



POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Time-Optimal Guidance and Control for Spacecraft Interception in LEO

TESI DI LAUREA MAGISTRALE IN
AUTOMATION AND CONTROL ENGINEERING

Author: **Riccardo Marinò**

Student ID: 10700016
Advisor: Prof. Andrea Colagrossi
Academic Year: 2025-26

*"Your future hasn't been written yet. No one's has.
Your future is whatever you make it.
So make it a good one."*

— Dr. Emmett Brown

Abstract

This thesis presents a novel hybrid Guidance, Navigation and Control (GNC) architecture designed for the rapid, time-optimal interception of non-cooperative targets in Low Earth Orbit (LEO) during the terminal rendezvous phase. The thesis addresses the critical challenge of autonomously and safely approaching uncooperative space assets in minimal time, while adhering to hard constraints such as thrust saturation and limited propellant.

To reduce the computational load of continuous re-optimization while guaranteeing robustness against disturbances, the proposed architecture decouples the control problem into two layers. The Guidance Layer employs a quasiconvex bisection search combined with linear quasi-nonsingular Relative Orbital Elements (ROE) dynamics to solve the free-final-time optimal control problem. This formulation guarantees global optimality and strict constraint satisfaction. Concurrently, the Tracking Layer utilizes a Sliding Mode Controller (SMC), formulated using the Clohessy-Wiltshire (CW) equations, to track the trajectory and actively reject disturbances (such as J_2 oblateness). The SMC parameters are automatically tuned via Particle Swarm Optimization (PSO). Additionally, an Artificial Potential Field (APF) strategy is integrated into the final approach phase to enforce safe approach corridor and ensure collision avoidance around the target.

The overall GNC system is tested and simulated under realistic conditions. Specifically, the simulations evaluate the system's performance during the critical navigation handover, transitioning from state estimation via an Inertial Navigation System (INS) to direct relative measurements provided by proximity sensors. Crucially, the architecture's robustness is further validated against severe actuator faults, such as stuck-open thrusters and navigation failures. The system successfully demonstrates the reliable execution of passive safety abort maneuvers via eccentricity/inclination (E/I) vector separation. Ultimately, this thesis provides a computationally efficient, globally optimal and robust framework for time-critical spacecraft interception missions.

Keywords: Rendezvous, Time-Optimal Control, GNC, Relative Orbital Elements (ROE), Clohessy-Wiltshire (CW), Sliding Mode Control (SMC), Particle Swarm Optimization (PSO), Artificial Potential Field (APF), Convex Optimization, Failure.

Abstract in lingua italiana

Questa tesi presenta un'architettura ibrida di Guidance and Control progettata per l'intercettazione a tempo minimo di target non cooperativi in orbita bassa terrestre (LEO) durante la fase di rendezvous terminale. La tesi affronta la complessa sfida legata all'avvicinamento autonomo ad asset spaziali non cooperativi nel minor tempo possibile, rispettando rigorosamente vincoli operativi quali la saturazione dei propulsori e limiti del propellente.

Per ridurre il carico computazionale derivante da una continua ri-ottimizzazione, garantendo al contempo robustezza rispetto ai disturbi, l'architettura proposta disaccoppia il problema di controllo su due livelli. Il Guidance Layer impiega un algoritmo di ricerca quasi-convessa, combinato con la dinamica lineare dei Relative Orbital Elements (ROE), per risolvere il problema di controllo ottimo a tempo finale libero. Questa formulazione assicura un'ottimalità globale e il rispetto dei vincoli. Parallelamente, il Tracking Layer utilizza un controllore Sliding Mode (SMC), formulato sulle equazioni di Clohessy-Wiltshire (CW), per seguire la traiettoria di riferimento e reiettare i disturbi (come l'effetto J_2). I parametri del controllore SMC sono calibrati automaticamente tramite un algoritmo di Particle Swarm Optimization (PSO). Inoltre, nella fase di docking, viene integrata una strategia basata sul Artificial Potential Field (APF) per garantire un corridoio di approccio ed evitare collisioni con il target.

L'intero sistema GNC è testato e simulato in condizioni realistiche. Nello specifico, le simulazioni valutano le prestazioni del sistema durante la fase di handover della navigazione, ovvero il passaggio dalla stima dello stato tramite un sistema di navigazione inerziale (INS) all'utilizzo delle misurazioni relative fornite dai sensori di prossimità. Aspetto fondamentale, la robustezza dell'architettura è stata ulteriormente validata a fronte di guasti agli attuatori, come il blocco o l'attivazione permanente dei propulsori, e di anomalie nel sistema di navigazione. Il sistema dimostra la capacità di eseguire in modo affidabile manovre di interruzione tramite la separazione del vettore eccentricità/inclinazione (E/I). In sintesi, questa tesi fornisce un framework computazionalmente efficiente, globalmente ottimo e robusto per missioni di intercettazione spaziale.

Parole chiave: Rendezvous, Controllo Ottimo a Tempo Minimo, GNC, Relative Or-

bitol Elements (ROE), Clohessy-Wiltshire (CW), Sliding Mode Control (SMC), Particle Swarm Optimization (PSO), Artificial Potential Field (APF), Ottimizzazione Convessa.

Contents

Abstract	iii
Abstract in lingua italiana	v
Contents	vii
1 Introduction	1
1.1 Technological Context and Motivation	1
1.2 Literature Review: Guidance and Control in Relative Orbital Dynamics . .	3
1.3 Scope and Contribution of This Thesis	7
2 Theoretical Background	9
2.1 Reference frames	9
2.2 Orbital Propagation	13
2.3 Satellite Modeling	17
3 Relative Motion Dynamics	23
3.1 Clohessy-Wiltshire (CW) Equations	24
3.2 Relative Orbital Elements (ROE)	26
3.3 Modeling Error and Linearity Comparison	31
4 Control System Design	33
4.1 Optimal Control	34
4.2 Sliding Mode Controller (SMC)	46
4.3 Final Approach Strategy	52
4.4 Hybrid Control	57
5 Particle Swarm Optimization (PSO)	61
5.1 Topology	62
5.2 PSO Algorithm	63

5.3	Optimization Results	68
6	Simulations and Results	73
6.1	Mission Scenario Definition	74
6.2	Open-Loop Error Characterization	76
6.3	Comparative Analysis of Guidance Strategies	79
6.4	Final Approach	97
6.5	Failures	102
7	Conclusion and Future Developments	111
7.1	Conclusion	111
7.2	Future Developments	112
	Bibliography	115
A	Appendix A	119
A.1	Transformation from Relative Orbital Elements to the Hill Frame	119
A.2	Transformation from Keplerian Elements to Relative Orbital Elements	120
A.3	Simulation Constants	122
	List of Figures	123
	List of Tables	127
	List of Symbols	130

1 | Introduction

1.1. Technological Context and Motivation

Rendezvous and Proximity Operations (RPO) have evolved from a purely cooperative capability developed for docking missions into a strategic and technologically critical function of modern space systems. While early rendezvous missions, such as crewed docking with space stations, were characterized by fully cooperative targets, precise navigation aids and a strong emphasis on fuel efficiency and safety margins, the current operational landscape is fundamentally different.

The progressive congestion of Low Earth Orbit (LEO) has been matched by the emergence of dual-use and security-sensitive missions, effectively turning space into a contested operational domain. Recent real-world events have demonstrated the strategic necessity of rapid, non-cooperative rendezvous. Notable examples include the 2020 shadowing of the US national security satellite USA 245 by the Russian inspector satellites Cosmos 2542 and 2543 [1]. In that event, the chaser spacecraft executed substantial phasing maneuvers to synchronize its orbit with the non-cooperative target, effectively demonstrating the operational reality of hostile interception and close-proximity inspection. Furthermore the high-mobility towing operations demonstrated by China's Shijian-21 highlight the growing capability for physical engagement and relocation. In particular, recent developments indicate that China has significantly accelerated its acquisition of expertise regarding non-cooperative rendezvous in both GEO and LEO regimes [2]. In these 'gray zone' scenarios, the target is inherently non-cooperative, may perform evasive maneuvers, or may be tumbling, requiring the chaser to execute interception under severe operational pressure rather than cooperative planning.

In this context, close-proximity rendezvous at relative distances below 100 km represents a particularly challenging regime. At such separations, the relative dynamics are strongly coupled through orbital geometry and small errors in semi-major axis or velocity translate into significant along-track drift. Furthermore, operational constraints become stringent: thrust saturation, limited propellant availability, navigation uncertainty and strict colli-

sion avoidance requirements must all be simultaneously satisfied.

Unlike classical cooperative docking, where fuel minimization and passive safety are paramount, defense-oriented interception introduces a different primary objective: time minimization. In adversarial or rapid-response scenarios, mission effectiveness is strictly coupled to reaction time. The chaser must intercept the target to inspect, characterize, or neutralize it before the target can maneuver out of range or conceal its intent. Consequently, the problem is framed not as a soft dock, but as a time-optimal interception, where the goal is to reach a 'point of engagement' in the minimum possible time while respecting hard physical constraints.

Despite the extensive literature on rendezvous guidance, several limitations remain evident when considering this operational framing. Pure open-loop time-optimal control solutions are highly sensitive to modeling inaccuracies and disturbances. Model Predictive Control (MPC) frameworks, while capable of handling constraints explicitly, require the solution of a constrained optimization problem at every sampling instant, which can impose a computational burden incompatible with small spacecraft processors and high update rates.

These gaps motivate the need for a guidance and control architecture that satisfies the following operational requirements:

- **Minimum maneuver time:** the final time t_f must be minimized under bounded thrust.
- **Hard thrust saturation:** actuator limits must be strictly enforced.
- **Fuel feasibility:** total ΔV and propellant consumption must remain within mission allocation.
- **Terminal accuracy:** the final relative state error must be compatible with interception requirements.
- **Robustness:** performance must be maintained under modeling errors, J2 perturbations, atmospheric drag, navigation uncertainty and failures.
- **Computational Efficiency:** the control algorithm must be formulated to minimize computational load, enabling potential implementation on resource-constrained on-board processors.

The central motivation of this thesis is therefore to develop a control architecture framework capable of generating time-optimal interception trajectories while maintaining robustness and computational efficiency where the control parameters are not tuned on the

specific mission but are automatically optimized based on the mission profile. The proposed approach addresses this challenge through a hierarchical structure that combines convex trajectory generation in Relative Orbital Elements (ROE) with a robust nonlinear tracking layer.

1.2. Literature Review: Guidance and Control in Relative Orbital Dynamics

The problem of spacecraft rendezvous has been extensively studied for over six decades, evolving from cooperative docking missions to complex proximity operations involving non-cooperative targets. The literature spans three interconnected domains: modeling of relative orbital dynamics, trajectory generation under constraints and robust tracking control. In recent years, the growing interest in on-orbit servicing, debris removal, formation flying and time-critical interception missions has significantly expanded the state of the art.

Relative Motion Modeling: Cartesian vs. Orbital Element Approaches

The foundation of modern relative motion modeling lies in the linearization of the two-body problem. The Clohessy–Wiltshire (CW) equations [3] provided the first tractable linear time-invariant (LTI) representation of relative motion in the Hill frame for circular reference orbits. Their analytical solution enabled onboard implementation during the Gemini and Apollo programs and remains widely used for short-range rendezvous and controller design.

Extensions to elliptical reference orbits were introduced by Tschauner and Hempel [4], resulting in linear time-varying (LTV) formulations. These models preserve analytical tractability but introduce periodic coefficients and increased complexity in controllability analysis.

Despite their historical relevance, Cartesian linearizations suffer from geometric limitations. The CW model approximates the mean motion and truncates nonlinear coupling terms, leading to modeling errors that scale with $(\rho/a)^2$. Alfriend and Yan [5] demonstrated that although CW and geometric formulations share the same formal error order, linearization in Cartesian space induces artificial secular drift effects not present in the true nonlinear dynamics.

To overcome these limitations, Schaub and Alfriend [6] formalized the nonlinear mapping between Cartesian states and differences in orbital elements, establishing the theoretical basis for Relative Orbital Elements (ROE). Subsequently, D’Amico and Montenbruck [7] introduced the quasi-nonsingular ROE formulation for near-circular LEO orbits, which separates in-plane and out-of-plane motion and avoids singularities associated with classical Keplerian elements.

ROE-based approaches offer significant advantages for long-duration proximity operations: the state variables evolve slowly under Keplerian motion, drift conditions can be enforced directly through $\delta a = 0$ and passive safety constraints can be expressed geometrically ($E \setminus I$ Separation [7]). These properties have made ROE the dominant representation in formation flying and distributed space systems. However, most ROE-based works focus on fuel-optimal or formation-keeping problems rather than explicitly time-critical interception.

Trajectory Generation Under Constraints: Convex and Successive Convexification Methods

Optimal control for rendezvous has traditionally been addressed through Pontryagin’s Minimum Principle [8–10], yielding fuel-optimal impulsive or low-thrust solutions. While theoretically rigorous, these formulations are difficult to generalize, as incorporating path constraints, actuator dynamics, or fuel limitations requires deriving complex costate equations for each new scenario.

In recent decades, convex optimization has emerged as a powerful alternative. When the relative dynamics are linear and constraints are convex, the trajectory generation problem can be formulated as a second-order cone program (SOCP) or quadratic program (QP). Convex methods guarantee global optimality and deterministic feasibility detection, properties particularly attractive for safety-critical applications.

For nonlinear dynamics, successive convexification (SCvx) techniques have been developed to iteratively linearize and convexify the problem around a reference trajectory [11]. These methods have demonstrated strong performance in planetary landing and constrained spacecraft guidance problems [12]. However, SCvx increases computational complexity and typically requires multiple convex solves per guidance update.

Time-optimal rendezvous introduces additional challenges. The free-final-time problem is inherently non-convex due to the coupling between state evolution and horizon length [8] this problem can be addressed via indirect methods or discretized nonlinear programming,

but convergence and robustness are often sensitive to initialization. In this thesis, quasi-convex reformulations using bisection on the final time is proposed when the fixed-time feasibility problem remains convex. This approach guarantees convergence to the global minimum time provided the reachable set is monotonic in time [13].

Nevertheless, the literature shows limited integration of ROE-based linear dynamics with explicit quasiconvex minimum-time search under hard thrust saturation and fuel constraints. Most convex rendezvous formulations either assume fixed time horizons or prioritize fuel minimization rather than time-critical objectives.

Tracking and Robust Control Strategies

While trajectory generation determines the nominal guidance profile, the tracking layer ensures robustness against disturbances, model mismatch and navigation errors. Several control strategies have been proposed in the literature.

Linear Quadratic Regulators (LQR) have been widely adopted for CW-based relative motion due to their simplicity and closed-form gain computation [14]. LQR provides optimality with respect to a quadratic cost but assumes accurate modeling and does not inherently handle actuator saturation or bounded disturbances.

Model Predictive Control (MPC) extends LQR by explicitly incorporating state and input constraints. In rendezvous applications, MPC has been used to enforce thrust bounds, collision avoidance regions and terminal constraints [15]. Robust MPC variants introduce constraint tightening to account for bounded disturbances. However, the computational cost of MPC remains a primary concern. For interception missions, this creates a strict computational budget, as the processor must support both the iterative control solver and the intensive data processing required for precise target tracking.

Robust control approaches, including H_∞ synthesis [16] and Sliding Mode Control (SMC), introduced by Utkin [17], which has gained popularity in aerospace applications due to its finite-time convergence properties and invariance to matched uncertainties [18, 19]. In rendezvous problems, SMC is typically employed as a tracking controller to enforce convergence to a precomputed trajectory despite perturbations such as J2 effects, drag, or sensor noise [20]. However, SMC does not generate optimal trajectories and must be integrated within a hierarchical framework.

Operational Constraints: Thrust, Fuel, Safety, Sensors and Perturbations

Operational constraints are central to real missions and strongly influence architecture selection. Hard thrust saturation limits acceleration authority and directly impacts minimum achievable maneuver time. Fuel consumption constraints, governed by the rocket equation, introduce nonlinear coupling between control effort and available mass.

Safety constraints are typically implemented through Keep-Out Zones (KOZ), line-of-sight restrictions, or passive safety orbits [21]. Cartesian formulations often require nonlinear position constraints, while ROE-based methods enable geometric separation of eccentricity and inclination vectors to enforce collision avoidance indirectly [7].

Navigation architecture further complicates the problem. During far-range phases, spacecraft may rely on inertial navigation or GNSS, while close-range operations employ relative sensors such as lidar or vision-based systems. Sensor handover introduces estimation discontinuities and increased uncertainty. Few works explicitly integrate navigation regime transitions within time-optimal guidance design.

Environmental perturbations, including J2 oblateness and atmospheric drag, induce secular drift and model mismatch. Many convex guidance works assume purely Keplerian dynamics, delegating perturbation rejection entirely to the tracking layer. The interaction between modeling accuracy in the guidance layer and robustness in the tracking layer remains an open architectural question.

Research Motivation

From the literature survey, several observations emerge that highlight the lack of a unified solution for rapid interception.

First, while ROE-based formulations provide superior geometric consistency for long-duration relative motion, most optimal control works employ Cartesian CW dynamics or focus on fuel-optimal formation flying rather than explicitly time-minimal interception.

Second, convex optimization guarantees global optimality and deterministic feasibility but is often implemented in fixed-time formulations or within receding-horizon MPC frameworks. However, the continuous re-optimization inherent to MPC introduces a computational overhead that can be reduced by decoupling the high-frequency control loop from the lower-frequency guidance updates.

Third, robust tracking methods such as SMC effectively reject disturbances but are rarely

integrated with globally optimal time-critical trajectory generators in a computationally efficient hierarchical structure.

Finally, operational constraints, thrust saturation, propellant limits, safety margins and sensor handover, are frequently treated separately rather than embedded coherently within a unified architecture.

1.3. Scope and Contribution of This Thesis

The objective of this thesis is to bridge these gaps by designing and validating a hybrid Guidance, Navigation and Control (GNC) architecture for time-optimal interception of non-cooperative targets in Low Earth Orbit.

To address the limitations of existing methods, the proposed solution adopts a hierarchical architecture composed of two decoupled layers:

1. **Guidance Layer (Convex ROE):** A minimum-time optimal control problem is posed as a quasiconvex feasibility search in Relative Orbital Elements. This ensures global optimality and strict enforcement of thrust/fuel constraints without the computational load of real-time MPC.
2. **Tracking Layer (SMC):** The reference trajectory is tracked in the Hill (RTN) frame using Sliding Mode Control with automatic gain selection, providing robustness against J2 perturbations, drag and navigation errors.

The primary contributions of this thesis are:

- A consistent formulation of the interception problem as a time-optimal rendezvous in quasi-nonsingular ROE space.
- The integration of a quasiconvex bisection search with linear ROE dynamics to solve the free-final-time problem with guaranteed convergence.
- The explicit modeling of propellant consumption and hard thrust saturation within a convex framework.
- The validation of the decoupled architecture (Guidance vs. Tracking) under non-linear orbital propagation.

To quantitatively evaluate the proposed method, the following performance metrics are defined:

- **Final time t_f :** minimum achievable maneuver duration.

- **Terminal state error:** norm of the final relative position and velocity error.
- **Fuel consumption:** mass depletion computed from the propulsion model.
- **Constraint satisfaction:** verification of thrust limits and safety distances.

It is important to clarify the scope of the work. This thesis does not address full mission-level autonomy, onboard target detection, or attitude control coupling. The target is assumed to follow natural orbital dynamics without active maneuvers and the focus is restricted to translational relative motion. Furthermore, the linear ROE model is adopted as a deliberate trade-off between accuracy and convex tractability; higher-order nonlinear ROE corrections are not included.

Within these boundaries, the thesis demonstrates that this hybrid architecture provides a computationally efficient, globally optimal and robust solution to time-critical interception problems.

2 | Theoretical Background

Following the definition of the mission scope and the architectural requirements in Chapter 1, it is essential to establish a rigorous simulation environment to validate the proposed control strategies. While the guidance and control algorithms developed in subsequent chapters often rely on linearized approximations, the validation of their performance requires a high-fidelity representation of the physical reality that captures the full nonlinear behavior of the spacecraft.

This chapter presents the mathematical and physical foundations used to model the orbital motion of the Chaser and Target spacecraft. The primary objective is to define the “ground truth” environment needed to assess the robustness of the hybrid architecture against the inevitable gap between the internal guidance models and the actual orbital dynamics. To this end, the fundamental reference frames, nonlinear orbital propagation methods and dominant environmental perturbations, specifically Earth’s oblateness (J_2) and atmospheric drag, are rigorously defined.

The chapter is organized as follows:

- **Section 2.1** defines the inertial and relative reference frames employed to describe the spacecraft state.
- **Section 2.2** details the nonlinear orbital propagation methods based on Gauss’s Planetary Equations, including the mathematical modeling of environmental perturbations.
- **Section 2.3** specifies the physical properties of the Chaser and Target satellites, including mass, propulsion characteristics and navigation sensor models used in the simulation.

2.1. Reference frames

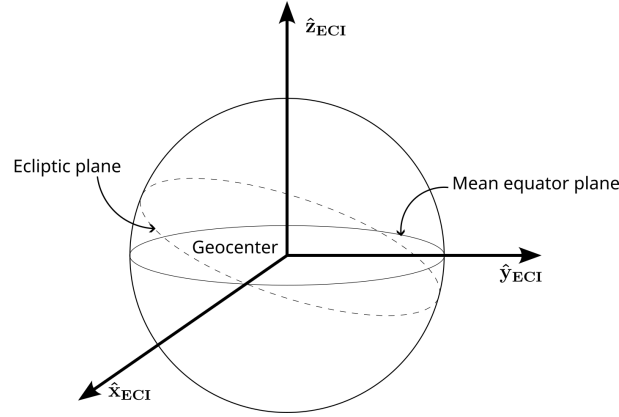


Figure 2.1: The Earth Centered Inertial (ECI) reference frame.

Earth-Centered-Inertial

The Earth-Centered Inertial (ECI) frame, denoted as $\mathcal{F}_{ECI}$, serves as the fundamental inertial reference system for formulating the equations of motion. As illustrated in Figure 2.1, the frame is defined by the orthonormal basis vectors $\{\hat{\mathbf{x}}_{ECI}, \hat{\mathbf{y}}_{ECI}, \hat{\mathbf{z}}_{ECI}\}$ and is centered at the Earth's center of mass (Geocenter).

The geometric definition of the axes relies on two fundamental astronomical planes shown in the figure: the *Mean Equator Plane* (defined by Earth's rotation) and the *Ecliptic Plane* (defined by Earth's orbit around the Sun).

- **Origin (O_{ECI}):** Located at the Geocenter. This serves as the inertial point of reference for the satellite's position vector $\mathbf{R}_{ECI}$.
- **The Principal Axis ($\hat{\mathbf{x}}_{ECI}$):** This axis points toward the *Vernal Equinox*. As depicted in Figure 2.1, this direction is geometrically defined by the line of intersection between the Mean Equator Plane and the Ecliptic Plane. It points in the direction of the Sun's position relative to Earth on the first day of Spring (ascending node).
- **The Fundamental Axis ($\hat{\mathbf{z}}_{ECI}$):** This axis is normal to the Mean Equator Plane, pointing in the direction of the Earth's geographic North Pole. It coincides with the Earth's mean rotational axis.
- **The Third Axis ($\hat{\mathbf{y}}_{ECI}$):** The y-axis completes the right-handed orthogonal triad, lying within the equatorial plane such that $\hat{\mathbf{y}}_{ECI} = \hat{\mathbf{z}}_{ECI} \times \hat{\mathbf{x}}_{ECI}$.

Since this frame is non-rotating relative to the fixed stars, Newton's laws of motion are valid without the addition of fictitious forces.

Hill Frame

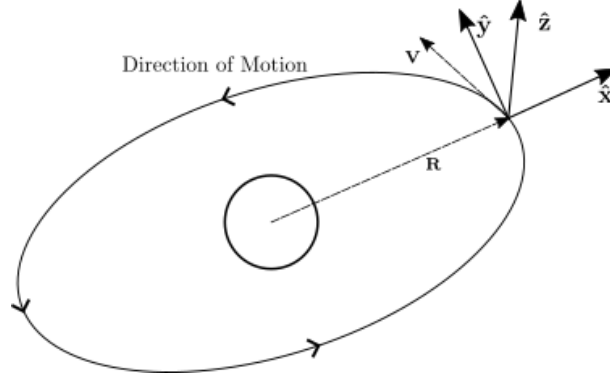


Figure 2.2: Hill reference frame

To describe the relative motion of the satellite, we employ the Hill (RTN) frame (shown in Figure 2.2). Unlike the inertial ECI frame, this is a rotating reference frame centered on the satellite's center of mass.

The orientation of the Hill frame basis vectors $\{\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}\}$ is defined by the satellite's instantaneous position $\mathbf{r}$ and velocity $\mathbf{v}$ in the inertial frame:

- **Origin (O_{Hill}):** Located at the center of mass of the reference satellite (often referred to as the "Chief" or "Target").
- **Radial Axis ($\hat{\mathbf{x}}$):** This axis is aligned with the local vertical, pointing outward from the Earth's center through the satellite. Mathematically, it is defined by the normalized position vector:

$$\hat{\mathbf{x}} = \frac{\mathbf{r}}{|\mathbf{r}|}$$

- **Cross-Track Axis ($\hat{\mathbf{z}}$):** This axis is normal to the orbital plane, aligned with the specific angular momentum vector of the orbit. It points in the direction of the orbit pole:

$$\hat{\mathbf{z}} = \frac{\mathbf{r} \times \mathbf{v}}{|\mathbf{r} \times \mathbf{v}|}$$

- **Along-Track Axis ($\hat{\mathbf{y}}$):** The y-axis completes the right-handed orthogonal triad ($\hat{\mathbf{y}} = \hat{\mathbf{z}} \times \hat{\mathbf{x}}$). It points in the direction of the transverse motion.

$$\hat{\mathbf{y}} = \hat{\mathbf{z}} \times \hat{\mathbf{x}}$$

Note: For a perfectly circular orbit, the $\hat{\mathbf{y}}$ axis aligns exactly with the velocity vector $\mathbf{V}$.

However, in an elliptical orbit (as depicted in Figure 2.2), the velocity vector deviates from $\hat{\mathbf{y}}$ by the flight path angle γ . Therefore, $\hat{\mathbf{y}}$ is strictly defined as perpendicular to the radius vector, rather than tangent to the path.

Keplerian Orbital Elements

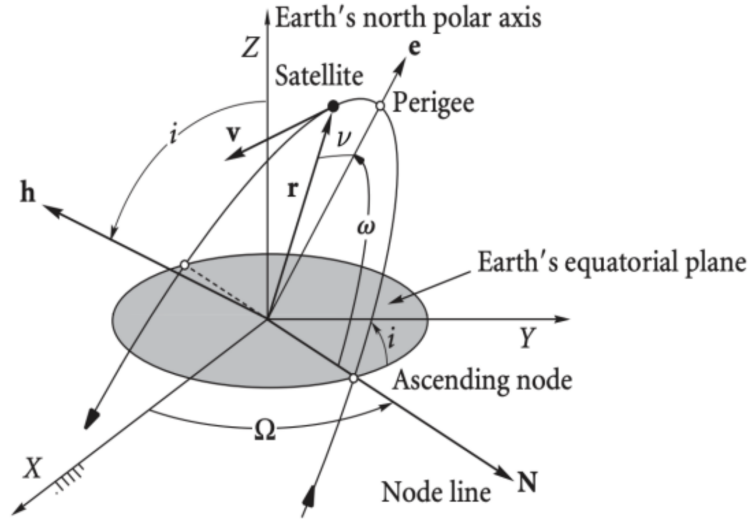


Figure 2.3: Orbital elements definition, adapted from [22]

While the Cartesian state vector $(\mathbf{r}, \mathbf{v})$ provides a complete instantaneous description of a satellite's motion, it offers little intuitive insight into the orbit's geometry or long-term behavior. A more fundamental parametrization is provided by the set of six Keplerian Orbital Elements. These elements describe the size, shape and orientation of the orbit in inertial space, as well as the satellite's position along it.

The classical set is defined as $\mathbf{e}_{kep} = [a, e, i, \Omega, \omega, \nu]^T$, as illustrated in Figure 2.3, where:

- **Semi-major Axis (a):** Defines the size of the orbit and directly determines the orbital energy and period.
- **Eccentricity (e):** Defines the shape of the orbit (0 for circular, $0 < e < 1$ for elliptical).
- **Inclination (i):** The tilt of the orbital plane relative to the Earth's equatorial plane.

- **Right Ascension of the Ascending Node (Ω):** The angle measured in the equatorial plane from the vernal equinox to the ascending node.
- **Argument of Perigee (ω):** The angle measured in the orbital plane from the ascending node to the **perigee**. The perigee is the point in the orbit where the satellite makes its closest approach to the Earth, corresponding to the maximum orbital velocity.
- **True Anomaly (ν):** The angle defining the satellite's specific position in the orbit, measured from the perigee.

While the True Anomaly ν is geometrically intuitive, it varies non-linearly with time due to the conservation of angular momentum. For analytical derivations and propagation, it is often advantageous to use the **Mean Anomaly (M)**. Physically, M represents the angular position of a fictitious body moving in a circular orbit with the same period as the actual satellite, offering the mathematical advantage of growing linearly with time ($M(t) = M_0 + n(t - t_0)$).

However, the classical set suffers from geometric singularities in specific regimes. Notably, for near-circular orbits ($e \approx 0$) typical of LEO operations, the position of the perigee becomes undefined. Consequently, the Argument of Perigee (ω) and the Anomalies (ν or M) become singular or numerically unstable, as the reference point for these angles vanishes. To address this, the **Mean Argument of Latitude ($\tilde{u}$)** is adopted as the robust angular coordinate. Defined as $\tilde{u} = \omega + M$, it measures the angle of the satellite from the Ascending Node rather than the perigee. This parameter remains well-defined and stable even as the eccentricity approaches zero, making it the preferred parametrization for the Relative Orbital Elements (ROE) formulation utilized in this thesis.

2.2. Orbital Propagation

To precisely simulate the Chaser and Target satellites for the evaluation of the control algorithm, a nonlinear orbital propagator is employed. The propagator is based on the classical Gauss planetary equations and includes the effects of the Earth's oblateness (J_2) and atmospheric drag, third-body perturbations and solar radiation pressure are neglected, as their effects are negligible over the rendezvous time horizon considered.

Gauss's Planetary Equations

Instead of propagating the trajectory using the classical Cartesian formulation (Cowell's method), the system dynamics are formulated using Gauss's Planetary Equations. This

method describes the motion using the set of six classical orbital elements, which provides a more intuitive understanding of the orbital geometry and is less sensitive to integration errors over long durations.

The state vector is defined by the classical orbital elements $[a, e, i, \Omega, \omega, \nu]^T$, described before.

Let the control acceleration vector $\mathbf{u}$, defined in the previously described Hill reference frame, be denoted as $\mathbf{u} = [u_x, u_y, u_z]^T$. Mapping these components to the Radial (f_r), Transverse ($f_\perp$) and Normal (f_h) directions required for Gauss's formulation:

$$f_r = u_x, \quad f_\perp = u_y, \quad f_h = u_z$$

The time evolution of the state vector is governed by the following system of nonlinear differential equations:

$$\begin{aligned} \frac{da}{dt} &= \frac{2a^2}{h} \left(e \sin \nu \cdot f_r + \frac{p}{r} \cdot f_\perp \right) \\ \frac{de}{dt} &= \frac{1}{h} (p \sin \nu \cdot f_r + [(p+r) \cos \nu + re] \cdot f_\perp) \\ \frac{di}{dt} &= \frac{r \cos(\omega + \nu)}{h} \cdot f_h \\ \frac{d\Omega}{dt} &= \frac{r \sin(\omega + \nu)}{h \sin i} \cdot f_h \\ \frac{d\omega}{dt} &= \frac{1}{he} (-p \cos \nu \cdot f_r + (p+r) \sin \nu \cdot f_\perp) - \frac{r \sin(\omega + \nu) \cos i}{h \sin i} \cdot f_h \\ \frac{d\nu}{dt} &= \frac{h}{r^2} + \frac{1}{he} (p \cos \nu \cdot f_r - (p+r) \sin \nu \cdot f_\perp) \end{aligned} \tag{2.1}$$

where the auxiliary variables are defined as:

- $p = a(1 - e^2)$: Semi-latus rectum
- $r = p/(1 + e \cos \nu)$: Orbital radius magnitude
- $h = \sqrt{\mu p}$: Specific angular momentum

It is important to note that the equation for the True Anomaly (ν) includes the natural motion term (h/r^2) in addition to the perturbation terms. This formulation is valid for the Chaser's elliptic orbit where $e > 0$.

J_2 Perturbation

The Earth is not a perfect sphere but rather an oblate spheroid. This lack of symmetry means that the gravitational force acting on an orbiting body is not directed strictly towards the center of the Earth. While the gravitational field of a point mass depends only on the distance from the center, the Earth's oblateness introduces a variation dependent on latitude (the angular distance from the equator). This is known as a zonal harmonic variation.

This effect is mathematically described through an infinite series of spherical harmonics, where the J_2 coefficient represents the dominant term after the primary gravitational mass.

Following the formulation in Vallado [23], the perturbing accelerations caused by J_2 can be expressed in the Radial (r), Transverse ($\perp$) and Normal (h) directions as follows:

$$\begin{aligned} a_r &= -\frac{3}{2} \frac{J_2 \mu R_e^2}{r^4} (1 - 3 \sin^2 i \sin^2(\omega + \nu)) \\ a_{\perp} &= -\frac{3}{2} \frac{J_2 \mu R_e^2}{r^4} (\sin^2 i \sin(2\omega + 2\nu)) \\ a_h &= -\frac{3}{2} \frac{J_2 \mu R_e^2}{r^4} (\sin 2i \sin(\omega + \nu)) \end{aligned}$$

These components form the disturbance vector:

$$\mathbf{a}_{J_2} = \begin{bmatrix} a_r \\ a_{\perp} \\ a_h \end{bmatrix} \quad (2.2)$$

This perturbation has significant effects on the orbital elements over time. Most notably, it induces a secular variation (drift) in the Right Ascension of the Ascending Node (RAAN). The instantaneous rate of change of Ω due to the normal component of the perturbation is given by:

$$\dot{\Omega} = \frac{r \sin(\omega + \nu)}{h \sin i} a_h$$

When averaged over an orbital period, this results in the phenomenon known as nodal regression, where the orbital plane rotates around the Earth's polar axis.

Atmospheric Drag

Second only to the Earth's oblateness (J_2), atmospheric drag is the most significant perturbation acting on a satellite in Low Earth Orbit (LEO). Physically, drag arises from the collision of atmospheric particles with the spacecraft surface, creating a non-conservative force that dissipates orbital energy. This effect primarily manifests as a secular decay in the semi-major axis and eccentricity.

Following the formulation in Vallado [23], the acceleration vector due to atmospheric drag is modeled as:

$$\mathbf{a}_{drag} = -\frac{1}{2}\rho_a \frac{C_D A}{m} |\mathbf{v}_{rel}| \mathbf{v}_{rel} \quad (2.3)$$

where:

- ρ_a is the local atmospheric density.
- C_D is the drag coefficient, a dimensionless parameter dependent on the satellite's shape, surface material and angle of attack.
- A is the cross-sectional area projected in the direction of motion.
- m is the satellite mass.
- $\mathbf{v}_{rel}$ is the velocity vector of the satellite relative to the rotating atmosphere.

Modeling the atmospheric density ρ is complex due to its dependence on multiple dynamic factors, including solar flux, geomagnetic activity, diurnal variations and altitude. For the purpose of this simulation, a constant atmospheric density corresponding to the nominal orbital altitude is assumed. Similarly, C_D is approximated as a constant (typically between 2.0 and 2.2).

Equation 2.3 is naturally defined in the inertial (ECI) frame. To incorporate this perturbation into the relative dynamics (e.g., Equation 2.2), it must be transformed into the Hill (RTN) frame.

First, the atmosphere is assumed to rotate rigidly with the Earth so the relative velocity vector in the ECI frame is computed by accounting for the Earth's rotation:

$$\mathbf{v}_{rel} = \mathbf{v}_{sat} - \boldsymbol{\omega}_e \times \mathbf{r}_{sat}$$

where $\mathbf{r}_{sat}$ and $\mathbf{v}_{sat}$ are the satellite's inertial position and velocity and $\boldsymbol{\omega}_e = [0, 0, \omega_\oplus]^T$ is the Earth's angular velocity vector.

The transformation from ECI to RTN is achieved using the basis vectors defined in Section 2.1. Constructing the rotation matrix $\mathbf{R}_{RTN}^{ECI}$ using the inertial basis vectors as columns:

$$\mathbf{R}_{RTN}^{ECI} = [\hat{\mathbf{x}} \mid \hat{\mathbf{y}} \mid \hat{\mathbf{z}}]$$

The drag acceleration in the Hill (RTN) frame is then obtained by applying the transpose of this matrix (since the rotation matrix is orthogonal, $\mathbf{R}^{-1} = \mathbf{R}^T$):

$$\mathbf{a}_{drag}^{RTN} = (\mathbf{R}_{RTN}^{ECI})^T \cdot \mathbf{a}_{drag}^{ECI}$$

Propagator Implementation

The complete propagator numerically integrates the Gauss Planetary Equations defined in Equation (2.1). At each integration time step, the net perturbing acceleration is computed by summing the active control inputs, the J_2 gravitational perturbation and the atmospheric drag.

Crucially, all vectors must be expressed in the Hill (RTN) frame before summation:

$$\mathbf{f} = \begin{bmatrix} f_r \\ f_{\perp} \\ f_h \end{bmatrix} = \mathbf{u} + \mathbf{a}_{J_2} + \mathbf{a}_{drag}^{RTN}$$

2.3. Satellite Modeling

Chaser Satellite

The Chaser spacecraft is modeled as a rigid body with a dry mass of $m_{dry} = 405$ kg. The propellant mass allocated for the rendezvous maneuver is treated as an operational constraint within the optimal control formulation. To enable the high agility required by the time-optimal rendezvous algorithm, the propulsion system utilizes a standard orthogonal configuration. Six fixed chemical thrusters are aligned with the principal body axes ($\pm x_b, \pm y_b, \pm z_b$), providing fully decoupled, holonomic control authority in all three translational directions.

The thrusters are characterized by a specific impulse of $I_{sp} = 236$ s. Each thruster is capable of delivering a maximum thrust of $T_{max} = 275$ N, subject to hard saturation constraints ($0 \leq T \leq T_{max}$). The propellant consumption is governed by the standard relationship between the instantaneous thrust magnitude T and the mass flow rate $\dot{m}$:

$$\dot{m} = -\frac{T}{I_{sp} \cdot g_0}$$

where $g_0 = 9.81 \text{ m/s}^2$ is the standard gravitational acceleration. The physical propulsion parameters and navigation thresholds used in the simulation are summarized in Table 2.1.

Parameter	Value	Unit
Dry Mass (m_{dry})	405	kg
Max Thrust (T_{max})	275	N
Specific Impulse (I_{sp})	236	s
Sensor Range Threshold (ρ^*)	10	km

Table 2.1: Chaser Satellite Physical Parameters

Navigation and Sensor Handover The chaser employs a hybrid navigation architecture depending on the relative distance ρ to the target:

- **Far Range** ($\rho \geq \rho^*$): The chaser relies on an Inertial Navigation System (INS). In this phase, relative measurements are unavailable and the state estimate is subject to drift accumulation due to sensor biases.
- **Close Range** ($\rho < \rho^*$): When the relative distance falls below the threshold $\rho^* = 10 \text{ km}$, the relative navigation sensors become active. This allows for precise relative state estimation and thrust error compensation.

The initial relative orbital conditions defining the start of the rendezvous maneuver are detailed in Table 2.2.

Relative Keplerian Parameter	Value	Unit
Semi-major Axis Difference (Δa)	-1.2	km
Eccentricity Difference (Δe)	0.0008	-
Inclination Difference (Δi)	0.08	deg
True Anomaly Difference ($\Delta \nu$)	-0.45	deg

Table 2.2: Initial Relative Orbital Conditions

Target Satellite

The target selected for this interception scenario is the **Cosmos 2506**, a Russian electro-optical reconnaissance satellite of the Persona class (3rd generation). Launched in June 2015, this spacecraft represents a high-value, non-cooperative military asset operating in a Sun-Synchronous Orbit (SSO).

Unlike a cooperative docking target (e.g., the ISS), Cosmos 2506 is classified as a “heavy” non-cooperative object. For the purpose of the Guidance, Navigation and Control (GNC) design, the target is modeled as a passive rigid body subject to natural orbital perturbations but without active attitude or orbit control.

The specific orbital parameters used for simulation correspond to the satellite’s operational epoch (refer to Table 2.4). The orbit is characterized by a mean altitude of approximately 711 km and an inclination of 98.1° , ensuring a frozen nodal regression rate that maintains constant solar illumination conditions typical for optical spy satellites.

From a safety perspective, the physical dimensions of a Persona-class satellite necessitate a conservative **Keep-Out Zone (KOZ)**. Consequently, the collision avoidance radius ρ_{safe} in the final approach phase is sized to encompass the target’s estimated structural volume plus a safety margin for solar panel appendages. The estimated physical and mission parameters are detailed in Table 2.3.

Parameter	Value	Note/Unit
Satellite Class	Persona	3rd Gen.
NORAD ID	40699	-
Orbit Type	SSO	-
Mean Altitude	711	km
Estimated Mass	$\approx 6,500$	kg
Dimensions	7.0×2.7	m
Status	Non-Cooperative	Tumbling
Perturbations	J_2 Dominant	-

Table 2.3: Target Physical and Mission Data (Estimated for Cosmos 2506)

The target’s state vector $[\mathbf{R}_t^T, \mathbf{V}_t^T]^T$ evolves according to the natural orbital mechanics.

Inertial Navigation System Model (INS)

To enhance robustness and ensure continuous navigation during the close-range approach, the chaser satellite implements an Inertial Navigation System (INS) to compensate for acceleration mismatch. While a Global Navigation Satellite System (GNSS) offers superior

Orbital Element	Value	Unit
Semi-major Axis (a)	7089	km
Eccentricity (e)	$6.017 \cdot 10^{-4}$	-
Inclination (i)	98.0877	deg
RAAN (Ω)	314.59	deg
Argument of Perigee (ω)	250.39	deg
True Anomaly (ν)	0	deg

Table 2.4: Target Orbit Parameters (Initial Conditions)

absolute positioning accuracy, it may be subject to signal occultation or jamming during proximity operations. Conversely, while the INS provides continuity, it is intrinsically limited in orbital environments.

It is critical to note that accelerometers measure specific force rather than total kinematic acceleration. In a free-falling orbital environment, the specific force corresponds only to non-gravitational perturbations (e.g., atmospheric drag, solar radiation pressure and thruster firings). The sensor cannot directly measure gravitational acceleration; consequently, the INS must rely on an internal mathematical model to estimate gravity. This reliance, combined with the integration of sensor errors, leads to a rapid divergence in the navigation solution compared to GNSS.

The INS model is derived from the performance characteristics of a space-grade IMU, specifically the Honeywell QA-2000. The accelerometer sampling frequency was set to $f_c = 50$ Hz. The bias was modeled with a magnitude of $10 \mu g$ to represent the in-run residual error after on-orbit calibration, which is a significant improvement over the raw turn-on repeatability of $< 160 \mu g$. The sensor's intrinsic noise was modeled based on the manufacturer's specification of $< 70 \mu g_{\text{RMS}}$ over the frequency range $[10 - 500]$ Hz.

The measurement model is implemented as follows:

$$a_i^* = a_i + b_{res} + \xi(t) \quad \forall i \in \{\hat{x}, \hat{y}, \hat{z}\}$$

where a_i^* is the measured specific force, a_i is the true non-gravitational acceleration derived from the high-fidelity propagator, b_{res} is the residual sensor bias and $\xi(t)$ is the discrete sensor noise drawn from a Gaussian distribution $\xi(t) \sim \mathcal{N}(0, \sigma^2)$.

To compute the variance σ^2 , the spectral noise density (N_0) is first derived from the RMS

noise specification and the sensor bandwidth ($BW \approx 500$ Hz):

$$N_0 = \frac{70}{\sqrt{500}} \approx 3.13 \text{ } [\mu\text{g}/\sqrt{\text{Hz}}]$$

The discrete noise variance for the simulation step is then calculated as:

$$\sigma^2 = \left(N_0 \cdot g_0 \cdot 10^{-6} \cdot \sqrt{f_c} \right)^2$$

where g_0 is standard gravity (9.81 m/s^2).

The measured acceleration a_i^* is integrated via the Clohessy–Wiltshire (CW) equations to estimate the relative position. A simulation comparing the trajectory derived from true acceleration against the measured acceleration reveals a position error of approximately 120 m after 1000 s. The dominant source of this divergence is the residual sensor bias of $10 \text{ } \mu\text{g}$.

Failure Scenarios

In this thesis, potential failure scenarios are analyzed to assess the robustness of the proposed controller. The following three critical failure modes are considered:

- **Stuck-Open Fault:** A specific thruster remains permanently active at maximum thrust levels, regardless of the control command. This results in a continuous, unintended perturbation force acting on the satellite.
- **Blocked-Closed Fault:** A thruster fails to activate despite a command being issued. This results in a partial loss of actuation authority, effectively reducing the control input to zero in the affected direction.
- **Navigation Failure:** The primary relative navigation sensors fail, depriving the Chaser of direct measurements of the Target’s position. In this scenario, the guidance system must rely solely on the Inertial Navigation System (INS).

3 | Relative Motion Dynamics

Having defined the high-fidelity inertial environment and the physical spacecraft characteristics in Chapter 2, it is necessary to translate these absolute states into a formulation suitable for proximity operations. While the nonlinear Gauss equations provide the “ground truth” for simulation, they are ill-suited for onboard guidance and control; computing relative motion by subtracting large inertial position vectors is not only numerically prone to errors but also fails to provide intuitive insight into the specific geometry of the rendezvous maneuver.

This chapter derives the mathematical models used to describe the motion of the Chaser relative to the Target. The primary motivation for introducing two distinct formulations, Cartesian coordinates and Relative Orbital Elements, used in the hierarchical control architecture proposed in this thesis. The Clohessy-Wiltshire (CW) formulation offers a direct link to relative position and velocity, making it ideal for the short-range feedback loops in the Tracking Layer. Conversely, the Relative Orbital Elements (ROE) formulation exploits the geometric properties of Keplerian motion to provide a slowly varying state representation.

The chapter is organized as follows:

- **Section 3.1** derives the Clohessy-Wiltshire (CW) equations, establishing the linear Cartesian state-space model used for short-range tracking.
- **Section 3.2** introduces the Quasi-Nonsingular Relative Orbital Elements (ROE), defining the geometric state representation used for trajectory optimization.
- **Section 3.3** performs a comparative analysis of the linearization errors inherent in both models, justifying the selection of ROE for long-duration propagation and CW for instantaneous control.

3.1. Clohessy-Wiltshire (CW) Equations

The mathematical foundation for relative motion was first established by George Hill in 1878 to describe the motion of the Moon relative to the Earth. However, the formulation gained significant prominence in the astro-dynamics community in 1960, when W.H. Clohessy and R.S. Wiltshire published their paper in the *Journal of the Aerospace Sciences* [3]. Their work was specifically motivated by the need to develop rendezvous techniques for the Gemini and Apollo programs. They derived a simplified set of equations that allowed for the analytic solution of the relative trajectory, enabling early onboard computers to compute rendezvous burns in real-time.

Derivation and Assumptions

The CW equations are derived by linearizing the nonlinear equations of relative motion about a circular reference orbit. The validity of this model relies on several key assumptions:

1. The Target satellite is in a circular orbit ($e_t = 0$).
2. The separation distance ρ between the Chaser and Target is small compared to the orbital radius r ($\rho \ll r$).
3. No external perturbations (e.g., J_2 or drag) are acting on the system; only central body gravity is considered.

Under these assumptions, the decoupled motion in the Cross-Track (z) direction and the coupled motion in the Radial (x) and Along-Track (y) directions are described by the following linear differential equations:

$$\begin{aligned} \ddot{x} - 2n\dot{y} - 3n^2x &= u_x \\ \ddot{y} + 2n\dot{x} &= u_y \\ \ddot{z} + n^2z &= u_z \end{aligned} \tag{3.1}$$

where $n = \sqrt{\mu/a^3}$ is the mean motion (angular velocity) of the Target satellite and u_x, u_y, u_z are the control accelerations (force per unit mass) applied by the Chaser's thrusters.

State-Space Representation

For the purpose of control system design, it is convenient to express Eq. 3.1 in the continuous state-space form $\dot{\mathbf{x}} = \mathbf{A}_{CW}\mathbf{x} + \mathbf{B}_{CW}\mathbf{u}$. Defining the state vector $\mathbf{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T$, the system matrices are:

$$\mathbf{A}_{CW} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 0 & 2n & 0 \\ 0 & 0 & 0 & -2n & 0 & 0 \\ 0 & 0 & -n^2 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B}_{CW} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

- Coriolis terms: $\pm 2n$
- Gravity gradient terms: $3n^2, -n^2$

LTI Stability and Analysis

It can be observed that the structure of this Linear Time-Invariant (LTI) system is relatively straightforward. The first three rows of the state matrix $\mathbf{A}$ represent pure kinematic relationships (velocity to position integration), while the control inputs act directly on the acceleration terms in the last three rows.

A key property of the CW formulation is that the system dynamics can be decoupled into two independent subsystems:

- **In-Plane Dynamics:** Governed by the state vector $[x, y, \dot{x}, \dot{y}]^T$, representing the coupled radial and along-track motion.
- **Out-of-Plane Dynamics:** Governed by the state vector $[z, \dot{z}]^T$, representing the simple harmonic motion in the cross-track direction.

This decoupling property simplifies the control design and is explicitly exploited in Chapter 5.

Analyzing the stability properties, the eigenvalues of the system matrix $\mathbf{A}$ are:

$$\lambda(\mathbf{A}_{CW}) = \{0, 0, \pm ni, \pm ni\}$$

The presence of purely imaginary eigenvalues ($\pm ni$) indicates the existence of oscillatory modes that are marginally stable, exhibiting bounded periodic motion in the absence of

damping. However, the repeated eigenvalues at the origin imply the existence of secular modes associated with nontrivial Jordan blocks. As a result, the CW system is not asymptotically stable and is Lyapunov unstable due to the presence of linearly growing solutions.

Furthermore, the repeated eigenvalues at the origin ($\lambda = 0$) correspond to a secular mode. Physically, this represents the potential for unbounded drift in the along-track direction (y) if the initial relative velocity does not perfectly match the constraint required for a bounded orbit (specifically, $\dot{y}_0 = -2nx_0$).

It is important to recognize that the CW formulation is obtained by truncating the Taylor series expansion of the nonlinear relative dynamics, retaining only the first-order terms. This linearization introduces an inherent modeling error that scales with the separation distance. Therefore, the stability properties derived from the matrix $\mathbf{A}$ are strictly valid only locally, within a small neighborhood of the Target.

A fundamental step is to verify that the LTI system defined in Equation (3.2) is fully controllable. This can be established using the standard rank condition theorem:

Theorem 3.1. Controllability

An LTI system is controllable if and only if the controllability matrix $\mathcal{C}$ has full rank n (where n is the dimension of the state vector). The matrix is defined as:

$$\mathcal{C} \triangleq [\mathbf{B}, \mathbf{AB}, \dots, \mathbf{A}^{n-1}\mathbf{B}]$$

For the fully actuated Chaser satellite, calculating this matrix yields $\text{rank}(\mathcal{C}) = 6$, confirming that the system is completely controllable.

Note: *It is worth noting that the CW system remains controllable even in specific under-actuated configurations. For instance, full controllability in the orbital plane (x, y) can be achieved using thrust only in the Along-Track (y) direction. This is possible due to the strong coupling between radial and along-track motion: a change in orbital velocity (Δv_y) naturally induces a change in altitude (x) over time. However, the Chaser's 6-thruster configuration exceeds these minimum requirements, providing direct authority in all axes.*

3.2. Relative Orbital Elements (ROE)

While the Clohessy-Wiltshire (CW) equations provide an intuitive Cartesian representation of relative motion, they suffer from significant limitations when applied to non-circular orbits or large separation distances. To overcome these drawbacks, the relative dynamics

can alternatively be described using Relative Orbital Elements (ROE).

Fundamentally, ROEs represent the state of the Chaser by examining the difference between its orbital elements and those of the Target. Instead of tracking the separation in meters (Radial, Along-Track, Cross-Track), this formulation tracks the geometric differences in the orbit shape and orientation.

The primary advantage of this approach lies in the geometry of Keplerian motion. In the absence of perturbations, orbital elements are constants of motion. Consequently, a relative state defined by element differences is naturally slowly varying or constant, unlike Cartesian coordinates which oscillate rapidly at the orbital frequency. This makes ROEs particularly powerful for:

- **Trajectory Safety:** Passive safety can be guaranteed by separating the eccentricity and inclination vectors.
- **Long-term Stability:** Matching the semi-major axis ($\delta a = 0$) guarantees zero drift, a condition that is numerically sensitive in Cartesian coordinates.

Seminal work by Alfrend and Schaub [6] formalized the nonlinear mapping between Cartesian states and element differences. Later, D'Amico and Montenbruck [7] introduced the "quasi-nonsingular" formulation of ROEs utilized in this thesis. This specific set of elements avoids singularities for near-circular orbits and decouples the in-plane and out-of-plane relative motion, effectively combining the intuitive benefits of the Hill frame with the geometric robustness of orbital elements. This modern formulation has since become the standard for relative guidance in distributed space systems.

Derivation and Assumptions

To derive the set of Relative Orbital Elements (ROE), we start from the classical Keplerian elements $\mathbf{e}_{kep} = [a, e, i, \Omega, \omega, \nu]^T$. For near-circular orbits ($e \approx 0$) typical of LEO, the argument of perigee ω becomes ill-defined, so it is convenient to use a quasi-nonsingular representation.

The non-singular state vector for a single satellite is defined as

$$\boldsymbol{\alpha} = \begin{bmatrix} a & \tilde{u} & e_x & e_y & i & \Omega \end{bmatrix}^T$$

where $\tilde{u} = M + \omega$ is the argument of latitude and the eccentricity vector components are defined as $e_x = e \cos \omega$ and $e_y = e \sin \omega$.

The ROE state vector, denoted as $\delta \boldsymbol{\alpha}$, is defined as the first-order differences between the

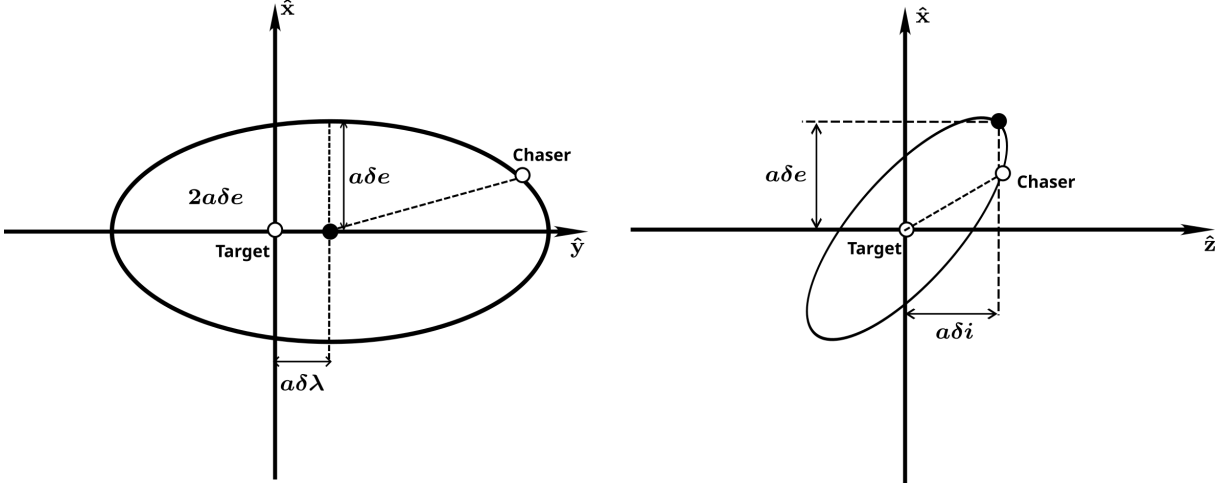


Figure 3.1: Projections of bounded relative motion in the along-track/radial (y/x) (left) and cross-track/radial (z/x) (right) directions for arbitrary relative orbital elements, adapted from [7]

Chaser (subscript c) and Target (subscript t) orbital elements. For the specific case of a near-circular target orbit and small relative separations, the standard dimensionless ROE set is

$$\delta\alpha = \begin{pmatrix} \delta a \\ \delta\lambda \\ \delta e_x \\ \delta e_y \\ \delta i_x \\ \delta i_y \end{pmatrix} = \begin{pmatrix} (a_c - a_t)/a_t \\ (\tilde{u}_c - \tilde{u}_t) + (\Omega_c - \Omega_t) \cos i_t \\ e_c \cos \omega_c - e_t \cos \omega_t \\ e_c \sin \omega_c - e_t \sin \omega_t \\ i_c - i_t \\ (\Omega_c - \Omega_t) \sin i_t \end{pmatrix} \quad (3.3)$$

Here, δa represents the normalized semi-major axis difference, $\delta\lambda$ is the relative mean longitude and the vectors $\delta\mathbf{e} = [\delta e_x, \delta e_y]^T$ and $\delta\mathbf{i} = [\delta i_x, \delta i_y]^T$ represent the relative eccentricity and inclination vectors, respectively.

Assumptions

The validity and utility of this formulation in the context of the proposed control algorithm rely on the following assumptions:

- **Near-Circular Target Orbit:** The derivation of the relative dynamics matrices assumes that the Target's eccentricity is small ($e_t \ll 1$). While the geometric definition in Equation (3.3) is exact, the linear mapping to Cartesian states degrades if e_t is large.
- **Small Relative Separations:** Although ROEs are more robust than Cartesian

coordinates at distance, the mapping between $\delta\boldsymbol{\alpha}$ and the local Hill frame positions (detailed in Appendix A.1) assumes that the separation distance is small compared to the orbital radius ($\rho \ll r$).

State-Space Representation

Similar to the CW formulation, the dynamics of the quasi-nonsingular Relative Orbital Elements can be expressed in a continuous state-space form:

$$\delta\dot{\boldsymbol{\alpha}} = \mathbf{A}_{ROE}\delta\boldsymbol{\alpha} + \mathbf{B}_{ROE}(\tilde{u})\mathbf{u} \quad (3.4)$$

where $\mathbf{u} = [u_x, u_y, u_z]^T$ represents the control accelerations in the Radial, Transverse and Normal Hill frame directions.

For unperturbed Keplerian motion, the state matrix $\mathbf{A}_{ROE}$ is constant and sparse, capturing only the relative drift in mean longitude caused by the difference in semi-major axis (and consequently, mean motion):

$$\mathbf{A}_{ROE} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -1.5n & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The control input matrix $\mathbf{B}_{ROE}(\tilde{u})$ is **time-varying**, depending on the instantaneous mean argument of latitude $\tilde{u}(t)$. The matrix is given by:

$$\mathbf{B}_{ROE}(\tilde{u}) = \frac{1}{na} \begin{bmatrix} 0 & 2 & 0 \\ -2 & 0 & 0 \\ \sin \tilde{u} & 2 \cos \tilde{u} & 0 \\ -\cos \tilde{u} & 2 \sin \tilde{u} & 0 \\ 0 & 0 & \cos \tilde{u} \\ 0 & 0 & \sin \tilde{u} \end{bmatrix}$$

where n is the target's mean motion and a is the target's semi-major axis.

Consistent with the dynamic decoupling observed in the CW equations, the ROE for-

mulation maintains the separation between planar and out-of-plane motion. The relative inclination vector $\delta \mathbf{i}$ is driven exclusively by normal thrust (u_z), while the in-plane elements ($\delta a, \delta \lambda, \delta \mathbf{e}$) evolve solely according to radial (u_x) and transverse (u_y) inputs.

LTV Stability and Analysis

Inspecting the state matrix $\mathbf{A}_{ROE}$, it is observed that the system possesses a very sparse structure with a single non-zero off-diagonal term. All eigenvalues of $\mathbf{A}_{ROE}$ are equal to zero. Due to the presence of a nontrivial Jordan block, the system is Lyapunov unstable and exhibits secular drift in the relative mean longitude. Physically, this corresponds to the natural Keplerian dynamics: if there is an initial difference in the semi-major axis ($\delta a \neq 0$), a secular drift in the relative mean longitude ($\delta \lambda$) will accumulate linearly over time. In the absence of control, any state with $\delta a = 0$ is an equilibrium point, while states with $\delta a \neq 0$ exhibit unbounded drift.

Similar to the CW equations, this ROE model is a linearization of the nonlinear relative dynamics. Consequently, the stability analysis is strictly valid only within a small neighborhood of the reference orbit where the linearization assumptions hold.

Unlike the CW formulation, the control input matrix $\mathbf{B}_{ROE}(t)$ in the ROE state-space model is time-varying. Therefore, the standard rank condition for LTI systems (Theorem 3.1) cannot be directly applied. Instead, the controllability must be assessed using the Controllability Gramian.

Theorem 3.2. Controllability of LTV Systems

Consider the Linear Time-Varying (LTV) system:

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t)$$

defined on the interval $t \in [t_0, t_f]$. The system is controllable on this interval if and only if the Controllability Gramian $\mathbf{W}_c(t_0, t_f)$ is positive definite (i.e., all its eigenvalues are strictly positive). The Gramian is defined as:

$$\mathbf{W}_c(t_0, t_f) = \int_{t_0}^{t_f} \Phi(t_f, \tau) \mathbf{B}(\tau) \mathbf{B}(\tau)^T \Phi(t_f, \tau)^T d\tau$$

where $\Phi(t, \tau)$ denotes the State Transition Matrix (STM) of the homogeneous system.

Since the ROE system is periodic with the orbital period $T = 2\pi/n$, the controllability is analyzed over the interval $[t_0, t_0 + T]$.

Although the system is LTV due to $\mathbf{B}_{ROE}(t)$, the state matrix $\mathbf{A}_{ROE}$ is constant for the unperturbed case. This simplifies the computation of the State Transition Matrix, which admits the closed-form exponential solution:

$$\Phi(t, \tau) = e^{\mathbf{A}(t-\tau)}$$

Substituting this into the Gramian definition yields:

$$\mathbf{W}_c(t_0, t_0 + T) = \int_{t_0}^{t_0+T} e^{\mathbf{A}(t_0+T-\tau)} \mathbf{B}(\tau) \mathbf{B}(\tau)^T e^{\mathbf{A}^T(t_0+T-\tau)} d\tau$$

Evaluating this integral confirms that $\mathbf{W}_c(t_0, t_0 + T)$ is symmetric positive definite (has full rank). This result proves that the system is fully controllable over the duration of one orbital period, meaning any relative orbital element configuration can be achieved given sufficient time and thrust authority.

3.3. Modeling Error and Linearity Comparison

Both the Clohessy-Wiltshire (CW) equations and the linearized Relative Orbital Elements (ROE) formulation rely on the fundamental assumption that the separation distance ρ is significantly smaller than the orbital semi-major axis a (i.e., $\rho/a \ll 1$). Consequently, both models are first-order approximations obtained by truncating the Taylor series expansion of the nonlinear relative motion dynamics.

Order of Accuracy

A rigorous analysis of the relative motion geometry reveals that both formulations share the same order of linearization error. Specifically, the neglected nonlinear terms scale with the square of the normalized separation distance. Thus, the modeling error ε for both CW and linear ROE is of the order:

$$\varepsilon \propto \mathcal{O}\left(\left(\frac{\rho}{a}\right)^2\right) \quad (3.5)$$

This implies that as the separation distance increases, the accuracy of both models degrades quadratically. However, Alfrend and Yan demonstrated that despite sharing the same formal error order, the magnitude of the error differs significantly between the two formulations [5].

Linearization Errors

The superior accuracy of the ROE formulation arises from the domain in which the linearization is performed:

- **CW:** The CW equations linearize the differential equations of motion directly in the Cartesian frame. A critical side effect of this linearization is the approximation of the mean motion n . In the nonlinear reality, the orbital period depends on the semi-major axis ($n \propto a^{-3/2}$). The linearized CW model fails to capture this relationship accurately, leading to an artificial secular drift in the along-track direction that grows quadratically with time, even if the initial conditions theoretically satisfy the bounded orbit constraint.
- **ROE:** The ROE formulation linearizes the geometric transformation between the orbital elements and the Cartesian state. As noted by Alfrend and Yan, performing the linearization in the orbital element space results in a smaller modeling error compared to solving the equations of motion in Cartesian space [5]. Because the unperturbed orbital elements are constants of motion, the ROE model preserves the correct period matching condition ($\delta a = 0$). Consequently, it does not suffer from the artificial dynamic drift found in CW and the error remains purely geometric.

To eliminate the $\mathcal{O}((\rho/a)^2)$ error entirely, second-order nonlinear corrections (such as the Yan-Alfrend theory) are required [5]. However, for the operational range considered in this work, the linear ROE formulation provides a sufficiently robust approximation that outperforms the CW equations for long-term propagation.

4 | Control System Design

Building upon the theoretical foundations of relative motion established in Chapter 2, this chapter presents the design of the hybrid Guidance, Navigation and Control (GNC) architecture. While the dynamic models, specifically the Clohessy-Wiltshire equations and Relative Orbital Elements provide the mathematical description of the spacecraft's motion, the primary objective of this thesis requires a control strategy capable of achieve time-optimal interception under operational constraints.

The control problem for non-cooperative rendezvous is characterized by two conflicting requirements: the need for global optimality to minimize maneuver time and the need for a robust control to reject disturbances and modeling errors. A single control layer is often not sufficient to satisfy both simultaneously; optimal control solvers are typically computationally demanding for real-time feedback, while robust linear controllers lack the global look-ahead capability required for trajectory optimization.

To address these challenges, this thesis proposes a **hybrid architecture** that decouples the control problem into two complementary layers:

- **Guidance Layer:** This layer exploits the geometric linearity of the ROE described in Section 3.2 to generate globally time-optimal reference trajectories via convex optimization.
- **Tracking Layer:** This layer utilizes the Clohessy-Wiltshire formulation described in Section 3.1 to design a Sliding Mode Controller (SMC) that robustly tracks the reference trajectory and rejects local disturbances.

This chapter is organized as follows:

- **Section 4.1** details the Guidance Layer, formulating the convex optimization problem used to generate the reference trajectory subject to propellant consumption and actuator saturation constraints.
- **Section 4.2** describes the Tracking Layer, deriving the Sliding Mode Controller (SMC) designed to ensure precise trajectory tracking in the presence of navigation

errors and external perturbations.

- **Section 4.3** introduces the Final Approach Strategy, where the control logic shifts from time-minimization to safety approach. An Artificial Potential Field (APF) method is developed to guide the chaser to the docking port while strictly enforcing collision avoidance.
- **Section 4.4** integrates these components into the complete Hybrid Architecture, defining the supervisory logic that governs the autonomous transitions between long-range rendezvous, close-range tracking and safety-critical operations.

4.1. Optimal Control

The primary objective of the guidance layer is to minimize the rendezvous time while strictly satisfying propellant and actuator constraints. The joint optimization of the control trajectory $\mathbf{u}(t)$ and the final time t_f constitutes a non-convex problem, primarily due to the non-linear dependence of the state evolution on the integration horizon. Consequently, t_f cannot be included directly as a decision variable in a standard convex solver.

To address this, a bi-level optimization strategy is employed that exploits the property of quasiconvexity. The free final time formulation can be treated as a quasiconvex optimization problem, since for any fixed final time t_f , the corresponding feasible set is convex [13]. This property allows the problem to be decomposed into two nested loops:

- **Outer Loop (Time Optimization):** A bisection method (Algorithm 4.1) iterates on the final time t_f to find the minimum horizon for which the problem remains feasible.
- **Inner Loop (Convex Optimization):** For a fixed t_f , the problem is posed as a convex trajectory tracking problem (specifically, an SOCP).

This strategy guarantees convergence to the global minimum time t^* due to two fundamental properties:

1. **Exact Feasibility Detection:** Since the inner loop is convex, the solver is guaranteed to find a feasible trajectory if one exists. Unlike non-convex formulations, there are no local minima to trap the solver into returning a false "infeasible" status.
2. **Monotonicity of the Reachable Set:** The feasibility of the maneuver is monotonic with respect to time, provided the target state $\delta\alpha_f$ represents a relative equilibrium (or a drift-free relative orbit). If the target state satisfies the unforced relative dynamics (i.e., $\mathbf{A}\delta\alpha_f \approx \mathbf{0}$ or can be maintained with valid control), any

trajectory that successfully reaches the target at time t_a can be extended to any later time $t_b > t_a$ by simply maintaining the state. Mathematically, this implies:

$$\text{feasible}(t_a) \implies \text{feasible}(t_b) \quad \forall t_b > t_a$$

This monotonicity ensures that the set of feasible times forms a single, connected interval $[t^*, \infty)$. Therefore, the bisection algorithm is guaranteed to strictly converge to the unique boundary t^* , which represents the global minimum time.

Solver

The optimal control problem is implemented in Matlab using the Yalmip optimization framework together with the Mosek solver. Yalmip provides a high-level modeling interface for defining decision variables, constraints and the cost function, while Mosek performs the actual numerical optimization.

Problem Setup

The optimal control can be written in the Bolza form:

$$\min_{u(\cdot)} J = \phi(t_0, \mathbf{x}_o, t_f, \mathbf{x}_f) + \int_{t_0}^{t_f} L(t, \mathbf{x}, \mathbf{u}) dt$$

subject to:

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

$$\mathbf{x}(t_0) = \mathbf{x}_o$$

$$u \in U$$

$$\Psi(t_0, \mathbf{x}_o, t_f, \mathbf{x}_f) = \mathbf{0}$$

where U is the set of admissible controls and Ψ defines the terminal (boundary) constraints.

Dynamics Constraints

The relative orbital element dynamics, as derived in Section 3.2, are given by:

$$\delta \dot{\boldsymbol{\alpha}} = \mathbf{A}_{ROE} \delta \boldsymbol{\alpha} + \mathbf{B}_{ROE}(\tilde{u}) \mathbf{u} \quad (4.1)$$

where the matrix $\mathbf{A}_{ROE}$ is:

$$\mathbf{A}_{ROE} = \begin{bmatrix} 0 & \mathbf{0}_{1 \times 5} \\ -1.5n & \mathbf{0}_{1 \times 5} \\ \mathbf{0}_{4 \times 1} & \mathbf{0}_{4 \times 5} \end{bmatrix}$$

and the control input matrix $\mathbf{B}_{ROE}(\tilde{u})$ is defined as:

$$\mathbf{B}_{ROE}(\tilde{u}) = \frac{1}{na} \begin{bmatrix} 0 & 2 & 0 \\ -2 & 0 & 0 \\ \sin \tilde{u} & 2 \cos \tilde{u} & 0 \\ -\cos \tilde{u} & 2 \sin \tilde{u} & 0 \\ 0 & 0 & \cos \tilde{u} \\ 0 & 0 & \sin \tilde{u} \end{bmatrix}$$

Computing the input matrix $\mathbf{B}_{ROE}$ requires a priori knowledge of the mean argument of latitude $\tilde{u}(t)$. Assuming the chaser satellite possesses knowledge of the target's orbital elements at the initial epoch t_0 , the evolution of $\tilde{u}$ can be predicted for the entire optimization horizon.

First, the eccentric anomaly E_0 corresponding to the true anomaly ν_0 at epoch t_0 is computed:

$$E_0 = 2 \arctan \left(\sqrt{\frac{1-e}{1+e}} \tan \frac{\nu_0}{2} \right).$$

The mean anomaly at epoch t_0 is then obtained via Kepler's equation:

$$M_0 = E_0 - e \sin E_0.$$

Propagating the mean anomaly to a generic time t yields:

$$M(t) = M_0 + n(t - t_0).$$

Finally, the mean argument of latitude is expressed as:

$$\tilde{u}(t) = \omega + M(t).$$

In the context of the inner optimization loop, the time horizon is fixed to $t \in [t_0, t_{end}]$ with a discrete time step Δt . Since the evolution of $\tilde{u}(t)$ depends only on the target's initial state and not on the control variables, the matrix $\mathbf{B}(\tilde{u})$ is time-varying but deterministic. Therefore, it is implemented as a sequence of precomputed matrices evaluated at each

time step k prior to solving the optimization problem.

Propellant Consumption Constraint

An important aspect of the guidance problem is the modeling of propellant consumption, which depends on the propulsion system specific impulse. In this mission scenario, the spacecraft is equipped with independent thrusters aligned with the body axes. Consequently, the total mass flow rate is the sum of the mass flow rates of the individual thrusters active at any given time:

$$\dot{m}(t) = \sum_{i \in \{x,y,z\}} \dot{m}_i(t) = - \sum_{i \in \{x,y,z\}} \frac{|T_i(t)|}{g_0 I_{sp}},$$

where T_i represents the thrust magnitude along the i -th axis, I_{sp} is the specific impulse (assumed identical for all axes) and g_0 is the standard gravitational acceleration.

Since the thrust is related to the applied acceleration through $T_i(t) = m(t)|a_i(t)|$, the mass flow equation can be rewritten using the L_1 norm of the acceleration vector:

$$\dot{m}(t) = -\frac{m(t)}{g_0 I_{sp}} \sum_{i \in \{x,y,z\}} |a_i(t)| = -\frac{m(t)}{g_0 I_{sp}} \|\mathbf{a}(t)\|_1.$$

In this work, the control input directly represents the commanded acceleration, i.e.,

$$\mathbf{a}(t) \equiv \mathbf{u}(t).$$

The spacecraft mass is assumed constant in the acceleration thrust relationship, while the exact exponential mass depletion is retained for propellant accounting purposes. The mass inside the differential equation can be approximated by its initial value m_0 (conservative assumption). Under this standard approximation, integration yields

$$m(t) = m_0 \exp\left(-\frac{1}{g_0 I_{sp}} \int_{t_0}^t \|\mathbf{u}(\tau)\|_1 d\tau\right),$$

where $m_0 = m_{\text{sat}} + m_{\text{prop}}$ denotes the initial total mass at time t_0 .

The total propellant consumed at the final time t_f is therefore

$$\Delta m(t_f) = m_0 - m(t_f).$$

Discretizing the time interval $[t_0, t_f]$ using an equally spaced grid with time step Δt and

N intervals, the equivalent velocity increment (Delta-V) cost is approximated as

$$\Delta v = \int_{t_0}^{t_f} \|\mathbf{u}(\tau)\|_1 d\tau \approx \sum_{k=0}^{N-1} \|\mathbf{u}_k\|_1 \Delta t.$$

The corresponding discrete mass evolution becomes

$$m_N = m_0 \exp\left(-\frac{1}{g_0 I_{sp}} \sum_{k=0}^{N-1} \|\mathbf{u}_k\|_1 \Delta t\right),$$

and the total propellant consumption constraint can be written as

$$\Delta m_N = m_0 \left(1 - \exp\left(-\frac{1}{g_0 I_{sp}} \sum_{k=0}^{N-1} \|\mathbf{u}_k\|_1 \Delta t\right)\right) \leq m_{\text{prop}}.$$

For uniform time steps, this inequality can be equivalently rearranged as

$$\Delta t \sum_{k=0}^{N-1} \sum_{j=1}^3 |u_{j,k}| \leq g_0 I_{sp} \ln\left(\frac{m_0}{m_0 - m_{\text{prop}}}\right).$$

which admits a direct physical interpretation as a bound on the maximum achievable velocity increment:

$$\Delta V_{\max} = g_0 I_{sp} \ln\left(\frac{m_0}{m_0 - m_{\text{prop}}}\right).$$

This formulation accurately captures the propellant consumption of independently actuated thrusters while preserving convexity. The resulting constraint is linear in the decision variables and can be efficiently enforced within the proposed convex optimization framework, yielding a conservative yet physically meaningful bound on propellant usage.

Control Action Constraints

The maximum admissible acceleration along each axis is physically limited by the available thrust and the instantaneous satellite mass:

$$u_i^{\max}(t) = \frac{T_i^{\max}}{m_{\text{tot}}(t)}.$$

Since the total mass $m_{\text{tot}}(t)$ decreases as propellant is consumed, this constraint is non-convex. To preserve convexity, a conservative approximation is adopted by fixing the mass to its initial value $m(t_0)$. Because $m(t) \leq m(t_0)$, the acceleration computed with

the initial mass is always achievable throughout the maneuver (a stronger constraint):

$$u_i^{\max} = \frac{T_i^{\max}}{m_0}.$$

Additionally, a constraint on the rate of change of acceleration (jerk) can be implemented to model the physical rise time of the thruster (T_r) and reduce structural stress. Assuming the thruster requires T_r seconds to ramp from zero to full authority ($u_{\max}$), the maximum allowable jerk is defined as $J_{\max} = u_{\max}/T_r$. The corresponding discretized constraint is:

$$\frac{|u_i(k+1) - u_i(k)|}{\Delta t} \leq J_{\max}.$$

Note: This constraint is relevant only when the discretization step Δt is comparable to the thruster rise time. In this thesis, the thruster response is faster than the simulation time step ($T_r \ll \Delta t$); therefore, this jerk constraint is not active.

Cost Function & Optimization Algorithm

For the inner fixed-time problem, the cost function minimizes the weighted state error energy to ensure a smooth trajectory towards the target:

$$J = \int_{t_0}^{t_f} \delta \boldsymbol{\alpha}^T \mathbf{Q} \delta \boldsymbol{\alpha} dt$$

Consequently, the inner convex optimal control problem is formally defined as:

$$\min_{\mathbf{u}(\cdot)} J = \int_{t_0}^{t_f} \delta \boldsymbol{\alpha}(\tau)^T \mathbf{Q} \delta \boldsymbol{\alpha}(\tau) d\tau \quad (4.2)$$

subject to

$$\begin{aligned} \delta \dot{\boldsymbol{\alpha}}(t) &= \mathbf{A}_{ROE} \delta \boldsymbol{\alpha}(t) + \mathbf{B}_{ROE}(\tilde{u}) \mathbf{u}(t), & \forall t \in [t_0, t_f] \\ \delta \boldsymbol{\alpha}(t_0) &= \delta \boldsymbol{\alpha}_0, \\ \|\delta \boldsymbol{\alpha}(t_f) - \delta \boldsymbol{\alpha}_f\|_{\infty} &\leq \varepsilon, \\ |u_i(t)| &\leq u_{\max}, & \forall i \in \{x, y, z\} \\ |\dot{u}_i(t)| &\leq \frac{u_{\max}}{T_r}, & \forall i \in \{x, y, z\} \\ \int_{t_0}^{t_f} \|\mathbf{u}(\tau)\|_1 d\tau &\leq \Delta V_{\max}. \end{aligned}$$

Algorithm 4.1 Bisection Optimization Loop

Require: T_{high} {Upper bound on rendezvous time}
Require: T_{low} {Lower bound on rendezvous time}
Require: $iter$ {Maximum number of iterations}

```

1:  $tol \leftarrow 1$ 
2:  $ii \leftarrow 0$ 
3:  $res \leftarrow tol + 1$ 
4: while  $ii < iter \wedge res > tol$  do
5:    $T \leftarrow (T_{\text{high}} + T_{\text{low}})/2$ 
6:    $[feas, x_{\text{sol}}, u_{\text{sol}}, m_{\text{sol}}, t] \leftarrow \text{optimization}(x_0, x_{\text{final}}, T, N, \mathbf{A}, \mathbf{B}, \dots)$ 
7:   if  $feas = 1$  then
8:      $res \leftarrow |T - T_{\text{high}}|$ 
9:      $T_{\text{high}} \leftarrow T$ 
10:  else
11:     $res \leftarrow |T - T_{\text{low}}|$ 
12:     $T_{\text{low}} \leftarrow T$ 
13:  end if
14:   $ii \leftarrow ii + 1$ 
15: end while

```

To enable feasibility detection within the outer time-optimization loop, the terminal state constraint is enforced with a small tolerance:

$$\|\delta\alpha(t_f) - \delta\alpha_f\|_\infty \leq \epsilon,$$

where ϵ is a prescribed tolerance vector. This relaxed hard constraint allows the solver to distinguish between feasible and infeasible final times during the bisection procedure, while still guaranteeing high terminal accuracy.

Solver Behavior Analysis

To illustrate the mechanics of the proposed bi-level optimization strategy, a representative test case is analyzed. This section focuses on visualizing the solver's internal behavior specifically the convergence of the outer time loop and the structure of the resulting control signal, rather than assessing performance in a specific mission scenario (which is addressed in Chapter 6).

Bisection Convergence

The efficiency of the outer loop (Algorithm 4.1) is determined by its ability to rapidly narrow the search interval $[T_{\text{low}}, T_{\text{high}}]$. Figure 4.1 displays the evolution of these time bounds over the iterations. The interval as expected half each iteration converging to

the optimal time t^* , demonstrating the algorithm's stability in detecting the boundary between feasible and infeasible solution spaces.

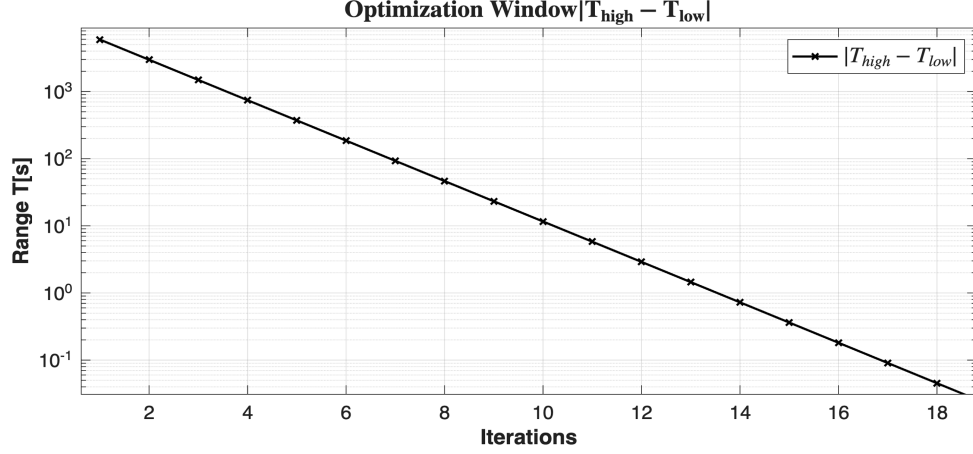


Figure 4.1: Evolution of the time horizon bounds (semilogarithmic scale on y-axis) during the bisection process. The gap $T_{high} - T_{low}$ decreases by half at each iteration, converging to the minimum feasible time.

Control Structure Analysis

The output of the inner optimization loop is governed by the cost function defined in Equation (4.2). By minimizing the rendezvous time, the solver naturally generates a trajectory that maximizes control authority, resulting in a distinct "Bang-Coast-Bang" profile.

As illustrated in Figure 4.2 (top), the acceleration profile exhibits three distinct phases:

1. **Initial Burn** ($t \approx 0 - 100$ s): The actuators saturate at the maximum limit $\|\mathbf{a}_{max}\|$ to instantaneously change the ROE and initiate the transfer.
2. **Coast Phase** ($t \approx 100 - 1450$ s): The control input drops to zero. During this phase, the spacecraft drifts passively under natural orbital dynamics to close the distance to the target.
3. **Final Braking** ($t \approx 1450 - 1640$ s): A final burn is applied to nullify the relative velocity and converge to the target state.

The mass variation shown in Figure 4.2 (bottom) confirms this behavior. A step-wise decrease in mass is observed only during the active thrusting phases. Furthermore, it can be observed that the entire propellant budget of 20 kg has been utilized, confirming that the solver is pushing the system to its performance limits to minimize time.

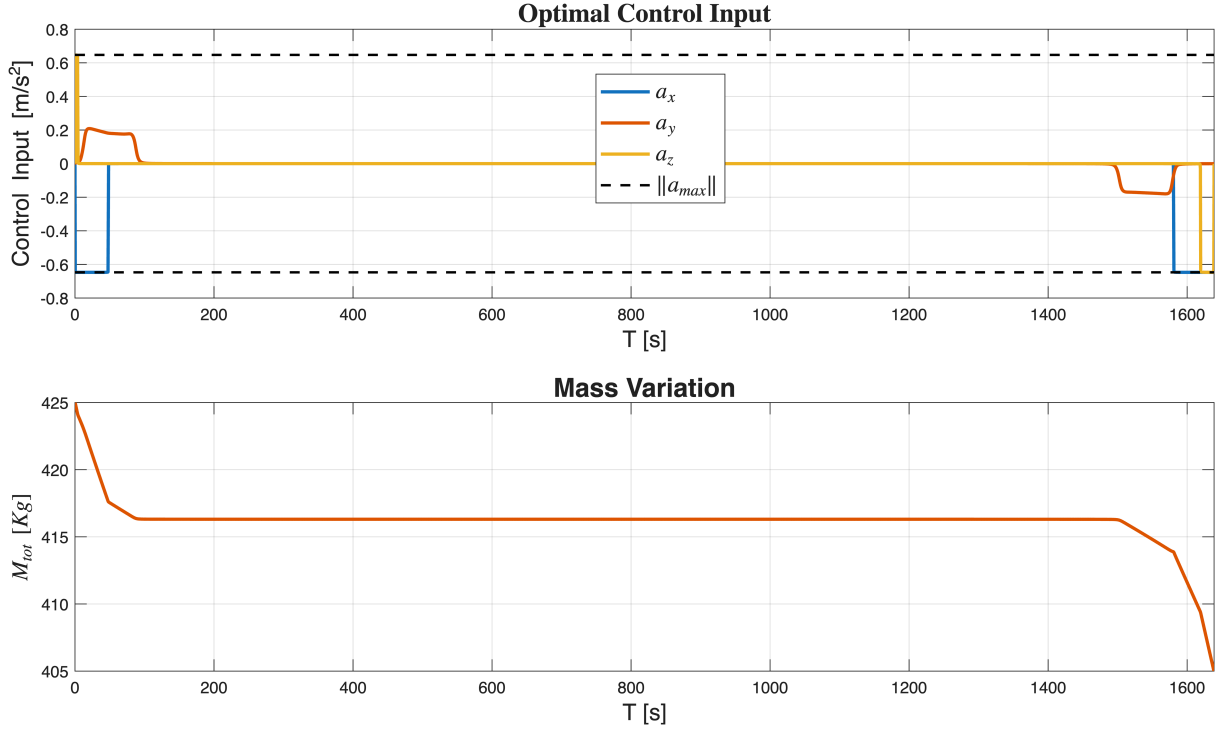


Figure 4.2: Control profile generated by the optimal control solver (top) and total satellite mass evolution (bottom).

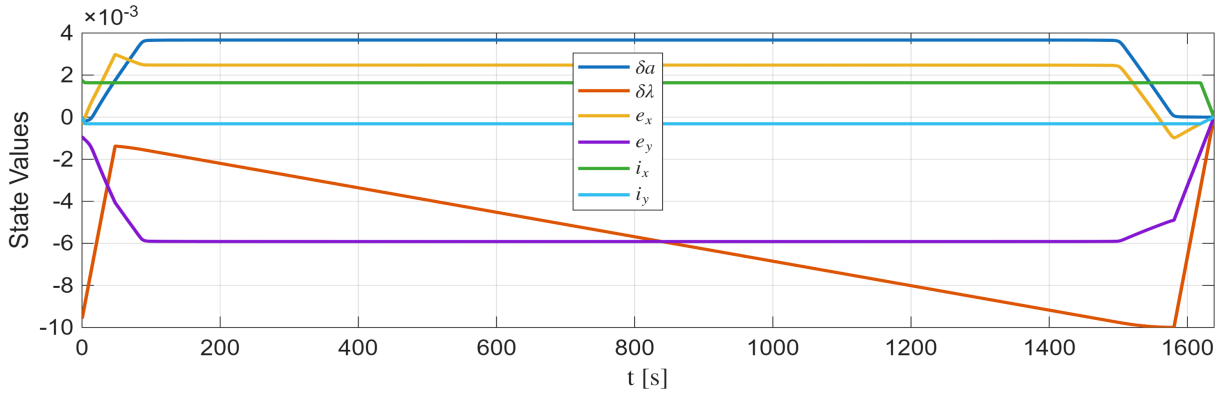


Figure 4.3: Relative Orbital Elements (ROE) reference trajectory generated by the optimal control.

Figure 4.3 shows the evolution of the state variables corresponding to this control profile. The initial conditions are derived from the physical orbital difference between the chaser and the target. In this specific scenario, the chaser begins with a semi-major axis difference $\Delta a = -1.300$ km, an eccentricity vector difference magnitude of $\Delta e = 0.001$, a mean anomaly difference $\Delta \theta = -0.55^\circ$ and a difference in inclination of $\Delta i = 0.1^\circ$.

The interaction between the control and the state dynamics is clearly visible:

- During the *Initial Burn*, the semi-major axis variation δa (blue line) is adjusted to a constant value suitable for the transfer.
- This constant δa induces a linear drift in the relative mean longitude $\delta \lambda$ (orange line) during the *Coast Phase*, effectively reducing the along-track distance without consuming propellant.
- During the *Final Braking*, δa and the remaining element errors are forced to zero, successfully completing the rendezvous.

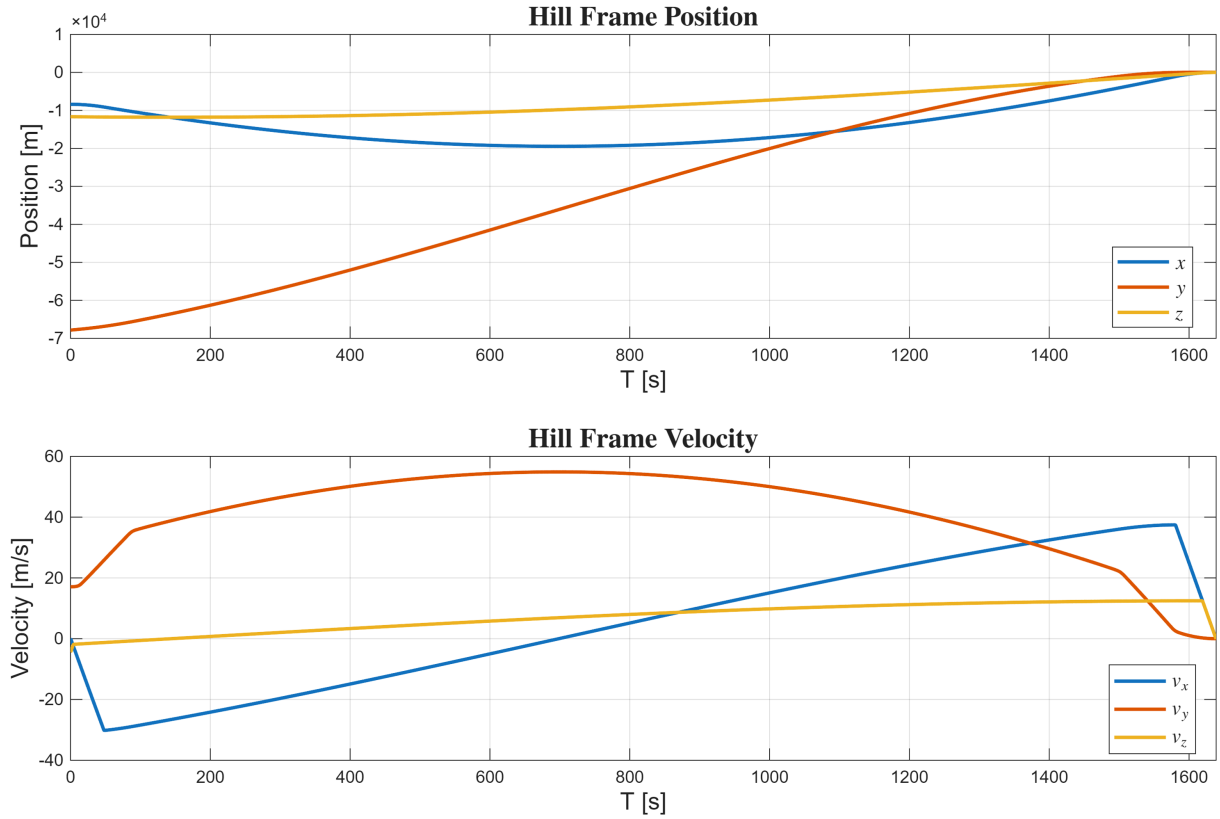


Figure 4.4: Hill frame reference trajectory generated by the optimal control.

Figure 4.4 illustrates the reference profile in Hill frame coordinates. Several key features of the dynamics can be observed:

- **Radial Position (x):** It can be noted that the radial component decreases from approximately -9 km to a minimum of -20 km. While this may seem counterintuitive, increasing the radial distance in the negative direction places the chaser in a lower orbit. This lower altitude naturally increases the orbital velocity, allowing the spacecraft to close the large separation distance in the along-track (y) direction effectively.

- **Coupled Velocity (v_x, v_y):** The radial and along-track velocities are coupled; therefore, even during the coasting phase, they are not constant. The negative radial displacement (x) generates a continuous acceleration in the along-track direction (Coriolis effect). As the spacecraft begins to ascend back toward $x = 0$ around $t = 700$ s, this "boost" effect diminishes and v_y begins to decrease as the target is approached.
- **Cross-Track Velocity (v_z):** The z component is completely decoupled. Although the Clohessy–Wiltshire equations describe this motion as a harmonic oscillator, the velocity profile appears quasi-linear here. This is because the maneuver duration is short relative to the full orbital period, making the sinusoidal trajectory appear straight.

Abort Strategy: E/I Vector Separation

In the event of a critical system failure (e.g., navigation failure) during the final approach (when the relative distance is small), the primary objective of the Guidance Layer shifts from rendezvous to collision avoidance. To ensure the safety of the Target spacecraft, the aim is to establish a passively safe relative orbit around the target. This facilitates a safe holding position for a second approach once the problem is fixed. Specifically, the strategy targets a safety ellipse defined by Eccentricity/Inclination (E/I) vector separation that guarantees a minimum keep-out distance while the spacecraft awaits fault recovery. This idea was originally proposed by D’Amico and Montenbruck [7].

Final Relative Orbit Selection

A critical risk during an abort maneuver is the potential for the recovery trajectory to cross the Target’s position (the origin of the Hill frame) while attempting to reach a predefined orbit. To mitigate this, the reference state for the safety ellipse, $\delta\alpha_{safe}$, is not fixed *a priori* but is determined based on the Chaser’s instantaneous state at the moment of failure, t_{abort} .

The algorithm analyzes the signs of the current relative eccentricity ($\delta\mathbf{e}$) and inclination ($\delta\mathbf{i}$) vectors. The target safe orbit is selected to be in the same quadrant as the current position, forcing the trajectory to expand outward rather than crossing the origin. For a required safety radius R_{safe} and target semi-major axis a_t , the target relative eccentricity component $\delta e_{x,safe}$ is defined as:

$$\delta e_{x,safe} = \begin{cases} +\frac{R_{safe}}{a_t} & \text{if } \delta e_x(t_{abort}) \geq 0 \\ -\frac{R_{safe}}{a_t} & \text{if } \delta e_x(t_{abort}) < 0 \end{cases} \quad (4.3)$$

Similarly, to ensure separation in the cross-track plane, the target relative inclination component $\delta i_{x,safe}$ is defined by the sign of the current inclination vector:

$$\delta i_{x,safe} = \begin{cases} +\frac{R_{safe}}{a_t} & \text{if } \delta i_x(t_{abort}) \geq 0 \\ -\frac{R_{safe}}{a_t} & \text{if } \delta i_x(t_{abort}) < 0 \end{cases} \quad (4.4)$$

The safety strategy strictly constrains these x -components ($\delta e_x, \delta i_x$) via eq. (4.3) and eq. (4.4) because they directly determine the minimum physical separation distance. However, the crucial design choice lies in the handling of the y -components. To guarantee passive safety via E/I vector separation, the relative eccentricity and inclination vectors must be collinear. This alignment ensures that the radial and normal oscillations are phased such that they never simultaneously pass through zero (creating a circular or elliptic orbit). Formally, the phase angles of the two vectors must satisfy the collinearity condition:

$$\delta \mathbf{e} \parallel \delta \mathbf{i} \implies \arctan\left(\frac{\delta e_y}{\delta e_x}\right) = \arctan\left(\frac{\delta i_y}{\delta i_x}\right) + k\pi, \quad k \in 0, 1 \quad (4.5)$$

By strictly setting the target y -components to zero ($\delta e_y = \delta i_y = 0$) for the recovery orbit, the guidance strategy forces both vectors to align perfectly with the x -axis. This trivially satisfies the collinearity condition in eq. (4.5), thereby simplifying the target state while mathematically guaranteeing the required spatial separation geometry.

Transient Keep-Out Constraint

Equations (4.3) and (4.4) compute a safe final coordinate and ensure that the desired state lies in the direction of expansion at the end of the maneuver. However, to guarantee safety during the transient transfer to the parking orbit, a linear keep-out constraint is embedded within the convex optimization problem.

This ensures that if, for example, the initial position is $\delta e_x > 0$ (target above), the final position will not only have a final constraint of $\delta e_{x,safe} > 0$, but the solver will also enforce a constraint requiring the Chaser to maintain or increase its current separation magnitude throughout the maneuver. Mathematically, for the radial component, this is expressed as:

$$\delta e_x(t) \cdot \text{sgn}(\delta e_x(t_{\text{abort}})) \geq |\delta e_x(t_{\text{abort}})| \cdot \beta \quad \forall t \in [t_{\text{abort}}, t_f]$$

The safety buffer coefficient $\beta \in (0, 1]$ acts as a scaling factor for the KOZ; by reducing β below unity, the constraint is relaxed, allowing the Chaser the necessary flexibility to drift slightly closer to the Target's to maintain feasibility or improve maneuver efficiency.

Operational Use

It is important to note that this E/I separation strategy is exclusively reserved for failure scenarios. In nominal conditions, applying these constraints would be counter-productive, as they strictly prevent the Chaser from reducing its relative distance to the Target, effectively rendering the final rendezvous impossible. Therefore, the Supervisory Logic (Section 4.4) is responsible for triggering this mode only when specific fault flags are raised.

4.2. Sliding Mode Controller (SMC)

The Sliding Mode Controller (SMC) is designed as a robust tracking layer to ensure the spacecraft accurately follows the reference trajectory generated by the optimal control solver, even in the presence of unmodeled perturbations or actuator faults (e.g., a stuck-open thruster).

Control Law Formulation

The underlying relative dynamics are modeled using the Clohessy-Wiltshire (CW) equations. Since these equations represent a Linear Time-Invariant (LTI) system, the sliding surface can be defined as a time-invariant linear combination of the state errors. Furthermore, relative navigation sensors provide measurements directly in the Hill reference frame, allowing for direct computation of the error state without transformations.

The sliding surface $\boldsymbol{\sigma} \in \mathbb{R}^3$ is defined as:

$$\boldsymbol{\sigma}(\mathbf{e}) = \mathbf{C}\mathbf{e}$$

where $\mathbf{e} = \mathbf{x} - \mathbf{x}_{\text{ref}}$ is the state error vector in the Hill frame and $\mathbf{C}$ is a positive definite gain matrix.

The control law is structured as the sum of an equivalent control term $\mathbf{u}_{\text{eq}}$ and a switching

control term $\mathbf{u}_{\text{sw}}$:

$$\mathbf{u} = \mathbf{u}_{\text{eq}} + \mathbf{u}_{\text{sw}}$$

In this hybrid architecture, the feedforward control input generated by the optimal control solver plays the role of an equivalent control $\mathbf{u}_{\text{eq}}$ under nominal conditions. This ensures that in nominal conditions, the switching term is zero. Different types of Sliding Mode Controller exist, for the sake of this thesis a classical First Order SMC is implemented as follow:

$$\mathbf{u}_{\text{sw}} = -\eta \tanh(\alpha \boldsymbol{\sigma}) \quad (4.6)$$

where η is the switching gain determining the robustness amplitude, the hyperbolic tangent function is used instead of the sign function to reduce chattering (Quasi-Sliding Mode) and α is used to tune the slope of the tanh function.

Parameters

To tune the parameters of the SMC defined in Equation (4.6) ($\mathbf{C}, \alpha, \eta$), various techniques can be employed. For the sliding surface matrix $\mathbf{C}$, classical techniques include Quadratic Minimization, eigenvalue placement and the unit vector approach. The switching gain η is typically tuned to compensate for matched disturbances; specifically, η must be chosen close to U_{max} to compensate for the maximum expected matched disturbance.

In this thesis, tuning is performed using Particle Swarm Optimization (PSO), as described in Chapter 5. This approach provides the flexibility to optimize different sliding surface structures and tailor the system's dynamic behavior by selecting a suitable cost function.

To simplify the tuning process by reducing the number of PSO optimization variables, as mentioned in Section 3.1, the Clohessy–Wiltshire (CW) model is decomposed into two decoupled subsystems representing the in-plane (S_1) and out-of-plane (S_2) relative dynamics:

$$S_1 : \dot{\mathbf{x}}_1 = \mathbf{A}_1 \mathbf{x}_1 + \mathbf{B}_1 \mathbf{U}_1, \quad \mathbf{A}_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 2n \\ 0 & 0 & -2n & 0 \end{bmatrix}, \quad \mathbf{B}_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (4.7)$$

$$S_2 : \dot{\mathbf{x}}_2 = \mathbf{A}_2 \mathbf{x}_2 + \mathbf{B}_2 U_2, \quad \mathbf{A}_2 = \begin{bmatrix} 0 & 1 \\ -n^2 & 0 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (4.8)$$

In-Plane SMC

Focusing on the first subsystem S_1 , which governs the in-plane motion, the sliding surface is defined as:

$$\boldsymbol{\sigma}(\mathbf{x}_1) = \mathbf{C}_1 \mathbf{x}_1$$

In this case, the matrix $\mathbf{C}_1 \in \mathbb{R}^{2 \times 4}$ relates two control inputs to four states. Consequently, the sliding surface matrix theoretically contains 8 parameters. However, the objective is to decouple the sliding surfaces to drive the errors $\mathbf{e}_x = [e_x \ e_{\dot{x}}]^\top$ and $\mathbf{e}_y = [e_y \ e_{\dot{y}}]^\top$ to zero independently. Therefore, a standard diagonal choice for the sliding surface is:

$$\mathbf{C}_1 = \begin{bmatrix} \lambda_x & 0 & 1 & 0 \\ 0 & \lambda_y & 0 & 1 \end{bmatrix}.$$

The resulting two sliding surfaces are defined as follows:

$$\begin{cases} \sigma_x = e_{\dot{x}} + \lambda_x e_x \\ \sigma_y = e_{\dot{y}} + \lambda_y e_y \end{cases} \Rightarrow \begin{cases} \dot{\sigma}_x = e_{\ddot{x}} + \lambda_x e_{\dot{x}} \\ \dot{\sigma}_y = e_{\ddot{y}} + \lambda_y e_{\dot{y}} \end{cases}$$

Substituting the in-plane Clohessy-Wiltshire (CW) dynamics into the derivative equations yields:

$$\begin{cases} \dot{\sigma}_x = (3n^2 e_x + 2n e_{\dot{y}} + u_x + d_x) + \lambda_x e_{\dot{x}} \\ \dot{\sigma}_y = (-2n e_{\dot{x}} + u_y + d_y) + \lambda_y e_{\dot{y}} \end{cases}$$

Rearranging terms highlights the cross-coupling:

$$\begin{cases} \dot{\sigma}_x = (3n^2 e_x + \lambda_x e_{\dot{x}}) + \mathbf{2n e_{\dot{y}}} + u_x + d_x \\ \dot{\sigma}_y = \lambda_y e_{\dot{y}} - \mathbf{2n e_{\dot{x}}} + u_y + d_y \end{cases}$$

where d_x and d_y are the matched disturbances acting on the system. It can be clearly observed that the two sliding surface dynamics are not decoupled; they contain the Coriolis coupling terms $2n e_{\dot{y}}$ (in $\dot{\sigma}_x$) and $-2n e_{\dot{x}}$ (in $\dot{\sigma}_y$).

To eliminate this coupling and simplify the controller tuning, a better choice for matrix $\mathbf{C}_1$ is proposed by introducing off-diagonal terms to counteract the Coriolis effect:

$$\mathbf{C}_1 = \begin{bmatrix} \lambda_x & -2n & 1 & 0 \\ 2n & \lambda_y & 0 & 1 \end{bmatrix}.$$

With this structure, the derivative of the sliding surface $\dot{\boldsymbol{\sigma}}$ naturally cancels the coupling

terms from the plant dynamics, resulting in decoupled error dynamics:

$$\begin{cases} \dot{\sigma}_x = 3n^2 e_x + \lambda_x e_{\dot{x}} + u_x + d_x \\ \dot{\sigma}_y = \lambda_y e_{\dot{y}} + u_y + d_y \end{cases}$$

This decoupling ensures that the control action on one axis does not perturb the sliding surface of the other axis, thereby improving tracking performance.

The complete set of optimization parameters for the Out-of-Plane dynamics is:

$$[\lambda_x, \lambda_y, \eta, \alpha].$$

Out-of-Plane SMC

Focusing on the second subsystem S_2 , which governs the out-of-plane motion, the sliding surface is defined as:

$$\sigma(\mathbf{x}_2) = \mathbf{C}_2 \mathbf{x}_2$$

where $\mathbf{C}_2 = [\lambda_z \ 1]$. The parameter $\lambda_z > 0$ determines the slope of the sliding manifold in the state-space phase plane.

Unlike the in-plane dynamics, the out-of-plane motion (z -axis) is naturally decoupled from the in-plane states (x, y) in the linear CW model. Therefore, no additional coupling terms are required in the sliding surface definition.

The complete set of optimization parameters for the Out-of-Plane dynamics is:

$$[\lambda_z, \eta, \alpha].$$

Parameter Tuning and Weight Selection

As stated above the SMC parameters are tuned using the PSO algorithm detailed in Chapter 5. Since the optimal parameters are obtained by numerically simulating the system, a critical question arises: how should the initial conditions for these simulations be set? Because the initial tracking errors and disturbances are not known *a priori* during real-world operations, tuning the controller to a single arbitrary initial condition might lead to an overfitted and fragile parameter set.

To ensure robustness across the entire operational envelope, this thesis implements a tuning strategy based on averaging the optimal parameters over multiple randomized initial conditions. When the tuning procedure initializes, it generates a set of random initial conditions that model various possible tracking errors relative to the steady-state

target. The random states are sampled from a uniform distribution, $\mathcal{U}(a, b)$, to ensure an equal probability of nominal and worst-case errors. For the xy plane, the initial states are sampled as follows:

$$\begin{aligned} x_0, y_0 &\sim \mathcal{U}(-400, 400) \text{ [m]} \\ v_{x0}, v_{y0} &\sim \mathcal{U}(-10, 10) \text{ [m/s]} \end{aligned}$$

The PSO algorithm is executed multiple times, once for each generated initial condition. Finally, the resulting optimal parameter sets are averaged to yield a generalized, robust set of SMC gains.

Beyond the initial conditions, the definition of the weighting matrices Q and R in the cost function heavily control the transient behavior of the satellite. For the xy plane, the state penalty matrix is defined as:

$$Q_{xy} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}, \quad R_{xy} = r_{xy} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Notice that the velocity states (v_x, v_y) are penalized more heavily than the position states (x, y) . This is a design choice driven by the difference in magnitude between the states: the velocity error usually an order of magnitude smaller than the position error. Weighting the velocity more heavily ensures that the optimizer properly accounts for it in the total cost function. This forces the system to brake adequately and approach the target smoothly.

The control penalty matrix R_{xy} , is defined as an identity matrix multiplied by a tunable scalar parameter r_{xy} :

Figure 4.5 illustrates the phase portrait displaying the reaching and sliding phases alongside the time-domain step response. When the control penalty r_{xy} is excessively high, the step response is slow and exhibits a slight overshoot. Furthermore, the phase portrait reveals that the controller struggles to maintain the trajectory on the sliding surface. This occurs because the severe penalty on control effort forces the optimization algorithm to select low values for both the reaching gain η and the boundary layer steepness α .

Conversely, lowering r_{xy} allows the optimizer to safely increase both η and α . As a result, the step response becomes significantly faster with zero overshoot, and the phase portrait demonstrates that the state trajectory adheres better to the sliding surface. Decreasing r_{xy} even further provides only marginal performance improvements; at this stage, the optimal reaching gain η has already hit the physical saturation limit of the actuators ($U_{\max}$) in the preceding configuration, preventing any additional control aggressiveness.

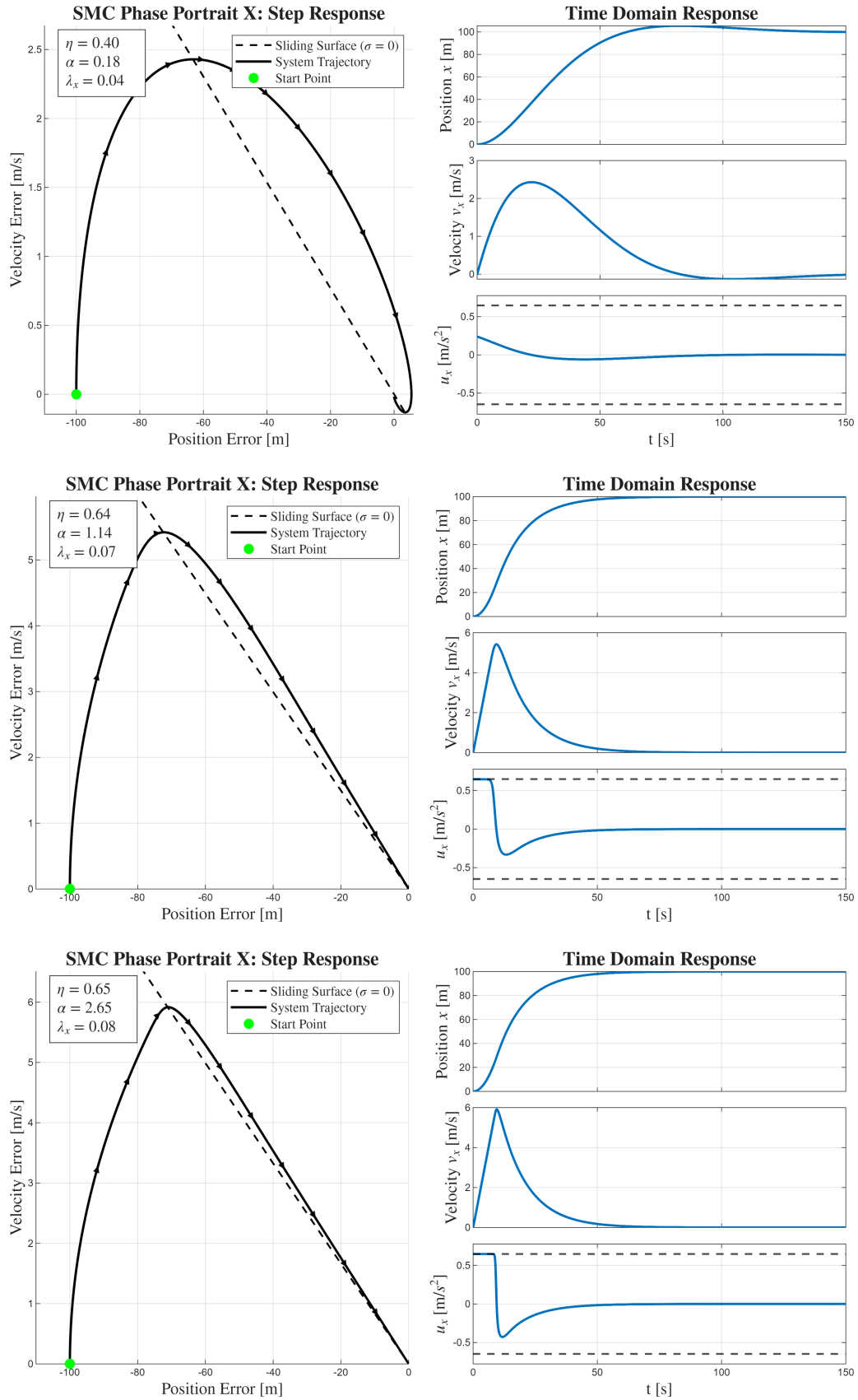


Figure 4.5: Comparison of SMC step responses for a 100 m initial error on the x -axis. The three vertically stacked plots illustrate the system trajectory as the control penalty scalar r_{xy} decreases (top to bottom).

4.3. Final Approach Strategy

In satellite rendezvous operations, the *Final Approach* is among the most critical phases, requiring precise relative navigation and strict collision avoidance. The chaser must approach the target surrounded by a predefined Keep-Out Zone (KOZ) starting from various orbital configurations, which makes the guidance problem complex.

To ensure a safe and autonomous approach, the guidance strategy must drive the chaser toward the docking point while actively avoiding restricted regions. Among existing methods, the **Artificial Potential Field (APF)** offers an effective and computationally efficient solution.

Artificial Potential Field (APF) Guidance

Originally proposed for robotic path planning, the APF models the operational space as a virtual potential field where the target generates an attractive potential and obstacles (or KOZs) generate repulsive potentials. This concept has been successfully adapted for space applications, enabling real-time, collision-free trajectory generation.

In this thesis, the APF is utilized to compute the direction of maximum potential decrease at each integration step, similar to the approach in [24]. This direction defines a desired velocity vector for a lower-level **SMC**, combining the global path planning capability of APF with the robustness of SMC. Although artificial potential fields may exhibit local minima, the proposed cardioid–cone combination is designed such that the only minimum within the operational region corresponds to the desired docking point.

Cardioid-Based Potential Function

The choice of the potential function is of primary importance. Since no assumptions are made regarding the chaser's initial position relative to the target (the maneuver may start from any direction), classical conical approach corridors are insufficient.

To address this, a Cardioid based potential function adapted from [25] is employed. It is defined as:

$$h(\boldsymbol{\rho}) = \|\boldsymbol{\rho} + \boldsymbol{\Delta}\|_2^2 + \mathbf{c}_s^\top (\boldsymbol{\rho} + \boldsymbol{\Delta}) - \varphi \|\boldsymbol{\rho} + \boldsymbol{\Delta}\|_2$$

where $\boldsymbol{\rho} = [x \ y \ z]^\top$ is the relative position vector, $\boldsymbol{\Delta} = [\varrho \ 0 \ 0]^\top$ is a positive offset vector shifting the cardioid along the x-axis and $\mathbf{c}_s = [\varphi \ 0 \ 0]^\top$ scales the cardioid size.

The total potential function J_t combines this boundary term with an attractive quadratic

well:

$$J_t(\boldsymbol{\rho}) = N(\boldsymbol{\rho}) \frac{k_1}{h(\boldsymbol{\rho})} + k_2 N(\boldsymbol{\rho}) \quad (4.9)$$

where k_1 regulates the steepness of the potential near the cardioid boundary, effectively adjusting the thickness of the **Keep-Out Zone (KOZ)** boundary, while k_2 increases the influence of the **attractive potential**. The term

$$N(\boldsymbol{\rho}) = \frac{1}{2} \boldsymbol{\rho}^\top \boldsymbol{\rho} \quad (4.10)$$

represents a quadratic attractive potential guiding the chaser toward the target.

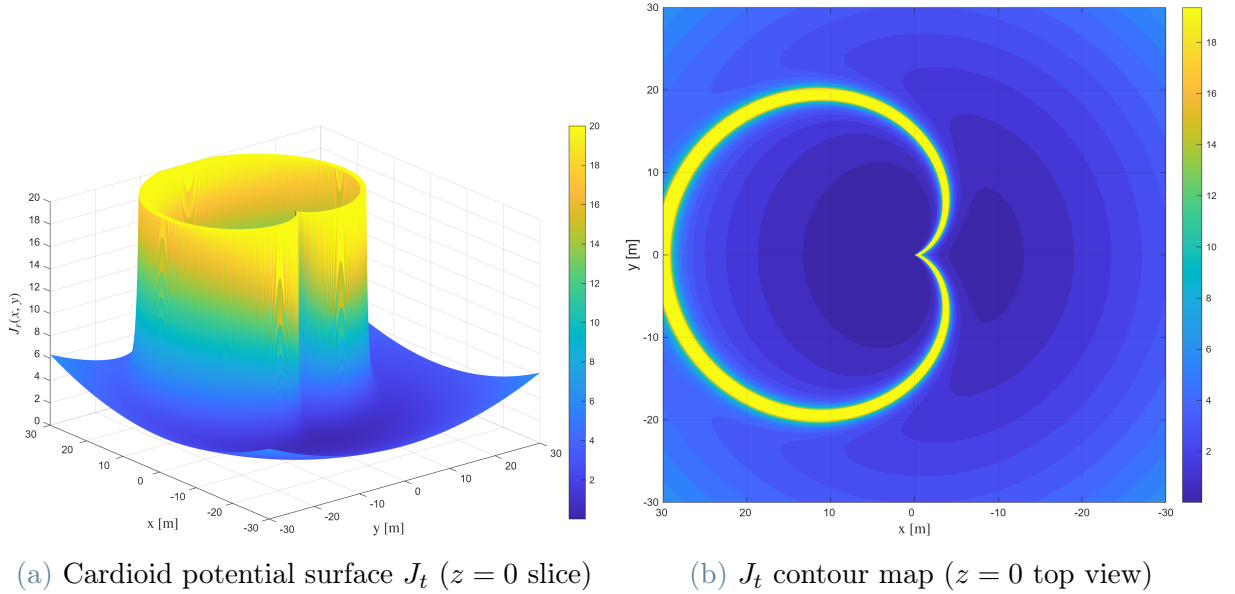


Figure 4.6: Artificial Potential Field (APF) sliced at $z = 0$.

From Figure 4.6, it is possible to observe the **cardioid boundary** is visible in the xy -plane (at $z = 0$) and the presence of two local minima: one located inside the cardioid (theoretically unreachable from the outside) and another located outside. The latter represents the minimum within the *region of interest* and can therefore be considered the **global minimum** within this subspace.

Quadratic Cone Potential

As seen in Figure 4.6b, the potential field's minimum is offset from the origin at the cardioid's cusp. Translating the cardioid via $\boldsymbol{\rho}$ corrects this offset but leaves a critically narrow safety margin of only a few meters around the docking point. While increasing

the parameter φ expands this margin, it generates a significantly larger **Keep-Out Zone (KOZ)**.

To resolve this trade-off, the cardioid is shifted ($\varphi = -10$) to completely encapsulate the target, acting purely to steer the chaser onto the proper docking axis.

For this final docking phase, an approach corridor is defined along the negative $-x$ direction. A local coordinate frame is introduced to align with this operational geometry:

$$x_l = -x.$$

Within this frame, the geometric boundary of the approach corridor is a cone, defining the strict spatial constraint for the switch point from the cardioid to the cone phase:

$$y^2 + z^2 \leq (x_l \tan \phi_c)^2 \quad (4.11)$$

where ϕ_c is the cone aperture angle.

Once the chaser aligns with the $-x$ axis and crosses this boundary, a secondary **potential function** provides the internal steering logic to actively guide the spacecraft to the docking port. This steering potential is defined as a weighted quadratic well:

$$J_{cone}(x_l, y, z) = x_l^2 + k_{lat}(y^2 + z^2) \quad (4.12)$$

where k_{lat} is the lateral tuning weight. Explicitly decoupling this control potential from the physical geometric boundary allows the control responsiveness to be aggressively tuned without altering or violating safety limits. Consequently, the gradient vector used to generate the velocity reference is computed as:

$$\nabla J_{cone}(\boldsymbol{\rho}) = \begin{bmatrix} 2x_l \\ 2k_{lat}y \\ 2k_{lat}z \end{bmatrix}$$

Furthermore, by assigning a high lateral weight ($k_{lat} \gg 1$), the potential is mathematically forced to prioritize rapid centering. The chaser tightly aligns with the centerline before progressing forward, effectively preventing wide, sluggish trajectories while safely descending the cone.

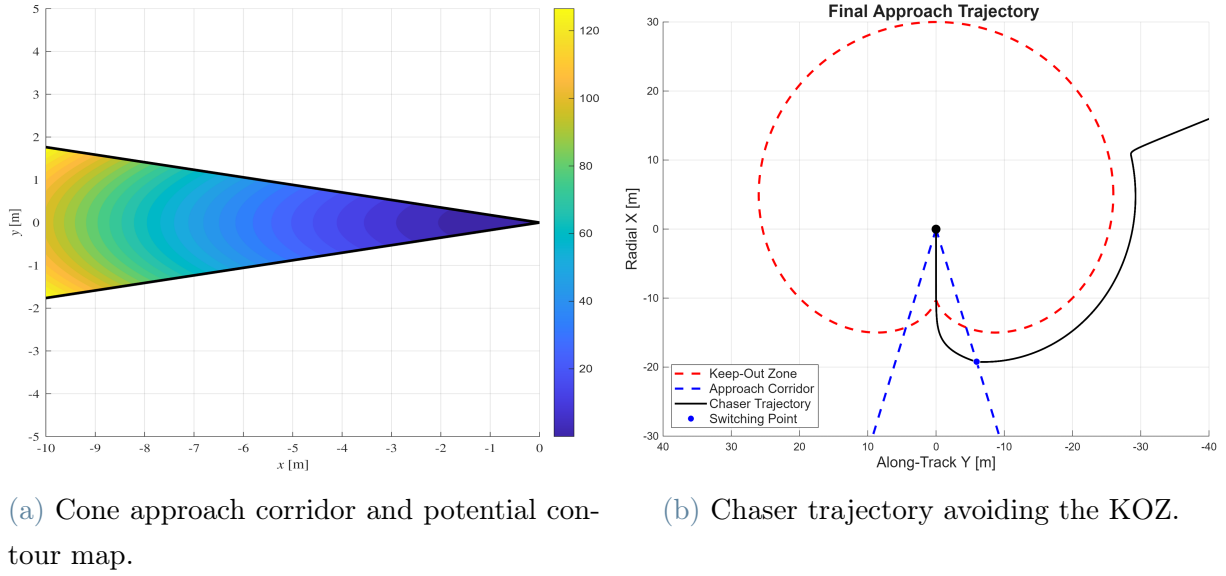


Figure 4.7: Illustration of the cone potential function and its overall effect on the final docking trajectory.

Figure 4.7a illustrates the external boundary of the **Quadratic Cone**, which defines the geometric region where the corresponding artificial potential function becomes active, alongside its contour map. In Figure 4.7b, a simulated rendezvous scenario demonstrates how the **Cardioid Potential Function** initially guides the chaser along the $-\bar{R}$ direction to safely avoid the Keep-Out Zone (KOZ). Once the chaser crosses the geometric boundary and enters the Line-of-Sight corridor, it is smoothly captured and driven toward the docking origin by the secondary quadratic potential function.

Velocity-Based Sliding Mode Control

The guidance layer computes the desired direction of motion:

$$\mathbf{E}_u = \frac{-\nabla J}{\|\nabla J\|}$$

This is converted into a reference velocity vector for the control layer:

$$\mathbf{v}_{\text{ref}} = v_{\text{max}} \mathbf{E}_u \quad (4.13)$$

Where v_{max} is the maximum velocity allowed in the final approach. To ensure robust tracking of this reference, a simplified Sliding Mode Controller (SMC) proposed in [24].

The sliding surfaces $\boldsymbol{\sigma} \in \mathbb{R}^3$ are defined as:

$$\boldsymbol{\sigma}(\mathbf{x}) = c(\mathbf{v} - \mathbf{v}_{\text{ref}}) \quad (4.14)$$

where $c > 0$ is a design parameter. The corresponding control law is:

$$u_i = U_i^{\max} \tanh(k\sigma_i) \quad \forall i \in \{x, y, z\}$$

Adaptive Velocity Constraints

To ensure safe docking of the chaser in the final meters, the maximum velocity $v_{\max}$ is adaptively reduced as the relative distance $\|\mathbf{x}\|$ falls below a proximity threshold $\bar{\rho}$:

$$v_{\max} = \begin{cases} v_{\max}, & \text{if } \|\mathbf{x}\| > \bar{\rho} \\ v_{\max} \frac{\|\mathbf{x}\|}{\bar{\rho}}, & \text{if } \|\mathbf{x}\| \leq \bar{\rho} \end{cases}$$

This formulation guarantees that $\|\mathbf{v}(t)\|_2 < v_{\max}$ at all times (as shown in Figure 4.8b), satisfying the safety constraints intrinsic to close-range proximity operations.

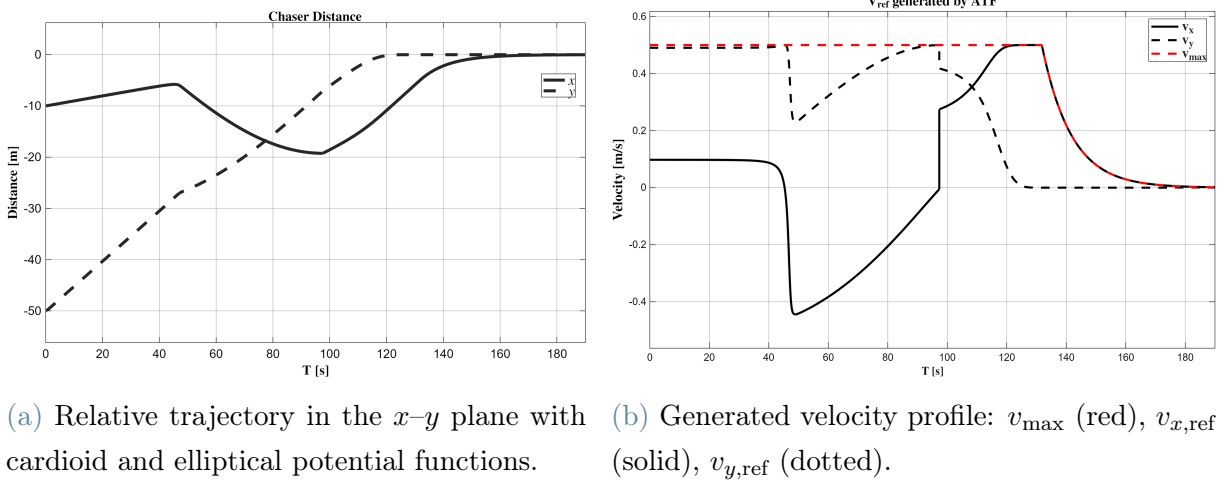


Figure 4.8: Chaser relative position and velocity during the final approach phase.

Figure 4.8 shows the chaser's relative motion during the final approach phase. The left plot illustrates the trajectory in the x - y plane, where the **KOZ** can be clearly observed along the x direction. The right plot displays the velocity references ($v_{x,\text{ref}}$, $v_{y,\text{ref}}$) generated by Equation (4.13) (with $v_{z,\text{ref}} = 0$, as z is set to zero). It can be observed that the magnitude of the relative velocity reference always satisfies the constraint on $v_{\max}$ and decreases during the final phase.

4.4. Hybrid Control

This section introduces the hybrid control architecture adopted in this thesis to address the requirements of optimality, constraint satisfaction and robustness in spacecraft rendezvous and proximity operations. The term *hybrid* is used here to denote the combination of distinct control modes, each governed by different objectives and design principles, together with a logic-based switching mechanism that selects the appropriate mode based on the system state and mission phase.

Purely optimal control solutions, while attractive from a theoretical standpoint, are often sensitive to modeling uncertainties and computationally demanding when implemented in real time. Instead, classical feedback controllers offer robustness and simplicity but may fail to guarantee optimal resource usage or strict constraint satisfaction. The proposed hybrid framework aims to leverage the strengths of both approaches by separating the control problem into complementary layers.

Control Architecture

At a high level, the control system is hierarchically structured into four main functional blocks, as illustrated in Figure 4.9:

- **Guidance Layer:** Responsible for generating reference trajectories or control profiles while satisfying mission constraints. It operates in two distinct modes depending on the mission phase.
- **Tracking Layer:** Tasked with ensuring the spacecraft accurately follows the guidance references while rejecting disturbances and compensating for modeling errors.
- **Supervisory Logic:** A finite state machine that governs the activation of and transition between different control and navigation modes based on real-time conditions.
- **Sensors:** Handles the estimation of the relative state. It switches between an internal Inertial Navigation System (INS) for long-range propagation (subject to drift as discussed in Section 2.3 and precise relative sensors when the target is within operational range.

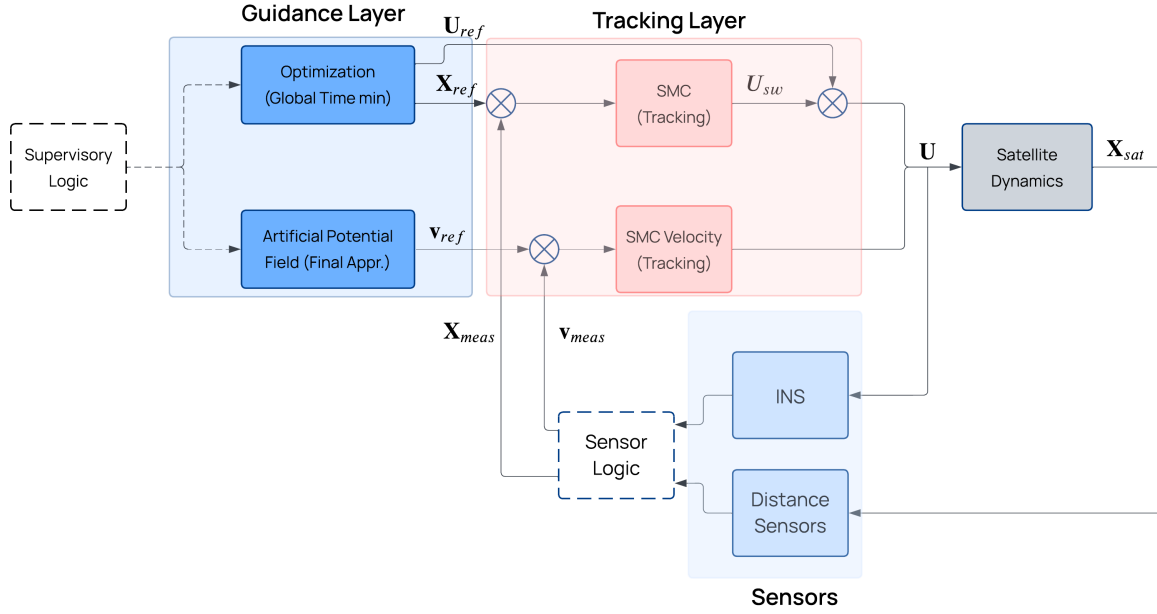


Figure 4.9: Hybrid Control Architecture Diagram.

The **Guidance Layer** has two modules:

1. **Optimization Module:** Active during the rendezvous and proximity phases, this module generates a full state reference $\mathbf{x}_{ref} \in \mathbb{R}^6$ and a feedforward control reference $\mathbf{U}_{ref} \in \mathbb{R}^3$ via the optimization solver.
2. **APF Module:** Active exclusively during the final approach, the Artificial Potential Field module generates a velocity reference vector $\mathbf{v}_{ref} \in \mathbb{R}^3$ to ensure collision avoidance and enforce maximum approach velocity constraints.

The **Tracking Layer** implements two distinct Sliding Mode Controllers (SMC) corresponding to the active guidance mode:

1. **Trajectory Tracking SMC** (defined in Equation (4.6)) which tracks $\mathbf{x}_{ref}$. In this configuration, the optimal control output $\mathbf{U}_{ref}$ serves as the equivalent control input and the SMC adds a robust switching term $\mathbf{U}_{sw}$ to compensate for uncertainties.
2. **Velocity Tracking SMC** (defined in Equation (4.14)) used during the final approach to track the velocity vector $\mathbf{v}_{ref}$ generated by the APF.

Finally, the **Navigation Layer** closes the feedback loop. The *Sensor Logic* block selects the appropriate feedback source to construct the measured state $\mathbf{x}_{meas} \in \mathbb{R}^6$. When the target is outside sensor range, the system relies on the INS, which estimates the relative

state by propagating the applied control commands $\mathbf{U}$ through the internal satellite dynamic model (Section 2.3). Once the target is within range, the logic switches to the *Distance Sensors*, providing direct and accurate measurements of the relative state.

5 | Particle Swarm Optimization (PSO)

In the previous chapter, a SMC was established as the robust tracking layer for the hybrid control architecture. While the SMC structure guarantees asymptotic stability in theory, its practical performance is highly sensitive to the selection of its design parameters, specifically the sliding surface slopes (λ) and the switching gains (η).

Determining the optimal set of these parameters analytically is challenging. The control problem involves hard constraints (actuator saturation), non-linear dynamics and the feasible parameter space is constrained by closed-loop stability. This creates a non smooth landscape where classical gradient-based methods, such as Sequential Quadratic Programming (SQP), are prone to failure due to numerical noise and local minima.

To address these challenges, this chapter presents a customized simulation-based optimization framework utilizing Particle Swarm Optimization (PSO). Unlike standard off-the-shelf implementations, the proposed algorithm incorporates specific topological and dynamic modifications to ensure robust convergence. Specifically, a Von Neumann neighborhood topology is adopted to maintain swarm diversity and prevent premature convergence, coupled with a constriction factor formulation to guarantee stability. Furthermore, a specialized cost function is designed to explicitly penalize actuator saturation, ensuring that the generated control gains remain physically feasible.

The chapter is organized as follows:

- **Section 5.1** defines the swarm topology, justifying the selection of the Von Neumann structure over the standard Global Best topology to balance exploration and exploitation.
- **Section 5.2** details the custom PSO algorithm, including the constriction factor implementation, the integration with the variable-step ODE solver and the formulation of the penalty-based cost function.
- **Section 5.3** presents the optimization results, demonstrating the algorithm's con-

vergence and providing a comparative analysis that highlights the superiority of this custom PSO approach over deterministic gradient solvers for this specific application.

5.1. Topology

An essential aspect of the PSO algorithm is the manner in which information is shared among particles known as the *topology*. The topology defines how each particle communicates with others within the swarm. Several common topological structures are used in the literature:

- **Star (Global Best):** each particle has access to the best position found by the entire swarm, commonly referred to as the *global best* (*gbest*). This topology enables rapid convergence but increases the risk of premature convergence to local minima.
- **Ring (Local Best):** each particle communicates only with its two immediate neighbors. Information therefore propagates more slowly through the network, leading to slower convergence but better coverage of the search space. The corresponding PSO variant is referred to as the *local best* (*lbest*) PSO.
- **Wheel:** in this structure, all particles communicate through a single central particle (the hub), while being isolated from each other. This slows down the diffusion of information but can preserve diversity in the swarm for longer periods.
- **Von Neumann:** particles are arranged in a two-dimensional grid where each particle communicates with its four immediate neighbors (up, down, left, right).

For this work, the Von Neumann topology has been selected. Empirical studies have shown that this topology offers advantages in terms of consistency and provides a better balance between exploration and exploitation [26–28]. In Fig. 5.1(a), the two-dimensional grid structure of the Von Neumann neighborhood is illustrated. When the particles located on the borders are connected to those on the opposite sides left to right and top to bottom a toroidal structure is formed, as depicted in Fig. 5.1(b).

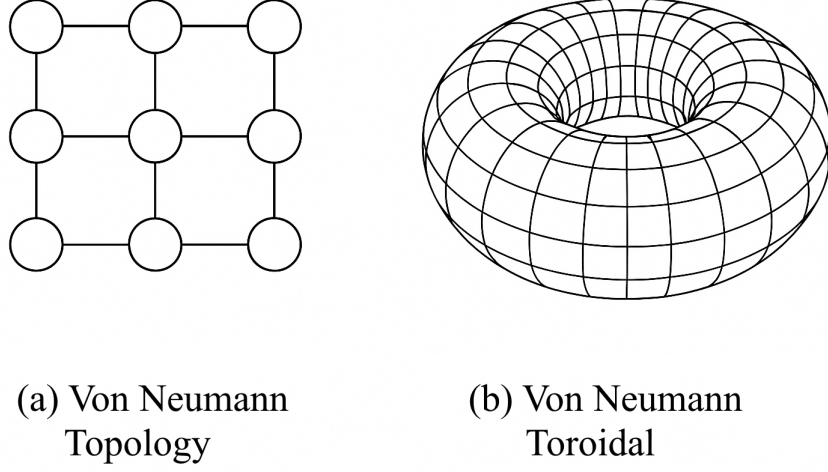


Figure 5.1: Von Neumann topology: (a) 2D grid structure, (b) toroidal wrapping.

5.2. PSO Algorithm

Let $x_i(k)$ denote the position of the i -th particle in the search space at iteration k , where the number of dimensions corresponds to the number of optimization parameters. The particle's position is updated by adding a velocity term $v_i(k)$ to its current position:

$$x_i(k+1) = x_i(k) + v_i(k+1)$$

In the classical formulation of the Particle Swarm Optimization (PSO) algorithm, the velocity of each particle is updated according to

$$v_{ij}(k+1) = v_{ij}(k) + c_1 r_{1j}(k) [y_{ij}(k) - x_{ij}(k)] + c_2 r_{2j}(k) [\hat{y}_{ij}(k) - x_{ij}(k)], \quad (5.1)$$

where $v_{ij}(k)$ denotes the velocity of particle i in dimension $j = 1, \dots, n_x$ at iteration k and $x_{ij}(k)$ is its corresponding position. The constants c_1 and c_2 are the cognitive and social acceleration coefficients, respectively, which scale the influence of the personal and neighborhood components. The random terms $r_{1j}(k), r_{2j}(k) \sim U(0, 1)$ are uniformly distributed in the range $[0, 1]$ and introduce stochastic behavior, enhancing the exploratory capability of the algorithm.

The personal best position, $\mathbf{y}_i(k)$, represents the best position (set of parameters) visited

by particle i up to iteration k . For a minimization problem, it is updated as follows:

$$\mathbf{y}_i(k+1) = \begin{cases} \mathbf{y}_i(k) & \text{if } J_f(\mathbf{x}_i(k+1)) \geq J_f(\mathbf{y}_i(k)) \\ \mathbf{x}_i(k+1) & \text{if } J_f(\mathbf{x}_i(k+1)) < J_f(\mathbf{y}_i(k)) \end{cases}$$

Where $J_f : \mathbb{R}^n \rightarrow \mathbb{R}$ is the cost function associated with the problem; it measures the fitness of the particles' parameters. The construction of the cost function will be discussed later.

The local best, denoted by $\hat{\mathbf{y}}_i$, represents the best solution among the neighbors of particle i , as defined by the swarm topology. In this work, for each particle, $\hat{\mathbf{y}}_i$ corresponds to the best among its adjacent particles (up, down, left and right).

Let Π_i denote the set of neighboring particles of particle i , as defined by the chosen topology. The local (or neighborhood) best position $\hat{\mathbf{y}}_i$ is the position corresponding to the minimum cost among all particles in Π_i , including particle i itself. Formally,

$$\hat{\mathbf{y}}_i = \arg \min_{\mathbf{y}_n(k)} \{ J_f(\mathbf{y}_n(k)) \mid n \in \Pi_i \}.$$

A well known issue with the classical PSO formulation Equation (5.1) is the potential divergence of the particle velocities, which can cause particles to move erratically or leave the feasible search space. Several strategies have been proposed in the literature to mitigate this problem, such as velocity clamping, position boundary reflections, or random re initialization.

In this work, the constriction factor approach proposed by Clerc and Kennedy [29] has been adopted. This method introduces a scaling factor, denoted as χ , that effectively damps the velocity update, ensuring convergence while maintaining exploration capability. The modified update equation becomes

$$\begin{cases} v_{ij}(k+1) = \chi \left[v_{ij}(k) + c_1 r_{1j}(k) [y_{ij}(k) - x_{ij}(k)] + c_2 r_{2j}(k) [\hat{y}_{ij}(k) - x_{ij}(k)] \right], \\ \chi := \frac{2}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|}, \end{cases} \quad (5.2)$$

where $\phi = c_1 + c_2$ and convergence is guaranteed when $\phi > 4$.

The constriction factor χ acts as a damping coefficient that limits the growth of the velocity term, preventing oscillations and divergence. This formulation provides a theoretically grounded balance between convergence speed and swarm diversity, allowing stable explo-

ration of the search space while avoiding premature stagnation.

Algorithm 5.1 Particle Swarm Optimization (PSO) with Von Neumann Topology

Require: N , n_{param} , N_{max} , $Boundary$, J_f , ε

- 1: Initialize parameters c_1 , c_2 , $\phi = c_1 + c_2$
 - 2: Compute constriction factor $\chi = \frac{2}{\left|2 - \phi - \sqrt{\phi^2 - 4\phi}\right|}$
 - 3: Initialize particles' positions $\mathbf{x}_{ij}(0)$ uniformly in $Boundary$
 - 4: Initialize velocities $\mathbf{v}_{ij}(0) = \mathbf{0}$
 - 5: Evaluate initial costs $J_f(\mathbf{x}_{ij}(0))$
 - 6: Set personal bests $\mathbf{y}_{ij}(0) = \mathbf{x}_{ij}(0)$
 - 7: Define neighborhood Π_{ij} using Von Neumann topology
 - 8: Determine local bests $\hat{\mathbf{y}}_{ij}(0)$ within Π_{ij}
 - 9: **while** $iter < N_{max}$ **and** $Stop == 0$ **do**
 - 10: **for** each particle (i, j) **do**
 - 11: Generate random numbers $r_1, r_2 \sim U(0, 1)$
 - 12: Update velocity:
 - 13: $\mathbf{v}_{ij}(k+1) = \chi[\mathbf{v}_{ij}(k) + c_1 r_1(\mathbf{y}_{ij}(k) - \mathbf{x}_{ij}(k)) + c_2 r_2(\hat{\mathbf{y}}_{ij}(k) - \mathbf{x}_{ij}(k))]$
 - 14: Update position: $\mathbf{x}_{ij}(k+1) = \mathbf{x}_{ij}(k) + \mathbf{v}_{ij}(k+1)$
 - 15: Evaluate cost $J_f(\mathbf{x}_{ij}(k+1))$
 - 16: **if** $J_f(\mathbf{x}_{ij}(k+1)) < J_f(\mathbf{y}_{ij}(k))$ **then**
 - 17: $\mathbf{y}_{ij}(k+1) = \mathbf{x}_{ij}(k+1)$
 - 18: **else**
 - 19: $\mathbf{y}_{ij}(k+1) = \mathbf{y}_{ij}(k)$
 - 20: **end if**
 - 21: **end for**
 - 22: Update local bests $\hat{\mathbf{y}}_{ij}(k+1)$ in each neighborhood Π_{ij}
 - 23: Compute best global cost $J_{best}(k+1) = \min J_f(\mathbf{y}_{ij}(k+1))$
 - 24: **if** stopping criterion $< \delta_{stop}$ is met **then**
 - 25: $Stop = 1$
 - 26: **end if**
 - 27: **end while**
 - 28: **return** Best parameters $\mathbf{x}^* = \arg \min_{\mathbf{x}_{ij}} J_f(\mathbf{x}_{ij})$
-

Cost Function

The cost function $J_f : \mathbb{R}^n \rightarrow \mathbb{R}$ that gave the best results in this context is

$$J_f = \int_0^T \left(\mathbf{u}(\tau)^\top R \mathbf{u}(\tau) + \mathbf{x}(\tau)^\top Q \mathbf{x}(\tau) \right) d\tau + k_\phi \int_0^T \phi(\tau)^{n_\phi} d\tau, \quad (5.3)$$

where $\mathbf{x}(\tau) \in \mathbb{R}^n$ is the state vector, $\mathbf{u}(\tau) \in \mathbb{R}^m$ is the control vector, and $R \in \mathbb{R}^{m \times m}$, $Q \in \mathbb{R}^{n \times n}$ are symmetric positive semi-definite weighting matrices.

The violation function $\phi(\tau)$ penalizes control amplitudes beyond the actuator limit $U_{\max}$. Since the thrusters operate independently on each axis, we enforce a box constraint using the infinity norm $\|\cdot\|_\infty$. We define

$$\phi(\tau) = \begin{cases} 0, & \text{if } \|\mathbf{u}(\tau)\|_\infty \leq U_{\max}, \\ \|\mathbf{u}(\tau)\|_\infty - U_{\max}, & \text{if } \|\mathbf{u}(\tau)\|_\infty > U_{\max}, \end{cases}$$

where $\|\mathbf{u}(\tau)\|_\infty = \max_i |u_i(\tau)|$. This formulation ensures that the control input for every individual axis i satisfies $|u_i| \leq U_{\max}$.

- The first term in (5.3) is the finite-horizon LQR-style cost (state and control quadratic penalties). Choosing $Q \succeq 0$ and $R \succ 0$ yields a standard quadratic regulation objective.
- The second term penalizes violations of the control bound $U_{\max}$. Raising ϕ to the power n_ϕ allows control of the severity of the penalty: $n_\phi = 1$ gives a linear penalty, while $n_\phi > 1$ yields a stronger penalization of large violations.
- Without this penalty, the PSO tends to identify control laws that generate short, high-amplitude spikes beyond the saturation threshold. Increasing R can mitigate this effect, but it typically slows the system's convergence to equilibrium. The proposed penalty provides a better trade-off.

Each time the PSO evaluates the cost function J_f , the closed-loop system dynamics must be numerically integrated. The solver employed is **ODE15s**, a variable-step, variable-order implicit solver designed for stiff systems.

It is important to note that actuator saturation is explicitly accounted for in the system dynamics during numerical simulation, ensuring that the state evolution reflects the physical limitations of the actuators. However, the control signal used in the evaluation of the cost function J_f is the unsaturated (ideal) control input.

This design choice prevents saturation from artificially masking excessive control demands during optimization. Instead, any attempt by the controller to exceed the actuator limits is explicitly penalized through an additional penalty term in the cost function. As a result, the optimizer is encouraged to select control parameters that remain within feasible bounds, while still accurately capturing the impact of saturation on the system dynamics.

This separation between the simulated control input and the cost evaluation improves numerical robustness and allows the penalty mechanism to effectively discourage infeasible control actions, without compromising the physical realism of the closed-loop simulation.

Stopping Criteria

In the literature, several stopping criteria for PSO have been proposed [30]. In this work, an improvement-based stopping criterion is adopted. The optimization process is terminated when the improvement in the global best cost value becomes sufficiently small over a given number of iterations.

To robustly evaluate the convergence behavior, a moving average of the best cost values is used to smooth out fluctuations. Let $J_{\text{best}}(k)$ denote the global best cost value at iteration k and let N_s represent the number of samples used to compute the moving average. The averaged cost values at iteration t are defined as:

$$\begin{aligned}\bar{J}_{\text{old}}(k) &= \frac{1}{N_s} \sum_{n=k-2N_s+1}^{k-N_s} J_{\text{best}}(n), \\ \bar{J}_{\text{new}}(k) &= \frac{1}{N_s} \sum_{n=k-N_s+1}^k J_{\text{best}}(n).\end{aligned}$$

The normalized improvement between these two averages is computed as:

$$\Delta_J(k) = \frac{|\bar{J}_{\text{new}}(k) - \bar{J}_{\text{old}}(k)|}{|\bar{J}_{\text{old}}(k)| + \varepsilon},$$

where ε is a small positive constant to prevent division by zero.

The stopping criterion is then expressed as:

$$\text{if } \Delta_J(k) < \delta_{\text{stop}}, \quad \text{then the optimization process is terminated.}$$

Here, δ_{stop} is a tolerance determining the sensitivity of the stopping condition.

5.3. Optimization Results

The PSO algorithm described above was applied to tune the parameters for the out-of-plane SMC. To visualize the optimization landscape and the PSO iterations, only two parameters are optimized: the sliding surface slope λ_z and the switching gain η_z and α fixed at 1, this allows a two-dimensional visualization of the optimization process and the evolution of the particle swarm.

In all the simulations below, the PSO is configured with the parameters listed in Table 5.1. The swarm uses a Von Neumann topology with a grid size of 7×7 , resulting in 49 particles. The cognitive and social coefficients are set to $c_1 = c_2 = 2.14$ to utilize the constriction factor $\chi \approx 0.73$ for stability.

Parameter	Value
Swarm Topology	Von Neumann (7×7)
Number of Particles (N_{pop})	49
Max Iterations (N_{max})	100
Cognitive/Social Coeff. (c_1, c_2)	2.14
Search Space Boundary	$[0, 3]$
State Weight (Q)	diag(10, 1)
Control Weight (R)	1000

Table 5.1: Parameters used for the Particle Swarm Optimization.

Given the initial conditions

$$\mathbf{x}_2 = \begin{bmatrix} \delta z \\ \dot{\delta z} \end{bmatrix} = \begin{bmatrix} -1500 \text{ m} \\ 10 \text{ m/s} \end{bmatrix},$$

Figure 5.2 illustrates the convergence profile of the algorithm. It can be observed that the cost function J_f decreases rapidly in the initial iterations (exploration phase) and settles into a minimum value after approximately 15 iterations (exploitation phase).

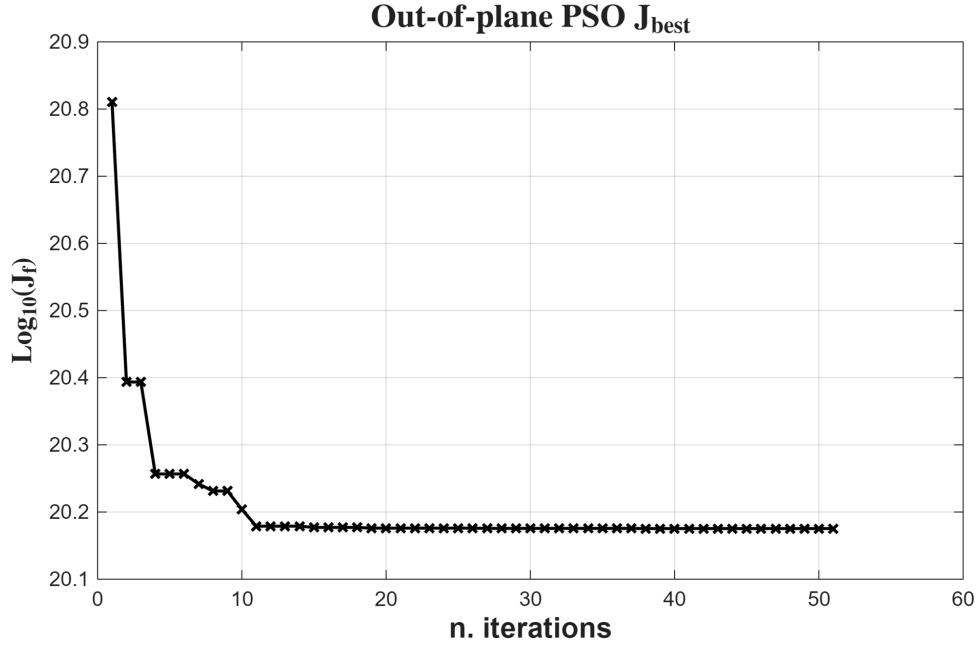


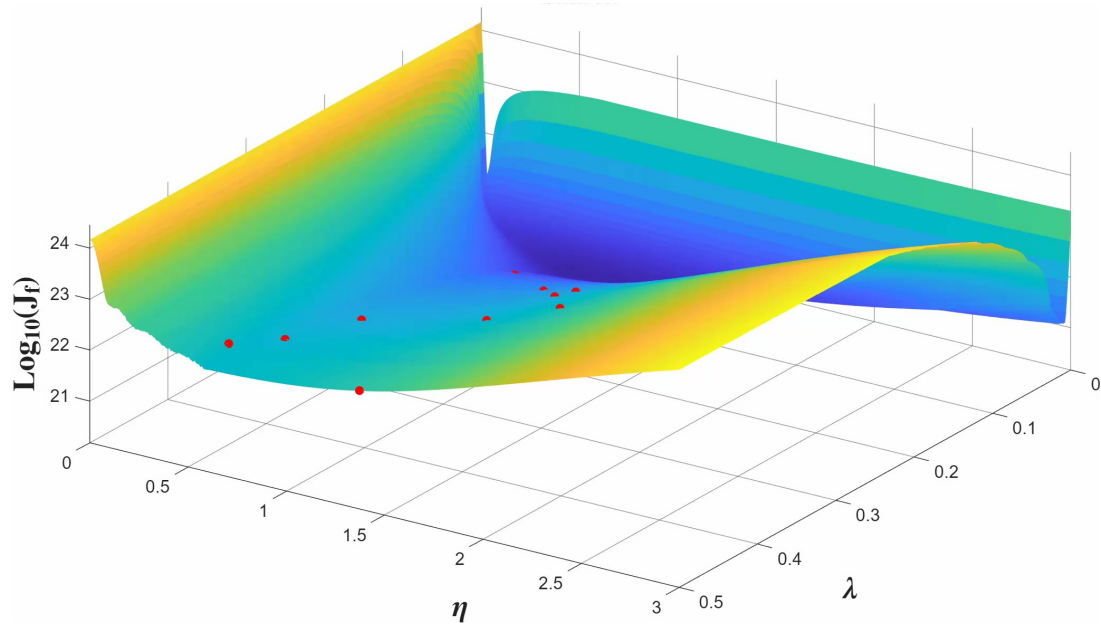
Figure 5.2: Convergence of the global best cost J_{best} over iterations.

The resulting optimal parameters were found to be:

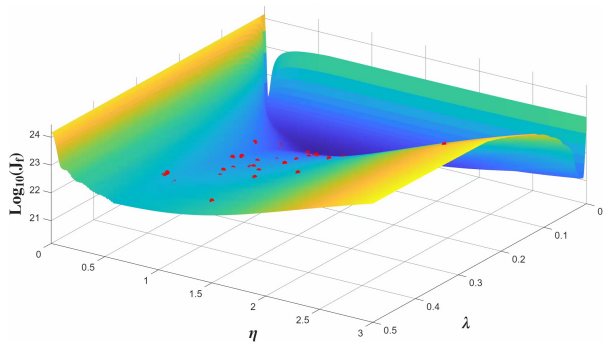
$$\eta = 0.6, \quad \lambda = 0.0423$$

These parameters yielded a control response characterized by minimal chattering and effective saturation usage. It is important to note that the optimal gain $\eta = 0.6$ coincides exactly with the saturation limit $U_{max} = 0.6$ set for this simulation. This indicates that the cost function, defined by Equation (5.3), forced η to its maximum allowable value.

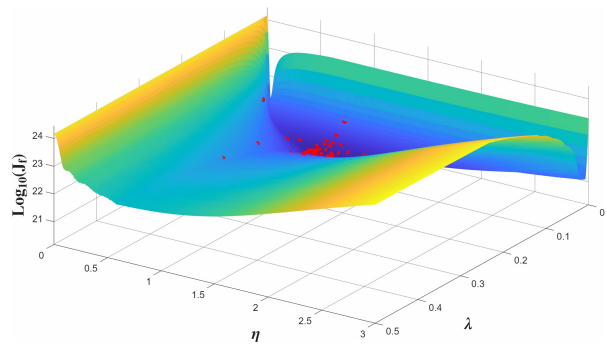
Although the PSO algorithm **does not guarantee convergence** to the global optimum, this is not strictly required in this context. The objective is to obtain an acceptable compromise between state convergence and control effort rather than the absolute minimum of J_f .



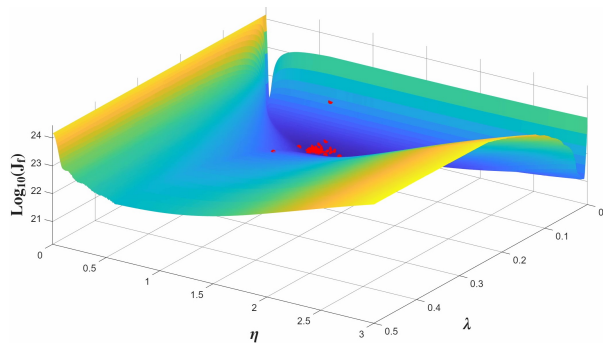
(a) Iteration 1



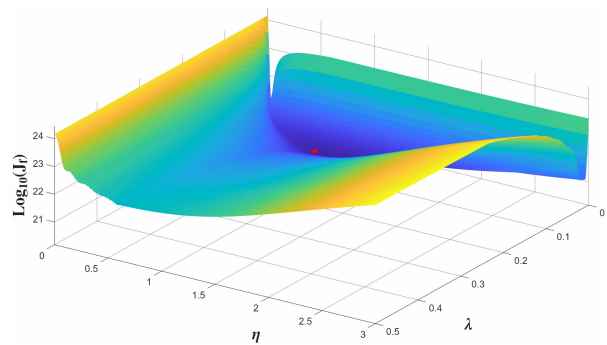
(b) Iteration 6



(c) Iteration 18



(d) Iteration 23



(e) Iteration 51

Figure 5.3: Evolution of the particle swarm during the PSO optimization process.

In Fig. 5.3, the evolution of the swarm during the PSO process is illustrated:

- **Fig. 5.3a:** the initial iteration of the PSO. The particles are randomly distributed over the cost function surface (represented on a logarithmic scale) within the defined search boundaries $\lambda \in [0, 3]$ and $\eta \in [0, 3]$, the major part of the particles are outside the image boundary.
- **Fig. 5.3c:** corresponding to iteration 18. The swarm begins to converge toward the global minimum, although several particles still explore different regions of the search space.
- **Fig. 5.3d:** by iteration 23, most particles have gathered around the optimal region, while a few continue to explore due to the stochastic nature of the PSO dynamics.
- **Fig. 5.3e:** the final iteration. The optimization process terminates according to the convergence criterion previously defined. At this stage, the particles are tightly clustered around the global minimum, although perfect convergence is not achieved because of the algorithm's inherent randomness.

Comparison between PSO and SQP

As observed in Figure 5.3a, the cost function appears macroscopically convex. However the objective here is to demonstrate why standard gradient-based algorithms exhibit convergence issues in this specific context. To do so, the PSO algorithm is compared against a classical deterministic solver, specifically Sequential Quadratic Programming (SQP).

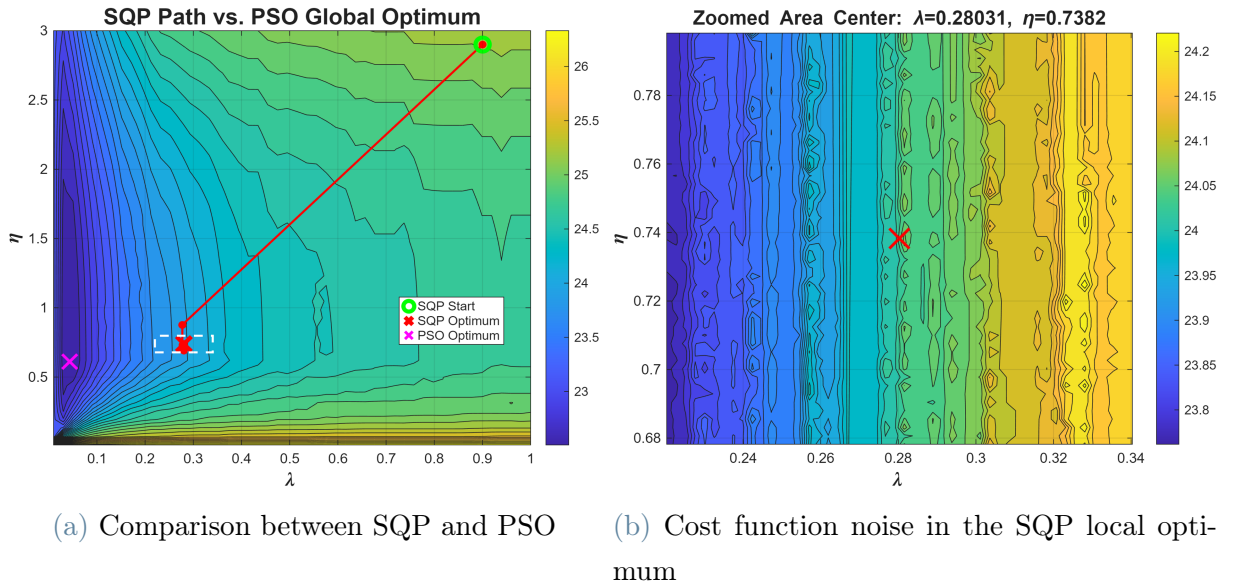


Figure 5.4: Analysis of the optimization landscape and solver performance.

In Figure 5.4a, the red line shows the optimization path generated by the SQP algorithm (solved using `fmincon` in MATLAB). It can be clearly seen that the SQP solver gets stuck in a fictitious local minimum. This local minimum is only attributable to the numerical noise inherent in the cost function, which is generated by the variable-step ODE solver used for the simulation. In contrast, the magenta cross represents the global minimum successfully identified by the PSO. Additionally, another distinct local minimum artifact can be observed near $\lambda = 0.57$, $\eta = 0.86$.

Figure 5.4b presents a magnified view of the area surrounding the SQP solution. This zoom clearly reveals the significant "roughness" and numerous local minima generated by the solver noise.

A fundamental limitation of the SQP approach in this specific control framework lies in the practical impossibility of computing an analytical gradient for the cost function J_f . The cost is evaluated by numerically integrating the closed-loop system dynamics, which includes discontinuous elements such as the actuator saturation function.

Formally, the optimization problem treats the cost function as a black-box mapping where the state trajectory is the solution to a discontinuous differential equation solved via a variable-step integrator (ODE15s). Because the control input hits hard saturation limits and the sliding surface introduces non-smooth transitions, the gradient ∇J_f does not exist in a classical sense at the switching points. Consequently, standard gradient-based solvers like SQP attempt to estimate the gradient numerically using finite differences:

$$\frac{\partial J_f}{\partial \theta_i} \approx \frac{J_f(\theta + \delta e_i) - J_f(\theta)}{\delta}$$

However, as shown in Figure 5.4b, the numerical integration of these stiff dynamics introduces high-frequency noise into the cost landscape. This noise renders finite difference approximations unreliable, causing the SQP solver to follow false gradients and become trapped in local minima that are essentially numerical artifacts.

Therefore, although the cost function could theoretically be smoothed by adjusting simulation techniques, most simply by tightening the integration tolerances of the ODE solver, this approach drastically increases the computational cost. Furthermore, it is not robust enough, as it does not guarantee the complete absence of numerical noise. Finally, as previously discussed, there is no guarantee that the cost function remains convex for the general case involving different parameters and cost function. Consequently, PSO represents a more robust and reliable choice for the considered optimization problem.

6 | Simulations and Results

Having established the theoretical foundations of the hybrid control architecture, comprising the ROE-based optimal guidance (Chapter 4), the robust Sliding Mode Tracking Layer (Chapter 4) and the PSO-based tuning methodology (Chapter 5), this chapter moves to the validation of the proposed system through high-fidelity numerical simulations.

While the previous chapters focused on the mathematical derivation of the control laws under ideal or simplified assumptions, this chapter assesses the system's performance in a realistic operational environment. The simulations are designed to bridge the gap between theory and practice by incorporating critical disturbances that were neglected in the guidance model, specifically the J_2 gravitational perturbation, atmospheric drag and sensor noise.

The primary objective is to demonstrate that the **hybrid architecture** can successfully transition the Chaser from a long-range rendezvous phase (characterized by high uncertainty) to a precise close-range docking, even when the internal models differ from the physical reality. To this end, the analysis is structured to isolate and quantify different performance metrics:

- **Section 6.1** defines the **Mission Scenario**, detailing the orbital parameters of the target (Cosmos 2506) and the specific uncertainty models for the Inertial Navigation System (INS) and initial state estimation.
- **Section 6.2** presents an **Open-Loop Error Characterization**, which quantifies the divergence between the linear guidance model and the nonlinear orbital reality. By isolating the effects of linearization errors and INS drift before the control loop is closed, this analysis establishes the disturbance magnitude that the Tracking Layer must compensate for.
- **Section 6.3** presents a **Comparative Analysis** of two distinct guidance strategies: a static strategy where the SMC is responsible for compensating the position error at the handover point, versus an adaptive re-optimization approach. This comparison highlights the efficiency gains achieved by updating the optimal trajectory at the

sensor handover point.

- **Section 6.4** evaluates the **Final Approach**, focusing on the safety-critical transition to the Artificial Potential Field (APF) guidance and the effectiveness of the velocity-based tracking.
- **Section 6.5** tests the system's **Robustness to Failure**, subjecting the control loop to severe actuator faults (stuck-open and blocked-closed thrusters) and testing the abort strategy.

6.1. Mission Scenario Definition

This section sets up the specific mission scenario used to test the spacecraft's hybrid control architecture. It establishes the primary objective of the mission: performing a time-optimal rendezvous starting from approximately 70 km behind the target, without exceeding a 20 kg propellant limit. To ensure the simulation accurately reflects operational realities, this scenario also incorporates uncertainties, specifically imperfect initial position data, J_2 effect and INS drift as the spacecraft travels.

Orbital Configuration and Initial Relative State

The target satellite is the Cosmos 2506 reconnaissance spacecraft. Its Sun-Synchronous Orbit (SSO) serves as the reference for the relative motion. The initial Keplerian elements for the simulation are summarized in Table 6.1.

Orbital Element	Value	Unit
Semi-major Axis (a)	7089	km
Eccentricity (e)	$6.017 \cdot 10^{-4}$	-
Inclination (i)	98.0877	deg
RAAN (Ω)	314.59	deg
Argument of Perigee (ω)	250.39	deg
True Anomaly (ν)	0	deg

Table 6.1: Target Orbit Parameters (Initial Conditions).

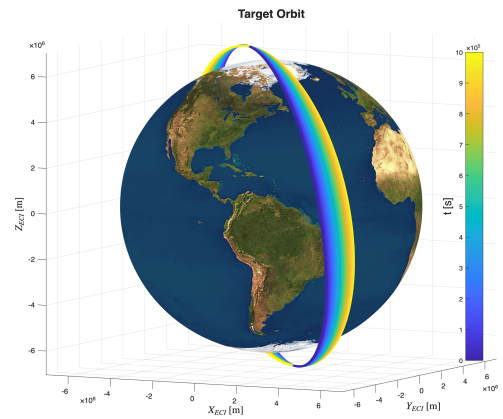


Figure 6.1: Visualization of the Target Orbit.

The Chaser is initially positioned approximately 70 km behind the target. The mission objective is to perform a time-optimal rendezvous with a 20 kg propellant limit (for

the optimization). Table 6.2 provides the initial Keplerian Elements differences. Using the Quasi-Nonsingular Relative Orbital Elements (ROE) defined in Equation (3.3), these differences are converted into the state vector $\delta\alpha_0$ shown in Table 6.3.

Rel. Parameter	Value	Unit
Semi-major Axis (Δa)	-1.2	km
Eccentricity (Δe)	0.0008	-
Inclination (Δi)	0.08	deg
True Anomaly ($\Delta \theta$)	-0.45	deg

Table 6.2: Initial Keplerian differences.

ROE	Value	Unit
δa	$-1.69 \cdot 10^{-4}$	-
$\delta \lambda$	$-7.83 \cdot 10^{-3}$	rad
δe_x	$-2.68 \cdot 10^{-4}$	-
δe_y	$-7.53 \cdot 10^{-4}$	-
δi_x	$1.39 \cdot 10^{-3}$	rad
δi_y	0	rad

Table 6.3: Initial Quasi-Nonsingular ROE.

Uncertainty Sources and Monte Carlo Setup

The chaser relative state is affected by multiple uncertainty sources, which motivate the hybrid structure of the proposed control system. These uncertainties arise from imperfect initial knowledge of the relative state, modeling approximations, and navigation error accumulation during open-loop flight.

Specifically, the main uncertainty contributions are grouped as follows:

- Relative state uncertainty, denoted as $\Delta \mathbf{x}$, which accounts for both the initial relative state uncertainty (originating from TLE-based orbit knowledge) and the modeling errors caused by linearization and neglected perturbations.
- Inertial Navigation System (INS) drift due to residual accelerometer bias, denoted as $e_{INS}(t)$.

To assess the robustness of the hybrid controller, a Monte Carlo analysis is performed. Initial position uncertainties are modeled in the RTN frame, reflecting the degradation of Two-Line Element (TLE) data accuracy over time. Following the analysis presented in [31], standard deviations for the error distribution are:

$$\sigma_R = 500 \text{ m}, \quad \sigma_T = 800 \text{ m}, \quad \sigma_N = 100 \text{ m}.$$

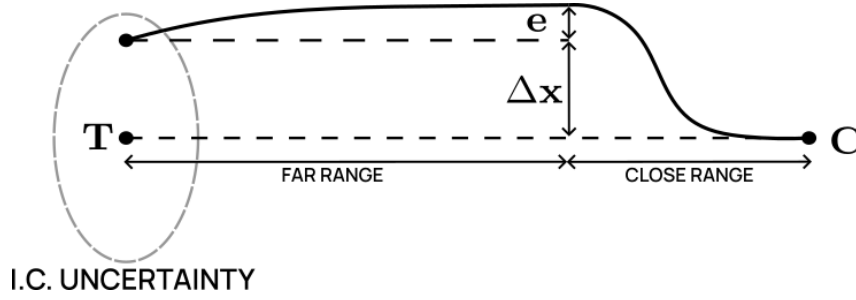


Figure 6.2: Conceptual schematic of the rendezvous maneuver under uncertainty.

Operational Mission Phases

The operational logic of the mission is visualized in Fig. 6.2. The dashed line represents the nominal time-optimal solution calculated at the start. However, due to initial state dispersion ($\Delta \mathbf{x}$), the Chaser's actual initial position is offset from the model. As the mission progresses through the **Far Range**, the internal INS accumulates a navigation error (e_{INS}) due to residual sensor bias, causing the actual trajectory (solid line) to diverge from the planned path. This open-loop drift persists until the Chaser crosses the 10 km sensor threshold.

The trajectory is formally divided into three distinct operational phases:

- **Far Range:** Initiated at ≈ 70 km. The system follows the open-loop optimization trajectory. Navigation is provided by the INS, which is subject to a $10 \mu g$ residual bias, leading to a position divergence of approximately 120 m after 1000 s.
- **Close Range:** Triggered at ≈ 10 km. Relative sensors provide measurements to close the SMC loop and compensate for the accumulated INS drift, guiding the Chaser back to the reference path.
- **Final Approach:** Employs the APF-based SMC to enforce collision avoidance through the cardioid-cissoid potential functions while maintaining velocity constraints.

6.2. Open-Loop Error Characterization

Before evaluating the closed-loop performance, it is essential to quantify the intrinsic errors arising from model linearization and inertial navigation drift. These effects represent disturbances that must be rejected by the tracking layer.

Reference Trajectory vs. Nonlinear Dynamics

The Guidance Layer generates a reference trajectory using the linearized ROE model. In an ideal scenario, applying the computed control profile would result in the spacecraft following this path exactly. However, the actual orbital environment is nonlinear and subject to perturbations (J_2 , drag) that are not captured by the guidance model.

Figures 6.3a and 6.3b illustrate this comparison. Figure 6.3a displays the reference trajectory generated by the optimizer, while Figure 6.3b shows the actual path flown by the spacecraft when the same control inputs are applied in the high-fidelity nonlinear propagator.

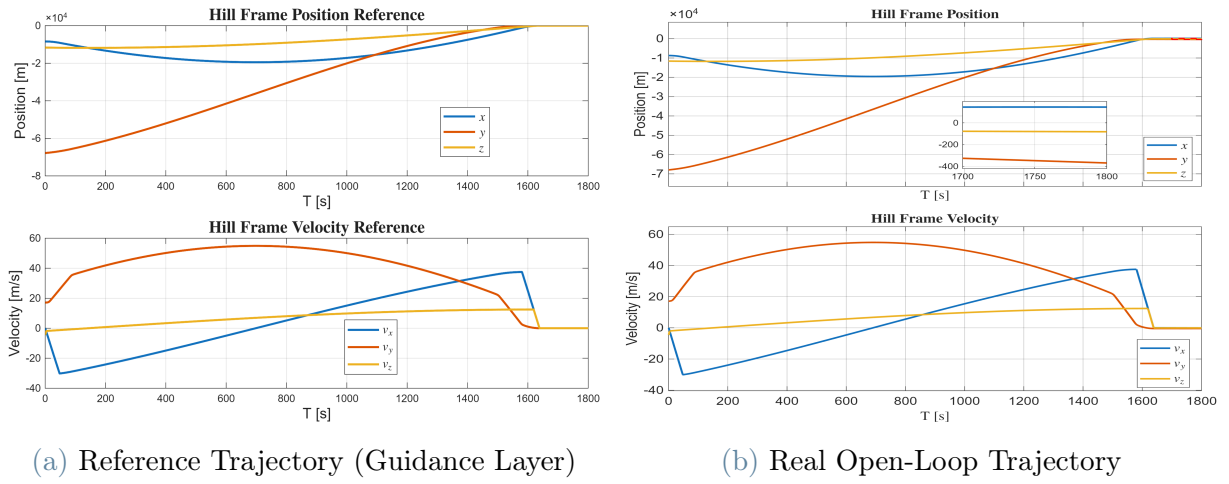


Figure 6.3: Comparison between the linear reference trajectory (left) and the real nonlinear open-loop trajectory (right). The inset on the right highlights the deviation at the target point.

The difference between the reference trajectory (Figure 6.3a) and real trajectory (Figure 6.3) represents the **Navigation Error**, plotted in Figure 6.4. This error in the open loop case arises because the linear guidance model cannot perfectly predict the nonlinear orbital mechanics. As observed in the figure, the error exhibits an oscillatory behavior corresponding to the orbital frequency. However, due to the short duration of the rendezvous maneuver relative to the orbital period, only a partial oscillation is visible, manifesting as a secular-like drift in the In-Plane components. This mismatch acts as a disturbance that the closed-loop Tracking Layer must reject.

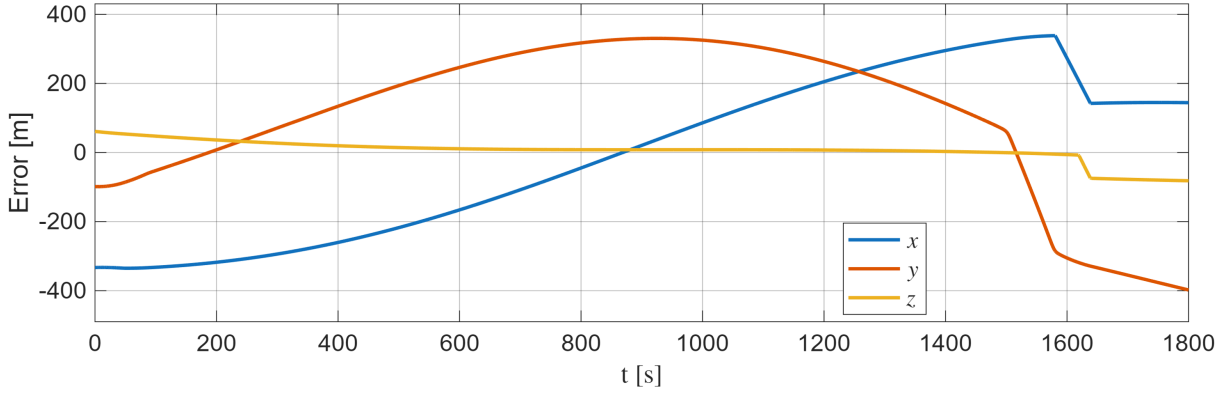


Figure 6.4: Navigation Error: Difference between the nonlinear real trajectory and the reference ($\|\mathbf{x}_{meas} - \mathbf{x}_{ref}\|$).

Inertial Navigation Drift

During the Far Range phase, the Chaser relies on the Inertial Navigation System (INS) describe in Section 2.3. The INS estimates the relative state by integrating the measured accelerations using clohessy-wiltshire equations. A critical characteristic of this navigation mode is the accumulation of error due to sensor bias.

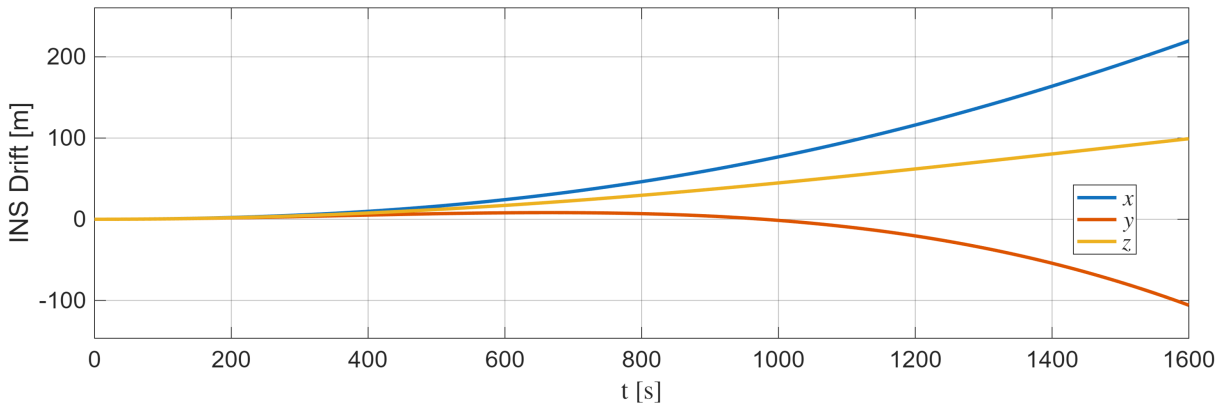


Figure 6.5: Divergence of the INS estimation error over time.

As shown in Figure 6.5, the INS error grows quadratically over time. It is important to note that the error starts at zero. This is because the INS propagates its state starting from its own initial estimate; it has no knowledge of any initial position uncertainty. Consequently, the plotted divergence represents purely the internal drift caused by the accelerometer residual bias ($10 \mu g$) accumulating during the maneuver.

6.3. Comparative Analysis of Guidance Strategies

To evaluate the trade-offs among computational complexity, propellant efficiency, and mission robustness, two distinct guidance architectures were simulated under identical uncertainty conditions. The core objective of this analysis is to assess how the hybrid control system manages the transition from the Far Range phase (characterized by navigation uncertainty) to the Close Range phase (characterized by precise relative sensing).

It is important to note that, for these simulations, the final approach control is **disabled**. This isolation ensures that the evaluation focuses exclusively on the performance of the optimal control coupled with the Sliding Mode Control (SMC). A dedicated analysis of the final approach phase is provided subsequently in Section 6.5.

The mission profile dictates that at a relative distance of $\rho^* \approx 10$ km, the chaser's navigation solution switches from an estimated state to a precise relative measurement provided by onboard sensors. This “Sensor Handover” event reveals the true accumulated position error, necessitating a control response.

Two strategies are compared to handle this transition:

Scenario A: Static Reference Tracking

In this scenario, the reference trajectory is computed exclusively at the mission start (t_0) using the initial estimate of the chaser's state. The Guidance Layer remains static throughout the maneuver, forcing the controller to react to navigation updates as if they were disturbances.

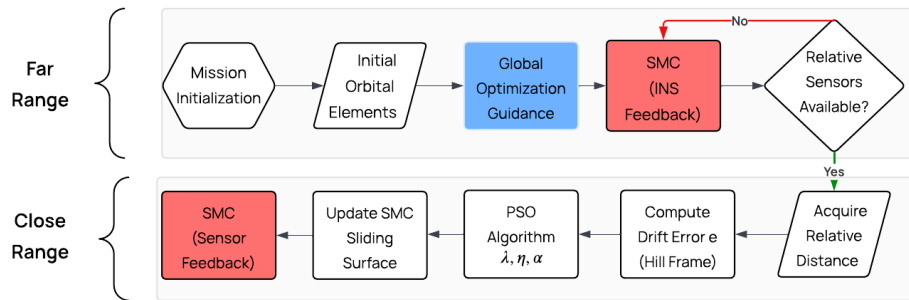


Figure 6.6: Flowchart of the control architecture for Scenario A.

The control algorithm follows the sequence illustrated in Figure 6.6:

- **Far Range Phase:** The mission initiates with the generation of the reference trajectory via global optimization. The chaser tracks this path using the SMC with

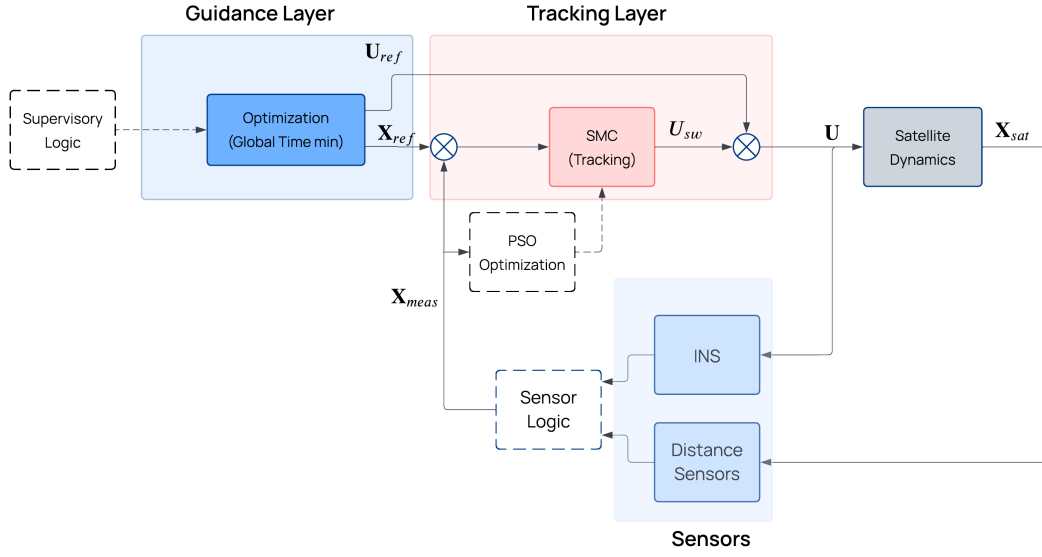


Figure 6.7: Control Scheme for Scenario A.

INS feedback, during which navigation drift naturally accumulates due to the lack of relative measurements.

- **Close Range Phase:** At the 10 km handover, relative sensors become active. The system computes the drift error (e) and executes the **PSO** routine to **update** the SMC parameters (λ, η, α) , ensuring the specific accumulated error is rejected efficiently.
- **Final Approach Phase:** The Final Approach controller is disabled to test the performance of the Hybrid controller.

Nominal Mission Performance

In this simulation, the system is initialized without any position uncertainty. Consequently, the Chaser possesses exact knowledge of both its own state and the Target's state. The error term Δx (referenced in Figure 6.2) is therefore composed solely of the linearization error inherent to the dynamic model Equation (3.5), as previously illustrated in Figure 6.4.

The corresponding optimal control acceleration profile generated by the guidance layer at t_0 is presented in Figure 6.8.

Figure 6.10 illustrates the performance of the Sliding Mode Controller throughout the approach. As shown in Figure 6.10a, for $t < t^* \approx 1380$ s, the chaser relies exclusively

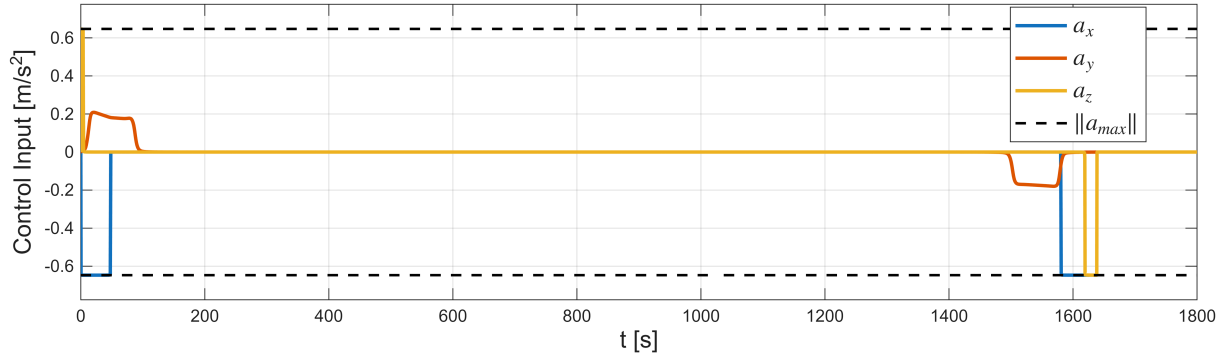


Figure 6.8: Control profile generated by the optimal control solver.

on the Inertial Navigation System (INS) for state estimation. During this phase, the **navigation error** (Figure 6.4), induced by the model linearization, is unobservable. The SMC regulates the estimated error, which remains close to zero, resulting in the minimal control activity observed in Figure 6.10b.

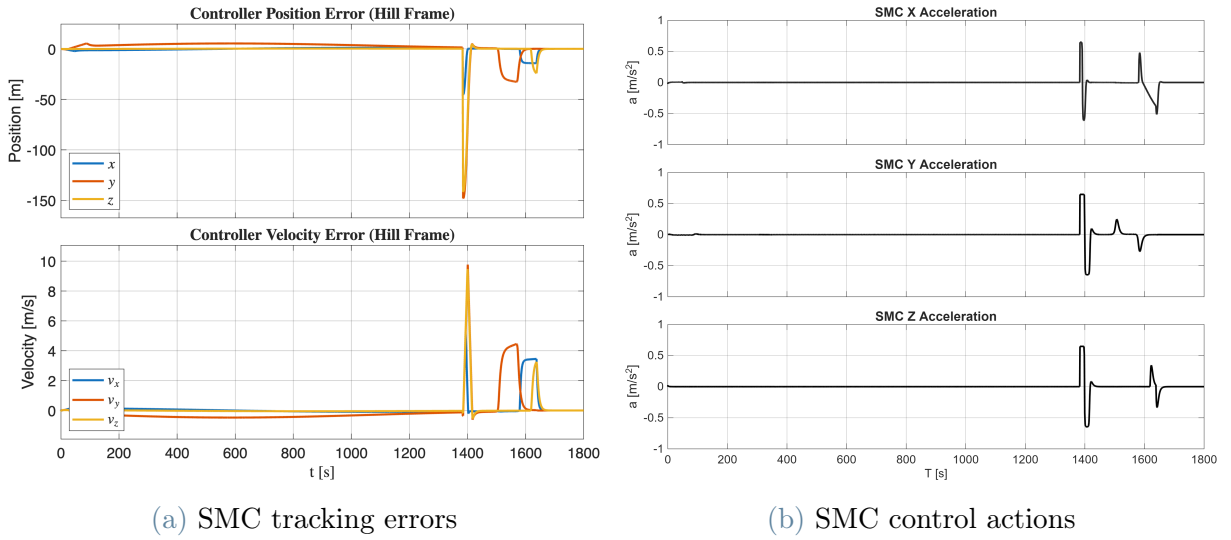


Figure 6.10: Evolution of the tracking error in input to the SMC (a) and corresponding control accelerations (b) for Scenario A.

At the sensor handover epoch (t^*), distance sensors measured the real relative separation, revealing the true accumulated navigation error (Spikes in Figure 6.10a). This transition produces a sharp discontinuity in the error signals across all three axes. It is important to note that these spikes do not represent instability, but rather the instantaneous correction of the state estimate.

Upon detection of the true error $\mathbf{e}(t^*)$, the PSO routine is triggered to retune the SMC parameters $\{\lambda, \eta, \alpha\}$. The optimization yields the following optimal parameters:

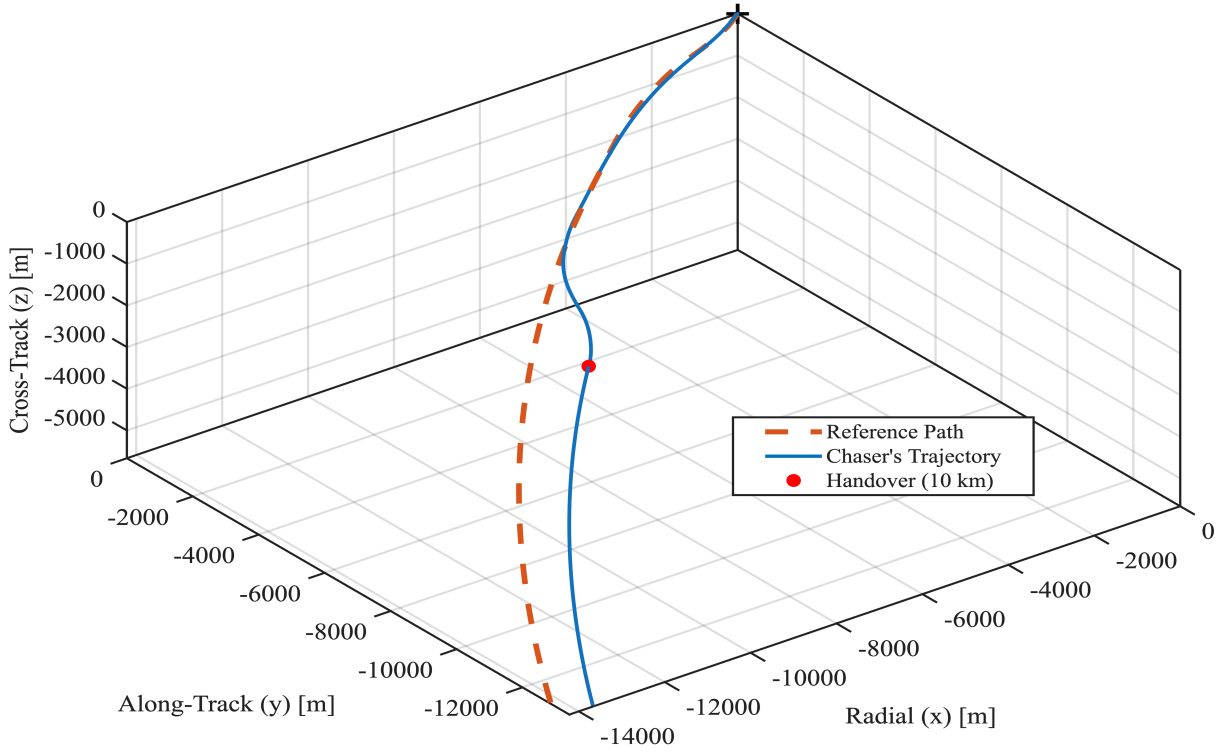


Figure 6.9: Chaser's trajectory in (Range < 20 km), after the Handover point the SMC compensate the Navigation error.

Control Axis	λ	η	α
In-Plane (xy)	$\lambda_x = 0.223$ / $\lambda_y = 0.136$	0.647	1.0
Out-of-Plane (z)	0.138	0.647	1.0

Table 6.4: Optimized SMC parameters for Scenario A.

Notably, the switching gain η for both planes converges to the actuator saturation limit ($U_{max} \approx 0.65$). This confirms the theoretical behavior discussed in Chapter 5, where the optimizer maximizes the available control authority to reject the matched disturbance in minimum time.

Following the re-optimization, the linearization error is rapidly suppressed. However, approaching $t = 1500$ s, a residual tracking error re-emerges. This deviation coincides with the final braking maneuver commanded by the guidance layer. Since the optimal acceleration profile is derived using the linear ROE model, it cannot perfectly replicate the nonlinear orbital dynamics during the braking maneuver. Consequently, the actual spacecraft trajectory diverges slightly from the planned reference, forcing the SMC to compensate for the modeling mismatch.

Monte Carlo Simulations

This section presents the results of the Monte Carlo analysis performed for Scenario A.

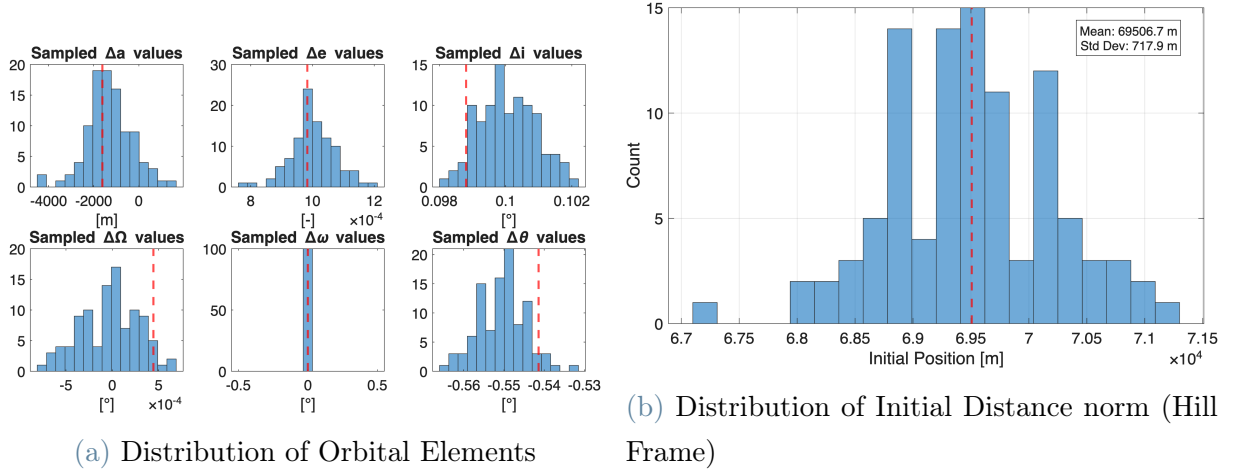


Figure 6.11: Initial statistical distribution of the Chaser's state for the Monte Carlo analysis.

Figure 6.11a displays the distribution of the sampled Orbital Elements. It is important to note that the Argument of Perigee (ω) is set to zero for all samples to avoid singularities, given the near-circular nature of the target orbit.

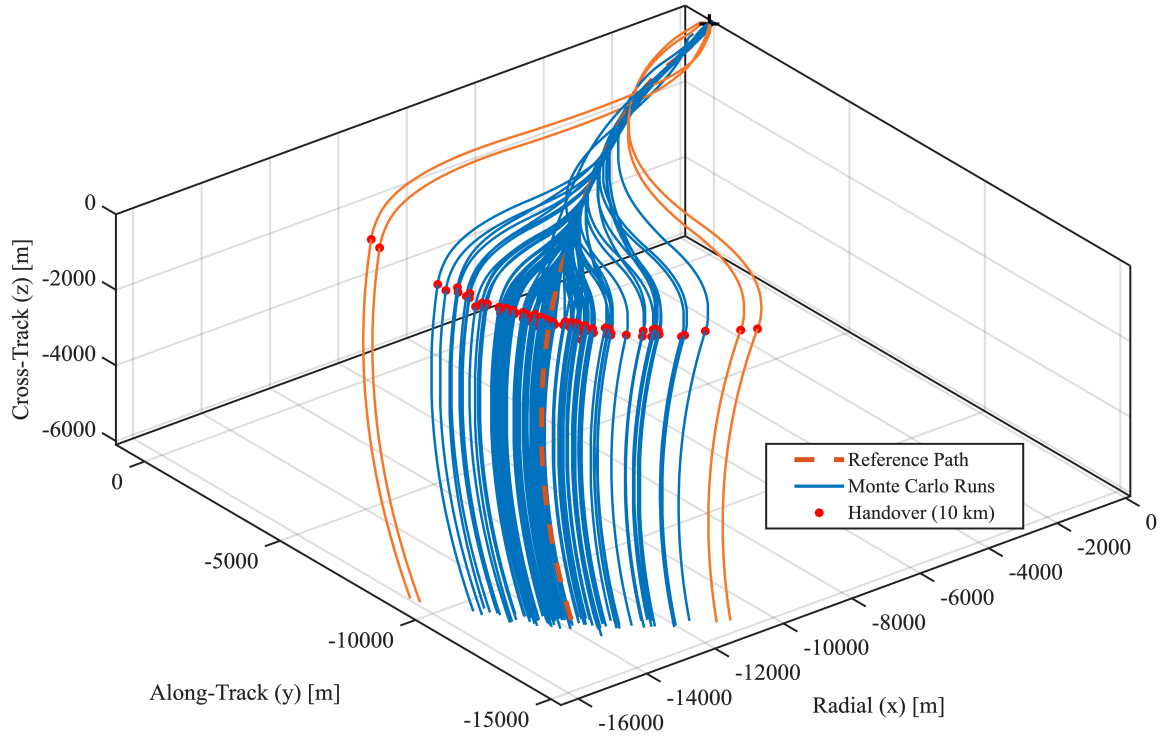


Figure 6.12: Monte Carlo trajectories (Range < 20 km).

Figure 6.12 presents the relative trajectories for all Monte Carlo simulation runs, focusing on the final 20 km of the approach. The plot highlights the system's behavior at the sensor handover point, indicated by the red markers. The approach can be divided into two distinct phases:

- **Phase 1 ($t < t^*$):** Prior to the handover, the Chaser operates without precise knowledge of its accumulated navigation error. The controller tracks the static reference trajectory relying solely on Inertial Navigation System (INS) feedback. This lack of absolute state updates leads to a visible dispersion of the trajectories.
- **Phase 2 ($t \geq t^*$):** Upon sensor handover, the PSO-tuned SMC utilizes the newly acquired precise relative measurements to rapidly nullify the accumulated drift. Unlike in Scenario B, the reference trajectory in Scenario A remains fixed. Consequently, the controller forces the Chaser to aggressively converge back to the original nominal path. A slight increase in tracking error is observed in the final phase, which corresponds to the braking maneuver discussed previously.

The trajectories highlighted in orange represent worst-case scenarios where the random initial conditions generate a substantial accumulated error by the time of handover. In these instances, the sudden availability of true state feedback creates a massive instantaneous tracking error. Even with the optimally tuned SMC, forcing the spacecraft back to the fixed reference path induces a transient overshoot of approximately 5%. Given the spatial scale of the maneuver, this translates to an absolute error on the order of hundreds of meters. These specific cases highlight a fundamental limitation of static reference tracking: encountering a large navigation error at the handover point demands aggressive control efforts that inevitably lead to overshoots.

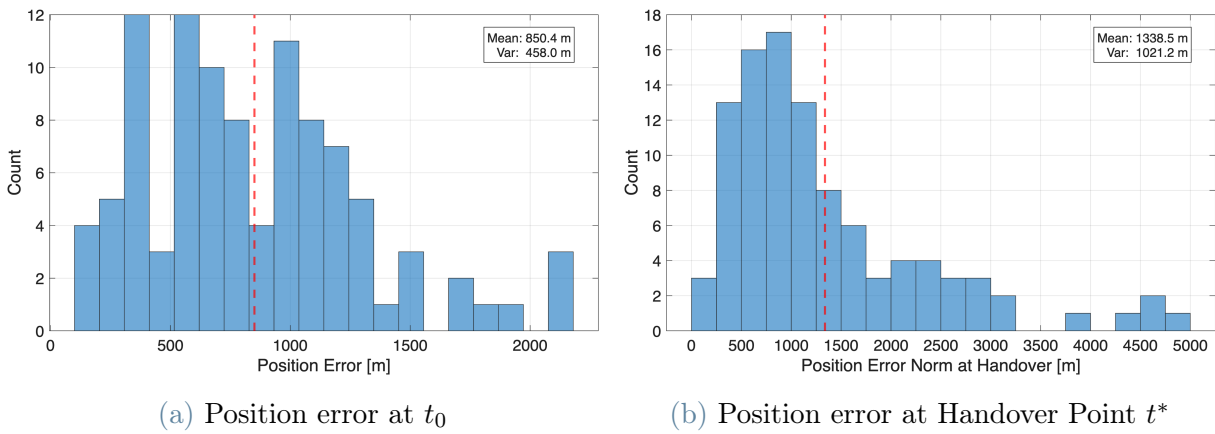


Figure 6.13: Comparison of the position error relative to the reference path at initialization (t_0) and at the sensor handover epoch t^* .

Figure 6.13 compares the position error distribution at the mission start (t_0) against the error at the sensor handover epoch (t^*). As expected, the dispersion grows significantly over time (as seen in Figure 6.13b) due to the absence of absolute relative distance measurements during the INS-only phase. Furthermore, the outliers present in Figure 6.13b directly correspond to the orange trajectories discussed previously, illustrating the huge positional discrepancies that the SMC must compensate after t^* .

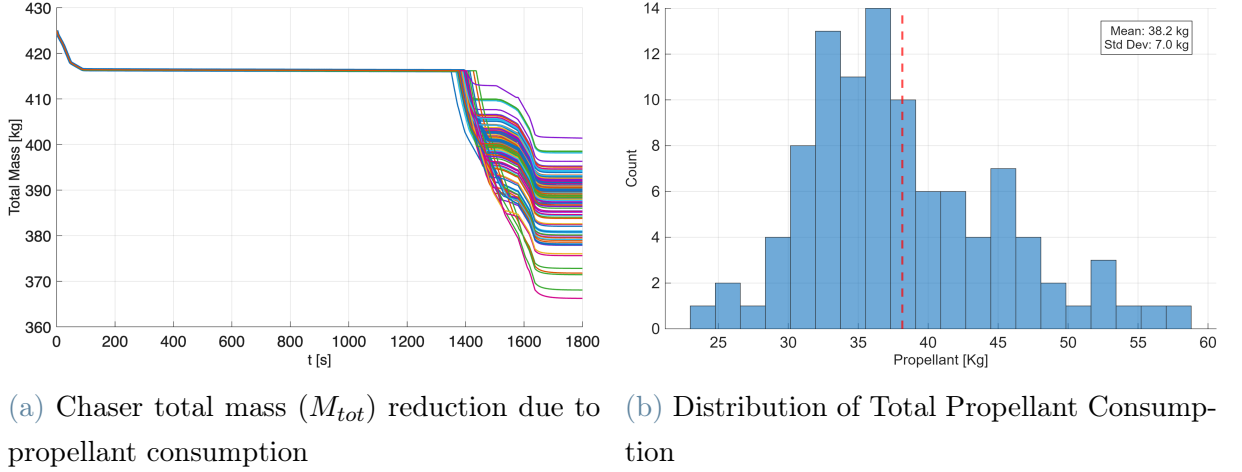


Figure 6.14: Analysis of propellant usage during Monte Carlo runs.

Figure 6.14 analyzes the propellant consumption across the simulated scenarios. The mass evolution profile (Figure 6.14a) clearly delineates three distinct mission phases: the initial boost, the coasting phase, and the terminal phase. During this final segment, the Chaser first expends propellant to compensate for the accumulated navigation error, which is then immediately followed by the terminal braking maneuver. For trajectories with exceptionally high initial drift, these two corrective accelerations overlap and are superimposed.

Finally, Figure 6.15 presents the distribution of the optimal SMC parameters $\{\lambda, \eta, \alpha\}$ generated by the PSO for each simulation run. The results confirm the theoretical expectations regarding the controller's behavior:

- **Switching Gain (η):** For both the in-plane (η_{xy}) and out-of-plane (η_z) controllers, the gain consistently converges to the actuator saturation limit ($U_{max} \approx 0.65$). This indicates that the optimization algorithm correctly identifies that maximizing control authority is necessary to reject the large initial navigation errors.
- **Sliding Surface Slope (λ):** The parameters λ_x , λ_y and λ_z exhibit a wider distribution. This variance demonstrates the adaptability of the PSO; it tunes the

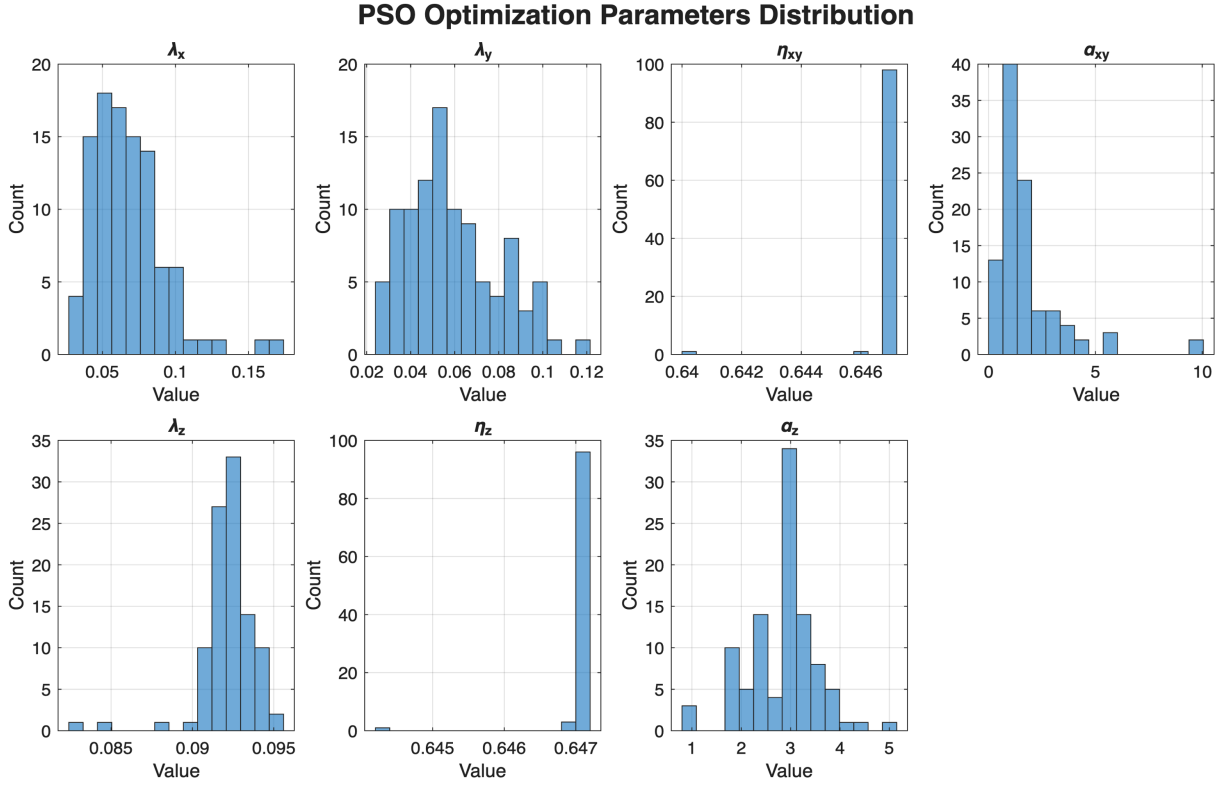


Figure 6.15: Statistical distribution of the SMC parameters optimized by PSO across the Monte Carlo runs.

slope of the sliding surface to tailor the convergence rate to the specific initial state dispersion of each run, balancing tracking speed against overshoot.

- **Hyperbolic Tangent Slope (α):** The distribution of α reflects the optimizer's mechanism for chattering alleviation. By tuning this parameter, the PSO effectively smooths the discontinuous control action near the sliding manifold, ensuring that tracking precision is maintained without inducing high-frequency oscillations.

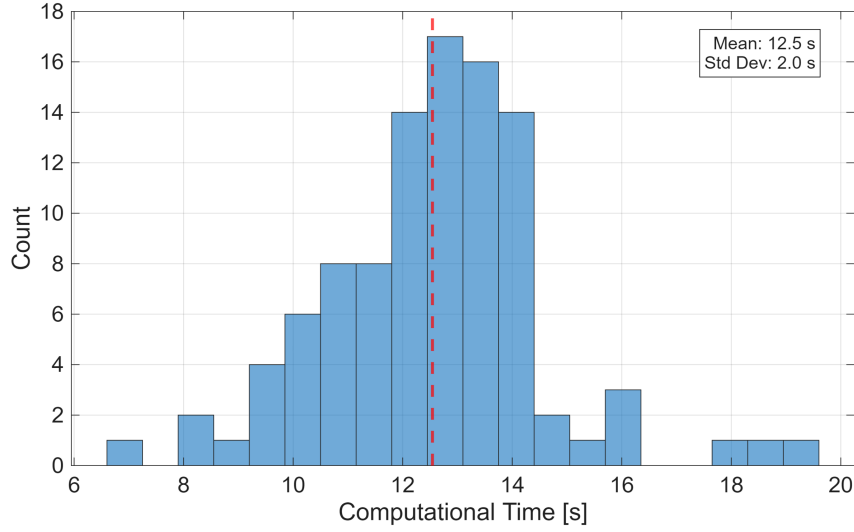


Figure 6.16: Statistical distribution of the rendezvous time delay (Control Lag) for Scenario A. The histogram displays the difference between the actual convergence time (where $\|\rho\| < 1$ [m]) and the nominal optimal duration (T_{opt}).

The histogram reveals a consistent positive lag. This behavior is inherent to the static guidance strategy (Scenario A): the open-loop optimal control profile terminates at t_{opt} . However, due to the modeling mismatch involved in the final braking maneuver, the Chaser arrives with a residual deviation. The feedback loop must then actively suppress this error, extending the mission duration beyond the planned optimal time to achieve asymptotic convergence.

Scenario B: Adaptive Re-Optimization

In this scenario, the guidance architecture is adaptive. The “Sensor Handover” event at ρ^* km serves as a trigger for the supervisory logic to re-engage the optimization solver. Unlike the static approach, the Guidance Layer effectively discards the obsolete reference, treating the current true position not as an error to be corrected, but as a new initial condition.

The control algorithm follows the sequence illustrated in Figure 6.6

- **Far Range Phase:** Similar to the previous scenario, the mission begins with a reference trajectory generated via global optimization. The chaser follows this path using SMC with INS feedback, while navigation drift accumulates as a result of the $10 \mu g$ residual bias.
- **Close Range Phase:** Upon acquiring precision relative measurements at the 10 km

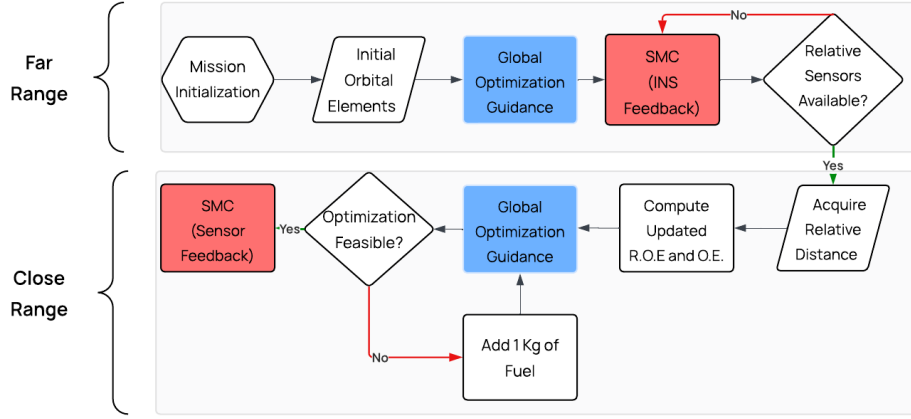


Figure 6.17: Flowchart of the control architecture for Scenario B.

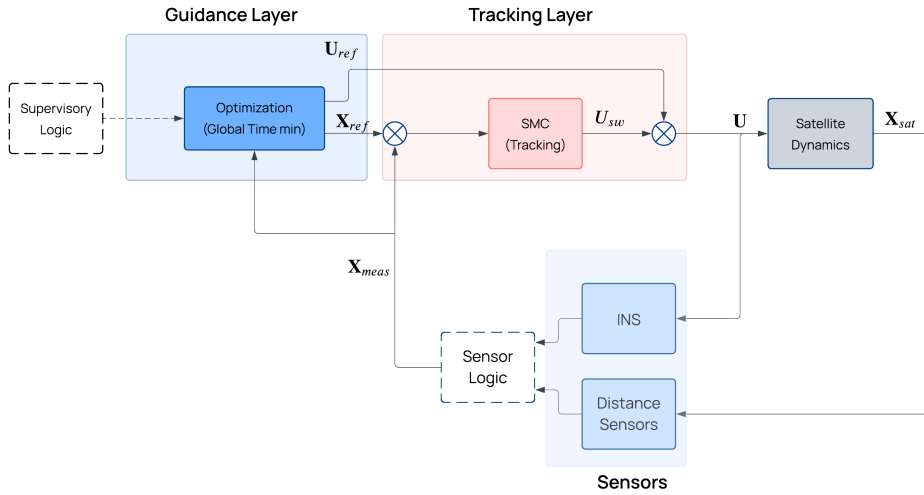


Figure 6.18: Control Scheme for Scenario B.

threshold, the system triggers a Mid-Course Update. The optimization algorithm computes a new time-optimal trajectory originating from the current state. By defining the chaser's actual location as the new starting point, the updated path becomes dynamically smooth and feasible, eliminating the need for the SMC to "fight" the navigation error.

- **Final Approach Phase:** The Final Approach controller is disabled to test the performance of the Hybrid Controller.

The re-optimization algorithm is detailed in Algorithm 6.1. Because the primary mission objective is time minimization, the initial search time window is bounded by $[0, T_{opt} - t^*]$. If a feasible optimization solution cannot be found within the remaining time $T_{opt} - t^*$

using the remaining propellant ($M_{prop}(t_0) - M_{prop}(t^*)$), the allowed propellant budget is iteratively **increased** in 1 kg increments. This iterative relaxation continues until a predefined maximum propellant limit is reached (in this case, an additional 10 kg). If the problem remains infeasible even after reaching the maximum propellant limit, the time window is subsequently **expanded**. All of these parameters and step sizes can be tuned according to specific mission requirements.

Algorithm 6.1 Scenario B: Adaptive Optimization (Structure Only)

Require: $\mathbf{x}_0$ (Current State), M_{prop}^{init} , M_{prop}^{max} , T_{range}

```

1: Solved  $\leftarrow$  FALSE
2: while Solved == FALSE and  $T_{range} \leq T_{limit}$  do
3:    $M_{curr} \leftarrow M_{prop}^{init}$  {Reset propellant Budget}
4:   while Solved == FALSE and  $M_{curr} \leq M_{prop}^{max}$  do
5:     Run Bisection Search:
6:      $T_{low} \leftarrow 0$ ,  $T_{high} \leftarrow T_{range}$ 
7:     while  $|T_{high} - T_{low}| > \varepsilon$  do
8:        $T_{test} \leftarrow \frac{T_{low} + T_{high}}{2}$ 
9:        $[J^*, \mathbf{u}, \mathbf{x}, \text{flag}] \leftarrow \text{Optimization}(\mathbf{x}_0, M_{curr}, T_{test})$ 
10:      if flag == Feasible then
11:         $T_{high} \leftarrow T_{test}$  {Feasible: Try faster}
12:        Solved  $\leftarrow$  TRUE
13:        Store Best Solution:  $\mathbf{u}^*$ ,  $\mathbf{x}^*$ 
14:      else
15:         $T_{low} \leftarrow T_{test}$  {Infeasible: Need more time}
16:      end if
17:    end while
18:    if Solved == FALSE then
19:       $M_{curr} \leftarrow M_{curr} + \Delta_{prop}$  {Adaptive: Add Propellant}
20:    end if
21:  end while
22:  if Solved == FALSE then
23:     $T_{range} \leftarrow 2 \times T_{range}$  {Fallback: Double Time Horizon}
24:  end if
25: end while
26: return Optimal Control Reference  $\mathbf{u}^*$  and State Reference  $\mathbf{x}^*$ 

```

Nominal Mission Performance

The simulation setup remains identical to Scenario A, with the system initialized with zero position uncertainty. For reference, the baseline optimal control acceleration profile, generated by the guidance layer at mission start (t_0), is presented again in Figure 6.19.

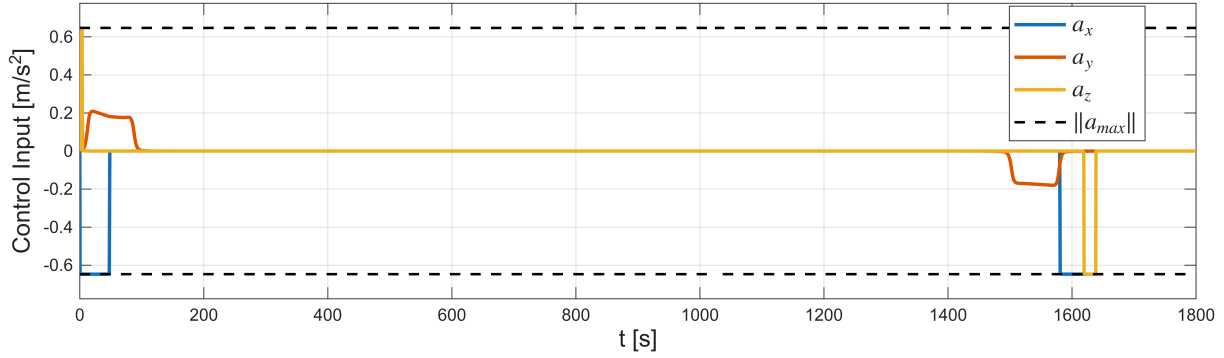


Figure 6.19: Nominal control profile generated by the optimal control solver at t_0 .

When the actual relative distance becomes observable at the sensor handover epoch t^* , the supervisory logic triggers the adaptive response. The updated Relative Orbital Elements (ROE), derived from the precise range measurements, are used to initialize a second optimization problem. This generates a new reference trajectory and a corresponding feedforward control profile starting from the current state.

Figure 6.20 illustrates the composite feedforward acceleration profile resulting from the two optimization stages. The first segment ($t < t^* \approx 1390$ s) corresponds to the initial solution computed at t_0 . Following the handover, the second optimization computes the corrective maneuver. It can be observed that the solver generates a new bang-bang control sequence.

Unlike Scenario A, where the SMC fights the error, here the acceleration profile is optimal; it exploits the natural system dynamics to correct the state deviation.

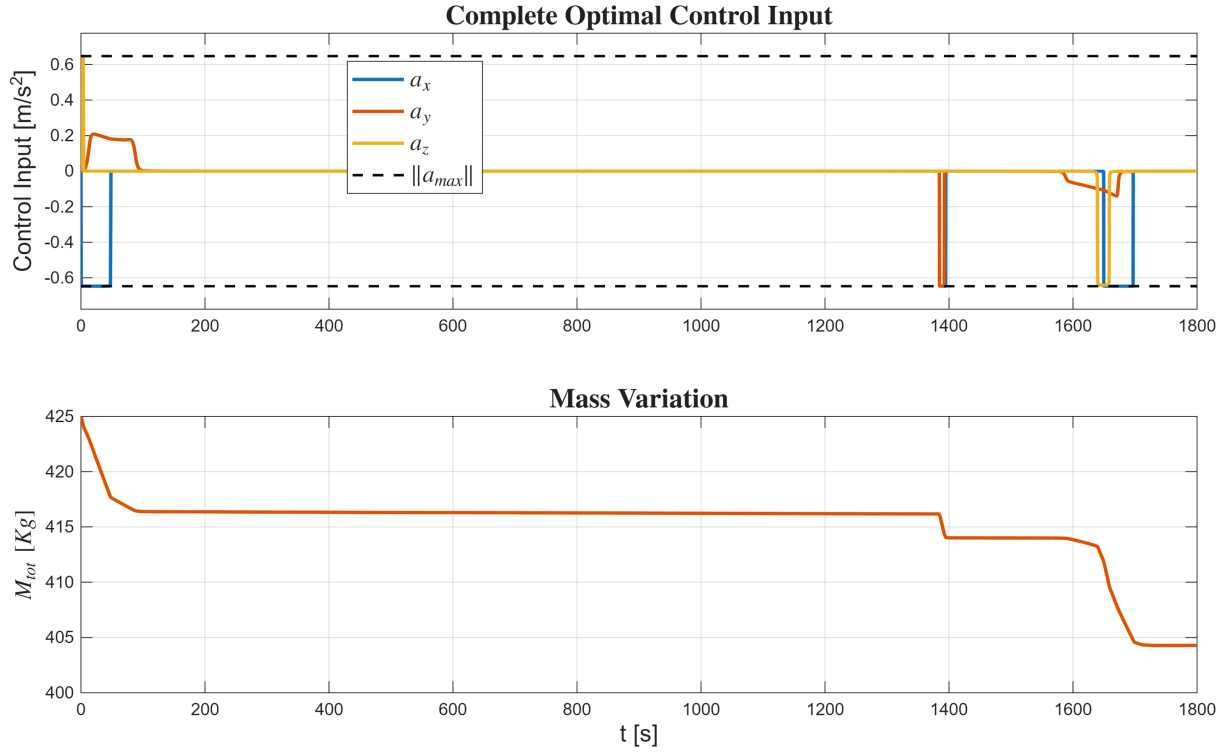


Figure 6.20: Composite control profile resulting from the initial and adaptive optimizations.

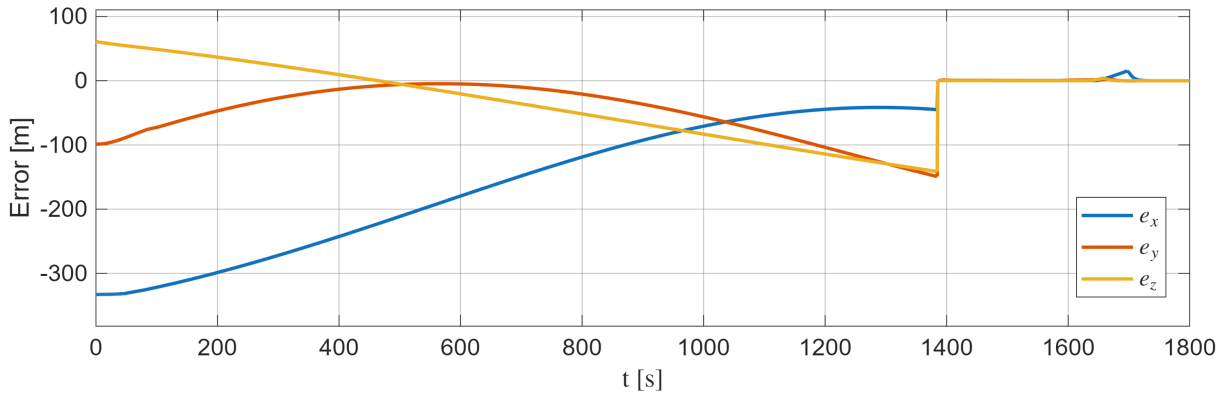


Figure 6.22: Navigation Error ($\|\mathbf{x}_{meas} - \mathbf{x}_{ref}\|$) in Hill Frame (Scenario B).

Figure 6.22 displays the navigation error, defined as the difference between the reference path and the actual trajectory. At the handover epoch t^* , the error drops instantly to zero because the reference trajectory is redefined to start exactly at the current measured state. Although the new optimal path is still subject to model linearization errors, these deviations are significantly smaller due to the reduced relative distance to the target. Furthermore, since the SMC now has access to precise relative measurements, any residual linearization error is actively compensated.

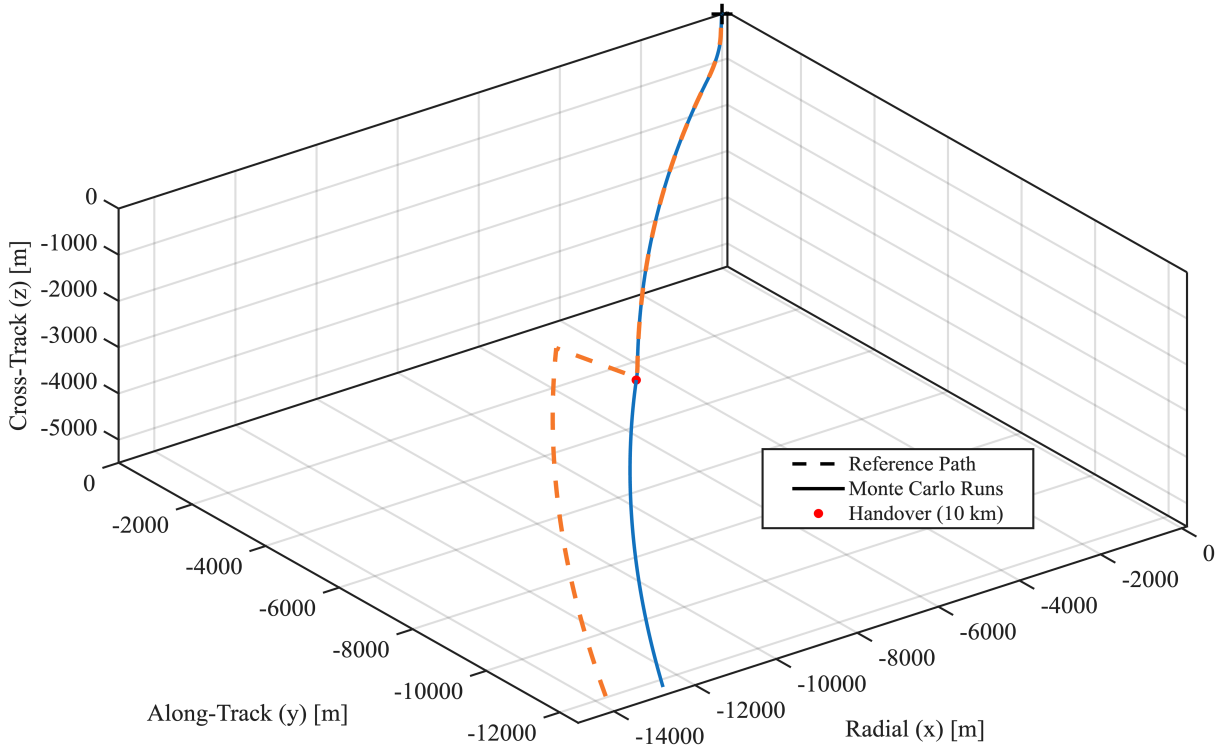


Figure 6.21: Chaser's trajectory in (Range < 20 km)

Monte Carlo Simulations

This section presents the results of the Monte Carlo analysis for Scenario B.

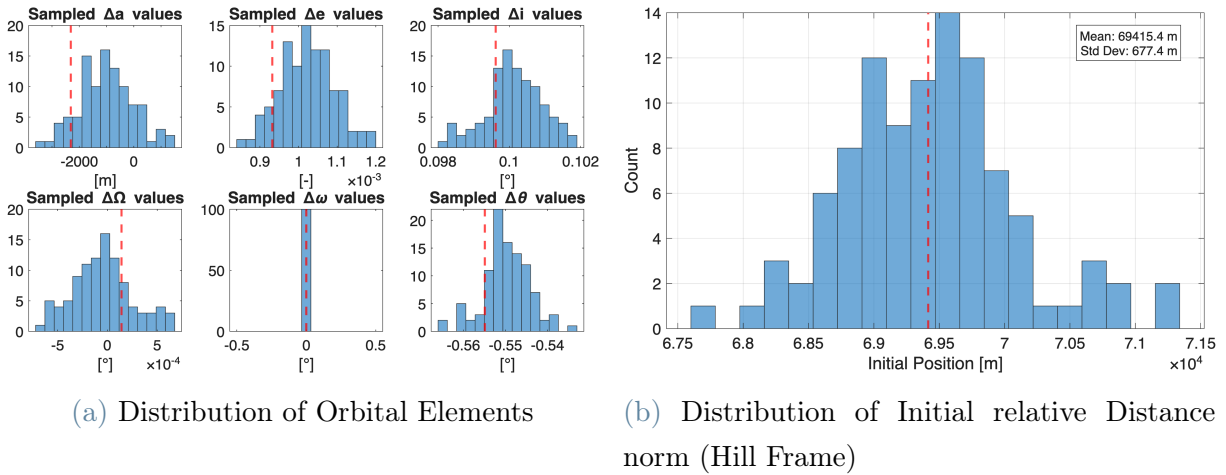


Figure 6.23: Initial statistical distribution of the Chaser's state for Scenario B. The red lines represent the nominal state.

Figure 6.23 confirms that the initialization of the samples is consistent with the uncertainty model defined in Section 6.1. Figure 6.23a displays the sampled Orbital Elements.

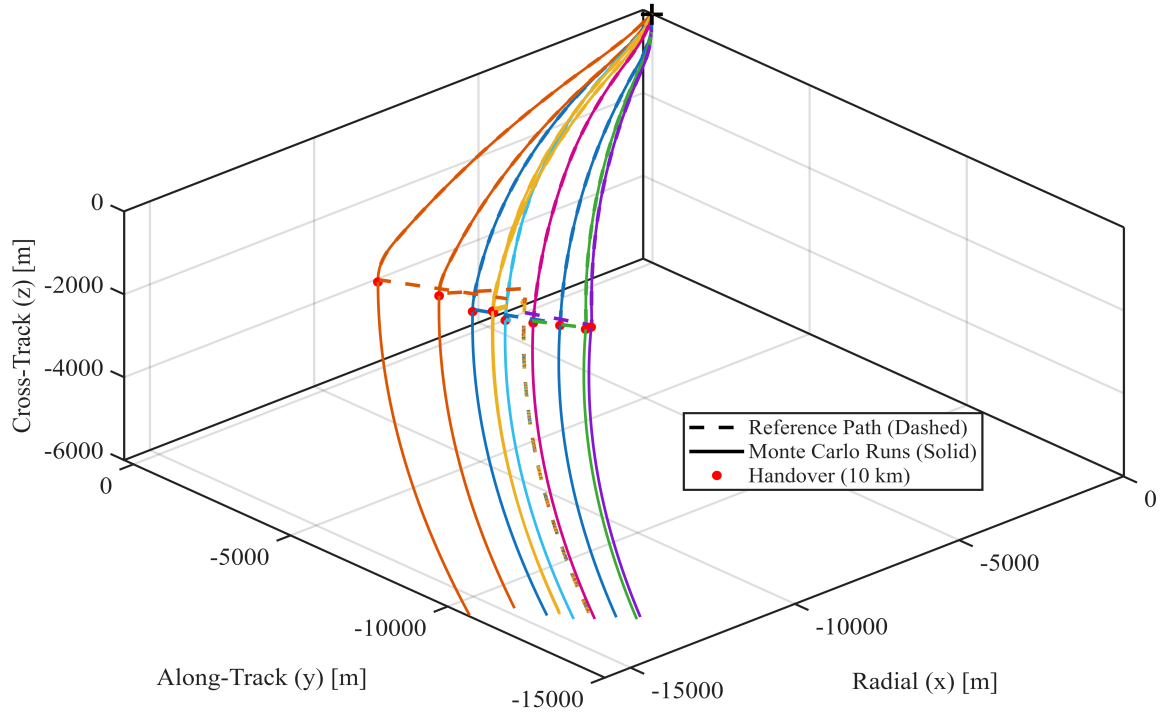


Figure 6.24: Monte Carlo trajectories (Range < 20 km), first 10 runs.

The relative trajectories for the first five runs are illustrated in Figure 6.24. Only a subset of runs is shown to maintain clarity. A distinct feature of Scenario B is observable at the sensor handover time t^* :

- **Phase 1** ($t_0 \rightarrow t^*$): The Chaser has no knowledge of the initial execution error. Consequently, the reference trajectory is identical for all runs (overlapping in the figure) and the controller fights the accumulated drift.
- **Phase 2** ($t \geq t^*$): At the handover, the guidance logic re-optimizes the path starting from the Chaser's actual measured position. This causes the reference trajectory to “jump” and align with the specific state of each run. As a result, the Chaser tracks a new, feasible path rather than attempting to converge back to the original nominal trajectory.

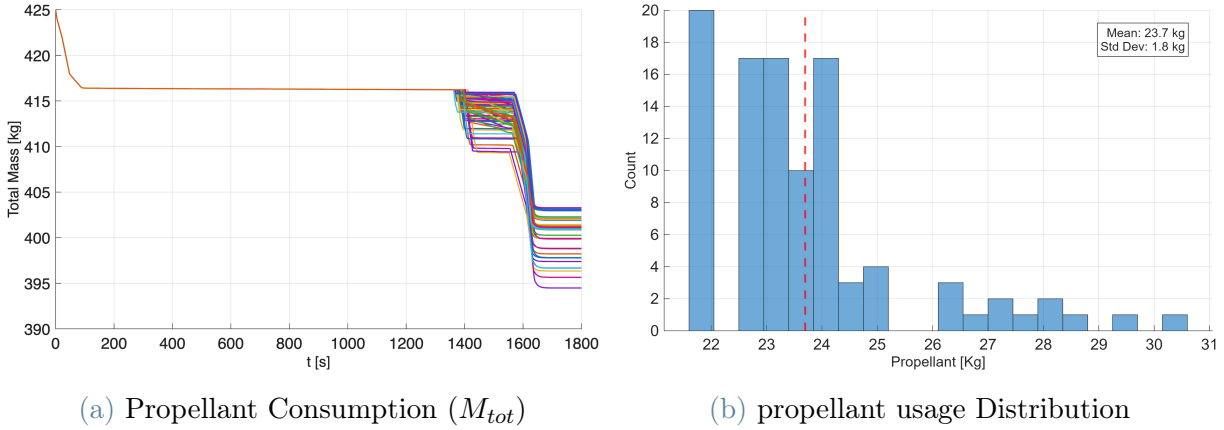


Figure 6.25: Analysis of propellant usage during Monte Carlo runs.

Figure 6.25 analyzes the propellant consumption by examining the evolution of the Chaser's total mass. Compared to Scenario A, the propellant consumption is significantly lower. Three distinct propellant consumption events are observable: the initial acceleration burn, the corrective maneuver at t^* (generated by the second optimization) and the final braking burn.

The distribution of total propellant consumption (Figure 6.25b) reveals a mean usage of 23.7 [kg]. This indicates that the adaptive strategy (Scenario B) is significantly more efficient, requiring on average only 20% more propellant than the nominal baseline, compared to the 100% increase observed in Scenario A. Furthermore, the distribution exhibits a clustered pattern. This behavior derives from the optimization logic: if the trajectory error is too large for the current propellant allocation, the algorithm incrementally adds 1 kg of propellant. Consequently, the final propellant consumption values tend to cluster around these discrete increments.

Analogous to the previous analysis, Figure 6.26 presents the statistical distribution of the rendezvous timing errors. However, strictly distinct from Scenario A, this figure compares two specific datasets:

- **Nominal Re-optimization (Blue):** This distribution represent the planned arrival time generated by the secondary optimization at the handover point respect to T_{opt} . Notably, the mean value is negative (-5.6 s), indicating a planned arrival earlier than the original T_{opt} . This occurs because the adaptive algorithm restricts the temporal search space to $t \in [t^*, T_{opt}]$. By relaxing the propellant constraint only when necessary, the solver generally identifies a feasible trajectory that is faster than the originally computed optimal time.

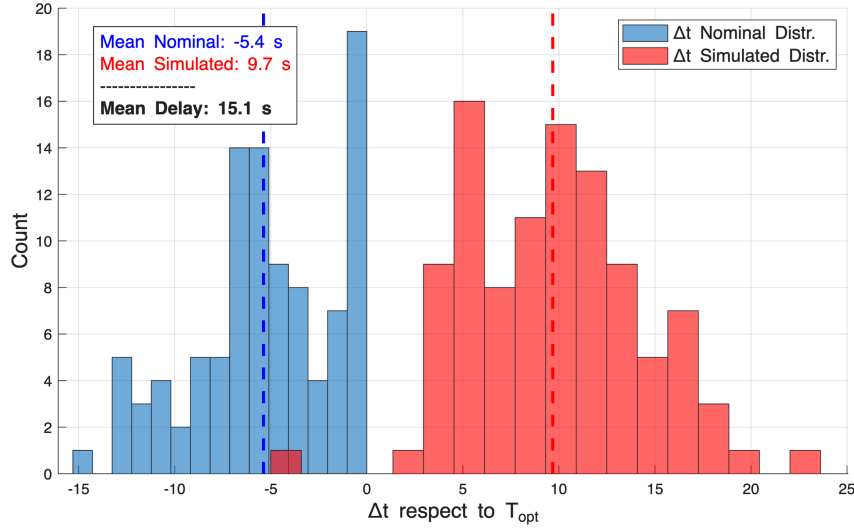


Figure 6.26: Distribution of the rendezvous time deviation relative to the initial optimal duration (T_{opt}) for Scenario B. The blue histogram represents the theoretical arrival time computed by the adaptive re-optimization, while the red histogram represents the actual physical convergence.

- **Simulated Convergence (Red):** This distribution shows the actual time required to satisfy the docking condition ($\|\rho\| < 1 [m]$). A “control lag” remains present, attributable to the same nonlinear modeling mismatches in the braking phase observed in Scenario A.

Crucially, this lag is significantly smaller than the 40.9 s observed in Scenario A. This improvement is due to the timing of the second optimization: because the trajectory is recalculated closer to the target (at the sensor handover), the error induced by the braking maneuver is substantially reduced.

Overall, the adaptive strategy proves superior: despite the lag, the average real rendezvous time in Scenario B is approximately 31 s faster than in Scenario A.

Performance Comparison

The comparison between Scenario A (Static Reference Tracking) and Scenario B (Adaptive Re-Optimization) strongly favors the second. In Scenario A, the controller fights accumulated navigation drift to return to the originally planned nominal path upon sensor handover. This aggressive correction leads to significant **transient overshoots**, **heavy propellant consumption**, and **interception delays**.

Conversely, Scenario B dynamically re-optimizes a new reference path taking the actual

measured position at t^* as a new initial condition. This adaptive approach eliminates the initial transient tracking error, drastically reduces propellant usage, and decreases the overall rendezvous time, proving to be a much more efficient and robust strategy.

To conclude the performance analysis, Figure 6.27 shows the computational times required for both PSO tuning and real-time trajectory re-optimization across the Monte Carlo runs. Because these simulations were executed on standard computing hardware rather than flight-qualified satellite processors, these metrics serve as a comparative baseline. While the absolute times do not perfectly reflect onboard performance, their order of magnitude confirms that both algorithms are feasible for real-time applications.

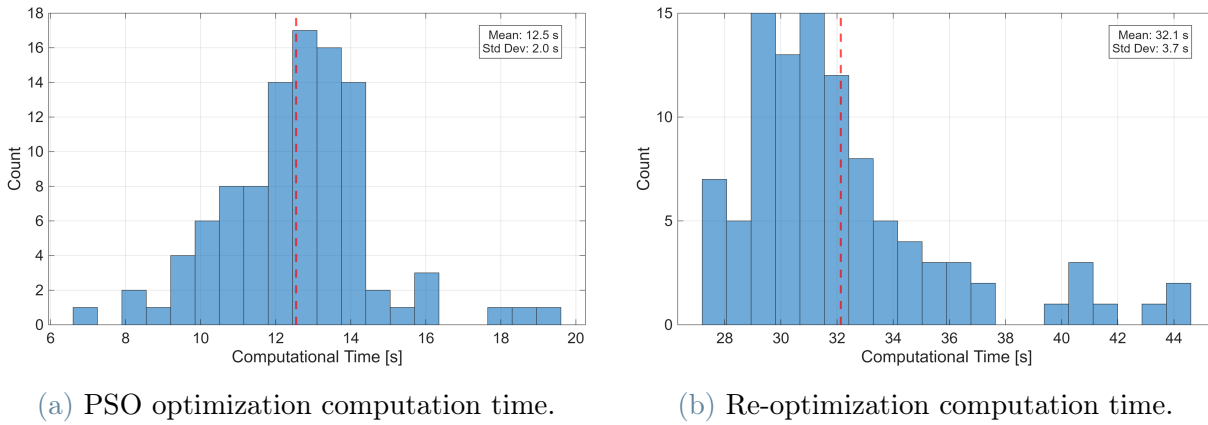


Figure 6.27: Computational time distributions across Monte Carlo runs. Note: recorded on standard desktop hardware, not flight-representative processors.

Performance Metric	Scenario A (Static)	Scenario B (Adaptive)
Operational Strategy		
Nav. Error Recovery Logic	Path-following (attempts to recover nominal trajectory)	Re-optimization (computes new optimal path)
Handover Transient Error	Handled entirely by SMC (high initial control effort)	Eliminated (initial state matches the new trajectory)
Numerical Results		
Mean Propellant Mass (m_p)	38.2 kg	23.7 kg
Mean Control Lag	+40.93 s	+9.93 s (~ 31 s faster)
Maximum Position Overshoot	Significant ($\sim 5\%$, hundreds of meters)	Negligible (smooth transition)
Mean Computational Time	12.5 s	32.1 s

Table 6.5: Performance metrics comparison between the Static (A) and Adaptive (B) guidance strategies.

6.4. Final Approach

This section evaluates the final approach phase. The underlying control structure, previously defined in Section 4.3, is briefly summarized below:

1. **Cardioid Artificial Potential Field (APF):** Activated when the chaser enters the final approach distance ρ^f (Equation (4.10)). It generates a Keep-Out Zone (KOZ) boundary combined with an attractive quadratic potential.
2. **Cone APF:** Activated during the final docking phase (Equation (4.11)), establishing a cone approach corridor with a quadratic potential to guide the chaser precisely to the docking port.

To demonstrate the effectiveness of this guidance strategy, the adaptive re-optimization approach is evaluated under nominal conditions, with the final target state of the approach specified in Table 6.6. In the Hill reference frame, this condition translates to the following state vector: $[-42.5, -42.5, 0, 0, 0.0675, 0]^T$.

Physically, this corresponds to an approach from **behind** the target (negative along-track) and **below** it (negative radial). Due to the radial offset, there is a minor residual velocity in the along-track direction. This specific final condition intentionally places the chaser inside the final approach boundary while maintaining a sufficient relative distance to **prevent collisions**.

If the final condition were set to perfectly match the target spacecraft's exact position (the origin), the chaser would arrive with too much momentum. Because the final approach controller enforces a strict, low velocity cap, it would be unable to decelerate the chaser sufficiently before it breaches the KOZ. Therefore, this offset final position is deliberately chosen to facilitate a safer, more manageable approach that naturally aligns the chaser with the proper docking axis.

ROE	Value	Unit
δa	$-6 \cdot 10^{-6}$	-
$\delta \lambda$	$-6 \cdot 10^{-6}$	rad
δe_x	0	-
δe_y	0	-
δi_x	0	rad
δi_y	0	rad

Table 6.6: Final Quasi-Nonsingular ROE.

Only the concluding segment of the trajectory is presented, corresponding to the moment the chaser enters the final approach phase (Figure 6.28).

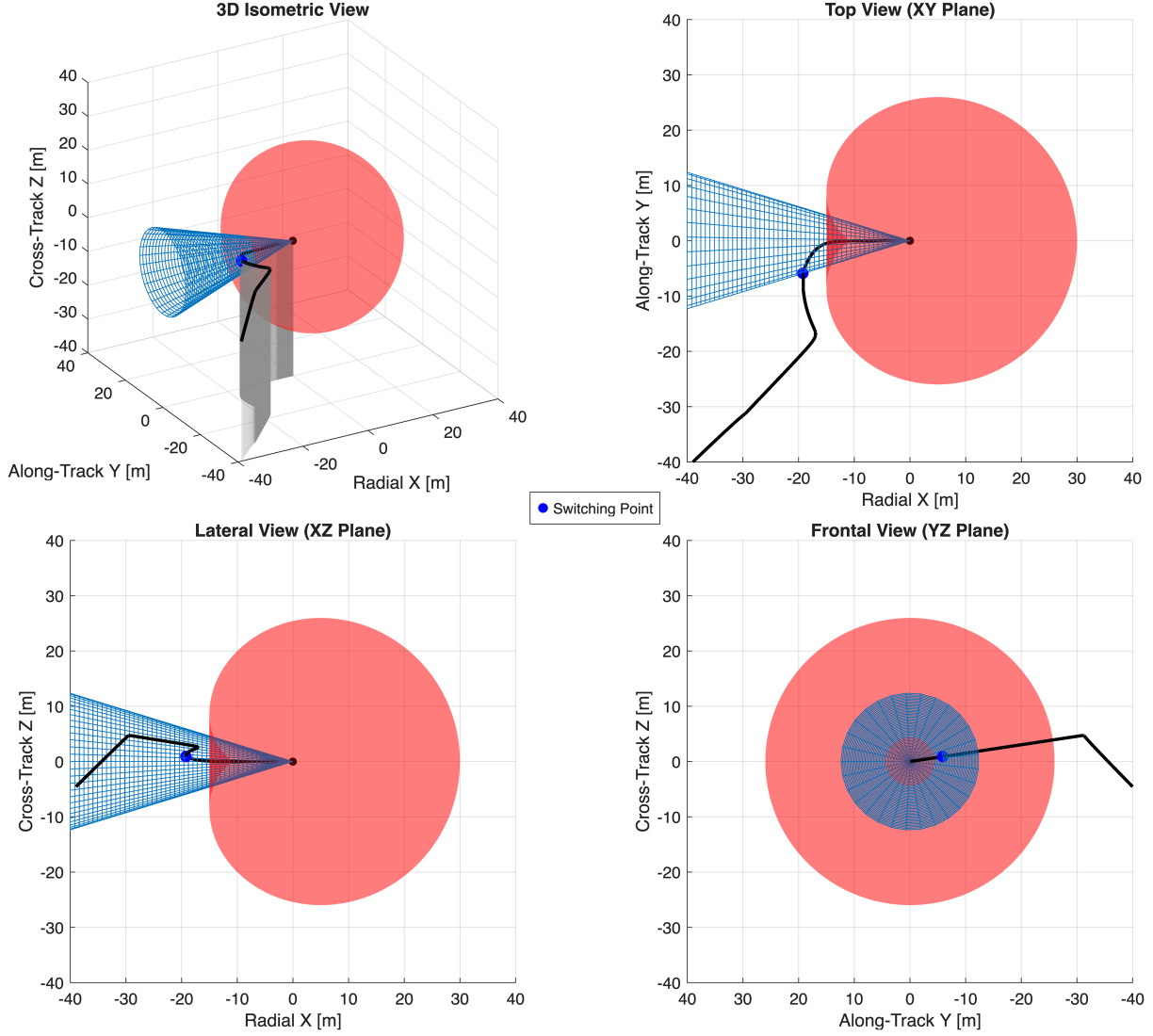


Figure 6.28: Views of the chaser's relative trajectory during the final approach, illustrating the transition between the Cardioid and Cone Artificial Potential Fields.

Views of the chaser initially approaches in a tangential direction with respect to the cardioid boundary. This behavior is governed by the quadratic attractive term of the cardioid APF (Equation (4.10)), which continuously draws the spacecraft toward the target. As the chaser nears the target, its trajectory begins to deviate toward the global minimum of the cardioid potential. The blue marker indicates the switching point where the chaser enters the cone region. At this stage, the secondary potential function assumes control, driving the spacecraft toward the final docking point.

A Monte Carlo simulation is performed to evaluate the robustness of the guidance strategy

against various approach directions. It is important to note that in a classical rendezvous, the chaser is typically positioned behind the target in the along-track direction (Table 6.2); therefore, the approach direction usually involves negative y and negative x coordinates. However, because this control structure is designed to achieve docking safely regardless of the starting position, its performance must be validated across a wider range of initial conditions.

The simulation setup consists of two distinct Monte Carlo campaigns. In the first campaign, all runs initiate within the hemisphere characterized by a negative radial position ($x < 0$), simulating a frontal approach. In the second campaign, all runs start with a positive radial position ($x > 0$), simulating an approach from behind. For both campaigns, the initial positions are uniformly distributed across the respective hemisphere at a starting distance of approximately 100 m, with a zero initial relative velocity.

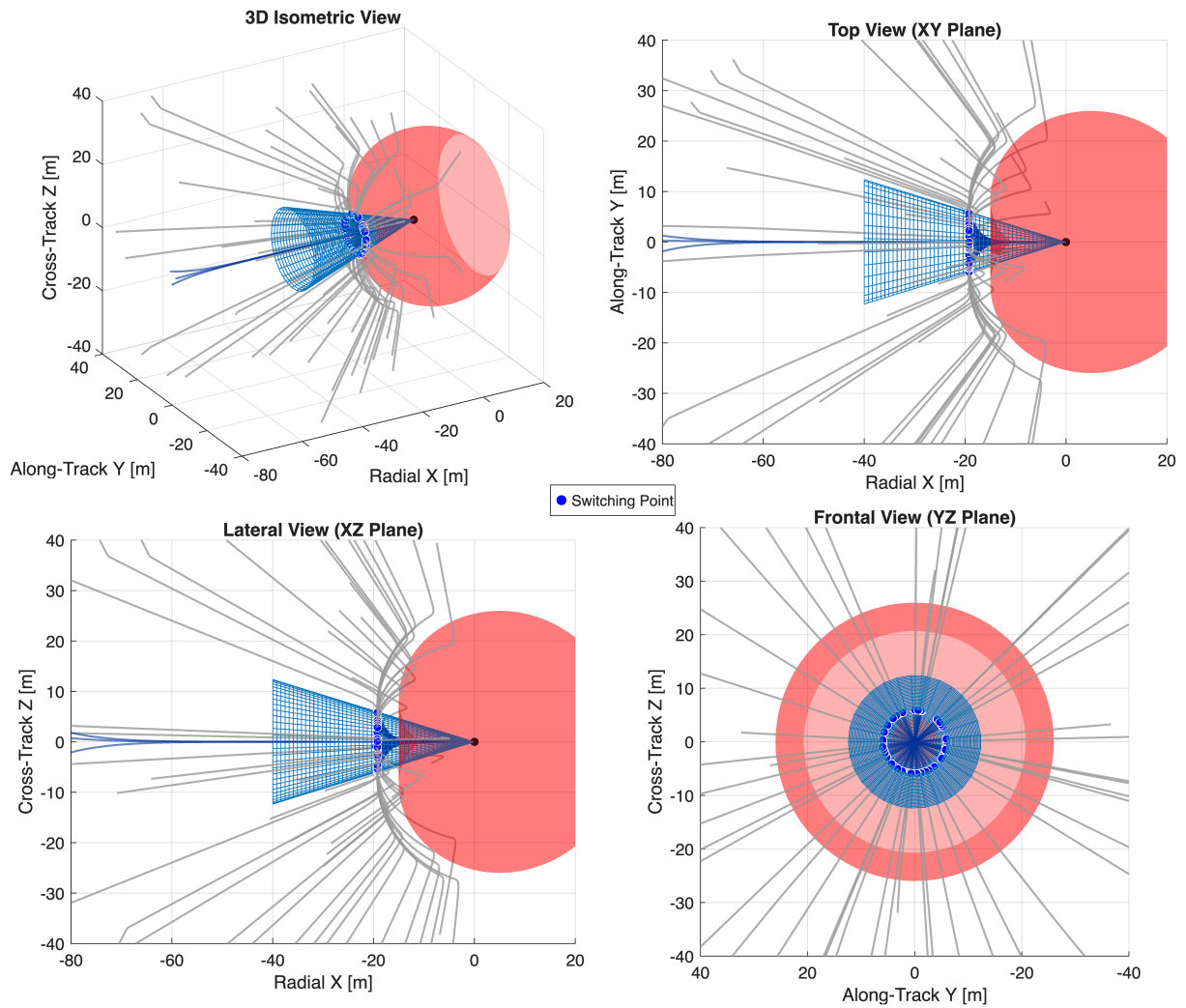


Figure 6.29: Views of the chaser's relative trajectories during the final approach Monte Carlo simulation for the frontal approach.

The simulated trajectories for the frontal approach campaign can be observed in Figure 6.29. The blue segment of the trajectory indicates the point at which the chaser enters the cone-like potential field; notably, all simulated runs successfully reach this secondary phase. Furthermore, Figure 6.30 presents the performance metrics for these maneuvers. The left plot illustrates the propellant consumption, which averages 0.71 kg. The right plot displays the time-to-dock, where a successful docking is recorded once the chaser reaches a relative distance of $\rho < 0.1$ m. Given that the controller enforces a maximum velocity of $v_{max} = 0.5$ m/s, the theoretical minimum time to cover a 100 m distance along a straight line is 200 s. Therefore, the observed mean docking time of 259 s aligns perfectly with expectations.

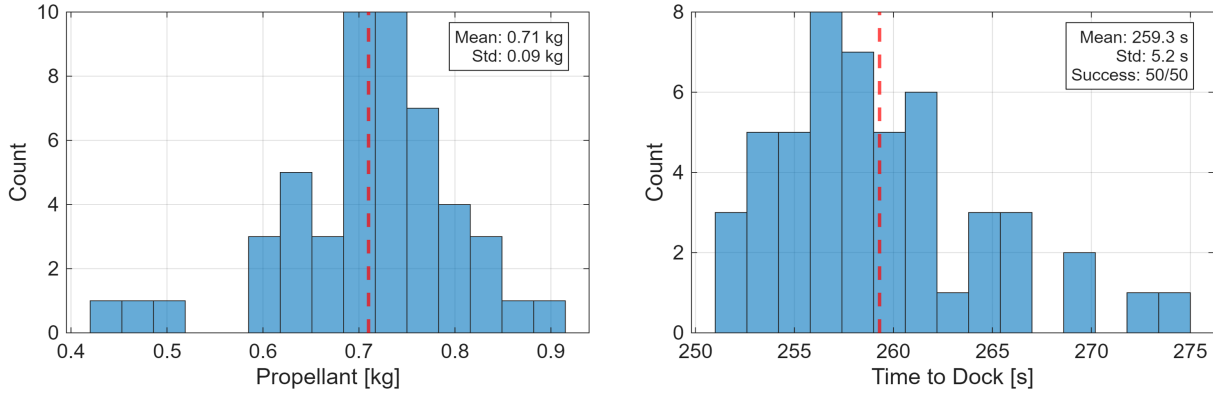


Figure 6.30: Statistical distribution of propellant consumption (left) and time-to-dock (right) for the frontal approach Monte Carlo simulation.

As shown in Figure 6.32, the mean propellant consumption increases to 1.0 kg. This higher propellant usage is expected; when starting from behind the target, the chaser must cover a greater distance to safely navigate around the Keep-Out Zone (KOZ) and align with the proper docking axis. The extended trajectory required to bypass the KOZ also explains the increased duration of the maneuver, resulting in a higher mean time-to-dock of 293 s.

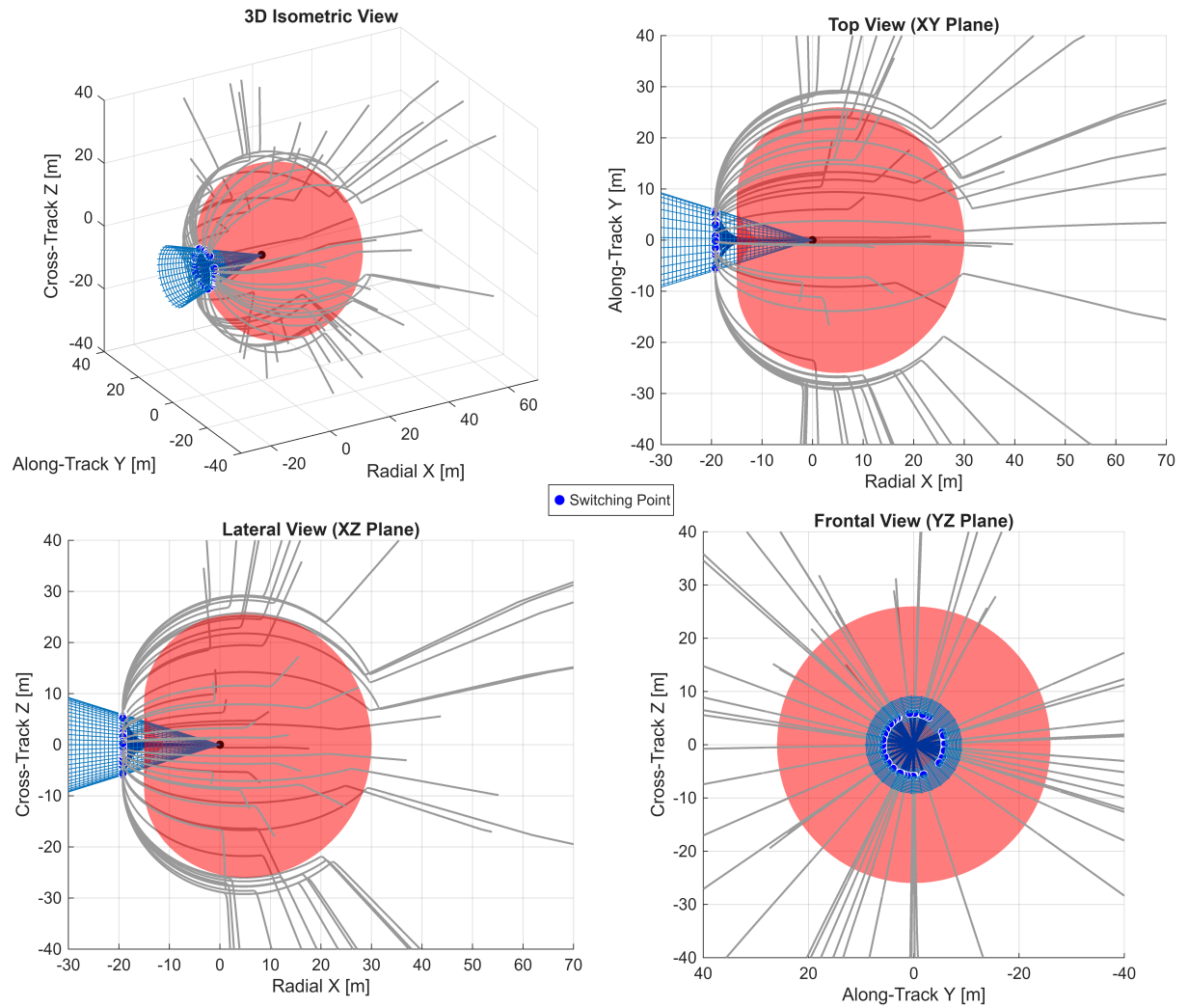


Figure 6.31: Views of the chaser's relative trajectories during the final approach Monte Carlo simulation for the behind approach.

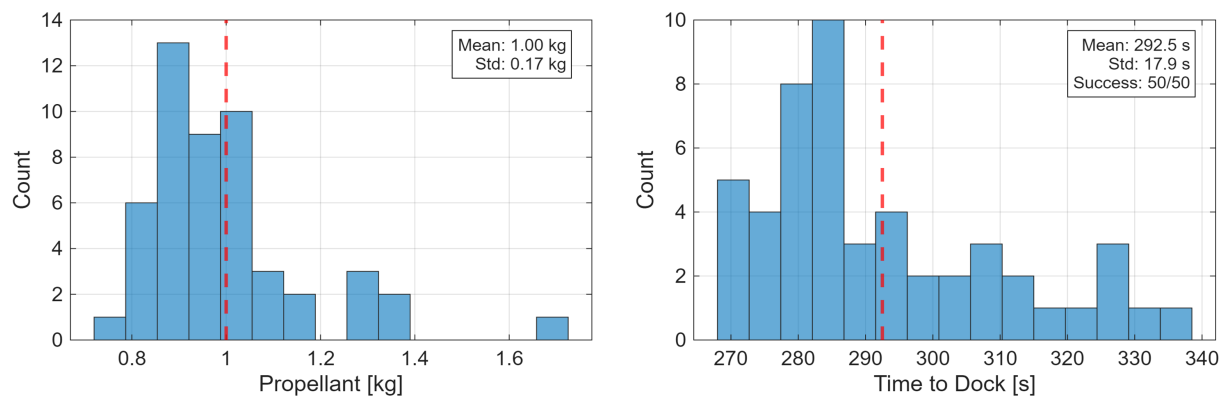


Figure 6.32: Statistical distribution of propellant consumption (left) and time-to-dock (right) for the behind approach Monte Carlo simulation.

6.5. Failures

In addition to nominal and stochastic performance evaluations, the robustness of the proposed control architecture is tested under actuator fault conditions. The primary objective is to assess whether safety and controllability can be preserved during critical phases of the rendezvous maneuver.

Thruster Fault

In this section, thruster fault scenarios are simulated to evaluate the robustness of the control system. Two primary fault cases are analyzed:

- **Stuck-open Fault:** A thruster turns on at maximum power without any control command, producing a continuous, unintended thrust.
- **Blocked-closed Fault:** A thruster fails to activate when a control command is issued, resulting in a loss of actuation in that specific direction.

It is important to note that propellant consumption is not considered for the following fault analyses. The scope of this section is strictly to verify the robustness of the control architecture. Furthermore, in certain scenarios (e.g., a stuck-open fault), the propellant consumed would exceed realistic physical limits.

This analysis is divided into two main phases:

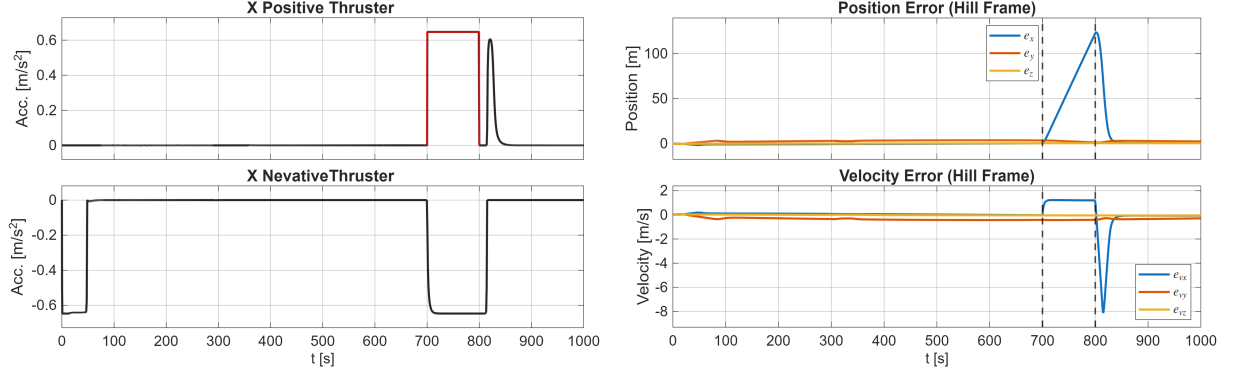
- **Far Range:** Faults occurring during the far-range phase are analyzed to demonstrate the SMC system's ability to compensate for disturbances and return the chaser to the reference path.
- **Final Approach:** Faults during the final approach are analyzed. Because the chaser is operating in extremely close proximity to the target, these scenarios highlight critical safety risks.

Far Range Fault

For the far-range fault simulations, the initial target position is defined by the same orbital element differences outlined in Table 6.2. During this phase, navigation relies solely on measurements from the INS.

A stuck-open fault is simulated on the positive X-axis thruster (positive radial direction) for a duration of 100 s, specifically from $t = 700$ s to $t = 800$ s. Under nominal conditions, no acceleration is required during this interval. This fault introduces an 100-second thrust

burst that continuously attempts to push the chaser into a higher orbit relative to the reference trajectory.

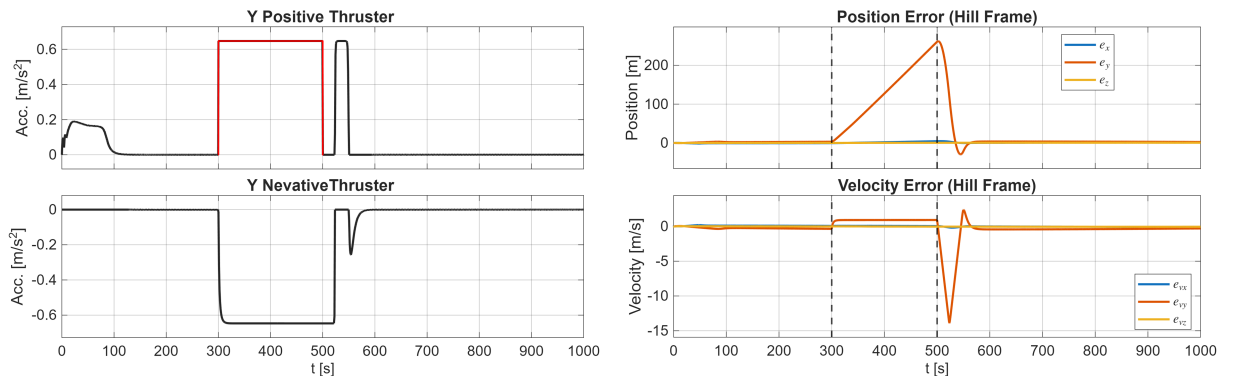


(a) Accelerations generated by the positive and negative X thrusters. The red area indicates the failure.

(b) Position error between the reference trajectory and the actual INS measurements. Vertical lines indicate the failure window.

Figure 6.33: Far-range performance under a positive X-axis stuck-open thruster fault.

As shown in Figure 6.33, the SMC immediately counteracts the faulted thruster. However, because the control law (Equation (4.6)) uses a hyperbolic tangent rather than a sign function, the counter-acceleration takes approximately 20 s to saturate. This delay causes a temporary thrust mismatch that increases radial velocity and position error (Figure 6.33b, blue line). Once the fault clears, the SMC applies positive thrust to nullify the accumulated errors, fully recovering in about 30 s and demonstrating the architecture's robustness.



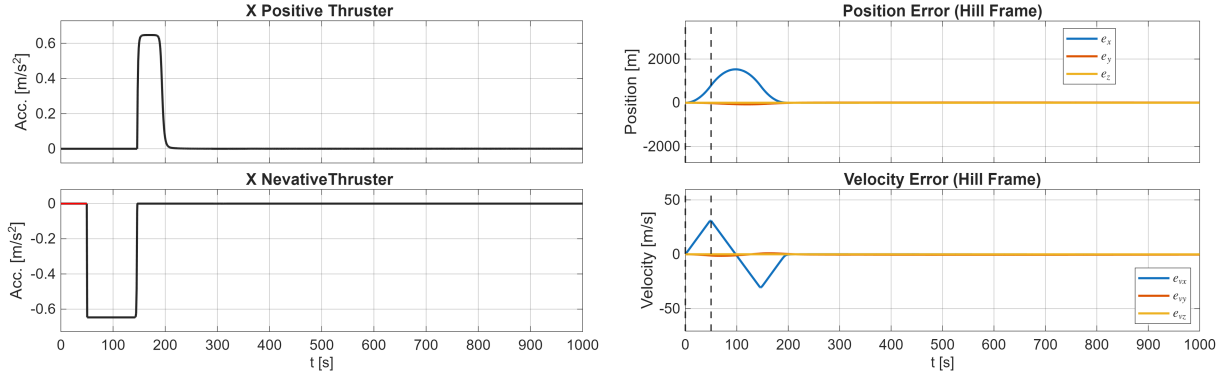
(a) Accelerations generated by the positive and negative Y thrusters. The red area indicates the duration of the failure.

(b) Position error between the reference trajectory and the actual INS measurements. Vertical lines indicate the failure window.

Figure 6.34: Far-range performance under a positive Y-axis stuck-open thruster fault.

Figure 6.34 illustrates the tracking errors generated by a stuck-open fault on the positive Y-axis, which remains active from $t = 300$ s to $t = 500$ s.

Next, a blocked-closed fault is simulated on the negative X-axis thruster from $t = 0$ s to $t = 50$ s. This specific interval coincides with the initial maneuver where the optimal control policy commands a maximum acceleration, as previously shown in Figure 6.8.



(a) Accelerations generated by the positive and negative X thrusters. The red area indicates the duration of the failure. (b) Position error between the reference trajectory and the actual INS measurements. Vertical lines indicate the failure window.

Figure 6.35: Far-range performance under a negative X-axis blocked-closed thruster fault.

During the interval $t \in [0, 50]$ s, the blockage completely prevents the optimal control sequence from firing the intended thruster. Consequently, the SMC actively compensates for this total loss of actuation to successfully maintain the spacecraft on the reference trajectory.

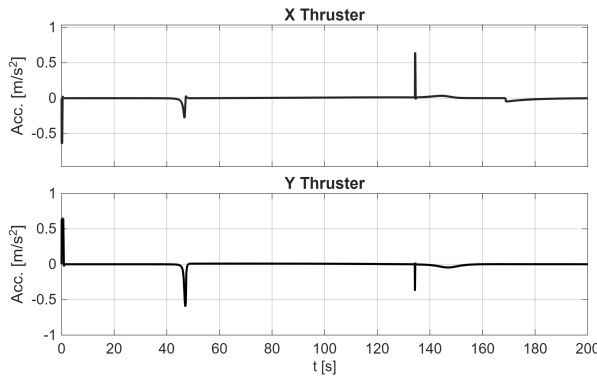
Final Approach

For the final approach phase, the initial conditions detailed in Table 6.7 are used. The inclination is set to zero to ensure a planar approach, which facilitates easier trajectory visualization.

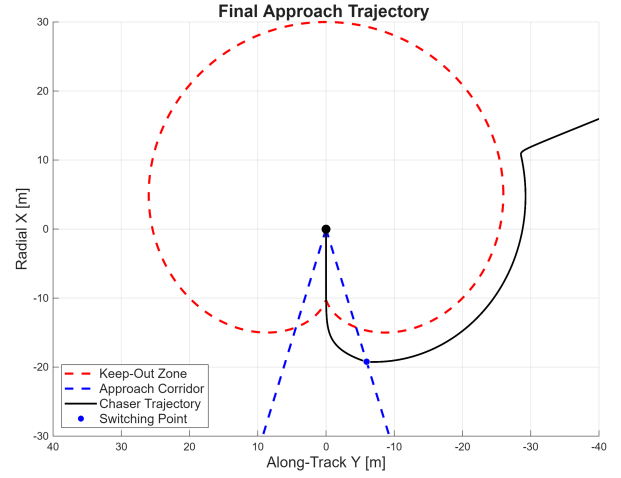
Rel. Parameter	Value	Unit
Semi-major Axis (Δa)	20	m
True Anomaly ($\Delta \theta$)	$-4.04 \cdot 10^{-4}$	deg

Table 6.7: Initial Keplerian differences for the final approach fault simulation.

The nominal trajectory for these initial conditions is shown in Figure 6.36.



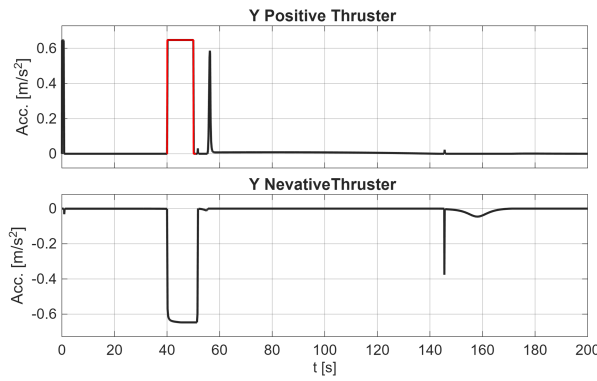
(a) Accelerations generated by the X and Y thrusters under nominal final approach conditions.



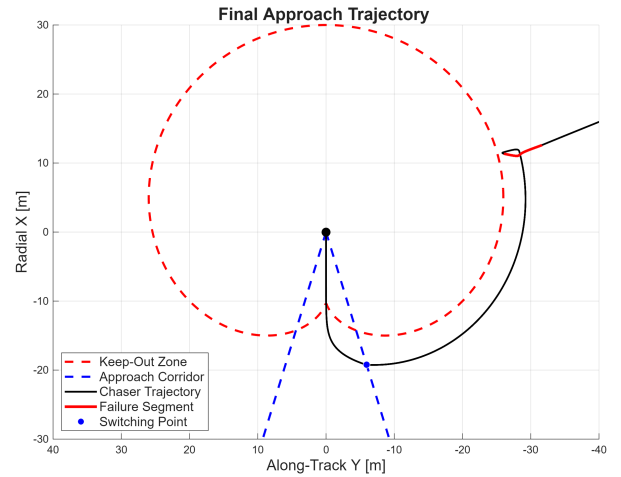
(b) Final approach trajectory in the xy-plane.

Figure 6.36: Nominal final approach trajectory utilizing the initial conditions from Table 6.7.

It is of particular interest to analyze how the final approach trajectory changes when a failure occurs during the maneuvering accelerations.



(a) Accelerations generated by the positive and negative Y thrusters. The red area indicates the failure.

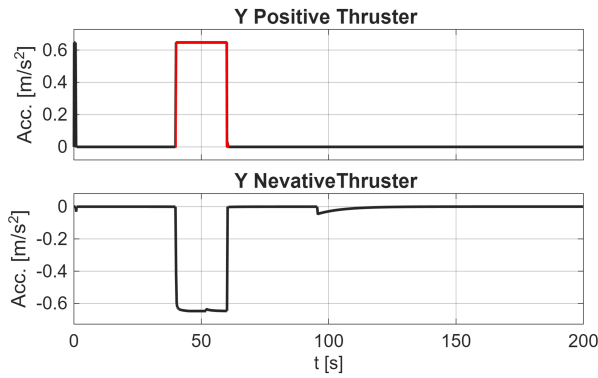


(b) Final approach trajectory in the xy-plane.

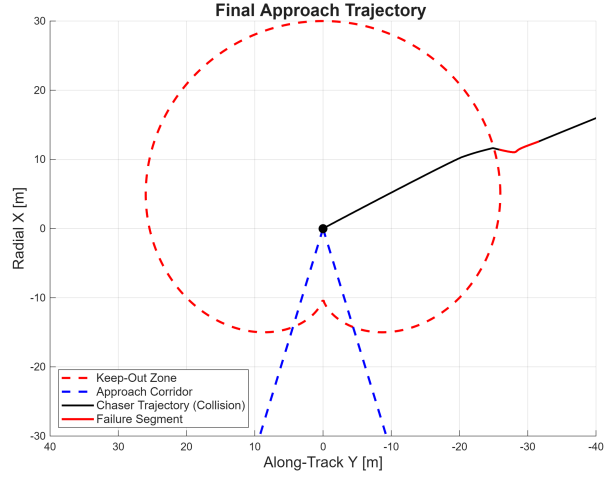
Figure 6.37: Final approach performance under a positive Y-axis stuck-open fault.

As illustrated in Figure 6.37, a stuck-open fault is simulated on the positive Y-axis thruster from $t = 40$ s to $t = 50$ s. (Note that if the fault had occurred on the negative Y thruster, the SMC would have simply counteracted it by reducing the net thrust during the nominal acceleration phase at $t = 44$ s). Because of this specific fault, the required nominal deceleration is severely compromised. This loss of braking capability causes the

chaser to drift dangerously close to the KOZ boundary. However, once the fault clears, the controller perfectly recovers the trajectory and safely completes the docking approach.



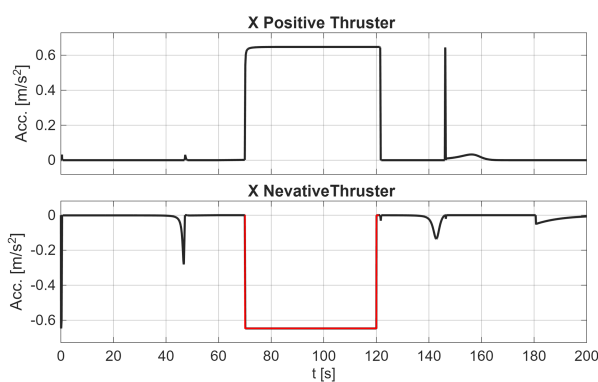
(a) Accelerations generated by the positive and negative Y thrusters. The red area indicates the failure.



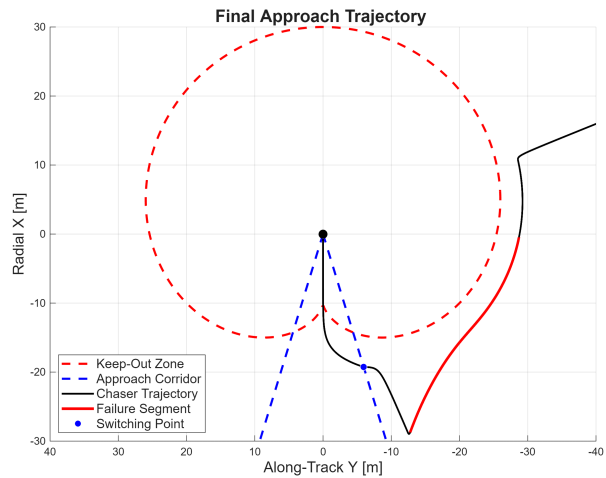
(b) Final approach trajectory in the xy-plane.

Figure 6.38: Final approach performance under a positive Y-axis stuck-open fault, resulting in a collision.

Figure 6.38 illustrates a similar positive Y-axis fault, but occurring between $t = 40$ s and $t = 60$ s. This disturbance forces the chaser to breach the KOZ. Because the cardioid potential is mathematically attractive inside this region (as previously shown in Figure 4.6), the failure results in a collision. A simple and effective solution to mitigate this vulnerability would be to implement a classical repulsive spherical potential inside the cardioid, ensuring the chaser is actively repelled if the primary KOZ boundary is ever crossed.



(a) Accelerations generated by the negative and positive X thrusters. The red area indicates the failure.



(b) Final approach trajectory in the xy-plane.

Figure 6.39: Final approach performance under a negative X-axis stuck-open fault.

Navigation Fault

During the final approach phase, accurate relative distance information is critical to avoid collisions and prevent entering the Keep-Out Zone (KOZ). Consequently, in the event of a failure in the navigation sensors responsible for relative distance measurement, the Chaser must revert to the Inertial Navigation System (INS).

However, due to the inherent bias and drift accumulation in INS solutions, the state estimate degrades rapidly. Therefore, the Chaser must immediately abort the mission. There are two primary abort strategies:

- **Parking Orbit:** The Chaser maneuvers to establish a stable holding orbit around the Target. This allows for system diagnosis and a potential second rendezvous attempt.
- **Deorbiting / Drift:** The Chaser performs a maneuver to significantly change its semi-major axis, creating a drift that maximizes the separation distance from the Target to ensure passive safety.

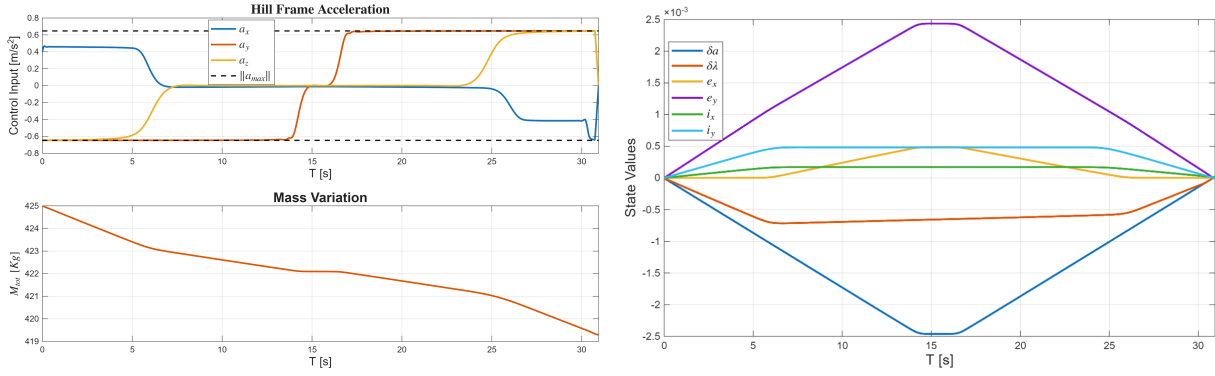
In this analysis, the **Parking Orbit** option is implemented as it presents a more complex control challenge. The primary requirement is that the maneuver must be executed as fast as possible to secure the spacecraft before the INS error grows beyond acceptable safety limits. Additionally, the generated trajectory must strictly ensure no collision occurs with the Target during the transition.

The Chaser is assumed to be in a docking position near the Target, characterized by the relative orbital parameter differences listed in Table 6.8.

Relative Keplerian Parameter	Value	Unit
Semi-major Axis Difference (Δa)	-30	m
True Anomaly Difference ($\Delta \theta$)	-0.0003	deg

Table 6.8: Initial Keplerian differences of the Chaser with respect to the Target at the onset of failure.

The abort control architecture is detailed in Section 4.1. The resulting optimal control input and Reference trajectory are presented below.

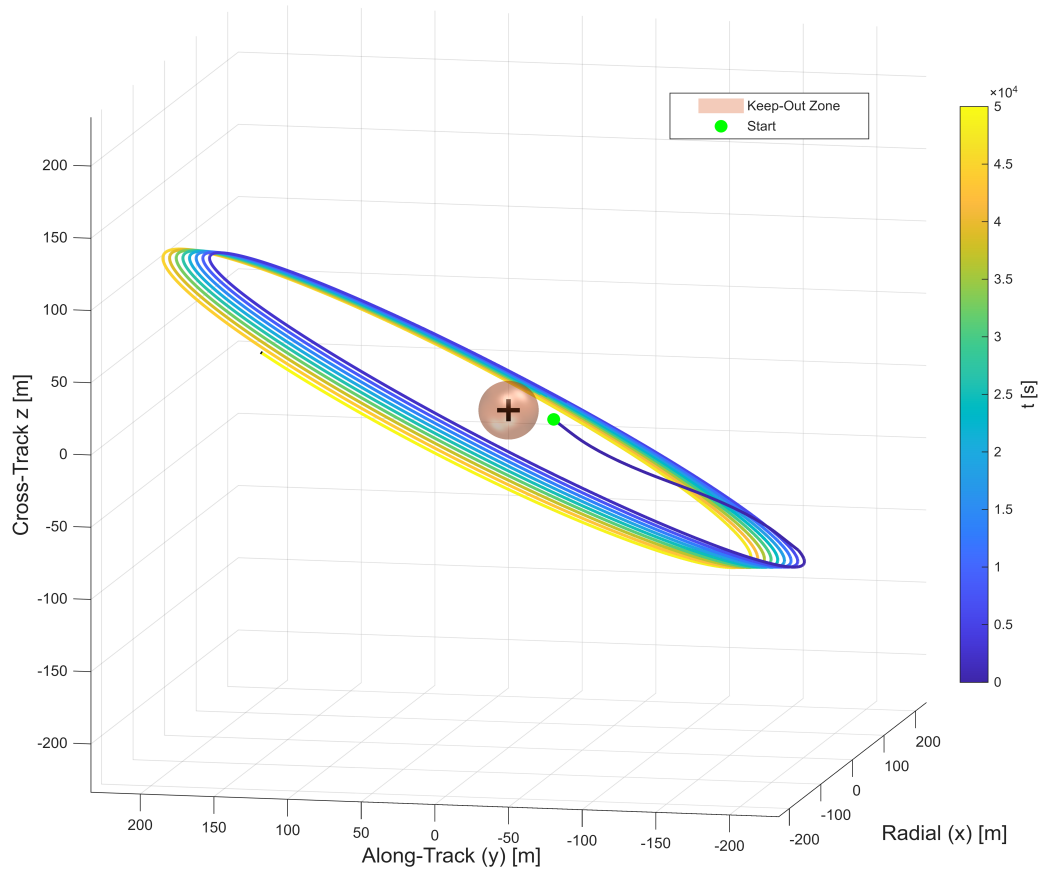


(a) Nominal control profile generated by the optimal control. (b) ROE reference generated by the optimal control.

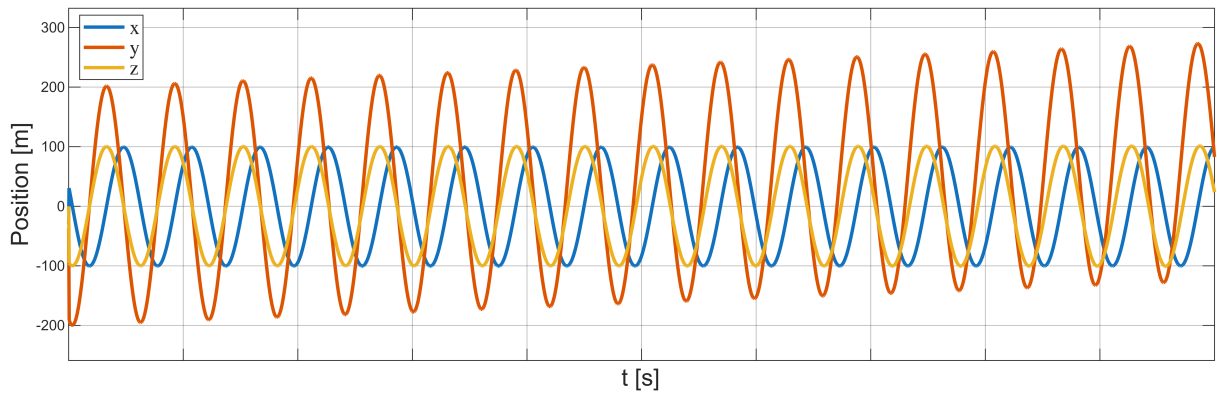
Figure 6.40: Time-Optimal Abort Maneuver results.

The optimal control input and propellant usage are shown in Figure 6.40a. Due to the short distance of the maneuver, the optimal time is found to be 32 seconds. As observed in the acceleration profile, the maneuver is not constrained by propellant consumption (which was left unconstrained for the safety abort), but rather by the actuator saturation (U_{max}).

The control profile exhibits a clear "bang-bang" behavior: the thrusters accelerate at maximum capacity to initiate the orbit change and immediately switch to maximum deceleration to stabilize into the parking orbit. This saturation confirms that the solution is time-optimal.



(a) 3D relative trajectory demonstrating the transition from the initial starting point to a safe parking orbit outside the Target's Keep-Out Zone.



(b) Time evolution of the relative position components in the Hill frame, highlighting the along-track (y) secular drift due to the J_2 perturbation.

Figure 6.41: Time-Optimal Abort Maneuver Analysis. The system rapidly maneuvers the Chaser away from the Keep-Out Zone into a passive relative orbit to ensure collision avoidance in the event of a navigation failure.

7 | Conclusion and Future Developments

7.1. Conclusion

This thesis has demonstrated the feasibility and performance of the hybrid Guidance and Control architecture specifically designed for time-critical orbital interceptions. By strategically decoupling the problem into an optimal Guidance Layer and a robust Tracking Layer, the work proved that global optimality and disturbance rejection can be achieved simultaneously without the computational costs of continuous re-optimization.

As detailed in Chapter 6, the simulations showed several key results that validate the effectiveness of this approach:

- **Optimized Tracking Performance:** The integration of Particle Swarm Optimization (PSO) for the offline tuning of the Sliding Mode Controller (SMC) provided an ideal balance between convergence speed and propellant economy. The controller demonstrated an exceptional ability to eliminate tracking errors within a few seconds. Crucially, the SMC maintained high precision even when subjected to non-linear J_2 perturbations and atmospheric drag disturbances.
- **Superiority of Adaptive Guidance:** The comparative analysis between a static trajectory and the adaptive guidance strategy at the handover point (ρ^*) revealed a significant reduction in both final approach time and propellant consumption. By re-optimizing the remaining trajectory based on actual navigation data at the start of the close range, the system successfully compensated for the accumulated drift and modeling mismatches inherent in far range approach.
- **Collision Avoidance and Safety:** The Artificial Potential Field (APF) effectively enforced the safety constraints required for proximity operations. The Cardioid Keep-Out Zone (KOZ) and the attractive approach cone ensured that the chaser remained within a safe corridor. Monte Carlo simulations confirmed that the chaser

reliably converged to the docking axis without violating safety volumes, demonstrating consistent performance regardless of whether the initial approach was initiated from the V-bar, R-bar, or arbitrary spatial orientations.

- **Robustness to Actuator and Navigation Failure:** The architecture exhibited high robustness to critical anomalies. The SMC proved capable of maintaining stability and trajectory tracking even during simulated thruster failures (stuck-closed/open faults). Furthermore, the implementation of the E/I vector separation maneuver provided a critical layer of passive safety; in the event of a total navigation failure, the chaser was shown to enter a safe drift orbit, preventing any catastrophic collision with the target asset.

In summary, the architecture proposed in this thesis constitutes a comprehensive and modular **GNC framework**. It is designed to be **general for any near-circular LEO orbit**, as the internal parameters from the ROE-based guidance to the PSO-tuned control gains are **automatically tuned** based on the specific mission profile and target orbital parameters.

7.2. Future Developments

While the proposed GNC architecture demonstrates strong performance within the defined scope, several avenues for future research and improvement remain open:

- **Proximity Operations and Capture of Uncooperative Tumbling Targets:** A critical next step is to expand the final approach phase by modeling the target not as a point mass, but as a complete 3D rigid body. In realistic non-cooperative scenarios, the target is often tumbling uncontrollably. Future developments must address the complex kinematics required to safely approach a rotating structure. This includes designing GNC algorithms that synchronize the chaser's trajectory and attitude with the target's tumbling rates, alongside integrating physical capture mechanisms.
- **Application to Kinetic Interception and ASAT Missions:** The robust, time-optimal nature of the proposed SMC and guidance architecture makes it highly suitable not only for soft rendezvous but also for kinetic interception, including Anti-Satellite (ASAT) operations and active space debris removal via direct impact. Future work could adapt the framework for Kinetic Kill Vehicles (KKVs). In such scenarios, the objective shifts from velocity-matching and soft docking to achieving a zero-miss-distance at high relative velocities, meaning the terminal phase would

bypass the repulsive APF boundaries in favor of aggressive, high-speed terminal homing guidance.

- **6-DOF Control and Attitude Coupling:** The scope of this work was restricted to translational relative motion. A critical next step is the integration of attitude dynamics to achieve full 6-Degrees-of-Freedom (6-DOF) control. Coupling translational maneuvers with spacecraft orientation is essential for sensor pointing constraints, thrust vectoring and physical docking.
- **APF Adaptation for Evasive Targets:** While the current Artificial Potential Field (APF) strategy effectively manages passive non-cooperative targets, it does not currently account for active evasion. A crucial future development involves enhancing the APF formulation to dynamically compensate for evading maneuvers by the target. This would ensure a successful interception and maintain strict safety boundaries even against a reactive or adversarial space asset.
- **Full Mission Autonomy and Vision-Based Navigation:** Integrating onboard target detection algorithms and processing pipelines for relative sensors (such as LiDAR or optical cameras) would transition the system closer to full mission autonomy. Studying the specific noise profiles and delays of these sensors within the SMC tracking loop will be crucial.
- **Hardware-in-the-Loop (HIL) Validation:** Validating the computational efficiency and robustness of the control framework on space-grade microprocessors via Hardware-in-the-Loop (HIL) simulations is necessary to confirm its viability for actual onboard implementation.

Bibliography

- [1] Todd Harrison, Kaitlyn Johnson, Joe Moye, Makena Young, et al. *Space threat assessment 2020*. Center for Strategic and International Studies (CSIS), 2022.
- [2] Clayton Swope, Kari A. Bingen, Makena Young, and Kendra LaFave. Title of the space report or article. *Journal Name or Publisher*, 2025.
- [3] WH Clohessy and RS Wiltshire. Terminal guidance system for satellite rendezvous. *Journal of the aerospace sciences*, 27(9):653–658, 1960.
- [4] J Tschauner and P Hempel. Rendezvous to a target orbiting in an elliptical path. *Astronautica Acta*, 11(2):104–+, 1965.
- [5] Kyle T Alfrend and Hui Yan. An evaluation and comparison of relative motion theories. *Advances in the Astronautical Sciences*, 116:1–17, 2004.
- [6] Kyle T Alfrend and Hanspeter Schaub. Dynamic and control of spacecraft formations: challenges and some solutions. *The Journal of the Astronautical Sciences*, 48(2):249–267, 2000.
- [7] Simone D’Amico and Oliver Montenbruck. Proximity operations of formation-flying spacecraft using an eccentricity/inclination vector separation. *Journal of Guidance, Control, and Dynamics*, 29(3):554–563, 2006.
- [8] James M Longuski, José J Guzmán, and John E Prussing. *Optimal control with aerospace applications*, volume 32. Springer, 2014.
- [9] George Boyarko, Oleg Yakimenko, and Marcello Romano. Optimal rendezvous trajectories of a controlled spacecraft and a tumbling object. *Journal of Guidance, Control, and dynamics*, 34(4):1239–1252, 2011.
- [10] Zhanhua Ma, Ou Ma, and Banavara N. Shashikanth. Optimal control for spacecraft to rendezvous with a tumbling satellite in a close range. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4109–4114, 2006.
- [11] Yuanqi Mao, Michael Szmuk, Xiangru Xu, and Behçet Açikmese. Successive con-

- vexification: A superlinearly convergent algorithm for non-convex optimal control problems. *arXiv preprint arXiv:1804.06539*, 2018.
- [12] Michael Szmuk and Behcet Acikmese. Successive convexification for 6-dof mars rocket powered landing with free-final-time. In *2018 AIAA Guidance, Navigation, and Control Conference*, page 0617, 2018.
 - [13] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
 - [14] Kyle Alfriend, Srinivas Rao Vadali, Pini Gurfil, Jonathan How, and Louis Breger. *Spacecraft formation flying: Dynamics, control and navigation*, volume 2. Elsevier, 2009.
 - [15] Avishai Weiss, Morgan Baldwin, Richard Scott Erwin, and Ilya Kolmanovsky. Model predictive control for spacecraft rendezvous and docking: Strategies for handling constraints and case studies. *IEEE Transactions on Control Systems Technology*, 23(4):1638–1647, 2015.
 - [16] Huijun Gao, Xuebo Yang, and Peng Shi. Multi-objective robust h_∞ control of spacecraft rendezvous. *IEEE Transactions on Control Systems Technology*, 17(4):794–802, 2009.
 - [17] Vadim I Utkin. *Sliding modes in control and optimization*. Springer Science & Business Media, 2013.
 - [18] Licheng Feng, Qing Ni, Yuzhu Bai, Xiaoqian Chen, and Yong Zhao. Optimal sliding mode control for spacecraft rendezvous with collision avoidance. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 2661–2668. IEEE, 2016.
 - [19] Qi Li, Jianping Yuan, and Huan Wang. Sliding mode control for autonomous spacecraft rendezvous with collision avoidance. *Acta Astronautica*, 151:743–751, 2018.
 - [20] Elisa Capello, Elisabetta Punta, Fabrizio Dabbene, Giorgio Guglieri, and Roberto Tempo. Sliding-mode control strategies for rendezvous and docking maneuvers. *Journal of Guidance, Control, and Dynamics*, 40(6):1481–1487, 2017.
 - [21] Christopher Petersen, Ryan J. Caverly, Sean Phillips, and Avishai Weiss. Safe and constrained rendezvous, proximity operations, and docking. In *2023 American Control Conference (ACC)*, pages 3645–3661, 2023.
 - [22] Howard D Curtis. *Orbital mechanics for engineering students*. Butterworth-Heinemann, 2019.

- [23] David A Vallado. *Fundamentals of astrodynamics and applications*, volume 12. Springer Science & Business Media, 2001.
- [24] Mauro Mancini, Nicoletta Bloise, Elisa Capello, and Elisabetta Punta. Sliding mode control techniques and artificial potential field for dynamic collision avoidance in rendezvous maneuvers. *IEEE Control Systems Letters*, 4(2):313–318, 2019.
- [25] Qi Li, Jianping Yuan, Bo Zhang, and Huan Wang. Artificial potential field based robust adaptive control for spacecraft rendezvous and docking under motion constraint. *ISA transactions*, 95:173–184, 2019.
- [26] James Kennedy and Rui Mendes. Population structure and particle swarm performance. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC’02 (Cat. No. 02TH8600)*, volume 2, pages 1671–1676. IEEE, 2002.
- [27] Huanqing Cui, Minglei Shu, Min Song, and Yinglong Wang. Parameter selection and performance comparison of particle swarm optimization in sensor networks localization. *Sensors*, 17(3):487, 2017.
- [28] Diogo Freitas, Luiz Guerreiro Lopes, and Fernando Morgado-Dias. Particle swarm optimisation: a historical review up to the current developments. *Entropy*, 22(3):362, 2020.
- [29] Maurice Clerc and James Kennedy. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE transactions on Evolutionary Computation*, 6(1):58–73, 2002.
- [30] Karin Zielinski and Rainer Laur. Stopping criteria for differential evolution in constrained single-objective optimization. In *Advances in differential evolution*, pages 111–138. Springer, 2008.
- [31] Noelia Sánchez-Ortiz, Raúl Domínguez-González, Nuria Guijarro-López, and Jaime Nomen-Torres. Calibration and analysis of orbital accuracy through sst observations. In *8th European Conference on Space Debris. ESA Space Debris Office*, 2021.
- [32] Arthur Richards, Tom Schouwenaars, Jonathan P How, and Eric Feron. Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming. *Journal of Guidance, Control, and Dynamics*, 25(4):755–764, 2002.
- [33] James Blake Rawlings, David Q Mayne, Moritz Diehl, et al. *Model predictive control: theory, computation, and design*, volume 2. Nob Hill Publishing Madison, WI, 2020.

- [34] IS Khalil, JC Doyle, and K Glover. *Robust and optimal control*, volume 2. Prentice hall New York, 1996.
- [35] Vincenzo Pesce, Andrea Colagrossi, and Stefano Silvestrini. *Modern spacecraft guidance, navigation, and control: from system modeling to AI and innovative applications*. Elsevier, 2022.

A | Appendix A

A.1. Transformation from Relative Orbital Elements to the Hill Frame

This appendix details the analytical transformation used to map the relative state of a chaser satellite from the quasi-circular Relative Orbital Elements (ROE) space into the Hill frame.

The relative state in the ROE framework is defined by the state vector $\delta\boldsymbol{\alpha}$:

$$\delta\boldsymbol{\alpha} = [\delta a, \delta\lambda, e_x, e_y, i_x, i_y]^T$$

where δa represents the relative semi-major axis, $\delta\lambda$ is the relative mean longitude, e_x and e_y define the relative eccentricity vector, and i_x and i_y define the relative inclination vector.

At any given instant, the transformation to the Cartesian Hill frame relative position $\delta\mathbf{r} = [x, y, z]^T$ and relative velocity $\delta\mathbf{v} = [v_x, v_y, v_z]^T$ depends on the target satellite's mean argument of latitude, $\tilde{u}$, evaluated at that same instant. The mean argument of latitude is defined as:

$$\tilde{u} = \omega + M$$

where ω is the target's argument of periapsis and M is its mean anomaly at the instant of evaluation.

Under the assumption of unperturbed Keplerian motion and a quasi-circular reference orbit, the linearized mapping from the ROE space to the relative position in the Hill frame is given by:

$$\begin{aligned} x &= a(\delta a - e_x \cos \tilde{u} - e_y \sin \tilde{u}) \\ y &= a(\delta\lambda + 2e_x \sin \tilde{u} - 2e_y \cos \tilde{u}) \\ z &= a(i_x \sin \tilde{u} - i_y \cos \tilde{u}) \end{aligned}$$

where a is the semi-major axis of the target's orbit.

The corresponding relative velocity components are obtained by evaluating the time derivatives of the position coordinates. Recognizing that the instantaneous time derivative of the mean argument of latitude is the mean motion ($\dot{u} = n$), and incorporating the secular drift in the along-track direction ($\dot{\delta\lambda} = -\frac{3}{2}n\delta a$), the instantaneous relative velocity components are:

$$\begin{aligned} v_x &= an(e_x \sin \tilde{u} - e_y \cos \tilde{u}) \\ v_y &= an \left(2e_x \cos \tilde{u} + 2e_y \sin \tilde{u} - \frac{3}{2}\delta a \right) \\ v_z &= an(i_x \cos \tilde{u} + i_y \sin \tilde{u}) \end{aligned}$$

A.2. Transformation from Keplerian Elements to Relative Orbital Elements

This appendix details the transformation from classical Keplerian orbital elements to the quasi-nonsingular Relative Orbital Elements (ROE) used throughout this thesis.

Mean Anomaly Computation

The available orbital states are expressed in terms of osculating Keplerian orbital elements

$$\mathbf{e}_{kep} = [a, e, i, \Omega, \omega, \nu]^T,$$

where ν denotes the true anomaly. Since the ROE framework is formulated using mean elements, the true anomaly is first converted into the mean anomaly M .

For each satellite, the eccentric anomaly E is obtained from the true anomaly through the standard relation

$$E = 2 \arctan \left(\sqrt{\frac{1-e}{1+e}} \tan \frac{\nu}{2} \right),$$

from which the mean anomaly is computed as

$$M = E - e \sin E.$$

This transformation removes the dependence on the true anomaly and enables the con-

sistent definition of the mean argument of latitude

$$\tilde{u} = M + \omega.$$

Relative Orbital Elements Definition

Let the subscripts c and t denote the Chaser and Target satellite, respectively. The normalized relative semi-major axis is defined as

$$\delta a = \frac{a_c - a_t}{a_t}.$$

The relative mean longitude is computed using the quasi-nonsingular definition

$$\delta \lambda = (M_c - M_t) + (\omega_c - \omega_t) + (\Omega_c - \Omega_t) \cos i_t,$$

where angular differences are wrapped to the interval $(-\pi, \pi]$ to ensure continuity.

The relative eccentricity vector components are given by

$$\begin{aligned} \delta e_x &= e_c \cos \omega_c - e_t \cos \omega_t, \\ \delta e_y &= e_c \sin \omega_c - e_t \sin \omega_t, \end{aligned}$$

while the relative inclination vector components are defined as

$$\begin{aligned} \delta i_x &= i_c - i_t, \\ \delta i_y &= (\Omega_c - \Omega_t) \sin i_t. \end{aligned}$$

Collecting all terms, the ROE state vector is expressed as

$$\delta \boldsymbol{\alpha} = \begin{bmatrix} \delta a & \delta \lambda & \delta e_x & \delta e_y & \delta i_x & \delta i_y \end{bmatrix}^T.$$

A.3. Simulation Constants

Variable	Description	Value	SI unit
m_{sat}	Chaser dry mass	405	kg
m_{fuel0}	Initial fuel mass allocated for the maneuver	20	kg
m_0	Chaser total initial mass ($m_{sat} + m_{fuel0}$)	425	kg
R_e	Earth equatorial radius	6378×10^3	m
$M_\oplus$	Earth mass	5.9722×10^{24}	kg
G	Universal gravitational constant	6.6743×10^{-11}	$\text{m}^3/(\text{kg} \cdot \text{s}^2)$
μ	Earth gravitational parameter	$G \cdot M_\oplus$	m^3/s^2
J_2	Second zonal harmonic coefficient	1.08263×10^{-3}	-
$\omega_\oplus$	Earth angular velocity vector	$[0, 0, 7.292 \times 10^{-5}]^T$	rad/s
ρ_a	Atmospheric density	4.5×10^{-14}	kg/m^3
C_D	Drag coefficient	2.2	-
A	Satellite cross-sectional area	2.25	m^2
I_{sp}	Thruster specific impulse	236	s
T_{max}	Maximum thrust magnitude	275	N
T_r	Thruster rise time	0.2	s
g_0	Standard gravitational acceleration	9.80665	m/s^2
a_0	Target initial semi-major axis (COSMOS 2506)	7089000	m
e_0	Target initial eccentricity	6.017×10^{-4}	-
i_0	Target initial inclination	98.0877	deg
Ω_0	Target initial right ascension of the ascending node	314.5895	deg
ω_0	Target initial argument of perigee	250.3851	deg
ν_0	Target initial true anomaly	0	deg
n	Target mean motion	$\sqrt{\mu/a_0^3}$	rad/s
b_{res}	INS residual bias ($10 \mu g$)	9.81×10^{-5}	m/s^2
f_c	Inertial Navigation System (INS) sampling frequency	50	Hz
ρ_{sensor}	Relative distance threshold for proximity sensor handover	10000	m
ρ_{FA}	Distance threshold to initiate the final approach phase	100	m
ρ_{bar}	Distance under which the max velocity (V_{max}) decreases	5	m
$V_{max,SMC}$	Maximum allowable velocity during final approach SMC	0.5	m/s
k_{SMC}	Smoothing parameter for the SMC $\tanh(k\sigma)$ function	30	-
C_{xy}, C_z	Sliding surface coefficients for the XY and Z planes	Tuned via PSO	-
η_{xy}, η_z	SMC robust switching gains for the XY and Z planes	Tuned via PSO	-
α_{xy}, α_z	SMC proportional gains for the XY and Z planes	Tuned via PSO	-
α_{card}	Cardioid potential field shape parameter	20	-
δ_{card}	Cardioid potential field offset parameter	-10	-
k_1, k_2	Cardioid artificial potential field tuning gains	20, 0.012	-

List of Figures

2.1	The Earth Centered Inertial (ECI) Frame	10
2.2	Hill reference frame	11
2.3	Orbital elements definition, adapted from [22]	12
3.1	Projections of bounded relative motion in the along-track/radial (y/x) (left) and cross-track/radial (z/x) (right) directions for arbitrary relative orbital elements, adapted from [7]	28
4.1	Evolution of the time horizon bounds (semilogarithmic scale on y-axis) during the bisection process. The gap $T_{\text{high}} - T_{\text{low}}$ decreases by half at each iteration, converging to the minimum feasible time.	41
4.2	Control profile generated by the optimal control solver (top) and total satellite mass evolution (bottom).	42
4.3	Relative Orbital Elements (ROE) reference trajectory generated by the optimal control.	42
4.4	Hill frame reference trajectory generated by the optimal control.	43
4.5	Comparison of SMC step responses for a 100 m initial error on the x -axis. The three vertically stacked plots illustrate the system trajectory as the control penalty scalar r_{xy} decreases (top to bottom).	51
4.6	Artificial Potential Field (APF) sliced at $z = 0$	53
4.7	Illustration of the cone potential function and its overall effect on the final docking trajectory.	55
4.8	Chaser relative position and velocity during the final approach phase.	56
4.9	Hybrid Control Architecture Diagram.	58
5.1	Von Neumann topology: (a) 2D grid structure, (b) toroidal wrapping.	63
5.2	Convergence of the global best cost J_{best} over iterations.	69
5.3	Evolution of the particle swarm during the PSO optimization process.	70
5.4	Analysis of the optimization landscape and solver performance.	71
6.1	Visualization of the Target Orbit.	74

6.2	Conceptual schematic of the rendezvous maneuver under uncertainty. . . .	76
6.3	Comparison between the linear reference trajectory (left) and the real non-linear open-loop trajectory (right). The inset on the right highlights the deviation at the target point.	77
6.4	Navigation Error: Difference between the nonlinear real trajectory and the reference ($\ \mathbf{x}_{meas} - \mathbf{x}_{ref}\ $).	78
6.5	Divergence of the INS estimation error over time.	78
6.6	Flowchart of the control architecture for Scenario A.	79
6.7	Control Scheme for Scenario A.	80
6.8	Control profile generated by the optimal control solver.	81
6.10	Evolution of the tracking error in input to the SMC (a) and corresponding control accelerations (b) for Scenario A.	81
6.9	Chaser's trajectory in (Range < 20 km), after the Handover point the SMC compensate the Navigation error.	82
6.11	Initial statistical distribution of the Chaser's state for the Monte Carlo analysis.	83
6.12	Monte Carlo trajectories (Range < 20 km).	83
6.13	Comparison of the position error relative to the reference path at initialization (t_0) and at the sensor handover epoch t^*	84
6.14	Analysis of propellant usage during Monte Carlo runs.	85
6.15	Statistical distribution of the SMC parameters optimized by PSO across the Monte Carlo runs.	86
6.16	Statistical distribution of the rendezvous time delay (Control Lag) for Scenario A. The histogram displays the difference between the actual convergence time (where $\ \boldsymbol{\rho}\ < 1 [m]$) and the nominal optimal duration (T_{opt}).	87
6.17	Flowchart of the control architecture for Scenario B.	88
6.18	Control Scheme for Scenario B.	88
6.19	Nominal control profile generated by the optimal control solver at t_0	90
6.20	Composite control profile resulting from the initial and adaptive optimizations.	91
6.22	Navigation Error ($\ \mathbf{x}_{meas} - \mathbf{x}_{ref}\ $) in Hill Frame (Scenario B).	91
6.21	Chaser's trajectory in (Range < 20 km)	92
6.23	Initial statistical distribution of the Chaser's state for Scenario B. The red lines represent the nominal state.	92
6.24	Monte Carlo trajectories (Range < 20 km), first 10 runs.	93
6.25	Analysis of propellant usage during Monte Carlo runs.	94

6.26	Distribution of the rendezvous time deviation relative to the initial optimal duration (T_{opt}) for Scenario B. The blue histogram represents the theoretical arrival time computed by the adaptive re-optimization, while the red histogram represents the actual physical convergence.	95
6.27	Computational time distributions across Monte Carlo runs. Note: recorded on standard desktop hardware, not flight-representative processors.	96
6.28	Views of the chaser's relative trajectory during the final approach, illustrating the transition between the Cardioid and Cone Artificial Potential Fields.	98
6.29	Views of the chaser's relative trajectories during the final approach Monte Carlo simulation for the frontal approach.	99
6.30	Statistical distribution of propellant consumption (left) and time-to-dock (right) for the frontal approach Monte Carlo simulation.	100
6.31	Views of the chaser's relative trajectories during the final approach Monte Carlo simulation for the behind approach.	101
6.32	Statistical distribution of propellant consumption (left) and time-to-dock (right) for the behind approach Monte Carlo simulation.	101
6.33	Far-range performance under a positive X-axis stuck-open thruster fault. . .	103
6.34	Far-range performance under a positive Y-axis stuck-open thruster fault. . .	103
6.35	Far-range performance under a negative X-axis blocked-closed thruster fault. .	104
6.36	Nominal final approach trajectory utilizing the initial conditions from Table 6.7.	105
6.37	Final approach performance under a positive Y-axis stuck-open fault. . . .	105
6.38	Final approach performance under a positive Y-axis stuck-open fault, resulting in a collision.	106
6.39	Final approach performance under a negative X-axis stuck-open fault. . . .	106
6.40	Time-Optimal Abort Maneuver results.	108
6.41	Time-Optimal Abort Maneuver Analysis. The system rapidly maneuvers the Chaser away from the Keep-Out Zone into a passive relative orbit to ensure collision avoidance in the event of a navigation failure.	109

List of Tables

2.1	Chaser Satellite Physical Parameters	18
2.2	Initial Relative Orbital Conditions	18
2.3	Target Physical and Mission Data (Estimated for Cosmos 2506)	19
2.4	Target Orbit Parameters (Initial Conditions)	20
5.1	Parameters used for the Particle Swarm Optimization.	68
6.1	Target Orbit Parameters (Initial Conditions).	74
6.2	Initial Keplerian differences.	75
6.3	Initial Quasi-Nonsingular ROE.	75
6.4	Optimized SMC parameters for Scenario A.	82
6.5	Performance metrics comparison between the Static (A) and Adaptive (B) guidance strategies.	96
6.6	Final Quasi-Nonsingular ROE.	97
6.7	Initial Keplerian differences for the final approach fault simulation.	104
6.8	Initial Keplerian differences of the Chaser with respect to the Target at the onset of failure.	107

List of Symbols

Variable	Description	SI unit
a	Semi-major axis	m
e	Eccentricity	-
i	Inclination	rad
Ω	Right Ascension of the Ascending Node (RAAN)	rad
ω	Argument of Perigee	rad
ν	True Anomaly	rad
M	Mean Anomaly	rad
$\tilde{u}$	Mean Argument of Latitude	rad
n	Mean motion	rad/s
μ	Earth's standard gravitational parameter	m ³ /s ²
J_2	Second zonal harmonic coefficient	-
$\mathbf{r}$	Chaser position vector in ECI frame	m
$\mathbf{v}$	Chaser velocity vector in ECI frame	m/s
$\mathbf{x}$	Clohessy–Wiltshire state vector	-
x, y, z	Relative position components in Hill (RTN) frame	m
$\dot{x}, \dot{y}, \dot{z}$	Relative velocity components in Hill (RTN) frame	m/s
$\boldsymbol{\rho}$	Relative position vector Hill (RTN) frame	m
ρ	Relative position norm $\ \boldsymbol{\rho}\ _2$	m
ρ^*	Distance at which distance sensor can function (Handover point)	m
$\mathbf{u}$	Control acceleration vector	m/s ²
$\mathbf{A}_{CW}$	System Matrix for Clohessy–Wiltshire equations	-
$\mathbf{B}_{CW}$	Input Matrix for Clohessy–Wiltshire equations	-
$\mathbf{A}_{ROE}$	System Matrix for ROE equations	-
$\mathbf{B}_{ROE}$	Input Matrix for ROE equations	-
$\delta\boldsymbol{\alpha}$	Relative Orbital Elements (ROE) state vector	-
δa	Relative semi-major axis	-
$\delta\lambda$	Relative mean longitude	rad
δe	Relative eccentricity vector	-
δi	Relative inclination vector	-

Variable	Description	SI unit
m	Chaser mass	kg
m_0	Chaser total mass at t_0	kg
T	Thrust magnitude	N
I_{sp}	Specific Impulse	s
g_0	Standard gravitational acceleration	m/s ²
ρ_a	Atmospheric density	kg/m ³
C_D	Drag coefficient	-
A	Cross-sectional area	m ²
$\mathcal{F}_{ECI}$	Earth-Centered Inertial reference frame	-
$\mathcal{F}_{RTN}$	Hill (Radial-Transverse-Normal) reference frame	-
J	Cost function	-
σ	Sliding surface	-
η	Sliding Mode Control switching gain	-
T_{opt}	Rendezvous time computed by the Optimal control at t_0	s
t^*	Handover time	s
KOZ	Keep Out Zone	-

