UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Dipartimento di Ingegneria Industriale

Corso di Laurea Magistrale in Ingegneria Aerospaziale

Autonomous Rendezvous with a non-cooperative satellite: trajectory planning and control

Relatore: Dott. Andrea Valmorbida

Correlatore: Prof. Enrico Lorenzini

Laureando: Fortuna Simone

Matricola: 1236495

Anno Accademico 2021-2022

Sommario

Con la nascita di nuove problematiche e nuove esigenze in ambito spaziale, le più importanti riguardanti il tema della mitigazione dei detriti spaziali o dell'assistenza e del servizio dei satelliti in orbita, lo scenario di rendez-vous autonomo tra un satellite inseguitore e un satellite target non cooperativo sta diventando sempre più centrale, ambizioso e accattivante. Il grande scoglio da superare, tuttavia, consiste nell'individuazione di una strategia di approccio robusta e vincente: mentre l'esecuzione di una manovra di rendez-vous e docking o cattura con satellite cooperativo è già stata collaudata e possiede una consolidata eredità di volo, il rendez-vous autonomo con satellite non cooperativo ed in stato di tombolamento è uno scenario agli albori, con pochi studi al riguardo. Lo scopo di questa tesi consiste nell'identificazione di una strategia di approccio che consideri le principali problematiche legate al tema in questione, ovvero la non-cooperazione e le scarse informazioni sullo stato di moto del target da raggiungere. Queste due complicazioni portano alla necessità di eseguire un moto di ispezione del satellite target e alla considerazione di numerosi vincoli nella progettazione della traiettoria di ispezione e di approccio. Un controllore adatto a trattare questo problema complesso e multi-vincolato è il *Model Predictive Controller*, in forma lineare o non lineare, abbinato ad un filtro di Kalman. La capacità di questo controllore di previsione e pianificazione di una traiettoria d'approccio, a partire da stime di posizione relativa tra target e inseguitore, permette di portare a termine la manovra in modo sicuro e robusto.

Abstract

According to the rise of new problems and new demands in the space field, the most important concerning the mitigation of space debris or the spacecraft on-orbit servicing and assistance themes, the Autonomous Rendezvous scenario between a chase satellite and a non-cooperative target satellite is becoming increasingly significant, ambitious, and attractive. The main issue to overcome, however, consists in the identification of a robust and successful approach strategy: while the execution of a rendezvous and docking or capture maneuver with a cooperative satellite has already been tested and holds a solid flight heritage, the autonomous rendezvous with a non-cooperative satellite in a state of tumbling motion is a scenario in the early days, with few studies about it and a not yet mature technology. The aim of this thesis consists in the identification of an approach strategy that deals with the main challenges related to the considered problem, namely non-cooperativeness and exiguous information about the target to be reached. These two issues lead to the need of performing an inspection motion and considering several constraints in the trajectory design. A controller suitable to handle this complex and multi-constrained problem is the Model Predictive Controller, in a linear or non-linear form, paired with a Kalman filter. The ability of this controller to predict and plan an approaching trajectory, starting from estimates of the relative position between the target and the chaser, allows to complete the approaching maneuver safely and in a robust way.

Contents

List of Figures

Chapter 1

Introduction

Rendezvous and Docking or Capture is an interesting on-orbit scenario in which a chase spacecraft arrives in the proximity of a target spacecraft and, after a few maneuvers or operations such as fly-around, finally captures it (or is captured). All these proximity maneuvers required to accomplish docking or capture are commonly enclosed into the Rendezvous and Proximity Operations (RPO).
Thinking about the growing importance given to scientific explorations, it's possible to extend the first definition by considering a generic space object as target instead of the more specific case of a spacecraft.

Nowadays, becoming the new space economy a reality even more, this scenario is getting increasingly popular, especially in the case of Autonomous Rendezvous and Docking or Capture (AR&D/C), where the need to assist and support particular on-orbit operations is gradually replaced by the use of robust Guidance, Navigation and Control systems and increasingly performant on-board computers.
A larger distinction can be made by considering AR&D/C maneuvers with cooperative or non-cooperative target objects: while for operations with cooperative bodies there is an already consolidate background and study (the major example is the ISS refurbishment) and a considerable flight heritage, executing a rendezvous and proximity maneuver with an object that does not interact or cooperate with the chaser is a more recent issue and, as one can intuit, a really challenging matter.

Non-cooperative bodies are all resident space objects, such as orbital debris, de-

funct satellites, and natural objects like asteroids and comets, that are of unknown appearance, uncooperative, and tumbling[37]. All these characteristics turn out to be obstacles for a rendezvous: for example, the knowledge of inertial properties and the state of motion is a crucial aspect for the conception and planning of a possible trajectory of approach, or, to name another, non-cooperativeness implies that the chaser can't rely on transponders, reflectors, artificial markers that allow friendly response to its sensors, but must rely on docking or grappling fixtures eventually offered by the target. All these uncertainties lead to the consideration of robustness approaching to the problem, besides fuel consumption and engineering constraints: the chase satellite has to track both the time-varying desired relative positions and attitude trajectories accurately with respect to a target spacecraft, primarily ensuring safety.

Despite AR&D/C problem with a non-cooperative target results in an undemonstrated, risky, and challenging operation, it has attracted increasing attention in recent years. Why does this scenario attract?

1.1 Non-cooperative AR&D/C performing motivations

As stated by Setterfield[37], there are several reasons to want visiting these resident space objects with an inspector satellite: to deorbit or deflect dangerous debris, to inspect and/or repair defunct satellites (on-orbit servicing) or to make scientific observations of natural objects.

1.1.1 Deorbiting or deflecting Space Debris

Space debris are all man-made objects including fragments and elements thereof, in Earth orbit or re-entering the atmosphere, that are non-functional.
It's possible to group them into these categories:

- Operational spacecraft;
- Defunct spacecraft;
- Spent rocket bodies;

- Mission related objects;

- Explosion fragments.

Debris orbital velocities are extremely high and range from 16 km/s in very Low Earth Orbits (LEO) to 2 km/s in Geostationary Earth Orbits (GEO), therefore it's clear that collision with even small debris can largely damage a satellite, with consequences from a "simple" component malfunctioning to the catastrophic destruction of the entire spacecraft, which implies the obvious loss of the mission.

Large orbital debris poses a risk of colliding with other debris and aggravating the problem, and thus are good candidates for preliminary debris removal efforts.

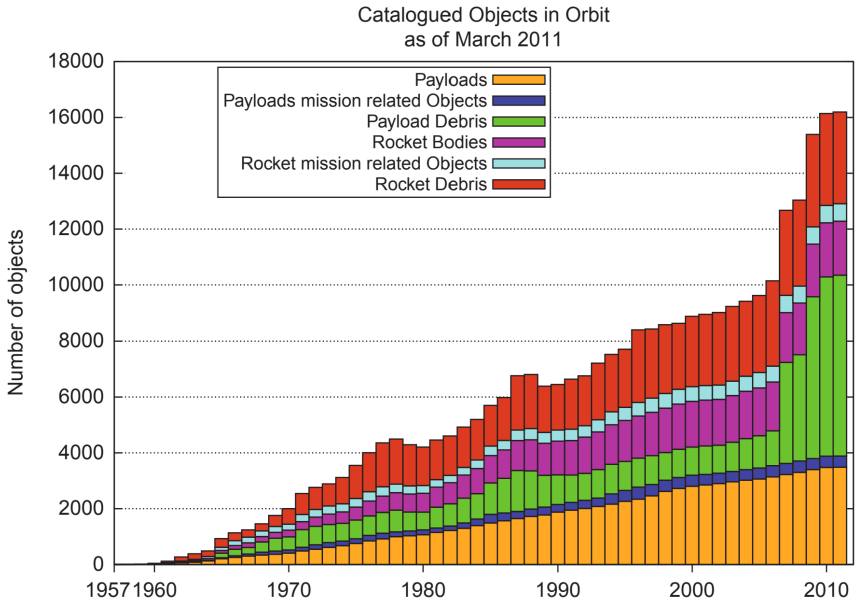


Figure 1.1: Space objects tracked by the US Space Surveillance Network (SSN) [19].

Recent studies have shown that, even without any new launches, the environment has reached a point where collisions among existing debris will result in the population to increase. This fact, called *'Kessler syndrome'* and first predicted by Kessler and Cour-Palais[18], attests how much this scenario is dramatically worrying.
To mitigate the space debris growth, it is imperative to reduce the release of critical-size objects in densely populated orbit regions. The principal mitigation guidelines, issued by various international organizations[34], are listed below:

- limit the number of mission related objects released into space;
- avoid on-orbit break-up;
- limit the orbital resident time spent in space of spacecraft and rocket stages after mission completion (non-manoeuvrable space objects post-mission lifetime should not exceed 25 years).

The various companies and associations are slowly agreeing to these mitigations, and it is now known the need to complement them with environment remediation measures. This can be done by performing "Active Debris Removal" missions, that depend on the space environment where the orbit debris is situated: in LEO the primary issue for proximity operations is to capture and deorbit large and massive debris with a high impact probability with operative satellites, in GEO their capture and relocation to a graveyard orbit.

In Europe, ESA is promoting the Clean Space Program, with the aim of studying and developing new technologies for debris mitigation as well as environment remediation, pursuing the capture and deorbit of debris using nets, simple and electrodynamic tethers (EDTs), harpoons, and ion-beam shepherding[44].

Figure 1.2: Envisat model (courtesy of [11]).

The most striking example with regard to environment remediation is the case of

the *e.deorbit* mission, targeting Envisat, on which the studies have focused the most. Envisat is an 8 tons mass, ESA-owned space debris in the densely populated near-polar region in the 600–800 km altitude range and currently represents a potential trigger for space debris proliferation in low Earth orbit.
Even if ambitious, this project is really challenging, due to Envisat huge mass, complicated capture access (due to tumbling motion), and high collision risk with debris in its current orbit. A possible success, in addition to the prestige acquired by ESA, would encourage and promote other missions, making the space environment increasingly clean and sustainable.

A quite different matter regards deflection purposes, herein not developed. Only an example, due to its recentness, is given: that is the DART (Double Asteroid Redirection Test) mission, belonging to NASA's AIDA (Asteroid Impact & Deflection Assessment) program and aimed at studying a method of planetary defense against minor bodies. It is a recent mission, launched on November 2021 aboard a Falcon 9 and directed towards the Didymos system, a $780m$ asteroid with a $160m$ diameter moon. The probe will act as a kinetic impactor on the small moon to slightly change its orbit. The success or failure of the impact will be necessary for the study of models and solutions to be applied in the event that minor bodies threaten the Earth, passing too close to the surface or even being on a collision course.

Figure 1.3: DART mission (courtesy of [42]).

1.1.2 On-Orbit Servicing

According to Barnhart[4], on-orbit satellite servicing (OOS) or assembling have the potential to foster the next economic revolution in space. The ability to approach, inspect, grasp, assembly, manipulate, repair, refuel, integrate, and build completely new platforms and spacecraft on orbit would enable new business models, innovation, and opportunities in space.

Beyond the operations to be performed at the end of life of a satellite for deorbiting, just think of all the on-orbit failures: many of these could have indeed been remedied (without losing billions of dollars) by servicing if the required systems were available. Several space programs have been carried out as technology demonstration for servicing a cooperative spacecraft, including NASA's Demonstration for Autonomous Rendezvous Technology (DART) program, JAXA's Engineering Test Satellite-VII (ETS-VII) project, DARPA's Orbital Express (OE) program, and AFRL's Experimental Satellite System (XSS) etc. In these RVD missions, the cooperative targets are assumed to move smoothly in different Keplerian orbits without rapid attitude maneuver, making docking easier.

Although autonomous satellite servicing has not yet been carried out on an uncooperative satellite, a few demonstration missions are shyly coming out.

- **Elsa-D**

  According to [5], ELSA-d, which stands for End of Life Services by Astroscale, is an in-orbit demonstration mission (launched in March 2021) for key end-of-life technology and capabilities of future debris removal missions. In particular, it consists of a servicer and small client launched together to demonstrate test capture, tumbling/non-tumbling capture, inspection, search & approach and deorbit. Regarding the latter, ELSA-d represents the world's first commercial Active Debris Removal (ADR) demonstration mission.

- **AVANTI**

  The Autonomous Vision Approach Navigation and Target Identification (AVANTI) experiment, developed and carried out by DLR in November 2016, realized the

first fully-autonomous vision-based rendezvous to a non-cooperative target spacecraft in low Earth orbit[14].

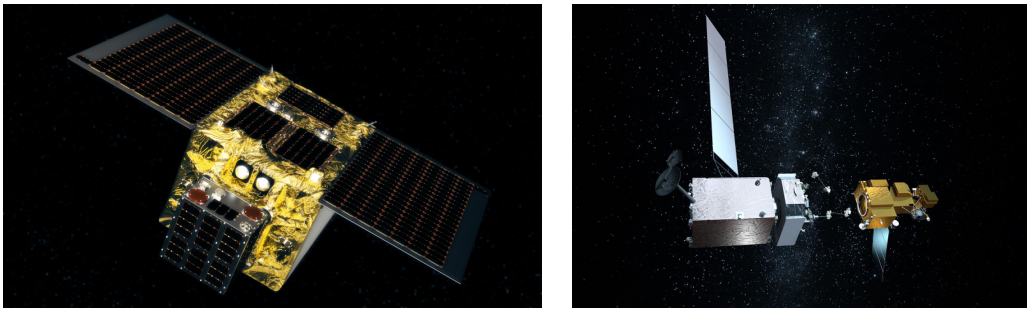
- **DEOS**

  This mission, developed by DLR (German Aerospace Research Institute) and Airbus Defence and Space, is an in-flight technology demonstration to find and evaluate procedures and techniques for rendezvous, capture and de-orbiting of an uncontrolled satellite from its operational orbit. As reported by Reintsema[36], the overall space segment of DEOS consists of two satellites which will perform and demonstrate all the afore mentioned aspects. One spacecraft represents the target to be captured, which is passive, non-cooperative and tumbling, the other one is the active servicing chase satellite.

  According to planning, as reported in [31], DEOS launch was scheduled in 2018, but the project was cancelled after the definition phase.

- **Resore-L or OSAM-1**

  As Vavrina[40] wrote, in the early 2020s, precisely in 2023, NASA will conduct a robotic satellite servicing mission, autonomously capturing the aging Landsat 7 spacecraft in low-Earth orbit (LEO) to refuel and relocate it in its original orbit, extending its mission life. This mission will be presented in more detail, having been chosen as a case study.



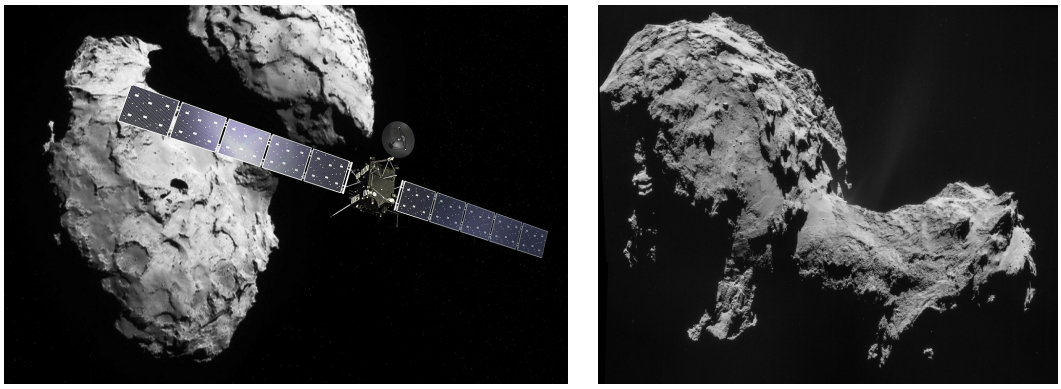
(a) *Elsa-d* **(b)** *Restore-L*

Figure 1.4: AR&D/C mission examples (courtesy of [12], [10]).

1.1.3 Scientific Explorations

One last possible motivation for performing autonomous rendezvous concerns scientific explorations. In recent years, given the explosive growth of the space sector in parallel with a more marked common interest, scientific missions for the exploration of celestial or extra-terrestrial bodies, carried out for a scientific purpose but no less ambitious or interesting, have also acquired greater importance.

The most glaring example is Rosetta, a ten-year expedition began in March 2004 to visit the Jupiter family comet 67P/Churyumov-Gerasimenko. After several flybys, in August 2014, Rosetta rendezvoused with the comet and started vision-based relative maneuvers to gradually approach to it, finally landing on its surface to acquire data.



(a) *Rosetta mission concept*

(b) *Comet 67P/Churyumov-Gerasimenko*

Figure 1.5: Rosetta mission (courtesy of [2], [9]).

1.2 Non-cooperative AR&D/C challenges

As already specified, autonomous RVD to an uncontrolled and possibly tumbling target has yet to be fully demonstrated in orbit, and so there is an evident lack of clear, widely accepted technical and safety standards – regarding maneuver planning, onboard sensing and control strategies - to perform RPO. According to Leomanni[20], there is a need for RVD techniques accounting for rotational motion of the target, which optimize meaningful performance indexes and are easy to implement onboard the chaser space-

craft. The main challenges to perform autonomous and noncooperative rendezvous are listed in the following:

- Necessity to conduct an inspection and rendezvous autonomously with 3D vision sensors (optical or LIDAR), infrared or others;

- Target non-cooperativeness: lack of sensor fiducial or capture interface to facilitate docking or grasping operations;

- If found, the docking point or the fixture to be grasped may move over time with the entire target, eliminating the possibility of using widely tested R-bar or V-bar approaches, traditionally employed by cooperative rendezvous;

- Specific trajectory constraints and relative navigation limitations make the servicer vehicle design to be single fault tolerant;

- Trajectory account for a variety of competing systems considerations such as missed/cold or hot burns, client lighting requirements and relative navigation sensor capabilities (for example sensor accuracy and maximum operating range).

Taking all these considerations into account, four complementary and consecutive operations, reasonably defined by Setterfield in its study of on-orbit inspection of a rotating object[37], or by Flores-Abad, in its review of space robotics technologies for OOS operations[13], can be found:

1. Observation and planning: the chase satellite maintains a safe distance from the target, acquiring information on it (through vision system or laser scan) and planning the next steps;

2. Approach of the chase inspector, synchronously with target attitude, to the berthing box, to dock or capture with a robotic arm;

3. Physical capture or dock of the target satellite;

4. Stabilization of captured target along with the servicing system through the application of external torques.

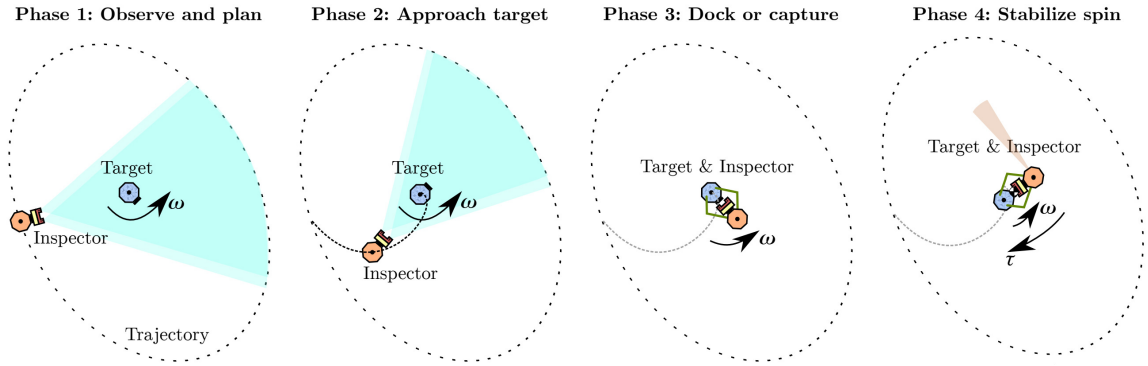
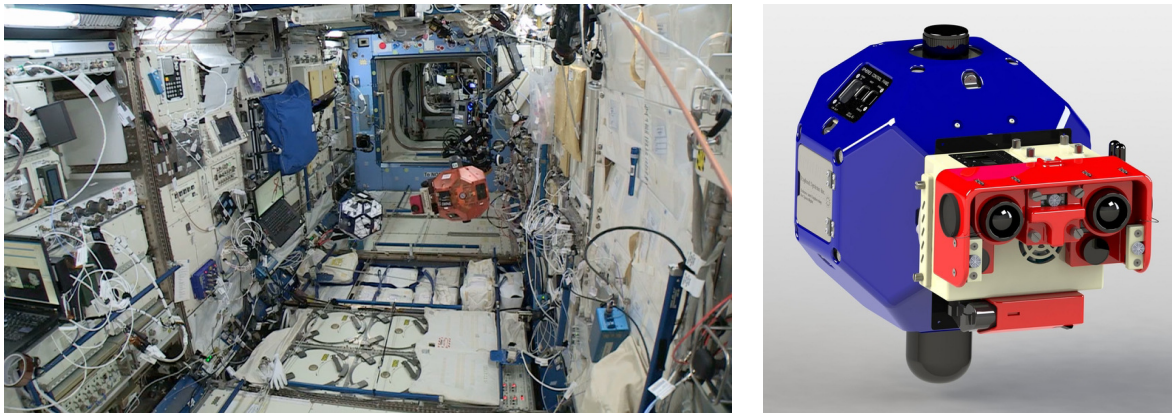


Figure 1.6: Overview of concept of non-cooperative proximity operations (courtesy of [37]).

As for the first phase, it is clear the need to have an estimate of the angular velocity as well as of the inertial properties of the target.

An interesting approach is provided by Setterfield in the work mentioned above, where an inspector equipped with a RADAR system or a 3D visual sensor, a star tracker and an inertial measurement unit is used to estimate the target's center trajectory, angular velocity, acceleration, and orientation in the inertial frame. The estimates of angular velocity and inertial properties are intimately connected: from an initial estimate of motion, through an algorithm that determines the polhode of the target object, it is possible to trace the orientation of the principal axes and a measurement of the main inertia ratios. This algorithm has been tested with the SPHERES-VERTIGO robotic platform, shown in Figure 1.7, on the International Space Station.

As regards the approach phase, many studies are being carried out. Due to the presence of many constraints to handle, such as the maximum action that can be delivered by actuators, velocity constraint or the collision avoidance constraint, the controller that best suits the problem is the Model Predictive Control (MPC). Several studies have been conducted on AR&D/C with a tumbling target through this controller. Dong[8] proposed a tube-based robust output feedback model predictive control technique; Li[21] simulated 4 scenarios demonstrating the MPC robustness; Zhou[45] proposed a novel control scheme based on motion planning and pose (or attitude) tracking, by adding a field of view constraint to ensure the target visibility; Weiss[41] and Park[32] showed, using an MPC, that is possible to plan trajectories avoiding eventual obstacles (debris) while approaching to the target. Park, in a further study[33] on this spacecraft



(a) *Test on ISS* **(b)** *SPHERES-VERTIGO unit*

Figure 1.7: SPHERES (Synchronized Position Hold Engage and Reorient Experimental Satellite) and VERTIGO (Visual Estimation and Relative Tracking for Inspection of Generic Objects) (courtesy of [29], [30]).

RVD problem with a rotating target platform, developed a nonlinear model predictive control (NMPC) approach. This NMPC strategy has been implemented on an air-bearing test bed to demonstrate the capability to perform real-time computation and to satisfy constraints at the same time.
The last two phases of capture and stabilization will not be developed in this thesis, although they still represent another very complex issue with regard to these missions, which motivated the development of new space robotics technologies and several experimental demonstration missions including both manned and unmanned missions[13].

1.3 Thesis outline

This thesis is organized as follows. Chapter 2 presents translational and rotational dynamic models suitable for a AR&D/C scenario. In Chapter 3 the principal Autonomous Rendezvous and Docking or Capture phases are discussed, from the delivering orbit to the capture phase, emphasizing the main problems and the choices made. Chapter 4 presents the background theory on model predictive controllers, linear and nonlinear, showing the reasons that led to their consideration. Alongside the controllers, Kalman filters - linear or Extended - of the navigation system will be briefly presented, showing

their close link with the control system. Chapter 5 presents the case study chosen for the simulations and the spacecraft simulator implemented in Matlab.
Chapters 6 and 7 are dedicated to results, the first concerning the near-field proximity operations, the second concerning the inner proximity operations. Finally, a conclusion is provided in Chapter 8.

Chapter 2

Spacecraft dynamic models

The case study involves a scenario in which a chase satellite follows another one (target): it is therefore useful to conduct and develop the problem considering the relative motion between the two bodies.
The following theory was taken from Curtis[6]. Refer to the mentioned book for a more careful and exhaustive analysis.

2.1 Hill Clohessy-Wiltshire translational model

As illustrated in the following, by linearizing the equations of relative motion through more than reasonable mathematical simplifications, it is possible to write them in such a way that they can be solved analytically without making considerable errors.

Let A and B be the target and the chaser respectively and consider the Earth-Centered Inertial frame. Consider then a second reference frame, this time non-inertial, centered in A and with the x-axis pointed in the radial direction, the y-axis in the direction of motion of the target and the z-axis perpendicular to the target orbital plane, to close the triad.
Referring to Figure 2.1, let's denote with $\mathbf{R}$ and $\mathbf{r}$ the inertial positions of bodies A and B respectively, and let $\delta\mathbf{r}$ be the position vector of the chase vehicle relative to the target, such that $\mathbf{r} = \mathbf{R} + \delta\mathbf{r}$. By substituting the last equation in the orbital equation of the target:

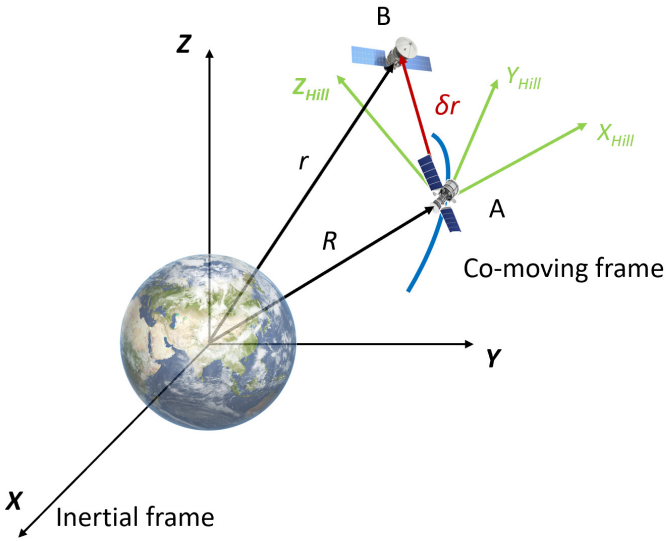


Figure 2.1: Relative motion: Inertial and Hill frame (green).

$$\ddot{\mathbf{r}} = -\mu \frac{\mathbf{r}}{r^3}$$

and assuming $\frac{\delta r}{r} << 1$, given the proximity between the two bodies, through mathematical steps it is possible to get a well-condensed equation:

$$\delta\ddot{\mathbf{r}} = -\frac{\mu}{R^3}\left[\delta\mathbf{r} - \frac{3}{R^2}\left(\mathbf{R}\cdot\delta\mathbf{r}\right)\mathbf{R}\right]$$

still expressed, however, in the inertial frame.

This expression, linearized in terms of $\delta\mathbf{r}$, can be written in the Hill reference defined above, considering that:

$$\delta\mathbf{r} = \delta x\hat{\mathbf{i}} + \delta y\hat{\mathbf{j}} + \delta z\hat{\mathbf{k}}$$

where $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$ and $\hat{\mathbf{k}}$ are the Hill versors.

By developing the term $\delta\ddot{\mathbf{r}}$ using the acceleration decomposition formula, one can write:

$$\delta\ddot{\mathbf{r}} = \dot{\boldsymbol{\Omega}} \times \delta\mathbf{r} + \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \delta\mathbf{r}) + 2\boldsymbol{\Omega} \times \delta\mathbf{v}_{\mathbf{rel}} + \delta\mathbf{a}_{\mathbf{rel}},$$

with $\delta\mathbf{a}_{\mathbf{rel}}$ relative acceleration measured in the comoving frame and $\boldsymbol{\Omega} = n\hat{\mathbf{k}}$, with $n = \sqrt{\frac{\mu}{R^3}}$ the mean orbit motion. By combining the expressions written above, it is possible to get the three scalar equations below:

$$\begin{cases} \delta\ddot{x} - \left(\frac{2\mu}{R^3} + \frac{h^2}{R^4}\right)\delta x + \frac{2(\mathbf{V}\cdot\mathbf{R})h}{R^4}\delta y - 2\frac{h}{R^2}\delta\dot{y} = 0 \\ \delta\ddot{y} + \left(\frac{\mu}{R^3} - \frac{h^2}{R^4}\right)\delta y - \frac{2(\mathbf{V}\cdot\mathbf{R})h}{R^4}\delta x + 2\frac{h}{R^2}\delta\dot{x} = 0 \\ \delta\ddot{z} + \frac{\mu}{R^3}\delta z = 0 \end{cases} \tag{2.1}$$

where h represents the orbital angular momentum of A.

These are linear second-order equations and they have to be solved to obtain the relative position coordinates. It is possible to notice how the first two expressions are coupled with each other since δx and δy appear in each one of them. Instead, δz appears alone, and this means that the out-of-plane component is independent.

If the target's orbit is circular or it has low ellipticity, the product $\mathbf{V}\cdot\mathbf{R}$ is quite zero and $h = \sqrt{\mu R}$, and Eqn 2.1 can be re-written in the following way:

$$\begin{cases} \delta\ddot{x} - \frac{3\mu}{R^3}\delta x - 2\sqrt{\frac{\mu}{R^3}}\delta\dot{y} = 0 \\ \delta\ddot{y} + 2\sqrt{\frac{\mu}{R^3}}\delta\dot{x} = 0 \\ \delta\ddot{z} + \frac{\mu}{R^3}\delta z = 0 \end{cases} \tag{2.2}$$

These second order, linear and time invariant differential equations are called Hill Clohessy-Wiltshire equations (HCW), here expressed in a homogeneous form.

This model is sufficiently accurate to describe the relative motion between the two bodies in a few-kilometer range, or better, when the distance between the chaser and the target is small compared with the radius of the target's orbit: that is the case of proximity maneuvers.

Unlike the set of equations 2.1, where the target orbit can be an ellipse, the HCW ones

have constant coefficient so a straightforward analytical solution exists. This solution can be found through a standard Laplace transformation or by combining homogeneous and particular components:

$$\begin{cases} x(t) = \left(4x_0 + 2\dfrac{v_0}{n}\right) + \dfrac{u_0}{n}\sin(nt) - \left(3x_0 + 2\dfrac{v_0}{n}\right)\cos(nt) \\ y(t) = \left(y_0 - 2\dfrac{u_0}{n}\right) + \left(6x_0 + 4\dfrac{v_0}{n}\right)\sin(nt) - 2\dfrac{u_0}{n}\cos(nt) - (6nx_0 + 3v_0)t \\ z(t) = z_0\cos(nt) + \dfrac{w_0}{n}\sin(nt) \\ u(t) = u_0\cos(nt) + (3nx_0 + 2v_0)\sin(nt) \\ v(t) = (6nx_0 + 4v_0)\cos(nt) - 2u_0\sin(nt) - (6nt + 3v_0) \\ w(t) = -z_0 n\sin(nt) + w_0\cos(nt) \end{cases} \tag{2.3}$$

where n is the mean motion afore mentioned and $[u, v, w]$ is the relative velocity vector, so $[u, v, w] = [\dot{x}, \dot{y}, \dot{z}]$. The subscripted zero indicates an initial condition.

All three components of the position vector oscillate with a frequency equal to the mean motion. Only $y(t)$ has a secular term, $-(6nx_0 + 3v_0)t$, that grows with time: if an average relative motion between chaser and target is not desired, it is appropriate to set the condition $v_0 = -2nx_0$, thus compensating for the secular effect.

In addition, choosing the initial position and velocity properly, it is possible to obtain a relative motion, suitable for a natural observation, as explained in the next paragraph.

Equations 2.3 can be organized in matrix form:

$$\begin{bmatrix} \mathbf{r}(t) \\ \mathbf{v}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi_{rr}}(t) & \mathbf{\Phi_{rv}}(t) \\ \mathbf{\Phi_{vr}}(t) & \mathbf{\Phi_{vv}}(t) \end{bmatrix} \begin{bmatrix} \mathbf{r_0}(t) \\ \mathbf{v_0}(t) \end{bmatrix} \tag{2.4}$$

with $\mathbf{\Phi_{ij}}(t)$ illustrated below:

$$\mathbf{\Phi_{rr}}(t) = \begin{bmatrix} 4-3\cos nt & 0 & 0 \\ 6(\sin nt - nt) & 1 & 0 \\ 0 & 0 & \cos nt \end{bmatrix} \quad \mathbf{\Phi_{rv}}(t) = \begin{bmatrix} \frac{\sin nt}{n} & \frac{2(1-\cos nt)}{n} & 0 \\ \frac{2(\cos nt-1)}{n} & \frac{4\sin nt-3nt}{n} & 0 \\ 0 & 0 & \frac{\sin nt}{n} \end{bmatrix}$$

$$\mathbf{\Phi_{vr}}(t) = \begin{bmatrix} 3n\sin nt & 0 & 0 \\ 6n(\cos nt - 1) & 0 & 0 \\ 0 & 0 & -n\sin nt \end{bmatrix} \quad \mathbf{\Phi_{vv}}(t) = \begin{bmatrix} \cos nt & 2\sin nt & 0 \\ -2\sin nt & 4\cos nt - 3 & 0 \\ 0 & 0 & \cos nt \end{bmatrix}$$

To facilitate the future controller design, the Hill's differential equations 2.2 can be also rewritten in terms of a state space form as follows:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) \tag{2.5}$$

where $x = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T$ is the state vector and A is the state matrix given by:

$$\mathbf{A_{Hill}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 0 & 2n & 0 \\ 0 & 0 & 0 & -2n & 0 & 0 \\ 0 & 0 & -n^2 & 0 & 0 & 0 \end{bmatrix} \tag{2.6}$$

It is possible to combine the HCW model in state space form seen in Eqn 2.5 with the 6×6 matrix in Eqn 2.4:

$$x(t) = \mathbf{\Phi}\left(t - t_0\right)\mathbf{x}(t_0) \tag{2.7}$$

where $\mathbf{\Phi}$ is the *State Transition Matrix* (*STM*) just mentioned:

$$\mathbf{\Phi}(\Delta t) = e^{(\mathbf{A}\Delta t)} = \begin{bmatrix} \mathbf{\Phi_{rr}}(\Delta t) & \mathbf{\Phi_{rv}}(\Delta t) \\ \mathbf{\Phi_{vr}}(\Delta t) & \mathbf{\Phi_{vv}}(\Delta t) \end{bmatrix} \tag{2.8}$$

where $\Delta t = t - t_0$.

2.1.1 LVLH frame

In addition to the Hill frame, another non-inertial and target-centered reference system will be used in this thesis. This frame, showed in Figure 2.2, will be named as "LVLH frame", an acronym that stands for "Local Vertical - Local Horizontal".
Its x-axis is pointed along the direction of motion, the z-axis is pointed to Nadir and the y-axis is perpendicular to the orbital plane.

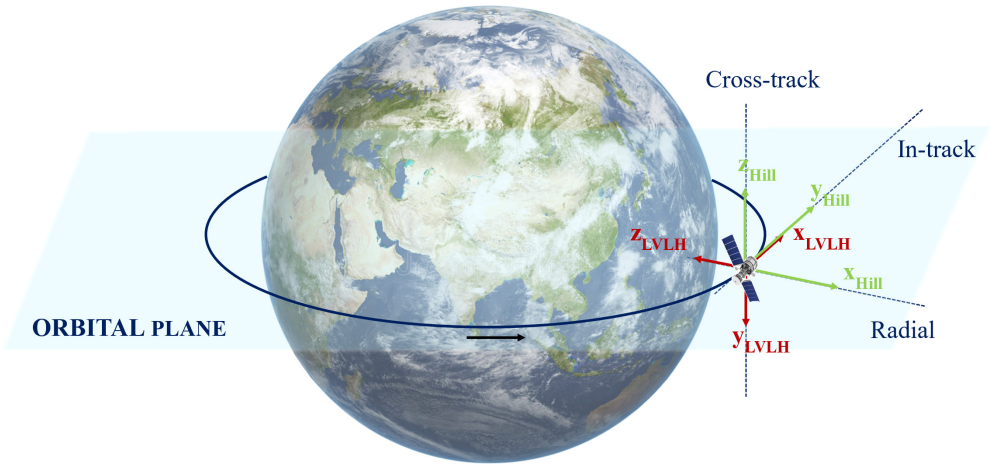


Figure 2.2: Hill's and LVLH orbital frames.

Between this reference and Hill's reference afore presented there is a simple and constant rotation, which involves just a formal change of the HCW equations and an adjustment of the state space matrix:

$$\mathbf{A_{LVLH}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 2n \\ 0 & -n^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3n^2 & -2n & 0 & 0 \end{bmatrix} \tag{2.9}$$

In the following discussion, to avoid confusion, the directions of the axes of orbiting

systems will often be explicitly specified with the names 'Radial', 'In-track' and 'Cross-track', referring to the generic orbiting system with the 'RIC frame' name.

2.2 Spacecraft attitude dynamics

The attitude is the orientation of a reference system, co-moving with the body-satellite, with respect to another system, chosen as a reference to evaluate the orientation of the satellite itself. To describe the attitude with a dynamic model, consider the definition of angular momentum $\mathbf{H}_O$ of a generic body of mass m, with reference to a generic point O, as shown in Figure 2.3:

$$\mathbf{H}_O = \int_{\mathcal{C}} OP \times \mathbf{v}_P \, dm \tag{2.10}$$

where $\mathbf{v}_P$ is the translational velocity of point P.

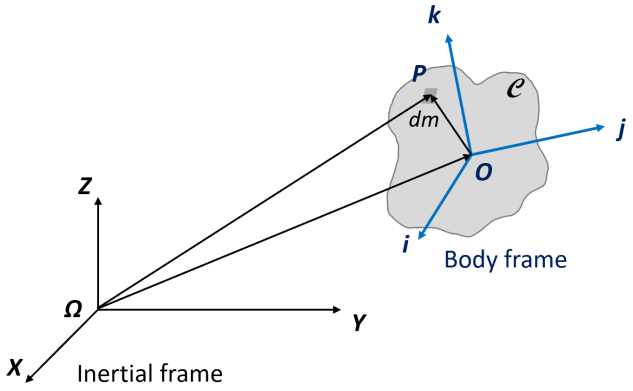


Figure 2.3: General Body frame.

After some rearrangement, the second equation of dynamics can be written in the following form:

$$\frac{d\mathbf{H}_O}{dt} = -\mathbf{v}_O \times m\mathbf{v}_G + \mathbf{T}_O^{ext}$$

where $\mathbf{v}_O$ and $\mathbf{v}_G$ are velocities in the inertial frame of O and G (that is the center

of mass) respectively and $\mathbf{T}_O^{ext}$ represents the sum of the external torques acting on the considered body. By imposing the coincidence between O and C, the expression is simplified into the following:

$$\frac{d\mathbf{H}_O}{dt} = \mathbf{T}_O^{ext}$$

written in the inertial reference. This equation can be written in the body reference as follows:

$$\dot{\mathbf{H}} + \boldsymbol{\omega} \times \mathbf{H} = \mathbf{T}^{ext} \tag{2.11}$$

where all terms are expressed in the body frame and the additional term $\boldsymbol{\omega} \times \mathbf{H}$ represents the contribution due to the non-inertiality of the body system. Being able to write the angular momentum in the body reference as $\mathbf{H} = \mathcal{I}\boldsymbol{\omega}$, where $\mathcal{I}$ is the inertial body tensor, the expression 2.11 takes the following form:

$$\mathcal{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathcal{I}\boldsymbol{\omega} = \mathbf{T}^{ext} \tag{2.12}$$

that, if developed, represents the Euler equations set, which describes the attitude dynamics.

Returning to the initial definition, to describe the attitude one can use angles or a rotation matrix between the body reference and a given system of reference. These methods, however, keep two weaknesses: both require the use of trigonometric functions, which result in non-linearity for the model and therefore complications in the algorithms; moreover, there can be singularities that appear through precise combinations that annul the determinant of the rotation matrices. Usually, therefore, a third alternative is preferred, which coincides with the use of quaternions.

These three alternatives will be briefly shown in the following.

Euler angles and Rotation matrices

Euler angles define 3 successive rotations around the current axes resulting from them. The order in which the rotations are carried out is arbitrary and can be chosen according to the particular situation to be analysed.

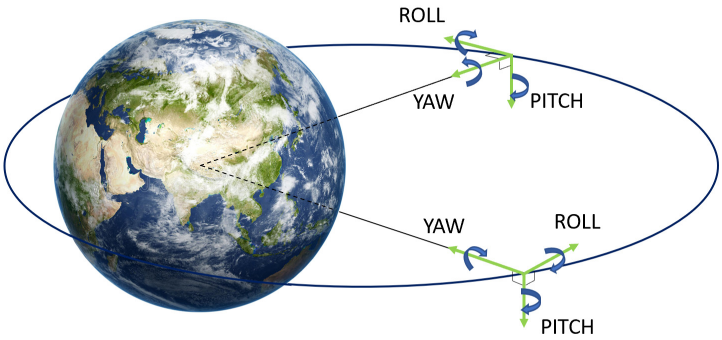


Figure 2.4: RPY orbital Frame.

Assuming to start from an orbiting tern (for example the RPY orbital reference represented in Figure 2.4), through these three rotations it is possible to align this tern with the desired body reference. One sequence to do this, typical of three-axis stabilized satellites, is the 3-2-1 one, which indicates a succession of three rotations:

- Rotation of Ψ around the current third axis;
- Rotation of Θ around the current second axis;
- Rotation of Φ around the current first axis,

where Ψ, Θ and Φ are the Euler angles.
Knowing these three angles means knowing the attitude and allows, through the subsequent multiplication of 3 rotation matrices, to identify the rotation matrix to pass from one reference to another:

$$R_O^B = R_\Phi \; R_\Theta \; R_\Psi \tag{2.13}$$

where O stands for Orbital and B for Body.
The angular velocity of the body system with respect to the orbiting system, following the sequence 3-2-1, can be written in the body reference as follows:

$$\boldsymbol{\omega}_{BO} = \begin{bmatrix} \dot{\Phi} - \sin\theta\dot{\Psi} \\ \dot{\Theta}\cos\Phi + \dot{\Psi}\sin\Phi\cos\Theta \\ -\dot{\Theta}\sin\Phi + \dot{\Psi}\cos\Phi\cos\Theta \end{bmatrix} \tag{2.14}$$

This angular velocity, however, cannot be used in Euler's equations, being referred to an orbiting system and therefore not inertial.
By considering the relative velocity theorem, it is possible to write, in the body reference:

$$\begin{aligned} \boldsymbol{\omega}_{BI} &= \boldsymbol{\omega}_{BO} + \boldsymbol{\omega}_{OI} \\ &= \begin{bmatrix} \dot{\Phi} - \sin\theta\dot{\Psi} \\ \dot{\Theta}\cos\Phi + \dot{\Psi}\sin\Phi\cos\Theta \\ -\dot{\Theta}\sin\Phi + \dot{\Psi}\cos\Phi\cos\Theta \end{bmatrix} + R_O^B \begin{bmatrix} 0 \\ -n \\ 0 \end{bmatrix} \end{aligned} \tag{2.15}$$

where $\boldsymbol{\omega}_{BI}$ is the angular velocity between body and inertial, to be used in Eqn 2.12, $\boldsymbol{\omega}_{BO}$ is the one given above and $\boldsymbol{\omega}_{OI}$ is the inertial angular velocity of the orbiting reference.
Thinking of a circular orbit, this last velocity can be easily written in the RPY reference shown above as follows:

$$\boldsymbol{\omega}_{OI}^{(O)} = \begin{bmatrix} 0 \\ -n \\ 0 \end{bmatrix} \tag{2.16}$$

and, multiplying it by the rotation matrix R_O^B in Eqn 2.13, it can be written in the body reference.
By integrating the equations of dynamics and kinematics 2.12 and 2.15, it is possible to pass from accelerations and angular velocities to Euler angles. However, as written in the previous paragraph, this method can lead to problems, having to invert matrices.

Quaternions or Euler's parameters

A quaternion is a 4 parameters vector that identify the instantaneous direction of the axis of rotation and the rotation itself around that axis. It is always possible to consider any rigid motion as a single rotation around the axis of instantaneous rotation.

To do this it is sufficient to provide the azimuth angle A, the elevation angle E and

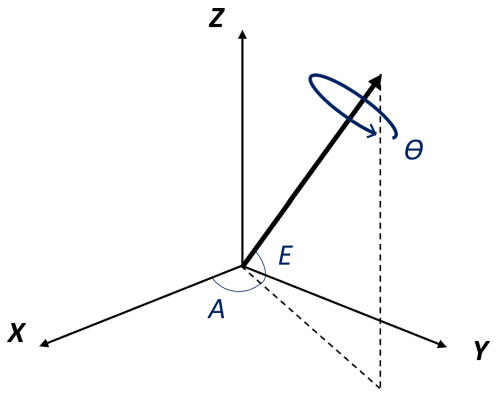


Figure 2.5: Quaternion components.

the quantity θ to determine the rotation around the axis: the fourth parameter, to complete the quadruplet, introduces the algebraic condition of normalization.
It can be demonstrated that there is a bijective correspondence between the rotation matrix from body to orbital and the quaternion: the integration of the equations of kinematics, therefore, can exploit this parallelism to turn into an integration of the following term:

$$\dot{\mathbf{q}} = \frac{1}{2}\left[\mathcal{Q}\right]\mathbf{q} = \frac{1}{2}\begin{bmatrix} 0 & -\omega_{BO,x} & -\omega_{BO,y} & -\omega_{BO,z} \\ \omega_{BO,x} & 0 & \omega_{BO,z} & -\omega_{BO,y} \\ \omega_{BO,y} & -\omega_{BO,z} & 0 & \omega_{BO,x} \\ \omega_{BO,z} & \omega_{BO,y} & -\omega_{BO,x} & 0 \end{bmatrix}\mathbf{q} \tag{2.17}$$

Once the integration has been carried out, it is possible to go back through simple steps, not shown for brevity, to Euler angles, for a more immediate comprehension.

Chapter 3

Rendezvous principal phases

Always remaining within the Autonomous Rendezvous scenario, the purpose of the mission studied is on-orbit servicing: a servicer chase spacecraft, equipped with robotic arms and an $AR\&C$ sensor suite, is launched and maneuvered into a target spacecraft co-elliptical orbit. The chaser (or servicer, due to its aim) then performs RPO with the target, culminating in the initiation of a natural o forced motion around it. After inspecting the target (for hours or possibly for several days), the servicer begins moving down a capture axis and then, co-moving with the tumbling target, reaches a specific target body-fixed box next to the target, until it is able to grapple it using robotic arms. After these latter are rigidly connected to the target structure, the chase servicer carry out its on-orbit servicing operations. When the mission is accomplished, the servicer vehicle releases the target and moves away safely, towards other clients to assist, until it has only enough fuel left to safely dispose of itself.

Concerning autonomous rendezvous and proximity operations, the main issues involve maneuver planning, onboard sensing and of course the actuation of control actions calculated by the GNC system. In order to perform a more detailed analysis, it is useful to look for a criterion of spatial and temporal subdivision of the mission in its entirety.

Keep-out volumes

To facilitate the safe design of the rendezvous trajectory and for the purposes of mission analysis, it is useful to divide the target vicinity region into three nested keep-out

volumes of gradually decreasing dimensions, as shown in Figure 3.1. The subdivision is based on the navigation uncertainty and therefore on the requirements of position and attitude, initially bland but gradually more critical getting closer to the target. Tracking data quality and on-board filtering algorithms directly affect orientation and size of the covariance ellipsoid of the relative position error and thus collision probability estimation.

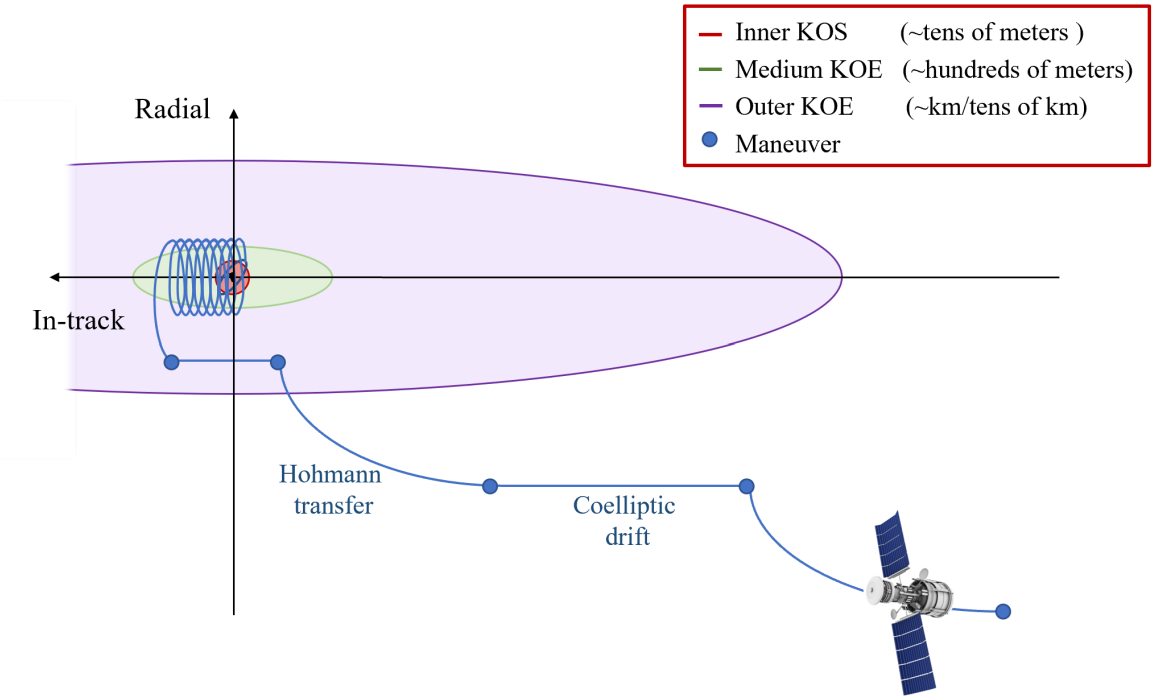


Figure 3.1: Keep-out ellipsoids concept.

Typically, during rendezvous with cooperative satellites, for safety purposes, the servicer spacecraft must stay outside of the keep-out volume of consideration until the ground provides an Authority To Proceed (ATP) command.

The instrumentation used to obtain the necessary measures cannot remain the same in all the rendezvous phases, but must change and becomes more precise as the chaser approaches the target. The sensors used must have measuring ranges and field of view compatible with the functions they must perform from time to time. The main R&D/C sensors accuracies are represented in Figure 3.2.

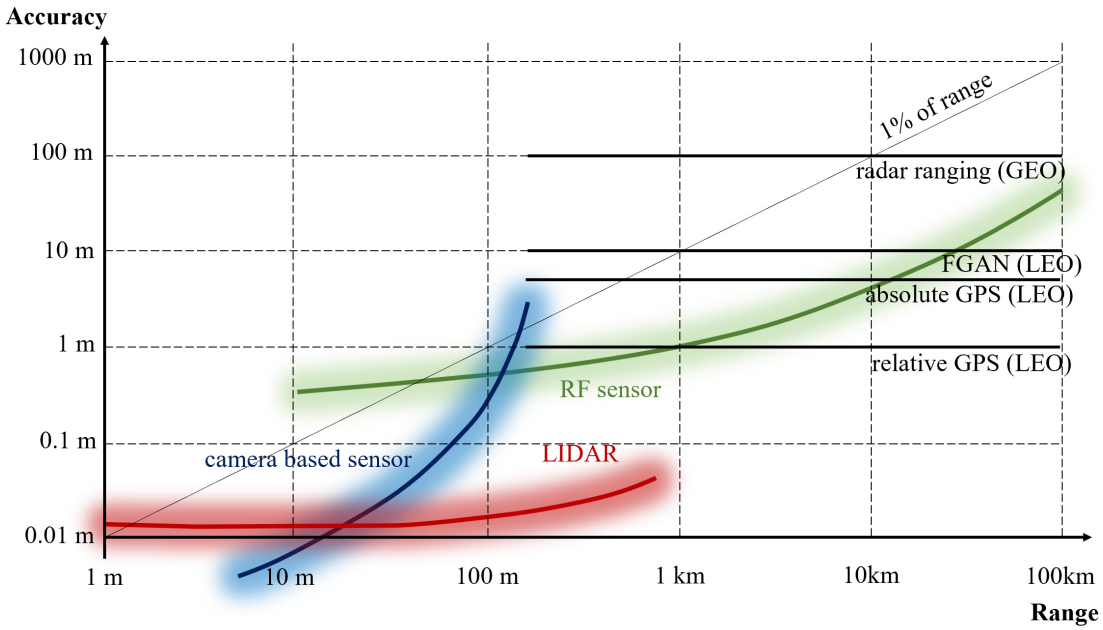


Figure 3.2: Main sensors to perform a rendezvous, with accuracies (based on [43]).

A distinction must be made between absolute sensors, that provide position and attitude with respect to an external absolute reference, and relative sensors, that give the position/attitude with respect to the target.
In the following, the three phases of a rendezvous are presented:

1. **Far-field rendezvous**

   These are the operations performed outside and straddling the outermost keep-out ellipsoid, whose indicative dimensions are shown in Figure 3.1. The sensors used by the chaser to know its position are the absolute ones, so mainly GPS, with no direct relative state measurements yet: given the distances between the two bodies, no struggling accuracies are required, and an uncertainty of tens of meters is considered acceptable. The position of the target, not being cooperative and therefore not being able to transmit information to the chaser, is estimated by the ground RADAR system. Considering passive safety as first driver, the far-field trajectory, being tens of kilometers behind, is designed to maintain a safe radial buffer from the target.

2. **Near-field rendezvous**

   It includes those operations around a hundred meters straddling the second innermost ellipsoid around the target: relative position sensors such as narrow field-of-view (NFOV) visible wavelength camera, visible wavelength stereo cameras, long-wave infrared camera (IRCam) or Light Detection and Ranging (LIDAR) sensor come into play.

3. **Inner Proximity Operations**

   These are those operations performed within the innermost keep-out sphere, where it is not possible to enter in unless certain requirements are met. The sensors used are obviously those of relative navigation, which during the approach provide increasingly accurate information on the status of the target. In this phase, more than others, six degrees of freedom must be guaranteed for the motion (translation and rotation), to comply with the strictest constraints and for an obvious safety requirement.

3.1 Far-field RPO

As done by Barbee[3] for geosynchronous orbit, it is assumed that the servicer spacecraft has already been placed on a co-planar and co-elliptical orbit with the target spacecraft's orbit, regardless of whether directly through the launch vehicle upper stage or rather through maneuvers planned by ground operators following launch.

The condition of co-ellipticity means, as one can intuit, that the orbits have the same eccentricity and aligned apse lines. The relative motion (drift, fly-around or both combined) between the two bodies is generated by the difference in semimajor axis between the orbits. The chaser can generally be above or below the target along its radial axis, but the latter may be considered a safer option to begin $AR\&C$ operations, as it allows to gradually reduce the relative distance and because the chaser does not have to cross the target's orbit prior to final rendezvous and proximity operations. Having a shorter semimajor axis and then a shorter period, the servicer spacecraft will be moving faster than the target when below it, so in-track distance can be easily

cancelled.

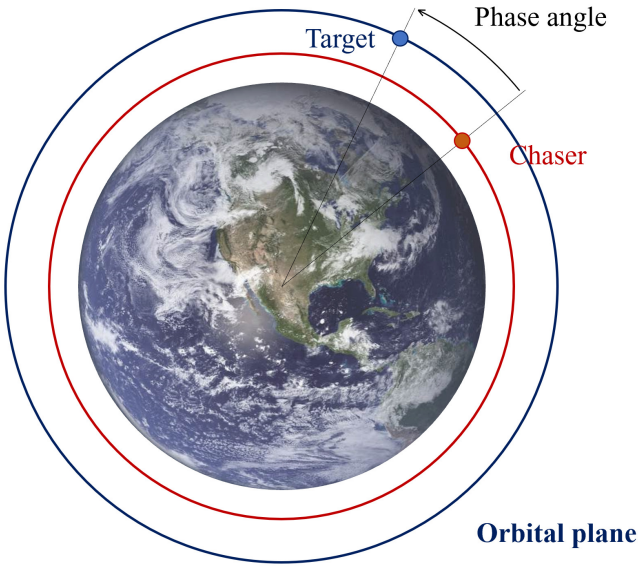


Figure 3.3: Co-elliptic motion.

Generally, the servicer begins its maneuvers tens or even a few hundred kilometers behind the target along the in-track axis (y-axis in Hill's frame) and a few dozen kilometers below it: these differences in position will be slowly cancelled through Hohmann transfers and co-elliptic drifts (Figure 3.3).
According to Barbee[3], this multiple-step approach is advantageous for many reasons.

1. There are several parameters that can be adjusted to find the optimal solution: by handling the choice of initial in-track and radial separations or the choice of radial raises covered by Hohmann transfers, the duration and strategy of the rendezvous can be controlled within the limits of orbital mechanics. For instance, drift time between altitude raise maneuvers can be controlled and adjusted for ensuring navigation filter convergence and accuracy.

2. This approach of consecutive drifts and altitude raises ensures passive safety to the relative motion as regards collision avoidance: if the servicer, due to any malfunctioning, loses the ability to maneuver at any time, or chooses to not carry out the AR&D/C after situational awareness sensors reveal an unsafe condition in the target proximity, the chaser will merely and safely continue to drift below the

target, purely passing it by, without colliding or breaching predefined keep-out volumes.

3. The Hohmann transfer is the most fuel efficient two impulse maneuver for transferring between two coplanar and co-elliptic orbits of different altitudes. In any case, assuming that a budget delta-v is given, altitude raises can be performed by other types of transfers, depending on mission requirements.

3.2 Near-field RPO

After the co-elliptical far-field approach, the main issue becomes the choice of the relative motion of observation to obtain information on the state of motion and inertial properties of the target, if they are unknown. This latter is the most difficult challenge. While approaching the target, relative sensors, such as LIDAR, optical sensors and infrared, come into play: despite the absolute sensors phase, it's mandatory that the inspector continually performs attitude maneuvers that keep the inspection sensor pointing inward and directly at the target while the circumnavigation is performed.
It is possible to group relative motions into two broad categories: forced motions and natural motions.

3.2.1 Natural motions

Unlike forced motions, natural motions exploit the geometry of orbits to circumnavigate the target satellite.
To give an analytical description, consider the HCW equations in Eqn 2.3 once more. With the purpose of describing natural relative motions, it is useful to bring the set of equations back to a more intuitive one, which highlights the main parameters it is possible to handle:

$$\begin{cases} x(t) = x_{max}\sin(nt+\psi_0) - \dfrac{2\dot{y}_c}{3n} \\ y(t) = 2x_{max}\cos(nt+\psi_0) + \dot{y}_c\dfrac{(nt+\psi_0-\pi/2)}{n} + \Delta y \\ z(t) = z_{max}\sin(nt+\phi_0) \\ \dot{x}(t) = nx_{max}\cos(nt+\psi_0) \\ \dot{y}(t) = -2nx_{max}\sin(nt+\psi_0) + \dot{y}_c \\ \dot{z}(t) = nz_{max}\cos(nt+\phi_0) \end{cases} \tag{3.1}$$

where the parameters are defined through the relative orbit initial conditions as follows:

$$\begin{aligned} x_{max} &= \sqrt{x_0^2 + \frac{\dot{x_0}^2}{n^2}} && \Rightarrow \text{ maximum radial offset} \\ \Delta y &= y_0 - 2\frac{\dot{x_0}}{n} && \Rightarrow \text{ initial in-track separation} \\ z_{max} &= \sqrt{z_0^2 + \frac{\dot{z_0}^2}{n^2}} && \Rightarrow \text{ maximum cross-track offset} \\ \psi_0 &= \arctan(\frac{nx_0}{\dot{x_0}}) && \Rightarrow \text{ initial in-plane phase} \\ \phi_0 &= \arctan(\frac{nz_0}{\dot{z_0}}) && \Rightarrow \text{ initial cross-track phase} \end{aligned}$$

The two main configurations that can be obtained by varying these parameters are presented below.

The "safety" concept can be realized choosing the correct initial parameters such that the trajectory never crosses the velocity vector of the target and such that the chaser never enters the inner keep-out volume, even flying around it.

Flying around Safety Ellipse

As already noted in Section 2.1, if one desires to have a relative fly around motion of the chaser without any drift from the target, the secular terms must be cancelled through the condition $v_0 = -2nx_0$. By imposing this condition, it can demonstrated how the two orbits of the client and the servicer have the same semi-major axis and therefore the same period. In the case of the equations seen above, this condition simply becomes $\dot{y}_c = 0$, which leads to the following simplification:

$$\begin{cases} x(t) = x_{max}\sin(nt+\psi_0) \\ y(t) = 2x_{max}\cos(nt+\psi_0) + \Delta y \\ z(t) = z_{max}\sin(nt+\phi_0) \\ \dot{x}(t) = nx_{max}\cos(nt+\psi_0) \\ \dot{y}(t) = -2nx_{max}\sin(nt+\psi_0) \\ \dot{z}(t) = nz_{max}\cos(nt+\phi_0) \end{cases} \tag{3.2}$$

It's important to evidence the possibility for the center of the safety ellipse to be offset from the target's center of mass, along the in-track direction, by an amount Δy.

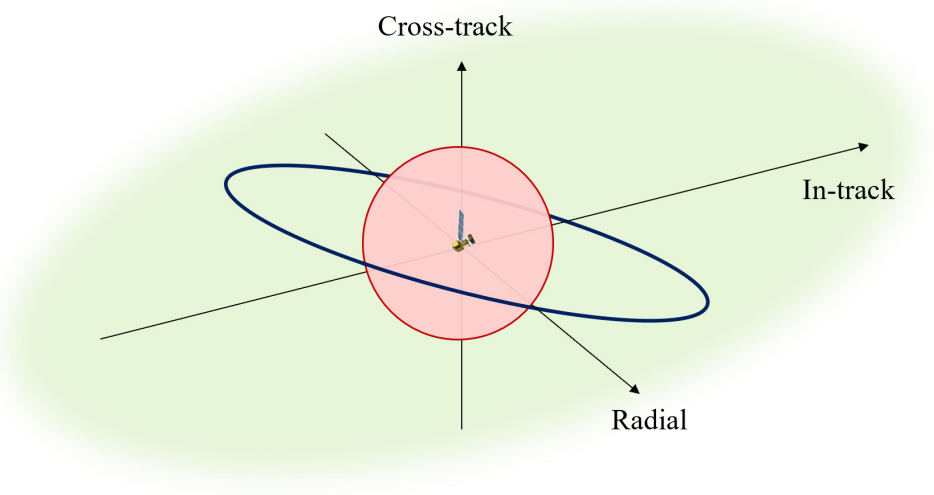


Figure 3.4: Flying around Safety Ellipse example.

Walking Safety Ellipse

A Walking Safety Ellipse (WSE) is obtained when $\dot{y_c} \neq 0$, yielding a relative motion that spirals around the x-axis (in-track) towards or away from the target depending on the signs of the initial in-track coordinate and $\dot{y_c}$. Because of this additional term, it can be noticed how the WSEs are not exactly centered on the x-axis because they have a radial bias proportional to $\dot{y_c}$.

It is useful to say something more about the phases of motion: indeed it is possible to

find a link between the two phases ψ_0 and ϕ_0. Let $\chi = nt + \psi_0$ be the safety ellipse plane motion polar angle, and let $\gamma = nt + \phi_0$ be the cross-track phase space angle: it can be demonstrated that:

$$\xi = \chi - \gamma = \frac{\pi}{2} \tag{3.3}$$

that can be seen as a condition on the initial difference between the in-plane and the out-of-plane phase angles: this is the core safety ellipse criterion.

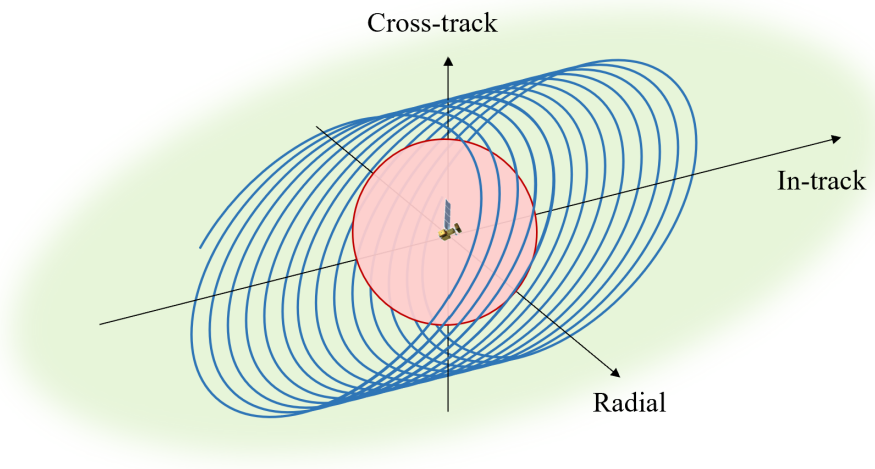


Figure 3.5: Walking Safety Ellipse example.

Note that the tubular manifold around the in-track axis can be exploited to safely approach to the target but also to inspect it through the circumnavigation.

3.2.2 Forced motion

These types of motions imply the need to recursively perform maneuvers and use propellant, since the difference in shape of the orbits does not provide for these scenarios of relative motion. Some examples are provided below.

Motions along V-bar and R-bar

These approaches are typical of cooperative rendezvous, so it is not worth dwelling too much on these strategies. A striking example for these types of motion are the rendezvous with the International Space Station, which uncompromisingly requires the

fulfillment of precise requirements to perform docking, thus dividing the rendezvous into several phases interconnected by authorizations to proceed subsequently granted by the station and ground operators.

Moving along the two main in-plane directions (R-bar or V-bar in Figure 3.6), outside the safety ellipsoid, typically requires two-pulse radial or tangential maneuvers: these maneuvers are designed to decrease the relative distance along the respective directions by taking advantage as much as possible of the natural relative motion between the two satellites.

A quite different matter, instead, is the so-called closing approach, or rather the approach within the safety ellipsoid (or sphere): in this case, to follow an almost straight trajectory (not fully, due to thrusters controlled in a Pulse Width Modulation mode), there is the need of multiple delta-vs. These delta-vs are delivered to compensate for Coriolis accelerations or any gravity gradients dictated by HCW equations in the presence of non-natural motion.

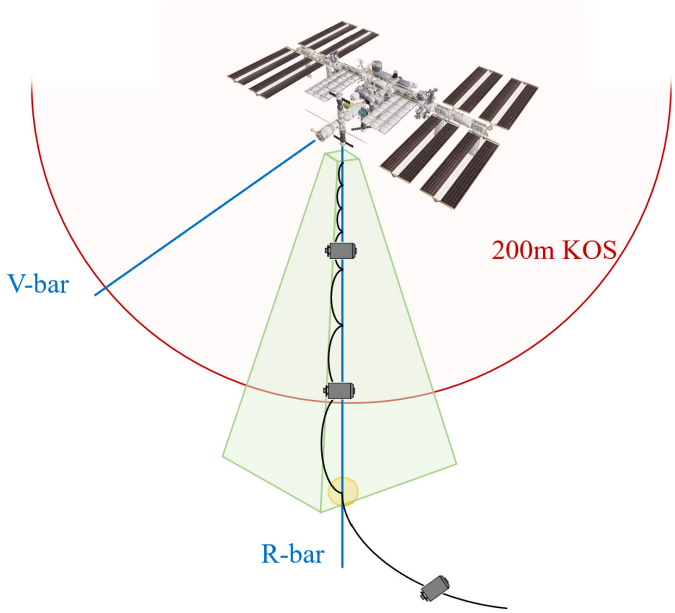


Figure 3.6: Rendezvous with the ISS: R-bar and V-bar approaches.

As shown in Figure 3.6, the docking ports of the ISS are located in the central part and facing Earth (radial) or along the flight direction of the station, behind it: the choice and execution of one or the other strategy will therefore depend on the port

chosen for the approach.

Waypoints on a octahedron

An interesting forced motion scheme can be the David E Gaylor's one[15]: the idea is to circumnavigate the target and perform inspection by travelling to six waypoints that offer views of the primary from all six faces of a virtual cube centered on the target, outside the inner keep-out sphere. A brief explanation on the latter must be provided, given that the observation structure of the case study considered in this thesis is built on this.

To remember, being an uncooperative rendezvous, the position of the target spacecraft's center of mass is unknown and must be estimated with sensors. Uncertainties on relative position are inevitable, but can be managed by representing them by three-dimensional Gaussian probability densities, linked to the covariance matrices. This covariance information can be used to build the error ellipsoid, and this ellipsoid constitutes the keep-out region: if the chaser remains outside this ellipsoid, the probability of collision is less than or equal to a certain value. For the sake of simplicity, this ellipsoid is considered as spherical: that is the inner keep-out sphere.

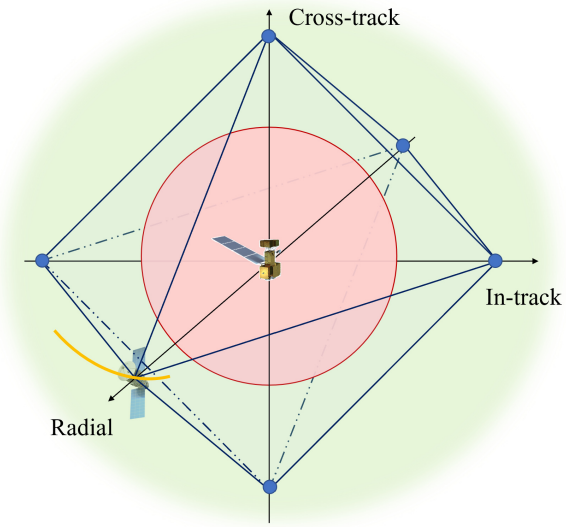


Figure 3.7: Waypoints on a octahedron.

The six points, due to an optimal arrangement consideration, can be the vertices of an octahedron or, as already stated, the centers of the six faces of a cube. The distance of

these vertices from the center is fixed and depends, as explained above, on the safety requirements adopted. For the sake of simplicity, the waypoints can all lie on the radial, in-track and cross-track orbital frame axes, as shown in Figure 3.7.

The design of the circumnavigation trajectory segments includes considerations of safety, time occurred to move from one waypoint to the next one and obviously total delta-v required to complete the circumnavigation. The last two requirements are variable and can be fixed in the design, while the first, on the other hand, is a hard constraint. Another variable parameter can be the travel sequence of the six waypoints, opting for the best choice with similar considerations on the total delta-v but also taking into account the orientation of the error ellipsoid.
The total delta-v can simply be calculated as the sum of all the delta-vs needed at each waypoint:

$$\Delta v_{tot} = \sum_{j=1}^{N} \Delta v_j \qquad \text{with j from 1 to N=6} \tag{3.4}$$

It is emphasized that the motion, after corrections, can be forced only near the observation points, and not along the trajectory that joins them, taking advantage of the geometry of the orbits and therefore creating an appropriate relative motion, as done for the natural orbits in the previous paragraph.
It's important that no trajectory segment passes through the origin or intersect the inner KOS. The steps to select the waypoint range are usually the following ones:

- Consideration of the ellipsoid of uncertainty;
- Circumscription of the ellipsoid with a sphere;
- Circumscription of the sphere with a cube, considering an additional margin of safety.

Further examples

There are other interesting examples that are worth mentioning. These examples have been studied mostly by AGI, an Ansys Company founded in 1989 to plan, design, build, and operate multi-faceted space missions[1]. Both examples are not stable trajectories, so they require frequent delta-vs to maintain.

- **Teardrop trajectory**

  The chase satellite operates in the radial direction (either positive or negative), and maintains a "tear drop" looking relative orbit to the target satellite. From this relative position, the chaser can operate fully above or below the target satellite. This relative orbit is very cheap regarding the propellant request, and it's no more risky than other forced approaches (Figure 3.8).

- **Perch point**

  This relative orbit is one where the chase satellite "parks" at a specific relative position with respect to the target satellite. This relative position is denoted in the orbital target centered frame (Hill or LVLH). An example can be a single waypoint considered in the previous strategy. It should be emphasized that not all points require a control action to maintain the position: in particular, the points along the in-track axis are stationary, unless differential perturbations, and therefore are more preferable than others.

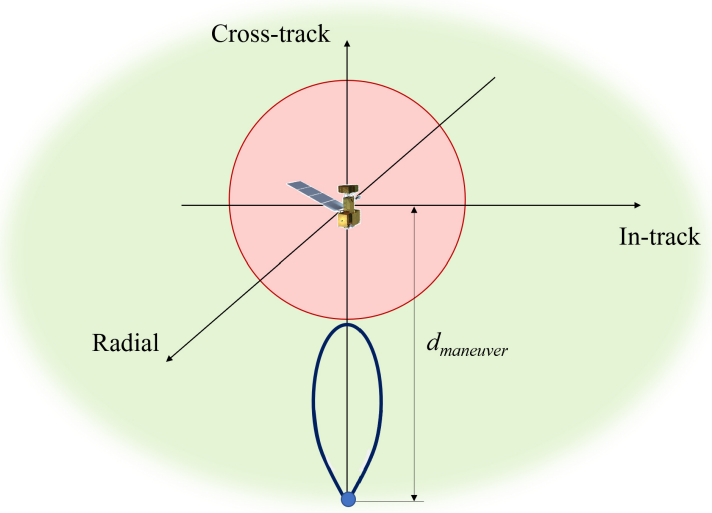


Figure 3.8: Teardrop orbit.

3.3 Inner Proximity Operations

This is the last phase in the approach to the target, the concerned area is the one identified by the innermost sphere with a radius of a few tens of meters. Entering this sphere is forbidden unless precise requirements are met, mainly coinciding with a sufficiently accurate knowledge of the target position, inertial properties and its rotational motion.

The approach to the sphere depends on the "intentions" of the chaser, that is, on the mission aim. As presented in the introduction, the reasons to perform an Autonomous Rendezvous and Docking or Capture with a non-cooperative satellite can be on-orbit servicing, scientific exploration or deflecting/deorbiting the target. Even if the first scenario has been chosen, the key issue, regardless of the mission aim, is largely to identify a particular point of the target to be reached to carry out the mission operations.

The problem is extremely complex as regards control and actuation: the chaser satellite must perform forced maneuvers to get closer to the target, chasing a precise attitude as long as necessary to perform this maneuver.
As anticipated in the introductory chapter, trajectory planning must consider the following aspects:

1. Inertial properties (unless AR&D/C is performed with defunct but known satellites) and target motion are known from the previous observation phase but with some uncertainty;

2. The target, no longer being active, is no longer even controlled and will be in a tumbling state, more or less evident depending on the environmental conditions and depending on the occurrence of particular events at the end of the mission (for example, consequences due its magnetic dipole moment);

3. Six degrees of freedom on the control and actuation must be guaranteed for the motion of the chaser, which therefore must have an adequate actuation system, suitable to pursuit any trajectories in a sufficiently accurate way, also dependent on its mass and inertia to be moved;

4. The target must always stay within the sensors field of view used to "look" at the target: the attitude control must also consider this aspect;

5. The trajectories, for obvious safety reasons, must not in any way breach or intersect the immediate target vicinity: another keep-out sphere, of a few meter radius (5-m radius), should be considered to represent the target in its uncertainty.

As one might guess, these aspects are non-trivial and greatly complicate the problem. In addition, as already noted, there are also no cases of successful missions that have been carried out, being the AR&D/C with non-cooperative satellites a relatively recent theme.

From the few studies carried out on rendezvous with non-cooperative targets in a tumbling motion, however, a common line emerges, namely the idea of chasing the capture axis, that is the axis that connects the center of gravity of the target to the target body-fixed point concerned for the future maneuver (docking or capture) to be performed (Figure 3.9).

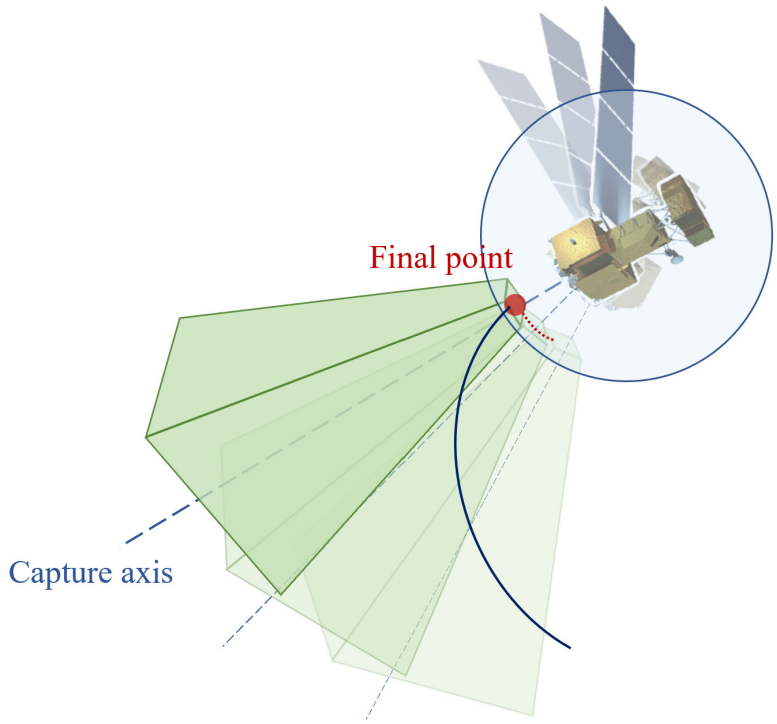


Figure 3.9: Approach to a tumbling object concept.

In this way, the chase satellite is co-moving with the target, thus largely cancelling the relative motion, a necessary condition for the maneuvers execution.
For the sake of simplicity, this final point, thinking about a robotic arm that would grasp any target fixture, will be named as "capture point" throughout the rest of this description, even if the mission aim is quite different from this.
This point, seen in the orbiting reference, will have a certain translational velocity, obviously linked to the angular velocity of the target:

$$\begin{aligned} \dot{\mathbf{p}}_c(t) &= \mathbf{v}_c(t) \\ \mathbf{v}_c(t) &= \boldsymbol{\omega}(t) \times \mathbf{p}_c(t) \end{aligned} \tag{3.5}$$

where $\mathbf{p}_c \in \mathbb{R}^3$ the linear position, $\mathbf{v}_c \in \mathbb{R}^3$ is the linear velocity and $\boldsymbol{\omega}(t) \in \mathbb{R}^3$ is the instantaneous angular velocity of the target body frame relative to the orbital frame.
Once the capture axis has been identified, even on several occasions during the approach[17], the trajectory planning follows, and this, being the target in a rotational motion, must take into account the future evolutions of its state of motion.
It can therefore be guessed that the on-board tasks of the GNC system are computationally onerous, and that certainly not all controllers are suitable for such issues.

3.3.1 Rendezvous and Docking or Capture Constraints

As reasonably stated by Leomanni[20], in order to safely achieve the rendezvous and capture objective, collisions must be avoided. Moreover, the servicer position must be confined within a suitable visibility region during the final part of the maneuver. This region commonly coincides with an approaching cone, centered in the capture axis and therefore in a rotational motion if observed in the orbital reference (see Figure 3.9).
The AR&D/C mission can be simplified through its division into two distinct phases, as done by Park[33]: a quite free-flying phase, aiming to reach the corridor, and a final approaching phase within the corridor. In each phase, the desired position to be reached by the chaser is set as an holding position.

In the free-flying phase, the holding position is considered to be along the capture axis, or the centerline of the docking cone corridor. The transition from the free-flying phase to the final docking approach occurs when the chaser is within a certain

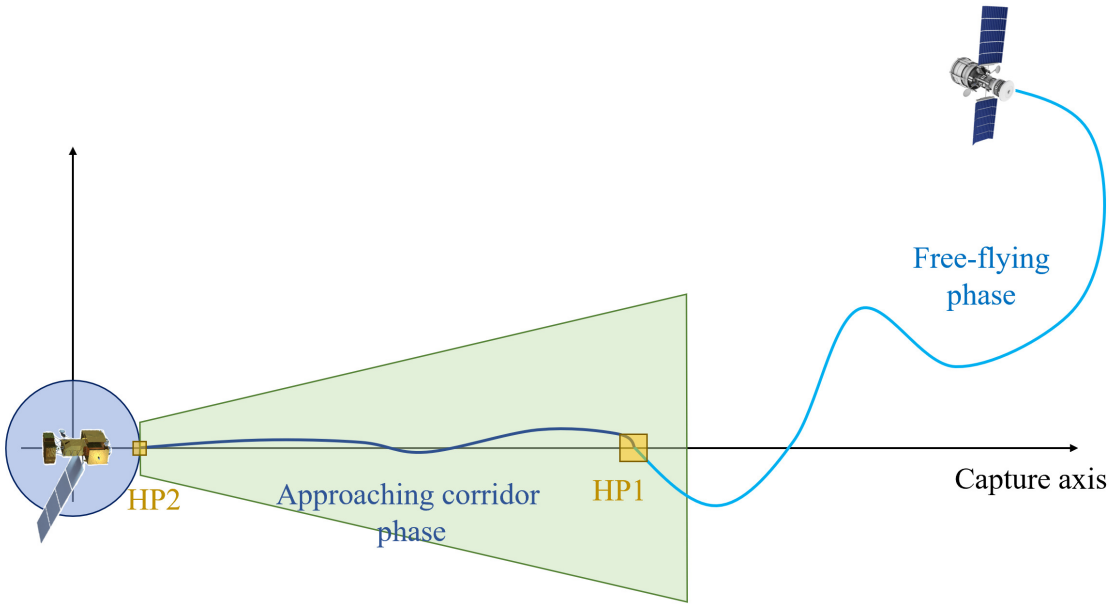


Figure 3.10: Approaching phases in body frame (in yellow the holding points HP).

distance from the holding position, named HP1 in Figure 3.10. When the second phase, within the approaching corridor, begins, the new desired holding position, named HP2 in Figure 3.10, is translated along the centerline of the cone such that the chaser successfully reaches the final point for grasping or docking with the target. The main constraints for both phases are summarized below.

Collision avoidance

The collision avoidance constraint is introduced by defining a keep-out zone in a form of an ellipsoid or sphere around the rotating target platform. For spherical keep-out zones, it is easily expressed as follows:

$$\|\mathbf{r}\|^2 \geq R^2 \tag{3.6}$$

where $\mathbf{r} \in \mathbb{R}^3$ is the position vector of the chaser in the orbital frame and R is the keep-out sphere radius.

A collision avoidance constraint can be set for the first phase: the chaser spacecraft reaches the first holding position within the corridor maintaining a minimum distance from target along its trajectory. For the afore mentioned second phase, another non-linear collision avoidance constraint guides to the second holding position for safely

capturing and precise attitude pointing to the target fixture to be grasped. Being a more critical phase, this last constraint must be set as harder than the first one.
A soft-docking constraint can be imposed so that the velocity of the spacecraft, once it reaches the holding point, is close to the velocity of the platform.

Maximum acceleration

It is reasonable to assume a maximum acceleration for the chase spacecraft, given by the union of the maximum acceleration that the spacecraft can stand (especially due to the on-board payload) and the maximum thrust that can be delivered by the thrusters:

$$|u_j| \leq u_{max} \quad \text{with } j = x_b, y_b, z_b, \tag{3.7}$$

where x_b, y_b, z_b are body principal axes.
Thinking of thrusters, it is clear that such a physical limit is also present for the minimum thrust that can be delivered, being the classic thrusters built to work in a pulsed manner (bang-bang). This point will be considered by the simulator presented in the next chapters.

Approaching cone

This constraint must be respected during the second final phase, that is, during the achievement of the final holding box, to ensure safety. Since the cone is centered around the tumbling capture axis (Figures 3.9 and 3.10), it's clear the need to predict the motion of the capture point in order to plan an approaching corridor constrained trajectory.

Considering a Model Predictive Control (linear or non-linear, in case these constraints are linearized) for example, these constraints are defined on the prediction horizon, that represents a "look to the future", as explained in the next chapter.

As done by the afore cited Park, the approaching cone constraints can be linearized by constructing four hyperplanes that define the edges of the entry cone and intersect at the capture point. When these constraints are activated, the chaser is forced to stay within these hyperplanes until capture is achieved.

Line of sight

Assuming a vision system to inspect and track the target motion, the chase spacecraft must approach during both phases maintaining the target within a Line-of-Sight (LOS) cone, that is the field of view of sensors. This is a mandatory issue: trajectory planning must take into account future evolutions of motion, and thus the chase spacecraft can't give up pointing its sensors at the target. Even assuming the possibility of mounting vision sensors on gimbal systems, in any case the set-up must take this aspect into account and continuously orient the servicer depending on its current relative position **r** from the target.

Referring to Figure 3.11, this condition can be summarized by imposing a constraint on the angle between the position vector **r** and the vector **s** along the optical axis, assuming for simplicity a vision sensor (this condition doesn't change if any other sensor, such as LIDAR, is considered):

$$\theta = \arccos\left(\frac{\mathbf{r}\cdot\mathbf{s}}{|\mathbf{r}||\mathbf{s}|}\right) \leq \theta_{max} \tag{3.8}$$

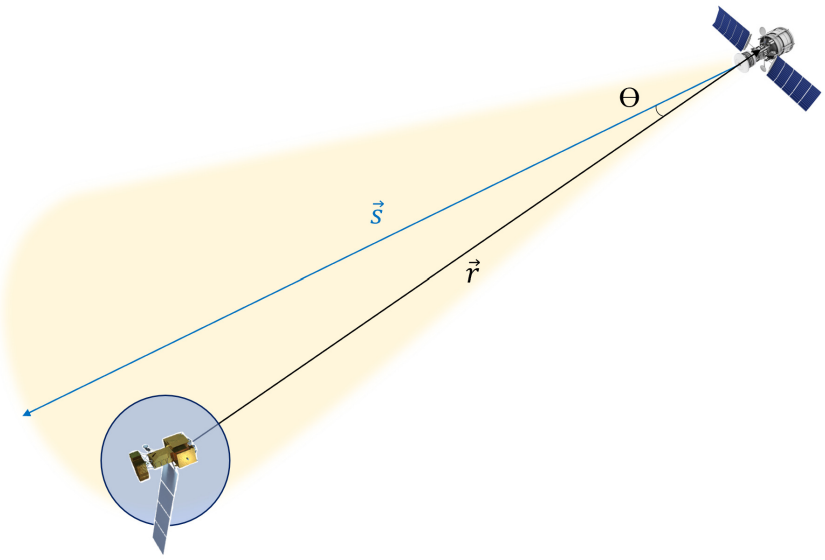


Figure 3.11: Line of Sight concept.

3.3.2 Tumbling motion of defunct satellites

As regards the AR&D/C problem, as already noted, a key point consists in having an approximate estimation of the candidate object attitude dynamics. Indeed, it is reasonable to think that not all rotational motions can be followed by the chase satellite: there will be a threshold angular velocity beyond which the latter will be unable to co-move with the target.
But how does the rotational motion of an uncontrolled body evolve in orbit?
In the absence of external torques, the angular momentum is preserved and has a fixed orientation in the inertial reference. Generally, however, in orbit there's no torques absence, although they are a minor entity: depending on the environment and therefore on the shape and height of the orbit, a satellite is more or less influenced.
The main torques acting on a generic uncontrolled body are listed below:

- Gravity gradient torques;
- Magnetic torques (eddy currents or residual magnetic moment);
- Torques due to solar radiation pressure;
- Torques due to residual atmospheric pressure.

As described by Sergey Efimov et al in a study on long-term evolution of attitude motion of defunct satellites in nearly polar orbits[35], it is generally agreed upon that the attitude dynamics of a large debris object in LEO can be qualitatively divided into three major stages. These stages are the transition to "flat" spin due to internal dissipation, exponential deceleration due to eddy-current torque, and the stage of slow chaotic motion culminating with a specific regime.

Let us assume that the defunct satellite is in a state of fast rotation, with angular velocity values such that $\omega/n \geq 10$, with n the orbit mean motion. Rotations with this magnitude order or even higher may occur as a consequence of a malfunction of the AOCS (Attitude and Orbit Control System), that lead to the mission failure (this is the case, for example, of Envisat). Defining as θ the angle between the axis with the minimal moment of inertia and the angular momentum, Sergey Efimov et al conduct simulations on its evolutions during a few years, as showed in Figure 3.12.

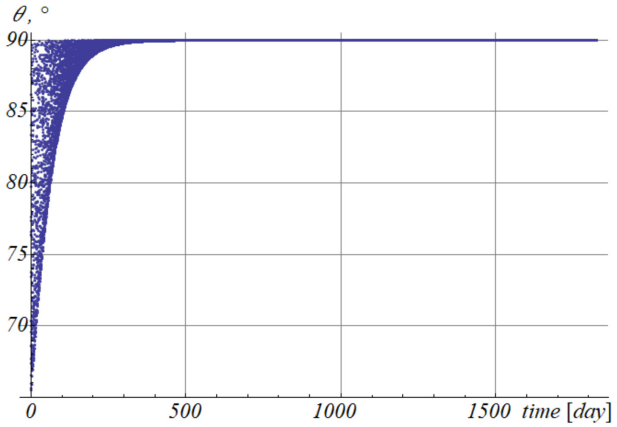


Figure 3.12: Evolution of θ (courtesy of [35]).

Starting from a generic value ($\theta = 50°$ in their simulations), it can be stated that after around one year, this angle is quite 90°.

This result is not accidental but is due to precise physical explanations: during the rotational motion there are internal dissipations that cause a decrease in the total kinetic energy, while the angular momentum of the spacecraft remains constant. Given the absolute value of the angular momentum, the kinetic energy of a rigid body is minimal when the rotation axis coincides with the axis with the greatest moment of inertia, hence internal dissipation always transforms arbitrary rotation into "flat" spin. Transition to flat spin can be aided by other dissipative factors, such as dissipative torque due to eddy currents or liquid sloshing, that is the interaction of the satellite system with residual fuel. For the sake of simplicity, these factors, as the other above listed torques, will not be considered throughout this study, and the satellite to be captured will be found in a flat spin motion.

Chapter 4

Control Strategies

GNC is the acronym for Guidance, Navigation and Control, three indispensable tasks to carry out a generic maneuver. The Navigation unit allows to know, with a certain precision, the state vector of the concerned vehicle, that often includes position, velocity, and attitude information. Guidance uses the current state to determine the maneuver sequence to follow a real-time or pre-designed trajectory. Control refers to the dynamics - forces, torques - to achieve in order to execute commanded maneuvers to follow the designed trajectory. Control design must account for navigation uncertainty.
In this chapter only the control system will be illustrated, describing the navigation system – linear or non-linear, depending on the model considered – within and intimately coupled with it.

4.1 Introduction

Still considering a continuous, linear, time-invariant model, it is possible to enlarge Eqn 2.5 assuming the presence of control and measurement systems as follows:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \end{cases} \tag{4.1}$$

where $\mathbf{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T \in \mathbb{R}^6$ is the same state vector of Eqn 2.5, $\mathbf{u}(t) \in \mathbb{R}^3$ is the

input or control action vector, $\mathbf{y}(t) \in \mathbb{R}^6$ is the output vector, $\mathbf{A} \in \mathbb{R}^{6\times6}$ is the state matrix defined in Eqn 2.6, $\mathbf{B} \in \mathbb{R}^{6\times3}$ is the control matrix in Eqn 4.2 and $\mathbf{C} = \mathcal{I}^{6\times6} \in \mathbb{R}^{6\times6}$ is the output matrix.

$$\mathbf{B} = \begin{bmatrix} 0^{3\times3} \\ \mathcal{I}^{3\times3} \end{bmatrix} \tag{4.2}$$

Being in a continuous form, Eqn 4.1 are not suitable for a controller, which instead acts in a discrete way. The dynamic model, given a time step T_s for the control, must be discretized in the following form:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}_d\mathbf{x}(k) + \mathbf{B}_d\mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{C}_d\mathbf{x}(k) \end{cases} \tag{4.3}$$

where the subscript d stands for discretized, and $\mathbf{A}_d$ and $\mathbf{B}_d$ can be derived from their continuous form as follows:

$$\mathbf{A}_d = \mathbf{\Phi}(t)|_{t=T_s} \tag{4.4}$$

$$\mathbf{B}_d = \int_0^{T_s} \mathbf{\Phi}(t)\mathbf{B}(t)dt \tag{4.5}$$

where $\mathbf{\Phi}(t)$ is the HCW continuous-time State Transition Matrix defined in Eqn 2.8.
The actions of the control must be provided in such a way as to pursue a trajectory, minimizing the error with respect to this reference: to accomplish this objective, several controllers with different features are available.
The controller chosen for the AR&D/C problem, as anticipated in the introductory chapter, is the Model Predictive Control, for reasons that will be explained in the next section.

4.2 Model Predictive Control

As suggested by J. M. Maciejowski[22], the Model Predictive Control is the only advanced control technique to have had a significant and widespread impact on industrial process control, and this is because it can deal routinely with equipment and safety constraints.

Stemmed from the chemical processing industries, MPC was developed and used in industry for several years before attracting much serious attention from academic control community. Nowadays, it can be used in different levels of the process control structure and is also able to deal with a wide variety of state and control constraints. Furthermore, the constant increase in computing speed and power certainly makes that a real prospect.

The basic idea of the MPC is using a process model to predict the output at future steps: this is done by applying an on-line numerical optimization method to obtain a sequence of control inputs that minimizes a predefined cost function over a finite receding horizon, while subjected to any certain constraints. The idea of a receding strategy is original: at each sampling instant t_k, by using the current measurements y_k and system states x_k, the entire control sequence $u_{t_k|t_{k+p}}$ is computed over the prediction horizon p, that is a finite number of future steps, but only the first element of the calculated control sequence, u_k, is used to control the system. This happens recursively: the prediction horizon window advances continuously with each step, as if it was, in a certain sense, a look to the future, and this is an enormous potential.

Actually, it is always possible to change the behaviour of the control if the system begins to deviate too much from its reference trajectory or if some parameters change in the dynamics: this guarantees a greater stability than the other control methods, as for example PID.

The other key advantage of MPC is the ability to handle constraints and make them part of the optimization: the ability to set hard or soft ranges for output and control variables perfectly fits the needs of a normal control problem, which has well-defined limits and safety and optimization requirements.

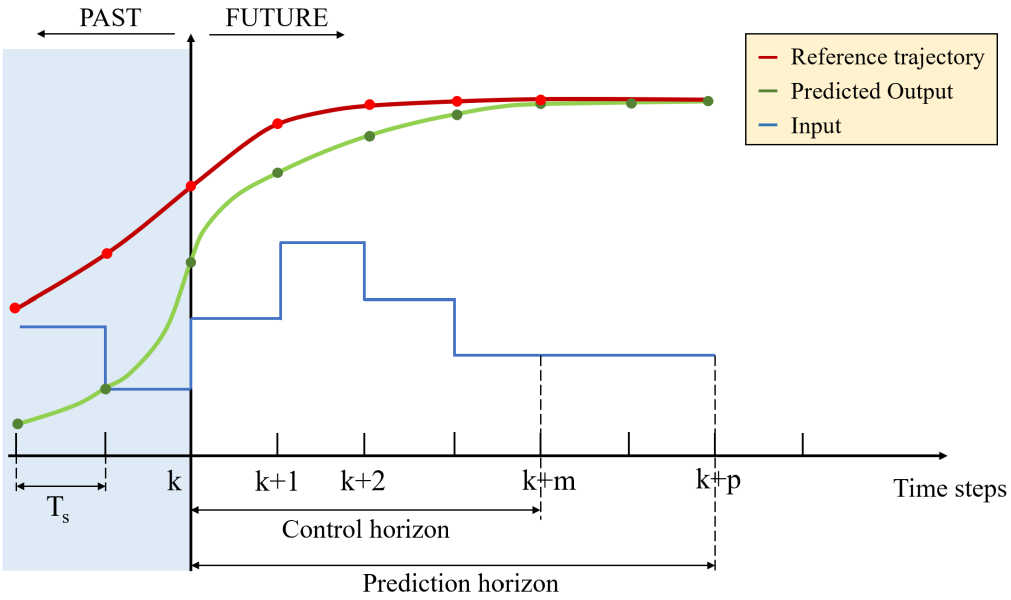


Figure 4.1: Basic idea of MPC operation.

4.2.1 Basic MPC formulation

The choice of a particular model plays a crucial role in the MPC control strategy. This dynamic model must be chosen in such a way as to provide an adequate description of the real plant dynamics and, at the same time, to allow an efficient numerical simulation. In the following description it is assumed that the real plant and the model are governed by the same equations, although this is not always acceptable.

Let us assume that the plant state space model is linear, discrete and time-invariant, in the form of Eqn 4.3, that the cost function is quadratic and that constraints are in the form of linear inequalities. Let's substitute Eqn 4.3 with a simpler and slender notation, generalizing the dimension of vectors:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned} \tag{4.6}$$

where x in an n-dimensional state vector, u is an l-dimensional input vector and y is an m-dimensional vector of measured outputs. Moreover, let's assume that the state variables are measured, so there's no need to estimate the state.

Let's denote $u(k+i|k)$ a future value of the input u and $x(k+i|k)$ and $y(k+i|k)$ the

predictions of variables x and y at time $k+i$, made at time k, on the assumption that some sequence of inputs $u(k+j|k)$ (with $j = 0, 1, \ldots, i-1$) will have occurred.

Cost function

The purpose of the cost function is to ensure the pursuit of the future exit y in the considered horizon, minimizing the control action. What it does, therefore, is to penalize deviations of the predicted controlled outputs $y(k+i|k)$ from a vector reference trajectory $r(k+i|k)$, that often is a predetermined trajectory, and deviations of control inputs $u(k+i|k)$ from some ideal resting values $u_{ref}(k+i|k)$, if any:

$$\begin{aligned} J(k) \;=\; & \sum_{i=0}^{p} \|y(k+i|k) - r(k+i|k)\|^2_{Q(i)} + \sum_{i=0}^{m-1} \|\Delta u(k+i|k)\|^2_{R(i)} + \\ & + \sum_{i=0}^{p-1} \|u(k+i|k) - u_{ref}\|^2_{S(i)} + \rho_\epsilon \epsilon(k)^2 \end{aligned} \tag{4.7}$$

where p and m are the prediction and control horizons respectively, $Q(i)$, $R(i)$, $S(i)$ are weighting symmetric matrices of output, control rates and control errors respectively, at index i, and ϵ is a slack variable to set the constraint relaxation. It's to be assumed that $m \leq p$ and that $\Delta u(k+i|k) = 0$ for $m \leq i \leq p$, so that $u(k+i|k) = u(k+m-1|k)$ for all $m \leq i \leq p$.

As stated by Maciejowski[22], the prediction an control horizons p and m, the weights $Q(i)$, $R(i)$ and $S(i)$ and the reference trajectory $r(k+i)$ all affect the behaviour of the closed-loop combination of the plant and MPC controller. Some of these parameters, weights above all, reveal to be tuning parameters to be adjusted not only to give satisfactory dynamic performance, but also to better achieve the economic objectives.

Constraints

As already noted before, constraints are a key point in favour of MPC controllers. All real processes, more or less, are subject to constraints, due to construction, efficiency or safety reasons. These constraints can be placed on the output variable y, on the amplitude of the control signal u (for example actuator ranges), or on the variation of the control signal Δu (for example actuator slew rates).

As already established, constraints in the form of linear inequalities will be considered, as follows:

$$\begin{cases} u_{min} \leq u(k) \leq u_{max} \\ \Delta u_{min} \leq \Delta u \leq \Delta u_{max} \\ y_{min} \leq y(k) \leq y_{max} \end{cases} \tag{4.8}$$

These conditions can be re-arranged in a less intuitive but more general form as follows:

$$\begin{cases} E\left[\Delta u(k|k), ..., \Delta u(k+m-1|k,1)\right]^T \leq [0, ..., 0]_{m\times 1} \\ F\left[u(k|k), ..., u(k+m-1|k,1)\right]^T \leq [0, ..., 0]_{m\times 1} \\ G\left[y(k|k), ..., y(k+p|k,1)\right]^T \leq [0, ..., 0]_{p\times 1} \end{cases} \tag{4.9}$$

where E, F and G are matrices of suitable dimensions.

Problem formulation

Once the main features of this controller have been formulated, it's possible to show how it works. The MPC solves on-line the minimization of Eqn 4.7, subject to the constraints in Eqn 4.9, respecting the relations between the state x, the output vector y and the control vector u dictated by the equations of state 4.6.
The solution is the optimal control sequence:

$$\Delta\mathcal{U}^*(k) = [\Delta u^*(k|k), \Delta u^*(k+1|k), \ldots, \Delta u^*(k+m-1|k)], \tag{4.10}$$

but what interests is the first control action, calculated as:

$$u(k) = u(k-1) + \Delta u^*(k|k) \tag{4.11}$$

Given the multiple parameters, it is possible to intervene in various ways on the performance of the MPC:

- lengthening or shortening predictions and control horizons;
- loosening or tightening the constraints, with the possibility of setting them hardly or softly;

- weighing variables more than others.

As one can intuit, lengthening the values of prediction and control horizons is possible up to a certain extent: actually, the advantage of prediction can turn into a disadvantage if one thinks of the computational load required. There is another possible reason for not lengthening the prediction horizon too much: if the real dynamics of the system is slightly different from that of the model, the prediction provided by this latter on future system evolution, starting from an estimate of the current state, accumulates errors for very long prediction horizons. The solution, therefore, depends on the model accuracy and is a trade-off between obtaining better performance and consuming on board resources in terms of computing power.

4.2.2 Formulation as a QP problem

Due to its form, it is useful to formulate the optimization problem seen above in a Quadratic Programming problem. A general QP problem can be expressed as follows:

$$\begin{cases} \min\limits_{\theta} \quad \frac{1}{2}\theta^T \Phi \theta + \phi^T \theta \qquad (\Phi = \Phi^T \geq 0) \\ \Omega\theta \leq \omega \end{cases} \tag{4.12}$$

In the following, the steps needed to lead the problem presented in the previous section back to the general formulation in Eqn 4.12 are presented. It's possible to rewrite the cost function in Eqn 4.7 as

$$J(k) = \|\mathcal{Z}(k) - \mathcal{T}(k)\|^2_{\mathcal{Q}} + \|\Delta\mathcal{U}(k)\|^2_{\mathcal{R}} + \|\mathcal{U}(k) - \mathcal{U}_{ref}(k)\|^2_{\mathcal{S}} + \rho_\epsilon \epsilon(k)^2 \tag{4.13}$$

with vectors defined as follows:

$$\begin{cases} \mathcal{Z}(k) = [y(k|k)^T, ..., y(k+p|k)^T]^T \\ \mathcal{T}(k) = [r(k|k)^T, ..., r(k+p|k)^T]^T \\ \mathcal{U}(k) = [u(k|k)^T, ..., u(k+p-1|k)^T]^T \\ \mathcal{U}_{ref}(k) = [u_{ref}(k|k)^T, ..., u_{ref}(k+p-1|k)^T]^T \\ \Delta\mathcal{U}(k) = [\Delta u(k|k)^T, ..., \Delta u(k+m-1|k)^T]^T \end{cases} \tag{4.14}$$

and where $\mathcal{Q}$, $\mathcal{R}$, $\mathcal{S}$ are larger diagonal weighting matrices built by setting Q_j, R_j, S_j as diagonal terms. Neglecting mathematical steps[22], it is possible to rewrite the first equation of Eqn 4.6 as follows:

$$\mathcal{Z}(k) = \Psi x(k) + \Upsilon u(k-1) + \Theta\Delta\mathcal{U}(k) \tag{4.15}$$

where

$$\Psi = \begin{bmatrix} CA \\ \vdots \\ CA^{m-1} \\ CA^m \\ \vdots \\ CA^p \end{bmatrix}; \quad \Upsilon = \begin{bmatrix} CB \\ \vdots \\ \sum_{i=0}^{m-1} CA^iB \\ \sum_{i=0}^{m} CA^iB \\ \vdots \\ \sum_{i=0}^{p-1} CA^iB \end{bmatrix}; \quad \Theta = \begin{bmatrix} CB & \cdots & 0 \\ CAB+CB & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{m-1} CA^iB & \cdots & CB \\ \sum_{i=0}^{m} CA^iB & \cdots & CAB+CB \\ \vdots & \vdots & \vdots \\ \sum_{i=0}^{m} CA^iB & \cdots & \sum_{i=0}^{p-m} CA^iB \end{bmatrix};$$

and to define the 'tracking error', or the response that will occur over the prediction horizon if no input changes were made ($\Delta\mathcal{U}(k) = 0$), as:

$$\mathcal{E} = \mathcal{T}(k) - \Psi x(k) - \Upsilon u(k-1) \tag{4.16}$$

It is then possible to write $\mathcal{U}(k) - \mathcal{U}_{ref}(k)$ of Eqn 4.13 as a function of $\Delta\mathcal{U}(k)$:

$$\mathcal{U}(k) - \mathcal{U}_{ref}(k) = \Lambda\Delta\mathcal{U}(k) + \mathcal{I}u(k-1) - \mathcal{U}_{ref}(k) = \Lambda\Delta\mathcal{U}(k) + \mu(k) \tag{4.17}$$

where Λ is the simple matrix (made of identity matrices) that maps $\mathcal{U}(k)$ to $\Delta\mathcal{U}(k)$. Due to these consideration, it is possible to reformulate Eqn 4.13 as follows:

$$\begin{aligned} J(k) &= \|\Theta\Delta\mathcal{U}(k) - \mathcal{E}(k)\|^2_{\mathcal{Q}} + \|\Delta\mathcal{U}(k)\|^2_{\mathcal{R}} + \|\mathcal{U}_{ref}(k) - \mathcal{U}(k)\|^2_{\mathcal{S}} + \rho_\epsilon\epsilon(k)^2 \\ &= \left[\Delta\mathcal{U}(k)^T\Theta^T - \mathcal{E}(k)^T\right]\mathcal{Q}\left[\Delta\mathcal{U}(k)\Theta - \mathcal{E}(k)\right] + \Delta\mathcal{U}(k)^T\mathcal{R}\Delta\mathcal{U}(k) + \\ &\quad \left[\Lambda\Delta\mathcal{U} + \mu(k))\right]^T\mathcal{S}\left[\Lambda\Delta\mathcal{U} + \mu(k))\right] + \rho_\epsilon\epsilon(k)^2 \\ &= \mathcal{E}(k)^T\mathcal{Q}\mathcal{E}(k) - 2\Delta\mathcal{U}(k)^T\Theta^T\mathcal{Q}\mathcal{E}(k) + 2\Delta\mathcal{U}(k)^T\Lambda^T\mathcal{S}\mu(k) + \\ &\quad \Delta\mathcal{U}(k)^T\left[\Theta^T\mathcal{Q}\Theta + \Lambda^T\mathcal{S}\Lambda + \mathcal{R}\right]\Delta\mathcal{U}(k) + const \end{aligned}$$

But this has the form

$$J(k) = \Delta\mathcal{U}(k)^T\mathcal{H}\Delta\mathcal{U}(k) - \Delta\mathcal{U}(k)^T\mathcal{G} + const, \tag{4.18}$$

where $\mathcal{G} = 2\Theta^T\mathcal{Q}\mathcal{E}(k)$ and $\mathcal{H} = \Theta^T\mathcal{Q}\Theta + \Lambda^T\mathcal{S}\Lambda + \mathcal{R}$.
As can be noted, Eqn 4.18 is analogue to the first equation of Eqn 4.12.
Consider now the second part of QP general form, namely the constraints. The consecutive steps, here reported, necessary to get to the final form, are those listed in the aforementioned book of Maciejowski[22]. Eqn 4.9 can be recalled in the form:

$$E\begin{bmatrix} \Delta\mathcal{U}(k) \\ 1 \end{bmatrix} \leq 0 \tag{4.19}$$

$$F\begin{bmatrix} \mathcal{U}(k) \\ 1 \end{bmatrix} \leq 0 \tag{4.20}$$

$$G\begin{bmatrix} \mathcal{Z}(k) \\ 1 \end{bmatrix} \leq 0 \tag{4.21}$$

These have to be expressed as constraints on $\Delta\mathcal{U}(k)$. Consider Eqn 4.20 as first, and

suppose F has the form:

$$F = [F_1, F_2, \ldots, F_p, f]$$

where F_i is a $q \times m$ matrix, and f has the size $q \times 1$, Eqn 4.20 can be written as:

$$\sum_{i=1}^{p} F_i \hat{u}(k+i-1|k) + f \leq 0$$

Since

$$\hat{u}(k+i-1|k) = u(k-1) + \sum_{j=0}^{i-1} \Delta\hat{u}(k+j|k)$$

the previous equation can be written as:

$$\sum_{j=1}^{p} F_j \Delta\hat{u}(k|k) + \sum_{j=2}^{p} F_j \Delta\hat{u}(k+1|k) + \cdots + F_p \Delta\hat{u}(k+p-1|k) + \sum_{j=1}^{p} F_j u(k-1) + f \leq 0$$

Finally, defining $F_i = \sum_{j=i}^{p} F_j$ and $F = [F_1, \ldots, F_p]$, Eqn 4.20 can be rewritten as:

$$F\Delta\mathcal{U}(k) \leq -F_1 u(k-1) - f \tag{4.22}$$

with the right-hand side vector known at time k.

As can be noted, Eqn 4.22 is now expressed as a linear inequality constraint on $\Delta\mathcal{U}(k)$.

A similar thing can be done for Eqn 4.21: assuming full state measurements, by using Eqn 4.15, it's possible to write it as:

$$G \begin{bmatrix} \Psi x(k) + \Upsilon u(k-1) + \Theta\Delta\mathcal{U}(k) \\ 1 \end{bmatrix} \leq 0$$

Now letting $G = [\Gamma, g]$, where g is the last column of G, and after some rearrangement, this is the same as:

$$\Gamma\Theta\Delta\mathcal{U}(k) \leq -\Gamma[\Psi x(k) + \Upsilon u(k-1)] - g \tag{4.23}$$

which is in the required form.

Finally, what remains is putting the inequality Eqn 4.19 into the form:

$$W\Delta\mathcal{U}(k) \leq w \tag{4.24}$$

Then it is possible to assemble inequalities in Eqn 4.22, 4.23 and 4.24 into the following single inequality:

$$\begin{bmatrix} F \\ \Gamma\Theta \\ W \end{bmatrix} \Delta\mathcal{U}(k) \leq \begin{bmatrix} -F_1 u(k-1) - f \\ -\Gamma[\Psi x(k) + \Upsilon u(k-1)] - g \\ w \end{bmatrix} \tag{4.25}$$

As can bee seen, Eqn 4.25 is in the form $\Omega\theta \leq w$, and with Eqn 4.18 completes the Quadratic Programming problem definition presented at the beginning of this section.

4.2.3 Observer-based MPC

The assumption that all state variables can be always measured directly is quite ideal: what one has is rather an estimate $\hat{x}(k|k)$ of the state $x(k)$, basing on measurements of $y(k)$ up to time k and on previous control actions.

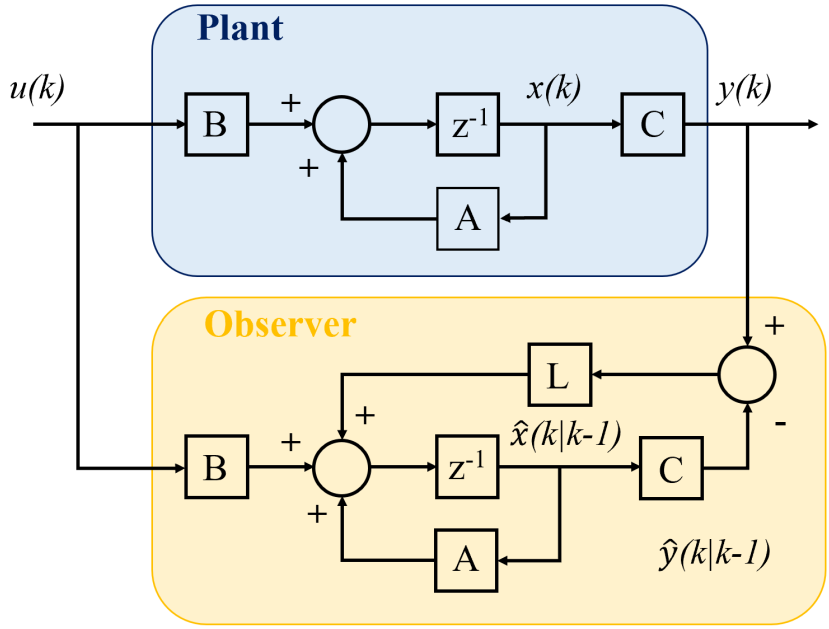


Figure 4.2: General structure of a state observer.

To obtain this estimate of the state vector, an observer must be used. Also in this case, the theory here reported has as reference the Maciejowski's book. The general

structure of a state observer is shown in Figure 4.2, for a plant described by the following equations:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned} \tag{4.26}$$

These are the same equations of the plant ones in Eqn 4.6, but, looking at Figure 4.2, with feedback from the measured plant output, through the gain matrix L, to correct the state estimate $\hat{x}$.

Starting from these equations, the observer can be defined through the following:

$$\hat{x}(k|k) = \hat{x}(k|k-1) + L'[y(k) - \hat{y}(k|k-1)] \tag{4.27}$$

$$\hat{x}(k+1|k) = A\hat{x}(k|k) + Bu(k) \tag{4.28}$$

$$\hat{y}(k|k-1) = C\hat{x}(k|k-1) \tag{4.29}$$

Substituting Eqn 4.29 into Eqn 4.27, and eliminating $\hat{x}(k|k)$, it can be written:

$$\begin{aligned} \hat{x}(k+1|k) &= A(I - L'C)\hat{x}(k|k-1) + Bu(k) + AL'y(k) \\ &= (A - LC)\hat{x}(k|k-1) + Bu(k) + Ly(k) \end{aligned} \tag{4.30}$$

where $L = AL'$. This is a stable system if the eigenvalues of $A - LC$ lie inside the unit disk. Furthermore, defining the state estimation error as $e(k) = x(k) - \hat{x}(k|k-1)$, and then using Eqn 4.26, this final expression can be given:

$$e(k+1) = (A - LC)e(k) \tag{4.31}$$

Eqn 4.31 clearly shows that the state estimation error converges to zero if the observer is stable, at a rate determined by the eigenvalues of $A - LC$.

As Maciejowski[22] stated, if the pair (A, C) is observable, then given an arbitrary set of locations in the complex plane, a gain matrix L exists which places the observer eigenvalues at these locations. The problem of finding L is equivalent to the state-feedback pole-placement problem, and can be solved using the same algorithm.

Assuming that the state and output equations of the plant are subjected to white noise disturbances with known covariance matrices, then L can be chosen in such a way to minimize the mean square state estimation error. The observer is then known as a *Kalman Filter*.

Kalman filter

Let's assume that the measurements are affected by a random error, called $v(k)$, and that there is a random error in the process, called $w(k)$, that is, an error or an uncertainty in the mathematical model. This latter sentence means that A is not known exactly, and therefore it's more difficult to control how quickly the error in Eqn 4.31 will vanish, or how faster the estimated state $\hat{x}$ converge to the true state x.
Eqn 4.26 can then be rewritten considering these errors as follows:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Ww(k) \\ y(k) &= Cx(k) + Vv(k) \end{aligned} \tag{4.32}$$

where W and V are state and measurement noise matrices.
Although these errors are random, it is possible to give a description of their mean properties using probability theory. Let us assume that the noises have Gaussian distribution, centered at the origin, with matrices of covariance R_k for measurement noise and Q_k for process noise:

$$\begin{cases} v(k) \sim \mathcal{N}(0, R_k) \\ w(k) \sim \mathcal{N}(0, Q_k) \end{cases} \tag{4.33}$$

Kalman filters combine the measurements and the prediction state estimate to find the optimal estimate of the state, in presence of process and measurement noise. This is done by multiplying the two probability functions together, thrusting more measurements or the prediction state estimate depending on the noise they are affected by.
The resulting curve is also a Gaussian function with a smaller variance, arose from the interaction of the previous two. The multiplication of these two probability density functions relates to the discrete Kalman filter equation that follows:

$$\hat{x}_k = A\hat{x}_{k-1} + Bu_k + L_k(y_k - C(A\hat{x}_{k-1} + Bu_k)) \tag{4.34}$$

where indices k and $k-1$ are subscripted for compactness. This equation is very similar to Eqn 4.30 of the general state observer: while this latter deals with deterministic

systems, the Kalman filter one deals with stochastic systems.
The first part of the equation predicts the current state through the state estimate from the previous time step and the current input. This can be called *a priori estimate*, and is calculated before the current measurement is taken:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \tag{4.35}$$

The second part of the equation uses the measurement y_k to update the a priori estimate in $\hat{x}_k^+$, called *a posteriori estimate*. Eqn 4.34 can be rewritten as:

$$\hat{x}_k^+ = \hat{x}_k^- + L_k(y_k - C\hat{x}_k^-) \tag{4.36}$$

The Kalman filter algorithm is initialized by providing the initial estimates $\hat{x}_0^-$ and P_0^-, which is the state estimate error covariance, built considering the process noise and the propagation of the uncertain $\hat{x}_{k-1}^-$.
It is then possible to identify two major steps in the Kalman filter process.

1. **State correction through measurements updated**

   Firstly the Kalman gain matrix is calculated:

   $$K_k = \frac{P_k^- C^T}{CP_k^- C^T + VR_kV^T} \tag{4.37}$$

   The Kalman gain K_k (that is L_k in Eqn 4.34) is calculated such that it minimizes the a posteriori error covariance: as mentioned before, it is possible to rely more on measurements than on estimate states or vice versa, depending on the respective covariance matrix and trying to minimize the combination of these.
   Secondly, this matrix is used to update the state estimate and the state estimate error covariance

   $$\begin{cases} \hat{x}_k^+ = \hat{x}_k^- + K_k(y_k - C\hat{x}_k^-) \\ P_k^+ = (I - K_kC)P_k^- \end{cases} \tag{4.38}$$

2. **Prediction step**

   Having now $\hat{x}_k^+$ and P_k^+, it's possible to predict the next state estimate and the

covariance matrix:

$$\begin{cases} \hat{x}_{k+1}^{-} = A\hat{x}_k + Bu_k \\ P_{k+1}^{-} = AP_kA^T + WQ_kW^T \end{cases} \tag{4.39}$$

Once the prediction step is over, the a posteriori estimates are used to predict the new a priori estimates of the next step, and so on.
This algorithm can also be performed by using multiple sensors: actually, the Kalman filter acts as a sensor fusion algorithm, mixing the different information from multiple sensors and weighing them depending on their reliability to provide a more accurate estimate. In this case, the equations are unaltered, what changes is only the size of matrices, which increases as the number of sensors increases.

4.3 Nonlinear Model Predictive Control

A linear time-invariant MPC can be used if system and constraints are linear and the cost function is quadratic: an MPC problem with these properties gives rise to a convex optimization problem where the cost function has a single global optimum, and the aim of the optimization is to find this optimum.

If the plant exhibits soft nonlinearities, linear MPC can still be used and one can benefit from the nice properties of the convex optimization problem. However, classic MPC can no longer be used and there's the need of other MPC formulations. These are, for example, the Adaptive and Gain-scheduled MPCs [26], whose base principle is to linearize the nonlinear system around consequent operating points and to use the latest linear model as the internal model at each step. The linearized model is not necessarily updated at each step, but may be left unchanged as long as the plant is in the vicinity of a particular operating condition. This results in a QP problem to be solved at each step, although the model changes from time to time.

In the Adaptive MPC a linear model is computed on-line as the operation conditions change, but the number of states and constraints must be unaltered along the prediction horizon. In the Gain-scheduled MPC this last condition is guaranteed by using multiple

MPCs for different operating conditions, but in a off-line mode, thus demanding a larger memory. Further details of their working principles are neglected, since these controllers have not been used in this thesis.

If the plant exhibits severe nonlinearities and constraints and cost function are not linear too, the usefulness of predictive control based on a linearized model is limited and the obvious solution is to use a nonlinear model and then a nonlinear MPC.
On one hand, this method is the most powerful one as it uses a nonlinear plant model, thus more representative of the real dynamics: the predictions made by the controller are more accurate, and this leads to better performances. On the other hand, it is also the most challenging one to solve in real-time, because when there are nonlinear constraints and a nonlinear cost function, convexity of the optimization is then lost: the problem becomes nonconvex. Therefore, the cost function may have multiple local optima and finding the global optimum may be hard, especially for on-line applications.

In reality, as stated by J.M. Maciejowski[22], the disadvantage is not so much that the global optimum may not be found, since the use of a linearized model only gives the illusion of finding a global optimum – one finds the global optimum of the wrong problem. The real disadvantage for on-line use is that one has very little idea of how long each optimization step will take to complete, whether it will ever terminate, what to do if a feasible solution cannot be found, and so on.
Due to significant nonlinearities, a time-varying linear model which varies during the prediction horizon must be used: at each point in the horizon one uses the linearization corresponding to the state at which one expects to be at that time.
The efficiency of solving a nonconvex optimization problem that requires a large number of computations depends on the nonlinear solver available.

Sequential Quadratic Programming

If a nonlinear internal model is used, then several optimization algorithms are available. The one which is closest to the algorithms presented for MPC is the Sequential Quadratic Programming, SQP, that essentially is Newton's method of optimization with constraints added through Lagrange multipliers. To give an approximative idea

of its working principle, some steps from the Maciejowski's book will follow.
Suppose that a general constrained optimization problem is to be solved, in the following form:

$$\min_{\theta} \{V(\theta) : H(\theta) = 0, \Omega_j(\theta) \leq 0\} \tag{4.40}$$

where $H_i(.)$ and $\Omega_j(.)$ are sets of nonlinear functions, and that an iterate θ_k has been obtained. The SQP algorithm makes a quadratic approximation of $V(\theta_k)$:

$$q_k(d) = \frac{1}{2} d^T \nabla^2_{\theta\theta} L(\theta_k, \lambda_k) d + \nabla V(\theta_k)^T d \tag{4.41}$$

where $L(\theta, \lambda) = V(\theta) + \sum_i \lambda_i H_i(\theta) + \sum_j \lambda_j \Omega_j(\theta)$ is the Lagrangian. The next iterate, θ_{k+1}, is given by

$$\theta_{k+1} = \theta_k + d_k d \tag{4.42}$$

where d_k is found by solving the QP problem which results from minimizing $q_k(d)$, subject to local linear approximations of the constraints:

$$\min_{\theta} \left\{ q_k(d) : H_i(\theta_k) + \nabla H_i(\theta_k)^T = 0, \Omega_j(\theta_k) + \nabla \Omega_j(\theta_k)^T d \leq 0 \right\} \tag{4.43}$$

This is the basic idea of the SQP algorithm, even if there are several variations: an example is the feasible SQP, a strategy that, ensuring to satisfy the nonlinear constraints, allows the iterates to converge even if there is insufficient time and the best iterate must be found in any case.

4.3.1 Observer-based nonlinear MPC

Another problem which becomes very prominent with nonlinear MPC is the state estimation: since the model is nonlinear, obtaining an estimate $\hat{x}(k|k)$ of the true state $x(k)$ becomes a non-trivial issue. As mentioned before, the standard way of estimating the state of a dynamic system from input-output measurements is to use a state observer. As regards linear models, there is a solid background theory, and, if statistical information is available about measurement noise and disturbances, a Kalman filter can be used to obtain optimal estimates.

Concerning nonlinear models, instead, there is some observer theory, but it is much less complete. Even if optimal state estimation problems can still be defined, they usually cannot be solved exactly, and one must necessarily make ad-hoc approximations.

When the model is no longer linear, the state transition function and measurement function also become non-linear.
State equations in Eqn 4.32 should be turned on the following form:

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) + W w_k \\ y_k &= g(x_k) + V v_k \end{aligned} \tag{4.44}$$

where f, g are nonlinear functions and, for simplicity, process and measurement noises are still additive terms. If this were not the case, the latter should be incorporated respectively into the state and measurement functions, thus getting a more general form, as reported in [24]:

$$\begin{aligned} x_{k+1} &= f(x_k, w_k, u_k^s) \\ y_k &= h(x_k, v_k, u_k^m) \\ v(k) &\sim \mathcal{N}(0, R_k) \\ w(k) &\sim \mathcal{N}(0, Q_k) \end{aligned} \tag{4.45}$$

where u_k^s and u_k^r are additional input arguments and the other terms have already been defined in the previous section.
Unfortunately, the Kalman filter works and converges only with linear systems: considering the probabilistic theory of Gaussian functions, the linearity of the model does not lead to a change in the shape of the probability density curve, which, on the contrary, happens in the case of nonlinear models.
Since the normal Kalman filter does not work, another filter form should be considered: the Extended Kalman filter (or EKF).

Extended Kalman Filter

In broad terms, an EKF behaves as the nonlinear MPC does with nonlinear systems: it linearizes the nonlinear function around the mean value of the current state estimate. In this case, however, not only the state function, but also the measurement function requires linearization. Through corrections of the predicted states and predictions of the future states, generated knowing the state function and therefore the model, it is possible, also in this case, to converge with this filter.

At each time step, the linearization is performed locally, and the resulting Jacobian matrices are then used in the prediction update states of the Kalman filter algorithms. Following what reported in [24], the general approach is given:

1. Given Eqn 4.45, the filter must be initialized with initial values of the estimated state $\hat{x}_{k|k-1}$ and state estimation error covariance matrix $P_{k|k-1}$;

2. Given a measurement of the state y, the estimated state and matrix P must be updated. To perform this point, Jacobians of the measurement function must be calculated:

$$\begin{aligned} C_k &= \left.\frac{\partial h}{\partial x}\right|_{\hat{x}(k|k-1)} \\ S_k &= \left.\frac{\partial h}{\partial v}\right|_{\hat{x}(k|k-1)} \end{aligned} \tag{4.46}$$

   and then, with K the Kalman gain, the update can be carried out:

$$\begin{aligned} K_k &= P_{k|k-1}C_k^T(C_kP_{k|k-1}C_k^T + S_kR_kS_k^T)^{-1} \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k(y_k - h(\hat{x}_{k|k-1}, 0, u_k^m)) \\ \hat{P}_{k|k} &= P_{k|k-1} - K_kC_kP_{k|k-1} \end{aligned} \tag{4.47}$$

3. After correcting the state estimation with measurements, then the prediction of the state and state estimation error covariance for the next step is necessary. To do this, Jacobians of the state function must be calculated:

$$\begin{aligned} A_k &= \left.\frac{\partial f}{\partial x}\right|_{\hat{x}(k|k)} \\ G_k &= \left.\frac{\partial f}{\partial w}\right|_{\hat{x}(k|k)} \end{aligned} \tag{4.48}$$

   Given these two matrices, the prediction step can be performed:

$$\begin{aligned} P_{k+1,k} &= A_kP_{k|k}A_k^T + G_kQ_kG_k^T \\ \hat{x}_{k+1|k} &= f(\hat{x}_{k|k}, 0, u_k^s)) \end{aligned} \tag{4.49}$$

Using the EKF, the calculation of Jacobians can be difficult and reveal to be a high computational cost demanding problem, especially when the system is highly non-linear. In this case, an unscented Kalman filter, here not developed for brevity, is recommended[24].

Chapter 5

AR&D/C scenario definition

In this chapter the AR&D/C problem with a non-cooperative target will be presented and applied to a real space mission. Specifically, the most critical issues of this rendezvous scenario and the tools used to solve them will be illustrated and discussed.

As already mentioned, only the last two phases of the three presented in Chapter 3 will be analysed, to focus mainly on proximity operations rather than phasing and far-field approach maneuvers, already widely tested and with a wide flight heritage (cooperative RVD). Furthermore, for simplicity, any missed burns, unfavourable lighting conditions or additional disturbance actions were not considered in the computation of the approaching trajectories.

As for the near-field rendezvous phase, the trajectories generated will be purely translational, with 3 degrees of freedom, since the attitude evolution is not of particular relevance given the high distance between the chaser and the target in this phase. As for the inner proximity operations phase, instead, attitude will also be managed by the control law and the actuators, as will be explained in the appropriate chapter.

Since the problems are widely distinct, even the controllers and the navigation system are different: for the first part, in fact, being both the HCW equations and the constraints linear, a Model Predictive Control, paired with a linear Kalman filter, will be sufficient to perform the trajectory tracking; for the last part, on the other hand, the attitude control (clearly nonlinear) and the presence of nonlinear constraints also for the translational part lead to the implementation and development of a nonlinear

Model Predictive Control, coupled with an extended formulation of the Kalman filter. A first part regarding the presentation of the mission chosen as a case study will be followed by a section on the implemented simulator and its working principles.

5.1 Case study: mission OSAM-1

The case study considered for the simulations is OSAM-1, an on-orbit servicing mission to restore Landsat-7, scheduled to launch in 2023. Some specifications regarding this mission will be presented in the following, after an overview of the target to be rescued.

LandSat7

As reported in eoPortal Directory [7], the Landsat-7 satellite (Figure 5.1) is part of NASA's ESE (Earth Science Enterprise) program, a joint venture of NASA and USGS (United States Geological Survey). The overall mission objective is to extend and improve upon the long-term record of medium-resolution multispectral imagery of the Earth's continental surfaces provided by the earlier Landsat satellites.

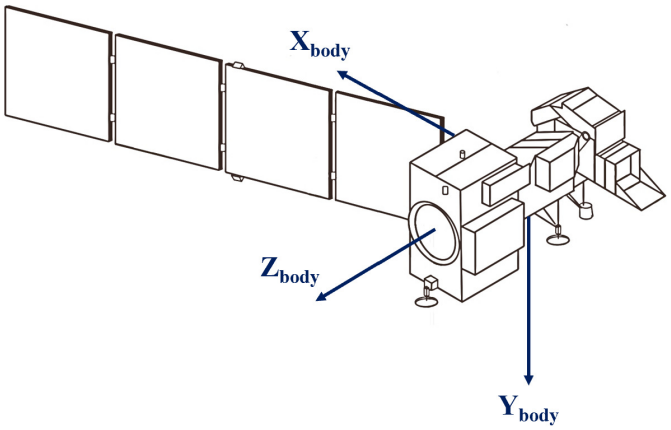


Figure 5.1: LandSat 7 model.

Operating since 1999, Landsat7 weighs 2200 kg, is 4.3 m long and 2.8 m in diameter, and lies on an orbit with the specifics listed in Table 5.1:

Orbital parameters	
Regime	sun-synchronous orbit
Altitude	705 km
Inclination	98.21°
Period	98.83 minutes
Repeat interval	16 days

Table 5.1: Landsat7 orbital parameters

Despite a 5-year life planned, Landsat7 remained on-station much longer than expected, fulfilling its science mission until September 2021, when Landsat9 was successfully launched to take its place in the standard orbit altitude, which, combining with the Landsat8 one, still operating, allows data to be collected every eight days over every location on Earth. As reported in [39], Landsat7 is now exiting the constellation and lower its orbit by 8 km to prepare for servicing by NASA's On-Orbit Servicing, Assembly, and Manufacturing-1 (OSAM-1) mission. This mission will provide Landsat7 with the needed fuel for a successful decommissioning, as described in the next paragraph.

According with [7], further ideas from NASA concern the possibility of turning the satellite into a transfer radiometer, acting as a calibration instrument for Landsats 8 and 9, and perhaps even extend its scientific utility. The first possibility, in any case, is materializing, being OSAM-1 under construction.

5.1.1 OSAM-1 overview

NASA's On-Orbit Servicing Assembly and Manufacturing (OSAM-1) mission (Figure 5.2), formerly called Restore-L, after passing the Critical Design Review (CDR) in April 2021, is now entering the mechanical building phase of the spacecraft.
This is an innovative mission, the first of its kind in low Earth orbit, and, according with [28], the success of this demonstration would spur a new era in sustainable space operations essential for the science, national security and space operations.

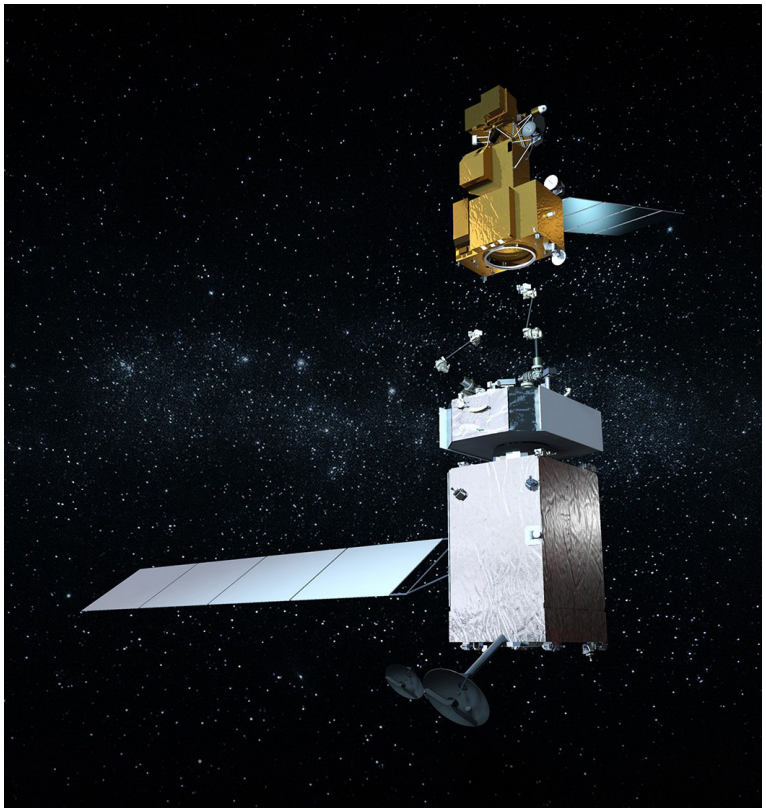


Figure 5.2: OSAM-1 mission concept (courtesy of [12]).

At the Maxar Technologies facility in Palo Alto, California, mounted on the Maxar's decades-proven 1300 class spacecraft platform, the propulsion system was completed, including pipes and their large propellant and oxidant tanks, which are inserted into the central cylindrical structure; the thrusters, visible in Figure 5.3 in both the upper and lower decks of the spacecraft, and the two spherical silver tanks filled with monopropellant fuel intended for the satellite to be supplied. The latter are units from the Space Shuttles, on which they powered the Auxiliary Power Units (APUs) that provided hydraulic power during launch and re-entry.

As written in Maxar website [28], the propulsion system will allow the spacecraft to maneuver with six degrees of freedom (translation and rotation). That means OSAM-1 will have a unique three-axis translational maneuver capability, including the ability to execute "back away" maneuvers. This feature ensures safety during the approach and rendezvous with a non-cooperative satellite such as Landsat7.

According to Shoemaker[38], OSAM-1 has a planned wet mass at launch of around

6500 kg. Prior to shipping the spacecraft to NASA's Goddard Spaceflight Center in Greenbelt (Maryland), all OSAM-1 subsystems are now being tested.

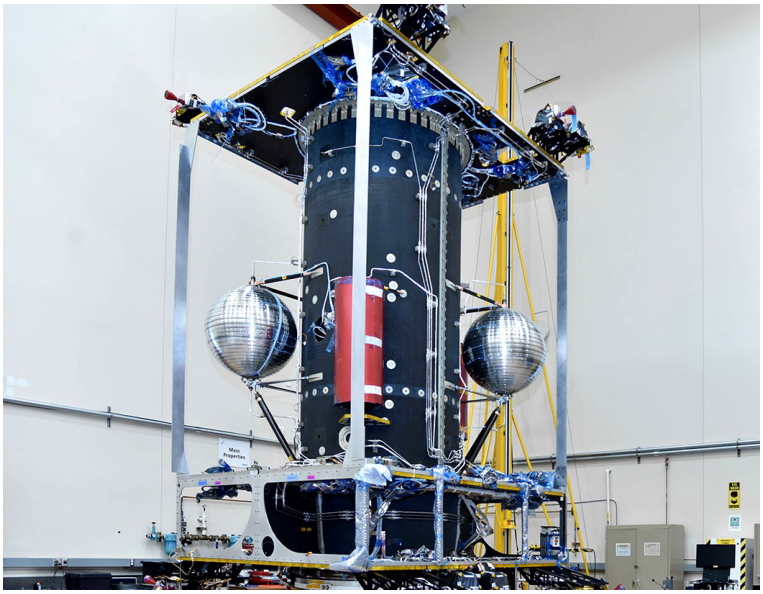


Figure 5.3: The basic structure of OSAM-1 being assembled (courtesy of [28]).

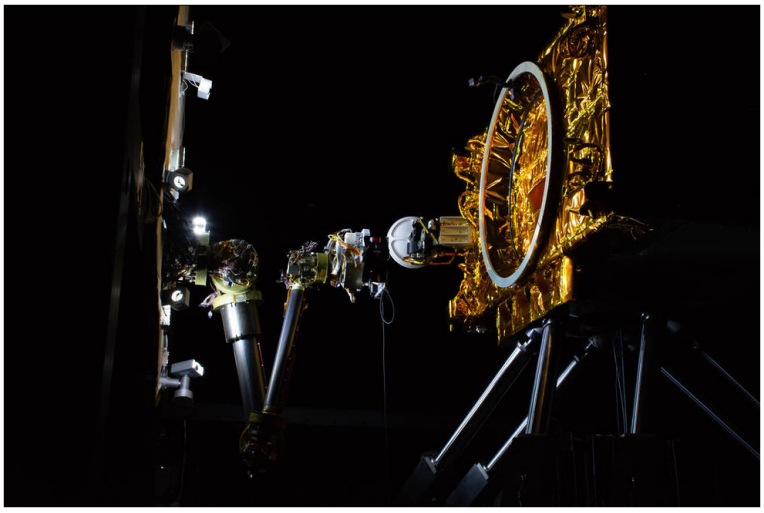
5.1.2 Servicing Technologies

Three robotic arms are planned for the OSAM-1 mission, two of which will be used to refuel Landsat7. These will grab a ring encircling the base of Landsat 7, then will guide Landsat 7 into so-called "berthing posts," where it will be secured to its rescue satellite, and finally, after cutting away the insulation blanket and opening the fill/drain valves, will transfer propellant to it (Figure 5.4, a).

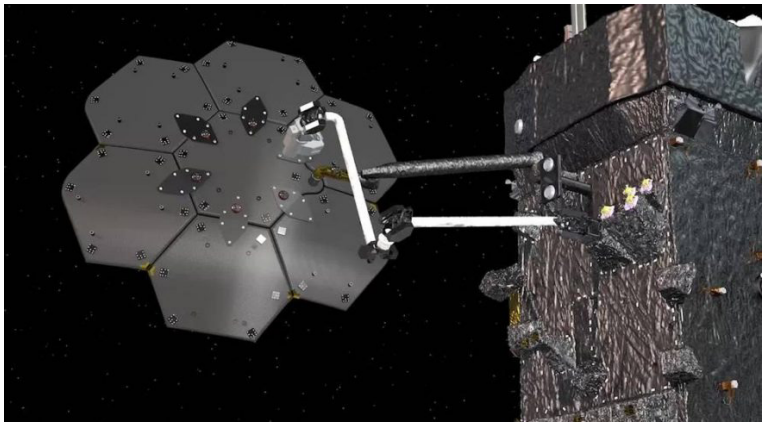
The other robotic arm, named "SPIDER" (*Space Infrastructure Dexterous Robot*), will be used to demonstrate in-space assembly and manufacturing for sustainable space exploration. "We're going to send a [radio] signal through that antenna to ground stations, and then we're going to disassemble the antenna and assemble it again to show it's a repeatable task" said Brent Robertson, the OSAM-1 Project Manager.
To accomplish all its tasks, OSAM 1 will be provided with essential tools and adequate techniques and technologies, as shown in [12].

- *Autonomous, real-time relative navigation system*

  Being a quite totally autonomous mission, OSAM-1 needs a relative navigation



(a) *Lights-out grapple testing of OSAM-1's Robotic Servicing Arm*



(b) *Concept of demonstration of in-space assembly (SPIDER)*

Figure 5.4: OSAM-1 issues (courtesy of [12]).

system composed by cameras, sensors, computers, algorithms and avionics that join forces to independently track the target satellite at different ranges, all in real time.

Once this system visually locks into the target, it can safely "guide and drive" the chase OSAM-1 satellite through precise rendezvous maneuvers around and then towards Landsat 7. On OSAM-1 will be mounted RAVEN, a NExIS (*Nasa's Exploration & In-Space Services*) project that fulfills the afore mentioned issue, already used on the ISS in 2016 to test key elements of the autonomous relative navigation system. Raven box, showed in Figure 5.5, contains a carefully curated and highly-performing system that includes visible, infrared and lidar sensors, a

high-speed processor and advanced algorithms.

- *Servicing avionics*

  In addition to ingesting and crunching sensor data, these elements control OSAM-1's rendezvous and robotic tasks.

- *Dexterous robotic arms*

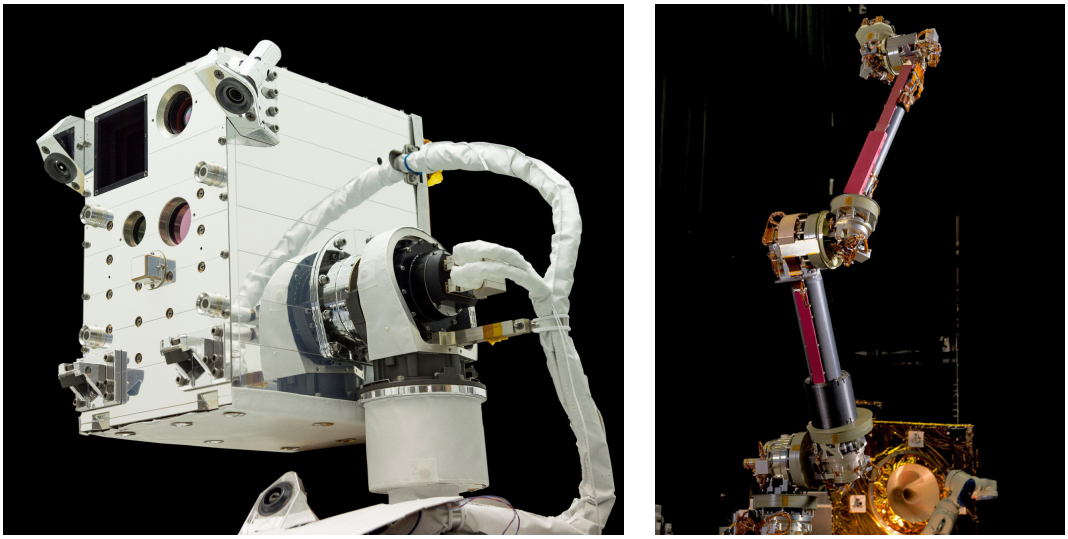
  These are Robotic Servicing arms, three seven-degree-of-freedom manoeuvrable robot arms to accomplish the two OSAM-1 issues, on-orbit servicing and on-orbit assembling.

- *Advanced tool drive and tools*

  Sophisticated, multifunction tools are manufactured to execute each servicing task, such as removing the caps and wires on the target satellite fuel valves.

- *Propellant transfer system*

  This system delivers measured amounts of fuel to the client at the right temperature, pressure and rate.



(a) *Raven module* **(b)** *Robotic Servicing Arm*

Figure 5.5: Servicing Technologies (courtesy of [12]).

5.2 Spacecraft Simulator

The programming software used in this thesis is Matlab, which, through specific toolbox, reveals to be an ideal environment for the study of a scenario such as rendezvous and docking or capture, either for the trajectory planning or for the control task.
In particular, the toolbox used are listed in the following.

- *Model Predictive Control Toolbox*, used to design and simulate linear or nonlinear model predictive controllers. It's possible to specify plant and disturbance models, horizons, constraints, and weights, to evaluate controller performance.

- *Global Optimization Toolbox*, that provides functions that search for global solutions to problems that contain multiple maxima or minima. In particular, genetic algorithms will be used to plan the observation trajectories.

- *Optimization Toolbox*, employed to solve linear, quadratic, conic, integer, and nonlinear optimization problems. According to [27], it provides functions for finding parameters that minimize or maximize objectives while satisfying constraints. This toolbox includes many solvers for linear or nonlinear programming, least squares or equations.

- *System identification toolbox* used to create linear and nonlinear dynamic system models from measured input-output data. The Extended Kalman filter tool belongs to this latter.

Since the study of the motion of a satellite – and the satellite itself – is complex and articulated, the simulator is organized through classes, which, describing only a single aspect of the general problem, allow greater clarity and streamline the global structure. Each class contains the functions used to set some parameters and update its variables during the simulation. All the classes converge in a single plant, which represents the general model for the study of the problem, including, of course, dynamics.
As regards the proximity operations within the inner keep-out sphere, the plants created and simultaneously developed will be two, one for the target and one for the chaser.

Since the target is not controlled, its created plant will be extremely simplified, actually incorporating only the propagation of the attitude dynamics, being the translational dynamics null due to the reference system taken. For these reasons, only the chase plant will be developed in the following.

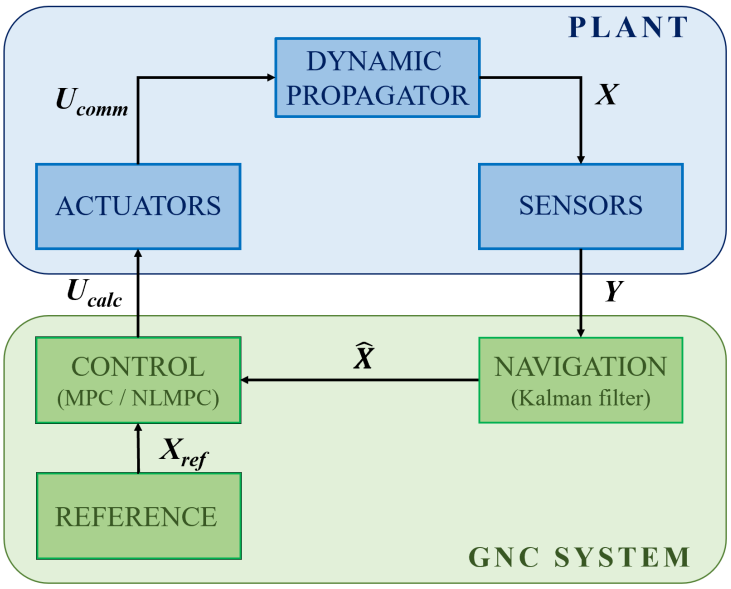


Figure 5.6: Chase spacecraft simulator scheme.

The main classes handle geometries and inertia, the propulsion system and the simulation parameters necessary to set the methods to solve differential equations of dynamics. Control and navigation do not affect the plant directly, but deal with the control input and the output measurement, as shown in Figure 5.6.

5.2.1 Dynamic propagator

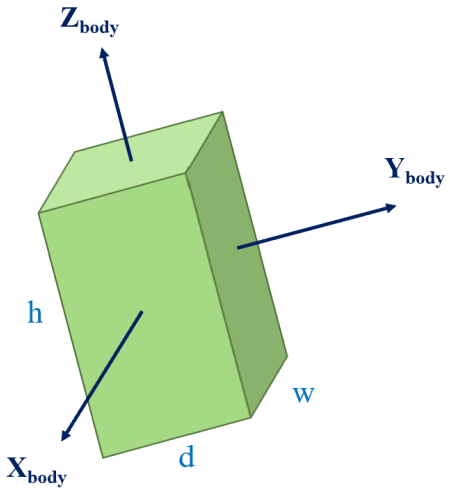
Geometries and Inertia

For simplicity, both the chaser, OSAM-1, and the target satellite, Landsat 7, will be represented as parallelepipeds of appropriate size and with uniform density. Intuitively, the body system considered will be the principal one, with principal axes perpendicular to body faces, as shown in Figures 5.7 and 5.8, where d stands for depth, w for width and h for height.

For a parallelepiped, given a principal axis i, the inertia around this axis can be written as follows:

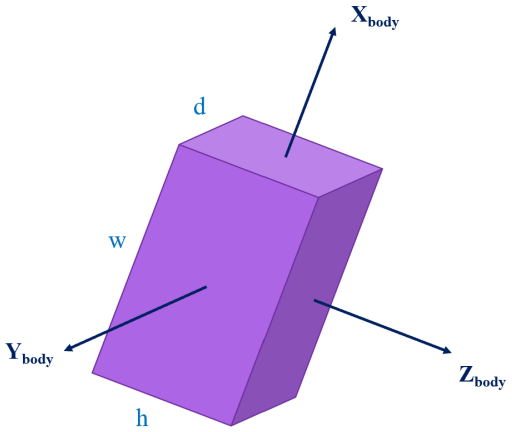
$$I_{ii} = \frac{1}{12} m(l_j^2 + l_k^2) \tag{5.1}$$

where l_j and l_k are the sides of the rectangular face perpendicular to axis i. The inertia tensor can be then written as a diagonal matrix: $\boldsymbol{\mathcal{I}} = diag[I_{xx}, I_{yy}, I_{zz}]$.



Inertial properties	
mass	2200 kg
width	2.5 m
depth	2.5 m
height	4.2 m
I_{xx}	4379.8 kgm^2
I_{yy}	4379.8 kgm^2
I_{zz}	2291.7 kgm^2

Figure 5.7: Target model.



Inertial properties	
mass	6500 kg
width	4.4 m
depth	2.7 m
height	2.7 m
I_{xx}	7897.5 kgm^2
I_{yy}	14435.4 kgm^2
I_{zz}	14435.4 kgm^2

Figure 5.8: Chaser model.

Equation of motion

The integration of the equations of dynamics constitutes the engine of the simulator. These equations, decoupled and divided into attitude equations and translational equations, are integrated through the Matlab ode45 function, a nonstiff differential equations solver that implements a Runge-Kutta method and which is designed to solve problems in the form:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(t, \mathbf{x}), \qquad \mathbf{x}(t_0) = \mathbf{x}_0 \tag{5.2}$$

where $\mathbf{x}$ is a vector of variables dependent on time t, the independent variable, and $\mathbf{f}$(t,$\mathbf{x}$) is a function of t and $\mathbf{x}$.

The equations of translational dynamics implied are the HCW equations 2.2, as already mentioned. The reference system used here, however, will be the LVLH orbital one presented in Chapter 2. Considering $\mathbf{x}$ as a vector containing position and velocity, $\mathbf{x} = [x, y, z, v_x, v_y, v_z]^T$, and adding the control action, they can be rewritten as follows:

$$\begin{cases} d\mathbf{x}(1:3) = \mathbf{x}(4:6); \\ d\mathbf{x}(4) = u_{comm,x} + 2n\mathbf{x}(6) + (F_{dist,x} - \dot{m}\mathbf{x}(4))/m; \\ d\mathbf{x}(5) = u_{comm,y} - n^2\mathbf{x}(2) + (F_{dist,y} - \dot{m}\mathbf{x}(5))/m; \\ d\mathbf{x}(6) = u_{comm,z} + 3n^2\mathbf{x}(3) - 2n\mathbf{x}(4) + (F_{dist,z} - \dot{m}\mathbf{x}(6))/m; \end{cases} \tag{5.3}$$

where n is the target mean motion, $\mathbf{F}_{dist}$ are possible disturbances (here set to zero) and $\dot{m}$ is the mass flow, set to zero in these equations, given the exiguous mass variation.

The equations of attitude dynamics are the Euler equations, written in the body reference, as in Eqn 2.12. The vector $\mathbf{x}$ now contains the angular velocity $\boldsymbol{\omega}_{BI}$ between inertial and body frames, written in body, and the quaternion, describing the body attitude referred to the orbital frame.

$$\mathbf{x} = [\omega_{BI,x}, \omega_{BI,y}, \omega_{BI,z}, q_1, q_2, q_3, q_4]^T$$

The first three components are known through the simple integration of Euler's equations:

$$d\mathbf{x}(1:3) = \boldsymbol{\alpha}_{comm} + \frac{\boldsymbol{\tau}_{dist} - \dot{\boldsymbol{\mathcal{I}}}\mathbf{x}(1:3)}{\boldsymbol{\mathcal{I}}} + \mathbf{x}(1:3) \times \boldsymbol{\mathcal{I}}\mathbf{x}(1:3) \tag{5.4}$$

where $\boldsymbol{\alpha}_{comm}$ is the control input vector and $\boldsymbol{\tau}_{dist}$ are possible disturbances (here set to zero). As explained in Chapter 2, body-inertial and body-orbital angular velocities are intimately related, so by assuming the first as known it's possible to derive the second one and to integrate the quaternion dynamics:

$$d\mathbf{x}(4:7) = \frac{1}{2}\,[\mathcal{Q}]\,\mathbf{x}(4:7) \tag{5.5}$$

where $[\mathcal{Q}]$ is the matrix shown in Eqn 2.17.

5.2.2 Actuator system: Thrusters

Due to the huge masses involved and the complicated maneuvers to be performed, especially in the last phase when the chaser has to co-move with the target, the choice of the propulsion system is not an easy task.

The chaser must be able to perform considerable thrusts, but without exceeding a certain acceleration value, and follow a well-defined thrust profile deriving from the control and navigation system. As reported by Leomanni[20], a range of reasonable values of maximum acceleration for the problem at hand is from $10^{-3}\ m/s^2$ to $10^{-1}\ m/s^2$, on a rough estimate.

Although during the proximity approach the attitude is a function of the position, the translational and rotational control actions remain distinct: the thrusters, therefore, have also been chosen separately, based on the thrust profile to chase the studied trajectories. In particular, for the translational part, twelve 200 N thrusters were chosen, two for each of the six faces normal of the parallelepiped. The main features, given by [16], are shown in Figure 5.9. As regards the attitude control, being less thrust demanding, four 10 N thrusters for each principal axis are used. The characteristics for these engines are shown in Figure 5.10, still given by [16].

Properties	
Thrust Nominal	$216N \pm 10\ N$
Specific Impulse	$> 270\ s$
Flow rate nominal	$78\ g/s$
Minimum on time	$28\ ms$
Mass	$1.9\ kg$
Throat Diameter	$12\ mm$
Nozzle End Diameter	$95\ mm$
Fuel	MMH/UDMH
Oxidizer	MON-3/N2O4

Figure 5.9: 200 N bi-propellant thrusters, courtesy of [16].

Properties	
Thrust Nominal	$10\ N$
Specific Impulse	$292\ s$
Flow rate nominal	$3.5\ g/s$
*Minimum on time**	$10\ ms$
Mass	$0.65\ kg$
Throat Diameter	$2.85\ mm$
Nozzle End Diameter	$35\ mm$
Fuel	MMH
Oxidizer	MON-1/MON-3/N2O4

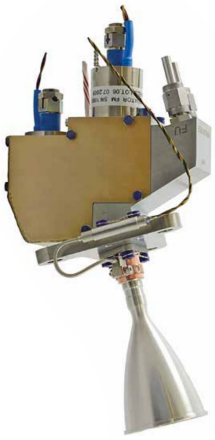


Figure 5.10: 10 N bi-propellant thrusters, courtesy of [16] (assumed*).

Both engines work in a Pulse Width Modulation mode: this means that under a certain value, that is correlated to the simulation time-step and the minimum on time value listed above, a control action cannot be given (Figure 5.11):

$$F^B_{calc} \geq F_{min} = \frac{F_{max}\ \tau}{dt_{simul}}$$

where B stands for Body and τ is the minimum on time. The upper limit, on the other hand, is implicitly imposed by the maximum thrust value that thrusters can provide.

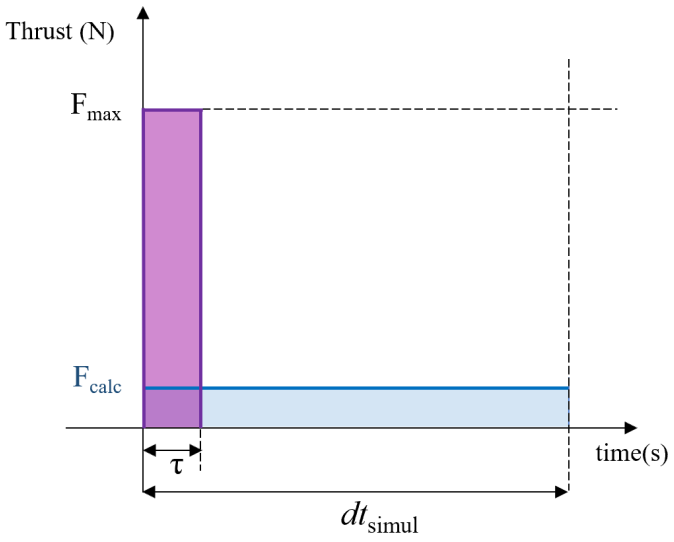


Figure 5.11: Pulse Width Modulation concept (in blue the calculated force, in purple the equivalent commanded force).

Anyway, the translational control actions $u_{comm} = [u_x, u_y, u_z]^T$ showed in Eqn 5.3 are provided in the LVLH orbiting frame: upper and lower limits, reasonably defined in the body frame, cannot be imposed directly on these values to find the commanded actions u_{comm} and then the forces F_{comm}. Given the peculiarity of predictive controllers, it is possible to set a limit on the maximum acceleration calculated for each direction of the orbiting reference:

$$\boldsymbol{u}_{max} = \frac{\boldsymbol{F}_{max}}{m\sqrt{3}} \tag{5.6}$$

Dividing $\boldsymbol{F}_{max}$ by $\sqrt{3}$ is a precaution that solves the problem: this factor ensures that, even in the case of an unfavourable relative orientation between the body and orbital references, the control action can be faithfully reproduced in the body reference, without any simplifications. This is decisive in the near-field RVD study, where the attitude is not provided; in proximity operations, on the other hand, these control actions u_{calc}, multiplied by mass, are transformed into forces and, through the rotation

matrix from orbital to current body R_O^B, these forces are provided in the body reference, where upper and lower limits can finally be imposed to find the commanded forces:

$$\begin{aligned} |F_{calc}^B| &\leq F_{max} \\ |F_{calc}^B| &\geq \frac{\tau F_{max}}{dt_{simul}} \end{aligned}$$

Also regarding the attitude control actions, a maximum limit is set:

$$\boldsymbol{\alpha}_{max} = \frac{\boldsymbol{\tau}_{max}}{\boldsymbol{\mathcal{I}}}$$

Here dividing by $\sqrt{3}$ is useless: torques are given in the body reference, so it is not necessary.

To simulate the non-repeatability of the thrusters and therefore the errors at each impulse, a random error was added to the commanded thrust value, with a standard deviation given on a percentage basis:

$$F_{act} = (1 + \sigma_{thrust} a) F_{comm}$$

with $\sigma_{thrust} = 1\%$ and a a random value.

To calculate the propellant mass used for translational motion two methods are available:

- Calculation through forces:

$$m_{prop} = \frac{\dot{m}\ dt_{Simul}}{F_{max}} \sum_{k=0}^{N} |F_{x,k}| + |F_{y,k}| + |F_{z,k}|$$

  where N is the total number of steps and $\dot{m}$ the constant mass flow;

- Calculation through Tsiolkovsky equation:

$$m_{prop} = m_{in} \left(1 - e^{-\Delta V_{tot}/(g\ I_{sp})}\right)$$

  where I_{sp} is the Specific Impulse, already given for both thrusters.

The latter equation takes into consideration the loss of mass, neglected instead in the simulated dynamics due to its exiguity, so results will be found through the first equation, even if there is a very slight difference between the two results.

To calculate the propellant mass used for the attitude control, the following equation has been employed:

$$m_{prop} = \frac{\dot{m}\ dt_{Simul}}{2\ b\ F_{max}^{att}} \sum_{k=0}^{N} |\tau_{x,k}| + |\tau_{y,k}| + |\tau_{z,k}|$$

where b is the arm (here set to $1m$) from the center of the satellite to the position of a generic attitude thruster, and all τ are torques referred to the chase body frame.

5.2.3 Control system

The purpose of the control system is to generate a command given the information coming from the navigation system and considering the reference trajectory to be followed. As already announced, Model Predictive controllers, whose working principle is shown in the previous chapter, are implemented in the control system.

Model Predictive Control

As mentioned at the beginning of this chapter, a simple MPC will be used where the model is linear, that is in the near field RVD phase, having to be handled only the translational part with the linear HCW equations.
Concerning its implementation, Matlab, through the Model Predictive Control toolbox, provides a series of commands to set up and manage an MPC. Using the predicted plant outputs, the controller solves a quadratic programming optimization problem (or QP problem) to determine the control action. The main steps (Figure 5.12) are listed in the following.

1. *Set the principal parameters*

   Control horizon, prediction horizon and duration of control action are here set in order to achieve the best performance possible. These are chosen taking into account all the considerations done in the previous chapter on lengthening or shortening the horizons.

2. *State space model creation*

   The HCW equations are re-proposed in the form of Eqn 4.1, with the matrices

A, B, C already defined, and then a state space model is created. Using the command of Matlab `c2d`, through the zero-order hold method, the state space model is then discretized.

3. *Controllability and observability tests*
   Having state-space systems, controllability and observability tests are recommended. To do this, Matlab gives two commands:

   - `obsv`, that computes the observability matrix Ψ showed in Eqn 4.15 for state-space systems; the model is observable if this matrix has full rank;
   - `ctrb`, that computes the controllability matrix of state-space systems; as the previous one, the system is controllable if the matrix has full rank.

4. *Weights definition*
   After providing the model with the number of state variables and the number of output measured variables through the function `setmpcsignals`, cost function matrices must be defined. To control the evolution of the latter and therefore the tracking performance, it is necessary to set the correct weights on the controlled variables and on the output variables.

5. *MPC model creation and constraints definition*
   The command `mpc` creates a model predictive controller object based on the discrete-time prediction model plant, enclosing all the values previously set.
   After the model is created, one can impose constraints on manipulated variables (i.e. control actions) and output variables, specifying the degree of hardness in the constraint imposition.

Once the controller is set, the Matlab command acting in the dynamic propagation loop is `mpcmove`, which requests as input the object created with the `mpc` command, the current state, the measured state (if one wants to employ the Kalman filter model created by default with the `mpc` command) and the reference used in the prediction horizon, and returns the control action u_{calc}.

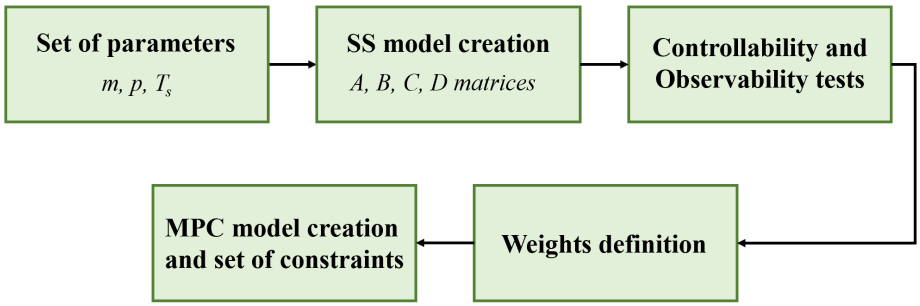


Figure 5.12: MPC definition steps.

Nonlinear MPC

When the problem ceases to be linear, that is for the use of attitude equations (Euler) or the management of nonlinear constraints, the tool to be used is the nonlinear MPC. Also in this case, Matlab provides a set of commands and tools, always belonging to the Model Predictive Control toolbox, to manage this controller.
As reported in the Matlab webpage[25], the key differences with a simple MPC are:

- The prediction model can be nonlinear and include time-varying parameters;
- The equality and inequality constraints can be nonlinear;
- The scalar cost function to be minimized can be a nonquadratic function of the decision variables.

Being this problem no longer totally quadratic, alternate optimization techniques, or other solvers, are needed. In this thesis the nonlinear MPC controller is set to solve a nonlinear programming problem using the `fmincon` function, in which the Sequential Quadratic Programming algorithm is implemented. This latter requires the Optimization Toolbox of Matlab.
The main steps required to implement the controller are presented below (Figure 5.13).

1. *Set the principal parameters*

   As for MPC, control horizon, prediction horizon and duration of control action are here set in order to achieve the best performance possible. All the considerations

made in the linear case, for these parameters, are here equally applied.

Through the command `nlmpc`, the nonlinear MPC controller is created. Next steps will enter this object as properties.

2. *State and output functions*

   Since the matrices to express the equations of the model in a state-space form can no longer be found, a state function must be directly provided. This function, describing the model, if provided in a continuous-time form, is automatically discretized using the implicit trapezoidal rule.

   The equations implemented in the function are the same as those of the dynamic propagator, but this time gathered: the overall state vector, therefore, will contain 13 variables:

   $$\mathbf{x} = [x, y, z, v_x, v_y, v_z, \omega_{BI,x}, \omega_{BI,y}, \omega_{BI,z}, q_1, q_2, q_3, q_4]^T$$

   Jacobians used in the linearization are directly calculated inside the tool and not provided externally.

   In addition to the state function, an output function must be set, that is, it must be specified which part of the state is then supplied at the output. Having to provide the reference to the nlmpc object in all the variables of the vector output but being the reference trajectory provided only in position, only $[x, y, z]$ will be provided at the output. To manage the measurement and prediction part there will be an external Extended Kalman filter.

3. *Cost function*

   Even the cost function, no longer being quadratic but in a general user-provided form, is to be specified, customizing it in order to accomplish all the requirements. Alongside this, anyway, a standard quadratic MPC cost function can be preserved: the final optimization will result in a combination between the custom optimization function and this latter, managed through weight tuning. The cost function used will be presented in the chapter of inner proximity operations study.

4. *Constraint function*

Regarding constraints too, a special function must be specified in the prediction horizon, in the form of equality or inequality. As for the latter, an inequality constraint is satisfied when the corresponding output is less than or equal to 0. Again, the function used will be presented in the final chapter.

5. *Nlmpc object validation*

   When the nlmpc model is completely defined, a control on what has been set is needed. To do this, a specific command, `validateFcns`, is given. It requires, as input, the `nlmpc` object created, all the parameters used and the initial state to test the entire configuration of the nonlinear MPC controller for potential problems such as whether information is missing, whether input and output arguments of any user custom functions are incompatible with object settings.

As for the linear MPC, there is a Matlab command acting in the dynamic propagation loop, `nlmpcmove`, which requests as input the object created with the `nlmpc` command, the current state, the measured state (or a part of it) and the reference used in the prediction horizon and returns the control action u_{calc}.

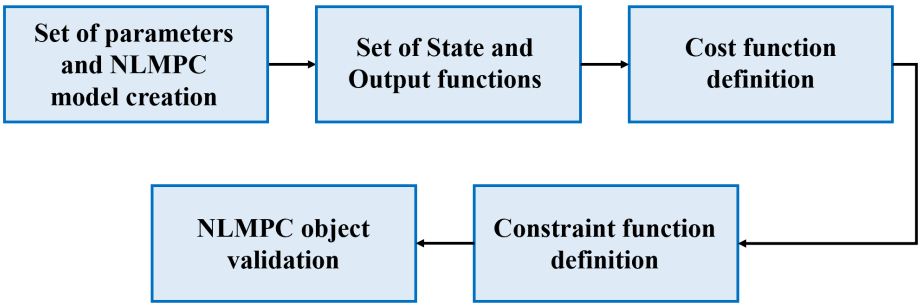


Figure 5.13: NLMPC definition steps.

5.2.4 Navigation system

To better estimate the state of the system by exploiting the measurements coming from the sensors and the knowledge of the model, a filter is required.

Kalman filter

As regards the linear translational model, a Kalman filter is coupled with the MPC. No particular functions were exploited to implement the KF, just used the equations in Section 4.2.3. It's recommended to refer to the afore mentioned section to have a complete view of its working principle.

Extended Kalman filter

As regards instead the nonlinear model, an Extended Kalman filter is needed, pairing with the nonlinear MPC, to estimate the states.
Through the command `extendedKalmanFilter` an EKF object is created. A state transition function and a measurement function must be provided as input: the first one is the same state function of the nonlinear MPC (that is the union of HCW and Euler equations), but discretized with a trapezoidal rule, the second one specifies which state variables are to be measured. In this case, only the position and the attitude (quaternions or Euler angles) will be measured.
Once the object is created and the initial state and measure are defined, it's possible to specify the process and the measurement noise covariances and create two matrices, specified in Subsection 4.3.1 as R and Q.

The way the filter acts has already been presented in Subsection 4.3.1: to implement the steps reported, Matlab[23] provides the commands `correct`, to correct the state and state estimation error covariance at time step k using measured data at time step k, and `predict`, to predict the state and state estimation error covariance at time the next time step. A further command, `residual`, return the difference between the actual and predicted measurements, then giving the possibility to know how the filter is performing.

5.2.5 Sensors

Since not any real measurements are available, to simulate the presence and action of a sensor system some possible dispersions of values have been defined.
Given that the satellite will mount visible, infrared and LIDAR sensors, it is reasonable

to think that a measure of the relative position $[x, y, z]$ between the two satellites can be provided. As for the attitude of the chaser, where present, to be measured will be the angles of Euler.
As explained, these measurements will be used by Kalman filters to correct predictions previously made by relying on the model. Although the handling of errors is the same, they are given differently in the two areas studied.

- *Near-field RVD*: the standard deviations of measurement noise, regarding relative position, is the same in the three orbital frame directions.
- *Inner Proximity Operations*: the standard deviations of the relative position and attitude measurement noises vary linearly during the approach, as shown in Figure 5.14.

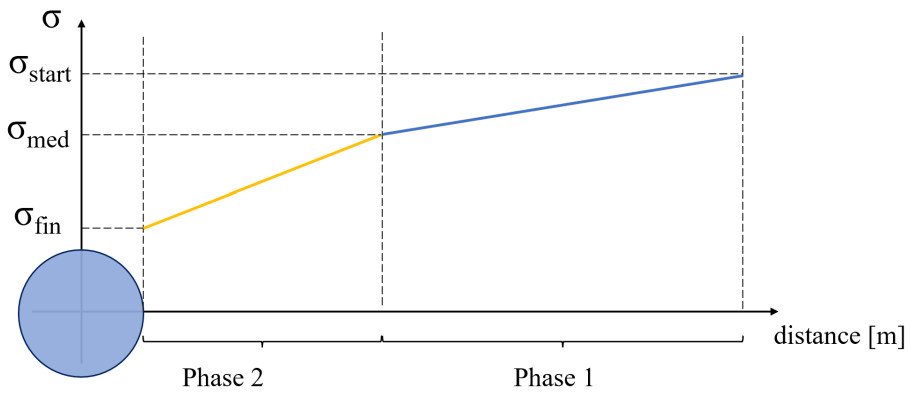


Figure 5.14: Linear variation of standard deviations.

To simulate as realistically as possible the results of an observation motion, however, standard deviations on a percentage basis were added on the estimated angular velocity and inertia, gradually reduced as the chaser approaches the target. These last errors, however, only affect the reference along the prediction horizon, created by the controller at each control step to compute the action u_{calc}, hence they constitute an error on the prediction of the motion rather than an error on the dynamics of the chaser.
Due to this reason, they are not managed by the filter, and this is the reason

why, with refer to Figure 5.6, navigation and reference blocks are not linked.

5.2.6 Reference

The reference is also managed differently in the two phases. As regards the first one, i.e. the near field rendezvous, the trajectory is calculated at the head of simulation, without changes in itinere. As regards the second one, the inner proximity rendezvous, as explained in the previous section, the reference is created at each control step based on the current estimates of the target inertia and angular velocity.
Further details will be provided in the following chapters, where the two phases will be analysed separately.

Chapter 6

Near-field Rendezvous Study

As already mentioned, the study of the rendezvous maneuver considered in this thesis begins from an initial distance of a few kilometers, thus taking for granted the method of approach for large initial distances. As an example, however, a simple Hohmann maneuver and subsequent propagation on a co-elliptical orbit will be shown. Relative motion, for these two propagations, is realized as a simple difference of absolute poses, so the HCW model is not used.

Concerning the observation maneuvers, which represent the core of this chapter, the HCW equations presented in Chapter 2 were used, being the considered distances between the chaser and the target of a few hundred meters.
The creation of the reference trajectories and therefore the implementation of the observation motions already presented was performed using genetic algorithms, simplifying the maneuvers, where present, in impulsive maneuvers, and conceiving the trajectories between one delta-v and the next one as simple propagations dictated by the HCW equations of relative motion.

Genetic algorithms are heuristic algorithms, used to attempt to solve optimization problems by exploiting the analogy with Darwinian principles of natural selection and biological evolution. In summary, these algorithms allow to evaluate different starting solutions (as if they were different biological individuals), which, recombining them (similarly to biological reproduction) and introducing elements of disorder (similarly to random genetic mutations), produce new solutions (new individuals) that are evaluated

by choosing the best (environmental selection) in an attempt to converge towards "excellent" solutions.

In Matlab, their implementation is possible through the `ga` function, which requires in input the so-called 'fitness function' to be minimized and the constraints to be respected during the search for the minimum, expressed in the form of equalities or inequalities. These two specifications will be provided for each observation motion studied.

The ellipsoid-shaped macro regions identified and used to analyse the approach have the dimensions illustrated in Figure 6.1.

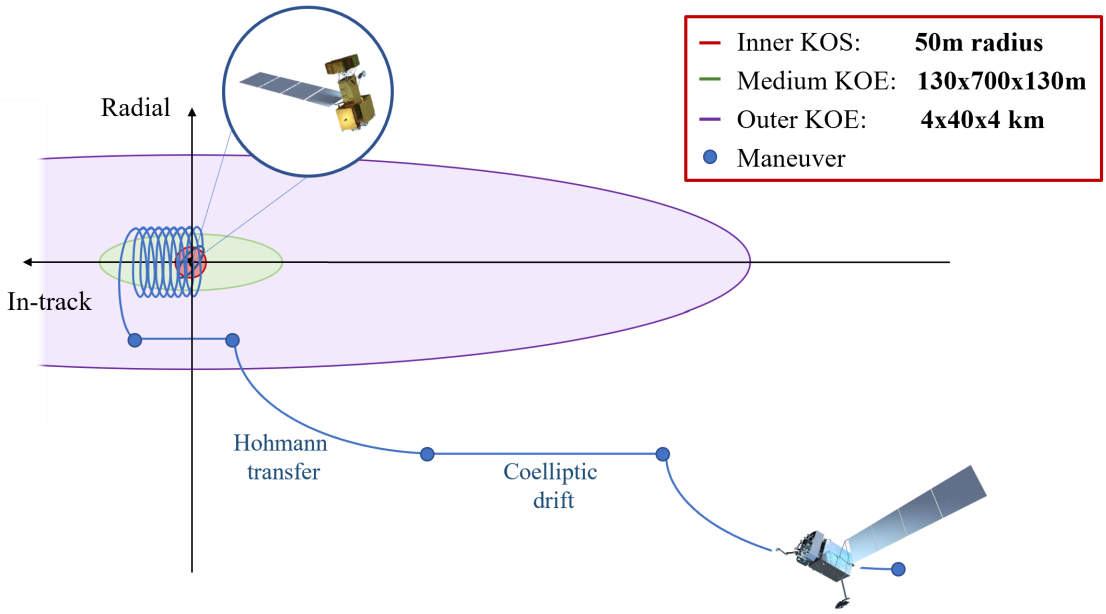
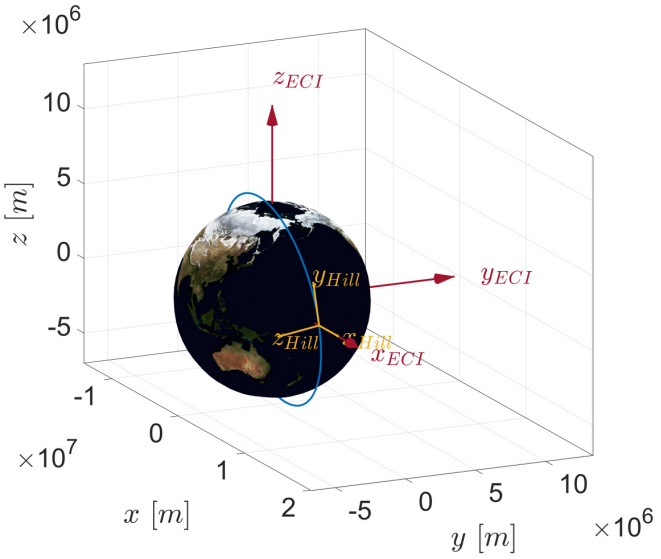


Figure 6.1: Keep-Out Ellipsoids (KOE).

After showing, as promised, a first example of a Hohmann maneuver followed by a co-elliptic propagation, the observation motions will be presented. These are thought as a first phase (*"Phase A"*) of natural approach through a Walking Safety Ellipse, straddling the middle ellipsoid, and a second phase of observation (*"Phase B"*) – through another WSE, an ellipse or a structure of waypoints – within this. As reported in the previous chapter, only translational motion will be analysed.

6.1 Hohmann and Phasing approach

The orbital parameters of both chase and target satellites considered in the studied scenario are shown in Figure 6.2.



	Target	**Chaser**
h	$697km$	$694km$
e	0	0
i	98°	98°
Ω	0°	0°
ω	0°	0°
θ	0.1°	0°
T	$98m\ 38.84s$	$98m\ 42.61s$

Figure 6.2: Orbital Parameters of the target and the chaser.

At the initial time, an impulsive delta-v is performed to enter a Hohmann trajectory and, after a propagation for half a target period, another impulsive delta-v is carried out to enter the co-elliptical orbit with respect to the target one, therefore circular, with a radius equal to the altitude of the apogee of the Hohmann trajectory.
The propagation in the co-elliptical orbit is a fraction of the target orbital period, chosen to approach along the in-track line and get closer to the target for subsequent maneuvers.
The main parameters are listed in Table 6.1 and referred to Figure 6.3.
It is possible to transform the trajectory into a relative trajectory, computed as a simple difference of absolute poses regarding position and dictated by the relative velocity theorem with regard to velocity. This trajectory, when expressed in the Hill's reference, appears as shown in Figure 6.4.

Perigee altitude	$694km$
Apogee altitude	$696.1km$
T_{HOH}	$0.5\ T_{targ}$
T_{COELL}	$0.3\ T_{targ}$
ΔV_1	$0.5572m/s$
ΔV_2	$0.5572m/s$
$\mathbf{X}_{1,Hill}$	$[-3010.77, -12343.20, 0]\ m$
$\mathbf{X}_{2,Hill}$	$[-900.71, -3159.51, 0]\ m$
$\mathbf{X}_{3,Hill}$	$[-900.03, -615.23, 0]\ m$

Table 6.1: Main Parameters of the Hohmann maneuver and the co-elliptic propagation.

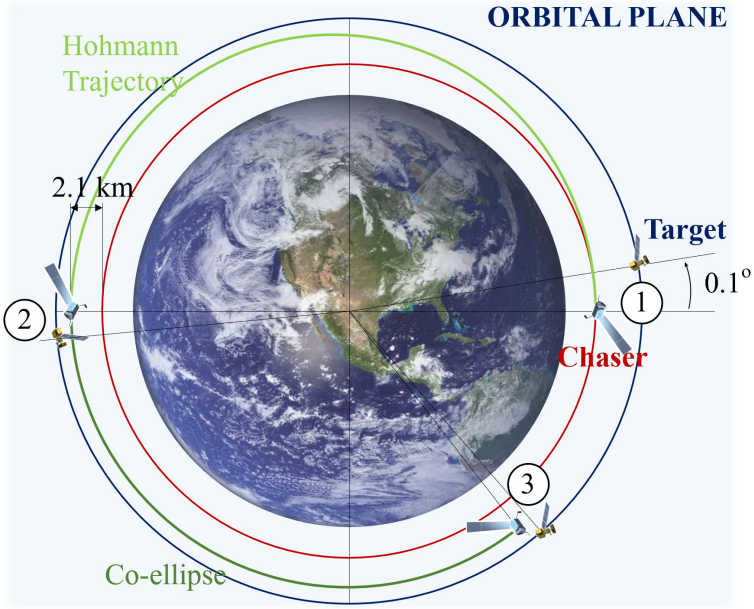


Figure 6.3: Hohmann transfer and Co-ellipse concept.

6.2 Phase A: Walking Safety Ellipse approach

The HCW model was used from the last point, named as $\mathbf{X}_{3,Hill}$ in Table 6.1, once the propagation on the phasing orbit is done. Being the chaser still hundreds of meters far from the target, a good solution, shown here, is to approach through a natural orbit,

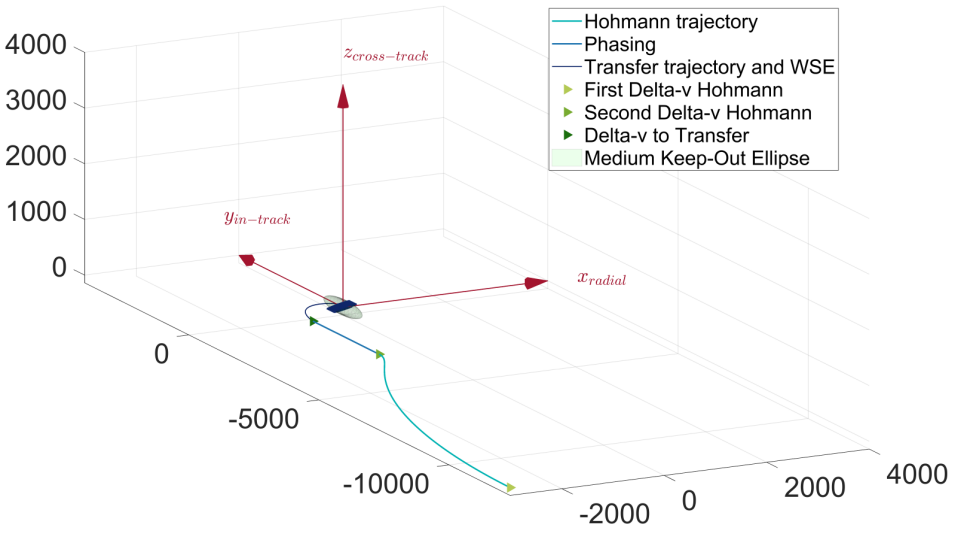


Figure 6.4: Hohmann and Phasing approach in Hill frame (in meters).

namely a Walking Safety Ellipse, already presented in Chapter 3.
As already mentioned at the beginning of the chapter, genetic algorithms were used to find the optimal impulsive delta-v and transfer trajectory, together with the WSE geometric parameters.

6.2.1 Genetic Algorithms for WSE

What must be provided to the algorithm is a function to be minimized, namely the fitness function, and some constraints (thinking about the construction of a WSE) that the trajectory must respect.

Limits

The parameters needed to define a WSE are the following: x_{max}, y_c, Δy, z_{max}, ψ_0, ϕ_0, with the meanings already provided in Chapter 3. In addition to these geometric parameters, it is possible to add a temporal one, Δt_{Transf}, or the time the chaser takes to go from the starting point to the entry point of the WSE.
Remembering that the semi axes of the middle ellipsoid are $[130, 700, 130]$ m (Figure 6.1), the limits for these parameters are presented in Table 6.2.

x_{max}	$[140 \div 160]\ m$
$\dot{y_c}$	$[-0.005 \div 0.005]\ m/s$
Δy	$[-200 \div 200]\ m$
z_{max}	$[140 \div 160]\ m$
ψ_0	$[0 \div 2\pi]$
ϕ_0	$[0 \div 2\pi]$
Δt_{Transf}	$[0 \div T_{targ}]$

Table 6.2: First WSE ranges for Genetic Algorithm.

Fitness function

This is the function to be minimized and can be set to direct the construction of the trajectory towards the desired one by adding additional costs to the function:

$$FitnessValue = \Delta V_{tot} + C_1 + C_2$$

where the three contributions have the following meaning:

- ΔV_{tot}: sum of the two delta-v, the first one to take the transfer trajectory, the second one to enter the WSE;
- C_1: cost on discord of the signs of Δy and $\dot{y_c}$, to ensure the approach to the target;
- C_2: cost on the minimum distance from the target, d_{min}, during the transfer to the WSE, imposed in the following way:

  $$C_2 = e^{\gamma R_{KOS} - d_{min}} - 1$$

  where R_{KOS} is the keep-out sphere radius and γ is a safety factor equal to 1.2.

6.2.2 Example of WSE

Since genetic algorithms are not repeatable, the solution found, even respecting the limits imposed, can change running the code several times, so what follows is only an

example of a possible WSE. The afore cited geometric parameters are listed in Table 6.3, while the main results are presented in Table 6.4. The propagation on the WSE is done in order to cover 1.5 times the initial in-track distance. A 3D plot of the spiral-shaped trajectory is given in Figure 6.5.

x_{max}	160 m
$\dot{y_c}$	-0.0046 m/s
Δy	200 m
z_{max}	140 m
ψ_0	1.0566
ϕ_0	5.9172
Δt_{Transf}	0.411 T_{targ}

Table 6.3: WSE parameters.

$\mathbf{x}_{to\ transf1}$	$[-900.03, -615.23, 0]$ m
$\mathbf{x}_{WSE,inserction}$	$[142.18, 359.61, -50.10]$ m
$\Delta V_{to\ transf1}$	0.4338 m/s
$\Delta V_{WSE,inserction}$	0.2575 m/s
Δt_{WSE}	$18h\ 11m\ 30.20s$

Table 6.4: Main parameters of the trajectory to WSE.

6.3 Phase B: Observation Phase

Once the distance has been reduced considerably, the second phase, that is the observation motion, can be initialized. The observation, this time, is performed inside the middle ellipsoid, while maintaining an appropriate safety distance from the target.
In the following, three strategies, all suitable and possible, will be presented: firstly an observation from a safety ellipse, then a walking safety ellipse and finally an observation performed from waypoints placed on an octahedron, as presented in Chapter 3. As

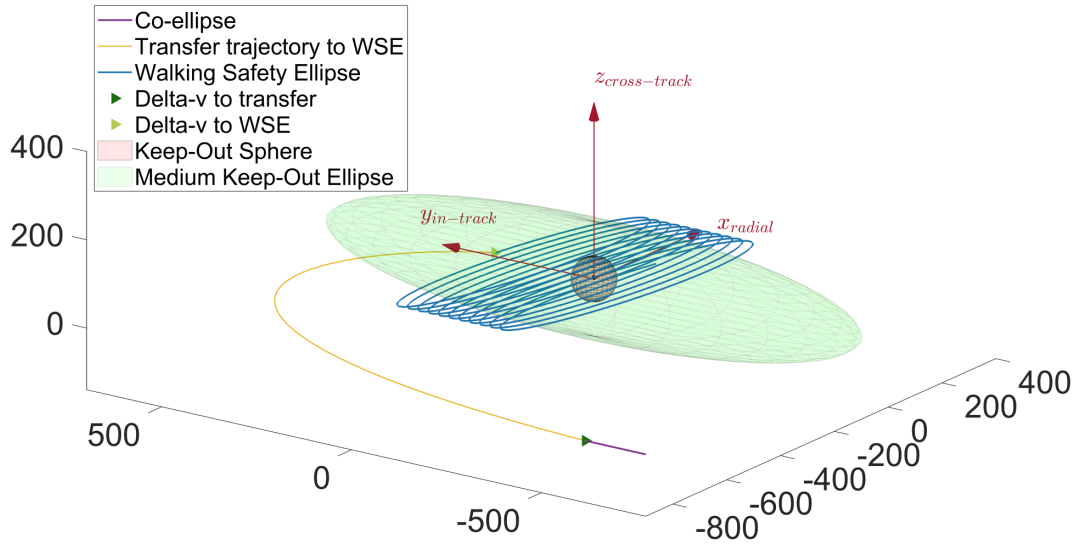


Figure 6.5: WSE Example in Hill frame (in meters).

done for the first WSE, the main parameters and costs used in the genetic algorithm will be shown for each alternative. The keep-out distance considered is 60 m, that is the inner KOS radius multiplied by a safety factor equal to 1.2.

6.3.1 Safety Ellipse

This is a natural fly-around motion which, ideally, does not require any intervention of the actuator system to be performed. It is therefore a very simple motion, suitable for observing or waiting, without any drift.

Genetic Algorithms for Safety Ellipse

The ellipse, projected on the radial-in-track plane, has a proportion of 1:2: any eventual condition can be placed only on one of the two variables, being interdependent.
The parameters to describe the ellipse are the same as those already shown for the WSE, with y_c, however, placed at zero.

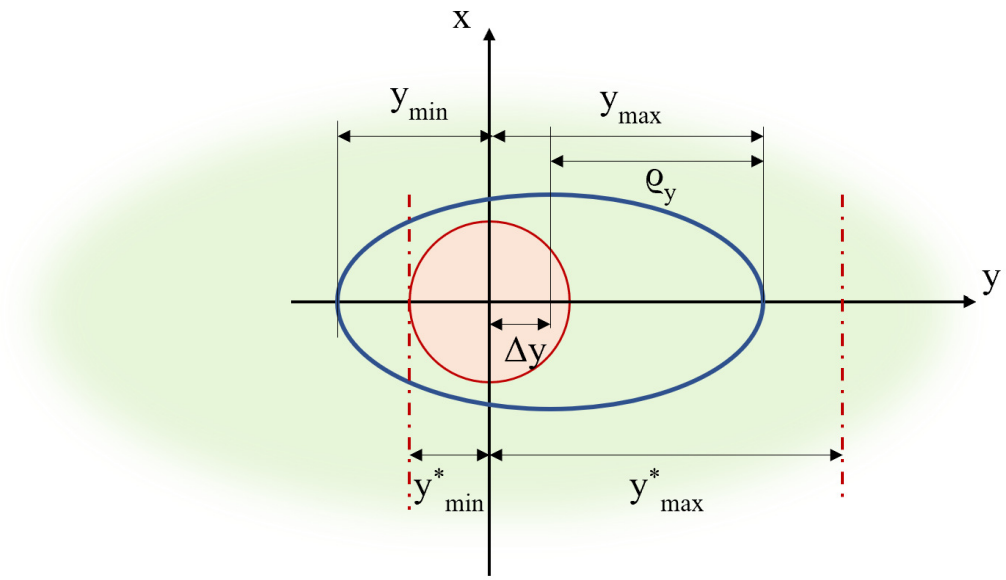


Figure 6.6: Ellipse limits.

Referring to Figure 6.6, the following conditions must be met:

$$\begin{cases} \rho_y \in [y^*_{min}, y^*_{max}] \\ \Delta y \in [-\Delta y^*, \Delta y^*] \end{cases}$$

where $\Delta y^* = \dfrac{y^*_{max} - y^*_{min}}{2}$. Choosing $y^*_{min} = 60\ m$ and $y^*_{max} = 120\ m$, ranges can be listed in Table 6.5:

x_{max}	$[30 \div 60]\ m$
Δy	$[-90 \div 90]\ m$
z_{max}	$[0 \div 80]\ m$
ψ_0	$[0 \div 2\pi]$
ϕ_0	$[0 \div 2\pi]$
Δt_{Transf}	$[0 \div 0.5\ T_{targ}]$

Table 6.5: Safety Ellipse ranges.

As regards the fitness function, the form is the same of the previous one but with more

contributes:

$$FitnessValue = \Delta V_{tot} + E_1 + E_2 + E_3 + E_4$$

where the additive terms are explained in the following:

- ΔV_{tot}: sum of the two delta-v, the first one to take the second transfer trajectory, the second one to enter the Safety Ellipse;

- E_1, E_2: costs to assure the conditions below

$$\begin{cases} \rho_y - |\Delta y| < y^*_{min} \\ \rho_y + |\Delta y| > y^*_{max} \end{cases}$$

  Writing $k_1 = \rho_y - (|\Delta y| + y^*_{min})$ and $k_2 = y^*_{max} - |\Delta y| - \rho_y$, the costs E_1 and E_2 can be written as follows:

$$\begin{cases} E_1 = 10 + e^{-k_1} - 1 \\ E_2 = 10 + e^{-k_2} - 1 \end{cases}$$

- E_3: cost on minimum safe relative trajectory distance, imposed to be greater than R_{KOS}:

$$b = \sqrt{\frac{x^2_{max} + z^2_{max} - \sqrt{x^4_{max} + z^4_{max} + 2x^2_{max} z^2_{max} \cos(2\psi_0 - 2\phi_0)}}{2}} \geq 1.2\ R_{KOS}$$

  Writing $k_3 = b - 1.2\ R_{KOS}$, E_3 can be written as follows:

$$E_3 = 10 + e^{-k_3} - 1$$

- E_4: cost on the minimum distance d_{min} of the transfer trajectory from the target, calculated in the same way of C_2 for WSE.

Example of Safety Ellipse

An example of Safety Ellipse is given, showing the results starting after the WSE is performed. The geometric parameters are listed in Table 6.6, while the main results are presented in Table 6.7. The propagation on the Safety Ellipse is done the time it takes to completely and accurately observe the target. 3D plots are given in Figures 6.7 and 6.8.

x_{max}	$60\ m$
Δy	$-0.0545\ m$
z_{max}	$80\ m$
ψ_0	6.0574
ϕ_0	4.4918
Δt_{Transf}	$0.423\ T_{targ}$

Table 6.6: Safety Ellipse parameters.

$\mathbf{x}_{to\ transf2}$	$[161.03, -49.34, -0.515]\ m$
$\mathbf{x}_{SE,inserction}$	$[-49.67, -67.26, 44.48]\ m$
$\Delta V_{to\ transf2}$	$0.1390\ m/s$
$\Delta V_{SE,inserction}$	$0.0976\ m/s$
Δt_{SE}	$kT_{targ},\ k \in \mathbb{R}$

Table 6.7: Main parameters of the trajectory to Safety Ellipse.

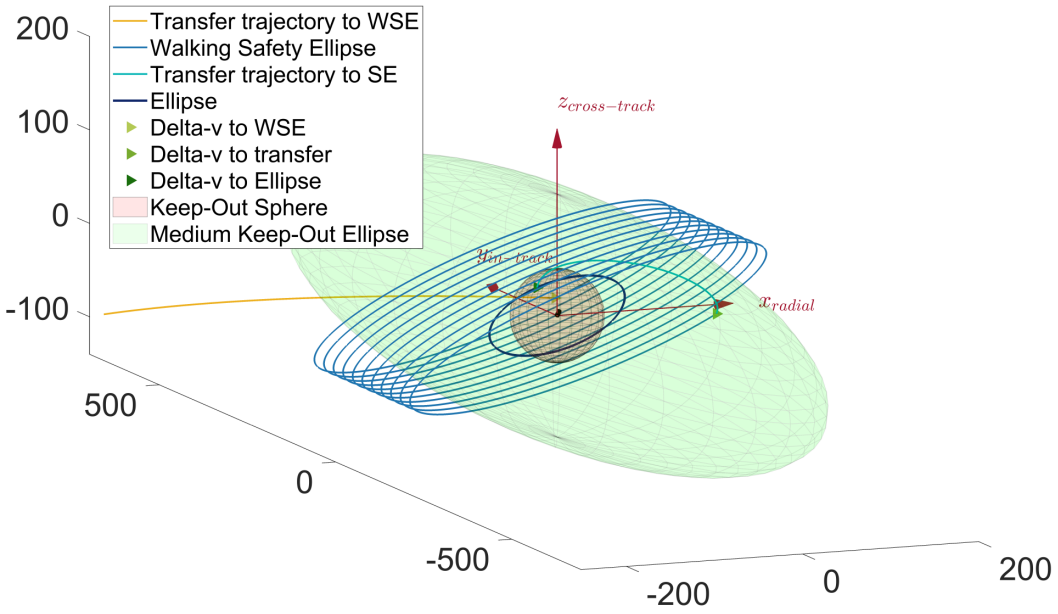


Figure 6.7: Safety Ellipse Example in Hill frame (in meters).

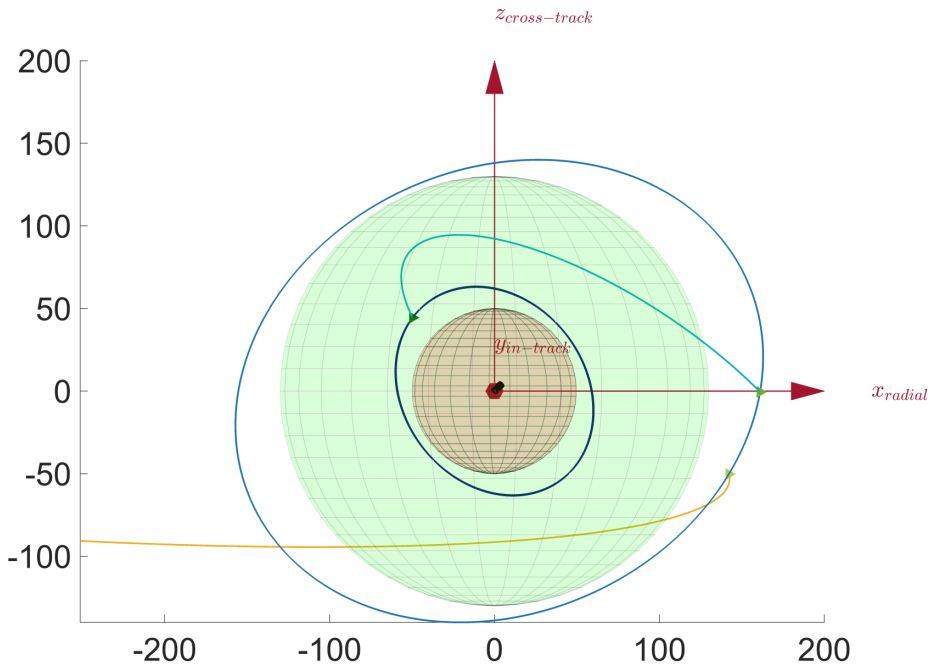


Figure 6.8: Safety Ellipse Example: Radial - Cross-track plane (in meters).

There is a particular combination of parameters such that the safety Ellipse has a circular shape (Figure 6.9): maintaining a constant distance may turn out to be an

advantage and a convenience. This trajectory is realized by imposing the conditions $\phi_0 = \psi_0$, $z_{max} = \sqrt{3}x_{max}$ and cancelling the quantity Δy to make the center of the ellipse coincide with the real position of the target: the number of variables to be checked by the algorithm, therefore, is reduced. A peculiar feature of these orbits is the angle of 60° between the radial direction and the projection of the orbit on the radial - cross-track plane, as visible in Figure 6.10.

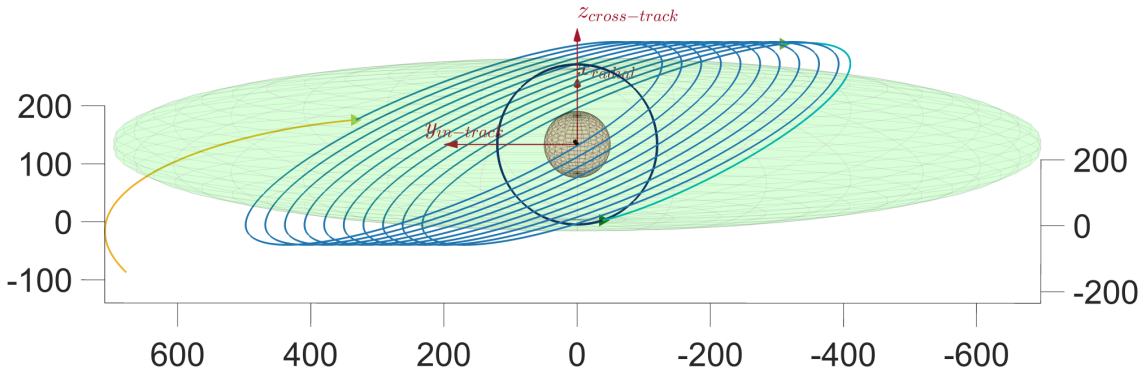


Figure 6.9: Circular Safety Ellipse Example in Hill frame (in meters).

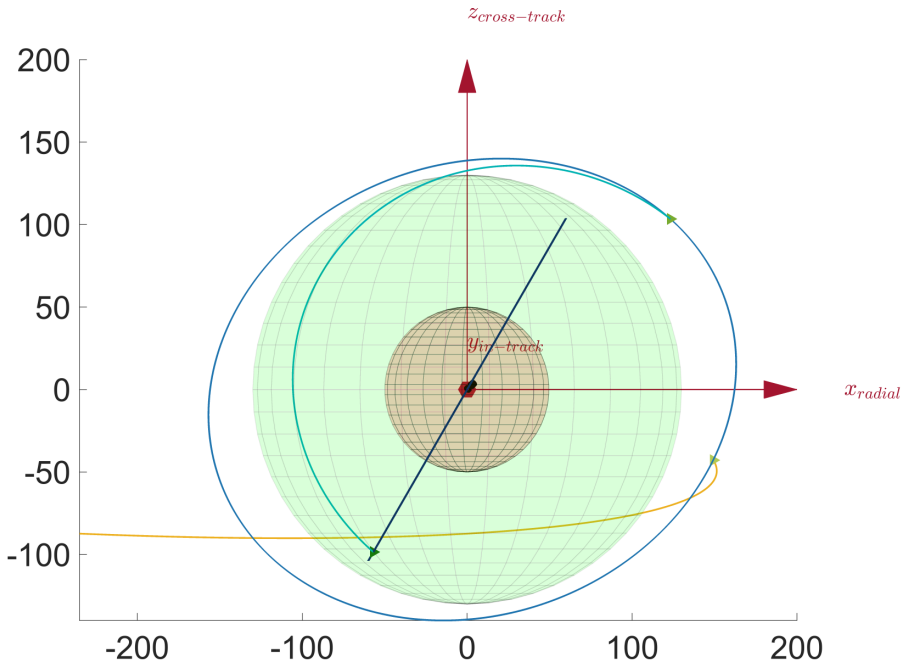


Figure 6.10: Circular Safety Ellipse Example: Radial - Cross-track plane (in meters).

6.3.2 Walking Safety Ellipse

To observe the target exploiting a natural motion, it is possible to use another WSE that is closer to the target with respect to the previous one. For this second case the same considerations of the first one apply, and only the ranges of parameters will be different: these will be set to ensure smaller dimensions to the spiraling motion and thus allowing a closer observation, as one can see looking at Table 6.8. The cost function, containing the same contributions as the first case, will not be reported.

x_{max}	$[60 \div 70]\ m$
$\dot{y_c}$	$[-0.003 \div 0.003]\ m/s$
Δy	$[30 \div 60]\ m$
z_{max}	$[0 \div 60]\ m$
ψ_0	$[0 \div 2\pi]$
ϕ_0	$[0 \div 2\pi]$
Δt_{Transf}	$[0 \div 0.5\ T_{targ}]$

Table 6.8: Inner Walking Safety Ellipse ranges.

Example of an inner Walking Safety Ellipse

As done for the previous subsection, an example is given in the following. The geometric parameters are listed in Table 6.9.

x_{max}	$69.99\ m$
$\dot{y_c}$	$-0.003 m/s$
Δy	$30.03\ m$
z_{max}	$59.97\ m$
ψ_0	4.9140
ϕ_0	3.3371
Δt_{Transf}	$0.4117\ T_{targ}$

Table 6.9: Inner Walking Safety Ellipse parameters.

The main results are presented in Table 6.10. The propagation on the WSE is done in such a way to cover 2 times the initial in-track distance along the in-track axis.

$\mathbf{x}_{to\ transf2}$	$[143.21, -252.10, 87.25]\ m$
$\mathbf{x}_{WSE,inserction}$	$[-66.72, 48.75, -11.65]\ m$
$\Delta V_{to\ transf2}$	$0.1222\ m/s$
$\Delta V_{WSE,inserction}$	$0.1565\ m/s$
Δt_{WSE}	$5h\ 38m\ 45.02s$

Table 6.10: Main parameters of the trajectory to the inner WSE.

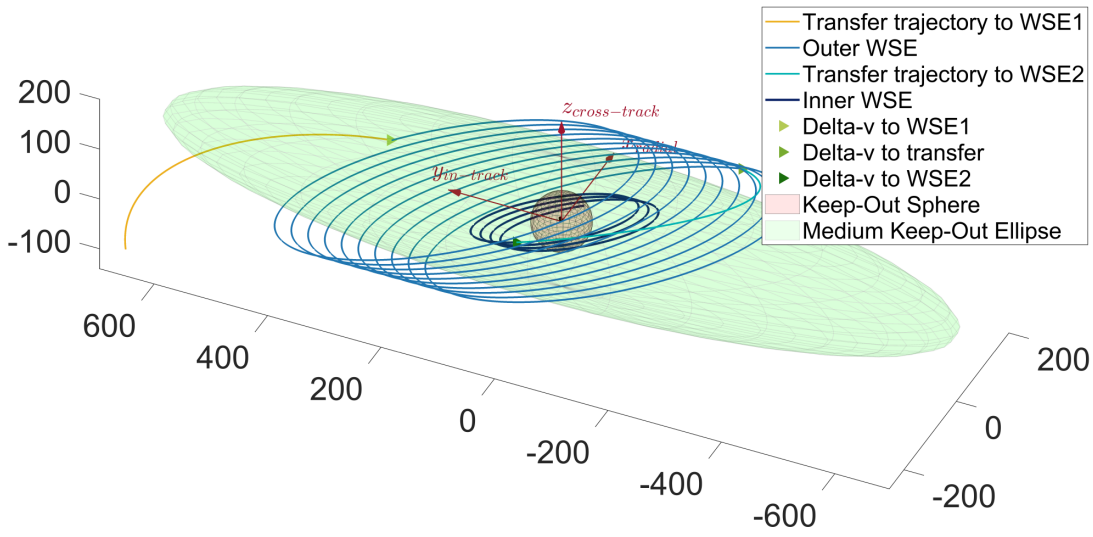


Figure 6.11: Inner WSE Example in Hill frame (in meters).

6.3.3 Waypoints

The last observation method analysed, unlike the others already presented, requires a higher cost in terms of propellant: maintaining a fixed position with respect to the target, if outside the in-track line (or V-bar), implies the use of the propulsion system, as explained in Chapter 3.

On the other hand, this method has the advantage of moving in the vicinity of the

target as needed, allowing an observation from multiple angles through a vision system and therefore improving the quality of information acquired on it. Among the forced motions mentioned in Chapter 3, only the observation from waypoint, arranged in an octahedron, considered more interesting and complete, will be analysed.

The waypoints are the vertices of an octahedron, rotated in such a way that each point lies on the axes of the Hill's frame, as shown in Figure 3.7. The distance of these waypoints from the target is equal to the radius of the KOS multiplied by the usual safety margin of 1.2 and by $\sqrt{2}$, to ensure that the KOS is entirely contained in the volume of the octahedron.

Genetic Algorithms for Waypoints

The algorithm is used 6 times, that is, before arriving to each of the 6 waypoints. As for the implementation, limits are placed only on the transfer time needed to reach the next waypoint:

$$\Delta t_{Transf} \in [300s \div 0.7\ T_{targ}]$$

The cost function, as in other cases, is imposed as follows:

$$FitnessValue = \Delta V_{tot} + W_1 + W_2$$

where the additive terms are explained in the following:

- ΔV_{tot}: sum of two delta-v, the first one to take the transfer trajectory between consequent waypoints, the second one to stop at the next waypoint;
- W_1: cost on the minimum distance from the target, d_{min}, during the transfer trajectory between waypoints, imposed in the following way:

  $$W_1 = e^{\gamma R_{KOS} - d_{min}} - 1$$

  where parameters have the same meaning of previous examples;
- W_2: cost on the maximum distance from the target, d_{max}, during the transfer trajectory between waypoints, to guarantee that the chaser is kept in the target vicinity. W_2 is imposed in the following way:

  $$W_2 = e^{d_{max} - D} - 1, \quad with\ D = 75m$$

D is set equal to half the X-semiaxe of the medium Keep-Out Ellipse.

Example of an Octahedron structure

Before showing the results, the following aspect must be emphasized: the optimal sequence of waypoints used was found by attempts, imposing the passage through a given point a single time.

The time spent on each waypoint was set as a reasonable fraction of the period, to ensure sufficient time for the vision system to acquire information.

Main results are given in Tables 6.11, 6.13 and 6.12.

$\mathbf{x}_{to\ WP1}$	$[-117.54, 108.74, -65.00]\ m$
$\mathbf{x}_{WP1}$	$[-84.85, 0, 0]\ m$
$\mathbf{x}_{WP2}$	$[0, 84.85, 0]\ m$
$\mathbf{x}_{WP3}$	$[84.85, 0, 0]\ m$
$\mathbf{x}_{WP4}$	$[0, 0, 84.85]\ m$
$\mathbf{x}_{WP5}$	$[0, -84.85, 0]\ m$
$\mathbf{x}_{WP6}$	$[0, 0, -84.85]\ m$

Table 6.11: Waypoints trajectory: principal points.

$\Delta t_{to\ WP1}$	$0h\ 42m\ 7.61s$
$\Delta t_{WP1-WP2}$	$0h\ 34m\ 47.69s$
$\Delta t_{WP2-WP3}$	$0h\ 34m\ 47.69s$
$\Delta t_{WP3-WP4}$	$0h\ 25m\ 29.49s$
$\Delta t_{WP4-WP5}$	$0h\ 28m\ 6.69s$
$\Delta t_{WP5-WP6}$	$0h\ 28m\ 6.69s$
Δt_{keep}	$1/10\ T_{targ}$

Table 6.12: Waypoints trajectory: propagation times.

$\Delta V_{to\ WP1}$	0.1047 m/s
$\Delta V_{stop\ at\ WP1}$	0.2565 m/s
$\Delta V_{to\ WP2}$	0.1586 m/s
$\Delta V_{stop\ at\ WP2}$	0.0290 m/s
$\Delta V_{to\ WP3}$	0.0290 m/s
$\Delta V_{stop\ at\ WP3}$	0.1586 m/s
$\Delta V_{to\ WP4}$	0.1891 m/s
$\Delta V_{stop\ at\ WP4}$	0.0579 m/s
$\Delta V_{to\ WP5}$	0.0562 m/s
$\Delta V_{stop\ at\ WP5}$	0.1061 m/s
$\Delta V_{to\ WP6}$	0.1061 m/s
$\Delta V_{stop\ at\ WP6}$	0.0562 m/s
$\mathbf{\Delta V_{TOT}}$	**1.3081 m/s**

Table 6.13: Waypoints trajectory: Delta-v.

Note that ΔV_{TOT} in Table 6.13 only takes into account the contributions to arrive/start from a given waypoint: station-keeping delta-vs, even if small, are not included nor calculated, since this phase only focuses on the computation of the reference trajectory. In the next section, however, this aspect will be analysed using an MPC. As done for the previous cases, 3D plots are provided to visualize this strategy (Figures 6.12, 6.13 and 6.14).

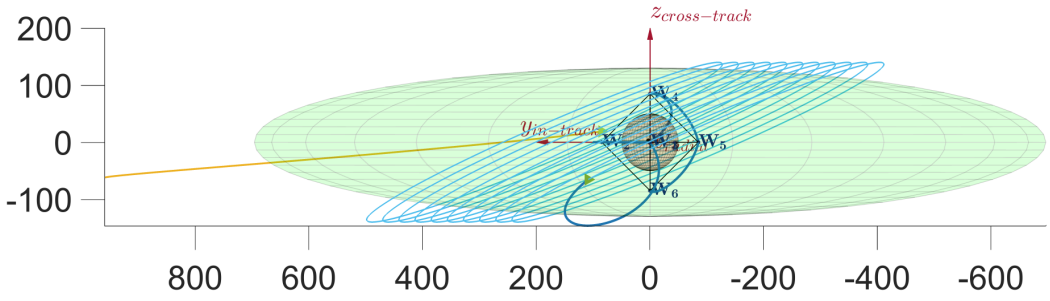


Figure 6.12: Octahedron structure Example: In-track - Cross-track plane (in meters).

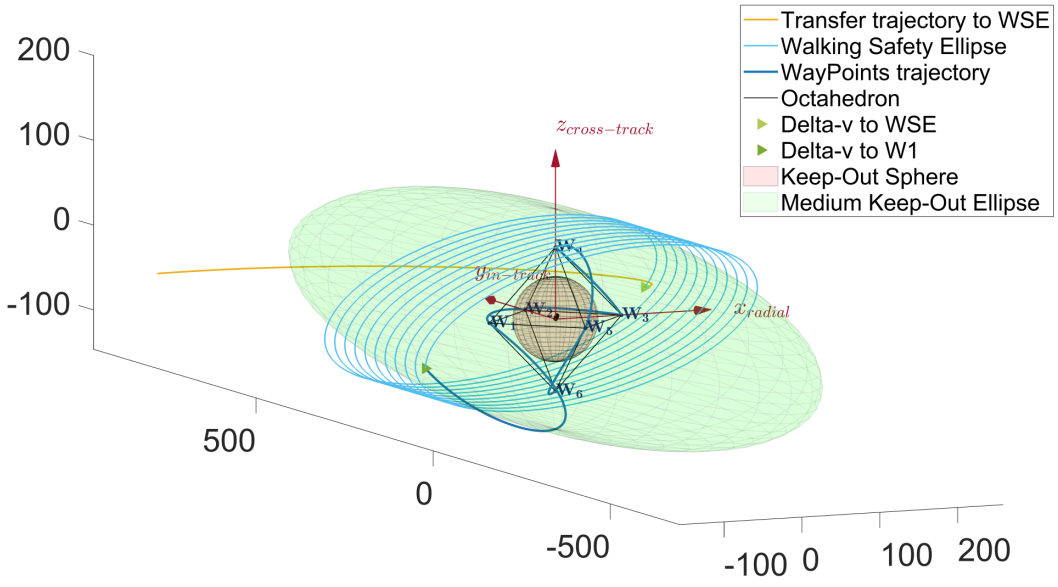


Figure 6.13: Octahedron structure Example in Hill frame (in meters).

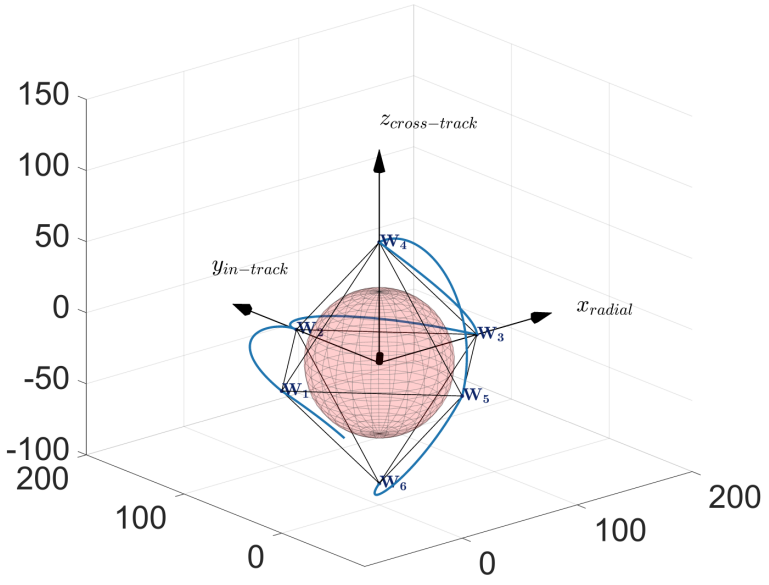


Figure 6.14: Octahedron structure Example: Particular (in meters).

6.4 Control on Observation Trajectories

In this section the tracking of the trajectories previously created, performed by the spacecraft simulator presented in Chapter 5, will be shown, at least in some details. Since these are mostly natural trajectories with some impulsive delta-v, what is interesting is the area straddling a delta-v, or, eventually, how the system behaves at a waypoint. With regard to the natural propagations between one delta-v and the next one, therefore, only some final considerations will be made.

6.4.1 Set of Simulator subsystems

Before presenting what promised, however, it is necessary to give some details of the simulator settings to successively perform the simulations.

Simulation Parameters

As first, the simulation parameters used are set in Table 6.14.

dT_{simul}	$1\ s$
Relative tolerance ($ode45$)	10^{-7}
Absolute tolerance ($ode45$)	10^{-7}

Table 6.14: Simulation parameters.

Given the time required for the observation phase, a simulation time step of $1s$ allows a sufficiently accurate description. The initial and final times will be discussed and provided in the results subsection.

Thrusters

Since the maneuvers to be performed are not onerous, the propulsion system explained in Subsection 5.2.2 is not used entirely: the maximum translational thrust set along any body direction is $200N$, so half of the total deliverable.
Since small values of the control force are required, smaller than the minimum provided by the $200N$ engine, that is $(F_{max}\ \tau)/dt_{simul} = (200\ 0.028)/1 = 5.6N$, the simulator is

set in such a way to switch thrusters under a certain thrust value, using two attitude control thrusters instead of the larger engine.
To sum up, this is the range of thrusts that can be delivered:

$$F_{calc}^{B} \in [0.2, 200]N$$

As already explained, not having the matrix R_O^B, it would not be possible to directly impose these body constraints on the forces calculated by the control system, being instead in the LVLH frame: for simplicity, therefore, it is assumed that the body system is always aligned with the orbital system. In addition to this, as explained in Subsection 5.2.2, a maximum control action has been defined in order to protect against any problems (Table 6.16).

MPC controller

The main parameters of the Model Predictive Controller, resulting from tuning, are listed in Table 6.15. The control action is calculated and delivered every simulation time-step. Constraints are presented in Table 6.16. The value of 0.0178 m/s^2 for the acceleration is found through the condition in Eqn 5.6, already discussed. Finally, the weights are presented in Table 6.17, where $n = 0.00106088\ rad/s^2$ is the orbital mean motion. Only the diagonal of R, S and Q matrices presented in Chapter 4 is given.

Sample Time T_s	1 s
Control Horizon m	15
Prediction Horizon p	65

Table 6.15: MPC parameters.

$\mathbf{u_{max}}$	0.0178 $[1, 1, 1]$ m/s^2
$\mathbf{u_{min}}$	-0.0178 $[1, 1, 1]$ m/s^2
$\mathbf{\Delta u_{max}}$	0.0178 $[1, 1, 1]$ m/s^2
$\mathbf{\Delta u_{min}}$	-0.0178 $[1, 1, 1]$ m/s^2

Table 6.16: MPC constraints.

$diag[Q]$	$[1, 1, 1, 1, 1, 1]$
$diag[R]$	$1e^{-9}/n^2\ [1, 1, 1]$
$diag[S]$	$5e^{-4}/n^2\ [1, 1, 1]$

Table 6.17: MPC weights.

Noise and Kalman Filter

As already mentioned in the previous chapter, the state vector contains relative position and velocity, but only the position is measured. The errors of the sensors, not being present, were simulated by adding a random normal distribution on the real values:

$$\tilde{y}_k = C\hat{x}_{k-1}^- + [\sigma_x a_1, \sigma_y a_2, \sigma_z a_3]$$

where a_1, a_2, a_3 are random values with normal distribution and $\sigma = \sigma_x = \sigma_y = \sigma_z = 0.7m$ is the standard deviation, reasonably assumed considering sensors accuracies and distances from the target involved.
As explained in Subsection 4.2.3, this measure, $\tilde{y}_k$, then contributes to update the state estimate that leads to the prediction step in the Kalman filter.
To initialize the Kalman filter, the initial estimate $\hat{x}_0^-$ and the state estimate error covariance matrix P_0^- must be provided. Assuming Gaussian distribution, also the matrices of covariance R_{KF} for measurement noise and Q_{KF} for process noise have to be defined, tuning them to make the filter converge. Table 6.18 groups all these matrices, while the initial estimate will be provided in the following, when the initial state will be already defined.

P_0^-	$diag\,[[4[1, 1, 1], 10^{-3}[1, 1, 1]]]$
Q_{KF}	$diag\,[10^{-7}\ [1, 1, 1, 1, 1, 1]]$
R_{KF}	$diag\,[30\sigma^2\ [1, 1, 1]]$

Table 6.18: Kalman filter specifics.

Note that the matrix of covariance R_{KF} of measurement noise is built using σ, as it should be.

6.4.2 Control on the Observation trajectory

As specified at the beginning of the section, it will not be shown how the simulator performs along the entire trajectory, but only in some specific cases, that is, straddling an impulsive delta-v or on a waypoint.

Behaviour at any ΔV

The maneuver examined is that of insertion in the Safety Ellipse, highlighted in Figure 6.15. The simulation time interval has not only to include the impulsive delta-v, but

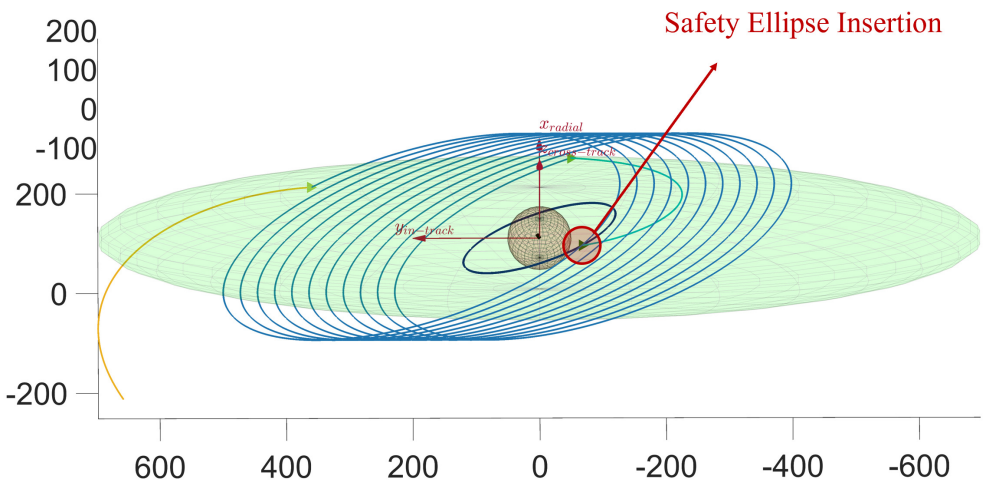


Figure 6.15: Safety Ellipse insertion point (in meters).

the beginning of this interval has to be chosen according to two factors: the prediction horizon and the time it takes the Kalman filter to converge. As for the first, to test the peculiarity and bounty of the Model Predictive Controller correctly, the impulsive delta-v must be "seen" at the end of the prediction horizon, or starting even before, outside of it. In particular, considering the parameters of the Table 6.15, it will be enough to start at least 65 s before the impulsive delta-v is actuated and simulate the execution of the maneuver up to a final time far enough from the instant of the impulsive delta-v itself. In any case, in order to clearly distinguish the control actions

due to the delta-v and those due to the settling of the filter, it was chosen to start almost 5 prediction horizons before.
Table 6.19 shows the main parameters to initialize the simulation:

$\mathbf{x_0}$	$[[-56.568, -113.358, 68.3947]\ m,\ [0.0014, 0.1584, -0.0693]\ m/s]$
$\hat{\mathbf{x}}_\mathbf{0}^-$	$[[-56.473, -113.026, 68.093]\ m,\ [0.0013, 0.1582, -0.0692]\ m/s]$
ΔT_{simul}	$449\ s$

Table 6.19: Main parameters for delta-v to the Safety Ellipse.

Main results are presented and discussed in the following. A comparison with the case of perfect measurements (that is, there's no need of the Kalman filter) is given. Looking at Figure 6.17, it's clear that the filter is acting: the curves are smooth, with no rapid oscillations. After some initial, large fluctuations due to the filter convergence, the shapes of the real curve and the curve with no measurement errors become similar. This fact can be seen also for velocities or forces, especially with regard to the latter, as one can see in Figure 6.20.

It is emphasized, however, that the reference trajectory is ideal and cannot be followed perfectly by a real system, being imposed an instantaneous change of velocity along a continuous position profile.

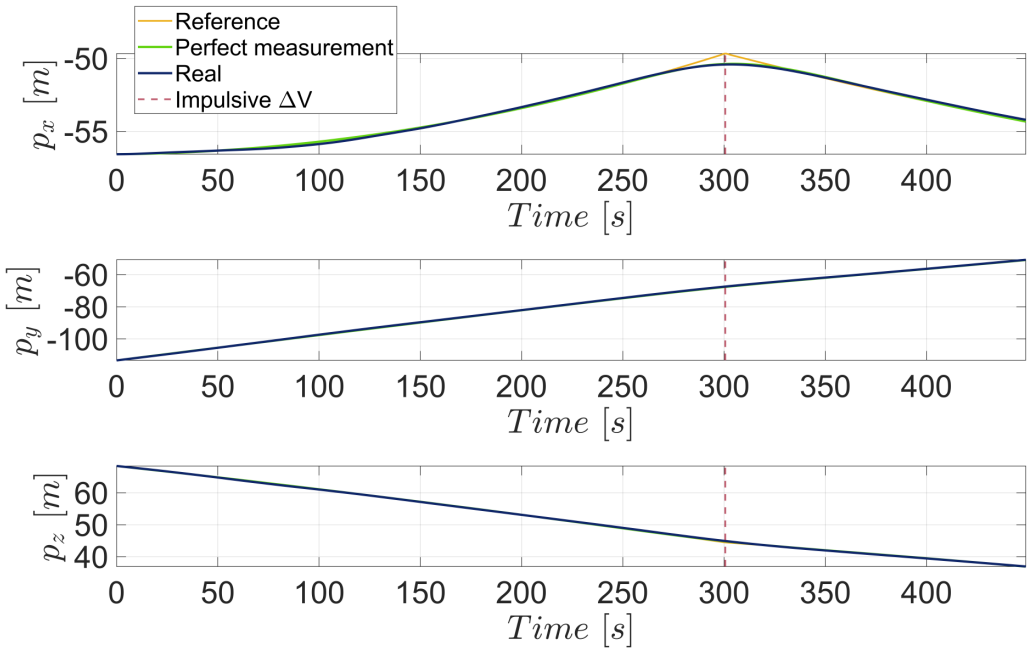


Figure 6.16: Delta-v to Ellipse: relative position.

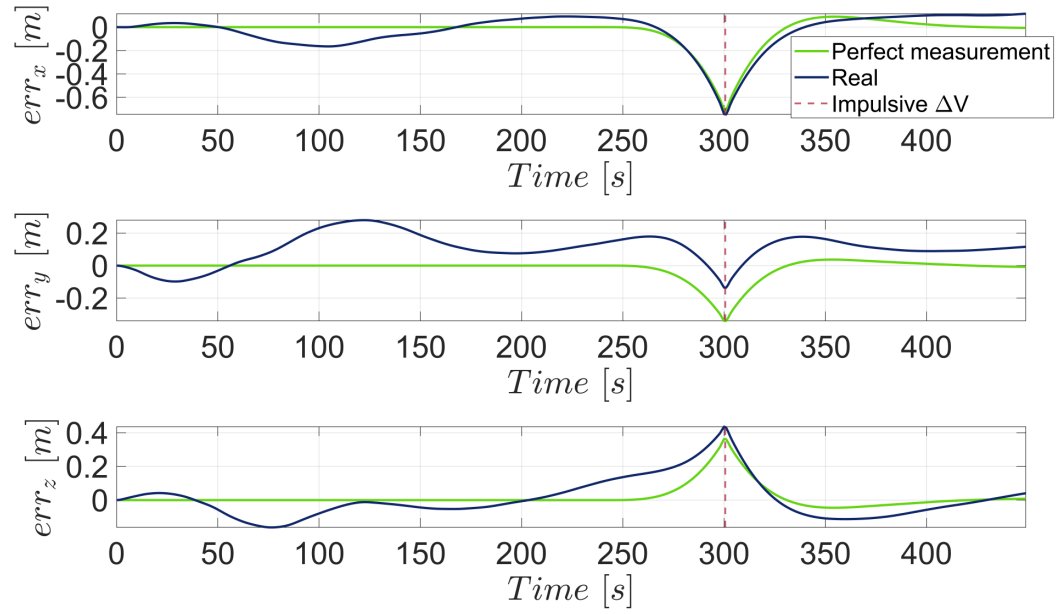


Figure 6.17: Delta-v to Ellipse: errors on relative position.

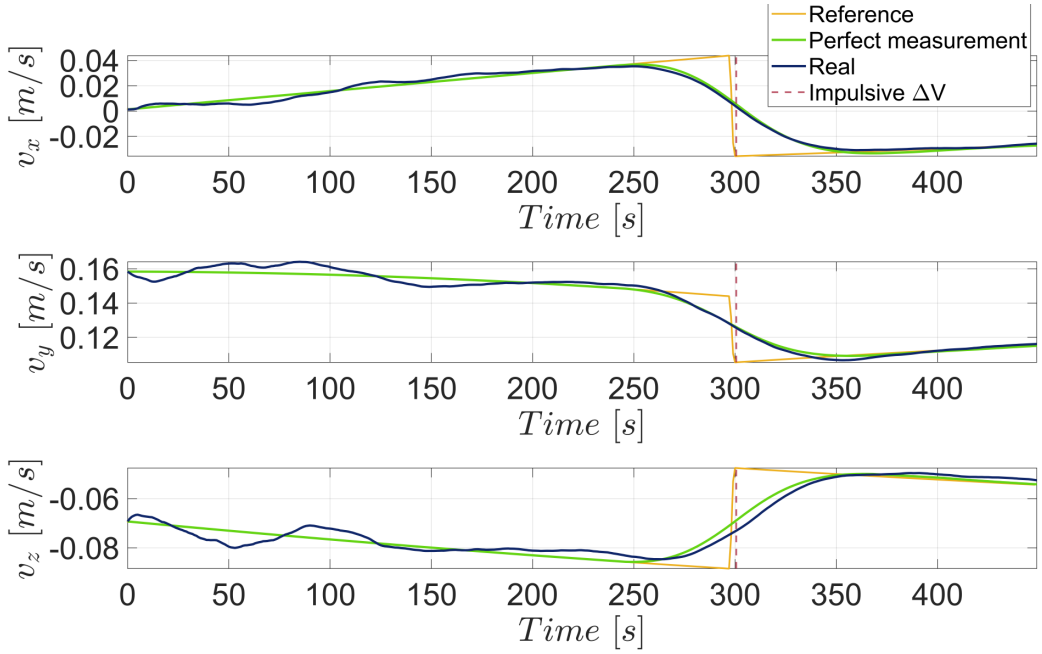


Figure 6.18: Delta-v to Ellipse: relative velocity.

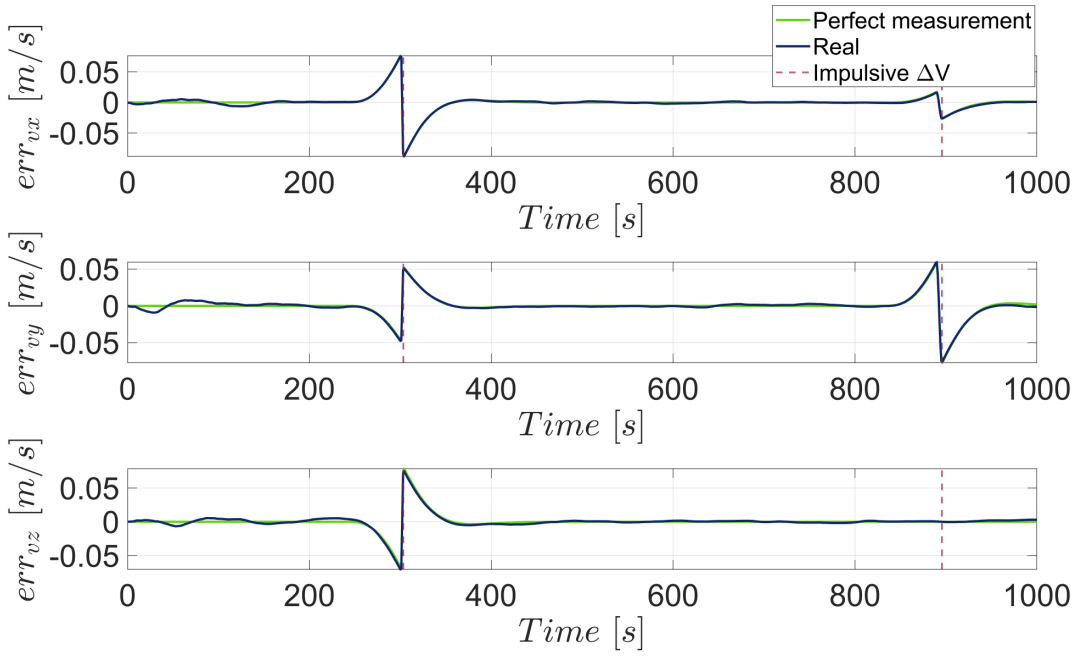


Figure 6.19: Delta-v to Ellipse: errors on relative velocity.

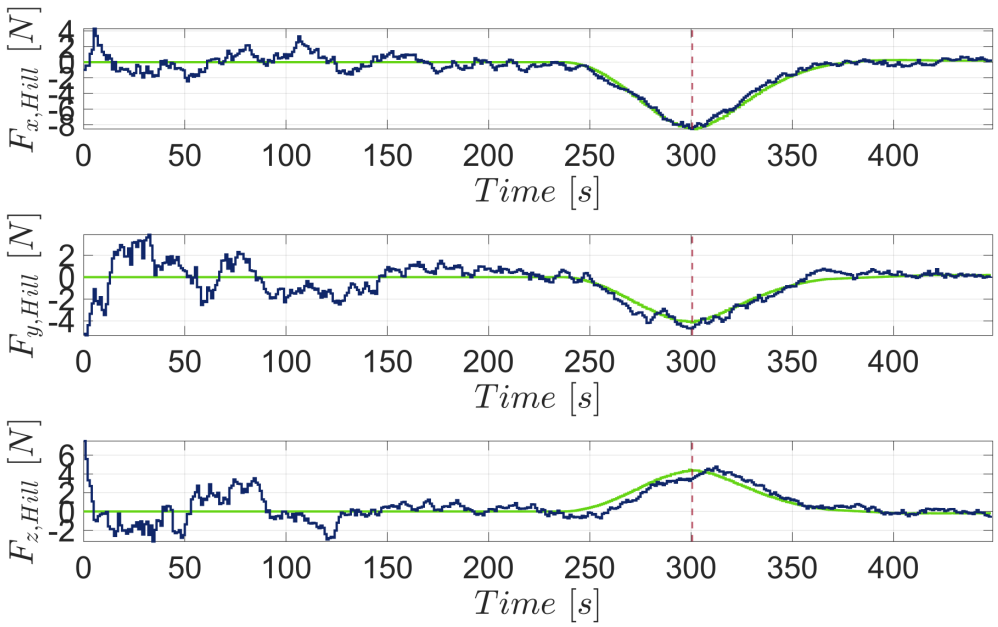


Figure 6.20: Delta-v to Ellipse: forces in Hill frame.

Especially in Figure 6.18, it is possible to appreciate how the controller begins to act well before the instant of the impulsive ΔV, confirming the main feature of the MPC, namely the ability to provide control actions calibrated on a time horizon that includes several control time-steps.

With regard to the actuated forces, given in Hill's frame (which, by assumption, coincides with the body reference), it is necessary to specify that the initial forces, which are irregular due to the initialization of the filter, and the forces due to the maneuver are comparable due to the exiguity of the latter, as visible in Table 6.20. Figures 6.21 and 6.22 prove the convergence of the filter: the difference between the estimated value and the real one, both for the position and for the velocity, is always confined within the 3σ dispersion and shows a slight decrease over time.

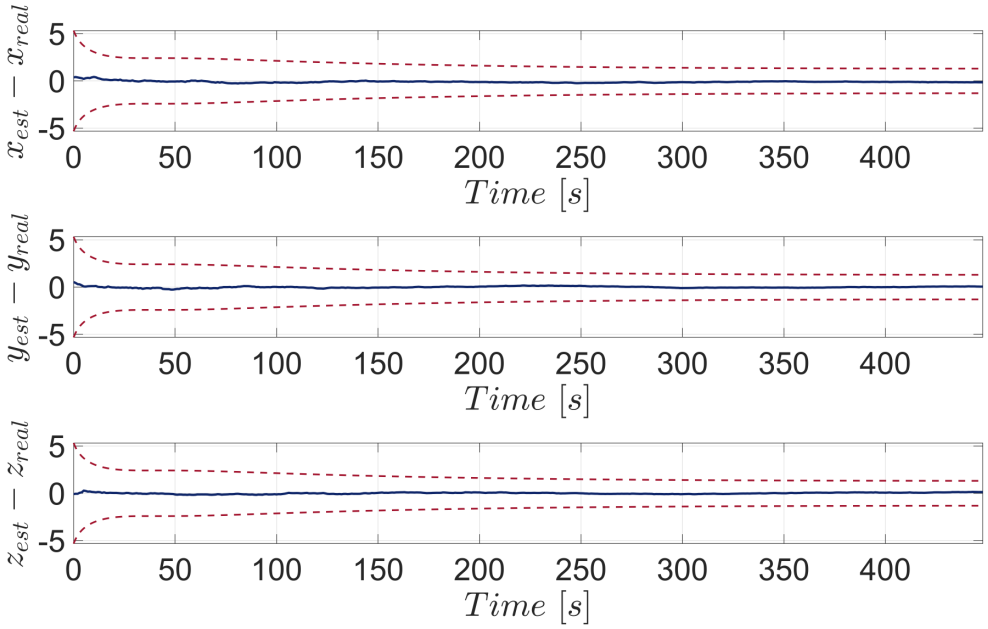


Figure 6.21: Delta-v to Ellipse: Kalman filter on relative position [m].

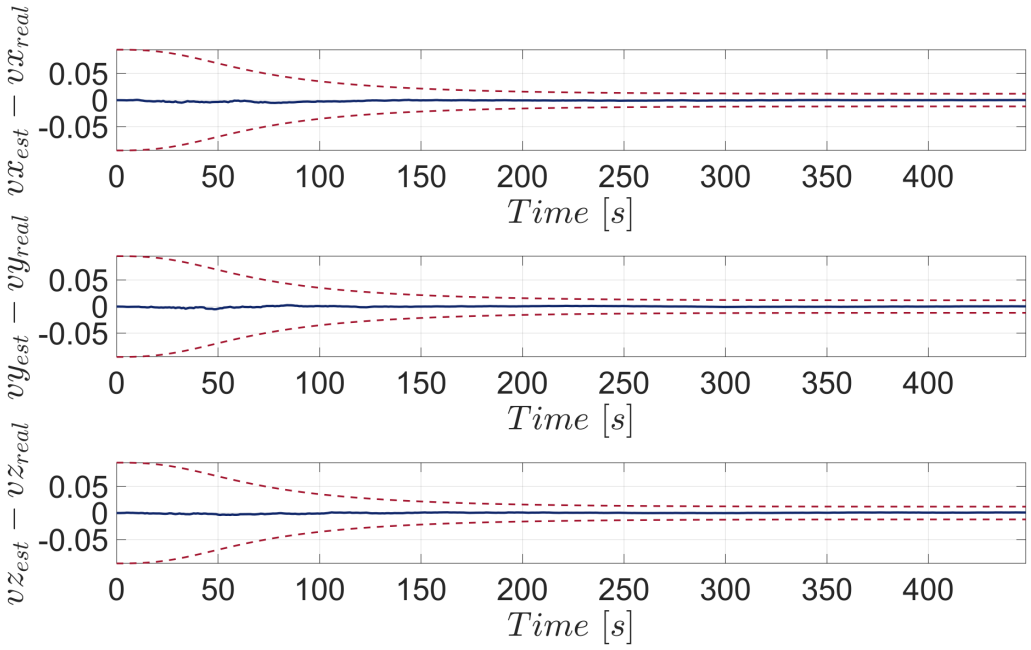


Figure 6.22: Delta-v to Ellipse: Kalman filter on relative velocity [m/s].

Finally, differences in consumption are listed in Table 6.20.

$\Delta V_{impulse}$	0.0976 m/s
$\Delta V_{perfect\ meas}$	0.1691 m/s
$\mathbf{\Delta V_{real}}$	**0.2877 m/s**
$\mathbf{m_{prop}}$	**0.8384 kg**

Table 6.20: ΔV_{tot} comparison for Ellipse insertion.

Consider that ΔV_{real} takes into account the initial phase of filter settling, so the difference between this value and the one without measurement errors is mainly due to this fact.

Behaviour at waypoint

Another interesting example is the relative position-keeping at a waypoint, with no relative velocity. Considering Figure 6.13, the waypoint examined is the first one. The calculated impulsive maneuvers are two, the first one to stop at the waypoint and the second one to get away from it. In the middle, as already specified, the motion will be forced, so the system will have to compensate for the gravity gradients dictated by the HCW equations.
The considerations made for the previous case regarding the prediction horizon and the convergence of the filter also apply to this scenario, so the simulation will start sufficiently before reaching the waypoint.
Table 6.21 shows the main parameters to initialize the simulation.

$\mathbf{x_0}$	$[[-35.5889, -15.6675, -46.2954]\ m,\ [-0.1472, 0.0009, 0.1473]\ m/s]$
$\hat{\mathbf{x}}_\mathbf{0}^-$	$[[-35.4977, -15.6137, -46.2628]\ m,\ [-0.1466, 0.0009, 0.1466]\ m/s]$
ΔT_{simul}	1000 s

Table 6.21: Main parameters for delta-v to the first waypoint.

The main results of kinematics and dynamics are presented in figures from 6.23 to 6.27, while the graphs concerning the convergence of the filter are omitted due

to redundancy, being very similar to those of the previous case. The two impulsive maneuvers are marked with vertical dotted lines.

Since the simulation time is longer and delta-vs are greater than that of the ellipse insertion, it is possible to notice, more than in the previous case, how the oscillations due to the convergence of the filter are less visible and occupy a quite limited initial time range.

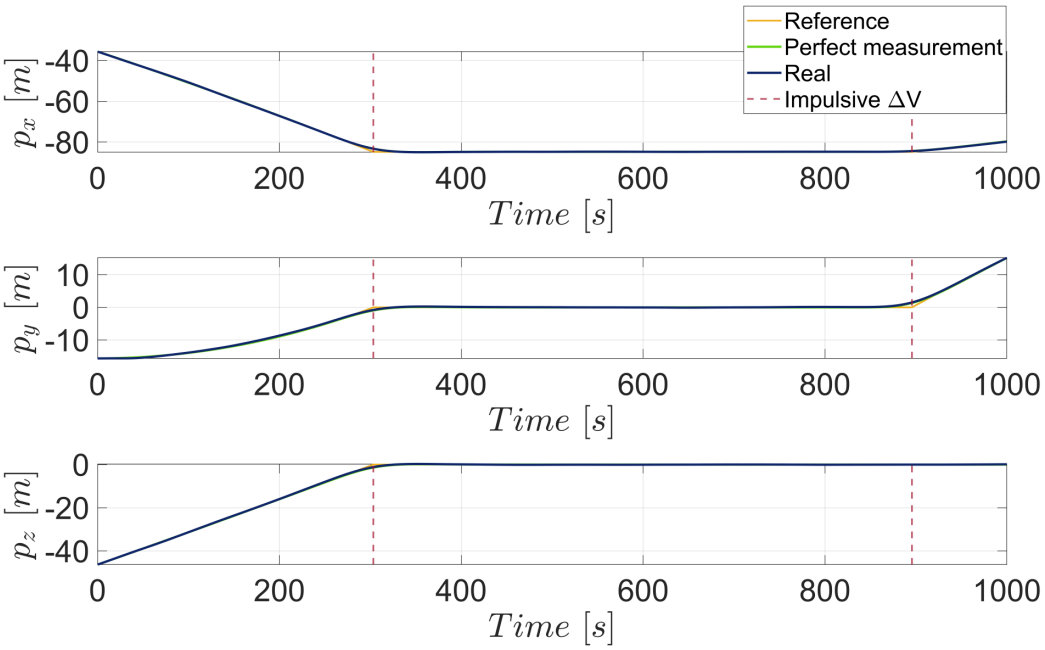


Figure 6.23: Delta-v at waypoint: relative position.

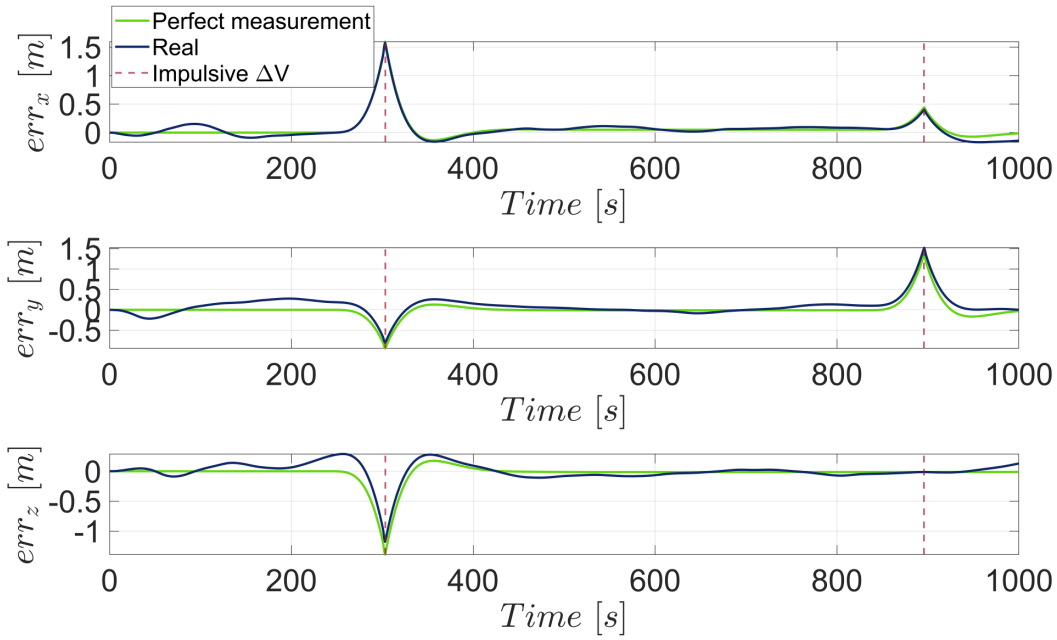


Figure 6.24: Delta-v at waypoint: errors on relative position.

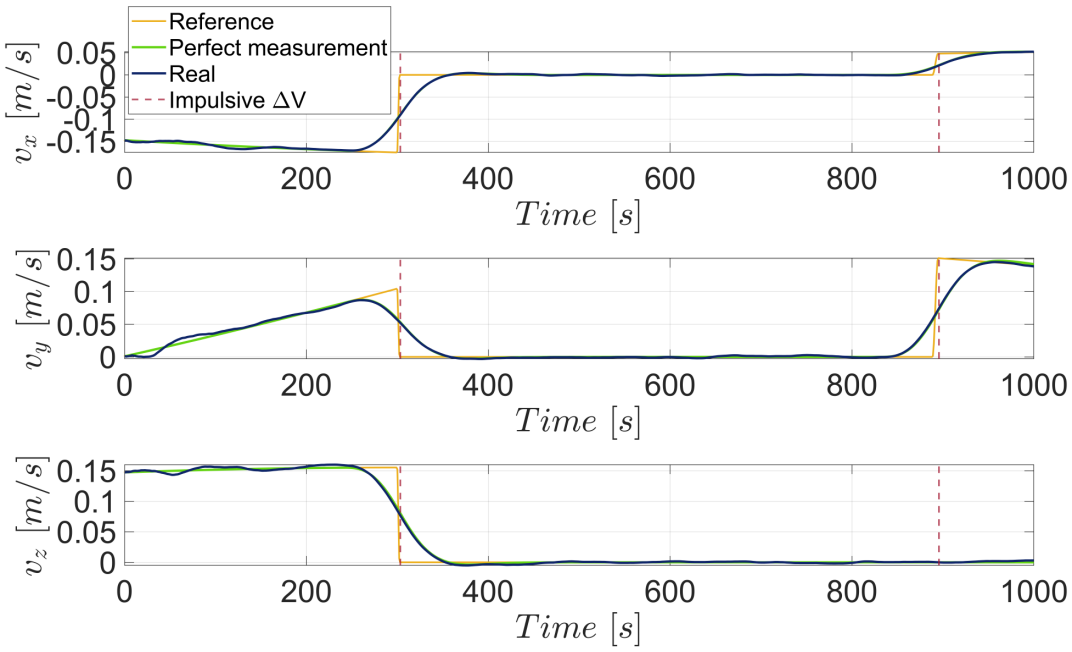


Figure 6.25: Delta-v at waypoint: relative velocity.

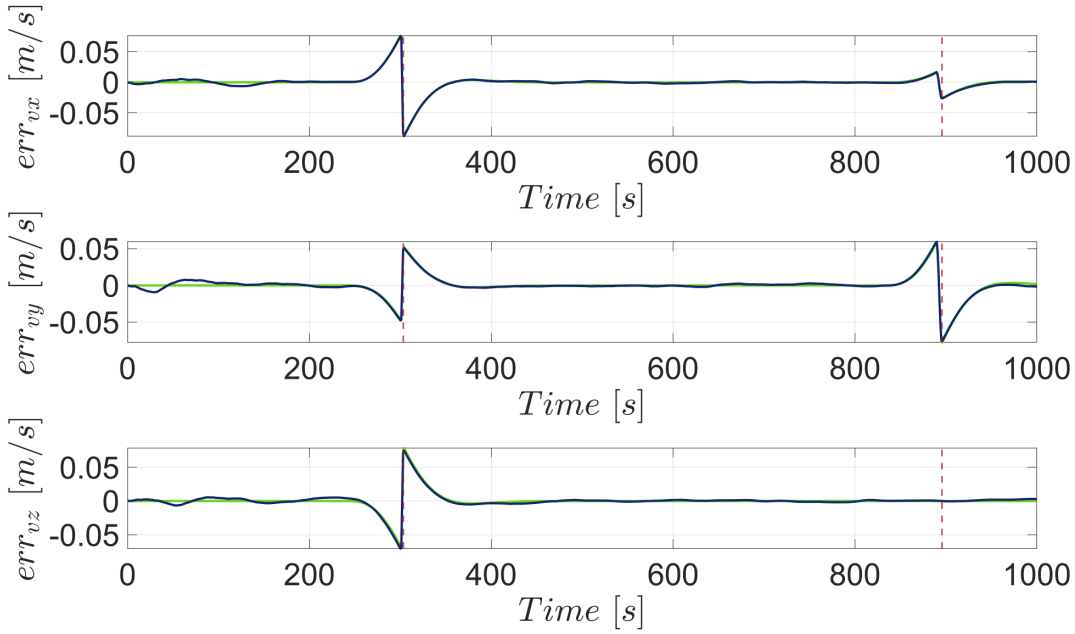


Figure 6.26: Delta-v at waypoint: errors on relative velocity.

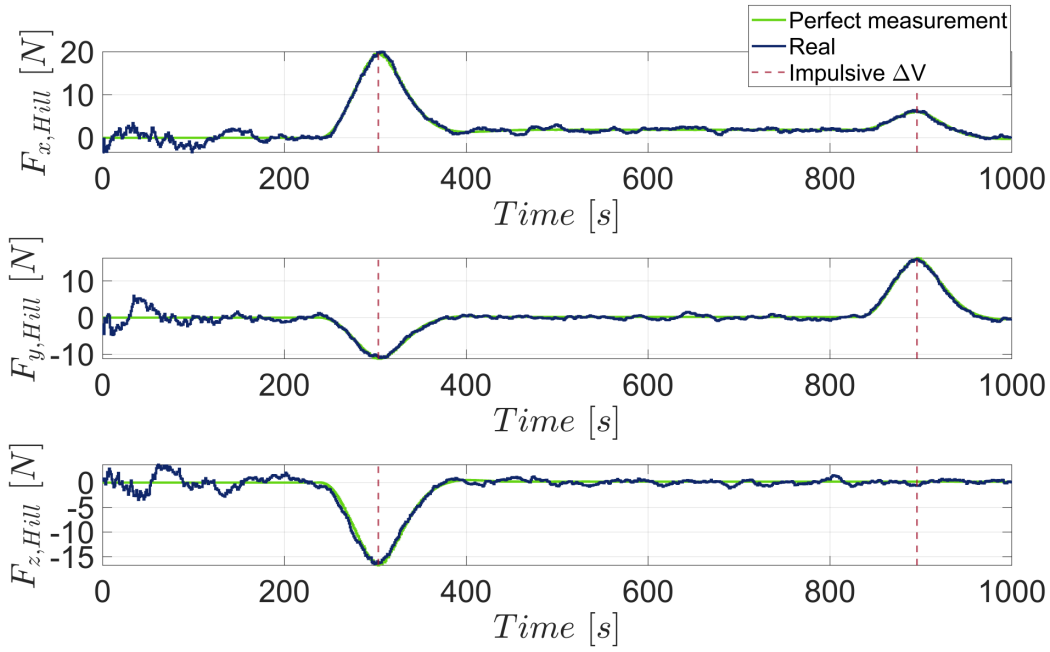


Figure 6.27: Delta-v at waypoint: forces in Hill frame.

As visible in the first graph of Figure 6.27 between the two impulsive maneuvers, having the waypoint only a radial offset, it is comprehensible, considering the HCW

equations of dynamics (Eqn 2.2), that the control action is delivered mainly in the radial direction to compensate for the gradient.

As for the previous case, a comparison between impulsive, perfect measurements and real consumptions are listed in Table 6.22. $\Delta V_{impulse}$ considers both the maneuver contributes.

$\Delta V_{impulse}$	0.4151 m/s
$\Delta V_{perfect\ meas}$	0.8518 m/s
$\mathbf{\Delta V_{real}}$	**1.0083 m/s**
$\mathbf{m_{prop}}$	**2.9390 kg**

Table 6.22: ΔV_{tot} comparison for waypoint.

Further considerations

Regarding natural evolutions along the global trajectory, for example along the walking safety ellipse, the control strategy to be used must be different. To take advantage of the naturalness of the orbit, what should be done is to loosen the position constraints for the tracking of the trajectory and perform sporadic and impulsive delta-vs. If this were not the case, the continuous correction supplied by actuators, even just trying to compensate the measurement noise, would lead to a consumption of propellant even where it could be saved, making the use of planned natural trajectories useless.

This case, however, was not developed in this thesis.

Chapter 7

Inner Proximity Rendezvous Study

Once the observation has been carried out, the last phase of approaching within the inner keep-out volume can be initialized. This is the most complicated, energy-demanding and most challenging phase for the GNC system, for the reasons already discussed in Chapter 3. The strategy adopted is the one described in the same chapter: the maneuver is divided into two minor phases, a first free-flying phase to get within the cone of approach, and a second constrained phase to reach the last holding point moving within the same cone.

Remember that, unlike the previous phase, as explained in Chapter 5, to manage the attitude and several non-linear constraints, the controller used is the NLMPC and the filter is an EKF: their specifications will be provided for both phases.
Before analysing the two phases separately, it is useful to provide some global information, valid for both maneuvers.

Main geometric parameters

Figure 7.1 shows the dimensions of the main geometric structures within the 50m radius sphere. The presence of an intermediate 25m sphere represents the collision avoidance constraint for the first free-flying phase. The dimensions of the boxes at the holding points are reasonably set regarding the position uncertainties, which depend on the involved distances from the target.
The approaching cone half-angle and the size of the final shrinkage have been chosen

in agreement with studies in the literature and considering the size of the chaser.

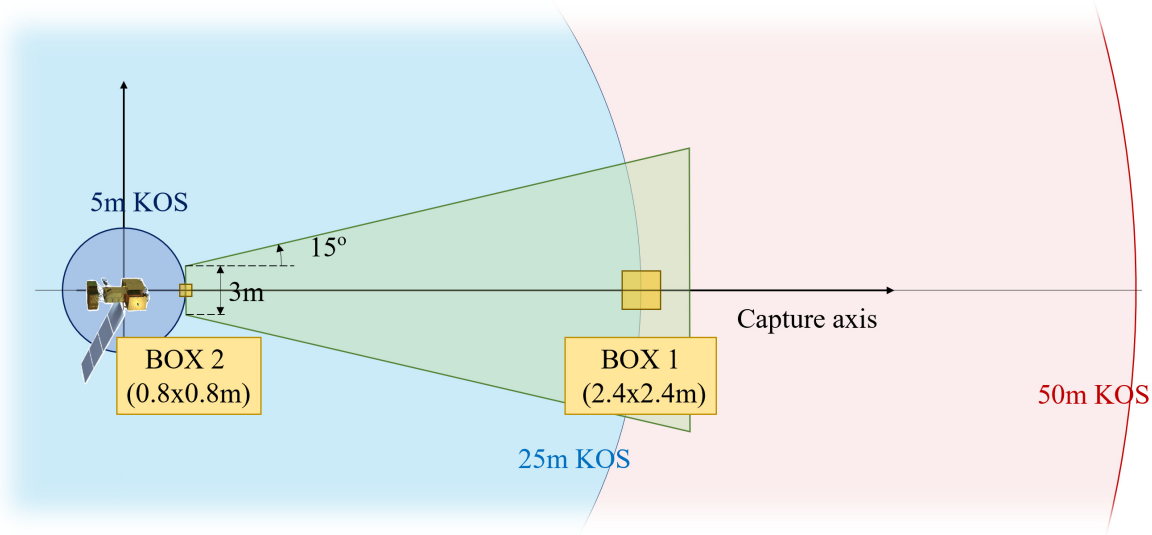


Figure 7.1: Principal dimensions of geometric structures.

Target

The grasping point identified on the target is well defined and coincides with a ring encircling the base of Landsat7 (Figure 5.4). Even if the grasping phase has not been simulated, it is necessary to identify the position of the capture point, with an offset of a few meters along the capture body axis, assuming the action of robotic arms. According to this last consideration, the chosen point lies on the smaller sphere of 5m radius centered in the target.

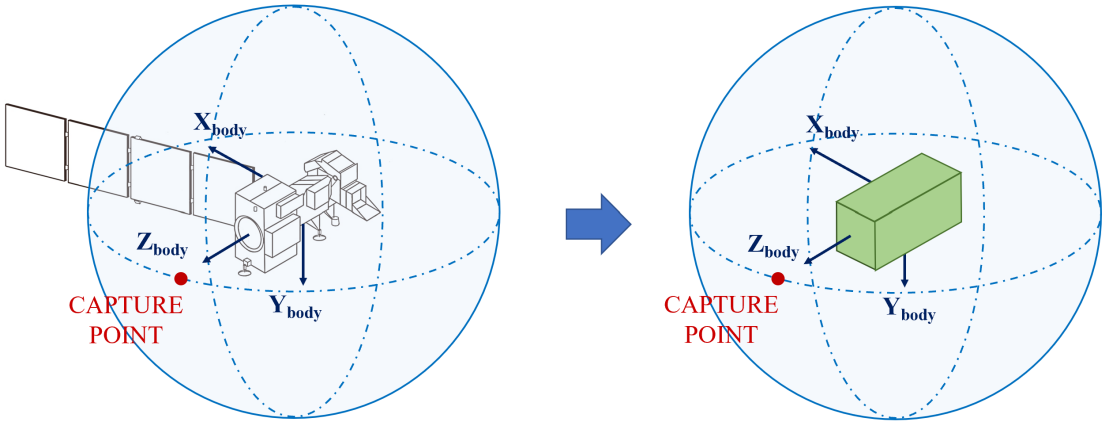


Figure 7.2: Capture Point.

Remember that the model of the two satellites involved has been simplified representing them as parallelepipeds (Figure 7.2), as reported in Chapter 5.

Another aspect to be specified, with regard to the target, is its tumbling motion, as stated at the end of Chapter 3: it is assumed that Landsat7 is already found in a state of flat spin, or a rotation around the axis of greater inertia, without contemplating any disturbing actions, considering them as negligible also because of the total duration of the maneuver (a few minutes).

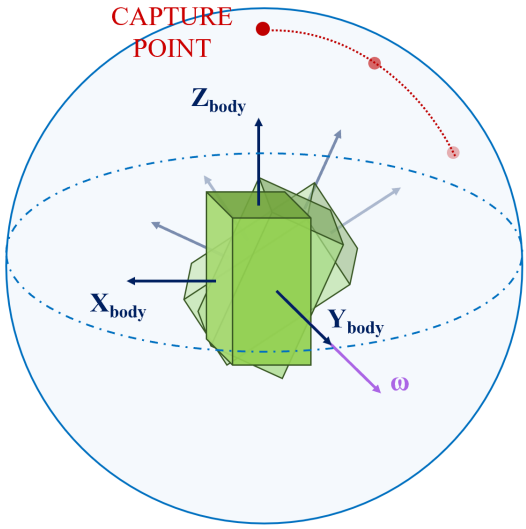


Figure 7.3: Tumbling motion of the target satellite: flat spin.

Being the target parallelepiped square based, the axes with greater inertia coincide with the two body principal axes x_B and y_B. Being indifferent which of the two axes is chosen, in the simulations the angular velocity of the target will be aligned with the y_B axis. As regards the magnitude of the angular velocity modulus, the rendezvous maneuvers will be studied for three target angular velocities, as reported in Table 7.1:

$\boldsymbol{\omega}_{BI,1}^{(B)}$	$[0,\ 0.5,\ 0]\ ^\circ/s$
$\boldsymbol{\omega}_{BI,2}^{(B)}$	$[0,\ 1.0,\ 0]\ ^\circ/s$
$\boldsymbol{\omega}_{BI,3}^{(B)}$	$[0,\ 1.5,\ 0]\ ^\circ/s$

Table 7.1: Considered angular velocities between body and inertial frames.

These values were chosen based on the typical motion of end-of-life satellites, excluding very low (below $0.1°/s$) and very high angular velocities ($4-5°/s$), these latter caused by failed three-axis stabilization.

Chaser

The side used to look at the target is the one indicated in Figure 7.4: the condition of visibility of the target is implemented through the versor x_B, considered for simplicity as the center axis of the hypothetical vision sensor.

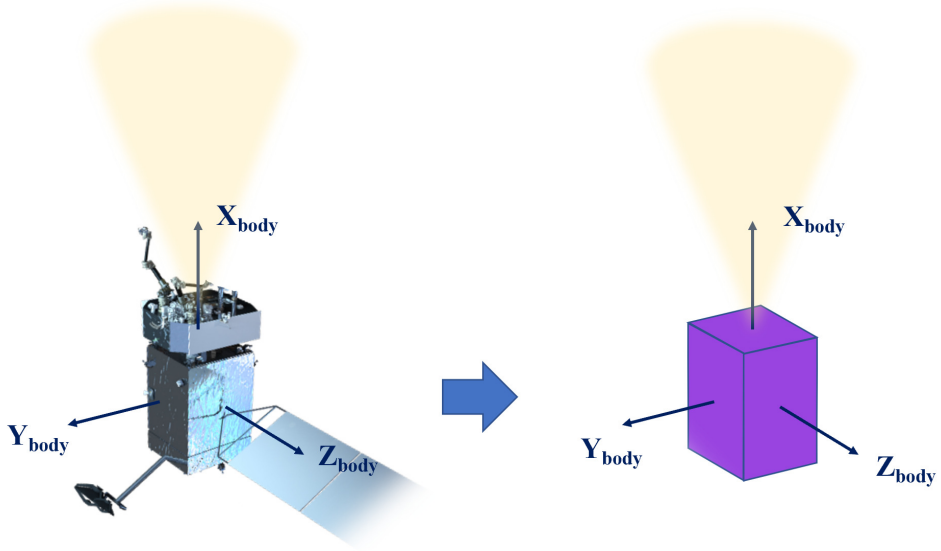


Figure 7.4: Field of View of the chaser.

Thrusters

As for the actuation system, in this final phase the use of thrusters is not reduced: there are one or two $200N$ thrusters for translational control along each body direction, and two or four $10N$ thrusters for attitude control around each main body axis.
Knowing the current attitude, and therefore the matrix R_O^B, it is possible to impose limits on the forces (commanded in the LVLH reference) without making assumptions. To sum up, the following is the range of thrusts that can be actuated:

$$Translation: \quad F_{calc}^B \in [5.6,\ 400]N$$

$$Attitude: \quad F_{calc}^B \in [0.2,\ 40]N$$

where the minimum values are calculated considering the minimum opening time of the valves in PWM. As regards attitude torques, a common distance of $1m$ between the center of mass and the position of a generic attitude thruster is considered.

7.1 First free-flying phase

The first phase, as already pointed out, coincides with the achievement of the first holding point within the approaching cone, starting outside the $50m$ radius sphere. The position of this point to be reached, in the body frame, is:

$$pos_{HP1} = [0,\ 0,\ 25]^T\ m$$

The reference trajectory coincides with the motion of the holding point in the orbiting reference, and this is created online at each control step, propagating the estimate of the current position of this point throughout the prediction horizon, with the current information about inertia and angular velocity of the target.

To study the behaviour of the NLMPC controller, for each angular velocity of the target, six simulations were performed, each with a different initial position. These positions, listed in Table 7.2, coincide with the face centers of a hypothetical cube centered in the target, similarly to the octahedron concept already studied. A common starting distance of $55m$ have been chosen.

Position 1	$[-55, 0, 0, 0, 0, 0]\ m$
Position 2	$[55, 0, 0, 0, 0, 0]\ m$
Position 3	$[0, -55, 0, 0, 0, 0]\ m$
Position 4	$[0, 55, 0, 0, 0, 0]\ m$
Position 5	$[0, 0, -55, 0, 0, 0]\ m$
Position 6	$[0, 0, 55, 0, 0, 0]\ m$

Table 7.2: Starting positions in LVLH frame.

As for the initial attitudes, it is assumed that the chaser starts from an orientation that allows it to perfectly point the chaser with a face, condition that coincides with

$\theta = 0°$, while for the target the initial attitude position, unaltered for all simulations, is provided in Euler angles:

$$[roll,\ pitch,\ yaw] = [20,\ 20, 20]^\circ$$

This orientation is random: in any case, starting from different positions, it will still be possible to distinguish favourable and unfavourable relative initial conditions looking at the performance of the controller.

7.1.1 Set of Simulator subsystems

Before showing the main results for each angular velocity, it is necessary to provide additional details of the simulator, as done for the near-field RVD study.

Simulation Parameters

As first, the simulation parameters used are set in Table 7.3.

dT_{simul}	1 s
Final time	200 s
Relative tolerance ($ode45$)	10^{-7}
Absolute tolerance ($ode45$)	10^{-7}

Table 7.3: Simulation parameters.

The time step simulation is the same of the previous phase: although the total maneuvering time is indeed shorter when compared to the observation motions one, it was preferred to keep dT_{simul} unchanged, due to the high computational cost required if this parameter is lowered.

Nonlinear MPC controller

The main parameters of the nonlinear Model Predictive Controller, resulting from tuning, are listed in Table 7.4. The control sample time is twice the simulation one: the control action is calculated and delivered every two simulation time-steps.

	$\boldsymbol{\omega_{BI} = 1.5°/s}$	$\boldsymbol{\omega_{BI} = 1.0°/s}$	$\boldsymbol{\omega_{BI} = 0.5°/s}$
Sample Time T_s	2 s		
Control Horizon m	4	4	3
Prediction Horizon p	24	24	20

Table 7.4: Nonlinear MPC parameters.

As can be noted, m and p are shorter than the MPC ones in the near-field RVD study: the computational cost, compared to linear MPC, is considerably higher. Despite this, these values proved to be suitable for accomplishing the mission. For the case of the lowest angular velocity, these values have even been slightly lowered, since a too long predictive horizon was not necessary.

As for the cost function (and therefore weights) and the constraints, the matter is a bit more delicate than the case with linear MPC, as explained in Chapter 5: functions must be provided.

Cost function

Keeping in mind that the state vector has increased, although the translation and attitude equations are decoupled, the cost function over the prediction horizon is common. Beside the classical standard quadratic MPC cost function, used to weigh the relative position error, the control action u and the control action rate Δu (for these latter the weights are quite zero for all simulations), another customized function can be provided. This latter is set in order to control the attitude dynamics: as already stated, the angle between the relative position vector and the line of sight θ defined in Eqn 3.8 must be small, thinking of a vision or LIDAR system that point at the target. This additional cost is made with the following function:

$$J_{LOS} = W_{LOS} \ |\theta|^2 \tag{7.1}$$

where W_{LOS} is another weight to be tuned. All weigh parameters are defined separately for each angular velocity.

Constraints

Constraints can be both linear or nonlinear. Linear constraints are imposed on control inputs and their variation. Considering the input vector as the union of the three angular accelerations α_x, α_y and α_z and the three translational acceleration u_x, u_y and u_z commanded, the linear constraints on the control inputs are provided in Table 7.5.

$\mathbf{u_{max}}$	$\big[[0.0051, 0.0028, 0.0028]\ rad/s^2,\ 0.0355\ [1,1,1]\ m/s^2\big]$
$\mathbf{u_{min}}$	$-\big[[0.0051, 0.0028, 0.0028]\ rad/s^2,\ 0.0355\ [1,1,1]\ m/s^2\big]$
$\mathbf{\Delta u_{max}}$	$\big[\ [\ ]\ ,\ 0.0355\ [1,1,1]\ m/s^2\big]$
$\mathbf{\Delta u_{min}}$	$-\big[\ [\ ]\ ,\ 0.0355\ [1,1,1]\ m/s^2\big]$

Table 7.5: NMPC linear constraints.

Nonlinear constraints over the prediction horizon must be specified with a function, in the form of inequalities:

$$\begin{cases} x^2 + y^2 + z^2 \geq R^2 \\ \sqrt{v_x^2 + v_y^2 + v_z^2} \leq V_{max} \end{cases}$$

As for the first one, R is set to $24.8m$: a margin of $20cm$ is kept to allow eventual oscillations when the holding point has been reached, in the box within the corridor. As for the second one, V_{max} is the translational velocity of the holding point to be reached, calculated through Eqn 3.5, multiplied by a factor greater than one, to ensure the capability to follow the reference.
For both these nonlinear constraints, a slack variable of 0.2 is set, in order to slightly relax them, not being in the immediate proximity of the target yet.

Noise and Extended Kalman Filter

As already mentioned in Chapter 5, the state vector contains relative position and velocity, the angular velocity between body and inertial and the quaternion, but only the position and quaternion (through Euler angles) are measured. Also in this case, the errors of the sensors, not being present, were simulated by adding a random normal

distribution on the true values. The standard deviations, especially as regards the relative position, become smaller as the chaser gets closer to the target, as it should be. Instead, as regards the attitude sensors, this improvement is not linked to the relative position, so the standard deviation remains quite constant.

distance from the target	$55m$	$25m$
σ_{pos}	$0.1m$	$0.07m$
σ_{ang}	$0.06°$	$0.05°$

Table 7.6: Standard deviations for relative positions and Euler angles.

Table 7.6 gives the standard deviations adopted at the initial and final distances: for intermediate distances, the standard deviation varies linearly between the indicated values, as explained in Subsection 5.2.5.

To manage these measures, in combination with the estimates of the states, an Extended Kalman filter is used. For its initialization, the matrices Q and R must be provided (Table 7.7), basing on process and measurement noise covariances respectively.

Q_{EKF}	$diag\big[10^{-3}\ [1,1,1,1,1,1,0.1,0.1,0.1,0.1,0.1,0.1,0.1]\big]$
R_{EKF}	$diag\big[\sigma^2_{pos\ @55m}\ [1,1,1],\ 10^{-3}\ [1,1,1,1]\big]$

Table 7.7: Extended Kalman filter specifics.

Note that, while the measurement noises are provided on Euler angles, the quaternions are part of the measuring vector $y = [x, y, z, q_1, q_2, q_3, q_4]$: at each step dT_{simul}, to simulate the measurement, the quaternions are transformed into Euler angles in order to add the measurement errors, and then transformed back into quaternions to be managed by the filter.

Alongside the EKF action, as explained in Subsection 5.2.5, some measurement errors are imposed on the initial estimate of the holding point coordinates (with the same uncertainty given for measures of relative position) and on the current target angular velocity and inertia information, to simulate the real uncertainties concerning

their knowledge. These errors are given on a percentage basis (see Table 7.8), in the form of standard deviations:

$$\omega_{real} = (1 + \sigma_{\omega,\%} a)\ \omega_{perfect}$$

where a is a random value. As regard the inertia, an initial estimate is given through an expression similar to the previous one, and this estimate is converged towards the real value as the chaser approaches the target.

distance from the target	$55m$	$25m$
$\sigma_{\omega,\%}$	5	1
$\sigma_{In,\%}$	0.2	0.15

Table 7.8: Standard deviations on target motion estimate.

$\sigma_{In,\%}$ is very low due to this mission case: Landsat7 properties are well-known. Also in this case, there is a linear variation between initial and final points.

7.1.2 Simulation results for the first phase

The cases are ordered according to angular velocities, in descending order. For the first case, the one with maximum angular velocity, the weights have been kept constant to see the behaviour of the controller only by changing the initial starting position. For the other two cases, however, the weights have been arranged in order to arrive and stay within the box and within the chosen final time of $200s$.

Case A: $1.5°/s$

Weights used for this first case are given in Table 7.9.

	Sim 1	*Sim 2*	*Sim 3*	*Sim 4*	*Sim 5*	*Sim 6*
$\mathbf{W_{pos}}$	2500 $[1]^{3\times1}$	2500 $[1]^{3\times1}$	2500 $[1]^{3\times1}$	2500 $[1]^{3\times1}$	2500 $[1]^{3\times1}$	2500 $[1]^{3\times1}$
$\mathbf{W_u}$	2600 $[1]^{6\times1}$	2600 $[1]^{6\times1}$	2600 $[1]^{6\times1}$	2600 $[1]^{6\times1}$	2600 $[1]^{6\times1}$	2600 $[1]^{6\times1}$
$\mathbf{W_{LOS}}$	1200	1200	1200	1200	1200	1200

Table 7.9: NLMPC weights for case A: target angular velocity of $1.5°/s$.

As regards the maximum relative velocity, this is set as 2 times the translational velocity of the holding point (Eqn 3.5), in the orbital reference:

$$V_{max} = 2 \times 0.6791\ m/s = 1.3581\ m/s$$

Main results are presented in figures from 7.5 to 7.10.

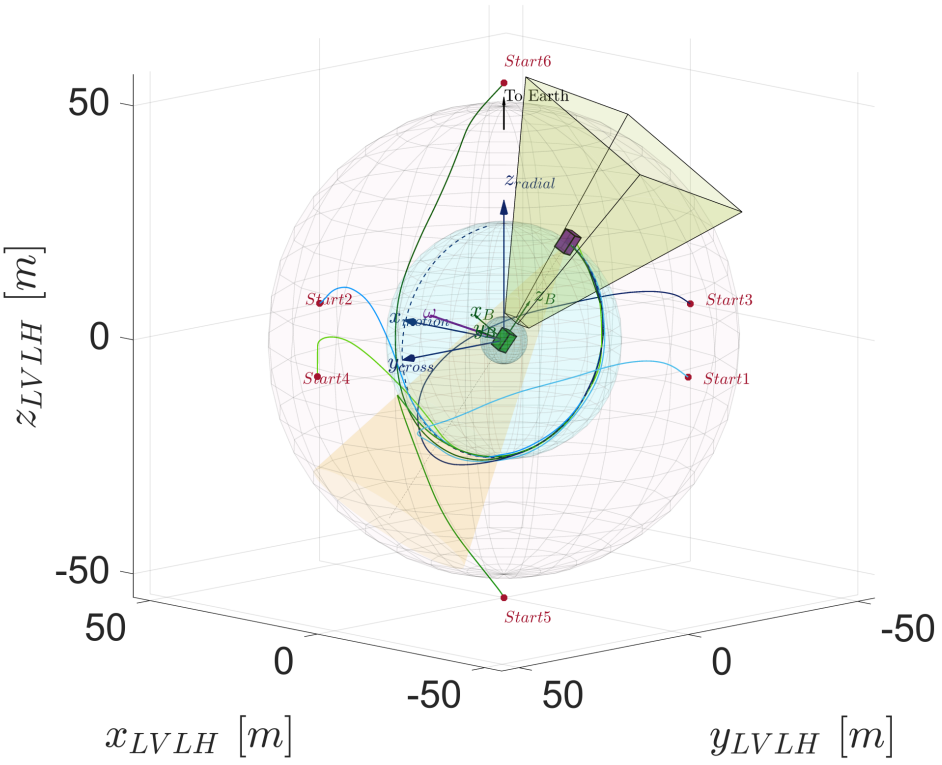


Figure 7.5: 3D trajectories in LVLH frame, starting from 6 different points. The yellow cone represents the sensor FOV, the green cone is the corridor approach. The nested KOS, starting from the outermost one, have a radius of 50m, 25m and 5m respectively. The dotted line on the 25m radius sphere represents the trajectory of the first holding point inside the approaching corridor.

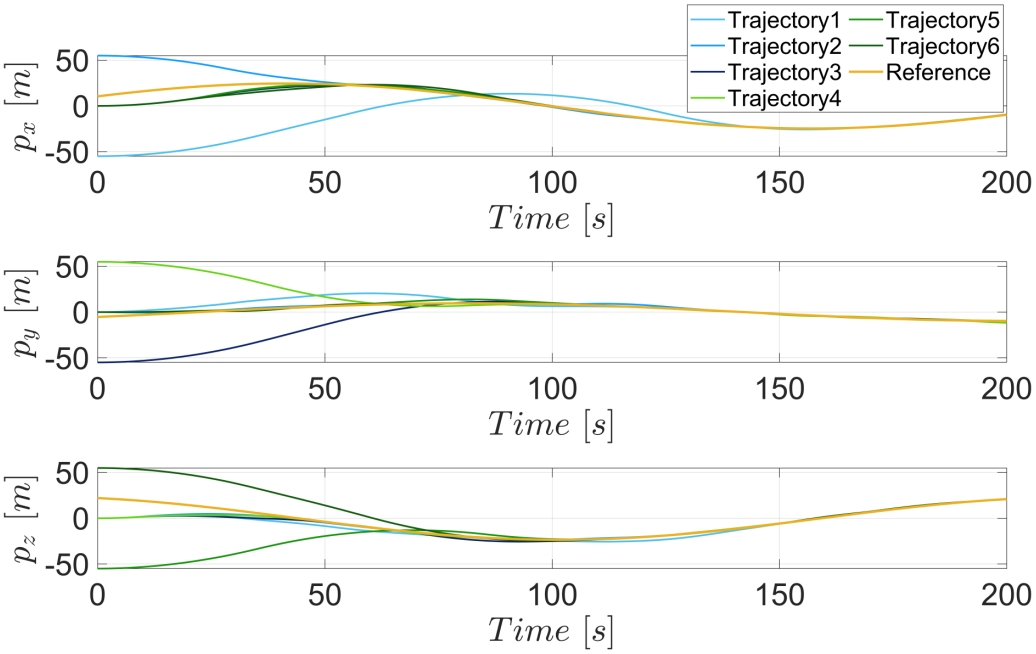


Figure 7.6: Relative positions and holding point reference trajectory in LVLH frame

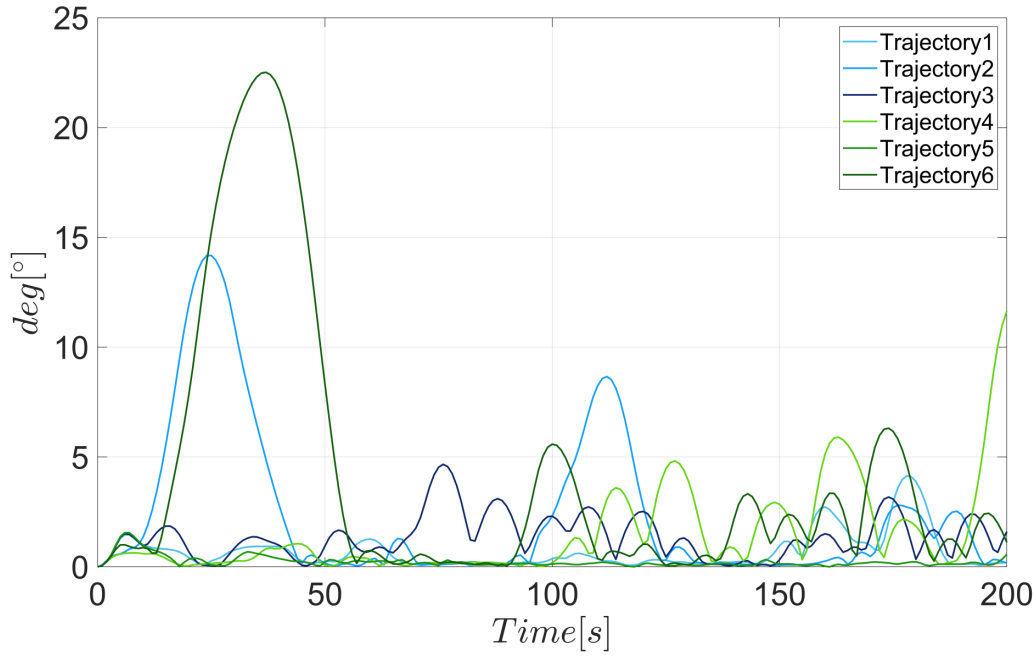


Figure 7.7: Angle θ between LOS and relative position vector.

As can be seen in Figure 7.8, simulation 4 does not end inside the box, unlike the others: this combination of weights, due to the angular velocity, is therefore not

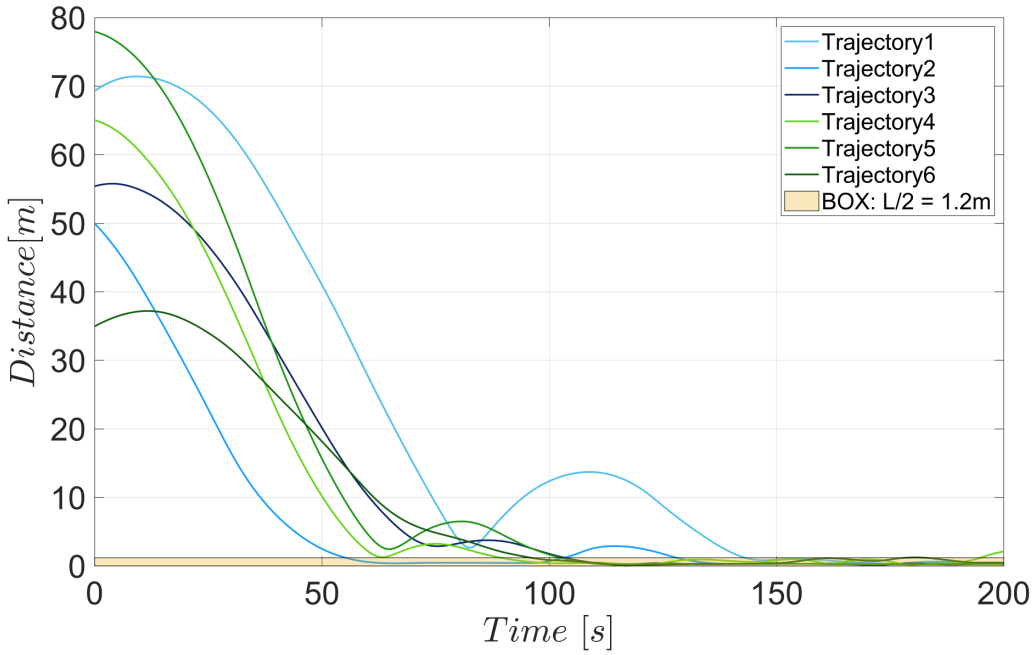


Figure 7.8: Distances from the holding point.

suitable to meet the requirements.

It is possible to see in Figure 7.7 an initially high value of θ for the sixth simulation: given the initial relative position, the chaser accelerates to cancel the relative distance, thus reaching such velocities as to imply difficulties with regard to the attitude control. Table 7.10 summarizes the main results. T_{box} represents the time it takes to stably enter the box, with reference to Figure 7.8.

	Sim 1	*Sim 2*	*Sim 3*	*Sim 4*	*Sim 5*	*Sim 6*
T_{box} $[s]$	145	130	105	–	104	97
Δv_{transl} $[m/s]$	11.7315	8.2821	9.6815	11.0378	9.0155	9.6123
m_{prop}^{transl} $[kg]$	34.19	24.14	28.22	32.17	26.28	28.01
m_{prop}^{att} $[kg]$	2.39	2.64	3.66	3.11	1.52	3.66
m_{prop}^{TOT} $[kg]$	36.58	26.78	31.88	35.28	27.79	31.67

Table 7.10: Principal results for case A: target angular velocity of $1.5°/s$.

Before considering the second case, two considerations about how the constraints and the EKF filter work are given: in Figure 7.9, as one can see, the collision avoidance

requirement is met, while in Figure 7.10 a slight decrease in the difference between estimates and measures, proving the convergence of the filter, can be noted.

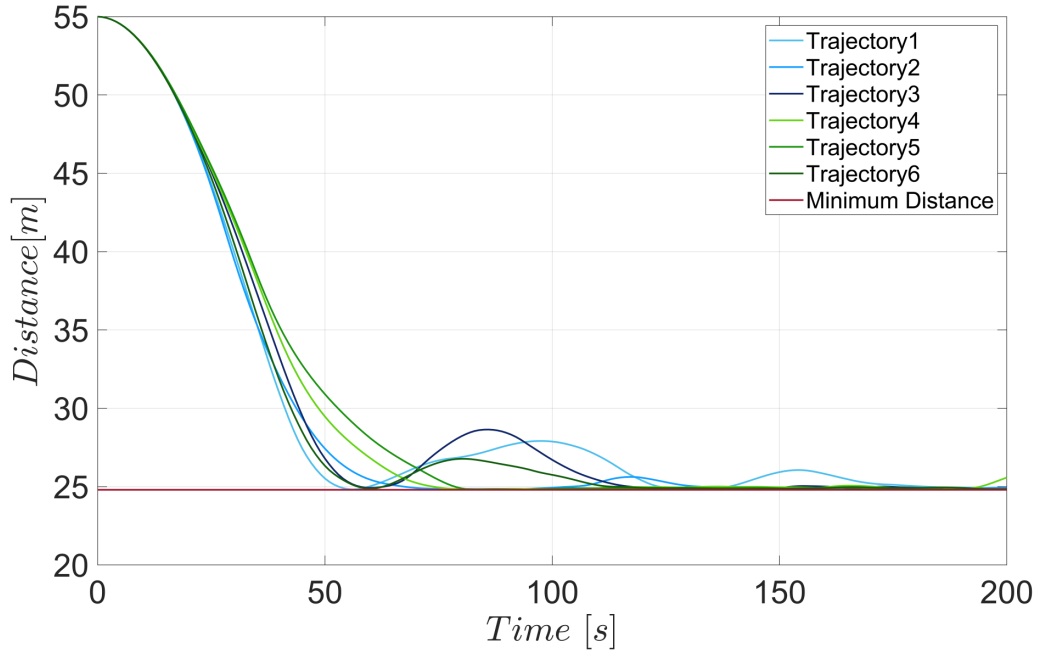


Figure 7.9: Distances from the target.

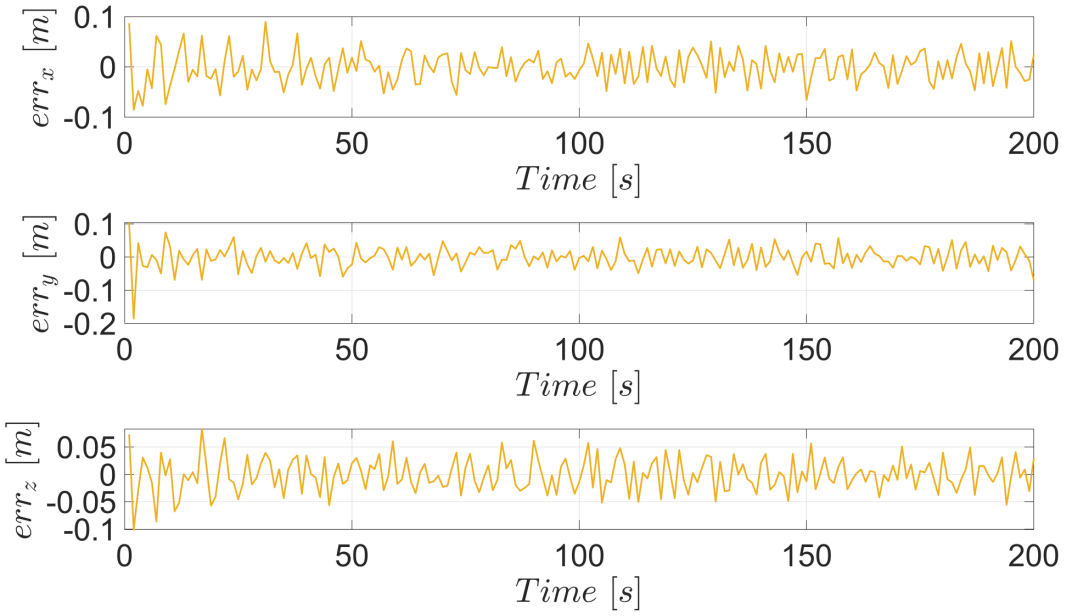


Figure 7.10: EKF on position: differences between estimated and measured values.

Case B: $1.0°/s$

Weights used are given in Table 7.11.

	Sim 1	*Sim 2*	*Sim 3*	*Sim 4*	*Sim 5*	*Sim 6*
$\mathbf{W_{pos}}$	2500 $[1]^{3\times1}$	2800 $[1]^{3\times1}$	1500 $[1]^{3\times1}$	2500 $[1]^{3\times1}$	1500 $[1]^{3\times1}$	800 $[1]^{3\times1}$
$\mathbf{W_u}$	2600 $[1]^{6\times1}$	2600 $[1]^{6\times1}$	2600 $[1]^{6\times1}$	2600 $[1]^{6\times1}$	2600 $[1]^{6\times1}$	2600 $[1]^{6\times1}$
$\mathbf{W_{LOS}}$	1200	2000	1500	1200	1600	1500

Table 7.11: NLMPC weights for case B: target angular velocity of $1.0°/s$.

As specified, in this second case the weights have been adapted to ensure the permanence inside the box (Figure 7.14). As for the maximum relative velocity, this is set as 2 times the translational velocity of the holding point (Eqn 3.5):

$$V_{max} = 2 \times 0.4609\ m/s = 0.9219\ m/s$$

Main results are presented in figures from 7.11 to 7.14.

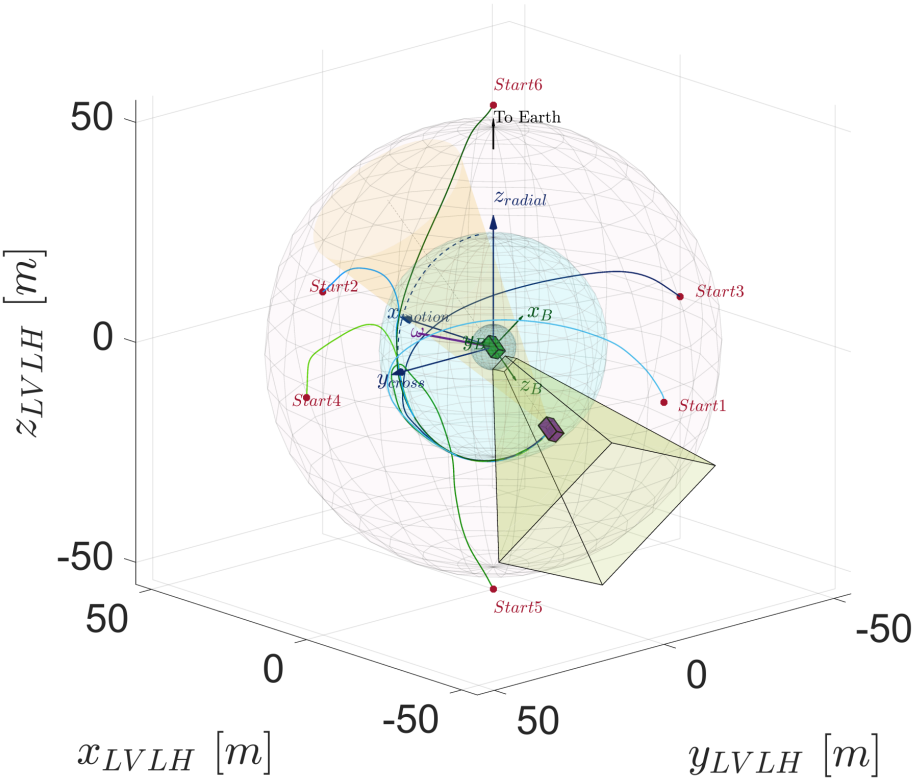


Figure 7.11: 3D trajectories in LVLH frame (see Figure 7.5 for a complete description).

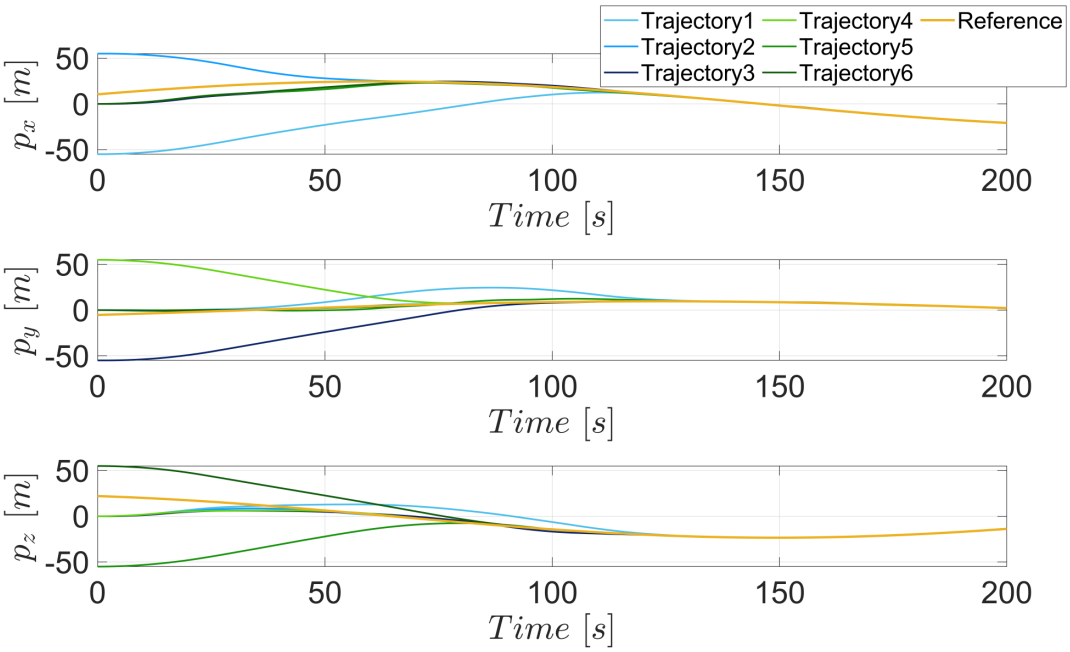


Figure 7.12: Relative positions and holding point reference trajectory in LVLH frame.

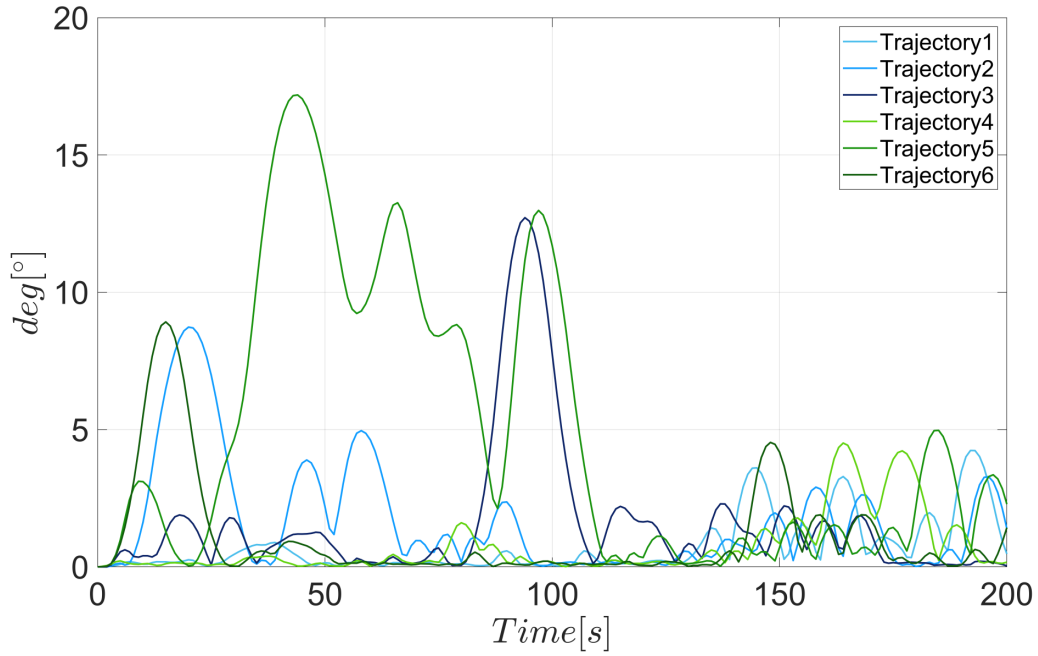


Figure 7.13: Angle θ between LOS and relative position vector.

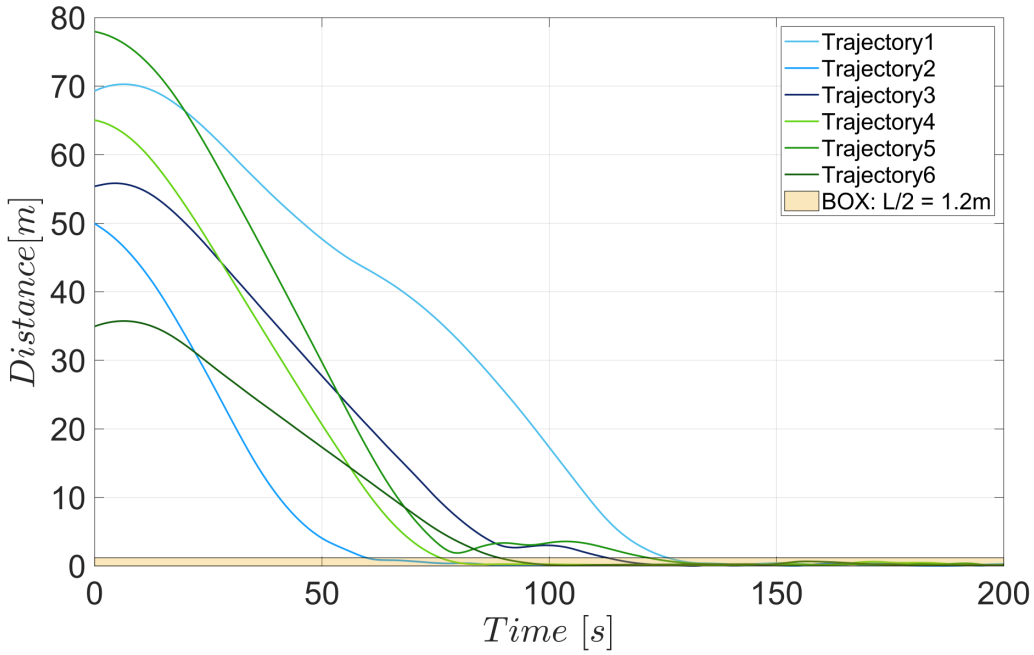


Figure 7.14: Distances from the holding point.

Looking at Figure 7.13, the same considerations made before for the sixth simulation can be made now for the fifth: an excessive chaser translational speed results in higher θ, and therefore a greater use of attitude thrusters (see m_{prop}^{att} in Table 7.12). As for translational delta-vs, in general, it is possible to notice a decrease compared to the first case with higher angular velocity, as it should be.

	Sim 1	*Sim 2*	*Sim 3*	*Sim 4*	*Sim 5*	*Sim 6*
T_{box} $[s]$	128	61	114	78	123	90
Δv_{transl} $[m/s]$	9.1405	6.6406	7.1345	6.5385	8.4718	4.9823
m_{prop}^{transl} $[kg]$	26.64	19.35	20.79	19.06	24.69	14.52
m_{prop}^{att} $[kg]$	2.43	3.11	2.99	2.15	3.96	2.02
m_{prop}^{TOT} $[kg]$	29.07	22.47	23.79	21.21	28.65	16.54

Table 7.12: Principal results for case B: target angular velocity of $1.0°/s$.

Case C: $0.5°/s$

Weights used for the last case are given in Table 7.13.

	Sim 1	*Sim 2*	*Sim 3*	*Sim 4*	*Sim 5*	*Sim 6*
$\mathbf{W_{pos}}$	1500 $[1]^{3\times1}$	400 $[1]^{3\times1}$	1500 $[1]^{3\times1}$	1500 $[1]^{3\times1}$	400 $[1]^{3\times1}$	1500 $[1]^{3\times1}$
$\mathbf{W_u}$	2600 $[1]^{6\times1}$	2600 $[1]^{6\times1}$	2500 $[1]^{6\times1}$	2500 $[1]^{6\times1}$	2600 $[1]^{6\times1}$	2600 $[1]^{6\times1}$
$\mathbf{W_{LOS}}$	2000	1300	2000	2000	1600	2000

Table 7.13: NLMPC weights for case C: target angular velocity of $0.5°/s$.

As regards the maximum relative velocity, this is set as 3 times the translational velocity of the holding point (Eqn 3.5). A lower maximum speed would not have allowed to appreciate the decline in the translational delta-v required with this minor angular velocity: the time to reach the holding point would have lengthened considerably.

$$V_{max} = 3 \times 0.2429\ m/s = 0.7286\ m/s$$

Main results are presented in figures from 7.15 to 7.18.

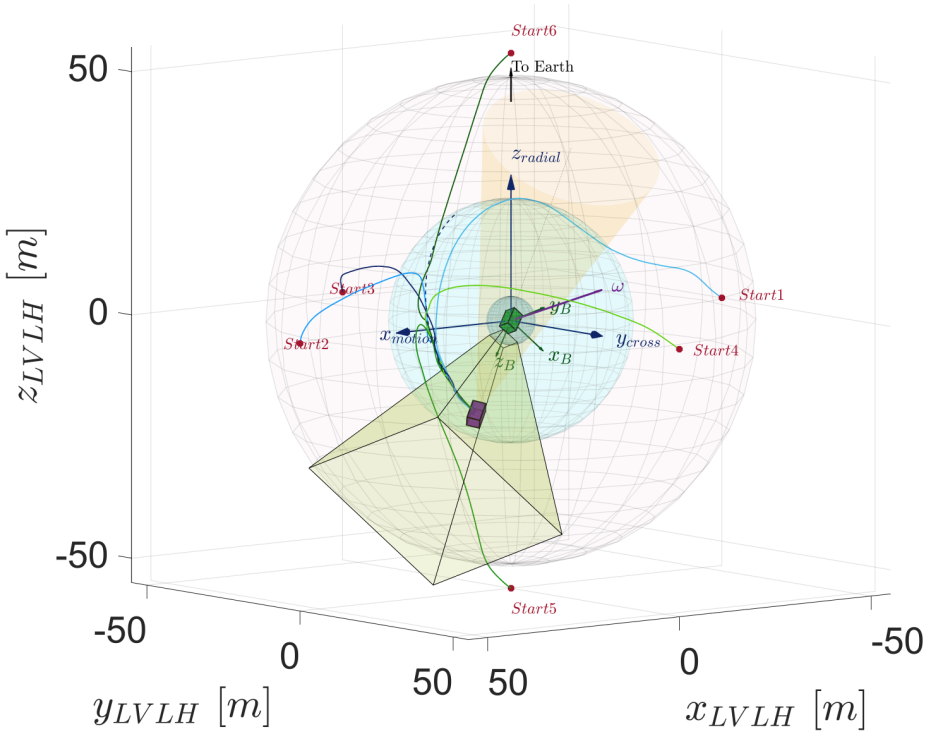


Figure 7.15: 3D trajectories in LVLH frame (see Figure 7.5 for a complete description).

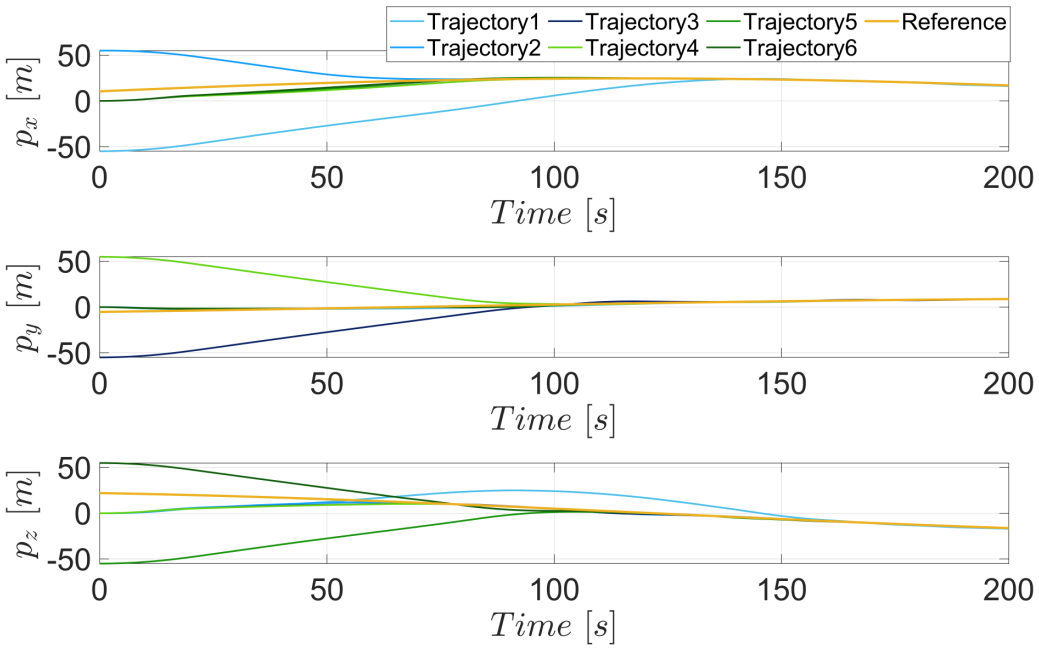


Figure 7.16: Relative positions and holding point reference trajectory in LVLH frame.

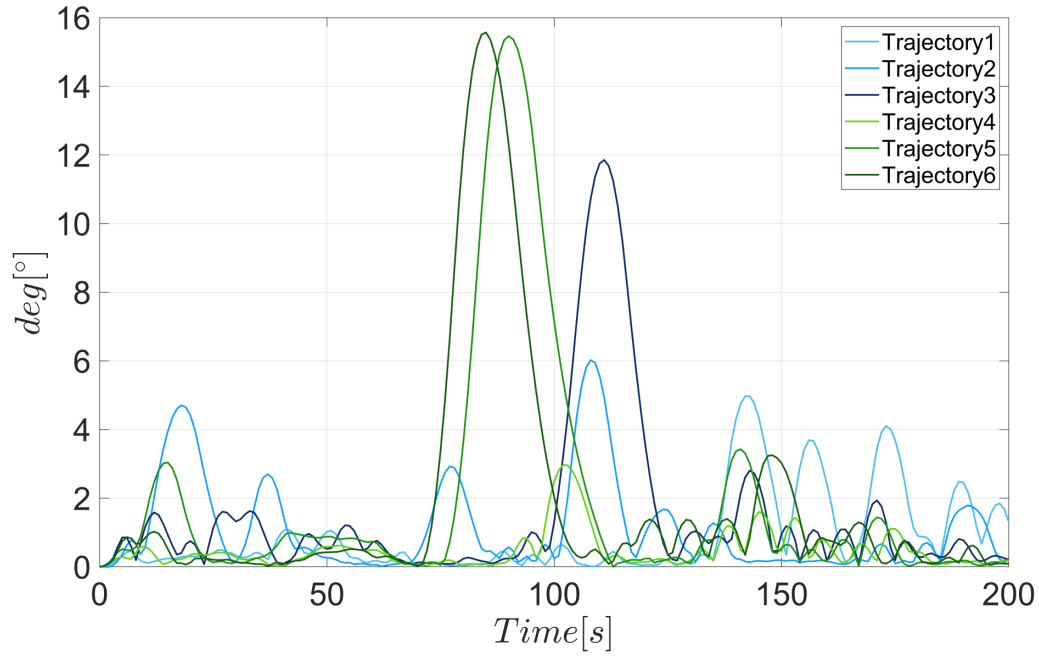


Figure 7.17: Angle θ between LOS and relative position vector.

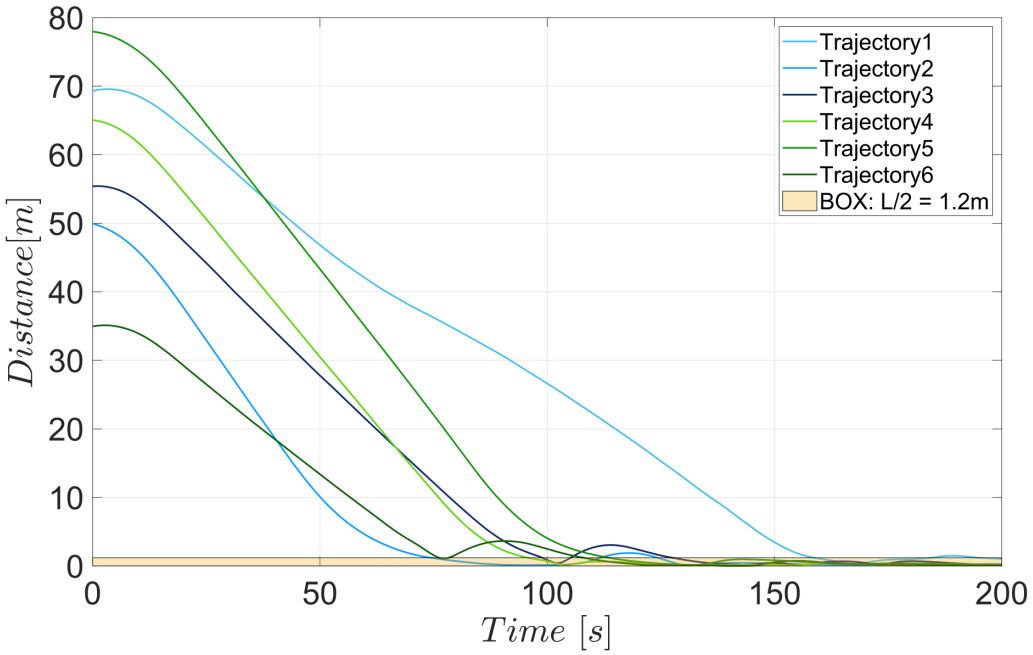


Figure 7.18: Distances from the holding point.

Finally, principal results are given in Table 7.14. What mostly influences this latter case is the limit on the maximum achievable translational velocity: for simulation 1, which starts from the most unfavourable point, the final time imposed is borderline.

	Sim 1	*Sim 2*	*Sim 3*	*Sim 4*	*Sim 5*	*Sim 6*
T_{box} $[s]$	160	126	129	97	115	109
Δv_{transl} $[m/s]$	6.8518	5.1372	6.3837	4.7336	4.9853	5.1002
m_{prop}^{transl} $[kg]$	19.97	14.97	18.61	13.80	14.53	14.86
m_{prop}^{att} $[kg]$	2.58	2.53	3.26	1.92	2.48	2.78
m_{prop}^{TOT} $[kg]$	22.55	17.50	21.87	15.72	17.01	17.64

Table 7.14: Principal results for case C: target angular velocity of $0.5°/s$.

Final considerations

Summing up, the following considerations can be made about this first phase.

- As the angular velocity of the target decreases, the overall maneuver becomes less onerous in terms of propellant consumption and therefore easier to be tracked by the system.
- Concerning the attitude control, before reaching the holding point, there is little dependence on the angular velocity of the target, since the amount of propellant used has only slightly decreased overall. What most influences, instead, is the combination of the weights adopted.
- Generally, different initial positions must correspond to different weights, based not only on the initial position, but also on the target initial attitude and the motion – favourable or unfavourable – of the final point to be reached. The following simulations, in any case, were performed to study the behaviour of the controller: in reality, in fact, it is possible to wait for the best condition to start the maneuver.

7.2 Final constrained phase

The second and last phase coincides with the achievement of the final holding point on the $5m$ radius sphere, always remaining within the approaching cone and starting from the first box at a distance of 25m. The coordinates of the final point in the target body reference are the following:

$$pos_{HP2} = [0,\ 0,\ 5]^T\ m$$

Starting from a position inside the cone, with a relative translational speed equal to that of the first holding point, the initial orientation of the target in the LVLH frame is quite indifferent. In any case, the same initial orientation of the previous phase was chosen:

$$[roll,\ pitch,\ yaw] = [20,\ 20, 20]^\circ$$

As previously, the reference trajectory coincides with the motion of the $5m$ - holding point in the orbiting reference, created online at each control step propagating the estimate of the current position of this point throughout the prediction horizon and with the current information about inertia and angular velocity of the target.

7.2.1 Set of Simulator subsystems

As for the first phase, it is necessary to define some details of the simulator.

Simulation Parameters

The simulation parameters used are listed in Table 7.15.

dT_{simul}	1 s
Final time	150 s
Relative tolerance ($ode45$)	10^{-7}
Absolute tolerance ($ode45$)	10^{-7}

Table 7.15: Simulation parameters.

The simulation time step is the same of the previous phase, but the final time is lower, being the distance to be covered shorter.

Nonlinear MPC controller

The main parameters of the nonlinear Model Predictive Controller, resulting from tuning, are listed in Table 7.16. The values have been used for all three analysed cases of angular velocity of the target. The control time interval is now equal to the simulation one, to control the execution of the maneuver more accurately.

Sample Time T_s	1 s
Control Horizon m	3
Prediction Horizon p	15

Table 7.16: Nonlinear MPC parameters.

As can be noted, m and p are shorter than the ones of the previous case: the computational cost, due to the additional constraints, is higher than before. Despite this, the selected values have led to satisfactory results.

Cost function

This time, the components added to the quadratic cost function, which controls the relative position error and the control action u, are two. Beside the condition of the line of sight regarding the attitude, already expressed in the previous phase, an additional weight has been included on the distance from the capture axis during the approach: by calling ϵ the 3×1 error-vector between the current relative position and its respective projection on the capture axis, the cost term can be written as follows:

$$J_{CA} = \epsilon^T \; diag(W_{CA}) \; \epsilon$$

where $diag(W_{CA})$ is a 3×3 diagonal weight matrix. This term allows the chaser to approach along the capture axis, helping to comply with the constraint on the approaching cone. All the weights used are given below, being unchanged for the three simulations.

$\mathbf{W_{pos}}$	200 $[1]^{3\times1}$ for $d \geq 7m$ 800 $[1]^{3\times1}$ for $d < 7m$
$\mathbf{W_u}$	3000 $[1]^{6\times1}$
$\mathbf{W_{LOS}}$	800
$\mathbf{W_{CA}}$	Linear variation: $5000[1]^{3\times1}$ at $d = 25m \rightarrow 800[1]^{3\times1}$ at $d = 5m$

Table 7.17: NLMPC weights for the last phase. d represents the distance from the target.

Constraints

The linear constraints on the input vector are the same as in the first phase, expressed in Table 7.5. Beside the nonlinear constraints already presented, another one, namely the constraint for the chaser to remain inside the approaching corridor, is implemented. All the nonlinear constraints are listed below.

$$\begin{cases} x^2 + y^2 + z^2 \geq R^2 \\ \sqrt{v_x^2 + v_y^2 + v_z^2} \leq V_{max} \\ \begin{cases} x_B - x_{off} - m(z_B - 5) \leq 0 \\ -x_B - x_{off} - m(z_B - 5) \leq 0 \\ y_B - y_{off} - m(z_B - 5) \leq 0 \\ -y_B - y_{off} - m(z_B - 5) \leq 0 \end{cases} \end{cases}$$

R is set to $4.9m$: as done for the previous case, a margin of $10cm$ is kept to allow eventual oscillations when the final holding point has been reached.

As for the second constraint, V_{max} is calculated as the translational velocity of the final holding point to be reached, using Eqn 3.5, multiplied by a factor that varies linearly from 5 at a relative distance $d = 25m$ to 2 at $d = 5m$, to control the approach to the target. This constraint, as regards the case with $\omega_{BI} = 0.5°/s$, is set as the case with $\omega_{BI} = 1.0°/s$: limiting the translational speed too much would not have allowed to reach the holding point in the required time ($150s$) and, above all, the delta-v would have been greater.

As for the last group of constraints, $m = \tan 15°$ and $x_{off} = y_{off} = 1.5m$ is half the side of the cone shrinkage shown in Figure 7.1. These conditions are nonlinear because they are referred to the body frame, thus implying a rotation during the calculations. All these nonlinear constraints are set as hard constraints.

Noise and Extended Kalman Filter

All the considerations made for the previous phase also apply in this second phase: therefore, the standard deviations with regard to the measurements of relative position and attitude are directly provided (Table 7.18).

distance from the target	$25m$	$5m$
σ_{pos}	$0.07m$	$0.01m$
σ_{ang}	$0.05°$	$0.05°$

Table 7.18: Standard deviations for relative positions and Euler angles.

As for the Extended Kalman filter, instead, the matrices Q and R provided in Table 7.19.

Q_{EKF}	$diag\big[10^{-2}\ [1,1,1,1,1,1,0.1,0.1,0.1,0.1,0.1,0.1,0.1]\big]$
R_{EKF}	$diag\big[\sigma^2_{pos\ @25m}\ [1,1,1],\ 10^{-3}\ [1,1,1,1]\big]$

Table 7.19: Extended Kalman filter specifics.

Approaching the target, the measurements provide a better estimation of the relative pose: it is recommended, therefore, to rely more on the measuring system than the process (the values of the Q matrix have increased by an order of magnitude).
Finally, as regards the estimation of the motion of the target, the values reported in Table 7.20 were used, always according to a linear variation.

distance from the target	$25m$	$5m$
$\sigma_{\omega,\%}$	1	0.5
$\sigma_{In,\%}$	0.15	0.10

Table 7.20: Standard deviations on target motion estimate.

7.2.2 Simulation results for the final phase

This section provides results for all three angular velocities. Before showing the comparison between the three cases, to give a global view of the problem, two three-dimensional plots, in the orbiting system and in the body system, are provided in Figure 7.19 and 7.20 respectively. These plots are referred to the case with maximum angular velocity. Figure 7.21, for the same case, demonstrates the action of the EKF, showing how the difference between estimates and measures, with regard to relative position, decreases over time.

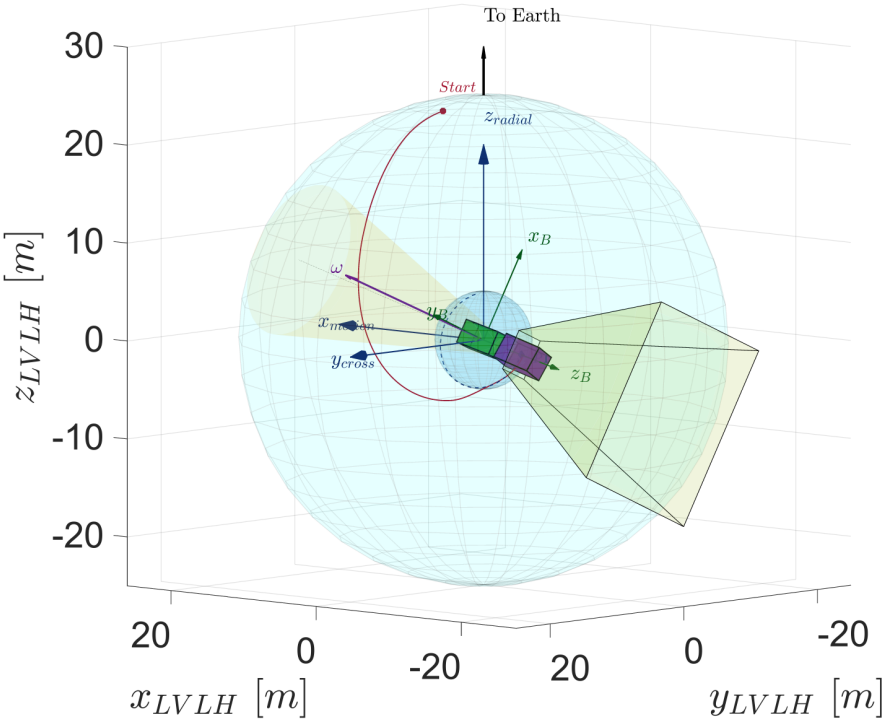


Figure 7.19: 3D plot of the approaching trajectory followed by the chaser for $\omega_{BI} = 1.5°$, in LVLH frame.

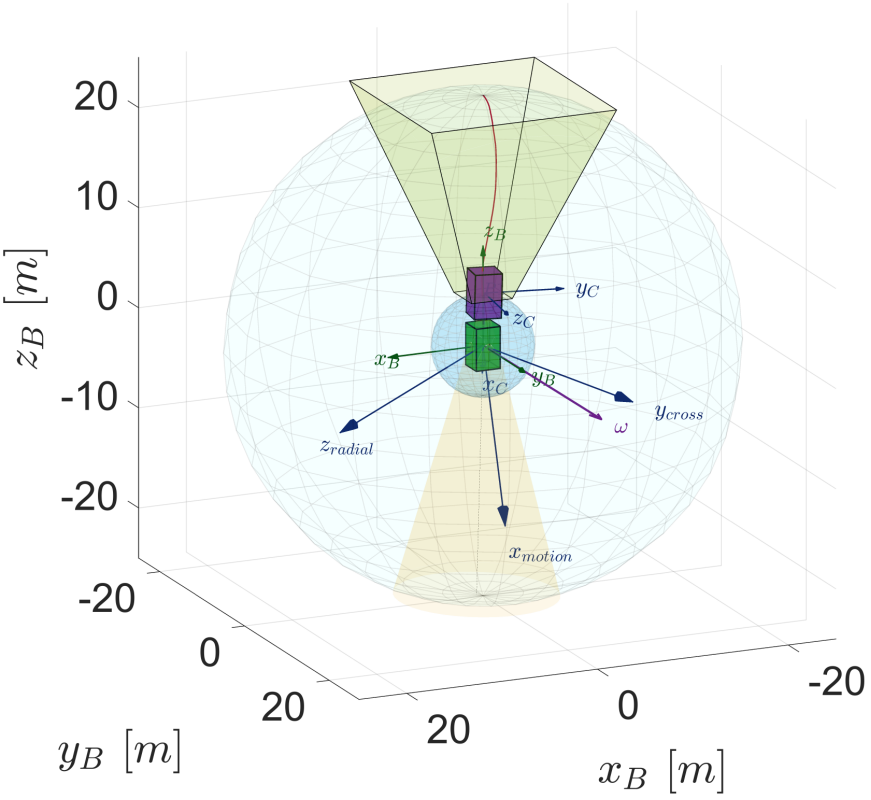


Figure 7.20: 3D plot of the approaching trajectory followed by the chaser for $\omega_{BI} = 1.5°$, in body frame.

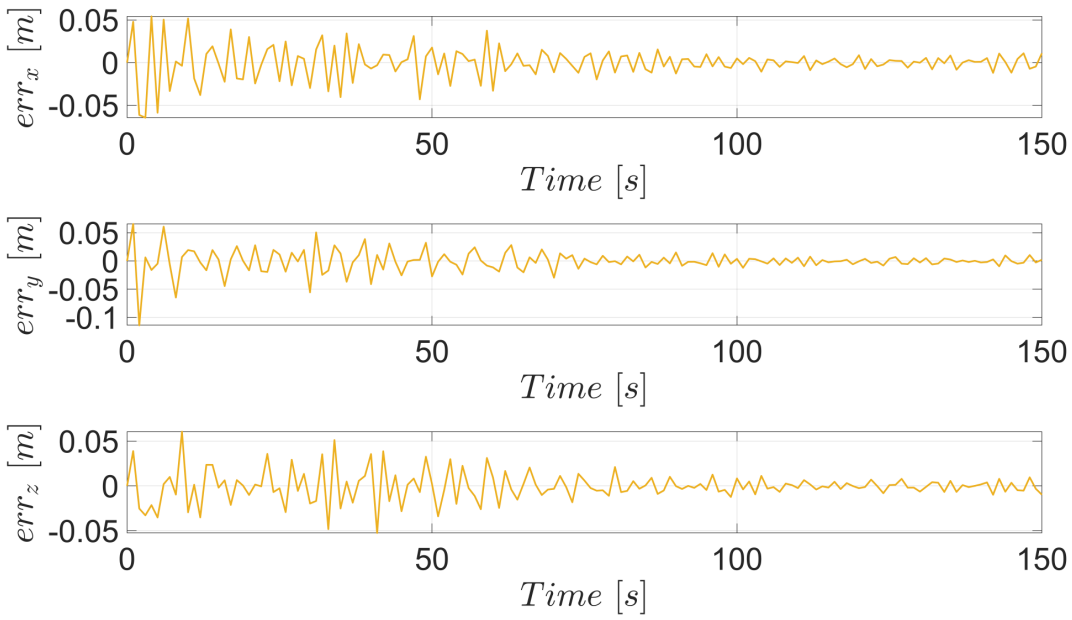


Figure 7.21: EKF on position: differences between estimated and measured values.

A comparison between the three cases is given. Figures 7.22, 7.23 and 7.24 show the approaching trajectory in the body frame.

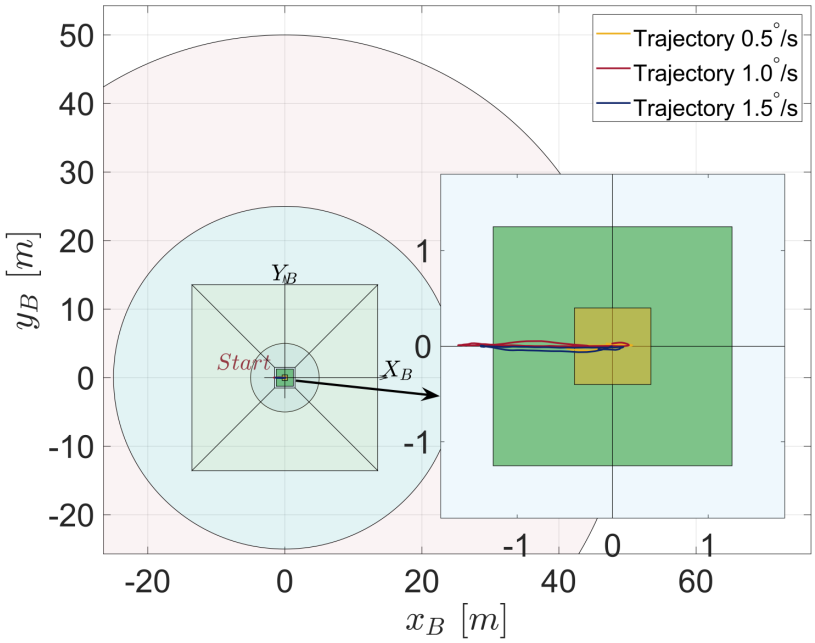


Figure 7.22: Approaching trajectories: x_B-y_B plane.

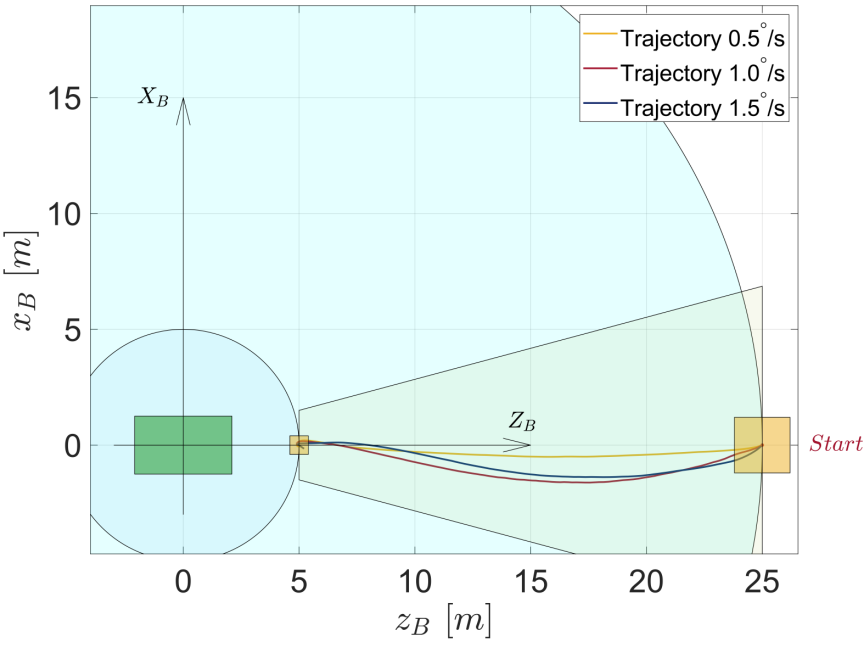


Figure 7.23: Approaching trajectories: z_B-x_B plane.

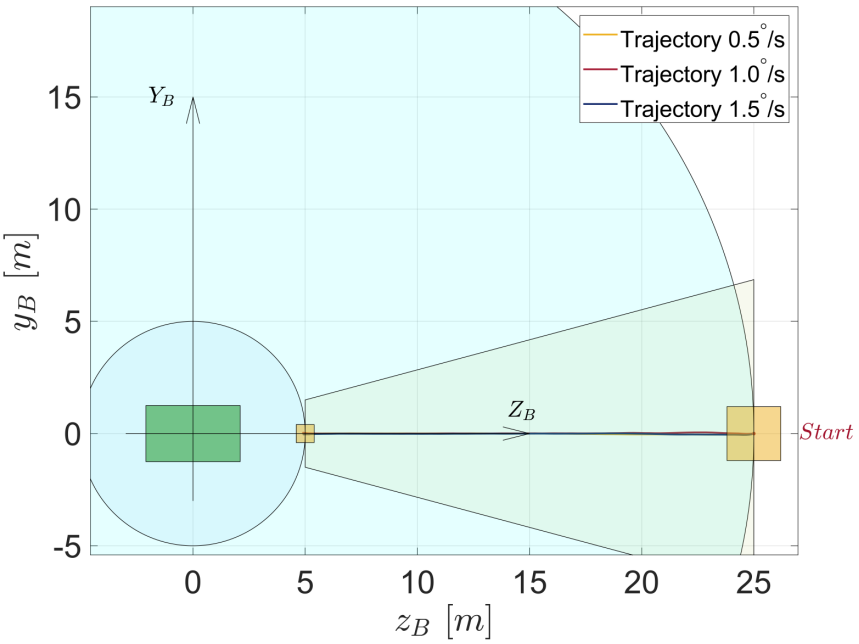


Figure 7.24: Approaching trajectories: z_B-y_B plane.

The collision avoidance constraint, here set as hard, is respected during the approach, as Figure 7.25 shows.

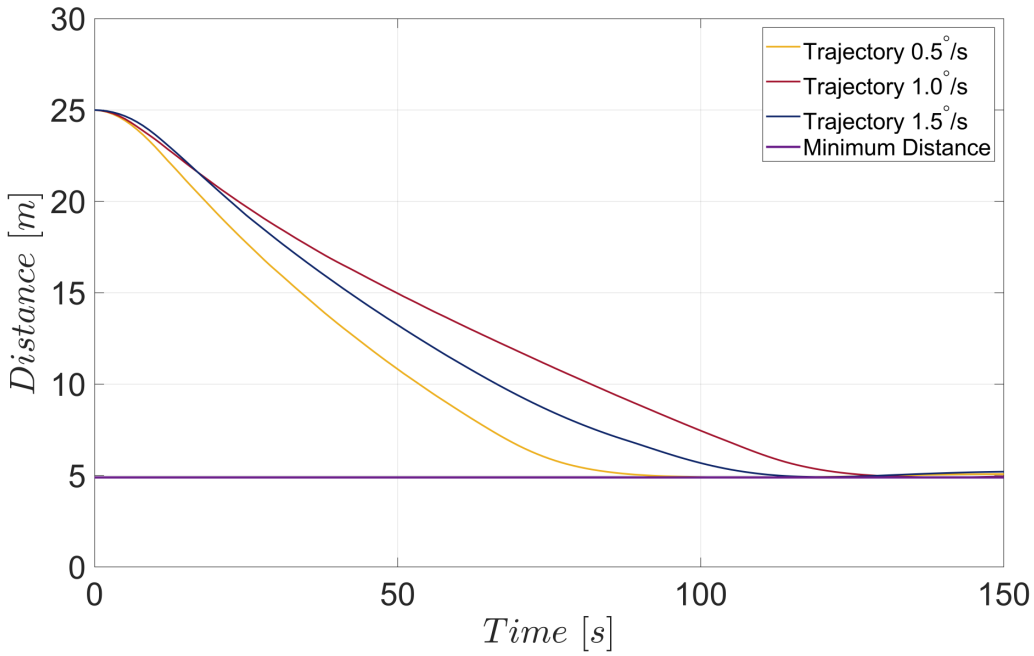


Figure 7.25: Distances from the target.

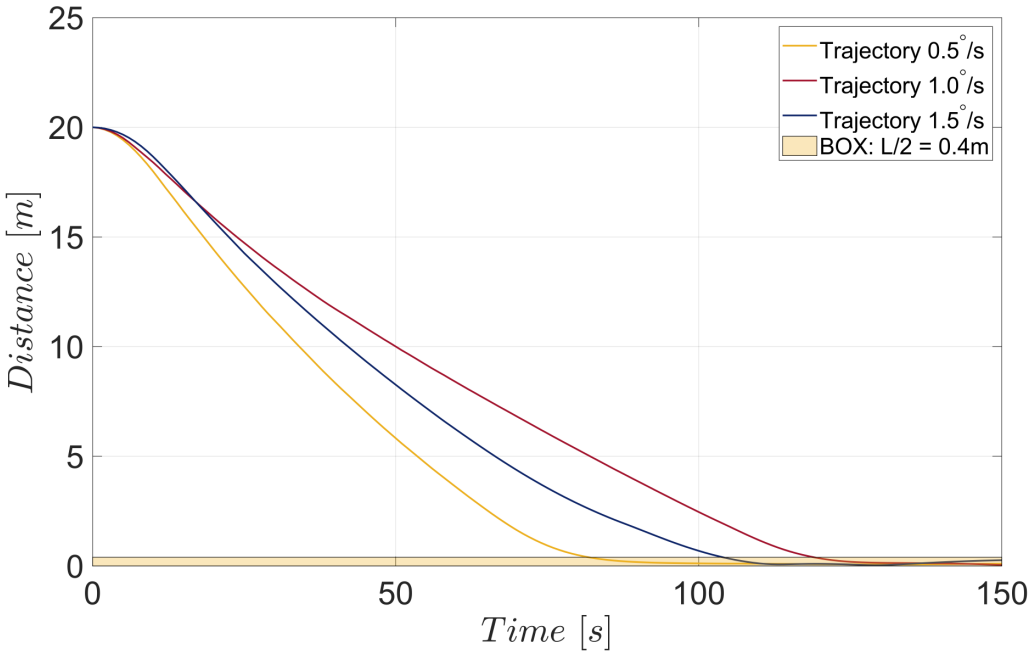


Figure 7.26: Distances from the holding point.

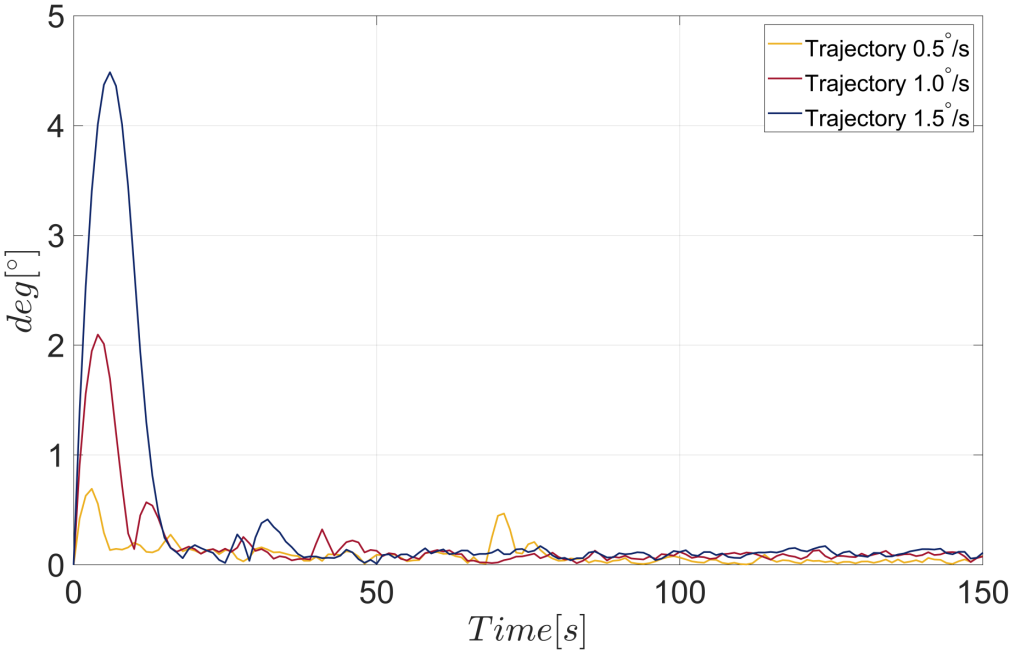


Figure 7.27: Angle θ between LOS and relative position vector.

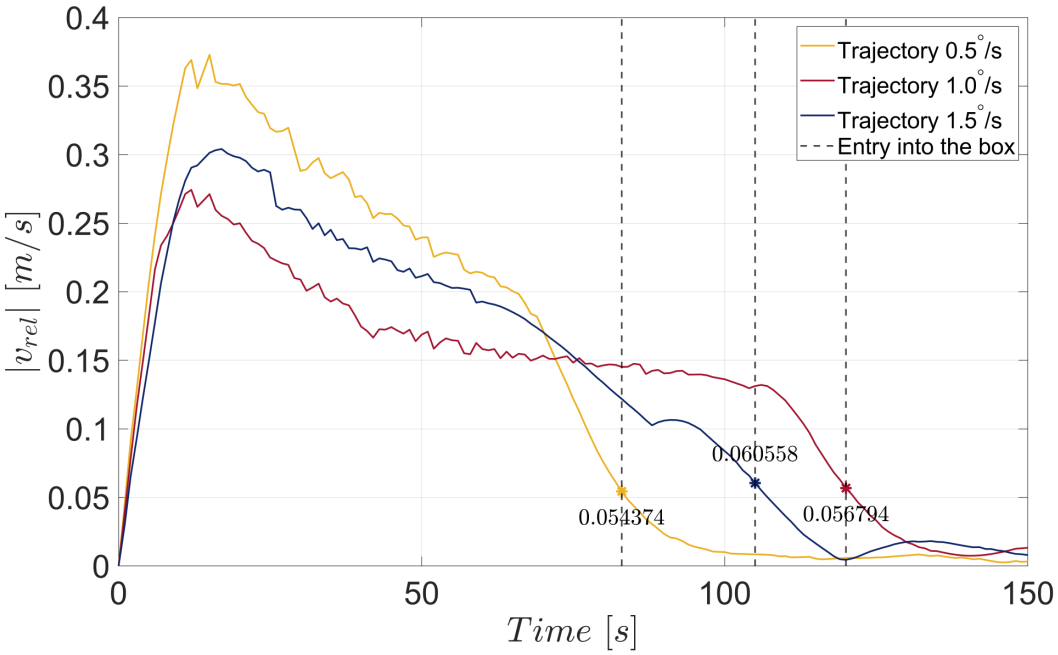


Figure 7.28: Translational velocities referred to the body frame.

Finally, the main results are listed in Table 7.21. It is possible to notice, as the rotation velocity of the target decreases, a reduction in the energy required, for both translation and rotation. For the latter, in particular, it is possible to notice how the weights found allow a very satisfactory control, looking at the profiles of the angle θ in Figure 7.27.

As for the velocities, it is possible to notice in Figure 7.28 and in Table 7.21 how at the entrance of the box the approaching velocity V_{box} is around $0.05 m/s$ for all cases, in accordance with the typical values for this RVD problem[43], considering the very close distance from the target.

Since the constraints on the translational velocity are the same for $0.5°/s$ and $1.0°/s$, it is clear that the time required to reach the box is shorter for the slower angular velocity case. The constraints on translational velocities, therefore, mainly dictate this time T_{box}. In any case, as shown in Figure 7.26, all the three simulations lead stably within the final region, where the capture phase through a robotic arm can be initialized.

	$\boldsymbol{\omega_{BI} = 1.5°/s}$	$\boldsymbol{\omega_{BI} = 1.0°/s}$	$\boldsymbol{\omega_{BI} = 0.5°/s}$
T_{box} $[s]$	105	120	83
V_{box} $[m/s]$	0.0606	0.0568	0.0544
Δv_{transl} $[m/s]$	2.5693	2.0634	1.4134
m_{prop}^{transl} $[kg]$	7.49	6.01	4.12
m_{prop}^{att} $[kg]$	0.82	0.76	0.68
m_{prop}^{TOT} $[kg]$	8.31	6.77	4.80

Table 7.21: Results for the constrained phase.

Chapter 8

Conclusions

The aim of this thesis consists in the identification of an approach strategy for an Autonomous Rendezvous with a non-cooperative satellite, with trajectory planning and subsequent control. Even if the case study considered belongs to the on-orbit servicing theme and leads to a final capture with a robotic arm, the strategy identified does not preclude its application on other AR&D/C missions with different purposes or methods of capture. Since the approach is wide-ranging and thus multifaceted, the decision was to divide the trajectory into three phases, identifying three nested keep-out volumes around the target of gradually decreasing dimensions. Focusing mainly on Rendezvous and Proximity Operations, only the last two phases closer to the target have been analysed. The main achievements of the study are summarized below, with some considerations.

1. In the observation phase of the target, the typical relative motions, natural or forced, have been identified to conduct an inspection of the target exploiting the geometry of the orbits (HCW equations), minimizing the total delta-v required through genetic algorithms. In addition to this requirement on delta-v, two further constraints have been considered inside these algorithms: the execution time of the maneuver and safety, through the imposition of keep-out zones.

2. Assuming to chase the trajectory with a spacecraft simulator, the crucial points of the reference trajectory have been studied: an impulsive maneuver and a maintenance of an unnatural relative position. The Model Predictive Controller,

coupled with a Kalman filter, has proven to be a suitable and efficient tool, especially thanks to its predictive and constraint handling capabilities: the delta-v obtained do not differ particularly from the case of impulsive and ideal delta-v.

3. Regarding the inner proximity RVD phase, breaching the last KOS, a method of approach with a non-cooperative and tumbling target has been identified, by using another Model Predictive Controller, this time nonlinear. Taking advantage from the possibility of creating a reference trajectory from the current estimates of the target poses and managing the presence of multiple and even nonlinear constraints, the predictive controller, paired with an Extended Kalman filter, allowed to come up with a solution of the problem and to complete the rendezvous maneuver inside the final berthing box, with very good performance in terms of total delta-v consumption (around $10m/s$ for the worst case of target angular velocity) and maneuvering time (5/6 minutes starting from a distance of $55m$ from the target).

The computational cost that a nonlinear MPC requires is higher when compared to traditional control methods, especially when it needs to handle several nonlinear or time-varing constraints. The use of the proposed controller requires an on-board computer that lives up to these calculations. However, the key properties of handling the constraints, also related to the safety of the maneuver, and managing the behaviour of the system over a sufficiently wide predictive horizon made it preferable to other control methods.

Beyond the results obtained, which confirmed the validity and robustness of the proposed approaches, it is necessary to denote some possible improvements to the performed study. Some of these are listed below.

- Employment of a more accurate simulation environment: non-circular target orbits, orbital perturbations due to the Earth's gravitational field or to the residual air (even if irrelevant), or torques due to the magnetic field or gravitational gradient.

- Study of the control method for pursuing natural relative trajectories, such as

the Walking Safety Ellipse (briefly mentioned at the end of Subsection 6.4.2).

- Prediction and study of collision avoidance maneuvers, following a possible on-board malfunctioning or any missed burns, and consideration of the lighting requirement, with the presence of eclipses, to carry out the rendezvous.
- Use of real data regarding relative sensors, coming from an online or real (thinking of a laboratory test) simulation of an approach maneuver to a rotating object.

Finally, a possible final development consists in the implementation and testing of the adopted control strategy using two satellite mock-ups that move on a frictionless testing table, recreating in a simplified but representative way the rendezvous environment between a controlled chaser and a non-cooperative satellite.

Ringraziamenti

Al dott. Andrea Valmorbida, per la puntualità e la disponibilità prestatemi durante questi ultimi mesi di tesi, al professor Lorenzini, e a tutti i professori che hanno concorso alla mia formazione.

A Dario, Lorenzo, Mattia, Andrea e a tutti gli altri compagni di università, dai primi anni agli ultimi, che hanno reso questo mio percorso più ricco e divertente e mi hanno insegnato il lavoro di squadra.

Ai miei genitori, ai miei fratelli, alla nonna, presenze costanti, che mi hanno supportato e sopportato in questi anni, aiutandomi nelle scelte senza precluderne alcuna.

A Martina, per essermi sempre stata vicino in ogni momento, a Dario, Maria, Federico e a tutti gli amici scout, ai "magnifici" e alla mia squadra di pallanuoto, per aver contribuito a determinare ciò che sono ora.

Con affetto, grazie

Bibliography

[1] an Ansys Company AGI. *Rendezvous Proximity Operations*. URL: https://www.ansys.com/it-it/products/missions.

[2] ASI. *Rosetta*. URL: https://www.asi.it/esplorazione/sistema-solare/rosetta/.

[3] Brent W Barbee et al. "A guidance and navigation strategy for rendezvous and proximity operations with a noncooperative spacecraft in geosynchronous orbit". In: *The Journal of the Astronautical Sciences* 58.3 (2011), pp. 389–408.

[4] David A Barnhart et al. "Using historical practices to develop safety standards for cooperative on-orbit rendezvous and proximity operations". In: *69th International Astronautical Congress (IAC), Bremen, Germany*. 2018, pp. 1–5.

[5] Chris Blackerby et al. "ELSA-d: An In-Orbit End-of-Life Demonstration Mission". In: *69th International Astronautical Congress*. 2018.

[6] Howard Curtis. *Orbital mechanics for engineering students*. Butterworth-Heinemann, 2013.

[7] eoPortal Directory. *Landsat7 specifics*. URL: https://directory.eoportal.org/web/eoportal/satellite-missions/l/landsat-7.

[8] Kaikai Dong et al. "Tube-based robust output feedback model predictive control for autonomous rendezvous and docking with a tumbling target". In: *Advances in Space Research* 65.4 (2020), pp. 1158–1181.

[9] ESA. *Comet 67P/Churyumov-Gerasimenko*. URL: https://www.esa.int/ESA_Multimedia/Images/2014/09/Comet_on_19_September_2014_NavCam.

[10] ESA. *Elsa-D*. URL: https://www.esa.int/ESA_Multimedia/Images/2018/12/End-of-Life_Service_by_Astroscale_demonstrator_ELSA-d_satellite.

[11] ESA. *Envisat model*. URL: https://www.esa.int/ESA_Multimedia/Images/2012/05/Envisat.

[12] NASA'S EXPLORATION and IN-SPACE SERVICES. *OSAM-1*. URL: https://nexis.gsfc.nasa.gov/osam-1.html.

[13] Angel Flores-Abad et al. "A review of space robotics technologies for on-orbit servicing". In: *Progress in aerospace sciences* 68 (2014), pp. 1–26.

[14] Gabriella Gaias, Jean-Sébastien Ardaens, and Christian Schultz. "The AVANTI experiment: flight results". In: (2017).

[15] David E Gaylor and Brent William Barbee. "Algorithms for safe spacecraft proximity operations". In: *AAS/AIAA Spaceflight Mechanics Meeting*. 2007.

[16] Ariane group. *Chemical bi-propellant thruster family*. URL: https://www.space-propulsion.com/brochures/bipropellant-thrusters/bipropellant-thrusters.pdf.

[17] Marko Jankovic, Jan Paul, and Frank Kirchner. "GNC architecture for autonomous robotic capture of a non-cooperative target: preliminary concept design". In: *Advances in Space Research* 57.8 (2016), pp. 1715–1736.

[18] Donald J Kessler and Burton G Cour-Palais. "Collision frequency of artificial satellites: The creation of a debris belt". In: *Journal of Geophysical Research: Space Physics* 83.A6 (1978), pp. 2637–2646.

[19] Heiner Klinkrad. "Orbital debris and sustainability of space operations". In: *Handbook of Satellite Applications* (2017), p. 1413.

[20] Mirko Leomanni et al. "Variable-Horizon Guidance for Autonomous Rendezvous and Docking to a Tumbling Target". In: *Journal of Guidance, Control, and Dynamics* (2022), pp. 1–13.

[21] Qi Li et al. "Model predictive control for autonomous rendezvous and docking with a tumbling target". In: *Aerospace Science and Technology* 69 (2017), pp. 700–711.

[22] Jan Marian Maciejowski. *Predictive control: with constraints*. Pearson education, 2002.

[23] Mathworks. *Extended Kalman filter*. URL: https://it.mathworks.com/help/control/ref/ekf_block.html?searchHighlight=extended%20kalman%20filter&s_tid=srchtitle_extended%20kalman%20filter_1.

[24] Mathworks. *Extended Kalman filter theory*. URL: https://it.mathworks.com/help/control/ug/extended-and-unscented-kalman-filter-algorithms-for-online-state-estimation.html.

[25] Mathworks. *Nonlinear MPC*. URL: https://it.mathworks.com/help/mpc/ug/nonlinear-mpc.html.

[26] Mathworks. *Nonlinear MPC, MATLAB*. URL: https://www.youtube.com/watch?v=hkYf-Chqwdw.

[27] Mathworks. *Optimization Toolbox*. URL: https://it.mathworks.com/help/optim/index.html?searchHighlight=optimization%20toolbox&s_tid=srchtitle_optimization%2520toolbox_1.

[28] MaxarTechnologies. *OSAM-1*. URL: https://blog.maxar.com/space-infrastructure/2021/osam-1-passes-accommodation-cdr-propulsion-system-module-complete?utm_source=explore-space&utm_medium=website&utm_campaign=osam-1.

[29] MIT. *SPHERES-VERTIGO*. URL: https://news.mit.edu/2012/research-update-spheres-satellites-get-camera-magnets-0802.

[30] Nasa. *VERTIGO (Visual Estimation and Relative Tracking for Inspection of Generic Objects)*. URL: https://www.nasa.gov/content/vertigo-visual-estimation-and-relative-tracking-for-inspection-of-generic-objects-0.

[31] Gunter's Space Page. *DEOS mission*. URL: https://space.skyrocket.de/doc_sdat/deos.htmt.

[32] Hyeongjun Park, Stefano Di Cairano, and Ilya Kolmanovsky. "Model predictive control for spacecraft rendezvous and docking with a rotating/tumbling platform and for debris avoidance". In: *Proceedings of the 2011 American Control Conference*. IEEE. 2011, pp. 1922–1927.

[33] Hyeongjun Park et al. "Nonlinear model predictive control for spacecraft rendezvous and docking with a rotating target". In: *27th AAS/AIAA Spaceflight Mechanics Meeting*. Vol. 2. 2017.

[34] Committee on the Peaceful Uses of Outer Space. *Compendium of space debris mitigation standards adopted by states and international organizations*. June 2014. URL: https://www.unoosa.org/pdf/limited/l/AC105_2014_CRP13E.pdf.

[35] Dmitry Pritykin, Sergey Efimov, and Vladislav Sidorenko. "Defunct satellites in nearly polar orbits: Long-term evolution of attitude motion". In: *Open Astronomy* 27.1 (2018), pp. 264–277.

[36] D Reintsema et al. "DEOS–the German robotics approach to secure and de-orbit malfunctioned satellites from low earth orbits". In: *Proceedings of the i-SAIRAS*. Japan Aerospace Exploration Agency (JAXA) Japan. 2010, pp. 244–251.

[37] Timothy Philip Setterfield. "On-orbit inspection of a rotating object using a moving observer". PhD thesis. Massachusetts Institute of Technology, 2017.

[38] Michael A Shoemaker et al. "OSAM-1 decommissioning orbit design". In: *AAS/AIAA Astrodynamics Specialist Conference.[Google Scholar]*. 2020.

[39] U.S. Geological Survey. *Landsat7*. URL: https://www.usgs.gov/landsat-missions/landsat-7.

[40] Matthew A Vavrina et al. "Safe rendezvous trajectory design for the restore-l mission". In: *Proceedings of the 29th AAS/AIAA Space Flight Mechanics Meeting*. 2019, pp. 3649–3668.

[41] Avishai Weiss et al. "Model predictive control for spacecraft rendezvous and docking: Strategies for handling constraints and case studies". In: *IEEE Transactions on Control Systems Technology* 23.4 (2015), pp. 1638–1647.

[42] Wikipedia. *DART mission*. URL: `https://en.wikipedia.org/wiki/Double_Asteroid_Redirection_Test`.

[43] Markus Wilde. "CEU Session# 4-Space Robotics for On-Orbit Servicing and Space Debris Removal". In: (2019).

[44] Kjetil Wormnes et al. "ESA technologies for space debris remediation". In: *6th European Conference on Space Debris*. Vol. 1. ESA Communications ESTEC Noordwijk, The Netherlands. 2013, pp. 1–8.

[45] Bang-Zhao Zhou, Xiao-Feng Liu, and Guo-Ping Cai. "Motion-planning and pose-tracking based rendezvous and docking with a tumbling target". In: *Advances in Space Research* 65.4 (2020), pp. 1139–1157.