POLITECNICO MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

Receding-Horizon Guidance and Control Around an Unknown Non-Cooperative Spacecraft for Deep Learning Based 3D Reconstruction

Tesi di Laurea Magistrale in Space Engineering - Ingegneria Industriale

Author: **Davide Bellini**

Student ID: 247658
Advisor: Prof. Stefano Silvestrini
Co-advisors: Prof. Madhur Tiwari
Academic Year: 2024-25

Abstract

Autonomous close-proximity operations around non-cooperative spacecraft are increasingly required for inspection, servicing, and debris-related missions. In these scenarios, guidance and control must trade propellant usage and conservative safety margins against high-level mission objectives. When inspection relies on optical sensing, challenges become inherently correlated: the spacecraft must generate meaningful and informative viewpoints while maintaining high camera-pointing quality, thus requiring full 6-DOF motion with coupled translation and attitude. This is especially critical when the collected imagery is intended to support deep learning based 3D reconstruction, where viewpoint diversity and controlled overlap directly affect reconstruction completeness and consistency. To enable reconstruction-driven inspection, this work proposes an onboard, receding-horizon guidance and control architecture that explicitly couples relative motion regulation with mapping objectives and promotes a uniform, non-redundant viewpoints distribution. A coverage memory defined on a spherical tessellation tracks exploration progress, while a lightweight high-level planner selects the next viewing direction by maximizing expected coverage improvement and penalizing control effort. At the lower level, a constrained 6-DOF model predictive controller generates feasible translational and attitude trajectories based on Clohessy–Wiltshire–Hill relative translational dynamics and a successively linearized rotational model, while preserving favourable imaging conditions. End-to-end simulations with high-fidelity orbital propagation and robust model validation demonstrate dynamically feasible inspection trajectories, sustained progress toward uniform coverage completion, and image sets that enable deep learning reconstruction. Overall, the proposed framework enables mapping-oriented autonomy for inspection missions around non-cooperative targets under realistic operating conditions.

Keywords: Close-proximity operations, Receding-horizon guidance, Non-cooperative spacecraft, Autonomous inspection, mapping-driven objectives

Abstract in lingua italiana

Le *close-proximity operations* autonome attorno a veicoli spaziali non cooperativi sono sempre più richieste per missioni di ispezione, servizio e gestione dei detriti. In questi scenari, la guida e il controllo del satellite devono bilanciare consumo di propellente e margini di sicurezza con gli obiettivi di missione. Quando l'ispezione si basa su sensori ottici, le difficoltà diventano intrinsecamente correlate: il satellite deve essere in grado di generare punti di vista informativi mantenendo al contempo un'elevata qualità di orientamento della camera, richiedendo quindi un moto a 6 gradi di libertà in cui traslazione e assetto vengono accoppiati. Ciò è particolarmente critico quando le immagini acquisite devono supportare una ricostruzione 3D attraverso metodi di *deep learning*, per la quale la diversità delle viste influenza direttamente la completezza e la coerenza della ricostruzione. Questo lavoro propone un'architettura di guida e controllo a *receding-horizon* che unisce la regolazione del moto relativo con gli obiettivi di mappatura del target, promuovendo una distribuzione delle direzioni di vista uniforme e non ridondante. I progressi di esplorazione vengono monitorati da un indicatore di copertura definito su una tassellazione sferica attorno al target, mentre un pianificatore seleziona la direzione di osservazione successiva massimizzando l'incremento di copertura atteso e penalizzando il controllo richiesto. Un controllore MPC a 6 gradi di libertà genera le traiettorie ammissibili, includendo sia moto traslazionale, sfruttando le equazioni di Clohessy–Wiltshire–Hill, che assetto, tramite linearizzazioni successive. Attraverso delle simulazioni, basate su un'accurata propagazione orbitale e comprese di validazione del modello adottato, vengono generate traiettorie dinamicamente consistenti, che presentano un progresso con crescita regolare verso la copertura uniforme del target e la creazione di set di immagini idonei a supportare la ricostruzione 3D. Nel complesso, la struttura proposta fornisce un metodo per l'ispezione autonoma di target non cooperativi in scenari operativi realistici.

Parole chiave: Close-proximity operations, Guida con receding-horizon, Target non coperativo, Ispezione autonoma, mappatura del target

Contents

1 | Introduction

This chapter introduces the context, motivation, and objectives of the thesis. The problem of autonomous Close Proximity Operations (CPO) for imaging and deep learning based 3D reconstruction of non-cooperative targets is presented, together with the aim of this research and the main contributions of this work.

1.1. Motivation

Autonomous Close Proximity Operations are becoming increasingly relevant across a wide range of space-mission applications, spanning in-orbit servicing, end-of-life management, active debris removal and the inspection of uncooperative targets such as unresponsive spacecraft. These applications are unified by the need to operate safely in the immediate vicinity of an object that may not provide cooperative aids, while meeting stringent operational margins and retaining robustness to navigation errors and model mismatch [1]. Similar considerations also arise in the exploration of small bodies, such as asteroids, where sustained observation at a short range is required to infer shape, rotation state and characterize surface properties [2].

At close range, guidance and control are subject to tight constraints that must be enforced continuously, often at high update rates. Limited actuation authority, narrow safety envelopes and hard operational requirements demand feedback strategies that remain feasible under uncertainty. The autonomy problem is therefore not merely reaching a terminal condition, but of shaping robust motion in real time while explicitly trading propellant expenditure against mission objectives. Accordingly, guidance and control must balance safety, robustness and performance throughout the mission.

This coupling becomes particularly evident in inspection oriented missions, where proximity is primarily a means to acquire informative measurements. In an information driven inspection setting, mission success is often determined by the quality and completeness of the acquired data rather than by a terminal conditions. Optical imaging is especially attractive due to its low mass and power footprint, but it imposes tight geometric requirements that directly translate into constraints on the admissible relative geometry. These conditions must be satisfied throughout maneuvers within the actuation limits and within prescribed safety envelopes, making imaging performance an integral part of the guidance and control problem.

Imaging is a central sensing feature for CPO, as it supports both optical navigation and non-contact target identification and characterization through the analysis of appearance, relative motion, and observable geometric signatures. Beyond providing instantaneous situational awareness, an image dataset collected over time can be exploited to infer dynamical and geometrical properties of the target, for instance by estimating relative pose and angular motion and by integrating multiple observations into a coherent representation [3].

A particularly relevant outcome of this process is the possibility of generating quantitative three-dimensional models to characterize the observed body. To this end, several strategies have been investigated, ranging from model based approaches that match projected target features to image measurements, to multi-view pipelines that recover structure and motion jointly from overlapping views [4]. More recently, learning based methods have further expanded the operating envelope by improving robustness to partial observability, illumination variability and limited prior knowledge of the target geometry [5].

Within this broader landscape, recent deep learning methods, and in particular Neural Radiance Fields (NeRF)[6], provide a principled way to fuse multi-view imagery into a continuous scene representation that supports novel-view synthesis and, when needed, geometry extraction for downstream characterization [7, 8]. However, deep learning reconstruction remains sensitive to the distribution and accuracy of camera poses and data quality. It benefits from guidance strategies that shape the data acquisition trajectory to produce an image set compatible with the reconstruction pipeline, rather than treating viewpoint selection as a post-processing concern [9].

Building on this perspective, this work focuses on the autonomous generation of inspection trajectories for imaging-driven 3D reconstruction around a non-cooperative target. The core problem is to design a real-time guidance and control strategy that explicitly trades propellant usage against goal-oriented imaging performance.

1.2. Close Proximity Operations

Close Proximity Operations denote the set of guidance, navigation, and control activities that enable a spacecraft to maneuver safely in the immediate vicinity of another space object [10], while satisfying stringent safety and operational constraints. In Earth orbit, CPO are commonly framed within the broader Rendezvous and Proximity Operations (RPOD) discipline. RPOD organizes proximity operations into successive phases, each with its own verifications, contingency procedures and increasingly stringent constraint sets as the vehicles close in. Depending on the mission objective, CPO can support markedly different goals. These include close range inspection and characterization, assisted or autonomous rendezvous and docking for servicing, capture or deorbit preparation in debris removal scenarios, as well as proximity flight for interaction tasks such as landing site assessment and descent initialization in small-body missions . Across these use cases, the defining technical feature is the need to generate feasible trajectories and control actions under tight margins. Guidance decisions must simultaneously account for safety, actuation limits and operational constraints. In the remainder of this section, the focus is placed on the drivers that make CPO fundamentally constraint dominated and on the implications for autonomous, goal-oriented guidance.

The interest in in-orbit inspection and autonomous proximity capabilities is driven by multiple application domains. First, inspection is a prerequisite for in-orbit servicing and life-extension [11]. In this setting, pre-capture characterization reduces risk by refining target geometry, pose and potential hazards. Second, active debris removal and end-of-life operations require robust close range manoeuvring around non-cooperative satellites [12]. Here, the absence of docking aids and uncertain appendages amplify the need for conservative safety envelopes [13]. Demonstrations and mission concepts in this area highlight the operational value of inspection trajectories that can be executed with limited ground intervention, while maintaining verifiable safety margins [13]. A closely related motivation arises for small-body exploration. Proximity flight improves the fidelity of surface, shape and rotational state characterization beyond what is feasible from ground observations or distant flybys [14, 15]. Moreover, sustained visual coverage is explicitly addressed through dedicated planning formulations [16].

CPO inherently present significant challenges because they operate in regimes where safety margins are tight and sensing is strongly geometry dependent. Moreover, the quality of the available relative state estimate directly impacts both feasibility and risk. In Earth-orbit inspection of non-cooperative targets, these difficulties are primarily driven by the lack of cooperative navigation supports and by the fact that the target pose and geometry can be only partially observed and time varying. In addition, guidance must enforce keep-out safety regions and operational envelopes while maintaining admissible viewing conditions for perception [17, 18]. Similar challenges appear and in proximity operations about small-bodies [19, 20].

The literature commonly distinguishes orbital approaches, where the spacecraft exploits stable periodic orbits when they exist [21], and hovering approaches, where continuous control is used to maintain artificial equilibria above regions of interest [2]. In optical inspection and surface mapping, illumination constraints are frequently mission driving: observation opportunities can be restricted to sunlit time windows that depend on the body rotation and local site geometry [19]. Planning strategies therefore account explicitly for time-varying lighting feasibility, often coupling guidance with observation scheduling to preserve both safety and mapping utility [20].

Non-cooperative targets and guidance challenges

The degree of target cooperativeness strongly influences both guidance design and navigation strategy. A target is considered cooperative if it actively provides state information to the chaser or if it carries markers that facilitate relative state estimation. Exploiting this cooperativeness or accurate prior model knowledge enables classical guidance approaches to reduce uncertainty and to keep safety and operational constraints manageable.

For example, early work on formation flying and autonomous docking typically assumes a cooperative and partially stabilized target, as reflected in safe rendezvous strategies and in flight demonstrations of autonomous proximity manoeuvrers [22–24]. A related category includes cooperative but uncontrolled objects that still broadcast state related information, allowing trajectory design in close proximity despite the absence of target attitude control [25]. Other contributions remove active cooperation while retaining the ability to estimate relative states through passive elements such as docking interface markers, so that the target remains uncontrolled but can still be localized [26]. In orbit around Earth, recent experiments have further advanced autonomy by demonstrating vision-based rendezvous and close-range operations against passive, non-cooperative targets [27]. An analogous concept of relying on prior model knowledge rather than cooperation has also been explored in CPO around small bodies, where irregular geometry and rotation significantly shape feasible observation conditions [15, 28].

These approaches become inadequate when the model of the object under inspection is not known a priori or cannot be trusted due to configuration changes, structural damage, or evolving surface properties. In such regimes, guidance must explicitly account for stringent safety and observation constraints while operating under evolving uncertainty, which challenges classical trajectory design pipelines and motivates autonomy oriented inspection strategies. A notable example in this direction is the SPHERES-VERTIGO experiment [29], which demonstrated simultaneous localization and mapping of an unknown and uncooperative target in a representative environment. Although primarily focused on navigation and reconstruction rather than manoeuvre planning, this experiment directly encouraged the development of guidance algorithms that autonomously select inspection manoeuvres around previously unknown non-cooperative resident space objects.

Operationally, non-cooperativity affects planning and control in several coupled ways. Constraint design becomes strongly tied to perception requirements, as maintaining instrument pointing and limiting occlusions under pose uncertainty drives conservative standoff distances and viewpoint selection rules. Furthermore, when the target rotates, both geometry and observability become explicitly time dependent, so that static corridors or keep-out assumptions may be invalidated, motivating motion primitives and optimization formulations that handle time-varying exclusion regions and passive safety properties. Finally, guidance becomes more tightly coupled with attitude, since pointing requirements and actuator geometry directly constrain admissible translational manoeuvres.

Inspection centric operations

Inspection-centric CPO differ from rendezvous or capture driven scenarios in that the primary mission deliverable is information: the chaser spacecraft must generate trajectories that are simultaneously safe and dynamically feasible, yet also capable of acquiring viewpoints that satisfy sensing requirements for target characterization. This requirement is especially stringent in non-cooperative settings, where the target geometry and attitude motion are only partially known and where observation constraints evolve in time, as demonstrated by early in-space inspection experiments and inspection oriented frameworks [19, 29]. Under these conditions, guidance cannot be posed purely as reaching a terminal state, but must also encode task-driven objectives such as observing specific surface features, improving surface coverage and refining shape or pose estimates, together with operational envelopes and safety margins.

A representative illustration of how high-level goals translate into concrete observation driven constraints is provided by image-based mapping campaigns for small bodies. For the OSIRIS-REx mission[15], observational strategies and imaging sequences were explicitly designed to provide the inputs required by mission mapping results under tight navigation and operational constraints. These products then support decision processes related to deliverability, safety, sampleability, and scientific value assessment [30, 31]. This direct link between mission requirements, observation planning and derived products motivates autonomous formulations in which the spacecraft is assigned explicit task objectives, namely what to observe and under which quality conditions, and computes feasible trajectories accordingly instead of following a fixed reference optimized a priori.

Within this goal-oriented paradigm, several planning architectures have been proposed for small-body proximity operations. Sample-based abstract reachability techniques have been used to plan impulsive manoeuvres within a receding-horizon Model Predictive Control (MPC) loop, trading observation performance against feasibility under uncertainty [32]. The same concept has been extended to global mapping and to different mission driven metrics, including feature observation objectives [33]. More recently, repeated trajectory refinement strategies have been developed that define observation regions while satisfying physical and operational constraints, explicitly accounting for navigation and control uncertainties [20]. Collectively, these works motivate modern inspection architectures in which a task level planner expresses what to observe next, while a constraint aware optimizer enforces safety and feasibility at execution time.

Compared to the extensive small-body literature, where proximity dynamics are intrinsically uncertain and have long motivated dedicated modeling and guidance frameworks, the works that tackle inspection guidance around unknown satellites is still relatively limited, with only a few focused inspection-oriented contributions like [19].

In Earth orbit, CPO are widely treated in the literature, providing a solid dynamical background. They are most conveniently posed in a target-centered relative frame, typically a Local Vertical Local Horizontal (LVLH) or Hill frame, which provides an operational parametrization of range, closing rate, and approach geometry [34]. Guidance and constraint enforcement are then expressed in relative coordinates, while navigation and high-fidelity propagation may still be carried out in an inertial frame and mapped to LVLH through standard frame transformations.

For close-range phases in near-circular orbits, linearized relative dynamics offer multiple tractable solution strategies and enable the formulation of computationally efficient guidance laws. The Clohessy–Wiltshire–Hill (CWH) equations [35] provide a linear time-invariant approximation and are routinely adopted as the prediction layer in real-time guidance and optimization. Practical targeting and multi-maneuver design also frequently leverage on CWH equations as a baseline for constrained guidance synthesis [36, 37].

Beyond CWH, alternative representations are used when geometry, perturbations, or the operational envelope motivate a different trade-off. Relative Orbital Elements (ROEs) offer an interpretable description that is particularly effective for formation flying and maneuver planning, and have been used extensively in both guidance design and flight demonstrations [1, 38]. For non-circular reference orbits, linear time-varying relative motion formulations and State Transition Matrix (STM) solutions (e.g., Tschauner–Hempel, Yamanaka–Ankersen models) provide higher-fidelity predictors at increased modeling and implementation complexity [39].

Using linear predictors such as CWH (or ROEs based linear models) enables systematic inclusion of operational constraints, such as keep-out regions, corridor-like approach restrictions, and relative state bounds, within convex or Quadratic Programming (QP) subproblems that are compatible with onboard implementations [40]. For inspection tasks, however, model selection must remain consistent with sensing requirements: feasible trajectories must preserve admissible viewing geometry (visibility, illumination, and pointing), which motivates integrated architectures that coordinate translation with attitude pointing and imaging scheduling [18].

Overall, the literature suggests that robust non-cooperative CPO are most effectively addressed by architectures that couple constraint aware proximity guidance with perception-driven feasibility management, directly motivating the imaging-driven guidance and control strategies developed in the remainder of this work.

1.3. Model Predictive Control for Real-Time Spacecraft Guidance

Spacecraft guidance is often posed as a constrained optimal control problem. After discretization via direct transcription, it results in a finite dimensional Nonlinear Programming (NLP) problem, where the decision variables typically include predicted state and input sequences.

For onboard implementation, computational tractability is essential. This is often achieved by exploiting structural properties of the problem, such as linearized prediction models based on relative motion [40], convex or convexified constraint sets for safety and collision avoidance [41] and quadratic cost functions compatible with QP solvers [37], allowing efficient numerical strategies suitable for real-time execution. Alternatively, when nonlinear dynamics or constraints must be preserved, Nonlinear Model Predictive Control (NMPC) formulations can be adopted, supported by tailored solvers or suboptimal execution strategies.

This motivates a variety of formulations ranging from linear MPC using simplified relative dynamics [40] and convexified safety constraints [41], to nonlinear MPC approaches that explicitly incorporate 6-Degrees of Freedom (DoF) coupled dynamics or tight operational constraints.

Linear MPC formulations are particularly attractive in scenarios where the underlying dynamics can be approximated by linear models such as the CWH equations. These enable quadratic-programming formulations with linear inequality constraints. [40] proposes an Linear-Quadratic Model Predictive Control (LQ-MPC) framework that encodes thrust bounds, Line of Sight (LOS) corridor constraints, terminal velocity conditions, and obstacle avoidance as linear constraints. The same work explores the use of explicit MPC for the non-rotating target case, where the optimizer is replaced by a precomputed piecewise affine control law. For tumbling targets, it is empathize the importance of dynamically reconfiguring the constraint set as a function of predicted target motion.

Collision avoidance is a critical challenge in CPO, as keep-out zones and obstacles are inherently nonconvex. In [42] the keep-out region is often modelled as an ellipsoid and applies an adaptive convexification strategy via online linearization using geometry based hyperplanes. This approach improves feasibility and performance, especially for rotating targets. The importance of constraint handling quality for mission feasibility is also echoed in other previous studies for proximity maneuvers [43].

Inspection or capture operations often require 6-DoF modeling and control. This is address in [37] developing a time-constrained NMPC framework that allows suboptimal solution application within a fixed budget while retaining stability via suboptimal MPC theory. Their architecture combines a CWH-based translational model with a nonlinear attitude dynamic, and adopts warm-starting and real-time iteration techniques to meet computational constraints. Related efforts explore comparisons with robust/adaptive control, learning based MPC and Sequential Convex Programming (SCP) based methods for constrained 6-DoF proximity maneuvers [36, 41].

In inspection-driven missions, MPC is often embedded within goal-oriented architectures where high-level planners determine what observations are required, and MPC enforces feasibility and safety in execution. This structure is explored in small-body applications such as [32, 33], where viewpoint planning is coupled with receding-horizon MPC using reachability sets. Similar concepts are applied to urban and Low Earth Orbit (LEO) inspection contexts in [19].

In summary, the MPC literature for close-proximity operations spans efficient QP based linear formulations for constrained relative-motion guidance, NMPC schemes that handle coupled 6-DoF dynamics through solver aware implementations. Increasingly, these directions converge into hierarchical, goal-oriented architectures for inspection and servicing missions operating under uncertainty.

1.4. Deep learning and NeRF based 3D reconstruction in space

Deep learning methods, and in particular Neural Radiance Fields, have recently emerged as an attractive option for image-driven 3D reconstruction when classical multi-view pipelines struggle with sparse viewpoints, limited texture, or strong photometric variability [5]. In the context of close-proximity inspection, the interest is not limited to producing visually plausible renderings since they can provide dense, view-consistent information that supports downstream tasks such as model completion, viewpoint assessment, and pose refinement, thereby turning the image set collected during the CPO into a reusable geometric and photometric proxy of the target.

NeRF provide an implicit, continuous representation of a scene that can be optimized directly from multi-view imagery by enforcing photometric consistency through differentiable rendering [6]. In inspection scenarios, NeRF-like models are attractive because they transform a sequence of close-range images into a view-consistent representation that supports novel view synthesis and, when required, geometry extraction for downstream characterization [8]. This shifts part of the burden from hand-crafted feature extraction toward differentiable rendering based optimization, but it also exposes new sensitivities: pose accuracy, illumination variability, and the distribution of viewpoints directly shape reconstruction quality and convergence behaviour.

A first family of approaches targets 3D reconstruction under the assumption that camera geometry is known with sufficient accuracy, and focuses on mitigating the limited observability typical of space imagery. One representative direction augments NeRF training with predicted depth priors to stabilize optimization and improve geometric consistency, reducing ambiguity when views are sparse or poorly conditioned [44]. Related work addresses imaging degradations common in on-orbit inspection, combining restoration steps with efficient NeRF parameterizations to improve robustness and runtime in space-target scenarios [45]. Earlier studies also explored NeRF based reconstruction from optical imagery with wide baselines, emphasizing both the promise of implicit representations and their dependence on reliable camera modeling and sufficient viewpoint diversity [46].

For small-body mapping, NeRF type models are particularly appealing because observation geometry and illumination are inherently time varying and often mission driving. Recent work adapts neural rendering to asteroid imagery and combines implicit representations with shape regularization to recover detailed surface geometry under challenging photometric conditions [7]. Compared to satellite inspection, the small-body case places stronger emphasis on robustness to rotationally induced illumination changes, extended shadows, and heterogeneous albedo, reinforcing the need to treat viewpoint and illumination management as operational requirements rather than post-processing preferences.

A second line of work addresses a dominant bottleneck in non-cooperative proximity operations: accurate camera poses are rarely available a priori. A representative pipeline trains a NNeRF from a sparse set of spaceborne images and then uses the learned radiance field to synthesize a dataset for training a spacecraft pose estimation network, enabling pose estimation even when a reliable target CAD model is unavailable [47]. This approach treats illumination variability as a first-order issue by relying on in-the-wild NeRF variants with appearance embeddings [48]. Complementarily, bundle-adjusting NeRF formulations jointly refine camera registration and implicit reconstruction starting from imperfect pose priors, reducing pose-induced inconsistency during training [49].

Space-focused studies have also begun to tighten the coupling between reconstruction and state estimation. One line adapts NeRF variants to the 3D reconstruction of non-cooperative Resident Space Object (RSO) highlighting sensitivities that matter under flight-like conditions [50] [51]. Another frames joint attitude estimation and neural reconstruction as a space Simultaneous Localization and Mapping (SLAM) problem, aligning rendered predictions to observations to estimate both attitude evolution and 3D representation [52]. Together, these works reinforce the link to imaging-driven guidance: reconstruction quality is shaped by viewpoint and illumination sampling, naturally motivating guidance strategies that regulate relative motion and pointing to sustain admissible viewing conditions during the inspection.

1.5. Research Objectives

This thesis considers autonomous close-proximity inspection of a non-cooperative target with the goal of enabling image-based 3D reconstruction. In this setting, the chaser must operate within a restricted safety envelope and under strict actuation limits, while the target is considerate not tumbling and only coarse geometric information is available. As a result, the inspection cannot be framed solely as a terminal guidance problem, but rather as a persistent decision-making process in which safe relative motion and camera viewing conditions must be sustained over time, and the sequence of viewpoints becomes a key mission driver.

Within this context, the overarching objective is to acquire an image set that is suitable for reconstruction by enforcing the viewing conditions required by the imaging pipeline and by shaping the relative motion so that observations are distributed around the target in a sufficiently complete manner. At the same time, these requirements must be pursued without compromising feasibility, respecting safety margins and the limited control authority typically available in close-proximity operations.

The specific research objectives are:

1. *To define a unified problem in which imaging requirements and operational constraints are addressed simultaneously during close-proximity motion.*

2. *To characterize and promote inspection completeness through the viewpoint distribution, despite partial initial knowledge of the target geometry.*

3. *To pose the resulting decision-making problem in a form compatible with real-time, onboard execution.*

These objectives frame the problem addressed in this thesis and highlight the underlying trade-offs between safe, feasible proximity operations and strict viewing requirements, while ensuring that the collected observations are not only admissible but also sufficiently informative to support reliable reconstruction.

2 | Background

This chapter summarizes the mathematical framework adopted in this work, introducing the notation and reference frames, the dynamical models, and the optimization formulation employed throughout the thesis

2.1. Notation

A consistent notation is adopted to describe vectors, matrices, reference frames, and attitude representations. Bold lowercase letters denote vectors (e.g., $\mathbf{r}, \mathbf{v}, \boldsymbol{\rho}, \boldsymbol{\omega}$), while bold uppercase letters denote matrices (e.g., $\mathbf{A}, \mathbf{B}, \mathbf{J}$). The Euclidean norm is $\|\cdot\|$, the unit vector is $\hat{\mathbf{x}} = \mathbf{x}/\|\mathbf{x}\|$, and $\mathbf{I}_n$ and $\mathbf{0}_n$ denote identity and zero matrices.

Reference frames are denoted by calligraphic capital letters. The main frames considered are the Earth-Centered Inertial (ECI) frame $\mathcal{E}$, the target-centered Local Vertical Local Horizontal (LVLH) frame $\mathcal{L}$, the chaser body frame $\mathcal{B}$, and the camera frame $\mathcal{C}$. A quantity expressed in a given frame $\mathcal{A}$ is indicated by the subscript $(\cdot)_{\mathcal{A}}$ when required.

Rotations are represented by Direction Cosine Matrix (DCM) $\mathbf{C}_{\mathcal{A}/\mathcal{B}} \in SO(3)$, mapping coordinates from $\mathcal{B}$ to $\mathcal{A}$ as $\mathbf{x}_{\mathcal{A}} = \mathbf{C}_{\mathcal{A}/\mathcal{B}}\mathbf{x}_{\mathcal{B}}$, with composition $\mathbf{C}_{\mathcal{A}/\mathcal{C}} = \mathbf{C}_{\mathcal{A}/\mathcal{B}}\mathbf{C}_{\mathcal{B}/\mathcal{C}}$ and $\mathbf{C}_{\mathcal{B}/\mathcal{A}} = \mathbf{C}_{\mathcal{A}/\mathcal{B}}^T$.

The cross product is $\times$, and the associated skew-symmetric matrix is $[\cdot]_\times$, such that $[\mathbf{a}]_\times \mathbf{b} = \mathbf{a} \times \mathbf{b}$.

$$[\mathbf{a}]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}. \tag{2.1}$$

Attitude is represented primarily through Modified Rodrigues Parameters (MRPs) $\boldsymbol{\sigma}_{\mathcal{A}/\mathcal{B}} \in \mathbb{R}^3$ associated with the DCM $\mathbf{C}_{\mathcal{A}/\mathcal{B}}$ and angular velocity is denoted by $\boldsymbol{\omega}_{\mathcal{A}/\mathcal{B}}$, representing the angular velocity of frame $\mathcal{A}$ with respect to $\mathcal{B}$. When needed, the expression frame is specified via a superscript, e.g., $\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}/\mathcal{L}}$.

2.2. Reference Frames

The reference frames adopted throughout this work are summarized as follows:

- **Earth-Centered Inertial (ECI) Reference Frame** $\mathcal{E}$ is adopted as the inertial reference frame of the problem, with $\hat{\mathbf{z}}_{\mathcal{E}}$ aligned with the Earth's spin axis. The ECI frame is used to describe the absolute orbital motion of the spacecrafts and to perform high-fidelity propagation and validation of the system dynamics.

- **Local Vertical Local Horizontal (LVLH) Reference Frame** $\mathcal{L}$ is defined as a non-inertial frame associated with the orbital motion of the target spacecraft, with origin at the target center of mass. Let $\mathbf{r}_{\mathcal{E}}$ and $\mathbf{v}_{\mathcal{E}}$ denote the target inertial position and velocity in $\mathcal{E}$, the LVLH unit axes are $\hat{\mathbf{x}}_{\mathcal{L}} = \mathbf{r}_{\mathcal{E}}/\|\mathbf{r}_{\mathcal{E}}\|$, $\hat{\mathbf{z}}_{\mathcal{L}} = (\mathbf{r}_{\mathcal{E}} \times \mathbf{v}_{\mathcal{E}})/\|\mathbf{r}_{\mathcal{E}} \times \mathbf{v}_{\mathcal{E}}\|$, and $\hat{\mathbf{y}}_{\mathcal{L}} = \hat{\mathbf{z}}_{\mathcal{L}} \times \hat{\mathbf{x}}_{\mathcal{L}}$. Relative position, velocity, and control inputs of the chaser are expressed in this frame, and the translational dynamics are formulated accordingly. The rotation of $\mathcal{L}$ with respect to $\mathcal{E}$ is explicitly accounted for through the corresponding kinematic relations.

- **Body reference frame** $\mathcal{B}$ is defined as a frame rigidly attached to the chaser spacecraft, with origin at its center of mass and axes aligned with the principal inertia axes of the structure. This frame provides a natural description of the spacecraft rotational motion and is used to define attitude-dependent quantities and the orientation of onboard instruments and actuators.

- **Camera reference frame** $\mathcal{C}$ is rigidly attached to the imaging sensor mounted on the chaser spacecraft. The optical axis is aligned with $\hat{\mathbf{z}}_{\mathcal{C}}$, while $\hat{\mathbf{x}}_{\mathcal{C}}$ and $\hat{\mathbf{y}}_{\mathcal{C}}$ span the image plane. The relative orientation between $\mathcal{C}$ and $\mathcal{B}$ is assumed fixed and known and is used to evaluate pointing constraints, field-of-view conditions, and imaging-related performance metrics.

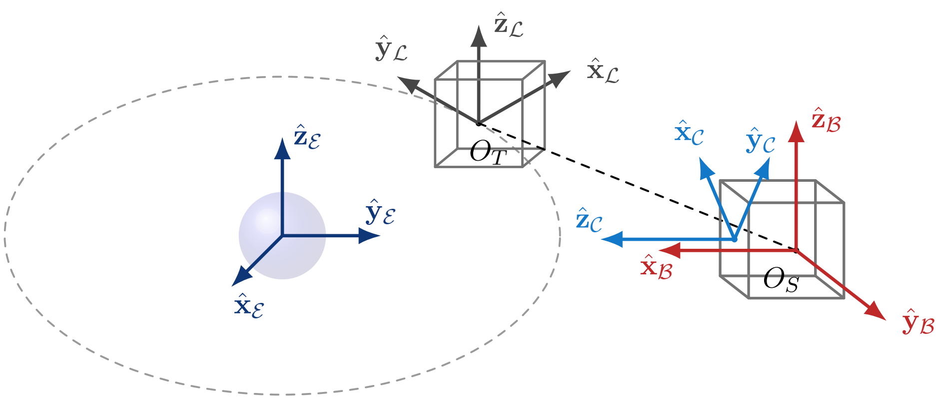


Figure 2.1: Reference frames overview.

2.3. Dynamic Models

Achieving an accurate and complete mapping of a non-cooperative target requires modeling both the relative translational motion between chaser and target and the rotational motion of the chaser. While translation governs the geometric evolution of the inspection trajectory, attitude determines the orientation of onboard sensors and, when thrust is body-fixed, the effective direction of the applied acceleration. Spacecraft attitude is therefore essential to satisfy pointing and imaging constraints, such as Field of View (FoV) and Line of Sight (LOS) requirements. Accordingly, autonomous inspection and imaging missions require a coupled treatment of translational motion and attitude.

2.3.1. Translational Dynamics

The translational dynamics are expressed in the Earth-centered inertial frame $\mathcal{E}$ ECI. Let $\mathbf{r}, \mathbf{v} \in \mathbb{R}^3$ be the absolute position and velocity expressed in $\mathcal{E}$. The equations of motion are modeled as a two-body dynamics augmented by additional perturbing accelerations,

$$\begin{cases} \dot{\mathbf{r}} = \mathbf{v} \\ \ddot{\mathbf{r}} = -\mu \dfrac{\mathbf{r}}{\|\mathbf{r}\|^3} + \mathbf{a}_{J2}(\mathbf{r}) + \mathbf{a}_{dist} \end{cases} \tag{2.2}$$

where μ is the Earth's gravitational parameter, $\mathbf{a}_{J2}$ models the acceleration due to the oblateness term J_2, and $\mathbf{a}_{dist}$ collects additional unmodeled or neglected perturbations. A compact expression for the J_2 acceleration in $\mathcal{E}$ is

$$\mathbf{a}_{J2}(\mathbf{r}) = \frac{3}{2} J_2 \frac{\mu R_\oplus^2}{r^5} \begin{bmatrix} x\left(5\frac{z^2}{r^2} - 1\right) \\ y\left(5\frac{z^2}{r^2} - 1\right) \\ z\left(5\frac{z^2}{r^2} - 3\right) \end{bmatrix}, \qquad r = \|\mathbf{r}\|, \ \mathbf{r} = [x \ y \ z]^T, \tag{2.3}$$

Close-proximity operations are commonly modeled using relative dynamics expressed in a target-centered local frame. The motion is described in the LVLH frame $\mathcal{L}$, whose origin is fixed at the center of mass of the target. Let $\mathbf{r}_T$ and $\mathbf{r}_C$ denote the inertial positions of the target and chaser, the relative position and velocity are then defined as

$$\boldsymbol{\rho} = \mathbf{r}_C - \mathbf{r}_T, \qquad \mathbf{v}_{rel} = \dot{\boldsymbol{\rho}}. \tag{2.4}$$

The relative acceleration in a rotating frame is given by

$$\ddot{\boldsymbol{\rho}} \;=\; \big(\mathbf{a}_C - \mathbf{a}_T\big) + \mathbf{u} - 2\,\boldsymbol{\omega}_{\mathcal{L}/\mathcal{E}} \times \mathbf{v}_{rel} - \dot{\boldsymbol{\omega}}_{\mathcal{L}/\mathcal{E}} \times \boldsymbol{\rho} - \boldsymbol{\omega}_{\mathcal{L}/\mathcal{E}} \times \big(\boldsymbol{\omega}_{\mathcal{L}/\mathcal{E}} \times \boldsymbol{\rho}\big). \tag{2.5}$$

where $\boldsymbol{\omega}_{\mathcal{L}/\mathcal{E}}$ is the angular velocity of the LVLH frame with respect to the inertial frame. The inertial accelerations $\mathbf{a}_T$ and $\mathbf{a}_C$ follow from (2.2), with the chaser additionally actuated by a control acceleration.

The relative motion of a chaser spacecraft with respect to a target on a circular reference orbit can be expressed through the Clohessy–Wiltshire–Hill (CWH) equations. The CWH model is obtained by performing a first-order linearization of the gravitational acceleration about the target trajectory, while accounting for the inertial effects associated with the rotating LVLH frame. This leads to a Linear Time-Invariant (LTI) representation under the standard CWH assumptions:

(i) the target follows a circular reference orbit of radius r_0;

(ii) the LVLH frame rotates at constant rate $\boldsymbol{\omega}_{\mathcal{L}/\mathcal{E}} = [0\ 0\ n]^T$ with $n = \sqrt{\mu/r_0^3}$ and $\dot{\boldsymbol{\omega}}_{\mathcal{L}/\mathcal{E}} = \mathbf{0}$;

(iii) relative distances satisfy $\|\boldsymbol{\rho}\| \ll r_0$, allowing the gravitational field to be linearized about the target position.

The resulting LTI form enables efficient discretization and fast solution of the optimization problem. Accordingly, the relative motion in $\mathcal{L}$ is described by the CWH equations

$$\begin{aligned} \ddot{x} - 2n\dot{y} - 3n^2 x &= u_x, \\ \ddot{y} + 2n\dot{x} &= u_y, \\ \ddot{z} + n^2 z &= u_z, \end{aligned} \tag{2.6}$$

where $\boldsymbol{\rho} = [x\ y\ z]^T$ is the relative position and $\mathbf{u} = [u_x\ u_y\ u_z]^T$ denotes the control acceleration expressed in LVLH. A detailed derivation of Equation (2.6) is available [35]

The translational state and the control input are defined as

$$\mathbf{x}_p \;=\; \begin{bmatrix} \boldsymbol{\rho} \\ \mathbf{v_{rel}} \end{bmatrix} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T, \qquad \mathbf{u}_p \;=\; \mathbf{a}_{\mathcal{L}} \in \mathbb{R}^3, \tag{2.7}$$

with these definitions, the Clohessy–Wiltshire–Hill equations can be written in compact form as

$$\dot{\mathbf{x}}_p \;=\; \boldsymbol{\Lambda}_p\,\mathbf{x}_p + \mathbf{G}_p\,\mathbf{u}_p, \tag{2.8}$$

with

$$\boldsymbol{\Lambda}_p = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 0 & 2n & 0 \\ 0 & 0 & 0 & -2n & 0 & 0 \\ 0 & 0 & -n^2 & 0 & 0 & 0 \end{bmatrix}, \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.9}$$

For a linear time-invariant system, its homogeneous solution can be expressed through the State Transition Matrix $\boldsymbol{\Phi}_p(t_k, t_{k+1})$ as

$$\mathbf{x}_p(t_{k+1}) = \boldsymbol{\Phi}_p(t_k, t_{k+1})\,\mathbf{x}_p(t_k) \tag{2.10}$$

Let $\Delta t \triangleq t_{k+1} - t_k$, for the CWH dynamics, the STM can be written as $\boldsymbol{\Phi}_p(t_k, t_{k+1}) = e^{\boldsymbol{\Lambda}_p\,\Delta t}$ and admits the following closed-form expression:

$$\boldsymbol{\Phi}_p(t_k, t_{k+1}) = \begin{bmatrix} 4-3c & 0 & 0 & \frac{s}{n} & \frac{2}{n}(1-c) & 0 \\ 6(s-n\Delta t) & 1 & 0 & \frac{2}{n}(c-1) & \frac{1}{n}(4s-3n\Delta t) & 0 \\ 0 & 0 & c & 0 & 0 & \frac{s}{n} \\ 3ns & 0 & 0 & c & 2s & 0 \\ 6n(c-1) & 0 & 0 & -2s & 4c-3 & 0 \\ 0 & 0 & -ns & 0 & 0 & c \end{bmatrix}, \tag{2.11}$$

where $s = \sin(n\Delta t), \;\; c = \cos(n\Delta t)$

Assuming a zero-order hold on the control acceleration over each sampling interval Δt, the CWH state-space model admits an exact discrete-time form

$$\mathbf{x}_{p,k+1} = \mathbf{A}_d\mathbf{x}_{p,k} + \mathbf{B}_d\mathbf{u}_{p,k}, \qquad \mathbf{A}_d = \boldsymbol{\Phi}_p(t_k, t_{k+1}), \qquad \mathbf{B}_d = \int_0^{\Delta t} \boldsymbol{\Phi}_p(\xi)\,\mathbf{G}_p\,d\xi, \tag{2.12}$$

with $\xi \in [0, \Delta t)$.

2.3.2. Rotational Dynamics and Attitude Kinematics

Attitude motion is modeled to support imaging-driven operations, where the feasibility and quality of each observation depend on the camera orientation. The attitude is described as the orientation of a body-fixed $\mathcal{B}$ with respect to a reference frame, and can be parametrized using Modified Rodrigues Parameters $\boldsymbol{\sigma} \in \mathbb{R}^3$ together with the angular velocity $\boldsymbol{\omega}$.

MRPs provide a minimal three-parameter attitude representation and admit direct mappings to quaternions and direction cosine matrices. Let $\mathbf{q}_{\mathcal{B}/\mathcal{L}} = [q_0\ \mathbf{q}_v^T]^T$ denote the unit quaternion associated with the rotation from $\mathcal{L}$ to $\mathcal{B}$ (i.e., mapping coordinates from $\mathcal{L}$ to $\mathcal{B}$). The corresponding MRPs are

$$\boldsymbol{\sigma} = \frac{\mathbf{q}_v}{1 + q_0}, \tag{2.13}$$

with inverse mapping

$$q_0 = \frac{1 - \boldsymbol{\sigma}^T\boldsymbol{\sigma}}{1 + \boldsymbol{\sigma}^T\boldsymbol{\sigma}}, \qquad \mathbf{q}_v = \frac{2\boldsymbol{\sigma}}{1 + \boldsymbol{\sigma}^T\boldsymbol{\sigma}}. \tag{2.14}$$

The attitude can equivalently be represented by the associated direction cosine matrix $\mathbf{C}_{\mathcal{B}/\mathcal{L}}(\boldsymbol{\sigma})$ is written in closed form as

$$\mathbf{C}_{\mathcal{B}/\mathcal{L}}(\boldsymbol{\sigma}) = \mathbf{I}_3 + \frac{8[\boldsymbol{\sigma}]_\times^2 - 4\left(1 - \boldsymbol{\sigma}^T\boldsymbol{\sigma}\right)[\boldsymbol{\sigma}]_\times}{\left(1 + \boldsymbol{\sigma}^T\boldsymbol{\sigma}\right)^2}, \tag{2.15}$$

Rigid-body rotational dynamics are governed by Euler's equation expressed in body coordinates. Let $\boldsymbol{\omega}_{\mathcal{B}/\mathcal{E}}$ denote the angular velocity of $\mathcal{B}$ with respect to the inertial frame $\mathcal{E}$ expressed in $\mathcal{B}$, and let $\mathbf{J} \in \mathbb{R}^{3\times 3}$ be the inertia matrix. Including control and disturbance torques,

$$\mathbf{J}\,\dot{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{E}} + \boldsymbol{\omega}_{\mathcal{B}/\mathcal{E}} \times \left(\mathbf{J}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{E}}\right) = \boldsymbol{\tau} + \boldsymbol{\tau}_{gg} + \boldsymbol{\tau}_{dist}, \tag{2.16}$$

where $\boldsymbol{\tau}$ is the commanded control torque, $\boldsymbol{\tau}_{gg}$ is the gravity-gradient torque, and $\boldsymbol{\tau}_{dist}$ collects additional unmodeled disturbances. A compact expression for the gravity-gradient torque is

$$\boldsymbol{\tau}_{gg} = 3\,\frac{\mu}{\|\mathbf{r}\|^3}\,\hat{\mathbf{r}}_{\mathcal{B}} \times \left(\mathbf{J}\,\hat{\mathbf{r}}_{\mathcal{B}}\right), \qquad \hat{\mathbf{r}}_{\mathcal{B}} = \frac{\mathbf{C}_{\mathcal{B}/\mathcal{E}}\mathbf{r}_{\mathcal{E}}}{\|\mathbf{r}_{\mathcal{E}}\|}. \tag{2.17}$$

In LVLH referenced formulations, the attitude kinematics are expressed in terms of the relative angular velocity $\boldsymbol{\omega}_{\mathcal{B}/\mathcal{L}}$, which drives the MRPs dynamics as

$$\dot{\boldsymbol{\sigma}}_{\mathcal{B}/\mathcal{L}} = \frac{1}{4}\,\Xi(\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{L}})\,\boldsymbol{\omega}_{\mathcal{B}/\mathcal{L}}, \tag{2.18}$$

with

$$\Xi(\boldsymbol{\sigma}) = \left(1 - \boldsymbol{\sigma}^T\boldsymbol{\sigma}\right)\mathbf{I}_3 + 2[\boldsymbol{\sigma}]_\times + 2\boldsymbol{\sigma}\boldsymbol{\sigma}^T, \tag{2.19}$$

A common representation expresses the rotational state in the body frame $\mathcal{B}$ and includes the inertial angular velocity $\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}/\mathcal{E}}$ together with attitude parameters such as $\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{L}}$. This choice retains the standard Euler equation for rigid-body dynamics, while the LVLH relative rate can be recovered from the frame kinematics. Accordingly, the rotational motion is governed by the coupled system

$$\begin{cases} \mathbf{J}\,\dot{\boldsymbol{\omega}}^{\mathcal{B}}_{\mathcal{B}/\mathcal{E}} + \boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}/\mathcal{E}} \times \left(\mathbf{J}\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}/\mathcal{E}}\right) = \boldsymbol{\tau} + \boldsymbol{\tau}_{gg} + \boldsymbol{\tau}_{dist}, \\ \dot{\boldsymbol{\sigma}}_{\mathcal{B}/\mathcal{L}} = \frac{1}{4}\,\Xi(\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{L}})\,\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}/\mathcal{L}}. \end{cases} \tag{2.20}$$

The LVLH-relative angular velocity appearing in (2.20) is obtained from the composition of angular velocities as

$$\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}/\mathcal{L}} = \boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}/\mathcal{E}} - \mathbf{C}_{\mathcal{B}/\mathcal{L}}(\boldsymbol{\sigma})\,\boldsymbol{\omega}^{\mathcal{L}}_{\mathcal{L}/\mathcal{E}}. \tag{2.21}$$

For a circular reference orbit of mean motion n, the LVLH frame rotates uniformly with respect to $\mathcal{E}$, so that

$$\boldsymbol{\omega}^{\mathcal{L}}_{\mathcal{L}/\mathcal{E}} = \begin{bmatrix} 0 \\ 0 \\ n \end{bmatrix}, \qquad \dot{\boldsymbol{\omega}}^{\mathcal{L}}_{\mathcal{L}/\mathcal{E}} = \mathbf{0}. \tag{2.22}$$

Substituting (2.22) into (2.21) provides an explicit algebraic mapping from the inertial rate state to the LVLH relative rate used in the attitude kinematics.

Defining the rotational state and input as

$$\mathbf{x}_r = \begin{bmatrix} \boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}/\mathcal{E}} \\ \boldsymbol{\sigma}_{\mathcal{B}/\mathcal{L}} \end{bmatrix} \in \mathbb{R}^6, \qquad \mathbf{u}_r = \boldsymbol{\tau} \in \mathbb{R}^3, \tag{2.23}$$

the system (2.20) can be written in compact nonlinear form as

$$\dot{\mathbf{x}}_r = \mathbf{f}(\mathbf{x}_r, t) + \mathbf{G}_r\,\mathbf{u}_r + \mathbf{w}_r(t), \qquad \mathbf{G}_r = \begin{bmatrix} \mathbf{J}^{-1} \\ \mathbf{0}_{3\times 3} \end{bmatrix}, \tag{2.24}$$

where $\mathbf{f}$ collects the nonlinear rigid-body terms together with the LVLH coupling introduced by (2.21), while $\mathbf{w}_r$ represents residual disturbances.

Given the nonlinear nature of (2.24), a Linear Time-Varying (LTV) approximation can be obtained by first order linearization along a reference trajectory. Let $\bar{\mathbf{x}}_r(t)$ denote a reference rotational trajectory. A first order expansion about $\bar{\mathbf{x}}_r(t)$ yields the LTV model

$$\dot{\mathbf{x}}_r(t) = \mathbf{\Lambda}_r(t)\,\mathbf{x}_r(t) + \mathbf{G}_r\,\mathbf{u}_r(t) + \mathbf{c}_r(t), \tag{2.25}$$

where

$$\mathbf{\Lambda}_r(t) = \mathbf{\nabla f}(\bar{\mathbf{x}}_r) = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}_r}\right|_{\bar{\mathbf{x}}_r(t)}, \qquad \mathbf{c}_r(t) = \mathbf{f}(\bar{\mathbf{x}}_r(t), t) - \mathbf{\Lambda}_r(t)\,\bar{\mathbf{x}}_r(t). \tag{2.26}$$

Assuming constant inputs over each interval $[t_k, t_{k+1})$ and freezing $\mathbf{\Lambda}_r, \mathbf{c}_r$ at t_k, the discrete-time dynamics become

$$\mathbf{x}_{r,k+1} = \mathbf{A}_{r,k}\,\mathbf{x}_{r,k} + \mathbf{B}_{r,k}\,\mathbf{u}_{r,k} + \mathbf{D}_{r,k}\,\mathbf{c}_{r,k}, \tag{2.27}$$

with

$$\mathbf{A}_{r,k} = \mathbf{\Phi}_r(t_k, t_{k+1}) = e^{\mathbf{\Lambda}_{r,k}\Delta t_k}, \tag{2.28}$$

$$\mathbf{B}_{r,k} = \int_{t_k}^{t_{k+1}} \mathbf{\Phi}_r(\xi, t_{k+1})\;\mathbf{G}_r\,d\xi = \int_0^{\Delta t_k} e^{\mathbf{\Lambda}_{r,k}(\Delta t_k - \xi)}\,\mathbf{G}_r\,d\xi, \tag{2.29}$$

$$\mathbf{D}_{r,k} = \int_{t_k}^{t_{k+1}} \mathbf{\Phi}_r(\xi, t_{k+1})\;d\xi = \int_0^{\Delta t_k} e^{\mathbf{\Lambda}_{r,k}(\Delta t_k - \xi)}\,d\xi, \tag{2.30}$$

where $\Delta t_k = t_{k+1} - t_k$ and $\xi \in [t_k, t_{k+1})$. Over a given reference trajectory, the matrices $\{\mathbf{A}_{r,k}, \mathbf{B}_{r,k}, \mathbf{D}_{r,k}, \mathbf{c}_{r,k}\}$ are evaluated and treated as constant over each interval.
Appendix A reports the explicit entries of $\mathbf{\Lambda}_r(t)$ used to compute $\mathbf{A}_{r,k}$, $\mathbf{B}_{r,k}$, and $\mathbf{D}_{r,k}$.

2.3.3. Camera Geometry and Field of View

The camera is modeled using a pinhole geometry, which provides a compact and widely adopted representation of the imaging process through a finite Field of View defined by the camera intrinsic parameters. Let (f_x, f_y) denote the focal lengths in pixel units and (W, H) the image width and height. The horizontal and vertical fields of view are

$$\mathrm{FOV}_h = 2\arctan\left(\frac{W}{2f_x}\right), \qquad \mathrm{FOV}_v = 2\arctan\left(\frac{H}{2f_y}\right). \tag{2.31}$$

These angles define a camera viewing cone in the camera frame, which characterizes the set of directions that can be imaged at each time instant.

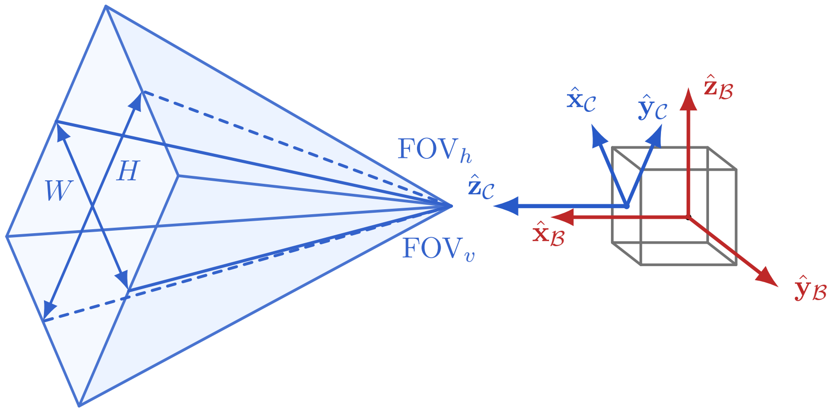


Figure 2.2: Pinhole camera geometry and field-of-view definition.

2.4. The Optimal Control Problem

Optimal Control Problem (OCP) provides a convenient mathematical framework to determine a control history that minimizes a prescribed performance index while satisfying the system dynamics and a set of operational constraints.

Let $t \in [t_0, t_f]$ denote a continuous time. The system is described by a state trajectory $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ and a control history $\mathbf{u}(t) \in \mathbb{R}^{n_u}$. The goal is to determine $\mathbf{x}(t), \mathbf{u}(t)$ to minimize an objective functional J, while satisfying the dynamics and a set of operational constraints. This can be stated as

$$\min_{\mathbf{u}\in\mathcal{U}} \; J\big(\mathbf{x}(t), \mathbf{u}(t)\big), \tag{2.32}$$

subject to the following constraints:

- *Dynamics.* The system evolution is governed by a set of differential equations

$$\dot{\mathbf{x}}(t) \;=\; \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t), t\big), \qquad \mathbf{x}(t_0) = \mathbf{x}_0, \tag{2.33}$$

  where $\mathbf{f} = [f_1, f_2, ..., f_n]^T \in \mathbb{R}^{n_x}$ may also depend on constant parameters assumed known over $[t_0, t_f]$.

- *Path constraints.* A set of limitations to be enforced along the optimal solution and modeled as inequality constraints

$$\mathbf{g}\big(\mathbf{x}(t), \mathbf{u}(t), t\big) \leq \mathbf{0}, \qquad t \in [t_0, t_f], \tag{2.34}$$

and as equality constraints

$$\mathbf{h}\big(\mathbf{x}(t), \mathbf{u}(t), t\big) = \mathbf{0}, \qquad t \in [t_0, t_f]. \tag{2.35}$$

- *Boundary conditions.* Initial and terminal constraints at t_0 and t_f

$$\boldsymbol{\psi}_{\min} \;\leq\; \boldsymbol{\psi}\big(\mathbf{x}(t_0), \mathbf{u}(t_0), t_0;\; \mathbf{x}(t_f), \mathbf{u}(t_f), t_f\big) \;\leq\; \boldsymbol{\psi}_{\max}. \tag{2.36}$$

  The equality case $\boldsymbol{\psi}(\cdot) = \mathbf{0}$ is recovered by setting $\boldsymbol{\psi}_{\min} = \boldsymbol{\psi}_{\max} = \mathbf{0}$.

- *Simple bounds.* Enforced on state and control through

$$\underline{\mathbf{x}}(t) \leq \mathbf{x}(t) \leq \overline{\mathbf{x}}(t), \qquad \underline{\mathbf{u}}(t) \leq \mathbf{u}(t) \leq \overline{\mathbf{u}}(t), \qquad t \in [t_0, t_f], \tag{2.37}$$

  which is equivalent to requiring $\mathbf{x}(t) \in \mathcal{X}(t)$ and $\mathbf{u}(t) \in \mathcal{U}(t)$ for suitable admissible sets.

The objective functional J is expressed in Bolza form,

$$J\big(\mathbf{x}(t), \mathbf{u}(t)\big) \;\triangleq\; \phi\big(\mathbf{x}(t_f), t_f\big) \;+\; \int_{t_0}^{t_f} L\big(\mathbf{x}(t), \mathbf{u}(t), t\big)\, dt, \tag{2.38}$$

where L is a function of state and control and time and ϕ denotes a terminal contribution.

To characterize optimal solutions of the continuous-time problem, first order necessary conditions from the calculus of variations are recalled. Lagrange multipliers are introduced for the dynamic constraint and for the boundary conditions. Let $\boldsymbol{\nu} \in \mathbb{R}^q$ denote the constant vector of multipliers associated with the boundary constraints, and let $\boldsymbol{\lambda}(t) \in \mathbb{R}^{n_x}$ denote the time-varying costate vector associated with the dynamics. The augmented objective functional is defined as

$$\begin{aligned} \bar{J} \;&= \phi\big(\mathbf{x}(t_f), t_f\big) + \boldsymbol{\nu}^\top \boldsymbol{\psi}\big(\mathbf{x}(t_f), \mathbf{u}(t_f), t_f\big) \\ &\quad + \int_{t_0}^{t_f} \Big[L\big(\mathbf{x}(t), \mathbf{u}(t), t\big) + \boldsymbol{\lambda}(t)^\top \Big(\mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t), t\big) - \dot{\mathbf{x}}(t)\Big)\Big]\, dt. \end{aligned} \tag{2.39}$$

which embeds the dynamics and the final boundary conditions into the objective functional. The necessary conditions for a stationary point of $\bar{J}$ are obtained by imposing that its first variation vanishes: $\delta\bar{J} = 0$. In order to write these conditions in compact form, it is convenient to define the Hamiltonian

$$H\big(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{u}, t\big) \triangleq L(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\lambda}^\top \mathbf{f}(\mathbf{x}, \mathbf{u}, t). \tag{2.40}$$

Accordingly, the necessary conditions can be expressed as

$$\dot{\mathbf{x}} = H_{\boldsymbol{\lambda}}, \qquad \dot{\boldsymbol{\lambda}} = -H_{\mathbf{x}}, \qquad \mathbf{0} = H_{\mathbf{u}}, \tag{2.41}$$

where subscripts denote partial derivatives. These relations are also referred to as the Euler–Lagrange equations. The first equation is equivalent to the system dynamics, the second describes the evolution of the costates, and the third provides an algebraic optimality condition for the control.

The Equation (2.41) must be solved together with the endpoint constraints and the associated transversality conditions, which depend on the boundary specifications.
A comprehensive treatment of the OCP formulation and of first order necessary conditions is available in [53, 54].

2.5. Nonlinear programming

The Optimal Control Problem defined in (2.32)–(2.37) is infinite-dimensional, as the decision variables are functions of time. In order to obtain a computable formulation, a standard approach consists in discretizing the time horizon and translating the continuous-time dynamics and constraints into a finite set of algebraic relations. After discretization, the problem can be recast as a finite dimensional Nonlinear Programming (NLP) problem.

In its general form, an NLP consists in determining a finite dimensional decision vector $\mathbf{z} \in \mathbb{R}^{n_z}$ that solves

$$\begin{aligned} \min_{\mathbf{z}\in\mathbb{R}^{n_z}} \quad & J_d(\mathbf{z}) \\ \text{s.t.} \quad & \mathbf{c}(\mathbf{z}) = \mathbf{0}, \\ & \mathbf{d}(\mathbf{z}) \leq \mathbf{0}, \\ & \mathbf{z}_{\min} \leq \mathbf{z} \leq \mathbf{z}_{\max}, \end{aligned} \tag{2.42}$$

where $J_d : \mathbb{R}^{n_z} \to \mathbb{R}$ denotes the discretized objective function, $\mathbf{c} : \mathbb{R}^{n_z} \to \mathbb{R}^{n_c}$ collects equality constraints, and $\mathbf{d} : \mathbb{R}^{n_z} \to \mathbb{R}^{n_d}$ collects inequality constraints. The bounds $\mathbf{z}_{\min}$ and $\mathbf{z}_{\max}$ represent admissible ranges and are typically treated separately due to their special numerical structure.

Through this construction, the differential equations (2.33) are replaced by algebraic constraints enforcing the discrete-time system evolution, while the continuous-time path and boundary constraints are translated into corresponding algebraic equalities and inequalities.

The first order optimality conditions for the NLP in Equation (2.42) are given by the Karush–Kuhn–Tucker (KKT) conditions. Let $\boldsymbol{\lambda} \in \mathbb{R}^{n_c}$ and $\boldsymbol{\mu} \in \mathbb{R}^{n_d}$ denote the Lagrange multipliers associated with equality and inequality constraints, respectively. The Lagrangian function is defined as

$$\mathcal{L}(\mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \triangleq J_d(\mathbf{z}) + \boldsymbol{\lambda}^\top \mathbf{c}(\mathbf{z}) + \boldsymbol{\mu}^\top \mathbf{d}(\mathbf{z}), \tag{2.43}$$

A point $\mathbf{z}^\star$ is said to satisfy the KKT conditions if there exist multipliers $(\boldsymbol{\lambda}^\star, \boldsymbol{\mu}^\star)$ such that

$$\nabla_{\mathbf{z}} \mathcal{L}(\mathbf{z}^\star, \boldsymbol{\lambda}^\star, \boldsymbol{\mu}^\star) = \mathbf{0}, \tag{2.44}$$

$$\mathbf{c}(\mathbf{z}^\star) = \mathbf{0}, \tag{2.45}$$

$$\mathbf{d}(\mathbf{z}^\star) \leq \mathbf{0}, \tag{2.46}$$

$$\boldsymbol{\mu}^\star \geq \mathbf{0}, \tag{2.47}$$

$$\boldsymbol{\mu}^{\star\top} \mathbf{d}(\mathbf{z}^\star) = 0. \tag{2.48}$$

Conditions (2.44)–(2.48) express, respectively, stationarity of the Lagrangian, primal feasibility, dual feasibility, and complementarity. Under suitable regularity assumptions, any local minimum of (2.42) must satisfy the KKT conditions. A complete and detailed formulation is presented in [53, 55].

2.5.1. Direct transcription

Direct transcription converts the continuous-time OCP into a finite-dimensional NLP by introducing a temporal discretization and parametrizing the decision functions over each grid interval. The resulting formulation is expressed in terms of a finite set of algebraic variables and constraints, while retaining the structure of the original cost functional and feasibility requirements.

Consider a temporal grid $\{t_k\}_{k=0}^{N}$ with

$$t_0 < t_1 < \cdots < t_N \triangleq t_f, \qquad \Delta t_k \triangleq t_{k+1} - t_k. \tag{2.49}$$

and define the nodal approximations $\mathbf{x}_k \approx \mathbf{x}(t_k)$ and $\mathbf{u}_k \approx \mathbf{u}(t_k)$. A direct transcription introduces a finite dimensional decision vector collecting all nodal values

$$\mathbf{w} \triangleq \left[\mathbf{x}_0^T, \quad \ldots, \quad \mathbf{x}_N^T, \quad \mathbf{u}_0^T, \quad \ldots, \quad \mathbf{u}_{N-1}^T\right]^T \in \mathbb{R}^{n_w}. \tag{2.50}$$

The continuous-time objective (2.38) is approximated by a quadrature rule compatible with the chosen parameterization, yielding a discrete cost of the form

$$J_d(\mathbf{w}) \triangleq \phi(\mathbf{x}_N, t_f) + \sum_{k=0}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k, t_k)\,\Delta t_k. \tag{2.51}$$

The differential constraints (2.33) are translated into algebraic relations that enforce discrete-time consistency between consecutive nodes. In general terms, this is achieved by introducing defect constraints of the form

$$\boldsymbol{\Delta}_k(\mathbf{x}_{k+1}, \mathbf{x}_k, \mathbf{u}_k) = \mathbf{0}, \qquad k = 0, \ldots, N-1, \tag{2.52}$$

where $\boldsymbol{\Delta}_k(\cdot)$ depends on the adopted discretization scheme.

Likewise, path and boundary constraints are discretized by enforcing them at the transcription points, producing sets of equalities and inequalities that approximate (2.34)–(2.37). Collecting the discrete objective (2.51) and all algebraic constraints yields an NLPin the general form stated in Equation (2.42), with $\mathbf{w}$ as decision vector.

Since constraints are enforced at a finite set of points, satisfaction between nodes is not guaranteed in general. The NLP solution accuracy and feasibility depend on the grid resolution and on the chosen transcription scheme. Moreover, the resulting NLP is typically nonconvex, so numerical solvers are expected to converge to locally optimal solutions.

2.6. Model Predictive Control

Model Predictive Control (MPC) is a control strategy in which control actions are obtained by repeatedly solving a finite-horizon OCP. At each sampling instant, the current state of the system is measured or estimated, an optimization problem is solved over a prediction horizon, and only the first control action of the resulting optimal sequence is applied to the system. The procedure is then repeated at the next sampling instant using updated state information. From a conceptual standpoint, MPC can be interpreted as the receding-horizon implementation of the corresponding finite-dimensional NLP obtained after discretization. Standard MPC formulations and receding-horizon concepts are discussed in [56].

Let the system be described by the discrete-time dynamics induced by the chosen transcription,

$$\mathbf{x}_{k+1} = \mathbf{F}_k(\mathbf{x}_k, \mathbf{u}_k), \tag{2.53}$$

and consider a finite prediction horizon of length N. At a generic time index k, MPC solves a finite-horizon optimal control problem of the form

$$\begin{aligned}
\min_{\{\mathbf{x}_i,\mathbf{u}_i\}_{i=k}^{k+N-1}} \quad & \phi(\mathbf{x}_{k+N}, t_{k+N}) + \sum_{i=k}^{k+N-1} L(\mathbf{x}_i, \mathbf{u}_i, t_i)\,\Delta t_i \\
\text{s.t.} \quad & \mathbf{x}_k = \hat{\mathbf{x}}_k, \\
& \mathbf{x}_{i+1} = \mathbf{F}_i(\mathbf{x}_i, \mathbf{u}_i), \qquad i = k, \dots, k+N-1, \\
& \mathbf{c}(\mathbf{x}_i, \mathbf{u}_i, t_i) \; = \; \mathbf{0}, \\
& \mathbf{d}(\mathbf{x}_i, \mathbf{u}_i, t_i) \leq \mathbf{0}, \\
& \underline{\mathbf{x}} \leq \mathbf{x}_i \leq \overline{\mathbf{x}}, \quad \underline{\mathbf{u}} \leq \mathbf{u}_i \leq \overline{\mathbf{u}},
\end{aligned} \tag{2.54}$$

where $\hat{\mathbf{x}}_k$ denotes the measured or estimated state at time t_k.
The optimization problem (2.54) has the same structure as the NLP derived from the continuous-time OCP, but with the initial state constraint updated at each time step.

Let $\{\mathbf{u}_i^\star\}_{i=k}^{k+N-1}$ denote an optimal solution of (2.54). The MPC control law is defined by the first element of the optimal input sequence

$$\boldsymbol{\kappa}_N(\hat{\mathbf{x}}_k) \;\triangleq\; \mathbf{u}_k^\star, \qquad \mathbf{u}_k = \boldsymbol{\kappa}_N(\hat{\mathbf{x}}_k), \tag{2.55}$$

which makes the resulting system evolve according to

$$\mathbf{x}_{k+1} = \mathbf{F}_k\big(\mathbf{x}_k, \boldsymbol{\kappa}_N(\hat{\mathbf{x}}_k)\big), \qquad k = 0, 1, 2, \dots \tag{2.56}$$

At the next sampling instant, the horizon is shifted forward by one step and the optimization problem (2.54) is re-solved with the updated initial condition $\mathbf{x}_{k+1} = \hat{\mathbf{x}}_{k+1}$.

This receding-horizon policy provides inherent feedback, allowing MPC to compensate for model mismatch, disturbances and uncertainties, while explicitly enforcing state and input constraints. Nonetheless, since the underlying NLP is generally nonlinear and nonconvex, practical MPC implementations typically guarantee convergence only to locally optimal solutions and must account for computational limitations in real-time applications.

3 | Problem Formulation

This work considers the guidance and control of a spacecraft operating in close proximity to a geometrically unknown target to support an optical imaging campaign for a deep learning based three-dimensional reconstruction. The reconstruction pipeline prescribes empirically tuned viewing conditions to ensure reliable neural rendering. This requires keeping the target within the camera Field of View, maintaining a bounded Line of Sight pointing error, and operating within a bounded range with a near constant standoff that preserves overlap across consecutive images. While the target orbital parameters are assumed known, its size and shape are not available a priori. As a result, the exploration trajectory cannot be fully specified in advance and must be determined sequentially.

A preliminary identification stage providing coarse geometric bounds is assumed completed prior to the phase addressed in this thesis.

Based on these assumptions, the chapter introduces the discrete-time predictive model used for guidance, formulates the associated finite-horizon Optimal Control Problem and outlines its transcription into a constrained finite-dimensional optimization problem.

3.1. Discrete-Time Predictive Model

Building upon the translational and rotational equations of motion introduced in Chapter 2, we now define the discrete-time Model Predictive Control within the receding-horizon optimizer. The purpose of this section is to formalize the optimization variables, the compact state–input representation and the discrete-time propagation used to enforce prediction consistency over the MPC horizon.

3.1.1. State and control parametrization

Let the MPC state and input be

$$\mathbf{x}_k \triangleq \begin{bmatrix} \boldsymbol{\rho}_k \\ \mathbf{v}_k \\ \boldsymbol{\omega}_k \\ \boldsymbol{\sigma}_k \end{bmatrix} \in \mathbb{R}^{12}, \qquad \mathbf{u}_k \triangleq \begin{bmatrix} \mathbf{a}_k \\ \boldsymbol{\tau}_k \end{bmatrix} \in \mathbb{R}^{6}. \tag{3.1}$$

Unless otherwise stated, $\boldsymbol{\rho}_k$ and $\mathbf{v}_k$ denote the chaser relative position and velocity expressed in the target LVLH frame $\mathcal{L}$, while $\boldsymbol{\omega}_k \equiv (\boldsymbol{\omega}_{\mathcal{B}/\mathcal{L}})_k$ and $\boldsymbol{\sigma}_k \equiv \boldsymbol{\sigma}_{\mathcal{B}/\mathcal{L},k}$ are the body angular rate and attitude (MRPs) defined in Chapter 2, expressed in $\mathcal{B}$. Similarly, $\mathbf{a}_k \equiv (\mathbf{a}_{\mathcal{L}})_k$ is the translational control acceleration in $\mathcal{L}$ and $\boldsymbol{\tau}_k$ is the control torque in $\mathcal{B}$. The sampling time is $T_s \triangleq \Delta t$ and the prediction horizon is N.

In the numerical implementation, the predictive model, constraints, and objective terms are formulated in dimensionless variables obtained through a fixed set of scaling constants, defined once and kept unchanged throughout the simulation. This normalization improves numerical conditioning and mitigates the large separation in magnitudes between translational and rotational quantities, while preserving the structure of the underlying model. Dimensional quantities are explicitly marked whenever used.

A distance scale $L^\star$ is adopted to represent the characteristic length of the inspection task. In this work, $L^\star$ is selected to be of the order of the nominal imaging distance, i.e., the range expected to provide favorable conditions for the reconstruction pipeline. The orbital time scale is defined as

$$T^\star \triangleq \frac{1}{n}, \tag{3.2}$$

namely the inverse mean motion, which sets the natural timescale of the LVLH dynamics. The associated velocity and acceleration scales follow as $V^\star = L^\star/T^\star$ and $A^\star = L^\star/(T^\star)^2$.

For the rotational dynamics, a representative inertia magnitude $J^\star$ is extracted from the physical inertia tensor, and a rotational rate scale $\Omega^\star_{\mathrm{rot}}$ is chosen to be of the order of the maximum admissible body rate. This yields the torque scale $\Gamma^\star = J^\star(\Omega^\star_{\mathrm{rot}})^2$. Finally, the dimensionless ratio

$$\eta \triangleq \frac{\Omega^\star_{\mathrm{orb}}}{\Omega^\star_{\mathrm{rot}}} = \frac{1/T^\star}{\Omega^\star_{\mathrm{rot}}}, \tag{3.3}$$

is introduced to capture the separation between the orbital and attitude time scales, and it appears explicitly in the dimensionless rotational model.

Scale	Definition	Normalization
Length	$L^\star$	$\boldsymbol{\rho} = \boldsymbol{\rho}_{\mathrm{phys}}/L^\star$
Time	$T^\star = 1/n$	$t = t_{\mathrm{phys}}/T^\star, \quad T_s = T_{s,\mathrm{phys}}/T^\star$
Velocity	$V^\star = L^\star/T^\star$	$\mathbf{v} = \mathbf{v}_{\mathrm{phys}}/V^\star$
Acceleration	$A^\star = L^\star/(T^\star)^2$	$\mathbf{a} = \mathbf{a}_{\mathrm{phys}}/A^\star$
Rate (orbital)	$\Omega^\star_{\mathrm{orb}} = 1/T^\star$	—
Rate (body)	$\Omega^\star_{\mathrm{rot}}$	$\boldsymbol{\omega} = \boldsymbol{\omega}_{\mathrm{phys}}/\Omega^\star_{\mathrm{rot}}$
Inertia	$J^\star$	$\mathbf{J} = \mathbf{J}_{\mathrm{phys}}/J^\star$
Torque	$\Gamma^\star = J^\star(\Omega^\star_{\mathrm{rot}})^2$	$\boldsymbol{\tau} = \boldsymbol{\tau}_{\mathrm{phys}}/\Gamma^\star$
Timescale ratio	$\eta = \Omega^\star_{\mathrm{orb}}/\Omega^\star_{\mathrm{rot}}$	—

Table 3.1: Adopted scaling constants and dimensionless variables.

3.1.2. Discrete-time predictive dynamics

The receding-horizon optimization requires a discrete-time predictor consistent with the translational and rotational equations of motion derived in Chapter 2. The aim here is to introduce a compact state space representation and the discrete-time propagation enforced over the horizon under piecewise constant inputs.

Define the translational state vector and control input as

$$\mathbf{x}_{p,k} \triangleq \begin{bmatrix} \boldsymbol{\rho}_k \\ \mathbf{v}_k \end{bmatrix} \in \mathbb{R}^6, \qquad \mathbf{u}_{p,k} \triangleq \mathbf{a}_k \in \mathbb{R}^3. \tag{3.4}$$

From the continuous-time CWH state-space model in Equation (2.8) and assuming $\mathbf{a}(t)$ constant over each interval $[t_k, t_{k+1})$, the exact discrete-time translational predictor reads

$$\mathbf{x}_{p,k+1} \;=\; \mathbf{A}_p\,\mathbf{x}_{p,k} + \mathbf{B}_p\,\mathbf{u}_{p,k}, \tag{3.5}$$

where $\mathbf{A}_p, \mathbf{B}_p$ follow from Equation (2.12). Since the translational dynamics are time-invariant for fixed (n, T_s), $\mathbf{A}_p, \mathbf{B}_p$ are computed once and reused across the horizon and across MPC iterations.

Define the rotational state vector and control input as

$$\mathbf{x}_{r,k} \triangleq \begin{bmatrix} \boldsymbol{\omega}_k \\ \boldsymbol{\sigma}_k \end{bmatrix} \in \mathbb{R}^6, \qquad \mathbf{u}_{r,k} \triangleq \boldsymbol{\tau}_k \in \mathbb{R}^3. \tag{3.6}$$

Since the rotational dynamics are nonlinear in $\mathbf{x}_r$, a first order affine approximation is therefore constructed along a reference rotational trajectory $\bar{\mathbf{x}}_r(t), \bar{\mathbf{u}}_r(t)$ over the prediction window, as detailed in Equation (2.25). By holding the resulting linearization coefficients constant over each sampling interval and applying the exact discretization presented in Equation (2.27), the discrete-time rotational predictor is written as

$$\mathbf{x}_{r,k+1} \;=\; \mathbf{A}_{r,k}\,\mathbf{x}_{r,k} + \mathbf{B}_{r,k}\,\mathbf{u}_{r,k} + \mathbf{f}_{r,k}. \tag{3.7}$$

Over the prediction horizon, the rotational dynamics are LTV affine through $(\mathbf{A}_{r,k}, \mathbf{B}_{r,k}, \mathbf{f}_{r,k})$. These coefficients are obtained from the current reference trajectory, kept fixed within each solve, and recomputed at the next MPC iteration after the reference is updated. The specific construction of the reference trajectory adopted in this work is part of the solution strategy and is discussed in Chapter 4.

Stacking the translational and rotational predictors yields the overall discrete-time model

$$\underbrace{\begin{bmatrix} \mathbf{x}_{p,k+1} \\ \mathbf{x}_{r,k+1} \end{bmatrix}}_{\mathbf{x}_{k+1}} = \underbrace{\begin{bmatrix} \mathbf{A}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{r,k} \end{bmatrix}}_{\mathbf{A}_k} \underbrace{\begin{bmatrix} \mathbf{x}_{p,k} \\ \mathbf{x}_{r,k} \end{bmatrix}}_{\mathbf{x}_k} + \underbrace{\begin{bmatrix} \mathbf{B}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{r,k} \end{bmatrix}}_{\mathbf{B}_k} \underbrace{\begin{bmatrix} \mathbf{u}_{p,k} \\ \mathbf{u}_{r,k} \end{bmatrix}}_{\mathbf{u}_k} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{f}_{r,k} \end{bmatrix}}_{\mathbf{d}_k}, \tag{3.8}$$

where $\mathbf{A}_k$ and $\mathbf{B}_k$ are block-diagonal and $\mathbf{d}_k$ collects the affine term induced by the attitude linearization. The coupled nature of the problem is captured by the optimization formulation introduced in the following sections.

3.2. Objectives

The guidance problem addressed in this work is formulated as a receding-horizon NLP with a finite prediction horizon, where the decision variables $\{\mathbf{x}_k, \mathbf{u}_k\}_{k=0}^{N}$ are constrained by the discrete-time dynamics in Equation (3.8) and by the operational constraints introduced in the following section. The objective is designed to optimize the close-proximity inspection performance over the prediction horizon. It promotes viewing conditions that support reliable image acquisition and downstream reconstruction, while explicitly accounting for the resource driven nature of spacecraft operations. In particular, imaging and surface mapping requirements must be balanced against actuation usage, since limited propellant and bounded control authority always constrain manoeuvres, long term mission performance and spacecraft mass.

Let $\ell_k(\mathbf{x}_k, \mathbf{u}_k)$ denote the stage cost at time t_k, and let $\ell_N(\mathbf{x}_N)$ denote the terminal contribution. The criterion minimized by the MPC is

$$J \triangleq \sum_{k=0}^{N-1} \ell_k(\mathbf{x}_k, \mathbf{u}_k) \;+\; \ell_N(\mathbf{x}_N). \tag{3.9}$$

Each ℓ_k is defined as a weighted combination of terms with different roles in the problem. Specifically,

$$\ell_k(\mathbf{x}_k, \mathbf{u}_k) \;=\; \ell_k^{\mathrm{img}}(\mathbf{x}_k) \;+\; \ell_k^{\mathrm{fuel}}(\mathbf{u}_k) \;+\; \ell_k^{\mathrm{exp}}(\mathbf{x}_k), \tag{3.10}$$

where ℓ_k^{img} promotes imaging-oriented performance, ℓ_k^{fuel} penalizes actuation effort as a fuel proxy, and ℓ_k^{exp} introduces additional state shaping to support exploration objectives. This partition is adopted to make the role of each component explicit within a single constrained NLP.

Each term is defined to reflect its intended physical role and is cast into a smooth penalty to ensure well behaved derivatives for gradient based solvers. Penalty functions with constraint like behavior are constructed using continuously differentiable surrogates, allowing requirements such as FoV compliance and range keeping to be enforced strongly while preserving smoothness. The choice of penalizing these requirements, rather than enforcing them as hard constraints, is discussed in Section 3.2.1 together with its implications for feasibility and robustness.

3.2.1. Imaging-Driven Terms

Achieving reliable deep learning based reconstruction requires maintaining viewing conditions identified by the reconstruction pipeline and validated through empirical tuning. In particular, the guidance is designed to keep the camera boresight within a prescribed Field of View about the Line of Sight to the target and to remain within prescribed range bounds while promoting a near-constant standoff distance around its nominal value. Accordingly, the imaging-driven stage cost is defined as

$$\ell_k^{\mathrm{img}}(\mathbf{x}_k) \;\triangleq\; w_{\mathrm{p}}\,\ell_k^{\mathrm{p}}(\mathbf{x}_k) \;+\; w_{\mathrm{r}}\,\ell_k^{\mathrm{r}}(\mathbf{x}_k), \tag{3.11}$$

where ℓ_k^{p} captures the pointing performance and ℓ_k^{r} enforces range keeping. The scalar weights w_{p} and w_{r} tune the relative importance of the two terms within the overall cost.

To interpret the imaging-related parameters, it is first useful to define the adopted viewing geometry. Let $\boldsymbol{\rho}_k \in \mathbb{R}^3$ denote the relative position expressed in $\mathcal{L}$, and define the range and the LOS direction as

$$r_k \;\triangleq\; \|\boldsymbol{\rho}_k\|_2, \qquad \mathbf{d}_{\mathcal{L},k} \;\triangleq\; -\frac{\boldsymbol{\rho}_k}{r_k}. \tag{3.12}$$

Let $\mathbf{b}_{\mathcal{B}} \in \mathbb{R}^3$ denote the camera boresight expressed in the body frame. Assuming a rigid mounting and the optical axis is aligned with the body x-axis

$$\mathbf{b}_{\mathcal{B}} \triangleq \hat{\mathbf{z}}_{\mathcal{C}}^{\mathcal{B}}, \qquad \hat{\mathbf{z}}_{\mathcal{C}}^{\mathcal{B}} \equiv \hat{\mathbf{x}}_{\mathcal{B}}. \tag{3.13}$$

Let $\mathbf{C}_{\mathcal{BL}}(\boldsymbol{\sigma}_k)$ be the direction cosine matrix mapping vectors from $\mathcal{L}$ to $\mathcal{B}$. The boresight expressed in $\mathcal{L}$ is then

$$\mathbf{b}_{\mathcal{L},k} \triangleq \mathbf{C}_{\mathcal{BL}}(\boldsymbol{\sigma}_k)^{\top}\,\mathbf{b}_{\mathcal{B}}. \tag{3.14}$$

The cosine of the boresight–LOS misalignment angle is

$$\varphi_k \triangleq \mathbf{b}_{\mathcal{L},k}^{\top}\mathbf{d}_{\mathcal{L},k} \tag{3.15}$$

so that $\varphi_k = 1$ corresponds to perfect alignment.

Pointing. The pointing penalty is designed with a dual purpose: it enforces a constraint-like behavior when the target leaves the camera FoV, and it simultaneously discourages unnecessary oscillations by promoting a perfect alignment. Let θ_{fov} denote the FoV half-angle and define $c_{\mathrm{fov}} \triangleq \cos\theta_{\mathrm{fov}}$, so that FoV compliance is expressed by $\varphi_k \geq c_{\mathrm{fov}}$. A smooth violation measure is introduced as

$$z_k \triangleq c_{\mathrm{fov}} - \varphi_k, \qquad s_k \triangleq \phi(z_k), \tag{3.16}$$

where $\phi(\cdot)$ is a smooth hinge surrogate hat activates only when the argument becomes positive, while remaining continuously differentiable. In this work, the implementation adopts the softplus form

$$\phi(z) \triangleq \frac{1}{\kappa}\log\bigl(1 + e^{\kappa z}\bigr), \qquad \kappa > 0. \tag{3.17}$$

The resulting pointing stage cost is defined as

$$\ell_k^{\mathrm{p}}(\mathbf{x}_k) \triangleq w_{\mathrm{p,out}}\, s_k^2 \;+\; w_{\mathrm{p,in}}\bigl(1 - \varphi_k\bigr), \tag{3.18}$$

where $w_{\mathrm{p,out}} \gg w_{\mathrm{p,in}} > 0$ imposes a large penalty outside the FoV cone ($z_k > 0$), while the second term provides a mild centering action within the cone ($z_k \leq 0$).

Range-keeping. Similarly to the pointing cost, the range term enforces an annulus-like behavior by penalizing deviations from an admissible inspection window while encouraging operation near a nominal distance, thus maintaining an approximately constant imaging scale along the inspection arc. Let $r_{\min}$ and $r_{\max}$ denote the admissible range bounds and let r_{opt} be a nominal distance reflecting the desired image scale for the reconstruction pipeline, such that $r_{opt} \in [r_{\min}, r_{\max}]$. Using the same smooth hinge surrogate $\phi(\cdot)$ introduced in Equation (3.17), the range-keeping cost is defined as

$$\ell_k^{\text{r}}(\mathbf{x}_k) \;\triangleq\; w_{\text{r,out}}\big(\phi(r_{\min} - r_k)^2 + \phi(r_k - r_{\max})^2\big) \;+\; w_{\text{r,in}}\left(\frac{r_k - r_{\text{opt}}}{r_{\max} - r_{\min}}\right)^2, \tag{3.19}$$

where $w_{\text{r,out}} \gg w_{\text{r,in}} > 0$ yields a high weight penalty outside the admissible window, while driving the chaser spacecraft towards r_{opt}.

The nominal distance r_{opt} is selected based on reconstruction-driven requirements expressed in terms of target image occupancy, defined as the fraction of the camera field of view subtended by the target. For a target of radius R_{tgt} and a camera FoV half-angle θ_{fov}, a desired occupancy level $\eta \in [\eta_{\min}, \eta_{\max}]$ defines an admissible apparent angular diameter $\delta = \eta\,\theta_{\text{fov}}$. Considering a spherical target, the corresponding line-of-sight distance ρ follows from the geometric relation

$$\sin(\delta/2) = \frac{R_{\text{tgt}}}{\rho}, \qquad r = \rho - R_{\text{tgt}}, \tag{3.20}$$

which induces an admissible radial interval $r \in [r_{\min}, r_{\max}]$ associated with $\eta_{\min}$ and $\eta_{\max}$. In this work, r_{opt} is chosen as a nominal value within this interval, consistently with the camera parameters and the occupancy targets adopted in the reconstruction pipeline.

3.2.2. Exploration Direction Tracking

A complete mapping of the unknown target requires multiple replanning cycles, as the high-level planner updates the desired exploration direction throughout the inspection. To guide the trajectory toward the current exploration direction, an additional state-shaping term is introduced. Let $\widehat{\mathbf{q}}_k \in \mathbb{R}^3$ denote the unit exploration direction in $\mathcal{L}$, and define the corresponding reference position as $\boldsymbol{\rho}_k^{\text{ref}} \triangleq r_{\text{opt}}\,\widehat{\mathbf{q}}_k$. A quadratic shaping term is then written as

$$\ell_k^{\text{exp}}(\mathbf{x}_k) \;\triangleq\; \big(\boldsymbol{\rho}_k - \boldsymbol{\rho}_k^{\text{ref}}\big)^\top \mathbf{Q}_\rho \big(\boldsymbol{\rho}_k - \boldsymbol{\rho}_k^{\text{ref}}\big), \tag{3.21}$$

where $\mathbf{Q}_\rho \succeq 0$ weights the attraction toward the exploration reference.

3.2.3. Fuel Proxy and Control Effort

In addition to imaging and exploration requirements, the optimization must remain compatible with spacecraft resource limitations. For this reason, propellant usage is accounted for through a proxy based on control actuation. Actuation is regularized via penalties on the commanded translational acceleration and attitude torque. Using the admissible maximum magnitudes $a_{\max}$ and $\tau_{\max}$ for normalization, the effort term is defined as the weighted quadratic form

$$\ell_k^{\text{fuel}}(\mathbf{u}_k) \triangleq w_a \frac{\|\mathbf{a}_k\|_2^2}{a_{\max}^2} + w_\tau \frac{\|\boldsymbol{\tau}_k\|_2^2}{\tau_{\max}^2}. \tag{3.22}$$

The acceleration term acts as a fuel proxy, whereas the torque term mainly regularizes attitude actuation and limits aggressive rotational commands.

3.3. Constraints

The NLP is subject to hard operational constraints enforcing safety and actuator limitations. Imaging-related requirements, namely FoV and range-keeping, are handled through penalty terms as in Section 3.2 rather than as hard constraints, since small and temporary deviations may help preserve feasibility and reduce control effort.

Dynamics constraints. For $k = 0, \ldots, N-1$, the state sequence is constrained to satisfy the discrete-time predictor Equation (3.8),

$$\mathbf{x}_{k+1} - \mathbf{A}_k\mathbf{x}_k - \mathbf{B}_k\mathbf{u}_k - \mathbf{d}_k = \mathbf{0}. \tag{3.23}$$

Keep-out constraint. Collision avoidance is enforced through a minimum relative distance from the target. The target is conservatively modeled as a sphere, although ellipsoidal keep-out sets could be adopted if a more detailed geometry were available. Let $r_k \triangleq \|\boldsymbol{\rho}_k\|_2$ and let $r_{\text{KOZ}} > 0$ denote the Keep-Out Zone (KOZ) radius. The constraint reads, for $k = 0, \ldots, N$,

$$r_k \geq r_{\text{KOZ}}. \tag{3.24}$$

Here r_{KOZ} is selected from a geometric bound on the target size R_{tgt} plus an operational margin, i.e., $r_{\text{KOZ}} = R_{\text{tgt}} + r_{\text{margin}}$, with $r_{\text{KOZ}} \leq r_{min}$. The constraint is enforced at the transcription nodes and for sufficiently small sampling times T_s inter-sample violations are unlikely in the present close-proximity regime and can be verified a posteriori by trajectory interpolation. While a strong range penalty may already discourage unsafe proximity in some scenarios, Equation (3.24) is imposed as a hard constraint to guarantee safety and maintain generality.

State and input bounds. Component-wise bounds are imposed to reflect admissible operating envelopes and actuator limits. For $k = 0, \dots, N$,

$$-\mathbf{v}_{\max} \leq \mathbf{v}_k \leq \mathbf{v}_{\max}, \qquad -\boldsymbol{\omega}_{\max} \leq \boldsymbol{\omega}_k \leq \boldsymbol{\omega}_{\max}, \tag{3.25}$$

and for $k = 0, \dots, N-1$,

$$-\mathbf{a}_{\max} \leq \mathbf{a}_k \leq \mathbf{a}_{\max}, \qquad -\boldsymbol{\tau}_{\max} \leq \boldsymbol{\tau}_k \leq \boldsymbol{\tau}_{\max}. \tag{3.26}$$

The attitude parametrization $\boldsymbol{\sigma}_k$ is left unconstrained, while the attitude motion is effectively limited through (3.25)–(3.26).

Command-rate constraints. To reflect actuator limitations and to prevent fast command changes, bounds are imposed on the control input variations. For $k = 1, \dots, N-1$,

$$-\Delta\mathbf{a}_{\max} \leq \mathbf{a}_k - \mathbf{a}_{k-1} \leq \Delta\mathbf{a}_{\max}, \qquad -\Delta\boldsymbol{\tau}_{\max} \leq \boldsymbol{\tau}_k - \boldsymbol{\tau}_{k-1} \leq \Delta\boldsymbol{\tau}_{\max}. \tag{3.27}$$

These bounds promote smoother control sequences and improve solver robustness.

3.4. MPC problem statement and transcription

The elements introduced in the previous sections can be collected into a finite-horizon constrained optimal control problem, which is solved in a receding-horizon formulation. At each MPC update, the guidance law is obtained by solving a finite-horizon constrained optimal control problem built from the predictive dynamics in Section 3.1.2, the objective function in Section 3.2 and the operational constraints in Section 3.3. Over a prediction window of N steps, the decision variables are the state and input sequences $\{\mathbf{x}_k, \mathbf{u}_k\}_{k=0}^{N}$, where $\mathbf{x}_k = [\mathbf{x}_{p,k}^\top \ \mathbf{x}_{r,k}^\top]^\top$ and $\mathbf{u}_k = [\mathbf{u}_{p,k}^\top \ \mathbf{u}_{r,k}^\top]^\top$ as stated in (3.4)–(3.6). Let $\hat{\mathbf{x}}_0$ denote the available state information at the current sampling instant, the finite-horizon MPC problem is stated as the NLP

$$
\begin{aligned}
\min_{\{\mathbf{x}_k,\mathbf{u}_k\}} \quad & \sum_{k=0}^{N-1} \ell_k(\mathbf{x}_k,\mathbf{u}_k) \ + \ \ell_N(\mathbf{x}_N) \\
\text{s.t.} \quad & \mathbf{x}_0 = \hat{\mathbf{x}}_0, \\
& \mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k + \mathbf{d}_k, && k = 0,\dots,N-1, \\
& \|\boldsymbol{\rho}_k\|_2 \geq r_{\text{KOZ}}, && k = 0,\dots,N, \\
& -\mathbf{x}_{\max} \leq \boldsymbol{x}_k \leq \mathbf{x}_{\max}, && k = 0,\dots,N, \\
& -\mathbf{u}_{\max} \leq \mathbf{u}_k \leq \mathbf{u}_{\max}, && k = 0,\dots,N-1, \\
& -\Delta\mathbf{u}_{\max} \leq \mathbf{u}_k - \mathbf{u}_{k-1} \leq \Delta\mathbf{u}_{\max} && k = 1,\dots,N-1.
\end{aligned}
\tag{3.28}
$$

Let $\{\mathbf{u}_k^\star\}_{k=0}^{N-1}$ be an optimal solution of Equation (3.28). The MPC law is defined by the first input of the optimal sequence

$$
\boldsymbol{\kappa}_N(\hat{\mathbf{x}}_0) \triangleq \mathbf{u}_0^\star, \qquad \mathbf{u}_0 = \boldsymbol{\kappa}_N(\hat{\mathbf{x}}_0), \tag{3.29}
$$

and the procedure is repeated at the next sampling instant after updating the initial condition and shifting the prediction window forward by one step.

4 | Software Architecture for Autonomous Target Mapping

This chapter presents the guidance, control and planning algorithms developed in this work. The objective is to acquire an image dataset that supports subsequent 3D reconstruction by promoting informative viewpoints and uniform coverage while limiting propellant consumption. Emphasis is placed on the rationale for the main design choices and on the practical implementation details that enable a modular, flexible architecture. The proposed framework is designed for online operation and to accommodate uncertainties inherent to the inspection task, including unknown target geometry, evolving coverage information, and data-driven updates available during the inspection.

The discussion first introduces the overall software architecture and the coverage memory representation used to track mapping progress and trigger replanning. It then details the episodic direction planner and the receding-horizon MPC implementation and finally presents the closed-loop setup used for validation.

4.1. High Level Architecture

The proposed architecture is organized into a set of interacting modules operating at different time scales. A high-level planner updates the exploration direction episodically, based on the current mapping progress and on replanning triggers. A receding-horizon MPC runs at every sampling instant and solves a constrained finite-horizon NLP to generate the control action. Mapping progress is tracked through a coverage memory that is updated at each step and provides the planner with a compact representation of what has already been observed. Closed-loop validation is carried out using a high-fidelity simulator, which propagates the system dynamics over each sampling interval under the applied control.

After the environment is initialized, the closed-loop execution follows a receding-horizon scheme. At each sampling time, the current coverage status is evaluated and the replanning logic decides whether the active exploration direction should be updated. Given the current state estimate and the selected direction, the corresponding NLP is posed consistently with the formulation in Chapter 3 and solved over the prediction horizon. The chaser then applies only the first control input of the optimal sequence over the next sampling interval. The high-fidelity simulator then integrates the state over the same interval and the coverage memory is updated based on the resulting geometry and visibility conditions. This cycle repeats until the termination criteria are met, as summarized in Figure 4.1.

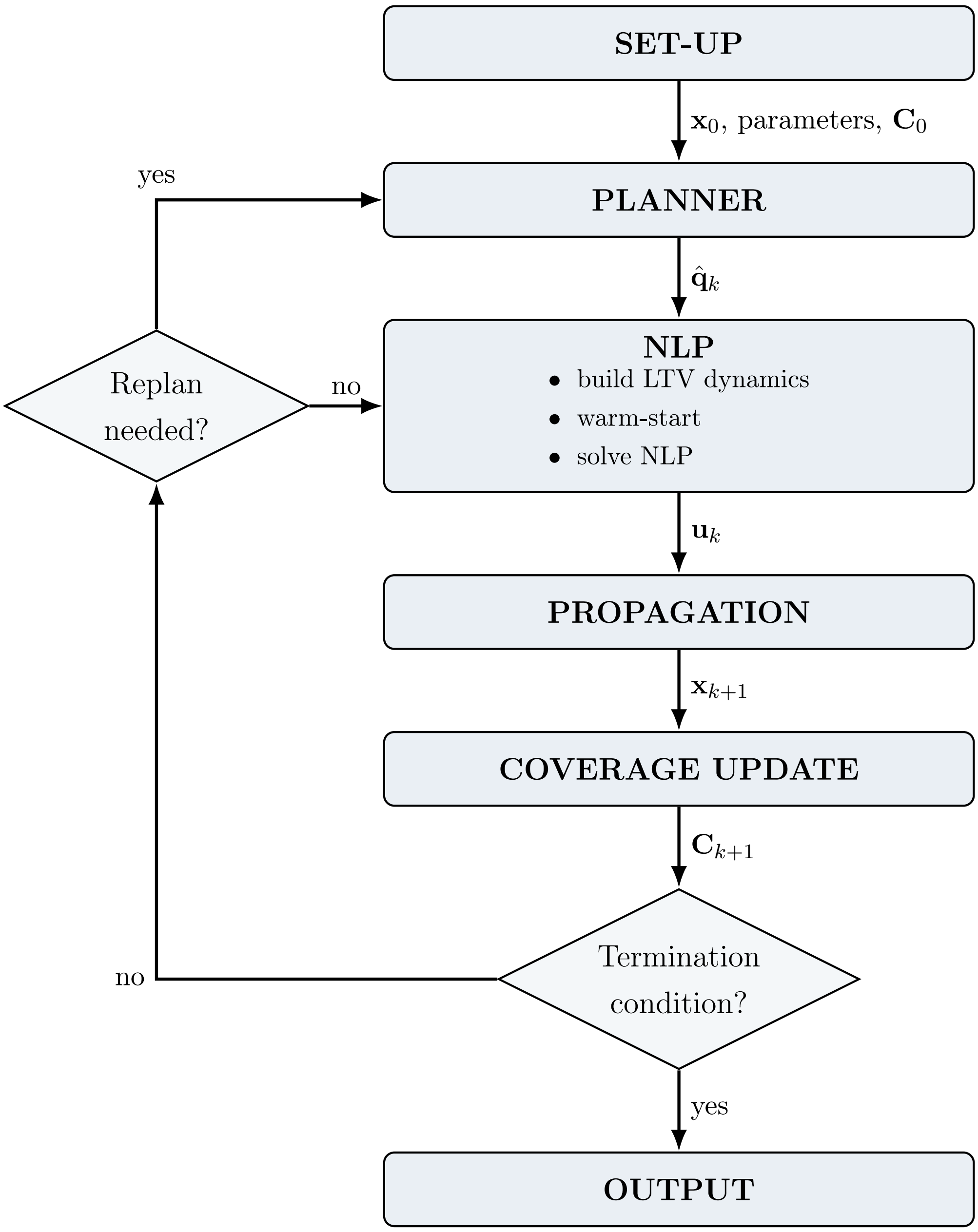


Figure 4.1: Closed-loop workflow.

4.2. Mapping process and coverage memory

This section introduces the geometric representation adopted to monitor mapping progress and to support the coverage driven planner.

4.2.1. Exploration space discretization

The goal of the mapping task is to acquire an image set that is as uniformly distributed over the target as possible. Since the target is non-cooperative and its geometry is not assumed known a priori, the discretization must be defined in a way that is simple, robust, and suitable for online replanning. For this reason, the target is initially approximated as a sphere. This assumption serves to provide a clean geometric baseline to define viewing regions and to quantify coverage. Once additional geometric information becomes available, the same construction can be generalized to an ellipsoid.

Given the admissible imaging radii $r_{\min}$ and $r_{\max}$ already defined in Section 3.2.1, the viewing space is restricted to the corresponding spherical shell. The remaining question is how to partition this shell into regions that support a uniform and well-distributed coverage of the target surface. This step is essential because it allows the mapping objective to be translated into a practical online requirement. This can be achieved by discretizing the unit sphere into a set of nearly equal-area directions and using them to induce a partition of the angular domain.

To generate N_d uniform viewing directions, a Spherical Fibonacci lattice is adopted. The method can be interpreted as wrapping a golden spiral around the sphere and sampling it at N_d points, as illustrated in Figure 4.2, providing a near-uniform distribution on the sphere. Let Φ denote the golden angle,

$$\Phi = \pi \left(3 - \sqrt{5}\right), \tag{4.1}$$

and let $i \in \{0, \ldots, N_d - 1\}$ be an index. The vertical coordinate z_i is chosen to be uniformly spaced, while the azimuth β_i is incremented by Φ,

$$z_i = 1 - \frac{2(i + \frac{1}{2})}{N_d}, \qquad \beta_i = i\,\Phi. \tag{4.2}$$

Defining $r_i = \sqrt{1 - z_i^2}$, the corresponding unit direction is then obtained as

$$\mathbf{q}_i = \begin{bmatrix} r_i \cos \beta_i \\ r_i \sin \beta_i \\ z_i \end{bmatrix}, \qquad \|\mathbf{q}_i\|_2 = 1. \tag{4.3}$$

The logic behind this parametrization is that linear spacing in z promotes equal-area sampling over latitude bands, while the irrational increment of β_i avoids repeated alignments along meridians. As a result, each direction $\mathbf{q}_i$ can be regarded as the nominal center of a region that, on average, represents approximately the same portion of the viewing sphere. This sampling strategy follows [57].

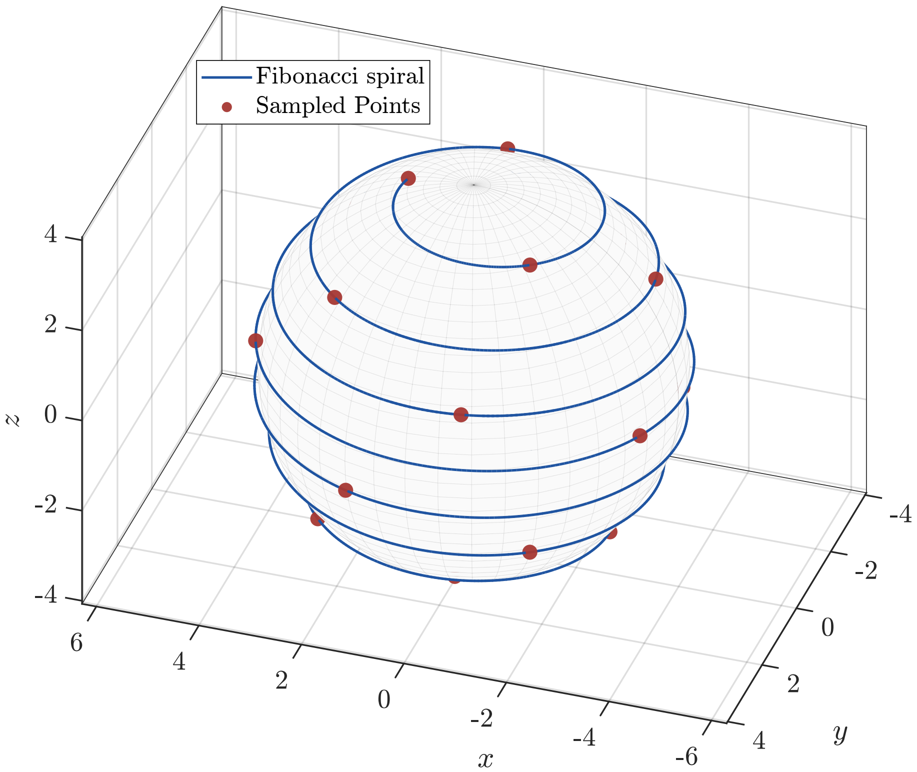


Figure 4.2: Fibonacci lattice with $N_d = 20$.

Once the direction set is defined, it is still necessary to associate each direction with a unique angular region. This is achieved by constructing a spherical Voronoi tessellation [58]. Intuitively, the Voronoi cell associated with $\mathbf{q}_i$ contains all unit vectors that are closer to $\mathbf{q}_i$ than to any other direction. The implementation follows a simple half-space construction: for each pair (i, j), the boundary between the two cells is described by a plane through the origin whose normal is the difference vector

$$\mathbf{n}_{ij} = \mathbf{q}_i - \mathbf{q}_j. \tag{4.4}$$

A unit vector $\mathbf{s} \in \mathbb{S}^2$ belongs to the half-space dominated by $\mathbf{q}_i$ if

$$(\mathbf{q}_i - \mathbf{q}_j)^\top \mathbf{s} \geq 0. \tag{4.5}$$

Therefore, the Voronoi cell associated with $\mathbf{q}_i$ is obtained as the intersection of all the corresponding half-spaces,

$$\mathcal{V}_i = \left\{ \mathbf{s} \in \mathbb{S}^2 \;\middle|\; (\mathbf{q}_i - \mathbf{q}_j)^\top \mathbf{s} \geq 0,\ \forall j \neq i \right\}. \tag{4.6}$$

For each index i the set of normals $\mathbf{n}_{ij}$ is precomputed and stored as a matrix $\mathbf{P}_i \in \mathbb{R}^{(N_d-1)\times 3}$, where each row corresponds to one separating plane. This makes region membership checks straightforward, since they reduce to evaluating a set of linear inequalities on the normalized direction.

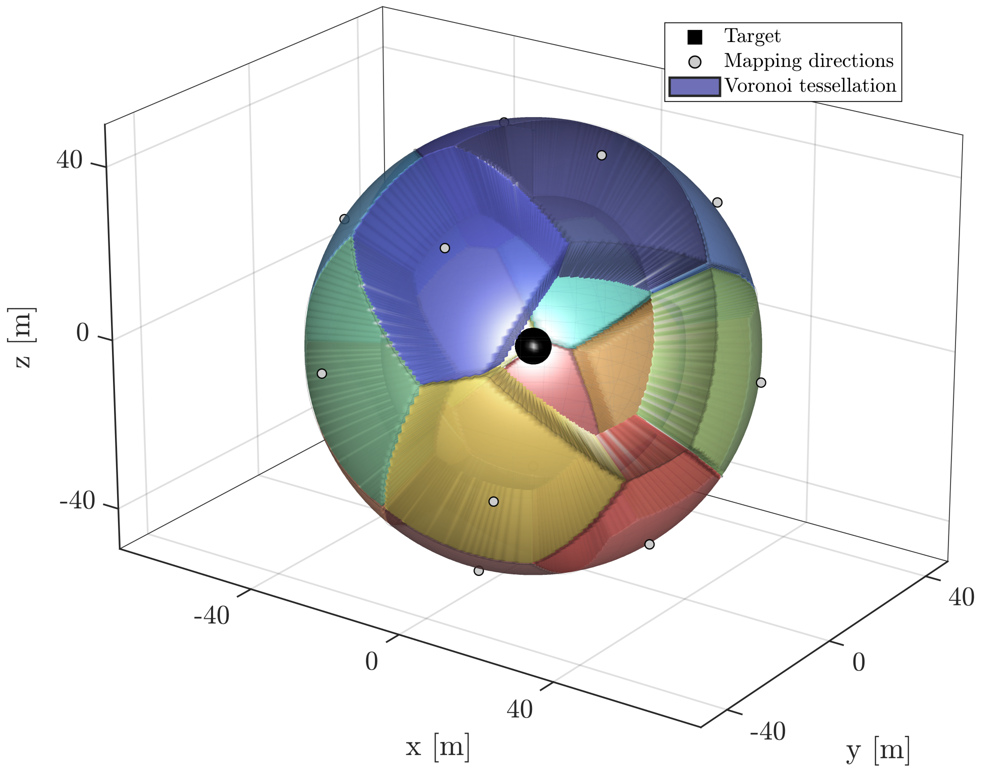


Figure 4.3: Voronoi tessellation and mapping zones with $N_d = 16$.

Finally, each angular cell is extended over the admissible radial interval $\left[r_{\min}, r_{\max}\right]$, yielding N_d volumetric mapping zones. As illustrated in Figure 4.3, the angular partition corresponds to a spherical Voronoi tessellation on the unit sphere. Each region then defines a volume inside the spherical shell bounded by the two radii, with lateral faces given by the separating planes and radial caps lying on the spheres of radii $r_{\min}$ and $r_{\max}$. This construction yields a clear geometric picture of the zones and leads to a simple membership check based on linear inequalities.

4.2.2. Coverage memory

The progress of target mapping is tracked online through a coverage memory that measures the amount of effective imaging time accumulated in each mapping zone. The objective is to encourage an approximately uniform coverage of the target, so that the collected dataset is informative for the subsequent 3-D reconstruction.

A key point is that the imaging requirements are enforced in the MPC through smooth penalties and are therefore not guaranteed at every sampling time. For this reason, the coverage is not incremented using the time spent in a region only. Instead, the update is conditioned to the instantaneous viewing geometry and range, so that the coverage represents time spent in a zone while the camera is effectively able to acquire usable images. At each step, the current state is processed by a dedicated module that evaluates whether the imaging conditions are satisfied and determines the mapping zones associated with the current line-of-sight direction.

Let $\mathbf{C}_k \in \mathbb{R}^{N_d}_{\geq 0}$ denote the coverage memory at time t_k, where $C_{k,i}$ is the accumulated effective imaging time in zone i. The update is computed from an imaging gate and a smooth zone membership vector. In the following, $\hat{\boldsymbol{\rho}}_k = \boldsymbol{\rho}_k / \|\boldsymbol{\rho}_k\|_2$ denotes the viewing direction (target $\rightarrow$ chaser), while the camera line-of-sight is $\mathbf{d}_{L,k} = -\hat{\boldsymbol{\rho}}_k$.

Imaging gate. The imaging gate is a scalar $g_k \in [0,1]$ that weights the update so that coverage accumulates only when the chaser lies within the admissible stand-off range and the camera orientation satisfies the FoV requirement. It is constructed as

$$g_k = g_k^{\text{range}} \, g_k^{\text{fov}}, \qquad g_k^{\text{range}} \in [0,1], \;\; g_k^{\text{fov}} \in [0,1]. \tag{4.7}$$

The range factor g_k^{range} depends on the current relative distance $r_k = \|\boldsymbol{\rho}_k\|$ and is designed to be close to one inside the admissible annulus and close to zero outside. In practice, it is implemented through smooth transitions around the two thresholds, so that the gate varies continuously when approaching or leaving the annulus.

The FoV factor g_k^{fov} is constructed from the same geometric quantities introduced in Section 3.2.1. It depends on the LOS unit vector $\mathbf{d}_{L,k}$ and on the camera boresight $\mathbf{b}_{L,k}$, both expressed in $\mathcal{L}$. The corresponding alignment metric

$$\varphi_k \;=\; \mathbf{b}_{L,k}^{\top}\mathbf{d}_{L,k} \;=\; \cos\theta_k \tag{4.8}$$

is used to define a smooth, near-binary gate such that $g_k^{\text{fov}} \approx 1$ when θ_k lies within the admissible FoV cone and $g_k^{\text{fov}} \approx 0$ otherwise. The FoV gate may be relaxed by a factor $\eta_{\text{fov}} \geq 1$ with respect to the cost threshold, to avoid discarding samples close to the boundary while still enforcing a meaningful usability criterion.

Zone membership weights. To avoid hard switching at region boundaries, the current angular location is mapped to a smooth membership vector $\mathbf{A}_k \in [0,1]^{N_d}$ with $\sum_{i=1}^{N_d} A_{k,i} \approx 1$. The weights are derived from directional cosines between the viewing direction $\hat{\boldsymbol{\rho}}_k$ and the pre-defined region directions $\{\mathbf{q}_i\}_{i=1}^{N_d}$. A smooth angular window assigns higher weight to the most compatible direction, and a normalized blending prevents numerical issues and erratic switching close to region boundaries.

Coverage accumulation. Given the sampling interval $\Delta t = T_s$, the per-step increment is defined as

$$\Delta\mathbf{C}_k \;=\; g_k\,\Delta t\;\mathbf{A}_k, \qquad \mathbf{C}_{k+1} \;=\; \mathbf{C}_k + \Delta\mathbf{C}_k, \tag{4.9}$$

which guarantees $\Delta\mathbf{C}_k \geq 0$ and accumulates coverage only when the current state satisfies the imaging conditions. A per-zone target level C_{tgt} is used as a completion reference. The resulting coverage memory is passed to the replanning logic, which triggers direction updates and declares mapping completion based on the current coverage status. Algorithm 4.1 compactly presents the resulting coverage update logic.

Algorithm 4.1 Coverage memory update

1: **Inputs:** current state $\mathbf{x}_k$, coverage memory $\mathbf{C}_k$, time step Δt, zone directions $\{\mathbf{q}_i\}_{i=1}^{N_d}$, range bounds $(r_{\min}, r_{\max})$, FoV threshold θ_{FOV}, FoV sharpness η_{fov}
2: **Outputs:** updated memory $\mathbf{C}_{k+1}$, incremental update $\Delta\mathbf{C}_k$
3: $r_k \leftarrow \|\boldsymbol{\rho}_k\|_2$
4: $\hat{\boldsymbol{\rho}}_k \leftarrow \boldsymbol{\rho}_k / r_k$ ▷ viewing direction
5: $\mathbf{d}_{L,k} \leftarrow -\hat{\boldsymbol{\rho}}_k$ ▷ line-of-sight
6: $g_k^{\mathrm{range}} \leftarrow \texttt{GateRange}\ (r_k,\ r_{\min}, r_{\max})$ ▷ soft range gate in $[0,1]$
7: $\varphi_k \leftarrow \mathbf{d}_{L,k}$ ▷ camera orientation metric
8: $g_k^{\mathrm{fov}} \leftarrow \texttt{GateFoV}\ (\varphi_k,\ \theta_{\mathrm{FOV}}, \eta_{\mathrm{fov}})$ ▷ soft FoV gate in $[0,1]$
9: $g_k \leftarrow g_k^{\mathrm{range}}\, g_k^{\mathrm{fov}}$
10: $\mathbf{A}_k \leftarrow \texttt{ZoneMembership}\ (\hat{\boldsymbol{\rho}}_k,\ \{\mathbf{q}_i\}_{i=1}^{N_d})$ ▷ find zone membership
11: $\Delta\mathbf{C}_k \leftarrow g_k\, \Delta t\, \mathbf{A}_k$ ▷ effective coverage increment for each zones
12: $\mathbf{C}_{k+1} \leftarrow \mathbf{C}_k + \Delta\mathbf{C}_k$ ▷ memory update

4.3. High-level planner

This section describes the supervisory planner that drives the exploration process by selecting a mapping direction. The planner does not compute a full spacecraft trajectory, rather it provides the MPC with a reference direction $\hat{\mathbf{q}}_k \in \mathbb{S}^2$ that shapes the guidance objective over the next sampling steps, while the MPC enforces the translational and attitude constraints and resolves the resulting closed-loop motion. At each call, the planner takes as input the current relative translational state subset $\mathbf{x}_{p,k} = [\boldsymbol{\rho}_k^\top\ \mathbf{v}_k^\top]^\top$, the coverage memory $\mathbf{C}_k$, the fixed candidate direction set $\{\mathbf{q}_i\}_{i=1}^{N_d}$ used for the mapping space discretization, and a short recent history memory that discourages immediate repetitions of the most recently selected directions. The main output are the selected direction $\hat{\mathbf{q}}_k$ to be tracked by the MPC and the predicted coverage increment $\mathbf{C}_{\mathrm{pred}}$ consistent with the update rule in Equation (4.9). To meet onboard runtime requirements, the planner follows a hierarchical coarse to fine procedure: cheap proxies are first used to screen the full set of candidates, a reduced set of short direction sequences is then evaluated more accurately and only the best few options are refined through a constrained short horizon optimization. This structure keeps the computational cost bounded while maintaining a tight coupling with the mapping progress encoded in $\mathbf{C}_{\mathrm{pred}}$.

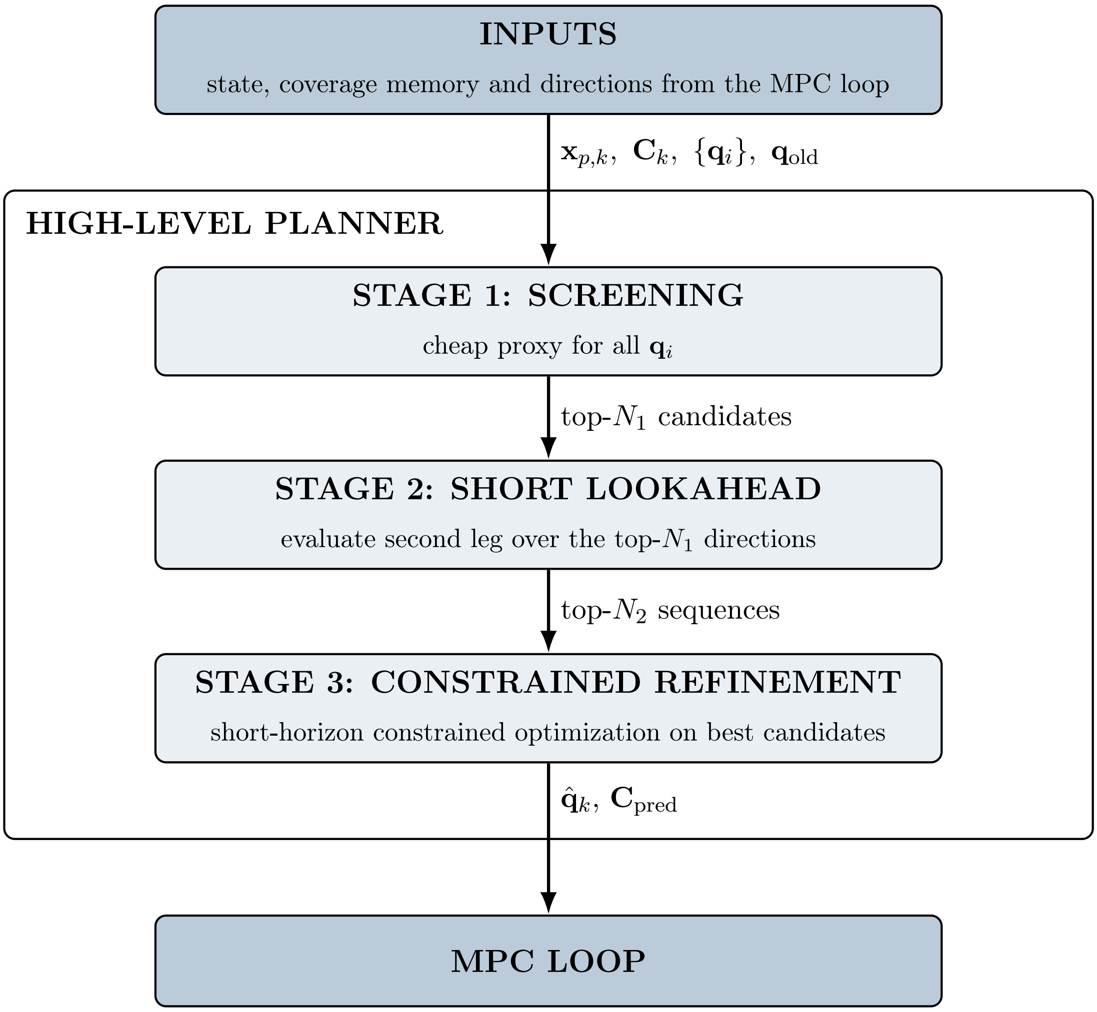


Figure 4.4: High-level planner architecture and MPC interface

Extending the lookahead beyond a few steps could, in principle, improve the quality of the selected directions. However, this is poorly suited to onboard execution and is fragile in an online mapping context: both the coverage memory and the current understanding of the target evolve in real time, so deeper open-loop optimizations are not only more expensive but also less informative, as their conclusions could readily superseded by the next measurements and by the resulting data-driven updates. For these reasons, the planner deliberately limits the lookahead depth and relies on repeated episodic updates to achieve long-term adaptation

4.3.1. Coarse stages: fast screening and short lookahead

The target direction generation process begins with two coarse stages that reduce the combinatorial set of candidate direction sequences to a size compatible with the subsequent refinement and with a bounded, predictable runtime. This is essential for real-time onboard operation, where timely updates of the mapping direction are required without compromising the responsiveness of the MPC loop.

Stage 1

Each candidate direction $\mathbf{q}_i$ is treated as a possible next reference for the mapping process and assigned a scalar score obtained by combining a mapping gain proxy with a turning angle based penalty. These two terms provide a fast approximation of the two quantities that ultimately matter at the MPC level, namely the expected coverage improvement and the actuation effort required to steer the motion toward the candidate viewpoint. The mapping proxy is computed from the coverage deficit encoded in $\mathbf{C}_k$ and favors directions associated with under observed zones, while the turning term discourages sharp changes in the local tangential motion, promoting smooth viewpoint evolution and reducing unnecessary manoeuvring.

Let C_{tgt} denote the target per-zone coverage. A normalized deficit is defined component-wise as

$$I_i \triangleq \max\left(0, \frac{C_{\text{tgt}} - C_{k,i}}{C_{\text{tgt}}}\right), \qquad i = 1, \dots, N_d, \tag{4.10}$$

and associated to any candidate $\mathbf{q}$ through a nearest direction rule on the canonical set $\{\mathbf{q}_i\}_{i=1}^{N_d}$. Specifically, define the nearest direction association

$$j(\mathbf{q}) \triangleq \arg \max_{j \in \{1,\dots,N_d\}} \mathbf{q}_j^\top \mathbf{q}, \tag{4.11}$$

and recall that $\{I_i\}_{i=1}^{N_d}$ denotes the per-direction normalized deficit introduced in eq. (4.10). The Stage-1 arrival mapping proxy is then

$$S^{(1)}_{\text{map,base}}(\mathbf{q}) \triangleq I_{j(\mathbf{q})}. \tag{4.12}$$

To account for mapping opportunities encountered while moving from the current radial direction $\hat{\boldsymbol{\rho}}_k = \boldsymbol{\rho}_k / \|\boldsymbol{\rho}_k\|_2$ to the candidate $\mathbf{q}$, the proxy is augmented by sampling L intermediate viewing directions along the great-circle arc connecting $\hat{\boldsymbol{\rho}}_k$ and $\mathbf{q}$ on $\mathbb{S}^2$. Let $\theta(\mathbf{q}) \triangleq \arccos(\hat{\boldsymbol{\rho}}_k^\top \mathbf{q})$ and select L angles $\theta_\ell \in (0, \theta)$ uniformly spaced, i.e.,

$$\theta_\ell \triangleq \frac{\ell}{L+1}\, \theta(\mathbf{q}), \qquad \ell = 1, \dots, L. \tag{4.13}$$

The corresponding unit directions on the arc can be written in closed form via spherical linear interpolation. Defining the unit tangent at $\hat{\boldsymbol{\rho}}_k$ toward $\mathbf{q}$ as

$$\mathbf{t}(\mathbf{q}) \triangleq \frac{\mathbf{q} - (\hat{\boldsymbol{\rho}}_k^\top \mathbf{q})\, \hat{\boldsymbol{\rho}}_k}{\left\|\mathbf{q} - (\hat{\boldsymbol{\rho}}_k^\top \mathbf{q})\, \hat{\boldsymbol{\rho}}_k\right\|_2}, \tag{4.14}$$

each intermediate direction is

$$\mathbf{m}_\ell(\mathbf{q}) \triangleq \cos(\theta_\ell)\, \hat{\boldsymbol{\rho}}_k + \sin(\theta_\ell)\, \mathbf{t}(\mathbf{q}), \qquad \ell = 1, \dots, L. \tag{4.15}$$

Each midpoint yields an intermediate LOS $\mathbf{d}_\ell(\mathbf{q}) = -\mathbf{m}_\ell(\mathbf{q})$ that is mapped to the nearest canonical direction through eq. (4.11), i.e.,

$$j_\ell(\mathbf{q}) \triangleq j(\mathbf{d}_\ell(\mathbf{q}))\,, \qquad \ell = 1, \dots, L. \tag{4.16}$$

The transition contribution is computed by accumulating the deficits of the distinct directions visited along the arc. Introducing

$$\delta_i(\mathbf{q}) \triangleq \begin{cases} 1, & \exists\, \ell \in \{1, \dots, L\} \text{ s.t. } j_\ell(\mathbf{q}) = i, \\ 0, & \text{otherwise}, \end{cases} \qquad i = 1, \dots, N_d, \tag{4.17}$$

the transition term is

$$S^{(1)}_{\text{map,trans}}(\mathbf{q}) \triangleq \kappa \sum_{i=1}^{N_d} \delta_i(\mathbf{q})\, I_i, \tag{4.18}$$

and the overall Stage-1 mapping proxy becomes

$$S^{(1)}_{\text{map}}(\mathbf{q}) \triangleq S^{(1)}_{\text{map,base}}(\mathbf{q}) + S^{(1)}_{\text{map,trans}}(\mathbf{q}). \tag{4.19}$$

The turning term discourages directions that would require a large change in the current orbital velocity around the target. The underlying rationale is that sharp changes in the direction of the velocity typically correlate with higher control effort, whereas candidates that remain aligned with the present motion tend to yield smoother, lower-ΔV evolution.

Let $\mathbf{P}_{T,k} = \mathbf{I} - \mathbf{d}_{L,k}\mathbf{d}_{L,k}^\top$ be the projector onto the plane orthogonal to $\mathbf{d}_{L,k}$. The motion around the target is captured by $\mathbf{v}_{t,k} = \mathbf{P}_{T,k}\mathbf{v}_k$. For each candidate $\mathbf{q}_i$, the corresponding in-plane direction is obtained by projection and renormalization, $\hat{\mathbf{q}}_{t,i} \propto \mathbf{P}_{T,k}\mathbf{q}_i$, and compared to the normalized tangential velocity $\hat{\mathbf{v}}_{t,k} \propto \mathbf{v}_{t,k}$ through the turn angle

$$\theta_i \triangleq \arccos\!\left(\hat{\mathbf{v}}_{t,k}^\top \hat{\mathbf{q}}_{t,i}\right) \in [0, \pi]. \tag{4.20}$$

A smooth saturating map then converts θ_i into a bounded penalty $S^{(1)}_{\text{turn}}(\mathbf{q}_i) \in [0,1]$, with low values for small turns and a rapidly increasing penalty as θ_i approaches π (near reversal).

The resulting Stage-1 score is summarized as

$$S^{(1)}(\mathbf{q}_i) \; = \; w_{\text{map}}\, S^{(1)}_{\text{map}}(\mathbf{q}_i) \; - \; w_{\text{turn}}\, S^{(1)}_{\text{turn}}(\mathbf{q}_i), \tag{4.21}$$

where w_{map} and w_{turn} are design weights, $S^{(1)}_{\text{map}}$ is the mapping proxy and $S^{(1)}_{\text{turn}}$ is the turning penalty. The top n candidates are retained for the next stage. The parameter n is chosen to discard clearly suboptimal directions while preserving enough diversity for the subsequent short lookahead evaluation.

Stage 2

Stage 2 extends the one step screening by explicitly evaluating short sequences $(\mathbf{q}_0, \mathbf{q}_1)$ built from the Stage 1 shortlist. For each retained first step direction $\mathbf{q}_0$, the same scoring logic is repeated over all admissible second step candidates. This two step lookahead mitigates the myopia of purely one step decisions: a direction that appears optimal at the next update may lead to poor follow-up options once the viewpoint has shifted, whereas a shallow two step evaluation already captures the dominant consequence of committing to $\mathbf{q}_0$ before selecting the subsequent update.

For each $\mathbf{q}_0$, Stage 2 forms short sequences by retaining the Stage 1 score already computed for $\mathbf{q}_0$ and by evaluating an additional second step contribution associated with candidate directions $\mathbf{q}_1$. The latter uses the same deficit-based mapping proxy introduced in Stage 1, while the turning penalty is updated using a predicted tangential heading at the end of the first segment. This heading is obtained through a purely geometric approximation on the unit sphere, using $\mathbf{d}_{L,k}$ as the current LOS direction and $\mathbf{d}_1(\mathbf{q}_0) \triangleq -\mathbf{q}_0$ as the nominal LOS direction after reaching the reference radius r_{opt} along $\mathbf{q}_0$. A geodesic arrival tangent is defined by projecting $\mathbf{d}_{L,k}$ onto the tangent plane at $\mathbf{d}_1$ and normalizing,

$$\mathbf{t}_{\text{geo}}(\mathbf{q}_0) \; \triangleq \; \frac{\left(\mathbf{I} - \mathbf{d}_1\mathbf{d}_1^\top\right)\mathbf{d}_{L,k}}{\left\|\left(\mathbf{I} - \mathbf{d}_1\mathbf{d}_1^\top\right)\mathbf{d}_{L,k}\right\|_2}. \tag{4.22}$$

If the projection in eq. (4.22) becomes ill-conditioned (nearly zero norm), the arrival heading is regularized by falling back to the current tangential direction.

To preserve continuity with the current motion when a meaningful tangential component exists, the predicted arrival heading blends $\mathbf{t}_{\text{geo}}(\mathbf{q}_0)$ with the current unit tangential velocity direction $\hat{\mathbf{v}}_{t,0} = \mathbf{v}_{t,k}/\|\mathbf{v}_{t,k}\|_2$,

$$\hat{\mathbf{v}}_{t,1}(\mathbf{q}_0) \triangleq \frac{(1-\lambda)\,\mathbf{t}_{\text{geo}}(\mathbf{q}_0) + \lambda\,\hat{\mathbf{v}}_{t,0}}{\left\|(1-\lambda)\,\mathbf{t}_{\text{geo}}(\mathbf{q}_0) + \lambda\,\hat{\mathbf{v}}_{t,0}\right\|_2}, \qquad \lambda \in [0,1], \tag{4.23}$$

where λ is a tuning parameter that regulates how strongly the predicted arrival heading follows the current tangential motion. When $\|\mathbf{v}_{t,k}\|_2$ is negligible, the geodesic term is used alone. The second step turning penalty for $\mathbf{q}_1$ is then evaluated exactly as in Stage 1 by replacing the reference direction $\hat{\mathbf{v}}_{t,0}$ with $\hat{\mathbf{v}}_{t,1}(\mathbf{q}_0)$ in the tangent plane cosine similarity, while keeping the same projection and saturation rules.

The cumulative score for a pair is finally computed as

$$S^{(1,2)}(\mathbf{q}_0, \mathbf{q}_1) \triangleq S^{(1)}(\mathbf{q}_0) + S^{(2)}(\mathbf{q}_0, \mathbf{q}_1), \tag{4.24}$$

where $S^{(2)}$ preserves the Stage 1 weight structure. Only the best g pairs are retained, where g is a hyperparameter that trades off exploration breadth against computational effort in the subsequent refinement stage.

Near completion, when the mean clamped coverage approaches the target level, an additional pruning step can be activated to discard pairs whose combined mapping proxy falls below a small threshold. This reduces late phase computation spent on sequences that are unlikely to increase $\mathbf{C}_k$ appreciably and accelerates convergence to a uniform coverage distribution, this threshold can be tuned depending on the desired balance between computational margin and conservative completion behavior.

Recent history discouragement. A short recent history memory $\mathbf{q}_{\text{old}}$ stores the indices of the most recently selected directions and is used to discourage immediate repetitions. In Stage 1, recently selected directions are treated as inadmissible and are removed from the candidate set. In Stage 2, the second step selection is additionally constrained to avoid repeating the current first step direction and the most recent element of the history. This lightweight exclusion rule prevents chattering between a small subset of directions, promotes exploration diversity and preserves the discrete nature of the planner, without introducing additional continuous decision variables or increasing the computational burden of the subsequent refinement stage.

4.3.2. Refinement stage: constrained optimization

The final stage refines the best few candidates by explicitly accounting for translational dynamics and control bounds through a constrained optimization. In contrast to the 6-

DoF problem in Chapter 3, this refinement layer is intentionally restricted to the 3-DoF translational subsystem. Attitude pointing and FoV feasibility are therefore not optimized at this level and are delegated to the MPC, which enforces the full set of translational and rotational constraints in closed loop. This separation yields a lightweight, fast subproblem that remains consistent with the same objective and with the translational dynamics and constraints adopted by the MPC.

Considering just the translational state and input vectors

$$\mathbf{x}_{p,j} \triangleq \begin{bmatrix} \boldsymbol{\rho}_j \\ \mathbf{v}_j \end{bmatrix} \in \mathbb{R}^6, \qquad \mathbf{u}_{p,j} \triangleq \mathbf{a}_j \in \mathbb{R}^3, \tag{4.25}$$

and, for a fixed candidate direction $\mathbf{q}$, defining the per-step tracking error

$$\mathbf{e}_{\rho,j}(\mathbf{q}) \triangleq \boldsymbol{\rho}_j - r_{\text{opt}}\mathbf{q}, \qquad \mathbf{e}_{\rho,N}(\mathbf{q}) \triangleq \boldsymbol{\rho}_N - r_{\text{opt}}\mathbf{q}, \tag{4.26}$$

the objective function of this optimization problem can be written in quadratic form as

$$J(\mathbf{q}) \triangleq \sum_{j=0}^{N-1} \big(w_j \, \mathbf{e}_{\rho,j}(\mathbf{q})^\top \mathbf{Q} \, \mathbf{e}_{\rho,j}(\mathbf{q}) \; + \; \Delta t \, \mathbf{a}_j^\top \mathbf{R} \, \mathbf{a}_j \big) \; + \; \mathbf{e}_{\rho,N}(\mathbf{q})^\top \mathbf{W} \, \mathbf{e}_{\rho,N}(\mathbf{q}), \tag{4.27}$$

where $\mathbf{Q}, \mathbf{R}, \mathbf{W} \in \mathbb{S}^3_{++}$ are design weight matrices. The coefficients $w_j > 0$ are normalized discount weights chosen to progressively emphasize late-horizon tracking while avoiding overly aggressive early motion; in the implementation, they are generated through an exponential profile and then normalized to preserve a consistent overall scaling of the stage terms.

The corresponding refinement subproblem for each $\mathbf{q}$ is formulated as a constrained finite-horizon NLP that combines:

(i) The discrete-time CWH dynamics presented in Section 2.3.1.

(ii) A radial annulus constraint $r_{\min}^2 \le \|\boldsymbol{\rho}_j\|_2^2 \le r_{\max}^2$ over the horizon.

(iii) Component-wise box bounds on $\boldsymbol{\rho}_j$, $\mathbf{v}_j$, and $\mathbf{a}_j$, consistently with the constraints introduced in Section 3.3.

Compared to the full 6-DoF closed-loop optimization, the present subproblem has lower dimension, LTI dynamics, and a quadratic objective. The only nonlinearity is the radial annulus constraint, implemented through $\|\boldsymbol{\rho}_j\|_2^2$ to avoid square roots and preserve a sparse constrained structure.

The refinement is applied only to the top g direction pairs retained after the coarse two-step stage. For each pair $(\mathbf{q}_0, \mathbf{q}_1)$, the translational subproblem is solved first for $\mathbf{q}_0$ starting from the current state $\mathbf{x}_{p,k}$, the terminal state of the resulting segment is then used as the initial condition for the solve associated with $\mathbf{q}_1$. This sequential roll-out preserves dynamic consistency across the pair while keeping the optimization horizon bounded.

To reduce the runtime, each solve is implemented in parametric form and handled through a small bank of pre-compiled solvers. The bank is indexed by a scaling factor α, which modulates an effective discretization step

$$\Delta t \;=\; \alpha\, T_s, \tag{4.28}$$

while keeping the horizon length N fixed. This provides a simple way to adjust the effective time allocated to reach a given direction, so that both nearby and distant candidates can be evaluated without changing the problem size. For each segment, the algorithm starts from the previously selected α and then explores the closest neighbouring values on the grid, proceeding outward only if necessary. The search is terminated as soon as the terminal tracking error satisfies $\|\mathbf{e}_{\rho,N}(\mathbf{q})\|_2 \leq \varepsilon_\rho$, where $\varepsilon_\rho > 0$ is a design tolerance.

Each refined candidate yields a short predicted translational trajectory $\{\boldsymbol{\rho}_j\}_{j=0}^{N}$ and the associated control sequence $\{\mathbf{a}_j\}_{j=0}^{N-1}$. Its expected mapping benefit is then estimated by applying, along the predicted trajectory, the same coverage memory update law used in the closed-loop. At this refinement level, the update is evaluated assuming feasible imaging conditions, while range and FoV constraints are enforced when the selected direction is executed by the MPC.

Starting from the current memory $\mathbf{C}_k$, the update is applied at each node $j = 0, \ldots, N-1$, producing a predicted sequence $\{\mathbf{C}_k^{(j)}\}_{j=0}^{N}$ with $\mathbf{C}_k^{(0)} \triangleq \mathbf{C}_k$ and $\mathbf{C}_k^{(j+1)}$ obtained by evaluating the mapping contribution at the predicted position $\boldsymbol{\rho}_j$. The resulting mapping score is then quantified as the cumulative increase of the coverage memory after capping each entry at the target level C_{tgt}, so that directions that have already reached the desired coverage do not contribute further to the predicted coverage increment.

$$S_{\mathrm{map}}(\mathbf{q}) \;\triangleq\; \sum_{j=0}^{N-1} \sum_{i=1}^{N_d} \left(\min\!\Big(C_{k,i}^{(j+1)},\, C_{\mathrm{tgt}}\Big) - \min\!\Big(C_{k,i}^{(j)},\, C_{\mathrm{tgt}}\Big) \right), \tag{4.29}$$

which is consistent with the online completion logic and prevents already saturated regions from dominating the score. The fuel proxy is computed from the refined control sequence as a normalized quadratic control effort,

$$S_f(\mathbf{q}) \triangleq \frac{1}{a_{\max}^2} \sum_{j=0}^{N-1} \|\mathbf{a}_j\|_2^2, \tag{4.30}$$

and the refinement score is defined for each segment as

$$S(\mathbf{q}) \triangleq w_{\text{map}}\, S_{\text{map}}(\mathbf{q}) - w_f\, S_f(\mathbf{q}), \tag{4.31}$$

with $w_{\text{map}}, w_f > 0$ denoting design weights. For each retained pair $(\mathbf{q}_0, \mathbf{q}_1)$, the total score is obtained by accumulating the contributions of the two consecutive segments,

$$S_{\text{tot}}(\mathbf{q}_0, \mathbf{q}_1) \triangleq S(\mathbf{q}_0) + S(\mathbf{q}_1), \tag{4.32}$$

The planner selects the pair that maximizes S^{tot} and returns the corresponding first step direction $\hat{\mathbf{q}}_k = \mathbf{q}_0^\star$, along with the associated predicted coverage after the first refinement segment,

$$\mathbf{C}_{\text{pred}} \triangleq \mathbf{C}_k^{(N)}(\mathbf{q}_0^\star),$$

obtained by evolving the coverage update model while tracking $\mathbf{q}_0^\star$.

Adaptive prioritization. The exploration–fuel trade-off is adjusted online based on a scalar completion indicator that reflects the current level of mapping completion. As coverage increases, the incremental contribution of the mapping score S_{map}decreases and eventually tends to zero, while the maneuvering cost remains of comparable magnitude. To maintain a balanced optimization and to guarantee completion of the mapping task, the weight associated with the mapping term is therefore increased according to a smooth exponential schedule as completion progresses, whereas the fuel-related weight is kept fixed. This adaptive weighting is applied consistently across all planning stages.

4.4. Receding-horizon MPC execution loop

This section describes the online receding-horizon execution of the MPC controller and summarizes the numerical steps that make the closed-loop routine efficient and repeatable in real time (Algorithm 4.2). At each sampling instant t_k, the MPC receives the estimated state $\hat{\mathbf{x}}_k$, the coverage memory $\mathbf{C}_k$, and the active exploration direction $\hat{\mathbf{q}}_k$, which is updated by the high-level planner when a replanning trigger occurs. Given these inputs, the finite-horizon NLP is assembled and solved, and only the first element of the optimal control sequence is applied over $[t_k, t_{k+1}]$. After propagation, the feedback state estimate and the coverage memory are updated and passed to the next step. The loop is repeated until the termination condition in Section 4.4.3 is met.

Algorithm 4.2 Receding-horizon MPC scheme

```
Inputs: initial conditions(x̂_0, C_0) and first guesses(X_guess, U_guess, X̄, Ū)_{k=0}
while C̄_k < α C_tgt, do                                  ▷ average coverage termination criteria
    CheckReplanConditions (x̂_k, C_k, q̂_k)                ▷ monitor mapping progress
    if replanning is triggered then
        w_map ← UpdateWeight (C_k)                        ▷ adaptive exploration–fuel balance
        q̂_k, C_pred ← HighLevelPlanner (x̂_k, C_k, w_map)  ▷ update target direction
    end if
    (X̄, Ū) ← BuildReference (x̂_k, X*, U*)                ▷ shift of the previous solution
    p_rot ← BuildLTV(X̄, Ū)                               ▷ build LTV dynamics
    p ← [x̂_k^⊤  q̂_k^⊤  p_rot^⊤]^⊤                         ▷ prepare solver parameters
    (X*, U*) ← SolveNLP(p, X_guess, U_guess)
    u_k ← U*(:, 1)
    x̂_{k+1} ← Propagate(x̂_k, u_k)                        ▷ real world propagation
    C_{k+1} ← UpdateCoverage(x̂_{k+1}, C_k)               ▷ coverage memory update
    (X_guess, U_guess) ← ShiftAndRollout(X*, U*, x̂_{k+1}) ▷ warm start for next solve
    C̄_k ← C_k
    x̂_k ← x̂_{k+1},   C_k ← C_{k+1}
    k ← k + 1
end while
```

4.4.1. Numerical solution of the NLP

At each MPC update, the next control action is obtained by solving the finite dimensional NLP in Equation (3.28), which results from transcribing the discrete-time predictive model over a horizon of length N. The decision vector stacks the state variables at the grid nodes and the piecewise-constant control inputs over the intervals,

$$\mathbf{w} \triangleq \begin{bmatrix} \mathbf{x}_0^\top & \cdots & \mathbf{x}_N^\top & \mathbf{u}_0^\top & \cdots & \mathbf{u}_{N-1}^\top \end{bmatrix}^\top, \tag{4.33}$$

while the parameter vector $\mathbf{p}$ collects the quantities updated online, namely the estimated initial condition and the planner provided direction, together with additional parameters required by the predictive model and by the transcription, which are detailed Section 4.4.2

$$\mathbf{p} \triangleq \begin{bmatrix} \hat{\mathbf{x}}_k^\top & \hat{\mathbf{q}}_k^\top & \mathbf{u}_{k-1}^\top & \mathbf{p}_{\text{rot}}^\top \end{bmatrix}^\top. \tag{4.34}$$

Using the notation introduced in Section 2.5 and the KKT conditions (2.44)–(2.48), Equation (2.54) can be written in the generic form

$$\min_{\mathbf{w}} \; f(\mathbf{w}, \mathbf{p}) \quad \text{s.t.} \quad \mathbf{g}(\mathbf{w}, \mathbf{p}) = \mathbf{0}, \qquad \mathbf{h}(\mathbf{w}, \mathbf{p}) \leq \mathbf{0}, \qquad \mathbf{w}_L \leq \mathbf{w} \leq \mathbf{w}_U, \tag{4.35}$$

where $\mathbf{g}$ enforces dynamic consistency across the horizon through the defect constraints and imposes the initial condition $\mathbf{x}_0 = \hat{\mathbf{x}}_k$, while $\mathbf{h}$ collects the nonlinear path constraints that are not representable as simple bounds. The horizon-local coupling of the constraints and the node wise enforcement of bounds yield sparse derivatives, which are directly exploited by the NLP solver.

The resulting sparse NLP is solved with IPOPT, which implements a primal-dual interior-point filter line-search method [59] and is consistent with the KKT framework presented in Section 2.5. Inequality constraints are handled via slack variables by introducing non-negative $\mathbf{s} \geq \mathbf{0}$ such that,

$$\mathbf{h}(\mathbf{w}, \mathbf{p}) + \mathbf{s} = \mathbf{0}, \qquad \mathbf{s} \geq \mathbf{0}, \tag{4.36}$$

and a logarithmic barrier is applied to enforce strict interiority of the slack and bound variables. Accordingly, for a given barrier parameter $\mu > 0$, the standard Lagrangian is replaced by the barrier augmented one,

$$\mathcal{L}_\mu = f(\mathbf{w}, \mathbf{p}) + \boldsymbol{\lambda}^\top \mathbf{g}(\mathbf{w}, \mathbf{p}) + \boldsymbol{\nu}^\top \big(\mathbf{h}(\mathbf{w}, \mathbf{p}) + \mathbf{s}\big) - \mu \sum_i \log s_i \; + \; \mathcal{B}_\mu^{\text{bnd}}(\mathbf{w}), \tag{4.37}$$

where $\mathcal{B}_\mu^{\text{bnd}}(\mathbf{w})$ denotes the logarithmic barrier associated with simple bounds. The KKT conditions (2.44)–(2.48) are then enforced in perturbed form, i.e., with complementarity relaxed along the central path as

$$\mathbf{S}\boldsymbol{\nu} = \mu \mathbf{1}, \qquad \mathbf{S} \triangleq \operatorname{diag}(\mathbf{s}), \quad \mathbf{s} > \mathbf{0}, \; \boldsymbol{\nu} > \mathbf{0}. \tag{4.38}$$

At each iteration, IPOPT computes a Newton step for the resulting primal-dual system by solving a sparse linear system whose blocks are built from the constraint Jacobians and the Hessian of the Lagrangian [59]. In this transcription, the time-local coupling of the defects yields structured sparsity, so sparse factorization constitutes the computational core of the solve.

Global convergence is promoted via a filter line-search that accepts trial iterates based on sufficient improvement in constraint violation and/or objective value, avoiding the need for a single penalty merit scalarization [59]. When progress stalls, IPOPT may enter a feasibility restoration phase to reduce constraint violation before resuming the main iterations. As the iterations progress, μ is reduced and the iterates track the central path toward a local KKT point of the original NLP.

In a receding-horizon setting, consecutive NLPs are initialized from the shifted previous solution and, when available, from the associated multipliers. The rotational coefficients $\mathbf{p}_{\mathrm{rot}}$ are refreshed online at each MPC update and then kept fixed over the subsequent solve.

Problem generation and derivatives via CasADi. The NLP functions in Equation (4.35) are implemented in CasADi, which provides symbolic graph construction and algorithmic differentiation for first and second order derivatives [60]. This enables consistent and efficient evaluation of gradients, Jacobians, and Hessian information required by IPOPT, while retaining the sparsity patterns induced by the horizon structure. This is particularly convenient because the translational predictor is LTI and can be discretized once, whereas the LTV rotational predictor is injected through the coefficients vector $\mathbf{p}_{\mathrm{rot}}$.

4.4.2. Online MPC update routine

To ensure reliable real-time execution in a receding-horizon setting, each MPC update requires a consistent initialization of both the decision variables and the time-varying rotational predictor. This is achieved through a lightweight online routine that reuses information from the previous solution and refreshes the model parameters at the current state. At time step k, the NLP is initialized from the previous optimal solution by **shifting** the primal trajectories forward by one node. In particular, the state and input sequences are used to build the initial guess

$$\mathbf{X}_k^{(0)} = \left[\mathbf{x}^\star_{1|k-1}, \ldots, \mathbf{x}^\star_{N|k-1}, \mathbf{x}^\star_{N|k-1}\right], \qquad \mathbf{U}_k^{(0)} = \left[\mathbf{u}^\star_{1|k-1}, \ldots, \mathbf{u}^\star_{N-1|k-1}, \mathbf{u}^\star_{N-1|k-1}\right] \quad (4.39)$$

where the last node and input are replicated to preserve dimensional consistency. The Lagrange multipliers are initialized from the previous solution as well, which improves robustness and typically reduces the number of interior-point iterations. The shift alone does not guarantee dynamic consistency, because the horizon starts from the updated estimate $\hat{\mathbf{x}}_k$ while the remaining nodes are inherited from the previous solution. The shifted inputs are thus propagated through the discrete-time predictors starting from $\hat{\mathbf{x}}_k$, yielding a reference rollout. The resulting reference sequence provides a dynamically consistent initialization and serves as the linearization trajectory for refreshing the time-varying rotational model over the horizon.

The rotational predictor is updated online through a successive linearization strategy. At each MPC update, the affine LTV model is reconstructed over the whole horizon by linearizing the rotational dynamics along the reference trajectory to obtain the coefficients $\{\mathbf{A}_{r,j}, \mathbf{B}_{r,j}, \mathbf{f}_{r,j}\}_{j=0}^{N-1}$. These coefficients are computed once prior to the NLP call and then kept frozen per-solve. This choice keeps the prediction constraints linear in the decision variables during the optimization, while still capturing the local time variation of the rotational dynamics through coefficients that are refreshed at the next sampling instant from the updated reference.

At each step the rotational coefficients are packed into a single vector $\mathbf{p}_{\text{rot}}$. This strategy keeps the NLP structure unchanged across MPC iterations: the translational predictor remains precomputed, whereas only the rotational blocks are refreshed online through $\mathbf{p}_{\text{rot}}$.

Finally, a few practical enforcement details ensure consistency across consecutive MPC updates. When limits are imposed on input variations ($\Delta\mathbf{a}$, $\Delta\boldsymbol{\tau}$), the variation at the first node must be referenced to the previously applied command; accordingly, the last applied input $\mathbf{u}_{k-1}$ is provided to the NLP and used to define the initial variation. In the implemented transcription, hard constraints are enforced at the transcription nodes and not explicitly within each sampling interval. In this close-proximity imaging regime, the guidance objective is to perform smooth fly-around and viewpoint changes rather than highly critical terminal operations such as docking or landing. The selected sampling time is sufficiently small to limit inter-sample excursions, so violations between nodes are unlikely and can be monitored a posteriori by trajectory interpolation when needed.

For implementation convenience, these online updated quantities are collected into the NLP parameter vector

$$\mathbf{p} = \begin{bmatrix} \hat{\mathbf{x}}_k^\top & \hat{\mathbf{q}}_k^\top & \mathbf{u}_{k-1}^\top & \mathbf{p}_{\text{rot}}^\top \end{bmatrix}^\top.$$

4.4.3. Replanning triggers and termination logic

Given the exploratory nature of the mapping task, dedicated logics are introduced to decide when to request a new mapping direction from the high-level planner and when to terminate the operation based on the current coverage state.

Replanning logic

Replanning relies on supervisory triggers that update the mapping direction $\hat{\mathbf{q}}_k$ whenever the current direction is no longer expected to provide meaningful additional information. The trigger logic aims to avoid spending control effort in configurations that mostly refine already seen regions or generate redundant views, which may still increase the instantaneous coverage metric but provide limited benefit to the underlying 3D reconstruction, where viewpoint diversity is critical.

A progress based trigger monitors whether the current direction is still producing a meaningful mapping gain and activates replanning when the improvement has effectively saturated. The metric is built consistently with the coverage memory definition in Section 4.2.2. Recalling $\mathbf{C}_k \in \mathbb{R}^{N_d}_{\geq 0}$ and the target coverage C_{tgt}, the average completion score is defined as the capped mean coverage level,

$$\bar{C}_k \triangleq \frac{1}{N_d} \sum_{i=1}^{N_d} \min\big(C_{k,i},\, C_{\mathrm{tgt}}\big), \tag{4.40}$$

so that zones that have already reached the target level do not artificially inflate the progress indicator. When a new exploration direction is selected at time t_{k_0}, the initial value $\bar{C}_{\mathrm{in}} \triangleq \bar{C}_{k_0}$ is stored and compared against the average reference level $\bar{C}_{\mathrm{max}}$, computed from the planner predicted coverage $\mathbf{C}_{\mathrm{pred}}$ via Equation (4.40). The latter represents the expected global completion score attainable by following the current planned direction until the next replanning event.

As the maneuver unfolds, the realized progress is measured through the normalized ratio

$$\eta_k \triangleq \frac{\bar{C}_k - \bar{C}_{\mathrm{in}}}{\big(\bar{C}_{\mathrm{max}} - \bar{C}_{\mathrm{in}}\big) + \varepsilon}, \tag{4.41}$$

with a small $\varepsilon > 0$ to handle the degenerate case $\bar{C}_{\mathrm{max}} \approx \bar{C}_{\mathrm{in}}$. The quantity $\eta_k \in [0, 1]$ estimates how much of the expected global gain associated with the current direction has already been achieved. Replanning is enabled only when η_k exceeds a prescribed threshold.

A geometry based trigger detects when the spacecraft has effectively reached the current direction command, after which the closed-loop motion may devolve into local circling around a similar viewpoint. This trigger is introduced specifically to prevent prolonged motion around the same region. Although such behavior can still increase the coverage score, it tends to produce images from similar viewpoints and therefore offers limited additional value for reconstruction quality. The direction is considered reached when an angular alignment and a position tolerance condition are satisfied, namely

$$\frac{\hat{\mathbf{q}}_k^\top \boldsymbol{\rho}_k}{\|\boldsymbol{\rho}_k\|_2} \;\geq\; \cos\big(\varepsilon_\theta\big), \qquad \big\|\boldsymbol{\rho}_k - r_{\text{opt}}\hat{\mathbf{q}}_k\big\|_2 \;\leq\; \varepsilon_r, \tag{4.42}$$

where ε_θ and ε_r are an angular and a radial tolerance, respectively.

When one of these conditions is met, the planner is called to select a new direction, promoting viewpoint diversity and a more uniform distribution of observations over the target surface.

Termination logic

The mapping task terminates when the global completion indicator derived from the coverage memory reaches a prescribed acceptance level. Consistently with Equation (4.40), the global completion score is taken as the capped mean coverage level $\bar{C}_k$. Termination is declared when

$$\bar{C}_k \;\geq\; \alpha\, C_{\text{tgt}}, \tag{4.43}$$

with $\alpha \in (0, 1)$ a high acceptance ratio. Selecting α close to one targets near complete coverage while avoiding a disproportionate final stage cost. Driving $\bar{C}_k$ all the way to C_{tgt} may require increasingly expensive maneuvers to access the last uncovered regions. The acceptance ratio α therefore provides a practical stopping criterion that balances reconstruction quality against operational effort in closed loop.

Upon satisfying Equation (4.43), the mapping phase is considered complete, but the closed-loop execution is not immediately stopped. Empirical evidence from the reconstruction pipeline indicates that reconstruction quality benefits when the final images are acquired from a viewpoint close to the one associated with the initial position, which improves viewpoint continuity across the dataset. Accordingly, once the completion threshold is reached, the planner switches the reference direction to the direction associated with the initial position and commands a return leg toward the initial configuration. The motion is then stopped using the same geometry based trigger introduced earlier, and mission completion is declared when the return condition is met.

4.5. High-fidelity propagation

This section describes the high-fidelity simulator adopted for closed-loop simulations. This framework is used to quantify the mismatch between the MPC predictive model and the nonlinear dynamics, and to evaluate the performance impact of the approximations required for real-time optimization. It also supports robustness analyses by introducing unmodeled perturbations and parameter mismatches in the simulated dynamics. The simulation is performed in the inertial frame $\mathcal{E}$, whereas the MPC operates on a relative state expressed in the target LVLH frame $\mathcal{L}$.

Over each sampling interval $[t_k, t_{k+1})$, the control action returned by the NLP is held constant. The MPC variables are mapped back to physical units through the scaling introduced in Section 3.1.1. For notational simplicity, the same symbols are retained throughout this section, and all quantities are understood to be dimensional whenever referring to the high-fidelity simulation.

The target translational motion is propagated in ECI according to the perturbed two-body model in Equation (2.2), retaining the J_2 term as the dominant disturbance. This yields the target inertial trajectory $(\mathbf{r}_T(t), \mathbf{v}_T(t))$ over $[t_k, t_{k+1})$. The chaser is then propagated with the full 6-DoF nonlinear equations of motion driven by the control inputs $\{\mathbf{a}_{\mathcal{L},k}, \boldsymbol{\tau}_{\mathcal{B},k}\}$. Both the target and chaser equations of motion are integrated numerically over $[t_k, t_{k+1})$ using a variable-step scheme with error control.

At the update instant t_{k+1}, the LVLH frame $\mathcal{L}_{k+1}$ is reconstructed from the inertial position and velocity of the propagated target. In particular, the radial axis is aligned with $\mathbf{r}_{T,k+1}$, the cross-track axis with $\mathbf{r}_{T,k+1} \times \mathbf{v}_{T,k+1}$, and the along-track axis completes the right-handed triad. This construction provides the direction cosine matrix $\mathbf{C}_{\mathcal{L}/\mathcal{E}}(t_{k+1})$ and the associated frame angular rate $\boldsymbol{\omega}^{\mathcal{L}}_{\mathcal{L}/\mathcal{E}}(t_{k+1})$.

The relative translational state used by the MPC is retrieved by projecting the inertial chaser target differences into LVLH. Relative position is

$$\boldsymbol{\rho}_{k+1} = \mathbf{C}_{\mathcal{L}/\mathcal{E}}(t_{k+1})\big(\mathbf{r}_{C,k+1} - \mathbf{r}_{T,k+1}\big), \tag{4.44}$$

and the relative velocity is reconstructed using the instantaneous LVLH frame kinematics,

$$\mathbf{v}_{k+1} = \mathbf{C}_{\mathcal{L}/\mathcal{E}}(t_{k+1})\big(\mathbf{v}_{C,k+1} - \mathbf{v}_{T,k+1}\big) - \boldsymbol{\omega}^{\mathcal{L}}_{\mathcal{L}/\mathcal{E}}(t_{k+1}) \times \boldsymbol{\rho}_{k+1}. \tag{4.45}$$

Attitude is propagated in the simulator using $\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{L}}$ together with the absolute angular velocity $\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}/\mathcal{E}}$, obtained by composing the LVLH relative body rate with the instantaneous LVLH angular rate. At the beginning of the interval, this composition is evaluated using $\boldsymbol{\omega}^{\mathcal{L}}_{\mathcal{L}/\mathcal{E}}(t_k)$,

$$\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}/\mathcal{E}}(t_k) = \boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}/\mathcal{L}}(t_k) + \mathbf{C}_{\mathcal{B}/\mathcal{L}}\big(\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{L}}(t_k)\big)\ \boldsymbol{\omega}^{\mathcal{L}}_{\mathcal{L}/\mathcal{E}}(t_k). \tag{4.46}$$

After propagation, the LVLH relative rate required by the MPC is recovered at t_{k+1} by inverting Equation (4.46) using the LVLH angular rate reconstructed from the target inertial motion $\boldsymbol{\omega}^{\mathcal{L}}_{\mathcal{L}/\mathcal{E}}(t_{k+1})$.

Finally, the reconstructed LVLH state is converted back to the nondimensional units adopted by the MPC. As a result, the subsequent MPC update receives a state that is consistent with the inertial propagation and expressed in the same variables assumed by the predictive model.

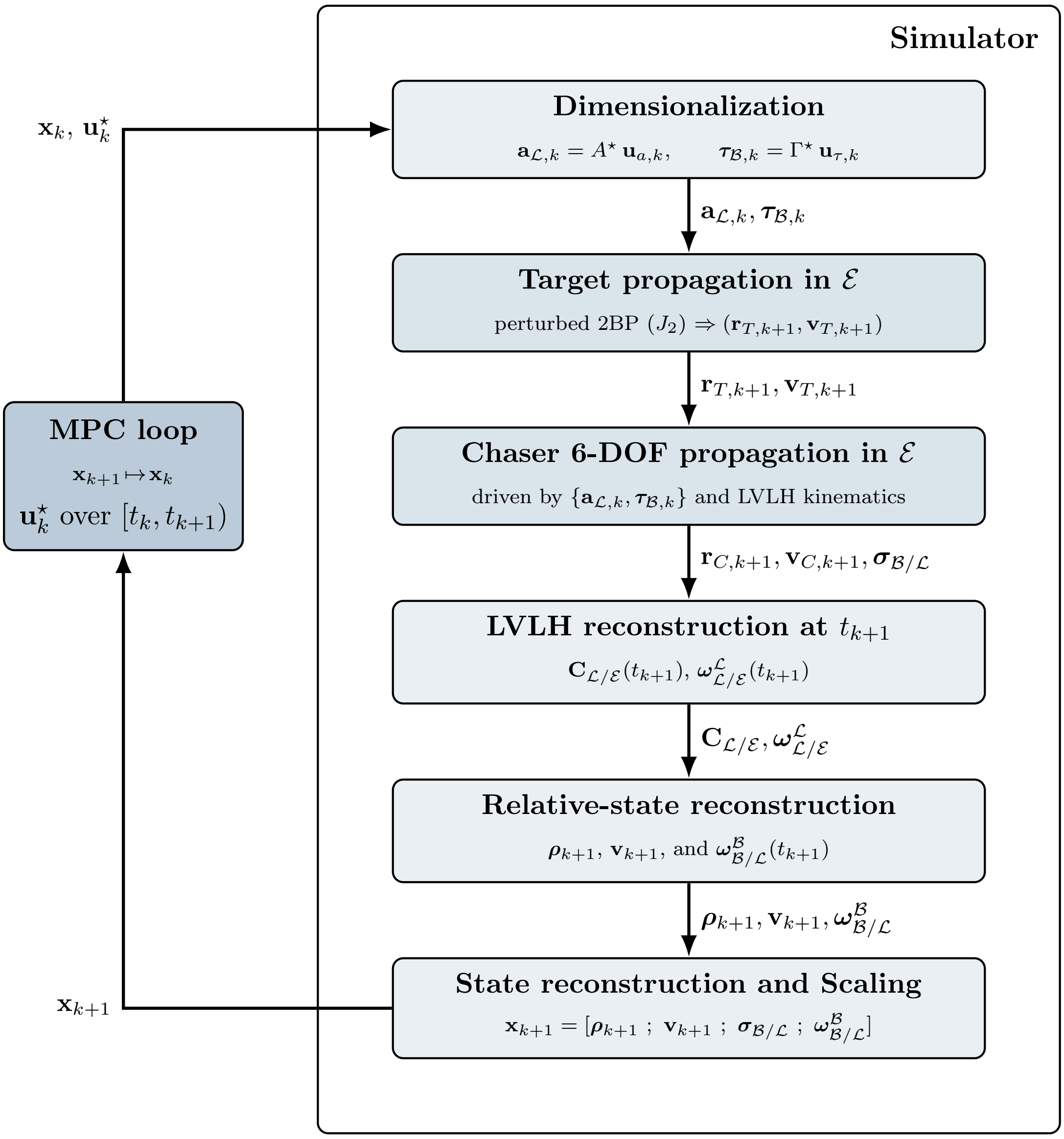


Figure 4.5: High-fidelity propagation strategy

Although the MPC operates on the same feedback variables returned by the simulator, the predictive model embedded in the optimization and the closed-loop propagation are

Component	MPC predictor	High-fidelity simulator
Translation	CWH in LVLH	Inertial propagation in ECI
LVLH	Constant angular rate	Constructed from the target orbit
Rotation	LTV on a moving reference	Nonlinear attitude dynamics
Perturbations	Neglected	J_2, $\boldsymbol{\tau}_{\mathrm{gg}}$
Numerical integration	Precomputed STM	Variable-step integration

Table 4.1: Comparison between the MPC predictor and the high-fidelity simulation adopted for closed-loop validation.

not identical. This mismatch is expected by design and does not compromise the approach, provided that the feedback state is reconstructed consistently at the sampling instants and that safety-related constraints are satisfied. The dominant contributions are summarized in Section 4.5 and mainly arise from the translational mismatch between the CWH predictor and the inertial propagation used in simulation, from the rotational approximation of the affine LTV model against full nonlinear 6-DoF dynamics, and from the different constraint handling, which is enforced at transcription nodes in the NLP but evolves continuously within each sampling interval in the plant. These discrepancies are mitigated in closed loop by the receding-horizon updates and the episodic replanning of the imaging direction, while inter-sample safety can be checked a posteriori when needed.

5 | Results

This chapter reports the simulation results of the proposed imaging-driven guidance architecture. The discussion focuses on whether the closed-loop behavior achieves complete and uniform coverage while remaining operationally feasible, respecting safety and actuation limits and maintaining a bounded control effort. The presentation follows a verification oriented order. First, the relative motion and the control inputs are analyzed to confirm feasibility, constraint compliance and realistic actuator usage. The imaging conditions are then assessed through range regulation and boresight-to-LOS alignment, which define the regime of usable images for reconstruction. Mapping performance is subsequently quantified by tracking coverage evolution and by summarizing the final distribution across the discretized viewing zones. The replanning timeline and the associated computational performance are then reported, with emphasis on online solvability at the adopted sampling rate.

5.1. Simulation Setup

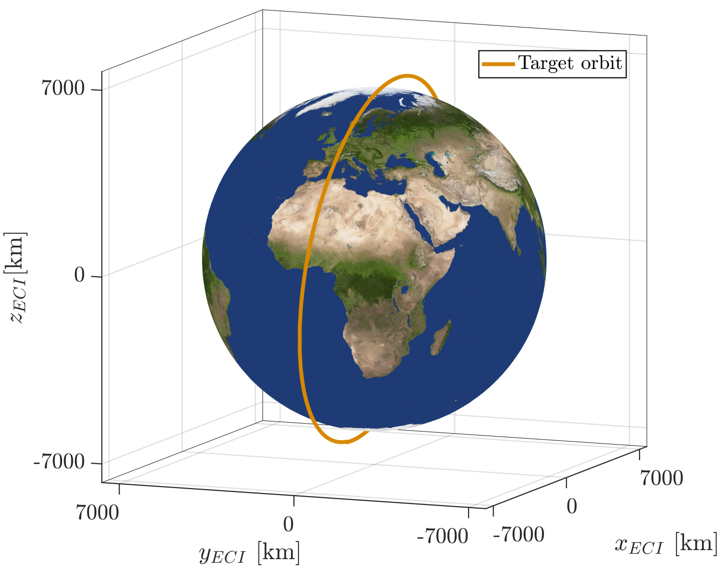
The following results adopt the notation, units, and reference frames introduced in Chapter 2. All quantities are reported in physical units, applying consistently the scaling defined in Section 3.1.1.

Relative motion states and translational control actions are expressed in the target-centered LVLH frame $\mathcal{L}$, whereas inertial trajectories are reported in the ECI frame $\mathcal{E}$. Attitude and torque quantities are expressed in the chaser body frame $\mathcal{B}$. When imaging conditions are evaluated, the camera boresight and the LOS vectors are mapped to $\mathcal{L}$.

5.1.1. Scenario definition and initialization

The numerical results are presented for a representative LEO test case, obtained by specializing the problem formulation with the scenario settings summarized below. The target is modeled as a generic non-cooperative spacecraft moving on a known reference orbit at altitude $h = 550$ km. This reference motion is used to define the target-centered

LVLH frame and is propagated in the inertial simulation environment to generate the target state history employed in closed loop. The orbit is not selected to reproduce a specific mission, but to represent a realistic LEO operating condition while remaining consistent with the modeling assumptions adopted in Chapter 2. For the same reason, a Sun-synchronous orbit is considered, with inclination $i_t = 97.5°$, which is representative of common Earth-observation operational regimes and therefore supports the relevance of the test case. As a result, the trends highlighted by the numerical implementation are not tied to a particular set of orbital elements, as long as the relative-motion regime remains within the validity range of the adopted LVLH based predictive model.



(a) Orbit and initial position.

Semi-major axis	a_t	6928 km
Eccentricity	e_t	10^{-4}
Inclination	i_t	97.5°
RAAN	Ω_t	0°
Argument of perigee	ω_t	0°
True anomaly at t_0	$\vartheta_{t,0}$	0°

(b) Orbital parameters.

Figure 5.1: Target reference orbit.

A near-circular orbit is considered to remain consistent with the assumptions underlying the CWH equations. A small but nonzero eccentricity ($e_t = 10^{-4}$) is retained to better reflect realistic LEO conditions, while remaining sufficiently low not to affect the LVLH relative dynamics in the operating regime considered. The remaining angular elements are set to zero for convenience, since for an almost circular orbit they mainly define an inertial orientation of the trajectory and have negligible influence on the qualitative behavior of the relative motion and on the imaging-related performance discussed next.

With the target reference orbit defined and the associated LVLH frame $\mathcal{L}$ established, the closed-loop simulations are initialized by prescribing the chaser state in relative coordinates. Initial translational conditions are assigned in $\mathcal{L}$ through the relative position $\boldsymbol{\rho}_0$

and velocity $\mathbf{v}_0$, while the initial attitude $\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{L},0}$ and angular velocity $\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}/\mathcal{L},0}$ define the rotational configuration. The corresponding numerical values are reported in Table 5.1 and are selected to represent a realistic post-detection condition for an non-cooperative target. The relative position is placed in the target orbital plane, which is consistent with a prior rendezvous phase that inserts the vehicle in the vicinity of the target before the imaging-driven guidance begins. This assumption provides a clean and representative starting configuration without restricting the generality of the conclusions, since the subsequent closed-loop behavior is governed by the same imaging and safety constraints in three dimensions.

The initial target-chaser relative distance is again chosen to be consistent with a preceding target identification phase, which is typically conducted at a larger stand-off distance than the nominal imaging distance, leaving sufficient margin for the guidance to converge toward the desired range corridor. The initial relative velocity is set to $\mathbf{v}_0 = \mathbf{0}$, so that the ensuing motion is driven by the closed-loop guidance action rather than by a prescribed initial drift. Finally, the initial attitude and angular rate are selected coherently with the assumed outcome of the preliminary inspection, by setting $\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}/\mathcal{L},0} = \mathbf{0}$ and choosing $\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{L},0}$ such that the target is immediately observable within the admissible camera viewing region given the camera mounting in $\mathcal{B}$ and the initial LOS defined by $\boldsymbol{\rho}_0$.

Initial conditions			
Relative position	$\boldsymbol{\rho}_0$	$[0,\ -46,\ 0]$	m
Relative velocity	$\mathbf{v}_0$	$[0,\ 0,\ 0]$	$\mathrm{m\,s^{-1}}$
Angular velocity	$\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}/\mathcal{L},0}$	$[0,\ 0,\ 0]$	$\mathrm{rad\,s^{-1}}$
MRP attitude	$\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{L},0}$	$[0,\ 0,\ -0.414]$	–

Table 5.1: Chaser initial relative state in $\mathcal{L}$ and attitude in $\mathcal{B}$.

The results in this chapter are obtained for a parameter set selected to represent a generic yet realistic close-proximity imaging campaign. The settings are chosen so that the guidance remains online-solvable and the commanded motion is compatible with representative translational and attitude control authority. For completeness, the adopted numerical values are reported in Tables 5.2 and 5.3. Their selection is driven by three coupled requirements: satisfying the admissible imaging geometry prescribed by the reconstruction pipeline, preserving conservative operational safety margins and keeping the computational load within a regime consistent with receding-horizon execution.

From an imaging standpoint, the camera parameters set the admissible field of view. The adopted FoV is representative of NeRF-based close-range space imagery and provides a practical balance between keeping the target sufficiently in-frame and allowing viewpoint changes without excessive pointing effort. The corresponding camera parameters are summarized in Table 5.2. Moreover, the relative motion is constrained to remain within the viewing corridor introduced in Chapter 3, namely within the range bounds $[r_{\min}, r_{\max}]$ while promoting the nominal standoff distance r_{opt} to stabilize apparent scale and inter-image overlap. The values of $r_{\min}$, $r_{\max}$, and r_{opt} reported in Table 5.3 follow the imaging-driven design rule in Equation (3.20) from an estimated target size. Target is enclosed within a bounding sphere of radius $R_{\text{tgt}} = 4$ m, which provides a geometry agnostic reference to set feasible viewing distances. Safety is further enforced through the Keep-Out Zone radius r_{KOZ} (also in Table 5.3), which guarantees a minimum separation from the target and accommodates residual uncertainty on its true shape and extent.

Camera parameters			
Image width	W	2048	px
Image height	H	2048	px
Focal length (horizontal)	f_x	5000	px
Focal length (vertical)	f_y	5000	px
Horizontal FoV	$\theta_{\text{FoV},h}$	23.15	deg
Vertical FoV	$\theta_{\text{FoV},v}$	23.15	deg

Table 5.2: Camera parameters used for imaging constraints and coverage evaluation.

Range and safety			
Minimum range	$r_{\min}$	30	m
Nominal standoff distance	r_{opt}	40	m
Maximum range	$r_{\max}$	50	m
Keep-out zone radius	r_{KOZ}	5	m

Table 5.3: Range and safety parameters.

The assumed actuation bounds pursue the same objective of realism without over-specializing the test case. Translational authority is limited through $a_{\max}$, while attitude control is limited in through $\tau_{\max}$, representing finite propulsion capability and reaction wheel or thruster torque authority. The adopted values, reported in Table 5.4, are selected to be representative of small spacecraft inspection scenarios and to remain compatible with online guidance generation. In addition to actuator saturation, the optimization enforces operational bounds on the relative speed and body rate, $v_{\max}$ and $\omega_{\max}$, which provide conservative safety margins and support imaging performance by avoiding overly fast translational or rotational motion that would produce poor quality images or challenge tracking. Step-to-step slew-rate limits $\Delta a_{\max}$ and $\Delta \tau_{\max}$ further prevent unrealistically abrupt command switching and act as a compact proxy for actuator bandwidth and implementation smoothness.

Actuation and rate limits			
Maximum translational acceleration	$a_{\max}$	$8 \cdot 10^{-4}$	$\mathrm{m\,s^{-2}}$
Maximum control torque	$\tau_{\max}$	$2 \cdot 10^{-3}$	$\mathrm{N\,m}$
Maximum relative speed bound	$v_{\max}$	0.1	$\mathrm{m\,s^{-1}}$
Maximum body rate bound	$\omega_{\max}$	1	$\mathrm{deg\,s^{-1}}$
Slew-rate limits			
Acceleration slew-rate bound	$\Delta a_{\max}$	$\frac{1}{3} a_{\max}$	$\mathrm{m\,s^{-2}}$
Torque slew-rate bound	$\Delta \tau_{\max}$	$\frac{1}{3} \tau_{\max}$	$\mathrm{N\,m}$

Table 5.4: Actuation and rate limits used in the simulations.

The MPC is implemented in discrete time on the sampling grid $\{t_k\}$, with sampling time T_s, and each update solves a finite-horizon problem of length N. Since the objective is sustained close-proximity imaging rather than terminal contact operations, the guidance does not require extremely high update rates. The selected values $T_s = 5$ s and $N = 200$ therefore provide a pragmatic compromise: sampling time is short enough for node-wise constraint enforcement to remain representative of the continuous time behavior in this regime, while the horizon length and the sample time retain sufficient computational margin for repeated online optimization. In an operational architecture, T_s can be interpreted as the frequency of the guidance layer, while navigation and low level attitude/actuator loops operate at higher rates, consistently with standard spacecraft Guidance, Navigation, and Control (GNC) practice.

Termination follows the supervisory logic introduced in Section 4.4.3. In the nominal configuration, the run stops when the progress indicator satisfies $\bar{C}_k \geq \alpha\, C_{\mathrm{tgt}}$, while a maximum simulation time provides a conservative fallback. The adopted acceptance ratio is set to $\alpha = 0.98$, which is intentionally close to unity to enforce near-complete coverage, yet it avoids unnecessary prolongation when residual deficits become marginal and progress may slow down due to episodic replanning and geometric constraints.

Simulation and mapping setup			
Sampling time	T_s	5	s
Prediction horizon	N	200	–
Number of mapping directions	N_d	16	–
Target coverage level	C_{tgt}	10	min
Acceptance ratio	α	0.98	–

Table 5.5: Simulation settings and mapping parameters.

The mapping space is discretized using $N_d = 16$ directions, providing a meaningful partition of the viewing domain given the assumed target size and the adopted FoV. This choice can be tuned depending on mission needs and sensor characteristics. Increasing N_d refines the angular resolution but also increases the burden on the supervisory logic and complicates convergence assessment, whereas too few directions may be overly coarse, promoting redundant image sets and degrading reconstruction quality. The selected value offers a practical balance between angular diversity and interpretable progress tracking. Consistently, the target per-direction accumulation level is set to $C_{\text{tgt}} = 10$ min, which provides sufficient imaging time to support viewpoint diversity within each sector while keeping the overall campaign duration bounded.

5.2. Guidance and Control Performance

This section presents the closed-loop guidance and control results obtained under the scenario and numerical settings summarized in Section 5.1. The analysis verifies that the resulting close-proximity motion remains consistent with the adopted relative dynamics framework, respects the operational bounds introduced in the problem formulation and generates well-behaved control actions compatible with the assumed actuation limits. The main replanning outcomes are reported to clarify how the supervisory logic updates the exploration direction over the run and how these decisions relate to the executed trajectory.

5.2.1. Relative trajectory in LVLH

Figure 5.2 reports a representative relative trajectory obtained from the simulation, expressed in the target-centered LVLH frame $\mathcal{L}$. The path exhibits a wide angular distribution around the target, indicating a good diversity of viewing directions and limited repeated passes over the same regions. This qualitative behavior is consistent with an imaging-driven exploration, where the motion is shaped to uniformly distribute viewpoints around the target. In the present simulation, the corresponding mapping campaign reaches completion in 4.36 hours, providing a compact summary of the time required to satisfy the prescribed coverage objective under the adopted constraints.

The trajectory also reflects the expected end-of-mission behavior discussed in Section 4.4.3. Once the completion condition is satisfied, the motion transitions toward a final viewpoint close to the one associated with the initial configuration. This return leg is visible in the terminal portion of the trajectory and is included to improve viewpoint continuity in the resulting image set.

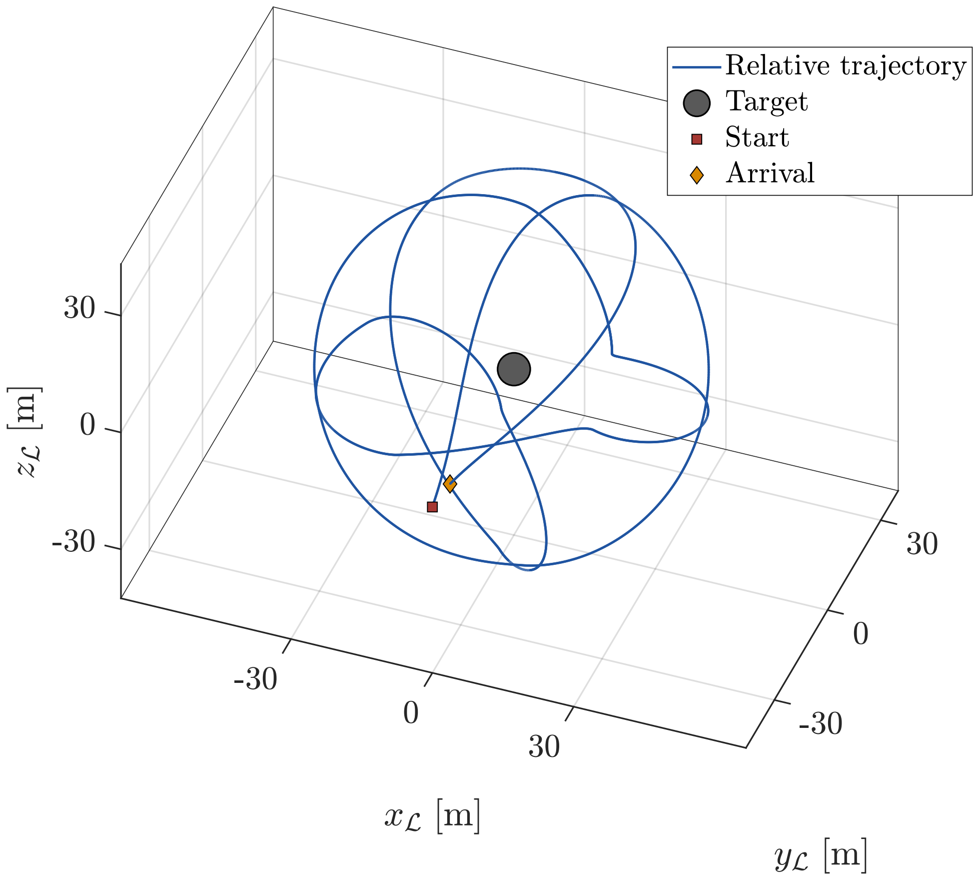
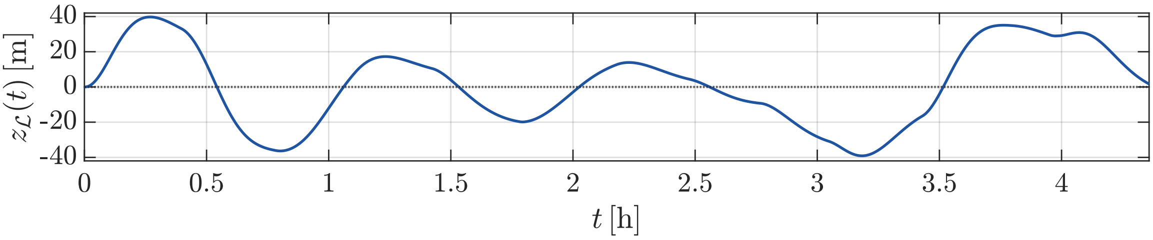


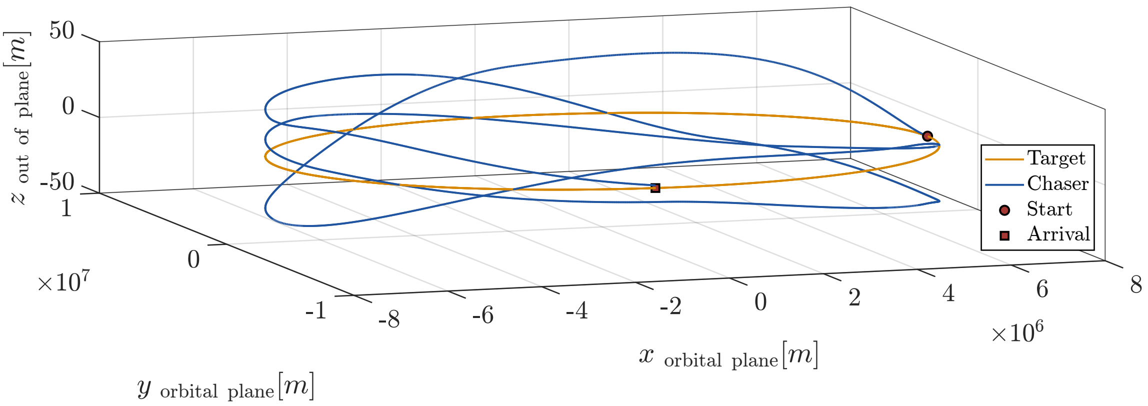
Figure 5.2: Relative trajectory in the target LVLH frame $\mathcal{L}$.

A further aspect of interest is the out-of-plane motion along the LVLH cross-track axis, which entails plane change components and is therefore inherently more demanding in terms of control effort, with a direct impact on propellant consumption. In the present run, the out-of-plane dynamics remain contained, as illustrated by the $z_{\mathcal{L}}(t)$ trace in Figure 5.3a. At the same time, a strictly planar inspection is not generally sufficient to ensure full observability of the target surface when target rotation is limited or not exploited to provide passive viewpoint diversity. In such cases, a cross-track component becomes necessary to acquire complementary perspectives.

The view in Figure 5.3b indicates that out-of-plane maneuvers are introduced primarily to depart from the orbital plane and then remain bounded. After an initial transition, the trajectory tends to settle around a slightly inclined inspection plane relative to the target orbit, reducing the need for recurrent cross-track corrections while still enabling the viewpoint variation required for mapping.



(a) Out-of-plane component $z_{\mathcal{L}}$ evolution.



(b) Out-of-plane component $z_{\mathcal{L}}$ with respect to the target orbital plane.

Figure 5.3: Out-of-plane component $z_{\mathcal{L}}$.

5.2.2. Range and pointing behavior

The following results focus on the imaging conditions discussed in Section 3.2.1. Since range and boresight-to-LOS alignment are primarily enforced through strong objective terms, rather than as hard constraints, their evolution is reported as evidence of compliance with the prescribed imaging requirements.

Figure 5.4 reports the relative range profile $r_k = \|\boldsymbol{\rho}_k\|$, highlighting the admissible shell bounds and the nominal standoff distance. After an initial transient, during which the chaser approaches from the injection conditions, the range settles within the prescribed corridor and remains close to r_{opt} throughout the run. The residual oscillations around the nominal value are consistent with the intended operating regime: since r_{opt} is promoted rather than imposed as a hard constraint, small deviations can be exploited to accommodate viewpoint changes and reorientation phases without over constraining the motion. This behavior preserves an admissible imaging geometry while avoiding unnecessary control effort devoted to strict range regulation.

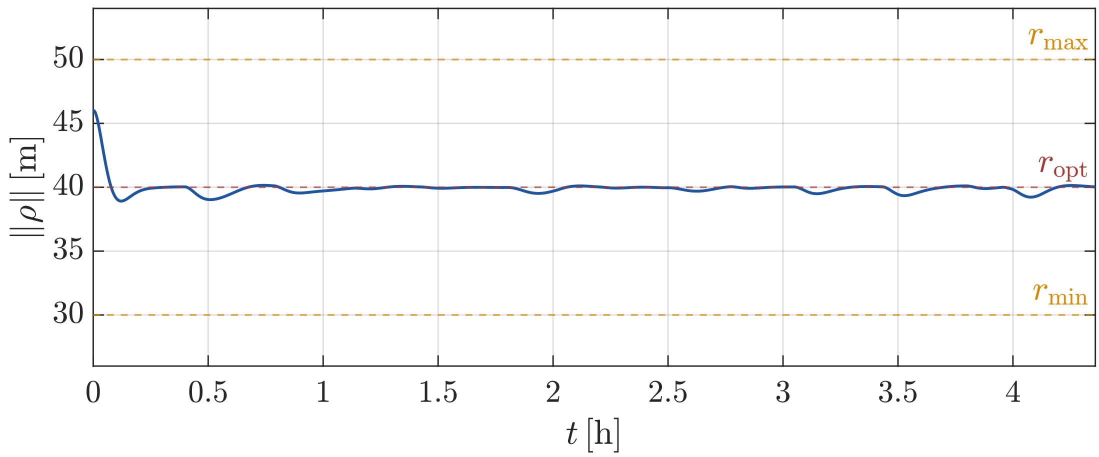


Figure 5.4: Relative range $r_k = \|\boldsymbol{\rho}_k\|$ over time.

Pointing performance is summarized through the off-boresight angle between the camera boresight $\mathbf{b}_{\mathcal{L},k}$ and the line-of-sight direction $\mathbf{d}_{\mathcal{L},k}$. As shown in Figure 5.5, the attitude guidance maintains a tight alignment with the LOS, remaining below the FoV half-angle threshold throughout the trajectory. Additionally, the off-boresight angle stays close to zero for most of the run, indicating that the target is consistently maintained near the center of the camera cone. This behavior matches the intended imaging regime of the reconstruction pipeline, as it provides stable framing and limits the occurrence of low-quality views associated with large misalignment. Residual deviations are confined to short transient phases, typically during more aggressive reorientation and repositioning maneuvers, which are often triggered by exploration updates following replanning events, but can also occur when the guidance accommodates local geometric transitions.

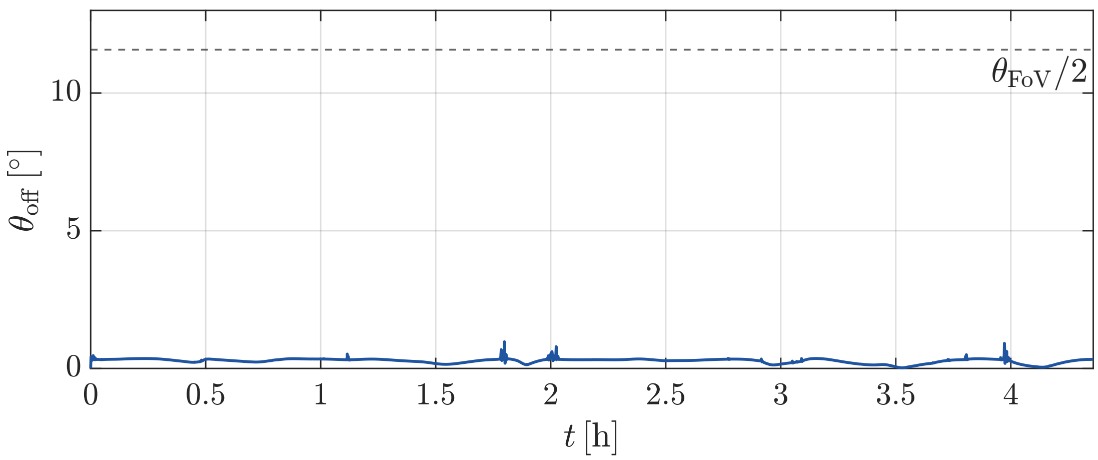
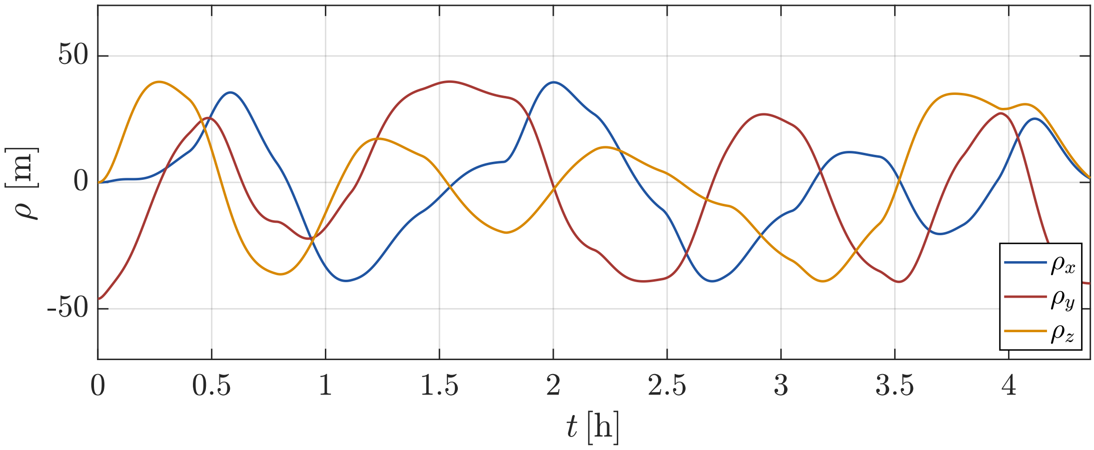


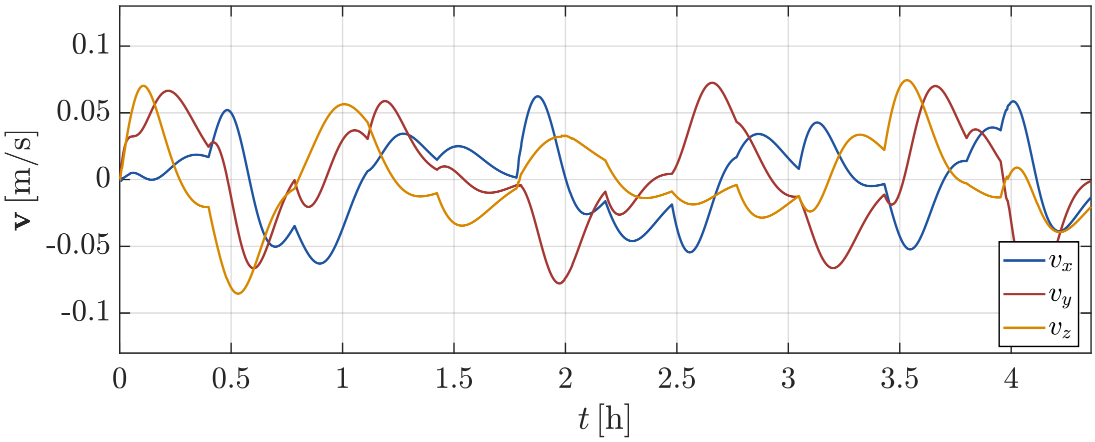
Figure 5.5: Off-boresight angle $cos^{-1}(\varphi_k)$ over time.

5.2.3. State evolution and dynamic consistency

Beyond geometric feasibility, it is important to confirm that the resulting motion remains within the intended close-proximity operating regime. This is characterized by the translational velocity and the body angular velocity with respect to $\mathcal{L}$, whose profiles capture the typical rates required to sustain the imaging-driven conditions. Alongside this, the full relative position in $\mathcal{L}$ is reported to provide a complete view of how the motion evolves over time. Bounded and well-conditioned velocity and rate traces support both dynamic consistency of the relative dynamics and imaging usability: excessively large translational or rotational rates would increase control effort and, more critically, would lead to degraded image quality (e.g., motion blur and reduced framing stability), undermining the reconstruction process.



(a) Relative trajectory $\boldsymbol{\rho}$.



(b) Relative trajectory $\mathbf{v}$.

Figure 5.6: Chaser relative motion in $\mathcal{L}$.

As shown in Figure 5.6, the translational velocity remains within the prescribed bounds throughout the run and varies smoothly, without sharp peaks that would be symptomatic of aggressive repositioning. The resulting profile is consistent with an imaging-driven close-proximity regime, where viewpoint changes are achieved through moderate relative speeds, limiting apparent target motion across frames and supporting stable image acquisition while respecting the imposed state and input constraints.

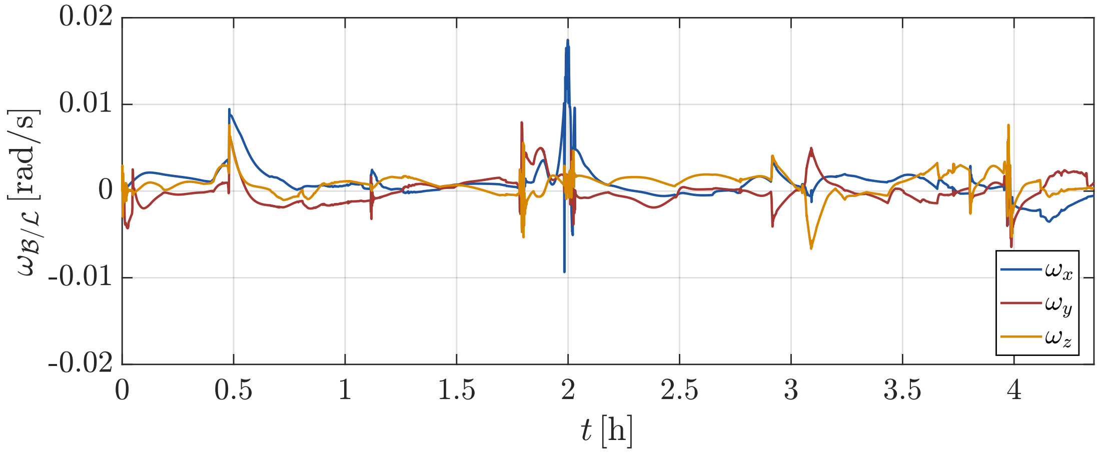


Figure 5.7: Angular velocity components $\boldsymbol{\omega}_{\mathcal{B}/\mathcal{L},k}$.

The angular-rate evolution in Figure 5.7 follows the same trend. Rates remain within the prescribed bounds for the entire duration and stay close to zero during steady imaging phases, confirming that the attitude is maintained tightly aligned with the viewing geometry. Brief spikes occur during more demanding reorientation intervals, for example when the guidance reacts to an updated exploration direction selected by the high-level planner. These transients are bounded and confined in time, indicating that viewpoint transitions are executed without sustained high motion rates.

5.2.4. Control effort and actuator feasibility

The feasibility of the closed-loop solution is further supported by the commanded input profiles. Figures 5.8 and 5.9 report the translational control acceleration in and the control torque, respectively. In both cases, the commands remain within the prescribed bounds and evolve smoothly over time, with no evidence of high-frequency chattering. Occasional localized peaks appear in the torque profile and, more mildly, in the translational acceleration, but they remain limited in magnitude and duration. Overall, the control effort is distributed over multiple sampling intervals rather than concentrated into impulsive corrections, consistent with the assumed actuation model.

A clear correspondence can be observed between the largest input magnitudes and the episodes in which the exploration direction is updated by the high-level planner. During these replanning events, the control inputs accommodate changes in the desired viewing geometry, leading to brief increases in actuation activity while preserving feasibility.

The attitude control shows occasional peaks in $\boldsymbol{\tau}_{\mathcal{B},k}$ when faster reorientation is required to maintain tight LOS alignment while redirecting the relative motion toward a new viewing sector. These peaks remain comfortably within the limits reported in Table 5.4 and are more apparent because the nominal torque demand is otherwise low. This behavior is consistent with the objective design discussed in Section 3.2.3, where the torque term primarily regularizes attitude transients and avoids oscillatory behavior, while allowing brief localized corrections when needed. The peaks do not persist over extended intervals, indicating that reorientation transients are handled without sustained torque saturation.

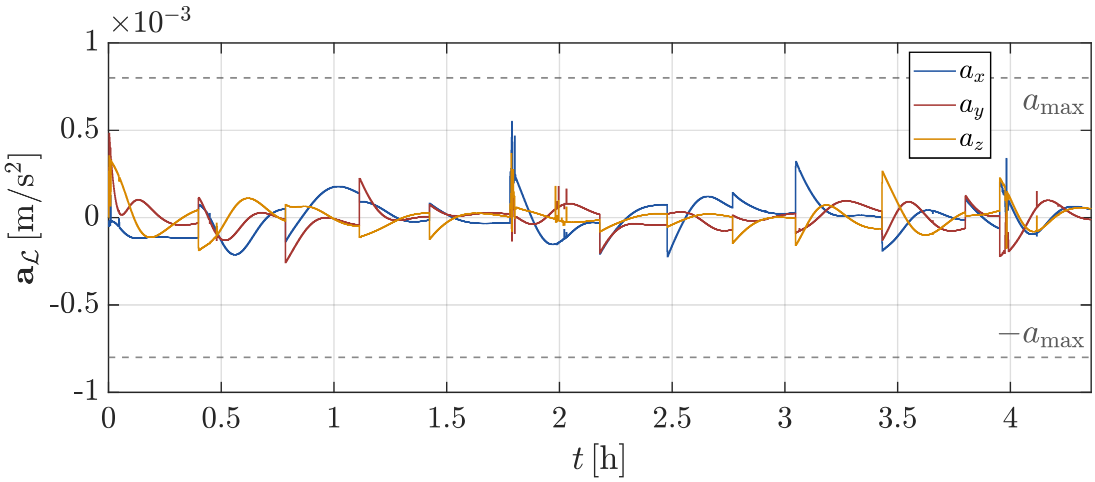


Figure 5.8: Translational control acceleration $\mathbf{a}_{\mathcal{L},k}$.

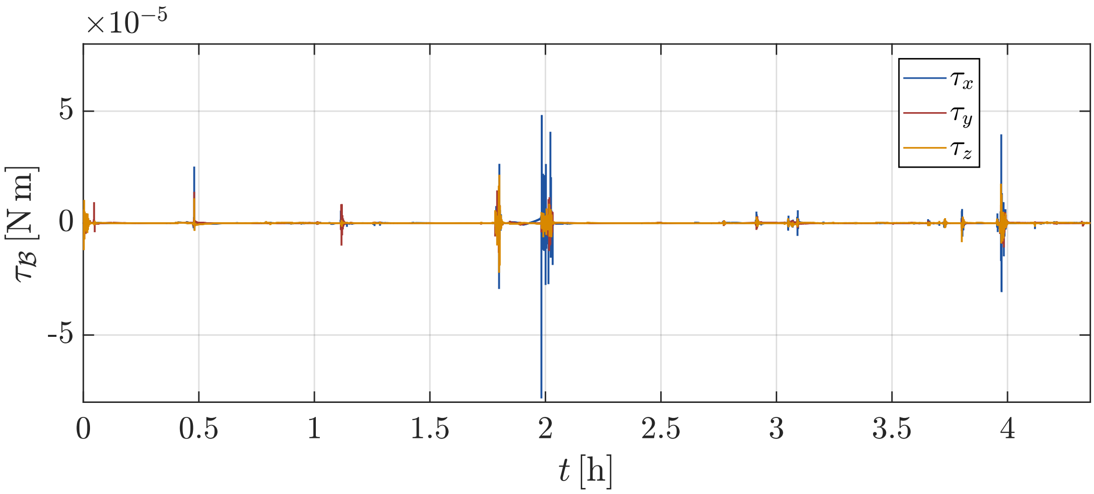


Figure 5.9: Control torque $\boldsymbol{\tau}_{\mathcal{B},k}$.

To complement the component-wise profiles, the step-to-step command variations provide a compact indication of slew activity and proximity to the imposed rate limits. Figures 5.10 and 5.11 report $\Delta\mathbf{a}_{\mathcal{L},k}$ and $\Delta\boldsymbol{\tau}_{\mathcal{B},k}$, respectively. The largest variations are typically observed in the translational acceleration during replanning updates, when a new exploration direction is selected and the guidance must redirect the motion accordingly. These episodes naturally lead to more demanding maneuvers because both the desired viewing geometry and the associated guidance objectives are updated. Nevertheless, the slew profiles remain within the prescribed bounds, confirming that the trajectory is obtained without unrealistically abrupt command switching.

Torque-rate variations exhibit analogous timing, being concentrated around critical reorientation phases, but they are markedly smaller in magnitude. This is consistent with brief, well-controlled attitude transients and supports the stable pointing behavior reported in Figure 5.5.

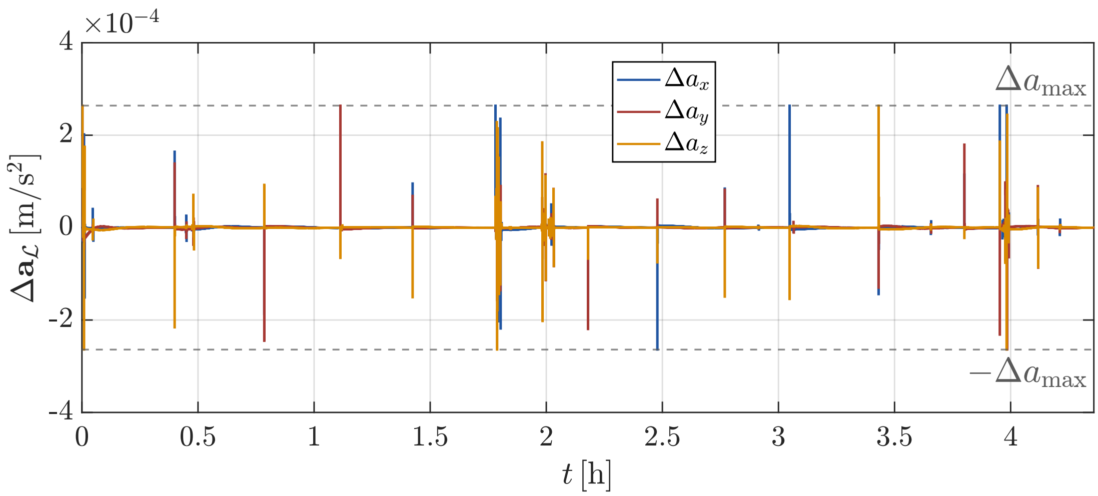


Figure 5.10: Acceleration step variation $\Delta \mathbf{a}_{\mathcal{L},k}$.

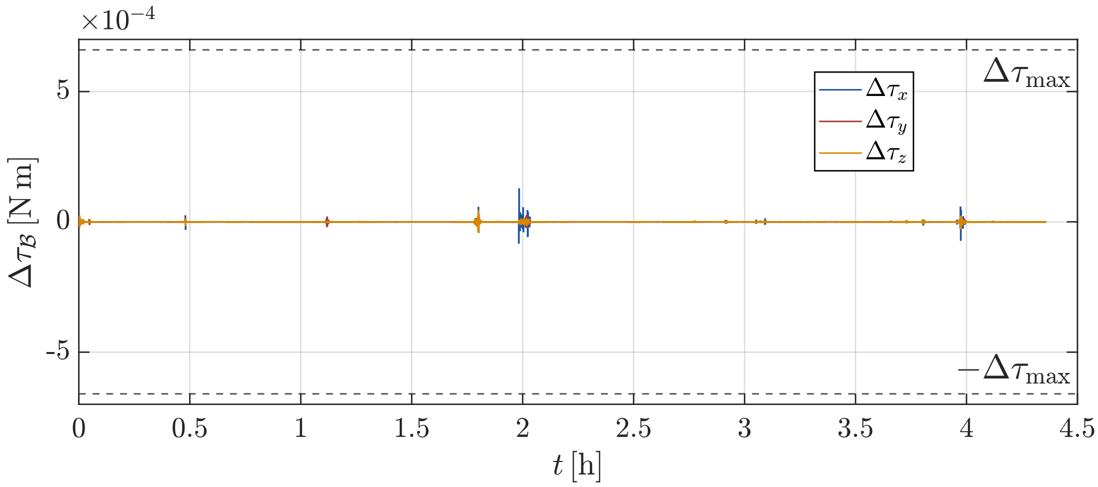


Figure 5.11: Torque step variation $\Delta \boldsymbol{\tau}_{\mathcal{B},k}$.

Total Δv budget. The overall mission cost can be readily computed by summing the contributions of the commanded accelerations over time.

$$\Delta v \triangleq \sum_{k} \left\| \mathbf{a}_{\mathcal{L},k} \right\| T_s \tag{5.1}$$

For the present run, the resulting budget is $\Delta v = 1.81\ \mathrm{m\,s^{-1}}$, consistent with the sustained low-thrust steering profile observed in Figure 5.8 and with the absence of extended high-acceleration phases.

5.2.5. Planner behavior

This section examines the supervisory layer that governs direction updates. The emphasis is on whether the replanning logic yields interpretable and stable decisions driven by coverage progress and by the avoidance of redundant viewpoints, rather than by rapid back and forth switching. The discussion relates the observed direction update timeline to the evolution of the coverage indicators, consistently with the supervisory policy introduced in Chapter 4.

The supervisory layer updates the exploration direction only when the progress-based conditions in Section 4.4.3 are satisfied. This evolution is summarized in Figure 5.12, which reports the selected mapping index i_{active} over the iterations. Direction changes are sparse with respect to the MPC sampling rate, and rapid back and forth switching between nearby zones is avoided, preserving both efficiency and interpretability of the exploration process.

Beyond sparsity, the index sequence exhibits an approximately cyclic progression, indicating that the planner does not simply target the most under covered region at every trigger. Instead, direction updates balance expected coverage improvement with the feasibility of the required transition in close proximity, so that reconfigurations remain limited and do not translate into unnecessary propellant expenditure. The observed pattern is consistent with a policy that steadily sweeps complementary sectors while exploiting the natural evolution of the relative geometry to keep maneuvering costs contained.

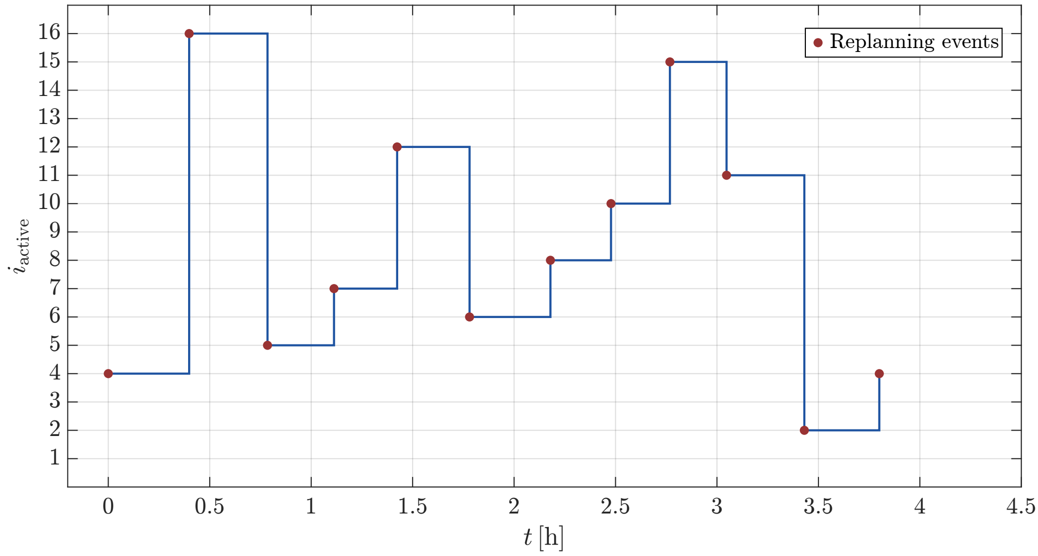


Figure 5.12: Replanning timeline with selected mapping direction index i_{active}.

5.3. Computational Performance

Since the MPC policy is applied at the sampling rate T_s, real-time execution requires that the per-step optimization completes within the available time budget, while the event driven replanning overhead must remain compatible with the same constraint. All simulations are executed with fixed software, solver and tolerance settings, so that the reported runtimes reflect the variability induced by the closed-loop evolution rather than by changes in the computational environment.

The per-step NLP problem must be solved within T_s to ensure that a new command is available before the next actuation update. Figure 5.13 reports the distribution of MPC solve times, enabling a direct comparison against the reference threshold T_s. In practice, a conservative margin must also be reserved for state estimation and command uplink within the same cycle. The results show that convergence is consistently fast: solve times concentrate well below the sampling interval and remain, on average, comfortably within the available time budget. Occasional peaks occur, typically associated with localized changes in constraint activity and with reference updates, but they do not challenge the timing requirement.

To support a robust timing interpretation, percentile-based indicators are reported in Table 5.6. In particular, the median and high-percentile values provide a more informative characterization than a single mean, as they capture both typical performance and the tail behaviour observed over the full simulation.

	Value	Unit
MPC solve time (median)	$t_{50} = 0.3428$	[s]
MPC solve time (mean)	$\bar{t} = 0.4987$	[s]
MPC solve time (95th percentile)	$t_{95} = 1.347$	[s]
MPC solve time (max)	$t_{\max} = 3.5810$	[s]
Configured per-solve time budget	$t_{\text{cap}} = 4$	[s]

Table 5.6: Runtime statistics for the MPC solve.

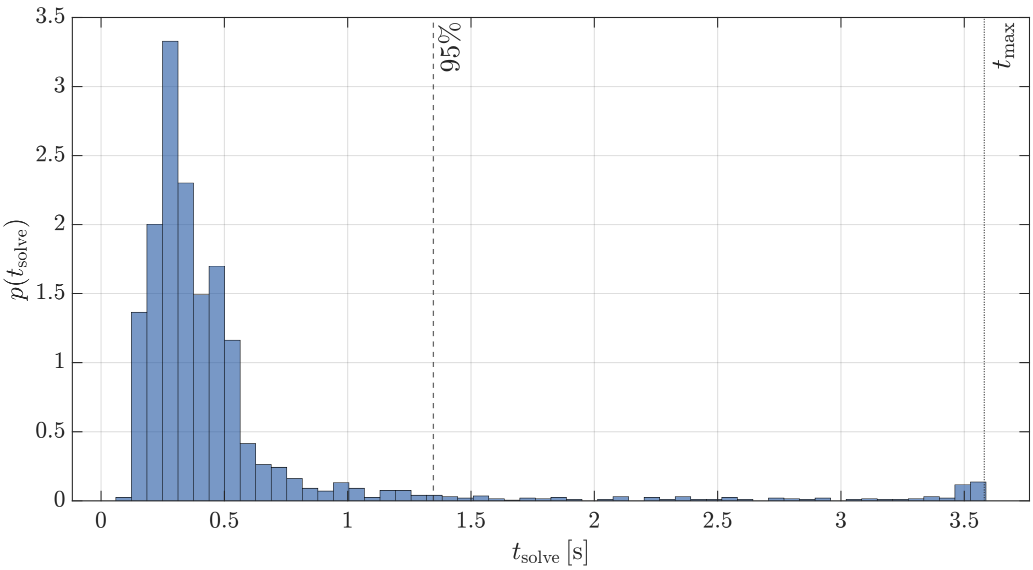


Figure 5.13: Per-step MPC solve time distribution.

Finally, it is worth noting that the reported runtimes are obtained with a high-level scripting implementation. Without changing the control formulation, further reductions are achievable by accelerating function and derivative evaluations via compilation and code generation and by moving the optimization loop to a lower-level implementation. Dedicated real-time NMPC toolchains based on Sequential Quadratic Programming (SQP) real-time iteration can reduce latency by splitting computations into preparation and feedback phases and by solving structured QPs efficiently. When tighter timing guarantees are required, an alternative is to adopt convexification based guidance solvers (e.g., SCP), which trade exact NMPC optimality for fast, reliable solves of a sequence of convex subproblems while retaining constraints and multi-objective structure.

The high-level planner introduces an extra computational cost whenever a direction update is triggered. Since replanning is event-driven and occurs at a slower effective rate than the MPC sampling, its impact has to be evaluated on a per-trigger basis. The

	Value	Unit
Planner runtime (mean per trigger)	$\bar{t}_{\text{plan}} = 19.744$	[s]
Planner runtime (median per trigger)	$t_{\text{plan},50} = 18.681$	[s]
Planner runtime (max per trigger)	$t_{\text{plan,max}} = 24.345$	[s]

Table 5.7: Runtime statistics for the high-level planner.

summary statistics in Table 5.7 complement the per-update runtimes already reported in Figure 5.12.

In the present implementation, replanning is not capable of being completed within a single MPC sampling interval. A new direction typically becomes available after a small number of MPC cycles (about 3–5, i.e., $\approx$ 15–25 s for $T_s = 5$ s), yet it is considered available at the next MPC call in the results shown here. While the associated scheduling problem is not addressed in this work, the same timing requirement can be satisfied in an on-board architecture without altering the guidance formulation. The planner can be executed asynchronously using the predicted state evolution over the upcoming MPC intervals, so that the updated direction is computed ahead of time, or the current direction can be held for one additional interval whenever a replanning result is delayed. These strategies are consistent with standard real-time NMPC implementations that split computations across cycles and employ asynchronous updates to accommodate variable solve times while preserving real-time execution.

The measured runtimes support the online feasibility of the proposed architecture at the selected sampling time T_s. MPC solve times remain comfortably within the available interval, while replanning overhead is both sparse and controlled and can be accommodated through standard scheduling choices without affecting the closed-loop formulation.

5.4. Mapping Assessment

This section reports the core mapping results of the proposed imaging-driven guidance strategy, with the objective of demonstrating complete and uniform coverage of the target. The discussion first follows the coverage indicators over time and the associated termination logic, then summarizes the final per-zone coverage distribution as a compact measure of uniformity and finally provides an intuitive spatial visualization of the achieved coverage on the discretized mapping tessellation.

Coverage metrics

Mapping progress is tracked through the coverage memory $\mathbf{C}_k \in \mathbb{R}_{\geq 0}^{N_d}$ defined in Section 4.2.2. Alongside the total accumulated imaging time, it is useful to recall the two completion oriented quantities adopted throughout the supervisory logic, namely the capped per-direction coverage and the corresponding global completion indicator,

$$\tilde{C}_{k,i} \triangleq \min\big(C_{k,i},\, C_{\text{tgt}}\big), \qquad \bar{C}_k \triangleq \frac{1}{N_d}\sum_{i=1}^{N_d} \tilde{C}_{k,i}. \tag{5.2}$$

This definition ensures that progress is driven by directions that have not yet reached C_{tgt}, so that $\bar{C}_k$ provides a faithful measure of mapping completion.

Figure 5.14 shows that $\bar{C}_k$ rises steadily until the prescribed acceptance level is reached. The initial phase exhibits a linear increase, close to the maximum attainable growth rate. This is expected when all directions are still below C_{tgt} and imaging valid configurations are readily available. It also provides a practical confirmation that the exploration policy converts available imaging time into effective global progress with limited overhead. As the run advances and the fraction of already covered directions increases, the curve transitions to a step-like evolution. This behavior reflects episodic replanning and finite-horizon decision making, in which direction updates are selected for their expected impact over a long interval rather than for instantaneous gain. In this regime, brief plateaus correspond to transitions over already fully observed regions: imaging time may still accumulate, while the capped indicator advances only when the effective coverage associated with the corresponding directions remains below C_{tgt}.

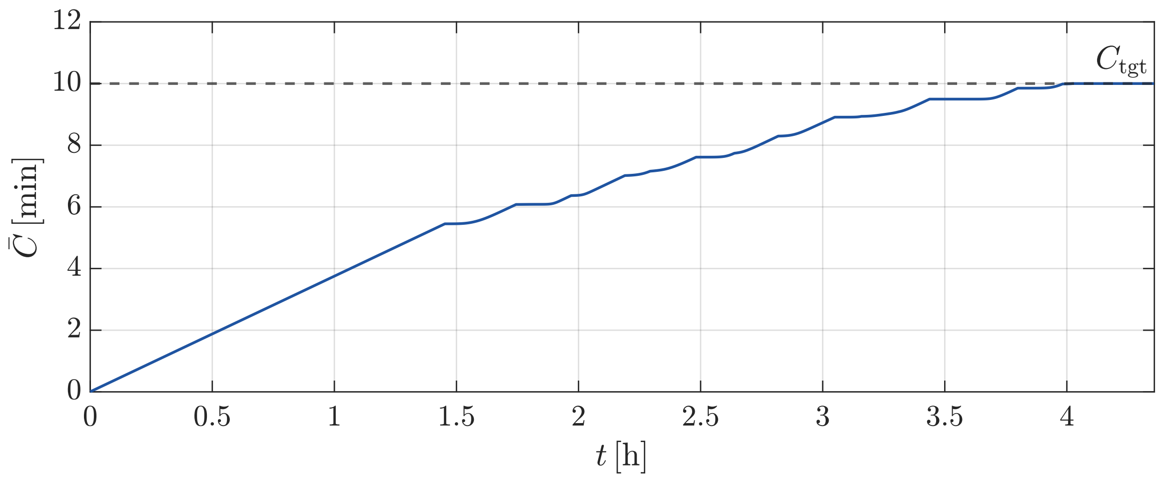


Figure 5.14: Global completion indicator $\bar{C}_k$ evolution.

The simulation terminates when the capped progress indicator reaches the prescribed acceptance ratio, namely when $\bar{C}_k \geq \alpha C_{\text{tgt}}$. This criterion matches the operational meaning of completion, since it certifies that the global mapping state is sufficiently close to the desired uniform coverage.

The final outcome is summarized by the distribution of $\tilde{C}_{k,i}$ across directions. As shown in Figure 5.15, the stopping rule guarantees a global completion level $\bar{C}_k \geq \alpha\, C_{\text{tgt}}$ at termination. In the simulation considered, the achieved outcome is even stronger, with all directions reaching at least C_{tgt}. This is enabled by the return leg toward the initial viewpoint, which adds additional imaging time along the traversed sectors and helps close any residual deficits, while naturally reinforcing directions that are already well covered.

The remaining dispersion in the bars, with some directions accumulating noticeably higher values, is therefore expected and reflects unavoidable revisits and viewpoint continuity along the executed motion. Achieving perfectly equal per-direction values would be unrealistic in closed loop and would typically require aggressive reconfiguration, with a disproportionate impact on control effort and fuel consumption.

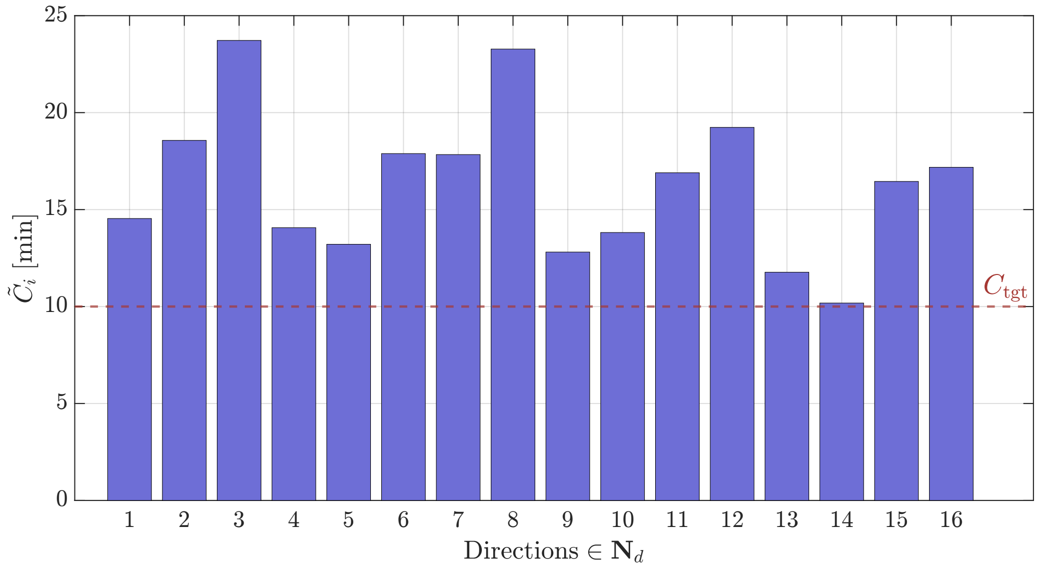


Figure 5.15: Final coverage distribution $\mathbf{C}_k$ across directions.

Overall, these results provide quantitative evidence of mapping completeness and uniformity: the capped completion indicator converges steadily to the prescribed level, and the final per-direction distribution confirms that no direction remains under covered at termination.

Spatial distribution

Distribution plots quantify the coverage outcome, whereas spatial visualizations provide a direct geometric interpretation of the final mapping result. Figure 5.16 reports the capped per-direction coverage on the spherical set of mapping directions introduced in Section 4.2.2. Each discretized direction is associated with its representative location on the unit sphere, providing an immediate assessment of whether coverage is balanced over the viewing domain and whether any region remains systematically under-covered. In the context of NeRF-based reconstruction, this perspective is particularly informative because it reflects the diversity and balance of viewpoints around the target, which are key drivers of reconstruction completeness and quality.

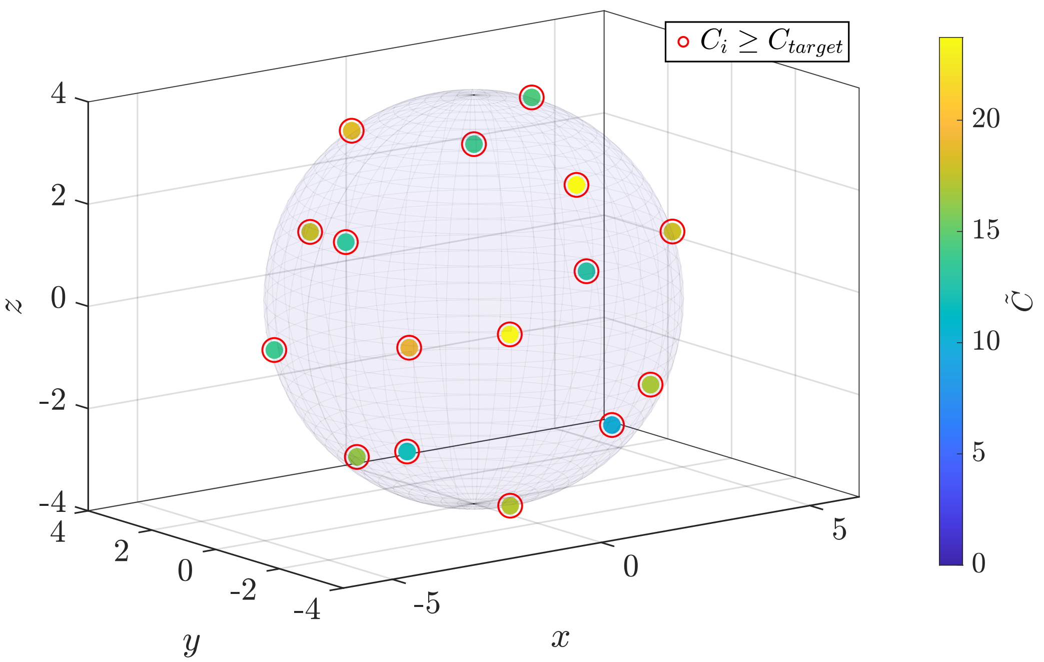


Figure 5.16: Spatial representation of the final coverage distribution.

To relate the spatial outcome to the executed behavior, Figure 5.17 shows the relative trajectory in $\mathcal{L}$ and highlights how it evolves across the discretized mapping sectors. The color coding reflects the direction, and associated Voronoi region, that yields the largest mapping contribution at each time, as determined by the zone membership weights used in the coverage update. This view highlights how the motion visits and covers different sectors of the tessellation. Moreover, It shows that the trajectory spans a range of relative positions even within each sector, supporting the collection of views that are not purely redundant at a fixed geometry.

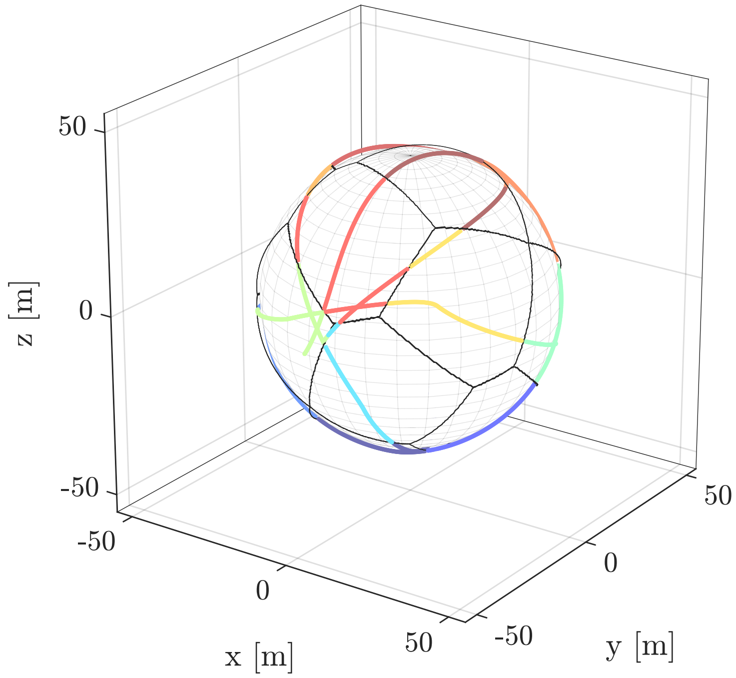


Figure 5.17: LVLH trajectory and Voronoi tesselation.

The spatial visualizations complement the convergence metrics by confirming that the achieved coverage is not only sufficient on average, but also well distributed around the target. The final pattern remains close to the desired level across the direction set and the trajectory-by-sector view indicates that viewpoints are progressively distributed over the tessellation rather than concentrated in limited regions.

5.5. Impact on 3D Reconstruction

Once the imaging campaign is completed, the acquired frames are assembled into a dataset and processed by a NeRF-based reconstruction pipeline. In this work, the NVIDIA software *Neuralangelo* [61] is adopted to assess the suitability of the viewpoint set produced by the proposed guidance strategy for downstream 3D reconstruction. In practice, Neuralangelo reconstructs both a radiance field and the corresponding mesh.

Reconstruction tuning and post-processing are outside the scope of this thesis, nevertheless, the qualitative consistency of the recovered model provides a direct end-to-end check of the effectiveness of the imaging-driven motion.

Figure 5.18 summarizes the image selection procedure adopted to build the dataset. Frames are extracted in batches so as to maintain, as much as possible, a uniform spacing along the trajectory while preserving viewpoint continuity between the initial and final configurations, consistently with the termination policy discussed in Section 4.4.3.

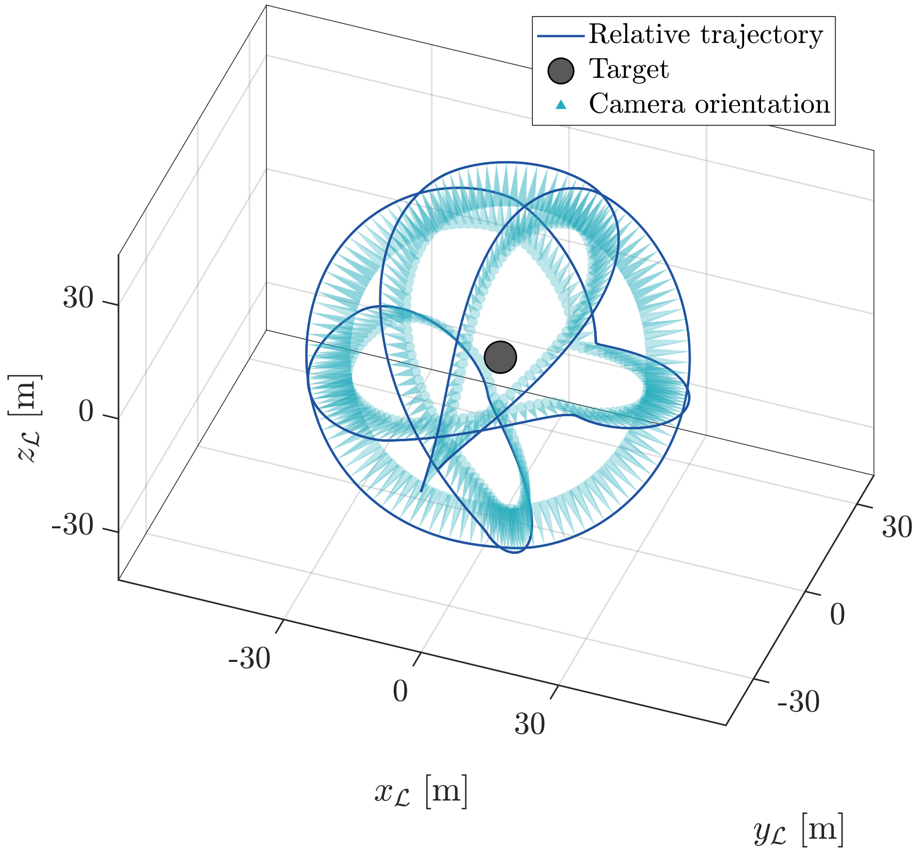


Figure 5.18: LVLH trajectory and camera pointing at sampled frames.

The final dataset is composed of 400 images selected from the generated viewpoints. Representative examples are reported in Figure 5.19 to provide visual context on the acquired viewpoints and on the target appearance throughout the run. Reconstruction is then carried out following the procedure described in [50].

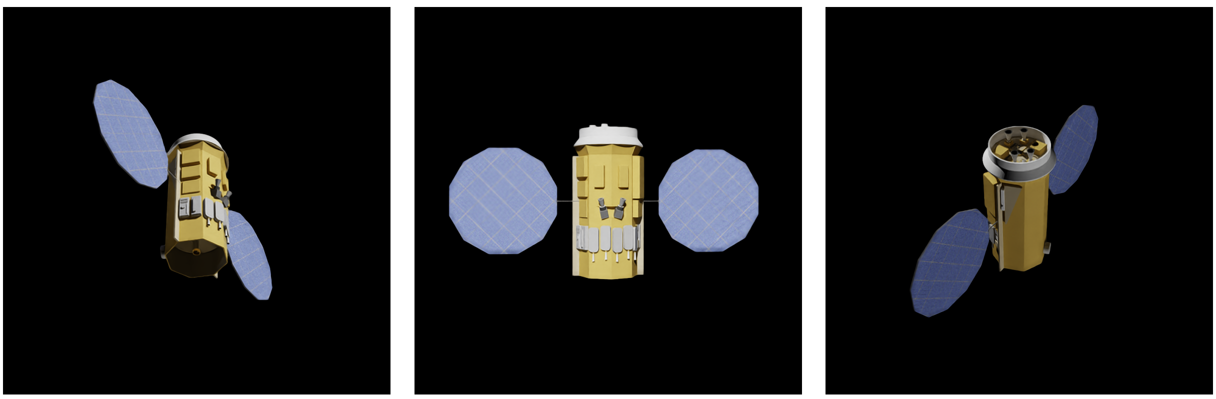


Figure 5.19: Examples of the images used to form the reconstruction dataset.

Reconstruction results.

Figure 5.20 reports representative renderings from the learned model at different viewpoints. The reconstruction preserves the main geometric and photometric features of the target and remains visually consistent across views, supporting that the proposed guidance produces a sufficiently diverse and balanced view set for NeRF-based learning. Minor artifacts are expected, since the reconstruction pipeline is not tuned within this work and no dedicated post-processing is applied. The goal here is qualitative validation rather than an optimized reconstruction benchmark.

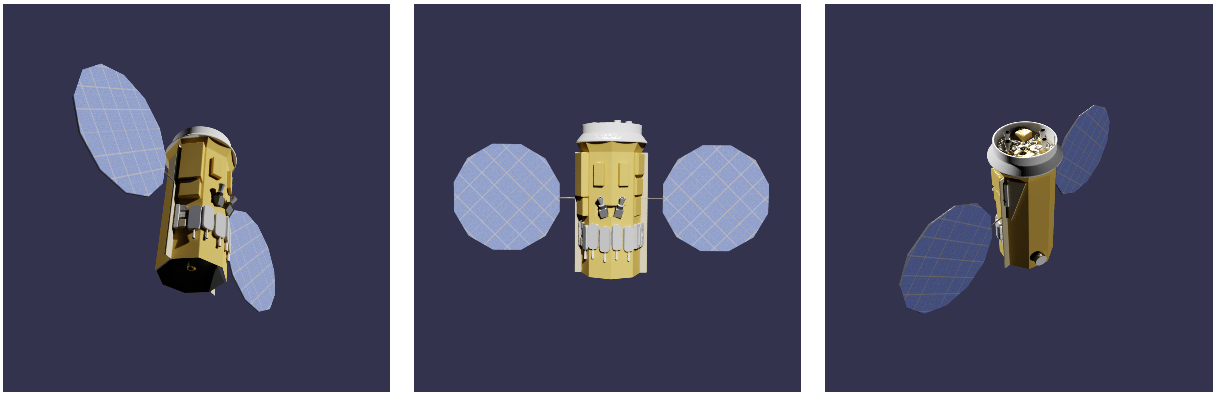


Figure 5.20: Representative reconstructed views from the proposed trajectory solution.

6 | Conclusions and Future Work

This thesis addressed the problem of autonomous close-proximity operations for inspection around a non-cooperative, unknown space object, where the primary operational goal is to actively generate a viewpoint set that is suitable for downstream deep learning based 3D reconstruction. In this context, inspection planning is still often treated separately from guidance and control, which is typically formulated to enforce safety and minimize propellant rather than to directly optimize inspection objectives. This separation can lead to trajectories that are dynamically feasible but poorly informative, or conversely to informative viewpoint plans that are difficult to execute under stringent operational constraints. The aim of this work was therefore to unify these aspects by embedding goal-oriented viewing requirements within a closed-loop guidance architecture, while still enforcing conservative safety margins, actuation limits and maintaining a computational footprint compatible with onboard applications.

The proposed solution is organized as a two layer architecture: a constrained MPC operating at the guidance rate and an episodic planner that updates the exploration direction sparsely based on mapping progress. At the MPC level, imaging requirements are handled through the objective function rather than imposed as hard constraints, to preserve feasibility under safety-driven maneuvers and transient reorientations, while still strongly promoting FoV compliance and range keeping, which are essential to the quality of the mission outputs. These imaging objectives are traded against actuation effort and an exploration shaping term that steers the vehicle toward the currently selected mapping direction. The supervisory layer then provides goal-oriented direction updates that bias the finite-horizon optimizer toward under covered regions while avoiding unnecessary manoeuvring.

The resulting closed-loop trajectory is consistent with the intended close-proximity regime and respects the imposed operational bounds on both translation and attitude. Range regulation meets the prescribed corridor requirements, with the motion settling near the nominal standoff distance and exploiting only limited deviations when they reduce reorientation effort and facilitate viewpoint transitions, rather than enforcing unnecessarily strict range tracking. Pointing compliance is maintained throughout the run, as the boresight-to-LOS misalignment remains within the FoV half-angle threshold, with short residual transients that are consistent with direction changes and reorientation phases. The commanded translational acceleration and attitude control torque remain within their bounds and evolve smoothly under rate limitations, including during direction update events, supporting actuation realism and a consistent effort profile compatible with sustained low-thrust steering. Mapping supervision remains effective at mission level, since viewpoint distribution sustains coverage growth and reaches the acceptance condition within finite time, after which the terminal viewpoint is selected near the initial configuration to preserve continuity in the assembled dataset in agreement with the termination logic.

A central outcome of this work is the explicit emphasis on autonomy under partial prior knowledge. Suitability to unknown or evolving targets is supported by the fact that exploration directions are generated online as a function of the inferred target geometry, rather than being prescribed a priori. In addition, the mapping process is carried out onboard as a function of the already covered area, so that the guidance strategy naturally adapts future viewpoints to what has been observed so far. Within the same structure, the architecture can accommodate updates to reconstruction and safety relevant parameters, such as the keep-out radius r_{KOZ}, the nominal standoff distance r_{opt}, or conservative shape envelopes, as additional information becomes available during the campaign, without altering the fundamental hierarchy of episodic planning, coverage supervision and constrained MPC.

Although the numerical validation in this thesis targeted a NeRF-based pipeline (with *Neuralangelo* used as a representative reconstruction backend), the proposed framework is more generally intended as a method for uniform and informative autonomous inspection under close-proximity constraints. Whenever a mission requires controlled viewpoint diversity, conservative separation, bounded actuation and limited onboard compute, the same principles extend naturally to spacecraft inspection and characterization, diagnosis of malfunctioning satellites or space debris and mapping-oriented operations around small bodies.

The thesis shows that embedding imaging requirements into a constrained receding-horizon controller, complemented by sparse high-level direction updates, yields an inspection strategy that is operationally feasible, interpretable and aligned with reconstruction-driven objectives. Episodic direction selection, combined with coverage-aware supervision and constrained MPC, gives a closed-loop behaviour that remains stable and goal-consistent while steadily driving the campaign toward completion.

6.1. Limitations and Future Developments

The proposed inspection framework was designed to be executable in real time, and the numerical evidence indicates that this objective is met for the considered scenario. Nevertheless, the same results also clarify the main limitations of the current implementation. From a computational budget and scheduling standpoint, while the NMPC solve time distribution remains compatible with the sampling interval adopted in this thesis, the episodic planner can become the bottleneck when the candidate set grows or the exploration lookahead horizon is extended. A natural evolution is to enforce a more deterministic timing structure without modifying the guidance logic. This could be achieved either by accelerating the planner via caching and coverage-driven pruning, or by adopting smoother surrogate objectives that enable faster local refinement (e.g., QP based stages). In parallel, the planner can be moved to an asynchronous execution mode, where it runs concurrently with the MPC and is allowed to refine its candidate evaluation until a predefined deadline, returning the best direction update available at that time. If an update is not available by the MPC call, the controller continues to operate using the last valid direction, so that closed-loop safety and feasibility are preserved.

Another limitation is the suboptimality by design. The method relies on a finite prediction horizon and on time-constrained online optimization, and therefore it typically converges to locally optimal solutions rather than guaranteeing global optimality for the full inspection problem. Additionally, the rotational prediction model is obtained via successive linearization and then frozen over each solve. This choice improves numerical robustness but can introduce conservatism when the attitude trajectory departs significantly from the linearization reference. Addressing these aspects does not require abandoning the receding-horizon philosophy, but rather strengthening it with mechanisms that mitigate myopic behavior. Examples include adaptive horizon selection, occasional long-horizon reoptimization steps activated when the coverage improvement rate falls below a prescribed threshold, and more informative warm-start policies.

A further aspect to consider is modeling realism and constraint enforcement. As in standard direct transcription, hard constraints are enforced at the nodes, and inter-sample satisfaction is not guaranteed in principle, although it is practically supported here by the small sampling time and by the smoothness of the resulting motion. This limitation can be addressed, for instance, by introducing state-triggered penalties or margins within the optimization, rather than relying solely on after the verification.

The proposed framework is designed to remain adaptable as additional information becomes available during the inspection and several flight oriented extensions can be incorporated with limited changes to the current structure. In particular, the target envelope can be progressively refined, for instance by replacing an initially spherical target model with an ellipsoidal one once the size and principal dimensions are better estimated. This refinement directly propagates to safety related quantities, including the keep-out radius and conservative margins, and to imaging-related choices, including the mapping discretization and the nominal stand-off distance that best supports the reconstruction pipeline. More generally, as the geometric model becomes more accurate, both the constraint set and the imaging objectives can be updated consistently within the same hierarchy of coverage supervision, episodic planning, and constrained MPC, enabling increasingly tailored trajectories as the campaign progresses.

Finally, the image layer itself is intentionally kept abstract. FoV satisfaction, pointing alignment and range keeping provide effective geometric proxies for viewpoint usefulness, yet in practice image quality is also governed by illumination and sensor level effects that are not modeled explicitly, including eclipse windows, phase-angle constraints, exposure limits,motion blur, occlusions and visibility losses. Moreover, the current framework assumes that relative navigation and attitude knowledge are sufficiently accurate to support closed-loop optimization, whereas realistic operations require a tighter coupling between viewpoint selection and estimation performance. Future work could therefore extend the imaging objectives by incorporating illumination and visibility aware constraints and should strengthen the guidance layer through uncertainty-aware mechanisms.

In summary, the limitations identified in this thesis are not structural weaknesses of the proposed architecture, but rather define a coherent progression toward increased flight realism by improving runtime determinism of the planner, strengthening safety guarantees beyond node-level enforcement and tightening the loop between perception and guidance. Within the same methodological core, these extensions also enable broader operational scenarios, including tumbling targets, time-varying exclusion regions and small-body environments where irregular gravity shapes the local dynamics. More generally, the results of this thesis support a clear design principle: inspection trajectories benefit from being generated end to end by tightly coupling constrained receding-horizon control with a persistent representation of mapping progress, so that safety, feasibility, and inspection-driven objectives are pursued together.

Bibliography

[1] Gabriella Gaias and Marco Lovera. Trajectory design for proximity operations: The relative orbital elements' perspective. *Journal of Guidance, Control, and Dynamics*, 44:1–9, 09 2021. doi: 10.2514/1.G006175.

[2] Daniel J Scheeres. Close proximity operations at small bodies: orbiting, hovering, and hopping. In *Mitigation of Hazardous Comets and Asteroids*, pages 313–337. Cambridge University Press, Cambridge, 2004.

[3] Antoine Petit, Éric Marchand, and Keyvan Kanani. Vision-based spacecraft pose estimation with model-based tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 50(3):2214–2231, 2014. doi: 10.1109/TAES.2014.120596.

[4] Linglong Zhou, Guoxin Wu, Yunbo Zuo, Xuanyu Chen, and Hongle Hu. A comprehensive review of vision-based 3d reconstruction methods. *Sensors*, 23(16):7241, 2023. doi: 10.3390/s23167241.

[5] Adam Herrmann and Hanspeter Schaub. Autonomous small body science operations using reinforcement learning. *Journal of Aerospace Information Systems*, 21:1–20, 07 2024. doi: 10.2514/1.I011376.

[6] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *CoRR*, abs/2003.08934, 2020. URL `https://arxiv.org/abs/2003.08934`.

[7] Shihan Chen and Bo Wu. Asteroid-nerf: Neural radiance fields for reconstructing asteroids from spacecraft images. *Astronomy & Astrophysics*, 687:A278, 2024. doi: 10.1051/0004-6361/202450053.

[8] Anne Mergy, Gurvan Lécuyer, Dawa Derksen, and Dario Izzo. Vision-based neural scene representations for spacecraft. *CoRR*, abs/2105.06405, 2021. URL `https://arxiv.org/abs/2105.06405`.

[9] Albert Ge. Importance sampling of views in nerf. *arXiv preprint*, 2023.

[10] Ping Lu. Autonomous trajectory planning for rendezvous and proximity operations by conic optimization. *Journal of Guidance, Control, and Dynamics*, 37(6):1749–1760, 2014.

[11] Benjamin Reed, Robert Smith, Bo Naasz, Joseph Pellegrino, and Charles Bacon. The restore-l servicing mission. In *Proceedings of the AIAA SPACE and Astronautics Forum and Exposition*, 09 2016. doi: 10.2514/6.2016-5478.

[12] Jason Forshaw, Ron Lopez, Akira Okamoto, Chris Blackerby, and Nobu Okada. The elsa-d end-of-life debris removal mission: Mission design, in-flight safety, and preparations for launch. In *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference*, pages 44–50, Kihei, Hawaii, 2019. Maui Economic Development Board. URL `https://amostech.com/TechnicalPapers/2019/Space-Based-Assets/Forshaw.pdf`. Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS).

[13] J. A. Starek, B. Açıkmeşe, I. A. Nesnas, and M. Pavone. *Spacecraft Autonomy Challenges for Next-Generation Space Missions*, volume 460 of *Lecture Notes on Control and Information Sciences*, chapter 1, pages 1–48. Springer, Berlin, 2016. doi: 10.1007/978-3-319-27060-9_1.

[14] Sei-ichiro Watanabe, Yuichi Tsuda, Makoto Yoshikawa, Satoshi Tanaka, Takanao Saiki, and Satoru Nakazawa. Hayabusa2 mission overview. *Space Science Reviews*, 208(1):3–16, 2017. doi: 10.1007/s11214-017-0377-1.

[15] OS Barnouin, MG Daly, EE Palmer, CL Johnson, RW Gaskell, M Al Asad, et al. Digital terrain mapping by the osiris-rex mission. *Planetary and Space Science*, 180: 104764, 2020. doi: 10.1016/j.pss.2019.104764.

[16] Sumeet G. Satpute, Per Bodin, and George Nikolakopoulos. Cooperative planning for multi-site asteroid visual coverage. *Advanced Robotics*, 35(21-22):1332–1346, 2021. doi: 10.1080/01691864.2021.1955003.

[17] Tochukwu Elijah Ogri, Muzaffar Qureshi, Zachary I Bell, Matthew Longmire, and Rushikesh Kamalapurkar. Monocular inspection of spacecraft under illumination constraints and avoidance regions. In *Proceedings of the American Control Conference (ACC)*, 2023. Oklahoma State University and Air Force Research Laboratory.

[18] Mark Stephenson, Daniel Huterer Prats, and Hanspeter Schaub. Autonomous satellite inspection in low earth orbit with optimization-based safety guarantees. In *AIAA Scitech 2021 Forum*, page 1061. Autonomous Vehicle Systems Lab, University of Colorado Boulder, 2021. doi: 10.2514/6.2021-1061.

[19] Michele Maestrini and Pierluigi Di Lizia. Guidance strategy for autonomous inspection of unknown non-cooperative resident space objects. *Journal of Guidance, Control, and Dynamics*, 45(6):1112–1127, 2022. doi: 10.2514/1.G006126.

[20] Antonio Rizza, Francesco Topputo, and Simone D'Amico. Goal-oriented asteroid mapping under uncertainties using sequential convex programming, 01 2024.

[21] Daniel J Scheeres. Orbital mechanics about small bodies. *Acta Astronautica*, 72: 1–14, 2012. doi: 10.1016/j.actaastro.2011.10.021.

[22] Marin Kobilarov and Sergio Pellegrino. Trajectory planning for CubeSat short-time-scale proximity operations. *Journal of Guidance, Control, and Dynamics*, 37(2): 566–579, 2014. doi: 10.2514/1.60289.

[23] Timothy E Rumford. Demonstration of autonomous rendezvous technology (dart) project summary. In *Space Systems Technology and Operations*, volume 5088, pages 10–19, Orlando, FL, April 2003. International Society for Optics and Photonics (SPIE). doi: 10.1117/12.487624.

[24] Staffan Persson, Per Bodin, Eberhard Gill, Jean-Alexis Harr, and Thomas Jörgensen. Flight results from the PRISMA formation flying and rendezvous demonstration mission. In *61st International Astronautical Congress*, pages 1–13, Prague, Czech Republic, 2010. International Astronautical Federation.

[25] Louis Breger and Jonathan P How. Safe trajectories for autonomous rendezvous of spacecraft. *Journal of Guidance, Control, and Dynamics*, 31(5):1478–1489, 2008. doi: 10.2514/1.34612.

[26] Isao Kawano, Masaaki Mokuno, Toru Kasai, and Tetsuo Suzuki. Autonomous rendezvous and docking results of ETS-VII. *Journal of Spacecraft and Rockets*, 38(4): 510–516, 2001. doi: 10.2514/2.3721.

[27] Gabriella Gaias and Jean-Sébastéan Ardaens. Flight demonstration of autonomous noncooperative rendezvous in low earth orbit. *Journal of Guidance, Control, and Dynamics*, 41(3):742–755, 2018. doi: 10.2514/1.G002821.

[28] Scott B. Broschart and Daniel J. Scheeres. Control of hovering spacecraft near small bodies: Application to asteroid missions. *Journal of Guidance, Control, and Dynamics*, 28(2):343–354, 2005. doi: 10.2514/1.3890.

[29] Daniel Fourie, Brent E Tweddle, Steve Ulrich, and Alvar Saenz-Otero. Flight results of vision-based navigation for autonomous spacecraft inspection of unknown objects. *Journal of Spacecraft and Rockets*, 51(6):2016–2026, 2014. doi: 10.2514/1.A32840.

[30] D. N. DellaGiustina, C. A. Bennett, K. J. Becker, D. R. Golish, L. Le Corre, D. A. Cook, and et al. Overcoming the challenges associated with image-based mapping of small bodies in preparation for the osiris-rex mission to (101955) bennu. *Earth and Space Science*, 5(12):1164–1191, 2018. doi: 10.1029/2018EA000382.

[31] D. S. Lauretta and et al. Osiris-rex: Sample return from asteroid (101955) bennu. *Space Science Reviews*, 212:925–984, 2017. doi: 10.1007/s11214-017-0405-1.

[32] Daniel A. Surovik. *Autonomous Mission Design in Extreme Orbit Environments*. PhD thesis, University of Colorado at Boulder, 2016. URL `https://ccar.colorado.edu/scheeres/scheeres/assets/ThesesAbstracts/surovik_dissertation.pdf`. PhD Thesis in Aerospace Engineering, University of Colorado at Boulder.

[33] Francesco Capolupo, Thierry Simeon, and Jean-Claude Berges. Heuristic guidance techniques for the exploration of small celestial bodies. *IFAC-PapersOnLine*, 50(1): 8279–8284, 2017. doi: 10.1016/j.ifacol.2017.08.1387.

[34] Howard D. Curtis. *Orbital Mechanics for Engineering Students*. Butterworth-Heinemann, 3rd edition, 2013. ISBN 978-0-08-097747-8. Discusses Hill and Clohessy–Wiltshire equations and relative motion dynamics.

[35] W. H. CLOHESSY and R. S. WILTSHIRE. Terminal guidance system for satellite rendezvous. *Journal of the Aerospace Sciences*, 27(9):653–658, 1960. doi: 10.2514/8.8704. URL `https://doi.org/10.2514/8.8704`.

[36] Jose Antonio Rebollo, Rafael Vazquez, Ignacio Alvarado, and Daniel Limon. Mpc for tracking applied to rendezvous with non-cooperative tumbling targets ensuring stability and feasibility. *Control Engineering Practice*, 122:105118, 2022. doi: 10.1016/j.conengprac.2022.105118.

[37] Gabriel Behrendt, Matthew Hale, Alexander Soderlund, Sean Phillips, and Evan Kain. Time-constrained model predictive control for autonomous satellite rendezvous, proximity operations, and docking. In *AIAA SCITECH 2022 Forum*, pages 1–15, 2022. doi: 10.2514/6.2022-0345.

[38] Simone D'Amico. Relative orbital elements as integration constants of hill's equations. Technical Report TN-05-08, German Space Operations Center (GSOC), DLR, 12 2005.

[39] K. Yamanaka and F. Ankersen. New state transition matrix for relative motion on an arbitrary elliptical orbit. *Journal of Guidance, Control, and Dynamics*, 25(1): 60–66, 2002. doi: 10.2514/2.4875.

[40] Stefano Di Cairano, Haesun Park, and Ilya V. Kolmanovsky. Model predictive control approach for guidance of spacecraft rendezvous and proximity maneuvering. *International Journal of Robust and Nonlinear Control*, 22:1398–1427, 2012. doi: 10.1002/rnc.2827.

[41] H. Zhou, A. Kelly, and J. J. Hull. Receding-horizon sequential convex programming for 6-dof guidance and control. *Journal of Guidance, Control, and Dynamics*, 42(10): 2269–2286, 2019. doi: 10.2514/1.G003208.

[42] Xun Wang, Yanyan Li, Xueyang Zhang, Rui Zhang, and Daoning Yang. Model predictive control for close-proximity maneuvering of spacecraft with adaptive convexification of collision avoidance constraints. *Aerospace Science and Technology*, 130:107873, 2022. doi: 10.1016/j.ast.2022.107873.

[43] 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):1–1, 2015.

[44] Tao Fu, Yu Zhou, and Ying Wang. Neural field-based space target 3d reconstruction with predicted depth priors. *Aerospace*, 11(12):997, 2024. doi: 10.3390/aerospace11120997.

[45] Chang Xu, Bu Sun, Zhan Wang, and Zhuang Wang. Optimized nerf for 3d reconstruction for space targets based on deblurring, low-light enhancement and hash encoding. *Advances in Space Research*, 2025. doi: 10.1016/j.asr.2025.01.065.

[46] Bu Wang, Bo Ren, Bu Sun, Wang Zhan, and Zhuang Wang. 3d reconstruction method of space target based on optical images with wide baseline via neural radiance fields. In *Journal of Physics: Conference Series*, volume 2347, page 012005, 2022. doi: 10.1088/1742-6596/2347/1/012005.

[47] Ewen Legrand, Clément Delporte, and Nicolas Delaygue. Leveraging neural radiance fields for pose estimation of an unknown space object during proximity operations, 2024.

[48] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[49] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-

adjusting neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[50] Bala Prenith Reddy Gopu, Timothy J Huber, George M Nehma, Patrick D Quinn, Madhur Tiwari, Matt Ueckermann, David Hinckley, and Christopher McKenna. Dynamic scene 3d reconstruction of an uncooperative resident space object. In *AIAA SCITECH 2026 Forum*, page 1038, 2026.

[51] Basilio Caruso, Trupti Mahendrakar, Van Minh Nguyen, Ryan T. White, and Todd Steffen. 3d reconstruction of non-cooperative resident space objects using instant ngp-accelerated nerf and d-nerf, 2023. URL `https://arxiv.org/abs/2301.09060`.

[52] Mátē Forray, David Huang, Tae Ha Park, and Simone D'Amico. Joint attitude estimation and 3d neural reconstruction for non-cooperative space objects: A new spaceborne slam perspective, 2025.

[53] John T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Advances in Design and Control. Society for Industrial and Applied Mathematics (SIAM), 2nd edition, 2009. ISBN 9780898716887.

[54] Francesco Topputo and Cheng Zhang. Survey of direct transcription for low-thrust space trajectory optimization with applications. *Abstract and Applied Analysis*, 2014, 2014. doi: 10.1155/2014/851720. Article ID 851720.

[55] Benoît C. Chachuat. Nonlinear and dynamic optimization: From theory to practice. Lecture Notes, IC-32, École Polytechnique Fédérale de Lausanne (EPFL), 2007. Winter Semester 2006/2007.

[56] James B. Rawlings, David Q. Mayne, and Moritz M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, LLC, Santa Barbara, CA, 2nd edition, 2022. ISBN 9780975937754. URL `http://www.nobhillpublishing.com`.

[57] Álvaro González. Measurement of areas on a sphere using fibonacci and latitude–longitude lattices. *Mathematical Geosciences*, 42(1):49–64, 2010.

[58] Jeffrey M. Augenbaum and Charles S. Peskin. On the Construction of the Voronoi Mesh on a Sphere. *Journal of Computational Physics*, 59(2):177–192, June 1985. doi: 10.1016/0021-9991(85)90140-8.

[59] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Pro-*

gramming, 106:25–57, 2006. URL `https://api.semanticscholar.org/CorpusID:14183894`.

[60] Joel A E Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 2018.

[61] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H. Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction, 2023. URL `https://arxiv.org/abs/2306.03092`.

A | Appendix A

Purpose and notation

The rotational predictor adopted in Equation (2.24) is linearized along a reference trajectory to construct the time-varying Jacobian $\mathbf{\Lambda}_r(t) = \partial \mathbf{f}/\partial \mathbf{x}_r$, which directly enters the rotational STM. This appendix reports the explicit entries of $\mathbf{\Lambda}_r$ used in the implementation. The derivation follows [41].

Given the rotational state,

$$\mathbf{x}_r = \begin{bmatrix} \omega_x & \omega_y & \omega_z & \sigma_1 & \sigma_2 & \sigma_3 \end{bmatrix}^\top, \tag{A.1}$$

starting from Equations (2.20) and (2.24) the rotational dynamics $\mathbf{f}(\mathbf{x}_r, t)$ is given by

$$\mathbf{f}(\mathbf{x}_r, t) = \begin{bmatrix} f_1 & f_2 & f_3 & f_4 & f_5 & f_6 \end{bmatrix}^\top. \tag{A.2}$$

The Jacobian $\mathbf{\Lambda}_r \in \mathbb{R}^{6\times 6}$ is

$$\mathbf{\Lambda}_r = \begin{bmatrix}
\frac{\partial f_1}{\partial \omega_x} & \frac{\partial f_1}{\partial \omega_y} & \frac{\partial f_1}{\partial \omega_z} & \frac{\partial f_1}{\partial \sigma_1} & \frac{\partial f_1}{\partial \sigma_2} & \frac{\partial f_1}{\partial \sigma_3} \\
\frac{\partial f_2}{\partial \omega_x} & \frac{\partial f_2}{\partial \omega_y} & \frac{\partial f_2}{\partial \omega_z} & \frac{\partial f_2}{\partial \sigma_1} & \frac{\partial f_2}{\partial \sigma_2} & \frac{\partial f_2}{\partial \sigma_3} \\
\frac{\partial f_3}{\partial \omega_x} & \frac{\partial f_3}{\partial \omega_y} & \frac{\partial f_3}{\partial \omega_z} & \frac{\partial f_3}{\partial \sigma_1} & \frac{\partial f_3}{\partial \sigma_2} & \frac{\partial f_3}{\partial \sigma_3} \\
\frac{\partial f_4}{\partial \omega_x} & \frac{\partial f_4}{\partial \omega_y} & \frac{\partial f_4}{\partial \omega_z} & \frac{\partial f_4}{\partial \sigma_1} & \frac{\partial f_4}{\partial \sigma_2} & \frac{\partial f_4}{\partial \sigma_3} \\
\frac{\partial f_5}{\partial \omega_x} & \frac{\partial f_5}{\partial \omega_y} & \frac{\partial f_5}{\partial \omega_z} & \frac{\partial f_5}{\partial \sigma_1} & \frac{\partial f_5}{\partial \sigma_2} & \frac{\partial f_5}{\partial \sigma_3} \\
\frac{\partial f_6}{\partial \omega_x} & \frac{\partial f_6}{\partial \omega_y} & \frac{\partial f_6}{\partial \omega_z} & \frac{\partial f_6}{\partial \sigma_1} & \frac{\partial f_6}{\partial \sigma_2} & \frac{\partial f_6}{\partial \sigma_3}
\end{bmatrix}. \tag{A.3}$$

Rigid-body dynamics

Assuming principal axes inertia $\mathbf{J} = \mathrm{diag}(I_{xx}, I_{yy}, I_{zz})$, the drift dynamics are

$$f_1 = \frac{I_{yy} - I_{zz}}{I_{xx}}\omega_y\omega_z, \qquad f_2 = \frac{I_{zz} - I_{xx}}{I_{yy}}\omega_z\omega_x, \qquad f_3 = \frac{I_{xx} - I_{yy}}{I_{zz}}\omega_x\omega_y. \tag{A.4}$$

Hence the partial derivatives are

$$\frac{\partial f_1}{\partial \omega_x} = 0, \qquad \frac{\partial f_2}{\partial \omega_x} = \frac{I_{zz} - I_{xx}}{I_{yy}}\,\omega_z, \qquad \frac{\partial f_3}{\partial \omega_x} = \frac{I_{xx} - I_{yy}}{I_{zz}}\,\omega_y, \tag{A.5}$$

$$\frac{\partial f_1}{\partial \omega_y} = \frac{I_{yy} - I_{zz}}{I_{xx}}\,\omega_z, \qquad \frac{\partial f_2}{\partial \omega_y} = 0, \qquad \frac{\partial f_3}{\partial \omega_y} = \frac{I_{xx} - I_{yy}}{I_{zz}}\,\omega_x, \tag{A.6}$$

$$\frac{\partial f_1}{\partial \omega_z} = \frac{I_{yy} - I_{zz}}{I_{xx}}\,\omega_y, \qquad \frac{\partial f_2}{\partial \omega_z} = \frac{I_{zz} - I_{xx}}{I_{yy}}\,\omega_x, \qquad \frac{\partial f_3}{\partial \omega_z} = 0. \tag{A.7}$$

and

$$\frac{\partial f_i}{\partial \sigma_j} = 0, \qquad i = 1,2,3, \quad j = 1,2,3. \tag{A.8}$$

MRP kinematics

Define

$$\Sigma \triangleq 1 - \sigma_1^2 - \sigma_2^2 - \sigma_3^2. \tag{A.9}$$

Let $\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}/\mathcal{L}} = [\omega_1, \omega_2, \omega_3]^\top$ denote the relative angular rate entering the MRP kinematics, so that

$$\dot{\boldsymbol{\sigma}}_{\mathcal{B}/\mathcal{L}} = \frac{1}{4}\Big((\Sigma)\mathbf{I}_3 + 2[\boldsymbol{\sigma}]_\times + 2\boldsymbol{\sigma}\boldsymbol{\sigma}^\top\Big)\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}/\mathcal{L}}. \tag{A.10}$$

In components, $f_4 = \dot{\sigma}_1$, $f_5 = \dot{\sigma}_2$, $f_6 = \dot{\sigma}_3$.

Partial derivatives with respect to $\boldsymbol{\omega}$

The partial derivatives with respect to $\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}/\mathcal{L}}$ are

$$\frac{\partial f_4}{\partial \omega_x} = \frac{1}{4}(\Sigma + 2\sigma_1^2), \qquad \frac{\partial f_4}{\partial \omega_y} = \frac{1}{2}(\sigma_1\sigma_2 - \sigma_3), \qquad \frac{\partial f_4}{\partial \omega_z} = \frac{1}{2}(\sigma_1\sigma_3 + \sigma_2), \tag{A.11}$$

$$\frac{\partial f_5}{\partial \omega_x} = \frac{1}{2}(\sigma_2\sigma_1 + \sigma_3), \qquad \frac{\partial f_5}{\partial \omega_y} = \frac{1}{4}(\Sigma + 2\sigma_2^2), \qquad \frac{\partial f_5}{\partial \omega_z} = \frac{1}{2}(\sigma_2\sigma_3 - \sigma_1), \tag{A.12}$$

$$\frac{\partial f_6}{\partial \omega_x} = \frac{1}{2}(\sigma_3\sigma_1 - \sigma_2), \qquad \frac{\partial f_6}{\partial \omega_y} = \frac{1}{2}(\sigma_3\sigma_2 + \sigma_1), \qquad \frac{\partial f_6}{\partial \omega_z} = \frac{1}{4}(\Sigma + 2\sigma_3^2). \tag{A.13}$$

Partial derivatives with respect to $\boldsymbol{\sigma}$

Since $\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}/\mathcal{L}}$ depends on $\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{L}}$ through the LVLH coupling, the derivatives $\partial f_4/\partial\sigma_i,\ f_5/\partial\sigma_i$, $f_6/\partial\sigma_i$ are written via chain rule as

$$\frac{\partial f_4}{\partial\sigma_1} = \frac{\sigma_1}{2}\omega_1 + \frac{\partial f_4}{\partial\omega_x}\frac{\partial\omega_1}{\partial\sigma_1} + \frac{\sigma_2}{2}\omega_2 + \frac{\partial f_4}{\partial\omega_y}\frac{\partial\omega_2}{\partial\sigma_1} + \frac{\sigma_3}{2}\omega_3 + \frac{\partial f_4}{\partial\omega_z}\frac{\partial\omega_3}{\partial\sigma_1}, \tag{A.14}$$

$$\frac{\partial f_4}{\partial\sigma_2} = -\frac{\sigma_2}{2}\omega_1 + \frac{\partial f_4}{\partial\omega_x}\frac{\partial\omega_1}{\partial\sigma_2} + \frac{\sigma_1}{2}\omega_2 + \frac{\partial f_4}{\partial\omega_y}\frac{\partial\omega_2}{\partial\sigma_2} + \frac{1}{2}\omega_3 + \frac{\partial f_4}{\partial\omega_z}\frac{\partial\omega_3}{\partial\sigma_2}, \tag{A.15}$$

$$\frac{\partial f_4}{\partial\sigma_3} = -\frac{\sigma_3}{2}\omega_1 + \frac{\partial f_4}{\partial\omega_x}\frac{\partial\omega_1}{\partial\sigma_3} - \frac{1}{2}\omega_2 + \frac{\partial f_4}{\partial\omega_y}\frac{\partial\omega_2}{\partial\sigma_3} + \frac{\sigma_1}{2}\omega_3 + \frac{\partial f_4}{\partial\omega_z}\frac{\partial\omega_3}{\partial\sigma_3}, \tag{A.16}$$

$$\frac{\partial f_5}{\partial\sigma_1} = \frac{\sigma_2}{2}\omega_1 + \frac{\partial f_5}{\partial\omega_x}\frac{\partial\omega_1}{\partial\sigma_1} - \frac{\sigma_1}{2}\omega_2 + \frac{\partial f_5}{\partial\omega_y}\frac{\partial\omega_2}{\partial\sigma_1} - \frac{1}{2}\omega_3 + \frac{\partial f_5}{\partial\omega_z}\frac{\partial\omega_3}{\partial\sigma_1}, \tag{A.17}$$

$$\frac{\partial f_5}{\partial\sigma_2} = \frac{\sigma_1}{2}\omega_1 + \frac{\partial f_5}{\partial\omega_x}\frac{\partial\omega_1}{\partial\sigma_2} + \frac{\sigma_2}{2}\omega_2 + \frac{\partial f_5}{\partial\omega_y}\frac{\partial\omega_2}{\partial\sigma_2} + \frac{\sigma_3}{2}\omega_3 + \frac{\partial f_5}{\partial\omega_z}\frac{\partial\omega_3}{\partial\sigma_2}, \tag{A.18}$$

$$\frac{\partial f_5}{\partial\sigma_3} = \frac{1}{2}\omega_1 + \frac{\partial f_5}{\partial\omega_x}\frac{\partial\omega_1}{\partial\sigma_3} - \frac{\sigma_3}{2}\omega_2 + \frac{\partial f_5}{\partial\omega_y}\frac{\partial\omega_2}{\partial\sigma_3} + \frac{\sigma_2}{2}\omega_3 + \frac{\partial f_5}{\partial\omega_z}\frac{\partial\omega_3}{\partial\sigma_3}, \tag{A.19}$$

$$\frac{\partial f_6}{\partial\sigma_1} = \frac{\sigma_3}{2}\omega_1 + \frac{\partial f_6}{\partial\omega_x}\frac{\partial\omega_1}{\partial\sigma_1} + \frac{1}{2}\omega_2 + \frac{\partial f_6}{\partial\omega_y}\frac{\partial\omega_2}{\partial\sigma_1} - \frac{\sigma_1}{2}\omega_3 + \frac{\partial f_6}{\partial\omega_z}\frac{\partial\omega_3}{\partial\sigma_1}, \tag{A.20}$$

$$\frac{\partial f_6}{\partial\sigma_2} = -\frac{1}{2}\omega_1 + \frac{\partial f_6}{\partial\omega_x}\frac{\partial\omega_1}{\partial\sigma_2} + \frac{\sigma_3}{2}\omega_2 + \frac{\partial f_6}{\partial\omega_y}\frac{\partial\omega_2}{\partial\sigma_2} - \frac{\sigma_2}{2}\omega_3 + \frac{\partial f_6}{\partial\omega_z}\frac{\partial\omega_3}{\partial\sigma_2}, \tag{A.21}$$

$$\frac{\partial f_6}{\partial\sigma_3} = \frac{\sigma_1}{2}\omega_1 + \frac{\partial f_6}{\partial\omega_x}\frac{\partial\omega_1}{\partial\sigma_3} + \frac{\sigma_2}{2}\omega_2 + \frac{\partial f_6}{\partial\omega_y}\frac{\partial\omega_2}{\partial\sigma_3} + \frac{\sigma_3}{2}\omega_3 + \frac{\partial f_6}{\partial\omega_z}\frac{\partial\omega_3}{\partial\sigma_3}. \tag{A.22}$$

LVLH coupling: partial derivatives $\partial\boldsymbol{\omega}_{\mathcal{B}/\mathcal{L}}/\partial\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{L}}$

Let n_{ref} denote the reference LVLH rotation rate. Using the MRP-to-DCM map, the LVLH angular velocity expressed in $\mathcal{B}$ yields

$$\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}/\mathcal{L}} = \boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}/\mathcal{E}} - n_{\text{ref}}\frac{1}{(2-\Sigma)^2}\begin{bmatrix} 8\sigma_1\sigma_3 - 4\sigma_2\Sigma \\ 8\sigma_2\sigma_3 + 4\sigma_1\Sigma \\ 4(-\sigma_1^2 - \sigma_2^2 + \sigma_3^2) + \Sigma^2 \end{bmatrix}, \tag{A.23}$$

and therefore the required partial derivatives are

$$\frac{\partial \omega_1}{\partial \sigma_1} = -n_{\text{ref}} \left(\frac{8\sigma_3 + 8\sigma_2\sigma_1}{(2-\Sigma)^2} - \frac{(8\sigma_1\sigma_3 - 4\sigma_2\Sigma)\,4\sigma_1}{(2-\Sigma)^3} \right), \tag{A.24}$$

$$\frac{\partial \omega_1}{\partial \sigma_2} = -n_{\text{ref}} \left(\frac{8\sigma_2^2 - 4\Sigma}{(2-\Sigma)^2} - \frac{(8\sigma_1\sigma_3 - 4\sigma_2\Sigma)\,4\sigma_2}{(2-\Sigma)^3} \right), \tag{A.25}$$

$$\frac{\partial \omega_1}{\partial \sigma_3} = -n_{\text{ref}} \left(\frac{8\sigma_1 + 8\sigma_2\sigma_3}{(2-\Sigma)^2} - \frac{(8\sigma_1\sigma_3 - 4\sigma_2\Sigma)\,4\sigma_3}{(2-\Sigma)^3} \right), \tag{A.26}$$

$$\frac{\partial \omega_2}{\partial \sigma_1} = -n_{\text{ref}} \left(\frac{4\Sigma - 8\sigma_1^2}{(2-\Sigma)^2} - \frac{(8\sigma_2\sigma_3 + 4\sigma_1\Sigma)\,4\sigma_1}{(2-\Sigma)^3} \right), \tag{A.27}$$

$$\frac{\partial \omega_2}{\partial \sigma_2} = -n_{\text{ref}} \left(\frac{8\sigma_3 - 8\sigma_1\sigma_2}{(2-\Sigma)^2} - \frac{(8\sigma_2\sigma_3 + 4\sigma_1\Sigma)\,4\sigma_2}{(2-\Sigma)^3} \right), \tag{A.28}$$

$$\frac{\partial \omega_2}{\partial \sigma_3} = -n_{\text{ref}} \left(\frac{8\sigma_2 - 8\sigma_1\sigma_3}{(2-\Sigma)^2} - \frac{(8\sigma_2\sigma_3 + 4\sigma_1\Sigma)\,4\sigma_3}{(2-\Sigma)^3} \right), \tag{A.29}$$

$$\frac{\partial \omega_3}{\partial \sigma_1} = n_{\text{ref}} \left(\frac{8\sigma_1 + 4\sigma_1\Sigma}{(2-\Sigma)^2} - \frac{\left(4(-\sigma_1^2 - \sigma_2^2 + \sigma_3^2) + \Sigma^2\right)4\sigma_1}{(2-\Sigma)^3} \right), \tag{A.30}$$

$$\frac{\partial \omega_3}{\partial \sigma_2} = n_{\text{ref}} \left(\frac{8\sigma_2 + 4\sigma_2\Sigma}{(2-\Sigma)^2} - \frac{\left(4(-\sigma_1^2 - \sigma_2^2 + \sigma_3^2) + \Sigma^2\right)4\sigma_2}{(2-\Sigma)^3} \right), \tag{A.31}$$

$$\frac{\partial \omega_3}{\partial \sigma_3} = -n_{\text{ref}} \left(\frac{8\sigma_3 - 4\sigma_3\Sigma}{(2-\Sigma)^2} - \frac{\left(4(-\sigma_1^2 - \sigma_2^2 + \sigma_3^2) + \Sigma^2\right)4\sigma_3}{(2-\Sigma)^3} \right). \tag{A.32}$$

List of Figures

List of Tables

List of Acronyms

CPO Close Proximity Operations
CWH Clohessy–Wiltshire–Hill
DCM Direction Cosine Matrix
DoF Degrees of Freedom
ECI Earth-Centered Inertial
FoV Field of View
GNC Guidance, Navigation, and Control
KKT Karush–Kuhn–Tucker
KOZ Keep-Out Zone
LEO Low Earth Orbit
LOS Line of Sight
LQ-MPC Linear-Quadratic Model Predictive Control
LTI Linear Time-Invariant
LTV Linear Time-Varying
LVLH Local Vertical Local Horizontal
MPC Model Predictive Control
MRPs Modified Rodrigues Parameters
NeRF Neural Radiance Fields
NLP Nonlinear Programming
NMPC Nonlinear Model Predictive Control
OCP Optimal Control Problem
QP Quadratic Programming
ROEs Relative Orbital Elements
RPOD Rendezvous and Proximity Operations
RSO Resident Space Object
SCP Sequential Convex Programming
SLAM Simultaneous Localization and Mapping
SQP Sequential Quadratic Programming
STM State Transition Matrix

List of Symbols

Variable	Description	SI unit
$\mathcal{E}$	Earth-Centered Inertial (ECI) reference frame	–
$\mathcal{L}$	Target LVLH reference frame	–
$\mathcal{B}$	Chaser body reference frame	–
$\mathcal{C}$	Camera reference frame	–
$\boldsymbol{r}$	Position vector	m
$\boldsymbol{v}$	Velocity vector	$\mathrm{m\,s^{-1}}$
$\boldsymbol{\rho}$	Relative position	m
r	Relative range	m
$\boldsymbol{d}$	Line of Sight unit vector	–
$\boldsymbol{b}$	Camera boresight	–
$\boldsymbol{C}_{A/B}$	Direction Cosine Matrix from B to A	–
$\boldsymbol{\sigma}_{A/B}$	Modified Rodrigues Parameters	–
μ	Earth gravitational parameter	$\mathrm{m^3\,s^{-2}}$
$R_\oplus$	Earth mean equatorial radius	m
$\boldsymbol{J}$	Inertia matrix	$\mathrm{kg\,m^2}$
$\boldsymbol{\tau}$	Torque	$\mathrm{N\,m}$
n	Mean motion of the target reference orbit	$\mathrm{rad\,s^{-1}}$
$\boldsymbol{\omega}$	Angular velocity	$\mathrm{rad\,s^{-1}}$
T_s	Sampling time	s
N	Prediction horizon length	–
θ_{fov}	Camera FoV half-angle	rad
W, H	Image width and height	px
f_x, f_y	Focal lengths in pixel units	px
N_d	Number of mapping directions	–
$\boldsymbol{q}_i$	Unit mapping direction	–
$\boldsymbol{C}$	Coverage memory vector	s

Acknowledgements

Thank you first to Politecnico di Milano, for offering me an academic path rich in challenges and opportunities, and for preparing me to pursue my goals.

A special acknowledgment goes to Professor Stefano Silvestrini for giving me the opportunity to develop this thesis, for his support throughout the work, and for enabling an international collaboration at the Florida Institute of Technology. I would also like to thank the host institution for providing the support and resources that made this experience possible.

In particular, I am deeply grateful to Professor Madhur Tiwari for his supervision and for guiding me through the topics addressed in this work, and for welcoming me not only as a visiting student, but as a member of his group.

My sincere thanks also go to all the researchers at the Autonomy Lab, especially Prenith Reddy and George Nehma, who not only provided valuable guidance but also welcomed me into their group, supported me along the way and made me feel truly part of the team.

Above all, I want to thank my family: my parents, my sister Martina, my brother Andrea, and my grandparents for their constant support. They have made it possible for me to achieve this milestone, and they have stood by me not only during this thesis and my studies, but throughout my life.