Constrained Quaternion Attitude Control of Satellites via Semi-Definite Programming

Shavy Kashyap

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2025

Committee:

Mehran Mesbahi

Behçet Açıkmeşe

Program Authorized to Offer Degree:
William E. Boeing Department of Aeronautics & Astronautics

University of Washington

Abstract

Constrained Quaternion Attitude Control of Satellites via
Semi-Definite Programming

Shavy Kashyap

Chair of the Supervisory Committee:
Mehran Mesbahi
William E. Boeing Department of Aeronautics and Astronautics

Spacecraft attitude control is critical for mission success in communication, navigation, and payload safety, requiring maneuvers that respect complex geometric and hardware constraints. Classical quaternion-based PD/PID controllers provide robust unconstrained attitude regulation but lack systematic enforcement of constraints such as sun-avoidance zones and actuator saturation limits.

This thesis presents a hybrid control framework that leverages semi-definite programming (SDP) to generate constraint-compliant, globally optimal attitude trajectories offline, integrating keep-in/out cones and actuator bounds via linear matrix inequalities (LMIs). A quaternion-feedback PD regulator then robustly tracks these trajectories in real time, enabling efficient onboard implementation.

MATLAB and Simulink simulations demonstrate that the proposed SDP-guided control outperforms classical PD/PID methods by rigorously respecting all constraints and improving maneuver safety and accuracy. The results suggest strong potential for future small-satellite missions requiring high-performance constrained attitude control.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Table Number Page

ACKNOWLEDGMENTS

I am deeply grateful to my advisor, Dr. Mehran Mesbahi, whose unwavering support, insightful guidance, and patience have been invaluable to me throughout this journey. His mentorship has profoundly shaped both my research and personal growth.

I would also like to thank my thesis committee members for their helpful feedback and encouragement, which significantly enriched this work.

My sincere thanks go to the entire RAIN Lab for fostering a warm and welcoming atmosphere filled with innovative minds. Special appreciation goes to Shiva Shakeri, Eomji Kim, Justin Chang, Ayaz Ahmed, and Po Chih Huang for their collaboration and support throughout this project.

I am also thankful to Gyan Prakash, Yolanda Everson, and Kelly Liu, whose friendship and encouragement provided much-needed balance and motivation during this journey.

I feel incredibly fortunate to have had this opportunity and am truly thankful to my family and friends for their constant belief in me. Their love and encouragement gave me the strength to persevere and explore beyond my comfort zone.

DEDICATION

To all the women who came before me, whose strength, courage, and resilience paved the way, and to all those who will come after me, may you continue to break barriers and inspire future generations.

Chapter 1

INTRODUCTION AND MOTIVATION

Spacecraft attitude control, the task of steering a vehicle's orientation in three-dimensional space, is fundamental to nearly every satellite mission. It enables critical functions such as pointing high-gain antennas toward ground stations, aiming optical instruments at celestial targets, and protecting sensitive payloads from direct Sun exposure. The Attitude Determination and Control System (ADCS) computes the spacecraft's current attitude from sensor measurements and commands actuator torques to achieve the desired reorientation [29]. Quaternions are the standard representation for attitude due to their compactness, lack of singularities, and efficient algebraic manipulation.

Actuators impose physical bounds on control torque, and spacecraft must adhere to maximum slew-rate limits to prevent sensor smearing and gyroscope saturation. Further, payload instruments introduce geometric constraints: **keep-out** cones surrounding bright celestial bodies (e.g., Sun, Moon) to protect star trackers, and **keep-in** cones to ensure sensors (e.g., sun sensors) maintain a valid line-of-sight(see Figure 1.1). The constrained attitude guidance (CAG) problem thus involves designing a steering law that respects actuator limits, slew-rate bounds, and geometric pointing constraints.

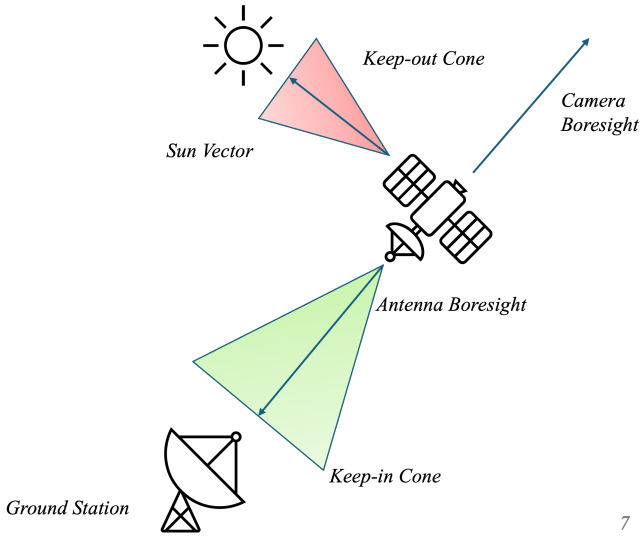


Figure 1.1: Spacecraft attitude constraints visualized. The keep-out cone (red) represents forbidden orientations near bright bodies (e.g., the Sun) to protect instruments. The keep-in cone (green) ensures continuous visibility of essential targets like ground stations. Accurate attitude control maintains spacecraft orientation within these cones.

1.1 Problem Statement

This section states the problem considered in this thesis. Our goal is simple to state but challenging to solve: develop a guidance algorithm that safely steers a spacecraft toward a desired orientation *while satisfying mission-critical constraints.*

Mission Constraints

The controller must ensure each of the following at all times:

- **Keep-Out Zones:** Avoid pointing sensitive payloads (e.g., star trackers, telescopes) at bright bodies (Sun, Moon).
- **Keep-In Zones:** Maintain line-of-sight with essential assets (ground stations, sun sensors).
- **Actuator Limits:** Enforce hard torque and angular-rate bounds to prevent reaction-wheel saturation or thruster over-exertion.

- **Rigid-Body Mechanics:** Respect the spacecraft's nonlinear quaternion kinematics and rotational dynamics.

1.2 Why This Problem is Challenging

Representing spacecraft attitude with **quaternions** introduces a non-intuitive, four-dimensional state space in which standard intuition about rotations no longer applies. Moreover, the keep-in/keep-out requirements manifest as **quadratic** constraints on the attitude states, and the underlying rigid-body kinematics and dynamics are inherently **nonlinear**. Actuator capabilities impose **hard torque and angular-rate bounds** that cannot be relaxed via soft penalties, demanding strict, real-time enforcement. Simultaneously satisfying geometric avoidance, line-of-sight, and physical limit requirements introduces multiple, often competing objectives, carving out a highly **nonconvex** feasible region. These factors together render classical PD/PID approaches insufficient, motivating the development of an optimization-based guidance strategy.

The two primary sources of complexity in constrained spacecraft attitude control-the nonlinearity of the rigid-body dynamics and the nonconvexity of the geometric and actuator constraints-have often been treated separately in the literature. Prior efforts have typically addressed either nonconvex quadratically constrained control problems for linear systems [14], linearly constrained control problems for nonlinear systems [4, 13, 30], or, more tractably, linearly constrained control problems for linear systems [10, 21].

Classical quaternion-feedback controllers, such as proportional-derivative (PD) and proportional-integral-derivative (PID), offer global stability and computational efficiency for unconstrained maneuvers[32, 34]. Nonetheless, they lack systematic methods to enforce hard constraints, typically relying on heuristic gain scheduling or supervisory logic. As mission requirements become more stringent, such ad hoc approaches risk compromising robustness and optimality.

In contrast, systematic treatments of the *constrained* attitude control problem remain relatively scarce despite its practical importance [5, 8, 11, 15, 22, 25, 26, 27, 37] . Prior works have addressed aspects such as keep-in/keep-out zone enforcement, actuator saturation, and slew-rate limits, but often in isolation rather than within a unified formulation. This gap

highlights the need for integrated methods that can simultaneously accommodate geometric, actuator, and kinematic constraints with provable guarantees.

Convex optimization, particularly semi-definite programming (SDP) utilizing linear matrix inequalities (LMIs), has emerged as a powerful framework for handling nonconvex attitude constraints [15]. By exploiting the quaternion's unit norm property, quadratic cone constraints can be transformed into convex LMIs, enabling globally optimal SDP formulations solvable in polynomial time [16]. This convexification facilitates explicit encoding of keep-in/keep-out cones, actuator saturation limits, slew-rate bounds, and quaternion normalization within a single offline optimization. In this work, quaternion trajectories are discretized via direct collocation to achieve accurate and tractable SDP solutions.

Building on these advances, this thesis proposes a hybrid architecture where an offline SDP guidance algorithm generates constraint-compliant, optimal reference trajectories under keep-in/out and hardware limits. A real-time quaternion-feedback PD controller then tracks these trajectories. This hybrid approach leverages the systematic constraint enforcement and global optimality of SDP, together with the simplicity and disturbance rejection properties of classical feedback control, providing a practical, high-performance solution for constrained spacecraft attitude maneuvers.

1.3 Prior Work

Hablani (1999) [11] preceded the modern convex-optimization era by tackling simultaneous bright-object avoidance and ground-station communication in a purely geometric framework. He derived closed-form "step" slews that keep the boresight tangential to forbidden Sun/Moon cones while superimposing a roll maneuver to maintain the antenna beam on Earth. By analytically choosing which side of the exclusion cone to follow, and sliding the boresight along its boundary when needed-Hablani guaranteed both safety and connectivity without any online optimization. While elegant, this approach could not systematically incorporate multiple simultaneous cones or enforce torque and rate limits, motivating convex-optimization methods that followed.

Kim and Mesbahi (2004) [15] reframed the Quadratically-Constrained Attitude Control (Q-CAC) problem as a Semi-Definite Program (SDP). Exploiting the quaternion's unit-norm

as an implicit magnitude constraint, they showed that each nonconvex attitude-cone inequality can be shifted and cast as a single Linear Matrix Inequality (LMI). Proposition 3.1 formally proves that once $\|q\| = 1$ is enforced, any finite collection of quadratic keep-out constraints admits a globally solvable SDP formulation. With torque and rate bounds also written as LMIs and discretized, an SDP is solved at each time step; a 50° sun-avoidance maneuver demonstrates the quaternion trajectory sliding along the exclusion boundary without violation. Although this work handled arbitrary keep-out cones, practical missions soon demanded handling both keep-in and keep-out simultaneously, which led to unified parameterizations.

Kim *et al.* (2010) [16] introduced a unified convex parameterization for any finite collection of quadratic attitude constraints-including both keep-in and keep-out cones-in a single SDP. By embedding the norm constraint into each LMI and applying an eigenvalue shift, they convexified multiple nonconvex inequalities simultaneously, paving the way for multi-cone attitude guidance in one optimization problem. However, the reliance on continuous SDPs and real-time solver calls spurred interest in purely geometric, discrete methods.

Kjellberg and Lightsey (2013) [17] proposed a geometric alternative that tessellates the unit-quaternion sphere into waypoints and casts constrained reorientation as an A* graph-search. Their method naturally handles arbitrary keep-in/keep-out sets by precomputing collision-free waypoint paths and then steering between them with a feedback controller-achieving real-time onboard planning without semi-definite optimization. While fast, graph-search approaches can become unwieldy as constraint complexity grows, motivating renewed focus on convex relaxations.

Sun and Dai (2015) [28] revisited the underlying Quadratically Constrained Quadratic Program (QCQP) relaxation, observing that standard SDP relaxations sometimes produce loose bounds. They developed an iterative rank-minimization algorithm that enforces the hidden rank-one structure of the quaternion relaxation, converging to the exact nonconvex solution and improving accuracy for highly constrained or large-scale maneuvers. To incorporate logical combinations of constraints, researchers then turned to mixed-integer formulations.

Tam and Lightsey (2016) [29] embedded binary decision variables into the SDP/QCQP framework, yielding a Mixed-Integer Convex Program (MICP) capable of expressing

"keep-one-of-many" and other Boolean conditions. Although this introduces combinatorial complexity, modern MICP solvers efficiently handle small numbers of binaries, enabling onboard logic-aware attitude planning. Complementing these discrete/convex hybrid methods, continuous-time feedback alternatives also matured.

Hu *et al.* (2020) [12] proposed potential-function methods with saturation. By designing a quaternion-based artificial potential with a unique global minimum and bounded gradient, then wrapping it in a saturated feedback law, these approaches guarantee almost-global convergence under actuator limits-without requiring SDP calls. Yet for multi-criteria missions, algorithms requiring trade-offs emerged.

Xu *et al.* (2020) [38] advanced the state of the art with a multi-objective dynamic-iteration planner that trades off energy, time, and path-quality metrics. Their piecewise-quadratic programming and rotational-path decomposition generate high-quality trajectories onboard at a fraction of the cost of full SDP optimization. Finally, dynamic exclusion/inclusion regions required further generalization.

Cruz and Bani Younes (2022) [6] extended constrained attitude control to handle dynamic keep-in/keep-out zones by deriving common-frame attitude kinematics and embedding them in a Lyapunov-backstepping steering law. This framework is critical for formation-flying missions where exclusion and inclusion regions move relative to the spacecraft.

Underpinning these methods is a hierarchy of convex optimization formulations. Quadratic Programs (QPs) and convex Quadratically Constrained Quadratic Programs (QCQPs) via LMIs form the foundation, ascending to semi-definite Programs (SDPs) [31]. Second-Order Cone Programs (SOCPs) lie intermediate, while Mixed-Integer extensions (MICP) and iterative rank enforcement sit atop, trading modeling expressiveness for computational complexity. In practice, one selects the lowest level of this hierarchy that exactly models the problem at hand, balancing solver speed against formulation richness.

This thesis focuses on foundational SDP-based trajectory planning with static keep-in/out cones and classical quaternion feedback control. While more advanced methods-including logical constraints, dynamic zones, and formation control-represent exciting directions, they lie beyond this work's scope and are discussed here to contextualize ongoing research.

1.4 Formal Problem Statement

This section formally states the constrained spacecraft attitude control problem considered in this thesis.

As surveyed by Wang [33], a key feature of autonomous guidance and control (G&C) systems is the onboard computation of steering laws that safely and efficiently reorient spacecraft under complex mission constraints. This problem can be posed as an *Optimal Control Problem* (OCP), which seeks a sequence of control torques $u(0), \ldots, u(N-1)$ that minimize a prescribed cost function over a fixed maneuver interval, subject to

- prescribed initial and terminal states (attitude and angular velocity),
- nonlinear quaternion kinematics and Euler rotational dynamics,
- bounds on angular velocities and control inputs,
- and geometric keep-in and keep-out constraints on the spacecraft's boresight directions.

The cost function typically penalizes attitude and rate errors along with control effort, ensuring smooth and efficient maneuvers.

Specifically, the problem is to steer a rigid spacecraft from a known initial attitude and angular velocity to a specified final state within a fixed maneuver time. The spacecraft's orientation is represented by a unit quaternion $q(k) \in \mathbb{R}^4$, and its rotational dynamics obey Euler's equations under control torques generated by reaction wheels or thrusters.

The guidance must satisfy several mission-critical constraints simultaneously:

- ***Geometric Exclusion (Keep-Out Cones):*** Sensitive optical instruments, such as star trackers or scientific cameras, must avoid pointing within specified half-angles of bright celestial bodies (e.g., the Sun or Moon). These forbidden directions define keep-out cones in body-fixed coordinates that the spacecraft's boresight must avoid throughout the maneuver.
- ***Geometric Inclusion (Keep-In Cones):*** Certain sensors, including sun sensors and high-gain antennas, require the target (e.g., Sun, Earth, or other reference) to remain within prescribed keep-in cones to maintain a valid line-of-sight for navigation or communication.
- ***Actuator Authority Limits:*** Control torques $u(k) \in \mathbb{R}^3$ are bounded by actuator

capabilities, such as $\|u(k)\|_\infty \leq \tau_{\max}$. Exceeding these limits may saturate actuators, risking loss of control or component damage.

- ***Angular Rate Restrictions:*** Body rates $\omega(k) \in \mathbb{R}^3$ must remain below thresholds imposed by sensor bandwidth and gyroscope saturation, often enforced as $\|\omega(k)\|_\infty \leq \omega_{\max}$, to preserve attitude determination accuracy.
- ***Quaternion Normalization:*** To maintain a consistent, singularity-free attitude representation, the quaternion must remain unit norm at all times: $\|q(k)\| = 1$.

These requirements present several challenges: the attitude dynamics are nonlinear and coupled; the keep-in/out and quaternion normalization constraints introduce nonconvexities; and actuator and rate limits impose hard bounds on inputs and states. The objective is to compute a control torque profile $u(0), \ldots, u(N-1)$ that respects all constraints, drives the spacecraft precisely to the desired terminal state, and minimizes a suitably defined cost reflecting attitude and rate errors.

1.5 Proposed Solution: Convex Optimization and Feedback Control

This thesis proposes a two-layer control architecture for constrained spacecraft attitude maneuvers, which combines the systematic rigor of convex optimization with the real-time robustness of classical feedback control.

Goal

The objective is to accurately track a desired spacecraft orientation trajectory while rigorously satisfying mission-critical constraints such as geometric keep-in/keep-out zones, actuator saturation limits, angular rate bounds, and the nonlinear dynamics inherent in spacecraft rotation.

Step 1: Convex Optimization for Guidance

- The constrained attitude guidance problem is reformulated as a convex optimization problem using Semi-Definite Programming (SDP) and Linear Matrix Inequalities (LMIs). This approach enables the explicit incorporation of complex geometric and

actuator constraints in a globally optimal trajectory planning framework.

- This convex optimization generates a safe and constraint-compliant reference trajectory, offline, that guides the spacecraft's orientation throughout the maneuver, avoiding forbidden attitude regions while respecting physical actuator and rate limits.
- By solving the trajectory planning problem offline, computational burden is reduced during real-time operation, enabling efficient onboard implementation.

Step 2: Feedback for Robust Execution

- A quaternion-feedback regulator, based on proportional-derivative (PD) control, is employed to track this precomputed reference trajectory in real time.
- This feedback controller compensates for model uncertainties, external disturbances, and unmodeled dynamics, thereby enhancing robustness and ensuring precise attitude tracking.
- The simplicity and low computational cost of the feedback control law make it suitable for implementation on spacecraft onboard processors.

By integrating the global optimality and rigorous constraint satisfaction of SDP-based planning with the disturbance rejection and simplicity of quaternion-feedback control, the proposed hybrid framework offers a practical, reliable, and high-performance solution for constrained spacecraft attitude control.

1.6 Thesis Organization

This thesis is organised as follows: Chapter 1 (Introduction) motivates constrained spacecraft attitude guidance and control, explains why the problem is challenging, reviews prior work, states the formal problem, and previews the proposed solution-an SDP-based constrained guidance layer paired with a quaternion feedback regulator-concluding with this organization. Chapter 2 (Background and Mathematical Preliminaries) assembles the required foundations: rigid-body dynamics, quaternion algebra and reference frames, classical PID control, and the convex-optimization tools (convex sets, LMIs, and the Schur complement) underlying constrained attitude control via SDP. Chapter 3 (Problem Formulation and Methodology)

presents the approach: it defines the quaternion feedback regulators; derives the SDP formulation encoding keep-in/keep-out cones, actuator-torque and body-rate limits, and quaternion normalization; and details the proposed control-and-guidance architecture in which the SDP solver generates constraint-feasible reference states tracked in real time. Chapter 4 (Results) evaluates performance across four scenarios-nominal (no constraint), one keep-out (1-KO SDP), two keep-out (2-KO SDP), and combined keep-in/keep-out-verifying constraint satisfaction and comparing against an unconstrained PD baseline in tracking accuracy and control effort. Chapter 5 (Conclusion) summarises contributions, discusses strengths and limitations, and outlines applications and future work. The Appendix lists where to find all Simulink models, MATLAB scripts, and simulation data used in this thesis.

Chapter 2

BACKGROUND AND MATHEMATICAL PRELIMINARIES

This chapter presents the mathematical foundations and theoretical background required for the development of the proposed constrained attitude control framework. We begin by reviewing quaternion-based formulations for satellite attitude representation and control, covering their advantages over alternative parameterisations. Fundamental concepts from rigid-body dynamics, reference frames, and classical control laws are then introduced, along with the principles of convex optimization that underpin the semidefinite programming (SDP) approach used in this work. The material in this chapter provides the essential tools for understanding the problem formulation and methodology discussed in subsequent chapters.

2.1 Quaternion-Based Satellite Attitude Control

This chapter introduces the theoretical framework of quaternion-based satellite attitude control. Quaternions provide an efficient and singularity-free method to represent rotations in three-dimensional space, essential for accurate satellite attitude control. The chapter describes the fundamental equations governing satellite motion, introduces quaternion mathematics, and then integrates these elements to describe the dynamics and kinematics of quaternion-based satellite attitude.

2.1.1 Satellite Attitude Dynamics

Satellite attitude dynamics describes how a spacecraft's orientation evolves in response to external torques and its internal inertia properties.

Rigid Body Mechanics

A rigid body is an idealization of a body that does not deform or change shape. Formally, it is defined as a collection of particles with the property that the distance between particles

remains unchanged during the course of motions of the body. [3] Mathematically, a rigid body can be described by a body-fixed coordinate frame **B**, within which the position vectors of all particles are constant. [20]

The motion of a rigid body in space consists of the translational motion of its center of mass and the rotational motion of the body about its center of mass; thus, a rigid body in space is a dynamic system with six degrees of freedom. Here we are concerned with the rotational motion of a rigid vehicle with or without the influence of gravitational and other external forces. [35]

Angular Momentum

Angular momentum, which is the rotational analog of linear momentum, is the measure of rotational motion of an object. It depends on how fast it's spinning, how its mass is distributed, and about what axis it's spinning.

The angular momentum **H** of a rigid spacecraft, about an arbitrary point O, of the linear momentum **mv** of a particle **P** of mass **m** is

$$\boldsymbol{H}^{O} = \boldsymbol{r} \times m\boldsymbol{v} \tag{2.1}$$

If we consider a spacecraft (or anything else) to be made up of a collection of n point masses. The total angular momentum of the spacecraft is obtained by summing the angular momenta of all its constituent particles:

$$\boldsymbol{H}^{O} = \sum_{i=1}^{n} \boldsymbol{r}^{iO} \times m_i \boldsymbol{v}^{iO} \tag{2.2}$$

where $\boldsymbol{m_i}$, $\boldsymbol{r^{iO}}$ and $\boldsymbol{v^{iO}}$ represent the mass, position and velocity vector of the particle $\boldsymbol{i}$ with respect to the center of mass $\boldsymbol{O}$, respectively.

Angular Velocity

Angular velocity is a vector quantity, often expressed as $\boldsymbol{\omega} = [\boldsymbol{\omega_x}, \boldsymbol{\omega_y}, \boldsymbol{\omega_z}]^{\boldsymbol{T}}$, that refers to the time rate of change of its angular position relative to the origin. Its direction indicates

the axis of rotation according to the right-hand rule, and its magnitude represents how fast the body is rotating about that axis.

Moment of Inertia

The moments of inertia characterize the distribution of mass within a rigid body relative to its axis of rotation. They determine the body's resistance to changes in its rotational state, that is, being set into rotary motion or stopped once rotation is underway. The response of a body to applied torques depends not solely on its mass, but on how that mass is spatially distributed. [7]

For a point mass, the moment of inertia is given by the product of the mass and the square of its distance from the axis of rotation:

$$J = mr^2 \tag{2.3}$$

For a distributed rigid body, the total moment of inertia is obtained by integrating r^2 over the entire mass distribution.

Therefore, in matrix notation, the angular momentum of a rigid body in motion about the origin can be expressed as

$$\boldsymbol{H} = J\boldsymbol{\omega} \tag{2.4}$$

where $\boldsymbol{J}$ is the moment of inertia, and $\boldsymbol{\omega}$ is the orbital angular velocity.

We can express this relation component-wise as:

$$\begin{Bmatrix} H_x \\ H_y \\ H_z \end{Bmatrix} = \begin{bmatrix} I_x & I_{xy} & I_{xz} \\ I_{yx} & I_y & I_{yz} \\ I_{zx} & I_{zy} & I_z \end{bmatrix} \begin{Bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix} \tag{2.5}$$

For rigid bodies with symmetric mass distributions, the inertia matrix is symmetric, implying that the off-diagonal terms satisfy $I_{xy} = I_{yx}$, $I_{xz} = I_{zx}$, and $I_{yz} = I_{zy}$.

2.1.2 Quaternion Representation

Motivation for Using Quaternions

Euler angles are a common method for describing the orientation of a rigid body, defining rotations around three successive axes to align the inertial frame with the body frame. Although they offer an intuitive and compact representation, Euler angles suffer from a critical limitation known as **gimbal lock**. Gimbal lock occurs when the pitch angle, θ, approaches $\pm 90^{\circ}$, causing two of the rotational axes to become aligned and resulting in the loss of one degree of freedom. It arises from the collinearity of the physical rotation axis vectors of the first and third rotations in the sequence. [20]

For many engineering systems involving constrained or limited three-dimensional motion, Euler angles are sufficient. However, for satellites, which require continuous and unrestricted rotational motion for tasks such as attitude pointing, target tracking, and orbital maneuvers, the risk of encountering gimbal lock becomes critical. Loss of rotational freedom can severely degrade control capabilities and destabilize the spacecraft.

Quaternions provide a robust alternative by offering a singularity-free, compact, and computationally efficient method for representing rotations. Unlike Euler angles, quaternions maintain smooth and stable interpolation between orientations, ensuring reliable performance during complex satellite attitude maneuvers.

Quaternion Introduction

Quaternions, also known as Euler symmetric parameters, were introduced in 1843 by the Irish mathematician Sir William R. Hamilton (1805-1865). [7] A quaternion is a four-component number system that extends complex numbers and provides an efficient means to represent three-dimensional rotations.

We consider a quaternion $\boldsymbol{q}$ to consist of a three-dimensional vector part and a scalar part, and express it as

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_{1:3} \\ q_4 \end{bmatrix} \quad \text{where} \quad \boldsymbol{q}_{1:3} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{2.6}$$

In this representation, $\boldsymbol{q}_{1:3}$ denotes the vector part and q_4 denotes the scalar part of the quaternion[1]. [20]

Axis-Angle Representation

A unit quaternion can be directly related to the axis-angle representation of a three-dimensional rotation. Specifically, any rotation in three-dimensional space can be described by a rotation of angle θ about a fixed unit vector $\boldsymbol{e}$, often referred to as the *Euler axis* or the *Eigen-axis*.

Given a rotation of angle θ about the axis $\boldsymbol{e} = (e_1, e_2, e_3)^T$, the corresponding unit quaternion $\boldsymbol{q}$ is defined as

$$\boldsymbol{q}_{1:3} = \boldsymbol{e} \sin\left(\frac{\theta}{2}\right), \quad q_4 = \cos\left(\frac{\theta}{2}\right) \tag{2.7}$$

where:

- $\boldsymbol{q}_{1:3}$ is the vector component,
- q_4 is the scalar component of the quaternion,
- $\boldsymbol{e}$ is the unit eigenaxis (Euler axis) about which the rotation occurs,
- θ is the rotation angle about the axis $\boldsymbol{e}$.

Thus, the quaternion compactly encodes both the rotation axis and the rotation angle within a four-dimensional structure. This representation avoids the singularities and discontinuities associated with other rotation parameterizations, such as Euler angles.

2.1.3 Quaternion Operations

This section introduces the basic quaternion operations utilized in this project. Quaternions support several fundamental operations that are critical for describing and manipulating

[1] MATLAB/Simulink follows a scalar-first quaternion convention $(q_4, \boldsymbol{q}_{1:3})$. This thesis adopts the scalar-last notation for consistency with most aerospace literature, but conversions are handled appropriately in code.

rotations. The main operations used in attitude control are multiplication, conjugation, normalization, and error computation.

Quaternion Multiplication

Quaternion multiplication combines two rotations represented by quaternions. Given two quaternions $\boldsymbol{q}_1 = (\boldsymbol{q}_{1,1:3}, q_{1,4})$ and $\boldsymbol{q}_2 = (\boldsymbol{q}_{2,1:3}, q_{2,4})$, their product $\boldsymbol{q} = \boldsymbol{q}_1 \otimes \boldsymbol{q}_2$ is defined as

$$\boldsymbol{q} = \boldsymbol{q}_1 \otimes \boldsymbol{q}_2 = \begin{bmatrix} q_{1,4}q_{2,4} - \boldsymbol{q}_{1,1:3}^T \cdot \boldsymbol{q}_{2,1:3} \\ q_{2,4}\boldsymbol{q}_{1,1:3} + q_{1,4}\boldsymbol{q}_{2,1:3} - \boldsymbol{q}_{1,1:3} \times \boldsymbol{q}_{2,1:3} \end{bmatrix} \tag{2.8}$$

where $\times$ denotes the vector cross product and $\cdot$ denotes the dot product. [20]

Quaternion multiplication is **non-commutative**, meaning that $\boldsymbol{q}_1 \otimes \boldsymbol{q}_2 \neq \boldsymbol{q}_2 \otimes \boldsymbol{q}_1$, in general.

Quaternion Normalization

Quaternions used to represent rotations must be unit quaternions, satisfying

$$\|\boldsymbol{q}\| = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2} = 1 \tag{2.9}$$

In numerical simulations, small errors can accumulate and cause the quaternion norm to drift from unity. Therefore, periodically re-normalizing the quaternion is standard practice to maintain its validity.

Quaternion Conjugate and Inverse

The conjugate of a quaternion $\boldsymbol{q} = (q_1, \boldsymbol{q}_{2:4})$ is given by

$$\boldsymbol{q}^* = \begin{bmatrix} \boldsymbol{q}_{1:3} \\ q_4 \end{bmatrix}^* = \begin{bmatrix} -\boldsymbol{q}_{1:3} \\ q_4 \end{bmatrix} = \begin{bmatrix} -q_1 \\ -q_2 \\ -q_3 \\ q_4 \end{bmatrix} \tag{2.10}$$

The inverse of a quaternion with nonzero norm is defined as

$$\boldsymbol{q}^{-1} = \frac{\boldsymbol{q}^*}{\|\boldsymbol{q}\|^2} \tag{2.11}$$

For a unit quaternion, where $\|\boldsymbol{q}\| = 1$, the inverse simplifies to the conjugate:

$$\boldsymbol{q}^{-1} = \boldsymbol{q}^* \tag{2.12}$$

Quaternion Error Computation

In attitude control, the difference between the current and the desired orientation is represented using the quaternion error. Given the current quaternion $\boldsymbol{q}$ and the desired quaternion $\boldsymbol{q}_d$, the quaternion error $\delta\boldsymbol{q}$ is defined as

$$\delta\boldsymbol{q} \equiv \begin{bmatrix} \delta\boldsymbol{q}_{1:3} \\ \delta q_4 \end{bmatrix} = \boldsymbol{q} \otimes \boldsymbol{q}_d^{-1} \tag{2.13}$$

where $\boldsymbol{q}_d^{-1}$ is the inverse of the desired quaternion $\boldsymbol{q}_d$, given by

$$\boldsymbol{q}_d^{-1} = \frac{\boldsymbol{q}_d^*}{\|\boldsymbol{q}_d\|^2} \tag{2.14}$$

and $\boldsymbol{q}_d^*$ denotes the conjugate of $\boldsymbol{q}_d$.

Expanding the Hamilton product, the quaternion error components are given by

$$\delta\boldsymbol{q} = \begin{bmatrix} q_4 q_{d,4}^{-1} - \boldsymbol{q}_{1:3}^T \cdot \boldsymbol{q}_{d,1:3}^{-1} \\ q_{d,4}^{-1}\boldsymbol{q}_{1:3} + q_4\boldsymbol{q}_{d,1:3}^{-1} - \boldsymbol{q}_{1:3} \times \boldsymbol{q}_{d,1:3}^{-1} \end{bmatrix} \tag{2.15}$$

where $\times$ denotes the vector cross product and $\cdot$ the dot product.

To ensure numerical stability, the error quaternion is normalized after computation:

$$\delta\boldsymbol{q} = \frac{\delta\boldsymbol{q}}{\|\delta\boldsymbol{q}\|} \tag{2.16}$$

2.2 Reference Frames

Accurate attitude determination and control depend critically on well-defined coordinate frames. A three-dimensional reference frame is completely specified by its origin's location and its orthonormal axes' orientation. In spacecraft attitude analysis, one typically works

with a small set of canonical frames-most notably the inertial frame, the body-fixed frame, and a local-level frame-each of which plays a distinct role in kinematics, dynamics, and guidance [20]. In this section, we introduce these frames and show how they are implemented in MATLAB for quaternion-based control.

2.2.1 Spacecraft Body Frame $\mathcal{F}_B$

The body frame is fixed to the spacecraft itself. Its origin is usually placed at the center of mass or a designated sufficiently rigid *navigation base*, and its three axes follow the vehicle's main structural or sensor directions. For example, one axis might point along the antenna boresight and another along a solar-panel boom. In practice, these axes guide component alignment on the ground during assembly and remain effectively rigid in orbit. Throughout our control algorithms, we track and command the orientation of

$$\mathcal{F}_B = \{\hat{\mathbf{x}}_B, \hat{\mathbf{y}}_B, \hat{\mathbf{z}}_B\}$$

relative to other frames to keep the spacecraft pointed correctly.

2.2.2 Inertial Frame ($\mathcal{F}_I$)

An inertial reference frame moves at constant velocity and does not rotate, so that Newton's laws of motion hold. The most precise realization of an Earth-centered inertial frame is the International Celestial Reference Frame (ICRF), whose axes are fixed relative to several hundred distant extragalactic radio sources via very-long-baseline interferometry [20].

We denote this frame by

$$\mathcal{F}_I \;=\; \{\hat{\mathbf{i}}, \hat{\mathbf{j}}, \hat{\mathbf{k}}\},$$

where

- $\hat{\mathbf{k}}$ points toward the mean north pole of Earth,
- $\hat{\mathbf{i}}$ lies in the equatorial plane, pointing toward the mean vernal equinox (the intersection of Earth's equator with the ecliptic),
- $\hat{\mathbf{j}} = \hat{\mathbf{k}} \times \hat{\mathbf{i}}$ completes a right-handed system.

Because $\mathcal{F}_I$ is nonrotating with respect to the distant universe, it provides a stable baseline for expressing spacecraft attitudes, kinematics, and trajectories.

2.2.3 Earth-Centered, Earth-Fixed Frame ($\mathcal{F}_E$)

The Earth-Centered, Earth-Fixed (ECEF) frame shares its origin with the inertial frame at Earth's center but rotates with the planet. We denote its axes by

$$\mathcal{F}_E = \{\hat{\mathbf{e}}_1,\ \hat{\mathbf{e}}_2,\ \hat{\mathbf{e}}_3\},$$

where

- $\hat{\mathbf{e}}_3$ coincides with the Earth's rotation axis (north pole),
- $\hat{\mathbf{e}}_1$ points toward the Earth's prime meridian (the Greenwich meridian),
- $\hat{\mathbf{e}}_2 = \hat{\mathbf{e}}_3 \times \hat{\mathbf{e}}_1$ completes a right-handed system.

2.3 Proportional-Integral-Derivative (PID) Control

Before we introduce quaternion feedback, it helps to recall the classical PID loop, which underpins many aerospace and industrial controllers [9].

2.3.1 Open-Loop Control

In an open-loop architecture (Figure 2.1), the controller $D_{ol}(s)$ acts directly on the reference $R(s)$, and the plant $G(s)$ produces the output $Y(s)$. Any disturbances $W(s)$ entering before the plant cannot be rejected, nor can model errors be corrected.

2.3.2 Closed-Loop (Feedback) Control

In closed-loop control (Figure 2.2), the plant output $Y(s)$ is subtracted from the reference $R(s)$, and the resulting error is sent through the controller $D(s)$. This feedback path allows the system to reject disturbances $W(s)$ and correct for model mismatches, at the cost of potentially amplifying sensor noise $V(s)$.

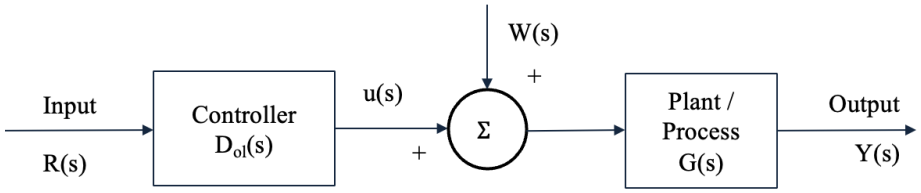


Figure 2.1: Open-loop control: the controller $D_{ol}(s)$ processes only the reference $R(s)$; disturbances $W(s)$ directly affect the plant output.

Explanation: Here $U(s) = D_{ol}(s)\,R(s)$ is applied to $G(s)$. No feedback means the system cannot compensate for $W(s)$ or plant uncertainties.

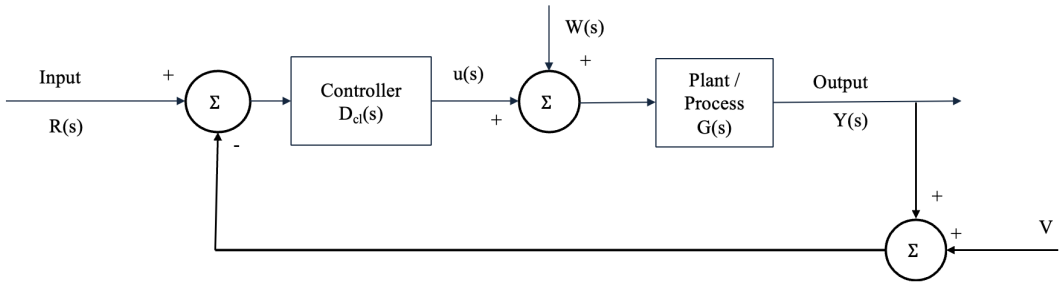


Figure 2.2: Closed-loop feedback control: the controller $D(s)$ acts on the error $E(s) = R(s) - Y(s)$; disturbances $W(s)$ and noise $V(s)$ enter at dedicated summing junctions.

Explanation: The control law $U(s) = D(s)\,E(s)$ enables disturbance rejection and robustness to uncertainties, while sensor noise $V(s)$ may be amplified by the loop gain.

The PID Law

The most common feedback structure is the PID controller, whose transfer function is

$$D(s) \;=\; K_P \;+\; \frac{K_I}{s} \;+\; K_D\, s.$$

here:

- K_P (*proportional*) shapes the overall responsiveness to error;
- K_I (*integral*) eliminates steady-state error by accumulating past errors (essential for regulation);
- K_D (*derivative*) adds damping and improves tracking of rapidly-changing references, at the risk of amplifying high-frequency noise.

Regulation Regulation is the task of maintaining the system output at a fixed set-point even in the presence of constant disturbances. In a feedback loop, the controller continually measures the error-the difference between the output and the desired value-and adjusts its input to drive that error toward zero. By including an integral term (which accumulates any persistent error over time), the controller guarantees that any steady-state offset is eliminated, ensuring zero steady-state error against constant disturbances.

Tracking Tracking is the task of forcing the system output to follow a time-varying reference signal as closely as possible. In a feedback loop, the controller continuously drives the error $e(t) = r(t) - y(t)$ toward zero even as $r(t)$ changes over time. Proportional gain (K_P) reduces the instantaneous error, derivative action (K_D) anticipates and damps rapid changes in the reference, and integral action (K_I) removes any steady-state offset for slowly varying or step-like commands. A well-tuned PID therefore, achieves both a fast, low-overshoot response to changing setpoints and zero long-term tracking error.

2.4 Constrained Attitude Control using Semi-Definite Programming (SDP)

This section begins by motivating the need for SDP in spacecraft attitude control before presenting the formulation and implementation details.

As discussed in Chapter 1, modern spacecraft missions demand highly precise attitude control under increasingly stringent operational constraints. While classical quaternion-based feedback controllers-such as proportional-derivative (PD) and proportional-integral-derivative (PID) laws-are well-established for unconstrained maneuvers due to their simplicity and computational efficiency, they lack systematic methods to rigorously enforce mission-critical constraints. These constraints include sun-avoidance keep-out cones to protect sensitive instruments, actuator torque limitations, and strict field-of-view requirements. The reliance on heuristic gain scheduling or conservative tuning in classical approaches often leads to suboptimal performance and reduced safety margins.

Convex optimization, specifically Semi-Definite programming (SDP), provides a promising framework for overcoming these challenges. By exploiting the geometric properties of quaternions and the mathematical structure of spacecraft rotational dynamics, originally nonconvex attitude constraints can be reformulated as convex problems through linear matrix inequalities (LMIs) and convex relaxations. This transformation enables the derivation of globally optimal, computationally efficient solutions with theoretical guarantees of constraint satisfaction.

The strength of SDP-based methods lies in their ability to deliver certifiable control performance for mission-critical spacecraft operations, supported by mature numerical solvers such as YALMIP and CVX. In this chapter, we develop a comprehensive SDP-based control framework tailored to quaternion-based spacecraft attitude maneuvers. Beginning with the essential principles of convex optimization, we derive an SDP formulation that incorporates critical constraints-including quaternion unit norms, sun-avoidance exclusion zones, and actuator torque bounds. We further discuss practical implementation strategies to ensure computational tractability for real-time and embedded applications.

2.4.1 Optimization Foundations for Attitude Control

Many spacecraft guidance and control problems can be formulated as constrained optimization tasks, where the objective is to compute trajectories or control inputs that satisfy system dynamics and mission-specific constraints. When these problems are structured to meet

the criteria of convexity, powerful tools from convex optimization theory can be applied to efficiently find globally optimal solutions. Convex optimization methods are especially attractive in this context due to their strong theoretical guarantees and robust numerical performance.

This section introduces key concepts from convex analysis that underpin the mathematical foundation of Semi-Definite Programming (SDP) approaches to constrained attitude control. The discussion follows the theoretical framework presented in *Convex Optimization* by Boyd and Vandenberghe [2].

Convex Sets and Convex Functions

A set $\mathcal{C} \subseteq \mathbf{R}^n$ is defined as *convex* if, for any pair of points $x_1, x_2 \in \mathcal{C}$, the entire line segment connecting them lies within the set. Mathematically, this condition is expressed as

$$\theta x_1 + (1-\theta)x_2 \in \mathcal{C}, \quad \forall \theta \in [0,1]. \tag{2.17}$$

Figure 2.3 provides visual examples of sets that satisfy or violate convexity. In the convex case, the line segment between any two points remains inside the set, whereas the nonconvex examples fail this property due to boundary gaps or concave geometry.

A function $f : \mathbf{R}^n \to \mathbf{R}$ is called *convex* if its domain is a convex set and it satisfies the inequality

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y), \quad \forall x, y \in \operatorname{dom} f,\ \theta \in [0,1]. \tag{2.18}$$

This condition implies that the function lies below the chord connecting any two points on its graph. Geometrically, the graph of a convex function is such that the straight line segment between any two points $(x, f(x))$ and $(y, f(y))$ lies above the curve itself.

If the inequality is strict for all $x \neq y$ and $\theta \in (0,1)$, the function is said to be *strictly convex*.

A function f is said to be *concave* if its negative, $-f$, is convex. This definition ensures that many analytical results developed for convex functions can also be applied to concave functions by sign inversion.

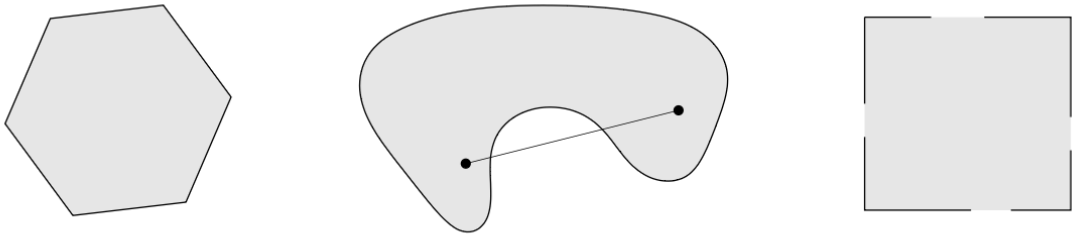


Figure 2.3: Illustration of convex and nonconvex sets. *Left*: A convex hexagon including its boundary, where any line segment between two points lies entirely within the set. *Middle*: A nonconvex, kidney-shaped region, where a chord between two interior points exits the boundary. *Right*: A square missing boundary segments violates the convexity condition. Adapted from *Convex Optimization* by Boyd and Vandenberghe [2].

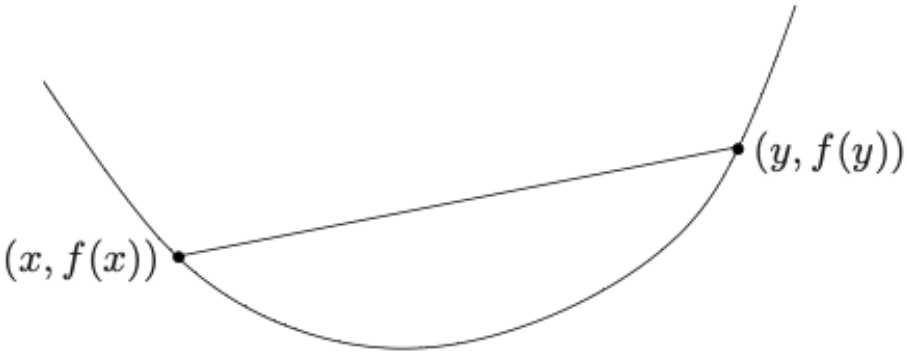


Figure 2.4: Visualization of a convex function. The line segment connecting two points on the graph lies above the function, illustrating the convexity condition. Adapted from *Convex Optimization* by Boyd and Vandenberghe [2].

One important property of convex functions is their continuity on the relative interior of their domain. While discontinuities can occur at the domain's boundary, convex functions are guaranteed to be continuous wherever the domain is not degenerate [2].

Convexity is central to optimization because it guarantees that any local minimum is also a global minimum. This property significantly simplifies both the theoretical analysis and the practical solution of optimization problems.

Linear Matrix Inequalities and the Schur Complement

Linear Matrix Inequalities in System and Control Theory by Boyd *et al.* [1] shows how many convex constraints can be recast as *linear matrix inequalities* (LMIs) and solved efficiently with Semi-Definite Programming (SDP).

Definition. An LMI is an affine matrix expression constrained to be positive semi-definite:

$$F(x) = F_0 + \sum_{i=1}^{m} x_i F_i \ \succeq \ 0, \qquad F_0, F_i \in \mathbf{S}^n, \ x \in \mathbf{R}^m. \tag{2.19}$$

Because semi-definiteness is preserved under affine combinations, the feasible set $\{x \mid F(x) \succeq 0\}$ is convex.

LMIs encompass linear inequalities, convex quadratic and norm bounds, and the Lyapunov-type matrix conditions are widely used in robust control applications [1]. When several LMIs appear simultaneously, they can be merged into a single block-diagonal constraint:

$$F^{(1)}(x) \succeq 0, \ \ldots, \ F^{(K)}(x) \succeq 0 \quad \Longleftrightarrow \quad \operatorname{diag}\big(F^{(1)}(x), \ldots, F^{(K)}(x)\big) \succeq 0. \tag{2.20}$$

Hence, we make no distinction between a set of LMIs and one LMI-an observation that greatly simplifies SDP modelling.

Schur complement. Let $X = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \in \mathbf{S}^n$, with $A \in \mathbf{S}^p$ non-singular (that is, $\det A \neq 0$) and $C \in \mathbf{S}^q$. The *Schur complement* of A in X is

$$S \ = \ C - B^T A^{-1} B. \tag{2.21}$$

The following statements are equivalent:

$$\begin{aligned} X \succ 0 &\iff A \succ 0 \text{ and } C - B^T A^{-1} B \succ 0 \quad \text{(Schur complement of } A\text{)} \\ &\iff C \succ 0 \text{ and } A - B C^{-1} B^T \succ 0 \quad \text{(Schur complement of } C\text{)}. \end{aligned} \tag{2.22}$$

This Schur-complement lemma turns many nonlinear convex constraints into LMIs and yields the determinant factorisation [2].

$$\det X = \det A \, \det S$$

Throughout the thesis, Schur-complement reformulations are employed to turn visibility limits and actuator-saturation bounds into LMIs, allowing these nonlinear constraints to be handled within the unified SDP framework for attitude-control framework.

Convex Optimization Problems

General Form [2] helps us explaining the optimization problem and the types .

A finite-dimensional optimization problem can be written in the form

$$\begin{aligned} \underset{x \in \mathbf{R}^n}{\text{minimise}} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \le 0, \quad i = 1, \ldots, m, \\ & h_j(x) = 0, \quad j = 1, \ldots, p, \end{aligned} \tag{2.23}$$

where $x \in \mathbf{R}^n$ is the *decision vector*, $f_0 : \mathbf{R}^n \to \mathbf{R}$ is the *objective* (*cost*) function, each f_i defines an *inequality constraint*, and each h_j defines an *equality constraint*.

Convex case Problem (2.23) is *convex* if

- the objective f_0 is convex,
- every inequality function f_i is convex, and
- each equality constraint h_j is affine, i.e. $h_j(x) = a_j^T x - b_j$.

Under these conditions, the feasible set is convex, and any local minimizer is therefore a global minimizer [2]. In other words, a convex optimization problem minimizes a convex objective over a feasible region formed by the intersection of convex inequalities and affine equalities.

Typical spacecraft constraints-such as actuator torque limits, sun-avoidance pointing cones, and momentum-wheel saturation-either are inherently convex or can be convexified. As a result, convex optimization provides a reliable, high-performance foundation for onboard guidance and control algorithms.

Semi-Definite Programming (SDP)

Semi-definite programming is an optimization framework in which a linear objective is minimized subject to a *linear matrix inequality* (LMI) that forces an affine combination of symmetric matrices to be positive semi-definite [31]. Since both the objective and the LMI constraints are convex, a semi-definite program is a convex optimization problem.

$$\begin{aligned} \underset{x\in\mathbf{R}^n}{\text{minimize}} \quad & c^T x \\ \text{subject to} \quad & F(x) \;=\; F_0 + \sum_{i=1}^{n} x_i F_i \;\succeq\; 0, \\ & Ax = b, \end{aligned} \tag{2.24}$$

where $F_0, F_1, \ldots, F_n \in \mathbf{R}^{n\times n}$ and "$F(x) \succeq 0$" means that all eigenvalues of $F(x)$ are non-negative, that is $F(x)$ is positive semi-definite. If every F_i is diagonal, the LMI reduces to a set of scalar linear inequalities, and Problem (2.24) collapses to an ordinary linear program.

Why Semi-Definite Programming?

Several convex optimization frameworks exist for constrained spacecraft attitude control, including Quadratic Programming (QP), Second-Order Cone Programming (SOCP), and semi-definite Programming (SDP). The suitability of each depends on the problem's constraints and mathematical structure.

SDP stands out because it can directly handle the nonconvex quadratic constraints arising from quaternion-based attitude representations and nonlinear dynamics. Unlike QPs and SOCPs, which are limited to linear or second-order cone constraints, SDP uses Linear Matrix Inequalities (LMIs) to represent matrix-valued constraints. This allows nonconvex quadratic

constraints-such as those encoding keep-in/keep-out cones and quaternion normalization-to be convexified exactly within a single optimization problem.

Kim and Mesbahi [15] demonstrated that each nonconvex cone constraint $q^\top \widetilde{A} q \leq 0$, where $\widetilde{A}$ is indefinite, can be transformed into an equivalent LMI by adding a suitably chosen scalar $\mu > \lambda_{\max}(-\widetilde{A})$. This transformation renders each attitude cone constraint convex without conservatism, an exactness that QP and SOCP formulations cannot guarantee, as they require conservative approximations.

Modern SDP solvers, such as MOSEK and SeDuMi interfaced via CVX or YALMIP, efficiently solve medium-scale problems typical of missions in real time [33]. While QPs and SOCPs may be faster per solve, they cannot simultaneously enforce multiple keep-in/out cones, actuator saturation, angular rate bounds, and quaternion normalization within a single tractable formulation.

Hence, SDP offers a balanced solution combining:

- **Expressiveness:** exact LMI encoding of all quadratic attitude constraints,
- **Optimality:** single-shot global optimum over the entire maneuver,
- **Practicality:** real-time capable solvers for typical onboard problem sizes.

In summary, SDP's rich representational power aligns naturally with the mathematical structure of constrained spacecraft attitude control problems, enabling globally optimal, constraint-compliant solutions that alternative convex frameworks cannot systematically guarantee.

Chapter 3

PROBLEM FORMULATION AND METHODOLOGY

This chapter develops the control methodology and problem formulation used in this work. We first present the quaternion feedback regulator, a well-established baseline control law for spacecraft attitude dynamics, and detail its mathematical structure, gain selection, and implementation in simulation. We then extend this framework to incorporate operational constraints through a semidefinite programming (SDP) formulation, enabling systematic handling of keep-in/keep-out zones, actuator limits, and rate bounds. Finally, we integrate these components into a hybrid guidance and control architecture capable of achieving precise, constraint-compliant maneuvers.

3.1 Quaternion Feedback Regulators

We adopt the quaternion feedback regulator of Wie [36] for three-axis, large-angle reorientation maneuvers. The law combines *(i)* a proportional term based on the vector part of the error quaternion and *(ii)* a derivative term based on the body rates:

$$\mathbf{u} \;=\; -\,K\,\delta\mathbf{q}_v \;-\; D\,\boldsymbol{\omega}, \tag{3.1}$$

where

- $\delta\mathbf{q}_v \in \mathbb{R}^3$ is the vector part of the attitude-error quaternion,
- $\boldsymbol{\omega} \in \mathbb{R}^3$ is the body angular-rate vector,
- K and D are 3×3 constant gain matrices.

Wie showed that choosing the gains [36]

$$K = k_p\,J, \qquad D = k_d\,J, \qquad k_p > 0,\; k_d > 0, \tag{3.2}$$

With J the spacecraft inertia matrix, yields a globally stabilizing, eigen-axis rotation. Equation (3.1) thus plays the same role as a classical PD controller.

3.1.1 Selection of Gain Matrices

Wie *et al.* [36] relate the scalar gains k_p and k_d to the closed-loop settling time t_s, damping ratio ζ, and natural frequency ω_n. For a large-angle eigen-axis rotation, the modified settling time formula is

$$t_s = \frac{8}{\zeta \omega_n} \tag{3.3}$$

with natural frequency ω_n and damping ratio ζ satisfying

$$k_p = 2\omega_n^2, \qquad k_d = 2\zeta\omega_n. \tag{3.4}$$

For a $t_s = 20$ s settling time with critical damping ($\zeta = 1$) and using (3.3):

$$20 = \frac{8}{\omega_n} \quad \Longrightarrow \quad \omega_n = 0.40 \text{ rad/s}.$$

Substituting $\omega_n = 0.40$ rad/s into (3.4) yields

$$k_p = 2 \times (0.40)^2 = 0.32, \qquad k_d = 2 \times 1 \times 0.40 = 0.80. \tag{3.5}$$

These gains are subsequently scaled by the inertia matrix through $K = k_p J$ and $D = k_d J$, ensuring energy-consistent feedback torques for all three body axes.

3.1.2 Simulink Implementation of the PD Regulator

To validate the quaternion-feedback law, we built two Simulink models: one that drives to a fixed desired attitude, and one that continuously tracks a moving Earth-pointing target.

Model 1: Constant Final Attitude

Figure 3.1 shows the block diagram for the fixed-attitude case. Table 3.1 lists all initialization parameters.

Simulation Results Simulation results for the fixed-attitude case are shown in Figures 3.2-3.5. In Figure 3.2, the scalar component q_1 decays smoothly from 1 to 0.902 while the vector components $q_{2:4}$ ramp up to their setpoints, demonstrating overall attitude convergence. Figure 3.3 shows initial oscillations in ω_y and ω_z rapidly damped by the

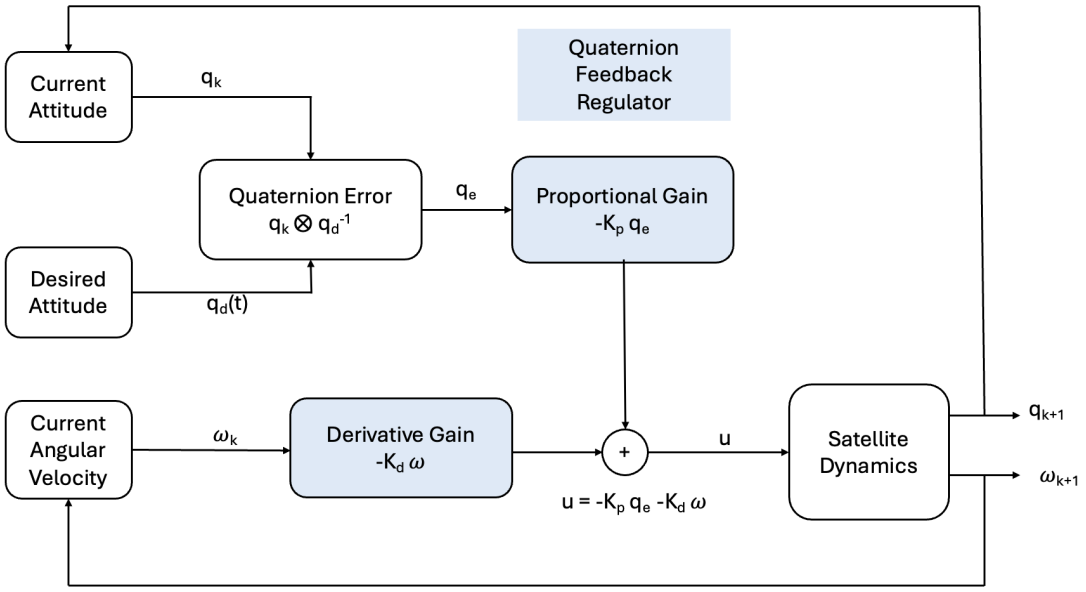


Figure 3.1: Closed-loop block diagram of the quaternion-feedback PD regulator for constant-attitude control.

Table 3.1: Simulation parameters: constant-attitude model

Parameter	Value
Sampling time, Δt	0.1 s
Inertia matrix, J	$\mathrm{diag}(0.0022,\ 0.0022,\ 0.0022)$ kg·m^2
PD gains	$K_P = 0.008,\ K_D = 0.004$
Initial quaternion, $q(0)$	$[1,\ 0,\ 0,\ 0]^T$
Initial angular rate, $\omega(0)$	$[0.003,\ 0.003,\ 0.003]^T$ rad/s
Desired quaternion, q_N	$[0.902,\ -0.001,\ -0.409,\ 0.136]^T$

derivative action, with all three rates settling quickly. The error quaternion in Figure 3.4 decays rapidly-its scalar part e_{q1} approaches unity and the vector parts $e_{q_{2:4}}$ drop to zero, confirming accurate tracking. Figure 3.5 illustrates that high initial torques correct the attitude error before decaying smoothly as the error vanishes. Finally, Figure 3.6 overlays actual quaternion trajectories (solid) and constant setpoints (dashed red), confirming that each component converges precisely to its desired value.

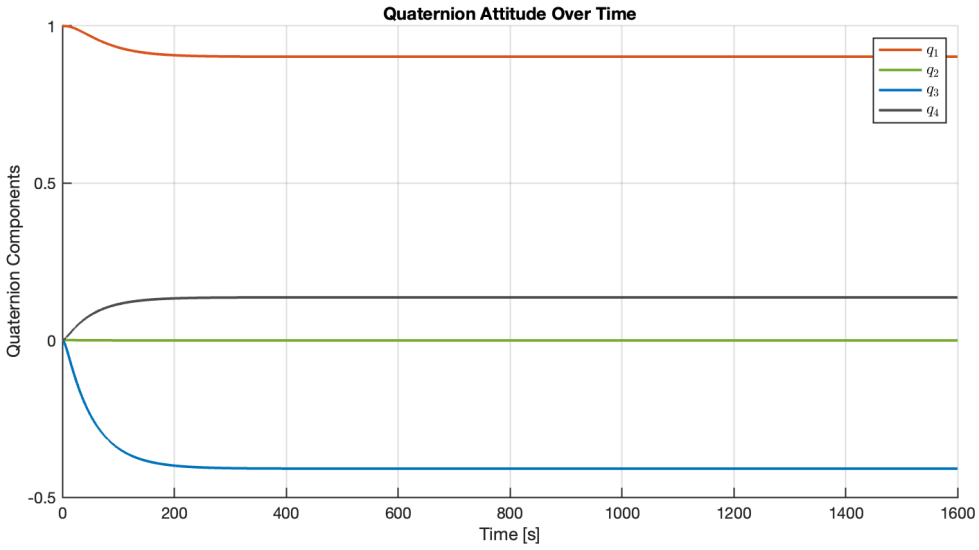


Figure 3.2: Quaternion components q_1, q_2, q_3, and q_4 versus time.

Model 2: Earth-Pointing Tracker

For a more realistic scenario, we replace the fixed q_N block with an Earth-pointing computation (Figure 3.7). At each step, the model computes the desired boresight quaternion $q_d(t)$ from the target's latitude, longitude, and altitude, here set to Seattle ($47.61°\text{N}$, $-122.33°\text{W}$, $57\,\text{m}$).

We will see that even as the desired attitude $q_d(t)$ moves to keep the satellite pointed at the desired location, the PD controller maintains a small tracking error and produces smooth torque profiles, demonstrating robustness to time-varying setpoints.

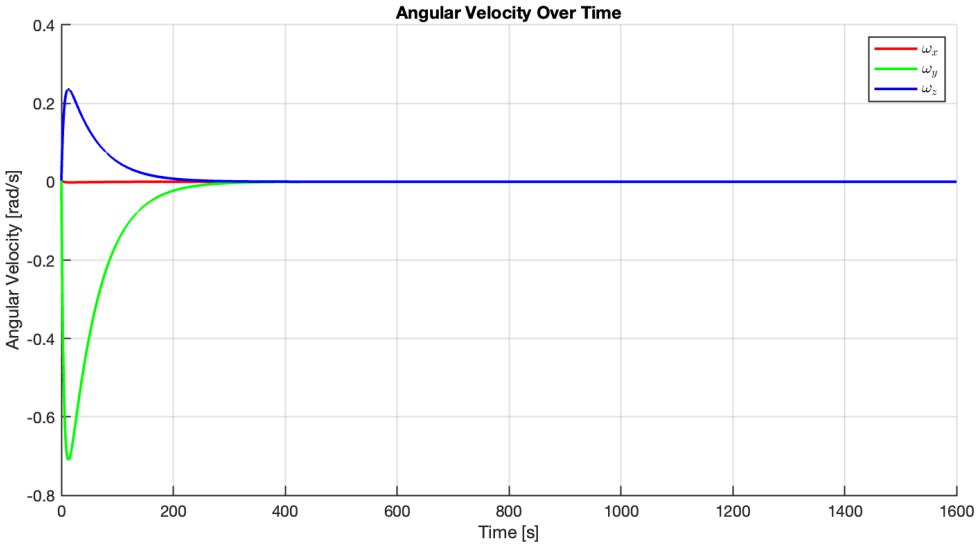


Figure 3.3: Angular-velocity components ω_x, ω_y, and ω_z versus time.

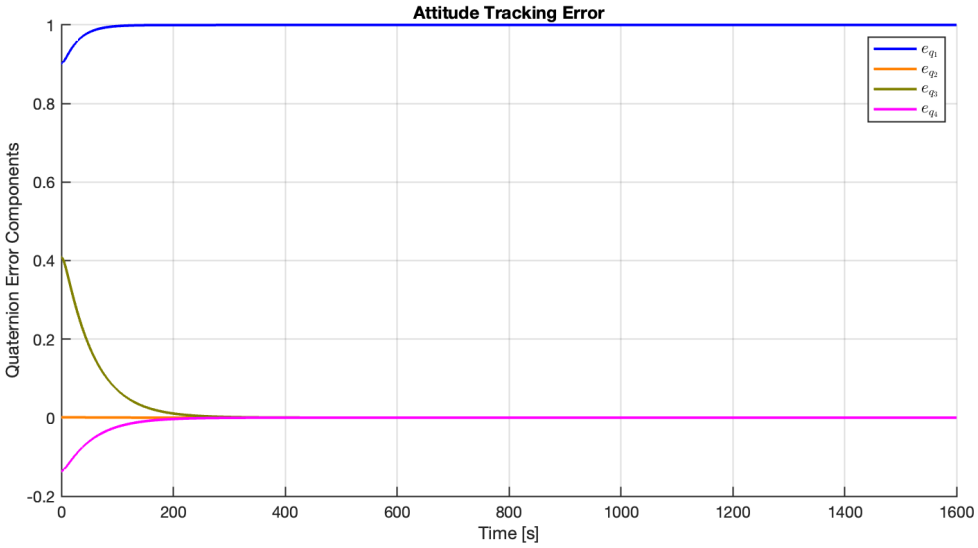


Figure 3.4: Error-quaternion components e_{q1}, e_{q2}, e_{q3}, and e_{q4} versus time.

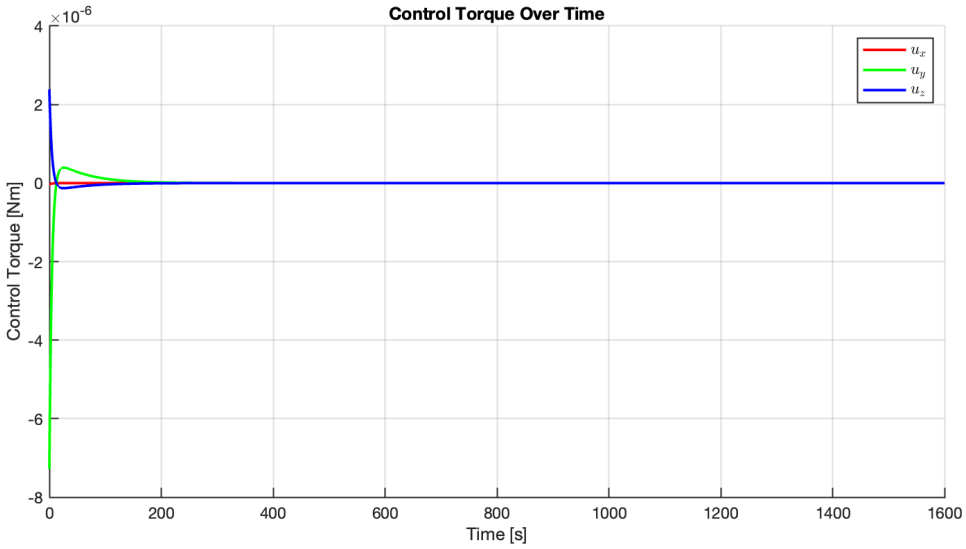


Figure 3.5: Control-torque components u_x, u_y, and u_z versus time.

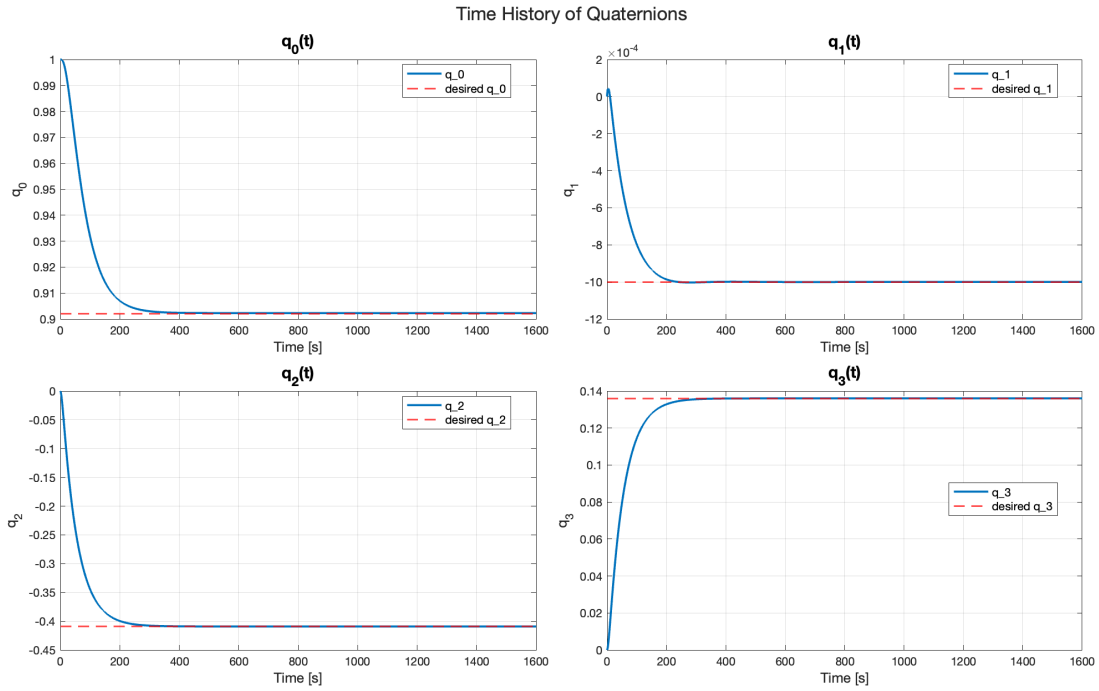


Figure 3.6: Time history of quaternion components with constant desired setpoints (dashed red).

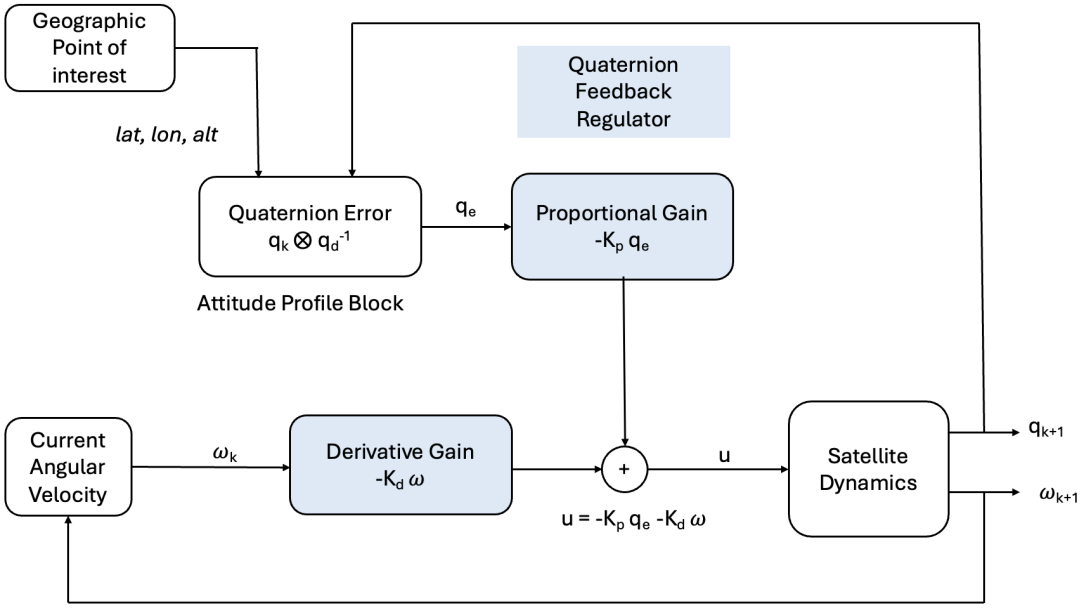


Figure 3.7: Simulink block diagram of the Earth-pointing PD regulator.

Table 3.2: Simulation parameters: Earth-pointing model

Parameter	**Value**
Sampling time, Δt	0.1 s
Inertia matrix, J	$\mathrm{diag}(0.0022,\ 0.0022,\ 0.0022)\ \mathrm{kg\,m^2}$
PD gains	$K_P = 0.008,\ K_D = 0.004$
Initial quaternion, $q(0)$	$[1,\ 0,\ 0,\ 0]^T$
Initial angular rate, $\omega(0)$	$[0.003,\ 0.003,\ 0.003]^T\ \mathrm{rad/s}$
Geographic POI	$(47.61°\mathrm{N},\ -122.33°\mathrm{W},\ 57\,\mathrm{m})$
Desired attitude	Earth-pointing quaternion $q_d(t)$

Simulation Results Simulation results for the tracking case are shown in Figures 3.8–3.12. In Figure 3.8, each quaternion component converges smoothly from its initial value to the time-varying target $q_d(t)$ computed by the Earth-pointing routine. Figure 3.9 shows the body rates exhibiting small initial transients as the controller locks onto the moving setpoint, then settling close to the commanded profile without overshoot. In Figure 3.10, the error quaternion $q_e = q \otimes q_d^{-1}$ remains small, with its scalar part near one and vector parts near zero throughout the maneuver. Figure 3.11 illustrates that torque commands update continuously to follow the desired attitude while staying within actuator limits. Finally, Figure 3.12 overlays actual quaternion components (solid) and the computed setpoints (dashed), demonstrating accurate tracking of the reference profile.

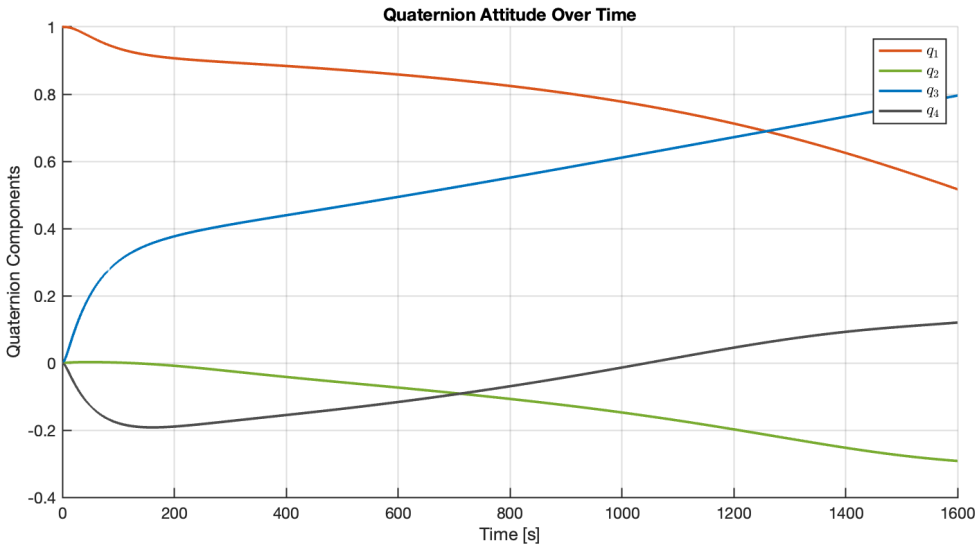


Figure 3.8: Quaternion components q_1-q_4 versus time under Earth-pointing PD control.

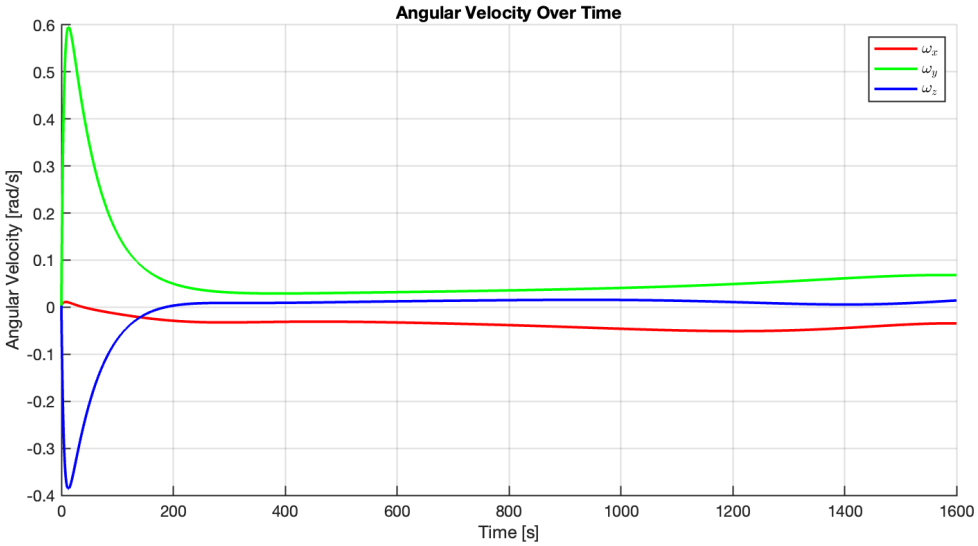


Figure 3.9: Angular-velocity components ω_x, ω_y, and ω_z versus time.

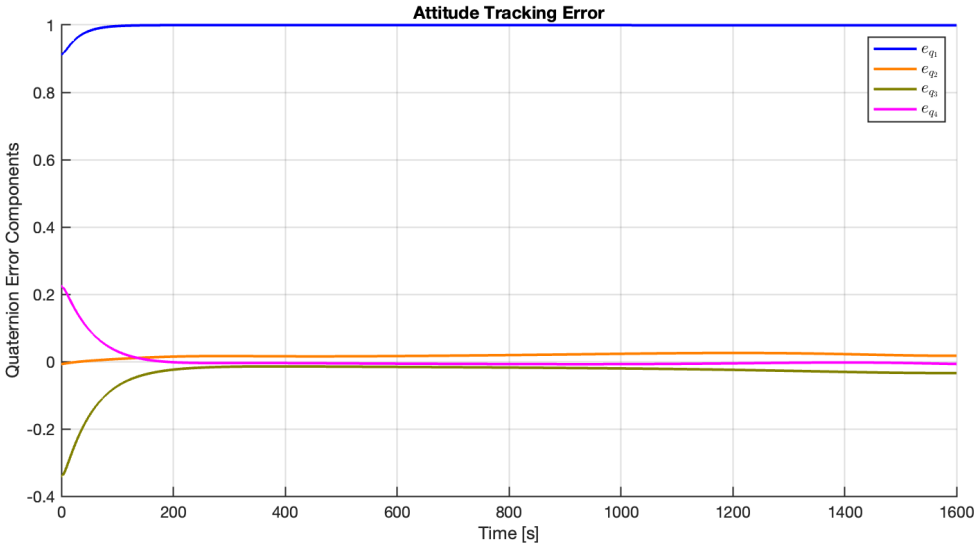


Figure 3.10: Error-quaternion components e_{q1}-e_{q4} versus time.

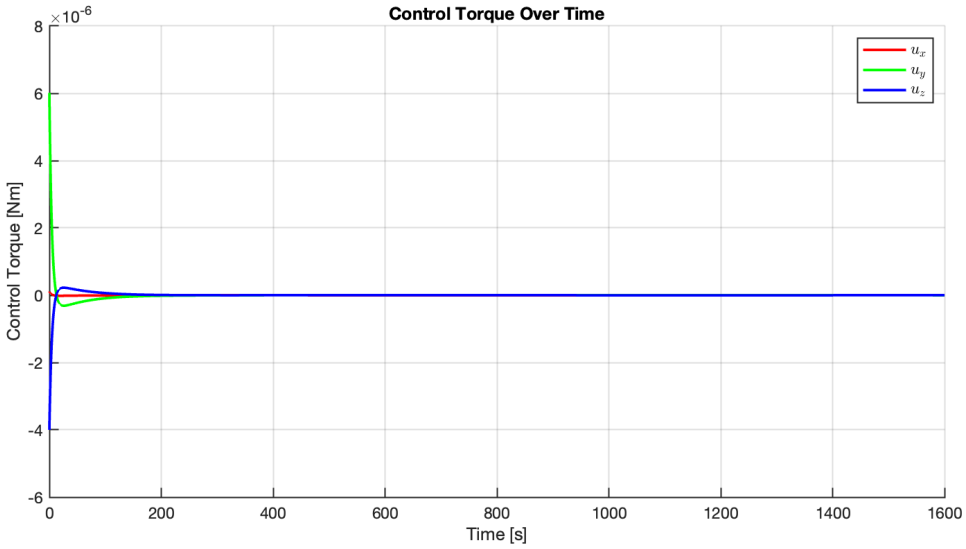


Figure 3.11: Control-torque components u_x, u_y, and u_z versus time.

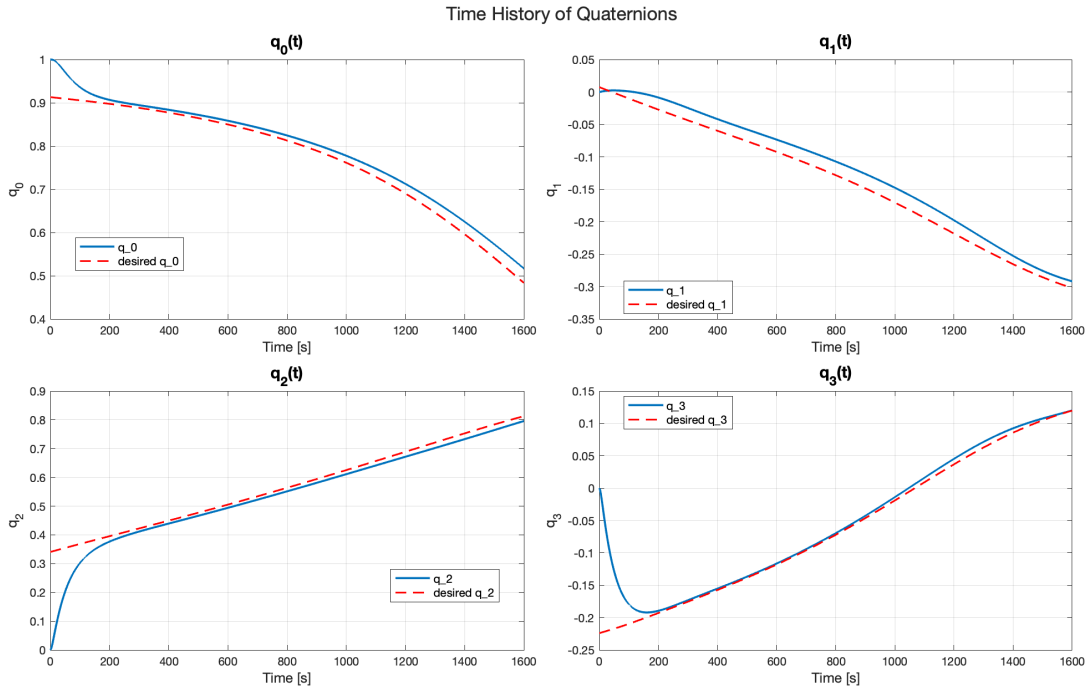


Figure 3.12: Time history of quaternion components with dynamic desired setpoints (dashed red).

3.2 SDP Formulation for Constrained Attitude Control

Following the convexification technique of Kim et al. [16], we exploit the spacecraft dynamics' inherent nonlinearities and the structured nonconvexity of the attitude constraints to recast the original Spacecraft Reorientation with Constraints (SRPC) problem as a convex semi-definite program.

Non-convex optimization can suffer from multiple local minima, a lack of global optimality guarantees, and high computational cost. To avoid these pitfalls, we adopt the convexification strategy developed by Kim et al. [15, 16], which converts the original non-convex SRPC formulation into an equivalent LMI-based semi-definite Program.

3.2.1 Decision Variable

At each discrete time step k, we solve for a decision vector

$$x(k) = \begin{bmatrix} u(k) \\ \omega(k+1) \\ q(k+2) \end{bmatrix} \in \mathbf{R}^{10}, \tag{3.6}$$

containing the control torque $u(k) \in \mathbf{R}^3$, the post-control body-rate $\omega(k+1) \in \mathbf{R}^3$, and the quaternion $q(k+2) \in \mathbf{R}^4$, which are the optimal body-rate and quaternion after the optimal control torque is applied.

The optimization minimizes a slack parameter $\alpha(k)$ that steers the current state toward the desired terminal angular velocity and quaternion while enforcing the following constraints. It represents the minimum bound of the error.

3.2.2 Keep-Out and Keep-In Constraint

Keep-Out constraint The Keep-out constraint, also referred to as the *forbidden region constraint*, ensures that sensitive optical instruments onboard the spacecraft avoid direct exposure to the bright celestial objects (e.g., the Sun) which could cause significant damage to the instruments. This is expressed mathematically, by defining a boresight vector $\boldsymbol{v_b} \in \mathbf{R}^3$

fixed in the spacecraft body frame and an inertially-fixed forbidden direction vector $\boldsymbol{w} \in \mathbf{R}^3$, representing, for instance, the direction of the sun.

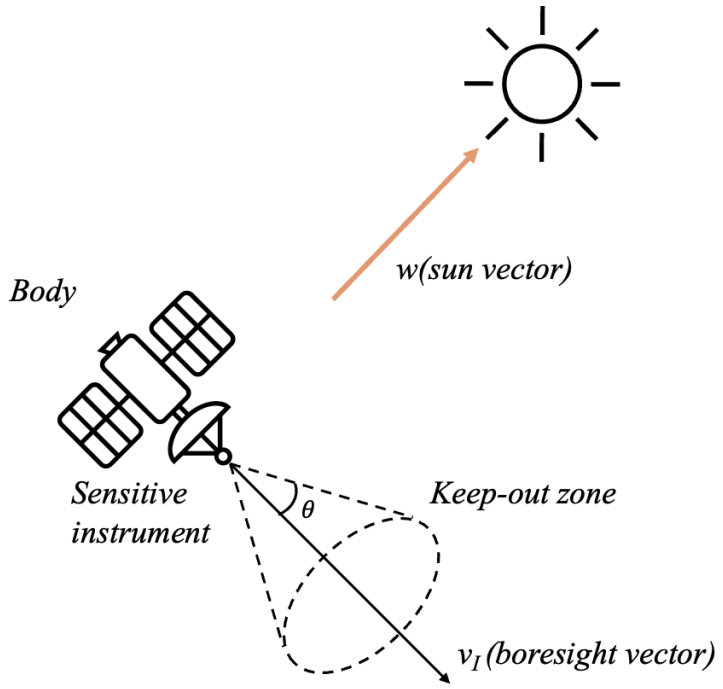


Figure 3.13: Geometry of the sun-avoidance keep-out cone

Mathematical formulation For a fixed body-frame boresight $v_b \in \mathbf{R}^3$ and an inertial keep-out direction $w \in \mathbf{R}^3$, the keep-out constraint requires a minimum angular separation θ between these vectors:

$$v_I(t)^T w \ \leq \ \cos\theta, \tag{3.7}$$

where $v_I(t)$ denotes the v_b vector expressed in inertial coordinates at time t. The quaternion representation, $q(t) = [\boldsymbol{q_o}(t), q_4(t)]^T$, facilitates transforming vectors from the spacecraft body frame to inertial frame:

$$v_I(t) = v_b - 2(\boldsymbol{q}_o^T(t)\boldsymbol{q}_o(t))v_b + 2(\boldsymbol{q}_o(t)^T v_b)\boldsymbol{q}_o(t) + 2q_1(t)(v_b \times \boldsymbol{q}_o(t)) \tag{3.8}$$

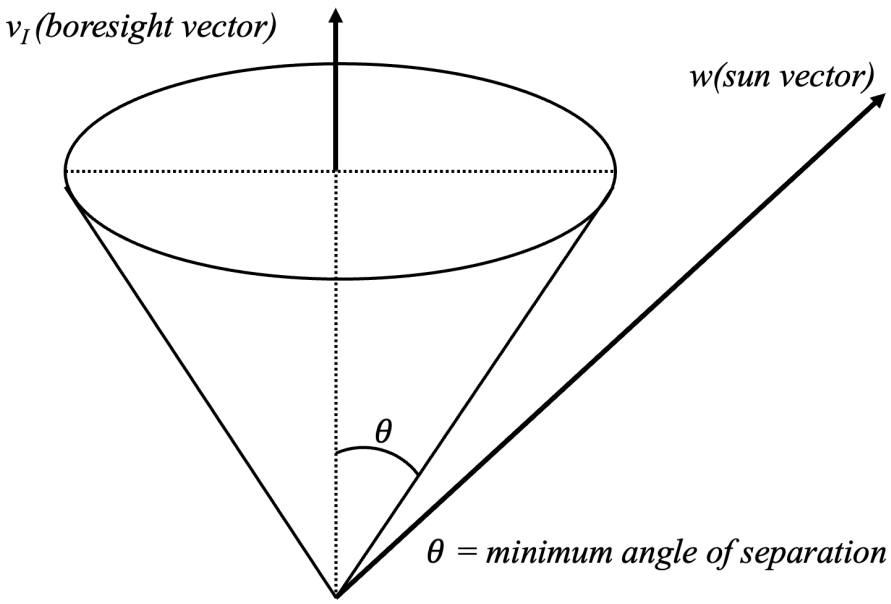


Figure 3.14: Exclusion cone in the spacecraft body frame.

Plugging (3.8) into (3.7) , we derive the following quadratic inequality:

$$[v_b - 2(\boldsymbol{q}_o^T \boldsymbol{q}_o)v_b + 2(\boldsymbol{q}_o^T v_b)\boldsymbol{q}_o + 2q_4(v_b \times \boldsymbol{q}_o)]^T w \leq \cos\theta \tag{3.9}$$

$$\implies v_b^T w - 2(\boldsymbol{q}_o^T \boldsymbol{q}_o)v_b^T w + 2(\boldsymbol{q}_o^T v_b)\boldsymbol{q}_o^T w + 2q_4(v_b \times \boldsymbol{q}_o)^T w - \cos\theta \leq 0 \tag{3.10}$$

Reorganising the terms to get the quadratic in $\boldsymbol{q}_o$term, the cross term, and the scalar term,

$$\underbrace{-2(\boldsymbol{q}_o^T \boldsymbol{q}_o)v_b^T w + 2(\boldsymbol{q}_o^T v_b)\boldsymbol{q}_o^T w}_{\textbf{Quadratic term}} + \underbrace{2q_1(v_b \times \boldsymbol{q}_o)^T w}_{\textbf{cross term}} + \underbrace{v_b^T w - \cos\theta}_{\textbf{scalar term}} \leq 0 \tag{3.11}$$

Matrix Representation The inequality in (3.11) can be expressed in matrix form as:

$$\boldsymbol{q}(k)^T \tilde{A} \boldsymbol{q}(k) \leq 0, \tag{3.12}$$

where

$$\tilde{A} = \begin{bmatrix} A & b \\ b^T & d \end{bmatrix}, \tag{3.13}$$

and

$$A := v_b w^T + w v_b^T - (v_b^T w + \cos\theta) I_3, \tag{3.14}$$

$$b := w \times v_b, \tag{3.15}$$

$$d := v_b^T w - \cos\theta. \tag{3.16}$$

Introducing the transformation matrix, defined as:

$$H = \begin{bmatrix} 0_{4\times 6} & I_4 \end{bmatrix}, \tag{3.17}$$

which isolates the quaternion states $q(k+2)$ from the decision variable $x(k)$:

$$x(k)^T \{H^T \tilde{A} H\} x(k) \leq 0. \tag{3.18}$$

The term μ_i is introduced to ensure that $\tilde{A}$ is positive definite by shifting the eigenvalues to be non-negative. This approach was proposed in [16] as a proposition. It shows that we can shift $\tilde{A}$ by a scalar μ as a convex parameterization:

$$\tilde{A}_\mu := \tilde{A} + \mu I_4 \qquad \mu > \lambda_{max}(-\tilde{A}),$$

so that $\tilde{A}_\mu \succ 0$.

This convexification approach, proposed by Kim *et al.* [16], leads to a Linear Matrix Inequality (LMI) representation:

$$\begin{bmatrix} \mu & (Hx(k))^T \\ Hx(k) & (\mu I_4 + \tilde{A})^{-1} \end{bmatrix} \geq 0. \tag{3.19}$$

Keep-In Constraint An Equivalent process can also be applied to the keep-in constraint, also known as *Mandatory region*, which is presented in Tam *et al.* [29] We know the

$$v_I(t)^T w \;\geq\; \cos\theta, \tag{3.20}$$

The angle between the 2 vectors should be within the half-angle θ of the inclusion cone of the instrument. Due to the sign change, the quadratic constraint becomes:

$$\boldsymbol{q}(k)^T (-\tilde{A}) \boldsymbol{q}(k) \leq 0, \tag{3.21}$$

The corresponding LMI becomes:

$$\begin{bmatrix} \mu & (Hx)^T \\ Hx & (\mu I_4 - \tilde{A})^{-1} \end{bmatrix} \geq 0, \tag{3.22}$$

with $\mu > \lambda_{max}(-\tilde{A})$ to ensure positive definiteness.

3.2.3 Angular Velocity Constraint

The spacecraft's angular velocity is restricted to a few degrees per second to maintain stable orientation, ensure mission objectives, and prevent overloading actuators such as reaction wheels or control moment gyros (CMGs). This constraint can be mathematically expressed as:

$$|\omega_i(k)| \leq \gamma_1, \qquad i = 1, 2, 3 \tag{3.23}$$

where $\boldsymbol{\omega}(k) = [\omega_1(k), \omega_2(k), \omega_3(k)]^T$ is the angular velocity vector at time step $k \in [t_0, t_f]$, and $\gamma_1 > 0$ denotes the mission-defined maximum allowable angular velocity (assumed constant).

To incorporate this constraint into the optimization framework, we rewrite (3.23) as follows:

$$|G_1 x(k)| \leq \gamma_1 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \tag{3.24}$$

where $x(k)$ is the decision variable defined in Section 3.2.1, and G_1 is a matrix used to extract the angular velocity components:

$$G_1 = \begin{bmatrix} 0_{3\times 3} & I_3 & 0_{3\times 4} \end{bmatrix}. \tag{3.25}$$

This constraint effectively prevents excessive rotational rates that could compromise the spacecraft's stability and operational integrity during the mission.

3.2.4 Torque Bound

Reaction wheels on a 1U CubeSat typically supply a maximum torque of approximately $2 \times 10^{-4}\, N \cdot m$. To ensure the commanded torques remain within the physical limits of the actuators, the following element-wise bound is imposed:

$$|u_i(k)| \leq \gamma_2, \qquad i = 1, 2, 3, \tag{3.26}$$

where $\mathbf{u}(k) = [u_1(k), u_2(k), u_3(k)]^T$ is the control torque vector at time step k, and $\gamma_2 > 0$ denotes the maximum allowable torque (assumed constant).

This constraint is embedded in the optimization problem using the matrix inequality:

$$|G_2 x(k)| \leq \gamma_2 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \tag{3.27}$$

where $x(k)$ is the decision variable defined in Section 3.2.1, and the selection matrix G_2 is given by:

$$G_2 = \begin{bmatrix} I_3 & 0_{3\times 7} \end{bmatrix}, \tag{3.28}$$

which isolates the torque components from $x(k)$.

Typical CubeSat actuators generate torques on the order of $10^{-6}\, N \cdot m$, while disturbance torques in Low Earth Orbit are typically around $5 \times 10^{-7}\, N \cdot m$. Reaction wheels can provide control torques up to approximately $10^{-3}\, N \cdot m$, which is sufficient for most attitude adjustment maneuvers.

This constraint ensures the optimization respects actuator limitations and prevents hardware overload, contributing to safe and reliable spacecraft operation.

3.2.5 Desired State Constraint

In [15], it is explained that the primary objective is to minimize the state errors, specifically the angular velocity and quaternion errors, between the current and desired states. This objective can be mathematically expressed as:

$$\|\omega(k+1) - \omega(N)\|^2 + \|C\left(q(k+1) \otimes q(N)^{-1}\right)\|^2 \tag{3.29}$$

where

$$C := \begin{bmatrix} I_3 & 0_{3\times 1} \end{bmatrix}, \tag{3.30}$$

with body angular velocity vector $\omega(k) = \begin{bmatrix} \omega_1(k), \omega_2(k), \omega_3(k) \end{bmatrix}^T$ and unit quaternion $q(k) = \begin{bmatrix} q_1(k), q_2(k), q_3(k), q_4(k) \end{bmatrix}^T$

The Hamilton product $q(k+1) \otimes q(N)^{-1}$ gives the attitude-error quaternion; the selector C extracts its vector part, so the second term penalises attitude misalignment, while the first term penalises residual angular-rate error. Equation (3.29) therefore measures the total state error as the sum of squared 2-norms of these two quantities.

We have already seen the Quaternion error computation in Equation [2.13]. The formulation adapted in the [15] is shown below, and we use the [20] to make the connection:

$$\begin{aligned} \delta q(k) &= q(k+1) \otimes q(N)^{-1} \\ &= q(N)^{-1} \odot q(k+1) \\ &= [q(N)^{-1} \odot] q(k+1) \\ &= [\Xi(q(N)^{-1}) q(N)^{-1}] q(k+1) \end{aligned} \tag{3.31}$$

because

$$[q\odot] = [\Xi(q)q]$$

where the inverse of the desired quaternion is

$$q(N)^{-1} = \begin{bmatrix} -q_1(N) \\ -q_2(N) \\ -q_3(N) \\ q_4(N) \end{bmatrix}$$

with $\Xi(q)$ being a 4×3 matrix

$$\Xi(q) = \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix}$$

therefore,

$$\Xi(q(N)^{-1}) = \begin{bmatrix} q_4(N) & q_3(N) & -q_2(N) \\ -q_3(N) & q_4(N) & q_1(N) \\ q_2(N) & -q_1(N) & q_4(N) \\ q_1(N) & q_2(N) & q_3(N) \end{bmatrix}$$

which leads to,

$$[\Xi(q(N)^{-1})q(N)^{-1}] = \begin{bmatrix} q_4(N) & q_3(N) & -q_2(N) & -q_1(N) \\ -q_3(N) & q_4(N) & q_1(N) & -q_2(N) \\ q_2(N) & -q_1(N) & q_4(N) & -q_3(N) \\ q_1(N) & q_2(N) & q_3(N) & q_4(N) \end{bmatrix}$$

Therefore 3.31 becomes :

$$\delta q(k) = \begin{bmatrix} q_4(N) & q_3(N) & -q_2(N) & -q_1(N) \\ -q_3(N) & q_4(N) & q_1(N) & -q_2(N) \\ q_2(N) & -q_1(N) & q_4(N) & -q_3(N) \\ q_1(N) & q_2(N) & q_3(N) & q_4(N) \end{bmatrix} \begin{bmatrix} q_1(k+1) \\ q_2(k+1) \\ q_3(k+1) \\ q_4(k+1) \end{bmatrix} \tag{3.32}$$

$$\begin{aligned} \delta q(k) &= q(k+1) \otimes q(N)^{-1} \\ &= Q(N)q(k+1) \end{aligned}$$

This is the matrix form of the quaternion error. When we add the C matrix to this and take

the square of the 2-norm, we get,

$$\begin{aligned}\left\|C\,\delta q(k)\right\|_2^2 &= \left\|C\,Q(N)\,q(k+1)\right\|_2^2 \\ &= \left\langle C\,Q(N)\,q(k+1),\ C\,Q(N)\,q(k+1)\right\rangle \\ &= \left(C\,Q(N)\,q(k+1)\right)^\top C\,Q(N)\,q(k+1) \\ &= q(k+1)^\top\,Q(N)^\top C^\top C\,Q(N)\,q(k+1).\end{aligned}$$

The Equation 3.29 becomes

$$\begin{bmatrix} x(k) \\ 1 \end{bmatrix}^T E(k) \begin{bmatrix} x(k) \\ 1 \end{bmatrix} \geq \alpha \tag{3.33}$$

We adopt this LMI from (Kim.*et al*) that captures the angular rate error and the quaternion error and drives it to zero:

$$\begin{bmatrix} \alpha & \left(\mathbf{E}(k)^{1/2} \begin{bmatrix} \mathbf{x}(k) \\ 1 \end{bmatrix}\right)^T \\ \mathbf{E}(k)^{1/2} \begin{bmatrix} \mathbf{x}(k) \\ 1 \end{bmatrix} & \mathbf{I}_{11} \end{bmatrix} \geq 0 \tag{3.34}$$

where $E(k)$ is built from the desired quaternion $q(N)$ and current angular-velocity $\omega(N)$ terms:

$$E(k) = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & 0_{3\times4} & 0_{3\times1} \\ 0_{3\times3} & I_3 & 0_{3\times4} & -\omega(N) \\ 0_{4\times3} & 0_{4\times3} & 0_{4\times4} & 0_{4\times1} \\ 0_{1\times3} & -\omega^T(N) & 0_{1\times4} & \omega(N)^T\omega(N) \end{bmatrix} + \begin{bmatrix} 0_{6\times6} & 0_{6\times4} & 0_{6\times1} \\ 0_{4\times6} & Q(N)^T C^T C Q(N) & 0_{4\times1} \\ 0_{1\times6} & 0_{1\times4} & 0_{1\times1} \end{bmatrix} \tag{3.35}$$

$$\mathbf{Q(N)} = \begin{bmatrix} q_4(N) & q_3(N) & -q_2(N) & -q_1(N) \\ -q_3(N) & q_4(N) & q_1(N) & -q_2(N) \\ q_2(N) & -q_1(N) & q_4(N) & -q_3(N) \\ q_1(N) & q_2(N) & q_3(N) & q_4(N) \end{bmatrix} \tag{3.36}$$

Minimizing $\alpha(k)$ forces the quaternion error to zero.

3.2.6 Discrete Dynamics and Kinematic Constraints

The rotational dynamics of the spacecraft are governed by Euler's equations, which relate angular velocities $\omega_i(t)$ to applied control torques $u_i(t)$:

$$J_1\,\dot{\omega}_1(t) - (J_2 - J_3)\,\omega_2(t)\,\omega_3(t) = u_1(t), \tag{3.37}$$

$$J_2\,\dot{\omega}_2(t) - (J_3 - J_1)\,\omega_3(t)\,\omega_1(t) = u_2(t), \tag{3.38}$$

$$J_3\,\dot{\omega}_3(t) - (J_1 - J_2)\,\omega_1(t)\,\omega_2(t) = u_3(t), \tag{3.39}$$

where J_i denotes the principal moment of inertia of the spacecraft about the i-th principal axis for $i = 1, 2, 3$.

To discretize Euler's equations (3.39), we employ the forward Euler approximation, a straightforward numerical method for approximating continuous-time derivatives using finite differences. This first-order method converts continuous-time equations into discrete-time update equations, making numerical implementation straightforward and efficient.

Specifically, the angular velocity derivative at a discrete time step k is approximated by:

$$\dot{\omega}(t) \approx \frac{\omega(k+1) - \omega(k)}{\Delta t}, \tag{3.40}$$

where Δt denotes the discrete sampling interval.

Applying this discretization to Euler's equations (3.39) yields the discrete-time dynamic constraints:

$$J_1\,\frac{\omega_1(k+1) - \omega_1(k)}{\Delta t} - (J_2 - J_3)\,\omega_2(k)\,\omega_3(k) = u_1(k), \tag{3.41}$$

$$J_2\,\frac{\omega_2(k+1) - \omega_2(k)}{\Delta t} - (J_3 - J_1)\,\omega_3(k)\,\omega_1(k) = u_2(k), \tag{3.42}$$

$$J_3\,\frac{\omega_3(k+1) - \omega_3(k)}{\Delta t} - (J_1 - J_2)\,\omega_1(k)\,\omega_2(k) = u_3(k). \tag{3.43}$$

These equations can be rearranged into the explicit discrete-time update rules for angular

velocities:

$$J_1\,\omega_1(k+1) = J_1\,\omega_1(k) + \Delta t\,[u_1(k) + (J_2 - J_3)\,\omega_2(k)\,\omega_3(k)]\,, \tag{3.44}$$

$$J_2\,\omega_2(k+1) = J_2\,\omega_2(k) + \Delta t\,[u_2(k) + (J_3 - J_1)\,\omega_3(k)\,\omega_1(k)]\,, \tag{3.45}$$

$$J_3\,\omega_3(k+1) = J_3\,\omega_3(k) + \Delta t\,[u_3(k) + (J_1 - J_2)\,\omega_1(k)\,\omega_2(k)]\,. \tag{3.46}$$

These updated equations form a recursive scheme for numerically propagating angular velocities ω_i through each discrete time step k, approximating the continuous-time dynamics. The accuracy of this numerical approximation depends on the choice of sampling interval Δt; smaller Δt values improve accuracy but increase computational requirements.

To ensure valid quaternion representations during numerical integration, we enforce a norm-preserving kinematic constraint, given by the kinematic equation from [20]

$$\dot{\mathbf{q}}(t) = \frac{1}{2}\boldsymbol{\omega}(t) \otimes \mathbf{q}(t) \tag{3.47}$$

$$= \frac{1}{2}\Omega(t)\,\mathbf{q}(t) \tag{3.48}$$

$$= \frac{1}{2}\Xi(t)\,\boldsymbol{\omega}(t) \tag{3.49}$$

where $\Omega(t)$ is defined as:

$$\Omega(t) = \begin{bmatrix} 0 & \omega_3(t) & -\omega_2(t) & \omega_1(t) \\ -\omega_3(t) & 0 & \omega_1(t) & \omega_2(t) \\ \omega_2(t) & -\omega_1(t) & 0 & \omega_3(t) \\ -\omega_1(t) & -\omega_2(t) & -\omega_3(t) & 0 \end{bmatrix}. \tag{3.50}$$

and

$$\Xi(t) = \begin{bmatrix} -q_4(t) & q_3(t) & -q_2(t) \\ -q_3(t) & -q_4(t) & q_1(t) \\ q_2(t) & -q_1(t) & -q_4(t) \\ q_1(t) & q_2(t) & q_3(t) \end{bmatrix} \tag{3.51}$$

This constraint ensures that the quaternion remains of unit norm, i.e., $||\mathbf{q}(t)|| = 1$, for all $t \in [t_0, t_f]$.

Applying the forward Euler method to discretize equation (3.48), we obtain:

$$\frac{\mathbf{q}(k+1)-\mathbf{q}(k)}{\Delta t} = \frac{1}{2}\Omega(k)\,\mathbf{q}(k), \tag{3.52}$$

$$\mathbf{q}(k+1) = \mathbf{q}(k) + \frac{\Delta t}{2}\Omega(k)\,\mathbf{q}(k), \tag{3.53}$$

$$\mathbf{q}(k+1) = \left[\mathbf{I}_4 + \frac{\Delta t}{2}\Omega(k)\right]\mathbf{q}(k). \tag{3.54}$$

Here, $\mathbf{I}_4$ is the 4×4 identity matrix. We use the first-order Taylor Expansion to represent Equation (3.54) in the exponential form as:

$$\mathbf{q}(k+1) \approx e^{\frac{\Delta t}{2}\Omega(k)}\,\mathbf{q}(k). \tag{3.55}$$

We use the Equation (3.55) to calculate $q(k+1)$ since we require that to solve the semi-definite programming algorithm at each timestep.

The simpler first-order approximation from equation (3.54) is sufficient, provided the discrete time step Δt remains small enough to ensure numerical stability and accuracy.

These kinematic constraints are enforced explicitly as equality constraints in the optimization formulation, guaranteeing a physically feasible quaternion trajectory at each discrete time step k.

Equation [3.46,3.54] can be used to form the constraint for the optimization problem:

$$\mathbf{F}(k)x(k) = y(k) \tag{3.56}$$

where

$$y(k) = \begin{bmatrix} J_1\omega_1(k) + \Delta t(J_2 - J_3)\omega_2(k)\omega_3(k) \\ J_2\omega_2(k) + \Delta t(J_3 - J_1)\omega_3(k)\omega_1(k) \\ J_3\omega_3(k) + \Delta t(J_1 - J_2)\omega_1(k)\omega_2(k) \\ q(k+1) \end{bmatrix} \tag{3.57}$$

$$\mathbf{F}(k) = \begin{bmatrix} -\Delta t I_3 & J & 0_{3\times 4} \\ 0_{4\times 3} & R(k+1) & I_4 \end{bmatrix} \tag{3.58}$$

and

$$J := \begin{bmatrix} J_1 & 0 & 0 \\ 0 & J_2 & 0 \\ 0 & 0 & J_3 \end{bmatrix} \tag{3.59}$$

$$R(k+1) := \frac{\Delta t}{2} \begin{bmatrix} -q_4(k+1) & q_3(k+1) & -q_2(k+1) \\ -q_3(k+1) & -q_4(k+1) & q_1(k+1) \\ q_2(k+1) & -q_1(k+1) & -q_4(k+1) \\ q_1(k+1) & q_2(k+1) & q_3(k+1) \end{bmatrix} \tag{3.60}$$

We use the Equation (3.49) to solve the optimization problem at each step. This is done because once we calculate the $q(k+1)$, we require the $\omega(k+1)$ as well to solve for $q(k+2)$, but since $\omega(k+1)$ is part of the decision variable, we can't use that inside the optimization, so instead we utilize the kinematic equation of the form

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

Discretising with a forward-Euler step

$$\frac{q(k+2) - q(k+1)}{\Delta t} = \frac{1}{2}\Xi(k+1)\,\boldsymbol{\omega}(k+1)$$

$$q(k+2) = \frac{\Delta t}{2}\Xi(k+1)\,\boldsymbol{\omega}(k+1) + q(k+1)$$

where

$$\Xi(k+1) = \begin{bmatrix} -q_4(k+1) & q_3(k+1) & -q_2(k+1) \\ -q_3(k+1) & -q_4(k+1) & q_1(k+1) \\ q_2(k+1) & -q_1(k+1) & -q_4(k+1) \\ q_1(k+1) & q_2(k+1) & q_3(k+1) \end{bmatrix}$$

3.2.7 Cost Function

We minimize a scalar parameter α and control a set of slack variables $x(k)$, where k is the time step.

$$\underset{x(k)}{\text{minimize}} \;\; \alpha \tag{3.61}$$

The role of the parameter α is to steer the spacecraft toward the desired terminal orientation $q(N)$ and angular velocity $\omega(N)$.

3.2.8 The Constraint Attitude Guidance Problem Statement

The CAG problem is to compute a control torque trajectory, **u(k)**, that autonomously reorients the spacecraft such that a specified attitude is attained. The pointing vector in the body frame and the inertial vector to the target are determined by the mission mode of the spacecraft.

At each time step, the semi-definite program (SDP) optimizes the decision vector $x(k)$, which includes the control torque $u(k)$, the post-control angular velocity $\omega(k+1)$, and the updated quaternion $q(k+2)$. By minimizing the slack variable α, the algorithm steers the spacecraft's attitude and angular velocity toward the desired terminal state, balancing tracking accuracy and constraint satisfaction.

The problem constraints are formulated as Linear Matrix Inequalities (LMIs), ensuring convexity and enabling efficient solution via interior-point methods. These constraints enforce spacecraft dynamics, actuator limits, angular velocity bounds, and attitude pointing requirements, all within a computationally tractable framework suitable for real-time implementation.

The linear inequality constraints also help maintain the quaternion norm near unity for small integration steps, preserving the physical feasibility and stability of the attitude representation throughout the maneuver.

Compared to classical nonlinear or heuristic methods, this SDP-based approach guarantees global optimality within the convexified problem formulation, offering robustness and certifiable constraint satisfaction for spacecraft attitude guidance and control.

SDP solved at each time step k

$$\min_{x(k)} \ \alpha$$

$$\text{s.t.} \quad \begin{bmatrix} \mu & (Hx(k))^{\mathsf{T}} \\ Hx(k) & \mu I_4 + \tilde{A} \end{bmatrix} \succeq 0 \quad \text{(Pointing)}$$

$$\begin{bmatrix} \alpha & \left(E^{1/2} \begin{bmatrix} x(k) \\ 1 \end{bmatrix}\right)^{\mathsf{T}} \\ E^{1/2} \begin{bmatrix} x(k) \\ 1 \end{bmatrix} & I_{11} \end{bmatrix} \succeq 0 \quad \text{(Tracking)}$$

$$F(k)\, x(k) = y(k) \quad \text{(Dynamics \& Kinematics)}$$

$$|G_1 x(k)| \leq \gamma_1 \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^{\mathsf{T}} \quad \text{(Body-Rate)}$$

$$|G_2 x(k)| \leq \gamma_2 \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^{\mathsf{T}} \quad \text{(Control Torque)}$$

where

$$x(k) = \begin{bmatrix} u(k) \\ \omega(k{+}1) \\ q(k{+}2) \end{bmatrix}, \quad G_1 = \begin{bmatrix} 0_{3\times 3} & I_3 & 0_{3\times 4} \end{bmatrix}, \quad G_2 = \begin{bmatrix} I_3 & 0_{3\times 7} \end{bmatrix},$$

$$y(k) = \begin{bmatrix} J_1\omega_1(k) + \Delta t(J_2{-}J_3)\omega_2(k)\omega_3(k) \\ J_2\omega_2(k) + \Delta t(J_3{-}J_1)\omega_3(k)\omega_1(k) \\ J_3\omega_3(k) + \Delta t(J_1{-}J_2)\omega_1(k)\omega_2(k) \\ q(k{+}1) \end{bmatrix},$$

$$F(k) = \begin{bmatrix} -\Delta t I_3 & J & 0_{3\times 4} \\ 0_{4\times 3} & R(k{+}1) & I_4 \end{bmatrix}.$$

3.3 Proposed Control and Guidance Architecture

In this thesis, we adopt a hybrid control and guidance architecture that leverages the complementary strengths of semi-definite programming (SDP)-based trajectory planning and classical quaternion-feedback control. This two-layer approach addresses the complexity of constrained spacecraft attitude maneuvers while maintaining real-time feasibility and robustness.

3.3.1 Guidance Layer: Optimal Trajectory Planning via SDP

The guidance layer is responsible for computing a feasible, constraint-compliant reference trajectory of spacecraft attitude and angular velocity. By formulating the constrained attitude guidance problem as a semi-definite program (SDP), the planner explicitly encodes mission-critical constraints such as:

- Keep-in and keep-out cones for payload protection and sensor line-of-sight requirements,
- Actuator torque saturation limits,
- Angular velocity bounds dictated by sensor and actuator capabilities.

The SDP planner generates globally optimal trajectories over a fixed maneuver horizon that minimize attitude and rate tracking errors while satisfying nonlinear rigid-body dynamics and quaternion normalization. These trajectories serve as time-indexed setpoints for the feedback controller.

This offline trajectory optimization provides *certifiable guarantees* of constraint satisfaction and optimality that classical heuristic or gain-scheduled methods cannot systematically ensure.

3.3.2 Control Layer: Real-Time Quaternion Feedback Regulation

While the SDP solver produces optimal reference trajectories, uncertainties in spacecraft dynamics, external disturbances, and modeling errors necessitate a control mechanism to track these trajectories accurately. The control layer implements a quaternion-feedback proportional-derivative (PD) regulator, which:

- Accepts the reference attitude and angular velocity from the guidance layer,

- Computes control torque commands based on the quaternion error and angular velocity feedback,
- Provides rapid disturbance rejection and compensates for unmodeled dynamics,
- Executes with minimal computational burden, suitable for onboard spacecraft processors.

This classical feedback control law complements the guidance planner by ensuring precise, real-time execution of complex constrained maneuvers.

3.3.3 Integration and Advantages of the Hybrid Framework

The separation of guidance and control into dedicated layers offers several advantages:

- **Computational Efficiency:** The computationally intensive SDP is solved offline or at a lower frequency, reducing onboard processor load.
- **Robustness:** The feedback controller manages transient errors and unmodeled perturbations that the SDP solver cannot anticipate.
- **Scalability:** This modular framework allows future improvements in trajectory optimization and feedback control independently.
- **Certifiability:** The SDP solver's explicit constraint handling provides safety assurances critical to mission success.

Overall, this architecture effectively combines rigorous optimization-based planning with proven, low-complexity feedback control to deliver reliable, high-performance constrained spacecraft attitude control.

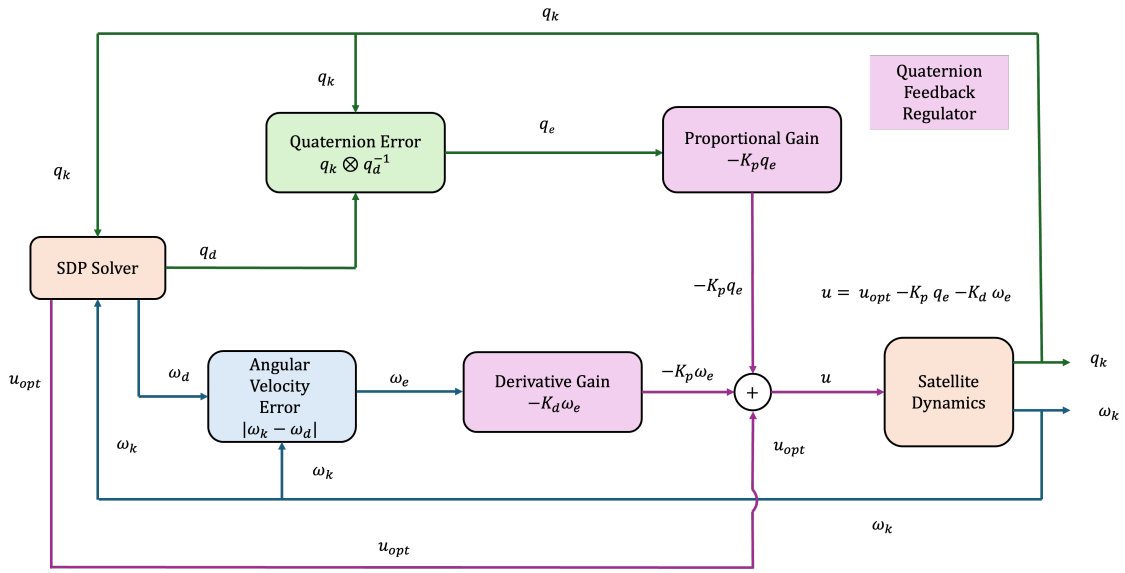


Figure 3.15: Hybrid control architecture integrating the semi-definite programming (SDP)-based guidance layer with a quaternion-feedback proportional-derivative (PD) control layer. The SDP solver computes constraint-compliant reference trajectories, which serve as inputs to the feedback regulator that computes real-time control torques. This modular design ensures robust, optimal, and computationally efficient spacecraft attitude control.

Chapter 4

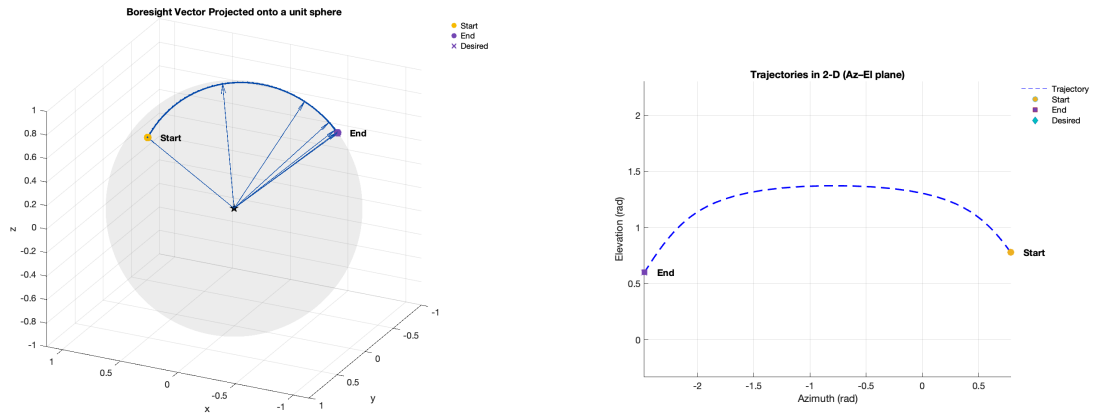
RESULTS

This chapter presents the simulation results that demonstrate the performance of the proposed constrained attitude control framework. We begin with the baseline *nominal* (unconstrained) case, which serves as a reference for evaluating constraint-handling performance. Subsequent sections introduce progressively more complex scenarios, including single and multiple keep-out constraints as well as combined keep-in/keep-out constraints. Each case is evaluated in terms of constraint satisfaction, tracking accuracy, and control effort. The simulation parameters used for all scenarios are summarised in Table 4.1.

4.1 Nominal (No Constraints)

Table 4.1 summarizes the simulation parameters for the unconstrained (nominal) case that serves as the baseline for all subsequent scenarios. Figure 4.1 shows the 3-D boresight trajectory on the unit sphere and the corresponding azimuth–elevation (Az–El) projection. The maneuver follows a smooth great-circle path from the initial to the final attitude with no evasive motion; the Az–El projection is monotonic in both coordinates.

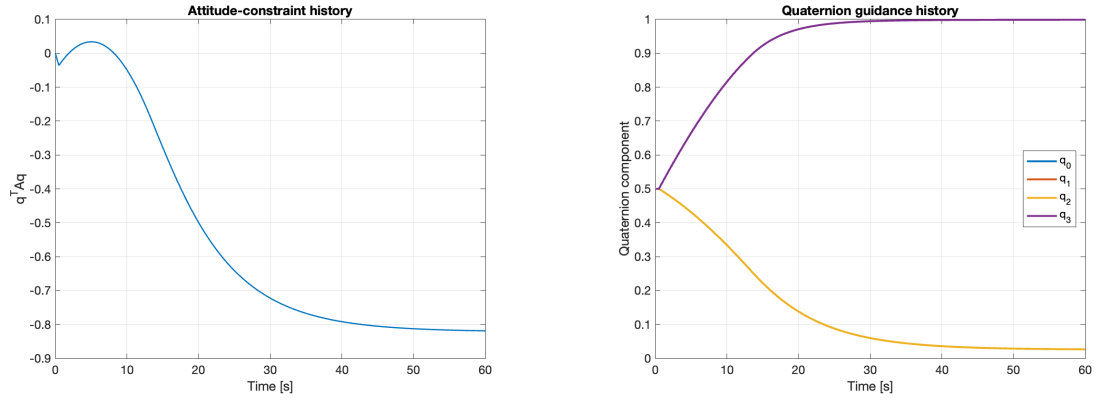
Closed-loop histories in Figures 4.2–4.3 illustrate the transient dynamics. The quaternion components converge rapidly to the target with well-damped behavior; body-rates remain well below the specified limit, and commanded torque stays under the actuator bound throughout the maneuver.



(a) 3-D boresight trajectory

(b) Azimuth–Elevation projection

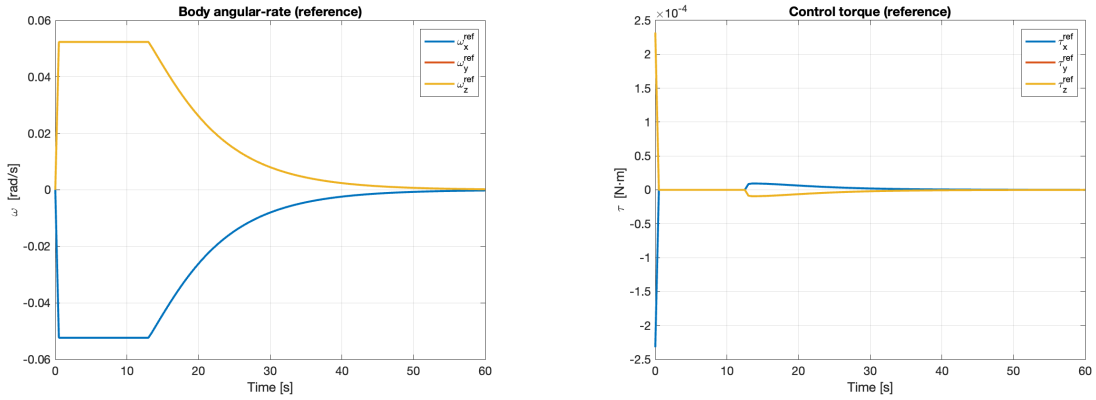
Figure 4.1: Nominal SDP guidance: spatial view (left) and Az–El projection (right).



(a) Azimuth / Elevation time history

(b) Quaternion components vs. time

Figure 4.2: Nominal SDP – angle traces (left) and quaternion convergence (right).



(a) Body-rate components vs. time

(b) Commanded torque vs. time

Figure 4.3: Nominal SDP – body-rate (left) and torque (right) time histories.

Table 4.1: Simulation Setup - 1

Parameter	**Value**
Max angular rate $\lvert\boldsymbol{\omega}_{\max}\rvert$	$\leq 3\,^{\circ}\,\mathrm{s}^{-1}$ ($\approx 0.0523\,\mathrm{rad/s}$)
Max control torque $\lvert\mathbf{u}_{\max}\rvert$	$\leq 2.5\,\mathrm{mN\,m}$
Final time t_f	$60\,\mathrm{s}$
Time Step dt	$0.5\,\mathrm{s}$
Initial attitude $\mathbf{q}_0$	$[0.5,\ 0.5,\ 0.5,\ 0.5]^{\mathsf{T}}$
Target attitude $\mathbf{q}_f$	$[0.0258,\ 0.0258,\ 0.0258,\ 0.9990]^{\mathsf{T}}$
Instrument boresight $\hat{\mathbf{b}}$	$[0.7,\ 0.5,\ 0.5]^{\mathsf{T}}$

4.2 One Keep-Out Constraint (1-KO SDP)

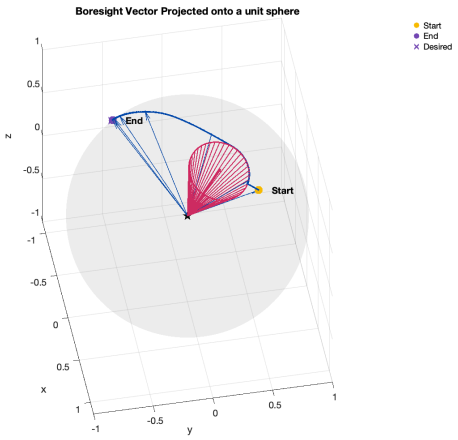
This experiment introduces a single inertially-fixed *keep-out* cone (e.g., sun-avoidance) that the boresight must not enter. The SDP solver produces a constraint-compliant reference trajectory that detours around the exclusion region, and the quaternion PD regulator tracks this trajectory.

Figures 4.4(a)–(b) compare the spatial boresight paths: the reference trajectory (a) skims the exclusion boundary before converging to the target, while the measured trajectory (b) follows the same arc closely. The corresponding Azimuth–Elevation projections in Figures 4.4(c)–(d) show the same single-detour maneuver in angular coordinates, with no oscillatory excursions.

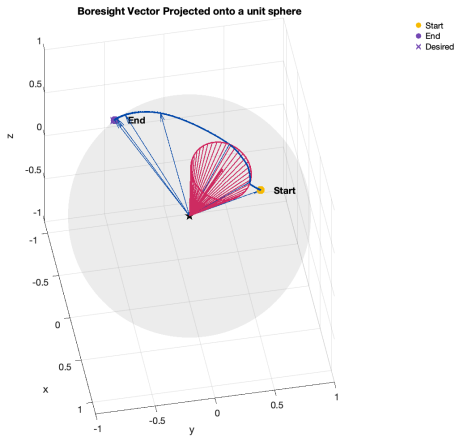
Constraint satisfaction is shown in Figure 4.5(a), where the algebraic condition $q^{\mathsf{T}} A q \leq 0$ remains strictly non-positive for the reference trajectory, indicating no cone violations. The measured quaternion tracking error in Figure 4.5(b) decays rapidly and remains small throughout the maneuver. Figures 4.5(c)–(d) show the quaternion components from the reference and measured trajectories, demonstrating close agreement.

Body-rate histories in Figures 4.6(a)–(b) remain within the specified limits and exhibit well-damped transients. Commanded torques in Figures 4.6(c)–(d) track the planned inputs without saturation.

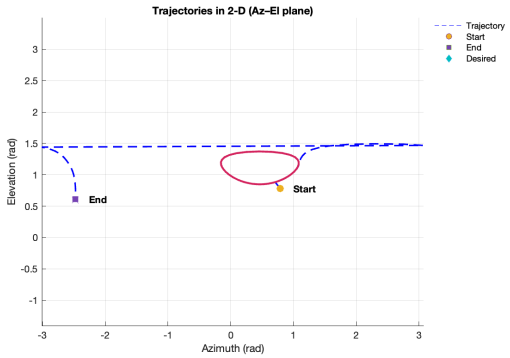
With a single keep-out region, the SDP solver generates a shortest-path detour that respects pointing constraints, and the feedback controller accurately executes the plan without violating actuator or rate bounds. The left–right figure layout (reference vs. measured) is kept consistent across plots to facilitate visual comparison.



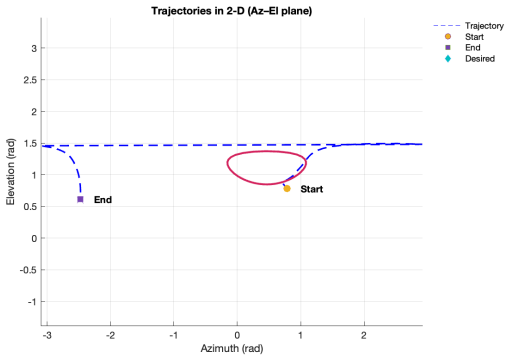
(a) 3-D boresight (reference)



(b) 3-D boresight (measured)

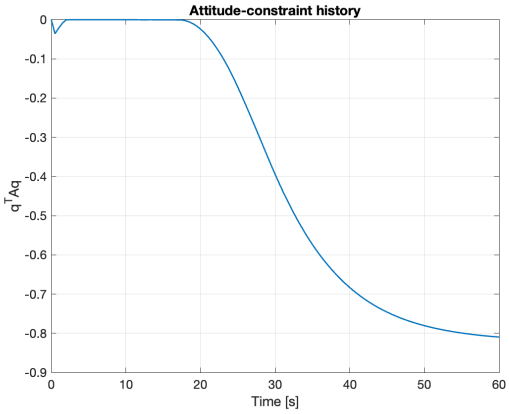


(c) Az–El projection (reference)

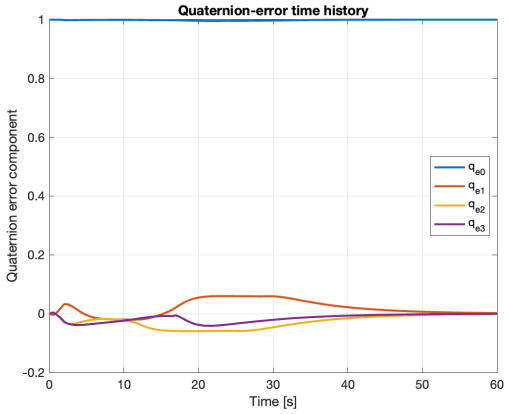


(d) Az–El projection (measured)

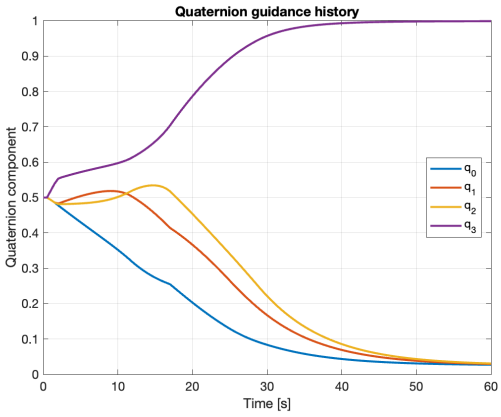
Figure 4.4: 1-KO SDP: spatial and Az–El views (reference vs. measured).



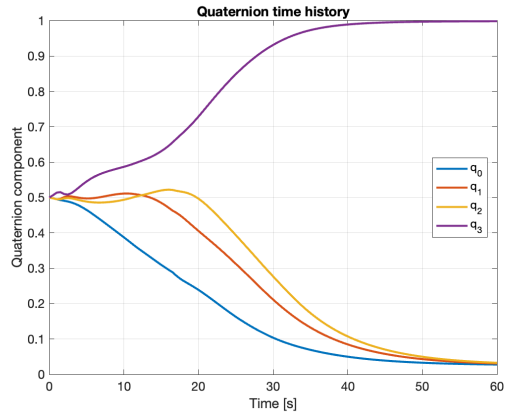
(a) Reference: constraint $q^{\mathsf{T}} Aq \leq 0$



(b) Measured: quaternion error

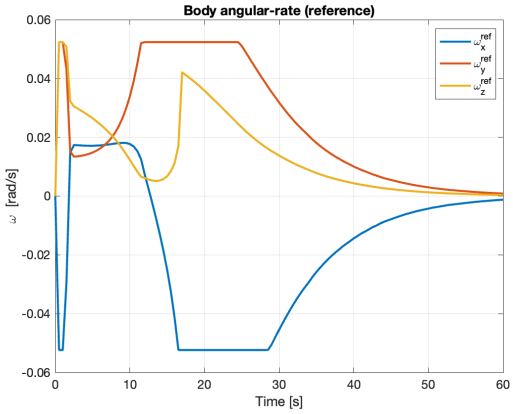


(c) Reference: quaternion components

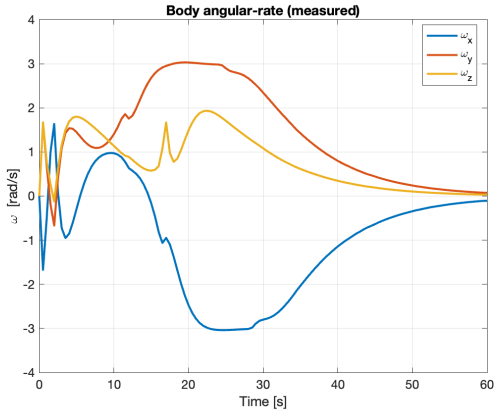


(d) Measured: quaternion components

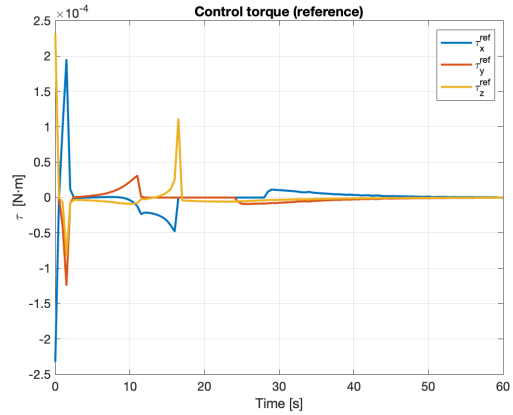
Figure 4.5: 1-KO SDP: top-constraint satisfaction (reference) vs. tracking error (measured); bottom-quaternion components (reference vs. measured).



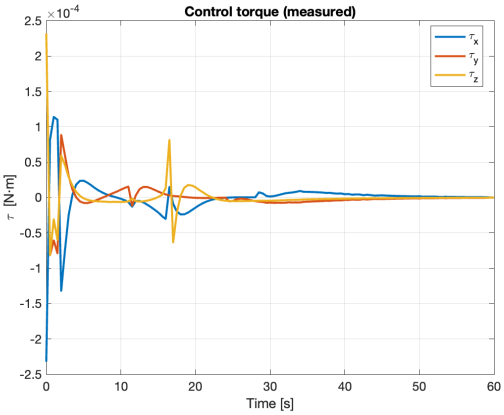
(a) Reference: body-rates



(b) Measured: body-rates



(c) Reference: commanded torque



(d) Measured: commanded torque

Figure 4.6: 1-KO SDP: body-rates (top) and commanded torque (bottom), reference vs. measured.

Inter-sample violation (ISV) and safety buffer. Because guidance and control are executed at a finite sampling period Δt, the continuous boresight can transiently approach the forbidden region between samples even if $q^{\mathsf{T}} A q \leq 0$ holds at the sampled instants. To mitigate this *inter-sample violation*, we inflate the keep-out half-angle in the guidance by a small buffer $\Delta\theta$:

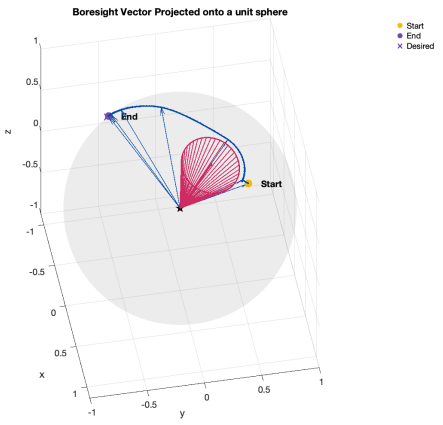
$$\theta_{\text{eff}} \;=\; \theta \;+\; \Delta\theta.$$

For the 1-KO results here, we use a nominal half-angle $\theta = 15^\circ$ and a safety buffer $\Delta\theta = 3^\circ$, so the guidance enforces

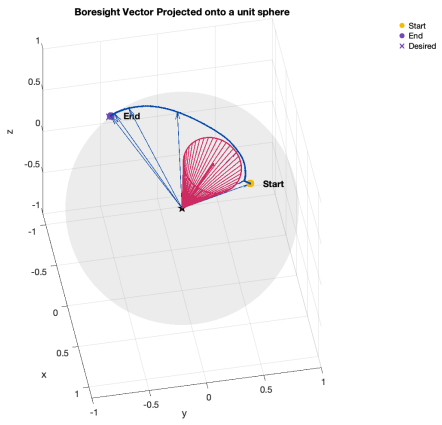
$$\theta_{\text{eff}} = 15^\circ + 3^\circ = 18^\circ.$$

This conservative margin keeps the continuous trajectory outside the true 15° exclusion cone even with zero-order-hold actuation between samples.

With the inflated half–angle $\theta_{\text{eff}} = 18^\circ$ (15° $+3^\circ$), both the planned and the measured boresight trajectories remain outside the true 15° keep–out cone for the entire maneuver; in particular, the measured path shows no constraint violation (Fig. 4.7, 4.8).

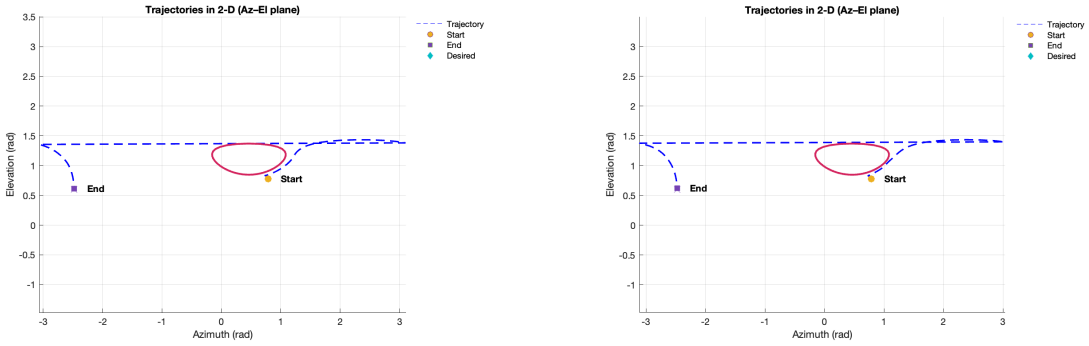


(a) 3-D boresight (reference)



(b) 3-D boresight (measured)

Figure 4.7: Inter-sample illustration for the 1-KO case (reference vs. measured).Guidance uses an inflated keep-out half-angle $\theta_{\text{eff}} = 18^\circ$ (15° nominal $+3^\circ$ buffer) to accommodate inter-sample excursions.



(a) Az–El projection (reference)

(b) Az–El projection (measured)

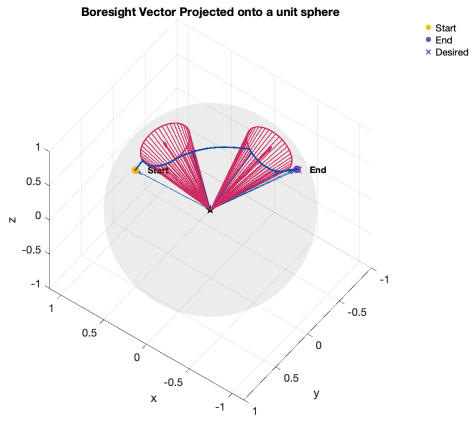
Figure 4.8: No constraint violations are observed in the measured trajectory once the 3° safety buffer is applied.

Moving forward, the same +3° half–angle inflation can be applied to the remaining cases to guard against inter–sample violation. For brevity, we do not replot the buffered trajectories or the corresponding measured responses in those sections; we report only the SDP guidance trajectories, noting that the buffer preserves constraint satisfaction without altering the qualitative behavior.

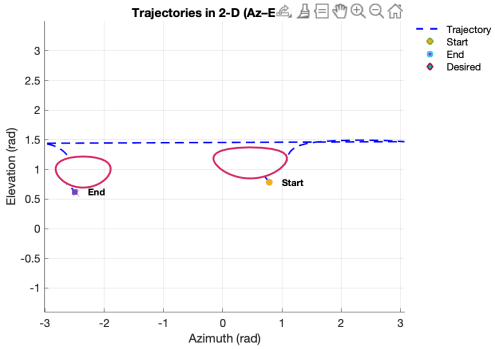
4.3 Two Keep-Out Constraint (2-KO SDP)

Figure 4.9a displays the 3-D antenna–boresight trajectory together with the two forbidden cones (rendered in red). The corresponding Azimuth–Elevation projection (Fig. 4.9b) and the key closed-loop time histories are given in Figures 4.9c–4.9f.

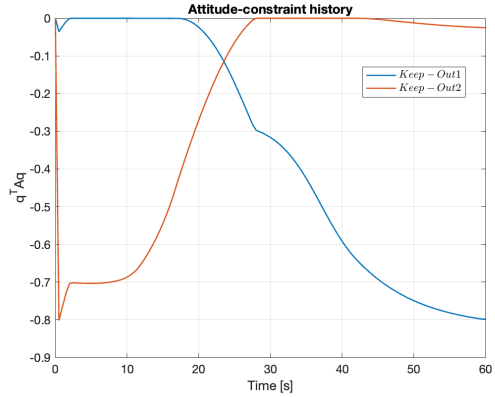
Compared with the nominal (unconstrained) run, the SDP now performs a deliberate detour to avoid *both* keep-out regions while still steering the boresight to the prescribed terminal attitude. Despite the added geometric complexity, all actuator (Fig. 4.9e) and rate limits (Fig. 4.9f) are respected and the algebraic attitude constraint $q^{\mathsf{T}} A q \leq 0$ remains strictly satisfied for the entire maneuver (Fig. 4.9c). The extra path length required to skirt both forbidden regions naturally increases the commanded control-torque with respect to the nominal run and the single–keep-out (1-KO) case.



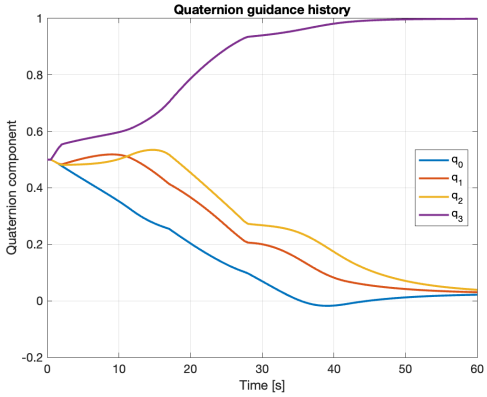
(a) 3-D boresight trajectory



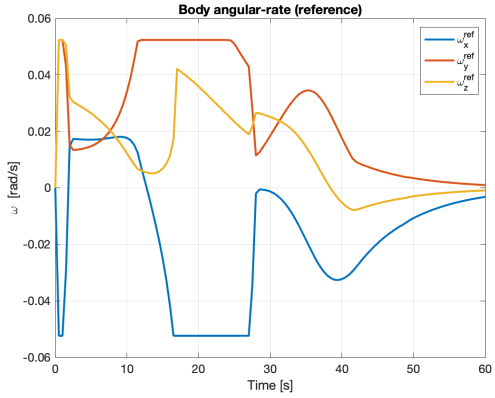
(b) Azimuth–Elevation projection



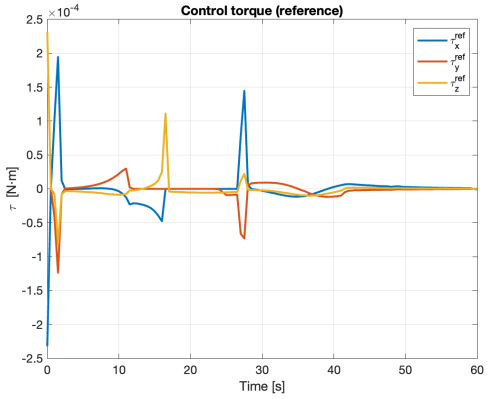
(c) Attitude constraint $q^{\mathsf{T}}Aq$



(d) Quaternion time history



(e) Body-rate reference



(f) Commanded torque

Figure 4.9: Two-keep-out (2-KO) SDP: spatial views (top row) and key closed-loop time histories (rows 2–3).

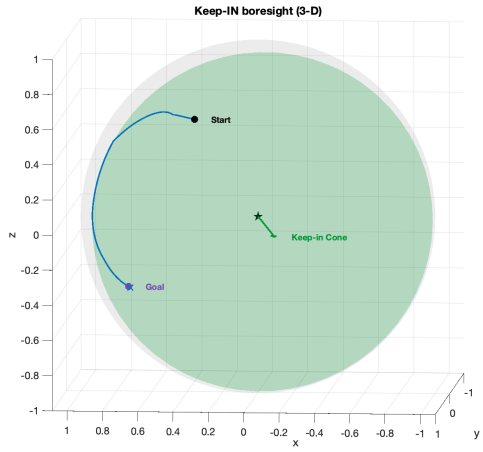
4.4 Keep-In and Keep-Out Constraint

This scenario performs simultaneous multi-axis keep-out and keep-in constraints. The maximum control torque and angular velocity are 2 N.m and 0.05 rad/s (= 2.86 deg/s), respectively. The antenna boresight, fixed along the body $+Y$ axis, must remain *inside* a single inclusion cone (green), whereas two optical sensors-aligned with the body $+X$ and $+Z$ axes-must steer clear of their respective exclusion cones (red). Figure 4.10 confirms that the antenna trajectory never exits the green keep-in region and converges to its prescribed target. Conversely, Figure 4.11 shows the telescope path detouring smoothly around both red keep-out zones before reaching its goal on the far side of the sphere. The time histories in Figure 4.12 indicate that the algebraic constraint $q^{\mathsf{T}}Aq$ is never violated (always ≤ 0), while body-rates and torques stay within the limits, and the slack variable α converges rapidly to zero, signalling that the optimization is pushing the quaternion to the desired states, rarely needing to soften the constraints.

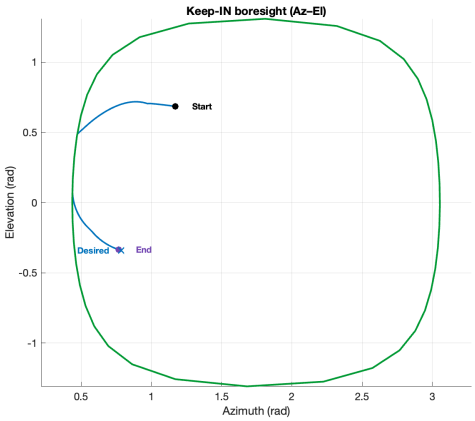
Overall, the controller achieves simultaneous inclusion and exclusion tracking without saturating the actuators or breaching rate limits, demonstrating its suitability for multi-sensor attitude missions.

Table 4.2: Simulation Setup – 2

Parameters	**Values**
Initial Attitude	$[0, -0.1632, -0.9254, -0.3420]$
Final Attitude	$[-0.9058, -0.2265, -0.1132, -0.3397]$
Final time t_f	$70s$
Time Step dt	$0.5s$
Boresights (keep-in)	$[0, 1, 0]$ (aligned with **y-axis**)
Boresights (keep-out)	$[1, 0, 0]$ (aligned with **x-axis**)
	$[0, 0, 1]$ (aligned with **z-axis**)

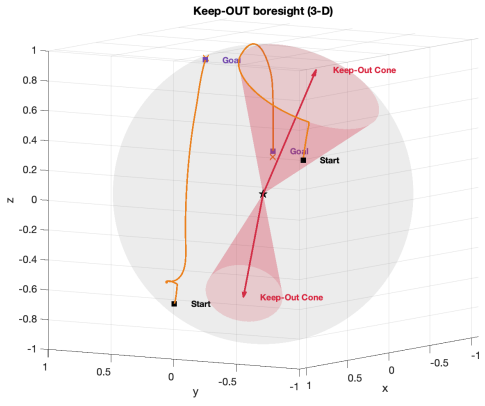


(a) 3-D view (unit sphere)

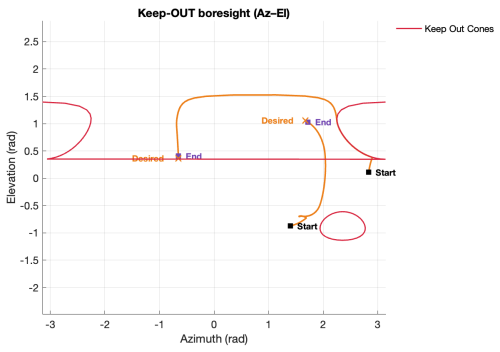


(b) Azimuth–Elevation projection

Figure 4.10: One–keep-in (1-KI) case. The antenna boresight stays inside the inclusion cone while tracking the reference.

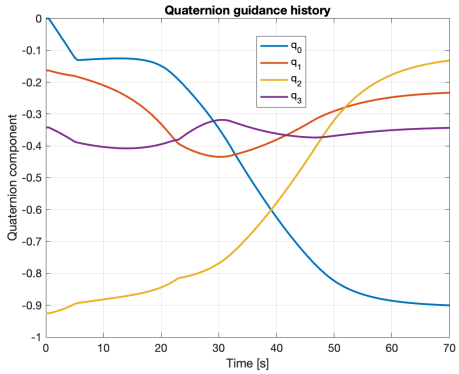


(a) 3-D view (unit sphere)

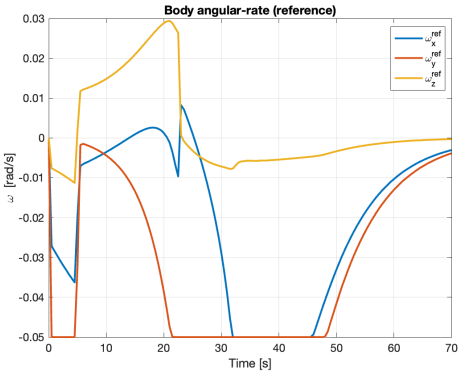


(b) Azimuth–Elevation projection

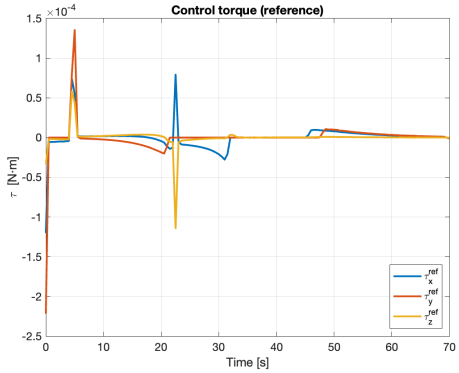
Figure 4.11: Two–keep-out (2-KO) case. The Optical Sensor boresight detours around exclusion zones before converging.



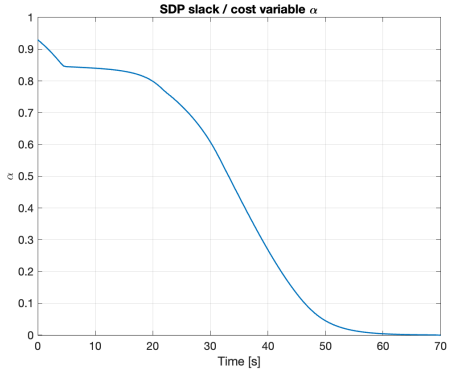
(a) Quaternion Time History



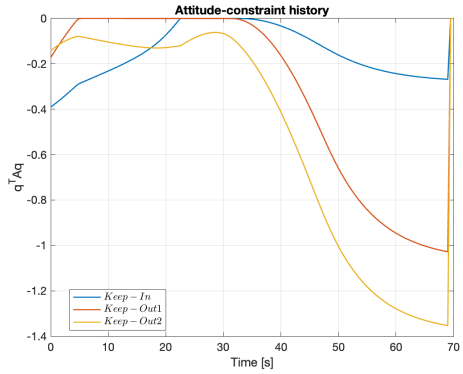
(b) Body-rate reference



(c) Commanded Torque



(d) Slack / cost α



(e) Constraint value $q^{\mathsf{T}}Aq$

Figure 4.12: Time histories of quaternion dynamics, body rates, torque commands, constraint satisfaction, and cost under the constrained SDP-based controller.

Chapter 5

CONCLUSION

This chapter synthesizes the findings of this work and reflects on its contributions, strengths, and limitations, as well as its relevance to current and future spacecraft missions. The discussion begins with a concise summary of the main technical advances achieved through the development of an SDP-based, quaternion-feedback attitude control framework. It then highlights the key pros and cons of the approach, outlines potential application domains where such methods offer significant benefits, and concludes by identifying promising research directions to extend the capabilities of the framework in more challenging and dynamic operational environments.

5.1 Summary of Contributions

This thesis used a quaternion-based spacecraft attitude control algorithm using semi-definite programming (SDP) to systematically enforce complex operational constraints, including actuator limits, keep-in and keep-out cones, and nonlinear rigid-body dynamics. Key contributions include:

- Formulation of the constrained attitude guidance problem as a convex SDP enabling globally optimal trajectory planning under multiple mission-critical constraints.
- Integration of a quaternion-feedback proportional-derivative controller to robustly track SDP-generated reference trajectories in real time despite model uncertainties and external disturbances.
- Demonstration of constraint satisfaction, feasibility, and high pointing accuracy across a variety of mission-relevant simulation scenarios.
- Analysis of strengths, including formal guarantees on constraint adherence and convex global optimality, and limitations such as computational demands and reliance on accurate modeling.

Together, these advances provide a rigorous framework bridging optimization theory and practical spacecraft attitude guidance and control.

5.2 Strengths and Limitations

The proposed semi-definite programming (SDP) based attitude control framework exhibits several strengths:

- Systematically handles complex operational constraints such as actuator torque limits and keep-out zones.
- Supports geometric constraint formulations well-suited for precision pointing tasks.
- The convex SDP formulation guarantees global optimality, eliminating local minima common in nonlinear attitude control problems.

However, there are limitations to note:

- The computational intensity of solving SDPs can pose challenges for real-time onboard implementation, especially for power- and resource-constrained spacecraft like CubeSats.
- The approach relies on accurate spacecraft dynamic and kinematic models; modeling errors can degrade performance.

This tradeoff between constraint fidelity and solver speed is a key consideration for practical deployment.

5.3 Applications

The SDP-based attitude control framework presented herein is especially well suited to emerging spacecraft missions requiring precise, constraint-aware maneuvers. Notable applications include:

- **Agile Earth Observation:** Enabling rapid retargeting without violating sun-avoidance zones or actuator saturations, critical for high-resolution imaging and sensor safety.
- **Satellite Formations and Laser Communications:** Maintaining tight relative orientations and line-of-sight constraints across multiple spacecraft despite jitter and

actuator limits.

- **Thruster and Boom Operations:** Preventing sensor damage by enforcing exclusion zones around plumes or deployable hardware during dynamic spacecraft configurations.
- **Space Telescopes and Sensitive Payloads:** Guaranteeing instrument protection through strict keep-in/out cone constraints while achieving accurate retargeting and calibration.

These examples illustrate the practical value of the convex optimization approach for current and next-generation space systems.

5.4 Future Work

Promising research directions to extend this work include:

- **Real-time onboard implementation:** Embedding efficient SDP solvers in flight processors to enable receding-horizon or model predictive control formulations for adaptive trajectory replanning.
- **Handling time-varying and moving constraints:** Developing methods to manage dynamic keep-in/out cones arising from tumbling targets or environmental changes using sequential convex approximations.
- **Robust and fault-tolerant control:** Incorporating parametric uncertainties, external disturbances, and actuator failures into the SDP framework through robust optimization and adaptive reconfiguration.
- **Integrated 6-DOF guidance and control:** Extending formulations to simultaneously handle translational and rotational dynamics for rendezvous, docking, and inspection missions.
- **Hybrid discrete-continuous optimization:** Combining convex SDPs with mixed-integer programming or global search methods to address non-convex constraint sets and multi-modal guidance problems.

These efforts will help realize fully autonomous, safe, and efficient spacecraft attitude guidance for complex mission scenarios.

BIBLIOGRAPHY

[1] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM Studies in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1994.

[2] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, England, March 2004.

[3] Brown University. Kinematics of rigid bodies - en4 notes. Lecture notes.

[4] Connie Kay Carrington and JL Junkins. Optimal nonlinear feedback control for spacecraft attitude maneuvers. *Journal of Guidance, Control, and Dynamics*, 9(1):99–107, 1986.

[5] Peng Cheng, Emilio Frazzoli, and Steven M LaValle. Improving the performance of sampling-based planners by using a symmetry-exploiting gap reduction algorithm. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 5, pages 4362–4368. IEEE, 2004.

[6] Arnold Christopher Cruz and Ahmad Bani Younes. Common frame dynamics for conically-constrained spacecraft attitude control. *Sensors*, 22(24):10003, 2022.

[7] Howard D Curtis. *Orbital Mechanics for Engineering Students*. Butterworth-Heinemann, 2019.

[8] E Feron, M Dahleh, E Frazzoli, and R Kornfeld. A randomized attitude slew planning algorithm for autonomous spacecraft. In *AIAA guidance, navigation, and control conference and exhibit*, page 4155, 2012.

[9] Gene F. Franklin, J. David Powell, and Abbas Emami-Naeini. *Feedback Control of Dynamic Systems*. Prentice Hall, Upper Saddle River, NJ, 6 edition, 2010.

[10] E.G. Gilbert and K.T. Tan. Linear systems with state and control constraints: the theory and application of maximal output admissible sets. *IEEE Transactions on Automatic Control*, 36(9):1008–1020, 1991.

[11] Hari B Hablani. Attitude commands avoiding bright objects and maintaining communication with ground station. *Journal of Guidance, Control, and Dynamics*, 22(6):759–767, 1999.

[12] Qinglei Hu, Yueyang Liu, Hongyang Dong, and Youmin Zhang. Saturated attitude control for rigid spacecraft under attitude constraints. *Journal of Guidance, Control, and Dynamics*, 43(4):790–805, 2020.

[13] John L Junkins and James D Turner. Optimal continuous torque attitude maneuvers. *Journal of Guidance and Control*, 3(3):210–217, 1980.

[14] Yoonsoo Kim, Mehran Mesbahi, and Fred Y Hadaegh. Dual-spacecraft formation flying in deep space: Optimal collision-free reconfigurations. *Journal of Guidance, Control, and Dynamics*, 26(2):375–379, 2003.

[15] Yoonsoo Kim, Mehran Mesbahi, Gurkirpal Singh, and Fred Hadaegh. On the constrained attitude control problem. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, page 5129, 2004.

[16] Yoonsoo Kim, Mehran Mesbahi, Gurkirpal Singh, and Fred Y Hadaegh. On the convex parameterization of constrained spacecraft reorientation. *IEEE Transactions on Aerospace and Electronic Systems*, 46(3):1097–1109, 2010.

[17] Henri C Kjellberg and E Glenn Lightsey. Discretized constrained attitude pathfinding and control for satellites. *Journal of Guidance, Control, and Dynamics*, 36(5):1301–1309, 2013.

[18] Henri Christian Kjellberg. *Constrained Attitude Guidance and Control for Satellites*. PhD thesis, The University of Texas at Austin, 2014.

[19] Jack B Kuipers. *Quaternions and rotation sequences: a primer with applications to orbits, aerospace, and virtual reality*. Princeton university press, 1999.

[20] F. Landis Markley and John L. Crassidis. *Fundamentals of Spacecraft Attitude Determination and Control*. Springer, 2014.

[21] David Q Mayne and WR Schroeder. Robust time-optimal control of constrained linear systems. *Automatica*, 33(12):2103–2118, 1997.

[22] Colin R Mclnnes. Large angle slew maneuvers with autonomous sun vector avoidance. *Journal of guidance, control, and dynamics*, 17(4):875–877, 1994.

[23] Daniel Selva and David Krejci. A survey and assessment of the capabilities of cubesats for earth observation. *Acta Astronautica*, 74:50–68, 2012.

[24] Malcolm D Shuster et al. A survey of attitude representations. *Navigation*, 8(9):439–517, 1993.

[25] Gurkirpal Singh, Glenn Macala, Edward Wong, Robert Rasmussen, Gurkirpal Singh, Glenn Macala, Edward Wong, and Robert Rasmussen. A constraint monitor algorithm for the cassini spacecraft. In *Guidance, navigation, and control conference*, page 3526, 1997.

[26] AM Sorensen. Iso attitude maneuver strategies. *NASA STI/Recon Technical Report A*, 95:975–987, 1993.

[27] Karlheinz Spindler. New methods in on-board attitude control (aas 98-308). *Spaceflight Dynamics 1998, Volume 100 Part 1, Advances in Astronautical Sciences*, 100:111, 1998.

[28] Chuangchuang Sun and Ran Dai. Spacecraft attitude control under constrained zones via quadratically constrained quadratic programming. In *AIAA guidance, navigation, and control conference*, page 2010, 2015.

[29] Margaret Tam and E Glenn Lightsey. Constrained spacecraft reorientation using mixed integer convex programming. *Acta Astronautica*, 127:31–40, 2016.

[30] SR Vadali, LG Kraige, and JL Junkins. New results on the optimal spacecraft attitude maneuver problem. *Journal of Guidance, Control, and Dynamics*, 7(3):378–380, 1984.

[31] Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM review*, 38(1):49–95, 1996.

[32] R Venkatachalam. Large angle pitch attitude maneuver of a satellite using solar radiation pressure. *IEEE Transactions on Aerospace and Electronic Systems*, 29(4):1164–1169, 2002.

[33] Zhenbo Wang. A survey on convex optimization for guidance and control of vehicular systems. *Annual Reviews in Control*, 57:100957, 2024.

[34] John Ting-Yung Wen and Kenneth Kreutz-Delgado. The attitude control problem. *IEEE Transactions on Automatic control*, 36(10):1148–1162, 1991.

[35] Bong Wie. *Space Vehicle Dynamics and Control*. AIAA, 1998.

[36] Bong Wie, Haim Weiss, and Ari Arapostathis. Quarternion feedback regulator for spacecraft eigenaxis rotations. *Journal of Guidance, Control, and Dynamics*, 12(3):375–380, 1989.

[37] Rafal Wisniewski and Piotr Kulczycki. Slew aneuver control for spacecraft equipped with star camera and reaction wheels. *Control engineering practice*, 13(3):349–356, 2005.

[38] Rui Xu, Hui Wang, Shengying Zhu, Huiping Jiang, and Zhaoyu Li. Multiobjective planning for spacecraft reorientation under complex pointing constraints. *Aerospace Science and Technology*, 104:106002, 2020.

[39] Zichen Zhao, Haibin Shang, and Yue Dong. Autonomous attitude planning for gravity wave detection using hybrid convex optimization. *Aerospace Science and Technology*, 130:107923, 2022.

Appendix A

WHERE TO FIND THE FILES

All code, scripts, and supplementary materials related to this thesis are available in the following GitHub repositories. These include MATLAB/Simulink files, SDP formulations, and documentation for reproducing results and simulations.

- **Feedback Controller:**
  https://github.com/shavykashyap/satellite-attitude-feedback-quaternion.git

- **Semi-Definite Programming and Trajectory Generation:**
  https://github.com/shavykashyap/satellite-attitude-sdp.git

For questions, updates, or to contribute, please refer to the README file in each repository or contact the author via GitHub.