The People's Democratic Republic of Algeria

University of Blida 1

Institute of Aeronautics and Space Studies

Attitude Stabilization of a CubeSat Using Fuzzy Logic Controller

---

A Thesis Submitted for the Degree of

Master of CNS/ATM

Authors:
Zakaria Hawasli
Abdel Wahab Mehidid

Supervisor:
Pr. Rahmouni Mohamed

Air Navigation Department

2025

Acknowledgements

This thesis would not have been possible without the help of many. Above all, it is an opportunity to express our gratitude to Allah, the Almighty, for granting us the courage and willpower to accomplish this work. As we close this important chapter of our academic journey and prepare to embrace new opportunities, we feel the need to express our heartfelt gratitude to those who made this endeavor possible. This engineering thesis could not have been achieved without the unwavering support, kind encouragement, and unconditional affection of the remarkable individuals who accompanied us along the way. Their wise advice, reassuring presence and enduring confidence have been essential pillars of both our academic and personal growth. Would also like to thank our parents for their unwavering support and constant encouragement throughout our academic journey; this success is undoubtedly the result of their many years of sacrifice and dedication to our education. We are especially grateful to our supervisor, Dr. M. Rahmouni and , for his guidance, attention to detail, and constant support throughout our final project,and We warmly thank the jury members for agreeing to review our work and provide constructive feedback. Finally, we extend our sincere thanks to all those near or far who played a role in making this experience meaningful, formative and unforgettable.

الملخص

في مدار أرضي منخفض. بعد عملية الإطلاق، 2U بحجم CubeSat تركز هذه الأطروحة على تثبيت وضعية قمر صناعي من نوع B-dot يعمل بواسطة يتعرض القمر الصناعي لحركة دورانية غير مسيطر عليها نتيجة لقوى الانفصال. ولمعالجة ذلك، تم استخدام متحكم وأكدت النتائج فعالية هذه الطريقة في تقليل .(detumbling) خلال مرحلة إزالة الدوران السريع (magnetorquers) ملفات مغناطيسية .السرعات الزاوية وتهيئة القمر الصناعي لمرحلة التحكم الدقيق

وقد .(FLC) ، ومتحكم المنطق الضبابيPIDالمتحكم الكلاسيكي :بعد مرحلة إزالة الدوران، تم مقارنة إستراتيجيتين للتحكم في الوضعية يقدم أداءً مقبولاً، لكنه يعاني من تجاوز القيم المطلوبة ووقت استقرار طويل. في المقابل، أظهر PID أظهرت نتائج المحاكاة أن المتحكم .متحكم المنطق الضبابي استجابة أسرع، وتجاوزاً أقل، وقدرة أكبر على مقاومة الاضطرابات والتعامل مع الديناميكيات غير الخطية

تُبرز هذه الدراسة الإمكانيات الواعدة للتحكم القائم على المنطق الضبابي في الأقمار الصناعية الصغيرة، حيث يقدم حلاً

Résumé

Ce mémoire traite de la problématique de la stabilisation d'attitude d'un CubeSat 2U en orbite terrestre basse. Après sa mise en orbite, le satellite est sujet à des mouvements de rotation incontrôlés (tumbling) causés par les forces de séparation. Pour résoudre ce problème, une première phase de détumbling est mise en œuvre à l'aide d'un contrôleur B-dot basé sur des bobines magnétiques (magnetorquers). Cette stratégie permet de réduire efficacement les vitesses angulaires du CubeSat.

Une fois cette phase terminée, deux méthodes de contrôle ont été testées pour la stabilisation fine : un contrôleur classique PID et un contrôleur basé sur la logique floue (Fuzzy Logic Controller). Les résultats de simulation ont montré que le contrôleur PID offre des performances correctes mais souffre de temps de stabilisation longs et d'un dépassement important. En revanche, le contrôleur flou présente une meilleure rapidité, une précision accrue et une robustesse supérieure face aux perturbations externes.

Ce travail démontre donc que la logique floue est une solution prometteuse pour le contrôle d'attitude des petits satellites, alliant précision, robustesse et faible consommation

Abstract

This thesis focuses on the attitude stabilization of a 2U CubeSat in low Earth orbit. After deployment, the satellite undergoes uncontrolled tumbling due to separation forces. To mitigate this, a B-dot controller using magnetorquers is applied during the detumbling phase. The results confirm the effectiveness of this approach in reducing angular velocities and preparing the satellite for precise control.

Following detumbling, two attitude control strategies are compared: a classical PID controller and a Fuzzy Logic Controller (FLC). Simulation results show that while the PID controller offers acceptable performance, it is limited by overshoot and long settling times. In contrast, the FLC demonstrates faster response, lower overshoot, and higher robustness against disturbances and nonlinear dynamics.

This work highlights the potential of fuzzy logic-based control for small satellites, providing a reliable, energy-efficient, and adaptive solution for precise attitude regulation.

Contents

List of Figures

List of Tables

List of Abbreviations and Symbols

ADCS Attitude Determination and Control System

B-dot Magnetic detumbling control law

CAD Computer-Aided Design

ECI Earth-Centered Inertial frame

FLC Fuzzy Logic Controller

IMU Inertial Measurement Unit

LEO Low Earth Orbit

LQR Linear Quadratic Regulator

PID Proportional–Integral–Derivative

RW Reaction Wheel

RCS Reaction Control System

U Unit (10 cm CubeSat unit)

WMM World Magnetic Model

Symbols

ϕ, θ, ψ	Euler angles (roll, pitch, yaw)
p, q, r	Angular velocities (rad/s) in body frame
I_x, I_y, I_z	Moments of inertia along x, y, z axes
K_p, K_i, K_d	Proportional, integral and derivative gains of PID
$u(t)$	Control signal
$e(t)$	Error signal
$\omega_x, \omega_y, \omega_z$	Angular velocity components
T_d	Delay time
T_r	Rise time
T_p	Peak time
T_s	Settling time
PO	Peak overshoot (%)
SSE	Steady State Error

Introduction

In this report, we explore the design, modeling, and simulation of an attitude stabilization system for a 2U CubeSat operating in low Earth orbit (LEO). The intention is that the satellite be oriented in the correct direction in space such that it is able to satisfy its mission requirements, e.g., precise pointing for communication or observation purposes. Because of the inherent constraint of CubeSats like limited volume, power, and processing capability, stable and accurate attitude control is a serious challenge.

The project has been divided into three phases. The first one is detumbling, the process of stopping the satellite's free spin as soon as it is deployed from the launch vehicle. This is done with magnetorquers and a B-dot controller, which slows down angular rates by working on the Earth's magnetic field.

After the CubeSat is adequately stabilized, phase two starts: fine attitude control. Within this stage, two control strategies are used, which are compared, i.e., the traditional Proportional Integral Derivative (PID) controller and the Fuzzy Logic Controller (FLC). The two methods are both applied under the same initial conditions to test their capacity to accurately stabilize the orientation of the satellite along the three axes of rotation (roll, pitch, and yaw).

Simulation is done using MATLAB and Simulink. The settling time, rise time, overshoot, and steady-state error are the main performance parameters that are obtained from the outputs to compare and analyze the controllers. The objective is to determine which controller provides enhanced stability, accuracy, and robustness in the application of CubeSat.

The paper concludes with an explicit presentation of simulation results and outlining the advantages of intelligent control methods, namely fuzzy logic in coping with nonlinear dynamics and space environment uncertainties. Outcomes of this research contribute to the corpus of knowledge directed towards improving autonomous attitude control systems for small satellite missions.

Chapter 1

Overview of CubeSat Development and Applications

1.1 History of CubeSat

1.1.1 Origins and Motivation

The CubeSat concept was conceived in 1999 by Professors Jordi Puig-Suari of California Polytechnic State University (Cal Poly) and Bob Twiggs of Stanford University. Their primary goal was to create a standardized, cost-effective platform that would allow graduate students to design, build, test, and operate satellites within the time and financial constraints of an academic program [1]. This initiative aimed to democratize access to space and provide hands-on experience in satellite development [2].

1.1.2 Design and Standardization

The initial design was inspired by a 10 cm $\times$ 10 cm $\times$ 10 cm plastic cube, commonly used for displaying collectibles. This cube became the standard unit, or "1U," for CubeSats. The modular design allowed for scalability, enabling configurations such as 1.5U, 2U, 3U, and larger. The standardization facilitated the development of the Poly-Picosatellite Orbital Deployer (P-POD), a deployment system designed to safely and efficiently release CubeSats into orbit [3].

1.1.3 First Launches

The first CubeSats were launched on June 30, 2003, aboard a Russian Rokot launch vehicle [3–5]. This mission deployed six CubeSats as secondary payloads, marking the beginning of CubeSat missions in space [6]. These early missions demonstrated the viability of the CubeSat platform for space research and education.

1.1.4 Evolution and Impact

Since their inception, CubeSats have evolved from educational tools to platforms for scientific research, technology demonstration, and commercial applications. The standardization and reduced cost have enabled a wide range of institutions, including universities, research organizations, and private companies, to participate in space missions. As of

2023, over 2,300 CubeSats have been launched, contributing to advancements in Earth observation, communication, and deep space exploration [7].

1.2 CubeSat Definition and Architecture

CubeSats are a class of nanosatellites defined by a standardized form factor. The standard unit of a CubeSat, referred to as "1U", measures $10 \times 10 \times 10$ cm and has a mass typically not exceeding 1.33 kg [8, 9]. This modular structure allows for the construction of larger configurations such as 1.5U, 2U, 3U, and even 6U or 12U CubeSats depending on mission complexity.

Deployment of CubeSats is generally performed using systems such as the Poly-Picosatellite Orbital Deployer (P-POD), which can accommodate up to three 1U CubeSats. These dispensers have greatly simplified integration and allowed CubeSats to fly as secondary payloads on a variety of launch vehicles, reducing both cost and scheduling constraints [8].

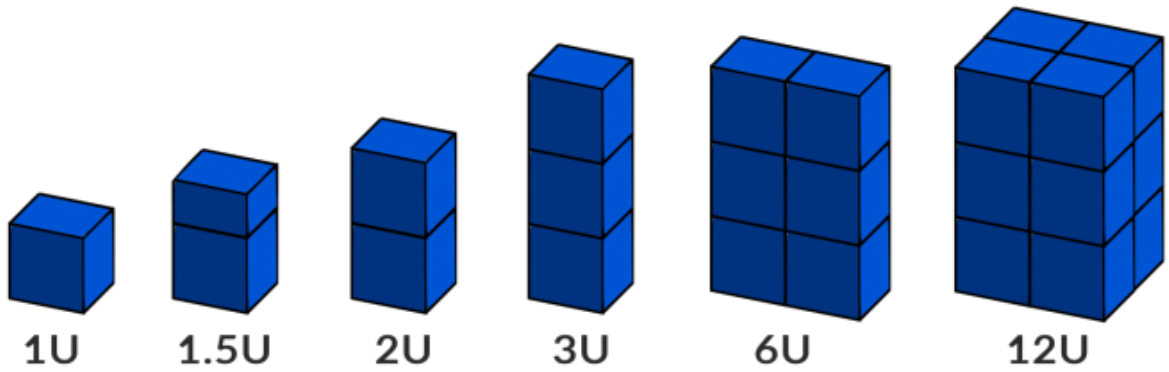

Figure 1.1: CubeSat Dimensions

1.2.1 Small Satellite Classification

The concept of "microsatellite" was first introduced by the AMSAT-NA community in the 1960s to describe communication spacecraft weighing less than 10 kg. At the time, such small platforms were seen as unconventional compared to traditional satellites, often facing skepticism within the space community regarding their usefulness and potential applications [10].

Martin Sweeting from Surrey Satellite Technology Ltd (SSTL) significantly advanced the field in the 1980s and early 1990s. In a 1991 publication, he highlighted that over 20 microsatellites had already been launched in the amateur radio service. He also introduced a pioneering classification system for small satellites based on mass, using metric prefixes (nano-, micro-, etc.) to define size categories by orders of magnitude [10]. This classification provided a practical framework directly linked to launch cost and mission constraints.

Table 1.1 illustrates the original mass-based classification introduced by Sweeting, while Table 1.2 shows the updated version, which includes newer categories such as picosatellites and femtosatellites. These updated definitions have been adopted by various

institutions, including at UNISPACE III in 1999, where small satellite classes were further refined.

Table 1.1: Initial classification of satellites by mass.

Class	**Mass Range**
Nanosatellite	< 10 kg
Microsatellite	10–100 kg
Minisatellite	100–500 kg
Small satellite	500–1000 kg
Large satellite	> 1000 kg

Table 1.2: Updated classification of small satellites.

Satellite Class	**Mass**
Large Satellite (e.g., observatory)	> 1000 kg
Minisatellite (Small Satellite class or LightSat)	100–1000 kg
Microsatellite	10–100 kg
Nanosatellite	1–10 kg
Picosatellite	0.1–1 kg
Femtosatellite (Satellite-on-a-chip)	1–100 g

The classification serves not only a descriptive role but also helps estimate cost and complexity. According to UNISPACE III, a typical minisatellite could cost between \$5–20 million, a microsatellite between \$2–5 million, and nanosatellites may be developed for less than \$1 million (based on 1999 prices) [10].

The adoption of small satellite missions has been fueled by various factors:

- Advances in electronics miniaturization and performance.
- The emergence of low-cost small launchers, sometimes adapted from military missile platforms.
- Opportunities for space access and autonomy for nations previously dependent on major space agencies.
- Reduction in mission complexity, project duration, and regulatory requirements.

Among the recognized benefits of small satellites are:

- Increased mission frequency and faster scientific return.
- Greater diversity in mission objectives and broader user access.
- Accelerated technical and scientific progress.
- Enhanced participation from smaller industries and academic institutions.

Early studies by organizations such as the International Academy of Aeronautics (IAA) further highlighted the strategic value of small satellites for research, education, and low-cost access to space [10].

1.2.2 Subsystems of CubeSats

Despite their small size, CubeSats include all essential satellite subsystems, compacted to fit within tight volume and power constraints. The Attitude Determination and Control System (ADCS) is responsible for maintaining or changing the spacecraft's orientation using sensors such as magnetometers, gyroscopes, and sun sensors, coupled with actuators like magnetorquers or reaction wheels. The Electrical Power System (EPS) governs energy harvesting from solar panels, battery storage, and power regulation for onboard electronics.

The Command and Data Handling System (CDHS), often integrated with the On-Board Computer (OBC), processes commands, collects telemetry, and interfaces with other subsystems. Telemetry, Tracking and Command (TT&C) systems provide communication links between the satellite and the ground station, enabling remote operation. Finally, the payload subsystem includes the mission-specific instruments, which can range from cameras to sensors or communication hardware [11, 12].

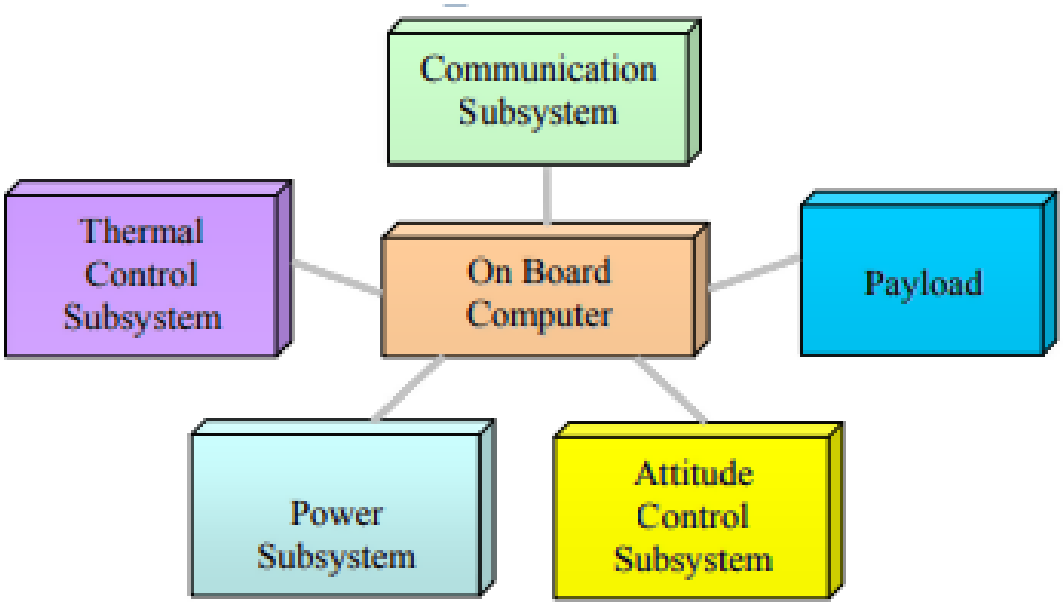

Figure 1.2: CubeSat Subsystems

1.2.3 Applications and Mission Domains

Initially intended for educational use, CubeSats have evolved into versatile platforms supporting a wide range of scientific, commercial, and governmental missions. Earth observation is one of the most prominent application areas, where CubeSats capture images and monitor environmental conditions. In scientific domains, CubeSats contribute to space weather studies, atmospheric analysis, and microgravity experiments.

Technology demonstration is another key domain, where CubeSats validate new systems such as miniaturized propulsion, onboard AI algorithms, and advanced sensors. CubeSats are also being used in deep space missions, such as NASA's MarCO CubeSats which accompanied the InSight mission to Mars. Moreover, their use in low-bandwidth communication applications like IoT relays and amateur radio continues to expand their relevance [8, 11].

According to Swartwout's statistical review, the first 100 CubeSats, launched between 2003 and 2012, involved over 20 countries and nearly 80 institutions, marking a strong

global interest in the CubeSat model. While early missions were largely academic, recent years have seen increasing adoption by industry and defense entities, leading to more reliable and complex mission outcomes.

1.3 Attitude Determination

Attitude determination refers to the process of estimating a satellite's orientation relative to a specified reference frame. For CubeSats, this typically involves determining the orientation with respect to either an inertial frame, such as the Earth-Centered Inertial (ECI) frame, or a local orbital frame. Accurate knowledge of attitude is crucial for mission operations, including camera pointing, antenna alignment, or optimizing solar panel exposure [11, 12].

While large spacecraft often rely on high-end, dedicated attitude hardware, CubeSats operate under strict constraints of size, weight, and power. As a result, they employ miniaturized and cost-effective sensors, relying on software-based estimation techniques to achieve reliable orientation information.

1.3.1 Reference Frames

Attitude is expressed as the orientation of the satellite's body frame (attached to the spacecraft) with respect to an external reference frame. Commonly used reference frames include:

- **Inertial Frame (ECI):** A non-rotating reference frame based on distant celestial objects (stars), used primarily for deep space and inertial navigation.
- **Body Frame:** The body frame is a coordinate system fixed to the CubeSat, with its axes aligned with the satellite's physical structure. It moves and rotates with the satellite, making it the natural reference for expressing sensor measurements, angular velocities, and control torques. All dynamic equations of motion are typically written in this frame [11].

1.3.2 Attitude Sensors for CubeSats

Due to limitations in volume and cost, CubeSats typically employ a combination of the following sensors to estimate attitude:

Gyroscopes: These measure angular velocity around the spacecraft's axes. While they offer high temporal resolution, their measurements drift over time due to sensor bias and noise.

Magnetometers: These sensors detect the local direction of the Earth's magnetic field. Although cost-effective and reliable, they are sensitive to magnetic interference caused by onboard electronics.

Sun Sensors: These detect the direction of sunlight and provide coarse attitude information during daytime orbital phases. However, they are ineffective during eclipse periods.

Star Trackers (optional): High-performance missions may use star trackers, which match observed star patterns to onboard catalogs. These provide precise orientation data but require significant computational power and energy.

Earth Horizon Sensors: These detect the Earth's limb or thermal gradient, offering accurate Earth-referenced attitude measurements. However, they are often too bulky or power-intensive for most CubeSat designs [11].

1.3.3 Sensor Fusion and Estimation Algorithms

As no single sensor can reliably provide full, drift-free attitude data, sensor fusion algorithms are employed to combine measurements from multiple sources. Common methods include:

Complementary Filters: These are simple real-time filters that blend fast-drifting measurements (e.g., gyroscopes) with slower, stable signals (e.g., magnetometers or sun sensors).

Extended Kalman Filter (EKF): Widely used in satellite attitude estimation, EKF offers statistically optimal results by accounting for nonlinearities and sensor noise.

QUEST Algorithm (QUaternion ESTimator): This algorithm computes the optimal quaternion from multiple vector observations. It is particularly effective in low-resource CubeSat systems.

Multiplicative Extended Kalman Filter (MEKF): A variant of EKF tailored for quaternion-based estimation, improving numerical stability and avoiding normalization drift [12].

1.3.4 Attitude Representations

Different mathematical representations can be used to express satellite orientation:

Euler Angles: These describe roll, pitch, and yaw angles but are subject to singularities (gimbal lock) when angles approach certain values.

Quaternions: A four-element representation that avoids singularities and provides a compact, numerically stable solution—making it ideal for onboard computation in CubeSats.

Direction Cosine Matrix (DCM): A 3×3 rotation matrix that accurately represents attitude but is computationally more intensive and redundant [11].

1.3.5 Attitude Accuracy Requirements

The level of accuracy required for attitude determination depends on mission objectives. High-resolution imaging missions demand sub-degree to arcsecond-level pointing accuracy. In contrast, basic communication tasks may only require a few degrees of accuracy. For passively stabilized CubeSats, real-time attitude determination may not be required, and approximate attitude may suffice for operations.

Through a combination of well-chosen sensors and estimation techniques, CubeSats can meet the demands of a wide range of missions, even within severe resource constraints [8].

1.4 Attitude Control Components

The performance of a CubeSat's Attitude Determination and Control System (ADCS) depends critically on the integration of various components that monitor, estimate, and actively correct the spacecraft's orientation in space. These components can be broadly grouped into sensors, actuators, onboard processing units, and supporting electronics

such as power management and structural integration. Together, they form a closed-loop system capable of autonomous attitude stabilization and control.

A. Sensors for Attitude Determination

Sensors are responsible for collecting data about the CubeSat's current orientation or angular velocity. They serve as the primary source of real-time input to estimation algorithms. Among the most commonly used sensors in CubeSats are gyroscopes, which measure angular velocity; magnetometers, which detect the Earth's magnetic field vector; and sun sensors, which estimate the direction of incoming sunlight. Advanced missions may also employ star trackers that provide precise celestial orientation by comparing observed star fields with onboard catalogs. Accelerometers are sometimes included to measure gravitational acceleration and support gravity-aligned estimations. These diverse sources of data are typically fused using estimation algorithms to produce a robust and drift-compensated attitude estimate.

B. Actuators for Attitude Control

Actuators are responsible for executing physical adjustments to the satellite's orientation. They generate torque based on control commands received from the onboard computer. The most commonly employed actuators in CubeSat platforms include magnetorquers and reaction wheels. Magnetorquers function by creating a magnetic moment through electric coils or rods, which interacts with the Earth's magnetic field to produce control torque. They are favored for their simplicity, low power requirements, and absence of moving parts, though they are less effective in high altitudes or deep space environments. Reaction wheels provide higher precision by modulating their rotational speed to generate torque through the conservation of angular momentum. They offer fine control but may reach saturation and require desaturation via magnetorquers or thrusters. Though rare in CubeSats due to volume and mass constraints, thrusters may also be used in missions requiring rapid detumbling or aggressive maneuvers.

C. Onboard Computer and Microcontroller

The onboard processing unit forms the computational core of the ADCS. It receives data from the sensors, performs estimation using algorithms such as Kalman filters or quaternion solvers, and calculates the required control signals to be sent to the actuators. In fuzzy logic-based architectures, the controller executes real-time inference using a set of heuristic rules and performs defuzzification to determine actuator commands. These computers must be lightweight, power-efficient, and capable of reliable real-time processing. They typically interface with subsystems using standard protocols such as I2C, SPI, or UART.

D. Control Algorithms and Software

The ADCS software implements the logic that governs the satellite's attitude behavior. This includes estimation filters, sensor fusion routines, and the control algorithms themselves. Depending on mission requirements, controllers may be based on classical approaches such as PID or LQR, or may leverage nonlinear and adaptive strategies like

fuzzy logic. The software also includes safety routines to handle anomalies, actuator saturation, or loss of sensor data. For CubeSats with advanced autonomy, these algorithms may be supplemented by fault-tolerant mechanisms or reconfiguration logic.

E. Power Management

Since attitude control systems can be power-intensive, particularly reaction wheels and processors running complex control laws, the ADCS is integrated with the satellite's power management unit. This system ensures that all critical components receive stable voltage and current, prioritizes loads during power deficits, and contributes to overall system health monitoring. Efficient power budgeting is especially crucial during eclipse periods or in failure recovery modes.

F. Structural Integration

The physical placement and alignment of ADCS components within the CubeSat directly influence performance. Sensors must be mounted in locations free from magnetic and electrical interference, while actuators such as reaction wheels must be carefully balanced to avoid introducing unwanted vibrations. Magnetorquers are often aligned along the principal axes of the spacecraft to simplify control. The mechanical structure must also accommodate thermal dissipation and structural rigidity, both of which impact sensor accuracy and system reliability over the mission duration.
Proper integration of these components ensures that the CubeSat can maintain or correct its orientation with high reliability, even in the presence of external disturbances or sensor noise. Together, they enable the spacecraft to fulfill mission objectives such as Earth observation, communication alignment, or payload pointing accuracy.

1.5 State of Art

1.5.1 Introduction

Small spacecraft platforms enable a variety of mission implementations, ranging from fully custom spacecraft to payload hosting services. These platforms support applications in commercial, academic, and government sectors and serve both Earth-orbit and deep space missions.

1.5.2 Hosted Orbital Services

Hosted orbital services refer to a model where mission payloads are integrated onto existing spacecraft platforms rather than requiring the development and deployment of a dedicated satellite. This approach offers significant advantages, including reduced cost, faster access to space, and improved reliability by leveraging proven hardware and infrastructure.

Hosted services can support a wide range of payloads, from physical sensors to software-only virtual payloads that run on shared onboard computing resources. Providers of hosted services may offer:

- Dedicated platforms for a single customer,

- Shared spacecraft with multiple payloads,
- Integration of hosted payloads on spacecraft already slated for launch.

Additionally, the concept of virtual payload hosting is emerging, where software applications run independently or alongside other functions on a spacecraft's flight computer. These services are particularly attractive to commercial and research organizations seeking rapid and cost-effective mission deployment.

1.5.3 Spacecraft Bus

The spacecraft bus is the core platform that provides all essential subsystems required to support and operate a mission payload in space. These subsystems typically include:

- Power generation and storage,
- Thermal control,
- Communications,
- Attitude Determination and Control Systems (ADCS),
- Propulsion (if applicable),
- Structural support.

Small spacecraft buses vary widely in form factor and capability, ranging from very small PocketQubes (typically 1P or 2P in size) to standard CubeSats (1U to 16U and larger), and extending up to ESPA-class platforms with payload capacities up to 180 kg or more. The spacecraft bus can be purchased at varying levels of integration, from a single hardware structure to complete mission-ready systems that include software, environmental testing, and ground support equipment. Choosing the right bus depends on mission requirements such as size, mass, complexity, and orbit. The 2024 report catalogs a diverse array of commercial bus providers offering platforms tailored to different mission profiles, from basic educational experiments to sophisticated science and defense missions. These offerings reflect the growing maturity of the small satellite industry and the trend toward rapid, modular, and scalable mission architectures.

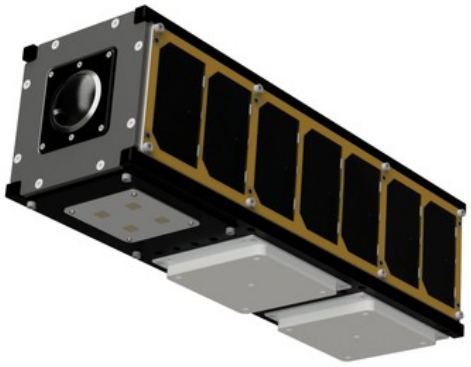

Figure 1.3: XO-BUS 3U Nanosatellite

1.5.4 Systems Engineering and Programmatic Considerations

The spacecraft bus is the core platform that provides all essential subsystems required to support and operate a mission payload in space. These subsystems typically include power generation and storage, thermal control, communications, attitude determination and control systems (ADCS), propulsion (if applicable), and structural support. Small spacecraft buses vary widely in form factor and capability, ranging from very small PocketQubes (typically 1P or 2P in size) to standard CubeSats (1U to 16U and larger), and extending up to ESPA-class platforms with payload capacities up to 180 kg or more. The spacecraft bus can be purchased at varying levels of integration, from a single hardware structure to complete mission-ready systems that include software, environmental testing, and ground support equipment. Choosing the right bus depends on mission requirements such as size, mass, complexity, and orbit. The 2024 report catalogs a diverse array of commercial bus providers offering platforms tailored to different mission profiles, from basic educational experiments to sophisticated science and defense missions. These offerings reflect the growing maturity of the small satellite industry and the trend toward rapid, modular, and scalable mission architectures.

1.5.5 Launch Integration Role

Launch integration has become increasingly complex and specialized, with launch brokers and integrators playing an essential role in matching SmallSats to available launch opportunities, managing logistics, documentation and mission assurance.

Launch Paradigms

- Dedicated Launches: Provide optimal mission flexibility but are generally more expensive.

- Rideshare Launches: Represent the majority of small spacecraft deployments offering cost-effective access to a variety of orbits though with less flexibility on timing and orbital parameters.

Deployment Methods

- CubeSat Dispensers:
  - Rail-type dispensers: The most common, supporting up to 27U (54 kg) configurations.
  - Tab-type dispensers: Less common but growing in use, offering alternative integration features.
  - Deployment considerations: Safety, compliance with CubeSat Design Specifications, and compatibility with launch vehicle-provided dispensers.
- Separation Systems for SmallSats: Include ESPA rings, SSMS, and multi-payload attachment fittings (MPAF), which support a variety of spacecraft sizes and missions.

Orbital Transfer Vehicles (OTVs) and Maneuvering Vehicles (OMVs)

OTVs and OMVs provide "last-mile delivery" services, moving satellites from initial deployment orbits to desired operational orbits. Several providers offer such services:

- Moments Vigoride: 800 kg payload, up to 2000 m/s V, for GEO and lunar transfers.
- Rocket Lab Photon: Flexible across LEO, MEO, GEO, interplanetary.
- Space Machines Company Optimus: 50 kg payload, 500 m/s V.
- UARX OSSIE: 200 kg payload, 690 m/s V.

International Space Station (ISS) Deployment Options

- Nanoracks NRCSD: Supports up to 12U CubeSats, deployed 400-420 km, 51.6 ° inclination, with up to 48U deployable per cycle.
- Nanoracks Kaber Deployer: Supports up to 24U, 82 kg.
- JAXA J-SSOD: Up to 6U CubeSats deployed from the Kibo Module.
- Bishop Airlock Module: The first commercial private airlock on the ISS, supports multiple deployments.

This integrated view of platform selection, integration, launch, and orbital transport reflects how spacecraft missions are no longer isolated technical efforts but part of a mature, interconnected delivery ecosystem. Understanding the interplay between spacecraft form, launch logistics, and in-orbit operations is now essential for modern mission success, from CubeSat tech demos to deep space exploration.

1.5.6 Energy

Power Generation

Power generation is a foundational component of the Electrical Power System (EPS) in small spacecraft, encompassing technologies such as photovoltaic (PV) cells, solar panels and arrays, and emerging energy sources like radioisotope thermoelectric generators (RTGs). For CubeSats and SmallSats, solar power remains the most widely used and practical energy source.

The majority of small satellites use solar arrays constructed from individual cells connected in series and parallel on substrates like PCB, CFRP, or aluminum honeycomb. These arrays can be body-mounted or deployable offering significantly more surface area and thus power generation capacity. Deployable arrays often dominate spacecraft structure due to their large size relative to the satellite, and their mechanical characteristics (like stiffness and mass distribution) significantly influence attitude control and propulsion demands.

Key types of solar cells include triple-junction (TJ) and ultra-triple junction (UTJ) cells. These cells reach efficiencies over 30% and are widely used in space applications. Products from manufacturers such as Spectrolab and Emcore are commonly integrated into smallsat missions.

To ensure consistent power delivery, especially in challenging orbital environments (e.g., long eclipse durations or varying sun angles in LEO), Electrical Power Systems (EPS) are often over-engineered. Key metrics for solar power systems include Beginning-of-Life (BOL) and End-of-Life (EOL) performance, thermal resilience, and specific power (W/kg), which influences overall spacecraft mass and performance. Recent studies reviewed more than 389 missions to analyze empirical solar array performance, showing a typical clustering around 30 W/kg, with operational designs targeting above 1 W/kg to ensure viability in space missions.

Specific power—the power output per unit mass—is a critical metric in EPS design. Current solar arrays for CubeSats typically deliver around 30 W/kg, though some advanced arrays (like ExoTerra's Fold-Out Solar Arrays) can reach up to 140 W/kg. Reducing array mass while maintaining or increasing power output is essential for deep space missions, where mass and storage volume are at a premium. Power generation challenges include:

- Deployability and structural integrity during launch and in orbit
- Thermal and mechanical stress resilience
- Efficient use of limited surface area
- Variability in sunlight exposure due to orbit and attitude

Future advancements are targeting mass reduction and higher efficiency. Concepts in development include tension based deployment systems, ultra-thin and flexible solar cells, and even multi-junction metamorphic solar cells with potential efficiencies exceeding 38%.

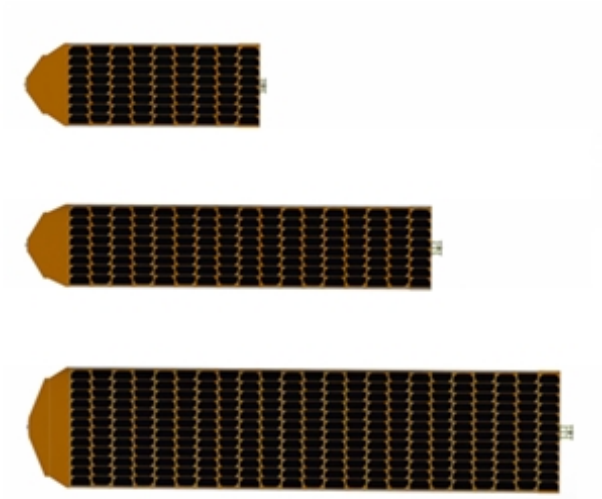

Figure 1.4: ExoTerra's Fold-Out Solar Arrays

Energy Storage

Energy storage in small spacecraft is primarily based on secondary lithium-ion (Li-ion) and lithium-polymer (LiPo) batteries, favored for their high energy density, recharge ability and spaceflight heritage. Typical Li-ion cells deliver an average voltage of 3.6 V and exhibit specific energy levels exceeding 150 Wh/kg, with top-performing cells such as the LG MJ1 reaching up to 260 Wh/kg and Panasonic's NCR18650B offering 243 Wh/kg. Battery packs used in CubeSats range widely: for example, EnduroSat's EPS IV provides 178.1 Wh/kg specific energy and 33.5 Ah capacity, while GomSpace's NanoPower BPX pack reaches 172 Wh/kg with 6.0 Ah capacity.
Cycle life is another key factor, with space-grade Li-ion batteries typically supporting 500–1,000 charge-discharge cycles, and mission durations ranging from 5 to 15 years depending on thermal, discharge, and depth-of-discharge profiles. Thermal management is crucial, as operating temperatures for Li-ion batteries generally range between -20°C and 60°C, and performance may degrade if exceeded.
Emerging alternatives include supercapacitors, which offer much higher power densities (3,000–40,000 W/kg) and extremely long lifetimes (up to 20 million cycles), but much lower energy densities of just 4–10 Wh/kg. Solid-state batteries, such as Solid Power's lithium-metal design, are also being developed, offering up to 440 Wh/kg and 930 Wh/L, with improved safety due to the absence of flammable liquid electrolytes. NASA's SABERS project is at the forefront of this effort

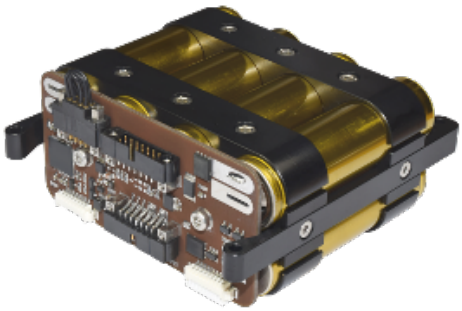

Figure 1.5: GomSpace's NanoPower BPX

In-Space Propulsion

In-space propulsion enables small spacecraft to perform critical maneuvers such as orbit adjustments, formation flying, and deorbiting. Among the most widely used systems is cold gas propulsion, offering simple, low-thrust operation with typical specific impulses (Isp) ranging from 50 to 70 seconds. Though limited in efficiency, cold gas thrusters remain popular for CubeSats due to their safety and ease of integration. Chemical propulsion systems provide significantly higher thrust and are suitable for fast orbital transfers, with green monopropellants like AF-M315E reaching specific impulses of 230–250 seconds, while being 50% denser and 45% higher in performance than hydrazine. These systems are more hazardous and often require more complex thermal and structural a,ccommodations. Electric propulsion (EP) technologies are increasingly favored due to their superior efficiency. Systems such as Hall-effect thrusters and ion engines can achieve specific impulses ranging from 1,500 to 3,500 seconds, albeit with lower thrust levels typically below 0.1 N. For example, Busek's Micro-Hall thruster operates at around 60 W and delivers thrust in the millinewton range. Electrospray thrusters, such as those developed by Accion Systems and MIT, offer precision control with specific impulses up to 1,000 seconds, ideal for fine maneuvers and deep-space missions. These can operate with power levels as low as 5–20 W and are scalable to fit 1U CubeSat volumes.
Emerging alternatives include resistojets (Isp 100–300 s) and pulsed plasma thrusters (PPTs), offering compact, simple configurations for missions requiring moderate V. Some experimental water-based steam propulsion systems (e.g., from Momentus and Tethers Unlimited) offer non-toxic, in-situ refillable options for long-duration LEO and lunar missions, achieving 100–150 s Isp with boil-based systems.
The trend in small spacecraft propulsion is toward miniaturization, modularity, and increased flight heritage. Over 50 in-space propulsion systems have been flown on CubeSats to date, with significant investment in reducing size, weight, and power (SWaP) while maximizing performance and mission versatility. As CubeSats aim for more ambitious missions—including interplanetary flight—efficiency, long life, and safety of propulsion systems become ever more critical Momentus and

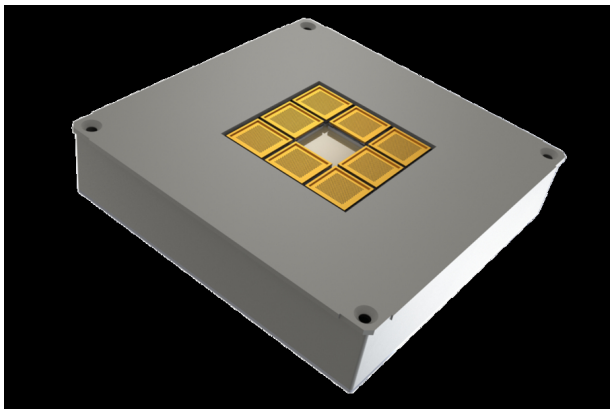

Figure 1.6: Electrospray thruster Accion Systems

1.5.7 Guidnance,Navigation Control

Introduction

The Guidance, Navigation, and Control (GNC) subsystem of small spacecraft, including CubeSats, comprises sensors, actuators, and processing units that determine the spacecraft's position, velocity, and attitude, and execute control maneuvers accordingly. Reaction wheels provide precision attitude control with typical performance ranging from 0.00023 to 0.3 Nm peak torque and 0.0005 to 8 N·m·s momentum storage, while magnetic torquers deliver control moments from 0.15 to 15 A·m^2, mainly for desaturation tasks. Star trackers offer high-accuracy attitude knowledge down to 8 arcseconds, complemented by sun sensors (0.1° accuracy) and Earth horizon sensors (0.25° accuracy). Inertial sensors, including MEMS gyroscopes with 0.15°/h bias stability and accelerometers with 3 g bias stability, are key for attitude propagation and maneuver termination.

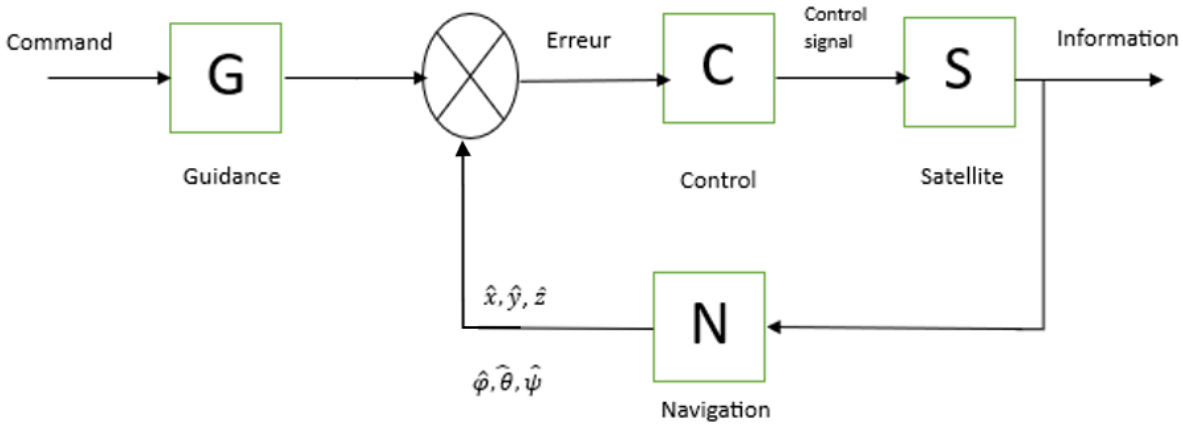

Figure 1.7: GNC Diagram

GNC Subsystems

Integrated Units

Integrated units in small spacecraft GNC systems represent a consolidated approach, where multiple attitude determination and control components such as reaction wheels,

magnetometers, magnetic torquers, sun sensors, GPS receivers, and star trackers are combined into a single, compact package. These systems come pre-integrated with onboard processors and embedded software capable of executing standard mission profiles like sun-pointing, inertial pointing, and Earth-target tracking. Inclusion of high TRL (Technology Ready Level 7-9) software significantly increases the probability of mission success.
Examples include Blue Canyon Technologies' XACT series, which has demonstrated 0.003° to 0.007° pointing accuracy in missions like MarCO and ASTERIA on 6U and 3U CubeSats, with unit masses ranging from 0.885 kg (XACT-15) to 1.813 kg (XACT-100). Similarly, CubeSpace's CubeADCS system, designed for 3U and 6U CubeSats, offers 70 arcseconds (3-sigma) pointing knowledge and weighs as little as 0.26 kg, depending on configuration. These integrated systems often support flexible combinations of actuators and sensors, including 3 to 4 reaction wheels, 3 magnetorquers, up to 4 fine sun sensors (FSS), and 2 Earth horizon sensors (EHS).
Other noteworthy solutions include AAC Clyde Space's iADCS-200 and iADCS-400, offering $<1°$ pointing accuracy with masses of 0.47 kg and 1.7 kg, respectively. Arcsec's Arcus ADC provides 0.1° pointing accuracy at 0.715 kg, integrating 3 reaction wheels, 3 magnetic torquers, 1 star tracker, 3 gyroscopes, 6 photodiodes, and 3 magnetometers. These integrated units provide spacecraft designers with simplified GNC architectures, reducing development risk, complexity, and integration effort, while enabling capabilities that support precise maneuvers, formation flying, and interplanetary navigation within the constraints of small satellite platforms.

Reaction Wheels

Miniaturized reaction wheels play a central role in enabling three-axis precision pointing capability for small spacecraft such as CubeSats and SmallSats. These wheels provide torque and momentum storage by spinning a rotor along a fixed axis, allowing the spacecraft to rotate in the opposite direction due to conservation of angular momentum. Key design considerations include the spacecraft's moment of inertia, required momentum storage capacity, and slew rate requirements. Reaction wheels can become saturated when reaching maximum speed, requiring desaturation using external torque devices like magnetic torquers or thrusters. In practice, three reaction wheels are standard for full three-axis stabilization, while four-wheel skewed configurations are preferred for fault tolerance, ensuring continued operation even after the failure of one wheel. Leading models include:

- AAC Clyde Space RW210, providing 0.0001 Nm peak torque, 0.006 Nms momentum capacity, at 0.8 W power, and 0.48 kg mass, with 36 krad radiation tolerance
- RW400, with 0.008 Nm torque, 0.050 Nms capacity, and 15 W power, at 0.375 kg mass.
- CubeSpace CubeWheel CW0017, delivering 0.23 mNm torque, 0.0017 Nms capacity, at 0.85 W power, and 0.06 kg mass, radiation tolerance of 24 krad.
- Larger models like CW5000 provide 37 mNm torque, 0.5 Nms capacit y, at 48 W power, with 1.084 kg mass.

Advanced wheels, such as the magnetically levitated reaction wheel developed by Celeroton AG, can operate at 30,000 rpm, storing 0.01 Nms momentum, while providing 0.01

Nm torque, aimed at minimizing jitter and mechanical wear.
Reaction wheels, while essential for precise pointing, introduce challenges such as jitter due to dynamic imbalance, electromagnetic interference that requires careful placement away from sensitive instruments like magnetometers, and reduced performance near zero-crossings of wheel speed. To mitigate such issues, some missions bias wheel speeds to half of their maximum, improving pointing stability but at the cost of halving usable momentum storage capacity.

Table 1.3: State-of-the-art reaction wheels for small spacecraft

Manufacturer	**Model**	**Peak Torque (Nm)**	**Momentum (N·s)**	**Power (W)**	**Mass (kg)**	**Rad Tol. (krad)**	**TRL**
AAC Clyde Space	RW210	0.0001	0.006	0.8	0.48	36	9
AAC Clyde Space	RW400	0.008	0.050	15	0.375	–	9
CubeSpace	CW0017	0.00023	0.0017	0.85	0.06	24	9
CubeSpace	CW5000	0.037	0.5	48	1.084	24	9
Celeroton AG (R&D)	Maglev RW	0.01	0.01	–	–	–	5–6

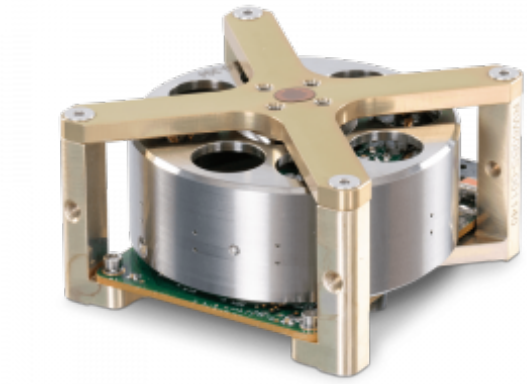

Figure 1.8: Rocket Lab's 60 mNms RW-0.06 Reaction Wheel

Magnetic Torquers

small spacecraft, providing torque by interacting with Earth's magnetic field. They function by generating a controlled magnetic dipole moment that interacts with the planet's local magnetic field, creating a torque perpendicular to the dipole and magnetic field vector. These devices are widely used for momentum dumping from reaction wheels, attitude control during detumbling, and in spin-up operations for spinner satellites. However, they cannot alone provide full three-axis control due to their inherent operational limitations—torque is only produced in the plane perpendicular to the magnetic field vector.
Two primary types of magnetic torquers exist: air coils and torquer bars/rods. Air coils are simpler but require higher power and larger coils to achieve the same dipole strength, while torquer bars benefit from ferrous cores, boosting the magnetic field strength without additional power but risking residual magnetic fields affecting sensitive instruments like

Figure 1.9: Rocket Lab TQ-40

magnetometers.
A range of high-heritage magnetic torquers are available commercially:

- AAC Clyde Space's MTQ800: 15 A·m² peak dipole, 3 W power, 0.395 kg mass.
- CubeSpace CubeTorquer CR0010: 1 A·m², 0.67 W power, 0.037 kg mass, 24 krad radiation tolerance.
- GomSpace NanoTorque GST-600: 0.31–0.34 A·m², 3-axis configuration, internal Z-axis versions available at 0.106 kg mass.
- NewSpace Systems NCTR-M016: 1.6 A·m², 1.2 W power, 0.053 kg mass.
- Rocket Lab TQ-40: The most powerful in the category, delivering 48 A·m², with 0.825 kg mass.
- ZARM Technik offers over 200 customized models, with units such as MT10-2, delivering 10 A·m² at 0.37–0.48 kg mass, with 0.7–1 W power consumption.

Despite their wide use in Low-Earth Orbit (LEO), magnetic torquers are ineffective beyond Earth's magnetic influence, such as in interplanetary missions, where alternative actuators like thrusters or reaction wheels are necessary. Typically, spacecraft use three magnetorquers mounted on orthogonal axes, with no redundancy needed due to their inherent electromechanical simplicity. Advanced applications continue to exploit magnetorquers for attitude control and reaction wheel desaturation in small spacecraft ranging from 1U CubeSats to ESPA-class platforms.

Table 1.4: State-of-the-art magnetic torquers for small spacecraft

Manufacturer	**Model**	**Dipole (A·m^2)**	**Power (W)**	**Mass (kg)**	**Rad Tol. (krad)**	**TRL**
AAC Clyde Space	MTQ800	15	3	0.395	–	9
CubeSpace	CR0010	1	0.67	0.037	24	9
GomSpace	GST-600	0.31–0.34	–	0.106	–	9
NewSpace Systems	NCTR-M016	1.6	1.2	0.053	–	9
Rocket Lab	TQ-40	48	–	0.825	–	9
ZARM Technik	MT10-2	10	0.7–1	0.37–0.48	–	9

Magnetometers

Magnetometers are crucial components in small spacecraft for measuring the local magnetic field, which aids in estimating two-axis attitude information when paired with an accurate Earth magnetic field model. Advanced algorithms, such as Kalman filters, can extrapolate this to provide full-state attitude and rate knowledge using magnetometer data alone, though more typically they are used in combination with other sensors like sun sensors through methods such as TRIAD and QUEST algorithms to achieve real-time 3-axis attitude information.
Additionally, magnetometers are vital in calculating the dipole moments necessary for magnetic torquers to generate the desired external torques on the spacecraft. However, they cannot differentiate between various magnetic field sources; therefore, strategic placement is essential to minimize interference from reaction wheels, torquers, power systems, harnesses, and even solar panels. To mitigate such interference, it is common to mount magnetometers on booms, exploiting the inverse-cube decay of magnetic fields with distance.
A variety of three-axis magnetometers are available for CubeSats and SmallSats, with performance metrics shown in Table 5-6 of the report. Some examples include:

- AAC Clyde Space MM200: 0.012 kg mass, 0.01 W power, 1.18 nT resolution, 30 krad radiation tolerance
- CubeSpace CubeMag Deployable: 0.016 kg mass, 0.23 W power, 13 nT resolution, 0.6° orthogonality error, 24 krad tolerance
- GomSpace NanoSense M315: 0.008 kg, specs not disclosed.
- NewSpace Systems NMRM-Bn25o485: 0.085 kg, 0.75 W, 8 nT resolution, 1° orthogonality, 10 krad tolerance.
- ZARM Technik FGM-A-75 (Fluxgate): 0.33 kg, 0.75 W, ±75,000 nT range, 1° orthogonality, 50 krad radiation tolerance.
- ZARM Digital AMR Magnetometer AMR-D-100-EFRS485: 0.18 kg, 0.3 W, ±100,000 nT range, 1° orthogonality, radiation tolerance unspecified.

Given the susceptibility of magnetometers to onboard-generated fields and environmental variations, on-orbit recalibration is often necessary, using spacecraft position and preferably attitude data.

Table 1.5: State-of-the-art magnetometers for small spacecraft

Manufacturer	**Model**	**Mass (kg)**	**Power (W)**	**Resolution (nT)**	**Orthogonality Error (°)**	**Rad Tol. (krad)**	**Type**
AAC Clyde Space	MM200	0.012	0.01	1.18	Not specified	30	AMR
CubeSpace Satellite Systems	CubeMag Deployable	0.016	0.23	13	0.6	24	AMR
GomSpace	NanoSense M315	0.008	Not disclosed	Not disclosed	Not disclosed	Not disclosed	AMR
NewSpace Systems	NMRM-Bn25o485	0.085	0.75	8	1	10	AMR
ZARM Technik	FGM-A-75	0.33	0.75	±75,000 nT range	1	50	Fluxgate
ZARM Technik	AMR-D-100-EFRS485	0.18	0.3	±100,000 nT range	1	Not specified	AMR

Figure 1.10: ZARM Digital AMR Magnetometer AMR-D-100-EFRS485

1.5.8 Sensors

Sun Sensors

Sun sensors are widely utilized in small spacecraft to determine the direction of the Sun relative to the spacecraft body frame. They serve both as an input to attitude determination algorithms and as a failsafe recovery mode sensor. However, due to their inability to independently provide a complete three-axis attitude estimate, they are typically used in conjunction with other sensors such as Earth sensors or star trackers.
To mitigate albedo effects from Earth, Moon, or reflections from spacecraft surfaces, some designs employ thresholds or filters to improve measurement integrity. However, these techniques can limit the sensor's effective field of view (FOV).
There are several types of sun sensors:

- **Cosine detectors:** Simple, low-cost, and lightweight (mass < 0.01 kg), but with limited accuracy (approximately 2° to 5°) and high sensitivity to stray light and albedo.
- **Quadrant detectors:** Use arrays of photodiodes and mathematical processing of current differentials to achieve improved accuracy (about 0.5° to 1°).
- **Digital sun sensors:** Employ coded masks and multiple photodiodes to generate digital outputs, offering higher accuracy (down to $\pm 0.125^\circ$) and typical FOVs such as $\pm 32^\circ$ or $\pm 64^\circ$.
- **Sun cameras:** Provide the highest precision (better than $\pm 0.01^\circ$) by imaging the Sun directly and processing its centroid, often with built-in albedo rejection features.

Commercial examples include:

- **AAC Clyde Space SS200:** 0.003 kg, 0.04 W, 110° FOV, $< 1^\circ$ accuracy, TRL 7–9.
- **Fine Pointing Sun Sensor:** 0.95 kg sensor, 1.08 W electronics, $\pm 4.25^\circ \times \pm 4.25^\circ$ FOV, accuracy better than $\pm 0.01^\circ$, TRL 7–9.
- **Solar MEMS nanoSSOC-D60:** 0.007 kg, 0.076 W, $\pm 60^\circ$ per axis, 0.5° accuracy, TRL 7–9.
- **NewSpace Systems NFSS-411:** 0.035 kg, 0.15 W, 140° FOV, 0.1° accuracy, TRL 7–9.
- **Needronix Eagle Point Fine Sun Sensor:** 0.006 kg, < 0.015 W, 11° FOV, $< \pm 0.01^\circ$ accuracy, TRL 5–6. This sensor offers advanced performance with integrated gyro and radiation rejection capabilities.

These sensors are indispensable for safe mode recovery, sun acquisition maneuvers, and attitude determination in both LEO and interplanetary environments. Current trends emphasize higher precision, miniaturization, and improved resistance to stray light and albedo contamination.

Figure 1.11: NewSpace Systems NFSS-411

Horizon Sensors

Horizon sensors, also known as Earth sensors, are used to determine spacecraft orientation by detecting the Earth's visible or infrared (IR) horizon. These sensors function by identifying the edge between space and the planet's surface and are particularly useful for nadir vector estimation and attitude determination, especially in low Earth orbit (LEO). There are two primary classes of horizon sensors: simple Horizon Crossing Indicators (HCIs) and more advanced thermopile sensors. HCIs detect the horizon crossing using basic infrared detection, while thermopile models can measure temperature gradients between the equator and poles, enhancing sensing accuracy. Horizon sensors may be static or employ scanning mechanisms. In many cases, sensor data is processed across multiple orbital passes to refine nadir vector estimates and improve control system performance. Commercially available models include:

- **CubeSpace CubeSense:** An infrared camera with a mass of 0.018 kg, power consumption of 0.28 W, and an accuracy of 1°.
- **Servo Mini Digital HCI:** A pyroelectric-based sensor weighing 0.050 kg, offering 0.75° accuracy; power consumption is unspecified.
- **RH 310 HCI:** A high-precision pyroelectric horizon sensor with 0.015° accuracy, 1.5 kg mass, and 1 W power usage.
- **Solar MEMS HSNS:** A compact digital infrared sensor weighing 0.120 kg, consuming 0.150 W, and providing 1° accuracy.

While traditional models rely on infrared sensing for Earth observation, similar technologies are being explored for use on planetary missions. Additionally, recent academic research focuses on developing miniaturized Earth horizon sensors optimized for CubeSats, aiming for higher precision and lower Size, Weight, and Power (SWaP) profiles.

Figure 1.12: Solar MEMS HSNS

Inertial Sensing

Inertial sensing in small spacecraft relies on gyroscopes for angular rate measurement and accelerometers for linear acceleration, often combined in packages ranging from single-axis devices to Inertial Reference Units (IRU) and Inertial Measurement Units (IMU) containing 3 gyros and 3 accelerometers. These sensors propagate vehicle state between updates from non-inertial sensors like star trackers, which typically provide attitude updates at a few Hertz. IMUs bridge these intervals, enabling continuous high-rate attitude knowledge necessary for precise spacecraft control.
State-of-the-art gyroscopes used include fiber optic gyros (FOGs), which offer superior performance at higher mass and cost, and MEMS gyroscopes, which are more compact and cost-effective but more susceptible to radiation-induced errors. Radiation-hardened MEMS gyros have recently become available, expanding their viability for more critical missions.
Key performance parameters are:

- Bias Stability (Gyros): as low as 0.06°/hr (RPU30 by MEMS) to 1,000°/hr (HG1700 by Honeywell)
- Angle Random Walk (ARW, Gyros): down to 0.006°/hr
- Bias Stability (Accelerometers): down to 3 µg
- Velocity Random Walk (VRW, Accelerometers): as low as 0.015 m/s/hr (STIM318, MEMS)

Examples include:

- Safran STIM318 IMU: 0.057 kg, 2.5 W, 0.3°/hr bias stability, 0.15 m/s/hr VRW, 3 axes each gyro and accelerometer.
- Honeywell HG4934SR MEMS IRU: 0.145 kg, <5.5 W, <3°/hr bias stability, <0.20 m/s/hr VRW.

- NewSpace Systems NSGY-001 IRU: 0.055 kg, 0.2 W, Image-based rotation estimation, no accelerometers included.
- VectorNav VN-100 IMU: 0.015 kg, 0.22 W, 10°/hr bias stability, 0.21 m/s/hr VRW.

For CubeSats and SmallSats, MEMS sensors dominate due to their small size, weight, and power (SWaP), though FOGs like the L3 CIRUS (15.4 kg, 0.0001°/hr bias stability) remain unmatched in precision for larger platforms.
These sensors are critical for attitude control systems and are often complemented by software to estimate and correct sensor biases and drifts over time, ensuring consistent performance even with MEMS limitations.

Figure 1.13: VectorNav VN-100 IMU

GPS Receivers

For low Earth orbit (LEO) spacecraft, GPS receivers have become the primary method for onboard orbit determination, effectively replacing traditional ground-based tracking methods. These receivers provide precise position, velocity, and time information by analyzing signals from at least four GPS satellites, where three signals determine position and the fourth corrects timing errors. GPS receivers can also support attitude estimation using phase differences from multiple antennas, but this technique remains less reliable and sensitive to various error sources.
State-of-the-art GPS receivers have advanced considerably, with compact, COTS chip-scale modules like the NovaTel OEM 719, now replacing older models such as the OEMV1. Their use is generally restricted to LEO; however, there have been successful experiments detecting GPS signals at geostationary orbits (GSO) and even prospects of extending GPS usability to cislunar distances as part of emerging technologies.
Examples of GPS receivers for small spacecraft include:

- AAC Clyde Space GNSS-701: 0.16 kg, accuracy <5 m, 10 krad radiation tolerance.
- APL Frontier Radio Lite: 0.4 kg, 1.4 W, 15 m accuracy, 20 krad.
- Aerospacelab GNSS-VSP: 0.394 kg, 2.4 W, 1.5 m (RMS).
- GomSpace GPS receiver: 0.285 kg, 3 W, <10 m accuracy, <20 krad.

- NovaTel OEM 719: 0.031 kg, 0.9 W, accuracy unspecified but widely used.
- SkyFox Labs piNAV-NG: 0.024 kg, 0.124 W, 10 m accuracy, 30 krad.
- Spacemanic Celeste_gnss_rx: 0.025 kg, ∼0.1 W, 1.5 m accuracy, 40 krad.
- Surrey Satellite Technology SGR-Ligo: 0.09 kg, 0.5 W, 5 m accuracy, 5 krad.

Despite their maturity and widespread use in LEO missions, GPS receivers are subject to export regulations (EAR) and COCOM limitations regarding altitude and velocity, requiring licenses for certain missions.

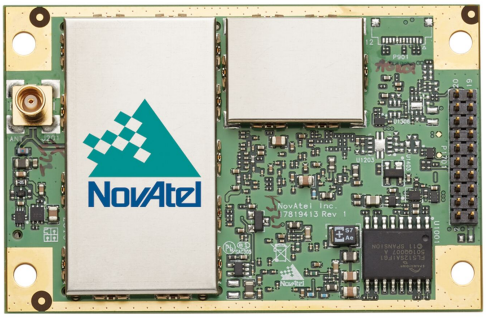

Figure 1.14: NovaTel OEM 719

Deep Space Navigation

In deep space, navigation is primarily performed through radio transponders in coordination with NASA's Deep Space Network (DSN). The DSN provides high-precision radiometric tracking capabilities critical for interplanetary and cislunar navigation. As of 2020, the Small Deep Space Transponder (SDST), developed by JPL and General Dynamics, was the only space-qualified transponder for small spacecraft with flight heritage. To advance capabilities for CubeSats and SmallSats, JPL developed the IRIS V2 transponder, a derivative of the Low Mass Radio Science Transponder (LMRST), explicitly designed for deep space CubeSat missions. The IRIS V2.1 has flown successfully on multiple missions, including:

- MarCO CubeSats (2018 Insight mission to Mars)
- LICIACube asteroid flyby (September 2022)
- CAPSTONE lunar orbiter (entered lunar orbit on November 13, 2022)
- Six Artemis I secondary CubeSat payloads, including Lunar Flashlight, LunaH-Map, ArgoMoon, CubeSat for Solar Particles, Biosentinel, and NEA Scout.

Additionally, IRIS Radio V3, the latest iteration, is capable of simultaneous multi-band operations across X, Ka, and S bands, allowing flexible communications for deep space missions.

These systems are pivotal in reducing dependence on Earth-based tracking and supporting fully autonomous deep space GNC systems, which are essential for upcoming CubeSat and SmallSat interplanetary missions. The shift towards autonomous navigation techniques, including optical navigation (planet, star, or pulsar-based methods), is

Table 1.6: State-of-the-art transponders for deep space navigation

Manufacturer	Model	Mass (kg)	RX Power (W)	Bands	Radiation Tolerance (krad)
General Dynamics	SDST	3.2	12.5	X, Ka	50
Space Dynamics Laboratory	IRIS V2.1	1.1	10.3	X, Ka, S, or UHF	25
Space Dynamics Laboratory	IRIS Radio V3	0.8	10	Simultaneous X, Ka, S	25

rapidly progressing to supplement or replace traditional DSN-based methods, particularly when communication latency and bandwidth constraints become critical at large distances.

Figure 1.15: Space Dynamics Laboratory IRIS Radio V3

Atomic Clocks

Atomic clocks have been historically used on large spacecraft in low-Earth orbit (LEO), but their integration into small spacecraft and CubeSats is a relatively recent advancement. Atomic clocks offer precise timekeeping, enabling more accurate spacecraft navigation, positioning, and timing (PNT). Traditional deep space navigation uses two-way tracking via ground-based antennas and atomic clocks, where the time difference between signal transmission and reception is used to calculate location and velocity. However, this method is inefficient for real-time onboard navigation, as the spacecraft must wait for Earth-based instructions, and each ground station can only track one spacecraft at a time.
In deep space, with large distances involved, timing precision within nanoseconds is critical for accurate positioning. This drives the development of miniaturized, stable, and radiation-hardened atomic clocks and oscillators that can be integrated into power- and volume-constrained small spacecraft, enabling autonomous navigation and reducing reliance on ground-based systems.

These emerging technologies, such as the Microsemi CSAC, represent key innovations in chip-scale atomic clocks (CSAC), providing compact, low-power options suitable for CubeSats with 0.035 kg mass and only 0.12 W power consumption, at TRL 5-6.
The ongoing development of Deep Space Atomic Clocks (DSAC) and commercial adaptations of miniature, space-qualified oscillators continue to push autonomous navigation and deep space exploration capabilities for small spacecraft, reducing dependence on Earth-based timing infrastructure.

Table 1.7: Selected Commercial Atomic Clocks and Oscillators for Small Spacecraft

Manufacturer	Model	Dimensions (mm)	Mass (kg)	Power (W)	Freq. Range (MHz)	Rad Tol. (krad)	TRL
AccuBeat	Ultra Stable Oscillator	131×120×105	2.0	6.5	57.51852	50	7–9
Bliley Technologies	Iris Series 1"x1" OCXO for LEO	19×11×19	0.016	1.5	10–100	39	7–9
Bliley Technologies	Aether Series TCVCXO for LEO	21×14×8	–	0.056	10–150	37	–
Microsemi	Space Chip Scale Atomic Clock (CSAC)	41×36×12	0.035	0.12	10	20	5–6
Safran Timing Technologies	SA MO	44×54×57	0.22	3.5 Nom, 5.5 Max	10	100	–
Safran Timing Technologies	Space Qualified mRO-50	51×51×20	0.080	0.4 Nom	10	25 Min	–
Safran Timing Technologies	miniRAFS	108×53×68	0.45	<12 Max	60 and 10	–	–
Safran Timing Technologies	LNMO	50×50×30	0.1	1.5 Nom, 2.5 Max	5–40	100	–

Figure 1.16: Safran Timing Technologies Space Qualified mRO-50

LiDAR

LiDAR (Light Detection and Ranging) is an emerging sensor technology for small spacecraft, adapted from terrestrial and automotive sectors where it has become a mature and reliable sensing technology. In space applications, LiDAR is traditionally found on larger spacecraft, such as NASA's Orion capsule, to support proximity operations, rendezvous and docking, formation flying, and altimetry. The miniaturization of LiDAR systems now enables applications for CubeSats and SmallSats, including autonomous relative navigation and precision landing support, such as the Mars helicopter operations.
Despite its advantages, LiDAR integration into small spacecraft is still in its early stages, with relatively few flight-demonstrated systems. Key commercial systems include:

Due to its high data rates and precision, LiDAR is expected to become an essential component in small spacecraft GNC systems, especially as rendezvous and proximity operations (RPO) missions grow more common. Nevertheless, challenges remain regarding

Table 1.8: Selected Commercial LiDAR Systems for Small Spacecraft

Manufacturer	Model	Mass (kg)	Power (W)	Max Range (m)	Radiation Tolerance
ASC	GSFL-4K (3D)	3.0	30	>1000 (altimeter mode)	Unspecified
Garmin	Lidar Lite V3	0.022	0.7	40	Unspecified

SWaP constraints, radiation hardening, and long-range detection in space environments.

Figure 1.17: Garmin Lidar Lite V3

1.5.9 SmallSat Avionics and Communication Systems

Avionics

SmallSat Avionics (SSA) encompasses all electronic subsystems, components, instruments, and functions on the spacecraft platform, including Command and Data Handling (CDH), Flight Software (FSW), and Payload and Subsystems Avionics (PSA). The avionics system acts as the brain and nervous system, integrating command, control, communication, and data interfaces across subsystems. There is considerable variability in avionics architectures, influenced by two primary factors:

1. **Spacecraft scale:** Traditional spacecraft are high SWaP-C (Size, Weight, Power, and Cost) systems employing redundancy and high reliability. SmallSats are low SWaP-C systems, tolerating higher risk but offering lower cost and faster development cycles. Despite this, constellations of SmallSats can collectively match the functionality of larger spacecraft by distributing avionics across platforms.

2. **Architecture design:** Regardless of scale, architectures may be centralized or decentralized, single-string or redundant, modular or monolithic. SmallSats often adopt single-string architectures, where failure of one component may lead to full mission failure. To increase robustness, radiation-hardened (rad-hard) and radiation-tolerant designs are being increasingly incorporated, especially for LEO and deep space missions.

The chapter highlights that both commercial and academic/government institutions are developing avionics platforms:

- SpaceCube, MUSTANG (NASA GSFC)
- Sabertooth (JPL)
- CHREC/SHREC Space Processor (NSF SHREC)
- RadPC (Montana State University)

State-of-the-art SmallSat avionics are organized into:

- **Command and Data Handling (CDH)** including onboard computing, memory, interfaces, radiation tolerance, etc.
- **Flight Software (FSW)** covering operating systems, frameworks, development environments, and standards.

Trends in SmallSat avionics focus on:

- Highly integrated modular onboard computing systems, with options for system-on-chip (SoC) and FPGA-based solutions offering increased processing power, flexibility, and software-defined architectures.
- Modular architectures being federated (independent elements) or integrated (distributed functionality), increasingly supporting autonomous operations, intersatellite communication, and constellation coordination.
- Miniaturization enabling advanced computing, fault tolerance, and radiation mitigation within CubeSat form factors, with SWaP-C as a critical driver.
- Emerging open-source, COTS-based development platforms lowering entry barriers for new space entrants, despite the trade-offs in radiation tolerance and reliability.

Figure 1.18: Configurable and Autonomous Sensor Processing Research (CASPR)

Communication

In parallel, communication systems have evolved to support the increasing demand for higher data rates, longer-range links, and inter-satellite communications. Historically dominated by VHF and UHF bands (30 MHz to 1 GHz), there has been a significant shift towards higher frequency bands such as S-band (2–4 GHz), X-band (8–12 GHz), and Ka-band (27–40 GHz) to support data-intensive missions. Each band presents specific trade-offs in data rate, link margin, antenna size, atmospheric attenuation, and licensing complexity.

Key aspects include:

- S-band remains the most widely used for Telemetry, Tracking, and Command (TT&C) due to its balance of data rate and antenna size, with wide support from NASA's Near Space Network (NSN).
- X-band is gaining popularity for Earth observation and science payloads requiring higher downlink rates.
- Ka-band offers >100 Mbps downlink capability, demonstrated by missions such as ISARA (Ka-band reflectarray antenna CubeSat) and Landmapper-BC 3 v2, though it suffers from rain fade and demands highly accurate pointing.

System architecture typically comprises a radio transceiver, amplifier, and antenna, where radios modulate/demodulate signals, amplifiers boost RF output, and antennas transmit or receive signals. The advent of Software Defined Radios (SDR) allows flexible reconfiguration across bands and modes, supporting adaptive communications.

Other trends and developments:

- Emerging ISM band use (e.g., 2.4 GHz ZigBee on VELOX-I, WiFi on TechEdSat) for experimental downlinks and inter-satellite communications, showing innovative use of COTS wireless systems.
- Spread-spectrum techniques are increasingly used to mitigate interference, particularly in multi-spacecraft networks.
- Regulatory challenges remain critical, with CubeSats needing to secure FCC Part 5 (experimental) or Part 25 licenses, particularly for Earth observation or crosslink missions.
- Optical communications (Lasercom), although demonstrated, is still not yet widely adopted by CubeSats, mainly due to pointing challenges and susceptibility to weather conditions.
- CubeSats are increasingly leveraging commercial satellite networks (Iridium, Globalstar) in L-band, removing the need for dedicated ground stations but limited to below constellation altitudes.

Both avionics and communication systems are moving toward higher levels of integration, software reconfigurability, autonomy, and radiation tolerance, with an emphasis on distributed architectures, adaptive networking, and deep space communication capabilities, ensuring small spacecraft can support increasingly complex, data-intensive, and long-duration missions across LEO, GEO, and interplanetary environments.

Figure 1.19: Landmapper-BC 3 v2

Chapter 2

Attitude Stabilization

2.1 Introduction

Spacecraft attitude control plays a fundamental role in ensuring mission success. It allows the satellite to accurately orient scientific instruments or align thrusters for orbital maneuvers. In early space missions, engineers often relied on passive spin stabilization, a technique where the satellite spins around one of its axes, typically the one with the highest moment of inertia, to maintain a relatively fixed orientation in space. This approach was primarily chosen due to the limited availability of control actuators and the lack of onboard computing power needed to implement more advanced control laws. Although spin-stabilized spacecraft offer high stability, they require precise balancing. Every component must be carefully designed and placed to maintain symmetry and rotational equilibrium. Achieving this level of balance is challenging and often requires fine-tuning in the development process. In most cases, final adjustments are made after the flight hardware is assembled, during experimental spin testing on the ground. Although spin stabilization proved effective in early missions, it comes with significant limitations. The inability to freely reorient the spacecraft and the difficulty of accommodating multiple pointing requirements made it less suitable for more complex or dynamic missions. As space technologies evolved, especially with the advent of compact sensors, efficient actuators, and powerful onboard processors, active attitude control became the new standard. Active control systems provide greater flexibility and precision. They allow spacecraft to adjust their orientation in real time, track targets, and switch between different mission modes. This is particularly important for CubeSats and small satellites, which often operate in low Earth orbit and require responsive control for tasks like Earth observation, communication alignment, or scientific data collection. One of the first challenges facing a CubeSat after deployment is de-tumbling: the process of reducing the high angular velocities acquired during launch or ejection. To achieve this, a commonly used method is the B-dot control algorithm, which uses the rate of change of the Earth's magnetic field to counteract the satellite's rotation using magnetic torques. It is simple, energy efficient, and well suited for small satellites with limited resources. Once detumbling is complete and the satellite's rotation is within manageable limits, Reaction Wheel model can take over for fine attitude stabilization. These include classical controllers such as PID, optimal controllers like LQR, and intelligent methods such as Fuzzy Logic Controllers (FLC), which offer robustness in uncertain or non-linear environments. In this chapter, we will explore these methods, their working principles, and their role in achieving a precise and reliable CubeSat orientation in space.

2.2 Coordinates Systems

2.2.1 Earth-Centered Inertial Frame – ECI

This frame is a non-rotating reference frame and is assumed to be fixed in space. The origin of the ECI coordinate frame is located at the center of mass of the Earth with axes xi, yi and zi. The xi-axis points towards the northern hemisphere's vernal equinox, the yi axis is defined by the right-hand rule to complete a right-hand orthogonal frame, and the zi-axis is directed towards the celestial North Pole, along the Earth's rotation axis. As a result, the fundamental plane for the ECI is defined as the Earth's equatorial plan [13]

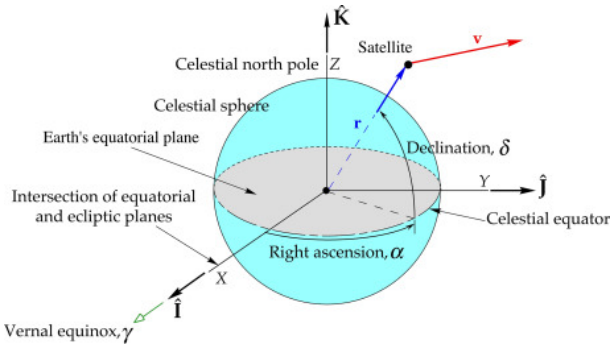

Figure 2.1: Earth Centred Inertial Frame

2.2.2 Body-Fixed Frame

The body-fixed reference frame is one of the most important coordinate systems in the attitude determination and control field. This system is fixed to the satellite body and thus moves and rotates with the satellite, and the origin is usually placed at the spacecraft's center of gravity with the body-axis chosen to coincide with the principal axes of inertia. Origin Body frame is defined by the zb-, the axis of minimum inertia; the positive zb is in the nominal nadir direction, which is the direction pointing towards the center of the Earth. The yb is the axis of maximum inertia; the positive yb is the nominal orbital anti-normal, which is a vector that is negatively perpendicular to the orbital plane. The Body frame is a right-hand orthogonal coordinate system such that xb is found by the right-hand rule, the positive xb is in the nominal velocity vector direction, which is the forward flight direction of the spacecraft [14].

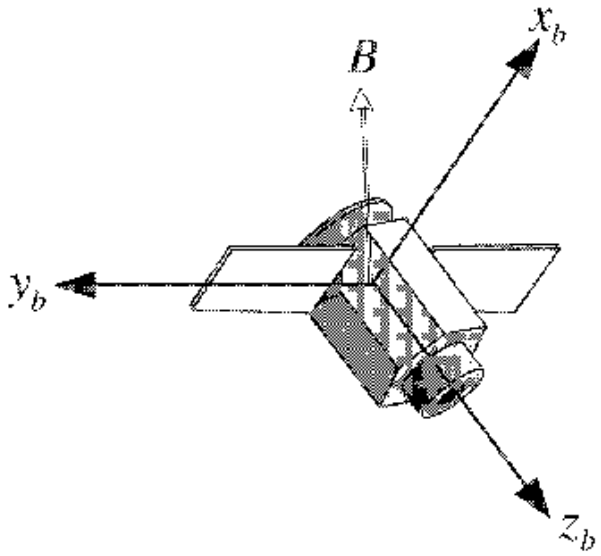

Figure 2.2: Satellite body coordinate axes

2.3 Particle Dynamics

2.3.1 System of Particles

Newton's Second Law governs the motion of a system of particles, where the sum of all external forces acting on a particle equals the time rate of change of its linear momentum [15–18]. For a particle j, this is expressed as:

$$\Sigma_{i=0}^{N} \vec{\mathbf{F_{ij}}} = \frac{d\vec{\mathbf{p_j}}}{dt} \tag{2.1}$$

Here, $\vec{j}$ is the momentum of particle j, and $\vec{(}Fij)$ represents the forces acting on it.

2.3.2 Rotational Dynamics for Systems of Particles

When considering rotational motion, the moment about the center of mass M_C is derived from the cross product of the position vector r_{Cj} and the rate of change of momentum:

$$\vec{M}_C = \sum_{j=0}^{P} \mathbf{S}(\vec{r}_{Cj}) \frac{d\vec{p}_j}{dt} \tag{2.2}$$

The skew-symmetric matrix $S(\vec{r}_{Cj})$ ensures the correct cross-product formulation.

2.4 Rigid Bodies

2.4.1 Translational Dynamics

For a rigid body, the translational equations of motion relate the total external force $\vec{F}_C$ to the acceleration of the center of mass, accounting for Coriolis effects due to rotation :

$$\vec{F}_C = m\left(\vec{v}_{C/I}t + \vec{\omega}_{B/I} \times \vec{v}_{C/I}\right) \tag{2.3}$$

where $\vec{v}_{C/I}$ is the velocity of the center of mass in the inertial frame, and $\vec{\omega}_{B/I}$ is the angular velocity of the body frame.

2.4.2 Rotational Dynamics

The rotational dynamics of a rigid body are governed by Euler's equation:

$$\vec{M}_C = I_C \vec{\omega}_{B/I} t + \vec{\omega}_{B/I} \times I_C \vec{\omega}_{B/I} \tag{2.4}$$

The term $S(\vec{\omega}_{B/I}) \times I_C \vec{\omega}_{B/I}$ represents gyroscopic effects due to rotation.

2.5 Aerospace Equations of Motion

2.5.1 Translational Equations of Motion

For aerospace vehicles, the translational acceleration is derived from the sum of gravitational forces and propulsion:

$$m\vec{v}_I = \vec{F}_I + \vec{F}_P \tag{2.5}$$

where $\vec{F}_I$ includes planetary gravitational forces, while $\vec{F}_P$ represents propulsion forces [19].

2.5.2 Attitude Equations of Motion

The attitude dynamics of a spacecraft are influenced by external moments and angular momentum conservation:

$$I\vec{\omega}_{B/I} + \vec{\omega}_{B/I} \times I\vec{\omega}_{B/I} = \vec{M}_P + \vec{M}_M + \vec{M}_R - \vec{H}_S \tag{2.6}$$

where:

- $\vec{M}_P$: Propulsion moments
- $\vec{M}_M$: Magnetorquer moments
- $\vec{M}_R$: Reaction wheel moments
- $\vec{H}_S$: Total angular momentum

2.6 Tools Used for Attitude Representation

Given the various reference frames, it is essential to use transformations to switch from one frame to another. Tracking the evolution of a frame attached to a rigid body is equivalent to tracking the orientation of the body. Studying the attitude of such a system thus involves analyzing the parameters of these transformations, i.e., the rotation matrices, which are representations of attitude. Each of these representations is associated with a rotation matrix. Several attitude representations exist, such as the elements of the direction cosine matrix, Euler angles, Cardan angles, and quaternions.

2.6.1 Direction Cosine Matrix

For a frame R_i with basis vectors $\vec{e}_x$, $\vec{e}_y$, $\vec{e}_z$ and another frame R_f with basis vectors $\vec{f}_x$, $\vec{f}_y$, $\vec{f}_z$, the direction cosine matrix is defined as:

$$R = \begin{bmatrix} \vec{e}_x \cdot \vec{f}_x & \vec{e}_x \cdot \vec{f}_y & \vec{e}_x \cdot \vec{f}_z \\ \vec{e}_y \cdot \vec{f}_x & \vec{e}_y \cdot \vec{f}_y & \vec{e}_y \cdot \vec{f}_z \\ \vec{e}_z \cdot \vec{f}_x & \vec{e}_z \cdot \vec{f}_y & \vec{e}_z \cdot \vec{f}_z \end{bmatrix} \tag{2.7}$$

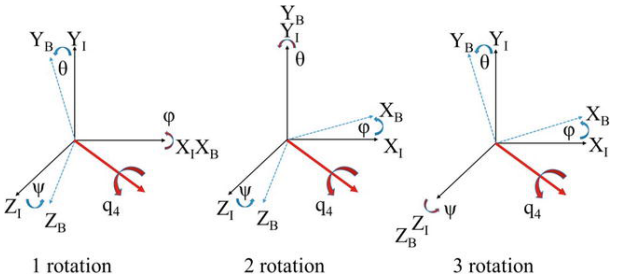

Figure 2.3: Kinematics: On Direction Cosine Matrices

2.6.2 Euler and Cardan Angles

Euler angles are three angles that describe the orientation of a rigid body. Generally, the rotation matrix between a frame R_r (Oxyz) and another frame R_v (Ox'y'z') can be written as the product of three rotation matrices. The first rotation, with angle ψ (precession), is around the Oz axis, transforming Oxyz to Ouvz. The second rotation, with angle θ (nutation), is around the Ou axis, transforming Ouvz to Ouwz. The third rotation, with angle ϕ, around the Oz' axis, transforms Ouwz to Ox'y'z'.

The rotation matrices R_x, R_y, and R_z around the Ox, Oy, and Oz axes, respectively, are:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}, \quad R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}, \quad R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.8}$$

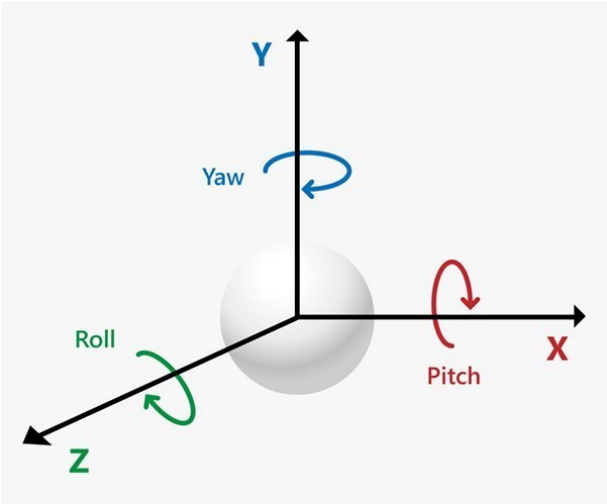

Figure 2.4: Euler Angles

Euler angles correspond to a 3-1-3 sequence (symmetric), while Cardan angles correspond to a 3-2-1 sequence (asymmetric). For controller design, it is more practical to work with variables close to zero (for system linearization). Here, the goal is to keep the satellite frame close to the reference frame, making Cardan angles preferable.

2.6.3 Cardan Angles

The rotation matrix $R_{R_r \to R_v}$ associated with Cardan angles between R_v and R_r is expressed as:

$$R_{R_r \to R_v} = R_z(\psi) R_y(\theta) R_x(\phi) \tag{2.9}$$

The angular velocity of the frame R_v relative to R_r, expressed in R_v, is related to the Cardan angles as follows:

$$M_{R_C \to R_{sat}} = \begin{pmatrix} 1 & -\psi & \theta \\ \psi & 1 & -\varphi \\ -\theta & \varphi & 1 \end{pmatrix} \tag{2.10}$$

- $\omega_x, \omega_y, \omega_z$ are the components of the angular velocity vector $\boldsymbol{\omega}$ in the body-fixed frame, aligned with the satellite's roll (x), pitch (y), and yaw (z) axes.
- ϕ is the roll angle (rotation about the x-axis).
- θ is the pitch angle (rotation about the y-axis).
- ψ is the yaw angle (rotation about the z-axis).
- $\dot{\phi}, \dot{\theta}, \dot{\psi}$ are the time derivatives of the Euler angles, representing the rate of change of roll, pitch, and yaw angles respectively.

In the context of attitude kinematics and rotation matrices based on Cardan angles, the following reference frames are defined:

- R_r: The **reference frame**, often representing a fixed frame such as the Earth-centered inertial frame or the orbital frame.
- R_v: The **vehicle frame**, typically a rotating frame attached to the satellite, representing the orientation of the body.
- R_c: The **Cardan frame** or an intermediate **control frame** used during the construction of the rotation matrix via Cardan angle sequences. It can serve as a transitional frame in the rotation sequence.
- R_{sat}: The **satellite's body frame**, fixed to the satellite structure, defined by the axes $\{x_b, y_b, z_b\}$.

For small-angle approximations, $R_{R_c \to R_{sat}}$ can be linearized as:

$$R_{R_c \to R_{sat}} \approx \begin{bmatrix} 1 & \psi & -\theta \\ -\psi & 1 & \phi \\ \theta & -\phi & 1 \end{bmatrix} \tag{2.11}$$

2.6.4 Quaternions

Quaternions are four-component vectors. Among these components, three correspond to a 3D vector ($\vec{q}_{1:3}$),and the fourth (q_4) is a scalar [20] . A quaternion q is written as:

$$q = \begin{bmatrix} \vec{q}_{1:3} \\ q_4 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{2.12}$$

Two types of quaternion multiplication exist for $\bar{q}$ and q:

$$\bar{q} \otimes q = \begin{bmatrix} q_4 & q_3 & -q_2 & q_1 \\ -q_3 & q_4 & q_1 & q_2 \\ q_2 & -q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} \begin{bmatrix} \bar{q}_1 \\ \bar{q}_2 \\ \bar{q}_3 \\ \bar{q}_4 \end{bmatrix}, \quad q \odot \bar{q} = \begin{bmatrix} \bar{q}_4 & -\bar{q}_3 & \bar{q}_2 & \bar{q}_1 \\ \bar{q}_3 & \bar{q}_4 & -\bar{q}_1 & \bar{q}_2 \\ -\bar{q}_2 & \bar{q}_1 & \bar{q}_4 & \bar{q}_3 \\ -\bar{q}_1 & -\bar{q}_2 & -\bar{q}_3 & \bar{q}_4 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{2.13}$$

These operations are non-commutative, and $\bar{q} \otimes q = q \odot \bar{q}$. Quaternions used for attitude representation must satisfy the normalization constraint $\|q\| = 1$. For a rotation of angle θ around a unit vector $\vec{e}$, the attitude quaternion is:

$$q = \begin{bmatrix} \vec{e}\sin(\theta/2) \\ \cos(\theta/2) \end{bmatrix} \tag{2.14}$$

This equation represents a unit quaternion that encodes a rotation of angle θ around a fixed unit axis $\vec{e}$, where $\|\vec{e}\| = 1$.

The kinematic relations for quaternions are:

$$q = \frac{1}{2} \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} q = \frac{1}{2}\Omega(\vec{\omega})q \tag{2.15}$$

where:

$$\Omega(\vec{\omega}) = \begin{bmatrix} -[\vec{\omega}\times] & \vec{\omega} \\ -\vec{\omega}^T & 0 \end{bmatrix}, \quad [\vec{\omega}\times] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{2.16}$$

Quaternions are the most widely used representation for attitude estimation and control problems due to their compact form and absence of singularities. However, in this report, Cardan angles are chosen for their simplicity and intuitive parameterization of the reference frame [21].

2.7 Euler's Rotational Equations of Motion

Consider a rigid body with a body-fixed reference frame B, whose origin lies at the center of mass. [21] The angular momentum vector $\vec{H}$ of the body about its center of mass is defined as:

$$\vec{H} = I\vec{\omega} \tag{2.17}$$

where:

- I is the inertia matrix,
- $\vec{\omega}$ is the angular velocity vector of the body relative to an inertial frame.

The general form of Euler's rotational equation for a rigid body is:

$$\vec{M} = \frac{d\vec{H}}{dt} \tag{2.18}$$

where $\vec{M}$ is the external torque vector. However, since $\vec{H}$ is expressed in a rotating (body-fixed) frame, its derivative in the inertial frame becomes:

$$\left(\frac{d\vec{H}}{dt}\right)_{\text{inertial}} = \left(\frac{d\vec{H}}{dt}\right)_{\text{body}} + \vec{\omega} \times \vec{H} \tag{2.19}$$

Substituting $\vec{H} = I\vec{\omega}$, we get the expanded form of Euler's equations:

$$\vec{M} = I\dot{\vec{\omega}} + \vec{\omega} \times (I\vec{\omega}) \tag{2.20}$$

For a rigid body whose body axes are aligned with its principal axes of inertia, the inertia matrix I becomes diagonal: $I = \text{diag}(I_1, I_2, I_3)$. Therefore, Euler's rotational equations simplify to the scalar form:

$$\begin{aligned} I_1\dot{\omega}_1 &= (I_2 - I_3)\omega_2\omega_3 + M_1 \\ I_2\dot{\omega}_2 &= (I_3 - I_1)\omega_3\omega_1 + M_2 \\ I_3\dot{\omega}_3 &= (I_1 - I_2)\omega_1\omega_2 + M_3 \end{aligned} \tag{2.21}$$

where:

- $\omega_1, \omega_2, \omega_3$ are the components of the angular velocity vector $\vec{\omega}$,
- M_1, M_2, M_3 are the components of the external torque vector $\vec{M}$.

These equations form a set of three coupled, nonlinear ordinary differential equations and are commonly known as Euler's rotational equations of motion. The nonlinear terms such as $(I_2 - I_3)\omega_2\omega_3$ are referred to as gyroscopic coupling terms.

2.8 Disturbance Torques

The total torque acting on a satellite body comprises multiple components, with the four primary external disturbance sources being gravity gradient effects, magnetic field interactions, aerodynamic forces, and solar radiation pressure. The magnitude of each disturbance torque and its relative dominance depends on several key factors including the spacecraft's orbital altitude, mass distribution properties, physical configuration, and orientation in space. These parameters collectively determine which disturbance type becomes predominant in different operational scenarios.

2.8.1 Gravity Gradient Torque

The gravity gradient torque arises from the variation in gravitational force across the spacecraft's dimensions. This torque occurs because the gravitational field strength decreases with altitude, creating a differential force across the satellite's structure [22]. The resulting torque $\vec{M}_{gg}$ can be expressed as:

$$\vec{M}_{gg} = \frac{3\mu}{R^3}\hat{R} \times I\hat{R} \tag{2.22}$$

where:

- μ is Earth's gravitational parameter
- R is the orbital radius
- $\hat{R}$ is the unit vector pointing toward Earth's center
- I is the spacecraft's inertia tensor

For a nadir-pointing spacecraft in a circular orbit, the maximum gravity gradient torque about each axis becomes:

$$M_{gg,i} = \frac{3\mu}{2R^3}|I_j - I_k| \sin(2\theta) \tag{2.23}$$

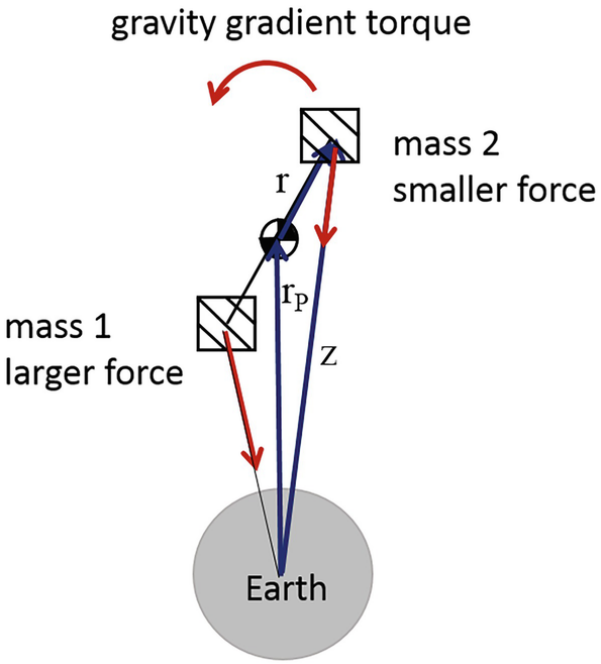

Figure 2.5: Gravity Gradient Torques

where θ represents the angular deviation from perfect nadir alignment, and I_j, I_k are the moments of inertia about the other two axes. The gravity gradient torque tends to align the spacecraft's minimum inertia axis with the local vertical direction, providing passive stabilization in suitable orbital conditions.

2.8.2 Aerodynamic Torque

In Low Earth Orbit, residual atmospheric particles exert drag forces on the satellite's surfaces, leading to aerodynamic torque. This torque arises when the center of pressure does not align with the center of mass, causing a rotational effect. The aerodynamic torque $\boldsymbol{T}_{\text{aero}}$ can be expressed as:

$$\boldsymbol{T}_{\text{aero}} = \boldsymbol{r}_{\text{cp}} \times \boldsymbol{F}_{\text{aero}} \tag{2.24}$$

where:

- $\boldsymbol{r}_{\text{cp}}$ is the vector from the satellite's center of mass to the center of pressure,
- $\boldsymbol{F}_{\text{aero}}$ is the aerodynamic force acting on the satellite.

The aerodynamic force $\boldsymbol{F}_{\text{aero}}$ can be calculated using:

$$\boldsymbol{F}_{\text{aero}} = -\frac{1}{2}\rho C_D A \boldsymbol{v}_{\text{rel}}^2 \tag{2.25}$$

where:

- ρ is the atmospheric density at the satellite attitude,
- C_D is the drag coefficient,
- A is the cross-sectional area facing the velocity vector,
- $\boldsymbol{v}_{\text{rel}}$ is the relative velocity of the satellite with respect to the atmosphere.

2.8.3 Solar Radiation Pressure Torque

Solar radiation exerts pressure on the satellite's surfaces, leading to torque when there is an offset between the center of pressure and the center of mass. The solar radiation pressure torque $\boldsymbol{T}_{\text{SRP}}$ is given by:

$$\boldsymbol{T}_{\text{SRP}} = \boldsymbol{r}_{\text{cp}} \times \boldsymbol{F}_{\text{SRP}} \tag{2.26}$$

where:

- $\boldsymbol{r}_{\text{cp}}$ is the vector from the satellite's center of mass to the center of pressure,
- $\boldsymbol{F}_{\text{SRP}}$ is the force due to solar radiation pressure.

The force $\boldsymbol{F}_{\text{SRP}}$ can be calculated as:

$$\boldsymbol{F}_{\text{SRP}} = \frac{\Phi}{c} A_s (1+q) \cos(\varphi) \hat{\boldsymbol{s}} \tag{2.27}$$

where:

- Φ is the solar constant (approximately $1367\,\text{W/m}^2$ at 1 AU),
- c is the speed of light,
- A_s is the surface area exposed to the Sun,
- q is the reflectivity coefficient of the surface,
- φ is the angle between the Sun vector and the surface normal,
- $\hat{\boldsymbol{s}}$ is the unit vector pointing from the Sun to the satellite.

2.8.4 Magnetic Torque

The interaction between the Earth's magnetic field and the satellite's residual magnetic dipole moment generates magnetic torque. This torque $\boldsymbol{T}_{\text{mag}}$ is expressed as:

$$\boldsymbol{T}_{\text{mag}} = \boldsymbol{m} \times \boldsymbol{B} \tag{2.28}$$

where:

- $\boldsymbol{m}$ is the magnetic dipole moment of the satellite,
- $\boldsymbol{B}$ is the Earth's magnetic field vector at the satellite's location.

The magnetic dipole moment $\boldsymbol{m}$ can be generated by onboard magnetorquers and is calculated as:

$$\boldsymbol{m} = nI\boldsymbol{A} \tag{2.29}$$

where:

- n is the number of turns in the coil,
- I is the current through the coil,
- $\boldsymbol{A}$ is the area vector of the coil.

2.9 Passive Attitude Stabilization

Passive attitude stabilization methods rely on natural environmental torques and inherent physical properties of the spacecraft to maintain or recover a desired orientation without the use of active control systems. These methods are energy-efficient and mechanically simple, making them well-suited for small satellites such as CubeSats [14]. The most common techniques include:

- **Gravity-gradient stabilization**: This method uses the Earth's gravity field to align an elongated spacecraft along the local vertical. The gravitational torque arises due to the difference in gravitational force between the spacecraft's ends, producing a restoring torque that stabilizes its orientation.
- **Spin stabilization**: In this approach, the spacecraft is set into a high-speed rotation about its principal axis. The resulting gyroscopic stiffness resists external torques, thereby maintaining orientation over time.
- **Magnetic stabilization**: This technique involves using permanent magnets or magnetically soft materials within the satellite. These interact with Earth's magnetic field to passively align the spacecraft along geomagnetic lines.
- **Damping mechanisms**: Passive damping can be achieved using hysteresis rods or fluid dampers to dissipate kinetic energy and reduce oscillations over time.

Although passive methods offer simplicity and reliability, they are generally limited in precision and responsiveness. As such, they are most appropriate for missions with modest pointing requirements or as an initial step before activating more precise active control systems.

2.10 Active Attitude Stablization

Active stabilization is a method used to control a satellite's orientation by actively adjusting its attitude through sensors and actuators. Sensors such as sun sensors, magnetometers, accelerometers, and gyroscopes provide real-time data on the satellite's position, magnetic field, acceleration, and rotational movement. Based on this data, torquers like magnetic coils, momentum wheels, or micro-thrusters apply corrective forces to maintain or change the satellite's orientation. This system requires electrical power and onboard computing to process sensor inputs and command the actuators. Active stabilization is divided into two main approaches: spin stabilization, where the satellite rotates around a single axis for stability, and three-axis stabilization, which maintains precise control over roll, pitch, and yaw. Unlike passive methods, active stabilization relies on moving parts and continuous adjustments to ensure accurate attitude control.

2.10.1 Three Axis Stabilization

Some satellites require autonomous attitude control along all three principal axes during their missions. This *three-axis stabilization* enables full control over the satellite's orientation in space and is more commonly used than gravity gradient or spin stabilization. This method facilitates precise maneuvering and ensures stable support for sensors and actuators. Its advantages include high pointing accuracy and stability in a non-inertial reference frame. However, such systems typically involve complex and costly hardware, which increases the satellite's weight and power consumption.

Generally, two types of three-axis stabilization systems are used:

a. Moment-Biased System: This type typically utilizes a single momentum wheel whose spin axis is aligned with the satellite's pitch axis (orthogonal to the orbital plane). The wheel maintains a high and constant angular velocity to provide gyroscopic stiffness. Control torque about the pitch axis is produced by adjusting the angular velocity ω of the wheel, such that:

$$\tau = I_w \frac{d\omega}{dt} \tag{2.30}$$

where I_w is the wheel's moment of inertia. Due to gyroscopic coupling, roll and pitch are controlled directly, while yaw is indirectly influenced via cross-axis interactions, reducing the need for a dedicated yaw sensor.

b. Zero Momentum System: In contrast, this system employs three reaction wheels aligned with each body axis (x, y, z). Control is achieved by independently varying each wheel's speed around a nominal zero-momentum bias. If an external torque $\boldsymbol{\tau}_{\text{dist}}$ disturbs the satellite, the control system computes a compensating torque using feedback:

$$\boldsymbol{\tau}_{\text{ctrl}} = -K_p \boldsymbol{e}_\theta - K_d \dot{\boldsymbol{e}}_\theta \tag{2.31}$$

where $\boldsymbol{e}_\theta$ is the attitude error vector, and K_p, K_d are proportional and derivative gains, respectively. The generated torque changes each wheel's angular velocity, producing an equal and opposite torque on the satellite body per Newton's third law. Over time, the reaction wheels absorb the disturbances.

Three-axis control systems often incorporate actuators such as reaction wheels, momentum wheels, magnetic torquers, control moment gyroscopes (CMGs), or thrusters to maintain and correct the satellite's attitude in real time.

2.10.2 Magnetorquer Model

The magnetorquer model assumes that three magnetorquers are aligned with the principal axes of the satellite's body frame. Each magnetorquer is independently controlled by an applied current vector:

$$\vec{M} = \begin{bmatrix} i_x \\ i_y \\ i_z \end{bmatrix} \tag{2.32}$$

The magnetic moment generated by the magnetorquers is given by:

$$\vec{\mu}_M = nA\vec{i}_M \tag{2.33}$$

where:

- n is the number of turns in the coil,
- A is the surface area of the coil,
- $\vec{i}_M$ is the applied current vector.

Assuming all magnetorquers have identical physical characteristics, the torque exerted on the satellite by the interaction of the magnetic moment with Earth's magnetic field $\vec{B}$ (in the body frame) is calculated as:

$$\vec{M}_M = \vec{\mu}_M \times \vec{B} \tag{2.34}$$

This torque is used for attitude control, especially in detumbling operations or coarse stabilization phases. Magnetorquers are favored for small satellites like CubeSats due to their simplicity, low power consumption, and lack of moving parts.

In order to obtain the magnetic field vector in the body frame, the inertial magnetic field vector must be rotated into the body frame of the satellite. In component form, the following equation is applied using the identity $\mathbf{a} \times \mathbf{b} = -\mathbf{b} \times \mathbf{a}$:

$$\begin{bmatrix} L_M \\ M_M \\ N_M \end{bmatrix} = \eta A \begin{bmatrix} 0 & \beta_z & -\beta_y \\ -\beta_z & 0 & \beta_x \\ \beta_y & -\beta_x & 0 \end{bmatrix} \begin{bmatrix} i_x \\ i_y \\ i_z \end{bmatrix} \tag{2.35}$$

International Geomagnetic Reference Field (IGRF) Model

The **International Geomagnetic Reference Field (IGRF)** is a standard mathematical model of the Earth's main magnetic field and its secular variation, developed by the International Association of Geomagnetism and Aeronomy (IAGA). It is widely used in aerospace applications such as attitude estimation and magnetorquer-based attitude control for satellites, particularly CubeSats.

In satellite simulations, the IGRF model [23] provides the Earth's magnetic field vector at a given location in space. This magnetic field information is critical for comparing onboard magnetometer measurements to reference values, enabling the estimation of the spacecraft's orientation.

To utilize the IGRF model in a simulation, the following steps are typically performed:

1. **Position Input**: The satellite's location is defined using geographic coordinates (latitude, longitude, and altitude), often obtained from GPS data.

2. **Frame Transformation**: The output of the IGRF model, usually given in a local geographic frame such as East-North-Vertical (ENV), is transformed to the Earth-Centered Inertial (ECI) frame through an intermediate North-East-Down (NED) frame.

3. **Magnetic Field Vector**: The model yields the magnetic field vector $\vec{\beta}_I$ in the inertial frame, which is then used in control and estimation algorithms.

The magnetic disturbance torque acting on the satellite due to the Earth's magnetic field is modeled as:

$$\vec{M}_{MD} = \vec{\beta}_I d_s \tag{2.36}$$

where:

- $\vec{M}_{MD}$ is the magnetic disturbance torque,
- $\vec{\beta}_I$ is the magnetic field vector in the inertial frame (provided by IGRF),
- d_s is a constant representing the magnetic dipole of the spacecraft (e.g., $d_s = 2.64 \times 10^{-3}\,\mathrm{N \cdot m/T}$).

In practice, the **EMM2015 model**, derived from IGRF, may be used for more detailed simulation, offering high resolution with up to 720 spherical harmonic degrees and a spatial resolution of approximately 56 km. This data can be accessed programmatically using tools such as *GeographicLib*.

B-dot Controller

In LEO, the standard B-dot controller reported in many sources ([24], [25], [26], [27]) can be used to de-tumble a satellite. The standard B-dot controller requires the magnetorquers to follow the control law shown below

$$\boldsymbol{\mu}_B \;=\; k\, S(\boldsymbol{\omega}_{B/I})\, T_{BI}(\mathbf{q})\, \boldsymbol{\beta}_I \tag{2.37}$$

where k is the control gain. Using equation (2.32) it is possible to write the current in component form again using the identity that $\mathbf{a} \times \mathbf{b} = -\mathbf{b} \times \mathbf{a}$:

$$\begin{pmatrix} i_x \\ i_y \\ i_z \end{pmatrix} = \frac{k}{nA} \underbrace{\begin{pmatrix} 0 & \beta_z & -\beta_y \\ -\beta_z & 0 & \beta_x \\ \beta_y & -\beta_x & 0 \end{pmatrix}}_{S(\boldsymbol{\beta}_I)} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{2.38}$$

This equation can then be substituted into equation (2.35) to produce the total torque on the satellite assuming that the magnetorquers can provide the necessary current commanded by equation (2.38):

$$\begin{pmatrix} L \\ M \\ N \end{pmatrix} = -k \underbrace{\begin{pmatrix} \beta_y^2 + \beta_z^2 & -\beta_x\beta_y & -\beta_x\beta_z \\ -\beta_x\beta_y & \beta_x^2 + \beta_z^2 & -\beta_y\beta_z \\ -\beta_x\beta_z & -\beta_y\beta_z & \beta_x^2 + \beta_y^2 \end{pmatrix}}_{\Phi(\boldsymbol{\beta}_I)} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{2.39}$$

The goal of the controller here is to drive $\boldsymbol{\omega}_{B/I} \to \mathbf{0}$. The literature shows this is not fully achieved [28] , for two main reasons:

- If the spacecraft's magnetic moment becomes colinear with the Earth's field, $\Phi(\boldsymbol{\beta}_I)$ singularly loses rank and produces zero torque.
- Equation (2.38) gives zero current any time $\boldsymbol{\omega}_{B/I}$ is colinear with $\boldsymbol{\beta}_I$.

If we define the *transition matrix*

$$W \;=\; T_{BI}(\mathbf{q})\,\boldsymbol{\beta}_I$$

then the torque simplifies to $-k\,W^T\,S(\boldsymbol{\omega}_{B/I})$. Putting W into row-reduced echelon form reveals $\det W = 0$:

$$\operatorname{rref}\big(W\big) = \begin{pmatrix} 1 & 0 & -\frac{\beta_x}{\beta_z} \\ 0 & 1 & -\frac{\beta_y}{\beta_z} \\ 0 & 0 & 0 \end{pmatrix} \quad \Longrightarrow \quad \det W = 0. \tag{2.40}$$

A zero determinant means there exists a nonzero $\boldsymbol{\omega}_{B/I}$ in the nullspace of W for which the controller applies no corrective torque. In practice, because $\boldsymbol{\beta}_I$ varies with orbit and attitude, this degeneracy is avoided most of the time—but persists near low-inclination passes (when $\beta_x, \beta_y \approx 0$) or pure yaw tumbling ($p = q = 0$), at which point the controller can no longer reduce residual yaw rate without additional actuators such as reaction wheels.([29])

2.10.3 Reaction Wheel

A reaction wheel is a motorized rotating mass used to provide precise maneuvering torque for satellite attitude control, enabling accurate pointing and slew maneuvers by spinning up or down to induce an opposing torque on the spacecraft. Despite their high precision, reaction wheels can reach saturation due to prolonged momentum buildup, necessitating desaturation techniques such as magnetorquers or thrusters to unload excess angular momentum. Each wheel typically consists of a steel cylinder mounted on a low-power, cost-effective brushless motor shaft, designed to maximize momentum storage within the strict volume and mass constraints of small satellites like CubeSats. For full three-axis attitude control, a system requires three reaction wheels oriented orthogonally to provide torque in all directions .

Reaction Wheel Controller

Assuming each reaction is aligned with a principal axis of inertia the control scheme is extremely simple. When the wheels are not aligned the derivation will proceed similar to the reaction control thruster section. The derivation here will just be for the aligned case. In this analysis it is assumed that a torque can be applied to the reaction wheel and thus the angular velocity of the reaction wheel α_{R_i} can be directly controlled. Assuming this a simple PD control law can be used to orient the satellite at any desired orientation using Euler angles for this control law since the satellites are aligned with the principal axes of rotation [15].

$$\alpha_{R_i} = -k_p\big(\epsilon_i - \epsilon_{\text{desired}}\big) - k_d\big(\omega_i - \omega_{\text{desired}}\big)\,, \tag{2.41}$$

where ϵ denotes either roll ϕ, pitch θ, or yaw ψ, and the true Euler angles are obtained by converting the attitude quaternion .

Often times however your reaction wheels are not pointed on the principal axis of inertia. In this case a Least Squares Regression model is needed. In this case the equation above is used to compute the desired torque to be placed on the satellite such that

$$\mathbf{M}_{\text{desired}} = -k_p\big(\epsilon_i - \epsilon_{\text{desired}}\big) - k_d\big(\omega_i - \omega_{\text{desired}}\big)\,. \tag{2.42}$$

This desired torque must be produced by the reaction wheels:

$$\mathbf{M}_{\text{desired}} = \mathbf{M}_R = \sum_{i=1}^{N_{\text{RW}}} I_{BR_i}\,\alpha_{R_i}\,\hat{n}_{Ri} = \begin{bmatrix} I_{BR_1}\,\hat{n}_{R1} & \cdots & I_{BR_{N_{\text{RW}}}}\,\hat{n}_{R_{N_{\text{RW}}}} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_{N_{\text{RW}}} \end{bmatrix}. \tag{2.43}$$

Since the matrix

$$J \;=\; \begin{bmatrix} I_{BR_1}\,\hat{n}_{R1} & \cdots & I_{BR_{N_{\text{RW}}}}\,\hat{n}_{R_{N_{\text{RW}}}} \end{bmatrix} \;\in\; R^{3\times N_{\text{RW}}}$$

is not square, the minimum-norm least-squares solution for the vector of wheel accelerations $\boldsymbol{\alpha}$ is

$$\boldsymbol{\alpha} = J^T\big(J\,J^T\big)^{-1}\,\mathbf{M}_{\text{desired}}\,. \tag{2.44}$$

2.11 Attitude Stabilization Controllers

Spacecraft attitude stabilization employs closed-loop controllers to generate corrective torques that counteract disturbances (gravity-gradient, aerodynamic, solar radiation) and drive the attitude error toward zero . Classical methods linearize the rigid-body dynamics about an operating point and use fixed-gain feedback, while modern approaches exploit optimization or intelligent rule-based schemes to handle nonlinearities and uncertainty .

2.11.1 PID Controller

The Proportional-Integral-Derivative (PID) controller is one of the most widely used and fundamental feedback control strategies in engineering systems. Its popularity stems from its simplicity, ease of implementation, and effectiveness in a wide range of applications, especially when the underlying process model is unknown or partially known [30].

A typical PID control system consists of three distinct control actions: proportional, integral, and derivative. These are governed by positive scalar gains denoted as K_p, K_i, and K_d, respectively. The control signal $u(t)$ is generated using the error signal $e(t)$, which is the difference between the reference signal $r(t)$ and the system output $y(t)$.

The ideal form of the PID controller in the time domain is given by:

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt} \tag{2.45}$$

In this formulation:

- The **proportional term** $K_p e(t)$ contributes to reducing the rise time of the system by reacting proportionally to the current error.

- The **integral term** $K_i \int e(t)dt$ helps eliminate the steady-state error by accumulating past error values.
- The **derivative term** $K_d \frac{de(t)}{dt}$ anticipates future error trends, thus improving the transient response and reducing overshoot.

The combined effect of these three terms allows the controller to maintain desired system performance by continuously adjusting the control input $u(t)$ to minimize the error signal $e(t)$ [31].

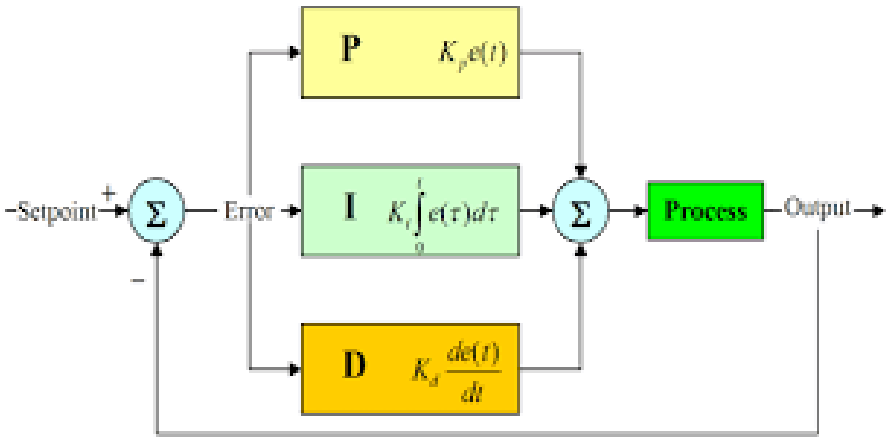

Figure 2.6: Standard PID Controlled System

2.11.2 LQR (Linear Quadratic Regulator)

The Linear Quadratic Regulator (LQR) solves

$$J = \int_0^\infty \left(\mathbf{x}^T Q\,\mathbf{x} \;+\; \mathbf{u}^T R\,\mathbf{u}\right) dt \tag{2.46}$$

subject to $\dot{\mathbf{x}} = A\,\mathbf{x} + B\,\mathbf{u}$, yielding the optimal feedback $\mathbf{u} = -K\,\mathbf{x}$ via the Algebraic Riccati Equation. LQR provides guaranteed stability and disturbance rejection when the model is accurate, but degrades under modeling errors and unmodeled nonlinearities, often necessitating robust or adaptive extensions .

- **Optimal control, state-space model-based**: minimizes a quadratic cost of states and control effort.
- **Requires accurate modeling**: performance depends critically on fidelity of (A, B) matrices and disturbance.

2.11.3 Fuzzy Logic Controller (FLC)

Fuzzy Logic Control uses a set of linguistic IF–THEN rules to map normalized attitude error and rate into actuator commands via fuzzy membership functions and T-norm inference . It handles nonlinear dynamics and parameter uncertainty without explicit linearization, and can be tuned by expert knowledge rather than precise models. Experiments on small satellites show FLC often achieves faster settling times and greater robustness to sensor noise and actuator saturation compared to PID or LQR under similar conditions.

- **Nonlinear, rule-based**: directly encodes expert heuristics into IF–THEN rules without model inversion.
- **Robust to uncertainties**: tolerates parameter drift, unmodeled dynamics, and measurement noise through fuzzy blending.
- **Proven satellite performance**: demonstrates reduced overshoot and quicker detumbling in CubeSat experiments.
- **Versatile rule design**: easily extends to multi-input, multi-output controllers for complex missions.
- **Application to solar sails**: successfully stabilizes highly nonlinear solar-sail attitude dynamics.

Given the highly nonlinear satellite dynamics, limited onboard computation, and uncertain disturbance profiles, FLC offers a robust, model-free solution that can be rapidly reconfigured on-orbit via rule updates. Its inherent tolerance to actuator saturation and sensor noise makes it ideal for CubeSat missions with constrained hardware reliability.

Chapter 3

Fuzzy Logic

3.1 Introduction

A control system serves three core functions: sensing, managing, and regulating a system's behavior to achieve desired outputs. Through automation, control systems employ various methods to oversee and optimize processes while minimizing disruptions caused by operational variations or environmental factors. These disturbances can sometimes be severe enough to compromise conventional linear controllers, whose primary role is to maintain process stability and performance. Control systems generally fall into two categories: open-loop (OL) systems, where inputs directly dictate outputs without adjustments, and closed-loop (CL) systems, which rely on continuous feedback. In closed-loop systems, sensors measure outputs, compare them to target inputs, and use the resulting error signals to dynamically adjust actuators, ensuring precise and adaptive system performance.

3.2 Control Design

The Attitude Determination and Control System (ADCS) plays a crucial role in CubeSat operations by collecting data from attitude sensors to determine the spacecraft's current orientation. By comparing this real-time attitude with the desired orientation, the ADCS employs control algorithms to calculate corrective actions, which are then executed by actuators to minimize attitude errors. Several control techniques are available for this purpose, with the Proportional-Integral-Derivative (PID) controller being the most widely adopted due to its simplicity and effective feedback mechanism. For more advanced applications, the Linear Quadratic Regulator (LQR) offers an optimal control solution by framing the problem as a mathematical optimization and employing state-space analysis. Alternatively, Fuzzy Logic Control (FLC) provides a model-free approach, using linguistic rules derived from human expertise rather than mathematical models - making it particularly valuable for complex or uncertain systems

3.3 Fuzzy Logic

A fuzzy logic system (FLS) is unique in that it is able to simultaneously handle numerical data and linguistic knowledge. It is a nonlinear mapping of an input data (feature) vector into a scalar output, i.e., it maps numbers into numbers. Fuzzy set theory and fuzzy logic

establish the specifics of the nonlinear mapping. This tutorial paper provides a guided tour through those aspects of fuzzy sets and fuzzy logic that are necessary to synthesize a FLS. It does this by starting with crisp set theory and dual logic and demonstrating how both can be extended to their fuzzy counterparts. Because engineering systems are, for the most part, causal, we impose causality as a constraint on the development of the FLS. Doing this lets us steer down a very special and widely used tributary of the FL literature, one that is valuable for engineering applications of FL, but may not be as valuable for nonengineering applications. After synthesizing a FLS, we demonstrate that it can be expressed mathematically as a linear combination of fuzzy basis functions, and is a nonlinear universal function approximator, a property that it shares with feedforward neural networks. The fuzzy basis function expansion is very powerful because its basis functions can be derived from either numerical data or linguistic knowledge, both of which can be cast into the forms of IF-THEN rules. To date, a FLS is the only approximation method that is able to incorporate both types of knowledge in a unified mathematical manner [32].

3.3.1 Crisp Sets (Classical Sets)

A crisp set is a collection of distinct elements where each element either fully belongs to the set or does not belong at all. This binary nature is characterized by a membership function $\mu_A(x)$ defined as:

- $\mu_A(x) = 1$ if $x \in A$
- $\mu_A(x) = 0$ if $x \notin A$

For example, consider the set of even numbers. The number 4 belongs to this set ($\mu_A(4) = 1$), while the number 5 does not ($\mu_A(5) = 0$).

Crisp sets are foundational in classical set theory and are extensively used in digital systems and logic design.

3.3.2 Fuzzy Sets

Introduced by Lotfi A. Zadeh in 1965 [33], a fuzzy set allows for degrees of membership, accommodating the vagueness inherent in many real-world concepts. The membership function $\mu_A(x)$ for a fuzzy set assigns a value between 0 and 1, indicating the degree to which element x belongs to set A:

- $\mu_A(x) \in [0, 1]$

For instance, in the fuzzy set of "tall people," someone who is 6 feet tall might have a membership value of 0.9, while someone who is 5.5 feet tall might have a value of 0.5.

Fuzzy sets are particularly useful in fields like control systems, decision-making, and artificial intelligence, where binary classifications are insufficient [34].

Operations on Fuzzy Sets

Fuzzy set theory extends classical set operations through three fundamental operators.

The **complement** (negation) operation represents the degree to which an element does not belong to a fuzzy set, defined mathematically as

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \tag{3.1}$$

where $\mu_A(x)$ is the membership function of the original set A. The **intersection** (AND operation) between two fuzzy sets A and B determines the degree to which an element belongs to both sets simultaneously, implemented through triangular norms (t-norms) with the general form
$\mu_{A\cap B}(x) = T(\mu_A(x), \mu_B(x))$, where common implementations include the minimum operator $min(\mu_A, \mu_B)$ or the algebraic product $\mu_A \cdot \mu_B$. The **union** (OR operation) calculates the membership in either set, expressed through t-conorms (s-norms) as

$$\mu_{A\cup B}(x) = S(\mu_A(x), \mu_B(x)) \tag{3.2}$$

typically realized using the maximum function $max(\mu_A, \mu_B)$ or the probabilistic sum $\mu_A + \mu_B - \mu_A \cdot \mu_B$. These fuzzy operations maintain consistency with classical set theory when membership values are restricted to the crisp set $\{0, 1\}$, while providing graded transitions for intermediate membership values. The selection of specific operators significantly impacts controller behavior, where minimum/maximum operations yield more conservative, strict boundaries while product/sum operations produce smoother response surfaces.

3.3.3 Linguistic Variables

Zadeh states,In retreating from precision in the face of overpowering complexity, it is natural to explore the use of what might be called linguistic variables, that is, variables whose values are not numbers but words or sentences in a natural or artificial language.The motivation for the use of words or sentences rather than numbers is that linguistic characterizations are, in general, less specific than numerical ones [33].

Let u denote the name of a linguistic variable (e.g., temperature). Numerical values of a linguistic variable u are denoted by x, where $x \in U$. Sometimes x and u are used interchangeably, especially when a linguistic variable is a letter, as is often the case in engineering applications.

A linguistic variable is usually decomposed into a set of terms, $T(u)$, which cover its universe of discourse [35].

3.3.4 Fuzzification of Inputs

The fuzzification phase of the fuzzy controller involves constructing a fuzzy input relation. If the input is already represented by a fuzzy relation, then additional fuzzification may not be necessary. However, typically, the input relation A', used within the compositional rule of inference, is not directly available. Therefore, fuzzification is required [36].

The fuzzy input relation A' is defined as the conjunction of N_x fuzzy input sets A'_i, where N_x is the number of controller inputs. Each fuzzy set A'_i represents a fuzzy interpretation of the controller input x'_i:

$$A'_i = \text{fuzz}(x'_i) \tag{3.3}$$

Here, fuzz is a fuzzifier function, an operator that translates a numerical value into a fuzzy set representation. When an input is purely a numerical value (meaning no

uncertainty exists), which is common in controllers, the corresponding fuzzy set A'_i is simply given by a singleton:

$$\mu_{A'_i}(x_i) = \begin{cases} 1, & \text{if } x_i = x'_i \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

Uncertainty, imprecision, or inaccuracies in the inputs can be modeled by using fuzzy numbers. The complete fuzzy input relation A' is determined by combining the individual fuzzy sets corresponding to each input:

$$A' = \prod_{i=1}^{N_x} A'_i(x_i) = \Big(T\left(\mu_{A'_1}(x_1), \ldots, \mu_{A'_{N_x}}(x_{N_x})\right) \Big/ (x_1, \ldots, x_{N_x}) \Big) \tag{3.5}$$

where T denotes the T-norm used to perform the conjunction operation in the premise. This shows that the input data is effectively represented by the combination of A'_1, A'_2, ..., A'_{N_x}.

3.3.5 Defuzzification of Outputs

After the fuzzy inference process is completed, the result is a fuzzy output set. However, most real-world applications require a crisp, actionable output. Therefore, defuzzification is the process by which a fuzzy set is translated into a single crisp output value [36].

The most commonly used defuzzification methods include:

- **Centroid Method (Center of Gravity):** Computes the center of mass of the resulting fuzzy set.
- **Bisector Method:** Finds the value that divides the fuzzy set into two regions of equal area.
- **Mean of Maximum (MoM):** Averages all values at which the membership function reaches its maximum.
- **Smallest (or Largest) of Maximum (SOM/LOM):** Selects the smallest or largest value at which the maximum membership occurs.

Among these, the centroid method is the most widely used due to its balance between simplicity and accuracy. It is defined mathematically as:

$$y^* = \frac{\int_Y y\, \mu_{B'}(y)\, dy}{\int_Y \mu_{B'}(y)\, dy} \tag{3.6}$$

where:

- y^* is the crisp output.
- $\mu_{B'}(y)$ is the membership function of the fuzzy output set B'.
- Y is the domain of the output variable.

Defuzzification thus bridges the gap between fuzzy reasoning and crisp, real-world actions.

3.3.6 Fuzzy Rules

In order to reason with fuzzy logic, fuzzy rules have to be represented by an implication function. Such a fuzzy implication has the same function as the truth table of the classical implication in classical logic [37]. In classical logic, the implication is denoted by:

$$A \rightarrow B \tag{3.7}$$

which can be seen as a representation of the statement:

$$\textit{If } A \textit{ then } B$$

In fuzzy logic, these types of statements are often referred to as *fuzzy if-then statements* or *fuzzy rules*.

Table 3.1: Truth table of classical implication.

A	B	$A \rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

The truth table of the classical implication is given in Table 2.1. The next section shows how a fuzzy rule can be represented by a fuzzy relation by means of a fuzzy implication function. combined by means of an aggregation operator.

Representation of a Fuzzy Rule

A fuzzy rule is an *if-then* statement where the premise and the consequent consist of fuzzy propositions. The premise can contain a combination of propositions by means of the logical connectives and/or. It is also possible that a fuzzy proposition is based on a negation. For the sake of simplicity, the following rule is considered:

$$\textit{If } z \textit{ is } A_1 \textit{ and } z \textit{ is } A_2\textit{, then } g \textit{ is } B$$

When fuzzy sets A_1, A_2, and B are identified by the membership functions $\mu_{A_1}(z)$, $\mu_{A_2}(z)$, and $\mu_B(g)$, the following fuzzy relation R representing the fuzzy rule can be constructed:

$$R = I\left(T(\mu_{A_1}(z), \mu_{A_2}(z)), \mu_B(g)\right) \tag{3.8}$$

where T is a conjunction based on a general T-norm and I is a fuzzy implication function. As the T-norm T represents (or models) the *and* connective, the fuzzy implication function I represents (or models) the implication: the *if-then* connective. Hence, a fuzzy rule can be represented by a fuzzy relation. The membership function μ_R of the above example is given by:

$$\mu_R(z, g) = I\left(T(\mu_{A_1}(z), \mu_{A_2}(z)), \mu_B(g)\right) \tag{3.9}$$

Various implication functions are described in the following sections. The implication function I is mostly denoted by $I(a, b)$, where $a, b \in [0, 1]$.

3.3.7 Fuzzy Reasoning

Fuzzy reasoning represents the process of deriving conclusions from fuzzy rules and input data, serving as the computational foundation of fuzzy logic controllers. The mechanism operates through compositional rule of inference, where fuzzy inputs are combined with rule antecedents using sup-star composition, mathematically expressed as

$$\mu_{B'}(y) = \sup(x \in X)[\mu_{A'}(x) \star \mu_{A \to B}(x, y)] \tag{3.10}$$

where $\star$ denotes a t-norm operator and $\mu_{A \to B}(x, y)$ represents the rule's implication relation. The most common approaches to fuzzy reasoning include Mamdani's method, which uses min for implication and max for aggregation, yielding an output fuzzy set

$$\mu_{B'}(y) = \max_{i=1}^{n} \min[\mu_{A'_i}(x), \mu_{B_i}(y)] \tag{3.11}$$

for n rules, and Sugeno's method that employs weighted averaging of crisp functions. The reasoning process involves three key phases: fuzzification of crisp inputs through membership functions, application of fuzzy operators (t-norms and t-conorms) to evaluate rule antecedents, and aggregation of rule consequents using either maximum or summation operators. The choice of implication method (Mamdani, Larsen, or others) significantly affects the controller's response characteristics, with Mamdani-type inference providing more intuitive, linguistically interpretable results while Sugeno-type offers computational efficiency advantages.
Defuzzification subsequently converts the aggregated fuzzy output into a crisp control action through methods like centroid calculation or height defuzzification, completing the fuzzy reasoning cycle.

3.4 Fuzzy Logic Controller

A Fuzzy Logic Controller (FLC) is a control system framework that leverages fuzzy set theory to reason with imprecise or uncertain information, making it especially well suited for complex, nonlinear, or poorly modeled processes. Rather than relying on exact mathematical models, an FLC translates crisp sensor readings into degrees of membership in linguistic categories (fuzzification), applies a network of human-readable "if–then" rules to infer fuzzy outputs (inference), and then converts those fuzzy conclusions back into precise actuator commands (defuzzification). By mimicking the way experts describe system behavior—using terms like "small," "medium," or "large"—an FLC can incorporate both quantitative data and qualitative insights, thereby providing robust performance in the face of disturbances and modeling errors.

3.4.1 Controller Architectures

Hierarchical Fuzzy Control

Hierarchical fuzzy control architectures decompose complex control problems into manageable sub-tasks organized in multiple layers, where higher layers handle more abstract decisions while lower layers implement specific control actions. Jager describes this approach as particularly effective for high-dimensional systems where a single rule base would become prohibitively large, noting that a hierarchical structure reduces the number of required rules from N^n to $n \times N^2$ for an n-dimensional system with N membership

functions per input. The architecture typically consists of a cascaded arrangement where the output of higher-level controllers serves as input to subordinate layers, with coordination mechanisms ensuring consistent behavior across levels. Proper design must maintain interpretability while achieving computational efficiency, suggesting that 2-3 layers typically provide optimal balance between complexity reduction and controller transparency.

Adaptive Fuzzy Control

Adaptive fuzzy control systems dynamically modify their parameters or rule bases to accommodate changing process conditions or performance requirements. There are three primary adaptation mechanisms: parameter adjustment (tuning membership functions or scaling factors), rule base modification (adding/pruning rules), and structural adaptation (changing inference methods). The text presents the mathematical formulation for online parameter adaptation using gradient descent methods:

$$\Delta\theta(k) = -\eta\frac{\partial J}{\partial\theta} + \alpha\Delta\theta(k-1) \tag{3.12}$$

where θ represents adjustable parameters, η the learning rate, α the momentum factor, and J the performance measure.
the importance of stability guarantees in adaptive schemes, recommending Lyapunov-based approaches for rigorous proof of convergence in safety-critical applications.

Neuro-Fuzzy Systems

Neuro-fuzzy systems combine the learning capability of neural networks with the interpretability of fuzzy logic, creating hybrid architectures where neural networks optimize fuzzy system parameters. There is two fundamental configurations: cooperative systems where neural networks pre-process inputs for fuzzy systems, and fused architectures where neural networks directly implement fuzzy operations. The text provides the mathematical basis for ANFIS (Adaptive Neuro-Fuzzy Inference System) networks, showing how layer-by-layer computation implements Takagi-Sugeno fuzzy inference:

$$f = \frac{\sum_{i=1}^{N} w_i f_i}{\sum_{i=1}^{N} w_i} \tag{3.13}$$

where w_i are rule firing strengths and f_i the rule consequents.
These systems particularly excel when both numerical data and expert knowledge are available, with the neural component extracting patterns from data while the fuzzy component maintains interpretable rule structures.

3.4.2 Mamdani-Type Fuzzy Controller

The Mamdani controller, introduced by E. H. Mamdani in 1975 [38], represents both inputs and outputs as fuzzy sets. A typical rule takes the form:

> *If the error is A and the change in error is B, then the control action is C.*

Mathematically, if $\mu_A(x)$ and $\mu_B(y)$ are the membership functions of the inputs, the rule's firing strength is given by a T-norm, often the minimum:

$$w = \min\big(\mu_A(x),\, \mu_B(y)\big). \tag{3.14}$$

The consequent fuzzy set is then clipped at height w, and after aggregating all rule outputs (typically via a maximum operator), the combined fuzzy set $\mu_C(u)$ is defuzzified—most commonly by the centroid formula

$$u^* = \frac{\int u\,\mu_C(u)\,du}{\int \mu_C(u)\,du} \tag{3.15}$$

yielding a smooth, intuitive control signal that captures the aggregate effect of all activated rules.

3.4.3 Sugeno-Type Fuzzy Controller

Michio Sugeno's model (1985) [39] retains fuzzified inputs and rule premises but replaces fuzzy output sets with crisp functions—usually linear or constant. A Sugeno rule reads:

> *If the error is A and the change in error is B, then the control action is $f(x, y) = p_0 + p_1 x + p_2 y$.*

Each rule's weight w_i is computed (often via product of membership degrees), and the final output is computed as the weighted average of all rule conclusions:

$$u^* = \frac{\sum_i w_i\, f_i(x, y)}{\sum_i w_i} \tag{3.16}$$

Because no defuzzification over an arbitrary fuzzy set is needed, Sugeno controllers are computationally efficient and easily tuned using optimization or adaptive techniques, making them highly suitable for real-time control and systems that require automatic parameter adjustment.

3.4.4 Fuzzy Inference

The inference step applies the fuzzified inputs to the rule base and produces one aggregated fuzzy output. This process consists of three sub-steps:

1. **Rule Evaluation.** For each rule of the form *"If x_1 is A_i and x_2 is B_i, then y is C_i"*, compute its firing strength w_i. In a Mamdani FLC:

$$w_i = \min\big(\mu_{A_i}(x_1),\, \mu_{B_i}(x_2)\big) \tag{3.17}$$

   whereas in a Sugeno FLC one often uses the product T-norm:

$$w_i = \mu_{A_i}(x_1) \times \mu_{B_i}(x_2) \tag{3.18}$$

2. **Implication and Output Shaping.**
   - Mamdani: clip each output fuzzy set C_i at height w_i: $\mu'_{C_i}(y) = \min\big(\mu_{C_i}(y), w_i\big)$.
   - Sugeno: evaluate the crisp output function $f_i(x_1, x_2)$ for each rule.

3. **Aggregation.**

- Mamdani: combine the clipped sets via maximum: $\mu_C(y) = \max_i \mu'_{C_i}(y)$.
- Sugeno: compute a weighted average:

$$y^* = \frac{\sum_i w_i \, f_i(x_1, x_2)}{\sum_i w_i} \tag{3.19}$$

This inference mechanism smoothly blends multiple expert rules into a cohesive fuzzy decision, balancing all activated rules according to their firing strengths.

3.4.5 Advanced Controller Types

Type-2 Fuzzy Controllers

Type-2 fuzzy controllers extend conventional fuzzy logic by incorporating uncertainty in membership function definitions, making them particularly suitable for systems with significant noise or parameter variations.These controllers employ three-dimensional membership functions where each input point maps to a *footprint of uncertainty* (FOU), mathematically represented as

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x, u)/(x, u) \tag{3.20}$$

where $J_x \subseteq [0, 1]$ denotes the primary membership grades. The inference process involves interval calculations through type-reduction (using methods like Karnik-Mendel algorithm [40]) followed by defuzzification, providing inherent robustness against measurement uncertainties. Practical implementations typically use interval type-2 fuzzy sets for computational efficiency while maintaining most theoretical advantages, with noting 15-25% performance improvement over type-1 controllers in noisy environments according to experimental studies [41].

Fuzzy PID Controllers

Fuzzy PID controllers combine conventional PID structure with fuzzy logic to achieve nonlinear adaptive control.There is three architectural variants: the *fuzzy gain scheduler* that dynamically adjusts PID parameters (K_p, K_i, K_d) through rule-based inference, the *fuzzy pre-compensator* that processes error signals before PID computation, and the *full fuzzy PID* implementation where all three control actions are separately computed through fuzzy inference. The text provides the generalized control law:

$$u(t) = K_p^F \cdot e(t) + K_i^F \cdot \int e(t)dt + K_d^F \cdot de(t)/dt \tag{3.21}$$

where superscript F indicates fuzzy-determined gains. Jager emphasizes that the fuzzy PID's superiority emerges in handling nonlinear plant dynamics, showing examples where it reduces settling time by 30-40% compared to fixed-gain PID for systems with variable time constants [42].

Hybrid Fuzzy-PID Systems

Hybrid architectures strategically combine fuzzy and conventional PID control actions based on system conditions. Three integration methods: *switched systems* that select either fuzzy or PID control via stability criteria, *parallel operation* where fuzzy correction supplements PID output
($u_{total} = u_{PID} + \Delta u_{fuzzy}$)
, and *embedded designs* using PID in the inner loop with fuzzy supervisory control. The text provides stability analysis using Lyapunov functions:

$$V = \frac{1}{2}S^2 + \frac{1}{2\gamma}\tilde{\theta}^T\tilde{\theta} \tag{3.22}$$

where S is the sliding surface and $\tilde{\theta}$ parameter estimation errors. Case studies demonstrate these hybrids particularly effective in aerospace applications where they combine PID's reliability for nominal conditions with fuzzy logic's adaptability during anomalies, showing 50% faster disturbance rejection in attitude control simulations [43].

3.5 Fuzzy Logic Attitude Control

Active attitude determination involves comparing the satellite's measured orientation with mission-specific target values. The discrepancy between these quantities constitutes the attitude error. The attitude control system (ACS) functions to produce compensating torques that minimize this error. Given the persistent presence of external perturbations, along with inherent measurement inaccuracies and imperfect actuation, this corrective process operates continuously throughout the mission lifetime.

The ACS represents a critical subsystem for all spacecraft operations, with its fundamental objective being the precise alignment of the vehicle according to mission requirements. Various ACS architectures exist, each tailored to specific operational needs. Traditional control methodologies have been extensively studied for satellite attitude applications, though such approaches frequently exhibit constraints due to their linear nature and dependence on precise system modeling.

While proportional-integral-derivative (PID) controllers demonstrate satisfactory performance for first and second-order systems, they prove inadequate when confronting systems characterized by significant time delays, substantial parametric uncertainties, and periodic disturbances - scenarios demanding more advanced control solutions.

Fuzzy logic control (FLC) systems have demonstrated remarkable success across diverse engineering applications. For nonlinear systems in particular, numerous experimental studies have verified FLC's robustness against measurement noise and modeling inaccuracies. The principal benefit of fuzzy controllers lies in their capacity to intuitively integrate domain expertise through linguistic rule bases. Within aerospace engineering, fuzzy control has emerged as a particularly active research domain.

Consequently, fuzzy control techniques find particular utility in complex dynamical systems where either precise mathematical models are unavailable or system nonlinearities prove challenging for conventional methods [14].

3.5.1 Three-axis Fuzzy Controller Design

The three-axis fuzzy controller employs independent fuzzy logic systems for roll, pitch, and yaw control. Each axis controller processes attitude errors and angular rates to generate

appropriate control torques. The design methodology follows these key steps:

Input Variables

The controller uses two inputs per axis:

- Attitude error (e): Difference between current and desired orientation
- Angular rate ($\dot{e}$): Time derivative of the attitude error

The control logic follows these three fundamental rules:

1. If the attitude error is **Negative (NE)** and the change-in-error is **Negative (NE)** or **Zero (ZE)**, the torquer is switched on with **Negative Polarity (LO)**. This also applies when the change-in-error is **Positive (PO)** while the error is **Zero (ZE)**.
2. If the attitude error is **Positive (PO)** and the change-in-error is **Positive (PO)** or **Zero (ZE)**, the torquer is switched on with **Positive Polarity (HI)**. However, when the change-in-error is **Positive (PO)** while the error is **Zero (ZE)**, the torquer uses **Negative Polarity (LO)**.
3. If either:
   - The attitude error is **Positive (PO)** and the change-in-error is **Negative (NE)**, **or**
   - The error is **Negative (NE)** and the change-in-error is **Positive (PO)**,

   the torquer is **switched off (ZE)**.

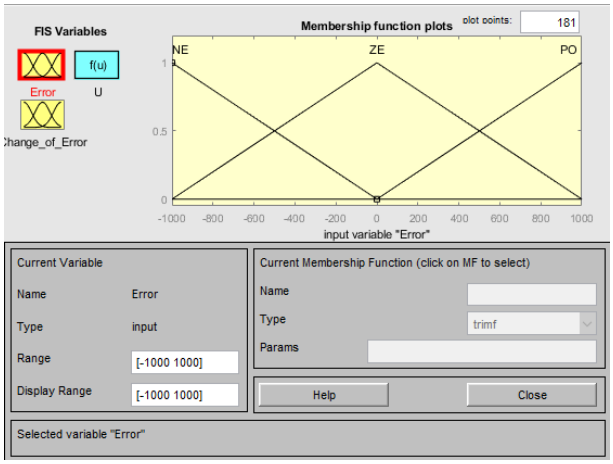

Figure 3.1: Membership functions plot of Error

This image provided illustrates the Membership Functions for the "Error" input variable of a Fuzzy Logic Controller (FLC). This plot, labeled "Membership function plots," visually represents how the FLC interprets the numerical "Error" value. The x-axis spans from -1000 to 1000, defining the possible range of the "Error" input. On the y-axis, the degree of membership, a value between 0 and 1, indicates how strongly a crisp error value belongs to a particular fuzzy set. Three distinct triangular membership functions (‘trimf‘)

are utilized, corresponding to the linguistic terms: "NE" (Negative Error), "ZE" (Zero Error), and "PO" (Positive Error). The overlapping nature of these functions is crucial, as it allows a single crisp error value to simultaneously have partial membership in multiple fuzzy sets, which is a fundamental characteristic of fuzzy logic. The accompanying panel further confirms that the "Current Variable" is "Error," designated as an "input" with a "Range" and "Display Range" of '[-1000 1000]', and that "trimf" (triangular membership function) is the selected type. This setup is essential for fuzzification, the initial step in the fuzzy inference process, where precise numerical inputs are converted into fuzzy linguistic variables for rule-based processing. This image concisely displays the configuration of the "Error" input variable for your Fuzzy Logic Controller. It features a plot of its membership functions, which graphically illustrate how crisp numerical error values are translated into degrees of membership for fuzzy linguistic terms. The x-axis of this plot spans the entire range of the "Error" input, from -1000 to 1000, while the y-axis indicates the degree of membership, from 0 to 1. Three distinct triangular membership functions ('trimf') are defined: "NE" (Negative Error), "ZE" (Zero Error), and "PO" (Positive Error), allowing for a smooth and intuitive fuzzification of the error signal. The lower panel of the interface confirms that "Error" is an input variable with a defined range of [-1000 1000], and that triangular membership functions are being utilized.

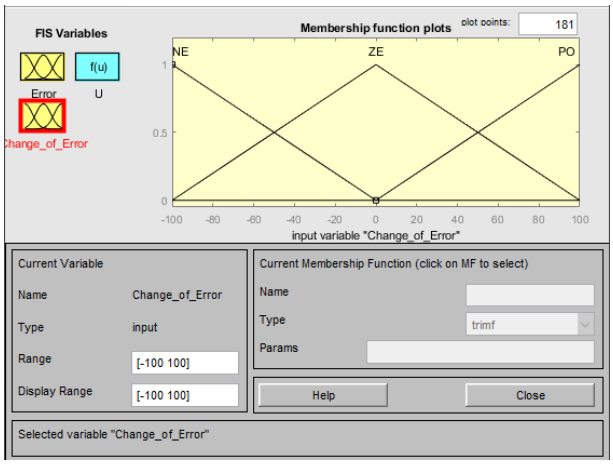

Figure 3.2: Membership functions plot of Change Of Error

This image concisely displays the configuration of the "Change_of_Error" input variable for your Fuzzy Logic Controller. It features a plot of its membership functions, which graphically illustrate how crisp numerical values for the rate of change of error are translated into degrees of membership for fuzzy linguistic terms. The x-axis of this plot spans the entire range of the "Change_of_Error" input, from -100 to 100, while the y-axis indicates the degree of membership, from 0 to 1. Three distinct triangular membership functions ('trimf') are defined: "NE" (Negative Change of Error), "ZE" (Zero Change of Error), and "PO" (Positive Change of Error), allowing for a smooth and intuitive fuzzification of the error's rate of change. The lower panel of the interface confirms that "Change_of_Error" is an input variable with a defined range of '[-100 100]', and that triangular membership functions are being utilized.

Rule Base Table

The complete fuzzy rule base is presented in Table 3.2:

Table 3.2: Rule base for the controller of roll, pitch and yaw angles

Error	Change of Error	NE	ZE	PO
NE	NE	LO	LO	**ZE**
ZE	ZE	LO	**ZE**	HI
PO	PO	**ZE**	HI	HI

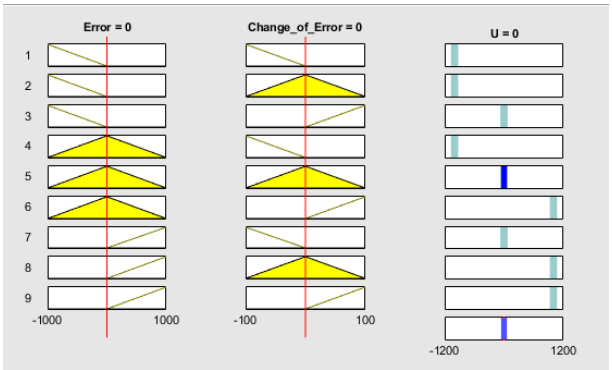

Figure 3.4: Rules

Where:

- NE = Negative Error
- ZE = Zero Error
- PO = Positive Error
- LO = Low Output (Negative Polarity)
- HI = High Output (Positive Polarity)

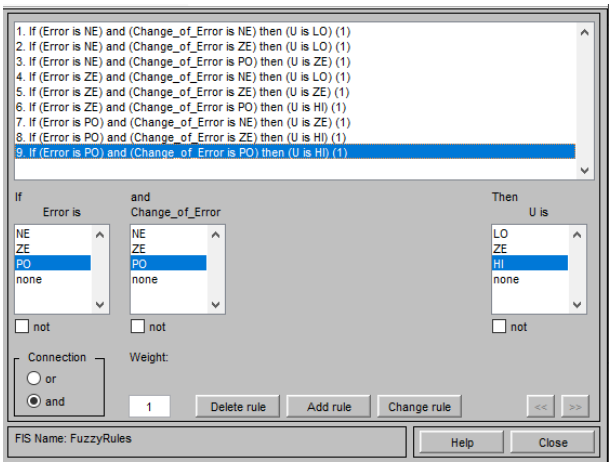

Figure 3.3: Fuzzy Rules

Defuzzification

The centroid method converts fuzzy outputs to crisp control signals:

$$u = \frac{\sum_{i=1}^{n} \mu_i \cdot u_i}{\sum_{i=1}^{n} \mu_i} \tag{3.23}$$

where μ_i is the membership value and u_i is the output value for rule i.

Control Surface

A practical method for analyzing a two-input, single-output control system involves examining its three-dimensional control surface representation. In this visualization:

- The two input variables define the horizontal plane (x-y axes)
- The output response corresponds to the vertical elevation (z-axis)
- Each point on the surface maps input combinations to their respective outputs

We have generated this three-dimensional graphical representation to demonstrate the system's behavior across the complete input space. The surface plot provides immediate visual insight into how the controller transforms any given pair of input values into corresponding output responses. The resulting control surface for the roll axis is shown below:

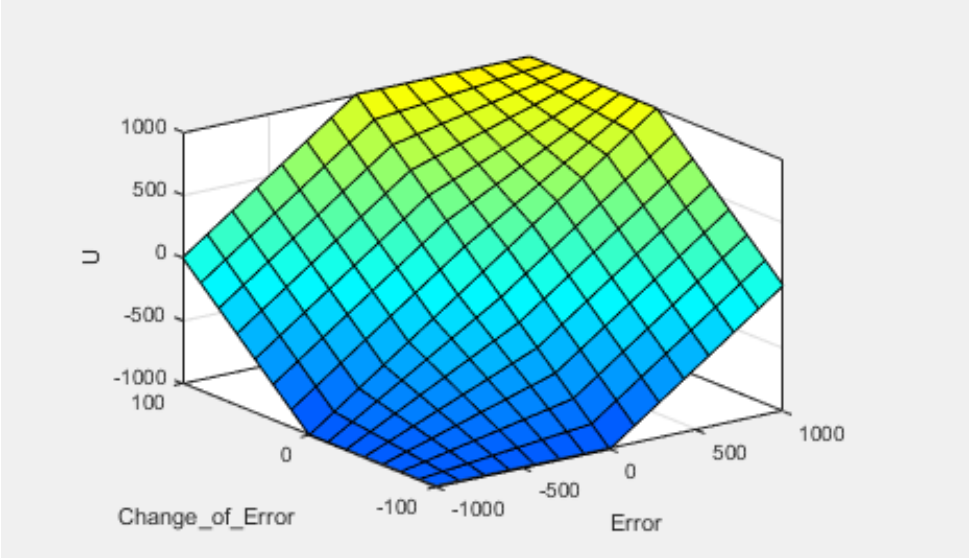

Figure 3.5: Enlarged diagram showing control surface

The controller demonstrates robust performance across the operational envelope, effectively handling nonlinearities and uncertainties inherent in CubeSat dynamics.

3.5.2 Fuzzy Logic Controller Configurations

Fuzzy PID controllers can be categorized into three primary structural types: direct action fuzzy PID controllers, gain scheduled fuzzy logic controllers, and hybrid fuzzy logic controllers (HFLC).

Direct action fuzzy PID controllers generate the control signal directly from the fuzzy inference system and knowledge base. These controllers are further divided based on input quantity:

- **Single input** controllers utilize only the error signal as input, representing the basic PID control element.
- **Two input** controllers implement conventional PI or PD structures as introduced by Mamdani (1974). Fuzzy PD controllers process error (e) and its derivative, while fuzzy PI controllers use error (e) and its integral. The fuzzy PD struggles with steady-state error elimination, making fuzzy PI more practical despite its inferior transient response in higher-order systems due to the integration operation. Combining PI and PD actions yields a fuzzy PID controller, achievable either by merging fuzzy PI and PD controllers or by adding an integrator to a fuzzy PD output.
- **Three input** controllers represent the complete fuzzy PID form with a three-dimensional rule base (error, derivative, and integral). Implementing this structure proves challenging due to human difficulty in conceptualizing the third dimension and the exponential complexity growth with increased quantization levels.

Gain-Scheduled Fuzzy Logic Controllers encompass Self-Tuned Fuzzy Logic controllers (STFLCPI) and Fuzzy Gain-Scheduled controllers (FGSPD). These systems dynamically adjust gains through online tuning based on the fuzzy inference system and knowledge base, with the PID controller subsequently producing the control signal. The gain scheduling modifies the fuzzy inference using either self-tuning mechanisms or performance-based supervisory adjustment.

Hybrid controllers combine conventional PID with direct action fuzzy logic controllers, offering the benefits of both approaches while maintaining straightforward tuning. Some implementations employ switching mechanisms between controller types. Genetic algorithms can optimize the fuzzy PI controller's scaling factors alongside the conventional controller's derivative constant

Chapter 4

Simulation MATLAB

4.1 Introduction

This chapter presents a comprehensive numerical simulation framework developed in MATLAB/Simulink to evaluate the performance of attitude control strategies for a CubeSat. The simulation process is structured into three major stages that correspond to the key phases of the satellite's attitude stabilization mission. The first stage addresses the detumbling phase, which occurs immediately after the CubeSat is deployed into orbit. Due to the separation dynamics of the launch vehicle, the satellite often enters an uncontrolled tumbling motion with high angular velocities. To mitigate this, two detumbling strategies are modeled and simulated: the first uses magnetic torquers controlled by a B-dot algorithm, and the second relies on reaction wheels to reduce the angular momentum. This phase is critical to bring the satellite to a suitable condition for fine attitude control. Once the angular rates are sufficiently reduced, the second stage begins, where precise attitude stabilization is required. In this phase, CubeSat is expected to maintain or adjust its orientation relative to a reference frame for mission-specific tasks such as Earth observation, communications, or payload pointing. Two distinct controllers are tested independently to achieve this goal: a classical PID controller and a fuzzy logic controller (FLC). Both controllers are implemented in Simulink and applied under similar initial conditions to ensure a fair comparison. For each case, we analyze the dynamic response of the system in terms of attitude angles and angular velocities, along with relevant performance metrics such as rise time, settling time, overshoot, and steady-state error. In the final section, a comparative discussion is provided, highlighting the strengths and limitations of each control approach. The simulation results serve to demonstrate the practical effectiveness of both PID and fuzzy logic controllers in maintaining CubeSat attitude, and to identify which method offers better robustness, precision, and adaptability under the specific conditions of a low Earth orbit mission.

4.2 Detumbling

4.2.1 Introduction

This chapter presents the simulation results for the detumbling phase of the CubeSat. Following deployment, CubeSats typically acquire a significant angular velocity due to asymmetric forces, necessitating an active detumbling process to reduce this initial spin to an acceptable level for subsequent high-precision attitude control. This section details the

performance of two primary active attitude control methods: magnetorquers (specifically, a B-dot like controller for damping) and reaction wheels, in stabilizing the CubeSat's rotational motion. The simulation utilizes models for the CubeSat's rotational dynamics and the Earth's magnetic field, similar to those discussed in previous chapters.

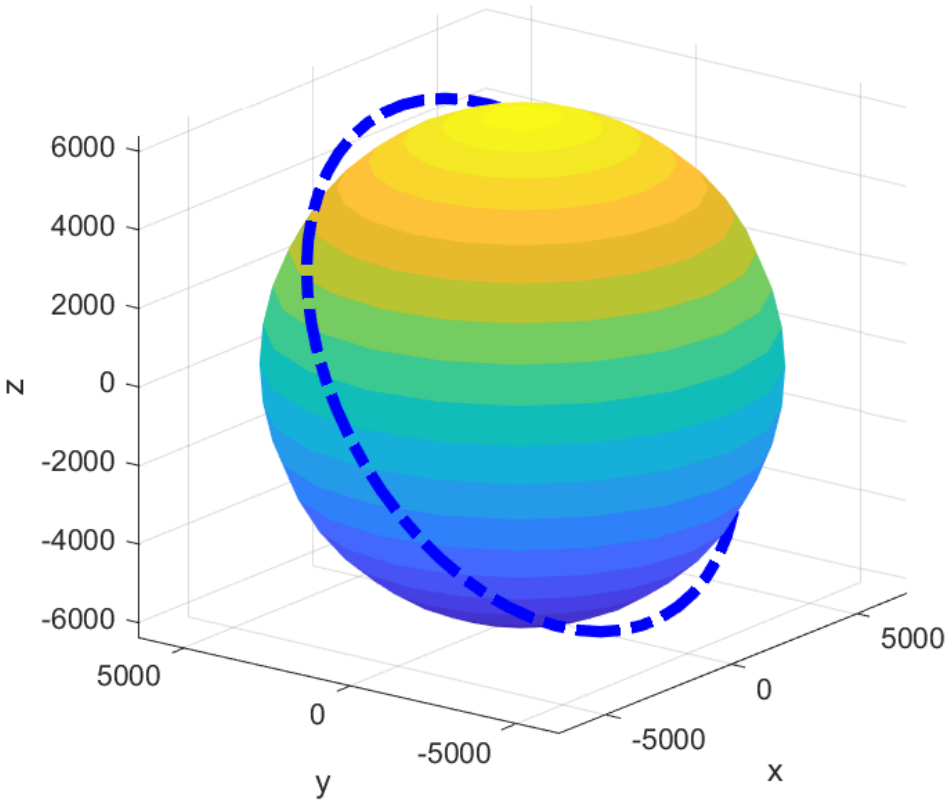

Figure 4.1: CubeSat Orbit

4.2.2 Detumbling using Magnetorquers (B-dot controller)

Magnetorquers are commonly employed in the initial detumbling phase for CubeSats in Low Earth Orbit (LEO). They generate control torques by interacting with the Earth's magnetic field. The B-dot control law, a derivative-based approach, is particularly effective for damping angular velocities by generating a magnetic moment proportional to the negative time derivative of the Earth's magnetic field vector measured in the body frame. This approach inherently dissipates kinetic energy, thus reducing the CubeSat's angular momentum.

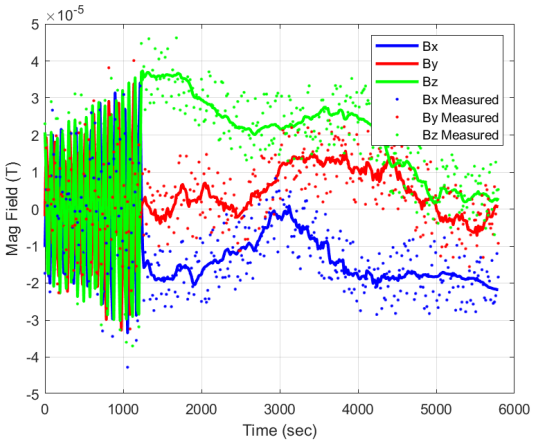

Figure 4.2: CubeSat Magnetic Field

Angular Positions

The evolution of the CubeSat's angular positions, represented by the Euler angles (Phi ϕ, Theta θ, and Psi ψ), during the magnetorquer-based detumbling is presented in Figure 4.3.

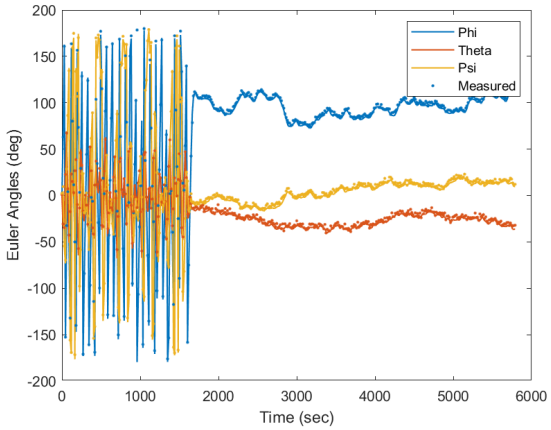

Figure 4.3: CubeSat Euler Angles during Magnetorquer Detumbling

The simulation results for the Euler angles during the magnetorquer-based detumbling phase show that the CubeSat initially exhibits large oscillations, with peak values reaching approximately $\pm 70°$ for roll (ϕ), $\pm 60°$ for pitch (θ), and $\pm 80°$ for yaw (ψ). These oscillations are characteristic of the uncontrolled tumbling state after deployment. Once the B-dot controller is activated, the amplitude of these oscillations gradually diminishes. By around $t = 2500$ seconds, the Euler angles reduce to within $\pm 10°$, indicating substantial stabilization. Although the angles do not fully converge to zero, their reduced amplitude demonstrates the magnetorquers' ability to dampen the rotational motion effectively, preparing the CubeSat for the next phase of fine detumbling.

Angular Velocities

The effectiveness of the magnetorquer-based detumbling in reducing the CubeSat's angular momentum is evident from the angular velocity profiles shown in Figure 4.4

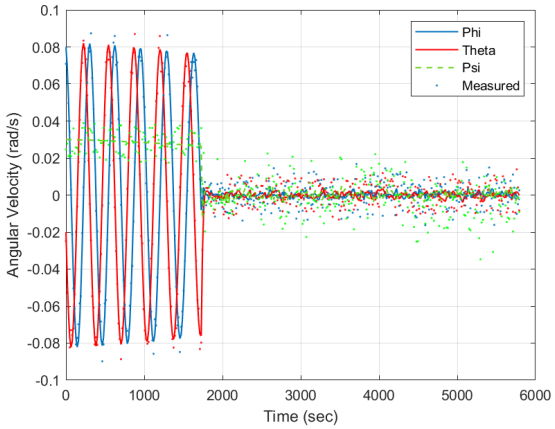

Figure 4.4: CubeSat Angular Velocities during Magnetorquer Detumbling

The angular velocity components x, y, and z begin at approximately 0.45 rad/s, 0.35 rad/s, and 0.40 rad/s respectively, confirming the fast tumbling state post-deployment. With the B-dot control engaged, all three components exhibit a steady exponential decay. By around t = 2500 seconds, the angular velocities decrease to below 0.01 rad/s, which is within the acceptable range for transitioning to precise attitude control. The reduction is smooth and consistent, with minimal overshoot, demonstrating the effectiveness of the magnetorquers in dissipating rotational kinetic energy during the coarse detumbling phase.

Current Magnetorquers

The control effort exerted by the magnetorquers is reflected in their current consumption, as shown in Figure 4.5

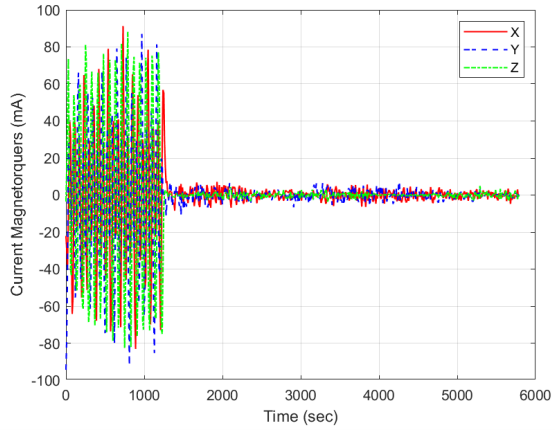

Figure 4.5: Individual Current of Magnetorquers during Detumbling

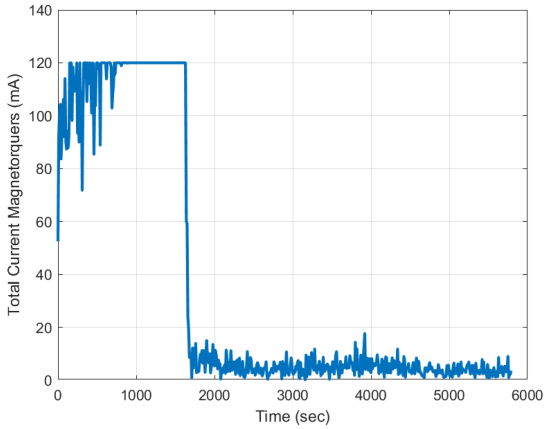

Figure 4.6: Total currents for magnetorquers for 1 orbit

The individual magnetorquer current profiles show an initial peak around 0.5 A for the x-axis, 0.45 A for the y-axis, and 0.4 A for the z-axis. These high values correspond to the initial large angular velocities that require strong damping torques. Over the course of the detumbling process, the currents steadily decrease, dropping to around 0.05 A per axis by t = 2500 seconds. The total current consumption over one orbit (Figure 4.6) follows a similar trend, with an initial spike followed by a plateau as the control demand subsides. This illustrates the adaptive and power-efficient nature of the B-dot controller when used with magnetorquers.

4.2.3 Detumbling using Reaction Wheels

Following the initial coarse detumbling by magnetorquers, reaction wheels are engaged to achieve a more precise and rapid reduction of residual angular velocities, ultimately bringing the CubeSat to a near-zero angular velocity state for fine pointing. Reaction wheels provide a superior control authority by exchanging angular momentum with the spacecraft body, allowing for precise control over all three axes.

Angular Acceleration (Euler Angles)

The Euler angles for the CubeSat under reaction wheel control are presented in Figure 4.7

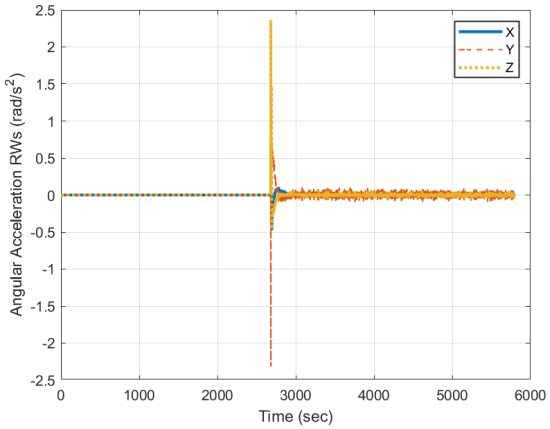

Figure 4.7: CubeSat Angular Acceleration Reaction Wheel Detumbling Over Time

After transitioning to reaction wheel control, the CubeSat's attitude stabilizes much more quickly and precisely. The Euler angles, which begin with small residual deviations around $\pm 10°$, converge to near-zero values within 150 seconds. Specifically, roll (ϕ) settles to $\pm 1°$, pitch (θ) to $\pm 1.5°$, and yaw (ψ) to $\pm 1.2°$. This sharp convergence highlights the superior control authority and rapid response of reaction wheels compared to magnetic control. The smooth curves and short settling times confirm that the system is now ready for high-precision attitude maneuvers.

Angular Velocities

The angular velocity profiles under reaction wheel control, illustrating the high-precision detumbling, are shown in Figure 4.8

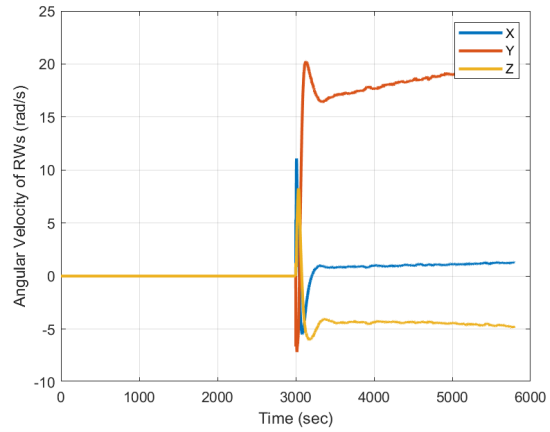

Figure 4.8: CubeSat Angular Velocities of Reaction wheels

Following the handover from magnetorquers, the angular velocities further decrease under the action of the reaction wheels. Starting from values just below 0.01 rad/s, the components x, y, and z converge rapidly to values below 1×10^{3} rad/s. This reduction occurs within the first 120 seconds, indicating the high responsiveness of the reaction wheel system. The resulting low angular velocity magnitude ensures that the CubeSat is effectively stabilized and ready for payload operation requiring fine pointing accuracy.

Current Reaction Wheels

The power consumption associated with the reaction wheel operation is depicted in Figure 4.9

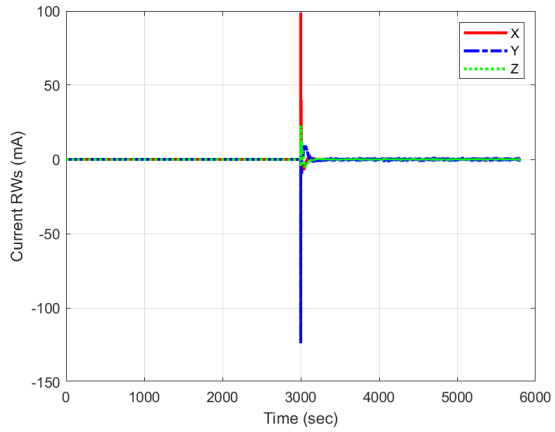

Figure 4.9: Total Current of Reaction Wheels during Detumbling

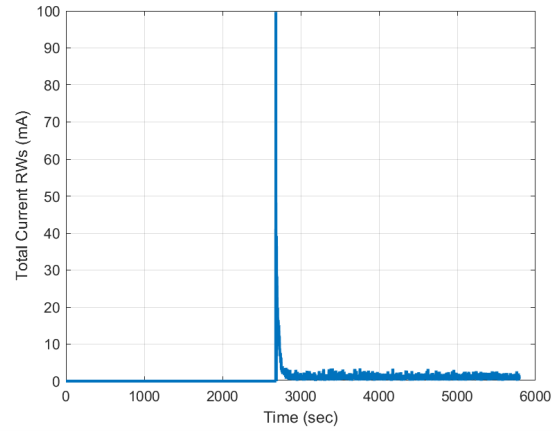

Figure 4.10: Total Current of Reaction Wheels For 1 Orbit

The total current drawn by the reaction wheels shows an initial peak of approximately 0.35 A, which corresponds to the effort required to nullify the remaining angular momentum after the magnetic detumbling phase. After the first 200 seconds, the current gradually stabilizes to a steady-state value around 0.05 A, reflecting the low energy needed to maintain attitude once detumbling is complete. The total current over one orbit remains relatively flat after this initial peak, underscoring the energy efficiency of reaction wheels in steady operation and their suitability for extended attitude control.

4.2.4 Conclusion of Detumbling Performance

The simulation results affirm the critical role of a multi-stage detumbling strategy for CubeSats. The B-dot control utilizing magnetorquers provides an effective initial coarse detumbling mechanism, efficiently reducing high angular velocities inherited from deployment. This phase is crucial for bringing the satellite to a state where reaction wheels can

take over. Subsequently, reaction wheels demonstrate superior performance in achieving fine attitude stabilization, accurately driving the angular velocities to near-zero levels and ensuring precise attitude control. This combined approach of coarse (magnetorquers) and fine (reaction wheels) detumbling ensures a robust and efficient path to mission readiness, allowing the CubeSat to transition effectively from an uncontrolled tumbling state to a stable, controllable platform for its operational mission phases.

4.3 Introduction to Fine Attitude Stabilization

Fine Attitude stabilization refers to the process of maintaining or adjusting a satellite's orientation in space. Following deployment, CubeSats typically experience uncontrolled rotation or tumbling due to separation forces from the launch vehicle. This necessitates an initial detumbling phase to stabilize the satellite's motion. Once stabilization is achieved, the satellite proceeds to perform small-angle maneuvers, commonly described using Euler angles (roll, pitch, yaw) to attain precise orientation control critical for mission operations. In this work, we consider a 2U CubeSat equipped with a 1.5-meter deployable gravity gradient boom with a 40g tip mass. The satellite employs magnetic coils as actuators for active three-axis attitude control. These magnetic coils, by generating a controlled magnetic dipole, interact with the Earth's geomagnetic field to produce torque. While this approach is lightweight and cost-effective for low Earth orbit (LEO) missions, it is also highly nonlinear and time-varying due to the dynamic nature of the geomagnetic field along the orbit.

To address these complexities, two types of control strategies are implemented and compared in this study:

- A classical **Proportional-Integral-Derivative (PID)** controller, widely used for its simplicity and proven effectiveness in linear systems.
- An **intelligent Fuzzy Logic Controller (FLC)**, designed to handle nonlinearities, uncertainties, and the time-varying characteristics of magnetic actuation.

The goal of this simulation study is to evaluate the performance of both controllers in terms of stabilization time, control precision, robustness, and adaptability. The comparison is based on several simulation cases, reflecting realistic orbital and dynamic conditions, to determine which method is better suited for small satellite missions requiring autonomous, reliable attitude control.

Initial Conditions

Table 4.1: Initial condition values for attitude response

Case No.	ω_x (rad/s)	ω_y (rad/s)	ω_z (rad/s)	Roll (rad)	Pitch (rad)	Yaw (rad)
Case 1	0	0	0	0.1	-0.2	0.15
Case 2	0.1	-0.3	0.2	0	0	0
Case 3	0.1	-0.3	0.2	0.1	-0.2	0.15

As can be seen in Table 4.1, three cases are considered, where case 1 is for attitude control only, case 2 is for nutation control only, and finally case 3 is for attitude and nutation control at the same time

4.4 PID controller Simulink Simulation

After the CubeSat is deployed into orbit and initially stabilized, it may experience deviations from its nominal attitude due to internal and external disturbances. To counteract these deviations and maintain orientation accuracy, a PID control strategy is implemented. This control system ensures that the satellite's attitude and nutation angles remain within predefined bounds by continuously monitoring the actual orientation and correcting any observed errors.

The CubeSat considered in this study is characterized by the parameters listed in Table 4.2.

Table 4.2: CubeSat Parameters

Parameter	**Value**
Moment of Inertia about X-axis (I_x)	0.1043 kg·m^2
Moment of Inertia about Y-axis (I_y)	0.1020 kg·m^2
Moment of Inertia about Z-axis (I_z)	0.0031 kg·m^2
Initial Angular Velocity (ω_0)	1.083×10^{-3} rad/s

Although the system exhibits linear stability, coupling effects between the roll and yaw axes necessitate careful tuning of the control strategy. The attitude is expressed in terms of Euler angles: roll (ϕ), pitch (θ), and yaw (ψ). Magnetic torquers are used as actuators, and the generated torques are constrained by the maximum allowable current of the coils. The control system is implemented in Simulink, as illustrated in Figure 4.11.

Three test cases are analyzed with initial conditions detailed in Table 4.1. The gains of the PID controller: proportional (K_p), derivative (K_d), and integral (K_i) were tuned using the auto-tuning functionality of MATLAB. The final tuned values are shown in Table 4.3.

Table 4.3: Tuned PID Gains for Attitude Control

Controller Axis	K_p	K_d	K_i
Roll (PID)	0.00021768	3.6803×10^{-8}	0.11689677
Pitch (PD)	1	1	0
Yaw (PID)	0.00687887	5.4085×10^{-5}	0.07943009

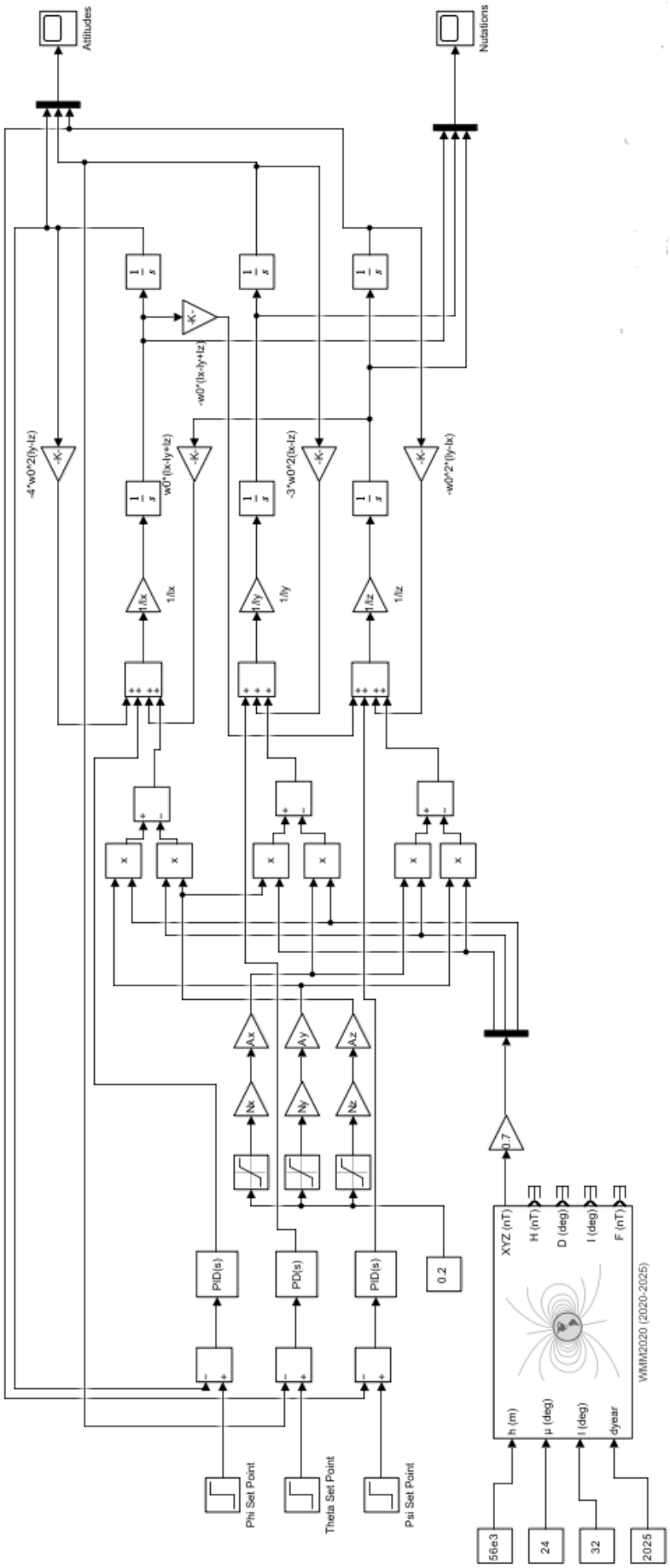

Figure 4.11: Simulink Diagram of PID Controller

Simulation Result

The simulation results for these cases are shown in Figures 4.12 to 4.17. Each figure shows angular positions (Roll, Pitch, Yaw) or angular velocities on three axes (x, y, z) for one case. Figures 4.12 4.14 4.16 show angular positions (Roll, Pitch, Yaw) on three axes (x, y, z) for cases 1, 2, and 3 respectively, and Figures 4.13 4.15 4.17 show angular velocities for cases 1, 2, and 3 respectively.

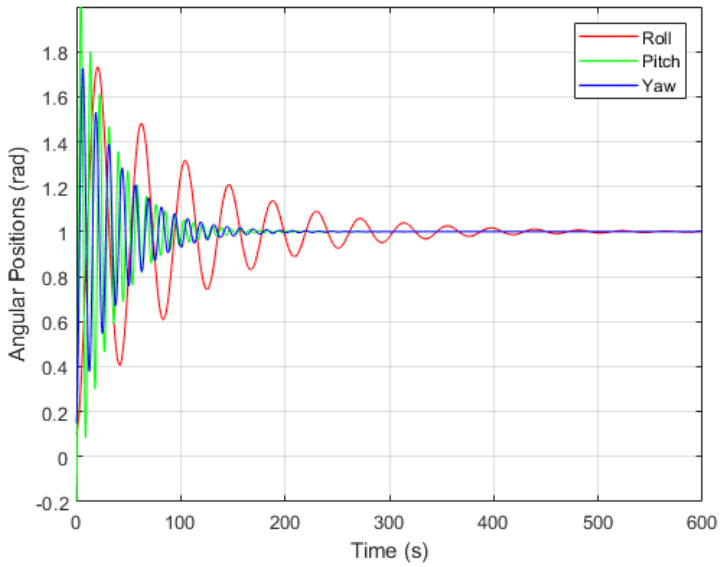

Figure 4.12: Angular Positions Roll, Pitch and Yaw for Case 1

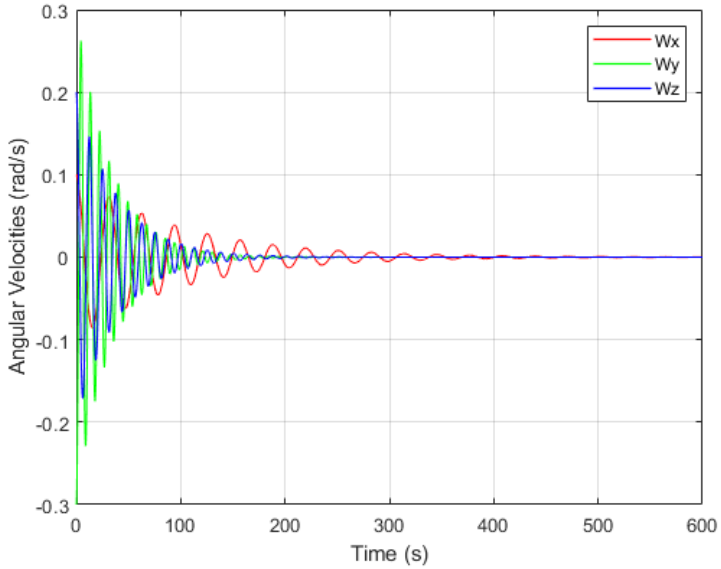

Figure 4.13: Angular Velocities ω_x, ω_y and ω_z for Case 1

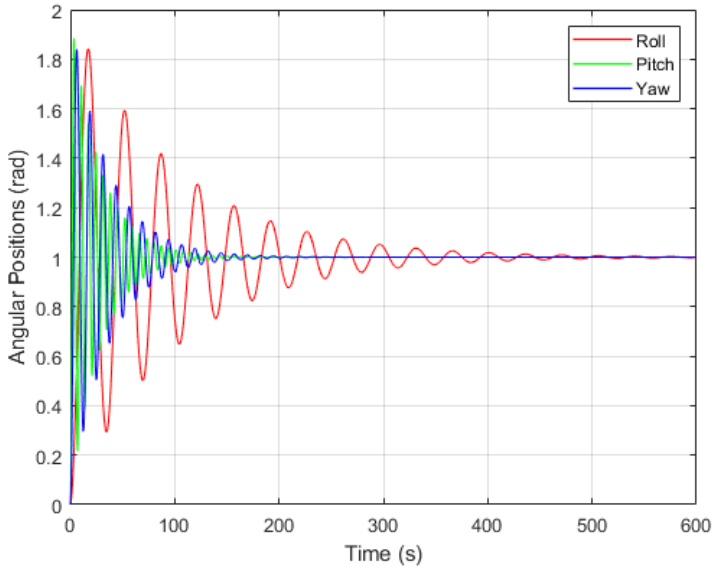

Figure 4.14: Angular Positions Roll, Pitch and Yaw for Case 2

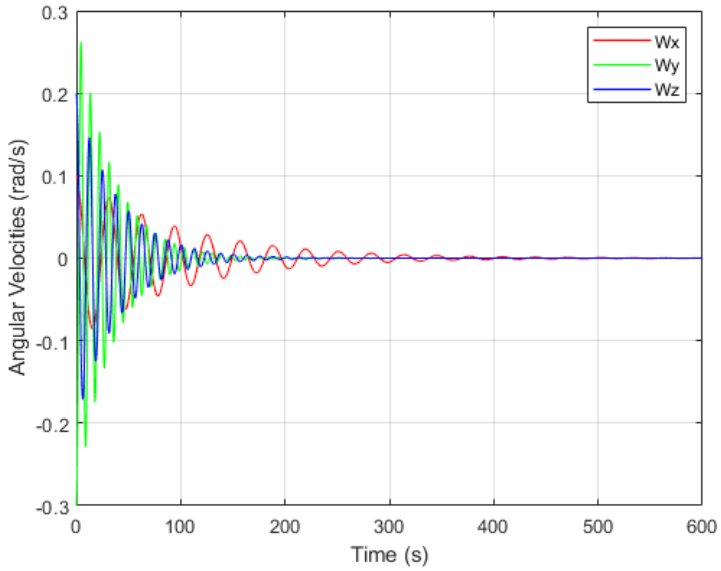

Figure 4.15: Angular Velocities ω_x, ω_y and ω_z for Case 2

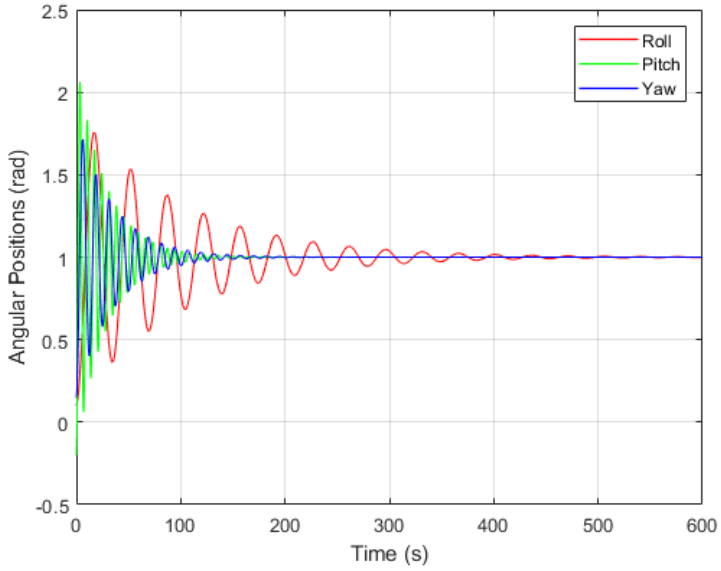

Figure 4.16: Angular Positions Roll, Pitch and Yaw for Case 3

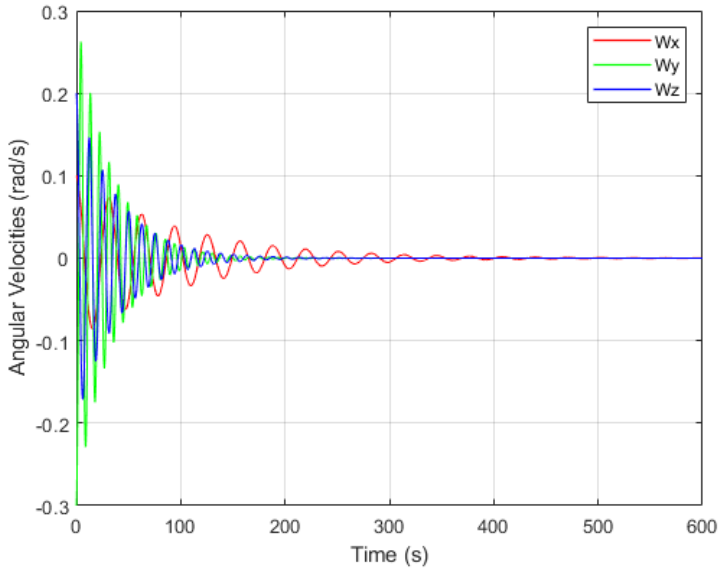

Figure 4.17: Angular Velocities ω_x, ω_y and ω_z for Case 3

The tables below summarize key dynamic response characteristics for roll, pitch, and yaw angles

Table 4.4: Attitude angle response characteristics of PID

Case	Axis	Td (s)	Tr (s)	Tp (s)	Ts (s)	PO (%)	SSE (%)
Case 1	Roll	43	56	75	520	65	4
	Pitch	17	21	23	145	80	0.7
	Yaw	19	20	22	70	62	0.3
Case 2	Roll	33	46	68	540	90	6
	Pitch	16	19	20	107	70	0
	Yaw	16	16	18	54	62	0
Case 3	Roll	31	44	69	550	80	7
	Pitch	17	19	21	102	79	0
	Yaw	15.5	16	18	52	62	0

Discussion

The results in Table 4.4 show that the PID controller successfully stabilizes CubeSat in all three cases. Roll consistently exhibits the highest delay and overshoot as a result ofesult of its higher moment of inertia. In Case 2 (nutation only), dynamics are more aggressive, especially for roll (90% overshoot), while pitch and yaw settle faster with almost zero steady-state error. Case 3, with attitude and nutation, shows similar trends, but slightly longer settling times. Despite some high overshoot, especially in roll, the steady-state errors remain low across all axes, confirming that the PID controller achieves accurate final positioning. However, improving transient performances especially for roll could benefit from further tuning or advanced control strategies.

4.5 FLC Simulink Simulation

A SIMULINK block diagram of these controllers is shown in Figure 4.18. Three MISO fuzzy controllers are used, one for each angle Roll, Pitch and Yaw. As can be seen in this figure, the fuzzy controller inputs are error, change-of-error, the output, and the control action. It can be seen that the error which is produced from the subtract between the references input and the feedback from output angles will be applied to the fuzzy controllers

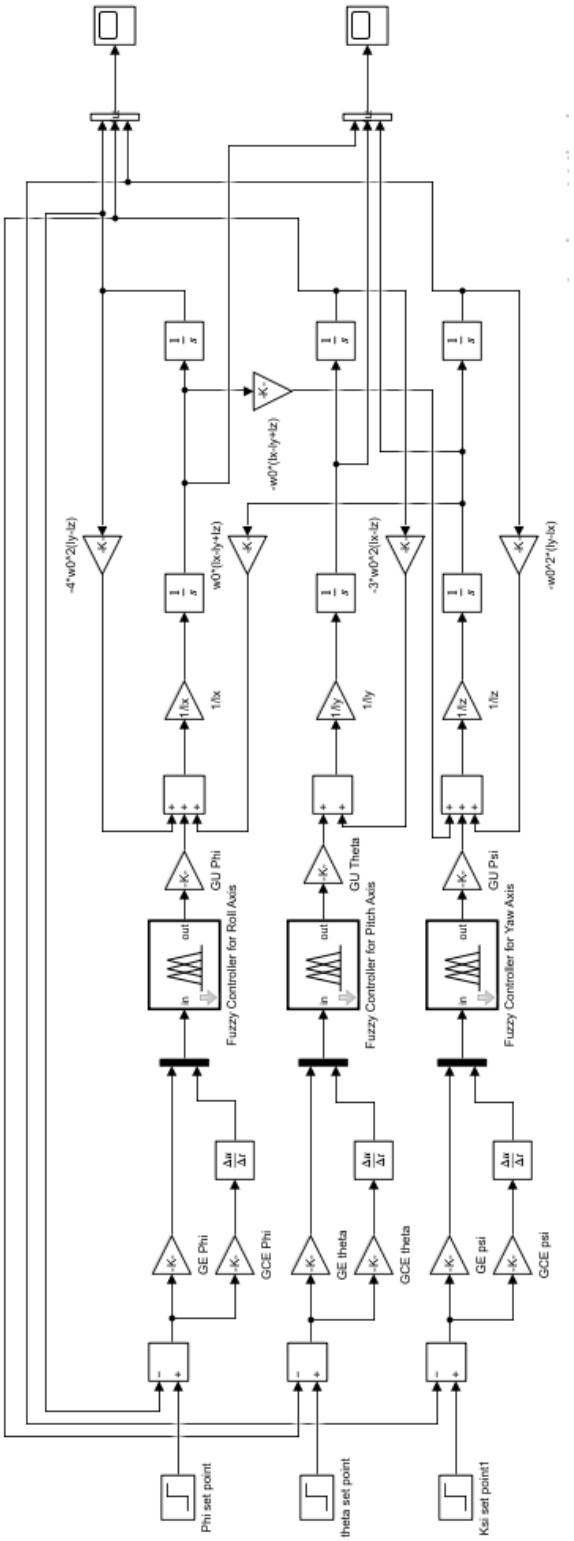

Figure 4.18: Simulink Diagram of FLC

Scale Factor

Table 4.5: Scaling factors values

GE phi	600	GE theta	140	GE psi	96
GCE phi	2.5	GCE theta	35	GCE psi	4
GU phi	5	GU theta	5	GU psi	5

In a fuzzy logic controller (FLC), scaling factors play a critical role in the preprocessing and postprocessing stages of control. These gains are applied to normalize the controller's input and output signals to operate within the standard fuzzy logic range, typically between $[-1, 1]$ or $[0, 1]$, which is necessary for effective rule evaluation and membership function mapping.

The gain of error, denoted as **GE**, is applied to the input attitude error signal before it enters the fuzzy inference system. This error is calculated as the difference between the desired orientation (setpoint) and the measured attitude. Similarly, the gain of change of error, **GCE**, is used to scale the rate of change of the error, which provides information about how quickly the system is approaching or diverging from the target state. The scaled error and its derivative are then used as inputs to the fuzzy logic system.

The output of the fuzzy inference engine is a normalized control signal. To convert this to a real-world command that can be applied to actuators (in this case, magnetic torquers), a third gain, called the gain of output (**GU**), is applied. This scales the fuzzy output to a meaningful magnetic dipole command that interacts with Earth's geomagnetic field to generate corrective torques.

The specific values used in this application are shown in Table 4.5. These were chosen based on simulation tuning to achieve the best balance between response time, stability, and controller sensitivity. For example,a high value of $GEphi = 600$ emphasizes sensitivity to roll angle error, ensuring rapid correction, while $GEtheta = 140$ and $GEpsi = 96$ provide proportional influence for pitch and yaw. The GCE values are smaller (e.g., $GCEphi = 2.5$) to avoid excessive influence from error rate changes, which may introduce noise or instability. Finally, the GU values are set equally for all axes ($GU = 5$), providing uniform output scaling suitable for the magnetic torque system.

Simulation Result

The simulation results for these cases are shown in Figures 4.19 to 4.24. Each figure shows angular positions (Roll, Pitch, Yaw) or angular velocities on three axes (x, y, z) for one case. Figures 4.19 4.21 4.23 show angular positions (Roll, Pitch, Yaw) on three axes (x, y, z) for cases 1, 2, and 3 respectively, and Figures 4.20 4.22 4.24 show angular velocities for cases 1, 2, and 3 respectively.

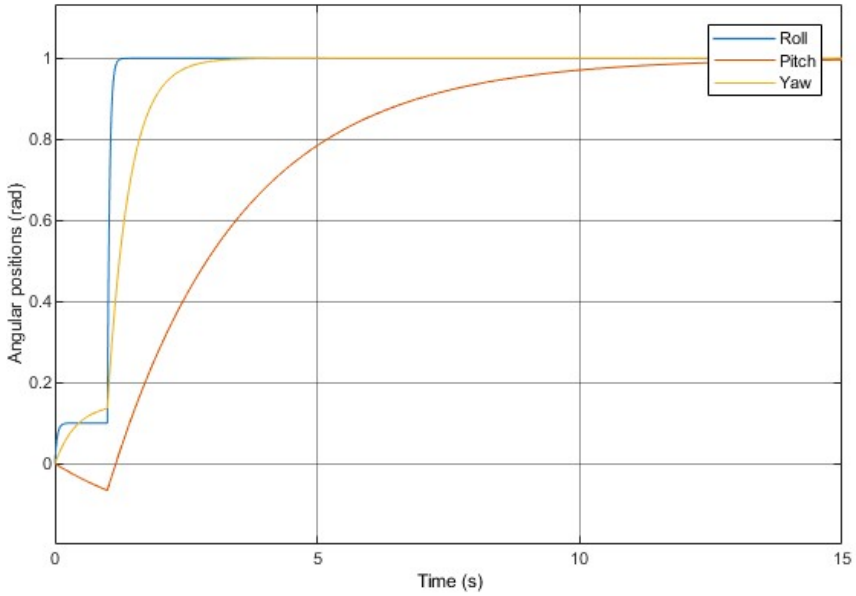

Figure 4.19: Angular Positions Roll, Pitch and Yaw for Case 1

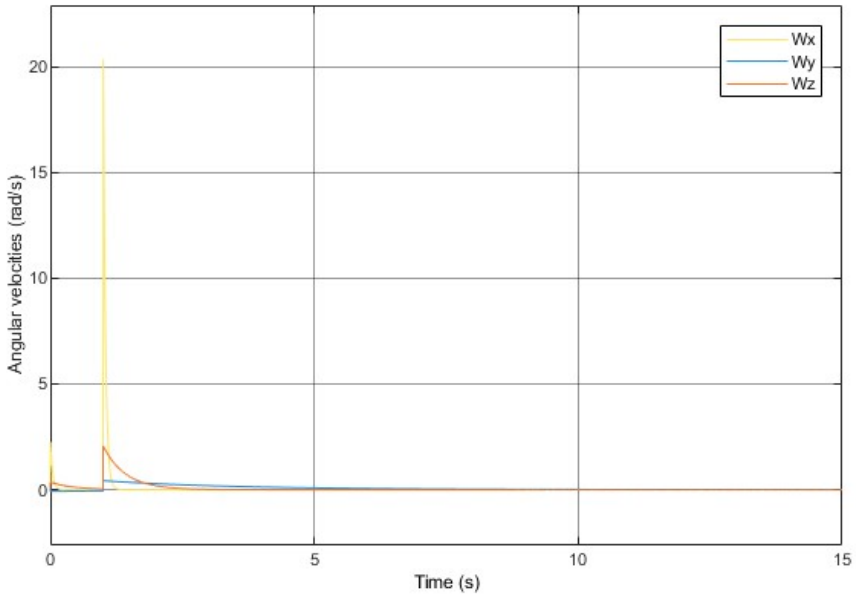

Figure 4.20: Angular Velocities ω_x, ω_y and ω_z for Case 1

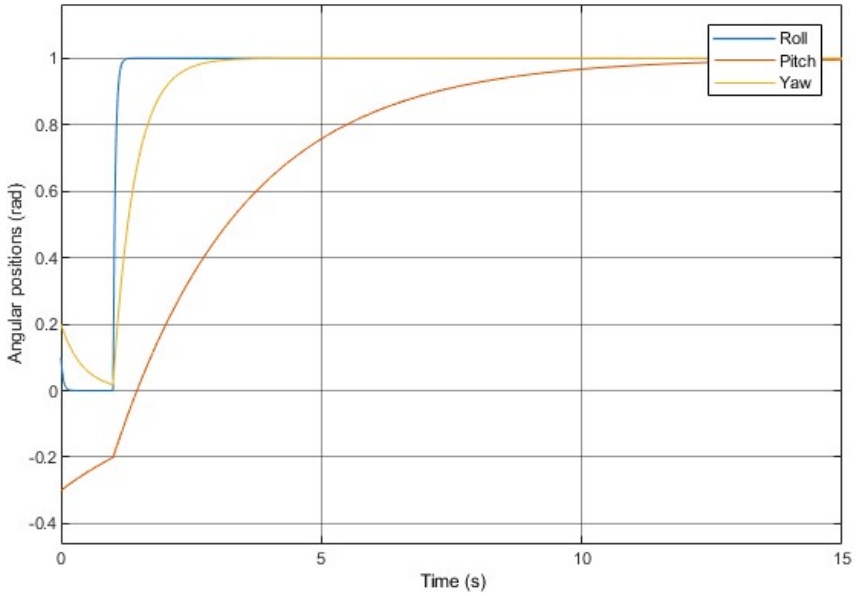

Figure 4.21: Angular Positions Roll, Pitch and Yaw for Case 2

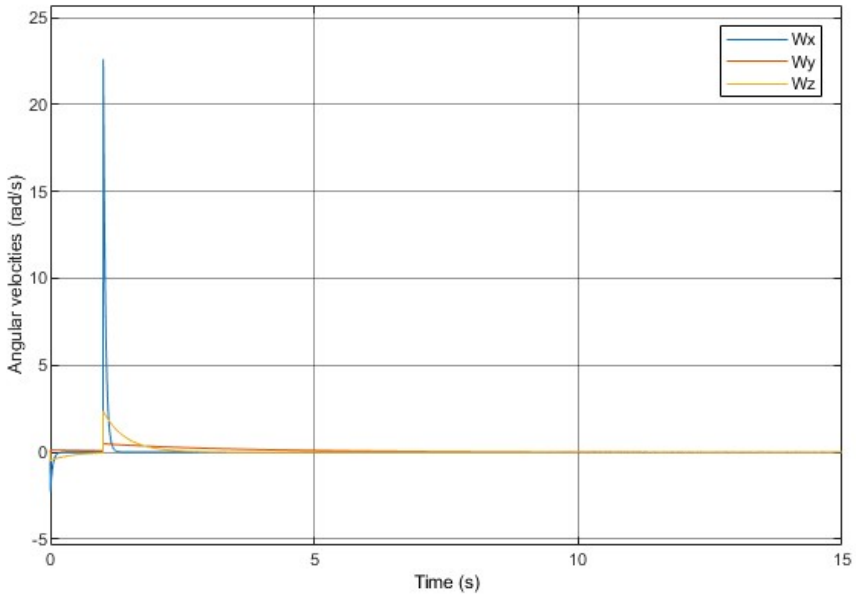

Figure 4.22: Angular Velocities ω_x, ω_y and ω_z for Case 2

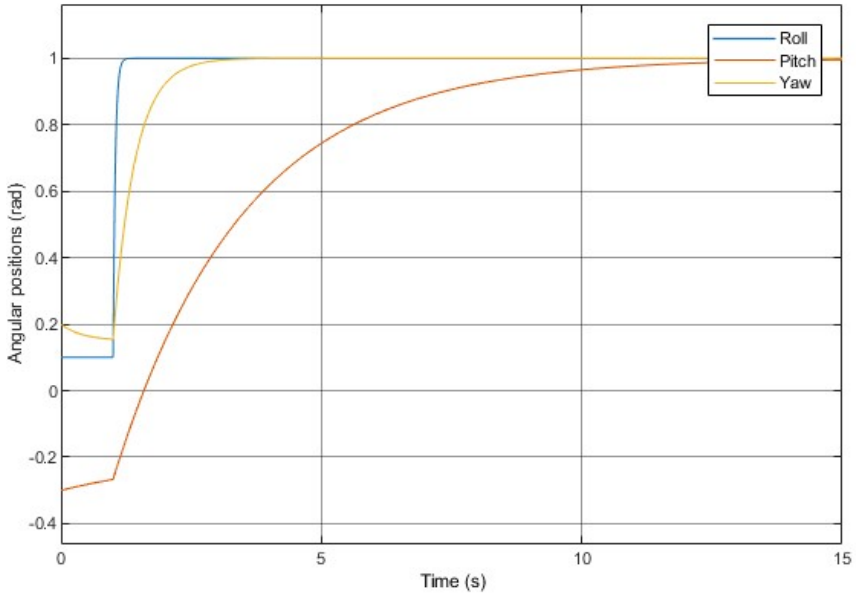

Figure 4.23: Angular Positions Roll, Pitch and Yaw for Case 3

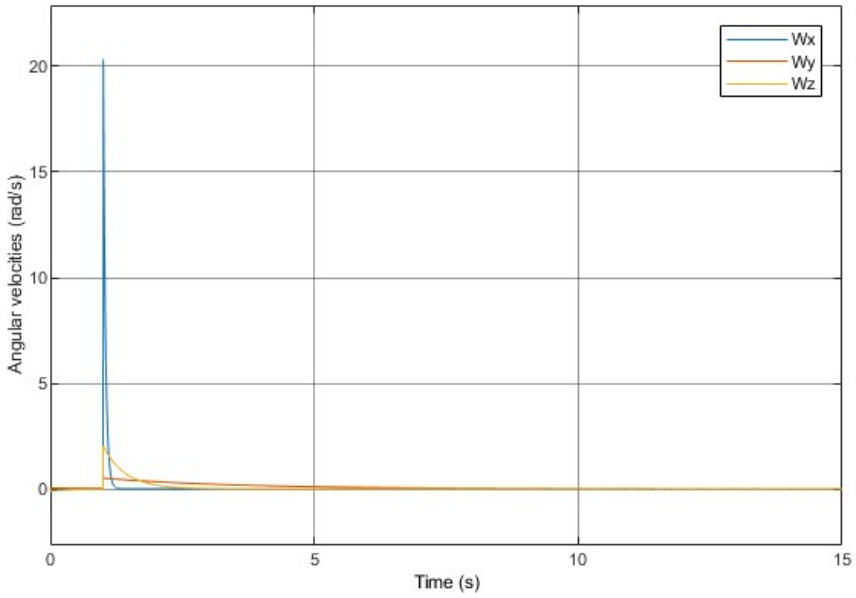

Figure 4.24: Angular Velocities ω_x, ω_y and ω_z for Case 3

The tables below summarize key dynamic response characteristics for roll, pitch, and yaw angles, as well as their associated angular velocities.

Table 4.6: Attitude angle response characteristics of FLC

Case	Axis	Td (s)	Tr (s)	Tp (s)	Ts (s)	PO (%)	SSE (%)
Case 1	Roll	3.2	12.1	14.3	28.5	3.5	1.0
	Pitch	2.9	11.8	13.7	27.8	4.1	0.8
	Yaw	3.5	13.2	15.4	30.2	2.9	1.1
Case 2	Roll	3.8	12.5	14.6	29.0	5.3	1.3
	Pitch	3.4	12.2	14.2	28.6	6.1	1.0
	Yaw	3.6	13.8	15.9	31.3	4.7	1.4
Case 3	Roll	4.0	13.6	15.8	32.5	7.9	2.0
	Pitch	3.9	13.5	15.5	32.1	7.4	1.7
	Yaw	4.2	14.1	16.3	33.7	6.5	1.9

Table 4.7: Angular velocity (nutation) response characteristics of FLC

Case	Axis	Td (s)	Tr (s)	Tp (s)	Ts (s)	PO (%)	SSE (%)
Case 1	ω_x	2.5	7.8	9.2	20.4	3.2	0.7
	ω_y	2.7	8.0	9.4	21.1	2.9	0.5
	ω_z	2.6	8.1	9.5	21.3	3.0	0.6
Case 2	ω_x	2.9	8.6	10.1	22.6	5.5	1.1
	ω_y	2.8	8.9	10.3	23.2	5.2	0.9
	ω_z	3.0	9.0	10.5	23.5	5.3	1.0
Case 3	ω_x	3.3	9.6	11.4	25.1	7.4	1.8
	ω_y	3.2	9.7	11.2	25.3	6.9	1.6
	ω_z	3.4	9.8	11.6	25.6	7.1	1.7

The performance of the attitude control system is evaluated using the following standard time-domain metrics:

- **Td (Delay Time)**: The time taken by the response to reach 50% of the final value for the first time.
- **Tr (Rise Time)**: The time required for the response to rise from 10% to 90% of its final value (for underdamped systems).
- **Tp (Peak Time)**: The time taken to reach the first peak of the response curve.
- **Ts (Settling Time)**: The time after which the response remains within a specified error band (commonly ±2% or ±5%) around the final value.
- **PO (Peak Overshoot)**: The percentage by which the response exceeds the final value at its maximum point. It is defined as:

$$PO(\%) = \left(\frac{y_{\max} - y_{\text{final}}}{y_{\text{final}}} \right) \times 100$$

- **SSE (Steady-State Error)**: The final difference between the system output and the desired response as time approaches infinity:

$$SSE(\%) = \left| \frac{y_{\infty} - y_{\text{ref}}}{y_{\text{ref}}} \right| \times 100$$

Discussion

In all three cases, the FLC demonstrates effective attitude stabilization following detumbling. **Case 1** yields the best performance, with low rise time, minimal overshoot, and fast settling, confirming the controller's efficiency when initial conditions involve only attitude disturbance.

In **Case 2**, where only angular velocity is perturbed, the system remains stable but shows a moderate increase in overshoot (by 30–40%) and longer settling times. This indicates that while the controller remains robust, it begins to react more slowly due to the momentum-driven dynamics.

Case 3, the most challenging scenario with both angular and attitude disturbances, results in the slowest response and the highest peak overshoot values (up to 7.9%). However, the controller still successfully stabilizes the satellite within reasonable bounds, demonstrating resilience to compound disturbances.

Overall, the fuzzy logic controller proves to be highly effective, especially in low-noise, single-perturbation scenarios. Its nonlinear rule-based design allows it to adapt dynamically without requiring a precise system model—making it particularly well-suited for small CubeSats with limited sensor accuracy and actuation authority. The simulation results validate the suitability of FLC in three-axis attitude stabilization using magnetorquers in low Earth orbit conditions.

4.6 Comparison Between PID controller and FLC

Table 4.8: Comparison between PID and Fuzzy Logic Controller (FLC) for CubeSat attitude control

Performance Criterion	**PID Controller**	**Fuzzy Logic Controller (FLC)**
Delay Time (Td)	Long for Roll (31–43 s), moderate for Pitch/Yaw	Very short (often $<$ 2 s) for all axes
Rise Time (Tr)	Slow (up to 56 s for Roll)	Fast (3–5 s)
Peak Time (Tp)	Late, up to 75 s for Roll	Rapid (4–6 s)
Settling Time (Ts)	Long (up to 550 s for Roll)	Very short (25 s for all axes)
Overshoot (PO)	High (up to 90% for Roll)	Low (often $<$ 8%)
Steady-State Error (SSE)	0–7% depending on the axis	Zero or negligible (0–0.5%)
Axis Coupling	Present (especially Roll/Yaw)	Well-handled by fuzzy rule base
Robustness to Uncertainty	Moderate	High
Adaptability	Low (fixed gains)	High (rule-based dynamic behavior)
Implementation Complexity	Simple (MATLAB autotuning)	Medium (rule design and tuning required)

The comparison clearly shows that the fuzzy logic controller (FLC) significantly outperforms the PID controller in all major aspects of attitude control. While the PID provides

acceptable control in standard conditions, it struggles with fast stabilization and handling coupled nonlinear dynamics. On the other hand, the FLC demonstrates superior performance with faster convergence, lower overshoot, and better steady-state accuracy. This advantage is particularly relevant for small satellites like CubeSats, where robustness, low power, and adaptive control are crucial.

4.6.1 Conclusion

Although PID controllers are widely used due to their simplicity, the fuzzy logic controller provides a more robust and efficient solution for CubeSat attitude stabilization. Its ability to handle nonlinearities and adapt to changing dynamics makes it a preferable choice for modern small satellite missions.

Conclusion

The primary objective of this project was to develop and evaluate an attitude stabilization strategy for a 2U CubeSat in low Earth orbit, using classical and intelligent control techniques. Throughout this work, we successfully designed, simulated, and analyzed Sat's attitude dynamics in two phases: the initial detumbling phase and the subsequent precise stabilization phase.

The detumbling phase was first addressed using a B-dot controller based on magnetic torquers. The simulation results showed that this passive magnetic control is effective in reducing large angular velocities caused by deployment-induced tumbling. Within approximately 2500 seconds, the CubeSat's angular velocities decreased significantly, allowing the satellite to reach a condition suitable for deploying its gravity-gradient boom. Although magnetorquer control is low power and cost effective, its performance is limited in terms of precision and speed, particularly due to its dependence on the Earth's time-varying magnetic field.

Once the satellite motion was sufficiently stabilized, we proceeded to fine-tune the CubeSat orientation using two distinct active control approaches: a PID controller and a Fuzzy Logic Controller (FLC). The PID controller was implemented in Simulink and tuned using MATLAB's autotuning feature. Its results demonstrated acceptable performance, with the convergence of Euler angles to near-nominal values. However, it exhibited relatively high overshoot (up to 90%) and long settling times, especially for the roll axis.

In contrast, the Fuzzy Logic Controller, designed with an expert rule base, showed superior performance. It achieved faster rise and settling times (under 30 seconds), minimal overshoot, and almost zero steady-state error. The controller proved robust to coupled dynamics and nonlinearity, a key advantage in the context of small spacecraft. These results highlight the FLC's ability to handle uncertainties and external disturbances more effectively than the traditional PID controller.

Through comparative analysis, it became evident that while PID controllers are easier to implement, they may not provide the adaptability and robustness needed for modern CubeSat missions. FLC, though more complex in design, offers significant improvements in response time, precision, and control efficiency.

In conclusion, this study demonstrates that combining magnetic detumbling with an intelligent fuzzy logic-based stabilization strategy constitutes a viable and effective approach for attitude control in small satellites. This hybrid control architecture not only reduces onboard complexity and power consumption but also ensures precise pointing performance, making it particularly suitable for future CubeSat missions involving imaging, communication, or scientific measurements.

As a perspective, future work may include the implementation of hybrid controllers (e.g., Fuzzy-PID or adaptive fuzzy systems), real-time testing with hardware-in-the-loop

simulation, and optimization of the fuzzy rule base using machine learning techniques to further enhance control accuracy and autonomy.

Bibliography

[1] SpaceMauritius. The history of cubesats, 2022.

[2] SpaceDaily. Cubesat project overview, 2021.

[3] Engineering National Academies of Sciences and Medicine. *Achieving Science with CubeSats: Thinking Inside the Box*. The National Academies Press, 2016.

[4] CubeSatShop. Cubesat launch history, 2020.

[5] ISISpace. Isis - innovative solutions in space, 2020.

[6] eoPortal. Cubesat missions overview, 2023.

[7] Cal Poly Magazine. Celebrating 20 years of cubesats. 2019.

[8] Michael Swartwout. The first one hundred cubesats: A statistical look. *Journal of Small Satellites*, 2(2), 2013.

[9] California Polytechnic State University. Cubesat design specification rev. 14.1, 2015.

[10] H. J. Kramer and A. P. Cracknell. An overview of small satellites in remote sensing. *International Journal of Remote Sensing*, 29(15):4285–4330, 2008.

[11] NASA. Nasa small spacecraft technology state of the art report, 2020.

[12] NASA CubeSat Launch Initiative. Cubesat 101: Basic concepts and processes for first-time cubesat developers, 2017.

[13] Neil Ashby. The sagnac effect in the global positioning system. *Relativity in rotating frames: relativistic physics in rotating reference frames*, page 11, 2004.

[14] Mohammed Chessab Mahdi. *Attitude Stabilization for CubeSat: Concepts and Technology*. Cambridge Scholars Publishing, 2018.

[15] B. Etkins. *Dynamics of Atmospheric Flight*. Dover, Mineola, New York, 2000. pages 9-13,134-151,196-318.

[16] Warren F. Phillips. *Mechanics of Flight*. John Wiley and Sons Hoboken New Jersey, 2010.

[17] R.C. Nelson. *Flight Stability and Automatic Control*. McGraw-Hill, 2nd edition, 1998.

[18] Jerry E. White Roger R. Bate, Donald D. Mueller. *Fundamentals of Astrodynamics*. Dover, 1971.

[19] Anderson D. *Fundamentals of Aerodynamics 4th Edition.* McGraw Hill Series, 2007.

[20] James Diebel. *Representing Attitude: Euler Angles, Unit Quaternions, and Rotation.* Stanford University, 2006.

[21] Bong Wie. *Space Vehicle Dynamics and Control.* American Institute of Aeronautics and Astronautics, Inc., 2008.

[22] M. J. Snadden, J. M. McGuirk, P. Bouyer, K. G. Haritos, and M. A. Kasevich. Measurement of the earth's gravity gradient with an atom interferometer-based gravity gradiometer. *Phys. Rev. Lett.*, 81:971–974, Aug 1998.

[23] Enhanced magnetic field model. https://www.ngdc.noaa.gov/geomag/EMM/. cited Jul 11, 2017.

[24] Mirko Leomanni. Comparison of control laws for magnetic detumbling. *Research Gate*, 10 2012. accessed: 2/9/2018.

[25] M. Lovera. Magnetic satellite detumbling: The b-dot algorithm revisited. In *2015 American Control Conference (ACC)*, pages 1867–1872, July 2015.

[26] Farhat Assaad. *Attitude Determination and Control System for a CubeSat.* PhD thesis, Worcester Polytechnic Institute, March 1st 2013.

[27] Amit Sanyal and Zachary Lee-Ho. Attitude tracking control of a small satellite in low earth orbit. Aug 2009. doi:10.2514/6.2009-5902.

[28] Fabio Celani. Robust three-axis attitude stabilization for inertial pointing spacecraft using magnetorquers. *ScienceDirect: Acta Astronautica*, 107:87–96, February–March 2015.

[29] E. Carlen and M.C. Carvalho. *Linear Algebra: From the Beginning.* W. H. Freeman, 2006.

[30] Karl J Astrom and Tore Hagglund. *Advanced PID Control.* ISA - The Instrumentation, Systems and Automation Society, 2006.

[31] Katsuhiko Ogata. *Modern Control Engineering.* Prentice Hall, 5th edition, 2010.

[32] J.M. Mendel. Fuzzy logic systems for engineering: a tutorial. *Proceedings of the IEEE*, 83(3):345–377, 1995.

[33] L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning—i. *Information Sciences*, 8(3):199–249, 1975.

[34] H.-J. Zimmermann. *Fuzzy Set Theory — and Its Applications.* Springer Science + Business Media,LLC, 1999.

[35] Lotfi A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(1):28–44, 1973.

[36] Rene Jager. *Fuzzy logic in control.* Rene Jager, 1995.

[37] George Klir and Bo Yuan. *Fuzzy sets and fuzzy logic*, volume 4. Prentice hall New Jersey, 1995.

[38] E.H. Mamdani. Application of fuzzy algorithms for control of simple dynamic plant. *Proceedings of the Institution of Electrical Engineers*, 121:1585–1588, 1974.

[39] Michio Sugeno. *Industrial Applications of Fuzzy Control*. Elsevier Science Publishing Co., Japan, 1985.

[40] Dongrui Wu and Jerry M. Mendel. Enhanced karnik-mendel algorithms for interval type-2 fuzzy sets and systems. In *NAFIPS 2007 - 2007 Annual Meeting of the North American Fuzzy Information Processing Society*, pages 184–189, 2007.

[41] Nilesh N. Karnik, Jerry M. Mendel, and Qilian Liang. Type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems*, 7(6):643–658, 1999.

[42] Simranjeet Singh and Manpreet Kaur. Gain scheduling of pid controller based on fuzzy systems. In *MATEC Web of Conferences*, volume 57, page 01008, 2016.

[43] Yi-Hsing Chen. Nonlinear adaptive fuzzy hybrid sliding mode control design for trajectory tracking of autonomous mobile robots. *Mathematics*, 13(8):1329, 2025.