Politecnico di Torino

ThalesAlenia Space
a Thales / Leonardo company

POLITECNICO DI TORINO

Master's Degree in Aerospace Engineering

Master's Degree Thesis

Design of an MPC-based controller for formation flight in VLEO missions

Supervisors
Prof. Elisa Capello
Fabio Faliero

Candidate
Giuseppe Santarella

Academic Year 2025-2026

The research presented in this thesis was carried out in collaboration with Thales Alenia Space Italia (TAS-I), within the framework of an industrial-academic study on formation flying.

Acknowledgements

Abstract

In the last two decades, the number of launched and under-study satellite missions involving formation flying has grown, with many of them aiming to contribute to autonomous proximity operations. Meanwhile, the size of the spacecraft involved has been decreasing due to the advantages in terms of mission costs, with small spacecraft (i.e., with a mass of less than 500 kg) being favored. This choice, however, comes at the cost of increased complexity in the Guidance, Navigation and Control (GNC) system design, due to limited onboard resources (computational capacity and thrust-to-power ratio). This thesis aims to tackle these constraints, presenting the use of a Model Predictive Control (MPC) controller applied to this context. After a brief survey of formation flying, a scenario with two small spacecraft (a chief and a deputy) in LEOs is analyzed, simulating only position dynamics, starting from previous results in literature. The chief is the orbiting reference point, and its absolute dynamics are simulated using a high-fidelity orbital propagator, assuming a null ballistic coefficient to represent it as non-decaying. The deputy is the satellite controlled by the MPC, and its relative dynamics with respect to the chief are described using quasi-nonsingular Relative Orbital Elements (ROE). The controller commands three pairs of low-thrust engines, each pair always aligned with the same Radial-Transverse-Normal reference frame axis. Since the real relative dynamics are nonlinear, the MPC controller, implemented in Yalmip, relies on the discretized version of the continuous-time dynamic system model obtained from a first-order Taylor expansion around the null ROE state vector. The plant matrix and the control input matrix are taken from the literature. In particular, the plant matrix accounts for atmospheric drag and J2 perturbations, in addition to Keplerian motion. Constraints on maximum thrust and input increments are imposed in the controller, and the matrices are updated at each call of the controller function. Finally, a maneuver example is simulated in Matlab/Simulink to validate the controller and show that the required computational times are suitable for onboard implementation.

Contents

1 Introduction 5
- 1.1 Formation flying: relevance in missions . . . 7
- 1.2 Survey on literature . . . 10

2 Formation flying modeling 13
- 2.1 Passive and periodic relative trajectories . . . 15
- 2.2 Dynamics description . . . 19
- 2.3 Plant matrices computation . . . 22
  - 2.3.1 Keplerian dynamics plant matrix . . . 22
  - 2.3.2 J_2 perturbation plant matrix . . . 23
  - 2.3.3 Atmospheric drag perturbation plant matrix . . . 24
  - 2.3.4 Overall plant matrix . . . 27
  - 2.3.5 Control input matrix . . . 27

3 Model Predictive Control 29
- 3.1 MPC formulation . . . 33

4 Simulations 37
- 4.1 Exact orbital propagator scheme validation . . . 37
  - 4.1.1 Scheme structure . . . 39
  - 4.1.2 Results . . . 39
- 4.2 Simulated scenario . . . 43
  - 4.2.1 Settings and desiderata . . . 47
  - 4.2.2 Results . . . 50

5 Conclusions 57
- 5.1 Future works . . . 58

Bibliography 61

List of Acronyms

CNES Centre national d'études spatiales. 8

DARPA Defense Advanced Research Projects Agency. 8

DART Demonstration of Autonomous Rendezvous Technology. 9

DEM digital elevation model. 8

DLR Deutsches Zentrum für Luft- und Raumfahrt. 8

ECI Earth-centered inertial. 13, 25, 37, 40, 50

ESA European Space Agency. 8

FDIR fault detection, isolation and recovery. 9

FF Formation Flying. 5–10, 19, 47

GNC Guidance, Navigation and Control. 5–9

GPS Global Positioning System. 5

GRACE Gravity Recovery and Climate Experiment. 8

GSFC Goddard Space Flight Center. 5

HCW Hill-Clohessy-Wiltshire. 10, 11, 15, 19

JPL Jet Propulsion Laboratory. 8

LEO Low Earth Orbit. 5, 6, 8, 9, 11, 15, 21, 57

LISA Laser Interferometer Space Antenna. 8

LPV Linear Parameter-Varying. 34, 50, 57, 58

LTV Linear Time-Varying. 57

MIQP Mixed-Integer Quadratic Programming. 59

MMS Magnetospheric Multiscale. 9

MPC Model Predictive Control. 6, 7, 9, 12, 15, 29, 30, 33, 36, 37, 43, 46, 47, 50, 51, 54, 55, 57–59

NASA National Aeronautics and Space Administration. 5, 8, 9

OCP Optimal Control Problem. 6, 29–32, 34–36, 46–48, 54, 55, 57

PCO Projected Circular Orbits. 10

PPRT passive and periodic relative trajectories. 15–17

PRISMA Prototype Research Instruments and Space Mission Technology Advancement. 8

RHC Receding Horizon Control. 29

ROE Relative Orbital Elements. 6, 11, 12, 15, 19–21, 23, 27, 40, 43–47, 50–52, 57, 58

RTN Radial-Transverse-Normal. 13–15, 17–19, 24, 27, 35, 37, 39–43, 45, 47, 50, 51, 53, 58

SDRE State-Dependent Riccati Equation. 11

SSC Swedish Space Corporation. 8

SWIFT Silicon Wafer Integrated Femtosatellites. 8

USAF United States Air Force. 9

VLEO Very Low Earth Orbit. 12, 13, 21, 54, 57

Chapter 1

Introduction

This thesis lies within the field of Guidance, Navigation and Control (GNC) for low-mass satellites in Low Earth Orbit (LEO) (orbits whose apogee altitude is below 2000 km) flying in Formation Flying (FF), with a focus on relative position control.
The definition of FF most commonly used in the literature is that provided by National Aeronautics and Space Administration (NASA)'s Goddard Space Flight Center (GSFC): *"The tracking or maintenance of a desired relative separation, orientation or position between or among spacecraft."*
It is important to clarify that, in order to speak of FF, the control law must act on the relative position and attitude dynamics. For instance, the Global Positioning System (GPS) satellites constitute a constellation but not a formation, because the desired relative positions among them are obtained by means of a control law that acts on their absolute position dynamics.
FF can be globally represented as the temporal combination of the following three tasks:

- Formation initialization (or establishment). The satellites reach the relative positions that yield the desired geometry. At the same time, their relative attitude is adjusted to obtain the configuration required by the mission objectives (for example, the assembly of a virtual scientific instrument).

- Formation keeping. The goal is to compensate the main orbital perturbations which, if left uncontrolled, would cause a progressive deviation of the relative orbits from the desired trajectories and a consequent loss of the formation geometry.

- Formation reconfiguration. This task comprises the maneuvers required to modify the shape and the size of the formation, or the relative attitude of the satellites during the mission, so as to adapt to new operational objectives or changes in the mission profile. The transition between configurations must occur in a coordinated, safe, and fuel-efficient manner.

The growing interest in FF, discussed in Sect. 1.1, concerns primarily small satellites, i.e., spacecraft with mass below 500 kg. This implies stringent constraints on physical dimensions and available onboard power. As a consequence, it is generally not possible to employ propulsion systems capable of providing large thrust levels. In addition, these

platforms typically offer limited onboard computational capacity, increasing the complexity in designing GNC algorithms.
The reduction of propellant consumption, therefore, becomes a key requirement for performing FF in LEOs. In this orbital regime, perturbations require frequent corrections which, if not properly planned, lead to significant ΔV expenditure. Efficient propellant usage is thus critical in determining both the operational lifetime and the overall sustainability of the mission. Designing control strategies able to guarantee accurate relative motion while minimizing propellant usage is consequently a central aspect for mission success.
Another fundamental aspect of FF is the coordination among the satellites in the formation. Formation coordination methods in the literature can be classified according to the underlying control structure, including leader-follower, behavior-based, virtual structure, cyclic pursuit and algebraic graph approaches, among others [44]. This classification should be regarded as a general guideline rather than a strict taxonomy. The works [44] and [1] provide useful references for a more in-depth discussion of these methods.
This thesis focuses on the leader-follower approach. In this framework, a reference satellite, called the leader or chief, defines the nominal orbit. The followers (or deputies) are controlled based on their relative states with respect to the leader.
The scenario considered in this work therefore involves at least two satellites. The chief is assumed to be controlled from ground, whereas the other satellites play the role of deputies: they are controlled exclusively through their relative position and attitude dynamics with respect to the chief and must arrange in space to achieve a formation with the desired shape and size by coordinating among themselves.
The problem of relative position control is addressed by means of a Model Predictive Control (MPC) algorithm. The used predictive model is obtained by linearizing the nonlinear relative position dynamics expressed in terms of Relative Orbital Elements (ROE) (nonlinear functions of the classical orbital elements and formally defined later in the dissertation). The use of an MPC algorithm combines optimal control with feedback, mitigating the errors introduced by approximating the nonlinear dynamics through linearization. The resulting state and input matrices are taken from the literature. In particular, the state matrix accounts for atmospheric drag and J_2 perturbations (i.e., the main orbital perturbations in LEOs), in addition to Keplerian motion.
To conclude this introductory discussion, the structure of the thesis is briefly outlined to clarify how the presented concepts are developed in the following chapters. The remainder of this chapter highlights the growing relevance of FF in both operational and planned space missions, and reviews how the modeling of relative dynamics, as well as formation configuration and reconfiguration strategies, have been addressed in the literature. Chapter 2 introduces the general scenario in which the present work is developed, including the reference frames, state variables, and fundamental quantities involved. It also recalls a key result from the literature that is later used to validate the Simulink implementation of the plant representing the relative dynamics of a deputy spacecraft with respect to the chief. The modeling approach adopted for the plant used within the MPC algorithm is then described. Chapter 3 presents the theoretical foundations of MPC. The concept of Optimal Control Problem (OCP) is introduced, and the specific

formulation adopted in this work is detailed, including the cost function, constraints, and algorithmic workflow. Finally, Chapter 4 describes the aforementioned Simulink scheme and its validation. The results of a representative formation reconfiguration maneuver controlled through the MPC algorithm are then presented and discussed.

1.1 Formation flying: relevance in missions

Starting from the year 2000, the number of launches performed or planned worldwide for missions involving FF has increased, as shown in Fig. 1.1 [13]. This count also includes missions that only reached the preliminary design phase and were subsequently canceled.

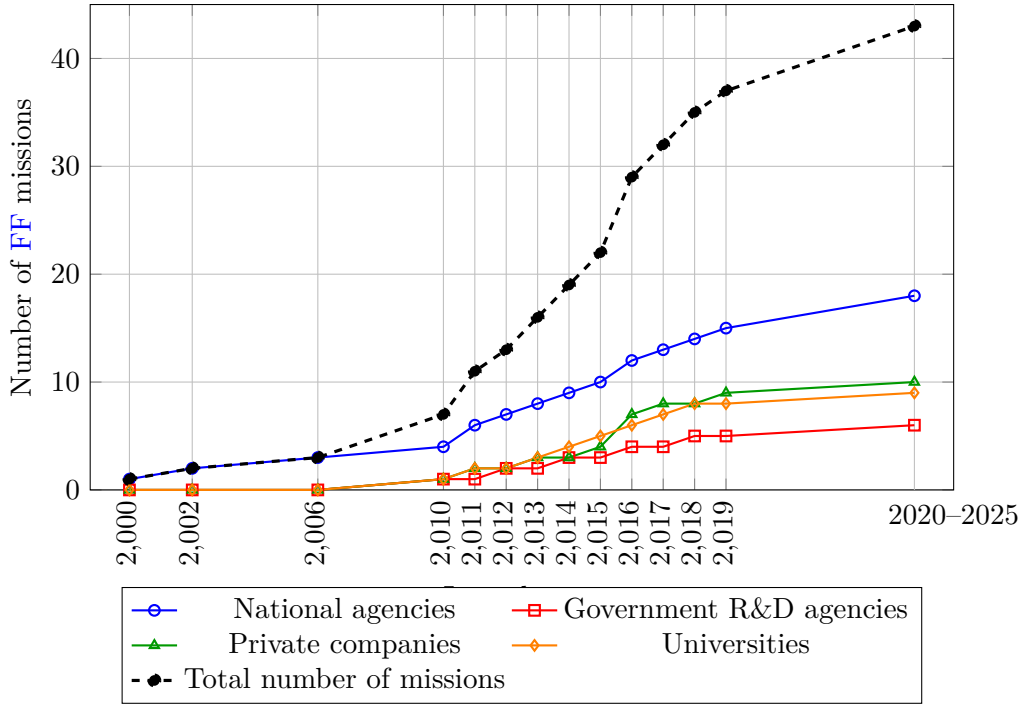


Figure 1.1: Temporal evolution since 2000 of the number of planned or proposed FF missions as a function of launch year, categorized by organization type and including the total number of missions. Data derived from [13].

As can be observed, interest in this area is shared across all types of agencies and organizations, both governmental and private. [13] provides a table listing the most significant of these missions. Moreover, it proposes a detailed classification based on three criteria: mission objective, mass of the satellites involved, and orbital regime.
Most of the missions considered are technology demonstrators, designed to test solutions developed in academic research on GNC in order to achieve a higher level of onboard autonomy, a feature that is becoming increasingly necessary in all types of FF missions (the motivation for this will emerge later in the text). However, a significant fraction is constituted by space and Earth science missions, whose number has grown over the years. Among the missions surveyed, 73% involve small satellites (i.e., with wet mass below 500

kg), whereas 24.3% involve large satellites (i.e., with wet mass equal to or greater than 1000 kg). Focusing on the orbital regimes characteristic of these missions, and considering the last five years, one may infer that the future trend will be dominated by small satellites used as technology demonstrators and Earth-observation instruments, and by large satellites employed in deep-space missions for astronomical observation.
This emerging trend is further supported by several FF mission concepts currently under study or planned for future deployment. An example of a deep-space mission under development for astronomical observation involving large spacecraft is Laser Interferometer Space Antenna (LISA) (expected launch in 2035), led by European Space Agency (ESA) with NASA contributing as a mission partner. The concept relies on three spacecraft flying in a triangular formation to enable space-based gravitational wave detection. A representative example of a small-satellite mission for Earth observation currently under development is the VULCAIN concept, coordinated by ESA and targeted for launch in 2027. The mission envisions two 12U CubeSats operating in Sun-synchronous orbit while maintaining an along-track formation to enable stereoscopic imaging of the Earth, with particular emphasis on monitoring volcanic and coastal regions. A further example, reflecting the ongoing shift toward extreme spacecraft miniaturization, is the Silicon Wafer Integrated Femtosatellites (SWIFT) mission concept, under study by the Jet Propulsion Laboratory (JPL) with support from Defense Advanced Research Projects Agency (DARPA). As a technology demonstrator mission, it explores the use of very large swarms of femtosatellites, each with a mass on the order of 100 g, operating in LEO, and aims to investigate whether distributed GNC strategies can be effectively scaled to swarms of this size, enabling robust and fault-tolerant FF through decentralized coordination rather than centralized control.
Ref. [13] can also be consulted for an overview of the most important missions launched so far, in particular with emphasis on the GNC architecture:

- Gravity Recovery and Climate Experiment (GRACE). A mission developed by NASA and Deutsches Zentrum für Luft- und Raumfahrt (DLR) and launched in 2002, employing two satellites to compute with very high accuracy the spherical harmonic coefficients of the Earth's gravitational field and to observe their temporal evolution.

- TanDEM-X and TerraSAR-X. Designed by DLR, EADS Astrium GmbH, and Infoterra GmbH and launched in 2010, this mission aims at generating a global high-precision digital elevation model (DEM).

- Prototype Research Instruments and Space Mission Technology Advancement (PRISMA). Designed by the Swedish Space Corporation (SSC) together with the Centre national d'études spatiales (CNES) and the Danish Technical University and launched in 2010, PRISMA is a demonstration mission for testing and validating GNC hardware, software, and algorithms for autonomous flight.

- Canadian Advanced Nanosatellite eXperiment-4 and -5. Designed by the Space Flight Laboratory at the University of Toronto Institute for Aerospace Studies and launched in 2014, this two-nanosatellite mission demonstrates the achievable accuracy in position control and relative position estimation.

- Magnetospheric Multiscale (MMS). Developed by NASA and launched in 2015, this mission aims to investigate magnetic reconnection, particle energization, and turbulence in the Earth's magnetosphere.

The orbital regime underlying most (73%) of the surveyed missions is LEO, which is therefore the focus of this thesis.
The motivations behind the increasing use of FF in space missions are now examined.
It is important to start from the original idea proposed in the literature in the paper by Sholomitsky et al. in 1977 [36] (which became a topic of growing interest from the 1990s onwards, when advances in small-satellite technology and in inter-satellite relative measurement and control systems made FF concretely feasible [44]): the satellites in the formation are used cooperatively to perform a function (possibly virtual, such as an infrared interferometer in [36]) that would previously have been assigned to a single, larger and more complex spacecraft.
The payload required by the monolithic satellite is distributed among the satellites, as are the different tasks that constitute the overall function of the emulated instrument.
Numerous advantages arise from this paradigm, as highlighted in [40]: improved performance of the system function, yielding higher scientific return (for example, enlarging the surveillance area in Earth observation missions), reduced launch and maintenance costs, increased flexibility in executing the required function (for instance, by modifying the relative positions of the satellites, by adding further satellites to the formation at a later stage, or by upgrading individual satellites at different times), and increased redundancy and system reliability. Indeed, the problem of having a single point of failure is mitigated: in the event of the failure of one satellite in the formation, its negative impact can be reduced by reconfiguring the formation (i.e., by reorganizing the mutual relative positions and attitudes of the remaining satellites not involved in the failure).
These advantages are, however, accompanied by limitations and challenges, which are the subject of ongoing research. One of the most significant is the quest for autonomy in LEO formations, namely the ability to manage all GNC functions independently, as opposed to relying on ground control, as discussed in [40]. Ground control requires coverage by a ground station, which imposes stringent constraints on visibility, introduces latency, and demands pre-planned maneuvers on the ground. Autonomous flight, instead, enables better performance and greater responsiveness to unexpected events.
This makes the problem more complex in terms of relative navigation (increasingly high accuracy required), actuation, maneuver planning and stability of the resulting configurations, distributed communication, and fault detection, isolation and recovery (FDIR).
This complexity has manifested itself in two emblematic real-world applications and has tempered the initial optimism surrounding multi-satellite systems [10]: NASA's Demonstration of Autonomous Rendezvous Technology (DART) mission in 2005, which ended with an unintended collision, and the United States Air Force (USAF) TechSat-21 project, canceled the same year after technical difficulties proved far greater than expected.
These events underline the crucial importance of preliminary technology demonstration missions to mature autonomous FF capabilities.
Within this broader context, the present thesis aims to contribute to the study of autonomy in FF. In particular, it investigates the use of MPC as a possible strategy for

executing the maneuvers required as a basis for the actual formation initialization and reconfiguration tasks.

1.2 Survey on literature

Having established the motivation and the increasing relevance of FF in current and future space missions, as well as the general framework and terminology adopted throughout this work, it is now appropriate to examine how the problem of relative motion control has been addressed in the literature. Among the various formation control architectures proposed in previous studies, this section focuses specifically on the leader-follower approach, which represents the foundation of the methodology developed in this thesis. In the last two decades, it has become the most widely used approach for satellite formation control. Its success lies in its intuitive structure.
Early analytical research on this topic relied on the linearized Hill-Clohessy-Wiltshire (HCW) equations, which describe the motion of one satellite relative to another moving in a circular orbit. Yeh and Sparks [43] provided a geometric interpretation of these equations, showing that all possible *force-free* relative trajectories (i.e., those requiring no control input) can be represented as the intersection between a plane and an elliptic cylinder in the local orbital frame of the leader. They defined these trajectories as *legal formation paths*, meaning relative orbits that satisfy the equations of motion naturally, without requiring active control. Such paths represent physically feasible and fuel-free configurations in which satellites can orbit cooperatively under purely gravitational dynamics. This analysis led to the definition of Projected Circular Orbits (PCO), a family of bounded, periodic relative trajectories that later became the reference model for formation keeping.
Building on this foundation, Wang and Hadaegh [39] formulated the problem of fuel-optimal formation reconfiguration as a sequence of impulsive maneuvers designed to minimize total propellant use. By combining linearized relative motion equations with an optimization method based on permutation cycles, they obtained efficient solutions for repositioning multiple satellites. Although highly fuel-efficient, their approach neglected orbital perturbations such as Earth's oblateness (the J_2 effect) and atmospheric drag, which become significant in low-altitude missions.
Palmer [32] extended this work by considering continuous low-thrust propulsion, which provides small but constant acceleration instead of instantaneous impulses. He derived analytical, energy-optimal trajectories for relative reconfiguration, demonstrating that smooth, low-thrust maneuvers can achieve similar fuel efficiency while better representing the dynamics of electric or cold-gas propulsion systems typically used on modern small satellites. In parallel, several studies aimed to improve both the accuracy of the dynamical model and the effectiveness of the control law. Kumar et al. [20] proposed a linear feedback controller that used thrust exclusively in the along-track direction to regulate the follower's relative position with respect to the leader. The control law, derived from the HCW equations under the assumption of a circular reference orbit, was designed to maintain circular or projected-circular formations with minimal control effort.

Simulation results showed that the method could keep relative position errors within approximately ±10 m, even when variations in initial conditions, small orbital eccentricities, and environmental disturbances such as atmospheric drag and solar radiation pressure were included. This demonstrated that satisfactory control performance can be achieved even when thrust capability is severely limited, as is typical for small spacecraft operating in LEOs.

Won and Ahn [41] extended the standard HCW formulation, which assumes a circular reference orbit and coplanar motion, by addressing the more general case of elliptical and non-coplanar absolute orbits for the leader and the follower. When the chief's orbit is elliptical or the orbital planes differ, the relative dynamics become inherently nonlinear and the HCW equations lose validity. To handle this complexity, the authors developed a State-Dependent Riccati Equation (SDRE) controller, a nonlinear feedback technique in which the gain matrix is recalculated at each time step based on the current system state. This approach locally linearizes the dynamics around the instantaneous operating point, enabling accurate control of the nonlinear relative motion caused by orbital eccentricity and inclination differences. The SDRE controller improved precision compared with constant-gain linear feedback laws, but its high computational cost and the absence of guaranteed global stability limited its suitability for real-time onboard implementation, especially for small satellites with restricted processing capability.

He and Han [16] later introduced a complementary approach that redefined the description of the relative motion in terms of ROE, offering a compact and physically intuitive representation of the differences between the leader's and the follower's orbital elements-namely, the semi-major axis, eccentricity, inclination, argument of perigee, and right ascension of the ascending node. This framework enables the in-plane and out-of-plane motions to be treated separately, allowing for the inclusion of long-term effects of perturbations such as the Earth's oblateness (J_2) and atmospheric drag through averaged dynamics. This work represented a key step forward in bridging analytical modeling accuracy with computational efficiency, providing a convenient foundation for modern optimal and predictive control designs.

Following these developments, Kim et al. [18] developed a hybrid optimization method based on a linearized, unperturbed two-body dynamics model to compute fuel-optimal reconfiguration maneuvers. A genetic algorithm was used to perform a global search for feasible multi-impulse solutions, while the primer vector theory, an optimal control tool derived from the calculus of variations, was applied to refine these trajectories locally and minimize the total ΔV. Although the approach achieved near-global fuel optimality, it relied on an ideal Keplerian framework without including perturbative effects, and its computational cost made onboard implementation challenging.

Cho and Park [7] later proposed an analytical solution for fuel-optimal control under a linearized, unperturbed two-body dynamics model, using the calculus of variations. Their formulation yielded a closed-form expression for the optimal continuous-thrust profile as a function of the system's fundamental matrix, eliminating the need for iterative numerical computation and reducing the overall computational load. This represented a significant step toward deriving analytical control laws for continuous-thrust formation reconfiguration within the ideal Keplerian framework.

In a subsequent study, Cho and Yu [6] addressed the limitations of linearized dynamics by applying the Udwadia–Kalaba method, an analytical framework that provides an exact solution for the equations of motion of a constrained system. In their formulation, the constraints represent the geometric relationships that must hold between the satellites to preserve the desired formation configuration, for example, maintaining a constant relative distance or a circular projected separation between the leader and the follower. The Udwadia–Kalaba approach allows the control acceleration required to satisfy these constraints to be computed explicitly, without resorting to linear approximations or small-angle assumptions. The resulting control law is fully nonlinear and remains valid even for large inter-satellite separations. The analysis is conducted under an ideal, unperturbed two-body environment and focuses on demonstrating the mathematical exactness and analytical tractability of the method, without extending the model to include external perturbations such as J_2 or atmospheric drag.

Moving beyond single-leader configurations, Ren and Sorensen [34] proposed a distributed consensus-based control architecture to coordinate formations with multiple leaders or agents. Their framework relies on local communication between neighboring satellites to estimate the desired group trajectory and achieve consensus on the collective motion. By eliminating the need for centralized control, this method enhances robustness to communication failures and improves scalability. Although initially developed for terrestrial multi-robot systems, its structure and principles have since been adapted to distributed satellite formation control.

Overall, the reviewed works reveal a clear trade-off between model fidelity, control performance (or optimality), and computational feasibility. Early impulsive-control methods [39] [32] achieved high fuel efficiency but neglected perturbations and did not capture the gradual accelerations generated by continuous low-thrust propulsion systems, such as electric and cold-gas thrusters, widely used on modern small satellites. More advanced nonlinear and hybrid approaches [41] [18] [6] improved the realism and precision of the models, in some cases incorporating perturbations such as J_2 or atmospheric drag, but at the expense of significantly higher computational effort. Linearized models expressed in ROE coordinates [16], especially when combined with optimal or predictive control frameworks, offer a balanced compromise: they preserve analytical simplicity, capture dominant perturbations through averaged models, and are suitable for real-time implementation on current onboard processors.

In this context, the present work builds upon the leader–follower paradigm and the ROE-based linearized dynamic model, extending it to the Very Low Earth Orbit (VLEO) regime, where atmospheric drag and J_2 perturbations dominate the relative dynamics. By coupling this model with a MPC strategy, the approach aligns with the current research trend toward autonomous, fuel-efficient, and computationally feasible formation control, bridging the gap between theoretical optimal control formulations and practical implementation in real missions.

Chapter 2

Formation flying modeling

The problem of describing the dynamics of the system under investigation is now addressed. The goal is to model the relative position dynamics of one or more small deputy (or follower) satellites with respect to the chief (or leader) satellite, whose trajectory is assumed as the reference for the formation. The latter results from the time propagation of an initial condition consisting of a closed Keplerian orbit with arbitrary eccentricity, under Keplerian motion with the inclusion of the J_2 perturbation. The chief is therefore assumed to be a non-decaying reference point. The intent is to contribute to laying the foundations for the realization of maneuvers required for the actual formation initialization and reconfiguration tasks in VLEO (i.e., situated at an altitude between 100 and approximately 450 km). Each deputy satellite considered in the analysis is equipped with six low-thrust engines, whose geometric arrangement will be illustrated shortly. The aim is to include in the treatment the main perturbing effects of the aforementioned orbital regime acting on the deputies' position dynamics.
For each deputy satellite considered, the objective is to visualize its relative position with respect to the chief in the Radial-Transverse-Normal (RTN) reference frame. To define the latter, it is first necessary to introduce the following quantities. $\mathbf{r}_c$ is the position vector of the chief satellite with respect to the Earth center, $\dot{\mathbf{r}}_c := d\mathbf{r}_c/dt$ is its velocity, and $\mathbf{h}_c := \mathbf{r}_c \times \dot{\mathbf{r}}_c$ is the angular momentum vector per unit mass of the chief. The unit vectors of the frame are

$$\hat{\mathbf{x}} := \frac{\mathbf{r}_c}{\underline{r}_c} \qquad \hat{\mathbf{z}} := \frac{\mathbf{h}_c}{\underline{h}_c} \qquad \hat{\mathbf{y}} := \hat{\mathbf{z}} \times \hat{\mathbf{x}} \tag{2.1}$$

where $\underline{r}_c := \|\mathbf{r}_c\|_2$ and $\underline{h}_c := \|\mathbf{h}_c\|_2$. The RTN reference frame is defined by the right-handed orthonormal triad $(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}})$ and is centered at the chief satellite. Denoting by $(\underline{x}, \underline{y}, \underline{z})$ the RTN coordinates, the relative position of the center of mass of the j-th deputy satellite with respect to the chief can therefore be expressed as $\boldsymbol{\rho}_j := \underline{x}_j\hat{\mathbf{x}} + \underline{y}_j\hat{\mathbf{y}} + \underline{z}_j\hat{\mathbf{z}}$.
A graphical overview of the quantities introduced above is provided in Fig. 2.1. The triad $(\hat{\mathbf{I}}, \hat{\mathbf{J}}, \hat{\mathbf{K}})$ represents the Earth-centered inertial (ECI) reference frame. The local triad $(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}})$ corresponds to the RTN frame defined in the previous paragraph and, at each instant, it is centered at the chief satellite (thus providing a moving, chief-centered description of the relative motion). Two deputy satellites are shown for illustrative purposes and are labeled D_1 and D_2. The blue curve represents the osculating Keplerian

orbit of the chief at the considered epoch. The red curve is an auxiliary circular orbit lying in the same orbital plane as the osculating one and passing, at the same epoch, through the chief position. The vector $\mathbf{n}_c$ is also reported and is defined as

$$\mathbf{n}_c := \hat{\mathbf{K}} \times \mathbf{h}_c, \tag{2.2}$$

i.e., it is directed along the line of nodes and points toward the ascending node of the osculating Keplerian orbit of the chief. Finally, the angles Ω_c, i_c, and θ_c shown in the figure refer to this orbit and are included to facilitate the interpretation of the orientation of its orbital plane and of the associated vectors (throughout the thesis, the subscript c denotes chief-related quantities). Ω and i denote, in general, the right ascension of the ascending node and the inclination of the osculating Keplerian orbit of a satellite, respectively, while $\theta := f + \omega$ is the (true) argument of latitude, where f is the true anomaly and ω is the argument of periapsis.

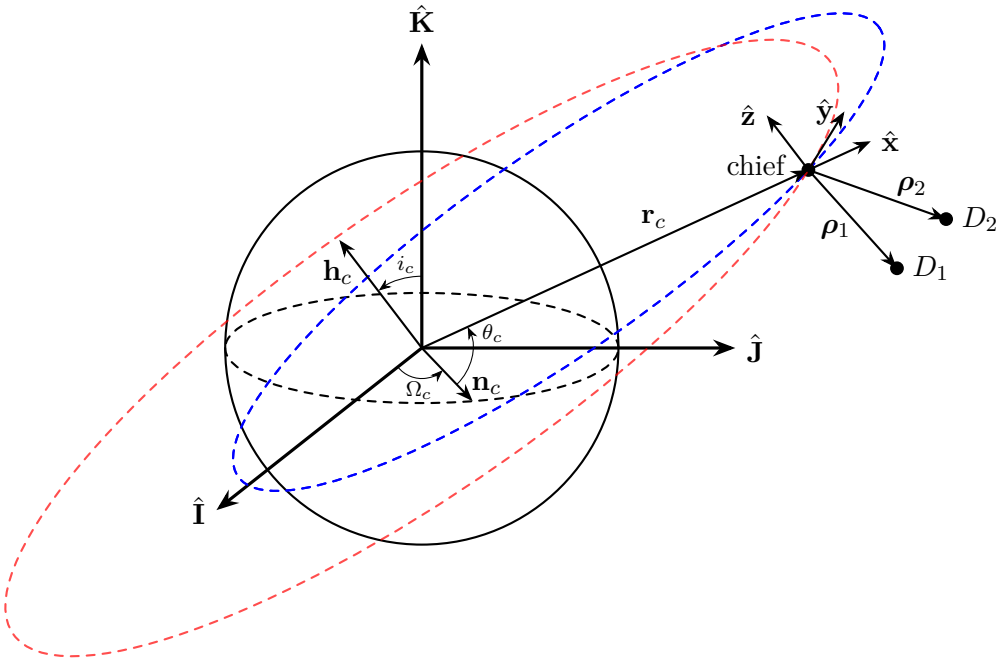


Figure 2.1: Geometrical representation of the reference frame adopted for the relative motion analysis. The chief satellite is shown at an arbitrary point along its osculating orbit (blue, dashed), with two deputies depicted at generic relative positions. The local RTN frame is attached to the chief at the considered epoch. An auxiliary circular orbit (red, dashed), lying in the same orbital plane and passing through the chief position, is included as a geometric construction to clarify the local transverse direction. The orbital angles reported in the figure provide a visual reference for the orientation of the orbital plane and for the direction of the associated vectors.

The thesis considers, for simplicity, the presence of a single deputy satellite (thus omitting the subscript j). The subscript d denotes quantities associated with the deputy. The work

focuses on verifying the adequacy of the adopted dynamic model and the suitability of the MPC algorithm for performing position control in this context. The goal is to perform a maneuver in which the shape and size of the deputy's relative orbit with respect to the chief are modified.
Before proceeding with the analysis, it is important for the reader to keep in mind the following concept, as it clarifies the rationale underlying the present work. When a spacecraft is modeled as a rigid body rather than as a point mass, its orbital position and attitude dynamics become intrinsically coupled (for further details, see [12]). In addition to this physical coupling, the action of the thrusters generally introduces an actuator-induced coupling, since any thrust command produces both a net force and a net torque unless a specific geometric configuration of thruster locations is adopted. In the present analysis, it is assumed that the six thrusters on board are arranged, in pairs, along the three principal inertia axes of the deputy satellite, with opposite directions. For simplicity, it is assumed that the attitude of this satellite is time-invariant in the RTN reference frame, and in particular that its principal inertia axes are parallel to the frame axes. Only the deputy's position dynamics are therefore considered, as the attitude dynamics have no degrees of freedom. It follows that the two dynamics are decoupled. In practice, this is equivalent to treating the deputy as a point mass. The intention is therefore to emphasize that the purpose of this work is not to obtain realistic results, but to help highlight the suitability of using MPC coupled with the chosen dynamic model to meet the needs outlined so far, from the perspective of future research.
With regard to the adopted dynamic model, reference [19] provides a comprehensive overview of the developments that have led to the formulation of the ROE (here defined in section 2.2), originating from the HCW equations. The work also harmonizes the various contributions in the literature, showing how they can be framed within a unified and coherent formulation.

2.1 Passive and periodic relative trajectories

Before illustrating the dynamic model adopted in this work, it is useful to mention the contribution of [40, Chap. 3]. The authors consider a relative position dynamics model of a generic deputy satellite with respect to the chief obtained through a linearization around a reference orbit of arbitrary eccentricity, in which the J_2 perturbation is included, and analytically derive the so-called passive and periodic relative trajectories (PPRT). In other words, there exist conditions, expressed in terms of orbital elements, that cancel the perturbative effect of J_2 on the relative position dynamics, thus making the use of control inputs unnecessary to obtain closed and periodic relative orbits.
In the scenario analyzed in the thesis, this is extremely useful for formation configuration mission planning. Indeed, as will be discussed in detail in section 2.2, the main orbital perturbations in LEOs are J_2 and aerodynamic drag. Their effect must be counteracted through the use of thrusters. Employing these relative orbits allows, as mentioned, the cancellation of the J_2 contribution to the relative dynamics, thereby saving fuel and enabling the analysis to focus exclusively on the impact of aerodynamic drag.

The necessary conditions for obtaining PPRT are now illustrated. Consider the six classical Keplerian orbital elements of a generic satellite: the semimajor axis a, the eccentricity e, the inclination i, the right ascension of the ascending node Ω, the argument of periapsis ω and the mean anomaly M. The second-order zonal harmonic of the geopotential (J_2) causes secular and periodic effects on each satellite's dynamics. In [40], as in the present work, only the first-order secular drift of the orbital elements is considered. In particular, in their works, Liu and Alford [21] [22] calculated the mean rates of variation of the classical orbital elements due to J_2 and J_3. If J_3 related terms are neglected, the following relations hold:

$$\begin{aligned}
&\frac{d\bar{a}}{dt} = 0, \quad \frac{d\bar{e}}{dt} = 0, \quad \frac{d\bar{i}}{dt} = 0, \\
&\frac{d\bar{\Omega}}{dt} = -\frac{3}{2}\bar{n}J_2\left(\frac{R_E}{\bar{p}}\right)^2 \cos(\bar{i}), \\
&\frac{d\bar{\omega}}{dt} = \frac{3}{4}\bar{n}J_2\left(\frac{R_E}{\bar{p}}\right)^2 \left[4 - 5\sin^2(\bar{i})\right], \\
&\frac{d\bar{M}}{dt} = \frac{3}{2}\bar{n}J_2\left(\frac{R_E}{\bar{p}}\right)^2 \left[1 - \frac{3}{2}\sin^2(\bar{i})\right]\sqrt{1-\bar{e}^2}
\end{aligned} \tag{2.3}$$

where the over bars denote the secular part of a variable, R_E is the Earth equatorial radius, p is the semilatus rectum of the considered satellite, and $n := \sqrt{\mu/a^3}$ is its mean motion (μ is the Earth gravitational constant).
Considering a chief satellite subject to the same dynamic contributions assumed in this thesis, and analyzing the effect of Eqs. (2.3) on the deputy relative position dynamics, one may conclude that obtaining PPRT requires the imposition of the so-called strict J_2 invariant trajectory conditions. For a generic variable l, define $\Delta l := l_d - l_c$. The strict J_2 invariant trajectory conditions can then be written as

$$\begin{cases} \Delta\bar{a} = 0 \\ \Delta\bar{e} = 0 \\ \Delta\bar{i} = 0 \end{cases} \tag{2.4}$$

These do not constitute a sufficient condition for PPRT, since they limit the tendency of the two spacecraft to drift apart, but the deputy satellite may still undergo tumbling motion in its relative trajectory and therefore requires active compensation to preserve the periodicity of the trajectory. Additional conditions are therefore required to obtain PPRT, as illustrated below.
There exist two classes of PPRT:

- First-class PPRT. They are obtained by imposing the conditions in Eqs. (2.4) together with

$$\bar{i} = i_c \tag{2.5}$$

  where $i_c = 63.435°$ or $116.565°$ is the critical inclination and corresponds to the orbital inclination that cancels the rotation of the line of apsides caused by the J_2 perturbation.

- Second-class PPRT. They are obtained by imposing the conditions in Eqs. (2.4) together with either $\bar{i} = 0$ or $\Delta\bar{\Omega} = 0$.

It is also important, for the purposes of this thesis, to mention the so-called quasi-periodic relative orbit conditions at the critical inclination, explored by the authors of [40] with the aim of obtaining a higher number of degrees of freedom allowed to design the satellite formation geometry (namely, four) compared to those provided by the first-class PPRT (namely, three). Defining $\eta := \sqrt{1-e^2}$, the quasi-periodic relative orbit conditions at the critical inclination are Eq. (2.5) together with:

$$\begin{cases} \Delta\mathcal{E} = 0 \\ \Delta\mathcal{H} = 0 \end{cases} \tag{2.6}$$

where

$$\mathcal{E} := -\frac{\mu}{2\bar{a}} + \frac{J_2 \mu R_E^2}{4\bar{a}^3\bar{\eta}^3}\left(1 - 3\cos^2(\bar{i})\right) \tag{2.7}$$

is the orbital energy, and

$$\mathcal{H} := \cos(\bar{i})\,\bar{\eta}\sqrt{\mu\bar{a}} \tag{2.8}$$

is the polar component of the orbital angular momentum. It is useful to consult [40] to explore further the effects of these conditions in terms of the relationships between the chief's and deputy's orbital elements and to visualize the geometric characteristics of the different types of relative orbits briefly illustrated so far.
In addition to constituting an important result for satellite formation geometry design, the conditions in Eqs. (2.6) are used in the present work to verify that the Simulink scheme containing the exact propagator of the position dynamics, discussed in chapter 4, has been correctly implemented. To do so, the procedure in [40, Subsec. 3.3.4] is followed. The authors verify, through the use of an exact propagator for relative satellite motion that accounts for orbital eccentricity, nonlinear dynamics, and the effects of the J_2 perturbation, that the trajectories obtained by imposing the conditions in Eqs. (2.6) are indeed quasi-periodic relative orbits, and they represent them in the RTN reference frame. To this end, the following initial conditions are first arbitrarily set:

- the orbital elements of the chief satellite;
- the position coordinates of the deputy satellite in the RTN reference frame; and
- one of the velocity components of the deputy satellite in the RTN reference frame, which is set to zero.

The remaining two components of the deputy's initial relative velocity are set by imposing the conditions in Eqs. (2.6). In particular, these are transformed into physically equivalent equations expressed in position coordinates and velocity components in the RTN frame. To illustrate them, it is necessary to define the following quantities

$$\mathbf{r}_d := \mathbf{r}_c + \boldsymbol{\rho} = (\underline{x} + \underline{r}_c)\hat{\mathbf{x}} + \underline{y}\hat{\mathbf{y}} + \underline{z}\hat{\mathbf{z}} \tag{2.9}$$

$$\dot{\mathbf{r}}_d := \underline{v}_x\hat{\mathbf{x}} + \underline{v}_y\hat{\mathbf{y}} + \underline{v}_z\hat{\mathbf{z}} \tag{2.10}$$

where $\mathbf{r}_d$ and $\dot{\mathbf{r}}_d$ are the absolute position and the absolute velocity of the deputy satellite, respectively. The components of $\dot{\mathbf{r}}_d$ appearing in Eq. (2.10) are defined as:

$$\begin{aligned} \underline{v}_x &:= \dot{\underline{x}} + \dot{\underline{r}}_c - \underline{y}\omega_z \\ \underline{v}_y &:= \dot{\underline{y}} + (\underline{r}_c + \underline{x})\omega_z - \underline{z}\omega_x \\ \underline{v}_z &:= \dot{\underline{z}} + \underline{y}\omega_x \end{aligned} \tag{2.11}$$

The quantities ω_x and ω_z are the components of the angular velocity of the RTN reference frame, i.e., such that

$$\boldsymbol{\omega} := \omega_x \hat{\mathbf{x}} + \omega_z \hat{\mathbf{z}} \tag{2.12}$$

and can be defined as follows

$$\omega_x := -\frac{k_{J2}\sin(2i_c)\sin(\theta_c)}{\underline{h}_c \underline{r}_c^3}, \qquad \omega_z := \frac{\underline{h}_c}{\underline{r}_c^2} \tag{2.13}$$

where $k_{J2} := \frac{3}{2} J_2 \mu R_E^2$. The quantity $\dot{\underline{r}}_c$ can be conveniently expressed as:

$$\dot{\underline{r}}_c = \sqrt{\|\dot{\mathbf{r}}_c\|_2^2 - \underline{h}_c^2/\underline{r}_c^2} \tag{2.14}$$

Defining

$$\begin{aligned} \underline{r}_d &:= \|\mathbf{r}_d\|_2 \\ \underline{r}_{dZ} &:= (\underline{r}_c + \underline{x})\sin(i_c)\sin(\theta_c) + \underline{y}\sin(i_c)\cos(\theta_c) + \underline{z}\cos(i_c) \end{aligned}$$

the authors show that the conditions in Eqs. (2.6) can be equivalently written as the following two equations:

$$\underline{v}_x^2 + \underline{v}_y^2 + \underline{v}_z^2 = \dot{\underline{r}}_c^2 + \frac{\underline{h}_c^2}{\underline{r}_c^2} + 2\mu\left(\frac{1}{\underline{r}_d} - \frac{1}{\underline{r}_c}\right) + 2k_{J_2}\left(\frac{1}{3\underline{r}_d^3} - \frac{1}{3\underline{r}_c^3} - \frac{\underline{r}_{dZ}^2}{\underline{r}_d^5} + \frac{\sin^2(\theta_c)\sin^2(i_c)}{\underline{r}_c^3}\right) \tag{2.15}$$

and

$$\begin{aligned} &\underline{v}_x\Big(\underline{z}\cos(\theta_c)\sin(i_c) - \underline{y}\cos(i_c)\Big) + \underline{v}_y\Big((\underline{x} + \underline{r}_c)\cos(i_c) - \underline{z}\sin(\theta_c)\sin(i_c)\Big) \\ &+ \underline{v}_z\Big(\underline{y}\sin(\theta_c) - (\underline{x} + \underline{r}_c)\cos(\theta_c)\Big)\sin(i_c) = \underline{h}_c\cos(i_c). \end{aligned} \tag{2.16}$$

Thus, the two aforementioned remaining components of the deputy's initial relative velocity are set by imposing conditions given by Eqs. (2.15) and (2.16).
To validate the Simulink scheme containing the exact position-dynamics propagator implemented in the thesis, the trajectory obtained by propagating the initial conditions described above with this scheme is compared with the trajectory obtained by using the same initial conditions with the propagator in [40], representing both in the RTN reference frame. Section 4.1 provides further details on the structure of the Simulink scheme and presents the results of the procedure just described.

2.2 Dynamics description

Deputy relative position dynamics with respect to the chief are described using ROE [19]. They are nonlinear functions of the six classical Keplerian orbital elements. Three versions of them exist in literature: singular, quasi-nonsingular and nonsingular. The differences between these three versions are here briefly described:

- singular ROE: not unique if either satellite orbit is circular (due to relative impossibility of defining ω) or equatorial (due to relative impossibility of defining ω and Ω).
- quasi-nonsingular ROE, not unique only in case the orbit of the deputy is equatorial (due to relative impossibility of defining Ω).
- nonsingular ROE: always uniquely defined.

Quasi-nonsingular ROE are the most frequently reported in the literature, due to the advantages that come from their use. Notably, the components of this state align with the integration constants of the HCW equations for near-circular orbits, and with the Tschauner-Hempel equations for eccentric orbits [9]. Moreover, quasi-nonsingular ROE provide a clear and intuitive framework to assess passive safety in FF, by exploiting the separation resulting from parallel (or anti-parallel) alignment of relative eccentricity and inclination vectors [11]. This version of ROE was originally introduced in [10]. This reference is important because the author illustrates the direct correlation between these elements and the resulting relative-orbit shape in the RTN reference frame, for near-circular reference orbits. Furthermore, he uses them to compute optimal impulsive-thrust maneuvers for formation keeping (taking into account J_2 and aerodynamic drag perturbations), with the objective of minimizing fuel consumption. This aspect is especially relevant since scientific instruments onboard satellites are sensitive to accelerations and must therefore operate when actuators are inactive. Consequently, impulsive thrusts must be scheduled at suitable intervals to counteract perturbations while ensuring acceleration-free operational windows during which the payloads can function properly. As a result, [10] serves as a starting point for generalizing the approach to reference orbits with arbitrary eccentricity.
Quasi-nonsingular ROE are valuable not only due to their unique definition for non-circular orbits, but also because, depending on the type of perturbation considered, they can enable large satellite separations, as they depend on the direct influence of these perturbations on the orbital elements. Therefore, they are preferable to the HCW equations [8].
This ROE state vector is defined starting from the the following vector of orbital elements

$$\alpha = \begin{pmatrix} a \\ u \\ e_x \\ e_y \\ i \\ \Omega \end{pmatrix} = \begin{pmatrix} a \\ \omega + M \\ e\cos(\omega) \\ e\sin(\omega) \\ i \\ \Omega \end{pmatrix} \tag{2.17}$$

where u is the mean argument of latitude, and e_x and e_y are the components of the eccentricity vector $\boldsymbol{e}$. Quasi-nonsingular ROE are then expressed as

$$\delta\alpha = \begin{pmatrix} \delta a \\ \delta\lambda \\ \delta e_x \\ \delta e_y \\ \delta i_x \\ \delta i_y \end{pmatrix} = \begin{pmatrix} (a_d - a_c)/a_c \\ u_d - u_c + (\Omega_d - \Omega_c)\cos(i_c) \\ e_{xd} - e_{xc} \\ e_{yd} - e_{yc} \\ i_d - i_c \\ (\Omega_d - \Omega_c)\sin(i_c) \end{pmatrix} \tag{2.18}$$

where $\delta\lambda$ is called relative mean longitude. The time derivative of this ROE state can be written as

$$\dot{\delta\alpha} = \begin{pmatrix} \dot{\delta a} \\ \dot{\delta\lambda} \\ \dot{\delta e_x} \\ \dot{\delta e_y} \\ \dot{\delta i_x} \\ \dot{\delta i_y} \end{pmatrix} = \begin{pmatrix} (\dot{a_d} - \dot{a_c})/a_c - \dot{a_c}\,\delta a/a_c \\ \dot{u_d} - \dot{u_c} + (\dot{\Omega_d} - \dot{\Omega_c})\cos(i_c) - (\dot{i_c})\delta i_y \\ \dot{e_{xd}} - \dot{e_{xc}} \\ \dot{e_{yd}} - \dot{e_{yc}} \\ (\dot{i_d}) - (\dot{i_c}) \\ (\dot{\Omega_d} - \dot{\Omega_c})\sin(i_c) + (\dot{i_c})\delta i_y/\tan(i_c) \end{pmatrix} \tag{2.19}$$

The nature of dynamic contributions that cause these variations in time can be:

- conservative, i.e. $\dot{\delta\alpha} = \dot{\delta\alpha}(\delta\alpha)$; or
- non-conservative, i.e. $\dot{\delta\alpha} = \dot{\delta\alpha}(\delta\alpha, \gamma)$, where γ stands for a generic set of parameters that must be considered for the perturbations taken into account in the relative dynamics model (e.g. third-body ephemerides, atmospheric data, pointing vector to the sun, ...).

For the purpose of obtaining the plant matrix, these relations must be made explicit and linearized. To do that, one must express the deputy orbital elements as $\alpha_d = \alpha_d(\delta\alpha\ ,\ \alpha_c)$:

$$\begin{aligned} a_d &= a_c\,(1 + \delta a)\,, \\ e_{x_d} &= e_{x_c} + \delta e_x, \\ e_{y_d} &= e_{y_c} + \delta e_y, \\ i_d &= i_c + \delta i_x, \\ \Omega_d &= \Omega_c + \delta i_y/\sin(i_c), \\ u_d &= u_c - \delta i_y/\tan(i_c) + \delta\lambda. \end{aligned} \tag{2.20}$$

In this work, satellites flying in VLEO are considered. However, the adopted dynamical model is, in general, applicable to satellites operating in LEO. An overview of the relative impact of the main perturbing effects on the differential dynamics of satellite formations in LEO can be found in [10, Sec. 2.3], where the orders of magnitude of the differential accelerations induced by several perturbations are discussed as a function of the inter-satellite separation, together with the assumptions underlying their computation. The analysis presented therein refers to satellites flying in near-circular orbits; in the present work, as discussed above, the chief satellite is instead allowed to follow an orbit of arbitrary eccentricity. Nevertheless, the qualitative conclusions regarding the dominant role of the Earth's non-sphericity (primarily the J_2 term) and atmospheric drag in LEO are expected to remain valid. Owing to the presence of aerodynamic drag, whose differential effect is inherently time-varying along the orbit, the time derivative of the ROE state cannot be described solely in terms of $\delta\alpha$, and the inclusion of the differential ballistic coefficient becomes necessary. The underlying reason for this requirement is discussed in the following.

The ballistic coefficient of a spacecraft is defined as $B := C_D S/m$, where C_D is the satellite drag coefficient, S is the satellite cross-section area and m is the satellite mass. In the following dissertation, C_D, S and m of each spacecraft are treated as constants, therefore ballistic coefficients too. As a consequence, influence of J_2 and atmospheric drag perturbations can be expressed as

$$\dot{\delta\alpha}(\delta\alpha, \Delta B) = \begin{pmatrix} \left(\dot{a_d}(\delta\alpha, \Delta B) - \dot{a_c}\right)/a_c - \dot{a_c}\,\delta a/a_c \\ \dot{u_d}(\delta\alpha, \Delta B) - \dot{u_c} + \left(\dot{\Omega}_d(\delta\alpha, \Delta B) - \dot{\Omega}_c\right)\cos(i_c) - (\dot{i_c})\delta i_y \\ \dot{e_{x_d}}(\delta\alpha, \Delta B) - \dot{e_{x_c}} \\ \dot{e_{y_d}}(\delta\alpha, \Delta B) - \dot{e_{y_c}} \\ \left(\dot{i_d}\right)(\delta\alpha, \Delta B) - (\dot{i_c}) \\ \left(\dot{\Omega}_d(\delta\alpha, \Delta B) - \dot{\Omega}_c\right)\sin(i_c) + (\dot{i_c})\delta i_y/\tan(i_c) \end{pmatrix} \tag{2.21}$$

where $\Delta B := B_d - B_c$ is the differential ballistic coefficient. In these equations, the dependence of the time derivatives of the chief and deputy orbital elements on α_c and B_c are implied, i.e. one should write rigorously that $\dot{\alpha_d} = \dot{\alpha_d}(\delta\alpha, \Delta B, \alpha_c, B_c)$ and $\dot{\alpha_c} = \dot{\alpha_c}(\alpha_c, B_c)$.

Let $0_{q\times s}$ denote the null array with q rows and s columns, where q and s are arbitrary positive integers. Linearizing Eq. (2.21) around $\delta\alpha = 0_{6\times 1}$ yields

$$\dot{\delta\alpha}(\delta\alpha, \Delta B) \approx \dot{\delta\alpha}(0_{6\times 1}, \Delta B) + \left.\frac{\partial\dot{\delta\alpha}}{\partial\delta\alpha}\right|_{\delta\alpha\,=\,0_{6\times 1}} \cdot (\delta\alpha - 0_{6\times 1}) \tag{2.22}$$

As shown earlier, ROE state time depends linearly on ΔB. This means it is possible to define an augmented version of ROE state, namely

$$\delta\widetilde{\alpha} = \begin{pmatrix} \delta\alpha \\ \Delta B \end{pmatrix} \tag{2.23}$$

so that Taylor first-order expansion becomes

$$\dot{\delta\widetilde{\alpha}}(\delta\widetilde{\alpha}) \approx \left.\frac{\partial\dot{\delta\widetilde{\alpha}}}{\partial\delta\widetilde{\alpha}}\right|_{\delta\tilde{\alpha}\,=\,0_{7\times1}} \cdot \delta\widetilde{\alpha} \tag{2.24}$$

in which

$$\left.\frac{\partial\dot{\delta\widetilde{\alpha}}}{\partial\delta\widetilde{\alpha}}\right|_{\delta\tilde{\alpha}\,=\,0_{7\times1}} = \left[\frac{\partial\dot{\delta\widetilde{\alpha}}}{\partial\delta a} \;\; \frac{\partial\dot{\delta\widetilde{\alpha}}}{\partial\delta\lambda} \;\; \frac{\partial\dot{\delta\widetilde{\alpha}}}{\partial\delta e_x} \;\; \frac{\partial\dot{\delta\widetilde{\alpha}}}{\partial\delta e_y} \;\; \frac{\partial\dot{\delta\widetilde{\alpha}}}{\partial\delta i_x} \;\; \frac{\partial\dot{\delta\widetilde{\alpha}}}{\partial\delta i_y} \;\; \frac{\partial\dot{\delta\widetilde{\alpha}}}{\partial\Delta B}\right]\Bigg|_{\delta\tilde{\alpha}\,=\,0_{7\times1}} \tag{2.25}$$

2.3 Plant matrices computation

For the purpose of obtaining plant matrices, one must analyze individually each contribution involved in the dynamics model. Equations expressing absolute time derivatives of the orbital elements of each satellite are substituted in Eq. (2.21), with Eqs. (2.20), too. Using the subscript "i" to denote the generic dynamic contribution, one can write:

$$A_i(\alpha_c) = \left.\frac{\partial\dot{\delta\widetilde{\alpha}}_i}{\partial\delta\widetilde{\alpha}}\right|_{\delta\tilde{\alpha}\,=\,0_{7\times1}} \tag{2.26}$$

This linearized dynamics plant matrix can be used without losing accuracy only in situations where the relative orbital elements appearing in the nonzero higher-order terms of the Taylor expansion are small.
Each dynamic contribution model is independent from each other, therefore one can apply superposition principle and the overall plant matrix can be obtained:

$$A(\alpha_c) = \sum_i A_i(\alpha_c) \tag{2.27}$$

Therefore linearized system dynamics with null control input will be described by:

$$\dot{\delta\widetilde{\alpha}}(\alpha_c) = A(\alpha_c)\delta\widetilde{\alpha} \tag{2.28}$$

2.3.1 Keplerian dynamics plant matrix

The derivatives in time of the orbital elements for each satellite are characterized by:

$$\dot{a} = \dot{e} = \dot{i} = \dot{\Omega} = \dot{\omega} = 0, \;\; \dot{M} = n = \frac{\sqrt{\mu}}{a^{3/2}} \tag{2.29}$$

Substituting these quantities in Eq. (2.19), one obtains [19]

$$\dot{\delta\widetilde{\alpha}} = \begin{pmatrix} 0 \\ \dot{M}_d - \dot{M}_c \\ 0_{5\times1} \end{pmatrix} = \sqrt{\mu}\begin{pmatrix} 0 \\ a_d^{-3/2} - a_c^{-3/2} \\ 0_{5\times1} \end{pmatrix} \tag{2.30}$$

Replacing deputy orbital elements with chief ones and ROE using Eqs. (2.20) and Taylor expanding Eq. (2.30) about $\delta\widetilde{\alpha} = 0_{7\times1}$, Keplerian plant matrix A_{Kep} can be obtained:

$$A_{\text{Kep}}(\alpha_c) = \left[\begin{array}{c:c} \begin{matrix} 0 \\ -1.5n_c \end{matrix} & 0_{2\times6} \\ \hdashline 0_{5\times1} & 0_{5\times6} \end{array}\right] \tag{2.31}$$

Only the semimajor axis appears in Eq. (2.30), therefore only δa will appear in the nonzero higher-order terms in the first-order Taylor expansion. Hence, this plant matrix is valid for small δa and arbitrary values of other components of $\delta\widetilde{\alpha}$.

2.3.2 J_2 perturbation plant matrix

The quantities in Eqs. (2.3) may be approximated by their osculating counterparts for the sake of simplicity, also because the influence of atmospheric drag on the relative dynamics (modeled in subsection 2.3.3) depends on the osculating state vector. With this premise, substituting these quantities into Eq. (2.19) yields the following expressions:

$$\delta\dot{\widetilde{\alpha}} = \kappa_{J_2 d}\begin{pmatrix} 0 \\ \eta_d\left[3\cos^2(i_d)-1\right] + 5\cos^2(i_d) - 1 - 2\cos(i_d)\cos(i_c) \\ -e_d\sin(\omega_d)\left[5\cos^2(i_d)-1\right] \\ e_d\left[5cos^2(i_d)-1\right]\cos(\omega_d) \\ 0 \\ -2\cos(i_d)\sin(i_c) \\ 0 \end{pmatrix}$$

$$-\kappa_{J_2 c}\begin{pmatrix} 0 \\ (1+\eta_c)\left[3\cos^2(i_c)-1\right] \\ -e_c\sin(\omega_c)\left[5\cos^2(i_c)-1\right] \\ e_c\left[5cos^2(i_d)-1\right]\cos(\omega_c) \\ 0 \\ -2\cos(i_c)\sin(i_c) \\ 0 \end{pmatrix} \tag{2.32}$$

By replacing the deputy orbital elements with Eqs. (2.20) and applying a Taylor expansion to Eq. (2.30) around $\delta\widetilde{\alpha} = 0_{7\times1}$, the J_2 contribution can be linearly approximated by

means of the following matrix [14]:

$$A_{J_2}(\alpha_c) = \kappa_{J_2 c} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{7}{2}EP & 0 & e_{x_c}GFP & e_{y_c}GFP & -FS & 0 & 0 \\ \frac{7}{2}e_{y_c}Q & 0 & -4e_{x_c}e_{y_c}GQ & -(1+4Ge_{y_c}^2)Q & 5e_{y_c}S & 0 & 0 \\ -\frac{7}{2}e_{x_c}Q & 0 & (1+4Ge_{x_c}^2)Q & 4e_{x_c}e_{y_c}GQ & -5e_{x_c}S & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{7}{2}S & 0 & -4e_{x_c}GS & -4e_{y_c}GS & 2T & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.33}$$

where the terms in Eqs. (2.3) and (2.33) have the following definitions:

$$\kappa_{J_2} := \frac{3}{4}\frac{J_2 R_E^2 \sqrt{\mu}}{a^{7/2}\eta^4}, \quad E := 1+\eta_c, \quad F := 4+3\eta_c, \quad G := \frac{1}{\eta_c^2}, \tag{2.34}$$
$$P := 3\cos^2(i_c) - 1, \quad Q := 5\cos^2(i_c) - 1, \quad S := \sin(2i_c), \quad T := \sin^2(i_c)$$

Eqs. (2.32) clearly show that Ω and M do not affect the state time derivative, unlike a, e, i and ω. Therefore the use of A_{J_2} does not impose constraints on $\delta\Omega$ and δM values. In [19], a correspondence is shown between these two quantities and δi_y and $\delta\lambda$, respectively. This implies the matrix applicability for arbitrary values of δi_y and $\delta\lambda$ and small values of δa, δe_x, δe_y, δi_x and ΔB.

2.3.3 Atmospheric drag perturbation plant matrix

The influence of atmospheric drag on the system dynamics has been widely explored in literature, e.g. in [19], in which a Density-Model-Specific Differential Drag plant matrix and a Density-Model-Free Differential Drag plant matrix are obtained.
The former approach explicitly relies on the Harris–Priester atmospheric density model and yields reliable results only for perigee altitudes between 200 and 900 km and for orbital eccentricities in the range 0.1 to 0.9. Since the present analysis considers a reference orbit with arbitrary eccentricity, this formulation is not suitable for the scope of this work.
The latter approach is applicable to orbits of arbitrary eccentricity and does not require an explicit ballistic-coefficient-based parameterization within the state equations. In this formulation, the effect of differential drag on the relative dynamics is captured by augmenting the state vector with the time derivatives of three differential orbital quantities. The values of these derivatives are obtained through an initialization phase, during which simulations are performed to reproduce navigation data representative of a real mission scenario. As a result, the model remains effective even in the presence of periodic variations of the ballistic coefficient of either spacecraft (assumed constant in this thesis), for instance due to deputy attitude maneuvers required to maintain alignment with the RTN frame, since such variations are implicitly embedded in the estimated drift terms

over the initialization window. Despite its generality, the Density-Model-Free formulation entails an increased state dimension and a more complex model structure. In the present work, a simpler plant matrix representation is preferred, as it provides adequate accuracy for the considered application while avoiding the need for state augmentation. For this reason, the Density-Model-Free approach is not adopted here, although it remains a valid alternative for modeling differential atmospheric drag effects in similar mission scenarios. To derive the matrix used in this thesis, a modified version of the procedure suggested by [4] has been followed. The following assumptions are made to derive the matrix:

- the spacecraft velocity relative to the atmosphere equals its velocity in the ECI reference frame. Therefore, acceleration on it caused by drag is tangent to its trajectory.

- the ballistic coefficient of the chief satellite is set to zero, in order to represent it as non-decaying. This implies $\Delta B = B_d$.

- the spacecraft separation is small enough to imply $\rho_c = \rho_d$, where ρ is the atmosphere density.

Gauss's Variational Equations [2] are used to describe drag effects on the dynamics. In particular, considering only the component of the acceleration along the spacecraft velocity vector, one has:

$$\begin{aligned}
\frac{da}{dt} &= \frac{2a^2 v}{\mu} a_{dt} \\
\frac{de}{dt} &= \frac{1}{v} 2\Big[e + \cos(f)\Big] a_{dt} \\
\frac{di}{dt} &= 0, \quad \frac{d\Omega}{dt} = 0 \\
\frac{d\omega}{dt} &= \frac{1}{ev} 2 \sin(f) a_{dt} \\
\frac{dM}{dt} &= -\frac{b}{eav} 2 \left(1 + \frac{e^2 r}{p}\right) \sin(f) a_{dt}
\end{aligned} \tag{2.35}$$

where a_{dt} is the component of the disturbing acceleration along the spacecraft velocity vector and b is the semiminor axis of the osculating orbit.
Due to the adopted assumptions, the drag effects are considered only for the deputy spacecraft. Hence, one has

$$\mathbf{a_{dt}} = -\frac{1}{2} B_d \rho_d v_d^2 \, \mathbf{i_t} \tag{2.36}$$

Following nomenclature used in [4], drag induced acceleration u_{drag} is defined as $u_{drag} := \|\mathbf{a_{dt}}\|_2 = \frac{1}{2} B_d \rho_d v_d^2$, hence $a_{dt} = -u_{drag}$.

By substituting Eqs. (2.35) and (2.36) in Eq. (2.21), one obtains:

$$\dot{\delta\widetilde{\alpha}} = \begin{pmatrix} -\frac{2a_d^2 v_d}{\mu a_c} u_{drag} \\ -\frac{2}{e_d v_d}\sin(f_d)\left[1 - \frac{b_d}{a_d}\left(1 + \frac{e_d^2 r_d}{p_d}\right)\right] u_{drag} \\ -\frac{2}{v_d}\left[\left(e_d + \cos(f_d)\right)\cos(\omega_d) - \sin(f_d)\sin(\omega_d)\right] u_{drag} \\ -\frac{2}{v_d}\left[\left(e_d + \cos(f_d)\right)\sin(\omega_d) + \sin(f_d)\cos(\omega_d)\right] u_{drag} \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{2.37}$$

Replacing deputy orbital elements with Eq. (2.20) and Taylor expanding Eq. (2.30) about $\delta\widetilde{\alpha} = 0_{7\times1}$, A_{drag} matrix is obtained:

$$A_{\text{drag}}(\alpha_c) = -\rho_c v_c^2 \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\mu} a_c v_c \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{v_c}\left[\left(e_c + \cos(f_c)\right)\cos(\omega_c) - \sin(f_c)\sin(\omega_c)\right] \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{v_c}\left[\left(e_c + \cos(f_c)\right)\sin(\omega_c) + \sin(f_c)\cos(\omega_c)\right] \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.38}$$

The absence of the B_d term in the resulting matrix, as compared to that presented in [4], is due to the different definition of ΔB, which in that paper is given as $\Delta B = (B_d - B_c)/B_d$. This discrepancy is fully consistent with the adopted normalization. Regarding the remaining differences, namely the signs of the matrix elements and the expression of the $A_{\text{drag}}(1,7)$ entry, they are likely attributable to a typographical error in the referenced paper.

Regarding the applicability range of A_{drag}, the assumption $\rho_c = \rho_d$ requires the distance between the chief and the deputy satellites to be small, so this matrix is valid only for small values of δa, $\delta\lambda$, δe_x, δe_y, δi_x and δi_y. Furthermore, since B_d appears in Eq. (2.37), ΔB must be small, too.

2.3.4 Overall plant matrix

By summing the matrices A_{Kep}, A_{J_2} and A_{drag}, the overall plant matrix of the model is found:

$$A = A_{\text{Kep}} + A_{J_2} + A_{\text{drag}} \tag{2.39}$$

Its range of applicability stems from those of each individual matrix. Therefore this matrix is valid for small values of δa, $\delta\lambda$, δe_x, δe_y, δi_x, δi_y and ΔB.

2.3.5 Control input matrix

To take into account the effects of thrusters on the relative position dynamics, a control input matrix B must be included, with the dynamic system model resulting in:

$$\dot{\delta\widetilde{\alpha}}(t) = A(t)\delta\widetilde{\alpha}(t) + B(t)u(t) \tag{2.40}$$

In [35], the matrix B is computed for the state vector $\delta\alpha$, with control input vector components expressed in the RTN frame. It is reported here with the addition of a row of zeros, to adapt it to the augmented ROE state vector $\delta\widetilde{\alpha}$. Hence the matrix used in this work is:

$$B = \frac{1}{a_c \eta_c} \begin{bmatrix} \frac{2}{\eta_c} e_c \sin(f_c) & \frac{2}{\eta_c}\left[1 + e_c\cos(f_c)\right] & 0 \\ -\frac{2\eta_c^2}{1+e_c\cos(f_c)} & 0 & 0 \\ \eta_c \sin(\theta_c) & \eta_c \frac{\left[2+e_c\cos(f_c)\right]\cos(\theta_c)+e_{x_c}}{1+e_c\cos(f_c)} & \frac{\eta_c e_{y_c}}{\tan(i_c)}\frac{\sin(\theta_c)}{1+e_c\cos(f_c)} \\ -\eta_c\cos(\theta_c) & \eta_c \frac{\left[2+e_c\cos(f_c)\right]\sin(\theta_c)+e_{y_c}}{1+e_c\cos(f_c)} & \frac{-\eta_c e_{x_c}}{\tan(i_c)}\frac{\sin(\theta_c)}{1+e_c\cos(f_c)} \\ 0 & 0 & \eta_c \frac{\cos(\theta_c)}{1+e_c\cos(f_c)} \\ 0 & 0 & \eta_c \frac{\sin(\theta_c)}{1+e_c\cos(f_c)} \\ 0 & 0 & 0 \end{bmatrix} \tag{2.41}$$

Chapter 3

Model Predictive Control

The idea underlying the MPC method, also called Receding Horizon Control (RHC), is the online recursive solution of an OCP over a finite time interval (the prediction window, also known as prediction horizon). The solution of the problem is an optimal sequence of input vectors, of which only the first is actually applied to the dynamic system. The resulting plant state vector obtained by using it (referred to as the successor state) is fed back to the controller as the initial condition for the next MPC iteration. The name 'receding horizon' comes from the fact that the prediction window in which the new problem is solved has the same duration as the previous one, but is shifted by a certain amount of time into the future. The core of the MPC controller is the predictive model. Indeed, it is used to propagate the system dynamics in response to the succession of input vectors to be optimized within the OCP. This model must be simple enough to reduce computational time as much as possible while simultaneously being representative enough of the plant behavior. The use of this method makes it possible to combine optimal control theory with feedback implementation and to handle the constraints arising from the practical context of the implemented controller.
The practical implementation of an MPC controller requires the time discretization of the simplified plant model used for prediction and optimization. As the controller is implemented on digital hardware, it cannot act continuously on the physical system, but must operate at a discrete set of time instants defined by the sampling frequency. At each sampling instant, the controller acquires in feedback the current state of the system, which determines the initial condition for solving the optimal control problem OCP. The internal model used for prediction is discretized consistently with the same time step adopted for control frequency.
The general mathematical formulation of the OCP is now introduced, considering a generic time-invariant nonlinear system which represents the plant model that the controller uses for prediction. It is time-discrete and is described by the following difference equation:

$$x^+ = f(x, u), \quad f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n \tag{3.1}$$

where $x \in \mathbb{R}^n$ is the current state, $u \in \mathbb{R}^m$ is the current control input and x^+ is the successor state. For simplicity, whenever no ambiguity arises and the array dimensions are clear from the context, the symbol 0 alone will be used to denote the null array $0_{q\times s}$,

for arbitrary q and s. The goal of the controller is regulation, namely, to reach the pair $(x, u) = (0, 0)$. In this chapter, the system state x is assumed known at every instant, so there are no uncertainties due to state estimation error.
Before formulating the OCP, it is important to describe the nomenclature that will be used. $a(i)$ denotes the generic quantity a evaluated at time $t = i\Delta t$, where Δt is the sampling step used. The pair (x, i) denotes a generic event, i.e., the fact that at instant i the state of a dynamic system has value x. In the following discussion, i denotes the current time step. The symbol $\mathbb{P}_N(x, i)$ denotes the OCP at event (x, i). N is the prediction horizon, i.e., such that the prediction window has duration $N\Delta t$. In the dissertation the set $\mathbb{I}_{j:k} := \{j, j + 1, ..., k\}$ will often be used, where $j, k \in \mathbb{I}_{\geq 0}$ (with $\mathbb{I}_{\geq 0}$ the set of nonnegative integers) and $k > j$.
Throughout the dissertation, it is useful to have a concise way to describe the evolution of the dynamics governed by $f(\cdot)$ when a specified sequence of inputs is applied to the predictive model, since this notion is employed in the definition of OCP. Let $x(k \mid (x, i))$ and $u(k \mid (x, i))$ denote the predicted system state and the control action at step k, computed at event (x, i). Define

$$\mathbf{u}(x, i) := \Big(u(i \mid (x, i)),\, u(i + 1 \mid (x, i)),\, \ldots,\, u(i + N - 1 \mid (x, i))\Big),$$
$$\mathbf{x}(x, i)) := \Big(x(i \mid (x, i)),\, x(i + 1 \mid (x, i)),\, \ldots,\, x(i + N \mid (x, i))\Big)$$

as the generic sequences of control actions and states predicted for the time steps within prediction window and computed at event (x, i), and

$$\mathbf{u}^0(x, i) := \Big(u^0(i \mid (x, i)),\, u^0(i + 1 \mid (x, i)),\, \ldots,\, u^0(i + N - 1 \mid (x, i))\Big),$$
$$\mathbf{x}^0(x, i) := \Big(x^0(i \mid (x, i)),\, x^0(i + 1 \mid (x, i)),\, \ldots,\, x^0(i + N \mid (x, i))\Big)$$

as the optimal control and state sequences, i.e., those obtained by solving $\mathbb{P}_N(x, i)$. In the dissertation, the superscript 0 indeed denotes optimality.
Since the system is time-invariant, the solution of $\mathbb{P}_N(x, i), \forall i \in \mathbb{I}_{\geq 0}$ coincides with the solution of $\mathbb{P}_N(x,0)$, i.e., $\mathbf{u}^0(x, i) = \mathbf{u}^0(x,0)$ and $\mathbf{x}^0(x, i) = \mathbf{x}^0(x,0)$. Therefore, for simplicity, the problem will be denoted simply by $\mathbb{P}_N(x)$ and the shorthand $x(k \mid x)$, $u(k \mid x)$, $\mathbf{u}(x) = (u(i \mid x), u(i + 1 \mid x), \ldots, u(i + N - 1 \mid x))$ and $\mathbf{x}(x) = (x(i \mid x), x(i + 1 \mid x), \ldots, x(i + N - 1 \mid x))$ will be used. By defining $\mathbf{u}_k(x) := (u(i \mid x), u(i + 1 \mid x), \ldots, u(i + k - 1 \mid x))$ and $\mathbf{x}_k(x) := (x(i \mid x), x(i + 1 \mid x), \ldots, x(i + k - 1 \mid x))$, the solution of (3.1) at time step $k + i$ with initial state x and control sequence $\boldsymbol{u}_k(x)$ is called $\phi(k; x, \boldsymbol{u}_k(x))$. In the MPC control method, only $u^0(i \mid x)$ (also denoted by $\kappa_N(x)$) is applied to the plant. At time step $k + i$, the solution of the closed-loop system composed of plant and MPC controller $x^+ = f(x, \kappa_N(x))$, propagated starting from (x, i), is denoted by $\xi(k; x)$.
The generic problem $\mathbb{P}_N(x)$ consists of two fundamental elements: the cost function and the constraints. A description of these components is provided below, followed by the formal definition of $\mathbb{P}_N(x)$.
The OCP is formulated to minimize the cost function $V_N : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$, which gathers

a performance metric over the prediction window and is generally defined as:

$$V_N(\boldsymbol{x}, \boldsymbol{u}) := \sum_{k=0}^{N-1} \ell(x(k \mid x), u(k \mid x)) + V_f(x(N \mid x)) \tag{3.2}$$

where the functions $\ell : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ and $V_f : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ are called stage cost and terminal cost, respectively. $V_N(\cdot)$ is to be minimized with respect to the decision variables $x(0 \mid x), x(1 \mid x), ..., x(N \mid x)$ and $u(0 \mid x), u(1 \mid x), ..., u(N-1 \mid x)$. In fact, it quantifies a penalty associated with the deviation of the system from a desired behavior. In particular, $\ell(x(k \mid x), u(k \mid x))$ represents the penalty assigned to the event $(x(k \mid x), u(k \mid x))$ based on the deviation from $(0, 0)$. $V_f(x(N \mid x))$ plays an identical role on the state $x(N \mid x)$. The presence of the terminal cost in the cost function is fundamental because it underlies the theorems commonly used in the literature to guarantee the property of asymptotic stability of the origin for the closed-loop system $x^+ = f(x, \kappa_N(x))$. For a deeper understanding of this property and how it can be guaranteed, see [33].
The constraints represent limitations on the infinite possible trajectories of the predictive model within the prediction window. They are classified into equality constraints and inequality constraints.
The equality constraints could be used to impose the initial condition of the prediction, usually $x(0 \mid x) = x(i)$, and the model dynamics, such that $x(k \mid x) = \phi(k; x(0 \mid x), \boldsymbol{u}_k(x)), \forall k \in \mathbb{I}_{1:N}$.
The inequality constraints instead represent physical limits of the system under consideration (for example, the existence of a maximum deliverable input) or desired conditions for the system states in the particular scenario in which the controller must operate. Given the closed sets $\mathbb{X}$ and $\mathbb{U}$, the inequality constraints for states and inputs are defined such that $x(k \mid x) \in \mathbb{X}$ and $u(k \mid x) \in \mathbb{U}$ $\forall k \in \mathbb{I}_{0:N}$. In addition, these constraints can take the form of mixed constraints, i.e., $(x(k \mid x), u(k \mid x)) \in \mathbb{Z}$ $\forall k \in \mathbb{I}_{0:N}$, where $\mathbb{Z} \subseteq \mathbb{X} \times \mathbb{U}$ is usually defined as $\mathbb{Z} = \{(x, u) \mid Fx + Eu \leq e\}$ with F and E generic matrices and e a generic vector, that is, a polyhedral set. Hence $u(k \mid x) \in \mathbb{U}(x) := \{ u \in \mathbb{U} \mid (x, u) \in \mathbb{Z} \}$ and $x(k \mid x) \in \{ x \in \mathbb{X} \mid \mathbb{U}(x) \neq \varnothing \}$ $\forall k \in \mathbb{I}_{0:N}$. If no mixed constraints are present, $\mathbb{Z} = \mathbb{X} \times \mathbb{U}$.
Finally, a terminal constraint set $\mathbb{X}_f$ is very often present. The constraint $x(N \mid x) \in \mathbb{X}_f \subseteq \mathbb{X}$ is imposed, with $\mathbb{X}_f$ a singleton (thus an equality constraint) or polyhedral or ellipsoidal (hence an inequality constraint). The presence of $\mathbb{X}_f$ has so far been shown in the literature to be necessary for guaranteeing recursive feasibility, defined as the property whereby if $\mathbb{P}_N(x_1)$ has a solution, then, $\forall k \in \mathbb{I}_{1:N}$, $\mathbb{P}_N(x_2)$ is also solvable, with $x_2 = \xi(k; x_1)$. The set $\mathbb{X}_f$, together with $V_f(\cdot)$, is also commonly used in results guaranteeing asymptotic stability of the origin for $x^+ = f(x, \kappa_N(x))$. However, it is not always strictly necessary [33].
The constraints presented here are defined as hard constraints, since the existence of a solution for $\mathbb{P}_N(x)$ cannot disregard their satisfaction, i.e., the OCP is defined feasible if and only if there exists at least one state-control trajectory $(x(k \mid x), u(k \mid x))_{k \in \mathbb{I}_{0:N}}$ that satisfies all the hard constraints of the problem.
Considering the current plant state $x(i)$, the OCP $\mathbb{P}_N(x(i))$ is therefore formally defined

as:

$$\begin{aligned}\mathbb{P}_N(x(i)): \qquad & V_N^0(x(i)) := \min_{\mathbf{x}(x),\mathbf{u}(x)} V_N(\mathbf{x}(x),\mathbf{u}(x)) \\ & \text{subject to} \\ & x(0 \mid x) = x(i), \\ & x(k \mid x) = \phi(k; x(0 \mid x), \boldsymbol{u}_k(x)),\ \forall k \in \mathbb{I}_{1:N}, \\ & u(k \mid x) \in \mathbb{U}(x(k \mid x)),\ \forall k \in \mathbb{I}_{0:N-1}, \\ & x(k \mid x) \in \{x \in \mathbb{X} \mid \mathbb{U}(x) \neq \varnothing\},\ \forall k \in \mathbb{I}_{1:N-1}, \\ & x(N \mid x) \in \mathbb{X}_f.\end{aligned} \tag{3.3}$$

By defining $\mathbb{Z}_N := \{(x,\mathbf{u}) \mid (\phi(k; x(0 \mid x), \mathbf{u}_k(x)), u(k \mid x)) \in \mathbb{Z},\ \forall k \in \mathbb{I}_{0:N-1},\ \phi(N; x(0 \mid x), \mathbf{u}) \in \mathbb{X}_f\} \subset \mathbb{X} \times \mathbb{U}^N$ and $\mathcal{U}_N(x) := \{\mathbf{u} \mid (x,\mathbf{u}) \in \mathbb{Z}_N\}$, a formulation of $\mathbb{P}_N(x(i))$ equivalent to (3.13) is obtained:

$$\mathbb{P}_N(x(i)): \qquad V_N^0(x(i)) := \min_{\mathbf{x}(x),\mathbf{u}(x)} \{V_N(\mathbf{x}(x),\mathbf{u}(x)) \mid \mathbf{u} \in \mathcal{U}_N(x(i))\}. \tag{3.4}$$

The problem $\mathbb{P}_N(x(i))$ is formally defined feasible if and only if $\mathcal{U}_N(x(i)) \neq \varnothing$. Otherwise, it is defined as infeasible.

It is useful to make an important clarification about the constraints of an OCP. The constraints defined so far are called hard constraints. The reason is that they are rigid: if no state-control trajectory exists that satisfies the inequalities representing them, the problem is infeasible. They are therefore used to represent physical limits of the system or constraints whose violation is unacceptable. In contrast to these, there exist the soft constraints. The latter are not rigidly binding, meaning that infeasibility cannot occur because of them, precisely due to the way they are defined. They are thus suitable for representing desirables, that is, conditions whose violation is tolerated, even though it is not desirable, and help avoid infeasibility.

For simplicity, consider the following hard constraints: $x(k \mid x) \in \{x \in \mathbb{R}^n \mid Gx \leq g\}$ and $u(k \mid x) \in \{u \in \mathbb{R}^m \mid Hu \leq h\},\ \forall k \in \mathbb{I}_{\geq 0}$, with $G \in \mathbb{R}^{q \times n}$, $H \in \mathbb{R}^{s \times m}$, $g \in \mathbb{R}^q$ and $h \in \mathbb{R}^s$ arbitrary. To transform the state constraints into soft constraints slack variables are introduced, $\varepsilon(k) \geq 0$, which are included among the decision variables of the OCP, together with $\mathbf{x}$ and $\mathbf{u}$. The soft state constraints can then be expressed as $Gx(k \mid x) \leq g + \varepsilon(k),\ k \in \mathbb{I}_{\geq 0}$. The quantity $\varepsilon(k)$ represents the violation of the hard state constraint at step k. With this reformulation, infeasibility due to the state constraint cannot occur: the constraint ceases to be hard. To include the slack variables among the decision variables, the augmented input $\tilde{u}(k \mid x) := [u(k \mid x);\ \varepsilon(k)]$ is defined. It is therefore possible to write

$$\tilde{G}x(k \mid x) + \tilde{H}\tilde{u}(k \mid x) \leq \tilde{h}, \quad k \geq 0$$

with

$$\tilde{G} = \begin{bmatrix} 0_{s\times n} \\ 0_{q\times n} \\ G \end{bmatrix}, \qquad \tilde{H} = \begin{bmatrix} H & 0_{s\times q} \\ 0_{q\times m} & -I_{q\times q} \\ 0_{q\times m} & -I_{q\times q} \end{bmatrix}, \qquad \tilde{u}(k \mid x) = \begin{bmatrix} u(k \mid x) \\ \varepsilon(k) \end{bmatrix}, \qquad \tilde{h} = \begin{bmatrix} h \\ 0_{q\times 1} \\ g \end{bmatrix}$$

where $I_{a\times a}$ denotes the identity matrix of dimension a. The slack variables are to be minimized; therefore, a term penalizing $\varepsilon(k)$ is included within the stage cost $\ell(\cdot)$.
After providing an overview of how the MPC method works, it is appropriate to highlight how the choice of the parameters Δt and N (the prediction horizon) plays a crucial role, since both require a careful trade-off. In particular, as Δt decreases, the controller can act more frequently on the plant and thus influence the faster components of the dynamics, though computational effort increases. Concerning N, as it decreases, the controller becomes more aggressive, since it can only make short-term predictions. However, the larger the N, the less accurate the prediction, since the employed model is always an approximation of reality.
Moreover, in the formulation, an additional parameter N_c may appear, called the control horizon and chosen arbitrarily, such that $N_c \in \mathbb{I}_{1:N}$. The introduction of this parameter is related to the intention to define, within the prediction window, a given control policy for $u(N_c \mid x), u(N_c+1 \mid x), ..., u(N-1 \mid x)$ (for example $u(N_c+k \mid x) = 0\ \forall k \in \mathbb{I}_{0:N-N_c-1}$, or $u(N_c+k \mid x) = Kx(N_c+k \mid x)\ \forall k \in \mathbb{I}_{0:N-N_c-1}$) and to treat as decision variables only the control actions $u(0 \mid x), u(1 \mid x), ..., u(N_c-1 \mid x)$. Increasing N_c provides a larger optimization margin, which theoretically improves performance, albeit at the expense of a higher computational burden due to increased problem complexity. Small N_c tends to promote, though not guarantee, internal stability.

3.1 MPC formulation

Within this thesis, the model used by the MPC controller derives from the continuous-time dynamic system model resulting from dynamics linearization, presented in the previous chapter. Let $x(i) := \delta\widetilde{\alpha}(\alpha_c(t_i))$ denote the current system state, computed at time $t_i := i\Delta t$, which marks the beginning of the prediction window of length $N\Delta t$. A dynamic model consisting of the following continuous-time linear time-invariant system is considered:

$$\dot{x}(t \mid i) = A(t_0)x(t \mid i) + B(t_0)u(t \mid i) \tag{3.5}$$

where $x(t \mid i) \in \mathbb{R}^7$ and $u(t \mid i) \in \mathbb{R}^3$ represent the predicted system state $\delta\widetilde{\alpha}$ and the control action at time t, computed at time t_i. Hence $x(t_i \mid i) = x(i)$. The use of t in $x(t \mid i)$ and $u(t \mid i)$, instead of the index k employed so far, reflects that this system is continuous in the time domain. Moreover, the use of i instead of x, as adopted previously, highlights that the system in Eq. (3.5), although time-invariant, depends on the plant and control input matrices evaluated at time t_i. Therefore, its description requires the initial event (x, i) rather than x alone.
A time discretization is then performed. The assumption made here is that the components of the input vector are constant during each sampling interval ("staircase inputs"); therefore, a zero-order hold method is used:

$$\underline{A} := e^{A(t_i)\Delta t}, \quad \underline{B} := \int_0^{\Delta t} e^{A(t_i)(\Delta t-\tau)} d\tau B(t_i) \tag{3.6}$$

The following equation describes the resulting time-discrete model:

$$x(k+1 \mid i) = \underline{A}x(k \mid i) + \underline{B}u(k \mid i), \ \forall k \in \mathbb{I}_{0:N-1} \tag{3.7}$$

where $x(k \mid i)$ and $u(k \mid i)$ represent the predicted system state and the control action at step k, computed at time t_i. Hence $x(0 \mid i) = x(i)$. The considered dynamic model is not time-invariant. With the same current state $x(i)$ and the same input sequence applied to the system within the prediction window, the predicted system state at step k depends on t_i. The resulting dynamic model is Linear Parameter-Varying (LPV). Within the prediction window, the matrices $\underline{A}$ and $\underline{B}$ are held constant at the values computed at t_i.
To account for limits in engine dynamics, the problem is augmented using the input increments $\Delta u(k \mid i) := u(k \mid i) - u(k-1 \mid i)$, extending the algebraic system with the state vector $x_u(k \mid i) := u(k-1 \mid i)$:

$$\begin{cases} x(k+1 \mid i) = \underline{A}x(k \mid i) + \underline{B}u(k-1 \mid i) + \underline{B}\Delta u(k \mid i), \\ x_u(k+1 \mid i) = x_u(k \mid i) + \Delta u(k \mid i). \end{cases}$$

Therefore, the following matrix equations hold:

$$\begin{cases} \begin{bmatrix} x(k+1 \mid i) \\ x_u(k+1 \mid i) \end{bmatrix} = \begin{bmatrix} \underline{A} & \underline{B} \\ 0_{3\times 7} & I_{3\times 3} \end{bmatrix} \begin{bmatrix} x(k \mid i) \\ x_u(k \mid i) \end{bmatrix} + \begin{bmatrix} \underline{B} \\ I_{3\times 3} \end{bmatrix} \Delta u(k \mid i) \\ y(k \mid i) = \begin{bmatrix} C & 0_{7\times 3} \end{bmatrix} \begin{bmatrix} x(k \mid i) \\ x_u(k \mid i) \end{bmatrix} \end{cases} \tag{3.8}$$

where $C := I_{7\times 7}$. For notational convenience, these equations can be equivalently rewritten in compact form as

$$\tilde{x}(k+1 \mid i) = \tilde{A}\tilde{x}(k \mid i) + \tilde{B}\Delta u(k \mid i), \tag{3.9}$$

where

$$\tilde{A} := \begin{bmatrix} \underline{A} & \underline{B} \\ 0_{3\times 7} & I_{3\times 3} \end{bmatrix}, \qquad \tilde{B} := \begin{bmatrix} \underline{B} \\ I_{3\times 3} \end{bmatrix}, \qquad \tilde{x}(k \mid i) := \begin{bmatrix} x(k \mid i) \\ x_u(k \mid i) \end{bmatrix}.$$

In this way, the input increments are included among the decision variables of the OCP; the cost function depends on $\tilde{\mathbf{x}}(i)$ and $\mathbf{\Delta u}(i)$, where $\tilde{\mathbf{x}}(i) := (\tilde{x}(0 \mid i), \tilde{x}(1 \mid i), ..., \tilde{x}(N \mid i))$ and $\mathbf{\Delta u}(i) := (\Delta u(0 \mid i), \Delta u(1 \mid i), ..., \Delta u(N-1 \mid i))$ are sequences of vectors. The solution of $\tilde{x}(k+1 \mid i) = \tilde{A}\tilde{x}(k \mid i) + \tilde{B}\Delta u(k \mid i)$ at step k with initial condition $\tilde{x}$ is denoted by $\beta(k; \tilde{x}, \mathbf{\Delta u}_k(i), t_i)$, where $\mathbf{\Delta u}_k(i) := (\Delta u(0 \mid i), \Delta u(1 \mid i), ..., \Delta u(k-1 \mid i))$. Thus $\tilde{x}(k \mid i) = \beta(k; \tilde{x}, \mathbf{\Delta u}_k(i), t_i) = \tilde{A}\tilde{x}(k-1 \mid i) + \tilde{B}\Delta u(k-1 \mid i)$.
The objective of the OCP to be formulated at time t_i is setpoint tracking, namely, to drive $\tilde{\mathbf{x}}$ toward a setpoint augmented state $\tilde{x}_{sp}(i)$, fixed within the prediction window, such that:

$$\begin{cases} \tilde{x}_{sp}(i) = \tilde{A}\tilde{x}_{sp}(i) + \tilde{B}\Delta u_{sp}, \\ \Delta u_{sp} = 0, \\ r = \tilde{C}\tilde{x}_{sp}(i). \end{cases} \tag{3.10}$$

where $\tilde{C} := \begin{bmatrix} C & 0_{7\times 3} \end{bmatrix}$ and r is the reference output. The index i in $x_{sp}(i)$ denotes that, while r remains constant throughout the entire simulation time T_{sim}, this vector is constant only within the prediction window and is computed at current event $(x(i), i)$. $\tilde{x}_{sp}(i)$ is therefore the solution of the following linear system:

$$A_{sp}\tilde{x}_{sp}(i) = b_{sp} \tag{3.11}$$

where

$$A_{sp} := \begin{bmatrix} \tilde{A} - I_{10\times 10} \\ \tilde{C} \end{bmatrix}, \qquad b_{sp} := \begin{bmatrix} 0_{10\times 1} \\ r \end{bmatrix}.$$

Define the weighted norm $\|d\|_D := \sqrt{d^\top D d}$, with $d \in \mathbb{R}^q$ and $D \in \mathbb{R}^{q\times q}$ symmetric positive definite, where q is arbitrary. Then the cost function $V_N(\cdot)$ used in the dissertation is:

$$V_N(\tilde{\mathbf{x}}(i), \boldsymbol{\Delta}\mathbf{u}(i), t_i) = \sum_{k=0}^{N-1} \left(\|\tilde{x}_{sp}(i) - \tilde{x}(k \mid i)\|_Q^2 + \|\Delta u(k \mid i)\|_R^2 \right) + \|\tilde{x}_{sp}(i) - \tilde{x}(N \mid i)\|_Q^2 \tag{3.12}$$

Introducing t_i as an argument of $V_N(\cdot)$ reflects that the terms in its expression depend on the event (x, i), not solely on the initial state x. In Eq. (3.12), the matrices Q and R are symmetric positive definite and constitute the weights, i.e., the controller tuning parameters. The ratio between the orders of magnitude of the components of Q and those of R reflects the design choice of balancing the speed with which to reduce the deviation $\tilde{x}_{sp}(i) - \tilde{x}(k \mid i)$ against the limitation of the control action variation $\Delta u(k \mid i)$. A high ratio favors faster convergence toward the setpoint trajectory, at the expense of more pronounced variations in control actions, and vice versa.
Define $\mathbb{X}_u := \{u \in \mathbb{R}^3 \mid -u_{max} \leq u_i \leq u_{max},\ \forall i \in \mathbb{I}_{1:3}\}$, $\Delta\mathbb{U} := \{\Delta u \in \mathbb{R}^3 \mid \Delta u_{min} \leq \Delta u_i \leq \Delta u_{max},\ \forall i \in \mathbb{I}_{1:3}\}$, where u_i and Δu_i denote the i-th component of u and Δu, respectively. Here $u_{max} = T_{max}/m_d$ is the maximum acceleration that each engine can produce on the deputy satellite, where T_{max} is the maximum thrust of the engine and m_d is the mass of the deputy satellite. It is important to emphasize that in this formulation there are no true slew-rate constraints, since each of the six thrusters is aligned along the same direction in the RTN frame; therefore, the components of the overall thrust vector are not due to thruster steering, but to the vector composition of the individual thrust vectors.
The initial condition of the OCP is $\tilde{x}(i) := [x(i); u(i-1)]$, where $u(i-1)$ represents the control action applied to the plant in the interval $[t_i - \Delta t, t_i)$. The OCP $\mathbb{P}_N(\tilde{x}(i), t_i)$ is thus defined as:

$$\begin{aligned} \mathbb{P}_N(\tilde{x}(i), t_i): \qquad & V_N^0(\tilde{x}(i), t_i) := \min_{\tilde{\mathbf{x}}(i), \boldsymbol{\Delta}\mathbf{u}(i)} V_N(\tilde{\mathbf{x}}(i), \boldsymbol{\Delta}\mathbf{u}(i), t_i) \\ & \text{subject to} \\ & \quad \tilde{x}(0 \mid i) = \tilde{x}(i), \\ & \quad \tilde{x}(k \mid i) = \beta(k; \tilde{x}, \boldsymbol{\Delta}\mathbf{u}_k(i), t_i),\ \forall k \in \mathbb{I}_{1:N}, \\ & \quad x_u(k \mid i) \in \mathbb{X}_u,\ \forall k \in \mathbb{I}_{1:N}, \\ & \quad \Delta u(k \mid i) \in \Delta\mathbb{U},\ \forall k \in \mathbb{I}_{0:N-1}. \end{aligned} \tag{3.13}$$

The resulting problem is a quadratic program, since the cost function is quadratic and the constraint sets are polyhedral. Algorithm 1 summarizes the main steps of the implemented MPC control scheme. At each iteration, the controller measures the system state and model parameters, updates the prediction model and the initial conditions of the optimization problem, solves the corresponding finite-horizon OCP, and applies the first control input of the optimal sequence to the plant.

Algorithm 1 MPC control scheme

Require: u_{max}, Δu_{min}, Δu_{max}, $x(0)$ and r

Set $u(-1) = 0$
Compute b_{sp}
for $i = 0 : \lfloor T_{\text{sim}}/\Delta t \rfloor$ **do**
    Evaluate $A(\alpha_c(t_i))$ and $B(\alpha_c(t_i))$
    if $i \neq 0$ **then**
        Measure $x(i)$
    end if
    Update $\tilde{x}(i) = [x(i); u(i-1)]$
    Compute $\underline{A}$, $\underline{B}$, $\tilde{A}$, $\tilde{B}$ and A_{sp}
    Solve $A_{sp}\tilde{x}_{sp}(i) = b_{sp} \Rightarrow$ obtain $\tilde{x}_{sp}(i)$
    Solve $\mathbb{P}_N(\tilde{x}(i), t_i) \Rightarrow$ obtain $\tilde{\mathbf{x}}^0(i)$ and $\mathbf{\Delta u}^0(i)$
    Compute $u(i) = u(i-1) + \Delta u^0(0 \mid i)$
    Apply $u(i)$ to the plant
end for

Chapter 4

Simulations

This chapter describes the simulations carried out in this master's thesis using MATLAB/Simulink. In particular, the chapter is structured as follows.
In Section 4.1, the strategy introduced in Section 2.1 is employed to validate the Simulink scheme of the plant, namely the model used to represent the real orbital dynamics with respect to the aspects of interest for this thesis. In Section 4.2, the simulated maneuver scenario is presented, together with the Simulink model architecture describing the interaction between the MPC algorithm and the plant, the adopted settings, and the obtained results.
The simulations were executed on an HP Pavilion 15 computer, equipped with an AMD Ryzen 7 5700U processor (8 cores, base frequency 1.80 GHz), 16 GB of RAM, and Windows 11 operating system, using MATLAB/Simulink R2024b.

4.1 Exact orbital propagator scheme validation

The model that constitutes the plant in this thesis has been developed using Simulink and makes use, among other features, of predefined Simulink blocks. It is described in detail in Subsection 4.1.1. The adopted scheme requires validation for two main reasons. First, it includes user-developed blocks that implement transformations between satellite positions and velocities expressed in the ECI reference frame (i.e., r_{ECI} and v_{ECI}), classical orbital elements, and position and velocity coordinates expressed in the RTN frame. Second, the scheme relies on the assumption, already introduced in Subsection 2.3.2, that mean orbital elements can be identified with their osculating counterparts. This same assumption is also adopted throughout the validation presented in this section.

As previously mentioned in Section 2.1, the Simulink scheme is validated by comparing the trajectory obtained by propagating the prescribed initial conditions using the scheme illustrated in Subsection 4.1.1 with the trajectory obtained by propagating the same initial conditions using the propagator presented in [40], representing both trajectories in the RTN reference frame. The satellite relative motion propagator adopted as reference in [40] accounts for eccentricity, nonlinearities, and the J_2 perturbation.

The assigned initial conditions for the simulation discussed in Section 2.1 are

$$\begin{aligned} &a_c(0) = 8000 \text{ km}, \quad e_c(0) = 0.1, \quad i_c(0) = 63.4349^\circ, \\ &\omega_c(0) = 45^\circ, \quad \Omega_c(0) = 0, \quad f_c(0) = 0. \end{aligned} \tag{4.1}$$

and

$$\underline{x}(0) = 0.5 \text{ km}, \quad \underline{y}(0) = 0, \quad \underline{z}(0) = 1 \text{ km}, \quad \dot{\underline{x}}(0) = 0. \tag{4.2}$$

The approximation mentioned above, namely about mean orbital elements and their osculating counterparts, enters the formulation through the initial conditions specified in Eqs. (4.1). The remaining components of the initial relative velocity of the deputy satellite (namely $\dot{\underline{y}}(0)$ and $\dot{\underline{z}}(0)$) are determined by imposing conditions in Eqs. (2.15) and (2.16) on the initial quantities.
In order to gain additional confidence, a comparison is also performed using alternative equations to determine the remaining components of the initial relative velocity of the deputy satellite. These equations were previously proposed in the literature (and are illustrated in [40, Subsecs. 3.3.3 and 3.3.4], together with Eqs. (2.15) and (2.16)) to obtain periodic relative orbits when nonlinearities and/or the J_2 perturbation are neglected. Specifically, Inalhan et al. [17] proposed to adopt the conditions

$$\dot{\underline{z}}(0) = 0 \tag{4.3}$$

and

$$\dot{\underline{y}}(0) = -\frac{n_c(2+e_c)\underline{x}(0)}{(1+e_c)^{1/2}(1-e_c)^{3/2}}\,, \tag{4.4}$$

which yield a periodic relative orbit when the J_2 perturbation and nonlinear effects are neglected.
In a different approach, Gurfil [15] and Xing et al. [42] adopted the condition given by Eq. (4.3), together with the energy-matching condition for classical elliptic Keplerian motion

$$(\dot{\underline{x}} + \dot{\underline{r}}_c - \underline{y}\omega_z)^2 + (\dot{\underline{y}} + \underline{r}_c\omega_z + \underline{x}\omega_z)^2 + \dot{\underline{z}}^2 = \dot{\underline{r}}_c^2 + \underline{r}^2\omega_z^2 + 2\mu\left(\frac{1}{\underline{r}_d} - \frac{1}{\underline{r}_c}\right). \tag{4.5}$$

These two initial relative velocity conditions allow the generation of a periodic relative orbit when nonlinearities are considered and the J_2 perturbation is neglected.
It is important to emphasize that Eqs. (2.15), (2.16), (4.4), and (4.5) are not conceptually formulated with reference to a specific time instant. Indeed, they derive from theoretical assumptions and simplifications intended to describe general properties of the relative dynamics, rather than instantaneous constraints valid only at a specific time. In the present work, however, these equations are operationally employed as auxiliary conditions to determine the remaining components of the initial relative velocity of the deputy satellite. Consequently, all quantities appearing in these equations must be interpreted as initial values of the simulation.

4.1.1 Scheme structure

The adopted scheme is illustrated in Fig. 4.1. The orange blocks represent the plant itself, namely the model describing the time evolution of the state vector x. The remaining block allows visualization of the results in terms of coordinates expressed in the RTN frame. The plant is based on two orbital propagators assumed to be exact, one for the chief satellite and one for the deputy satellite. For each of them, a predefined Simulink block named Orbit Propagator is used in the Numerical (high precision) propagation configuration [28]. Both the chief and deputy propagators use the oblate ellipsoid (J_2) gravitational potential model for the Earth. This model fully accounts for the effects induced by the J_2 perturbation on the orbital dynamics (see [29] for a detailed explanation). In addition, the deputy propagator includes the contribution of aerodynamic drag, assuming a constant ballistic coefficient for the deputy satellite (i.e., constant mass, drag coefficient, and drag area). The atmospheric density required to compute this contribution is obtained using the Jacchia–Bowman 2008 model [5], implemented using MATLAB functions developed by Mahooti [27], version 2.2.0. The aerodynamic drag contribution is not included in the chief propagator, since the ballistic coefficient of the chief satellite is assumed to be zero ($B_c = 0$). The chief therefore follows a non-decaying trajectory.
Within the context of the present scheme validation, the aerodynamic drag contribution in the deputy orbit propagator has been disabled. This choice is motivated by the fact that the Jacchia–Bowman 2008 atmospheric model is well established in the literature and therefore does not require validation. Moreover, in the simulations presented in [40], which are used here for comparison, aerodynamic drag is not included in the relative motion propagator adopted as reference. In the simulation discussed in Section 4.2, this contribution will instead be included.
The simulations were performed using the variable-step solver `ode45`, provided by MATLAB [38]. Both the time step and its maximum size were automatically determined by the solver, without imposing manual constraints. The simulated time interval has a duration of 96 hours.

4.1.2 Results

The aim of the validation is to test the Simulink model presented in Fig. 4.1, by verifying that it reproduces the results reported in [40, Subsec. 3.3.4].
Figs. 4.2, 4.3, and 4.4 show the results of the present validation in terms of the deputy relative trajectory represented in the RTN frame. Each figure represents the trajectory obtained by applying a specific subset of the aforementioned conditions. In detail, the trajectories shown in Figs. 4.2, 4.3, and 4.4 correspond to cases (i), (ii), and (iii), respectively, and are obtained by imposing the conditions in Eqs. (4.3) and (4.4), Eqs. (4.3) and (4.5), and Eqs. (2.15) and (2.16). In each case, the cited equations define the case-dependent portion of the initial conditions, while the remaining initial conditions are common to all simulations and are given in Eqs. (4.1) and (4.2).
In all three cases, an excellent agreement is observed between the results, both in terms of trajectory shape and oscillation amplitude. The validation procedure can therefore be considered successfully completed, confirming the correct implementation of the dynamic

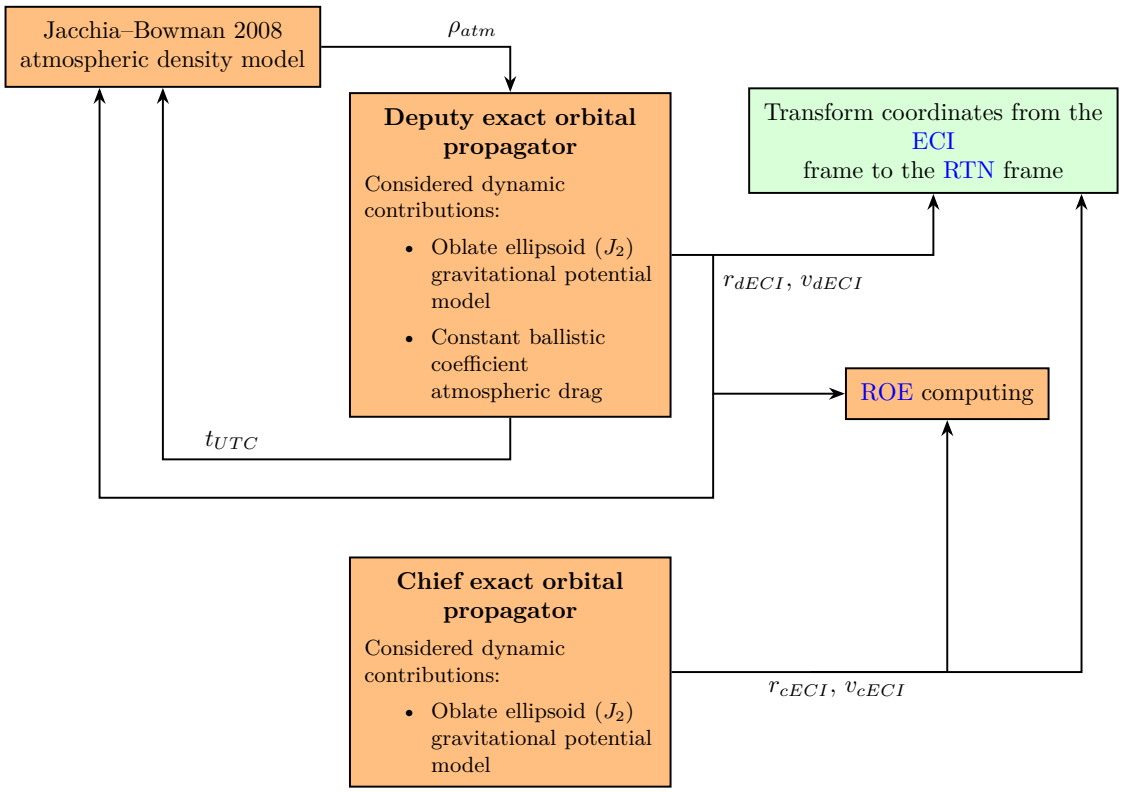
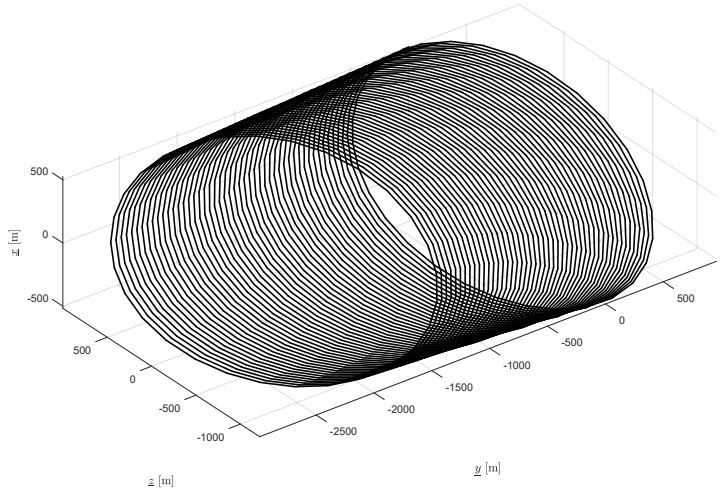
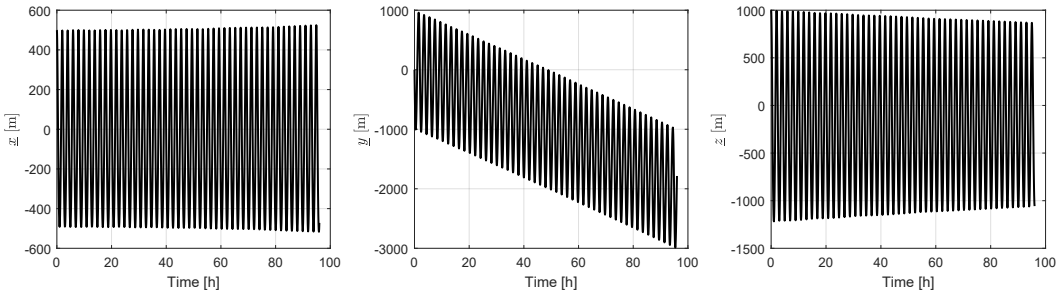


Figure 4.1: Exact orbital propagator scheme

model adopted as the plant in this work.

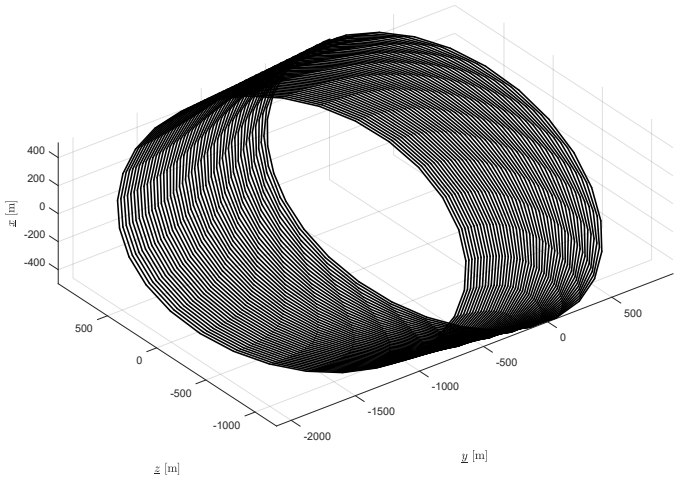


(a)

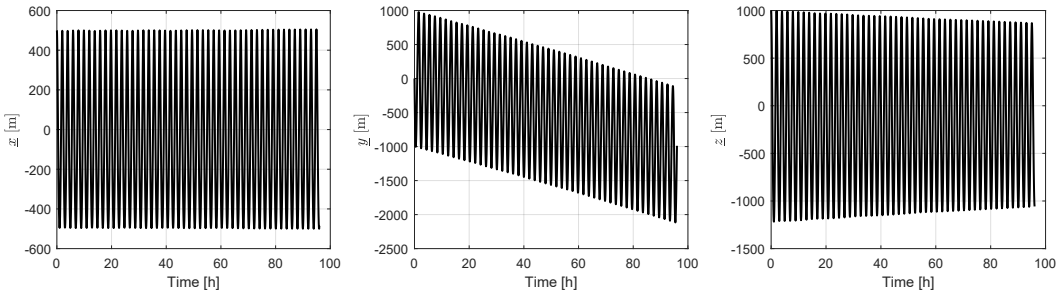


(b)

Figure 4.2: Numerical results obtained with the proposed Simulink model for case (i), corresponding to conditions given by Eqs. (4.3) and (4.4). (a) Three-dimensional relative trajectory in the RTN frame. (b) Relative motion projected onto the RTN frame axes.

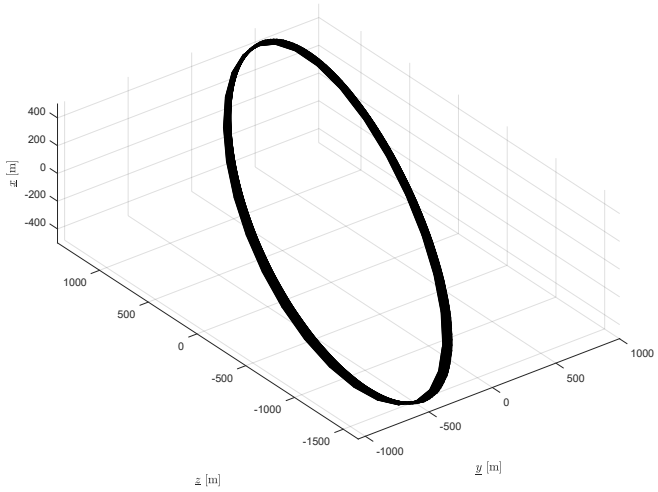


(a)

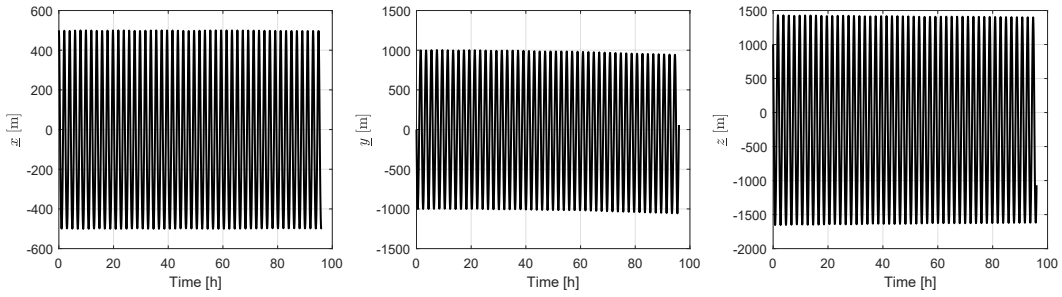


(b)

Figure 4.3: Numerical results obtained with the proposed Simulink model for case (ii), corresponding to conditions given by Eqs. (4.3) and (4.5). (a) Three-dimensional relative trajectory in the RTN frame. (b) Relative motion projected onto the RTN frame axes.



(a)



(b)

Figure 4.4: Numerical results obtained with the proposed Simulink model for case (iii), corresponding to conditions given by Eqs. (2.15) and (2.16). (a) Three-dimensional relative trajectory in the RTN frame. (b) Relative motion projected onto the RTN frame axes.

4.2 Simulated scenario

The scenario in which the implemented MPC algorithm is tested consists of a formation reconfiguration maneuver, in which the relative orbit of the deputy satellite is modified. In particular, the maneuver considered here is an out-of-plane transfer, so called because the projection of the target relative orbit onto the Radial–Transverse plane of the RTN frame is identical to that of the initial relative orbit. The starting classical orbital elements of the chief orbit are reported in Table 4.3. The starting and target ROE are reported in Table 4.1. Table 4.2 contains the relevant physical properties of the two satellites.

Table 4.1: Initial and target ROE for the maneuver

	$a_{c0}\delta a$ [m]	$a_{c0}\delta\lambda$ [m]	$a_{c0}\delta e_x$ [m]	$a_{c0}\delta e_y$ [m]	$a_{c0}\delta i_x$ [m]	$a_{c0}\delta i_y$ [m]
starting	0	0	0	200	0	180
target	0	0	0	200	0	420

Table 4.2: Satellites properties

Parameter	**Value**
m_c	20 kg
B_c	0
m_d	20 kg
S_d	0.1 m^2
C_{Dd}	2.1
T_{max}	0.65 mN

Table 4.3: Initial classical orbital elements of the chief satellite

a_{c0} [km]	e_{c0} [-]	i_{c0} [°]	Ω_{c0} [°]	ω_{c0} [°]	M_{c0} [°]
6771	0.001	97.004	30	90	0

The simulated maneuver is identical to one of those presented in [3] and is analogous to the one described in [4]. A particularly interesting aspect is that references already exist in the literature in which the same plant model is used both for prediction and optimization (also referred to as the internal plant model). The reason for choosing this specific example emerges from the following considerations.
Fig. 4.5 illustrates the geometric relationship between the osculating ROE and the osculating relative orbit represented in the RTN frame (see [37] for further details). From this representation, it follows that δa, $\delta\lambda$, δe_x, and δe_y are referred to as in-plane ROE, whereas δi_x and δi_y are referred to as out-of-plane ROE. The reader should note that the relative orbit depicted in Fig. 4.5 is osculating. It would represent the actual relative trajectory followed by the deputy satellite only if perturbations due to the J_2 term and atmospheric drag were absent, i.e., in the case of purely Keplerian dynamics.

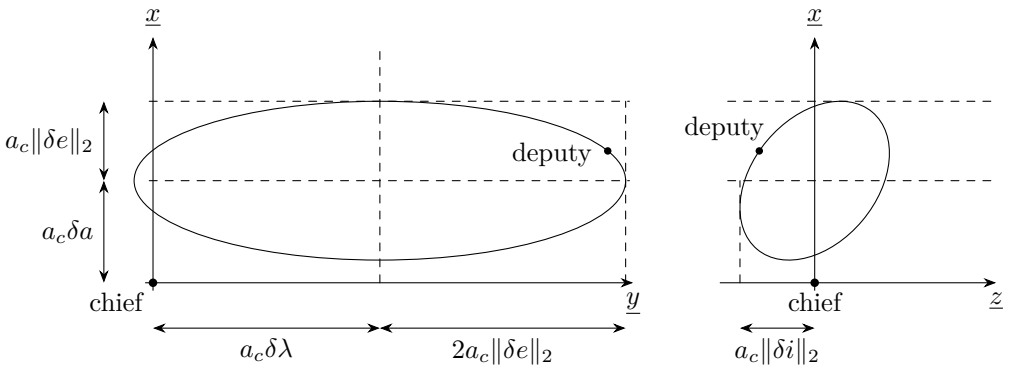


Figure 4.5: Unperturbed deputy relative motion in the RTN frame: in-plane motion on the left and out-of-plane motion on the right [37].

Steindorf et al. [37] further highlight how the structure of matrix B, shown in Eq. (2.41), implies that the most efficient way to modify δa and the vector δe consists in applying a purely transverse thrust. The relative mean longitude $\delta\lambda$ is instead directly controllable through a purely radial thrust; however, it can also be controlled indirectly by means of a transverse thrust that produces a suitable variation in δa, exploiting the effect of Keplerian dynamics (see Eq. (2.31)). It follows that all the in-plane ROE can be modified through a purely transverse thrust.
An analysis of the same matrix also shows that the out-of-plane ROE can be modified exclusively by means of a thrust applied in the normal direction. These considerations motivate the choice, proposed by the authors, of not applying thrusts in the radial direction in order to minimize propellant consumption. This choice is also adopted in [3] [4] and in the present thesis. In the latter case, in particular, considering the presence of three pairs of thrusters permanently aligned with the three axes of the RTN frame, this choice allows the elimination of the pair of thrusters aligned with the radial direction, with a consequent reduction in the mass of the deputy satellite. Such thrusters would in fact not be usable for relative attitude control due to their geometric arrangement on board the satellite (described in more detail in Chapter 2).

In the considered example, in the absence of perturbations due to the J_2 term and aerodynamic drag, the formation reconfiguration would be achievable through a purely normal thrust. However, Eq. (2.36) shows that aerodynamic drag acts on the in-plane ROE. Consequently, control of the relative motion also requires the use of a transverse thrust component. Moreover, Eq. (2.32) shows that the perturbative effects associated with the J_2 term require control actions with thrust components in both considered directions.
The choice of the considered formation reconfiguration example allows a clear interpretation of the role of the control components. In particular, the thrust in the normal direction is mainly employed to vary the component δi_y, whereas the thrust in the transverse direction is primarily devoted to compensating for the undesired variations of the in-plane ROE induced by aerodynamic drag. Both thrust components also contribute to counteracting the perturbative effects associated with the J_2 term.
With regard to the numerical implementation, in [3] [4] the MPC algorithm is formulated using modeling environments for OCP formulation such as CVX or CVXPY, employed within simulations developed in MATLAB or Python. In the reference works, the authors also resort to conceptual block-diagram representations to describe the simulation flow and the information exchange between the MPC controller and the plant, highlighting which quantities are updated and transferred at each iteration; however, the use of Simulink as a simulation environment is not explicitly stated. In the present work, the formulation of the OCP, denoted by $\mathbb{P}_N(\tilde{x}(i), t_i)$ and defined in Eqs. (3.13), is implemented using YALMIP, a MATLAB toolbox for optimization modeling. In this context, YALMIP is used as a tool to express the mathematical formulation of the control problem in a structured manner and to interface it with external numerical solvers used to compute the solution. It therefore plays a role analogous to that of CVX, albeit within a different implementation framework.
It is important for the reader to bear in mind that, although the maneuver scenario and the general structure of the MPC algorithm are similar, $\mathbb{P}_N(\tilde{x}(i), t_i)$ is not defined in an identical manner as the OCPs considered in the reference works. In particular, the latter include additional constraints and convexification procedures, which substantially modify the problem formulation. These aspects will be discussed in detail later in the text. Consequently, a direct quantitative comparison between the results obtained in the two cases is meaningful only within certain limits, since the observed differences are primarily attributable to the different control problem definitions and, to a lesser extent, to different implementation choices and solution methods.
Furthermore, unlike what is done in [3] [4], the plant considered here does not account for Solar Radiation Pressure nor for third-body perturbations. This choice is motivated by the objective of limiting model complexity and focusing the analysis on the behavior of the MPC controller in the presence of the main dynamical effects only.
In [3] [4], the use of a single low-thrust engine that can be reoriented as needed is assumed. In formation reconfiguration maneuvers, convex slew-rate constraints are therefore introduced in order to realistically limit the rate at which the thrust direction can be changed. Moreover, collision avoidance constraints between the various deputies in the formation, suitably convexified, are included in the OCP. In the present work, the introduction of a slew-rate constraint is not necessary, as anticipated in Section 3.1. As for collision

avoidance, it is not considered here, since the described scenario involves a single deputy only, and such a constraint would therefore have no effect. Nevertheless, its introduction certainly represents one of the next useful steps toward achieving FF.
In the reference works [3] [4], a constraint on the maximum thrust magnitude is also introduced, and an attempt is made to simulate the presence of a lower bound on the thrust magnitude (a constraint that is never fully implemented, as discussed later). The absence of a lower-bound thrust constraint in the OCP $\mathbb{P}_N(\tilde{x}(i), t_i)$ (Eqs. (3.13)) is due to the formulation of the optimization problem as a convex program, i.e., the minimization of convex functions over convex sets. Introducing a constraint on the lower thrust limit would require the use of a non-convex set, causing the optimization problem to lose its convexity. This issue is circumvented by ignoring control vectors $u(i)$ whose norm is below the minimum threshold. As a result, a saturation is applied outside the control law: the sequence of acceleration vectors applied to the plant in [3] [4] is not the direct solution of the OCP, but rather a subsequent approximation. Consequently, the lower thrust limit is not implemented in a rigorous manner. In the present thesis, the objective is to formulate the OCP $\mathbb{P}_N(\tilde{x}(i), t_i)$ as a quadratic program, i.e., an optimization problem with a quadratic cost function and a polyhedral constraint set. The constraint on the maximum thrust magnitude is therefore introduced through the set $\mathbb{X}_u$ defined in Section 3.1, whereas the lower thrust bound is neglected. In the present scenario, due to the choice of not applying thrust in the radial direction, this set is modified accordingly and takes the form

$$\mathbb{X}_u = \{\, u \in \mathbb{R}^3 \mid u_1 = 0 \ \wedge \ -u_{\max} \leq u_i \leq u_{\max}, \ \forall i \in \mathbb{I}_{2:3} \,\}. \tag{4.6}$$

In Subsection 4.2.2, the components u_1, u_2, and u_3 are also referred to as u_R, u_T, and u_N, respectively, according to the RTN reference frame.

4.2.1 Settings and desiderata

In the present simulation scenario, the plant used for orbital dynamics propagation coincides with the one defined and validated in Section 4.1. This high-fidelity model is coupled with the software assumed to be implementable on board, whose simulation flow scheme is illustrated in Fig. 4.6. In particular, the onboard software includes the computation of the ROE, the generation of matrices A and B (from which matrices $\underline{A}$, $\underline{B}$, $\tilde{A}$, $\tilde{B}$, and A_{sp} are subsequently derived), and the MPC algorithm, which outputs the control action to be applied to the deputy satellite.
The MPC algorithm is implemented within a MATLAB function, which is called inside the Simulink scheme through the Interpreted MATLAB Function block. The simulation is carried out with a fixed integration step of 10 s using the `ode4` solver. Within this framework, the MPC controller is invoked with a sample time $\Delta t = 100$ s, which coincides, as explained in Section 3.1, both with the time interval at which the control action $u(i)$ is updated during the simulation and with the discretization step used by the algorithm for predicting the dynamics by means of the internal plant model. This choice regarding the ratio between sample time and integration step provides an adequate compromise between simulation accuracy and computational burden. Indeed, selecting an integration step smaller than the controller sample time allows the evolution of the

state vector to be reconstructed through multiple integration steps within the interval $[t_i - \Delta t,\, t_i)$, thereby reducing the numerical error associated with the adopted integration method. On the other hand, an excessively small integration step, while marginally improving propagation fidelity, would lead to an unnecessary increase in computational complexity without providing significant benefits in controller performance.
The simulation is carried out over a total time interval equal to $7T_{c0}$, where T_{c0} denotes the initial osculating orbital period of the chief satellite.
Within the aforementioned MATLAB function, the OCP is formulated and the optimization solver is invoked at each sampling instant. As previously stated, this formulation is implemented using the YALMIP toolbox [23] [26] [24]. The parameterized optimization problem is defined through the `optimizer` function [25], which allows an efficient implementation of the controller. The solver employed to solve the OCP is MOSEK [30], version 11.0.27, selected for its high reliability and robustness in solving quadratic programs.
The prediction horizon is chosen so as to cover a time window of duration $N\Delta t$, as defined previously. In the considered case, the value $N = 5$ is adopted.
The following describes the selection of the weights Q and R used in the cost function $V_N(\cdot)$. Due to the structure of the vector $\tilde{x}(k \mid i)$, the matrix Q is imagined as

$$Q = \begin{bmatrix} Q_{\text{state}} & 0_{7\times 3} \\ 0_{3\times 7} & R_u \end{bmatrix}, \tag{4.7}$$

where $Q_{\text{state}} \in \mathbb{R}^{7\times 7}$ and $R_u \in \mathbb{R}^{3\times 3}$. The matrices Q_{state}, R_u, and R are diagonal and are constructed as follows. Using the subscript j to denote the (j, j) element of a matrix, the elements of Q_{state}, R_u, and R are preliminarily computed according to Bryson's rule. Denoting by δa_r, $\delta\lambda_r$, $\delta e_{x,r}$, $\delta e_{y,r}$, $\delta i_{x,r}$, and $\delta i_{y,r}$ the first six elements of vector r, the vector $z \in \mathbb{R}^{6\times 1}$ is defined as

$$z := \begin{pmatrix} |\delta a - \delta a_r| \\ |\delta\lambda - \delta\lambda_r| \\ |\delta e_x - \delta e_{x,r}| \\ |\delta e_y - \delta e_{y,r}| \\ |\delta i_x - \delta i_{x,r}| \\ |\delta i_y - \delta i_{y,r}| \end{pmatrix}. \tag{4.8}$$

Defining $z_{max} \in \mathbb{R}^{6\times 1}$ as the vector whose elements represent, in terms of order of magnitude, the maximum acceptable values during the simulation for the corresponding elements of z, and considering $\Delta u_{min} = -\Delta u_{max}$, the following preliminary expressions are adopted:

$$Q_{state,j} = \frac{1}{z_{max,j}^2}, \qquad j = 1, \ldots, 6, \tag{4.9}$$

$$R_{u,j} = \frac{1}{u_{max}^2}, \qquad j = 1, 2, 3, \tag{4.10}$$

$$R_j = \frac{1}{\Delta u_{max}^2}, \qquad j = 1, 2, 3. \tag{4.11}$$

In the considered scenario, the following choice is made:

$$z_{max} = 10^{-6} \begin{pmatrix} 10 \\ 1 \text{ rad} \\ 1 \\ 1 \\ 1 \text{ rad} \\ 1 \text{ rad} \end{pmatrix}. \tag{4.12}$$

As anticipated in Section 3.1, the maximum control acceleration is given by $u_{max} = T_{max}/m_d$. Using $T_{max} = 0.65$ mN and $m_d = 20$ kg, which are the same values adopted in [3] [4], one obtains $u_{max} = 0.0325$ mm/s^2. Moreover, in the absence of specific references, $\Delta u_{max} = 0.51 u_{min}$ is adopted, where $u_{min} = 0.0175$ mm/s^2 is the lower thrust limit used in the reference works.
It is observed that, in Eq. (3.12), the element $Q_{state,7}$ multiplies a quantity that is identically zero, since the seventh element of vector $\tilde{x}(k \mid i)$ is ΔB, which is here assumed to be constant (and therefore equal to the last element of vector r). Consequently, the value assigned to the diagonal element $Q_{state,7}$ does not directly affect the value of the cost function $V_N(\cdot)$ nor the optimal solution of the control problem. However, in order to preserve a numerically well-conditioned structure of matrix Q and to avoid introducing excessively different scales among the diagonal elements, $Q_{state,7}$ is assigned a finite value of the same order of magnitude as the other coefficients. In particular, $Q_{state,7}$ is set equal to the minimum among the diagonal elements of Q_{state} effectively used in the simulation. The values of the weights obtained through Bryson's rule provide only an initial estimate of the order of magnitude of the coefficients and are not generally optimal. They are therefore used as a starting point for the controller tuning process. Starting from this preliminary choice, the matrices Q_{state}, R_u, and R are subsequently refined through an empirical tuning procedure based on repeated trial-and-error simulations, with the aim of selectively prioritizing the control of specific state components and improving the overall controller performance in terms of convergence speed and tracking accuracy. The final

values used in the simulation, resulting from this heuristic procedure, are reported below:

$$
Q_{\text{state}} = \begin{bmatrix}
10^{10} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1.7 \cdot 10^{15}\ \text{rad}^{-2} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 \cdot 10^{13} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 3 \cdot 10^{13} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 10^{13}\ \text{rad}^{-2} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 10^{12}\ \text{rad}^{-2} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 10^{12}\ \text{kg}^2/\text{km}^4
\end{bmatrix}
$$

$$
R_u = \begin{bmatrix}
946.746 & 0 & 0 \\
0 & 946.746 & 0 \\
0 & 0 & 946.746
\end{bmatrix} \text{s}^4/\text{mm}^2
$$

$$
R = \begin{bmatrix}
12554.04 & 0 & 0 \\
0 & 12554.04 & 0 \\
0 & 0 & 12554.04
\end{bmatrix} \text{s}^4/\text{mm}^2
\tag{4.13}
$$

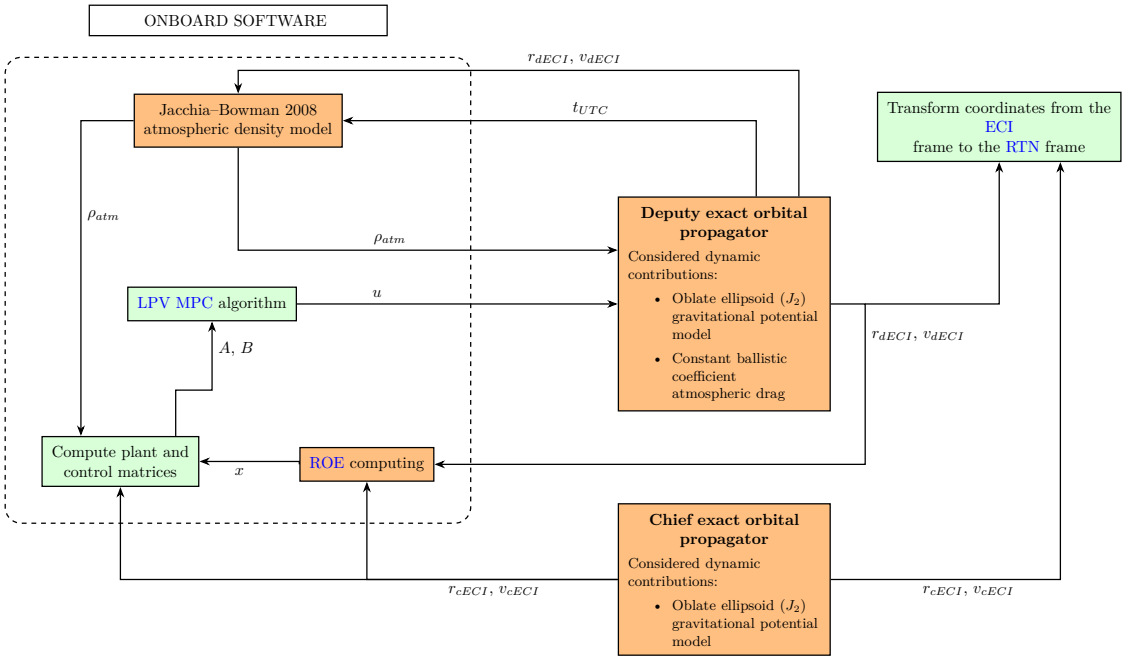


Figure 4.6: Onboard software simulation flow scheme

4.2.2 Results

Fig. 4.7 shows the time evolution of the ROE multiplied by the initial semimajor axis of the chief satellite a_{c0}. The obtained results are satisfactory. Indeed, the reconfiguration maneuver is completed in approximately 3.5 chief orbital periods T_{c0}, with an error

smaller than 1 m for $a_{c0}\delta i_x$ and smaller than 2 m for $a_{c0}\delta i_y$. In the remaining part of the simulation, the error with respect to the target value remains stable for these two quantities. The error on the remaining ROE components is also on the order of meters. However, while the error in $a_{c0}\delta a$ and $a_{c0}\delta\lambda$ remains bounded, the error in $a_{c0}\delta e_x$ and $a_{c0}\delta e_y$ exhibits an approximately constant secular growth. Considering that T_{c0} is about one and a half hours, this growth is relatively fast: after one day (corresponding to nearly 16 chief orbital periods), the error in $a_{c0}\delta e_x$ reaches approximately 6 m, while that in $a_{c0}\delta e_y$ reaches approximately 4 m.
This behavior highlights the inability of the implemented controller to fully counteract the combined effects of the considered perturbations on the vector δe, which would be necessary to achieve formation keeping after completion of the reconfiguration maneuver. As previously stated, however, formation keeping lies outside the scope of the present work, which focuses exclusively on initialization and reconfiguration maneuvers. The observed tendency toward maintaining the remaining four scaled ROE acquired after the first 3.5 periods T_{c0} should therefore not be interpreted as the result of a dedicated formation-keeping strategy, but rather as a consequence of the continuous action of the implemented MPC algorithm throughout the entire simulation.

Fig. 4.8 provides a clearer visualization of the executed maneuver by representing the relative trajectory of the deputy satellite in the RTN frame. It is important to note that the blue and red curves represent osculating relative orbits, computed at the initial and final instants of the simulation, respectively, using the data reported in Table 4.1. Fig. 4.9 shows the time history of the control acceleration components expressed in the RTN frame. As assumed during the controller design phase, the radial thrust component is identically zero, and control is therefore entirely entrusted to the transverse and normal components. In particular, the normal component u_N is the most heavily utilized during the reconfiguration phase, reaching values close to saturation during the initial stages of the maneuver, which highlights the dominant role of out-of-plane dynamics. After completion of the maneuver (i.e., after approximately 3.5 chief orbits), the magnitude of u_N decreases significantly, remaining confined within low values and exhibiting a predominantly oscillatory behavior.
The transverse component u_T instead assumes significantly smaller magnitudes and exhibits an almost periodic behavior throughout the entire simulation. This behavior is primarily attributable to the compensation of the considered perturbations, in particular aerodynamic drag.
It is also observed that, once the reconfiguration phase is completed, all control components remain well below the imposed saturation limits, indicating that the MPC algorithm continues to operate by counteracting residual perturbations without requiring large control accelerations. Fig. 4.10 shows the time evolution of the ellipsoidal altitude h of the deputy satellite throughout the entire simulation. This quantity is computed within the developed Simulink scheme using the Geocentric to Geodetic Latitude block, adopting the WGS84 Earth model [31].
The plot allows verification that the deputy satellite altitude remains within the validity range of the Jacchia–Bowman 2008 atmospheric model used to compute the atmospheric

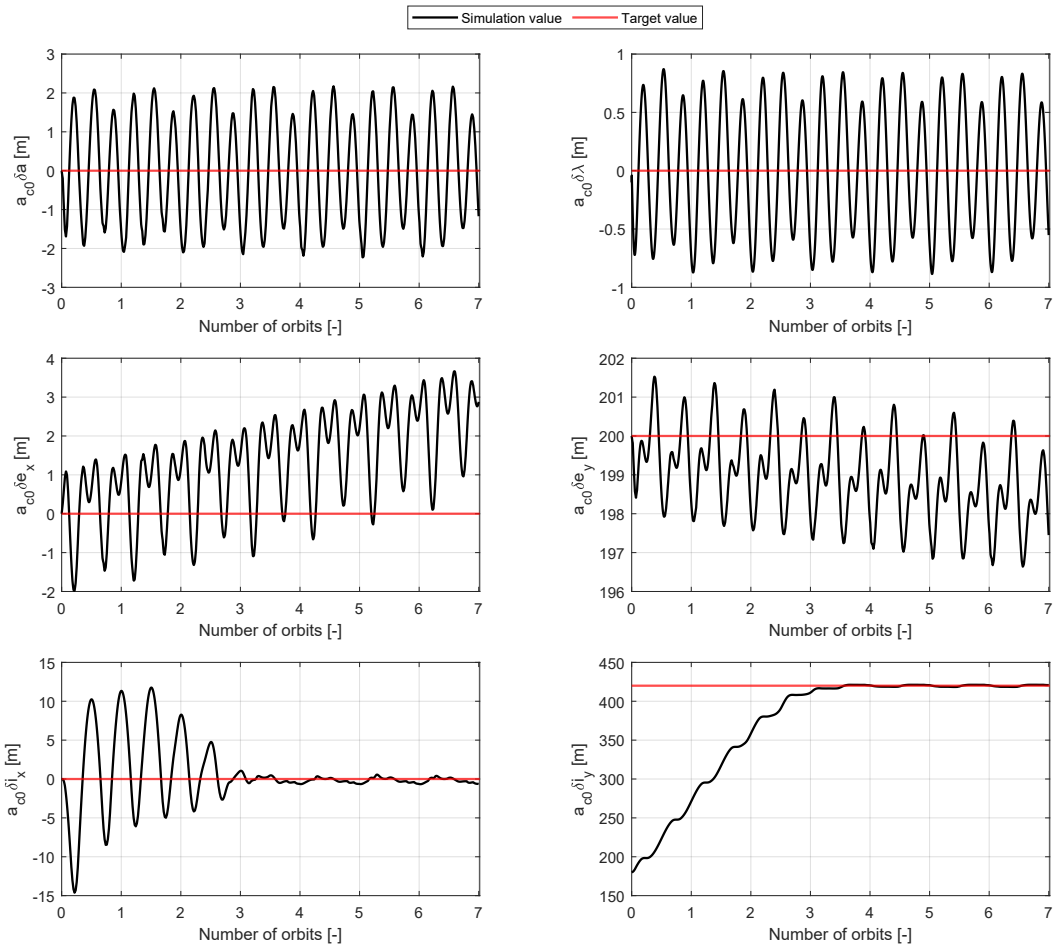


Figure 4.7: Time history of the scaled ROE

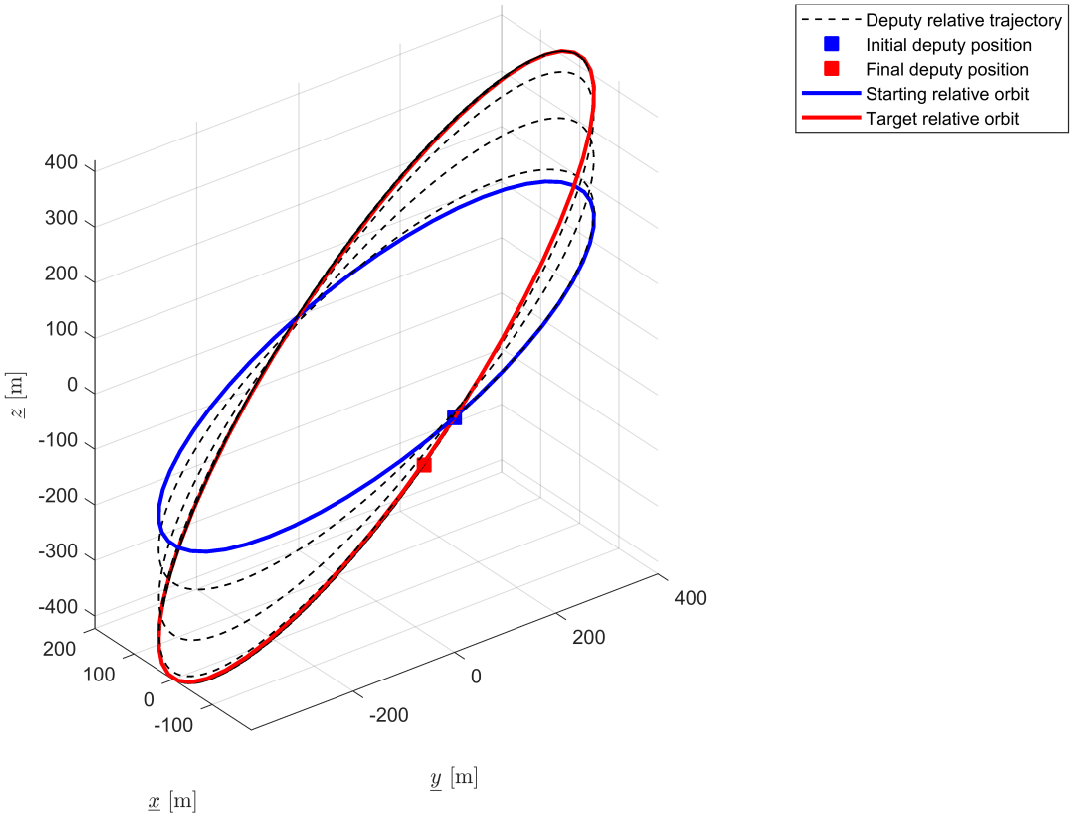


Figure 4.8: Simulated 3D relative trajectory of the deputy satellite in the RTN frame

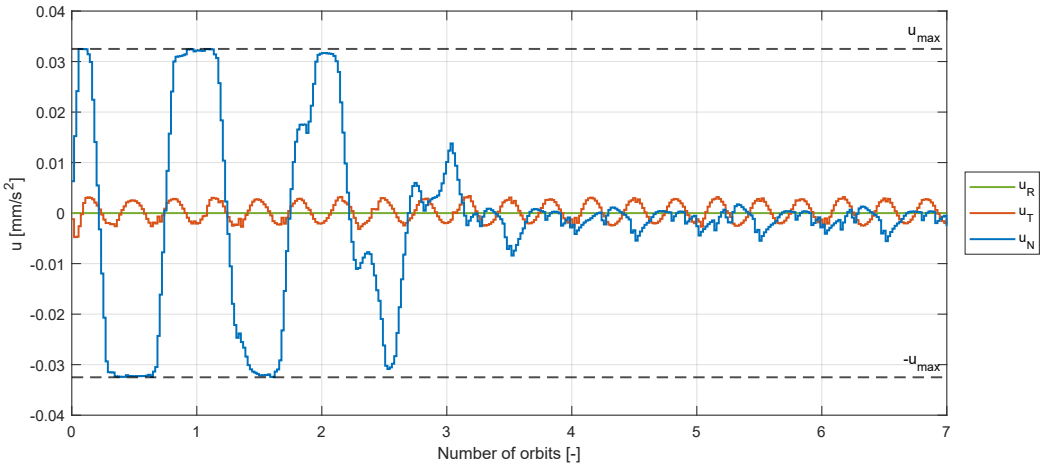


Figure 4.9: Control acceleration profiles in the RTN frame resulting from the numerical simulation

density required for aerodynamic drag evaluation. This model is considered valid for altitudes approximately between 175 km and 1000 km [5]. Since the altitude remains within this range for the entire duration of the simulation, the aerodynamic drag contribution modeled in this work can be considered physically consistent for the analyzed orbital scenario. Moreover, it clearly emerges that the deputy satellite operates throughout the simulation within the VLEO regime, consistently with the considered orbital scenario. The overall simulation required a total computation time of 134 s. This quantity, how-

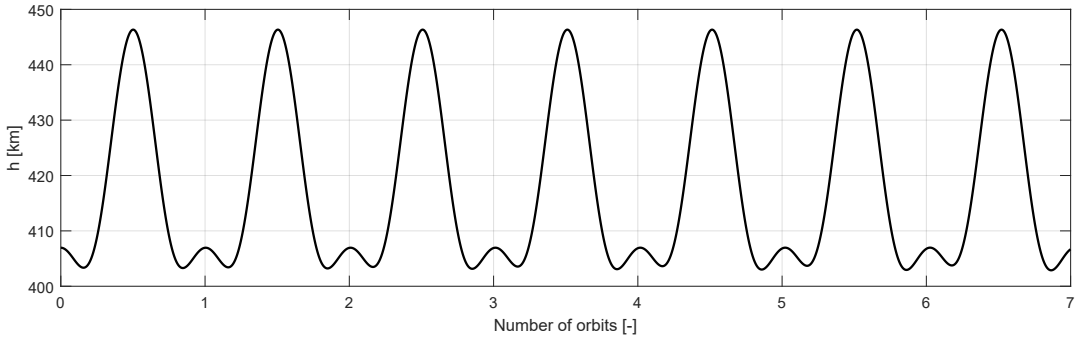


Figure 4.10: Ellipsoidal altitude profile of the deputy satellite resulting from the numerical simulation

ever, cannot be used to draw conclusions about the computational performance of the implemented MPC algorithm, since it also includes the numerical propagation of the plant dynamics in the Simulink model. In order for the algorithm to be implementable onboard, its execution time (i.e., the time to solve the formulated OCP) must be smaller than the sampling time Δt at each sampling instant. For this reason, the OCP solution time was measured at each sampling instant. Denoting by $t_{\text{solve}}(i)$ the time required by the controller, when invoked at the sampling instant t_i, to solve the corresponding OCP, the measured sequence is reported in Fig. 4.11.
The execution time appears approximately constant during most of the simulation (which can be regarded as the nominal operating regime of the controller), while a limited number of isolated peaks are observed. To quantify the variability of the execution times, the mean value and the standard deviation were computed. Denoting by $N_s := \lfloor T_{sim}/\Delta t \rfloor + 1$ the total number of sampling instants, these two quantities are defined as

$$\begin{aligned} t_{\text{mean}} &:= \frac{1}{N_s} \sum_{i=0}^{N_s-1} t_{\text{solve}}(i), \\ \sigma &:= \sqrt{\frac{1}{N_s - 1} \sum_{i=0}^{N_s-1} \left(t_{\text{solve}}(i) - t_{\text{mean}}\right)^2}. \end{aligned} \tag{4.14}$$

In order to obtain a representative value of the execution time in the nominal regime, while avoiding excessive sensitivity to sporadic peaks, a statistical upper bound was defined as

$$t_{\text{stat}} := t_{\text{mean}} + 3\sigma. \tag{4.15}$$

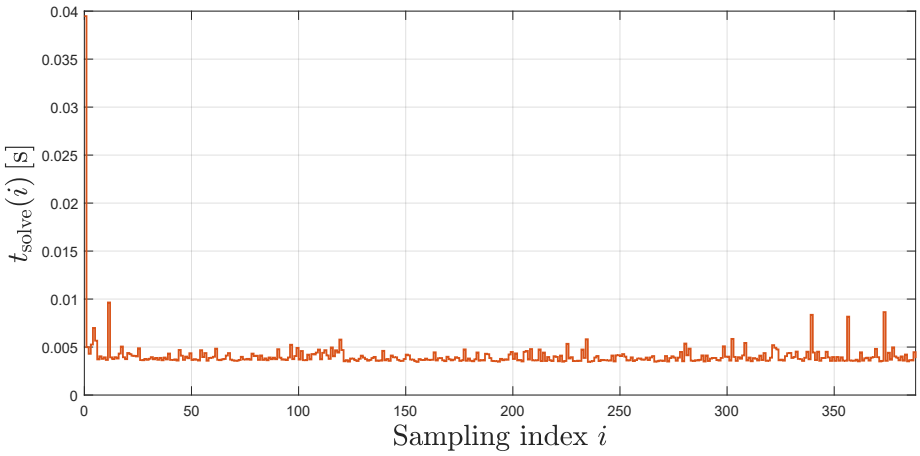


Figure 4.11: Measured OCP solution time $t_{\text{solve}}(i)$ as a function of the sampling index. After an initial transient peak, the execution time settles to an approximately constant nominal value with sporadic isolated spikes.

This quantity provides a conservative estimate of the typical solver execution time. It is more informative than the mean value alone and less affected by isolated peaks than the maximum observed time $t_{\max}$. The value t_{stat} therefore characterizes the typical worst-case execution time associated with the regular operation of the solver, while $t_{\max}$ accounts for occasional peaks. Table 4.4 reports the obtained values.

Table 4.4: Statistical characterization of the MPC solver execution time.

Performance metric	**Value** [s]
t_{mean}	0.0041
σ	0.0019
t_{stat}	0.0098
$t_{\max}$	0.0395

The condition $t_{\max} < \Delta t$ indicates that the control input was always computed before the next sampling instant during the simulation. Moreover, the fact that t_{stat} remains well below Δt indicates that the available computation time is significantly larger than the typical execution time, showing that the timing feasibility of the controller does not depend on isolated favorable events.
It is worth emphasizing that this analysis refers to the computational platform used for simulation. It shows that the implemented control strategy is not computationally prohibitive and satisfies the timing requirements imposed by the sampling interval in the considered setup. However, it does not directly guarantee execution on flight hardware, whose computational capabilities may differ.

Chapter 5

Conclusions

This thesis addresses the problem of formation flight in VLEO missions within a leader-follower architecture involving one chief spacecraft and one deputy. The objective has been to design an MPC-based control strategy for formation initialization and reconfiguration.
A review of existing formation flight control strategies and modeling approaches has been carried out in order to position the present work within the current state of the art. The approach adopted in this work builds upon the ROE-based linearized framework commonly used in formation flight literature and applies it to a VLEO scenario, a regime of increasing interest for future LEO missions involving small satellites, in which the dominant perturbations are atmospheric drag and J_2. In this context, the modeling of these effects plays a central role in the design of the controller.
The relative motion has been described using quasi-nonsingular ROE and linearized around the null ROE state vector (corresponding to the instantaneous position of the chief spacecraft). The plant matrices associated with Keplerian motion, J_2 perturbation, and atmospheric drag have been adopted from established formulations available in the literature. The differential ballistic coefficient ΔB, here assumed constant, has been introduced in order to represent the non-conservative perturbation due to aerodynamic drag within the linear relative dynamics.
Since the plant matrices depend on the osculating orbital elements of the chief, the linearized model is, in general, intrinsically Linear Time-Varying (LTV). In the MPC implementation, the system matrices are explicitly recomputed at each sampling instant using the current value of the chief orbital elements, resulting in a LPV predictive formulation.
The OCP has been formulated as a strictly convex quadratic program. This formulation ensures a unique global solution at each sampling instant of the algorithm. The cost function penalizes deviations from the desired relative configuration, control effort, and variations of the control input. Upper bounds on thrust magnitude have been introduced, while lower-bound thrust limits have been intentionally neglected in order to preserve the convex structure of the problem.
The plant used for closed-loop simulations has been implemented in MATLAB/Simulink

using a high-precision orbital propagator. In the considered maneuver, the deputy propagator includes both J_2 and aerodynamic drag effects, whereas the chief follows a non-decaying trajectory, being subjected to J_2 perturbation only. The adopted propagation scheme has been validated against reference results available in the literature.
The numerical results demonstrate that the proposed LPV-MPC controller is capable of steering the deputy from an initial configuration to a desired relative geometry while satisfying thrust constraints and compensating for perturbation-induced effects. The timing analysis of the solver execution shows that both the average computational time and the statistical worst-case execution time remain below the adopted sampling interval in the simulated setup on the adopted computational platform (specific hardware configuration to be reported), indicating that the implemented strategy can be executed within the selected control update rate.
Despite the achieved results, several aspects of the proposed framework can be further investigated and improved.

5.1 Future works

Several developments can extend and strengthen the present work.
A first direction concerns a quantitative assessment of propellant efficiency. Such an analysis would require the definition of a suitable benchmark solution, for example an impulsive analytical strategy or a nonlinear minimum-fuel trajectory, together with a systematic tuning of the cost-function weights. Since the relative weighting between state deviation and control effort directly shapes the control law, different mission objectives would lead to different tuning strategies. A structured comparison would therefore be necessary before drawing conclusions on fuel optimality.
Another relevant continuation of this work concerns the development of a control setup specifically aimed at formation keeping. The controller implemented in this thesis has been designed and tuned primarily for finite-time formation acquisition and reconfiguration. The closed-loop simulations show that, although several ROE components converge satisfactorily to their target values, the osculating relative eccentricity components exhibit residual oscillations with a slow drift over time. This behavior suggests that the current formulation, while effective for reconfiguration, is not fully suited for long-term steady maintenance of the desired relative geometry. A dedicated reformulation and tuning of the MPC problem would therefore be required.
The inclusion of spacecraft attitude dynamics could also represent a natural extension. In the present work, the deputy is modeled as a point mass with its attitude assumed fixed and permanently aligned with the RTN frame. This assumption allows the thrust components to be directly interpreted as forces along fixed RTN directions and fully decouples the position dynamics from the attitude dynamics. Introducing a six-degrees-of-freedom model would instead require explicitly modeling attitude kinematics and rigid-body dynamics. In such a framework, the thrust vector would depend on the instantaneous spacecraft attitude, and the same actuators could be required to contribute simultaneously to both translational and rotational control. Moreover, the current thruster configuration, in which each engine is aligned with a principal axis

and therefore generates no control torque, would no longer be sufficient. A consistent extension would therefore involve revisiting the geometric arrangement of the thrusters and reformulating the control problem to account for the coupled position and attitude dynamics.
Furthermore, the framework could be extended to formations composed of multiple deputies. Increasing the number of spacecraft would enlarge the dimension of the predictive model and require the introduction of inter-satellite safety constraints. Collision avoidance constraints are inherently non-convex when expressed as minimum-distance requirements in $\mathbb{R}^3$. Embedding them within the current convex MPC formulation would therefore require suitable convexification strategies in order to preserve the quadratic programming structure of the optimization problem. The formulation and validation of such convex approximations, while controlling the growth of computational complexity as the number of satellites increases, would represent a relevant continuation of this work.
Finally, a possible further improvement concerns actuator modeling. In this work, as discussed above, only upper bounds on the thrust are enforced in order to preserve a convex formulation. If a minimum thrust level were to be imposed while still allowing the thruster to be switched off, the actuator would need to be modeled with an explicit on/off logic. This would naturally lead to the introduction of binary decision variables and to a Mixed-Integer Quadratic Programming (MIQP) formulation and a consequent loss of convexity. Such an extension has not been considered in the present work.

Bibliography

[1] B. Andrievsky and Kostin I. Fadeeva J. Popov, A. M. Modeling and control of satellite formations: A survey. Automation, 3(3):511–544, 2022.

[2] R. Battin. An introduction to the Mathematics and Methods of Astrodynamics. American Institute of Aeronautics and Astronautics, 1999.

[3] Enrico Belloni. On-board model predictive control for optimal formation acquisition and maintenance in high-drag environments. Master's thesis, Politecnico di Milano, Milan, Italy, 2018.

[4] Silvestrini S. Prinetto J. Lavagna M. Belloni, E. Relative and absolute on-board optimal formation acquisition and keeping for scientific activities in high-drag low-orbit environment. Advances in Space Research Volume 73, 2024.

[5] Bruce R. Bowman, W. Kent Tobiska, Frank A. Marcos, Cheryl Y. Huang, Chin S. Lin, and William J. Burke. A new empirical thermospheric density model jb2008 using new solar and geomagnetic indices. In Proceedings of the AIAA/AAS Astrodynamics Specialist Conference, number AIAA 2008-6438. American Institute of Aeronautics and Astronautics, 2008.

[6] H. Cho and A. Yu. New approach to satellite formation-keeping: Exact solution to the full nonlinear problem. Journal of Aerospace Engineering, 22(4):445–455, 2009.

[7] H.-C. Cho and S.-Y. Park. Analytic solution for fuel-optimal reconfiguration in relative motion. Journal of Optimization Theory and Applications, 141(3):495–512, 2009.

[8] W. H. Clohessy and R. S. Wiltshire. Terminal guidance system for satellite rendezvous. Journal of Aerospace Science, 27(9):653–658, 1960.

[9] S. D'Amico. Relative orbital elements as integration constants of hills equations. German Aerospace Center (Deutsches Zentrum fur Luft- und Raumfahrt)-German Space Operations Center, TR 05-08, WeÃling, Germany, 2005.

[10] S. D'Amico. Autonomous Formation Flying in Low Earth Orbit. PhD thesis, Delft University of Technology, Delft, The Netherlands, 2010.

[11] S. D'Amico and O. Montenbruck. Proximity operations of formation-flying spacecraft using an eccentricity/inclination vector separation. Journal of Guidance, Control, and Dynamics, Vol. 29, No. 3, 2006, 2006.

[12] de Ruiter A. H. J., Damaren C. J., and Forbes J. R. Spacecraft Dynamics and Control: An Introduction. John Wiley & Sons, Chichester, UK, 2013.

[13] Lawn. M. Bevilacqua R. Di Mauro, G. Survey on guidance navigation and control requirements for spacecraft formation-flying missions. JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS Vol. 41, No. 3, 2018.

[14] D'Amico S. Lavagna M. Guffanti, T. Long-term analytical propagation of satellite relative motion in perturbed orbits. 27th AAS/AIAA Space Flight Mechanics Meeting, 1980.

[15] P. Gurfil. Relative motion between elliptic orbits: Generalized boundedness conditions and optimal formationkeeping. Journal of Guidance, Control, and Dynamics, 28(4):761–767, 2005.

[16] Q. He and C. Han. Dynamics and control of satellite formation flying based on relative orbit elements. In Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, Hawaii, 2008. 18–21 August 2008.

[17] Gokhan Inalhan, Michael Tillerson, and Jonathan P. How. Relative dynamics and control of spacecraft formations in eccentric orbits. Journal of Guidance, Control, and Dynamics, 25(1):48–59, 2002.

[18] B.; Park S.Y.; Choi K.H. Kim, D.Y.; Woo. Hybrid optimization for multiple-impulse reconfiguration trajectories of satellite formation flying. Adv. Space Res., 44, pp. 1257–1269, 2009.

[19] Guffanti T. Koenig, A. and S. D'Amico. New state transition matrices for spacecraft relative motion in perturbed orbits. AIAA/AAS Astrodynamics Specialist Conference, AIAA Paper 2016-5635, 2016.

[20] K. D. Kumar, H. C. Bang, and M. J. Tahk. Satellite formation flying using along-track thrust. Acta Astronautica, 61(6–7):553–564, 2007.

[21] J. J. F. Liu. Satellite Motion about an Oblate Earth. AIAA Journal, 12(11):1511–1516, 1974.

[22] J.J.F. Liu and R.L. Alford. Semianalytic theory for a close-earth artificial satellite. Journal of Guidance and Control, 1980.

[23] Johan Löfberg. Yalmip: A toolbox for modeling and optimization in matlab. In Proceedings of the CACSD Conference, Taipei, Taiwan, 2004.

[24] Johan Löfberg. Yalmip example: Standard mpc. https://yalmip.github.io/example/standardmpc/, 2016.

[25] Johan Löfberg. Yalmip optimizer object documentation. https://yalmip.github.io/solver/optimizer/, 2016.

[26] Johan Löfberg. Yalmip tutorial: Basics. https://yalmip.github.io/tutorial/basics/, 2016.

[27] Meysam Mahooti. Jacchia-bowman atmospheric density model, 2024. MathWorks File Exchange. Available at: https://it.mathworks.com/matlabcentral/fileexchange/56163-jacchia-bowman-atmospheric-density-model.

[28] MathWorks. Numerical high-precision orbit propagator, 2024. Available at: https://it.mathworks.com/help/aeroblks/orbitpropagatornumericalhighprecision.html.

[29] MathWorks. Orbit propagator algorithms, 2024. Available at: https://it.mathworks.com/help/aeroblks/orbit-pop-algorithms.html#mw_7ec8d2be-2b7a-4e1b-8384-1acb9e3586af.

[30] MOSEK ApS. The mosek optimization toolbox for matlab manual. https://www.mosek.com/documentation/.

[31] National Geospatial-Intelligence Agency (NGA). NGA.STND.0036_1.0.0_WGS84,

July 2014. https://nsgreg.nga.mil/doc/view?i=4085.

[32] P. Palmer. Optimal relocation of satellites flying in near-circular-orbit formations. Journal of Guidance, Control, and Dynamics, 29(3):519–526, 2006.

[33] Mayne D. Q. Diehl M. M. Rawlings, J. B. Model Predictive Control: Theory, Computation, and Design, 2nd Edition. Nob Hill Publishing, 2019.

[34] Sorensen N. Ren, W. Distributed coordination architecture for multi-robot formation control. J. Robot. Autonom. Syst., vol. 56, no. 4, pp. 324–333, 2008.

[35] D'Amico S. Riggi, L. Optimal impulsive closed-form control for spacecraft formation flying and rendezvous. In Proceedings of the 2016 American Control Conference (ACC), 2016.

[36] Prilulsky O.F. Rodin V.G. Sholomitsky, G.B. Infra-red space interferometer. In 28th Int. Astro. Fed. Congress, Praha, Czechoslovakia, Paper IAF-77-68, 1977.

[37] Lukas M. Steindorf, Simone D'Amico, Julian Scharnagl, Florian Kempf, and Klaus Schilling. Constrained low-thrust satellite formation-flying using relative orbit elements. In Proceedings of the AAS/AIAA Astrodynamics Specialist Conference, 2017. Paper AAS 17-291.

[38] The MathWorks, Inc. ode45 – solve nonstiff differential equations, medium order method. https://www.mathworks.com/help/matlab/ref/ode45.html.

[39] Hadaegh F. Y. Wang, P. K. C. Minimum-fuel formation reconfiguration of multiple free-flying spacecraft. J. Astronaut. Sci., vol. 47, no. 1, pp. 77–102, 1999.

[40] Wu B. Poh E. K. Wang, D. Satellite Formation Flying: Relative Dynamics, Formation Design, Fuel Optimal Maneuvers and Formation Maintenance. Springer, 2017.

[41] Ahn H.-S. Won, C.-H. Nonlinear orbital dynamic equations and state-dependent riccati equation control of formation flying satellites. Journal of the Astronautical Sciences, 51(4):433–449, 2003.

[42] Jian Xing, Guohua Tang, Xiaoqiang Xi, and Hong Li. Satellite formation design and optimal stationkeeping considering nonlinearity and eccentricity. Journal of Guidance, Control, and Dynamics, 30(5):1523–1528, 2007.

[43] Sparks A. Yeh, H. H. Geometry and control of satellite formations. In Proc. Amer. Control Conf., 2000.

[44] Liu G.P. Zhang, S. A survey on formation control of small satellites. Proceedings of the IEEE | Vol. 106, No. 3, 2018.