POLITECNICO DI TORINO

MASTER's Degree in SPACE ENGINEERING

MASTER's Degree Thesis

Predictive and Learning-Based Control of Underactuated Satellite Attitude Dynamics under Reaction Wheel Faults with Magnetorquer Assistance

Supervisors

Prof. Elisa CAPELLO

Ing. Simone CHESI

Candidate

FEDERICO LACEDONIO

April 2026

Abstract

The growing adoption of small satellites introduces significant challenges in terms of system reliability and on-board computational resources. In particular, the Attitude Determination and Control System (ADCS) must remain effective even in the presence of actuator degradations, while still satisfying the implementation constraints of embedded platforms. Within this context, this thesis investigates attitude-control strategies for a spacecraft operating under both nominal and degraded actuation conditions, with particular focus on the underactuated case resulting from reaction-wheel loss.

Three control directions are considered: a classical quaternion-based Proportional-Derivative (PD) baseline, a model-based Nonlinear Model Predictive Control (NMPC) strategy, and a data-driven neural-network approach intended as a possible surrogate policy. A benchmark-oriented simulation framework is defined to compare controller behavior in two representative scenarios: a fully actuated nominal configuration and a degraded underactuated configuration in which the surviving reaction wheels must cooperate with the magnetic-control subsystem. The comparison is supported by a compact set of quaternion-based pointing metrics, angular-rate indicators, requirement-style band-compliance measures, and actuator-usage quantities for both reaction wheels and magnetorquers.

The results show that, in the fully actuated benchmark, the PD controller remains the most effective overall solution. Although NMPC reduces several angular-rate indicators, it does not provide a sufficiently clear improvement in pointing performance or actuator usage to justify its additional complexity in the nominal case. In the underactuated benchmark, the comparison becomes more nuanced. Aggregate pointing metrics still favor PD in several cases, but NMPC proves more effective in suppressing the strongest rate excursions, preserving wheel-speed margin, reducing the most demanding magnetic-control episodes, and improving the timing of the most relevant degraded reorientation transient. These results indicate that the benefit of predictive control is not universal across all operating conditions, but becomes more meaningful when the control problem is structurally harder and residual actuator coordination becomes critical.

In parallel, a complete imitation-learning pipeline is developed, including simulation-based dataset generation, feature construction, output standardization, and supervised training of a neural policy. The resulting feed-forward controller is intended to replace the command-generation stage at actuator-interface level, so as to predict the full commanded torque and magnetic-dipole vector from the current spacecraft condition. In the degraded benchmark, the reduction in effective control authority is not imposed by reducing the NN output dimension, but by the faulted

actuator configuration and by the shared downstream actuation architecture. While this establishes a reproducible framework for learning a surrogate controller from expert data, the resulting closed-loop behavior is not yet satisfactory enough to support the neural network as a mature alternative to the model-based strategies. The data-driven branch should therefore be regarded as an exploratory outcome of the work and as a basis for future developments rather than as a validated final controller.

Overall, the thesis shows that NMPC becomes more relevant than the classical baseline mainly in degraded underactuated operation, whereas PD remains highly competitive, and often preferable, in nominal conditions. It also highlights that further progress on the learning-based branch will likely require more suitable machine-learning techniques, such as reinforcement learning, together with broader fault-injection campaigns and implementation-oriented validation.

Keywords: attitude control, underactuated spacecraft, reaction wheels, magnetorquers, nonlinear model predictive control, neural networks, imitation learning, fault-tolerant attitude control.

Acknowledgements

I would like to express my sincere gratitude to Professor Elisa Capello for her constant availability, competence, and guidance throughout the development of this thesis. Her support, valuable feedback, and technical insight have been fundamental in shaping this work and in helping me address its challenges with greater clarity and rigor.
I would also like to thank CUS-GNC for the training, guidance, and stimulating technical environment provided during this experience. The opportunity to work within such a context has been extremely valuable for my academic and professional growth, allowing me to deepen my knowledge and gain a more practical understanding of the topics addressed in this thesis.

Table of Contents

List of Tables

List of Figures

Acronyms

ADCS
Attitude Determination and Control System

AOCS
Attitude and Orbit Control System

LEO
Low Earth Orbit

VLEO
Very Low Earth Orbit

RW
Reaction Wheel

MTQ
Magnetorquer

PD
Proportional-Derivative

PID
Proportional-Integral-Derivative

MPC
Model Predictive Control

NMPC
Nonlinear Model Predictive Control

NN

Neural Network

MLP

Multilayer Perceptron

OCP

Optimal Control Problem

ECI

Earth-Centered Inertial

LVLH

Local Vertical Local Horizontal

DCM

Direction Cosine Matrix

IGRF

International Geomagnetic Reference Field

ECR

Equal Concern For Relaxation

ECSS

European Cooperation for Space Standardization

ReLU

Rectified Linear Unit

MSE

Mean Squared Error

RMSE

Root Mean Square Error

RMS

Root Mean Square

MPE

Mean Pointing Error

RPE

Residual Pointing Error

GPU

Graphics Processing Unit

RL

Reinforcement Learning

Chapter 1

Introduction

1.1 Motivation: Bridging the Reliability-Computation Gap

Small satellites increasingly require autonomous and reliable attitude-control capabilities to support demanding mission phases such as safe-mode recovery, attitude acquisition, commanded reorientation maneuvers, and precise payload pointing. In this context, the ADCS must operate under tight on-board constraints, including limited computational resources, restricted power availability, and hardware architectures that may offer reduced redundancy compared to larger platforms. At the same time, mission reliability remains essential, since the spacecraft must preserve safe and predictable behavior in the presence of environmental disturbances, actuator saturations, and component degradations.

These requirements expose a practical *reliability-computation gap*. On one side, classical feedback controllers such as quaternion-based PD laws are computationally lightweight, straightforward to implement, and generally robust under nominal conditions. Their design, tuning, and verification are usually simpler than those of optimization-based controllers, which makes them attractive as baseline and fallback solutions. However, classical controllers do not handle constraints directly. In practice, actuator saturations, rate limits, availability masks, and momentum-management logic are often enforced outside the feedback law through allocation blocks, command limiting, deadbands, and supervisory rules. When the spacecraft operates close to actuator bounds, or when control authority is reduced by faults, this indirect constraint handling can substantially modify the intended closed-loop behavior.

On the other side, MPC, and especially NMPC, can explicitly include nonlinear dynamics and actuator constraints in the control design. This is particularly relevant for spacecraft attitude motion, whose rotational behavior is governed

by quaternion kinematics and Euler rigid-body dynamics, including gyroscopic cross-coupling terms [1, 2]. In principle, NMPC offers a systematic way to trade off tracking performance, control smoothness, and actuation effort over a finite prediction horizon, while enforcing input bounds, rate limits, and soft output constraints [3, 4, 5, 6]. Its main drawback is computational cost: at each sampling instant, a constrained nonlinear optimization problem must be solved online. For embedded flight processors and strict real-time requirements, this may become a major implementation bottleneck.

The gap becomes even more pronounced in faulted or degraded actuation scenarios. RWs are widely used for fine attitude control, but the loss of one or more wheels may lead to an underactuated configuration in which arbitrary three-axis torque generation is no longer available. In such cases, attitude control must exploit the residual wheel-torque subspace, the nonlinear coupling of rigid-body dynamics, and the contribution of auxiliary actuators such as MTQs. However, MTQ authority is inherently limited because the generated magnetic torque is always orthogonal to the local geomagnetic field and therefore varies along the orbit [7, 8]. Moreover, the same MTQs are often shared with reaction-wheel momentum dumping, so attitude assistance and wheel desaturation are coupled at the actuation level [6]. As a result, the effective control authority is time-varying, anisotropic, and strongly scenario-dependent.

Within this framework, a key question arises: can part of the constraint-aware behavior of an optimal controller be retained while reducing the online computational burden to a level more compatible with embedded implementation? A possible answer is provided by imitation learning, in which an expert controller is used to generate demonstration data and a policy is trained to reproduce its input-output behavior through fast inference [9, 10, 11]. When the expert behavior can be represented as a direct mapping from the available measured variables to the commanded actuation, a feed-forward NN offers a particularly attractive compromise: it is simple to evaluate online, easy to integrate into embedded software, and capable of approximating complex nonlinear input-output relationships [12, 13].

This thesis addresses the reliability-computation gap by comparing three control strategies within a common spacecraft attitude-control framework: a classical quaternion-based PD baseline, an NMPC expert with explicit input and rate constraints, and a feed-forward neural controller trained to imitate the NMPC-driven control behavior. All strategies are assessed under a common spacecraft model and a consistent actuation and guidance framework, with particular attention to attitude-tracking performance, interaction with constraints, and reaction-wheel momentum management. Computational affordability is treated as an implementation-oriented motivation for the controller selection and as a direction for future validation, rather than as a dedicated benchmarked result of the present thesis.

1.2 Thesis Objectives and Research Questions

The general objective of this thesis is to investigate spacecraft attitude control under degraded actuation conditions and to assess how different control strategies balance reliability, constraint-awareness, and practical implementability. The work focuses on a common nonlinear attitude-control framework in which a classical controller, an optimization-based controller, and a learning-based surrogate are studied under consistent modeling assumptions. The comparative results reported in the thesis focus on closed-loop dynamic behavior and actuator usage, while a dedicated runtime benchmark is left for future work.

More specifically, the thesis pursues four main objectives.

1. **Establish a coherent benchmark framework.** Define a common spacecraft attitude-control problem in which all considered strategies are evaluated under the same physical model, the same actuation architecture, and the same guidance logic. The aim is to ensure that the comparison is meaningful from an engineering perspective and not biased by inconsistent assumptions.

2. **Assess the role of a classical baseline controller.** Analyze the behavior of a quaternion-based PD controller as a lightweight and practically implementable reference solution. This objective is important to quantify what can already be achieved with a simple feedback structure when the spacecraft operates under reduced control authority.

3. **Study the benefits of constraint-aware predictive control.** Formulate and evaluate an NMPC strategy capable of accounting explicitly for nonlinear dynamics, actuation limits, and degraded-control conditions. The purpose is to clarify the extent to which a predictive controller can improve closed-loop behavior when constraints and underactuation play a central role.

4. **Investigate a data-driven approximation of the expert policy.** Explore whether a feed-forward NN, trained through imitation learning from an NMPC expert, can reproduce a useful portion of the expert behavior while retaining the implementation simplicity that motivates learning-based surrogates for embedded ADCS applications.

These objectives lead to the following research questions.

1. To what extent can a classical quaternion-based PD controller remain effective in an underactuated spacecraft attitude-control problem?

2. What advantages does NMPC provide when actuator constraints, reduced control authority, and nonlinear dynamic coupling must be handled explicitly?

3. Can a feed-forward NN trained by imitation learning retain a meaningful portion of the control behavior generated by the NMPC expert?

4. From an implementation viewpoint, does the resulting learning-based controller suggest a plausible path toward reducing online complexity for embedded applications, even though a dedicated runtime benchmark is left outside the scope of the present results?

Taken together, these questions define the central theme of the thesis: understanding whether the robustness and constraint-awareness of predictive control can be approximated by a simpler learned controller within a coherent benchmark framework, while leaving quantitative runtime validation to future work.

1.3 Problem Statement

Spacecraft attitude control must ensure reliable stabilization and tracking even when the available actuation is limited, constrained, or partially degraded. This requirement is particularly relevant for small satellites, where the ADCS is expected to achieve accurate rotational control under tight on-board limitations in computation, power, and actuator authority. In such systems, the control problem is not only one of attitude regulation, but also one of maintaining acceptable performance in the presence of nonlinear dynamics, bounded actuation, and time-varying environmental interactions [1, 2].

In many practical architectures, fine attitude control is primarily assigned to RWs, while MTQs provide magnetic control authority and are also involved in auxiliary tasks such as reaction-wheel momentum management. Under nominal conditions, this combination already introduces coupling between rotational dynamics, actuator saturations, and environmental dependence. When one or more wheels become unavailable, the problem becomes more challenging: the spacecraft may enter an underactuated condition in which the commanded body torque can no longer be generated arbitrarily along all axes. As a result, control performance depends not only on the feedback law itself, but also on how the remaining actuation authority, rigid-body coupling, and magnetic actuation are exploited.

This motivates the central problem addressed in the thesis: how to control spacecraft attitude effectively in a degraded and constrained actuation scenario while preserving a computational burden compatible with realistic embedded implementation. Classical feedback laws such as quaternion-based PD controllers are attractive because of their simplicity and low computational cost, but they do not incorporate constraints directly and may become difficult to tune when the available control authority is reduced. By contrast, predictive control strategies such as NMPC can account explicitly for nonlinear dynamics and actuator limitations,

but their online optimization cost may hinder real-time deployment on flight-representative hardware [3, 4].

Within this context, a further question naturally arises: whether the behavior of a more expensive expert controller can be approximated through a data-driven policy that is much cheaper to evaluate online. This leads to a three-level comparison between a classical baseline, an optimization-based expert, and a learned surrogate. The purpose is not only to compare tracking performance, but also to understand how each strategy behaves when constraints, reduced actuation authority, and practical implementation requirements are considered simultaneously [11, 12, 13].

Accordingly, the thesis formulates a common benchmark attitude-control problem in which different control approaches are evaluated under the same spacecraft model, the same guidance structure, and the same degraded-actuation framework. The core issue is therefore to determine how the trade-off among reliability, constraint handling, and computational cost evolves across the considered strategies, and whether a learned controller can retain useful features of an NMPC expert while remaining more suitable for embedded execution.

1.4 Methodology and Comparative Approach

The thesis follows a model-based comparative methodology aimed at evaluating different attitude-control strategies within a common and coherent benchmark framework. Rather than analyzing each controller in isolation, the work adopts the same spacecraft dynamics, the same degraded-actuation architecture, and the same guidance logic as a shared basis for comparison. The orbital propagation used to generate environment-dependent quantities is treated only as a supporting simulation component and not as an independent topic of investigation. This choice is essential to ensure that the observed differences among controllers can be attributed to the control strategy itself, rather than to inconsistent modeling assumptions or scenario definitions.

The methodological workflow is organized around three control levels. First, a quaternion-based PD controller is considered as the classical baseline solution. Its role is to provide a lightweight and transparent reference against which more advanced methods can be assessed. Second, an NMPC controller is formulated as the optimization-based expert strategy. Because NMPC can account explicitly for nonlinear dynamics and operational constraints, it represents the high-performance reference in the comparative analysis [3, 4]. Third, a feed-forward NN is introduced as a data-driven surrogate trained to imitate the control behavior generated by the expert controller, following the general logic of imitation learning [9, 10, 11].

The comparison is therefore not limited to control performance alone. Instead, the thesis evaluates the considered strategies along multiple and complementary

dimensions: attitude-tracking capability, behavior under degraded actuation, interaction with actuator constraints, and compatibility with reaction-wheel momentum management. This broader perspective reflects the practical nature of the problem, where an effective ADCS design must balance control quality with implementation feasibility. In the present manuscript, however, implementation-oriented timing assessment is discussed qualitatively and deferred to future work, rather than being treated as a formal benchmark dimension.

Finally, the resulting control branches are discussed within a common simulation and assessment framework. In particular, the model-based strategies are compared directly at closed-loop level, while the learned surrogate is assessed as an exploratory branch whose current limitations and future improvements are also discussed.

This approach supports the central objective of the thesis: not simply identifying which controller performs best under ideal conditions, but understanding how different control paradigms trade off simplicity, constraint-awareness, and robustness in degraded scenarios within a common benchmark framework. In this sense, the comparative analysis is both methodological and engineering-oriented: methodological, because all controllers are assessed within a common framework; engineering-oriented, because the final discussion is driven by closed-loop behavior and actuator usage under realistic benchmark assumptions.

1.5 Thesis Outline

The remainder of the thesis is organized as follows:

- **Chapter 2 - Theoretical Background.** This chapter presents the theoretical background of the work. It introduces the reference frames adopted for spacecraft attitude analysis, the attitude parameterizations used throughout the thesis, the quaternion-based kinematics, the rigid-body rotational dynamics, the role of angular momentum, the general actuation principles of RWs and MTQs, and the main environmental effects relevant to spacecraft attitude control. The chapter concludes with a general formulation of the control problem, without introducing controller-specific details or benchmark-dependent numerical settings.

- **Chapter 3 - Control-Theoretic Background.** This chapter focuses on the control-theoretic background. It first reviews the classical quaternion-based PD approach as a baseline attitude controller, discussing its error definition, control law, and general closed-loop interpretation. It then introduces the NMPC framework, including the receding-horizon principle, the structure of the prediction model, the formulation of the cost function, and the handling of constraints. The chapter therefore provides the conceptual basis for the

two model-based controllers considered in the thesis, still without entering the numerical details of the adopted implementation.

- **Chapter 4 - Benchmark Definition and Controller Implementation.** This chapter translates the previous theoretical framework into the specific benchmark adopted in this work. It defines the spacecraft and mission assumptions, the orbital and environmental conditions, the degraded actuation scenario, the guidance profile, and the simulation framework. The same chapter also collects the practical implementation choices for the considered controllers, including the adopted parameters, constraints, and tuning settings, which are summarized through dedicated tables at the end of the relevant sections.

- **Chapter 5 - Data-Driven Strategy: Neural Network Control.** This chapter introduces the data-driven strategy based on a feed-forward NN. It discusses the imitation-learning logic adopted in the thesis, the generation of the training dataset from the expert controller, the pre-processing of the logged variables, the network architecture, and the integration of the trained model into the attitude-control loop.

- **Chapter 6 - Results and Comparative Analysis.** This chapter presents the simulation results and the comparative analysis. The closed-loop comparison is developed primarily for the classical and predictive controllers under a common benchmark framework, with attention to tracking behavior, interaction with actuator limitations, and robustness under degraded actuation. The neural branch is instead discussed in terms of its current status and limitations within the broader thesis perspective.

- **Chapter 7 - Conclusions and Future Work.** This chapter summarizes the main findings of the thesis, discusses the principal limitations of the present work, and outlines possible future developments.

Chapter 2

Theoretical Background

2.1 Reference Frames and Attitude Representation

A consistent definition of reference frames and attitude variables is required before introducing spacecraft rotational kinematics and dynamics. In attitude-control problems, the spacecraft orientation must be described with respect to a reference frame, environmental quantities must be expressed in a compatible coordinate system, and actuator directions must be interpreted in the same geometric framework. For this reason, the present chapter first defines the reference frames used throughout the thesis and then introduces the adopted attitude representation [1, 2].

2.1.1 Inertial, Orbital, and Body Reference Frames

Three reference frames are used throughout the thesis: an inertial frame, an orbit-related frame, and the spacecraft body frame. Together, they provide the geometric basis for describing orbital motion, spacecraft attitude, actuator directions, and environmental interactions.

The inertial frame adopted in this work is the ECI frame, denoted by

$$\mathcal{F}_I = \left\{ \hat{\mathbf{i}}_1, \hat{\mathbf{i}}_2, \hat{\mathbf{i}}_3 \right\}. \tag{2.1}$$

For the purposes of spacecraft attitude analysis, this frame is treated as non-rotating over the time intervals of interest. It provides the natural reference for expressing orbital position and velocity and for defining the absolute orientation of the spacecraft [1, 2].

An orbit-related frame is also introduced to describe the local geometry of the spacecraft along its trajectory. In this thesis, the orbital frame is the LVLH frame,

denoted by

$$\mathcal{F}_O = \{\hat{\mathbf{o}}_1, \hat{\mathbf{o}}_2, \hat{\mathbf{o}}_3\}\,. \tag{2.2}$$

This frame is constructed from the orbital position and velocity vectors and moves with the spacecraft along the orbit. It is useful for interpreting local pointing directions, orbital geometry, and environmental quantities that vary with orbital position, such as the geomagnetic field. Although the exact sign convention may vary across the literature, the LVLH frame is generally defined so that one axis is aligned with the local vertical direction, one lies in the local horizontal plane, and the third completes the right-handed triad [1, 2].

The spacecraft-fixed body frame is denoted by

$$\mathcal{F}_B = \left\{\hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \hat{\mathbf{b}}_3\right\}. \tag{2.3}$$

This frame is rigidly attached to the spacecraft structure. The inertia matrix, body angular velocity, actuator axes, control torques, and angular momentum quantities are all expressed in this frame. In practical ADCS formulations, the body frame is the natural coordinate system for control computation, since actuators generate torques directly in spacecraft-fixed coordinates.

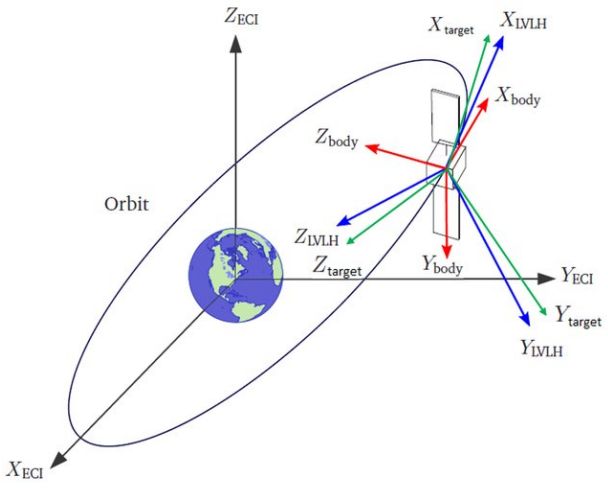


Figure 2.1: Reference frames adopted in the thesis. The figure illustrates the inertial frame $\mathcal{F}_I$, the orbital frame $\mathcal{F}_O$, and the spacecraft body frame $\mathcal{F}_B$, together with the attitude definition used to describe the rotation from inertial to body coordinates [14].

The spacecraft attitude is therefore the orientation of $\mathcal{F}_B$ with respect to the chosen reference frame, typically the inertial frame or, depending on the mission objective, an orbital frame. A rotation matrix can be used to map vectors between

coordinate systems. In particular, if $\mathbf{C}_{BI} \in \mathbb{R}^{3\times3}$ denotes the DCM from inertial to body coordinates, then any vector $\mathbf{v}_I$ expressed in $\mathcal{F}_I$ is mapped into body coordinates through

$$\mathbf{v}_B = \mathbf{C}_{BI}\,\mathbf{v}_I. \tag{2.4}$$

Because $\mathbf{C}_{BI}$ represents a proper rotation, it satisfies

$$\mathbf{C}_{BI}^T\mathbf{C}_{BI} = \mathbf{I}, \qquad \det(\mathbf{C}_{BI}) = 1. \tag{2.5}$$

The inverse transformation is therefore given by

$$\mathbf{v}_I = \mathbf{C}_{BI}^T\,\mathbf{v}_B. \tag{2.6}$$

These three frames play complementary roles in the thesis. The inertial frame provides the global reference for attitude definition, the orbital frame supports the interpretation of mission geometry and environment-dependent quantities, and the body frame is the natural domain for rotational dynamics and control actuation.

2.1.2 Attitude Parameterizations

A spacecraft attitude can be represented through several equivalent mathematical parameterizations. Common choices include rotation matrices, Euler angles, axis-angle descriptions, and quaternions. Each representation offers a different trade-off between physical interpretability, compactness, and suitability for control-oriented computation [1, 2].

Rotation matrices provide a direct and geometrically complete description of orientation, since they map coordinates from one frame to another through linear transformation. Their main drawback is redundancy: although a rotation matrix contains nine elements, only three independent parameters are needed to define a rigid-body orientation. Moreover, orthogonality must be preserved numerically.

Euler angles are often attractive because of their intuitive interpretation in terms of successive rotations. However, they suffer from singular configurations, commonly referred to as gimbal lock, where the mapping between angular velocity and angle rates becomes ill-conditioned or singular. For this reason, Euler angles are generally more suitable for visualization and interpretation than for global attitude propagation or feedback-law formulation.

Axis-angle representations provide a compact geometric description of a finite rotation by combining a unit rotation axis with a scalar rotation angle. While conceptually clear, they are not always the most convenient choice for continuous-time propagation and controller implementation.

Quaternions overcome many of these limitations. A unit quaternion offers a globally non-singular description of attitude, requires only four scalar components, and is numerically well suited for spacecraft kinematics and control design. The

only redundancy is the double covering property, according to which two opposite quaternions represent the same physical attitude. Despite this ambiguity, quaternions are widely preferred in spacecraft applications because they avoid the singularities of Euler angles while remaining more compact than rotation matrices [1, 2, 15].

For these reasons, quaternions are adopted throughout this thesis as the primary attitude representation for modeling, simulation, and control-oriented derivations. Euler angles may still be used in plots or post-processing when a more intuitive interpretation of orientation is useful, but they are not employed as the main internal state representation.

2.1.3 Quaternion Representation

The spacecraft attitude is represented by a unit quaternion written in scalar-first form as

$$q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T = \begin{bmatrix} q_0 \\ \mathbf{q}_v \end{bmatrix}, \tag{2.7}$$

where $q_0 \in \mathbb{R}$ is the scalar part and $\mathbf{q}_v \in \mathbb{R}^3$ is the vector part. The unit-norm condition

$$\|q\|_2 = 1 \tag{2.8}$$

ensures that the quaternion represents a proper rigid-body rotation.

If the spacecraft undergoes a rotation of angle θ about a unit axis $\hat{\mathbf{e}}$, the corresponding quaternion is

$$q = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ \hat{\mathbf{e}} \sin\left(\frac{\theta}{2}\right) \end{bmatrix}. \tag{2.9}$$

This relation highlights one of the key properties of quaternions: finite rotations are represented without the singularities associated with angle-based parameterizations.

The conjugate of the quaternion is defined as

$$q^* = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \end{bmatrix}^T, \tag{2.10}$$

and, for a unit quaternion, it coincides with the inverse rotation:

$$q^{-1} = q^*. \tag{2.11}$$

Quaternion composition is described by the quaternion product. Given two rotations represented by q_a and q_b, the composed rotation is written as

$$q_c = q_a \otimes q_b, \tag{2.12}$$

where $\otimes$ denotes quaternion multiplication. In scalar-vector form, the product can be expressed as

$$q_a \otimes q_b = \begin{bmatrix} q_{a,0}q_{b,0} - \mathbf{q}_{a,v}^T \mathbf{q}_{b,v} \\ q_{a,0}\mathbf{q}_{b,v} + q_{b,0}\mathbf{q}_{a,v} + \mathbf{q}_{a,v} \times \mathbf{q}_{b,v} \end{bmatrix}. \tag{2.13}$$

The associated DCM from inertial to body coordinates can be written as a function of the quaternion components. Under the scalar-first convention adopted here,

$$\mathbf{C}_{BI}(q) = \left(q_0^2 - \mathbf{q}_v^T \mathbf{q}_v\right) \mathbf{I} + 2\mathbf{q}_v \mathbf{q}_v^T - 2q_0[\mathbf{q}_v]_\times, \tag{2.14}$$

where $[\mathbf{q}_v]_\times$ denotes the skew-symmetric matrix associated with $\mathbf{q}_v$. This expression provides the direct link between quaternion and rotation-matrix formulations.

Finally, because quaternions exhibit double covering, the pair q and $-q$ represents the same physical attitude. This property is not problematic in itself, but it must be handled carefully whenever continuity of the attitude trajectory or uniqueness of the attitude error representation is required in later control formulations [15]. For the theoretical development of this chapter, it is sufficient to note that quaternions provide a compact, globally non-singular, and control-friendly representation of spacecraft orientation.

2.2 Attitude Kinematics and Dynamics

Once the spacecraft attitude has been represented through quaternions, the next step is to describe how the attitude evolves in time and how the rotational motion is governed by external and internal torques. In spacecraft attitude analysis, this requires two complementary ingredients: kinematics, which relates angular velocity to the time evolution of the attitude variables, and dynamics, which relates the angular acceleration to the applied torques and to the inertia properties of the rigid body [1, 2].

In the present thesis, the spacecraft is modeled as a rigid body. This assumption is standard in preliminary ADCS analysis and is appropriate when flexible appendage dynamics are either negligible or intentionally excluded from the control-oriented model. Under this assumption, the rotational state is described by the attitude quaternion and by the angular velocity vector expressed in the body frame.

2.2.1 Quaternion Kinematics

Let the spacecraft angular velocity expressed in the body frame be

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T. \tag{2.15}$$

The quaternion kinematics relates this angular velocity to the time derivative of the attitude quaternion. Under the scalar-first convention adopted in this thesis, the quaternion differential equation can be written as

$$\dot{q} = \frac{1}{2}\,\mathbf{\Omega}(\boldsymbol{\omega})\,q, \tag{2.16}$$

where

$$\mathbf{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}. \tag{2.17}$$

Equivalent forms of the same equation are often found in the literature, depending on the adopted convention for quaternion multiplication and for the direction of frame transformation. The form above is consistent with the scalar-first quaternion convention introduced in Section 2.1.

An equivalent scalar-vector expression is

$$\dot{q}_0 = -\frac{1}{2}\,\mathbf{q}_v^T\boldsymbol{\omega}, \tag{2.18}$$

$$\dot{\mathbf{q}}_v = \frac{1}{2}\,(q_0\mathbf{I} + [\mathbf{q}_v]_\times)\,\boldsymbol{\omega}, \tag{2.19}$$

where $[\mathbf{q}_v]_\times$ is the skew-symmetric matrix associated with the vector part of the quaternion.

The quaternion kinematics is bilinear in the attitude and angular-velocity variables. This equation does not involve torques directly; it only expresses how a given angular velocity changes the orientation of the body frame over time. In this sense, the kinematics provides the geometric propagation law of the attitude state.

A fundamental property of the quaternion representation is that the unit-norm condition should remain satisfied during propagation. In exact continuous-time dynamics, if the quaternion is initially normalized and the kinematic equation is integrated exactly, then the unit norm is preserved. In numerical simulation and discrete-time implementation, however, integration errors may introduce small norm deviations. For this reason, quaternion normalization is commonly enforced during simulation or signal processing to preserve a valid attitude representation [1, 2].

2.2.2 Rigid-Body Rotational Dynamics

The rotational dynamics of the spacecraft is governed by Euler's equation expressed in the body frame. Let $\mathbf{J} \in \mathbb{R}^{3\times3}$ denote the spacecraft inertia matrix about its

center of mass, expressed in body coordinates. The angular momentum of the rigid body is then

$$\mathbf{h} = \mathbf{J}\boldsymbol{\omega}. \tag{2.20}$$

Assuming a rigid spacecraft body, the rotational equation of motion can be written as

$$\mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathbf{J}\boldsymbol{\omega}) = \boldsymbol{\tau}_{ext}, \tag{2.21}$$

where $\boldsymbol{\tau}_{ext}$ is the resultant external torque expressed in the body frame.

Equation (2.21) shows that the rotational motion is influenced not only by the applied torque, but also by the gyroscopic coupling term

$$\boldsymbol{\omega} \times (\mathbf{J}\boldsymbol{\omega}), \tag{2.22}$$

which captures the nonlinear interaction among the angular-velocity components and the inertia distribution. This term is especially relevant in spacecraft applications because it creates cross-axis coupling even when the applied torque acts primarily along one direction.

If the inertia matrix is diagonal in the chosen body frame,

$$\mathbf{J} = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix}, \tag{2.23}$$

the dynamics can be written component-wise as

$$J_x\dot{\omega}_x - (J_y - J_z)\omega_y\omega_z = \tau_x, \tag{2.24}$$

$$J_y\dot{\omega}_y - (J_z - J_x)\omega_z\omega_x = \tau_y, \tag{2.25}$$

$$J_z\dot{\omega}_z - (J_x - J_y)\omega_x\omega_y = \tau_z. \tag{2.26}$$

These equations highlight that rotational motion is intrinsically coupled unless the body is perfectly spherical, that is, unless all principal moments of inertia are equal.

For control-oriented formulations, the total applied torque is usually decomposed into contributions from control actuation and from the environment:

$$\boldsymbol{\tau}_{ext} = \boldsymbol{\tau}_{ctrl} + \boldsymbol{\tau}_{dist}, \tag{2.27}$$

where $\boldsymbol{\tau}_{ctrl}$ is the torque generated by the attitude-control system and $\boldsymbol{\tau}_{dist}$ collects disturbance torques such as gravity-gradient, aerodynamic, solar-radiation-pressure, or residual magnetic contributions. This decomposition is conceptually useful because it separates the commanded action from the perturbing effects that the controller must reject or accommodate.

Together, the quaternion kinematics and Euler rotational dynamics define a nonlinear state-space model of spacecraft attitude motion. The state nonlinearity originates both from the quaternion propagation law and from the gyroscopic term in the rotational dynamics. These nonlinear features are central in spacecraft attitude control, especially when actuator limitations or degraded-control conditions prevent simple decoupled control action.

2.2.3 Angular Momentum Formulation

Angular momentum plays a central role in spacecraft attitude dynamics because it provides a compact physical description of rotational motion and of the interaction between the spacecraft body and internal actuators. In the absence of external torque, the total angular momentum of the spacecraft system is conserved. When internal momentum-exchange devices are present, the distribution of angular momentum among the spacecraft body and the actuators becomes a key aspect of attitude-control analysis [1, 2].

For the rigid spacecraft body alone, the body angular momentum is given by (2.20). If internal rotating devices such as RWs are included, the total angular momentum of the spacecraft system can be expressed more generally as

$$\mathbf{h}_{tot} = \mathbf{J}\boldsymbol{\omega} + \mathbf{h}_{int}, \tag{2.28}$$

where $\mathbf{h}_{int}$ represents the internal stored angular momentum associated with the rotating actuators.

The time derivative of the total angular momentum in the body frame satisfies

$$\dot{\mathbf{h}}_{tot} + \boldsymbol{\omega} \times \mathbf{h}_{tot} = \boldsymbol{\tau}_{ext}. \tag{2.29}$$

This equation emphasizes an important physical point: internal actuators cannot create net angular momentum from nothing. They redistribute momentum within the spacecraft system, while the overall variation of total angular momentum is driven by external torque.

This perspective is particularly useful in spacecraft equipped with momentum-exchange actuators. By accelerating or decelerating an internal wheel, the control system changes the actuator momentum and, by reaction, applies a torque to the spacecraft body. Although this mechanism enables precise attitude control, it also implies that momentum can accumulate in the wheels over time. If disturbance torques persist, or if the spacecraft repeatedly performs demanding maneuvers, the internal momentum may approach actuator limits. In that case, additional means are needed to unload or redistribute stored momentum.

From a control-theoretic standpoint, angular momentum therefore provides the link between instantaneous torque generation and long-term actuator operability. Torque determines the short-term evolution of angular velocity and attitude,

whereas stored momentum affects whether the same control authority can be sustained over time. This is why momentum management is not merely an auxiliary concern, but a structural aspect of spacecraft attitude control whenever internal torque actuators are involved.

In the context of this thesis, the angular-momentum formulation is important for two reasons. First, it clarifies the physical role of RWs as momentum-exchange devices rather than direct external torque sources. Second, it motivates the need to consider actuator availability, momentum buildup, and auxiliary actuation mechanisms within the broader attitude-control problem. The detailed actuator models and the role of auxiliary magnetic actuation are introduced in the next section.

2.3 Spacecraft Actuation Principles

Spacecraft attitude control is achieved by generating control torques that modify the rotational motion of the rigid body. Depending on the platform architecture, these torques can be produced by internal momentum-exchange devices, by actuators interacting with the external environment, or by a combination of both. In small-satellite ADCS architectures, two of the most common actuator classes are RWs and MTQs. Although both contribute to attitude control, they rely on different physical mechanisms and provide very different forms of authority [1, 2].

In the present thesis, these actuators are not introduced yet through the specific configuration adopted in the case study, but through their general operating principles. This distinction is important because the benchmark-dependent architecture will be defined later, whereas the present section only establishes the physical and mathematical concepts needed to interpret the control problem.

2.3.1 Reaction Wheel Torque Generation

RWs are internal momentum-exchange devices used to generate control torque by accelerating or decelerating a rotating flywheel mounted on the spacecraft. Their operation is based on conservation of angular momentum: when the wheel momentum changes, an equal and opposite torque is applied to the spacecraft body [1, 2].

Consider a wheel whose spin axis, expressed in the body frame, is given by the unit vector $\hat{\mathbf{a}}_i$. Let $h_{w,i}$ denote the angular momentum stored in the i-th wheel. Neglecting wheel structural flexibility and higher-order internal effects, the torque applied by that wheel to the spacecraft body can be written as

$$\boldsymbol{\tau}_{rw,i} = -\dot{h}_{w,i}\,\hat{\mathbf{a}}_i, \tag{2.30}$$

where $\dot{h}_{w,i}$ is the commanded rate of change of wheel momentum. If the wheel rotational inertia about its spin axis is $J_{w,i}$ and its spin rate is $\Omega_{w,i}$, then

$$h_{w,i} = J_{w,i}\Omega_{w,i}, \tag{2.31}$$

so that the body torque becomes

$$\boldsymbol{\tau}_{rw,i} = -J_{w,i}\dot{\Omega}_{w,i}\,\hat{\mathbf{a}}_i. \tag{2.32}$$

For a cluster of N_w wheels, the total reaction-wheel torque can be written compactly as

$$\boldsymbol{\tau}_{rw} = -\mathbf{A}_{rw}\,\dot{\mathbf{h}}_w, \tag{2.33}$$

where

$$\mathbf{A}_{rw} = \begin{bmatrix} \hat{\mathbf{a}}_1 & \hat{\mathbf{a}}_2 & \cdots & \hat{\mathbf{a}}_{N_w} \end{bmatrix} \in \mathbb{R}^{3\times N_w}, \tag{2.34}$$

and

$$\dot{\mathbf{h}}_w = \begin{bmatrix} \dot{h}_{w,1} & \dot{h}_{w,2} & \cdots & \dot{h}_{w,N_w} \end{bmatrix}^T \tag{2.35}$$

collects the wheel momentum rates. Equation (2.33) shows that the mapping from wheel commands to body torque depends directly on the wheel-axis geometry.

A major advantage of RWs is that they can provide continuous and finely controllable torque without consuming propellant. This makes them particularly well suited for accurate pointing and smooth attitude maneuvers. However, their torque generation is inherently linked to momentum storage. Since the wheels absorb angular momentum while controlling the spacecraft, persistent disturbance torques or repeated maneuvers may gradually drive them toward saturation. Once the wheel speed or stored momentum approaches its allowable limit, the available control authority is reduced unless momentum is unloaded through some auxiliary mechanism.

More generally, the action of RWs must be understood on two different time scales. On a short time scale, they are torque actuators capable of producing precise attitude corrections. On a longer time scale, they are momentum-storage devices whose usable range is finite. This dual nature is one of the main reasons why reaction-wheel control and momentum management are deeply coupled in practical spacecraft ADCS design.

2.3.2 Magnetorquer Torque Generation

MTQs generate control torque by interacting with the local geomagnetic field. Unlike RWs, which exchange momentum internally, MTQs produce an external torque through electromagnetic interaction with the environment. This makes them especially attractive for small satellites because they are lightweight, simple, and

naturally suited for functions such as detumbling and reaction-wheel momentum unloading [1, 7, 8].

Let $\mathbf{m} \in \mathbb{R}^3$ denote the magnetic dipole moment generated by the magnetorquer assembly and let $\mathbf{B} \in \mathbb{R}^3$ denote the local geomagnetic field, both expressed in the body frame. The resulting magnetic torque is

$$\boldsymbol{\tau}_{mtq} = \mathbf{m} \times \mathbf{B}. \tag{2.36}$$

This relation is fundamental because it immediately shows that the generated torque is always orthogonal to the magnetic field vector. As a consequence, magnetic actuation cannot provide arbitrary torque in all directions at a given instant.

A useful matrix form is obtained by introducing the skew-symmetric matrix associated with $\mathbf{B}$:

$$\boldsymbol{\tau}_{mtq} = -[\mathbf{B}]_\times \, \mathbf{m}. \tag{2.37}$$

The mapping from dipole command to body torque is therefore linear in $\mathbf{m}$ for a fixed magnetic field, but its effectiveness depends directly on the instantaneous direction and magnitude of $\mathbf{B}$.

This dependence has two important implications. First, magnetic control authority is time-varying along the orbit because the geomagnetic field changes with spacecraft position and attitude. Second, the actuation is anisotropic even at a fixed orbital point, since no torque can be produced along the direction of $\mathbf{B}$. Therefore, MTQs are intrinsically underactuated actuators when considered instantaneously [7, 8].

Despite this limitation, MTQs remain extremely useful in practice. Over time, the direction of the geomagnetic field changes along the orbit, so the set of achievable torques varies as well. This time-varying authority can be exploited for functions such as detumbling, coarse attitude correction, and wheel desaturation. In particular, when used together with RWs, MTQs can provide the external torque needed to unload accumulated wheel momentum, thereby restoring the long-term effectiveness of the momentum-exchange system.

For this reason, magnetic control should not be interpreted only as a low-authority backup solution. In many small-satellite architectures, it is an essential complementary actuation mechanism that enables sustainable attitude control over extended mission time scales.

2.3.3 Underactuated Attitude Control Concept

A spacecraft attitude-control system is said to be underactuated when the available actuators cannot generate arbitrary three-axis control torque instantaneously. In such a condition, the torque space accessible to the control system has dimension smaller than three, either because of actuator geometry, actuator failure, environmental dependence, or a combination of these factors.

This concept can be expressed in a general control-allocation form. Suppose that the total control torque is written as

$$\boldsymbol{\tau}_{ctrl} = \mathbf{G}(\xi, t)\,\mathbf{u}, \tag{2.38}$$

where $\mathbf{u}$ collects the actuator commands and $\mathbf{G}(\xi, t)$ is the actuator mapping matrix, possibly dependent on the system state ξ and on time. If

$$\operatorname{rank}(\mathbf{G}(\xi, t)) < 3, \tag{2.39}$$

then the control system is instantaneously unable to span the full three-dimensional torque space.

Underactuation may arise in different ways. For reaction-wheel systems, it may result from the loss of one or more wheels or from a wheel layout that does not provide full-rank torque authority after degradation. For magnetic actuation, underactuation is intrinsic, since the torque generated by (2.36) is always orthogonal to the magnetic field. More generally, a spacecraft combining multiple actuator classes may still be underactuated if the instantaneous authority of the remaining actuators does not fully cover the required torque directions.

The key consequence is that attitude control can no longer be interpreted as simple axis-by-axis torque assignment. Instead, the controller must exploit the available torque subspace, the nonlinear coupling of rigid-body dynamics, and, when present, the time-varying contribution of the environment. In this setting, performance depends not only on the nominal control objective, but also on how the controller prioritizes directions, handles unavailable torque components, and coordinates the residual actuation resources.

Underactuated spacecraft control is therefore fundamentally different from fully actuated control. In a fully actuated system, any commanded body torque can in principle be mapped directly to the actuators, up to saturation limits. In an underactuated system, this direct correspondence no longer holds. The controller must instead rely on indirect mechanisms such as sequential torque generation, exploitation of dynamic coupling, or the complementary action of actuators with different physical characteristics.

This distinction is central to the thesis. The goal is not merely to regulate spacecraft attitude under nominal authority, but to analyze how different control strategies behave when the available actuation is structurally limited. The detailed benchmark architecture will be defined later, but the present theoretical concept already explains why degraded spacecraft attitude control is a nontrivial problem even before numerical parameters or implementation details are introduced.

2.4 Environmental Effects and Perturbations

Spacecraft attitude motion is not determined only by the commanded control torques. Even when the primary objective is attitude regulation or tracking, the spacecraft remains embedded in an orbital environment that continuously affects both the available actuation and the disturbance torques acting on the rigid body. For this reason, a control-oriented attitude model must account not only for the spacecraft dynamics and actuators, but also for the main environmental interactions that influence rotational behavior [1, 2].

In the present thesis, environmental effects are relevant for two main reasons. First, they introduce disturbance torques that the control system must reject or accommodate. Second, they determine the operating conditions of magnetic actuation, since the torque generated by MTQs depends directly on the local geomagnetic field. The present section introduces these effects at a theoretical level, without yet specifying the numerical settings adopted in the benchmark case study.

2.4.1 Geomagnetic Field Interaction

The geomagnetic field is one of the most important environmental quantities in spacecraft attitude control whenever magnetic actuation or magnetic disturbance effects are involved. Along the orbit, the spacecraft is immersed in a magnetic field that varies with geographic position, altitude, and time. When expressed in the spacecraft body frame, this field also depends on the instantaneous attitude of the spacecraft.

Let $\mathbf{B}_I$ denote the geomagnetic field vector expressed in an external reference frame and let $\mathbf{C}_{BI}$ be the direction cosine matrix from inertial to body coordinates. The magnetic field expressed in the body frame is then

$$\mathbf{B}_B = \mathbf{C}_{BI}\,\mathbf{B}_I. \tag{2.40}$$

This relation shows that the magnetic field seen by the spacecraft is not only orbit-dependent, but also attitude-dependent.

The control relevance of the geomagnetic field follows directly from the magnetic-torque relation introduced in Section 2.3:

$$\boldsymbol{\tau}_{mtq} = \mathbf{m} \times \mathbf{B}_B. \tag{2.41}$$

Therefore, the available magnetic control authority depends on both the magnitude and the direction of $\mathbf{B}_B$. In particular, no magnetic torque can be generated along the direction of the field itself. This makes the instantaneous magnetic actuation authority inherently anisotropic and time-varying [7, 8].

Beyond actuation, the geomagnetic field also affects spacecraft attitude through magnetic disturbance mechanisms. Residual magnetic dipoles associated with

spacecraft materials, electronics, or imperfect magnetic cleanliness can interact with the ambient field and generate parasitic torques. For this reason, the geomagnetic field has a dual role in spacecraft attitude dynamics: it provides a useful source of external control authority through MTQs, but it can also act as a source of environmental perturbation.

From a modeling standpoint, the geomagnetic field is usually treated as a known or estimated exogenous input that varies along the orbit. In control-oriented formulations, what matters most is not only its absolute value, but the fact that its direction and magnitude evolve over time. This time variation is one of the key features that distinguish magnetic attitude-control problems from fully body-fixed actuation problems.

2.4.2 Disturbance Torque Sources

In addition to the commanded control torques, a spacecraft is subject to environmental torques that arise from its interaction with the surrounding orbital environment. These perturbations are generally small compared to the torque authority of the actuators, but they act continuously and can accumulate their effects over time. As a result, they are important both for closed-loop tracking performance and for long-term momentum management [1, 2].

A general disturbance-torque contribution can be denoted by

$$\boldsymbol{\tau}_{dist} = \boldsymbol{\tau}_{gg} + \boldsymbol{\tau}_{aero} + \boldsymbol{\tau}_{srp} + \boldsymbol{\tau}_{mag} + \cdots , \tag{2.42}$$

where the terms on the right-hand side represent, respectively, gravity-gradient, aerodynamic, solar-radiation-pressure, and magnetic disturbance torques, while the ellipsis indicates that additional sources may be included when relevant.

The gravity-gradient torque originates from the nonuniform gravitational field acting on an extended rigid body. Because different parts of the spacecraft experience slightly different gravitational attraction, a net restoring or destabilizing torque may arise depending on the attitude and inertia distribution. In classical form, this torque is related to the orbital radius vector and to the inertia matrix of the spacecraft. Its significance is strongly dependent on orbit geometry and spacecraft inertia asymmetry.

Aerodynamic torque is generated by the interaction between the residual atmosphere and the spacecraft surfaces. Although the density is low in many orbital regimes, the aerodynamic force acting away from the center of mass can still produce a measurable moment. This torque depends on atmospheric density, spacecraft velocity relative to the atmosphere, exposed area, surface properties, and the offset between aerodynamic center and center of mass.

Solar-radiation-pressure torque is caused by the momentum transfer associated with incident solar photons. The resulting force on illuminated surfaces can produce

an attitude-dependent torque when the line of action does not pass through the center of mass. This effect is usually smaller than aerodynamic torque in lower orbits, but it may become significant depending on spacecraft geometry, optical properties, and mission profile.

Magnetic disturbance torque is commonly modeled as the interaction between a residual spacecraft magnetic dipole $\mathbf{m}_{res}$ and the ambient geomagnetic field:

$$\boldsymbol{\tau}_{mag} = \mathbf{m}_{res} \times \mathbf{B}_B. \tag{2.43}$$

This disturbance is particularly relevant for spacecraft carrying magnetic actuators, magnetometers, or components that are difficult to magnetically clean, since residual dipole effects may vary with configuration and operational state.

These disturbance sources differ in physical origin, but they share a common control implication: they continuously bias the rotational dynamics and may drive the system away from the desired attitude unless compensated or rejected by the controller. Even when each individual disturbance is small, their cumulative effect may be sufficient to alter transient behavior, increase steady-state pointing error, or induce reaction-wheel momentum buildup over time.

2.4.3 Role of Disturbances in Attitude Control

From a control perspective, environmental disturbances are not merely secondary modeling details. They directly influence the difficulty of the attitude-control problem, the amount of control effort required to maintain pointing performance, and the long-term sustainability of actuator operation.

In the rotational dynamics introduced in Section 2.2, the total applied torque is written as

$$\boldsymbol{\tau}_{ext} = \boldsymbol{\tau}_{ctrl} + \boldsymbol{\tau}_{dist}. \tag{2.44}$$

This expression shows that disturbance torques enter the dynamics in the same channel as the control torque, that is, as direct contributors to angular acceleration. Consequently, the controller must generate additional action whenever it seeks to reject the perturbing effects and preserve the desired attitude evolution.

The impact of disturbances can be interpreted on at least three levels. First, disturbances affect instantaneous tracking and regulation performance by introducing torque components not commanded by the control law. Second, they modify the transient control effort required to achieve or maintain a target attitude. Third, when momentum-exchange actuators are used, persistent disturbances may lead to gradual internal momentum accumulation, thereby coupling environmental perturbations with actuator saturation and desaturation requirements.

This last point is particularly important in spacecraft equipped with RWs. Because reaction wheels store momentum while generating body torque, the continuous rejection of low-amplitude but persistent disturbances may progressively

consume the available momentum margin. In such cases, disturbance rejection cannot be viewed only as a short-term performance issue; it also becomes a long-term operability issue. Auxiliary actuation mechanisms, such as magnetic unloading, are therefore often needed not simply to improve attitude control, but to preserve the availability of reaction-wheel authority over time [6].

Disturbances also influence the relative advantages of different control strategies. A classical controller may reject moderate perturbations effectively when enough control authority is available, but it typically treats disturbances only indirectly through feedback action. A predictive controller, by contrast, may incorporate disturbance-related trade-offs within its constrained optimization framework, at least when the disturbance model or its effect is represented explicitly. More generally, the presence of environmental perturbations strengthens the importance of constraint-aware and architecture-aware control design.

For all these reasons, environmental effects must be considered an integral part of the spacecraft attitude-control problem. In the following section, the discussion is completed by formalizing the control problem in a general way, before introducing the specific benchmark settings and controller implementations adopted later in the thesis.

2.5 General Control Problem Formulation

The previous sections introduced the geometric, dynamic, actuation, and environmental foundations of the spacecraft attitude-control problem. These elements can now be collected into a general control-oriented formulation, whose purpose is to define the regulated variables, the available actuation channels, and the main constraints affecting the closed-loop behavior. At this stage, the formulation remains intentionally controller-independent: it describes *what* the control problem is, not yet *how* it is solved by a specific control strategy [1, 2].

2.5.1 Attitude Regulation and Tracking Objectives

The general goal of spacecraft attitude control is to drive the spacecraft orientation toward a desired attitude and, when required, to make the spacecraft follow a prescribed time-varying reference profile. Depending on the mission phase, this may correspond to attitude stabilization, reference tracking, pointing maintenance, or commanded slew maneuvers.

Let $q(t)$ denote the spacecraft attitude quaternion and let $q_{ref}(t)$ denote the desired reference attitude. The attitude-control objective can be expressed through an attitude error variable that measures the relative rotation between the current and desired orientations. Using the scalar-first quaternion convention, a common

definition of the error quaternion is

$$q_e(t) = q_{ref}(t) \otimes q^{-1}(t), \tag{2.45}$$

where $q^{-1}(t)$ is the inverse of the current attitude quaternion and $\otimes$ denotes quaternion multiplication. When the spacecraft is perfectly aligned with the reference attitude, the error quaternion corresponds to the identity rotation.

In addition to attitude alignment, rotational-rate behavior must also be controlled. Let $\boldsymbol{\omega}(t)$ be the spacecraft angular velocity expressed in the body frame and let $\boldsymbol{\omega}_{ref}(t)$ denote the desired reference angular velocity, when such a reference is defined. A general angular-velocity tracking error can then be introduced as

$$\tilde{\boldsymbol{\omega}}(t) = \boldsymbol{\omega}(t) - \boldsymbol{\omega}_{ref}(t). \tag{2.46}$$

For pure attitude regulation problems, the desired rate is typically zero, whereas for tracking problems it may be associated with the kinematics of the reference attitude.

Accordingly, the spacecraft attitude-control objective can be stated in general terms as follows:

$$q_e(t) \to q_{id}, \qquad \tilde{\boldsymbol{\omega}}(t) \to \mathbf{0}, \tag{2.47}$$

where $q_{id} = [1\ 0\ 0\ 0]^T$ is the identity quaternion. In words, the controller seeks to reduce the attitude error while simultaneously damping angular-velocity deviations.

This formulation emphasizes that spacecraft attitude control is fundamentally a nonlinear tracking problem. The controlled variables evolve on a nonlinear rotational manifold, the dynamics are coupled through Euler's equation, and the achievable behavior depends on the actuator architecture as well as on the environmental conditions. For this reason, even the abstract control objective already reflects the need for a coordinated treatment of orientation, angular velocity, and actuation authority.

2.5.2 Constraints and Degraded-Actuation Conditions

In practical spacecraft systems, the control objective cannot be considered independently of the physical limitations affecting the actuators and the state evolution. The control problem is therefore constrained by design.

At a general level, the spacecraft rotational dynamics can be written as

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)\,, \tag{2.48}$$

where the state vector $\mathbf{x}(t)$ collects the attitude and angular-velocity variables, while the input vector $\mathbf{u}(t)$ represents the actuator commands. The control torque applied to the body is generated through an actuator mapping of the form

$$\boldsymbol{\tau}_{ctrl}(t) = \mathbf{G}(\mathbf{x}(t), t)\,\mathbf{u}(t), \tag{2.49}$$

where the matrix $\mathbf{G}(\cdot)$ may be constant or time-varying depending on the actuation principle.

The admissible input set is generally bounded by amplitude and rate constraints:

$$\mathbf{u}_{min} \leq \mathbf{u}(t) \leq \mathbf{u}_{max}, \tag{2.50}$$

$$\dot{\mathbf{u}}_{min} \leq \dot{\mathbf{u}}(t) \leq \dot{\mathbf{u}}_{max}, \tag{2.51}$$

and additional internal actuator states, such as stored momentum, may also be subject to operational limits:

$$\mathbf{z}_{min} \leq \mathbf{z}(t) \leq \mathbf{z}_{max}, \tag{2.52}$$

where $\mathbf{z}(t)$ denotes generic actuator-related quantities that affect sustained operability.

These constraints are not peripheral details. They directly shape the set of feasible control actions and may alter the achievable closed-loop performance. A controller that ignores them at the design level may still be made operational through external saturation logic or supervisory mechanisms, but the resulting closed-loop behavior can differ significantly from the ideal unconstrained design.

A further complication arises in degraded-actuation conditions. If actuator failures, availability masks, or environmental dependence reduce the rank of the allocation matrix in (2.49), the system may become underactuated:

$$\operatorname{rank}(\mathbf{G}(\mathbf{x}(t), t)) < 3. \tag{2.53}$$

In this case, the controller cannot generate arbitrary body torque instantaneously in all directions. The control problem must then be reformulated in terms of the torque subspace that remains accessible, the directions that can only be influenced indirectly, and the possible role of time-varying or auxiliary actuation mechanisms.

Therefore, the spacecraft attitude-control problem considered in this thesis is constrained not only by state and input bounds, but also by structural actuation limitations. This is precisely what makes the problem nontrivial from both a theoretical and an implementation standpoint.

2.5.3 Benchmarking Logic and Comparative Perspective

Because the thesis aims to compare different control strategies rather than to analyze a single controller in isolation, the control problem must also be framed from a benchmarking perspective. In other words, the formulation should support a fair and meaningful comparison among controllers operating under the same physical assumptions and mission logic.

At the highest level, this means that all considered controllers should be evaluated on a common nonlinear spacecraft model, under the same actuation architecture,

and with the same reference-generation logic and environmental assumptions. Only under these conditions can the observed differences in performance be interpreted as genuine differences among control paradigms rather than as artifacts of inconsistent problem definition.

The comparative perspective adopted in the thesis is based on three complementary dimensions. The first is *control effectiveness*, namely the ability to regulate or track the desired attitude while maintaining acceptable rotational behavior. The second is *constraint compatibility*, that is, the extent to which actuator limitations, reduced control authority, and momentum-related considerations are handled consistently within the control architecture. The third is *computational suitability*, which concerns the effort required to evaluate the controller online and therefore its potential compatibility with embedded implementation.

These dimensions reflect a broader engineering principle: in spacecraft ADCS design, the best controller is not necessarily the one that achieves the smallest tracking error in ideal conditions, but the one that offers the most favorable compromise between performance, robustness to practical limitations, and implementability. A controller may be elegant from a theoretical viewpoint yet difficult to deploy in real time, while a simpler controller may be highly attractive operationally despite more limited optimality.

For this reason, the control problem formulated in this chapter is intentionally broad enough to accommodate different solution paradigms. A classical feedback controller, a predictive optimization-based controller, and a learned surrogate can all be interpreted as alternative ways of solving the same constrained spacecraft attitude-control problem. Their specific theoretical principles will be discussed in the next chapter, while the concrete benchmark settings adopted for the comparison will be introduced later in the thesis.

This closes the system-level theoretical background required for the remainder of the work. The next chapter builds on this foundation by introducing the control-theoretic principles of the two model-based strategies considered in the thesis, namely quaternion-based PD control and NMPC.

Chapter 3

Control-Theoretic Background

3.1 Classical Quaternion-Based PD Control

After formulating the spacecraft attitude-control problem in general terms, the next step is to introduce the theoretical principles of the first control strategy considered in the thesis, namely quaternion-based PD control. This class of controllers is widely used in spacecraft applications because it combines a compact attitude-error representation with a simple feedback structure based on angular-velocity damping and orientation correction [15, 1, 2].

From a broader control perspective, quaternion-based PD control can be viewed as the attitude-control counterpart of the classical PID paradigm. In standard single-axis form, a continuous-time PID law is commonly written as

$$u(t) = K_P e(t) + K_D \dot{e}(t) + K_I \int_0^t e(\tau)\, d\tau, \tag{3.1}$$

where the proportional term reacts to the instantaneous tracking error, the derivative term introduces damping and anticipatory action, and the integral term is typically used to reduce or eliminate steady-state error in the presence of constant biases or modeling mismatch.

In spacecraft attitude control, however, the direct extension of a classical PID structure is not always the most appropriate choice. First, the controlled variable is not a scalar Euclidean coordinate, but a rotational state evolving on a nonlinear manifold. Second, the derivative action is more naturally expressed through the measured body angular velocity rather than through the time derivative of an angle error. Third, and most importantly for the present work, the integral term is often omitted in practical attitude-control formulations when constrained or degraded actuation plays a central role.

The reason is not that integral action is useless in principle, but that it introduces additional issues that are particularly relevant in spacecraft applications. In nominal regulation problems, a well-designed quaternion-based proportional-plus-damping feedback is often sufficient to drive the attitude error to zero when enough control authority is available. By contrast, integral action may accumulate when the requested torque cannot be fully realized because of actuator saturation, underactuation, availability masks, or momentum-related supervisory constraints. In such cases, the integral term may lead to windup and may amplify the mismatch between the nominal control request and the torque that the actuator set can actually deliver. This is especially undesirable in the present thesis, where the benchmark is explicitly underactuated and where magnetic authority is shared with momentum dumping.

For these reasons, the classical controller adopted in this work is formulated as a quaternion-based PD law rather than as a full PID. The proportional part acts on the attitude misalignment, while the derivative part acts on body angular velocity and provides rotational damping. This retains the main stabilizing structure of the classical feedback law while avoiding the additional integral-state dynamics that would complicate the constrained underactuated benchmark considered here.

The interest of quaternion-based PD control lies in its favorable compromise between simplicity and effectiveness. On one side, it avoids the singularities associated with Euler-angle parametrizations. On the other side, it provides an intuitively interpretable feedback action: the proportional term drives the spacecraft toward the desired attitude, while the derivative term damps the rotational motion. For this reason, it represents a natural classical baseline against which more advanced control strategies can be compared.

In the present chapter, the controller is introduced only from a control-theoretic perspective. The practical implementation settings, the selected gains, and the specific benchmark-dependent actuation logic will be presented later in Chapter 4.

3.1.1 From Classical PID Logic to Quaternion-Based PD

The transition from a classical PID law to a quaternion-based PD law can be understood by reinterpreting the three standard feedback roles in the spacecraft attitude-control context.

The proportional role is retained, but the scalar tracking error $e(t)$ in (3.1) is replaced by a rotational error signal derived from the quaternion attitude mismatch. The derivative role is also retained, but instead of differentiating an angle error explicitly, the controller uses the measured body angular velocity, which is directly available from the spacecraft rotational state and is physically more meaningful for damping rigid-body motion. The integral role, on the other hand, is not retained in the controller adopted here.

Therefore, the resulting control architecture should be interpreted as a geometry-consistent specialization of classical feedback logic to the spacecraft attitude problem: proportional action for orientation correction and derivative action for rotational damping, without the integral term that would be more difficult to reconcile with the constrained and underactuated actuation framework considered in this thesis.

3.2 Nonlinear Model Predictive Control

While quaternion-based PD control provides a simple and effective classical baseline, spacecraft attitude control in constrained and degraded-actuation conditions often requires a more structured decision mechanism. In particular, when nonlinear rigid-body dynamics, actuator bounds, rate limits, and momentum-related considerations must be handled simultaneously, it is natural to formulate the control problem as a constrained optimization problem solved repeatedly over time. This is the fundamental idea behind MPC and, in the present nonlinear setting, behind NMPC [3, 4].

In NMPC, the controller uses an internal prediction model of the plant to forecast the future evolution of the system over a finite horizon. At each sampling instant, it computes the control sequence that optimizes a selected performance criterion while satisfying the imposed constraints. Only the first control move of the optimized sequence is actually applied; the optimization is then repeated at the next sampling instant using updated state information. This repeated finite-horizon optimization makes NMPC particularly attractive for spacecraft applications in which constraints and nonlinear couplings are central features of the problem.

The purpose of the present section is to introduce the theoretical principles of NMPC in a controller-independent way. The specific internal model, weights, solver settings, and practical tuning choices adopted in the benchmark case study will be introduced later in Chapter 4.

3.2.1 Receding Horizon Control Principle

The defining feature of NMPC is the receding-horizon strategy. Let t_k denote the current control-update instant and let $\mathbf{x}_k$ be the measured or estimated system state at that instant. Starting from $\mathbf{x}_k$, the controller predicts the future state evolution over a finite horizon by using an internal nonlinear model of the spacecraft dynamics. Over the same horizon, it searches for a control sequence that minimizes a performance index while satisfying state and input constraints [3, 4].

If the optimization horizon contains N_p prediction steps and the manipulated inputs are updated over N_u control moves, the optimization problem can be

interpreted conceptually as

$$\min_{\mathbf{u}_{k|k},\, \mathbf{u}_{k+1|k},\, \ldots} J \tag{3.2}$$

subject to the nonlinear prediction model and to the relevant constraints. The notation $\mathbf{u}_{k+i|k}$ denotes the control action planned for future step $k+i$ using the information available at time t_k.

Once the optimal sequence has been computed, only the first control move is applied to the plant:

$$\mathbf{u}(t_k) = \mathbf{u}^{\star}_{k|k}. \tag{3.3}$$

At the next update time t_{k+1}, a new state estimate is acquired, the prediction horizon is shifted forward, and a new optimization problem is solved. This is why the method is called receding horizon control: the optimization window moves forward in time as the controller operates.

The logic of this strategy is particularly relevant in spacecraft applications. Unlike purely reactive feedback laws, NMPC does not choose the current input only on the basis of the instantaneous error. Instead, it selects the present control action while accounting for its predicted consequences over future evolution. This allows the controller to anticipate the effect of nonlinear coupling, actuator saturation, and trajectory-dependent constraints rather than reacting to them only after they occur.

The receding-horizon principle also provides a convenient way to integrate competing objectives into a single decision framework. Attitude tracking, input moderation, rate smoothing, and momentum-related considerations can all be weighted within the same finite-horizon optimization problem. As a result, NMPC offers a systematic structure for solving control problems in which performance and feasibility must be balanced simultaneously.

3.2.2 Prediction Model Structure

At the core of NMPC lies a prediction model that approximates the dynamics of the controlled system. In the present thesis, the plant of interest is the spacecraft rotational dynamics, whose state is naturally described by attitude and angular-velocity variables, while the control inputs are actuator commands mapped into body torque.

In general terms, the continuous-time nonlinear system can be written as

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t), t)\,, \tag{3.4}$$

where $\mathbf{x}(t)$ is the state vector, $\mathbf{u}(t)$ is the control input, and $\mathbf{d}(t)$ collects exogenous quantities such as disturbances or measured environmental signals. In spacecraft attitude control, the nonlinear function $\mathbf{f}(\cdot)$ typically includes quaternion kinematics,

Euler rigid-body dynamics, and the mapping from actuator commands to control torque.

For digital implementation, the prediction model is used in discrete time:

$$\mathbf{x}_{k+1} = \mathbf{f}_d(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k)\,, \tag{3.5}$$

where $\mathbf{f}_d(\cdot)$ denotes the discrete-time state transition induced by the continuous nonlinear dynamics over one sampling interval. The exact form of $\mathbf{f}_d(\cdot)$ depends on the chosen discretization method and on the controller implementation details, which are benchmark-dependent and are therefore postponed to Chapter 4.

The role of the prediction model is not merely to propagate the state. It also defines the dynamic structure through which present control decisions influence future tracking errors, control effort, and feasibility. If the model captures the main nonlinear couplings of the spacecraft dynamics, then the controller can exploit these couplings when solving the finite-horizon problem. This is a major conceptual difference with respect to classical local feedback laws.

A further point is that the prediction model used in NMPC need not reproduce every physical detail of the plant. In practice, the internal model is a control-oriented representation that should be rich enough to capture the dominant dynamics and constraints relevant to decision making, while remaining tractable for online optimization. The model is therefore a compromise between physical fidelity and computational manageability [4].

In spacecraft attitude applications, this modeling choice is especially important because the dynamics is nonlinear, the attitude representation must remain consistent, and the actuation may depend on structural or environmental conditions. A well-posed NMPC design must therefore select a state representation and a prediction structure that are both dynamically meaningful and numerically suitable for repeated optimization.

3.2.3 Cost Function Formulation

The optimization performed by NMPC is driven by a finite-horizon cost function that quantifies the control objectives. In general, the cost is constructed to penalize deviations from the desired attitude behavior, excessive control usage, and undesired input variations. The resulting objective function reflects the trade-offs that the controller should enforce over the prediction horizon [3, 4].

A generic nonlinear finite-horizon cost can be written as

$$J = \sum_{i=0}^{N_p-1} \ell\Big(\mathbf{x}_{k+i|k}, \mathbf{u}_{k+i|k}, \Delta\mathbf{u}_{k+i|k}\Big) + V_f\Big(\mathbf{x}_{k+N_p|k}\Big)\,, \tag{3.6}$$

where $\ell(\cdot)$ is the stage cost, $V_f(\cdot)$ is an optional terminal cost, and

$$\Delta\mathbf{u}_{k+i|k} = \mathbf{u}_{k+i|k} - \mathbf{u}_{k+i-1|k} \tag{3.7}$$

denotes the input increment.

In control-oriented spacecraft applications, the stage cost is commonly designed to include three main components:

1. a tracking term, which penalizes attitude and angular-velocity error;
2. an input term, which penalizes actuator effort;
3. an input-rate term, which penalizes abrupt command variation.

In compact quadratic form, a common structure is

$$\ell = \left(\mathbf{y}_{k+i|k} - \mathbf{y}_{ref,k+i}\right)^T \mathbf{Q} \left(\mathbf{y}_{k+i|k} - \mathbf{y}_{ref,k+i}\right) + \mathbf{u}_{k+i|k}^T \mathbf{R} \mathbf{u}_{k+i|k} + \Delta \mathbf{u}_{k+i|k}^T \mathbf{S} \Delta \mathbf{u}_{k+i|k}, \tag{3.8}$$

where $\mathbf{Q} \succeq 0$, $\mathbf{R} \succeq 0$, and $\mathbf{S} \succeq 0$ are weighting matrices, and $\mathbf{y}$ denotes the selected output vector relevant to the control objective.

The importance of this structure is not limited to its mathematical convenience. It provides a direct language for encoding the engineering priorities of the controller. Increasing the output-tracking weight emphasizes reference-following accuracy. Increasing the input weight reduces aggressive actuator usage. Increasing the input-rate weight promotes smoother control action and may help preserve actuator compatibility. Thus, the cost function is the formal mechanism through which control priorities are translated into optimization criteria.

In a nonlinear spacecraft problem, the cost function may also be enriched with additional terms related to actuator-related states, soft constraints, or mission-specific priorities. More generally, the cost need not be purely quadratic in all circumstances. Nevertheless, the quadratic form remains a common and useful template because it captures the main control trade-offs while preserving a structure suitable for optimization.

3.2.4 Constraint Handling

A key advantage of NMPC over classical feedback laws is its ability to incorporate constraints explicitly into the control computation. Instead of designing an unconstrained control law and enforcing feasibility only afterward through external saturation logic, NMPC treats constraints as intrinsic parts of the decision problem [3, 4].

In general, the finite-horizon optimization can include:

- input-amplitude constraints,
- input-rate constraints,
- output or state constraints,

- internal actuator-state constraints,
- equality constraints imposed by system structure.

A generic constrained NMPC problem can therefore be written schematically as

$$\begin{aligned} \min_{\{\mathbf{u}_{k+i|k}\}} \quad & J \\ \text{subject to} \quad & \mathbf{x}_{k+i+1|k} = \mathbf{f}_d\Big(\mathbf{x}_{k+i|k}, \mathbf{u}_{k+i|k}, \mathbf{d}_{k+i|k}\Big), \\ & \mathbf{u}_{min} \leq \mathbf{u}_{k+i|k} \leq \mathbf{u}_{max}, \\ & \Delta\mathbf{u}_{min} \leq \Delta\mathbf{u}_{k+i|k} \leq \Delta\mathbf{u}_{max}, \\ & \mathbf{y}_{min} \leq \mathbf{y}_{k+i|k} \leq \mathbf{y}_{max}, \\ & \mathbf{z}_{min} \leq \mathbf{z}_{k+i|k} \leq \mathbf{z}_{max}, \end{aligned} \tag{3.9}$$

for the relevant horizon indices.

The main conceptual consequence is that feasibility and performance are optimized together. If the desired reference cannot be followed exactly without violating the constraints, the controller will choose the best admissible compromise according to the cost-function weights. This is fundamentally different from reactive controllers whose commanded torque may later be clipped or reshaped by the actuators without the controller explicitly accounting for such modifications during design.

Constraint handling is particularly valuable in spacecraft attitude control because many practically relevant limitations are not incidental but structural. Actuator bounds, torque anisotropy, momentum limitations, and smoothness requirements directly affect what the spacecraft can achieve at a given time. By incorporating these limitations inside the optimization, NMPC can reason about feasible future trajectories rather than requesting torque profiles that are only idealized.

In addition, NMPC allows the use of soft constraints when strict satisfaction of a requirement may be occasionally infeasible or unnecessarily restrictive. In that case, a slack-variable mechanism can be introduced so that limited violations are permitted but penalized in the cost function. This extends the flexibility of the controller while preserving a structured compromise between performance and feasibility.

3.2.5 Suitability for Underactuated Spacecraft Control

The theoretical features introduced above make NMPC especially attractive for underactuated spacecraft attitude control. In such problems, the controller must operate under nonlinear dynamics, limited instantaneous torque authority, actuator constraints, and environment-dependent actuation effectiveness. These characteristics align naturally with the strengths of a predictive constrained optimization framework [5, 6].

First, NMPC can exploit the nonlinear prediction model to account for cross-axis dynamic coupling. This is important because, in underactuated systems, the unavailable torque directions may still be influenced indirectly through the rigid-body dynamics and through the time evolution of the state. A purely reactive controller may fail to leverage this structure systematically, whereas NMPC can select current inputs based on their predicted indirect effect over the horizon.

Second, the explicit treatment of actuator and rate constraints is highly relevant when the available control authority is already reduced. In underactuated conditions, every feasible actuation direction becomes more valuable, and the controller must use it carefully. A predictive formulation can modulate tracking aggressiveness, command smoothness, and input allocation in a coordinated way rather than treating these aspects independently.

Third, NMPC can incorporate additional control priorities that are difficult to handle consistently within a simple feedback law. For example, short-term tracking objectives may need to be balanced against long-term actuator operability, or torque directions may need to be prioritized differently depending on the current configuration. Within the NMPC framework, these trade-offs can be expressed through appropriate costs and constraints rather than being left to ad hoc supervisory logic.

For these reasons, NMPC is a theoretically well-suited candidate for degraded spacecraft attitude control. Its main drawback is not conceptual adequacy, but computational demand. Because a nonlinear constrained optimization problem must be solved online at every update step, the achievable performance must always be assessed together with the associated computational burden. This trade-off is central to the thesis: NMPC is treated as the expert controller because of its constraint-aware and model-aware capabilities, but also as the source of the computational challenge that motivates the later data-driven approximation.

In summary, NMPC provides a principled way to control nonlinear constrained spacecraft dynamics under reduced actuation authority. It is precisely this combination of theoretical suitability and practical computational cost that makes it the pivotal control strategy in the comparative logic of the thesis.

3.3 Control Strategies in the Thesis Context

The previous sections introduced the two model-based control paradigms considered in this thesis: a classical quaternion-based PD controller and an NMPC strategy. At this stage, however, it is also necessary to clarify why these two controllers are selected, what roles they play within the comparative framework, and why their comparison naturally motivates the later introduction of a data-driven surrogate.

The thesis is not intended as a generic survey of spacecraft attitude-control

methods. Rather, it focuses on a specific engineering question: how different control paradigms behave when spacecraft attitude control must be performed under reduced actuation authority, practical constraints, and limited on-board computational resources. Within this perspective, the considered controllers are not arbitrary alternatives, but complementary reference points in a structured comparison.

3.3.1 Why PD as Baseline

Quaternion-based PD control is adopted as the classical baseline because it represents a widely understood, physically meaningful, and computationally lightweight solution for spacecraft attitude control [15, 1, 2]. Its structure is directly tied to the geometry of the attitude problem: the proportional action acts on the quaternion error, while the derivative action damps the angular motion. This makes the controller both interpretable and easy to relate to the underlying rigid-body dynamics.

From a comparative standpoint, the importance of the PD controller lies in its transparency. Because the feedback structure is simple, the relationship between error signals, gain selection, and commanded torque is comparatively easy to interpret. This is useful not only from a design perspective, but also from an evaluation perspective: the PD controller provides a clear lower-complexity reference against which the benefits of more advanced strategies can be assessed.

The PD baseline is also relevant because, in practical spacecraft engineering, simple controllers are often attractive precisely for the reasons that optimization-based methods are challenging: they require limited computation, they are straightforward to implement, and they can often be validated with comparatively modest design effort. Therefore, showing that a more advanced controller performs better is not, by itself, sufficient. The improvement must be interpreted relative to the simplicity and operational attractiveness of a classical baseline.

At the same time, using PD as baseline makes the limitations of purely reactive feedback especially visible in degraded-actuation conditions. When actuator feasibility, saturation, and momentum-related constraints become central, the gap between nominal commanded torque and physically realizable actuation may become significant. This is precisely why the PD controller is a meaningful reference: it is strong enough to be relevant, yet simple enough that its limitations under constrained underactuated conditions remain clearly identifiable.

3.3.2 Why NMPC for Fault-Tolerant Control

NMPC is adopted as the expert model-based strategy because its theoretical structure is particularly well aligned with the requirements of degraded spacecraft

attitude control. As discussed in Section 3.2, the method can explicitly incorporate nonlinear dynamics, input and rate constraints, and additional control priorities within a unified receding-horizon optimization framework [3, 4].

This makes NMPC especially attractive in fault-tolerant or underactuated scenarios. When the available actuator set cannot instantaneously produce arbitrary three-axis torque, the controller must do more than react to the current error. It must exploit the residual actuation authority, the nonlinear coupling of the rotational dynamics, and the future evolution of the system in order to determine the most effective admissible control action. These are precisely the kinds of tasks for which predictive control is theoretically well suited [5, 6].

Within the logic of the thesis, NMPC is not chosen simply because it is more sophisticated than PD. It is chosen because it provides a principled reference for what a constraint-aware, model-aware controller can achieve when applied to the same spacecraft problem. In other words, it represents the upper-structure benchmark in the comparison: the controller that is best positioned, at least conceptually, to manage the complexity of the problem formulation.

This role is particularly important in a thesis concerned with underactuated control. If the comparison were limited to simple feedback designs, it would be difficult to determine whether the observed limitations arise from the physical problem itself or from the chosen simplicity of the controller. By contrast, introducing NMPC provides a stronger reference point: if a difficulty remains even within a predictive constrained framework, then it is more likely to reflect a genuine structural challenge of the control problem rather than a weakness of a particular low-complexity design.

Therefore, NMPC is used in the thesis as the expert controller in both senses of the term: expert because of its richer control logic, and expert because it provides the control behavior that later serves as the basis for the data-driven imitation framework.

3.3.3 Motivation for the Later Data-Driven Approximation

The comparison between PD and NMPC naturally exposes the central trade-off that motivates the later neural-network-based controller. On one side, the PD controller is simple and computationally cheap, but structurally limited in how it handles constraints and degraded actuation. On the other side, NMPC offers a much richer decision framework, but at the cost of repeated online nonlinear optimization.

This trade-off suggests a natural question: whether part of the control behavior produced by the expert predictive controller can be approximated by a much cheaper policy evaluation mechanism. This is the point where imitation learning becomes relevant. In imitation learning, an expert policy is first used to generate state-action

demonstrations, and a parameterized function is then trained to reproduce the mapping from observed inputs to commanded outputs [9, 10, 11].

A feed-forward NN is particularly attractive in this context because of its expressive approximation capability and low online evaluation cost. Classical universal-approximation results show that sufficiently rich neural architectures can approximate broad classes of nonlinear mappings under suitable assumptions [12, 13]. In the present thesis, the interest is not in approximation theory for its own sake, but in its control relevance: if the input-output behavior of the NMPC expert can be learned with acceptable fidelity, then the resulting neural controller may retain some of the expert's useful behavior while being significantly easier to evaluate online.

This does not mean that the neural controller is expected to inherit all the theoretical guarantees or structural properties of the expert predictive controller. The point is more pragmatic. The neural controller is introduced as a possible compromise between the two extremes already established in this chapter: the structural simplicity of classical feedback and the constraint-aware sophistication of predictive optimization.

Accordingly, the control logic of the thesis becomes hierarchical. The PD controller serves as the classical baseline, the NMPC controller serves as the expert model-based reference, and the neural controller is introduced later as a data-driven surrogate intended to approximate the expert behavior at a reduced computational cost. This hierarchy is central to the comparative methodology of the work and explains why the three strategies should be interpreted as connected stages of the same investigation rather than as unrelated controller choices.

This concludes the control-theoretic background of the thesis. On this basis, the next chapter translates the previous theoretical concepts into the concrete benchmark used throughout the work, including the mission assumptions, the simulation framework, the degraded-actuation architecture, and the practical implementation settings of the considered controllers.

Chapter 4

Benchmark Definition and Controller Implementation

4.1 Case Study Definition

The theoretical framework introduced in Chapters 2 and 3 is now specialized to the benchmark adopted throughout the thesis. The purpose of this chapter is to define the concrete spacecraft configuration, the orbital environment, the degraded-actuation condition, and the practical controller settings used in the comparative analysis. In contrast with the previous chapters, the present one is therefore benchmark-specific and collects the numerical assumptions that make the comparison reproducible.

The case study is built to represent a spacecraft attitude-control problem in which accurate tracking must be maintained despite reduced reaction-wheel availability and shared use of the magnetic actuation subsystem for both auxiliary control and momentum management.

4.1.1 Mission and Spacecraft Assumptions

The benchmark considers a rigid spacecraft operating in Earth orbit and commanded to follow an attitude-guidance profile composed of successive pointing phases and reorientation maneuvers. The spacecraft is modeled as a single rigid body with constant mass and constant inertia properties over the full simulation horizon. Flexible dynamics, mass variation, and structural reconfiguration effects are neglected, since they are outside the scope of the present control-comparison study.

All rotational quantities are expressed in the body frame introduced in Chapter 2. The body frame is assumed aligned with the principal axes of inertia, so that the

inertia matrix is diagonal. The spacecraft mass is fixed to

$$m = 200 \text{ kg}, \tag{4.1}$$

and the rigid-body inertia matrix adopted throughout the benchmark is

$$\mathbf{J} = \text{diag}(9.7,\ 7.2,\ 16.8) \text{ kg m}^2. \tag{4.2}$$

These values are not introduced as a generic statement about all small satellites, nor as the result of an original spacecraft preliminary-design activity carried out within this thesis. Rather, the benchmark configuration is derived from the internal *generic* spacecraft configuration adopted within CUS-GNC, from which the reference mass properties, actuator architecture, and nominal actuator parameter ranges were inherited. In this sense, the case study should be interpreted as a control-oriented benchmark built on top of an existing internal spacecraft baseline, rather than as a mission-specific flight design.

4.1.2 Orbital and Environmental Conditions

The benchmark is defined on a near-circular low-Earth orbit used to generate the environmental quantities required by the simulator and by the actuation logic. The adopted orbital parameters are

$$h = 500 \text{ km}, \qquad e = 0.0011, \qquad i = 96.79^\circ, \tag{4.3}$$

with right ascension of the ascending node, argument of perigee, and initial true anomaly set to zero in the reference propagation setup. The corresponding semi-major axis is defined as

$$a = R_E + h, \tag{4.4}$$

where R_E is the Earth mean radius.

At the same time, it is important to clarify the scope of this modeling choice. The orbital dynamics is not a research focus of the present thesis and is not treated here as an original modeling contribution. Instead, the orbital propagation block was reused from the simulator framework developed in [16] and embedded in the present benchmark because a consistent orbital state was required to generate environment-dependent quantities for the attitude-control simulations.

Accordingly, the orbital model is used here only as an auxiliary environmental driver. Its role is to provide the propagated position and velocity needed for the local orbital geometry, the gravity-gradient torque evaluation, and the time-varying geomagnetic field used by the MTQs and by the residual magnetic disturbance model. The thesis does not investigate orbital-control aspects, orbit determination, or a dedicated analysis of orbital perturbations within this block.

The geomagnetic field is evaluated through the IGRF-14 model at the propagated spacecraft location and then rotated into the body frame using the current spacecraft attitude [17]. In this way, the orbital propagation enters the benchmark only indirectly, through the environmental quantities required by the attitude-control problem.

4.1.3 Spacecraft Inertia and Actuator Configuration

The nominal actuator suite adopted in the benchmark comprises four RWs arranged in a pyramidal configuration and three orthogonal MTQs. The reaction-wheel geometry is defined through the configuration matrix

$$\mathbf{Z} = \begin{bmatrix} \cos\beta & 0 & -\cos\beta & 0 \\ 0 & \cos\beta & 0 & -\cos\beta \\ \sin\beta & \sin\beta & \sin\beta & \sin\beta \end{bmatrix}, \qquad \beta = 30^\circ, \tag{4.5}$$

which corresponds to the pyramidal arrangement used throughout the thesis.

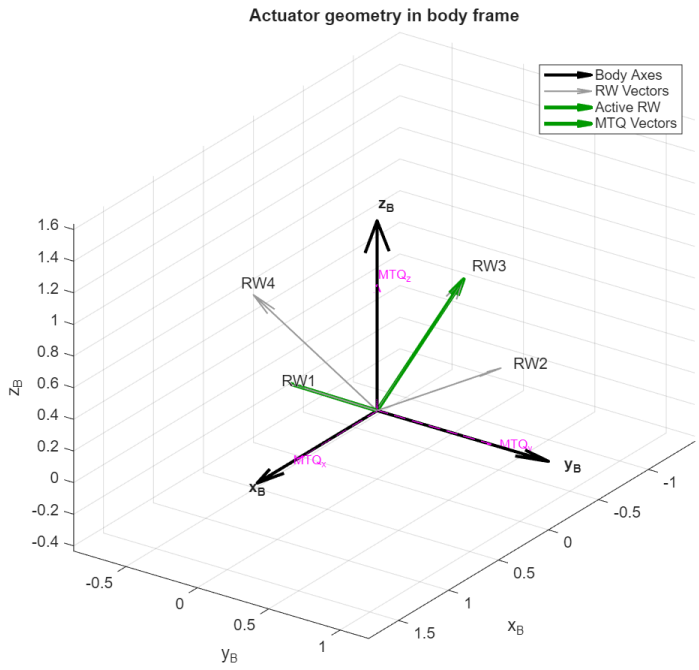


Figure 4.1: Actuator geometry adopted in the benchmark body frame. The figure shows the RWs spin axes and the three orthogonal MTQs dipole directions used in the common spacecraft actuation model.

Each wheel is modeled with the same physical parameters. The maximum motor torque per wheel is

$$\tau_{rw,\max} = 0.2 \text{ N m}, \tag{4.6}$$

the wheel-speed saturation is

$$\Omega_{\max} = 7500 \text{ rpm}, \tag{4.7}$$

the rotor inertia is

$$I_{rw} = 0.02 \text{ kg m}^2, \tag{4.8}$$

and the wheel torque dynamics is represented with a first-order time constant

$$\tau_{rw} = 0.1 \text{ s}. \tag{4.9}$$

The MTQ subsystem is modeled as a three-axis dipole generator with identical parameters on all axes. The commanded dipole is bounded component-wise by

$$|m_i| \leq 200 \text{ A m}^2, \qquad i \in \{x, y, z\}, \tag{4.10}$$

with coil parameters

$$N = 500, \qquad A = 0.03 \text{ m}^2, \qquad NA = 15 \text{ turns} \cdot \text{m}^2, \tag{4.11}$$

which correspond to a maximum current per axis of approximately

$$I_{\max} \approx 13.33 \text{ A}. \tag{4.12}$$

When actuator dynamics is enabled, the magnetic dipole response is modeled through a first-order electrical dynamics with time constant

$$\tau_e = 0.2 \text{ s}, \tag{4.13}$$

an internal update step of

$$\Delta t_{mtq} = 0.01 \text{ s}, \tag{4.14}$$

and an actuation delay of

$$t_{delay} = 0.5 \text{ s}. \tag{4.15}$$

The reaction-wheel and magnetorquer parameters adopted here are consistent with the same internal CUS-GNC generic configuration used as benchmark baseline for the thesis. These actuator parameters therefore define the common physical benchmark on top of which the different controllers are compared. Their role is structural: they are not controller-specific tuning values, but part of the plant and actuator model shared by all control strategies.

4.1.4 Failure Scenario and Underactuated Architecture

The control benchmark is intentionally defined in a degraded-actuation condition. Although the nominal spacecraft carries four RWs, the comparative simulations are performed under a permanent fault configuration in which only RW 1 and RW 3 remain available. This condition is enforced through the availability mask

$$\mathbf{M}_f = \mathrm{diag}(1,\ 0,\ 1,\ 0), \tag{4.16}$$

so that RW 2 and RW 4 are disabled over the entire simulation horizon.

Accordingly, the body torque generated by the wheel set becomes

$$\boldsymbol{\tau}_{rw}(t) = -\,\mathbf{Z}\,\mathbf{M}_f\,\mathbf{u}_{rw}(t), \tag{4.17}$$

where $\mathbf{u}_{rw}(t) \in \mathbb{R}^4$ is the full wheel-command vector. Because only the first and third wheel columns remain active, the directly available wheel-torque subspace is reduced. In the adopted geometry, the surviving wheel pair spans the x_B-z_B plane, which means that a direct wheel-generated torque about the body y axis is no longer available.

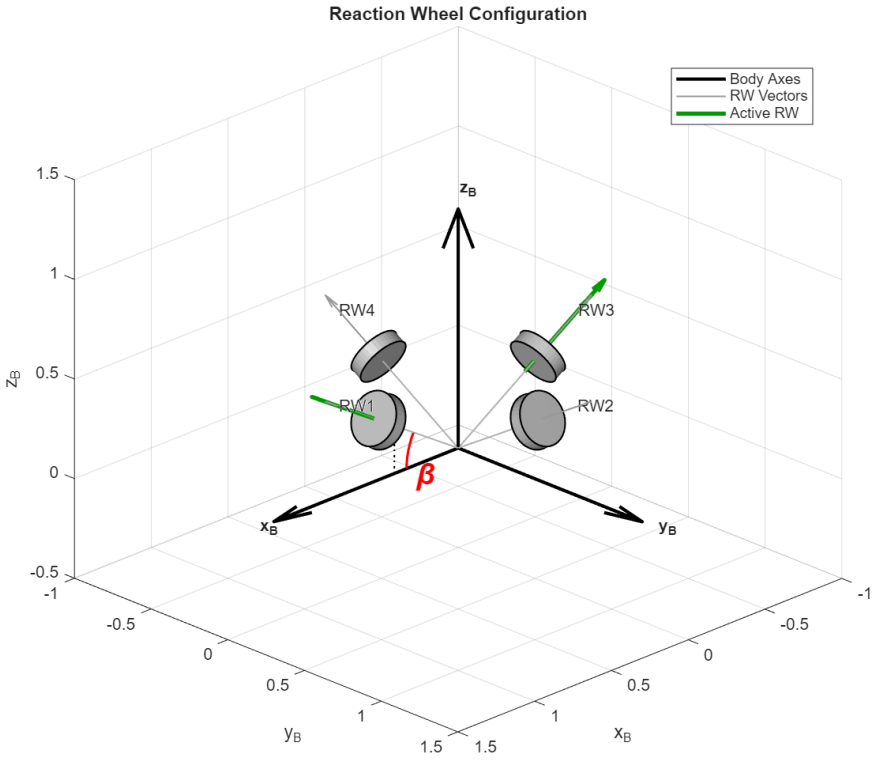


Figure 4.2: Pyramidal RW configuration adopted in the benchmark. In the permanent fault case considered in this work, only RW 1 and RW 3 remain active, which reduces the instantaneous wheel-torque authority to a two-dimensional subspace.

This is the structural reason why the benchmark is genuinely underactuated. The primary controller cannot rely on full three-axis wheel authority and must instead operate under a reduced torque interface. Auxiliary magnetic actuation remains available and is used within the common benchmark architecture for two purposes:

1. reaction-wheel momentum dumping;
2. limited attitude-assist action on the weakly controlled axis.

No mid-simulation fault injection is introduced in the comparison setup. The spacecraft starts directly in the degraded configuration and remains in that configuration for the whole run. This choice is deliberate, because it isolates the sustained underactuated-control problem of interest and avoids mixing transient fault-detection or fault-transition effects with the controller-comparison objective.

4.2 Guidance and Simulation Scenario Definition

Once the benchmark spacecraft and actuator architecture have been fixed, the comparative framework must be completed by defining the commanded attitude profile, the main simulation assumptions, the disturbance environment, and the performance indicators used in the analysis. These elements are shared by all controllers considered in the thesis and therefore constitute the common operational scenario on which the comparison is built.

The role of the present section is not to introduce new theoretical concepts, but to specify how the benchmark problem is actually exercised in simulation. In particular, the orbital propagation should be interpreted only as a supporting source of environment-dependent inputs for the attitude benchmark, not as a separate object of analysis within the thesis. In this sense, the guidance profile, the initial conditions, the disturbance environment, and the assessment metrics should all be interpreted as common test conditions rather than controller-specific design choices.

4.2.1 Attitude Guidance Profile

The benchmark guidance is assigned directly through a prescribed quaternion reference profile generated by the guidance block. The controller therefore receives the measured spacecraft attitude $q(t)$, the body angular velocity $\boldsymbol{\omega}(t)$, and the desired reference quaternion $q_{\mathrm{ref}}(t)$, consistently under the scalar-first quaternion convention adopted throughout the thesis.

Unless otherwise stated, the desired angular velocity is assumed to be

$$\boldsymbol{\omega}_{\mathrm{ref}}(t) = \mathbf{0}, \tag{4.18}$$

so that each commanded attitude change is interpreted as a rest-to-rest reorientation followed by a pointing phase.

The reference profile used throughout the comparative benchmark is piecewise constant and is defined as

$$q_{\mathrm{ref}}(t) = \begin{cases} \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T, & 0 \le t < 2600 \text{ s}, \\ \begin{bmatrix} 0.8349 & -0.1206 & 0.4236 & 0.33 \end{bmatrix}^T, & 2600 \le t < 4200 \text{ s}, \\ \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T, & t \ge 4200 \text{ s}. \end{cases} \tag{4.19}$$

This sequence defines three consecutive phases. The first phase corresponds to a nominal identity-attitude pointing segment. The second phase imposes a commanded reorientation toward a nontrivial target attitude. The third phase commands a return to the identity quaternion, thereby introducing a second reorientation maneuver followed by a final pointing segment.

The adopted profile is intentionally simple in structure but sufficiently rich to test the controllers under both steady-pointing and transient reorientation conditions. Because the same sequence is applied to all strategies, it provides a common basis for comparing tracking quality, transient damping, interaction with actuator bounds, and momentum-management behavior.

4.2.2 Initial Conditions and Simulation Horizon

The initial spacecraft attitude is set to

$$q(0) = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \end{bmatrix}^T, \tag{4.20}$$

which represents a nontrivial initial misalignment with respect to the first commanded reference. This choice ensures that the benchmark begins with an actual closed-loop correction phase rather than with the spacecraft already aligned with the reference.

The simulations are performed in discrete time with a controller update period equal to

$$T_s = 0.1 \text{ s}, \tag{4.21}$$

while the plant and actuator dynamics are integrated with fixed internal step

$$h_{int} = 0.01 \text{ s}. \tag{4.22}$$

This separation between controller rate and internal propagation step is adopted consistently throughout the benchmark in order to preserve numerical robustness while maintaining a realistic control-update frequency.

The commanded reference switches at

$$t_1 = 2600 \text{ s}, \qquad t_2 = 4200 \text{ s}, \tag{4.23}$$

so that the simulation horizon always includes:

1. an initial convergence segment toward the identity attitude;
2. a first commanded reorientation toward the intermediate target quaternion;
3. a second commanded reorientation back to the identity attitude.

In the current benchmark definition, the comparison is therefore structured around these three guidance phases and the associated transients. The exact aggregation of performance over the full run, including the final post-maneuver dwell interval, is discussed together with the results campaign so that the same evaluation windows can be applied consistently to all controllers.

4.2.3 Disturbance and Magnetic Environment Settings

The environmental quantities required by the simulation are generated from the orbital setup defined in Section 4.1. In particular, the geomagnetic field is evaluated along the propagated orbit through the IGRF model and then rotated into the body frame using the current spacecraft attitude [17]. The resulting body-frame magnetic field vector is used consistently both for MTQ torque generation and for residual magnetic disturbance evaluation.

Within the scope of the thesis, these quantities are introduced only insofar as they affect the attitude-control problem. No dedicated orbital-perturbation analysis is carried out at orbit-dynamics level; rather, the propagated orbit is used as an exogenous input source for the environmental terms entering the rotational dynamics.

The baseline disturbance model adopted in the comparative benchmark is

$$\boldsymbol{\tau}_d(t) = \boldsymbol{\tau}_{gg}(t) + \boldsymbol{\tau}_{mag}(t), \tag{4.24}$$

that is, the total environmental disturbance torque is taken as the sum of gravity-gradient torque and residual magnetic dipole torque, while other perturbations are neglected at this stage.

The residual magnetic disturbance is modeled as

$$\boldsymbol{\tau}_{mag}(t) = \mathbf{m}_{res} \times \mathbf{B}_B(t), \tag{4.25}$$

with constant residual magnetic dipole

$$\mathbf{m}_{res} = \begin{bmatrix} 0.05 & 0.05 & 0.05 \end{bmatrix}^T \text{ A m}^2. \tag{4.26}$$

The gravity-gradient contribution is computed through the standard rigid-body expression [1, 2]

$$\boldsymbol{\tau}_{gg}(t) = \frac{3\mu}{\|\mathbf{r}(t)\|^5}\, \mathbf{r}_B(t) \times \Big(\mathbf{J}\, \mathbf{r}_B(t)\Big), \tag{4.27}$$

where μ is the central-body gravitational parameter and $\mathbf{r}_B(t)$ is the spacecraft position vector expressed in the body frame.

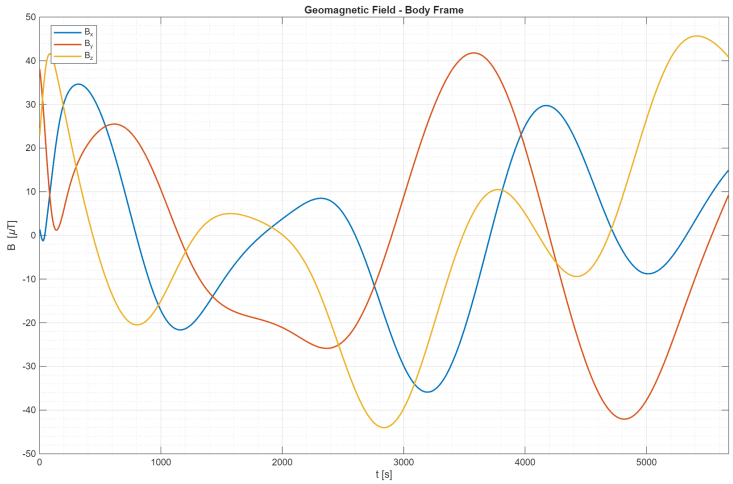


Figure 4.3: Example geomagnetic-field profile along the benchmark orbit.

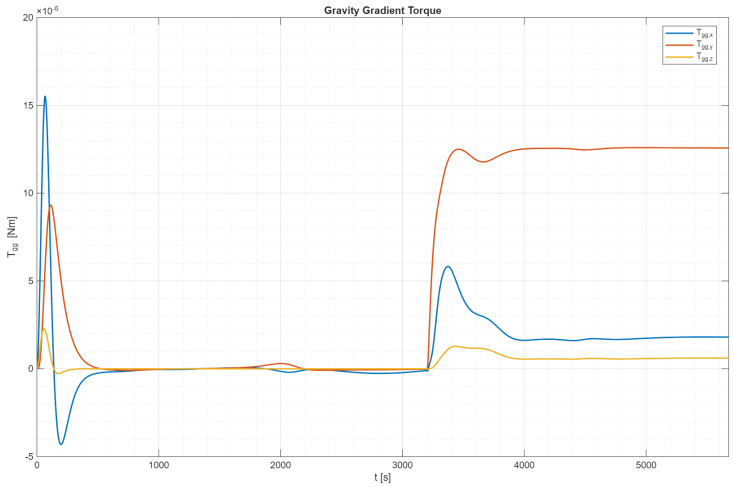


Figure 4.4: Representative gravity-gradient torque profile generated under the benchmark orbit and inertia assumptions. The plot is reported here to document the order of magnitude and time variation of the mechanical disturbance entering the comparative simulations.

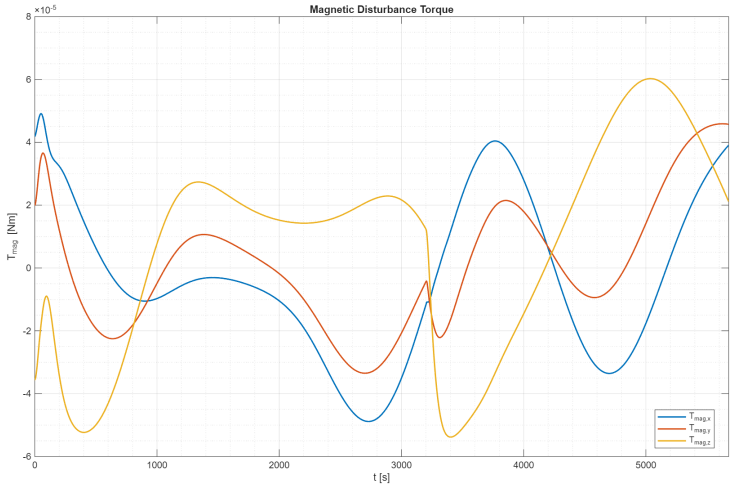


Figure 4.5: Representative residual magnetic disturbance torque profile associated with the benchmark residual dipole and the propagated geomagnetic field. The plot highlights the time-varying perturbation induced by magnetic-environment interaction.

These disturbances are intentionally moderate in complexity. Their function in the benchmark is not to maximize environmental fidelity, but to retain the perturbation mechanisms most relevant to the present control problem:

1. gravity-gradient torque as a persistent orbit-dependent mechanical perturbation;

2. magnetic disturbance torque as the direct counterpart of the same geomagnetic field that also governs MTQ authority.

This choice is consistent with the aim of the thesis, namely evaluating the control architectures under a common disturbance environment that remains sufficiently realistic to affect tracking and momentum accumulation, while still preserving a clear interpretation of the controller comparison.

4.2.4 Performance Assessment Framework

Controller performance is assessed through a common set of time-domain indicators computed over each simulation run. The comparative framework is intentionally multi-dimensional: it does not focus only on attitude-tracking accuracy, but also on angular-rate regulation, interaction with actuator limitations, and momentum-management behavior. Implementation-oriented considerations motivate the controller selection discussed throughout the thesis, but a dedicated runtime benchmark is not part of the reported assessment framework.

The attitude-tracking error is evaluated from the quaternion error

$$q_e(t) = q_{\text{ref}}(t) \otimes q^{-1}(t), \tag{4.28}$$

with sign consistency enforced by selecting the representation with non-negative scalar part. The scalar attitude indicator used throughout the results chapter is then

$$\theta_{err}(t) \equiv e_q(t) = 2\arccos(q_{e,0}(t)), \qquad q_{e,0}(t) \geq 0, \tag{4.29}$$

while the angular-rate error is defined as

$$e_\omega(t) = \|\boldsymbol{\omega}(t) - \boldsymbol{\omega}_{\text{ref}}(t)\|_2 . \tag{4.30}$$

Because the benchmark guidance adopts $\boldsymbol{\omega}_{\text{ref}}(t) = \mathbf{0}$, the scalar rate quantity reported in the results chapter is simply

$$\|\boldsymbol{\omega}(t)\|_2 \equiv e_\omega(t). \tag{4.31}$$

On this basis, the comparative assessment includes:

- attitude-tracking indicators, such as peak and root-mean-square quaternion-based error;
- transient indicators, such as settling behavior during commanded reorientation phases;
- actuation-related indicators, including boundedness of wheel torque and magnetic dipole commands;
- momentum-related indicators, including peak RW momentum and dumping activity.

In the requirement-style metrics later reported as `timeInBand(`θ`)` and `timeInBand(`ω`)`, the benchmark conditions are

$$\theta_{err}(t) \leq \theta_{\text{band}}, \qquad \|\boldsymbol{\omega}(t)\|_2 \leq \omega_{\text{band}},$$

with

$$\theta_{\text{band}} = 1^\circ, \qquad \omega_{\text{band}} = 0.01^\circ/\text{s}.$$

The metric definitions themselves are fixed at benchmark level and are shared by all controllers. Runtime benchmarking is instead left outside the scope of the present reported results and is discussed only as an implementation-oriented future extension.

Table 4.1: Scenario-definition and assessment settings adopted in the comparative benchmark.

Setting	Symbol	Value	Unit
Guidance			
Desired angular velocity	$\boldsymbol{\omega}_{\text{ref}}$	$\mathbf{0}$	rad/s
Initial attitude	$q(0)$	$\begin{bmatrix}0.5 & 0.5 & 0.5 & 0.5\end{bmatrix}^T$	–
Reference segment 1	$q_{\text{ref},1}$	$\begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^T$	–
Reference segment 2	$q_{\text{ref},2}$	$\begin{bmatrix}0.8349 & -0.1206 & 0.4236 & 0.33\end{bmatrix}^T$	–
Reference segment 3	$q_{\text{ref},3}$	$\begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^T$	–
First switch time	t_1	2600	s
Second switch time	t_2	4200	s
Simulation			
Controller sample time	T_s	0.1	s
Internal integration step	h_{int}	0.01	s
Environment and disturbances			
Magnetic-field model	–	IGRF-14	–
Included perturbations	–	Gravity-gradient torque + residual magnetic torque	–
Residual magnetic dipole	$\mathbf{m}_{\text{res}}$	$\begin{bmatrix}0.05 & 0.05 & 0.05\end{bmatrix}^T$	A·m^2
Assessment			
Attitude error metric	$\theta_{err}(t) \equiv e_q(t)$	$2\arccos\big(q_{e,0}(t)\big)$	rad
Rate error metric	$\|\boldsymbol{\omega}(t)\|_2 \equiv e_\omega(t)$	$\|\boldsymbol{\omega}(t)\|_2$	rad/s
Attitude compliance band	θ_{band}	1	deg
Rate compliance band	ω_{band}	0.01	deg/s
Momentum indicators	–	Peak momentum and dumping activity	–

With Sections 4.1 and 4.2, the benchmark problem is fully specified at spacecraft, environment, and scenario level. The next sections introduce how the PD and NMPC controllers are concretely implemented within this common framework.

4.3 PD Controller Implementation

The quaternion-based PD controller introduced in Chapter 3 is now specialized to the benchmark defined in Sections 4.1 and 4.2. In the present chapter, the objective is no longer to discuss the controller from a purely theoretical perspective, but to document how it is actually implemented in the comparative simulation framework, including gain selection, actuator interfacing, and the supervisory logic required by

the degraded actuation architecture.

In the adopted benchmark, the PD controller remains the primary source of attitude-control torque request. However, because the spacecraft is operated in a permanently underactuated reaction-wheel configuration, the final control action applied to the plant is not obtained through direct three-axis wheel authority. Instead, the nominal PD torque must be mapped onto the reduced wheel set and complemented by a shared MTQ management architecture that also handles momentum dumping. For this reason, the implemented baseline should be interpreted as a practical spacecraft-control architecture rather than as an isolated textbook PD feedback law.

4.3.1 Practical Control Architecture

The implemented PD baseline follows the same geometric logic introduced in Section 3.1. At each controller update instant, the measured quaternion and body angular velocity are used to compute the quaternion error

$$q_e(t) = q_{\mathrm{ref}}(t) \otimes q^{-1}(t), \tag{4.32}$$

with sign consistency enforced so that the scalar component satisfies

$$q_{e,0}(t) \geq 0. \tag{4.33}$$

The commanded body torque is then generated through the quaternion-based feedback law

$$\boldsymbol{\tau}_{cmd}(t) = -\mathbf{K}_p\, \mathbf{q}_{e,v}(t) - \mathbf{K}_d\, \boldsymbol{\omega}(t), \tag{4.34}$$

where the benchmark reference angular velocity is zero, consistently with the rest-to-rest guidance profile introduced in Section 4.2.

The implemented architecture is multi-rate. The main controller is updated at the benchmark sampling time

$$T_s = 0.1 \text{ s}, \tag{4.35}$$

while the plant and actuator internal dynamics are propagated using the finer internal step

$$h_{int} = 0.01 \text{ s}. \tag{4.36}$$

This choice is retained consistently for all considered control strategies so that the comparison is not biased by inconsistent simulation timing.

From a structural viewpoint, the implemented PD control chain consists of four stages:

1. computation of quaternion and angular-velocity feedback torque;

2. allocation of the achievable part of the requested torque to the available RWs;

3. generation of an auxiliary MTQ support action on the weakly controlled axis;
4. magnetic-priority supervision to combine attitude-assist and momentum-dumping requests.

This decomposition is important because, in the present underactuated benchmark, the PD law alone does not directly determine the final actuator commands. Instead, it provides the nominal attitude-control request that is subsequently reshaped by the reduced wheel geometry, magnetic inversion, and shared actuator supervisor.

4.3.2 Gain Selection and Tuning Logic

The main quaternion-based feedback gains are selected according to the inertia-scaled eigenaxis rationale discussed by Wie, Weiss, and Arapostathis [15]. The implemented gain structure is

$$\mathbf{K}_d = 2\zeta\omega_n \mathbf{J}, \qquad \mathbf{K}_p = 2\omega_n^2 \mathbf{J}, \tag{4.37}$$

which preserves the same inertia-weighted formulation adopted in the theoretical chapter.

In the benchmark implementation, the tuning objective is intentionally conservative. The PD baseline is not designed to produce aggressive minimum-time maneuvers, but to provide a smooth and well-damped reference controller under degraded actuation. Accordingly, the tuning was guided by the following practical requirements:

$$\zeta = 1, \qquad T_s^{des} = 150 \text{ s}, \tag{4.38}$$

which correspond to a critically damped design with maneuver times on the order of a few hundred seconds. Using the standard second-order approximation, this leads to

$$\omega_n = \frac{4}{T_s^{des}\zeta} = 0.0267 \text{ rad/s}, \tag{4.39}$$

and therefore to the scalar coefficients

$$d = 2\zeta\omega_n = 0.0533 \text{ s}^{-1}, \qquad k = 2\omega_n^2 = 1.422 \times 10^{-3} \text{ s}^{-2}. \tag{4.40}$$

With the benchmark inertia matrix

$$\mathbf{J} = \text{diag}(9.7,\ 7.2,\ 16.8) \text{ kg m}^2, \tag{4.41}$$

the resulting implemented gain matrices are

$$\mathbf{K}_p = \text{diag}(0.01380,\ 0.01024,\ 0.02389), \tag{4.42}$$

$$\mathbf{K}_d = \mathrm{diag}(0.5173,\ 0.3840,\ 0.8960). \tag{4.43}$$

These values were retained because they already provided a smooth response consistent with the intended benchmark timescale, while avoiding unnecessarily aggressive torque demand in the underactuated configuration.

The auxiliary magnetic support gains are tuned separately. Unlike the main PD channel, the MTQ support does not follow a full analytical eigenaxis synthesis, since magnetic actuation is inherently time-varying, anisotropic, and constrained by the local geomagnetic field. For this reason, the auxiliary magnetic gains are selected heuristically through nonlinear simulation refinement, in a spirit consistent with the simulation-oriented tuning rationale adopted in magnetic-control studies such as [8]. The final retained values for the weak-axis magnetic support are

$$K_{p,y}^{att} = 3.5 \times 10^{-2}, \qquad K_{d,y}^{att} = 1.5. \tag{4.44}$$

These gains are intentionally lightweight. Their purpose is not to replace the main wheel-based attitude control, but only to provide a complementary correction on the direction that is poorly actuated by the surviving wheel pair.

4.3.3 Applied Saturations and Interface with Actuators

In the benchmark configuration, only RW1 and RW3 are available. Therefore, the requested body torque produced by (4.34) must be projected onto the torque subspace spanned by the active wheel geometry. If

$$\mathbf{Z}_{act} = \mathbf{Z}(:, [1,3]), \tag{4.45}$$

the active-wheel torque command is computed through the minimum-norm least-squares allocation

$$\mathbf{u}_{rw,act}(t) = -\,\mathrm{pinv}(\mathbf{Z}_{act})\,\boldsymbol{\tau}_{cmd}(t). \tag{4.46}$$

The resulting two-wheel command is then embedded into the full four-wheel command vector by assigning zero torque to the failed channels and enforcing the per-wheel motor torque bound

$$|u_{rw,i}(t)| \le \tau_{rw,\max} = 0.2\ \mathrm{N\,m}. \tag{4.47}$$

Because the surviving wheel pair spans only the x_B-z_B plane in the adopted benchmark geometry, the PD law cannot realize a direct wheel torque about the body y axis. For this reason, a limited auxiliary MTQ support action is introduced only along that weak direction:

$$\tau_{y,att}(t) = -K_{p,y}^{att}\, q_{e,2}(t) - K_{d,y}^{att}\, \omega_y(t). \tag{4.48}$$

The corresponding auxiliary body-torque request is

$$\boldsymbol{\tau}_{att}^{mtq}(t) = \begin{bmatrix} 0 & \tau_{y,att}(t) & 0 \end{bmatrix}^T, \tag{4.49}$$

which is converted into a raw magnetic dipole request through the minimum-norm inversion of the magnetic torque relation:

$$\mathbf{m}_{att}^{raw}(t) = \frac{\mathbf{B}_B(t) \times \boldsymbol{\tau}_{att}^{mtq}(t)}{\|\mathbf{B}_B(t)\|_2^2}, \tag{4.50}$$

provided that the magnetic-field norm remains above the internal numerical threshold adopted in the simulator.

The same MTQ subsystem is also used for reaction-wheel momentum dumping. During dumping, the unloading wheel torque is generated through the proportional law

$$\tau_{rw,desat,i}(t) = K_{p,\mathrm{dump}}\, J_{rw,i}\, (\omega_{ref,i} - \omega_{rw,i}(t)), \tag{4.51}$$

with

$$K_{p,\mathrm{dump}} = 5 \times 10^{-4}, \tag{4.52}$$

and the corresponding body desaturation torque is then mapped into a magnetic dipole request.

Since attitude assistance and dumping share the same MTQ authority, a supervisory priority logic is required. Dumping is assigned priority, and the remaining available dipole margin is computed after component-wise saturation of the unloading request. During simultaneous dumping, the attitude-assist dipole is reduced by the factor

$$\gamma_d = \begin{cases} 1, & \text{if dumping is inactive,} \\ 0.2, & \text{if dumping is active.} \end{cases} \tag{4.53}$$

The final magnetic command is thus obtained by combining the dumping and assist requests under residual-authority saturation and global dipole bounds:

$$|m_i(t)| \leq 200\ \mathrm{A\,m^2}, \qquad i \in \{x, y, z\}. \tag{4.54}$$

This interface logic is a crucial part of the implemented PD baseline. In practice, the achieved closed-loop behavior depends not only on the feedback matrices in (4.42)–(4.43), but also on wheel projection, torque saturation, magnetic inversion, and the priority assigned to momentum dumping over weak-axis support.

4.3.4 Final Implemented PD Configuration

The final PD baseline used in the comparative benchmark can therefore be summarized as follows. The controller computes a quaternion-based body-torque request at

10 Hz using inertia-scaled proportional and derivative gains. This requested torque is then projected onto the two-wheel faulted configuration RW1–RW3 through least-squares allocation. Because the direct wheel authority is insufficient to span the full three-axis torque space, a dedicated magnetic support term is added only on the body y axis. The same magnetic subsystem is also responsible for reaction-wheel desaturation, with dumping assigned higher priority than attitude assistance.

From a methodological viewpoint, this implemented architecture remains faithful to the role assigned to the PD controller in the thesis. It is a lightweight classical baseline, designed to remain interpretable and computationally inexpensive, yet embedded in the same realistic underactuated actuator-management framework used by the other strategies. The purpose is therefore not to present the PD law in an ideal unconstrained setting, but to document how a classical controller behaves when integrated into a shared benchmark architecture that includes actuator limits, availability masks, magnetic projection constraints, and momentum-management requirements.

Table 4.2: Implemented PD controller parameters used in the benchmark.

Parameter	Symbol	Value	Unit
Timing			
Controller sample time	T_s	0.1	s
Internal integration step	h_{int}	0.01	s
Main PD gains			
Damping ratio	ζ	1	–
Desired settling-time scale	T_s^{des}	150	s
Natural frequency	ω_n	0.0267	rad/s
Scalar damping coefficient	d	0.0533	s^{-1}
Scalar stiffness coefficient	k	1.422×10^{-3}	s^{-2}
Proportional gain matrix	$\mathbf{K}_p$	diag(0.01380, 0.01024, 0.02389)	–
Derivative gain matrix	$\mathbf{K}_d$	diag(0.5173, 0.3840, 0.8960)	–
RW interface			
Active wheels	–	RW1, RW3	–
Failed wheels	–	RW2, RW4	–
Allocation method	–	Minimum-norm least squares	–
Per-wheel torque limit	$\tau_{rw,\max}$	0.2	N·m
MTQ assist			
Weak-axis support channel	–	Body y axis only	–
Proportional assist gain	$K_{p,y}^{att}$	3.5×10^{-2}	–
Derivative assist gain	$K_{d,y}^{att}$	1.5	–
Dumping and limits			
Dumping proportional gain	$K_{p,\mathrm{dump}}$	5×10^{-4}	–
Assist scaling during dumping	γ_d	0.2	–
Dipole bound per axis	$\lvert m_i \rvert$	≤ 200	$A{\cdot}m^2$

The implemented PD baseline is now fully defined. The next section introduces the benchmark implementation of the NMPC controller under the same spacecraft, scenario, and actuator assumptions.

4.4 NMPC Controller Implementation

The NMPC framework introduced in Chapter 3 is now specialized to the benchmark case study defined in Sections 4.1 and 4.2. In the present section, the focus shifts from the theoretical principles of receding-horizon control to the actual controller configuration used in the comparative simulations, including the internal prediction model, the adopted horizons, the optimization settings, the selected

weights and bounds, and the integration of the controller within the common actuation-management architecture.

Unlike the classical PD baseline, the implemented NMPC does not first compute an unconstrained desired body torque and then leave feasibility handling entirely to downstream logic. Instead, the main reduced wheel command is synthesized directly through a constrained optimization problem in which the admissible torque range and torque variation are embedded in the controller itself. At the same time, for fairness of comparison, the NMPC is still embedded in the same external actuator-level benchmark framework used by the other strategies: the same faulted wheel configuration is retained, and the same MTQ-based momentum-dumping manager remains active outside the optimization loop.

4.4.1 Internal Model Used in the Controller

The implemented NMPC is formulated on a control-oriented nonlinear model of the underactuated spacecraft rotational dynamics. The controller state is defined as

$$\mathbf{x}_k = \begin{bmatrix} q_{0,k} & q_{1,k} & q_{2,k} & q_{3,k} & \omega_{x,k} & \omega_{y,k} & \omega_{z,k} \end{bmatrix}^T, \tag{4.55}$$

where q_k is the quaternion error in scalar-first convention and $\boldsymbol{\omega}_k$ is the body angular velocity. Operating on the error state rather than on the absolute attitude allows the controller to treat each constant-reference segment of the benchmark guidance as a regulation problem about the equilibrium

$$q_e^\star = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T, \qquad \boldsymbol{\omega}^\star = \mathbf{0}. \tag{4.56}$$

The output vector used in the cost function is chosen as

$$\mathbf{y}_k = \begin{bmatrix} q_{1,k} & q_{2,k} & q_{3,k} & \omega_{x,k} & \omega_{y,k} & \omega_{z,k} \end{bmatrix}^T, \tag{4.57}$$

that is, the vector part of the quaternion error together with the full body angular-rate vector. The scalar quaternion component is not penalized directly in the output because it is redundant once quaternion normalization and hemisphere enforcement are imposed.

Consistently with the degraded benchmark architecture, the manipulated-variable vector contains only the directly commanded body-torque components on the controllable axes:

$$\mathbf{u}_k = \begin{bmatrix} \tau_{x,k} & \tau_{z,k} \end{bmatrix}^T, \qquad \boldsymbol{\tau}_k = \begin{bmatrix} \tau_{x,k} & 0 & \tau_{z,k} \end{bmatrix}^T. \tag{4.58}$$

Therefore, the NMPC acts directly only on the body x and z torque channels, while the body y direction remains weakly controlled and is handled indirectly

through the nonlinear rigid-body coupling and the shared external MTQ support architecture.

The internal state propagation is implemented through the nonlinear functions

$$\mathbf{x}_{k+1} = f_d(\mathbf{x}_k, \mathbf{u}_k), \qquad \mathbf{y}_k = h(\mathbf{x}_k), \tag{4.59}$$

where, in the MATLAB/Simulink implementation, $f_d(\cdot)$ corresponds to `nmpc_stateFcn_under` and $h(\cdot)$ corresponds to `nmpc_outputFcn_under` [18]. The state function includes:

1. forward-Euler discretization of quaternion kinematics and Euler rigid-body dynamics;

2. quaternion normalization after each prediction step;

3. hemisphere enforcement to maintain the convention $q_0 \geq 0$.

The rigid-body dynamics embedded in the prediction model is

$$\dot{\boldsymbol{\omega}}(t) = \mathbf{J}^{-1}\left(\boldsymbol{\tau}(t) - \boldsymbol{\omega}(t) \times \left(\mathbf{J}\boldsymbol{\omega}(t)\right)\right), \tag{4.60}$$

with benchmark inertia matrix

$$\mathbf{J} = \mathrm{diag}(9.7,\ 7.2,\ 16.8)\ \mathrm{kg\,m^2}. \tag{4.61}$$

The quaternion propagation remains consistent with the scalar-first convention adopted throughout the thesis.

Disturbance torques are not included explicitly in the internal optimization model. Accordingly, the NMPC predicts the nominal underactuated rigid-body dynamics and relies on the receding-horizon state update to compensate for gravity-gradient effects, residual magnetic disturbance, and mismatch introduced by the external MTQ supervisory logic. This is a deliberate modeling choice: it keeps the optimization problem compact while preserving the dominant nonlinear coupling needed to exploit the underactuated spacecraft dynamics.

Before being embedded in closed loop, the state and output functions are validated through MATLAB's `validateFcns` procedure using the nominal initialization

$$\mathbf{x}_0 = \begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}^T, \qquad \mathbf{u}_0 = \begin{bmatrix}0 & 0\end{bmatrix}^T. \tag{4.62}$$

This step is useful to verify dimension consistency and numerical compatibility of the prediction model before repeated closed-loop use in Simulink.

4.4.2 Horizon, Sampling, and Optimization Settings

The implemented controller operates at the same benchmark control-update period already introduced for the overall simulation framework:

$$T_s = 0.1 \text{ s}. \tag{4.63}$$

Within this discrete-time structure, the adopted prediction and control horizons are

$$N_p = 30, \qquad N_c = 7. \tag{4.64}$$

These values were retained as a compromise between sufficiently rich finite-horizon prediction and acceptable computational cost in repeated simulations. In particular, the chosen horizon is long enough to allow the optimizer to exploit nonlinear cross-axis coupling over multiple steps, while the shorter control horizon reduces the number of free decision variables.

The benchmark NMPC is implemented directly through MATLAB's `nlmpc` interface, without introducing a separate custom nonlinear-programming backend in the reported comparison setup [18]. The optimization problem is therefore solved using the solver infrastructure exposed by the `nlmpc` object.

In the retained configuration, early suboptimal termination is disabled:

$$\texttt{UseSuboptimalSolution} = \texttt{false}. \tag{4.65}$$

The main solver settings are

$$\begin{aligned} \texttt{MaxIterations} &= 50, \\ \texttt{MaxFunctionEvaluations} &= 6000, \\ \texttt{OptimalityTolerance} &= 5 \times 10^{-4}, \\ \texttt{ConstraintTolerance} &= 5 \times 10^{-4}, \\ \texttt{StepTolerance} &= 10^{-6}, \\ \texttt{FiniteDifferenceType} &= \texttt{forward}, \\ \texttt{UseParallel} &= \texttt{true}. \end{aligned} \tag{4.66}$$

These settings reflect the practical trade-off typical of nonlinear predictive control in simulation-oriented studies. On one hand, excessively loose tolerances may yield faster but noisier and less regular torque profiles. On the other hand, overly strict optimization settings may increase runtime without a proportional improvement in closed-loop behavior. The retained configuration was selected because it provided sufficiently smooth reduced wheel-torque commands while preserving acceptable convergence behavior over repeated closed-loop runs [3, 18].

4.4.3 Weights, Constraints, and Tuning Choices

The finite-horizon cost used in the implemented controller follows the standard quadratic structure discussed in Chapter 3:

$$J_k = \sum_{i=0}^{N_p-1} \left(\mathbf{y}_{k+i|k} - \mathbf{y}_{ref}\right)^T \mathbf{Q} \left(\mathbf{y}_{k+i|k} - \mathbf{y}_{ref}\right) + \sum_{i=0}^{N_c-1} \mathbf{u}_{k+i|k}^T \mathbf{R}\, \mathbf{u}_{k+i|k} + \sum_{i=0}^{N_c-1} \Delta\mathbf{u}_{k+i|k}^T \mathbf{S}\, \Delta\mathbf{u}_{k+i|k}, \tag{4.67}$$

with zero output reference

$$\mathbf{y}_{ref} = \mathbf{0} \tag{4.68}$$

for each constant-reference segment of the piecewise guidance profile.

In the implemented benchmark, the weighting matrices are

$$\mathbf{Q} = \mathrm{diag}(1000,\ 1000,\ 1000,\ 5000,\ 5000,\ 5000), \tag{4.69}$$

$$\mathbf{R} = \mathrm{diag}(0.1,\ 0.1), \tag{4.70}$$

$$\mathbf{S} = \mathrm{diag}(1500,\ 1500). \tag{4.71}$$

This choice penalizes angular-rate deviation more strongly than quaternion-vector error, while still regularizing both the absolute torque request and its step-to-step variation. In practical terms, the selected weights favor a well-damped rotational response and help suppress unnecessarily abrupt reduced wheel-torque sequences.

The manipulated variables are bounded according to the conservative benchmark limits

$$|\tau_x| \le 0.104\ \mathrm{N\,m}, \qquad |\tau_z| \le 0.060\ \mathrm{N\,m}, \tag{4.72}$$

which correspond to a fraction

$$\mathrm{satFrac} = 0.30 \tag{4.73}$$

of the theoretical body-torque authority associated with the active wheel pair in the degraded RW1–RW3 configuration. These conservative limits are intentional: they reduce the risk of the optimizer generating torque sequences that would later be strongly clipped by the downstream actuation chain.

To avoid excessive command chattering, bounds are also imposed on manipulated-variable increments:

$$|\Delta\tau_x| \le 3.64 \times 10^{-3}\ \mathrm{N\,m}, \qquad |\Delta\tau_z| \le 2.10 \times 10^{-3}\ \mathrm{N\,m}, \tag{4.74}$$

which correspond to the rate fraction

$$\mathrm{rateFrac} = 0.035. \tag{4.75}$$

These rate bounds are particularly relevant in the present reduced-actuation scenario, because they prevent the optimizer from trying to exploit dynamic coupling through unrealistically abrupt torque switching over a short horizon.

During tuning, optional soft output constraints on the angular-rate components were also considered:

$$\bar{\omega}_x = 0.010 \text{ rad/s}, \qquad \bar{\omega}_y = 0.005 \text{ rad/s}, \qquad \bar{\omega}_z = 0.010 \text{ rad/s}, \tag{4.76}$$

with slack-variable penalty

$$\rho_{ECR} = 10^6. \tag{4.77}$$

However, the comparative benchmark reported in the thesis is based primarily on the manipulated-variable and manipulated-variable-rate bounds above, while the soft rate constraints are retained as an implementation option explored during controller refinement.

The tuning procedure followed a pragmatic simulation-based approach. Horizons, weights, and bounds were iteratively adjusted through repeated nonlinear closed-loop runs with the objective of obtaining:

1. well-damped reorientation behavior;
2. rise and settling times on the order of a few hundred seconds;
3. limited oscillatory angular-rate activity;
4. smooth reduced wheel-torque commands compatible with the downstream actuation chain.

This empirical refinement is consistent with practical MPC design, where an initial structured formulation is often followed by repeated tuning in simulation to balance tracking quality, control smoothness, and constraint interaction [19, 3].

Although the NMPC computes the primary reduced wheel-torque request directly inside the optimization loop, the benchmark still retains the same external momentum-dumping and MTQ-management logic used by the other controllers. In particular, reaction-wheel desaturation is activated through wheel-speed hysteresis with thresholds

$$\omega_{high} = 6500 \text{ rpm}, \qquad \omega_{tgt} = 2000 \text{ rpm}, \qquad \Delta\omega_{db} = 800 \text{ rpm}, \tag{4.78}$$

and proportional unloading gain

$$K_p^{dump} = 5 \times 10^{-4}. \tag{4.79}$$

The resulting dumping dipole request is given priority within the shared MTQ manager, while any residual authority available for attitude-assist action is scaled by

$$\alpha_d = \begin{cases} 1, & \text{if dumping is inactive,} \\ 0.2, & \text{if dumping is active.} \end{cases} \tag{4.80}$$

This architecture preserves comparability with the PD and neural baselines: the main difference among the strategies lies in how the primary reduced wheel command is generated, not in how the common magnetic unloading supervisor is defined.

4.4.4 Final Implemented NMPC Configuration

The final benchmark NMPC can therefore be summarized as follows. The controller operates on the quaternion-error state and full body angular-rate vector, but optimizes only the reduced body-torque channels corresponding to the controllable wheel directions in the degraded RW1–RW3 configuration. The prediction model retains the nonlinear rigid-body coupling and quaternion propagation, while omitting explicit disturbance-torque prediction and explicit magnetic-assist modeling inside the optimization. The control action is synthesized at each step through a constrained nonlinear program with benchmark-specific horizons, weights, torque bounds, and torque-rate bounds.

From a methodological viewpoint, this configuration embodies the role assigned to the NMPC in the thesis. It is the expert controller because it provides a model-aware and constraint-aware synthesis of the primary actuation request under the same underactuated benchmark used by all strategies. At the same time, it remains embedded in the common multi-rate actuator architecture with shared momentum-dumping supervision, so that the comparison remains centered on the control law rather than on auxiliary implementation asymmetries.

In this sense, the implemented NMPC does not remove the underactuated nature of the spacecraft. Rather, it addresses it more systematically than the classical baseline by optimizing a feasible reduced wheel-torque sequence over a finite horizon, while exploiting the nonlinear rigid-body coupling captured by the prediction model.

Table 4.3: Implemented NMPC controller parameters used in the benchmark.

Parameter	Symbol	Value	Unit
State model			
State vector	$\mathbf{x}$	$\begin{bmatrix} q_0 & q_1 & q_2 & q_3 & \omega_x & \omega_y & \omega_z \end{bmatrix}^T$	–
Output vector	$\mathbf{y}$	$\begin{bmatrix} q_1 & q_2 & q_3 & \omega_x & \omega_y & \omega_z \end{bmatrix}^T$	–
Manipulated variables	$\mathbf{u}$	$\begin{bmatrix} \tau_x & \tau_z \end{bmatrix}^T$	N·m
State function	–	`nmpc_stateFcn_under`	–
Output function	–	`nmpc_outputFcn_under`	–
Timing and horizons			
Sample time	T_s	0.1	s
Prediction horizon	N_p	30	–
Control horizon	N_c	7	–
Weights			
Output weights	$\mathbf{Q}$	diag(1000, 1000, 1000, 5000, 5000, 5000)	–
Input weights	$\mathbf{R}$	diag(0.1, 0.1)	–
Input-rate weights	$\mathbf{S}$	diag(1500, 1500)	–
Manipulated-variable bounds			
Torque bound on x channel	$\lvert\tau_x\rvert$	≤ 0.104	N·m
Torque bound on z channel	$\lvert\tau_z\rvert$	≤ 0.060	N·m
Saturation fraction	`satFrac`	0.30	–
Manipulated-variable-rate bounds			
Increment bound on x channel	$\lvert\Delta\tau_x\rvert$	$\leq 3.64 \times 10^{-3}$	N·m
Increment bound on z channel	$\lvert\Delta\tau_z\rvert$	$\leq 2.10 \times 10^{-3}$	N·m
Rate fraction	`rateFrac`	0.035	–
Solver settings			
Use suboptimal solution	–	false	–
Max iterations	–	50	–
Max function evaluations	–	6000	–
Optimality tolerance	–	5×10^{-4}	–
Constraint tolerance	–	5×10^{-4}	–
Step tolerance	–	10^{-6}	–
Finite-difference type	–	forward	–
Parallel finite differences	–	true	–
Optional soft constraints			
Candidate rate bounds	$\bar{\omega}_{x,y,z}$	(0.010, 0.005, 0.010)	rad/s
Slack penalty	ρ_{ECR}	10^6	–
Shared dumping logic			
High-speed threshold	ω_{high}	6500	rpm
Target speed	ω_{tgt}	2000	rpm
Deadband	$\Delta\omega_{db}$	800	rpm
Dumping proportional gain	K_p^{dump}	5×10^{-4}	–
Assist scaling during dumping	α_d	0.2	–

With Sections 4.3 and 4.4, the two model-based controllers are fully defined within the same benchmark framework. The next section closes the chapter by summarizing the common comparison conditions adopted for the following neural-network and results chapters.

4.5 Comparative Setup for the Following Chapters

With the benchmark spacecraft, the operational scenario, and the two model-based controllers now fully defined, the comparative framework of the thesis can be stated explicitly. The purpose of the present section is to clarify which elements are kept fixed across the remainder of the work, how fairness of comparison is ensured, and how the same benchmark structure is reused for the later data-driven controller.

The importance of this clarification is methodological. The thesis does not compare controllers developed under different assumptions or evaluated on unrelated simulation setups. Instead, the comparison is intentionally organized so that the spacecraft model, the fault condition, the guidance sequence, the actuator limits, and the environmental inputs remain common. This is essential if the subsequent differences in tracking performance, momentum-management behavior, and computational cost are to be interpreted as genuine controller effects rather than artifacts of inconsistent problem definition.

4.5.1 Common Evaluation Conditions

All control strategies considered in the thesis are evaluated under the same benchmark conditions introduced in the previous sections of this chapter. In particular, the following elements are shared across the comparative analysis:

1. the same rigid-body spacecraft model with inertia matrix $\mathbf{J}$ and constant mass assumptions;

2. the same near-circular low-Earth orbit, used only to generate the environmental inputs required by the attitude benchmark, including orbital geometry, gravity-gradient torque, and geomagnetic-field evolution;

3. the same degraded actuation architecture with only RW1 and RW3 available;

4. the same three-axis MTQ subsystem with identical dipole bounds and internal dynamics;

5. the same piecewise-constant quaternion guidance profile with identical switching times;

6. the same controller update period and plant internal integration step;

7. the same disturbance environment, including gravity-gradient and residual magnetic torque;

8. the same wheel-speed-based momentum-dumping logic and shared magnetic-priority supervision.

As a result, each controller is required to solve exactly the same underactuated attitude-control problem. The benchmark therefore preserves a common plant, a common environment, and a common actuation-management structure across all later chapters.

This common basis also implies that the reported differences among controllers should be interpreted with care. For example, if one strategy produces lower attitude error than another, this cannot be attributed to a more favorable orbit, a milder fault case, or a relaxed disturbance setup, since all these aspects are fixed. Likewise, if one controller exhibits a less favorable overall behavior, this should be interpreted as a consequence of its control-synthesis mechanism rather than as an artifact of plant propagation or scenario complexity.

4.5.2 Consistency of the Comparison

Ensuring consistency of comparison requires more than simply reusing the same spacecraft model. It also requires that the interfaces between controller and plant remain conceptually aligned across the considered strategies.

In the present thesis, this alignment is obtained through the following principles.

First, all controllers operate on the same attitude and angular-rate information. The measured spacecraft quaternion and body angular velocity are always the fundamental state variables available for control. Although the internal state representation may differ among controllers, the underlying physical information is the same.

Second, all controllers are embedded in the same degraded actuator architecture. The structural fault affecting the wheel set is not redefined from one strategy to another, and the same MTQ authority remains shared with the desaturation logic throughout the comparison. This is particularly important because it prevents the comparison from becoming biased by controller-specific actuation privileges.

Third, the same guidance profile is used for all strategies. Therefore, each controller is exposed to the same sequence of initial convergence, commanded reorientation, and return maneuver. This ensures that differences in transient behavior, tracking quality, or control smoothness arise from the controller dynamics rather than from different command sequences.

Fourth, the same assessment variables are retained across the results analysis. Quaternion-based attitude error, angular-rate deviation, actuator activity, momentum-related quantities, and computational timing are all evaluated within the same benchmark logic. The controllers are therefore not judged according to different definitions of performance.

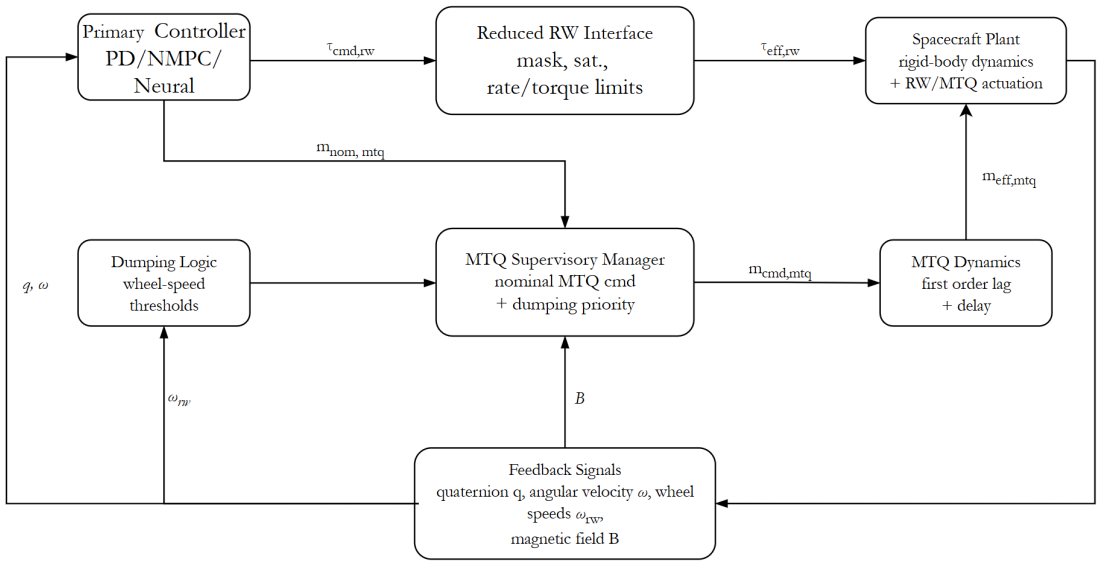


Figure 4.6: Common benchmark command-flow architecture used to ensure consistency of comparison across the benchmark. The primary controller changes across the considered strategies, whereas the actuator-level interface, dumping logic, MTQ supervision, actuator dynamics, and plant feedback remain common.

This consistency is especially important in the present work because the three strategies do not have the same internal nature. The PD controller is a classical reactive law, the NMPC is an optimization-based predictive controller, and the neural controller introduced later is a learned surrogate. Without a tightly controlled benchmark structure, the comparison among such different paradigms would risk becoming ambiguous. By fixing the surrounding benchmark conditions, the thesis keeps the comparison focused on the actual design question of interest: how the different control paradigms trade off simplicity, constraint-awareness, and computational burden under the same underactuated spacecraft scenario.

4.5.3 Link with the Data-Driven Controller Development

The benchmark and implementation framework defined in this chapter also plays a second role beyond the immediate comparison between PD and NMPC. It provides the common foundation for the data-driven controller introduced in the next chapter.

The neural-network-based controller is not developed as an independent control architecture with a different plant or a different mission setup. Instead, it is derived directly from the same benchmark problem. In particular, the data used for training are generated from closed-loop simulations carried out under the same spacecraft

model, the same degraded actuation condition, the same guidance profile, and the same actuation-management logic defined in the present chapter. This means that the later learning problem is grounded in the exact benchmark already established here.

From a methodological viewpoint, this is a crucial point. The neural controller is not meant to compete with the NMPC by solving a different problem. Rather, it is intended to approximate the control behavior of the expert predictive strategy within the same problem definition. The benchmark fixed in Chapter 4 therefore acts as the bridge between the model-based and data-driven parts of the thesis.

In practice, this implies that the feature vector, target commands, and deployment assumptions of the neural controller must remain consistent with the benchmark interfaces already defined. The later data-driven development is thus constrained by the same physical and operational assumptions rather than being treated as a separate and unconstrained machine-learning exercise.

This closes the benchmark-definition and implementation chapter. The thesis now moves from the model-based controller setup to the data-driven strategy, while preserving the same spacecraft, scenario, and comparative framework established here.

Chapter 5

Data-Driven Strategy: Neural Network Control

5.1 Imitation Learning Strategy

This chapter investigates a data-driven control strategy based on a feed-forward NN trained to approximate the expert control law generated by the NMPC strategy introduced in Chapter 4. The objective is to replace the online solution of a constrained optimization problem with the direct evaluation of a learned nonlinear map, while preserving the same state-feedback control philosophy adopted in the model-based benchmark.

In this framework, the NN is not derived from first-principles modeling. Instead, it is trained from input-output pairs collected from closed-loop simulations, so that it can reproduce the control action issued by the expert controller over the operating conditions represented in the dataset. This places the proposed approach within the broader family of imitation-learning methods, in which a policy is learned from demonstrations provided by a reference expert [9, 10, 11].

For the underactuated ADCS considered in this thesis, the learned controller is formulated as a static nonlinear map between the current spacecraft condition and the commanded actuator action:

$$\hat{\mathbf{u}}_k = \hat{\pi}_\theta(\mathbf{x}_k), \tag{5.1}$$

where $\mathbf{x}_k$ is the input feature vector, $\hat{\mathbf{u}}_k$ is the control command predicted by the network, and θ denotes the trainable parameters. The main appeal of this formulation is computational: once trained, the network can be evaluated much more efficiently than solving an NMPC problem online, which makes it a potentially attractive surrogate for resource-constrained on-board implementation.

5.1.1 Behavior Cloning Concept

The specific imitation-learning strategy adopted in this work is *behavior cloning*, namely a supervised-learning approach in which the neural controller is trained to reproduce the action selected by an expert for each observed state [9, 11]. In the present application, the expert is the NMPC-based controller used to generate the reference closed-loop trajectories.

Let the dataset be defined as

$$\mathcal{D} = \left\{ \left(\mathbf{x}^{(i)}, \mathbf{u}^{(i)} \right) \right\}_{i=1}^{N}, \tag{5.2}$$

where $\mathbf{x}^{(i)}$ is the feature vector associated with sample i and $\mathbf{u}^{(i)}$ is the corresponding command generated by the expert controller. The training problem can then be written as the minimization of a supervised regression loss:

$$\theta^{\star} = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \left\| \hat{\pi}_{\theta}\left(\mathbf{x}^{(i)} \right) - \mathbf{u}^{(i)} \right\|_2^2. \tag{5.3}$$

Under this interpretation, the NN does not solve the original OCP explicitly. Rather, it learns an approximation of the expert state-to-command mapping directly from the simulated demonstrations. This is a natural choice for the present study, since the goal is not to redesign the control law from scratch, but to investigate whether a learned surrogate can reproduce the command behavior of the reference NMPC over the relevant operating region.

A well-known limitation of behavior cloning is that its performance depends strongly on the quality and representativeness of the expert dataset [10, 11]. If important regions of the state space are underrepresented during training, the learned controller may exhibit limited generalization when exposed to unseen closed-loop conditions. For this reason, the dataset-generation process plays a central role in the methodology and is discussed in detail in the next section.

5.1.2 Universal Approximation Theorem

The use of a feed-forward NN as a policy approximator is supported, at a theoretical level, by the universal approximation property of multilayer networks. In its classical form, this result states that a sufficiently large feed-forward network with nonlinear activation functions can approximate any continuous function on a compact domain with arbitrary accuracy [12, 13].

This result is relevant here because the mapping between the spacecraft state and the corresponding control command is nonlinear and strongly coupled. The commanded action depends on the interaction between rigid-body attitude dynamics, underactuation, reaction-wheel internal state, and the magnetic control channel.

A feed-forward NN provides a flexible function approximator that can, in principle, capture such relationships without requiring an explicit analytical reformulation of the expert law.

At the same time, the universal approximation theorem does not guarantee practical success by itself. It does not specify how much data are required, how the model should be trained, or how robust the approximation will remain once the learned controller is reinserted in closed loop. For this reason, in the present thesis the theorem is used as a theoretical motivation for the selected approximation framework, whereas the practical viability of the approach depends on dataset quality, preprocessing consistency, network design, and subsequent validation.

5.2 Dataset Generation and Pre-processing

The effectiveness of the proposed imitation-learning strategy depends directly on the quality of the dataset used for training. In the present work, the dataset was generated through repeated closed-loop simulations of the same underactuated benchmark adopted for the model-based control analysis. The resulting data were then assembled into a supervised-learning structure and pre-processed before network training.

5.2.1 Simulation Campaign and Scenarios

A dedicated simulation campaign was constructed to expose the expert controller to multiple operating conditions representative of the considered attitude-control problem. The same spacecraft model, actuator architecture, and closed-loop simulation framework used for the NMPC analysis were retained, so that the demonstrations collected for training remained fully consistent with the benchmark adopted throughout the thesis.

In the baseline campaign used for the feed-forward policy approximation, the dataset was generated from repeated simulations of fixed duration, each initialized from different combinations of attitude error, body angular velocity, and reaction-wheel state. The initial attitude condition was randomized by selecting a rotation axis and angle, while body-rate and wheel-momentum states were varied to enrich the dataset with both maneuvering and regulation transients.

The scenario family was not limited to a single rest-to-rest maneuver. Instead, the simulation set was designed to include different levels of initial excitation, so that the expert policy could be sampled over a broader range of conditions. This is particularly important in the underactuated case, where the control action depends not only on attitude error, but also on the interaction between surviving RWs, MTQs, and internal momentum evolution.

5.2.2 Logged Signals and Feature Vector

During each simulation, the quantities required for supervised learning were logged directly from the closed-loop model. The input vector was defined so as to include the main variables influencing the expert control action, namely attitude error, body angular velocity, reaction-wheel momentum, and geomagnetic field expressed in the body frame. The resulting feature vector is

$$\mathbf{x}_k = [q_{0,k}\ q_{1,k}\ q_{2,k}\ q_{3,k}\ \omega_{x,k}\ \omega_{y,k}\ \omega_{z,k}\ h_{rw,1,k}\ h_{rw,2,k}\ h_{rw,3,k}\ h_{rw,4,k}\ B_{x,k}\ B_{y,k}\ B_{z,k}]^T , \tag{5.4}$$

which defines a 14-dimensional input space.

The first four components correspond to the quaternion error in scalar-first convention, the next three components are the spacecraft body angular velocity, the following four terms describe the RW momentum state, and the last three components contain the magnetic-field vector in body coordinates. This selection was adopted to provide the network with the minimum set of variables required to represent the instantaneous control condition of the spacecraft within the considered ADCS architecture.

The target vector was defined as

$$\mathbf{u}_k = [\tau_{x,k}\ \tau_{y,k}\ \tau_{z,k}\ m_{x,k}\ m_{y,k}\ m_{z,k}]^T , \tag{5.5}$$

where the first three components are the commanded body torques and the last three components are the magnetic dipole commands generated by the expert closed-loop controller. This output definition preserves consistency with the signal interface used in the simulation framework and allows the neural controller to be interpreted as a direct replacement of the expert command-generation block.

5.2.3 Dataset Assembly and Robust Logging

Once each simulation was completed, the logged signals were converted into numeric arrays and reorganized into a sample-by-sample supervised-learning format. Since the raw simulation outputs could be returned either as `timeseries` objects or as numeric arrays, a dedicated post-processing step was introduced to standardize their representation before concatenation.

For each run, the input samples were assembled into a matrix

$$\mathbf{X}^{(i)} \in \mathbb{R}^{N_i \times 14}, \tag{5.6}$$

while the corresponding targets were stored in

$$\mathbf{Y}^{(i)} \in \mathbb{R}^{N_i \times 6}, \tag{5.7}$$

where N_i denotes the number of valid time samples obtained from simulation i. To preserve input–output consistency, all logged channels within the same run were aligned to the minimum common sample length before being stored.

A robust logging procedure was adopted also at file level. Rather than building the full dataset only at the end of the campaign, each simulation was first saved as an individual checkpoint containing its local matrices $\mathbf{X}^{(i)}$ and $\mathbf{Y}^{(i)}$. After all runs had been processed successfully, the global dataset was assembled by concatenating the per-simulation matrices into

$$\mathbf{X} \in \mathbb{R}^{14\times N}, \qquad \mathbf{Y} \in \mathbb{R}^{6\times N}, \tag{5.8}$$

where N is the total number of samples collected over the full simulation campaign.

This two-stage procedure improved the robustness of dataset generation. First, it reduced the risk of losing the entire campaign because of a late interruption. Second, it preserved the simulation-level grouping of the data, which is useful during the subsequent train/validation split because samples belonging to the same trajectory remain strongly time-correlated.

5.2.4 Pre-processing for Training and Output Normalization

Before training, the global dataset was reorganized so that each sample could be associated with its original simulation. This made it possible to perform the train/validation split at trajectory level rather than at sample level. Such a choice is important because consecutive points extracted from the same closed-loop run are not statistically independent, and a fully random sample-wise split would lead to information leakage between training and validation sets.

The adopted preprocessing pipeline therefore retained a fraction of the available simulations for validation, while the remaining runs were used for training. Input conditioning was handled through standardization, so that each feature was presented to the network on a numerically comparable scale. The same principle was applied to the output channels, whose normalization parameters were estimated using only the training subset.

If $\boldsymbol{\mu}_u$ and $\boldsymbol{\sigma}_u$ denote the channel-wise mean and standard deviation of the training targets, the normalized output vector can be written as

$$\tilde{\mathbf{u}} = \frac{\mathbf{u} - \boldsymbol{\mu}_u}{\boldsymbol{\sigma}_u}. \tag{5.9}$$

Very small standard deviations were safeguarded numerically to avoid ill-conditioned scaling on channels with limited variability.

This preprocessing stage was introduced to improve numerical conditioning during optimization while preserving consistency between the expert-generated

commands and the learning problem posed to the network. In particular, the final feed-forward training configuration retained in the thesis relies on output standardization rather than on a bounded label mapping, since the former is more natural for a regression-oriented approximation of continuous control commands and remains fully consistent with the adopted actuator-interface output definition.

5.3 Network Architecture Design

Once the dataset structure had been defined, the next step was the selection of a NN architecture capable of approximating the expert control law while remaining simple enough to be trained and integrated consistently within the simulation framework. In this thesis, the main reference architecture is a feed-forward fully connected NN, coherently with the behavior-cloning formulation introduced in Section 5.1. Under this interpretation, the controller is represented as a static nonlinear map between the current spacecraft condition and the corresponding control action.

The objective of the architectural design was not to introduce a highly specialized deep-learning solution, but rather to verify whether a standard MLP could capture the dominant input–output relation induced by the NMPC expert over the operating conditions represented in the dataset. The design process therefore focused on three main aspects: the selection of the input/output interface, the hidden-layer topology, and the activation functions adopted within the network.

5.3.1 Input and Output Selection

The NN input and output definitions were kept consistent with the dataset construction presented in Section 5.2. In particular, the network input is the 14-dimensional feature vector defined in (5.4), which collects the quaternion error, spacecraft angular velocity, RW momentum state, and geomagnetic field vector expressed in the body frame.

This selection was adopted because it contains the main variables influencing the control action in the considered underactuated ADCS. The quaternion error and angular velocity describe the current attitude-control condition; the RW momentum terms account for the internal actuator state; and the magnetic-field vector is required because the effectiveness of the MTQs depends directly on the local geomagnetic environment. From a control perspective, these quantities provide a compact state description from which the expert command can be inferred.

The network output is defined directly at actuator-command level as the six-dimensional vector

$$u_{\mathrm{NN},k} = \begin{bmatrix} \tau_{\mathrm{cmd},x,k} & \tau_{\mathrm{cmd},y,k} & \tau_{\mathrm{cmd},z,k} & m_{\mathrm{cmd},x,k} & m_{\mathrm{cmd},y,k} & m_{\mathrm{cmd},z,k} \end{bmatrix}^T,$$

where the first three components represent the commanded body-torque vector $\tau_{\mathrm{cmd},k} \in \mathbb{R}^3$ and the last three components represent the commanded magnetic-dipole vector $m_{\mathrm{cmd},k} \in \mathbb{R}^3$. In implementation terms, the NN block therefore returns a single 6×1 command vector, which is subsequently split into torque and magnetic commands and passed to the same downstream actuation chain adopted in the model-based benchmark.

This choice is important because the neural policy is not designed to reproduce the internal manipulated variables of a specific controller, such as the reduced decision vector optimized by the underactuated NMPC. Instead, it is trained to approximate the actuator-interface command generated by the expert closed-loop architecture. In other words, the learning target is the final command signal made available to the allocation and actuation blocks, not the internal optimization variables of the expert controller.

For this reason, the output dimension of the NN is kept equal to six even in the underactuated benchmark. The loss of control authority is not encoded by removing output channels from the neural network, but by the downstream actuator configuration itself. More specifically, the torque command $\tau_{\mathrm{cmd},k}$ is processed by an allocation stage that reflects the actual faulted spacecraft architecture: in the degraded case, only the surviving reaction wheels contribute directly to the reduced RW torque subspace, while the remaining control authority must be recovered through the shared magnetic actuation and supervisory logic. As a consequence, the y-axis torque component may still appear in the command vector predicted by the NN, but it is not interpreted as a direct independent RW command. Rather, it is handled consistently by the same fault-aware downstream actuation architecture used throughout the benchmark.

5.3.2 Topology: Layers and Neurons

The selected feed-forward architecture is a MLP with three hidden fully connected layers, each containing 128 neurons. In compact form, the resulting structure can be summarized as

$$14 \rightarrow 128 \rightarrow 128 \rightarrow 128 \rightarrow 6.$$

This topology was chosen as a compromise between expressive capability and implementation simplicity. A smaller network could be too restrictive to reproduce the nonlinear state-to-command relation generated by the NMPC expert over a broad range of operating conditions, whereas a significantly larger model would increase training cost and complexity without offering a clear methodological advantage at this stage of the work. The adopted configuration was therefore retained as a reasonable baseline for supervised policy approximation.

The use of multiple hidden layers enables the NN to construct a hierarchical nonlinear transformation of the input features. In the present application, this is

useful because the expert control law depends on coupled effects involving rigid-body dynamics, actuator internal state, and magnetic actuation. At the same time, the network depth remains limited, which keeps the model relatively straightforward to train, validate, and deploy within the MATLAB/Simulink environment.

From an implementation viewpoint, the architecture was defined through a standard feature input layer followed by three fully connected hidden layers and a final regression output layer. Since the controller must predict continuous-valued commands rather than discrete classes, the overall problem is naturally formulated as a regression task rather than a classification task.

5.3.3 Activation Functions

The hidden layers use the ReLU activation function, defined element-wise as

$$\phi(z) = \max(0, z). \tag{5.10}$$

This choice was adopted because it provides a simple nonlinear transformation while keeping the training process computationally efficient and numerically well conditioned in many feed-forward regression problems.

Within the present application, ReLU activations are suitable because they enable the network to approximate a nonlinear control map without introducing the strong saturation effects associated with bounded sigmoidal nonlinearities. This is useful when the target outputs represent continuous actuator commands whose variability must be preserved across different operating conditions.

No bounded activation function was imposed at the final output layer. The output layer is therefore linear, consistently with the regression formulation and with the standardized target representation used during training. In practice, the network predicts normalized command values, which are then mapped back to physical units through the inverse transformation associated with the output-normalization parameters.

This architectural choice is coherent with the objective of reproducing the expert control action as accurately as possible in a MSE sense, while leaving physical actuator saturation and command allocation to the surrounding control framework rather than embedding them directly into the NN structure.

No saturation nonlinearity is introduced in the output layer. This choice is consistent with the supervised-learning formulation adopted in the thesis, in which the network predicts standardized continuous-valued commands that are later mapped back to physical units through inverse standardization and then interpreted by the shared downstream actuation architecture.

5.4 Training Process

Once the dataset and the NN architecture had been defined, the next step was the supervised training of the feed-forward controller. In the adopted implementation, training was performed on the dataset generated from the NMPC expert simulations, with the objective of minimizing the discrepancy between the command predicted by the network and the reference action generated by the expert controller.

The overall training pipeline was designed to remain consistent with the simulation-level organization of the dataset. In particular, the split between training and validation data was performed by simulation rather than by individual samples. This choice is important because consecutive points extracted from the same closed-loop trajectory are strongly time-correlated, and a purely random sample-wise split would introduce information leakage between training and validation subsets.

After reconstructing the simulation index from the nominal number of samples per run, a fraction of the available trajectories was reserved for validation, while the remaining simulations were retained for training. Output normalization parameters were computed using only the training subset and were then applied consistently to the validation data. This ensured that validation remained representative of previously unseen trajectories while preserving coherence with the supervised-learning setup.

5.4.1 Loss Function (MSE)

The NN training problem was formulated as a supervised regression task. Let $\mathbf{x}^{(i)}$ denote the input feature vector associated with sample i, and let $\tilde{\mathbf{u}}^{(i)}$ denote the corresponding normalized target vector. The network prediction can then be written as

$$\hat{\tilde{\mathbf{u}}}^{(i)} = \hat{\pi}_\theta\left(\mathbf{x}^{(i)}\right), \tag{5.11}$$

where θ is the set of trainable parameters.

The loss function adopted during training is the MSE,

$$\mathcal{L}_{\mathrm{MSE}}(\theta) = \frac{1}{N_b}\sum_{i=1}^{N_b}\left\|\hat{\tilde{\mathbf{u}}}^{(i)} - \tilde{\mathbf{u}}^{(i)}\right\|_2^2, \tag{5.12}$$

where N_b denotes the number of samples in the considered mini-batch. Since the output channels were standardized before training, the loss was evaluated on normalized quantities rather than on raw physical units. This avoids the risk that channels with larger numerical magnitudes dominate the optimization process.

The use of MSE is consistent with the behavior-cloning interpretation of the problem, since the aim is to reproduce a continuous control action selected by the expert controller for each observed spacecraft state.In this sense, the loss directly

measures the pointwise discrepancy between the NN command vector and the expert command vector at actuator-interface level over the training dataset.

In addition to the main regression loss, a small L_2 regularization term was included in the training configuration to limit excessive growth of the network weights and improve numerical conditioning. This does not alter the conceptual role of the MSE objective, but complements it by mildly penalizing overly large parameter values.

5.4.2 Optimization Algorithm (Adam)

The NN parameters were optimized using the Adam algorithm, which combines adaptive first- and second-moment estimates of the stochastic gradients and is widely used in supervised deep-learning applications [20]. In the present work, Adam was selected because it provides a practical compromise between robustness, convergence speed, and ease of implementation for fully connected regression networks.

At iteration k, given the gradient of the loss function with respect to the trainable parameters, the Adam update can be written in compact form as

$$\mathbf{m}_k = \beta_1 \mathbf{m}_{k-1} + (1-\beta_1)\nabla_\theta \mathcal{L}_k, \tag{5.13}$$

$$\mathbf{v}_k = \beta_2 \mathbf{v}_{k-1} + (1-\beta_2)\left(\nabla_\theta \mathcal{L}_k\right)^2, \tag{5.14}$$

$$\theta_k = \theta_{k-1} - \alpha \frac{\hat{\mathbf{m}}_k}{\sqrt{\hat{\mathbf{v}}_k} + \varepsilon}, \tag{5.15}$$

where α is the learning rate, β_1 and β_2 are decay coefficients, and ε is a small positive scalar introduced for numerical stability [20].

In the implemented training script, the main hyperparameters were set as follows:

- maximum number of epochs equal to 1000;
- mini-batch size equal to 4096;
- initial learning rate equal to 10^{-3};
- piecewise learning-rate schedule with drop factor 0.5 and drop period of 50 epochs;
- L_2 regularization coefficient equal to 10^{-4};
- gradient threshold equal to 1.

Training was executed on GPU hardware, with sample shuffling performed at every epoch. A validation-based stopping logic was also enabled, so that the training process could retain the network corresponding to the best validation loss observed

during optimization. This choice is coherent with the objective of preserving the parameter set showing the most favorable behavior on unseen trajectories rather than simply the lowest error on the training subset.

5.4.3 Training and Validation Performance

The behavior of the training process was monitored through the evolution of both training and validation loss. Since the validation subset was built from simulations excluded from the training set, its trend provided a more meaningful indicator of generalization than a fully random sample-wise split would have done.

After training, the normalized network outputs were mapped back to physical units through the inverse standardization, so that the predicted actuator-interface command vector could be interpreted again in physical units:

$$\hat{\mathbf{u}} = \hat{\tilde{\mathbf{u}}} \odot \boldsymbol{\sigma}_u + \boldsymbol{\mu}_u, \tag{5.16}$$

where $\boldsymbol{\mu}_u$ and $\boldsymbol{\sigma}_u$ are the channel-wise mean and standard deviation estimated on the training outputs, and $\odot$ denotes the element-wise product.

The validation error was then also evaluated in original physical units through the RMSE. For a generic output channel j, the corresponding metric can be expressed as

$$\mathrm{RMSE}_j = \sqrt{\frac{1}{N_{\mathrm{val}}} \sum_{i=1}^{N_{\mathrm{val}}} \left(\hat{u}_j^{(i)} - u_j^{(i)}\right)^2}, \tag{5.17}$$

where N_{val} is the number of validation samples. An overall aggregate indicator can also be defined as

$$\mathrm{RMSE}_{\mathrm{global}} = \sqrt{\frac{1}{6N_{\mathrm{val}}} \sum_{i=1}^{N_{\mathrm{val}}} \sum_{j=1}^{6} \left(\hat{u}_j^{(i)} - u_j^{(i)}\right)^2}. \tag{5.18}$$

However, they are not sufficient by themselves to characterize the closed-loop suitability of the learned controller. In the present thesis, they should therefore be interpreted only as intermediate learning indicators rather than as evidence of a satisfactory control solution.

Overall, the training stage established a coherent supervised-learning pipeline for approximating the expert policy from simulation data. This represents a useful methodological result, but the resulting NN does not yet provide a sufficiently reliable closed-loop behavior to be considered a mature alternative to the model-based controllers.

Chapter 6

Simulation Results and Comparative Analysis

6.1 Comparison Framework and Evaluation Metrics

This chapter discusses the simulation results used to assess the closed-loop behavior of the controllers introduced in the previous chapters. The analysis follows the benchmark logic of the thesis. Rather than presenting PD and NMPC as separate case studies, the discussion is organized around the actuation conditions in which the spacecraft operates. This makes the comparison more physically meaningful, since it separates the nominal reference scenario from the degraded configuration in which fault-tolerant control becomes the central issue.

Two benchmark conditions are considered. The first one is the nominal fully actuated case, in which the complete RW set is available and the spacecraft retains full three-axis control authority. The second one is the underactuated case, in which only the surviving wheels remain available and the control law must coordinate the residual RW authority together with the magnetic-control channel. The latter benchmark is the most relevant one for the present thesis, since it directly reflects the faulted condition motivating the development of advanced control strategies for degraded spacecraft operation.

The performance-assessment logic adopted in this chapter follows an ECSS-inspired perspective, in which control performance is described through a compact set of scalar indicators supporting interpretation, requirement-style verification, and controller comparison [21, 22, 23]. The aim is not to reproduce the full formal structure of the standards, but to retain a coherent set of quantities capable of describing pointing quality, transient behavior, and actuator usage in a concise and

physically interpretable way.

The primary attitude indicator is the equivalent attitude-error angle θ_{err}, computed from the error quaternion and used as the main scalar measure of pointing performance. This quantity is well suited to spacecraft-attitude analysis because it is independent of Euler-angle singularities and provides an immediate interpretation of global orientation error [1, 2]. In addition to the full time history, the analysis uses scalar descriptors such as peak value, RMS value, and percentile-based summaries, especially the 95th and 99th percentiles. In the notation of Chapter 4, this quantity is exactly the scalar quaternion-based metric e_q.

A second family of quantities follows an ECSS-like decomposition of pointing performance. In this framework, a moving-mean component of θ_{err} is interpreted as MPE, while the residual component is interpreted as RPE. This distinction helps separate slow residual drift from faster oscillatory behavior and therefore provides a more structured description of closed-loop pointing quality [21, 22].

A third family of metrics is associated with requirement-style compliance. Both attitude and angular-rate regulation are evaluated with respect to absolute bands, so that quantities such as time in band, maximum violation, and settling-related indicators can be used as compact measures of regulation quality. For angular velocity, the main scalar quantity is the norm of the body-rate vector $\|\boldsymbol{\omega}\|$, whose peak, RMS, and percentile-based summaries provide an axis-invariant representation of the residual rotational dynamics. Because the benchmark guidance adopts $\boldsymbol{\omega}_{\text{ref}} = \mathbf{0}$, the reported norm $\|\boldsymbol{\omega}\|$ coincides with the scalar metric e_ω introduced in Chapter 4. Accordingly, timeInBand(θ) denotes the fraction of time for which $\theta_{err}(t) \leq \theta_{\text{band}}$, while timeInBand($\omega$) denotes the fraction of time for which $\|\boldsymbol{\omega}(t)\| \leq \omega_{\text{band}}$, with $\theta_{\text{band}} = 1^\circ$ and $\omega_{\text{band}} = 0.01^\circ/\text{s}$.

Finally, the actuator-level assessment is based on two complementary groups of quantities. For the RW subsystem, the relevant indicators are the maximum speed across active wheels and the fraction of time spent above warning or maximum thresholds. For the MTQ subsystem, the analysis uses the magnetic-dipole norm, together with its peak, RMS, percentile-based summaries, duty-cycle interpretation, and the fraction of time spent near actuator limits. Taken together, these metrics keep the comparison centered on dynamic performance and actuator demand. Implementation-oriented timing assessment is not quantified in this chapter and is left for future work.

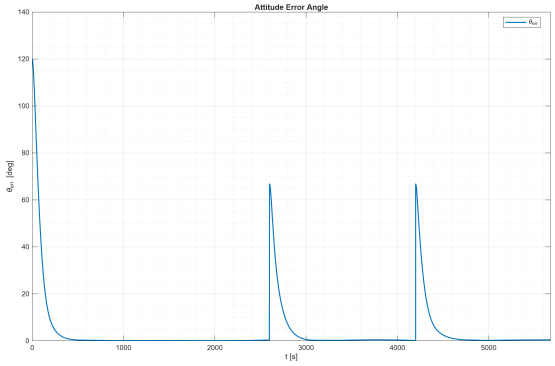
6.2 Nominal Benchmark: Fully Actuated Reference

The fully actuated benchmark provides the nominal reference scenario against which the degraded case can later be interpreted. In this condition, the spacecraft

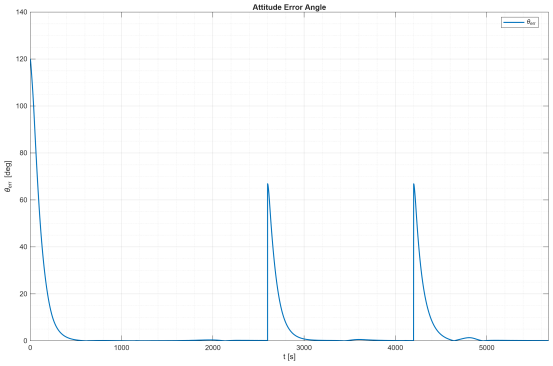
retains full three-axis torque authority through the complete RW set, and the control problem is no longer dominated by the need to recover effective control through residual actuation channels. For this reason, the nominal benchmark is not the most demanding case from the thesis viewpoint, but it remains useful because it clarifies how the two controllers behave when actuator redundancy is not compromised.

6.2.1 Primary Response

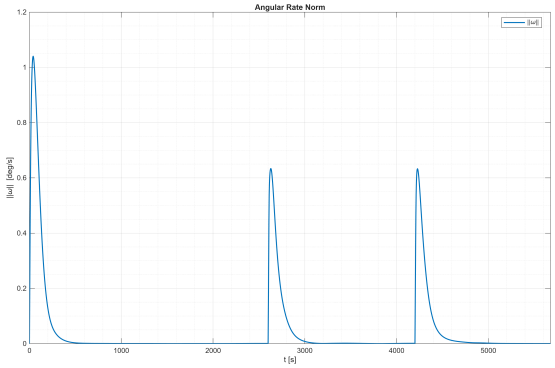
The nominal benchmark uses the same commanded maneuver sequence later adopted for the degraded comparison, namely an initial acquisition phase followed by two commanded reorientations. This preserves consistency across the chapter and allows the nominal and degraded responses to be interpreted within the same mission-like scenario.



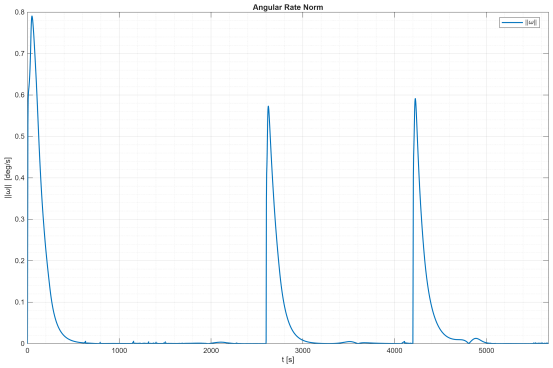
(a) PD - θ_{err}



(b) NMPC - θ_{err}

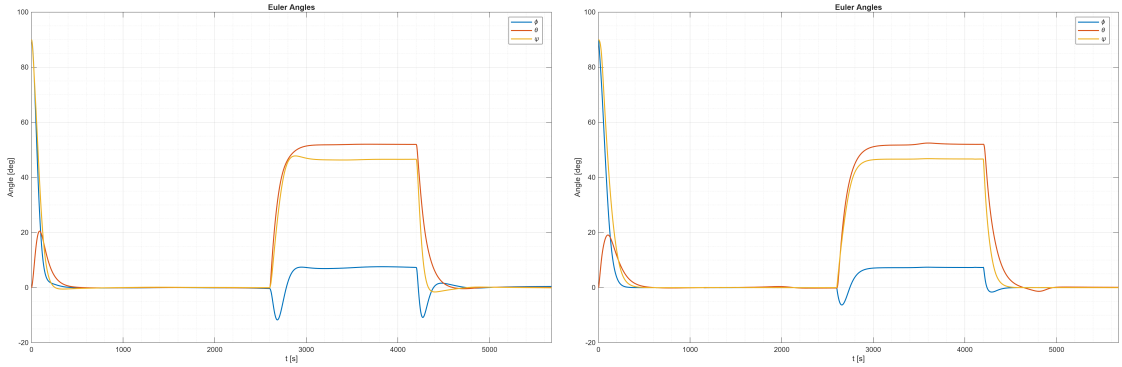


(c) PD - $\|\boldsymbol{\omega}\|$



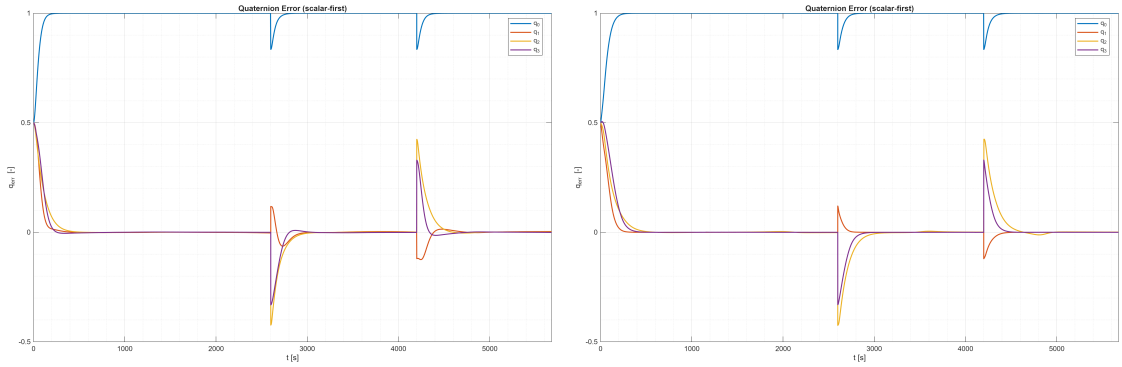
(d) NMPC - $\|\boldsymbol{\omega}\|$

Figure 6.1: Nominal fully actuated comparison in terms of equivalent attitude-error angle and body-rate magnitude.



(a) PD - Euler angles

(b) NMPC - Euler angles



(c) PD - quaternion error

(d) NMPC - quaternion error

Figure 6.2: Supplementary nominal comparison in terms of Euler-angle and quaternion-error histories.

Although Euler angles are not used as primary performance indicators, the corresponding histories provide a useful supplementary visualization of the maneuver sequence. Likewise, the quaternion-error histories remain fully consistent with the scalar attitude-error interpretation adopted in the benchmark analysis.

At aggregate level, the fully actuated benchmark favors the PD baseline in most of the attitude-error indicators. As shown in Table 6.1, PD yields a lower RMS attitude error, 14.72 deg against 16.12 deg, and a lower 95th-percentile attitude error, 28.12 deg against 34.63 deg. The time spent inside the absolute attitude band is also higher for PD, namely 79.75 % against 77.11 %. The same overall tendency is visible in the ECSS-like decomposition, where PD retains a lower RMS MPE and a very similar RPE behavior.

The body-rate comparison is more nuanced. Here, NMPC reduces the peak body-rate magnitude from 1.040 deg/s to 0.791 deg/s, while also lowering the RMS and

percentile-based rate metrics. Therefore, even in the nominal case, the predictive controller provides a smoother rate response. However, this does not translate into a better overall pointing envelope, since the full-run attitude-error statistics remain more favorable to PD. In the present benchmark, the nominal advantage of NMPC is therefore not compelling.

Metric	Unit	PD	NMPC
Peak(θ_{err})	deg	120.00	120.00
RMS(θ_{err})	deg	14.72	16.12
p95(θ_{err})	deg	28.12	34.63
p99(θ_{err})	deg	73.79	85.26
RMS(MPE)	deg	15.05	16.42
RMS(RPE)	deg	2.37	2.35
p95(RPE)	deg	1.85	1.89
Peak($\|\boldsymbol{\omega}\|$)	deg/s	1.040	0.791
RMS($\|\boldsymbol{\omega}\|$)	deg/s	0.161	0.142
p95($\|\boldsymbol{\omega}\|$)	deg/s	0.389	0.382
p99($\|\boldsymbol{\omega}\|$)	deg/s	0.846	0.675
timeInBand(θ)	%	79.75	77.11
timeInBand(ω)	%	78.82	75.84

Table 6.1: Primary fully actuated performance metrics for the nominal benchmark.

For clarity, Table 6.1 uses the compact notation introduced in Chapter 4: $\theta_{err} \equiv e_q$ and $\|\boldsymbol{\omega}\| \equiv e_\omega$, while timeInBand($\theta$) and timeInBand($\omega$) are computed with respect to the benchmark bands $\theta_{\text{band}} = 1^\circ$ and $\omega_{\text{band}} = 0.01^\circ/\text{s}$.

6.2.2 Actuator Usage

The actuator-side comparison further reinforces the same conclusion. Even without reporting the corresponding time histories in figure form, the scalar actuator metrics remain informative for assessing the nominal benchmark.

Table 6.2 shows that the fully actuated PD solution requires a lower maximum active-wheel speed and a substantially lower magnetic effort across all aggregate indicators. In particular, the maximum active-wheel speed increases from 116.62 rpm for PD to 156.45 rpm for NMPC. On the magnetic side, the predictive controller exhibits higher peak, RMS, and percentile-based dipole values, together with a much larger duty cycle, 42.35 % against 17.30 %.

This result is noteworthy because, in the nominal benchmark, the predictive controller does not only fail to improve the overall pointing envelope, but it also does so while demanding more from the actuation system. In other words, the

extra control structure of NMPC does not appear to buy a nominal benefit that compensates for the additional actuator usage reflected by the current benchmark.

Metric	Unit	PD	NMPC
maxActive(ω_{rw})	rpm	116.62	156.45
fracAboveWarn(RWs)	%	0.00	0.00
fracAboveMax(RWs)	%	0.00	0.00
Peak($\|\mathbf{m}\|$)	A m^2	140.07	154.24
RMS($\|\mathbf{m}\|$)	A m^2	10.19	16.42
p95($\|\mathbf{m}\|$)	A m^2	15.67	31.62
p99($\|\mathbf{m}\|$)	A m^2	45.81	80.76
duty($\mathbf{m}$)	%	17.30	42.35
fracAtLimit(MTQs)	%	0.00	0.00

Table 6.2: Actuator-side metrics for the nominal fully actuated benchmark.

6.2.3 Event-Wise Transient Interpretation

The event-wise transient indicators remain consistent with the aggregate-metric interpretation. As shown in Table 6.3, PD reaches shorter settling times after the peak in the second and third commanded reorientations for both θ_{err} and $\|\boldsymbol{\omega}\|$. For example, in the final maneuver the attitude-error settling time after the peak decreases from about 663.6 s for NMPC to 411.3 s for PD, while the corresponding body-rate settling time decreases from about 697.3 s to 399.6 s. Therefore, within the nominal benchmark, the transient behavior does not overturn the aggregate-metric interpretation. It remains broadly consistent with the view that PD is preferable in the fully actuated case.

Event	Signal	Peak time after event start [s]		Settling after peak [s]	
		PD	NMPC	PD	NMPC
Initial acquisition	θ_{err}	0.1	0.1	378.6	434.3
Initial acquisition	$\|\boldsymbol{\omega}\|$	40.3	49.2	354.1	407.2
Reorientation 2	θ_{err}	0.1	0.1	359.6	378.5
Reorientation 2	$\|\boldsymbol{\omega}\|$	29.7	24.4	352.4	359.9
Reorientation 3	θ_{err}	0.2	0.1	411.3	663.6
Reorientation 3	$\|\boldsymbol{\omega}\|$	27.7	25.7	399.6	697.3

Table 6.3: Event-wise transient comparison for the nominal fully actuated benchmark.

6.2.4 Interim Takeaway

Within the nominal benchmark, the current comparison does not support a convincing advantage of NMPC over the PD baseline. Although the predictive controller reduces several body-rate indicators, PD retains a tighter attitude-error envelope, better time-in-band performance, faster event-wise settling in the main reorientation windows, and lower overall actuator demand. At benchmark level, this suggests that the additional structure of NMPC is not justified in the fully actuated case.

6.3 Degraded Benchmark: Underactuated Comparison

The underactuated benchmark is the core result of this thesis, since it is the condition in which the controller must regulate the spacecraft with reduced direct torque authority and therefore exploit the residual actuation channels more effectively. In the representative degraded scenario considered here, the surviving RWs must cooperate with the magnetic-control subsystem in order to recover three-axis stability and track the commanded reorientation sequence. This is the benchmark in which the practical value of a predictive strategy can be assessed most clearly.

6.3.1 Benchmark Description and Maneuver Sequence

The maneuver schedule used for the transient interpretation is summarized in Table 6.4. The scenario is composed of an initial acquisition phase followed by two commanded reorientation windows. The selected run extends sufficiently beyond the third event to enable a meaningful transient analysis on the final maneuver as well, even though, as discussed later, the scalar event extraction is less robust there than in the intermediate maneuver.

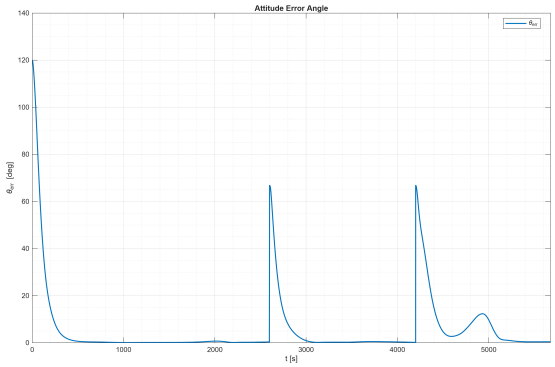
Window	Start time [s]	End time [s]	Interpretation
Initial acquisition	0	2600	Recovery from the initial large attitude error and early closed-loop stabilization.
Reorientation 2	2600	4200	Intermediate commanded attitude maneuver under degraded actuation.
Reorientation 3	4200	5676.9	Final commanded attitude maneuver with sufficient post-event horizon for transient assessment.

Table 6.4: Commanded maneuver windows used for the underactuated event-wise analysis.

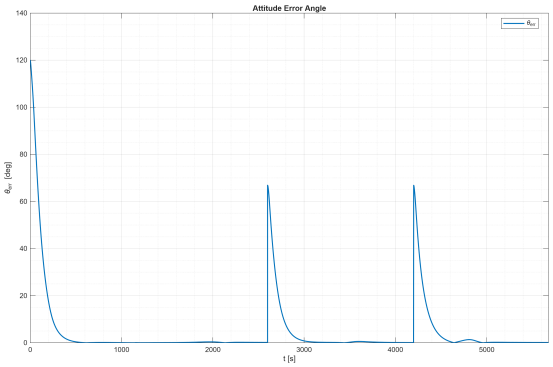
This benchmark should therefore be read as a representative comparison at chapter level, not as a statistical campaign. The objective is to characterize the physical differences between the two control laws in a coherent degraded scenario and to understand how those differences appear in full-run metrics, actuator usage, and event-wise transient behavior.

6.3.2 Pointing Performance and Rate Damping

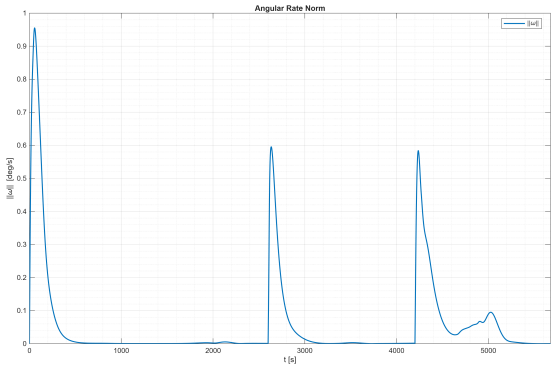
The first comparison concerns the primary closed-loop quantities, namely the equivalent attitude-error angle θ_{err} and the body-rate magnitude $\|\boldsymbol{\omega}\|$. These two signals provide the most compact description of the degraded response, because they directly represent global pointing quality and rotational damping.



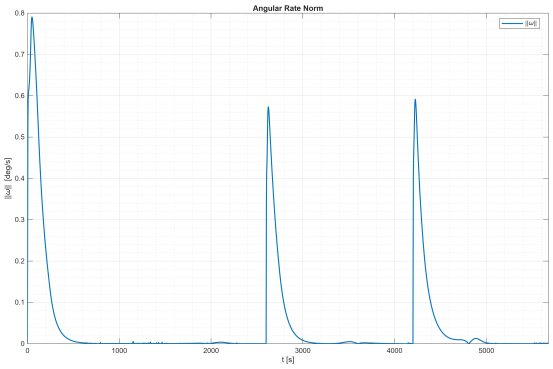
(a) PD - θ_{err}



(b) NMPC - θ_{err}

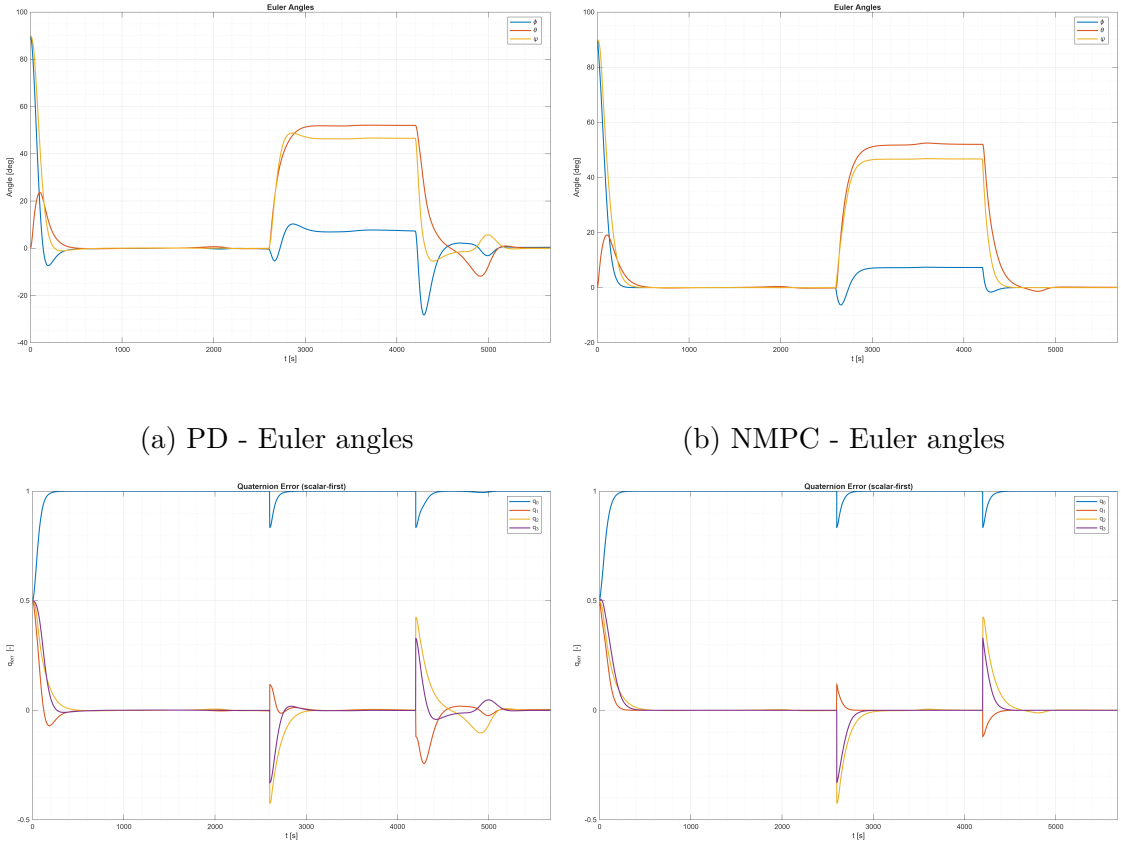


(c) PD - $\|\boldsymbol{\omega}\|$



(d) NMPC - $\|\boldsymbol{\omega}\|$

Figure 6.3: Underactuated comparison in terms of equivalent attitude-error angle and body-rate magnitude.



(a) PD - Euler angles

(b) NMPC - Euler angles

(c) PD - quaternion error

(d) NMPC - quaternion error

Figure 6.4: Supplementary underactuated comparison in terms of Euler-angle and quaternion-error histories.

Although Euler angles are not used as primary performance indicators, the corresponding histories provide a useful supplementary visualization of the maneuver sequence and help identify the temporal location of the main reorientation events. Likewise, the quaternion-error histories remain fully consistent with the scalar attitude-error interpretation adopted in the benchmark analysis.

The full-run aggregate pointing metrics do not show a uniform advantage of NMPC in the underactuated case. On the contrary, Table 6.5 shows that PD retains lower RMS attitude error, 14.76 deg against 16.12 deg, and a lower 95th-percentile attitude error, 24.21 deg against 34.63 deg. The same tendency appears in the ECSS-like decomposition, where PD yields lower RMS MPE, lower RMS RPE, and a lower 95th-percentile RPE. The fraction of time spent inside the absolute attitude band is also slightly higher for PD, namely 79.41 % against 77.12 %.

The body-rate comparison is more mixed. Here, NMPC reduces the peak

body-rate magnitude from 0.955 deg/s to 0.791 deg/s, while also lowering the 99th percentile. However, PD yields a lower 95th-percentile rate value and a slightly higher fraction of time spent inside the angular-rate band. Therefore, the degraded comparison does not support a blanket statement of superiority in terms of aggregate run-level regulation. It instead reveals a more specific trade-off between global pointing quality and peak-transient suppression.

Metric	Unit	PD	NMPC
Peak(θ_{err})	deg	120.00	120.00
RMS(θ_{err})	deg	14.76	16.12
p95(θ_{err})	deg	24.21	34.63
p99(θ_{err})	deg	84.36	85.26
RMS(MPE)	deg	15.14	16.42
RMS(RPE)	deg	1.71	2.35
p95(RPE)	deg	0.96	1.89
Peak($\|\boldsymbol{\omega}\|$)	deg/s	0.955	0.791
RMS($\|\boldsymbol{\omega}\|$)	deg/s	0.143	0.142
p95($\|\boldsymbol{\omega}\|$)	deg/s	0.254	0.382
p99($\|\boldsymbol{\omega}\|$)	deg/s	0.814	0.675
timeInBand(θ)	%	79.41	77.12
timeInBand(ω)	%	79.30	75.85

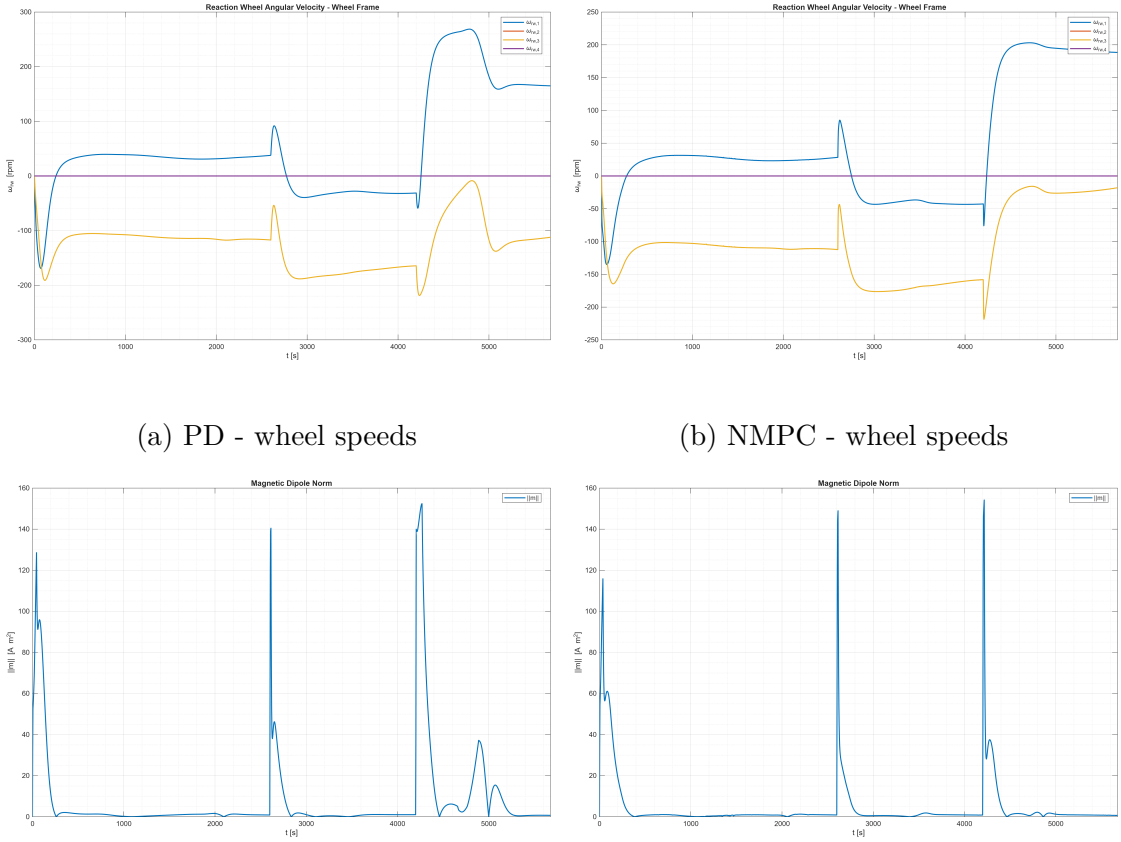
Table 6.5: Primary underactuated performance metrics for the representative degraded benchmark.

For clarity, Table 6.5 uses the same compact notation: $\theta_{err} \equiv e_q$ and $\|\boldsymbol{\omega}\| \equiv e_\omega$, while timeInBand($\theta$) and timeInBand($\omega$) are computed with respect to the benchmark bands $\theta_{\text{band}} = 1^\circ$ and $\omega_{\text{band}} = 0.01^\circ/\text{s}$.

This result is important because it clarifies that the degraded advantage of NMPC is not expressed as a uniform reduction of all aggregate pointing metrics. Instead, the predictive controller becomes more interesting when the analysis shifts from global statistics toward maneuver-phase organization and actuator coordination.

6.3.3 Actuator Usage: Reaction Wheels and Magnetorquers

The actuator-side comparison provides a second layer of interpretation. In an underactuated spacecraft, the relevant question is not only whether the spacecraft converges, but also how aggressively the surviving actuators are used to recover three-axis stability.



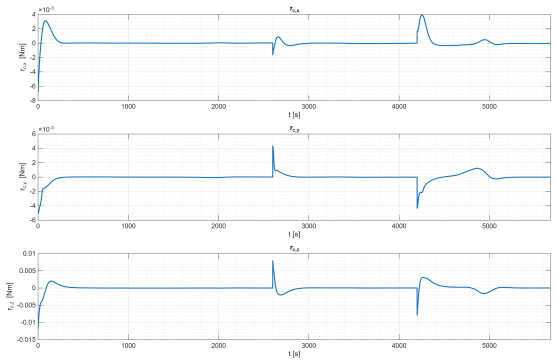
(a) PD - wheel speeds

(b) NMPC - wheel speeds

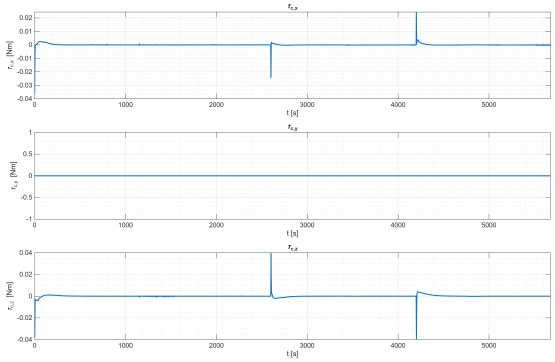
(c) PD - $\|\mathbf{m}\|$

(d) NMPC - $\|\mathbf{m}\|$

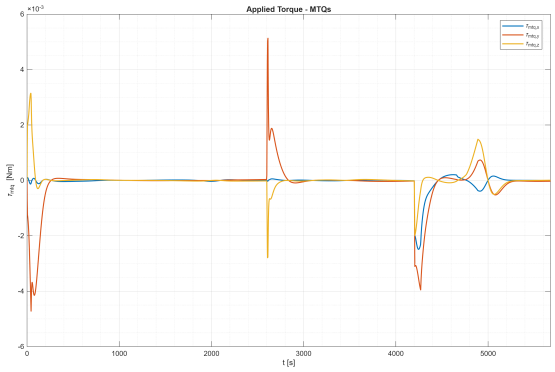
Figure 6.5: Underactuated actuator-level comparison. The top row shows the evolution of the active RWs, while the bottom row reports the magnetic-dipole norm requested from the MTQs.



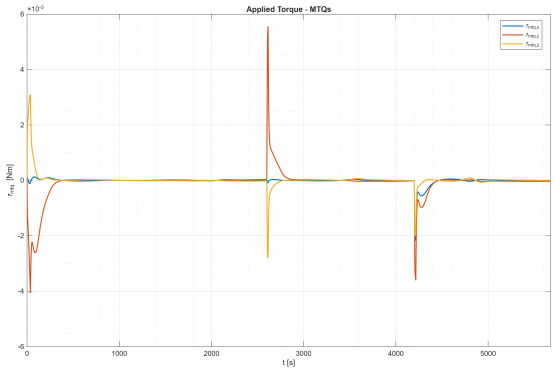
(a) PD - control torque in body frame



(b) NMPC - control torque in body frame



(c) PD - applied MTQs torque



(d) NMPC - applied MTQs torque

Figure 6.6: Supplementary underactuated actuator-level comparison in terms of commanded control torque and applied MTQ torque in the body frame.

A supplementary view is provided by the commanded and applied torque histories, which offer a more direct picture of how control effort is distributed over time, even though the main quantitative assessment remains based on the scalar actuator metrics introduced previously. The underactuated plots make this redistribution explicit: while the commanded body torque remains three-dimensional, the reduced wheel authority makes the magnetic channel more structurally involved during the main maneuver windows.

Metric	Unit	PD	NMPC
maxActive(ω_{rw})	rpm	320.53	218.63
fracAboveWarn(RWs)	%	0.00	0.00
fracAboveMax(RWs)	%	0.00	0.00
Peak($\|\mathbf{m}\|$)	A m^2	168.02	154.20
RMS($\|\mathbf{m}\|$)	A m^2	16.28	16.42
p95($\|\mathbf{m}\|$)	A m^2	26.64	31.64
p99($\|\mathbf{m}\|$)	A m^2	93.07	80.77
duty($\mathbf{m}$)	%	51.71	41.12
fracAtLimit(MTQs)	%	0.00	0.00

Table 6.6: Actuator-side metrics for the representative underactuated benchmark.

The actuator metrics show that the benefit of NMPC is not primarily a reduction of every magnetic-dipole summary statistic, but a more selective improvement of the residual actuation usage. The predictive controller clearly lowers active-wheel loading, reduces the maximum magnetic-dipole request, lowers the heavy upper tail of MTQ usage, and decreases the MTQ duty cycle. At the same time, some aggregate MTQ summaries such as the RMS and 95th percentile remain comparable or slightly less favorable. In the degraded benchmark, this means that the advantage of NMPC appears mainly in wheel-margin preservation and in the reduction of the most demanding magnetic episodes, rather than in a uniform reduction of all actuator metrics.

6.3.4 Event-Wise Transient Interpretation

The full-run metrics presented above are informative, but they do not tell the entire story. In the present benchmark, an important part of the controller difference emerges when the transient response is examined event by event. As shown in Table 6.7, the initial acquisition remains broadly similar for the two controllers, while the intermediate commanded reorientation reveals a much stronger distinction.

In the second maneuver, the PD response reaches the attitude-error peak about 600 s after the event boundary and settles only after about 726 s from the peak, whereas NMPC reaches the peak almost immediately and settles in about 379 s. A similar effect is visible in the body-rate norm, where the PD peak occurs about 630 s after the event start and settles in about 705 s, while NMPC peaks after about 24 s and settles in about 360 s. This is the clearest transient-level advantage of the predictive controller in the degraded case.

The final maneuver is visually informative in the time histories, but the scalar event extraction is less robust for direct comparison in the current workbook. In the PD case, the response enters the event window already close to the settling

band and later exhibits a secondary excursion within the same window. For this reason, the intermediate reorientation provides the cleanest and most defensible event-wise comparison.

Event	Signal	Peak time after event start [s]		Settling after peak [s]	
		PD	NMPC	PD	NMPC
Initial acquisition	θ_{err}	0.1	0.1	442.6	434.3
Initial acquisition	$\|\boldsymbol{\omega}\|$	56.3	49.2	385.4	407.2
Reorientation 2	θ_{err}	600.2	0.1	726.3	378.5
Reorientation 2	$\|\boldsymbol{\omega}\|$	629.8	24.4	704.5	359.9

Table 6.7: Event-wise transient comparison for the underactuated benchmark. The intermediate reorientation provides the most robust scalar comparison within the representative benchmark.

6.3.5 Interim Takeaway

The underactuated benchmark supports a more nuanced conclusion than the nominal case. At aggregate full-run level, PD still retains several stronger pointing indicators. However, NMPC reduces the most severe rate excursions, preserves more wheel-speed margin, lowers the heaviest MTQ demands, and organizes the intermediate degraded reorientation more promptly. Therefore, in the faulted benchmark, the predictive controller does not dominate every scalar metric, but it does provide a more structured use of the residual actuation authority during the most demanding maneuver phases.

6.4 Current Comparative Assessment

6.4.1 Dynamic Performance

Taken together, the two benchmarks support a differentiated interpretation of dynamic performance. In the fully actuated case, PD remains preferable overall. Although NMPC reduces several angular-rate indicators, the nominal benchmark still favors PD in the attitude-error metrics, time-in-band compliance, and event-wise settling behavior. Therefore, in a non-degraded configuration, the additional predictive structure does not translate into a clearly better closed-loop result.

The degraded benchmark leads to a more mixed conclusion. In aggregate run-level pointing metrics, PD remains competitive and, in several cases, preferable. However, once the interpretation shifts to peak-transient suppression and maneuver-phase timing, NMPC becomes more attractive. This contrast between the two

benchmarks is one of the most meaningful outcomes of the chapter. It suggests that the value of advanced predictive control is not general in an unconditional sense, but strongly scenario-dependent.

6.4.2 Actuator Coordination

The actuator-side evidence reinforces the same benchmark-dependent conclusion. In the nominal case, PD retains lower wheel loading and lower magnetic effort, which means that the predictive controller demands more from the actuator set without delivering a correspondingly strong nominal benefit. In the underactuated case, however, the interpretation changes. There, NMPC keeps the residual wheel speeds within a tighter envelope and reduces the most severe MTQ excursions and duty-cycle usage, even though not every MTQ summary metric improves uniformly.

This difference is central to the thesis logic. The main potential advantage of NMPC is not that it should always outperform a classical controller. Rather, it is that it may coordinate the remaining control authority more effectively when the system is faulted. The current benchmark-level evidence is consistent with exactly this interpretation.

6.4.3 Current Scope of the Evidence

The present chapter should be read as a benchmark-level comparison based on representative runs. This is sufficient to support a technically meaningful interpretation of the closed-loop behavior, but it does not yet replace a broader campaign-level assessment. Even with this limitation, the comparison already supports a clear thesis-level message.

The nominal benchmark does not provide a compelling reason to replace the PD baseline with a more complex predictive controller. The underactuated benchmark, by contrast, shows that NMPC becomes more meaningful once the actuation authority is reduced and the control problem becomes structurally harder. Its added value is not expressed as a universal improvement of every scalar metric, but as a more effective handling of the most demanding degraded maneuver phases and a more selective management of residual actuator loading.

6.5 Chapter Summary

This chapter has organized the controller comparison according to the benchmark logic of the thesis. The fully actuated case has been used as the nominal reference, while the underactuated case has been treated as the central degraded benchmark.

In the nominal benchmark, the current comparison favors the PD baseline. The classical controller retains better aggregate attitude-error performance, better

time-in-band behavior, faster settling in the main reorientation windows, and lower overall actuator demand. In the degraded underactuated benchmark, the conclusion becomes more nuanced. There, PD still retains several stronger aggregate pointing metrics, but NMPC reduces the strongest rate excursions, preserves more wheel-speed margin, lowers the heaviest MTQ demands, and settles more promptly in the most informative degraded reorientation.

Taken together, these results support the main interpretative point of the thesis. The additional structure of NMPC is not uniformly advantageous across all operating conditions. Its real value emerges most clearly in the faulted and underactuated regime, where the control law must make more effective use of limited residual actuation authority and where maneuver-phase coordination becomes more important than nominal steady benchmark performance alone.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

This thesis investigated attitude-control strategies for small satellites operating under both nominal and degraded actuation conditions, with particular focus on the underactuated case caused by RW loss. The work was motivated by the need to improve fault tolerance while remaining compatible with the computational constraints of embedded ADCS architectures.

A first contribution of the thesis is the definition of a benchmark-oriented comparison framework. Two operating conditions were considered: a nominal fully actuated configuration and a degraded underactuated configuration. This structure made it possible to separate the reference closed-loop behavior from the faulted regime in which residual actuator coordination becomes the dominant issue. The comparison was supported by a compact set of quaternion-based pointing metrics, angular-rate indicators, requirement-style band-compliance measures, and actuator-side quantities related to both RWs and MTQs.

A second contribution concerns the model-based comparison between PD and NMPC. In the nominal fully actuated benchmark, the results do not support a convincing advantage of the predictive controller. Although NMPC reduces several angular-rate indicators, PD retains better aggregate pointing performance, faster settling in the main reorientation windows, and lower actuator demand. In the underactuated benchmark, the conclusion becomes more nuanced. Aggregate pointing metrics still favor PD in several cases, but NMPC reduces the strongest rate excursions, preserves more wheel-speed margin, lowers the heaviest magnetic-control episodes, and improves the timing of the most informative degraded reorientation. This indicates that the value of predictive control is not universal, but becomes

more meaningful when the spacecraft operates in a structurally harder faulted regime.

A third contribution is methodological and concerns the neural-network control pipeline. A complete imitation-learning workflow was developed, including simulation-based dataset generation, feature construction, output standardization, and supervised training. The resulting feed-forward controller was formulated as a surrogate for the command-generation stage at actuator-interface level, so as to predict the full commanded torque and magnetic-dipole vector from the current spacecraft condition while remaining compatible with the shared benchmark actuation architecture. However, the resulting closed-loop behavior was not satisfactory enough to support the neural controller as a mature alternative to the model-based strategies. The neural-network branch should therefore be regarded as an exploratory outcome of the work rather than as a validated final solution.

Overall, the thesis shows that PD remains a strong and effective baseline, especially in nominal fully actuated conditions, while NMPC becomes more relevant in degraded underactuated operation, where residual actuator coordination and transient organization are more critical. The neural-network approach remains promising, but further methodological improvements are required before it can be considered a reliable control solution for the considered benchmark.

7.2 Future Work

Several directions naturally emerge from the present work.

The first and most immediate extension concerns the machine-learning strategy. The imitation-learning approach adopted in this thesis provided a useful first step, but it did not lead to a sufficiently robust closed-loop controller. Future work should therefore investigate more suitable learning paradigms, such as reinforcement learning, as well as hybrid solutions combining supervised initialization with subsequent policy refinement. In parallel, the feature set, network architecture, training curriculum, and loss formulation should be revised to improve both open-loop prediction quality and closed-loop reliability.

A second important direction concerns fault-injection testing. In the present work, nominal and underactuated regimes were treated as distinct benchmark conditions. A more realistic extension would consist in injecting faults during the mission timeline, so as to assess the transition from nominal to degraded operation within a single simulation. This would allow a more faithful evaluation of recovery capability, transient reconfiguration, and controller robustness.

A third extension concerns the breadth of the scenario set. Future work should consider different maneuver profiles, initial conditions, disturbance realizations, and fault configurations. This should be complemented by broader statistical validation,

including multi-run or Monte Carlo-style campaigns, in order to quantify robustness and variability more systematically.

Another relevant direction concerns implementation-oriented assessment. Since one of the motivations of the thesis is the trade-off between control quality and computational affordability, future work should complete the comparison by quantifying execution times under a consistent software and hardware setup, and by verifying compatibility with realistic embedded ADCS constraints.

Finally, it would be worthwhile to investigate supervisory or reconfiguration logics capable of handling transitions between nominal and degraded modes in a structured manner. Such an approach could combine the simplicity of PD in nominal operation with the stronger degraded-regime coordination offered by NMPC, or with future learned surrogates once they become sufficiently reliable.

In summary, the next steps are twofold: improving the learning-based branch, with reinforcement learning representing a particularly promising direction, and extending the model-based comparison through fault-injection campaigns, broader scenario exploration, and implementation-oriented validation.

Bibliography

[1] James R. Wertz, ed. *Spacecraft Attitude Determination and Control*. Vol. 73. Astrophysics and Space Science Library. Originally published by D. Reidel Publishing Company. Springer Dordrecht, 1978. DOI: 10.1007/978-94-009-9907-7 (cit. on pp. 2, 4, 8–13, 15, 16, 18, 20, 21, 23, 27, 35, 46, 79).

[2] Marcel J. Sidi. *Spacecraft Dynamics and Control: A Practical Engineering Approach*. Cambridge Aerospace Series 7. Cambridge: Cambridge University Press, 1997. ISBN: 0-521-55072-6 (cit. on pp. 2, 4, 8–13, 15, 16, 20, 21, 23, 27, 35, 46, 79).

[3] S. Joe Qin and Thomas A. Badgwell. «A Survey of Industrial Model Predictive Control Technology». In: *Control Engineering Practice* 11.7 (2003), pp. 733–764. DOI: 10.1016/S0967-0661(02)00186-7 (cit. on pp. 2, 5, 29, 31, 32, 36, 58, 60).

[4] James B. Rawlings, David Q. Mayne, and Moritz M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. 2nd ed. Madison, WI: Nob Hill Publishing, LLC, 2017. ISBN: 9780975937730 (cit. on pp. 2, 5, 29, 31, 32, 36).

[5] Christopher D. Petersen, Frederick Leve, and Ilya Kolmanovsky. «Model Predictive Control of an Underactuated Spacecraft with Two Reaction Wheels». In: *Journal of Guidance, Control, and Dynamics* 40.2 (2017), pp. 320–332. DOI: 10.2514/1.G000320 (cit. on pp. 2, 33, 36).

[6] Alberto Guiggiani, Ilya Kolmanovsky, Panagiotis Patrinos, and Alberto Bemporad. «Constrained Model Predictive Control of Spacecraft Attitude with Reaction Wheels Desaturation». In: *2015 European Control Conference (ECC)*. IEEE, 2015, pp. 1382–1387. DOI: 10.1109/ECC.2015.7330731 (cit. on pp. 2, 23, 33, 36).

[7] Jens Gießelmann. «Development of an Active Magnetic Attitude Determination and Control System for Picosatellites on Highly Inclined Circular Low Earth Orbits». Master of Engineering by Research thesis. RMIT University, June 2006. URL: https://www.raumfahrt.fh-aachen.de/compass-1/download/Development%20of%20an%20Active%20Magnetic%20Attitude%2

0Determination%20and%20Control%20System%20for%20Picosatellites%20on%20highly%20inclined%20circular%20Low%20Earth%20Orbits.pdf (cit. on pp. 2, 18, 20).

[8] Daniel M. Torczynski, Rouzbeh Amini, and Paolo Massioni. «Magnetorquer Based Attitude Control for a Nanosatellite Testplatform». In: *AIAA Infotech@Aerospace Conference*. Atlanta, Georgia, USA, 2010, pp. 1–9. DOI: 10.2514/6.2010-3511 (cit. on pp. 2, 18, 20, 52).

[9] Dean A. Pomerleau. «ALVINN: An Autonomous Land Vehicle in a Neural Network». In: *Advances in Neural Information Processing Systems 1*. Ed. by David S. Touretzky. Morgan Kaufmann, 1989, pp. 305–313 (cit. on pp. 2, 5, 37, 67, 68).

[10] Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. «A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning». In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Geoffrey Gordon, David Dunson, and Miroslav Dudík. Vol. 15. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, Apr. 2011, pp. 627–635. URL: https://proceedings.mlr.press/v15/ross11a.html (cit. on pp. 2, 5, 37, 67, 68).

[11] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. «Imitation Learning: A Survey of Learning Methods». In: *ACM Computing Surveys* 50.2 (2017), 21:1–21:35. DOI: 10.1145/3054912 (cit. on pp. 2, 5, 37, 67, 68).

[12] George Cybenko. «Approximation by Superpositions of a Sigmoidal Function». In: *Mathematics of Control, Signals and Systems* 2.4 (1989), pp. 303–314. DOI: 10.1007/BF02551274 (cit. on pp. 2, 5, 37, 68).

[13] Kurt Hornik. «Approximation Capabilities of Multilayer Feedforward Networks». In: *Neural Networks* 4.2 (1991), pp. 251–257. DOI: 10.1016/0893-6080(91)90009-T (cit. on pp. 2, 5, 37, 68).

[14] Giovanni Lavezzi, Mariusz Eivind Grøtte, and Marco Ciarcià. «Attitude Control Strategies for an Imaging CubeSat». In: *2019 IEEE International Conference on Electro Information Technology (EIT)*. IEEE, 2019, pp. 149–155. DOI: 10.1109/EIT.2019.8833806 (cit. on p. 9).

[15] Bong Wie, Haim Weiss, and Ari Arapostathis. «Quaternion Feedback Regulator for Spacecraft Eigenaxis Rotations». In: *Journal of Guidance, Control, and Dynamics* 12.3 (1989), pp. 375–380. DOI: 10.2514/3.20418 (cit. on pp. 11, 12, 27, 35, 51).

[16] Riccardo Cecchini. «Modeling of an Orbital Simulator and Design of MPC-based Attitude Control System for VLEO Maneuvers». Relator: E. Capello. Restricted to staff users until 20 October 2026 (embargo). M.Sc. thesis. Politecnico di Torino, 2025. URL: https://webthesis.biblio.polito.it/id/eprint/37467 (cit. on p. 39).

[17] International Association of Geomagnetism and Aeronomy (IAGA). *IGRF-14*. Zenodo. Version v1, published 2024-11-22. Nov. 2024. DOI: 10.5281/zenodo.14012303. URL: https://zenodo.org/records/14012303 (cit. on pp. 40, 45).

[18] MathWorks. *nlmpc - Nonlinear model predictive controller*. Model Predictive Control Toolbox documentation, last updated 2026-02-21, accessed 2026-03-16. MathWorks. 2026. URL: https://www.mathworks.com/help/mpc/ref/nlmpc.html (cit. on pp. 57, 58).

[19] Willy Wojsznis, John Gudaz, Terry Blevins, and Ashish Mehta. «Practical Approach to Tuning MPC». In: *ISA Transactions* 42.1 (2003), pp. 149–162. DOI: 10.1016/S0019-0578(07)60121-9 (cit. on p. 60).

[20] Diederik P. Kingma and Jimmy Ba. «Adam: A Method for Stochastic Optimization». In: *3rd International Conference on Learning Representations (ICLR)*. 2015. URL: https://arxiv.org/abs/1412.6980 (cit. on p. 76).

[21] *Space Engineering – Control Performance*. Tech. rep. ECSS-E-ST-60-10C. ECSS, Nov. 2008 (cit. on pp. 78, 79).

[22] *Space Engineering – Control Performance Handbook*. Tech. rep. ECSS-E-HB-60-10A. ECSS, Dec. 2010 (cit. on pp. 78, 79).

[23] *Space Engineering – Satellite Attitude and Orbit Control System Requirements*. Tech. rep. ECSS-E-ST-60-30C. ECSS, Aug. 2013 (cit. on p. 78).