POLYTECHNIC OF TURIN

Master's Degree in Mechatronics Engineering

Master's Degree Thesis

A NMPC-based spacecraft rendezvous maneuver with moving obstacles and variable prediction horizon

Supervisors

Prof. Carlo NOVARA

Prof. Michele PAGONE

Candidate

Andrea ALBASINI

July 2023

Abstract

Since the first orbiting explorations, the rendezvous maneuver has become a fundamental operation in space missions. The operation consists of a spacecraft alignment to a specific target point, usually another spacecraft or some debris orbiting in the space. A successful rendezvous mission usually requires the determination of the trajectory, the speed, and the timing of the approaching spacecraft to ensure a safe and accurate approach. In this work thesis, the rendezvous problem is explained and analysed. Then, a control algorithm that guides an autonomous spacecraft to a requested position close to the target is proposed. Usually a spacecraft during the mission could encounter obstacles along the orbit, such as a meteorite to be dodged along the trajectory. In order to make the rendezvous mission more realistic, the task is performed including an obstacle like a sphere moving towards the spacecraft, so that the control algorithm is able to guide the spacecraft along its trajectory, satisfying the path constraints, and making him reach the target point. The obstacles are implemented within the optimization problem by augmenting the cost function with a suitable weighted penalty term that prevents the satellite to closely approach the obstacle. In the context of the space rendezvous maneuver, the Nonlinear Model Predictive Control (NMPC) appears as an appealing control tool, providing an optimal control law in presence of constraints on both input and state. The NMPC optimal control problem is solved by means of the Pontryagin Minimum (maximum in the original form) Principle (PMP). Furthermore, a slightly version of the NMPC, featuring a variable prediction horizon, is presented. The effectiveness of the two NMPC algorithms are then tested in simulation in a Matlab/Simulink environment, where the performance of the fixed and variable prediction horizons controllers are compared.

Acknowledgements

Without commitment
you'll never start
but more importantly
without consistency
you'll never finish
cit. Denzel Washington

Table of Contents

List of Tables

List of Figures

Acronyms

MPC

Model Predictive Control

NMPC

Nonlinear Model Predictive Control

HCW

Hill-Clohessy-Wiltshire

BVP

Boundary value problem

Tp

Prediction Horizon

Ts

Sampling time

ECI

Earth centered inertial

LVLH

Local vertical local horizontal

S/C

Spacecraft

RH

Receding Horizon

RdV

Rendezvous

ISS

International Space Station

LEO

Low Earth Orbit

PMP

Pontryagin Minimum (or Maximum) Principle

OCP

Optimal Control Problem

CoM

Center of mass

Chapter 1

Introduction

Space missions are a critical aspect of human exploration, enabling humans to study the universe and expand the understanding of the cosmos. The advancement of space missions has been driven by technological developments, international cooperation and scientific discoveries.

Moreover, space missions are inherently complex, involving a wide range of scientific, engineering, and logistical challenges. The success of space missions depends on various factors, such as the reliability of spacecraft and launch vehicles, the accuracy of navigation and communication systems, and the effectiveness of scientific instrumentation.

The first space missions were conducted around the 1960s. Among them, the Apollo program marked a significant milestone in space exploration, see [1]. The program's primary goal was to land humans on the moon. Rendezvous was a crucial aspect of the Apollo missions, as it involved the alignment and docking of two spacecraft in orbit around the moon. This was necessary for the transfer of astronauts and supplies between spacecraft and for the eventual return to Earth.

The first successful lunar rendezvous mission was Apollo 8 in 1968, which involved the first manned flight to the Moon's vicinity. The Apollo 11 mission in 1969 marked the first successful landing on the Moon, and subsequent missions continued to demonstrate the effectiveness of the rendezvous technique in space exploration.

Since then, rendezvous has remained a critical aspect of space missions. It is necessary for various applications, such as the servicing of satellites and the assembly of space stations. The development of new technologies, such as autonomous guidance and navigation systems, has enabled more advanced and complex rendezvous maneuvers, making space exploration safer, more efficient, and more flexible.

In conclusion, the history of initial space missions highlights the importance of rendezvous as a crucial point in space exploration. From the early manned missions to the moon landings and beyond, the success of space missions has depended on the ability to precisely align and approach spacecraft in orbit. As space exploration continues to advance, the importance of rendezvous will only become more significant.

Rendezvous is a fundamental key in space exploration that involves the precise alignment and approach of two or more spacecraft in orbit. This maneuver is essential for a wide range of space missions, as explained before. The success of a rendezvous mission depends on various factors, such as the accuracy of navigation and guidance systems, the complexity of the spacecraft design, and the effectiveness of communication protocols.

In recent years, there has been a growing interest in developing advanced rendezvous technologies that can enhance the safety, efficiency, and flexibility of space missions. These technologies include autonomous guidance and navigation systems, advanced sensors and imaging techniques, and innovative propulsion systems. As the demand for space exploration continues to increase, the importance of rendezvous missions and the development of associated technologies will only become more significant.

The importance of rendezvous lies in its ability to facilitate a wide range of space activities, from satellite servicing to the assembly of space stations. Here are some of the key reasons why rendezvous is fundamental in space missions:

- Crew transfer: it enables spacecraft to transfer crew members between vehicles in orbit, a vital capability for long-duration manned missions.

- Satellite servicing: rendezvous is essential for servicing and repairing satellites in orbit.

- Cargo transfer: it enables the transfer of supplies and equipment between spacecraft. This is especially important for resupply missions to the International Space Station (ISS), where cargo vehicles have to rendezvous and dock with the station to deliver supplies.

- Space exploration: it is crucial for exploration missions, such as those to Mars and other planets.

In addition to these applications, rendezvous is also crucial for ensuring the safety of space missions. Accurate and precise rendezvous operations are essential to avoid collisions and other hazards in orbit.

After the successful completion of initial space missions, there was a need to improve their efficiency in order to achieve lower life cycle costs. This improvement involved various aspects, such as simplified and standardized mission planning and training, reducing the number of mission support personnel, increasing flight rates, eliminating extensive flight-to-flight analysis, and eliminating the need for computing flight-specific trajectory data. Furthermore, in recent years, Rdv operations have become fundamental in space research for debris removal missions. The accumulation of space debris poses a significant threat to operational spacecraft, and rendezvous operations are being developed and utilized to safely remove and dispose of debris from orbit.

The aim of this thesis is to develop a control algorithm capable of successfully accomplishing the Guidance and Control task for a wide range of RdV problems, ultimately achieving precise targeting of a default destination.

In this work, a spacecraft called the "target" with a circular orbit around the Earth has been utilized, having similar characteristics to the ISS. The target has the same orbital angular rate and is located at the same distance from the Earth as the ISS, which is approximately 400-450 kilometers above the Earth's surface. Due to this proximity, the analysis focuses on the control system that would be installed on a hypothetical spacecraft operating in Low Earth Orbit (LEO). The provided image, sourced from the European Space Agency site [2], illustrates the maximum range of LEO, which is approximately 1000 kilometers above the Earth's surface.

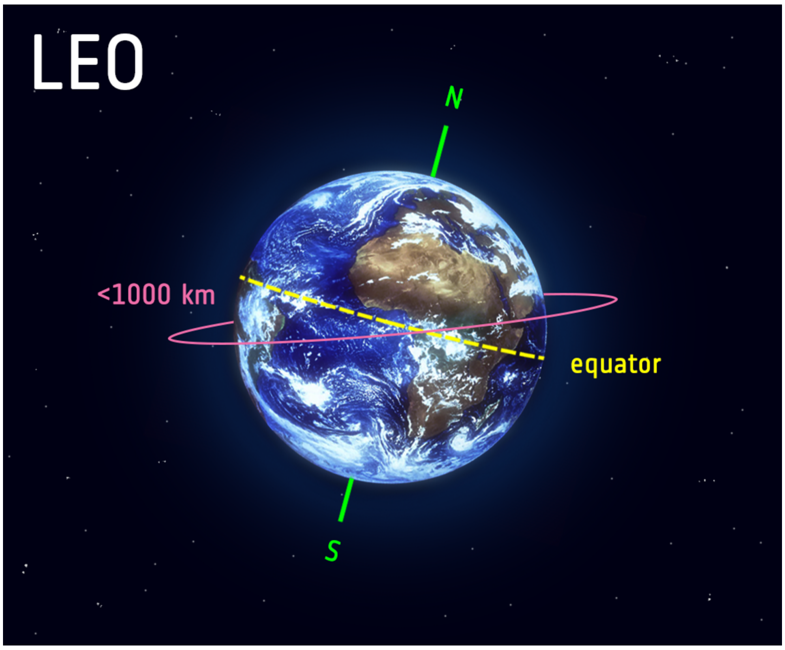


Figure 1.1: Low Earth orbit (taken from [2])

In this context, Model Predictive Control (MPC) plays a crucial role in the space control environment and holds great potential for various applications in future space missions. MPC is particularly advantageous for missions characterized by strong nonlinear dynamics. One of its main strengths is its ability to handle Multi-Input Multi-Output (MIMO) systems and effectively manage input and state constraints while optimizing different performance criteria. The key aspect of MPC design lies in its approach to addressing optimal control problems within a receding horizon strategy. This strategy involves repeatedly solving an optimization problem over a finite time horizon, considering the current state of the system and updating the control actions based on the obtained solution.

To address the challenges posed by nonlinear dynamics, constraints, and non-convex performance indexes, Nonlinear Model Predictive Control (NMPC) has been introduced. Thanks to its stability and robustness properties, NMPC is capable of effectively managing the constraints arising from various maneuvers and mitigating discrepancies between the model of the actual spacecraft and the predictive models utilized in the algorithm. The algorithm incorporates the Clohessy-Wiltshire equations to describe the model.
In order to ensure the simulation of a safe space around the target and to prevent potential collisions with asteroids or other objects along the planned trajectory, considering constraints is essential.

Furthermore, one of the main advantages of NMPC is its ability to integrate both the guidance and control tasks into a single algorithm, allowing for the autonomous planning of required maneuvers with minimal human intervention. Therefore, in this paper, a NMPC framework will be utilized for the studied rendezvous missions.

The NMPC optimization process utilizes the Pontryagin Minimum (or Maximum) Principle, which, under suitable conditions, provides an explicit control law even in the presence of non-linearity. This is a significant advantage because non-linearity often makes it challenging to find an explicit solution. By employing the dual formulation of the optimal control problem, the Hamiltonian scalar function and the Lagrange (or co-state) variables are introduced, transforming the Optimization Control Problem into a standard Two-Points Boundary Value Problem. Solving this problem yields the gains required for the explicit optimal control law.

1.1 Thesis Outline

Chapter 2 investigates the theoretical background for the GNC problem. It includes discussions on absolute and relative motion description, coordinate frames and general characteristics of orbits. Chapter 3 focuses on Nonlinear Model Predictive Control (NMPC), a powerful and flexible control approach. It covers the formulation and features of NMPC, with a reference to the Adaptive Horizon NMPC. Chapter 4 describes the application of Pontryagin's Minimum Principle to solve the algorithm for the GNC problem. It explores the advantages of this method in dealing with the control challenge. Chapter 5 delves into mission-specific topics and the software utilized for the research before to present the simulation results in the sixth Chapter. Lastly, in Chapter 7, there will be a discussion about the obtained results and suggestions for potential future works related to the GNC problem.

Chapter 2

Theoretical Background

2.1 Introduction

This chapter is necessary for the reader to understand the work explained in the following chapters. There will be a review of the fundamental concepts of orbital dynamics and the coefficient used. The following theory is taken from [3], [4] and [5].

A S/C can be seen as a rigid body that moves in orbital space. Essentially, the motion can be divided into translation and rotation. For this reason, the laws of motion will be described, as well as the various types of reference frames used among the planets.

2.1.1 Kepler's law

Kepler around the beginning of the *XVII* century published a correction of Copernicus' law, who thought the orbits were circular. Kepler, on the other hand, explained how the orbits were elliptical and how their velocity varied. The laws are reported below, from [6]:

1. The orbit of a planet is an ellipse with the Sun at one of the two foci.

2. A line segment joining a planet and the Sun sweeps out equal areas during equal intervals of time.

3. The square of a planet's orbital period is proportional to the cube of the length of the semi-major axis of its orbit.

These are the exact laws of Kepler related to a planet orbiting around the Sun, but they can be applied to any small body orbiting around a star, such as the Earth in this particular case, see [7].

- Orbits are elliptical, with the heavier body at one focus of the ellipse. A special case could be a circular orbit with the Star as center.
- A line drawn from the planet to the satellite sweeps out equal areas in equal times no matter which portion of the orbit is measured.
- The square of a satellite's orbital period is proportional to the cube of its average distance from the planet.

2.1.2 Newton's law

The laws of Newton form the basis of classical mechanics, describing how a body can move in space and interact with it. There are three laws of motion, along with one law of gravitation.

- Newton's laws of motion
  1. A particle remains at rest or continues to move at a constant velocity, unless acted upon by an external force.
  2. The rate of change of the linear momentum mv of a particle is given by

     $$\frac{d}{dt}(m\mathbf{v}) = F \tag{2.1}$$

     - m : particle mass;
     - $\mathbf{v}$: particle velocity;
     - $\mathbf{F}$: force acting on the particle.

     For an object with constant mass , the force will be equal to the mass of the body multiplied by the acceleration of it:

     $$\mathbf{F} = m\frac{d\mathbf{v}}{dt} = m\mathbf{a} \tag{2.2}$$

  3. For any force $\mathbf{F}_{12}$ exerted by a particle 1 on a particle 2, there exists a force

     $$\mathbf{F}_{21} = -\mathbf{F}_{12} \tag{2.3}$$

     exerted by particle 2 on particle 1 equal in magnitude and with opposite direction.

- Newton's law of gravitation:
  Every point mass attracts every other point mass by a force acting along the line intersecting the two points. The force is proportional to the product of the two masses, and inversely proportional to the square of the distance between them [8]:

  $$\mathbf{F} = \frac{Gm_1m_2\mathbf{r}}{r^3} \tag{2.4}$$

- m_1, m_2 : particle masses;
- $\mathbf{r}$: vector of magnitude $r = |r|$, the distance that connect the two particles;
- G : $6.67 \times 10^{-11} Nm^2/kg^2$: universal constant of gravitation.

2.2 The Two-Body problem

In order to understand the interplay of forces in space and how bodies and their gravitational forces interact with each other, it is necessary to explain their behavior. The simplest case is the interaction between only two bodies, known as the *Two-Body Problem*, as shown in the figure below.

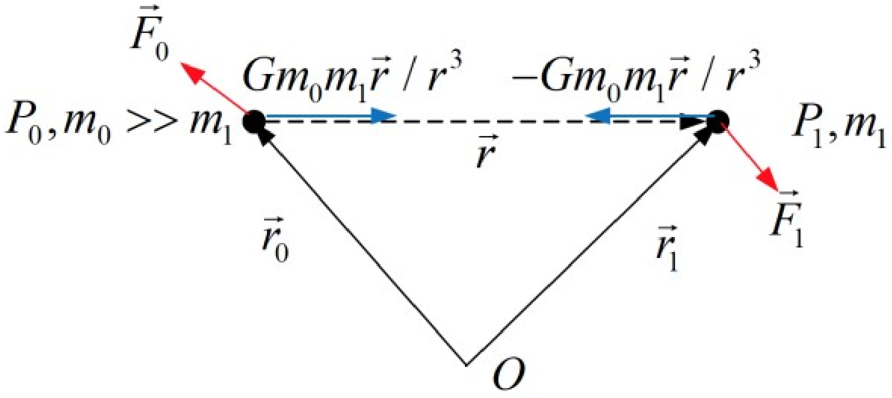


Figure 2.1: Two body in a reference frame

As seen in the figure, there are two different masses, denoted as m_0 and m_1, with m_0 considered greater than m_1. These masses are located at points P_0 and P_1, respectively, in an inertial frame with origin at O. The distances between the origin and the masses are represented by vectors $\vec{r_0}$ and $\vec{r_1}$. The attractive forces between the two bodies are shown in blue, with equal magnitudes and opposite directions. Additionally, there are two external forces represented in red. Here's a quick recap of the symbols used in the picture..

- $\mathbf{r}_0$ and $\mathbf{r}_1$: positions of the masses;
- $\mathbf{r} = \mathbf{r}_1 - \mathbf{r}_0$: relative position of the masses;
- $\mathbf{v}_0$ and $\mathbf{v}_1$: velocities of the masses;
- $\mathbf{F}_0$ and $\mathbf{F}_1$: external forces acting on the masses.

Considering the Newton's *II* law and the gravity law it is possible to obtain the following equations:

$$\begin{aligned} \dot{\mathbf{v}}_0 &= \frac{Gm_1}{r^3}\mathbf{r} + \frac{1}{m_0}\mathbf{F}_0 \\ \dot{\mathbf{v}}_1 &= -\frac{Gm_0}{r^3}\mathbf{r} + \frac{1}{m_1}\mathbf{F}_1 \end{aligned} \tag{2.5}$$

Here, $\mathbf{r} = \mathbf{r}_1 - \mathbf{r}_0$ represents the relative position between the two masses, and $r = |\mathbf{r}|$ represents the magnitude of the vector. It is important to note that constant masses have been assumed.

Considering the equation relative to the Center of Mass (CoM):

- $\mathbf{v} = \mathbf{v}_1 - \mathbf{v}_0$ (relative velocity)
- $\mathbf{r}_C = \frac{m_0}{m_0+m_1}\mathbf{r}_0 + \frac{m_1}{m_0+m_1}\mathbf{r}_1$ (CoM position)
- $\mathbf{v}_C = \frac{m_0}{m_0+m_1}\mathbf{v}_0 + \frac{m_1}{m_0+m_1}\mathbf{v}_1$ (CoM velocity)

From the previous computations, the following equations can be obtained:

$$\begin{aligned} \dot{\mathbf{v}} &= -\frac{G(m_0+m_1)}{r^3}\mathbf{r} + \frac{1}{m_1}(\mathbf{F}_1 - \frac{m_0}{m_1}\mathbf{F}_0) \quad (\textit{relative} \quad \textit{motion}) \\ \dot{\mathbf{v}_C} &= \frac{1}{m_1}\frac{\mathbf{F}_1 + \mathbf{F}_0}{1 + m_0/m_1} \qquad\qquad\qquad\qquad (\textit{CoM} \quad \textit{motion}) \end{aligned} \tag{2.6}$$

The first equation describes the relative motion of the two bodies, while the second equation represents the motion of the CoM between them. When $m_0 \gg m_1$, the relative motion equation described below is known as the Restricted Two-Body Equation.

$$\dot{\mathbf{v}} + \mu\frac{\mathbf{r}}{r^3} = \frac{1}{m_1}\mathbf{F}_1 \quad (\textit{R2B}) \tag{2.7}$$

Where $\mu = Gm_0$ represents the *gravitational parameter*, which is always associated with the larger body. For example, when considering a S/C orbiting around the Earth, the gravitational parameter of the Earth is taken into account.

2.2.1 Free motion of the restricted two-body problem

As mentioned earlier, the Restricted Two-Body Equation includes an external force in the second term. When this force is nullified, the problem is referred to as the *Free R2B*:

$$\dot{\mathbf{v}} + \mu \frac{\mathbf{r}}{r^3} = 0 \tag{2.8}$$

Below are the properties of this problem, which will be analyzed starting from the next page :

- the total mechanical energy of the FR2B system is conserved;
- the angular momentum of the FR2B system is conserved;
- the free response of the FR2B equation occurs on a plane;

Energy conservation

Taking the dot product of equation FR2B (2.8) with $\mathbf{v}$:

$$\begin{aligned} \dot{\mathbf{v}} \cdot \mathbf{v} + \frac{\mu}{r^3}\mathbf{r} \cdot \mathbf{v} &= \frac{1}{2}\frac{d}{dt}(\mathbf{v} \cdot \mathbf{v}) + \frac{\mu}{2r^3}\frac{d}{dt}(\mathbf{r} \cdot \mathbf{r}) \\ = \frac{d}{dt}\frac{v^2}{2} + \frac{\mu}{2r^3}\frac{d}{dt}r^2 &= \frac{d}{dt}\frac{v^2}{2} + \frac{\mu \dot{r}}{r^2} = \frac{d}{dt}\left(\frac{v^2}{2} - \frac{\mu}{r}\right) = 0. \end{aligned} \tag{2.9}$$

This proves the principle of energy conservation:

$$\dot{\mathcal{E}} = 0, \quad \mathcal{E} = const \tag{2.10}$$

- $\mathcal{E} = \frac{v^2}{2} - \frac{\mu}{r}$: total (mechanical) energy per unit mass
- $\frac{v^2}{2}$: kinetic energy per unit mass
- $-\frac{\mu}{r}$: potential energy per unit mass.

The total mechanical energy results constant proving the principle of energy conservation. From the energy equation is possible to find the corresponding orbital velocity which results $v = \sqrt{2\mu/r + 2\mathcal{E}}$.

Angular momentum conservation and planar motion

This time taking the cross product of FR2B equation with $\mathbf{r}$:

$$\mathbf{r} \times \dot{\mathbf{v}} + \frac{\mu}{r^3}\mathbf{r} \times \mathbf{r} = \mathbf{r} \times \dot{\mathbf{v}} = \mathbf{v} \times \mathbf{v} + \mathbf{r} \times \dot{\mathbf{v}} = \frac{d}{dt}(\mathbf{r} \times \mathbf{v}) = 0. \tag{2.11}$$

This proves the principle of angular momentum conservation:

$$\dot{\mathbf{h}} = 0, \quad \mathbf{h} = const \tag{2.12}$$

$\mathbf{h} = \mathbf{r} \times \mathbf{v}$: angular momentum per unit mass.

An important observation is that due to the conservation of specific angular momentum, denoted as $\mathbf{h}$, the position vector $\mathbf{r}$ and the velocity vector $\mathbf{v}$ always remain in the same plane. This plane is referred to as the orbital plane.

It is worth noting that the orbital plane also contains the two masses, m_0 and m_1, which are involved in the restricted two-body problem.

2.3 Orbit equation

This time, taking the cross product of the FR2B equation (2.8) with $\mathbf{h}$, the following equation is obtained:

$$\left(\dot{\mathbf{v}} + \frac{\mu}{r^3}\mathbf{r}\right) \times \mathbf{h} = \frac{d}{dt}\left(\mathbf{v} \times \mathbf{h} - \frac{\mu}{r}\mathbf{r}\right) = 0. \tag{2.13}$$

The above equality can be established thanks to:

$$\frac{d}{dt}(\mathbf{v} \times \mathbf{h}) = \dot{\mathbf{v}} \times \mathbf{h} + \mathbf{v} \times \dot{\mathbf{h}} = \dot{\mathbf{v}} \times \mathbf{h}, \quad (\dot{\mathbf{h}} = 0)$$

$$\begin{aligned} \frac{d}{dt}\left(-\frac{\mathbf{r}}{r}\right) &= \frac{\dot{r}}{r^2}\mathbf{r} - \frac{1}{r}\mathbf{v} = \frac{1}{2r^3}\left(\frac{d}{dt}r^2\right)\mathbf{r} - \frac{1}{r}\mathbf{v} \\ &= \frac{1}{2r^3}\left(\frac{d}{dt}(\mathbf{r} \cdot \mathbf{r})\right)\mathbf{r} - \frac{1}{r}\mathbf{v} \\ &= \frac{1}{r^3}((\mathbf{r} \cdot \mathbf{v})\mathbf{r} - (\mathbf{r} \cdot \mathbf{r})\mathbf{v}) \\ &= \frac{1}{r^3}\mathbf{r} \times (\mathbf{r} \times \mathbf{v}) \end{aligned} \tag{2.14}$$

The above equation (2.13) shows that

$$\mathbf{v} \times \mathbf{h} - \frac{\mu}{r}\mathbf{r} = const \doteq \mu\mathbf{e} \tag{2.15}$$

where $\mathbf{e}$ represents the *eccentricity vector* and $e = |\mathbf{e}|$ represents the *eccentricity value*.

Taking the dot product of $\mathbf{r}$ with the previous equation (2.15):

$$\mathbf{r} \cdot (\mathbf{v} \times \mathbf{h}) - \frac{\mu}{r}\mathbf{r} \cdot \mathbf{r} = \mu\mathbf{r} \cdot e. \tag{2.16}$$

Considering the scalar triple product: $\mathbf{r} \cdot (\mathbf{v} \times \mathbf{h}) = (\mathbf{r} \times \mathbf{v}) \cdot \mathbf{h}$. Moreover $(\mathbf{r} \times \mathbf{v}) \cdot \mathbf{h} = \mathbf{h} \cdot \mathbf{h} = h^2$. Thus,

$$h^2 - \mu r = \mu r e \cos\theta \tag{2.17}$$

where θ, the angle between $\mathbf{r}$ and $\mathbf{e}$, is called the *true anomaly*.

Defining $p = h^2/\mu$, called the *parameter* or *semilatus rectum*, and deriving r from (2.17), the Orbit Equation (ORE) is obtained:

$$r = \frac{p}{1 + e\cos\theta}. \tag{2.18}$$

2.4 Orbit Geometry

The ORE

$$r = \frac{p}{1 + e\cos\theta}. \tag{2.19}$$

represents the equation of a conic section, which is expressed in terms of polar coordinates r and θ. When $\theta \in [0,2\pi]$, the radial distance r traces out a conic.

A conic section, or simply a conic, is a curve formed by the intersection of a cone and a plane.

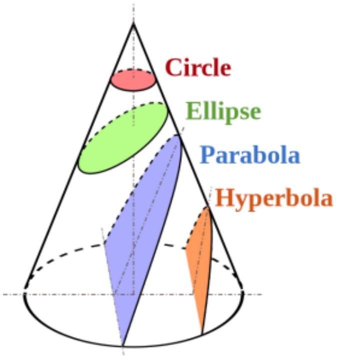


Figure 2.2: Intersections of a cone with different planes

Based on the value of the eccentricity, different types of conic sections can be distinguished:

- circle, $e = 0$;
- ellipse, $0 \leq e < 1$;
- parabola, $e = 1$;
- hyperbola, $e > 1$.

In spite of they are different curves, they have certain common characteristic:

- the origin is located at one focus;
- θ is measured from the point on the conic nearest to the focus;
- p determines the size;
- e determines the shape.

2.4.1 Ellipse

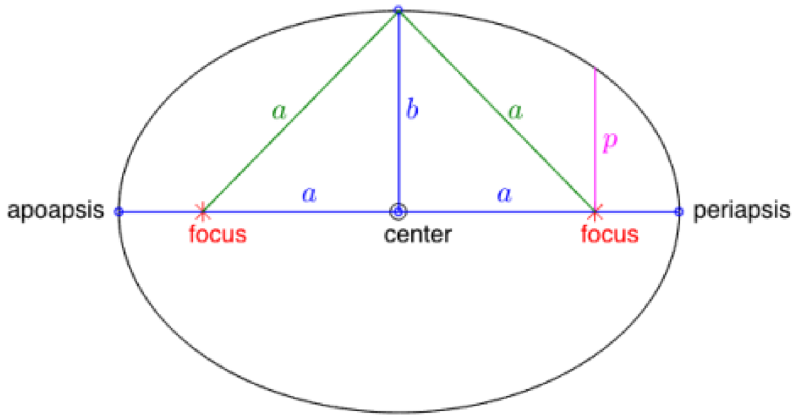


Figure 2.3: Ellipse

An ellipse is the locus of points the sum of whose distances from two fixed points (foci) is constant ($= 2a$ with a = semi-major axis).
Some usual parameters of an horizontal ellipse are:

- previously explained eccentricity $0 \leq e < 1$;

- semilatus rectum : $p = a(1 - e^2)$;
- semi-major axis: $a = p/(1 - e^2)$;
- semi-minor axis: $b = a\sqrt{1 - e^2}$;
- distance center-focus $c = ae$.

The two points of the curve corresponding at $\theta = 0$ and $\theta = \pi$ are called respectively *periapsis* and *apoapsis*. In the topic of planet orbiting around the Sun these two points take the names of *Perihelion* and *Aphelion*. While, in the case of a body orbiting around the Earth they take the names of *Perigee* and *Apogee*.

Considering the conservation of mechanical energy, it can be expressed at the Periapsis and the Apoapsis as follows:

$$\mathcal{E} = \frac{v_p^2}{2} - \frac{\mu}{r_p} = \frac{v_a^2}{2} - \frac{\mu}{r_a} = constant \tag{2.20}$$

As seen in [9], thanks to a specific reference frame (centered at a focus with an axis passing through the Periapsis, another axis perpendicular to it within the orbital plane, and a third axis in the same direction as h), it becomes possible to have the velocity and radius vectors perpendicular to each other at the Periapsis (and Apoapsis).

Therefore, it is possible to write:

$$h = r_p v_p = r_a v_a = constant, \quad so \quad v_p = \frac{r_a}{r_p} v_a \tag{2.21}$$

Taking the mechanical energy equation (2.20) and using the new formula of v_p:

$$\frac{v_a^2}{2} - \frac{v_p^2}{2} = \frac{\mu}{r_a} - \frac{\mu}{r_p}, \qquad \frac{1}{2}\frac{r_p^2 - r_a^2}{r_p^2} v_a^2 = \mu \frac{r_p - r_a}{r_a r_p} \tag{2.22}$$

Solving with respect to the kinetic energy at the Apoapsis:

$$\frac{v_a^2}{2} = \mu \frac{r_p}{r_a(r_a + r_p)} \tag{2.23}$$

Regarding that $r_p + r_a = 2a$. It is possible to write:

$$\frac{v_a^2}{2} = \mu \frac{2a - r_a}{2a r_a} = \frac{\mu}{r_a} - \frac{\mu}{2a} \tag{2.24}$$

Substituting the obtained value in ε (2.20) the total mechanical energy becomes:

$$\mathcal{E} = \frac{v_a^2}{2} - \frac{\mu}{r_a} = -\frac{\mu}{2a} = negative \quad constant \tag{2.25}$$

While, the velocity is given by the vis-viva equation:

$$v = \sqrt{\frac{2\mu}{r} - \frac{\mu}{a}} \tag{2.26}$$

In the case of circular orbit: $e = 0, a = r$ and $v = \sqrt{\mu/r}$.

2.4.2 Parabola

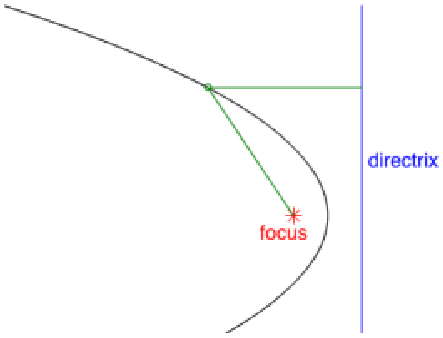


Figure 2.4: Parabola

A parabola is the locus of points whose distance from a fixed point (focus) is equal to the distance from a fixed line (directrix).

For a parabolic orbit e = 1, implying that: $r_a \to \infty$, $a \to \infty$;
the total energy is null: $\mathcal{E} \to 0$.

From the vis-viva equation, for any orbital position with radius r, we obtain the corresponding velocity

$$v_e = \sqrt{\frac{2\mu}{r}} \tag{2.27}$$

This is known as the *escape velocity.* Escape velocity, also referred to as escape speed, represents the minimum speed required for a free, non-propelled object to break free from the gravitational influence of a primary body and ultimately reach an infinite distance away from it.

2.4.3 Hyperbola

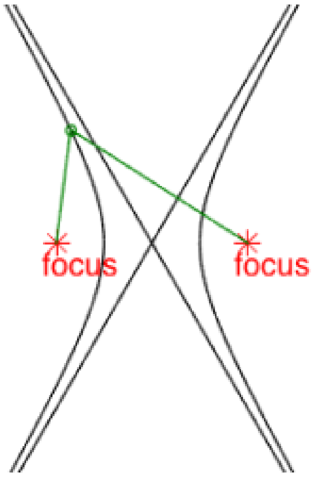


Figure 2.5: Hyperbola

An hyperbola is the locus of points the difference of whose distances from two fixed points (foci) is $const = -2a$. For a hyperbolic orbit, $e > 1$,implying that the total energy is positive

$$\mathcal{E} = \frac{v_\infty^2}{2} > 0 \qquad (2.28)$$

.

With $(r \to \infty)$ it is possible to obtain the following asymptotic quantities:

- angle: $\theta_\infty = \arccos \frac{-1}{e}$;
- velocity: $v_\infty = \sqrt{\mu/|a|}$.

A newsworthy comment is that using an hyperbolic orbit near to a moving planet leads to an increase of the velocity of the S/C without it remains captured from it. This phenomena is called *Hyperbolic passage*.

2.4.4 Parameters review

Table 2.1: Energy and orbital parameters

Energy and orbital velocity					
No.	Orbit	Eccentricity e	Semi-major axis a	Energy per unit mass E	Orbital velocity v
0	Circular	0	$r(t) = a$	$-0.5\mu/a < 0$	$v_e = \sqrt{\mu/r}$
1	Elliptic	<1	>0	$-0.5\mu/a < 0$	$\sqrt{\mu(2/r - 1/a)}$
2	Parabola	1	∞	0	$v_e = \sqrt{2\mu/r} = \sqrt{2}v_e$
2 bis	Idem,Earth				$v_e = 11.2km/s$
3	Hyperbola	>1	<0	$-0.5\mu/a > 0$	$\sqrt{\mu(2/r - 1/a)}$

2.4.5 Coordinate Transformation

The analysis mentioned above is built upon the FR2B equation, which allows us to derive the position trajectory using the ORE.
Now if integrating the FR2B with null initial components along z axis $(z(0) = 0, \dot{z}(0) = 0)$, the position vector r becomes only dependent on x and y, $\mathbf{r}(t) = (x(t), y(t), z(t) = 0)$. Therefore, for a two-dimensional x-y plane, the coordinates can be calculated as follows.

From Cartesian to Polar coordinates:

$$r(t) = \sqrt{x^2(t) + y^2(t)}$$
$$\cos\theta(t) = \frac{x(t)}{\sqrt{x^2(t) + y^2(t)}} \tag{2.29}$$

Inverse transformation:

$$\begin{aligned} x(t) &= r\cos\theta(t) \\ y(t) &= r\sin\theta(t) \\ z(t) &= 0 \end{aligned} \tag{2.30}$$

2.5 Reference frames

An essential aspect for comprehending the absolute and relative motion of a satellite orbiting around a star is the definition of an appropriate reference system. While various reference systems can be employed in orbital analysis, this work will focus on the utilization of two specific reference systems.

The reference frames are the following:

- **LVLH - local vertical local horizontal frame** (non inertial)
- **GE - geocentric equatorial frame** (inertial)

Local Vertical Local Horizontal Reference frame: origin at P_1

- l_3 (local vertical): defined along the direction $P_0 \rightarrow P_1$, on the orbit plane;
- l_1 (local horizontal): perpendicular to l_3, on the orbit plane, sign concordant with the orbital velocity;
- $l_2 = l_3 \times l_1$ (orbit pole): perpendicular to the orbit plane.

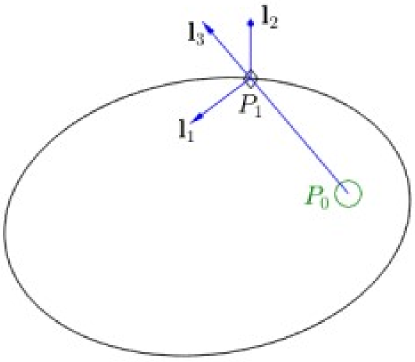


Figure 2.6: LVLH frame

Geocentric Equatorial Frame: origin at Earth CoM;

- $\hat{I}$: Earth → Sun direction, 1^{st} day of spring (vernal equinox) ;
- $\hat{K}$: polar rotation axis, positive from the earth CoM towards the North Pole;
- $\hat{J} = \hat{K} \times \hat{I}$: on the equatorial plane;

This last type of reference frame is fixed and it does not rotate with the Earth, so it is also independent on the S/C orbit.

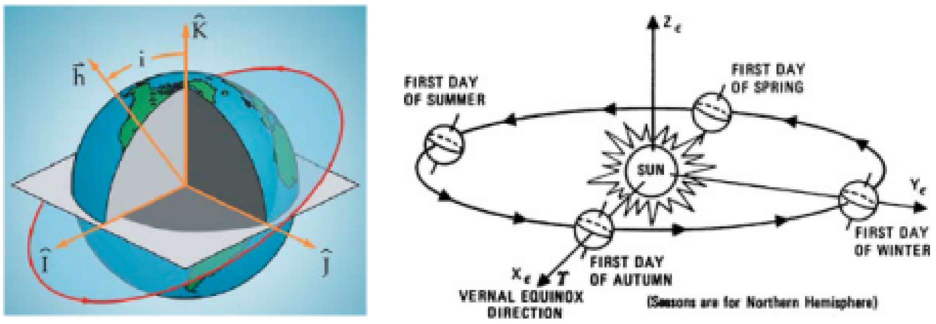


Figure 2.7: GE frame

2.6 Orbital elements

When focusing on elliptical orbits around the Earth, several distinct features can be identified to describe each orbit. First, two different planes can be distinguished: the orbital plane and the equatorial plane.
The intersection of these two planes is referred to as the *line of nodes*. The angle i between these two planes is known as the *inclination*.
Additionally, there are other key elements that help characterize an orbit. The *ascending node* represents the point where the orbit intersects the equatorial plane. The angle Ω is defined as the angle from the $\hat{I}$ axis of the Geocentric Equatorial (GE) frame to the ascending node. Similarly, the angle ω is the angle from the ascending node to the perigee.

Furthermore, ν (also denoted as θ) represents the true anomaly, which is the angle from the perigee to the spacecraft's position on the orbit.

The classical orbital elements consist of six quantities. Among them, a, e, Ω, i, and ω are five independent and constant parameters that fully describe the orbit.

The sixth quantity, ν, determines the exact position of the orbiting body along the orbit.

Table 2.2: Orbital elements

Element	Name	Description	Range of values	Undefined
a	Semi-major axis	Size	Depends on the conic section	Never
e	Eccentricity	Shape	$e = 0$: circle $0 < e < 1$: ellipse	Never
i	inclination	Tilt, angle from K unit vector to specific angular momentum vector h	$0 \leq i \leq 180$	Never
Ω	Right ascension of the ascending node	Swivel, angle from vernal equinox to ascending node	$0 \leq \Omega < 360$	If i = 0 or 180
ω	Argument of perigee	Angle from ascending node to perigee	$0 \leq \omega < 360$	If i = 0 or 180 or e = 0
ν	True anomaly	Angle from perigee to the S/C's position	$0 \leq \nu < 360$	If e = 0

From Position and Velocity to Orbital Elements

The orbital elements provide an equivalent representation of the position and velocity vectors of a S/C. Once the orbit geometry is fully defined in the GE frame, the motion of the spacecraft can be completely described by the true anomaly, ν.

It is crucial to understand the conversion between the position and velocity vectors ($\mathbf{r}$ and $\mathbf{v}$) and the six orbital elements, as it allows for a comprehensive understanding of the orbit.

$$\mathbf{h} = \mathbf{r} \times \mathbf{v}, \qquad \mathbf{e} = \frac{1}{\mu}\mathbf{v} \times \mathbf{h} - \frac{\mathbf{r}}{r}, \qquad \hat{\mathbf{I}}' = \hat{\mathbf{K}} \times (\mathbf{h}/h)$$

$$a = h^2/(\mu(1-e^2)), \qquad e = |\mathbf{e}|, \qquad \cos i = \hat{\mathbf{K}} \cdot \mathbf{h}/h, \tag{2.31}$$

$$\cos w = \hat{\mathbf{I}}' \cdot \mathbf{e}/e, \qquad \cos \Omega = \hat{\mathbf{I}} \cdot \hat{\mathbf{I}}', \qquad \cos \theta = \mathbf{r} \cdot \mathbf{e}/(re)$$

Other useful formula are the *eccentricity anomaly* E and the *Period* P:

$$\tan\frac{E}{2} = \sqrt{\frac{1-e}{1+e}}\tan\frac{\theta}{2}, \qquad P = 2\pi\sqrt{\frac{a^3}{\mu}} \tag{2.32}$$

From Orbital Elements to Position and Velocity

In the following, the inverse transformation from the orbital element to the position and the velocity of the S/C:

$$p = a(1 - e^2), \qquad r = \frac{p}{1 + e\cos\theta} \tag{2.33}$$

r and **v** expressed in the perifocal frame are :

$$r = \begin{bmatrix} r\cos\theta \\ r\sin\theta \\ 0 \end{bmatrix}, \qquad v = \begin{bmatrix} -\sqrt{\mu/p}\sin\theta \\ \sqrt{\mu/p}(e + \cos\theta) \\ 0 \end{bmatrix} \tag{2.34}$$

In order to switch from perifocal frame to GE frame it is possible to transform the coordinates through an appropriate transformation matrix T, developed using the Euler angles 313.

$$\begin{aligned} &Transformation \quad PF \rightarrow GE: \qquad \mathbf{T}_{313}(\Omega, i, \omega) \\ &Transformation \quad GE \rightarrow PF: \qquad \mathbf{T}_{313}(-\omega, -i, -\Omega) \end{aligned} \tag{2.35}$$

2.7 Orbit Perturbations

The orbital dynamics discussed thus far have been based on Kepler's and Newton's laws, focusing on non-perturbed orbits. However, in reality, orbits are subject to various perturbations caused by different factors. The principal perturbations are the following, see [5], [3]: gravity potential harmonics perturbing the central force, due to an irregular mass distribution of planets (e.g., Earth polar flattening); third-body forces like those due to the Sun or Moon gravity; aerodynamic forces due to the residual atmosphere and wind at low-Earth orbits; solar or cosmic radiation; others, such as Earth radiation and tides, and spacecraft thermal radiation; drag force at low-Earth orbits. It is important to note that some of these perturbations may take months or even years to significantly impact spacecraft, but for the purposes of this work, a conservative non-perturbed model will be considered due to the specific scenario of low Earth orbit and the short duration of the rendezvous mission, typically lasting more or less one hour.

2.8 HCW equations

After reviewing the parameters and concepts discussed so far, it should now be clear to the reader about the key aspect of the relative motion that will be employed in this work.
The HCW (Hill-Clohessy-Wiltshire) equations are a set of linear differential equations that provide a description of the relative motion between a small satellite (chaser) and a larger parent body (target) in a circular orbit. These equations are widely used in the field of astrodynamics to model the motion of spacecraft when they are in close proximity to a non-collaborative target and both are on near-circular orbits. This model is particularly useful for planning rendezvous maneuvers between the chaser and the target. It allows for the prediction and analysis of the relative motion dynamics, facilitating the design and execution of precise rendezvous strategies. Given its relevance and applicability, the HCW model will also be employed in this work to assist in the planning and analysis of the rendezvous mission. Now its derivation will be explained.

It can be considered a three bodies system P_0, P_1, P_2 with three different masses respectively $m_0 \gg m_1, m_2$ and three different positions r_0, r_1, r_2 in the inertial frame $R = \{O, \mathbf{i}_1, \mathbf{i}_2, \mathbf{i}_3\}$ where the origin $O = P_0$. Considering the assumption on the forces in the Two body problem the second Newton's laws becomes:

$$\begin{aligned} \dot{\mathbf{v}}_i &= \sum_{i \neq j}^{2} \frac{G m_j}{r_{ij}^3}(r_j - r_i) + \frac{\mathbf{F}_i}{m_i}, \qquad i = 0{,}1{,}2 \\ r_{ji} &= |\mathbf{r}_j - \mathbf{r}_i| \end{aligned} \tag{2.36}$$

If these equations will expand they will be of the following form:

$$\begin{aligned} \dot{\mathbf{v}}_0 &= \frac{G m_1}{r_{01}^3}(\mathbf{r}_1 - \mathbf{r}_0) + \frac{G m_2}{r_{02}^3}(\mathbf{r}_2 - \mathbf{r}_0) + \frac{\mathbf{F}_0}{m_0} \\ \dot{\mathbf{v}}_1 &= \frac{G m_0}{r_{10}^3}(\mathbf{r}_0 - \mathbf{r}_1) + \frac{G m_2}{r_{12}^3}(\mathbf{r}_2 - \mathbf{r}_1) + \frac{\mathbf{F}_1}{m_1} \\ \dot{\mathbf{v}}_2 &= \frac{G m_0}{r_{20}^3}(\mathbf{r}_0 - \mathbf{r}_2) + \frac{G m_1}{r_{21}^3}(\mathbf{r}_1 - \mathbf{r}_2) + \frac{\mathbf{F}_2}{m_2} \end{aligned} \tag{2.37}$$

Since the CoM is assumed to coincide with the larger body of mass m_0, the position vector r_0 becomes zero. Consequently, the gravitational forces between the smaller bodies with respect to the one induced by the larger body can be neglected. In other words, it is assumed that each small mass is only subject to the central gravitational field generated by the point P_0 with mass m_0. The mutual gravitational interaction between P_1 and P_2 is considered to be negligible in this context.

In this way the two small bodies obey the restricted two-body equations:

$$\begin{aligned} \dot{\mathbf{v}}_1 + \mu\frac{\mathbf{r}_1}{r_1^3} = \frac{1}{m_1}\mathbf{F}_1 \\ \dot{\mathbf{v}}_2 + \mu\frac{\mathbf{r}_2}{r_2^3} = \frac{1}{m_2}\mathbf{F}_2 \end{aligned} \tag{2.38}$$

Each row of (2.38) describes the motion of each mass in the gravitational field generated by the big body, regardless the motion of the other mass.

Studying the relative motion between the two small bodies is of great significance in space applications, particularly in rendezvous scenarios.
So, to properly describe the relative motion between P_2 and P_1 is essential to define the relative position $\mathbf{r} = \mathbf{r}_2 - \mathbf{r}_1$, with $r = |\mathbf{r}|$ and if the orbit of the chief (target) is Keplerian then $\mathbf{F}_1 = 0$ and $\mathbf{F} = \mathbf{F}_2$. As a consequence, the equation (2.38) can be rewrite in the following way:

$$\begin{aligned} \dot{\mathbf{v}}_2 = \dot{\mathbf{v}}_1 - \dot{\mathbf{v}} = -\frac{\mu}{r_2^3}(\mathbf{r}_1 + \mathbf{r}) + \frac{\mathbf{F}}{m_2} = \\ = -\frac{\mu}{|\mathbf{r}_1 + \mathbf{r}|^3}(\mathbf{r}_1 + \mathbf{r}) + \frac{\mathbf{F}}{m_2} \end{aligned} \tag{2.39}$$

From here on out, the inertial frame considered is a perifocal frame built on the chief S/C $R_{p_0} = \{P_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\}$. Instead as non-inertial frame $R_{p_1} = \{P_1, \mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3\}$ is chosen built on the body P_1.
Named this two reference frames, the inertial acceleration can be described:

$$\dot{\mathbf{v}} = \dot{\mathbf{v}}_h + 2w_1 \times \mathbf{v}_h + w_1 \times (w_1 \times \mathbf{r}) + \dot{w}_1 \times \mathbf{r} \tag{2.40}$$

Where $\dot{\mathbf{v}}_h$ and $\mathbf{v}_h$ represent respectively the acceleration and velocity vectors in the R_{p_1} frame. $\dot{w}_1$ and w_1 represent the acceleration and the angular velocity vectors of P_1 seen in the inertial frame R.

Now, considering the R_h frame constructed as follow
$R_h = \{P_1, \mathbf{h}_1 = \mathbf{r}_1/r_1, \mathbf{h}_2, \mathbf{h}_3 = w_1/|w_1|\}$ the expression can be rewritten:

$$\begin{aligned} &\mathbf{r}_1 = r_1\mathbf{h}_1, \quad \mathbf{r}_h = \mathbf{r} = x\mathbf{h}_1 + y\mathbf{h}_2 + z\mathbf{h}_3 = \mathbf{H}\mathbf{r}' \\ &\mathbf{v}_h = \dot{x}\mathbf{h}_1 + \dot{y}\mathbf{h}_2 + \dot{z}\mathbf{h}_3 = \mathbf{H}\dot{\mathbf{r}}' \\ &\dot{\mathbf{v}}_h = \ddot{x}\mathbf{h}_1 + \ddot{y}\mathbf{h}_2 + \ddot{z}\mathbf{h}_3 = \mathbf{H}\ddot{\mathbf{r}}' \\ &\mathbf{F}_h = F_1\mathbf{h}_1 + F_2\mathbf{h}_2 + F_3\mathbf{h}_3 = \mathbf{H}\mathbf{F}' \end{aligned} \tag{2.41}$$

where $\mathbf{r}' = [x, y, z]$ represents the coordinates of the P_2 body in the non-inertial frame; instead $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3]$ is the matrix which contains the unit vectors of R_h

frame. It can be observed that the chief orbit is described by the FR2B equation. Then combining the equations (2.38), (2.39) and the previous one (2.41):

$$\begin{aligned}
&\mathbf{r}_1 = r_1\mathbf{h}_1, \qquad \dot{\mathbf{v}}_1 = \dot{v}_1\mathbf{h}_1 = -\frac{\mu}{r_1^2}\mathbf{h}_1 \\
&\dot{\mathbf{v}}_2 = -\frac{\mu}{r_2^3}\mathbf{r}_2 + \frac{\mathbf{F}}{m_2} = -\frac{\mu}{r_2^3}(\mathbf{r}_1 + \mathbf{r}) + \frac{\mathbf{F}}{m_2} = \\
&\quad = -\frac{\mu}{r_2^3}((r_1 + x)\mathbf{h}_1 + y\mathbf{h}_2 + z\mathbf{h}_3) + \frac{F_1}{m_2}\mathbf{h}_1 + \frac{F_2}{m_2}\mathbf{h}_2 + \frac{F_3}{m_2}\mathbf{h}_3 \\
&w_1 = |w_1|\mathbf{h}_3, \quad \dot{w}_1 = |\dot{w}_1|\mathbf{h}_3
\end{aligned} \tag{2.42}$$

Now, replacing the results obtained in the equation (2.40) and after some calculations:

$$\begin{aligned}
\dot{\mathbf{v}} = \dot{\mathbf{v}}_2 - \dot{\mathbf{v}}_1 &= -\frac{\mu}{r_2^3}((r_1 + x)\mathbf{h}_1 + y\mathbf{h}_2 + z\mathbf{h}_3) \\
&+ \frac{F_1}{m_2}\mathbf{h}_1 + \frac{F_2}{m_2}\mathbf{h}_2 + \frac{F_3}{m_2}\mathbf{h}_3 + \frac{\mu}{r_1^3}r_1\mathbf{h}_1 = \\
&= \dot{\mathbf{v}}_h + 2w_1 \times \mathbf{v}_h + w_1 \times (w_1 \times \mathbf{r}) + \dot{w}_1 \times \mathbf{r} = \\
&= \ddot{x}\mathbf{h}_1 + \ddot{y}\mathbf{h}_2 + \ddot{z}\mathbf{h}_3 + 2|w_1|\dot{x}\mathbf{h}_2 - 2|w_1|\dot{y}\mathbf{h}_1 \\
&+ |w_1|\mathbf{h}_3 \times (|w_1|x\mathbf{h}_2 - |w_1|y\mathbf{h}_1) + |\dot{w}_1|x\mathbf{h}_2 - |\dot{w}_1|x\mathbf{h}_1 = \\
&= \ddot{x}\mathbf{h}_1 + \ddot{y}\mathbf{h}_2 + \ddot{z}\mathbf{h}_3 + 2|w_1|\dot{x}\mathbf{h}_2 - 2|w_1|\dot{y}\mathbf{h}_1 \\
&- |w_1|^2 x\mathbf{h}_1 - |w_1|^2 y\mathbf{h}_2 + |\dot{w}_1|x\mathbf{h}_2 - |\dot{w}_1|x\mathbf{h}_1
\end{aligned} \tag{2.43}$$

Rearranging along the unit vectors directions and writing in function of the relative accelerations:

$$\begin{aligned}
\ddot{x} &= 2w_1\dot{y} + \dot{w}_1 y + w_1^2 x + \frac{\mu}{r_1^3}r_1 - \frac{\mu}{r_2^3}(r_1 + x) + \frac{F_1}{m_2} \\
\ddot{y} &= -2w_1\dot{x} - \dot{w}_1 x + w_1^2 y - \frac{\mu}{r_2^3}y + \frac{F_2}{m_2} \\
\ddot{z} &= -\frac{\mu}{r_2^3}z + \frac{F_3}{m_2}
\end{aligned} \tag{2.44}$$

The previous equations can be linearized for a small separation between the two small bodies, $r = |\mathbf{r}_2 - \mathbf{r}_1| \ll r_1$. Using the binomial theorem it is possible to write:

$$\frac{\mu}{r_2^3} \approx \frac{\mu}{r_1^3}\left(1 - 3\frac{x}{r_1}\right) \tag{2.45}$$

Now, considering:

$$\begin{aligned}
&\mathbf{v}_1 = w_1 \times \mathbf{r}_1 \quad \text{at the apsides:} \quad \mathbf{v}_1 = \sqrt{\mu/a}, \quad \mathbf{r}_1 = a \\
&\implies w_1 = \sqrt{\mu/a^3} \approx \sqrt{\mu/r_1^3} = \textit{constant} \\
&w = w_1, \qquad \dot{w} = 0
\end{aligned} \tag{2.46}$$

It is worth to note that: $\sqrt{\mu/a^3} \approx \sqrt{\mu/r_1^3}$ only when $e \approx 0$. Replacing the previous results, the HCW equations are obtained:

$$\begin{aligned}
\ddot{x} &= 3w^2x + 2w\dot{y} + \frac{F_1}{m_2} \\
\ddot{y} &= -2w\dot{x} + \frac{F_2}{m_2} \\
\ddot{z} &= -w^2z + \frac{F_3}{m_2}
\end{aligned} \tag{2.47}$$

Chapter 3

Nonlinear Model Predictive Control

3.1 Introduction

As mentioned in the introduction, the algorithm used to control the rendezvous mission is Nonlinear Model Predictive Control. In this chapter, the reader will understand why this type of algorithm is one of the best in such situations.

One of the key advantages of NMPC, see [10], [11] and [3], is its ability to handle nonlinear systems, which are systems that do not follow a linear relationship between inputs and outputs. This makes it particularly useful for controlling complex processes that are difficult to accurately model using traditional linear control techniques. Additionally, it allows for dealing with constraints on variables and managing the trade-off between performance and command effort.
The control approach involves predicting the behaviour of the system over a finite time interval, known as the time horizon, using a model of the plant. Subsequently, the control selects the optimal command input to achieve the best predicted state for the desired output behavior obtained through an online optimization algorithm.

NMPC can be applied to a wide variety of applications across different industries. For instance, in the chemical industry, it can be used to control chemical reactors, distillation columns, and other chemical processes.In the field of robotics, NMPC is valuable for controlling the motion of robots in various applications. The automotive industry benefits from NMPC in controlling various systems in vehicles, including engine management and active suspension systems. Additionally, NMPC finds applications in the aerospace industry, as demonstrated in this work, among many other fields.

These examples serve as just a few illustrations of the numerous applications of NMPC. Its capability to handle nonlinear systems and optimize control actions through prediction makes it a valuable tool for enhancing efficiency, reducing costs, and improving performance across various industries.

3.2 Theoretical formulation

The formulation considers a nonlinear Multiple Input - Multiple Output (MIMO) system in a general form:

$$\dot{x} = f(x,u), \quad y = h(x,u) \tag{3.1}$$

where $x \in \mathbb{R}^n$ represents the state vector, $u \in \mathbb{R}^{n_u}$ represents the command input vector and $y \in \mathbb{R}^{n_y}$ corresponds to the output vector. In a general form of the system, the output can be measured. The measurements are collected at each sampling time T_s, so they are updated in real-time:

$$x(t_k) = T_s k \tag{3.2}$$

with k = 0,1,2...
At each time $t = t_k$, the state and output of the system are predicted over a time interval $[t, t+T_p]$. The prediction is performed by integration of differential equations of the model starting from initial conditions at time t. Instead, T_p is called *prediction horizon* and it is required to be larger than T_s.
At any time $\tau \in [t, t+T_p]$, the predicted output is a function of time, of initial state $x(t)$ and the input signal:

$$\hat{y}(\tau) \equiv \hat{y}(\tau, x(t), \hat{u}(t:\tau)) \tag{3.3}$$

where $\hat{u}(t:\tau)$ represents the command input applied from initial time t to final time τ when the prediction is computed. So, $\hat{u}(t:\tau)$ denotes the input signal within the time interval $[t,\tau]$.

In the picture below, it is evident that within the prediction interval $[t, t+T_p]$, $\hat{u}(\tau)$ represents an open-loop input. Therefore, it does not rely on the state $x(\tau)$. Instead, it is the predicted state that depends on a certain command input. Consequently, for each input signal, there exists a corresponding pair of predicted state and output. If the input changes within the prediction interval, the state and output will also change accordingly. The fundamental concept of predictive control is to search for the optimal command input for the system throughout each prediction interval. This approach aims to achieve the best state vector and, consequently, the optimal output vector for the mission, while maintaining the desired behavior of the system.

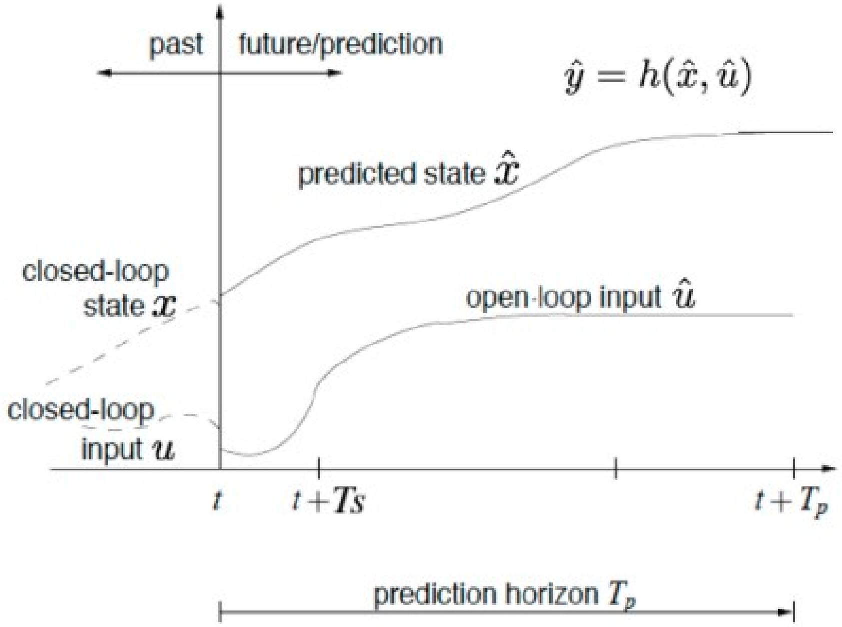


Figure 3.1: State and input vectors over prediction interval

At each time $t = t_k$, the best input signal $\hat{u}(t : \tau) = u^*(t : \tau)$ will be chosen, such that the prediction

$$\hat{y}(\tau, x(t), u^*(t : \tau)) \equiv \hat{y}(u^*(t : \tau)) \tag{3.4}$$

will have the desired behaviour for $\tau \in [t, t + T_p]$. This means that every prediction interval the right command input is chosen in order to achieve an optimal output. This concept of finding the optimal command input can be formalized by using an objective function, also known as a cost function, with the following form [12]:

$$J(u(t : t + T_p)) = \int_t^{t+T_p} (\|\tilde{y_p}(\tau)\|_Q^2 + \|u(\tau)\|_R^2) d\tau + \|\tilde{y_p}(t + T_p)\|_P^2 \tag{3.5}$$

where $\tilde{y}_P(\tau) \doteq r(\tau) - \hat{y}(\tau)$ represents the predicted tracking error and $r(\tau) \in \mathbb{R}^{n_y}$ represents the reference to track. The predicted tracking error is the difference between the reference and the predicted output over a certain time interval.
The input signal $u^*(t : t + T_p)$ is the best pick that minimizes the objective function.

This minimization occurs every time interval. Regarding the cost function, three terms can be highlighted:

- $\|\tilde{y_p}(\tau)\|_Q^2$: the predicted tracking error , that is integrated for all time interval. This term is being minimized during all the process.
- $\|u(\tau)\|_R^2$: this term allows to manage a trade-off between performance and command activity.
- $\|\tilde{y_p}(t+T_p)\|_P^2$: this term represents the predicted tracking error specifically at the final time step. Therefore, it is not part of the integrand.

The square weighted norm of a vector $v \in \mathbb{R}^n$ is defined as follow:

$$\|v\|_Q^2 \doteq v^T Q v = \sum_{i=1}^{n} q_i v_i^2, \quad Q = \text{diag}(q_1, ..., q_n) \in \mathbb{R}^{n \times n}, \quad q_i \geq 0 \tag{3.6}$$

Therefore, the coefficients R, P, and Q play a crucial role in designing the control system, and their selection will be explained in detail later.
The objective function is not the only aspect to consider when formulating an optimization problem. It is also essential to impose constraints on variables such as $\hat{y}(\tau)$ and $\hat{u}(\tau)$. Here are some examples of these constraints:

$$\begin{aligned} \dot{\hat{x}}(\tau) &= f(\hat{x}(\tau), \hat{u}(\tau)) \quad \hat{x}(t) = x(t), \quad \tau \in [t, t+T_p] \\ \hat{y}(\tau) &= h(\hat{x}(\tau), \hat{u}(\tau)). \end{aligned} \tag{3.7}$$

where $\hat{y}(\tau)$ depends on $\hat{x}$, which, in turn, is influenced by the command input $\hat{u}(\tau)$. As a result, $\hat{y}(\tau)$ is determined by the model equations based on the specific $\hat{u}(\tau)$ employed. Moreover, the initial condition of the predicted state corresponds to the measured state at the initial time t. In NMPC, the differential equation is integrated over the prediction time interval to obtain the predicted state signal $\hat{x}(\tau)$. Subsequently, the output equation is used to calculate the value of $\hat{y}(\tau)$, which is then utilized to construct the predicted tracking error within the objective function. However, when formulating the optimization problem, additional constraints are necessary to ensure that the variables satisfy the differential equations that describe the system model. It can be deduced that $\hat{y}(\tau)$ and $\hat{u}(\tau)$ are not independent variables.
Other constraints on the state, input and output vectors could be:

- $\hat{x}(\tau) \in X_c$, $\hat{y}(\tau) \in Y_c$, $\tau \in [t, t+T_p]$ for example for obstacles in the path or collision avoidance;
- $\hat{u}(\tau) \in U_c$, $\tau \in [t, t+T_p]$ for example for input saturation.

3.3 Intuitive idea

The intuitive idea of the NMPC controller can be visually represented by the two picture below. On the left, an example of a prediction interval is depicted, showing all the possible command inputs and their corresponding output signals. The red line represents the optimal command input that yields the best output, minimizing the predicted tracking error. On the right, an illustration showcases the application of NMPC in obstacle avoidance, where various possible paths to follow are depicted. The choice of these paths depends on the tuning of the parameters in the objective function.

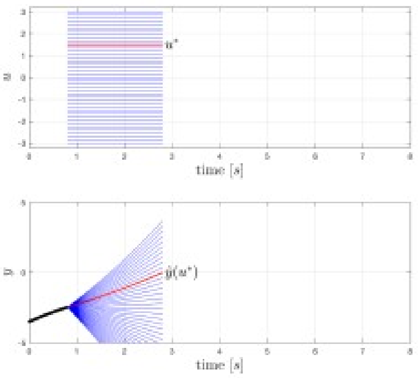


Figure 3.2: Intuitive idea

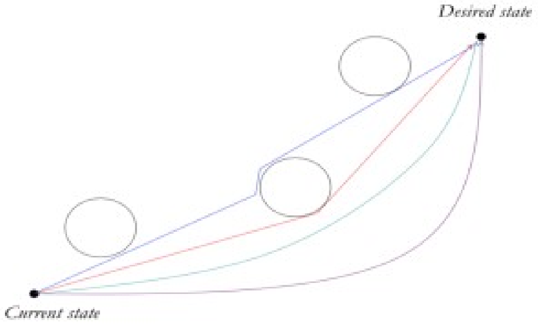


Figure 3.3: Obstacle avoidance

3.4 Mathematical Formulation

In this section the mathematical formulation of the NMPC control will be presented. At each time $t = t_k$ for $\tau \in [t, t + T_p]$ the following optimization problem is solved:

$$\begin{aligned}
& u^*(t : t + T_p) = arg\min_{u(\cdot)} \quad J(u(t : t + T_p)) \\
& subject \quad to : \\
& \dot{\hat{x}}(\tau) = f(\hat{x}(\tau), u(\tau)), \quad \hat{x}(\tau) = x(t) \\
& \hat{y}(\tau) = h(\hat{x}(\tau), u(\tau)) \\
& \hat{x}(\tau) \in X_c, \quad \hat{y}(\tau) \in Y_c, \quad u(\tau) \in U_c \\
& u(\tau) = u(t + T_c), \quad \tau \in [t + T_c, t + T_p]
\end{aligned} \tag{3.8}$$

where $\hat{u}(\tau)$ is considered constant during the interval. T_s represents the sampling time, T_c represents the control horizon, T_p denotes the prediction horizon and

$0 \leq T_s \leq T_c \leq T_p$. The control horizon is used to reduce the computational load. Indeed, the optimization problem mentioned above is generally non-convex due to the non-convex nature of the objective function J and the possibility of having non-convex sets of constraints X_c, Y_c, and U_c. Consequently, efficient numerical algorithms are required to find solutions to such problems.
In mathematics, a real-valued function is defined as convex if the line segment connecting any two points on the graph of the function lies entirely above or on the graph between those two points. Another equivalent definition is that a function is convex if its epigraph, which represents the set of points lying on or above the graph of the function, is a convex set. A non-convex function is wavy, with some 'valleys' (local minima) that are not as deep as the overall deepest 'valley' (global minimum).
Thanks to an iterative process, the algorithm begins from an initial point x_0. Using the gradient function, which indicates the direction of maximum variation, the algorithm explores points in the vicinity of x_0 until it finds a minimum. A fundamental challenge in this approach is that it is typically impossible to guarantee whether the discovered minimum is a local or global minimum. This uncertainty is determined by the choice of the starting point x_0. This issue is not specific to the algorithm used but rather a mathematical problem. Solving a multivariate nonlinear function is inherently difficult, and the complexity of the problem generally increases exponentially with the dimension of the decision variable vector. A local minimum refers to a point where the objective function attains its minimum value within a neighborhood, while a global minimum represents the point where the objective function attains its minimum value across the entire domain of the function.

The optimization problem is non-convex due to the non convexity of the cost function J or one of the constraints. This means that there can be several minima. So, the NMPC optimization algorithm cannot guarantee whether the found minimum is global or not. Furthermore, the sets X_c, Y_c and U_c can be non-convex. In the end, the NMPC usually works with a non-convex optimization problem.
Another feature of the optimization problem is that it must be solved online at each sampling time. In NMPC, it is crucial to have an efficient optimization algorithm capable of solving the problem within a reasonable time frame. Moreover, due to the non-convexity, the algorithm must be sufficiently reliable to ensure that the obtained minimum is sufficiently close to the global minimum, thereby fulfilling the control objectives.

3.4.1 Receding control horizon strategy

The Receding Control Horizon principle is a fundamental concept in Nonlinear Model Predictive Control. The basic idea is that, at each time step, the control action is computed based on a finite prediction horizon. However, only the first term of the command input is applied to the system. This process is repeated at each sampling time, with a new prediction horizon that considers the current state of the system.
Suppose that, at a time $t = t_k$, the optimal input signal $u^*(t : t + T_p)$ has been computed by solving the optimization problem (3.8). First of all, $u^*(t : t + T_p)$ is an open-loop input and depends on x(t) but not on $x(\tau)$, where $\tau > t$. Second of all, if $u^*(t : t + T_p)$ is applied for the entire time interval $[t, t + T_p]$, it does not provide a feedback action. Therefore, it cannot enhance precision, reduce errors and disturbance effects, or adapt to a varying scenario.
For this reason the NMPC uses the receding horizon strategy:

1. At time $t = t_k$:

   a. compute $u^*(t : t + T_p)$ by solving the optimization problem (3.8);

   b. apply only the first input value: $u(\tau) = u^*(t = t_k)$ and keep it constant for $\forall\tau \in [t_k, t_{k+1}]$.

2. Repeat steps 1a-1b for $t = t_{k+1}, t_{k+2}, ...$

where only the first element of the command input is sent to the system.

3.5 NMPC Design and Algorithm

3.5.1 NMPC Algorithm

In the following the general working principle of the algorithm:

1. At time $t = t_k$, for $\tau \in [t, t + T_p]$, solve the optimization problem

$$\begin{aligned}
&u^* = arg \min_{u \in \mathbb{R}^{n_u \times m}} \quad J(u) \\
&subject \quad to: \\
&\dot{\hat{x}}(\tau) = f(\hat{x}(\tau), u_p(\tau)), \quad \hat{x}(t) = x(t) \\
&\hat{y}(\tau) = h(\hat{x}(\tau), u_p(\tau)) \\
&\hat{x}(\tau) \in X_c, \quad \hat{y}(\tau) \in Y_c, \quad u_p(\tau) \in U_c \\
&u_p(\tau) = u_p(t + T_c), \quad \tau \in [t + T_c, t + T_p]
\end{aligned} \tag{3.9}$$

   - Open-loop optimal input: $u_p^*(\tau), \forall\tau \in [t_k, t_k + T_p]$.

- Closed-loop control law: $u(\tau) = u_p^*(t_k), \forall \tau \in [t_k, t_{k+1}]$.

2. Repeat step 1 for $t = t_{k+1}, t_{k+2}, ...$ and so on.

3.5.2 NMPC Design

Control scheme

The typical control scheme in NMPC is as follows. The NMPC algorithm aims to find the optimal solution to track the reference signal received as input. The output of the NMPC becomes the input signal for the system plant, which is then utilized to achieve the desired behavior and output.
Plant:

$$\dot{x} = f(x, u), \quad y = h(x, u) \tag{3.10}$$

Model plant contained in the NMPC block:

$$\dot{\hat{x}} = \hat{f}(\hat{x}, \hat{u}), \quad \hat{y} = \hat{h}(\hat{x}, \hat{u}) \tag{3.11}$$

In the nominal case, the real model and the predicted model are assumed to be equal, denoted as $\hat{f} = f$ and $\hat{h} = h$. However, in practice, these models are always different, reflecting the discrepancies between the predicted and actual behavior of the system.

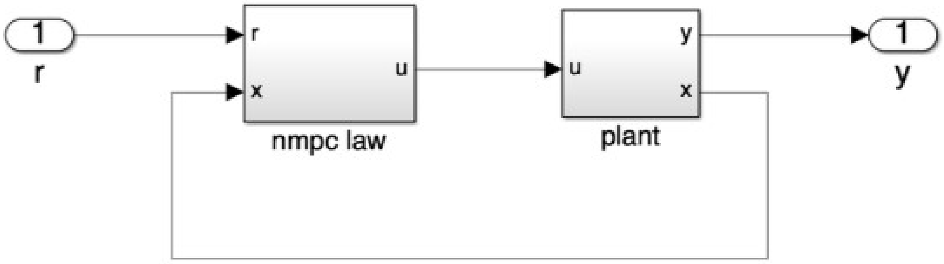


Figure 3.4: Control Scheme

Choice of Parameters

Here are the principal parameters of the NMPC algorithm, along with a brief explanation of how to choose them.

- T_p = A large value of T_p improves the closed-loop stability properties, but a too large value may result in reduced short-time tracking accuracy.

- T_s = It should be chosen as a small enough value to capture the dynamics of the system accurately. However, it should not be too small to avoid numerical problems and slow computations.

- T_c = In many applications, setting T_c to be equal to T_s is sufficient. Small values of T_c can reduce computational time without significantly impacting performance.

- m = In many cases, a low value of m is sufficient to achieve satisfactory behavior. For example, setting $m = 1$ often works well in various situations.

Choice of Weighted Matrices

As seen in the previous pages, the cost function in NMPC is defined using three different weighted matrices: *Q, P* and *R*. These matrices are related to three different aspect of the problem. Q is related to the predicted tracking error over the entire time interval. It quantifies the importance of minimizing the deviation between the predicted and desired outputs. The Q matrix needs to be positive semi-definite. P is related to the predicted tracking error at the end of the time interval. It captures the importance of achieving the desired output precisely at the final time. The P matrix needs to be positive definite. R is associated to the control input signal. It represents the cost or penalty on the control effort. The R matrix needs to be positive definite.
It is convenient to choose these matrices as diagonal, in fact as initial possible choice can be the following.

$$Q_{ii} = \begin{cases} 1, & \text{in the presence of requirements on} \quad y_i \\ 0, & \text{otherwise} \end{cases}$$

$$P_{ii} = \begin{cases} 1, & \text{in the presence of requirements on} \quad y_i \\ 0, & \text{otherwise} \end{cases}$$

$$R_{ii} = \begin{cases} 1, & \text{in the presence of requirements on} \quad u_i \\ 0, & \text{otherwise} \end{cases}$$

It is worth noting that the matrices are related to the signal norms, and the norms are related to the energy. Changing the diagonal coefficients means changing the energy of the associated signals.
So, a trial procedure could involve changing the values of Q_{ii}, P_{ii} and R_{ii} until the requirements are satisfied. Increasing Q_{ii} and P_{ii} means decreasing the energy of x_i and y_i,respectively, which can help reduce oscillations and the convergence time. On the other hand, increasing R_{ii} is equivalent to decreasing the energy of u_i, resulting in reduced command effort and energy consumption.

3.6 Advantages and Drawbacks

NMPC has several advantages and drawbacks, as with any control system. Here is an introduction to some of the key advantages and drawbacks of NMPC:

- Advantages :
  - *i)* General and flexible: it can handle complex nonlinear MIMO systems;
  - *ii)* intuitive formulation based on optimal concepts;
  - *iii)* It manages different constraints and input saturation at the same time;
  - *iv)* Efficient management of the performance/ input activity trade-off;
  - *v)* It is able to find optimal trajectories, over a finite prediction interval ;
  - *vi)* NMPC is competent to compute optimal trajectory and control law together.
- Drawbacks :
  - *i)* There are high on-line computational cost;
  - *ii)* There could be found local minima in the optimization problem;
  - *iii)* Problems in the case unstable zero-dynamics.

3.7 Adaptive Horizon MPC

As explained before, NMPC plays a key role in achieving the proposed task in this work. In order to improve the method and make the problem easier to solve while reducing computational time, the Adaptive Horizon MPC (AHMPC) will be introduced.

Firstly, Adaptive Horizon Model Predictive Control (AHMPC) is based on the formulation and assumptions of NMPC as described before. The main difference between the two algorithms lies in the modification of the prediction horizon interval. AHMPC is a scheme that allows for varying the length of the prediction horizon in Model Predictive Control as needed, see [13]. Its objective is to achieve stabilization with as small prediction horizons as possible, enabling a faster control algorithm and addressing complicated dynamic processes. The practical reason behind this is that using a shorter horizon in the terminal cost function will reduce the dimension of decision variables in the online program.

AHMPC addresses this limitation by adaptively adjusting the prediction horizon during runtime of the code. It continuously evaluates the system's performance and updates the horizon length accordingly to improve control performance. This adaptive mechanism enables the controller to respond to changes in the system's behavior and optimize control actions more effectively.

In addition to the standard requirements of NMPC, which include a terminal cost that serves as a control Lyapunov function, AHMPC also requires a terminal feedback that transforms the control Lyapunov function into a standard Lyapunov function for the closed-loop dynamics within a certain domain around the operating point. It is not necessary to precisely determine this domain in advance; AHMPC computes it online and verifies in real-time if the state is within the domain to assess if stabilization is being achieved.

In this work, the AHMPC is implemented by gradually reducing the prediction horizon interval by one sampling time each time the algorithm determines the best command input, starting when the state vector x approaches the target. The new prediction horizon, denoted as $T_{p_{new}}$, is calculated as the previous prediction horizon, $T_{p_{old}}$, minus the sampling time T_s:

$$T_{p_{new}} = T_{p_{old}} - T_s \tag{3.12}$$

This reduction in the prediction horizon allows the algorithm to adapt and optimize control actions more quickly as the system approaches the desired state.

Chapter 4

Pontryagin's Minimum Principle

This chapter will present the Pontryagin's Minimum (or Maximum) Principle, which is used in optimal control theory to determine the optimal control inputs for guiding a dynamical system from one state to another. In this work, this principle is employed to solve the optimization problem introduced in the NMPC chapter. However, the procedure will be discussed in the subsequent sections.

As demonstrated in the previous chapters, the NMPC approach involves selecting the optimal input signal by solving an optimization control problem. In this regard, Pontryagin's Minimum Principle is utilized due to its ability to provide an explicit control law, which is rarely available. Essentially, the principle introduces the Hamiltonian scalar function and the Lagrange variables, enabling the transformation of the original Optimal Control Problem (OCP) into a Two Point Boundary Value Problem. This transformation allows for the determination of the gains necessary for the explicit control law. The significance of the maximum principle lies in its simplification of the control problem by maximizing (or minimizing) the Hamiltonian, making it more manageable compared to the original infinite-dimensional control problem.

The advantages of the process used are the following:

i) the gradient of the Hamiltonian with respect to the input vector provides an efficient explicit strategy for the optimal guidance;

ii) input and state constraints, usually nonlinear, are incorporated within the OCP by adding penalty functions, see [14];

iii) tuning of the NMPC design parameters requires a minor effort;

iv) the PMP-based solution eliminates the need for input parameterization, unlike direct methods such as SQP (Sequential quadratic programming), resulting in a better accuracy in tracking the reference and improved error cancellation.

As will be shown in the next sections, the PMP is based on certain assumptions regarding the Hamiltonian function, which are crucial for computing an explicit control law.

4.1 Problem formulation

This approach is necessary to find an optimal solution for the NMPC, so it is important to remember the many features described in the previous chapters regarding the control algorithm.
Firstly, the optimization problem requires input parameterization in order to have a finite dimension. however, this issue is mitigated when the PMP approach is used , as it does not require any a-priori parameterization of the control signal. From this point forward, the implementation of the algorithm will be explained, with reference to [15] and [16].
It should be noted that Pontryagin's minimum principle does not always work with a system in a general form. In fact, the algorithm presented here considers an affine-in-the-input nonlinear system, which is commonly used due to the fact that many space models are typically in affine form.

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t) \tag{4.1}$$

where $x \in \mathbb{R}^{n_x}, u \in \mathbb{R}^{n_u}$ represent the state vector and the input vector of the system. It's implicit that the output coincides with the state vector.
Considering the cost function J (3.5) explained in the previous chapter with the three diagonal matrices $\mathbf{Q} = \mathbf{Q}^T \geq 0, \mathbf{P} = \mathbf{P}^T \geq 0$, and $\mathbf{R} = \mathbf{R}^T > 0$, where $\mathbf{Q}, \mathbf{P} \in \mathbb{R}^{n_x \times n_x}$ and $\mathbf{R} \in \mathbb{R}^{n_u \times n_u}$.
The optimization problem to solve has the following form:

$$\begin{aligned} & u^*(t : t + T_p) = arg\min_{u(\cdot)} \quad J(u(t : t + T_p)) \\ & subject \quad to: \\ & \dot{\hat{x}}(\tau) = f(\hat{x}(\tau)) + g(\hat{x}(\tau))u(\tau)), \quad \hat{x}(\tau) = x(t) \\ & \hat{x}(\tau) \in X_c, \quad u(\tau) \in U_c \end{aligned} \tag{4.2}$$

with X_C and U_C as state and input sets of constraints.
As explained for NMPC, the receding horizon strategy is also employed in this context.

Let be the following assumptions:
Assumption 1: Let $f \in \mathscr{C}^1(\mathbb{R}^{n_x} \to \mathbb{R}^{n_x})$ and $g \in \mathscr{C}^1(\mathbb{R}^{n_x} \to \mathbb{R}^{n_x})$.
Assumption 2: The admissible control set $U_C \subseteq \mathbb{R}^{n_u}$ is $U_C = \{u \in \mathbb{R}^{n_u} : u_{i_{min}} \leq u_i \leq u_{i_{max}}\}, i = 1, ..., n_u$.
Assumption 3: The state constraint set is $X_C = \{x \in \mathbb{R}^{n_x} : C(x) \leq 0\}$. Here, $C(x) \in \mathscr{C}^1(\mathbb{R}^{n_x} \to \mathbb{R})$ is generally a non convex function.

Since the state constraints will be incorporated into the cost function as an appropriate penalty term (which will be explained in more detail later), there won't be a significant distinction between the unconstrained and constrained cases. Therefore, for simplicity, the unconstrained optimization problem will be presented. According to [17], in order to find the appropriate trajectory x(t) and the optimal solution u(t), the Hamiltonian scalar function $H(x(t), u(t), \lambda(t)) \in \mathscr{C}^k(\mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_x} \to \mathbb{R})$ has to attain its minimum value $u = u^*$. In the same time the differential equations (4.1), the time evolution of Lagrangian multipliers λ (called co-variables) and a set of boundary conditions for optimality, see [18], have to be satisfied. Keeping implicit the time dependence the Hamiltonian is defined as follow:

$$H = \tilde{x}_P^T \mathbf{Q} \tilde{x}_P + u^T \mathbf{R} u + \lambda^T (f + gu) \tag{4.3}$$

While the Pontryagin formulation of the NMPC control problem has the form:

$$\begin{aligned}
&(x^*, u^*, \lambda^*) = arg\min_{u(\cdot)} \quad H \\
&subject \quad to: \\
&\dot{x} = f + gu \\
&\dot{\lambda} = -\nabla_x H^T \\
&x_k - x(t_k) = 0 \\
&\lambda^T(t_k) = -\mu^T \\
&\lambda^T(t + T_p) = 2\mathbf{P}\tilde{x}_P(t + T_p)
\end{aligned} \tag{4.4}$$

From the first two rows of the problem (4.4), it is evident that the dynamics of both the state and co-state variables are involved. These rows are called the Euler-Lagrange differential equations. In addition to these equations, the remaining rows in the problem formulation impose boundary conditions on both the state and co-state variables at the boundaries of the prediction horizon. These boundary conditions ensure the continuity of the variables, that cannot be arbitrarily chosen. For this reason at each time $t = t_k, x_k = x(t_k)$, in order to ensure the continuity between two successive steps. In the same way the state continuity constraint is represented by $\lambda(t_k) = -\mu$ at $t = t_k$. The Euler-Lagrange equations - describing the λ time evolution - take the compact form of:

$$\dot{\lambda} = -\nabla_x (H(x, u, \lambda)) \tag{4.5}$$

which, thanks to (4.3), becomes:

$$\dot{\lambda} = -\left(\lambda^T \nabla_x \left(f(x) + g(x)u\right) - 2\mathbf{Q}\tilde{x}_p\right)^T \tag{4.6}$$

Instead, in a similar manner, the optimal control law is obtained by minimizing the Hamiltonian with respect to the input variable u.

$$u^*(t) = arg\min_{u(\cdot)} \quad H(x, u, \lambda) \tag{4.7}$$

which is equivalent to solve the equation

$$\nabla_{u(\cdot)} H(x, u, \lambda) = 0 \tag{4.8}$$

In the case of affine-in-input systems, the solution will have the form:

$$u^* = -\frac{1}{2}\mathbf{R}^{-1}(\lambda^T g(x)) \tag{4.9}$$

where $\mathbf{R}$ is the diagonal, constant, square and full rank matrix introduced in the chapter 3.

By combining the PMP-based NMPC solution in (4.4) with the optimal control law in (4.9), it becomes evident how the optimal control problem turns in a Two-Points Boundary Value Problem. As a matter of fact, the differential equations of state variables, along with the Euler-Lagrange equations and the boundary conditions described in (4.4), constitute a TPBVP to solve every prediction interval $[t, t + T_p]$. The TPBVP is formalized as follows:

$$\begin{aligned} &\dot{x} = f + gu \\ &\dot{\lambda} = -\nabla_x H^T \\ &x_k - x(t_k) = 0 \\ &\lambda^T(t + T_p) = 2\mathbf{Q}\tilde{x}_P(t + T_p) \end{aligned} \tag{4.10}$$

The solution of this problem provides the λ and x variables of the explicit control law (4.9).

From the different formulas shown above, it is evident that in the optimal control law (4.9), the input $u(\tau)$ depends on $\lambda(\tau)$ and $x(\tau)$, whose values change at each sampling step of the TPBVP over the prediction horizon. Therefore, the PMP-based NMPC solution does not require a-priori parameterization of the input signal.

4.1.1 Input Constraints

In this work, the input is considered bounded linearly, such that $U_C = u(t) \in \mathbb{R}^{n_u} : u_{i_{min}} \leq u_i(t) \leq u_{i_{max}}, \forall t$. So, the optimal control input will be:

$$u^* = sat_{U_C}\left(-\frac{1}{2}\mathbf{R}^{-1}(\lambda^T g(x))\right) \tag{4.11}$$

The $sat(\cdot)$ represents the saturation operator and it is applied element-wise way to the input vector.
In other words, the limitations on the i^{th} component are performed in this way:

$$u_i^* = \begin{cases} u_{i_{min}}, & if - \frac{\lambda_i g_i(x)}{2r_i} \leq u_{i_{min}} \\ \\ u_{i_{max}}, & if - \frac{\lambda_i g_i(x)}{2r_i} \geq u_{i_{max}} \\ \\ -\frac{\lambda_i g_i(x)}{2r_i}, & otherwise \end{cases} \tag{4.12}$$

where r_i represents the i^{th} entry of the $\mathbf{R}$ diagonal. Instead, $u_{i_{min}}$ and $u_{i_{max}}$ are variables chosen based on the problem to solve.
In this work these values represent the maximum acceleration of the thrusters.

4.1.2 Path constraints

There are a lot of methods available to handle state constraints. One common approach is to relax both state and input constraints by introducing penalty terms, assuming that the nonlinear system has a well-defined relative degree.
However, in this work, the state constraint is addressed differently. It is replaced by an appropriate penalty term in the cost functional without making significant modifications to the algorithm for solving the OCP compared to the unconstrained case. By doing so, input constraints do not need to be relaxed and can be handled using the standard PMP.

To incorporate state constraints, an augmented cost function $\tilde{J}$ is constructed in such a way that when the state approaches the boundary of the forbidden set, its value increases and becomes larger than the original cost function J, $\lim_{C(x,t)\to 0} \tilde{J} \gg J$.
The approach used to augment the cost function J is by adding a suitable penalty function $k(x)$. This penalty function aims to prevent the states from approaching the boundary of the constrained set by assuming a large value, while having a null (or nearly null) value when the state is far from the boundaries.

For this reason, the augmented cost function assumes the following form:

$$\tilde{J}(u(\tau)) = J(u(\tau)) + \int_{t}^{t+T_p} \sum_{i=1}^{n} k_i(x) d\tau \tag{4.13}$$

where n means the number of state constraints to consider.
The augmented Hamiltonian is represented like this:

$$\tilde{H}(x,u,\lambda) = H(x,u,\lambda) + \sum_{i=1}^{n} k_i(x) \tag{4.14}$$

After some computations, it can be noticed that the Euler-Lagrange equations are affect by this type of modification:

$$\dot{\lambda} = -\nabla_x (H + \sum_{i=1}^{n} k_i(x)) \tag{4.15}$$

4.2 Gaussian Penalty Function

As mentioned earlier, the state constraints are incorporated as penalty functions added to the cost function J. This paragraph will demonstrate how the terms $k(x)$ are composed.

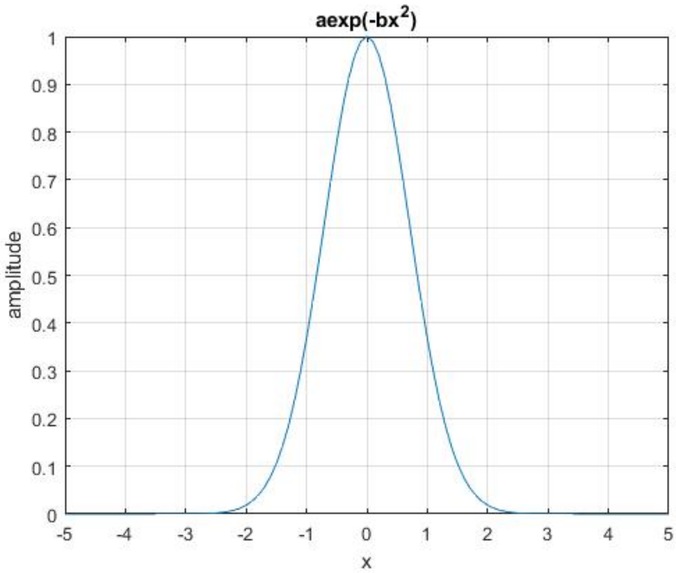


Figure 4.1: Gaussian function example

The state constraints are handled employing a Gaussian-like penalty function. In the figure above, an example of the simplest form of such a curve is depicted. The curve is represented by the equation $k(x) = aexp(-bx^2)$, as described in [19]. Here, a and b are parameters to tune by a trial and error procedure. The means of the two constants are the following: a represents the amplitude of the curve (in this case it is equal to 1); while b describes the slope of the curve, where a larger value of b corresponds to a steeper curve.
Instead in this work the state constraint is slight different assuming the following form:

$$k(x) = aexp(-bC(x)^2) \tag{4.16}$$

where *C(x)* represents the equation that describes the object or obstacle that the S/C has to avoid along the trajectory. In this work, the main objects to be avoided are represented by spheres. Therefore, the equation implemented for *C(x)* will be of the form:

$$C = d - \|r(t)\|_2 \tag{4.17}$$

where d represents the sphere radius and *r(t)* represents the position vector of the S/C.

Chapter 5

Rendezvous Mission

Firstly, in this chapter, the main topic of this work, which is the rendezvous mission, will be explained in detail. The chapter will provide a comprehensive overview of the rendezvous mission, its objectives, and the underlying principles involved.

Secondly, the chapter will focus on the presentation of the implementation of the rendezvous mission using Matlab and Simulink programs. These programs were utilized to conduct the work thesis, and the chapter will delve into the details of their application in carrying out the research.

5.1 Introduction

A space rendezvous involves a maneuver between two S/Cs that are in different circular orbits. One S/C is referred to as the target, while the other is the chaser. The rendezvous problem entails the approach of one S/C to the other.

As shown in figure below [20], the rendezvous process typically consists of several phases, including phasing, close-range rendezvous, final approaching, and docking. In the phasing phase, the chaser performs maneuvers guided by the ground telemetry tracking and command ($TT\&C$) network. These maneuvers aim to align the navigation sensors of the chaser with the target. The primary objectives of this phase include adjusting the phase angle between the two S/Cs, minimizing orbital plane differences, increasing the orbital altitude, and initiating relative navigation. During the homing phase, the chaser autonomously controls its movement, with the final position of this phase being designated as P_2. This position is a station-keeping point located a few kilometers away from the target. The key objectives of this phase are to establish the target's orbit and reduce the relative velocity between the two S/Cs.

In the closing phase, the chaser further reduces the relative distance and transitions its position to P_3, which is a station-keeping point situated hundreds of meters away from the target.

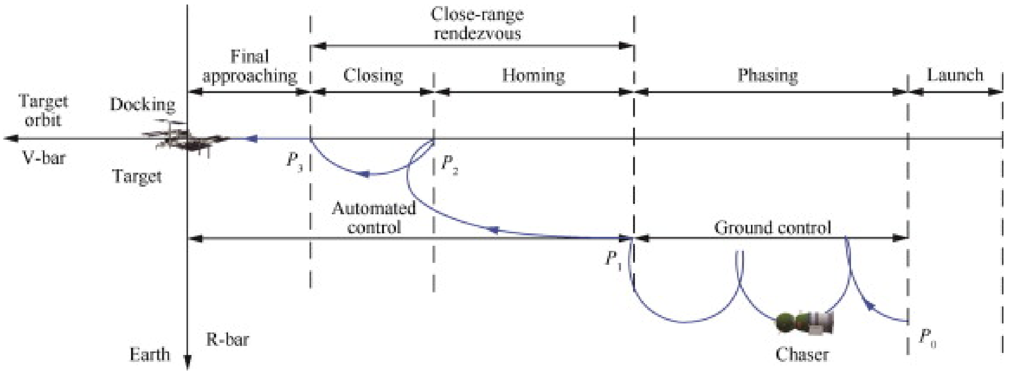


Figure 5.1: Space rendezvous [20]

This work focuses on analyzing the close rendezvous scenario, where an autonomous spacecraft (S/C) already in orbit approaches a target and ends its trajectory in a desired position close to the target. During this trajectory, a moving obstacle, represented as a sphere with its own trail, will be present and implemented as constraints to be avoided.
The algorithm implemented in this work models the obstacle and predicts its motion, assuming a uniform rectilinear trajectory. Additionally, the prediction of the constraint model incorporates additional time to allow the chaser spacecraft to observe the obstacle at closer proximity and avoid it more easily.
After successfully solving the rendezvous task, an approach to enhance performance will be implemented by reducing the prediction horizon interval of the control algorithm. The concept of reducing the prediction horizon has been explained previously, and no further explanations will be provided. The improved performance resulting from this modification will be illustrated in the subsequent chapter.

5.2 Control Algorithm

5.2.1 System model

In this section, the models used in the project are explained. Another important aspect of the project is understanding the distinction between the predictive model used in the algorithm and the real model. The physical system, known as the real plant, is composed of two S/Cs. The target follows a circular orbit around the Earth, while the chaser, initially positioned near the target, adjusts its own trajectory to align itself with the target.

Consequently, the real plant consists of these two bodies, which can be represented using the Two-Body Problem formulation. Each body is considered as a small body relative to the Earth. The dynamics of the system are influenced by the gravitational interaction between these bodies and the Earth. Accurately modeling and understanding these dynamics is crucial for simulating and controlling the real system in the project.

Indeed, the two models presented below represent the equations of the Two-Body Problem. The first model corresponds to the target, which is free to rotate around the Earth along its own trajectory.

The second model represents the chaser, which includes the command input. The chaser's trajectory needs to be adjusted in order to reach the target in the desired position. Therefore, the chaser model incorporates control inputs to facilitate this trajectory adjustment.

Target Model

$$\begin{aligned} \dot{x}_{(1:3)} &= x_{(4:6)} \\ \dot{x}_4 &= -\mu x_1/r \\ \dot{x}_5 &= -\mu x_2/r \\ \dot{x}_6 &= -\mu x_3/r \end{aligned} \tag{5.1}$$

Chaser Model

$$\begin{aligned} \dot{x}_{(1:3)} &= x_{(4:6)} \\ \dot{x}_4 &= -\mu x_1/r + u_1 \\ \dot{x}_5 &= -\mu x_2/r + u_2 \\ \dot{x}_6 &= -\mu x_3/r + u_3 \end{aligned} \tag{5.2}$$

Where μ represents the Earth gravitational parameter, $x = (x_1, x_2, x_3, x_4, x_5, x_6) = (x, y, z, \dot{x}, \dot{y}, \dot{z})$ represents the coordinates of the S/C in the Geocentric Equatorial frame. The vector $u = (u_1, u_2, u_3)$ represents the command input vector in the GE frame. In the end, r represents Euclidean norm (magnitude) of the position vectors of the S/C raised to the power of three.

$$r = \left(\sqrt{x^2 + y^2 + z^2}\right)^3 \tag{5.3}$$

Prediction Model

In contrast to the real plant of the system, the algorithm employs a predictive model that is not an exact replica but rather an approximation. The implemented prediction model in the NMPC is based on the Clohessy-Wiltshire equations, which were derived in the section (2.8). These equations serve as a simplified model for predicting the relative motion between the chaser and target spacecraft in a near-circular orbit.

$$\begin{aligned} \ddot{x} &= 3w^2x + 2w\dot{y} + u_1 \\ \ddot{y} &= -2w\dot{x} + u_2 \\ \ddot{z} &= -w^2z + u_3 \end{aligned} \tag{5.4}$$

where $u = (u_1, u_2, u_3)$ represents the command input vector of the NMPC in the LVLH frame.

Rotation Matrix

As explained, the NMPC and the real system have different models and work in two different frames. The NMPC model operates with the HCW equations, which represent the relative motion, and they are considered in the LVLH frame with the target spacecraft as the origin. On the other hand, in the real system, the target and chaser are two rockets that orbit the Earth along low orbit trajectories, meaning they are considered in the GE frame.

Therefore, in order to adapt the two models, a function is needed to transition between the frames and calculate the components of the relative motion. Based on these considerations, it is possible to determine the unit vectors of the three axes and establish the relationship between the two frames using a rotation matrix:

$$\begin{aligned} e_x &= \frac{r_0}{\|r_0\|} \\ e_z &= \frac{r_0 \times v_0}{\|r_0 \times v_0\|} \\ e_y &= e_z \times e_x \end{aligned} \tag{5.5}$$

$$R = [e_x, e_y, e_z]^T$$

Where r_0 and v_0 represent the position vector and velocity vector of the debris observed in the GE frame, respectively. Thus, R denotes the rotation matrix constructed in the GE frame to switch the reference system, thereby obtaining the coordinates in the LVLH frame.
Consequently, by considering the differences, denoted as p for the position vectors and v for the velocity vectors, between the chaser and the target, it becomes possible to derive the components of the satellite observed in the LVLH frame.

$$\begin{aligned} &p = x_c(1:3) - x_t(1:3); \quad x_{rel}(1:3) = R * p; \\ &v = x_c(4:6) - x_t(4:6); \quad x_{rel}(4:6) = R * v + x_{rel}(1:3) \times w; \end{aligned} \tag{5.6}$$

Where x_c represents the state vector of the chaser, x_t represents the state vector of the target, w denotes the angular velocity of the target, and x_{rel} represents the state vector of the chaser in the LVLH frame. The matrix R is orthogonal, so its inverse is equal to its transpose.

Below, we have the command input denoted as u, which is intended to be sent to the chaser in the GE frame. Therefore, the command input of the NMPC in the LVLH frame will be transposed into the GE frame and then sent to the chaser.

$$u = R^T * u_{rel} \tag{5.7}$$

5.2.2 System Plant

In this subsection, the Simulink plant is presented. The accompanying pictures depict both the general plant, also known as the Full Plant, and the Real Plant, which represents the actual system.

The Full Plant consists of a generic NMPC that receives the reference and the previous state of the system as inputs, and generates the command input to the system as an output. Additionally, the dynamics of the obstacle are included, and its position is sent to the NMPC to incorporate its coordinates into the algorithm. Further explanation of this topic will be provided in the subsequent sections.

On the other hand, the Real Plant showcases the two models of the chaser and target, along with the two rotation matrix functions. The function labeled as R takes the target coordinates, calculates the rotation matrix, and converts the input from the LVLH frame (used by the NMPC) into the GE frame. Similarly, the function labeled as *rot frame* takes the target coordinates, computes the position and velocity of the chaser in the LVLH frame, and subsequently sends them to the control algorithm.

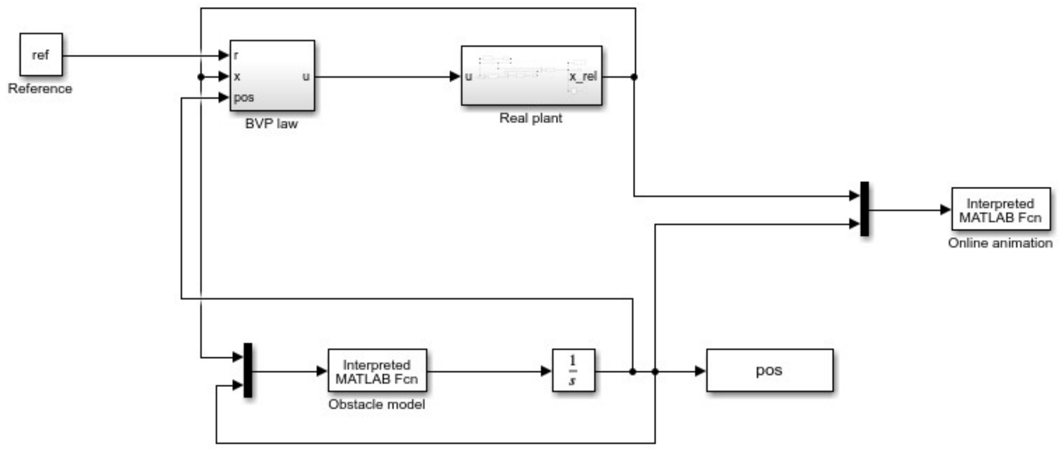


Figure 5.2: Full Plant

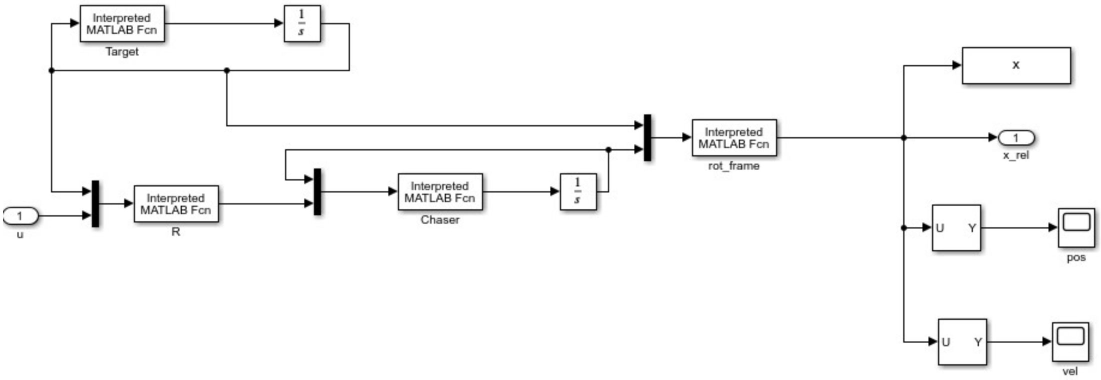


Figure 5.3: Real Plant

5.2.3 Algorithm Parameters

In the following, a list of tables will be presented, containing the parameters utilized for the simulations. In the first table, the orbital scenario parameters are listed. The table includes various parameters related to the Earth, such as the gravitational parameter used in the models. It is noteworthy that the target is positioned at an altitude of 6771 [km], indicating a low orbit, and its mean motion is equal to that of the International Space Station (ISS).

Table 5.1: Orbital Scenario Parameters

Description	Value	Unit
Earth's Planetary Const. (μ)	0.39e6	km^3s^{-2}
Earth's Radius (RE)	6371	km
Orbit Radius (r)	RE + 400 = 6771	km
Mean Motion (w)	$\sqrt{\mu/(r^3)} = 0.0011$	s^{-1}

Here are the parameters related to the initial and reference position and velocity, presented in the LVLH frame.

Table 5.2: CW Eq. Parameters

Description	Symbol	Value
Init. Position	r_0	$(-0.5, -0.1{,}0.3)^T km$
Init. Velocity	$\dot{r}_0$	$(-0.001, -0.008, -0.001)^T km/s$
Ref. Position	r_r	$(0.04{,}0{,}0)^T km$
Ref. Velocity	$\dot{r}_r$	$(0{,}0{,}0)^T km/s$

Here are the algorithm parameters that indicate the tuning employed to accomplish the desired goal.

- Sampling Time $T_s = 1$ s
- Prediction Time $T_p = 60$ s
- R-matrix $\mathbf{R} = 500 \times \mathbf{I}_{3\times3}$
- Q-matrix $\mathbf{Q} = 0.5 \times \mathbf{I}_{3\times3}$
- P-matrix $\mathbf{P} = diag(50{,}500{,}1{,}10{,}1{,}1)$
- Initial command input $u_0 = [0{,}0{,}0]^T$
- Initial co-state variables $\lambda_0 = [0{,}0{,}0{,}0{,}0{,}0]^T$
- Input saturation constraints $u_{max} = 10^{-5}\,[5{,}5{,}5]^T$

5.2.4 PMP : Case of study

In this section, all the theory discussed in the fourth chapter will be applied to the specific case studied in this work.
Therefore, the model employed in the prediction control algorithm is described by the Clohessy-Wiltshire equations. Consequently, the Hamiltonian, which is the sum of the integrand function of the cost function J and the multiplication of the co-state variables and the system, takes the following form:

$$\begin{aligned} H(x,u,\lambda) &= u^T R u + \tilde{x}_p^T \mathbf{Q} \tilde{x}_p + \lambda_x \dot{r}_x + \lambda_y \dot{r}_y \\ &+ \lambda_z \dot{r}_z + \lambda_{\dot{x}}(3w^2 r_x + 2w\dot{r}_y + u_x) \\ &+ \lambda_{\dot{y}}(-2w\dot{r}_x + u_y) + \lambda_{\dot{z}}(-w^2 r_z + u_z) \end{aligned} \tag{5.8}$$

As a result, the derivative of the Hamiltonian with respect to the state variables, which represents the Euler-Lagrange equations, can be expressed in compact form as follows:

$$\dot{\lambda} = \mathbf{A}\lambda - 2\mathbf{Q}\tilde{x} \tag{5.9}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{W} \\ -\mathbf{I}_{3\times3} & \mathbf{M} \end{bmatrix} \tag{5.10}$$

$$\mathbf{W} = \begin{bmatrix} -3w^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & w^2 \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} 0 & 2w & 0 \\ -2w & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.11}$$

And **I** represents the identity matrix. The boundary conditions on the co-state variables, $\lambda = (\lambda_x, \lambda_y, \lambda_z, \lambda_{\dot{x}}, \lambda_{\dot{y}}, \lambda_{\dot{z}})$, are imposed at the end of the prediction horizon. On the other hand, the boundary conditions on the state variables must be imposed at the beginning of the interval.

$$\begin{aligned} x_0 &= x(t_0) \\ \lambda(t_F) &= 2\mathbf{P}\tilde{x}_P(t_F) \end{aligned} \tag{5.12}$$

Now, the combination of the predictive model, described by the Clohessy-Wiltshire equations, and the Euler-Lagrange equations forms the augmented model utilized in the control algorithm:

$$\begin{aligned} \dot{s}_{(1:3)} &= x_{(4:6)} \\ \dot{s}_4 &= 3w^2 x_1 + 2wx_5 - \frac{\lambda_4}{2R_1} \\ \dot{s}_5 &= -2wx_4 - \frac{\lambda_5}{2R_2} \\ \dot{s}_6 &= -w^2 x_3 - \frac{\lambda_6}{2R_3} \\ \dot{s}_7 &= -3w^2\lambda_4 - 2\mathbf{Q}_1(x_1 - y_{r_1}) - der_{y_1} - gx \\ \\ \dot{s}_8 &= -2\mathbf{Q}_2(x_2 - y_{r_2}) - der_{y_2} - gy \\ \\ \dot{s}_9 &= w^2\lambda_6 - 2\mathbf{Q}_3(x_3 - y_{r_3}) - der_{y_3} - gz \\ \\ \dot{s}_{10} &= 2w\lambda_5 - \lambda_1 - 2\mathbf{Q}_4(x_4 - y_{r_4}) \\ \\ \dot{s}_{11} &= -\lambda_2 - 2w\lambda_4 - 2\mathbf{Q}_5(x_5 - y_{r_5}) \\ \\ \dot{s}_{12} &= -\lambda_3 - 2\mathbf{Q}_6(x_6 - y_{r_6}) \end{aligned} \tag{5.13}$$

The factor $\frac{\lambda_i}{2R_i}$ represents the components of the command input. This derive from $u = -\lambda^T/2R$, which is the optimal command input in this work, due to the matrix $g = [0_{3\times3}; I_{3\times3}]$.

Another important matter to note is the presence of six terms $der_{y_1}, der_{y_2}, der_{y_3}$ and gx, gy, gz. They represent the constraints on the position components that guide the S/C to avoid a specific surface. Here are the equations implemented in the model that describe these constraints.

$$\nabla_x k(x) = \begin{bmatrix} -2abr_x exp[-b(\|r\|_2 - d)^2]\frac{\|r\|_2-d}{\|r\|_2} \\ -2abr_y exp[-b(\|r\|_2 - d)^2]\frac{\|r\|_2-d}{\|r\|_2} \\ -2abr_z exp[-b(\|r\|_2 - d)^2]\frac{\|r\|_2-d}{\|r\|_2} \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.14}$$

These equations simply represent the partial derivatives of the Gaussian penalty functions with respect to the co-state variables. As mentioned earlier, the term d represents the radius of the sphere. This term is generic because it assumes different values depending on the specific application.

The first implemented constraint involves a sphere around the target with a radius of 10 $[m]$. This constraint is designed to prevent collision with the S/Cs. The coefficients used to enforce this constraint are $a = 1000$ and $b = 500$.

The second constraint is the focal point of this project, which involves avoiding a moving obstacle during the trajectory. The implementation of this constraint as part of the predictive model will be discussed in the following sections.

5.2.5 Obstacle Dynamics

As mentioned earlier, to make the rendezvous maneuver more realistic, it is important to consider the possibility of encountering obstacles or orbital bodies such as meteorites along the trajectory. In this work, a moving obstacle, with its own trail, is implemented to simulate such scenarios. In this case, the obstacle is moving in the direction of the chaser, and the chaser is equipped to predict its movement and take evasive action accordingly. The model of the meteorite is incorporated into the augmented model of the NMPC as state constraints on the co-state variables. This allows the chaser to be aware of the presence of the obstacle and determine the appropriate path to avoid it.

The obstacle is represented as a sphere with a radius of 70 meters followed by another sphere of same size that represents the trail. The bodies follow a uniform rectilinear motion. The initial conditions for the obstacle are taken from the unconstrained trajectory of the chaser, ensuring a collision would occur in the absence of any avoidance maneuver. The obstacle starts moving from a point that the chaser crosses, with a velocity equal to that of the spacecraft.

Table 5.3: CW Eq. Parameters

Description	Symbol	Value
Init. Position	p_0	$(0.3703, 0.2056, 0.1169)^T km$
Init. Velocity	$\dot{p}_0$	$(0.002, -0.0045, 0.002)^T km/s$

In fact, the model used for the moving obstacle is as follows:

$$\begin{aligned}
\dot{p}_1 &= p_4 \\
\dot{p}_2 &= p_5 \\
\dot{p}_3 &= p_6 \\
\dot{p}_4 &= 0 \\
\dot{p}_5 &= 0 \\
\dot{p}_6 &= 0
\end{aligned} \tag{5.15}$$

where $p = (p_1, p_2, p_3, p_4, p_5, p_6)$ represents the state vector of the obstacle, p_1, p_2, p_3 denote the position, while p_4, p_5, p_6 indicate the velocities.
The first image below illustrates how the body center is part of the unconstrained trajectory of the chaser. Without any constraints, a collision would occur between the two bodies.

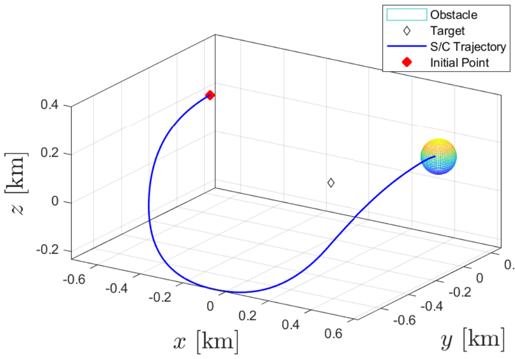


Figure 5.4: Collision

The second picture demonstrates the motion of the obstacle as it approaches the S/C, starting from the yellow point.

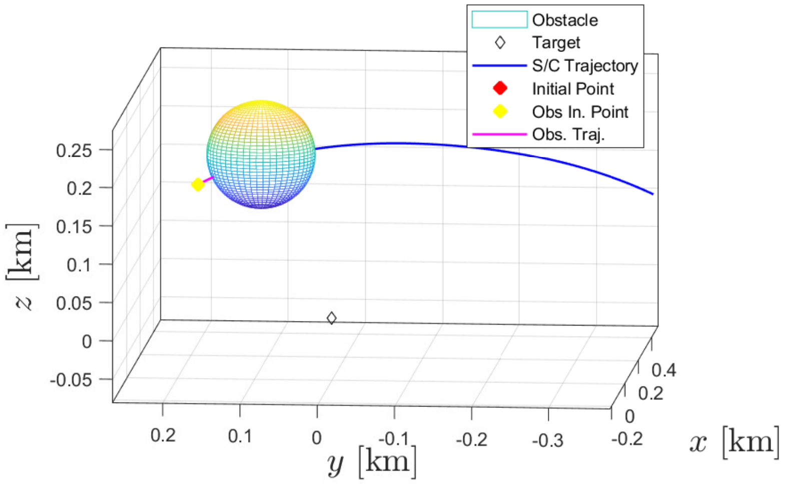


Figure 5.5: Zoom of the collision

After explaining the motion of the obstacle from the perspective of the obstacle itself, it is important to illustrate the avoidance strategy from the viewpoint of the S/C.

It should be noted that the spacecraft does not perceive a fixed sphere that changes its origin at each prediction horizon interval. In the model, both the position and velocity of the obstacle are considered, enabling the chaser to integrate the obstacle's model during each prediction interval. To achieve this, the command *ode23* is used.

The chaser is not only integrating the model of one body, but rather two spheres to avoid both the immediate presence and the trail of the moving obstacle. Consequently, the chaser will perceive two spheres approaching it. To enhance the predictive capability regarding the obstacle, the spacecraft integrates the model for an extended period of time. Therefore, a time increment of Δt of 5 *sec* is added to the prediction time. This approach ensures that the chaser has prior awareness of the obstacle, allowing sufficient time to avoid any potential collision.

Similarly in [21], the line of code implemented in the model has the following form:

$$[t, p_i] = \text{ode23}(@(t, p_i) obstacle(t, p), linspace(0, T_p + \Delta t, n), p) \tag{5.16}$$

Where *obstacle* represents the model of the object described earlier. The variable n is an integer that determines the number of divisions within the prediction interval. This simplifies the implementation of two spheres, with one positioned at the end of the prediction interval and the other in the middle of it. The variable p represents the initial condition state vector of the obstacle, and p_i denotes the resulting vector after integration. The constraints applied to avoid collisions with the obstacle are similar to those used for a general sphere. However, in this case, two spheres are present, positioned closer to the chaser. To enforce avoidance behavior, the coefficients $a = 700$ and $b = 5$ are employed. These coefficients play a crucial role in determining the strength of the avoidance behavior, enabling the chaser to navigate safely around the obstacle while fulfilling its mission objectives.

Chapter 6

Simulations Results

In this chapter, the simulation results will be presented, beginning without constraints and subsequently incorporating the moving obstacle. Finally, the outcomes with the variable prediction horizon will be presented for a comparison of the two approaches.

The simulations have been run with the plants explained in the chapter before with the solver *ode2* with *FixedStep* equal to $T_{sim} = 1$.

6.1 Unconstrained Trajectory

The following plots depict the unconstrained trajectory and its components.

6.1.1 Geocentric Equatorial Frame

The results presented below are considered in the Geocentric Equatorial Frame. These coordinates reveal that the trajectories of the target and chaser spacecrafts are consistently superimposed. However, the image below is provided to help the reader comprehend the circular orbit of both S/Cs around the Earth.

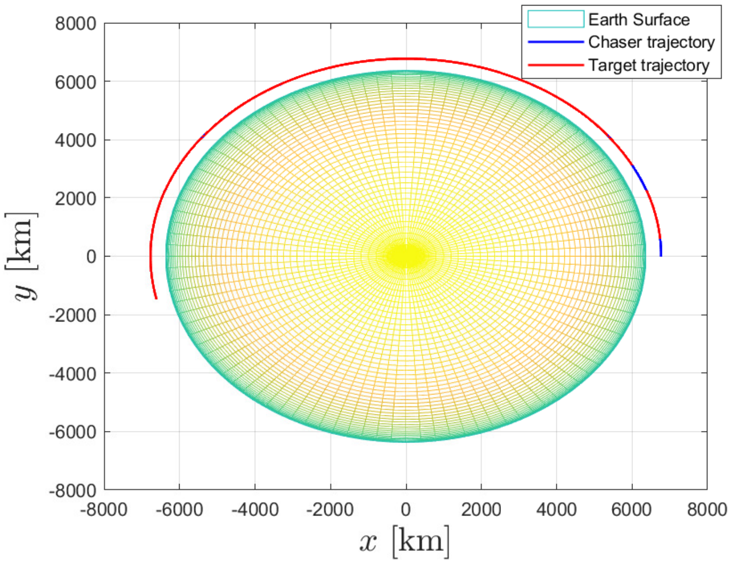
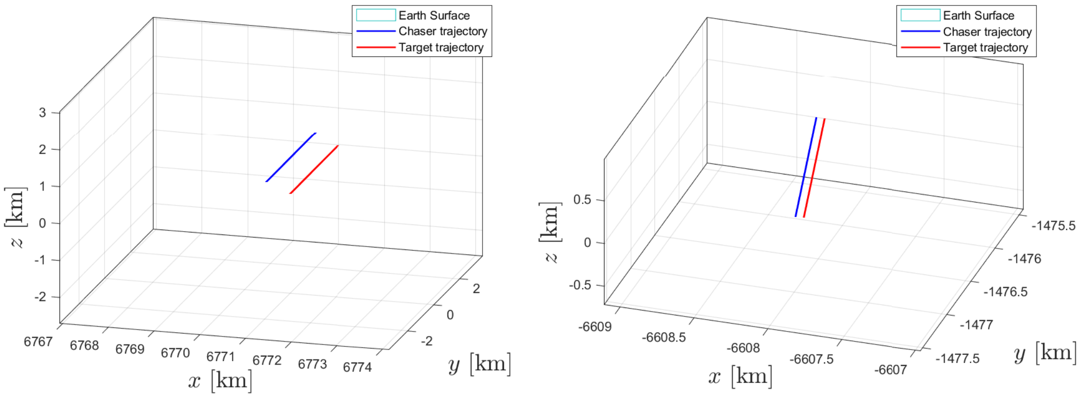


Figure 6.1: Trajectory in GE frame

By zooming in on the initial and final points of the trajectory, it becomes evident that the distance between the two spacecrafts decreases. Additionally, the alignment on the z-coordinate is clearly visible.



(a) Zoom Initial point

(b) Zoom Final Point

Figure 6.2: Zoom of trajectory

6.1.2 Local-Vertical-Local-Horizontal Frame

Now, the results are presented in the Local Vertical Local Horizontal (LVLH) frame, with the target as the origin.
The first plot illustrates the trajectory of the chaser, starting from the red point and concluding at the green point.

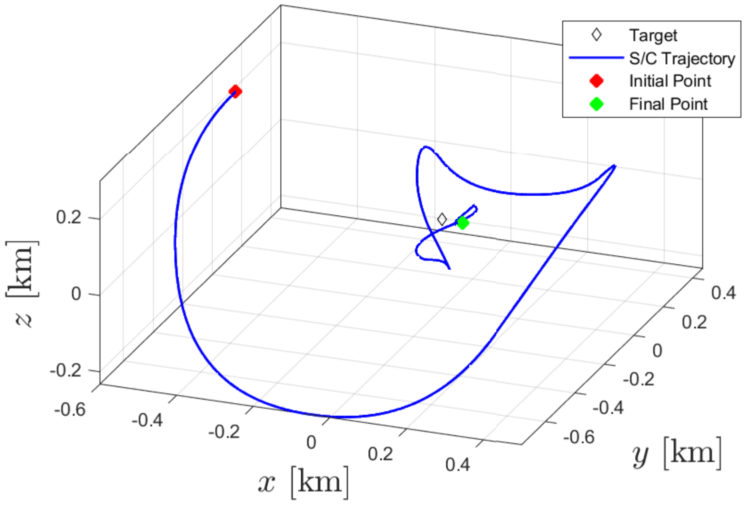


Figure 6.3: Unconstrained Trajectory

6.1.3 Components

The reference point is a specific location, but there is an acceptable tolerance defined as $tol = [0.015; 0.01; 5e-3]$ $[km]$ on the position and $[5e-4; 5e-4; 5e-4]$ $[km/s]$ on the velocity. The time it takes to reach the defined reference zone is:

$$t = 2700[s] = 45[min] \tag{6.1}$$

Once this reference zone is reached, the thrusters are deactivated or turned off.

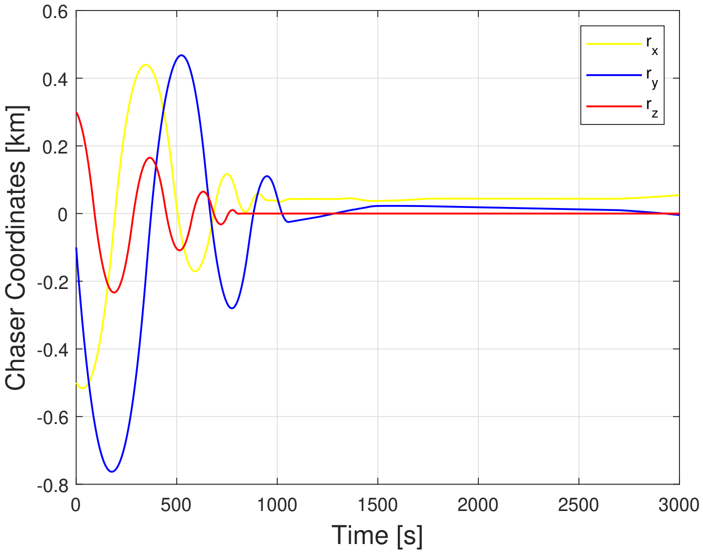


Figure 6.4: Position Coordinates

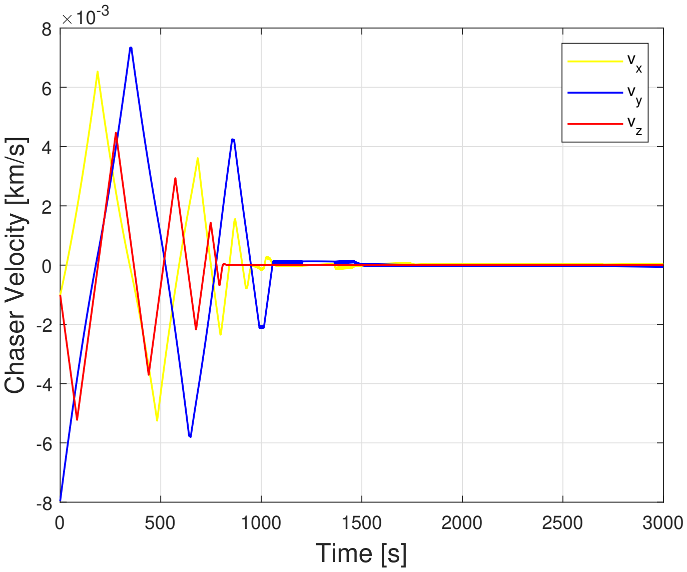


Figure 6.5: Velocity Coordinates

Here are the representations of the command inputs sent from the NMPC to the system in the LVLH frame. These inputs are divided into three coordinates, and it can be observed how the S/C reaches the designated zone around 1200 $[sec]$, gradually reducing its velocity until it comes to a complete stop at 2700 $[sec]$.

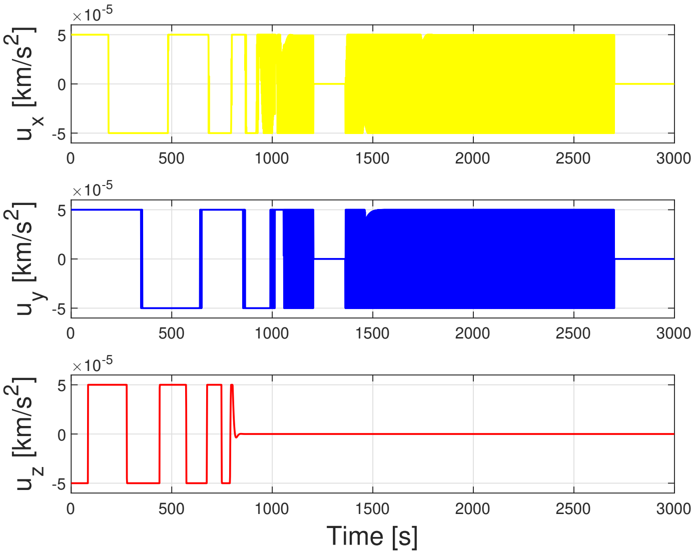
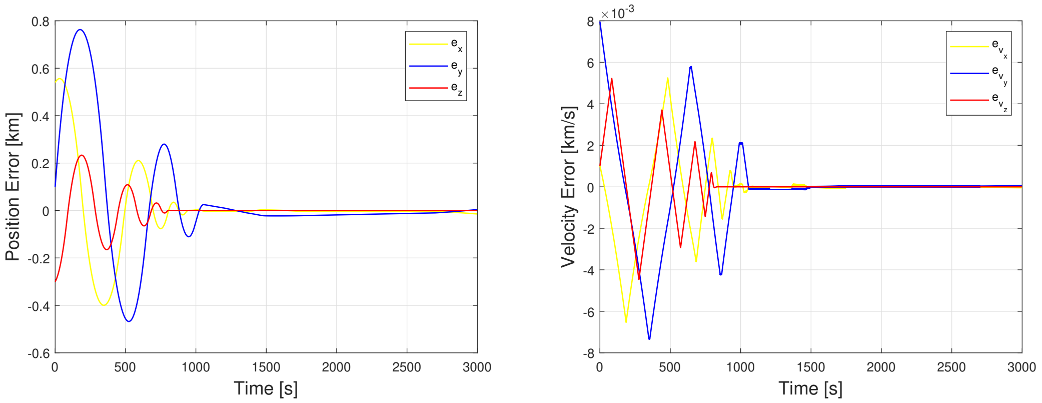


Figure 6.6: Command input

6.1.4 Errors

These plots display the errors, with the left side showing the errors relative to the position coordinates and the right side showing the errors relative to the velocity coordinates. These results are calculated by taking the difference between the state vector and the reference vector for the entire duration of the simulation.



(a) Position Error

(b) Velocity Error

Figure 6.7: Components errors

6.2 Constrained Trajectory

From now on, the plots will focus exclusively on the LVLH frame to emphasize the work thesis centered around obstacle avoidance. In this section, the accomplishment of the task is illustrated, including the trajectory with the implementation of the sphere's motion.

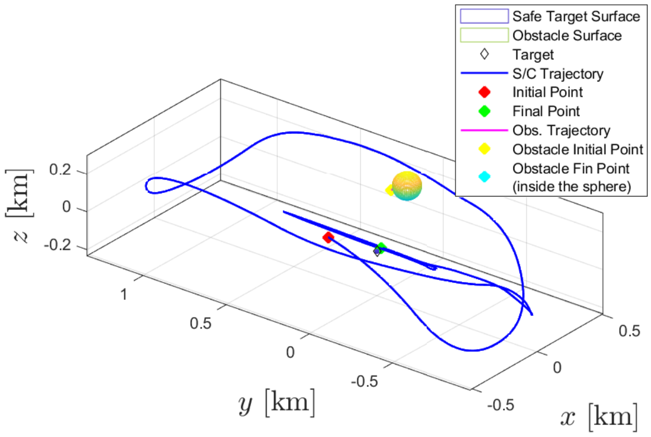


Figure 6.8: Constrained Trajectory

6.2.1 Constraints

The picture below demonstrates how the satellite is able to reach the desired position without entering the safe zone around the target.

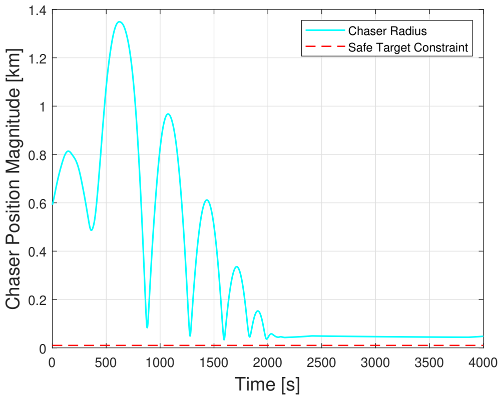


Figure 6.9: Safe Target

This plot is generated by calculating the Euclidean norm between the S/C and the origin of the obstacle. The minimum distance is achieved around 393 $[sec]$, measuring approximately 87 $[m]$. As the distance always remains greater than the radius of the sphere, a collision is successfully avoided.

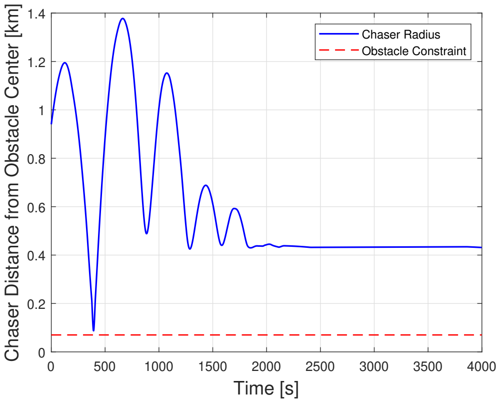
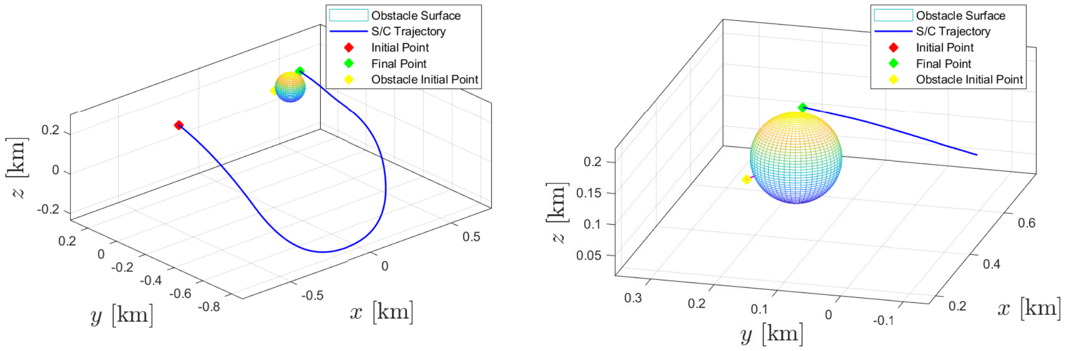


Figure 6.10: Obstacle Constraint

In the previous chapter, the collision between the two bodies was illustrated to help the reader understand the situation. However, in the current scenario, the satellite is able to continue its trajectory while effectively avoiding the imposed constraints.



(a) Minimum distance

(b) Zoom

Figure 6.11: Obstacle Avoidance

6.2.2 Components

Now, the coordinates over the time interval are presented. The position coordinates depict the deviation of the trajectory from the unconstrained path due to the presence of the obstacle. The most notable difference is observed along the x-coordinate.

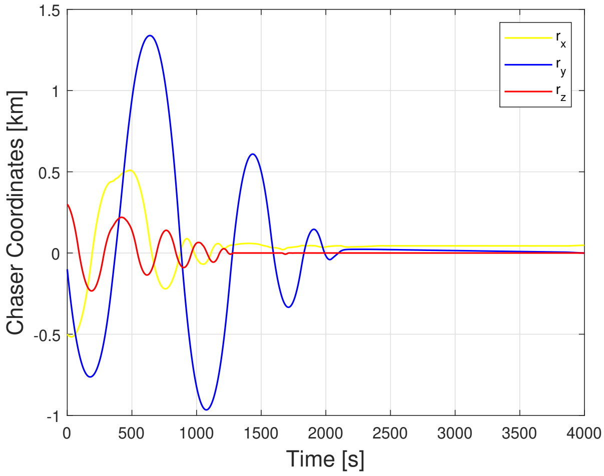


Figure 6.12: Position Coordinates

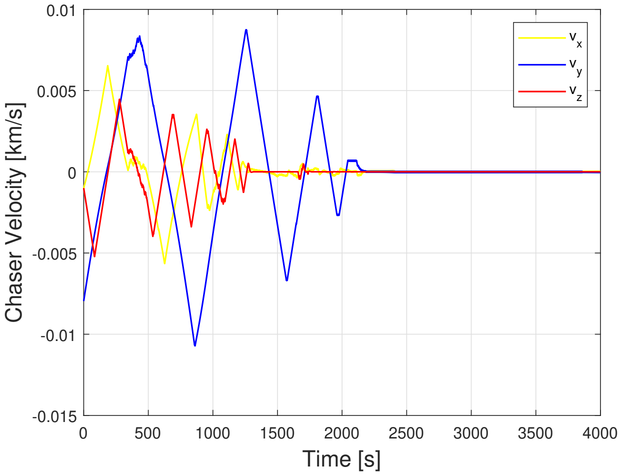


Figure 6.13: Velocity Coordinates

The plots of the command inputs indicate that the task is successfully accomplished in approximately 3860 *sec*, which is equivalent more or less to 64 *min*. This duration is approximately 1000 *sec* longer than the unconstrained case. The additional time is necessary for the spacecraft to perform the required maneuvering and adjustments to safely navigate around the obstacle while reaching the desired position.

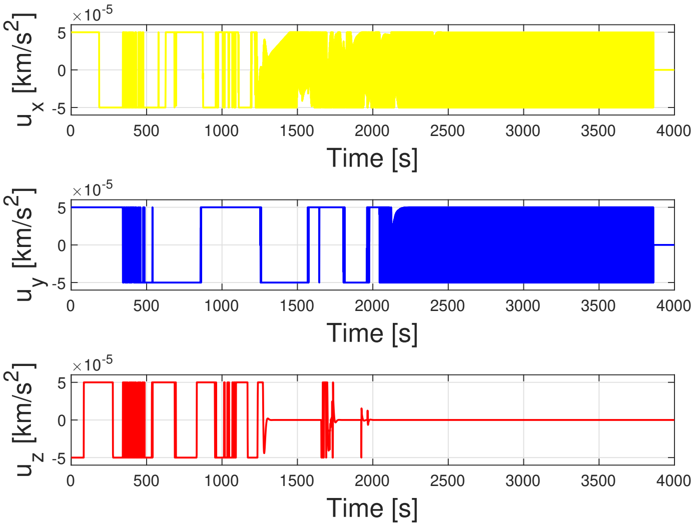
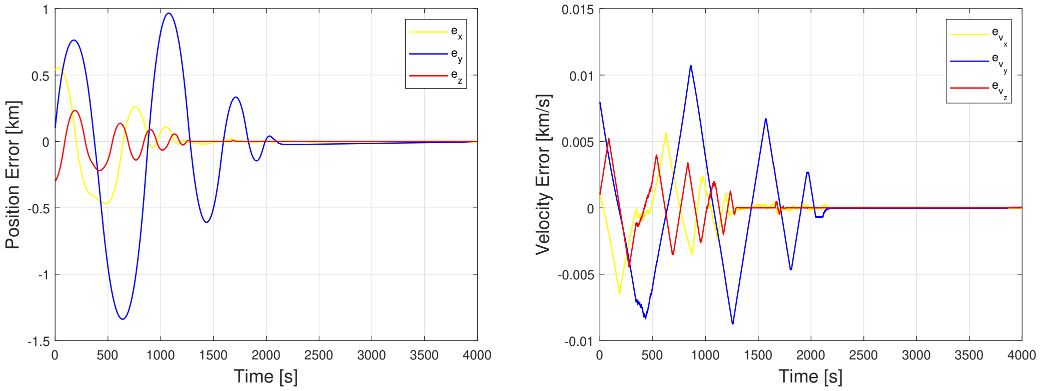


Figure 6.14: Command Input

6.2.3 Errors

The errors of position and velocity are displayed in the following plots. These plots illustrate the discrepancies between the desired values and the actual values of position and velocity.



(a) Position Error

(b) Velocity Error

Figure 6.15: Components errors

6.3 Variable Prediction Horizon - Unconstrained

After displaying the complete task plots, the results of the NMPC with a variable horizon are listed.

The trajectory shown in the picture is similar to the trajectory obtained with the NMPC, but the time required to reach the target differs. Specifically, the S/C using the NMPC with a variable horizon achieves the goal in 2500 $[sec]$ (approximately 42 $[min]$), which is about 200 sec (around 3 $[min]$) faster than in the initial simulation.

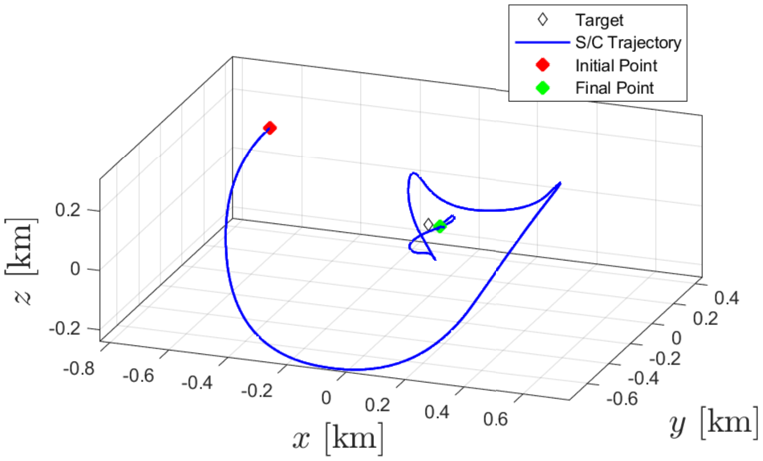


Figure 6.16: Unconstrained Trajectory with variable T_p

6.3.1 Components

Below are the plots of position, velocity, and command inputs for this simulation.

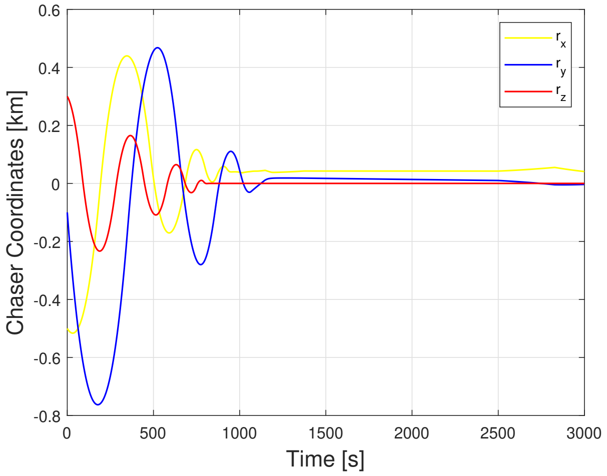


Figure 6.17: Position Coordinates

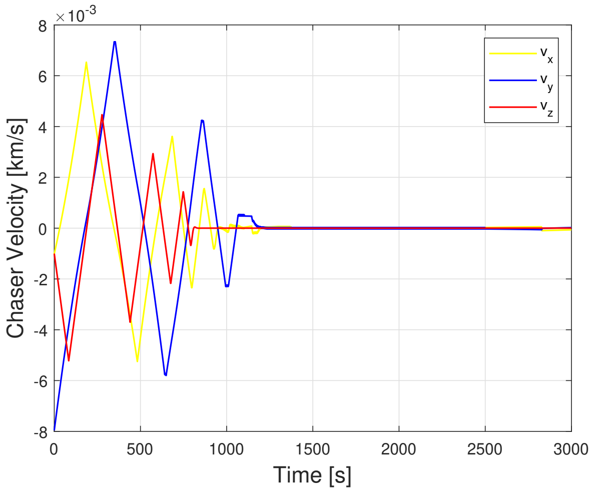


Figure 6.18: Velocity Coordinates

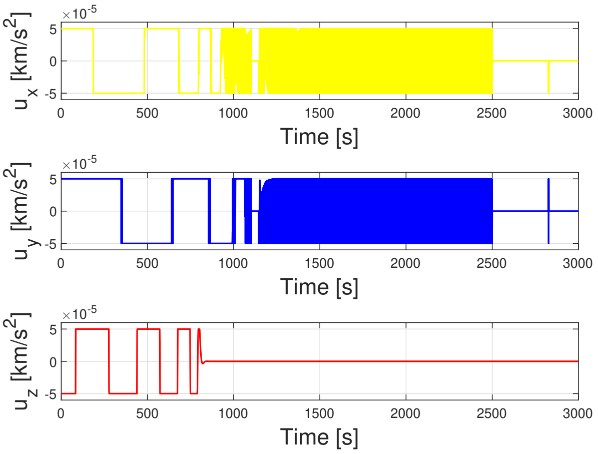
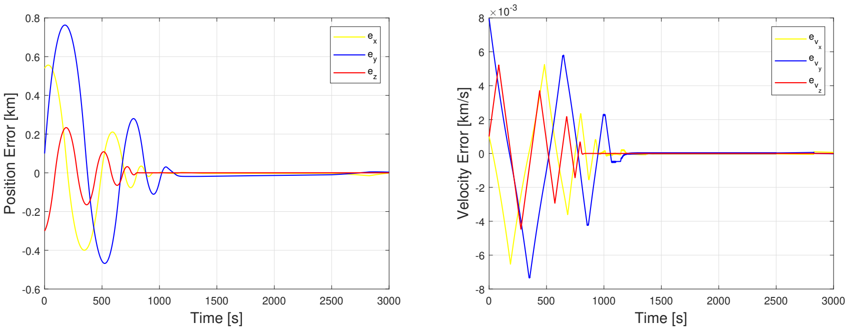


Figure 6.19: Command input

Thanks to the command inputs, it is indeed possible to calculate the energy of the signal over the time interval, referred to as the total impulse. By integrating the command inputs over time, the total impulse can be determined, providing insights into the overall energy expenditure of the system.
The total impulse calculated in the variable prediction horizon case is 1.4155e-05 [km/s], which is lower than the total impulse calculated in the normal case, which is 1.4583e-05 [km/s]. The value of the total impulse indicates that the NMPC with variable prediction horizon converges earlier and has a lower energy consumption compared to other methods. This observation highlights the efficiency

and effectiveness of the NMPC with variable prediction horizon in achieving the desired trajectory with minimal energy expenditure.

6.3.2 Errors



(a) Position Error **(b)** Velocity Error

Figure 6.20: Components errors

6.4 Variable Prediction Horizon - Constrained

In this section, the final results are illustrated. The trajectory depicted demonstrates the alignment of the S/C with the target and the successful avoidance of the obstacle. In this case, the satellite is controlled by the NMPC with a variable prediction horizon.

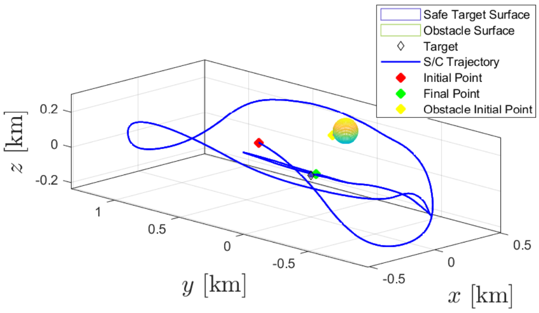


Figure 6.21: Constrained Trajectory

6.4.1 Constraints

The safe zone around the target and the constraints imposed by the obstacle are respected, similar to the previous simulations. The trajectory ensures that the spacecraft maintains a safe distance from the target and avoids any collision with the obstacle.

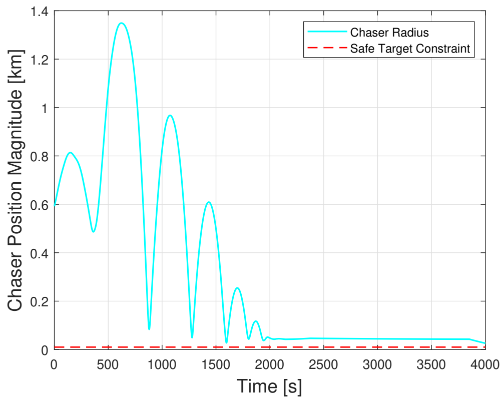


Figure 6.22: Safe Target

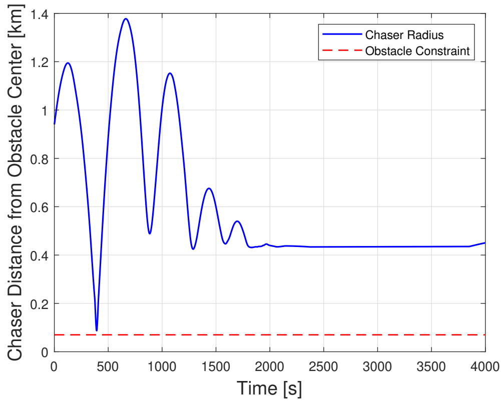


Figure 6.23: Obstacle Constraint

6.4.2 Components

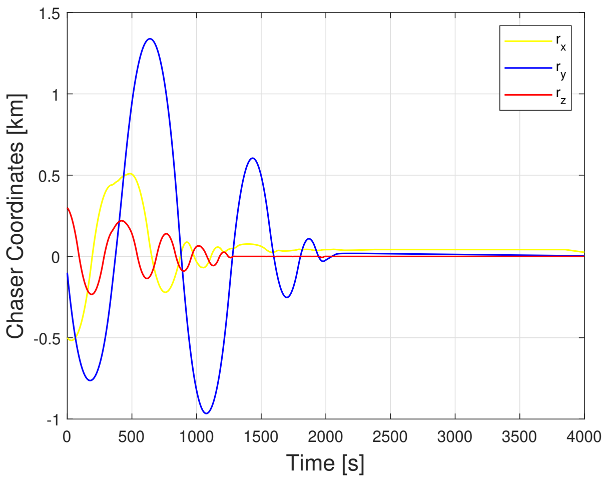


Figure 6.24: Position Coordinates

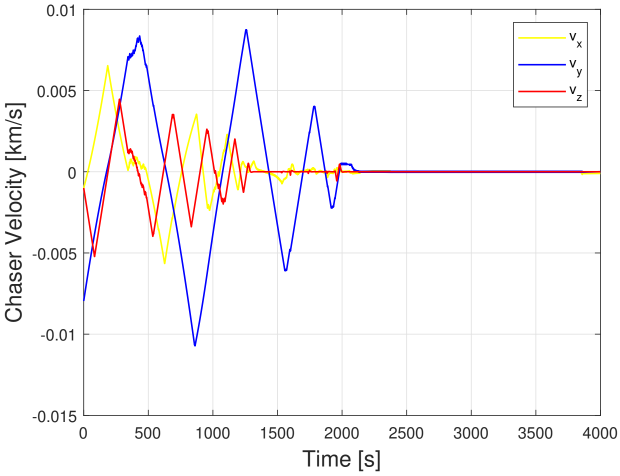


Figure 6.25: Velocity Coordinates

The plots of the command inputs reveal that the engines reduce effort around 3850 *sec*, which is 10 *sec* earlier than the case with the normal NMPC. This means that in this case, the effort is lower than the csae with NMPC. The total impulse of this simulation is 2.2054e-05 [km/s], which is less than the total impulse of 2.2087e-05 [km/s] consumed in the simulation with the NMPC.
This is a significant and noteworthy result from these simulations. The ability to

shut off the engines earlier indicates an improvement in control efficiency and the achievement of the desired trajectory in a shorter time frame.

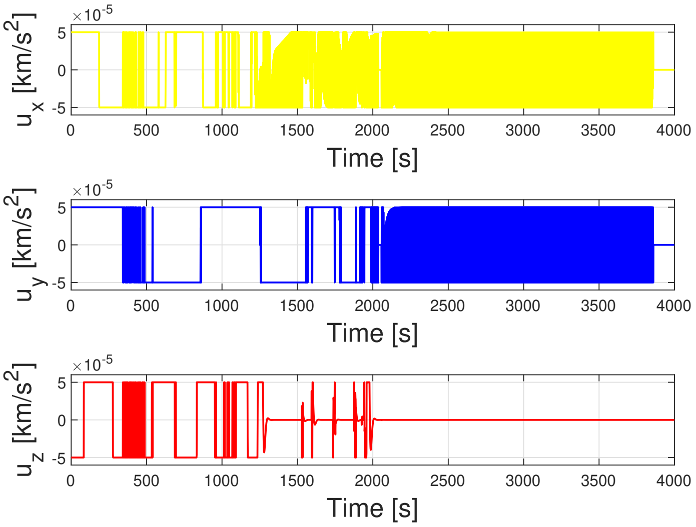
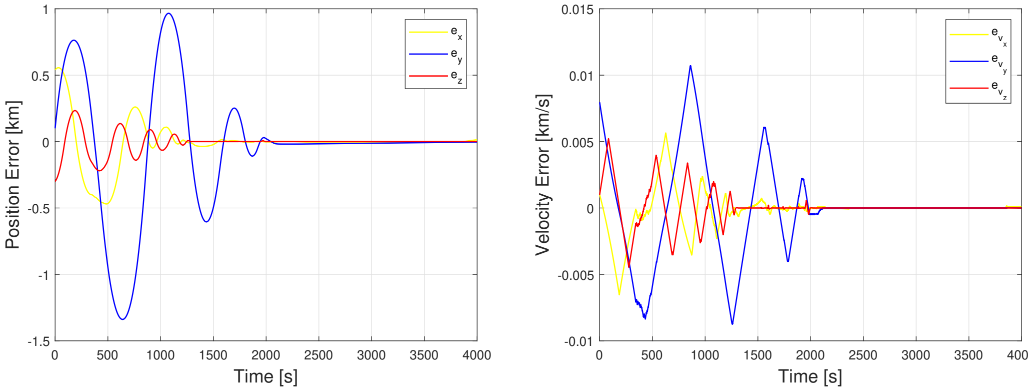


Figure 6.26: Command Input

6.4.3 Errors



(a) Position Error

(b) Velocity Error

Figure 6.27: Components errors

Chapter 7

Conclusions

In this thesis, an NMPC control algorithm based on Pontryagin's Minimum Principle has been implemented to accomplish the desired rendezvous maneuver. The model used for the S/Cs was described by the two-body equations. However, the predictive model used in NMPC was based on the HCW equations. With the algorithm's implementation, an S/C was capable of reaching a reference position near the target while avoiding obstacles along its trajectory. The obstacles were implemented as state constraints to be avoided. The control algorithm provided the S/C with information about the position and motion of obstacles, enabling it to take the necessary precautions to complete the task.

Following this performance, another control algorithm, AHMPC, was employed to perform the same task. AHMPC was introduced and successfully completed the maneuver. The results presented in the final chapter demonstrate that AHMPC reduced the convergence time and, consequently, the total impulse consumed throughout the maneuver. The differences were more evident in the unconstrained case, where the reduction of convergence time was approximately 200 seconds, while in the constrained case, it was around 10 seconds. Ultimately, these results highlight the characteristic of AHMPC to achieve the task with the smallest possible prediction horizon, enabling a faster control algorithm.

7.1 Further Works

These results have shown how NMPC is able to control S/Cs in their space missions. After this, some further work could be useful to improve the complexity and accuracy of future tasks. The main focus of further work could be on the uncertainty of obstacle coordinates. In this thesis, the position and velocity of the obstacles are known. Introducing a random error in the algorithm's knowledge of them could improve control efficiency, enabling the S/C to avoid a wider range of

obstacles. Similarly, changing the motion of the obstacles and giving them different trajectories could also be explored.
In this thesis, AHMPC has been implemented in a descending mode. An improvement could involve expanding the prediction horizon, allowing it to have more freedom to adapt to the task and complete it. This increased freedom can be implemented right from the beginning of the task.

As explained, NMPC is able to accomplish the task and handling space mission. However, the study on the efficiency and accuracy of maneuvers is not yet complete.

Bibliography

[1] J. L. Goodman. *History of space shuttle rendezvous.* Tech. rep. 2011 (cit. on p. 1).

[2] *Low Earth Orbit.* 2020. URL: `https://www.esa.int/ESA_Multimedia/Images/2020/03/Low_Earth_orbit` (cit. on p. 3).

[3] Carlo Novara. *Nonlinear Control and Aerospace Applications: lecture notes.* Politecnico di Torino, 2017 (cit. on pp. 6, 21, 26).

[4] E. Canuto, C. Novara, L. Massotti, D. Carlucci, and C. Perez Montenegro. *Spacecraft dynamics and control. The embedded model control approach.* Butterworth-Heinemann, 2018 (cit. on p. 6).

[5] Roger R. Bate, Donald D. Muller, Jerry E. White, and William W. Saylor. *Fundamental of Astrodynamics.* Mineola, New York: Dover Publications, 2020 (cit. on pp. 6, 21).

[6] *Kepler's laws of planetary motion.* URL: `https://en.wikipedia.org/wiki/Kepler's_laws_of_planetary_motion` (cit. on p. 6).

[7] *Orbital Mechanics.* URL: `https://en.wikipedia.org/wiki/Orbital_mechanics#Laws_of_astrodynamics` (cit. on p. 6).

[8] *Newton's law of universal gravitation.* URL: `https://en.wikipedia.org/wiki/Newton's_law_of_universal_gravitation` (cit. on p. 7).

[9] M. S. Lallaro. «Guidance and Control in Space Debris Removal Missions via Adaptive Nonlinear Model Predictive Control». MA thesis. Torino: Politecnico di Torino, 2021. URL: `http://webthesis.biblio.polito.it/id/eprint/17893` (cit. on p. 14).

[10] F. Allgöwer, R. Findeisen, and Z. K. Nagy. «Nonlinear model predictive control: From theory to application». In: *Journal-Chinese Institute Of Chemical Engineers* 35.3 (2004), pp. 299–316 (cit. on p. 26).

[11] J. Rawlings and D. Mayne. *Model predictive control: Theory and design.* Nob Hill Pub, 2009 (cit. on p. 26).

[12] A. Bemporad, F. Borrelli, and M. Morari. «Model predictive control based on linear programming - the explicit solution». In: *IEEE transactions on automatic control* 47.12 (2002), pp. 1974–1985 (cit. on p. 28).

[13] A.J. Krener. «Adaptive horizon model predictive control». In: *IFAC - PapersOnLine* 51.13 (2018), pp. 31–36 (cit. on p. 36).

[14] Z. Wang and Y. Li. «An indirect method for inequality constrained optimal control problems». In: *IFAC-PapersOnLine* 50.1 (2017), pp. 4070–4075 (cit. on p. 37).

[15] M. Pagone, M. Boggio, C. Novara, and S. Vidano. «A Pontryagin-based NMPC approach for autonomous rendez-vous proximity operations». In: *2021 IEEE Aerospace Conference*. 2021 (cit. on p. 38).

[16] M. Pagone, M. Boggio, C. Novara, A. Proskurnikov, and G. C. Calafiore. «A penalty function approach to constrained Pontryagin-based Nonlinear Model Predictive Control». In: *2022 IEEE 61st Conference on Decision and Control (CDC)*. 2022 (cit. on p. 38).

[17] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko. *The Mathematical Theory of Optimal Processes*. New York: Interscience Publishers, John Wiley and Sons, 1962 (cit. on p. 39).

[18] A. E. Bryson and Y. Ho. *Applied optimal control: optimization, estimation and control*. Taylor & Francis Inc, 1975 (cit. on p. 39).

[19] L. T. Luh. «The shape parameter in the Gaussian function». In: *Computers & Mathematics with Applications* 63.3 (2012), pp. 687–694 (cit. on p. 43).

[20] Luo Yazhong, Zhang Jin, and Tang Guojin. «Survey of orbital dynamics and control of space rendezvous». In: *Chinese Journal of Aeronautics* (2014), 27(1): 1–11 (cit. on pp. 44, 45).

[21] *ODE23*. URL: `https://www.mathworks.com/help/matlab/ref/ode23.html` (cit. on p. 55).