POLITECNICO DI TORINO

Corso di Laurea Magistrale
in Mechatronic Engineering

Tesi di Laurea Magistrale

Guidance and Control in Space Debris Removal Missions via Adaptive Nonlinear Model Predictive Control

Relatore:

Prof. Carlo Novara

Candidato:

Marco Scaffidi Lallaro

Anno Accademico 2020/2021

Abstract

Space debris orbiting around the Earth are becoming a major problem that could impair the future of space exploration.

Among the different approaches to this problem that have been proposed in the recent years, this work focuses on a possible solution consisting in an autonomous spacecraft that performs a rendezvous maneuver, collects a debris of unknown mass and then moves to a parking orbit.

When the spacecraft collects a debris of unknown mass, the dynamics of the system may change substantially and this may affect the control stability and performance of the spacecraft. In this study, a control system is designed, capable of handling situations with time-varying and uncertain parameters, as it occurs in space debris removal missions.

A control strategy based on an Adaptive Nonlinear Model Predictive Control (ANMPC) is developed. The unknown mass of the debris is treated as an uncertain parameter and is estimated by means of two different methods (Recursive Average and Extended Kalman Filter (EKF)). Then, the estimated mass is used to update the internal model of the ANMPC, which later solves an on-line optimization problem providing an optimal trajectory and control action for reaching the debris and the parking orbit.

Simulations shows that the proposed control system is able to effectively accomplish the requested task. The solutions coming from two different estimation methods are compared and provide similar results. If a simple NMPC strategy is employed without an estimation and adaptation process, the obtained results do not diverge significantly from the other case, meaning that NMPC is a control method that is intrinsically robust for this kind of applications. However, the mass estimation is useful in order to improve the overall performance.

The adaptive algorithm has also the potential to be extended to other space missions characterized by unknown parameters.

To my Mum and Dad,
who always picked me up on time
and encouraged me to go on every adventure,
especially this one

Contents

1 Introduction 1
1.1 Thesis Outline 5

2 Theoretical Background 6
2.1 Orbital Dynamics 6
2.1.1 Kepler's Laws 7
2.1.2 Newton's Laws 7
2.1.3 The Two-Bodies (2B) Problem 8
2.1.4 Orbit Equation 11
2.1.5 Orbit Geometry 12
2.1.6 Reference Frames 17
2.1.7 Orbital Elements 19
2.1.8 Hill-Clohessy-Wiltshire (HCW) equations 22
2.2 Controller Comparison 25
2.2.1 General Requirements 26
2.2.2 Nonlinear Control Design Methods 26
2.2.3 Controller Choice 27
2.2.4 Adaptive NMPC 28
2.3 Estimation Process 28

3 Model Predictive Control 29
3.1 Introduction 29
3.2 Theoretical formulation 30
3.2.1 Prediction 30
3.2.2 Optimization 32
3.2.3 Receding Horizon Principle 34
3.3 Mathematical Formulation 35
3.3.1 Features of the Optimization Problem 36
3.3.2 Features of the Objective Function 36
3.3.3 Input Parameterization 36
3.4 MPC Algorithm and Design 37
3.4.1 MPC Algorithm 37
3.4.2 MPC Design 38
3.5 MPC Properties 40
3.6 Advantages and Drawbacks 41

3.7 Comparison between NMPC and linear MPC 41

4 Space Mission and Control Algorithm 43

4.1 Space Mission 43

4.1.1 Mission Maneuvers 44

4.2 Control Algorithm 47

4.2.1 System Model 47

4.2.2 Prediction Models 50

4.2.3 General Settings 51

4.2.4 Maneuvers Settings 53

5 Simulation Results 71

5.1 Simulation Results 71

5.2 Rendezvous Maneuver 71

5.2.1 Time and Delta-V 71

5.2.2 Command Input, Position, and Velocity 72

5.2.3 Trajectory 73

5.2.4 Fuel Consumption 75

5.3 Docking Maneuver 76

5.3.1 Time and Delta-V 76

5.3.2 Command Input, Position, and Velocity 76

5.3.3 Trajectory 77

5.3.4 Fuel Consumption 79

5.4 Estimation Maneuver 80

5.4.1 Time and Delta-V 80

5.4.2 Command Input, Position, and Velocity 81

5.4.3 Trajectory 82

5.4.4 Fuel Consumption 84

5.4.5 Estimation Process 85

5.5 Second Rendezvous Maneuver 87

5.5.1 Time and Delta-V 87

5.5.2 Command Input, Position, and Velocity 87

5.5.3 Trajectory 88

5.5.4 Fuel Consumption 90

5.6 Orbit-Change Maneuver 91

5.6.1 Time and Delta-V 91

5.6.2 Command Input, Position, and Velocity 92

5.6.3 Trajectory 93

5.6.4 Fuel Consumption 95

5.7 Comparison Without Estimating m_2 96

6 Conclusion 99

6.1 Further Works 100

Acronyms

MPC Model Predictive Control

NMPC Nonlinear Model Predictive Control

MIMO Multiple Input - Multiple-Output

FBL Feedback Linearization

SMC Sliding Mode Control

GS Gain Scheduling

w.r.t. with respect to

RH Receding Horizon

LTI Linear Time Invariant

S/C Spacecraft

GNC Guidance, Navigation and Control

HCW Hill-Clohessy-Wiltshire

LEO Low-Earth Orbit

CoM Center of Mass

2B Two-Bodies

FR2B Free Two-Bodies

GE Geocentric Equatorial

LVLH Local-Vertical Local-Horizontal

LORF Local-Orbital Reference Frame

PF Perifocal (Perigee) Frame

DCM Direction Cosine Matrix

RLS Recursive Least Squares

EKF Extended Kalman Filter

ORE Orbit Equation

ESA European Space Agency

ADR Active Debris Removal

AOCS Attitude and Orbit Control Systems

LQR Linear Quadratic Regulator

Chapter 1

Introduction

Space debris orbiting around the Earth is becoming a major problem that could impair operational space missions and the future of space exploration. More than 15000 objects have been cataloged in orbit around the Earth at which only the 6% are active satellites [1]. The pollution of debris would continue to increase even if space launches are halted today. If a collision happens between a debris and an active satellite or a large debris (e.g. derelict satellites) a cloud of little objects could spread in Space. This was the case when the Iridium satellites collided with a retired Russian satellite. That accident created over 1500 pieces of orbital debris. In recent years, the mitigation problem was take care for the Space community and some steps were made towards a future with a clean Space. The debris problem was formulated the first time in the 1960 by the scientist Willy Ley. Then, in the 1978 the scientist Donald J. Kessler formulate a possible dystopian scenario that takes the name of *Kessler syndrome*. In this dystopian future the Space pollution could be great enough to make no more possible Space exploration.

In addressing this problem the UN stated some guidelines for Space Debris Mitigation, but mitigation alone is not sufficient to solve the problem. For this purpose several Active Debris Removal (ADR) methods were proposed in the last decade. Space agencies (e.g. NASA, ESA) already started with the study to their own programs for these kind of missions and in literature can be found different methods to approach to the debris problem [1] [2]. By considering how a debris can be removed a classification can be made:

- *Collective Methods*
- *Laser-based Methods*
- *Ion-beam Shepherd-based Methods*
- *Tether-based Methods*
- *Sail-based Methods*
- *Dynamical Systems-based Methods*

An important step was made in 2019. Indeed, in that year it was testing the first ADR for LEO debris [3].

Differently from the approaches to this problem that have been proposed in recent years, this thesis focuses on a possible solution exploiting an automated method in the most critical region for debris concentration. Similar studies that dealing with automated method are already presented to the space community. An important consideration has to be made on the basis of the different control algorithms considered in these works.

In [4] a feedback linearization method is used to accomplish an autonomous non-cooperative rendezvous maneuver. The equations used to describe the relative motion are a modified version of the HCW equations. The limitations of the feedback linearization method do not make possible to consider constraints and the dynamics equations must be written in affine form. As will be seen later, constraints must be taken into account and the used equations are not in affine form.

In [5] two CubeSats are provided with a Linear Quadratic Regulator (LQR). The command input is calculated from an optimization problem and applied in open-loop. Since the optimal sequence is applied in open-loop is not possible to dealing with varying constraints and/or to counteract unexpected behaviours.

In [6] is analyzed the GNC problem for a ADR mission that exploits a robotic arm as capture system. In this article two different controllers are propose for Guidance and Control modules. A LQR is used for Guidance purposes, while both a LQR and a PID controller are used for Control purposes. An EKF and another version of the Kalman filter (the unscented KF) are used in the Navigation module. Also in this case the possible problem coming from the LQR are not taken into account.

The aim of this thesis is to develop a control algorithm that is able to accomplish the Guidance and Control tasks for a wide range of ADR missions. As will be seen later, in order to consider varying constraints and to dealing with system equations that are not in affine form, an adaptive version of the MPC will be developed. With this control method is also possible to obtain optimal solutions and to solve in the same time Guidance and Control.

The critical region for debris concentration is found at orbital altitude between 600 to 800 [km]. For this reason, the analysis is concentrated only on the control system that has to be installed on a hypothetical Spacecraft (S/C) that has to work in a LEO (that includes the critical region seen before). In this way, the control algorithm could be extended to a wide range of ADR missions that could use different collective methods: robotic arm, tentacle system, harpoon, and nets.

However, recent analyzes found in literature focus their studies on two different regions: LEO and GEO. But, it is worth to note that in the GEO the debris density is much lower than in the LEO and possible collisions could cause lower damages for the lower impact velocity [1].

One important challenge, for the automation of a ADR system, is that when the space debris is collected by the S/C, its unknown mass will affect the dynamics of the whole system. In fact, this could be the case where the information on the S/C dynamics alone is not sufficient to successfully accomplish the different tasks of a space mission.

The control system that has to be developed must be capable of handling situations where not all the parameters are known a priori or change in time. These parameters are needed in order to accomplish the Guidance, Navigation and Control (GNC) problem for the ADR mission.

For this purpose, a control strategy based on an adaptive version of the Nonlinear Model Predictive Control (NMPC) is developed. This control approach is implemented ad-hoc in the debris removal space mission, where the unknown debris mass, treated as a parameter in the control problem, is estimated with two different methods (*Recursive Average* and *Extended Kalman Filter (EKF)*) in order to obtain an optimal trajectory needed for the debris successfully removal.

The models used by the controller and by the EKF are different from the one used as plant. Inside this last one are considered also effects as the drag force and other little perturbations. Another difference is that only in the plant the propellant consumption is accounted for. Thanks to its feasible and robustness properties the NMPC is able to optimally menage the the constraints coming from the different maneuvers and also to counteract the differences between the used models.

The mission is divided in five different maneuvers. In each one the controller sets its parameter in a gain scheduling fashion. The scheduled parameters, as the references and the sampling time, are selected thanks to some conditions. In this way, the system understands when a specific maneuver has to start. The mission start with two precision maneuvers: the Rendezvous and the Docking. After the Docking the system becomes the fusion between the S/C and the debris. The prediction model used by the controller in these maneuvers is based on the HCW equations. Then, the Estimation maneuver stars and debris mass is estimated online. After this estimation the system is driven on an empty orbit. This is taken as starting point for the next maneuver: Orbit-Change. In the estimation maneuver and in the Orbit-Change the prediction model is based on the 2B equations. It is important to note that this last maneuver is considered to bring debris with large size in an empty orbit by increasing the apogee distance. Another possible solution in order to remove debris is to burn them in the atmosphere by decreasing their orbit.

The adaptivity of the NMPC is in the estimation maneuver. The estimation algorithm finds and feeds the controller with the estimation of the debris mass. In this way, at each sampling time this value is updated and converges to the real mass value.

The debris mass is considered equal to 550 $[kg]$. In this way it is possible to extend the control approach also to systems that has to work with a large debris size (like old S/C stages).

Just to have an idea of the danger that a debris could represent, in the following picture it is showed a test done by European Space Agency (ESA) with a small debris (1.2 [cm]):

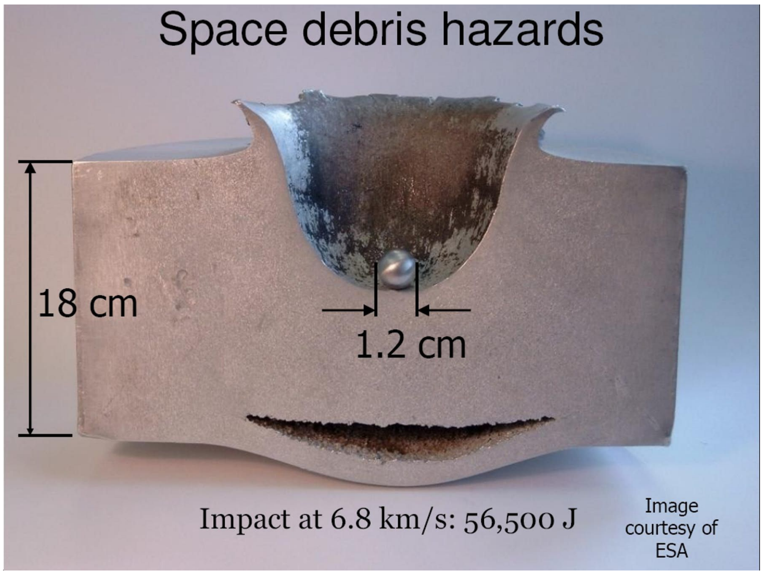

Figure 1.1: ESA Impact Test

The *Kessler syndrome* accounts for a possible scenario in which a lot of objects orbit in Low-Earth Orbit (LEO). In this situation the space pollution is high enough that collisions between objects could cause a cascade in which each collision generates space debris that increases further collisions. One important implication is that the distribution of debris in orbit could render space activities and the use of satellites in specific orbital ranges difficult for many generations.

Considering that, this problem will become more important, with the growing number of satellites, that will orbit around the Earth, the ADR missions will be necessary in order to reach the clean Space condition.

1.1 Thesis Outline

Chapter 2 investigates the theoretical background for the GNC problem. It contains discussions on absolute and relative motion description, coordinate frames, and finally controller comparison and estimation processes. Chapter 3 focuses on the powerful and flexible Nonlinear Model Predictive Control (NMPC). Chapter 4 indicates the various maneuvers needed for the mission. Then, it will be analyzed the whole control algorithm. Two different techniques are compared for the estimation process. Then, in chapter 5 the simulation results are shown and discussed. The work is closed by the conclusion chapter, which provides suggestions for further works.

Chapter 2

Theoretical Background

This chapter aims to introduce the essential basics needed by the reader to understand this thesis work.

The first part focuses on arguments about orbital dynamics, on coordinate frames, and on absolute and relative equations of motion. The second one, concerns a control comparison. Then, the last part focuses on the the formulation of the estimation process. The presented notions are taken from [10] and [16].

2.1 Orbital Dynamics

A S/C can be described by a rigid body which moves with respect to (w.r.t.) some inertial frame. Clearly, this is an approximate description (rigid body), but in many cases it is fine. It can be used to develop many analysis and also to design control systems.

However, the body motion is a combination of a *translation* of the body Center of Mass (CoM) and a *rotation* of the body about an axis passing trough its CoM.

By split the two motions, it is possible to analyze them separately. The aim of this thesis is to develop a control algorithm able to accomplish various mission tasks in the orbital dynamics field. For this reason, the analysis focuses only on the translational motion of the S/C in some gravitational field.

In this section, the basic concepts of celestial mechanics will be analyzed. In particular, the *Kepler's laws* and the *Newton's laws* are take into account. Both these two analysis are very important from the point of view of modern applications.

The Newton's laws come from general physics and they imply the Kepler's laws (which are empirical).

2.1.1 Kepler's Laws

Kepler formulated their three laws on the observation of planets orbits around the Sun. The three laws modified the Copernicus heliocentric theory, by replacing circular with elliptical orbits, and explaining how planetary velocities vary:

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

2. *The radius vector, that links the Sun and a planet, sweeps out equal areas during equal intervals of time. In other words, the area velocity is constant;*

3. *Planetary period of revolution are proportional to $r_m^{3/2}$. Where r_m is the mean distance from the Sun;*

2.1.2 Newton's Laws

The Newton's laws are more general. They can be divided into three laws of motion and one law of gravitation. The first three describe the relationship between the motion of an object and the forces acting on it. They are:

1. *An object either remains at rest or continues to move at a constant velocity, unless it is acted upon by an external force.*

2. *The rate of change of the linear momentum $m\boldsymbol{v}$ of an object is given by:*

   $$\frac{d}{dt}(m\boldsymbol{v}) = \boldsymbol{F}$$

   Where m is the object mass, $\boldsymbol{v}$ is the object velocity, F is the force acting on the object.

   For an object with constant mass, the net force on an object is equal to the mass of that object multiplied by the acceleration $\boldsymbol{a}$.

   $$\boldsymbol{F} = m\frac{d\boldsymbol{v}}{dt} = m\boldsymbol{a}$$

3. *When one object exerts a force on a second object $\boldsymbol{F}_{12}$, that second object exerts a force $\boldsymbol{F}_{21}$ that is equal in magnitude and opposite in direction on the first object.*

   $$\boldsymbol{F}_{21} = -\boldsymbol{F}_{12}$$

While, the Newton's law of gravitation states:

> *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.

$$\boldsymbol{F} = \frac{Gm_1m_2\boldsymbol{r}}{r^3}$$

In this last equation $G = 6.67e - 11$ $[Nm^2/kg^2]$ is the *Gravity universal constant*, m_1 and m_2 are the object masses. In order to indicate the direction of $\mathbf{F}$ is used the previous notation with $\mathbf{r}/r^3$ (r^3 is the cubic of the distance magnitude).

2.1.3 The Two-Bodies (2B) Problem

In order to understand the orbital dynamics it is essential to start the analysis from the forces that are acting on the space objects. For this reason, the formulation of the Two-Bodies (2B) problem plays a crucial role.

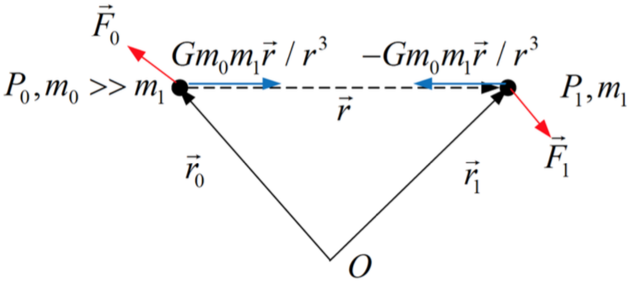

Figure 2.1: 2B Problem General Setting

The simplest analysis that can be make is done by considering only two point mass objects. As it can be see, there are two bodies located in two different points P_0 and P_1 w.r.t. a given reference frame with origin O. The position vectors from the reference frame are: $\mathbf{r}_0$ and $\mathbf{r}_1$. While, the relative position is indicated with $\mathbf{r}$. These two bodies have masses m_0 and m_1 respectively, with $m_0 \gg m_1$.

By means of gravity force, the two bodies are attracted each other. The big body m_0 is attracted with $\mathbf{F}_{01} = Gm_0m_1\mathbf{r}/r^3$ towards the small body m_1. For the third Newton's law of motion, the small body reacts with the same force, but in opposite direction $\mathbf{F}_{10} = -\mathbf{F}_{01}$.

In this situation, two external forces F_0 and F_1 acting on the two masses respectively are also considered. It is worth to note that these forces are not gravitational. They can be seen as the two resultants of the all external forces applied to the two bodies.

It is possible to write the second Newton's law on the two bodies. It is assumed that the two masses are constants :

$$\begin{aligned}\dot{\mathbf{v}}_0 &= \frac{1}{m_0}\sum \mathbf{F} = Gm_1\frac{\mathbf{r}}{r^3} + \frac{1}{m_0}\mathbf{F}_0 \\ \dot{\mathbf{v}}_1 &= \frac{1}{m_1}\sum \mathbf{F} = -Gm_0\frac{\mathbf{r}}{r^3} + \frac{1}{m_1}\mathbf{F}_1\end{aligned} \tag{2.1}$$

These two equation describe the absolute motion of the two bodies w.r.t. the inertial reference frame.
Now, considering:

- *Relative Position:*
$$\mathbf{r} = \mathbf{r}_1 - \mathbf{r}_0$$
- *Relative Velocity:*
$$\mathbf{v} = \mathbf{v}_1 - \mathbf{v}_0$$
- *Relative Acceleration:*
$$\dot{\mathbf{v}} = \dot{\mathbf{v}}_1 - \dot{\mathbf{v}}_0$$
- *CoM Position:*
$$\mathbf{r}_c = \frac{m_0}{m_0 + m_1}\mathbf{r}_0 + \frac{m_1}{m_0 + m_1}\mathbf{r}_1$$
- *CoM Velocity:*
$$\mathbf{v}_c = \frac{m_0}{m_0 + m_1}\mathbf{v}_0 + \frac{m_1}{m_0 + m_1}\mathbf{v}_1$$

It is possible to write:

- *Relative Motion:*
$$\dot{\mathbf{v}} = -G(m_0 + m_1)\frac{\mathbf{r}}{r^3} + \frac{1}{m_1}(\mathbf{F}_1 - \frac{m_1}{m_0}\mathbf{F}_0)$$
- *CoM Motion:*
$$\dot{\mathbf{v}}_c = \frac{1}{m_1}\frac{\mathbf{F}_1 + \mathbf{F}_0}{1 + (m_0/m_1)}$$

The first equation describe the relative motion of the two bodies. The second one describe the motion of the system CoM (the system is composed by the union of the two bodies).

By considering the previous assumption $m_0 \gg m_1$, it is obtained the *Restricted 2B equation*:

$$\dot{\mathbf{v}} + \mu\frac{\mathbf{r}}{r^3} = +\frac{1}{m_1}\mathbf{F}_1 \tag{2.2}$$

Where $\mu = Gm_0$ is the *Gravitational Parameter.* It is always associated with the big body.

It is interesting to note that the acceleration of the system CoM becomes zero $\dot{\mathbf{v}}_c = 0$. This means that the CoM coincide with the center of the big body. Moreover, this point can be chosen as the origin of an inertial reference frame (it has null acceleration).

In the following it is showed a table with some gravitational parameters and accelerations:

Celestial Body	m $[kg]$	$r_{equatorial}$ $[m]$	μ $[m^3/s^2]$	g $[m/s^2]$
Sun	$1.99e30$	$0.696e9$	$0.133e21$	274
Earth	$5.97e24$	$6.38e6$	$0.3986e15$	9.78
Mars	$0.642e24$	$3.40e6$	$0.0428e15$	3.70
Moon	$0.0755e24$	$1.74e6$	$4.90e12$	1.62

Table 2.1: Some Celestial Parameters

If the external force $\mathbf{F}_1$ is equal to zero it is possible to write the restricted Free Two-Bodies (FR2B):

$$\mathbf{F}_1 = 0, \qquad \dot{\mathbf{v}} + \mu \frac{\mathbf{r}}{r^3} = 0 \tag{2.3}$$

Using this equation it is possible to prove that:

- The total mechanical energy ε of the FR2B system is conserved;
- The angular momentum h of the FR2B system is conserved;
- The free response of the FR2B equation occurs on a plane called *Orbital Plane*;

It is also possible to derive a geometric description of the FR2B system orbits.

Energy Conservation

Taking the equation 2.3 and performing the dot product for $\mathbf{v}$ it is obtained:

$$\begin{aligned} &\dot{\mathbf{v}} \cdot \mathbf{v} + \mu \frac{\mathbf{r}}{r^3} \cdot \mathbf{v} = \frac{1}{2}\frac{d}{dt}(\mathbf{v}\cdot\mathbf{v}) + \mu \frac{1}{2r^3}\frac{d}{dt}(\mathbf{r}\cdot\mathbf{r}) = \\ &= \frac{d}{dt}\frac{v^2}{2} + \mu \frac{1}{2r^3}\frac{d}{dt}r^2 = \frac{d}{dt}\frac{v^2}{2} + \mu \frac{\dot{r}}{r^2} = \\ &= \frac{d}{dt}(\frac{v^2}{2} - \frac{\mu}{r}) = 0 \end{aligned} \tag{2.4}$$

The quantity $(\frac{v^2}{2} - \frac{\mu}{r}) = constant$ (since its derivative is equal to zero) is the *total mechanical energy* ε per unit mass of the system. In this way, it is proven the principle of energy conservation:

$$\varepsilon = \frac{v^2}{2} - \frac{\mu}{r} = constant$$

Where $\frac{v^2}{2}$ is the kinetic energy per unit mass and $\frac{\mu}{r}$ is the potential energy per unit mass.

Solving the previous equation for v it is possible to see that for a given total energy, the corresponding magnitude of the orbital velocity vector is:

$$v = \sqrt{2(\frac{\mu}{r} + \varepsilon)}$$

Conservation of the Angular Momentum and Planar Motion

Now, taking the equation 2.3 it is performed the cross product for $\mathbf{r}$:

$$\begin{aligned} &\mathbf{r} \times \dot{\mathbf{v}} + \frac{\mu}{r^3}\mathbf{r} \times \mathbf{r} = \mathbf{r} \times \dot{\mathbf{v}} = \mathbf{v} \times \mathbf{v} + \mathbf{r} \times \dot{\mathbf{v}} = \\ &= \frac{d}{dt}(\mathbf{r} \times \mathbf{v}) = 0 \end{aligned} \tag{2.5}$$

Note:

$$\frac{d(a \times b)}{dt} = a \times \frac{db}{dt} + \frac{da}{dt} \times b$$

The quantity $\mathbf{r} \times \mathbf{v} = \mathit{constant}$ (since its derivative is equal to zero) is the *angular momentum* $\mathbf{h}$ per unit mass. In this way, it is proven the principle of angular momentum conservation.

Since $\mathbf{h}$ is constant, another important implication is that $\mathbf{r}$ and $\mathbf{v}$ always remain in the same plane. This plane is called *Orbital Plane*.

The result of the cross product between two vectors is always perpendicular to the plane described by them. $\mathbf{h}$ constant implies that also the plane must be constant. In this way, the motion of the small body occurs on a plane if there aren't external forces applied to it.

2.1.4 Orbit Equation

Taking the equation 2.3 it is performed the cross product for $\mathbf{h}$:

$$\begin{aligned} &\dot{\mathbf{v}} \times \mathbf{h} + \frac{\mu}{r^3}\mathbf{r} \times \mathbf{h} = \dot{\mathbf{v}} \times \mathbf{h} + \mathbf{v} \times \dot{\mathbf{h}} + \frac{\mu}{r^3}\mathbf{r} \times \mathbf{h} = \\ &= \dot{\mathbf{v}} \times \mathbf{h} + \frac{\mu}{r^3}\mathbf{r} \times \mathbf{h} = \\ &= \frac{d}{dt}(\mathbf{v} \times \mathbf{h} - \frac{\mu}{r}\mathbf{r}) = 0 \end{aligned}$$

Note:

$$\begin{aligned} &\frac{d}{dt}(-\frac{\mathbf{r}}{r}) = \frac{\dot{r}}{r^2}\mathbf{r} - \frac{1}{r}\mathbf{v} = \frac{1}{2r^3}(\frac{d}{dt}r^2)\mathbf{r} - \frac{1}{r}\mathbf{v} = \frac{1}{2r^3}[\frac{d}{dt}(\mathbf{r} \cdot \mathbf{r})]\mathbf{r} - \frac{1}{r^3}(\mathbf{r} \cdot \mathbf{r})\mathbf{v} = \\ &= \frac{1}{r^3}[(\mathbf{r} \cdot \mathbf{v})\mathbf{r} - (\mathbf{r} \cdot \mathbf{r})\mathbf{v}] = \frac{1}{r^3}\mathbf{r} \times (\mathbf{r} \times \mathbf{v}) = \frac{1}{r^3}\mathbf{r} \times \mathbf{h} \end{aligned} \tag{2.6}$$

The quantity $\mathbf{v} \times \mathbf{h} - \frac{\mu}{r}\mathbf{r} = \mu\mathbf{e}$ is constant. $\mathbf{e}$ is the *eccentricity vector* (its magnitude $|\mathbf{e}| = e$ is called just *eccentricity* of the system).

Taking this last equation is performed the dot product for $\mathbf{r}$:

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

Considering the scalar triple product: $\mathbf{r} \cdot (\mathbf{v} \times \mathbf{h}) = (\mathbf{r} \times \mathbf{v}) \cdot \mathbf{h} = \mathbf{h} \cdot \mathbf{h} = h^2$ The previous equation becomes:

$$h^2 - \frac{\mu}{r}\mathbf{r} \cdot \mathbf{r} = h^2 - \mu r = \mu r e \cos\theta$$

Where θ is the *true anomaly.* It is the angle between the vectors $\mathbf{r}$ and $\mathbf{e}$. This angle is an important quantity used to describe the orbit of the small body around the big one.

Considering $p = \frac{h^2}{\mu} = constant$ (that is the *semilatus rectum* also called the *parameter*) and solving the previous equation for $\mathbf{r}$, it is obtained the so called *Orbit Equation (ORE)*:

$$r = \frac{p}{1 + e\cos\theta} \tag{2.7}$$

2.1.5 Orbit Geometry

The ORE, from standard analytical geometry, is the equation of a *conic section* written in terms of polar coordinates r and θ. For $\theta \in [0; 2\pi]$, r describes a conic.

A conic section (or just a conic) is a curve obtained as the intersection of a cone with a plane:

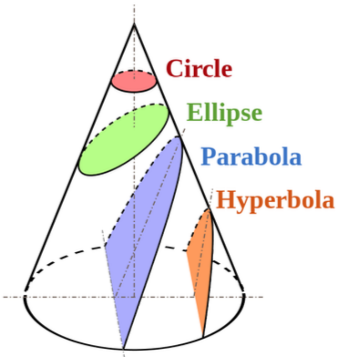

Figure 2.2: Conics

Staring from the ORE (see equation: 2.7) it can be distinguished:

1. *Circle* If $e = 0$;

2. *Ellipse* If $0 < e <= 1$;

3. *Parabola* If $e = 1$;

4. *Hyperbola* If $e > 1$;

The eccentricity is a very important parameter that determines the orbit shape.

From standard geometry the following properties hold:

- The origin is located at one focus;
- θ is measured from the point on the conic nearest to the focus;
- p determines the size. It depends on the angular momentum and on μ. By considering the same big body μ doesn't change, while p increase with h.
- e determine the shape;

Ellipse

A circle can be see as an ellipse where the two foci are coincident and the two semi-axes are equal. An ellipse is the locus of points the sum of whose distances from two fixed points (called *foci*) is constant and equal to $2a$.

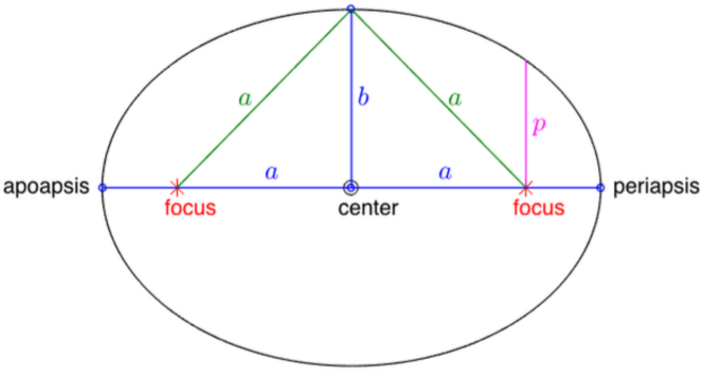

Figure 2.3: Ellipse

In the upper figure: a is the semi-major axis, b is the semi-minor axis, p is the semilatus rectum.

The two extreme points are called *Apsides*. One is the *Periapsis* and corresponding to $\theta = 0$. The distance from the main focus is:

$$r_p = \frac{p}{1+e} \tag{2.8}$$

The other one is called *Apoapsis* and corresponding to $\theta = \pi$. The distance from the main focus is:

$$r_p = \frac{p}{1-e} \tag{2.9}$$

In the case of planet orbiting around the Sun these two points take the names of *Perihelion* and *Aphelion* respectively. While, in the case of a body orbiting around the Earth they take the names of *Perigee* and *Apogee*.

Considering the ORE it is possible to write the main ellipses parameters:

- *Semi-major axis:*
$$a = \frac{p}{1-e^2}$$
- *Semi-minor axis:*
$$b = a\sqrt{1-e^2}$$
- *distance center-focus:*
$$c = ae$$

The magnitude of the velocity can be expressed by the *vis-viva equation.* Considering that the mechanical energy is constant, it can be written at the Periapsis and at the Apoapsis:

$$\varepsilon = \frac{v_p^2}{2} - \frac{\mu}{r_p} = \frac{v_a^2}{2} - \frac{\mu}{r_a} = constant$$

Now, it is considered a reference frame focus-centered with an axis passing thru the Periapsis, positive in the direction focus-periapsis, on the orbit plane. The other axis is perpendicular and stays on the orbit plane, with direction taken in order to have that the third one has the same direction of h. In this way, the velocity and radius vectors are perpendicular at Periapsis and Apoapsis. It is possible to write:

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

Rearranging the previous equation and substituting with v_p:

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

Solving w.r.t. the kinetic energy at the Apoapsis:

$$\frac{1}{2} v_a^2 = \mu \frac{r_p}{r_a(r_a + r_p)}$$

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

$$\frac{1}{2} v_a^2 = \mu \frac{2a - r_a}{2ar_a} = \frac{\mu}{r_a} - \frac{\mu}{2a}$$

Substituting this value in ε it obtained that the total mechanical energy is:

$$\varepsilon = \frac{v_a^2}{2} - \frac{\mu}{r_a} = -\frac{\mu}{2a} = negative\ constant$$

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

$$v = \sqrt{\frac{2\mu}{r} + 2\varepsilon} = \sqrt{\frac{2\mu}{r} - \frac{\mu}{a}}$$

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

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*).

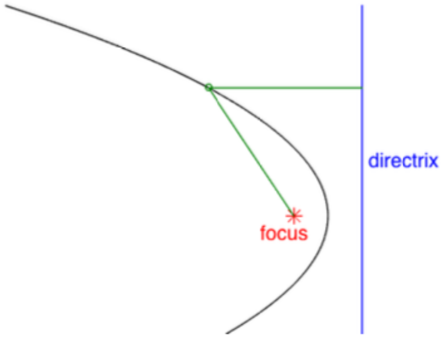

Figure 2.4: Parabola

As seen before, in this case $e = 1$. This implies:

- $r_a \to \infty$
- $a \to \infty$
- $\varepsilon = 0$

Using the *vis-viva equation*, for any orbital position with radius r, the corresponding velocity is:

$$v_{escape} = \sqrt{\frac{2\mu}{r}}$$

Where v_{escape} is the *escape velocity*. It allows to leaving a closed orbit. If an object is on an ellipsoidal orbit the velocity can be increased until v_{escape}. Then this limit the obit becomes no more closed and an open parabolic orbit it is obtained.

Hyperbola

An hyperbola is the locus of points the difference of whose distances from two fixed points (*foci*) is constant and equal to $-2a$.

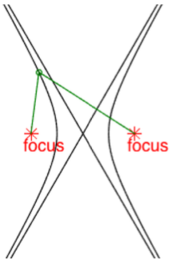

Figure 2.5: Hyperbola

In this case $e > 1$. This implies that the total energy is positive:

$$\varepsilon = \frac{v_\infty^2}{2} > 0$$

It is possible to compute asymptotic quantities for $r \to \infty$:

- *Angle θ:*

$$\theta_\infty = \arccos(-\frac{1}{e})$$

- *Velocity:*

$$v_\infty = \sqrt{\frac{\mu}{|a|}}$$

An interesting observation is that it is possible to increase the speed of an object (S/C) using an hyperbolic orbit. A body passing close to a moving planet is subject to a velocity increase, without being captured by the planet gravity. This maneuver is called *hyperbolic passage*. It was adopted first by US probe Mariner 10 (1973) to fly-by Venus (once) and Mercury (three times).

Orbit Geometry Considerations

Starting from the FR2B equation several conservation laws and the ORE were found. The ORE describes the geometry of the orbit of a small body traveling on an orbit around a big body. While, the FR2B describe the dynamics of the same motion. Indeed, it is a differential equation and its solution is given by integration starting from given initial conditions. The solution is $\mathbf{r}(t)$ in function of the time.

As seen, the orbit is always on a plane, by assuming that this plane coincides with an (x, y)-plane it is possible to write: $z(0) = z(t) = \dot{z} = 0$. In this way: $\mathbf{r}(t) = (x(t), y(t), z(t) = 0)$ where $x(t)$ and $y(t)$ satisfy the ORE when they are transformed as:

- (x, y) *Cartesian coordinates:*

$$r(t) = \sqrt{x^2(t) + y^2(t)}$$

- (r, θ) *Polar coordinates:*

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

While, the inverse transformation is given by:

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

2.1.6 Reference Frames

There are different types of reference frames that are used in the space field. Here, only four of them are considered:

1. *Local-Vertical Local-Horizontal (LVLH) Reference Frame:* $R_l = \{P_1, \mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3\}$. It is a non-inertial frame;

2. *Local-Orbital Reference Frame (LORF):* $R_o = \{P_1, \mathbf{o}_1, \mathbf{o}_2, \mathbf{o}_3\}$. It is a non-inertial frame;

3. *Perifocal (Perigee) Frame (PF):* $R_p = \{P_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\}$. It is an inertial frame;

4. *Geocentric Equatorial (GE) Reference Frame:* $R_{ge} = \{P_0, \hat{\mathbf{I}}, \hat{\mathbf{J}}, \hat{\mathbf{K}}\}$. It is an inertial frame;

The first three are more general and they are associated to an elliptic orbit. The fourth one is associated with the Earth.

The position of the big body is associated to P_0. While, the position of the small one is associated to P_1.

Local-Vertical Local-Horizontal Frame

The origin P_1 moves with the small body on the orbit. The unit vectors of the axes are:

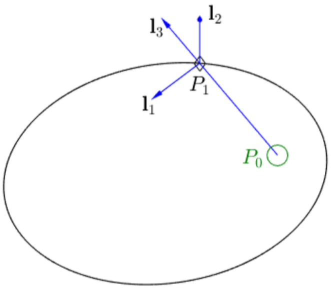

Figure 2.6: LVLH

- *Local vertical* $\boldsymbol{l}_3$*:* It is defined along the direction $P_0 \rightarrow P_1$ on the orbit plane;
- *Local horizontal* $\boldsymbol{l}_1$*:* It is perpendicular to $\mathbf{l}_3$ on the orbit plane. The sign is concordant with the velocity;
- *Orbit pole* $\boldsymbol{l}_1$*:* It defined by the vector product: $\mathbf{l}_2 = \mathbf{l}_3 \times \mathbf{l}_1$. It is perpendicular to the orbit plane;

Local-Orbital Reference Frame

The origin P_1 moves with the small body on the orbit. The unit vectors of the axes are:

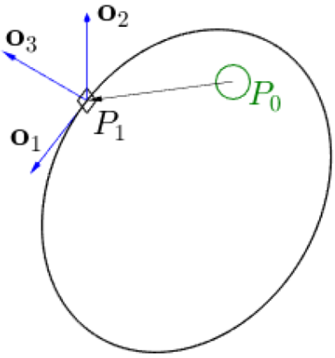

Figure 2.7: LORF

- $\boldsymbol{o}_1$*:* It has the same direction of the instantaneous normalized velocity. It is on the orbit plane and it is tangent to the orbit;
- *Orbit pole* $\boldsymbol{o}_2$*:* It is perpendicular to the orbit plane;
- $\boldsymbol{o}_3$*:* It defined by the vector product: $\mathbf{o}_3 = \mathbf{o}_1 \times \mathbf{o}_2$. It is on the orbit plane;

Perifocal Frame

The origin P_0 is fixed with the big body. This time the reference is inertial. The unit vectors of the axes are:

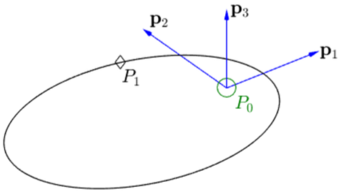

Figure 2.8: PF

- $\boldsymbol{p}_1$*:* It is defined along the direction of the eccentricity vector $\mathbf{p}_1 = \mathbf{e}/e$. This unit vector passes thru the periapsis on the orbit plane.

- *Orbit pole* $\boldsymbol{p}_3$*:* It is perpendicular to the orbit plane;

- $\boldsymbol{p}_2$*:* It defined by the vector product: $\mathbf{p}_2 = \mathbf{p}_3 \times \mathbf{p}_1$. It is on the orbit plane;

It is also called *Perigee Frame* when it is associated with the Earth.

It is worth to note that:

- P_0 is one focus of the ellipse;

- When the elliptic orbit is reduced to a circular orbit the first two frames coincide;

- The direction of the orbit pole is always the same and coincident with $\mathbf{h}$;

Geocentric Equatorial Frame

This time, this reference frame is defined only for the Earth. The origin P_0 is coincident with its CoM. The axes $\hat{\mathbf{I}}$ and $\hat{\mathbf{J}}$ are on the *Equatorial Plane.*

$\hat{\mathbf{I}} \leftrightarrow \mathbf{X}_\epsilon$ unit vector coincides with the *vernal equinox direction*: it is defined as the *Earth-Sun* direction when the Earth is at the first day of spring (north hemisphere). $\hat{\mathbf{K}}$ is coincident with the polar rotation axis, positive from the Earth CoM towards the North pole. $\hat{\mathbf{J}}$ is defined by the cross product: $\hat{\mathbf{J}} = \hat{\mathbf{K}} \times \hat{\mathbf{I}}$ and it leis on the equatorial plane.

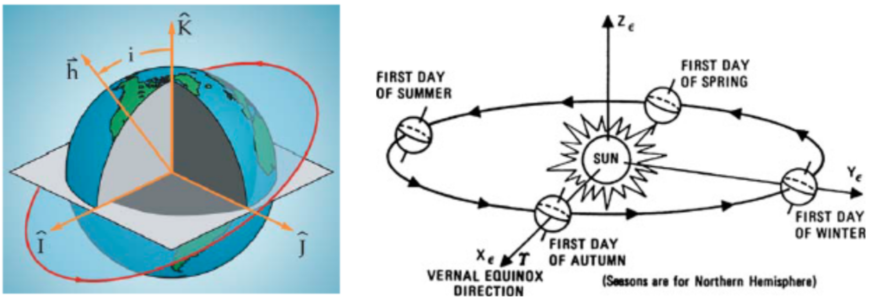

Figure 2.9: GE

It is worth to note that this frame doesn't rotate with the Earth and it is independent from the small body orbit.

2.1.7 Orbital Elements

It is considered an elliptic orbit around the Earth. In this situation two planes can be distinguished:

1. *Orbit Plane:* That is the plane where the orbit lays;

2. *Equatorial Plane:* That is the plane that passes through the Earth's equator;

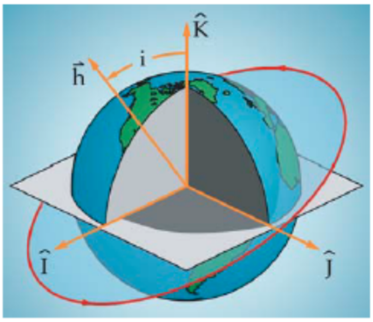

Figure 2.10: Elliptic Orbit around the Earth

A first important notion is the one of *line of nodes*. It is the intersection between these two planes. While, the angle between the two is called *inclination* i.

For this elliptical orbit another important quantity is the *ascending node*. It hold on the line of nodes and it is the point defined by the intersection between the orbit and the equatorial plane. This intersection defines two points, but it is considered only the point when the orbit goes from the lower side of the equatorial plane to the upper side. In this way, it is possible to uniquely define the ascending node.

Looking the following figure it is possible to found other important quantities:

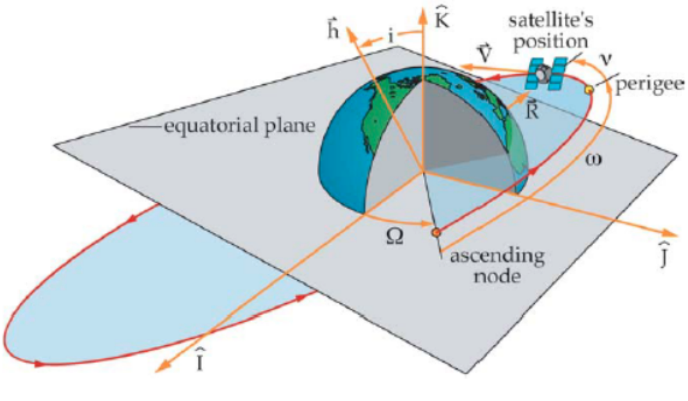

Figure 2.11: Orbital Elements Description

The angle from the unit vector $\hat{\mathbf{I}}$ to the ascending node is Ω and it is called *right ascension of the ascending node*. The angle from the ascending node to the perigee is ω and it is called *argument of perigee*. While, the angle ν is the the *true anomaly* (already defined in section 2.1.4 $\nu = \theta$). It is defined as the angle defined from the perigee to the S/C position.

These quantities are a part of the so called *six classical orbital elements.* They are five constant quantities $(a, e, \Omega, i, \omega)$ plus the true anomaly ν, that depends on the S/C position. Whit these six elements it is possible to uniquely define a given orbit. Indeed, the first five elements define: the shape, the size, and the features of the orbit. ν gives the position on that orbit (the position is not fixed, but it changes in time).

Sometimes the *perigee passage* time is used instead of the true anomaly.

From Position and Velocity to Orbital Elements

The orbital elements are equivalent to position and velocity vectors of the S/C. Indeed, once the orbit geometry is fully definite (by the constant orbital elements) in the GE frame, it is possible, thru ν, to define completely the motion of the S/C.

It is important to move from $\mathbf{r}$ and $\mathbf{v}$ to the six orbital element and vice-versa.

Starting from $\mathbf{r}$ and $\mathbf{v}$ expressed in the GE frame it is possible to write the following quantities:

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

$$a = \frac{p}{1-e^2} = \frac{h^2}{\mu(1-e^2)}, \quad e = |\mathbf{e}|, \quad \cos i = \hat{\mathbf{K}} \cdot \frac{\mathbf{h}}{h} \tag{2.10}$$

$$\cos\omega = \hat{\mathbf{I}}' \cdot \frac{\mathbf{e}}{e}, \quad \cos\Omega = \hat{\mathbf{I}} \cdot \hat{\mathbf{I}}', \quad \cos\theta = \mathbf{r} \cdot \frac{\mathbf{e}}{re}$$

Where $\hat{\mathbf{I}}'$ is the unit vector from the Earth CoM to the ascending node.

Other useful parameters are the *eccentric anomaly* E and the *Period* P:

$$\tan\frac{E}{2} = \sqrt{\frac{1-e}{1+e}}\tan\frac{\theta}{2}, \quad P = 2\pi\sqrt{\frac{a^3}{\mu}} \tag{2.11}$$

From Orbital Elements to Position and Velocity

Supposing that the six orbital element are known it is possible to write the position and the velocity of the S/C:

$$p = a(1-e^2), \quad r = \frac{p}{1+e\cos\theta} \tag{2.12}$$

$\mathbf{r}$ and $\mathbf{v}$ expressed in the PF are given by:

$$\mathbf{r} = \begin{bmatrix} r\cos\theta \\ r\sin\theta \\ 0 \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} -\sqrt{\mu/p}\ \sin\theta \\ \sqrt{\mu/p}\ (e+\cos\theta) \\ 0 \end{bmatrix} \tag{2.13}$$

In order to transform them in GE coordinates a Direction Cosine Matrix (DCM) is needed. Remembering that the transformation is the inverse of the rotation and exploiting the proper Euler angles 313. It is possible to write the transformation matrices. The angles used are:

$$\begin{aligned}&\textit{Transformation PF}\rightarrow\textit{GE:} \qquad \mathbf{T}_{313}(\Omega, i, \omega)\\ &\textit{Transformation GE}\rightarrow\textit{PF:} \qquad \mathbf{T}_{313}(-\omega, -i, -\Omega)\end{aligned} \tag{2.14}$$

2.1.8 HCW equations

In order to write the HCW equations it is considered a system composed by three bodies with masses (m_0, m_1, m_2) and $m_0 \gg m_1, m_2$. The positions of these bodies w.r.t. an inertial frame are (r_0, r_1, r_2). Considering the same assumption on the forces done for the 2B problem it is possible to write the second Newton's laws:

$$\begin{aligned}&\dot{\mathbf{v}}_i = \sum_{i\neq j}^{2} \frac{Gm_j}{r_{ij}^3}(\mathbf{r}_j - \mathbf{r}_i) + \frac{\mathbf{F}_i}{m_i}, \qquad i = 0,1,2\\ &r_{ji} = |\mathbf{r}_j - \mathbf{r}_i|\\ \\ &\dot{\mathbf{v}}_0 = \frac{Gm_1}{r_{01}^3}(\mathbf{r}_1 - \mathbf{r}_0) + \frac{Gm_2}{r_{02}^3}(\mathbf{r}_2 - \mathbf{r}_0) + \frac{\mathbf{F}_0}{m_0}\\ &\dot{\mathbf{v}}_1 = \frac{\mu}{r_{10}^3}(\mathbf{r}_0 - \mathbf{r}_1) + \frac{Gm_2}{r_{12}^3}(\mathbf{r}_2 - \mathbf{r}_1) + \frac{\mathbf{F}_1}{m_1}\\ &\dot{\mathbf{v}}_2 = \frac{\mu}{r_{20}^3}(\mathbf{r}_0 - \mathbf{r}_2) + \frac{Gm_1}{r_{21}^3}(\mathbf{r}_1 - \mathbf{r}_2) + \frac{\mathbf{F}_2}{m_2}\end{aligned} \tag{2.15}$$

Similarly to the 2B problem, the CoM of this system can be considered coincident with the center of the big body with mass m_0. In this way: $r_0 = 0$. The gravity force contribution between the two small bodies is negligible w.r.t. the one given by the the big body:

$$\left|\frac{Gm_1m_2}{r_{21}^3}(\mathbf{r}_1 - \mathbf{r}_2)\right| = \left|\frac{Gm_2m_1}{r_{12}^3}(\mathbf{r}_2 - \mathbf{r}_1)\right| \approx 0$$

It is assumed that each small body is subjected to the central gravitational field generated by the the big body, while the mutual gravity interaction between the two is negligible.

In this way, the two small bodies obey to the restricted 2B equations, written in a generic inertial frame ($R = \{P_0, \mathbf{i}_1, \mathbf{i}_2, \mathbf{i}_3\}$) centered with the big body:

$$\begin{aligned}\dot{\mathbf{v}}_1 + \mu\frac{\mathbf{r}_1}{r_1^3} &= +\frac{1}{m_1}\mathbf{F}_1\\ \dot{\mathbf{v}}_2 + \mu\frac{\mathbf{r}_2}{r_2^3} &= +\frac{1}{m_2}\mathbf{F}_2\end{aligned} \tag{2.16}$$

In several space applications, like rendezvous, it is of interest to describe the relative motion of the two small bodies. For this purpose, it is considering that they can be seen as: P_1 is the point mass of a chief S/C, P_2 is the point mass of a deputy S/C that has to *rendezvous* with P_1. While, P_0 is the point mass of the big body (it can be seen as the Earth).

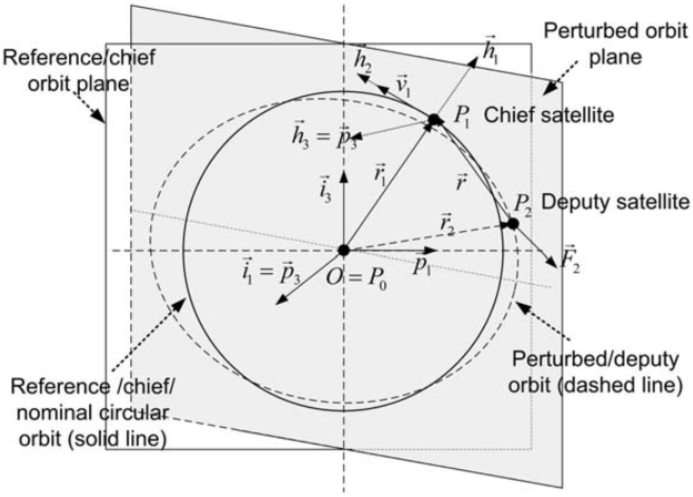

Figure 2.12: References and Perturbed Orbits

Considering that the relative reference is *chief-centered*, and assuming that the external force applied to it is zero $\mathbf{F}_1 = 0$, it is possible to write:

$$\begin{aligned} &\mathbf{r} = \mathbf{r}_2 - \mathbf{r}_1, \quad r = |\mathbf{r}|, \quad \mathbf{F} = \mathbf{F}_2 \\ &\dot{\mathbf{v}}_2 = \dot{\mathbf{v}}_1 - \dot{\mathbf{v}} = -\frac{\mu}{r_2^3}(\mathbf{r}_1 + \mathbf{r}) + \frac{\mathbf{F}}{m_2} = \\ &= -\frac{\mu}{|\mathbf{r}_1 + \mathbf{r}|^3}(\mathbf{r}_1 + \mathbf{r}) + \frac{\mathbf{F}}{m_2} \end{aligned} \tag{2.17}$$

This equation describes the motion of P_2 relative to P_1 in a generic inertial frame with origin at P_0. In the following, the inertial frame that will be chosen will be a PF build on the orbit of the chief S/C (as seen in section 2.1.6): $R_p = \{P_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\}$. While, the non inertial frame chosen is the LVLH frame build always on the chief S/C (this frame is also called *Hill-Frame*): $R_h = \{P_1, \mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3\}$.

The inertial acceleration can be written as [9]:

$$\dot{\mathbf{v}} = \dot{\mathbf{v}}_h + 2\,\omega_1 \times \mathbf{v}_h + \omega_1 \times (\omega_1 \times \mathbf{r}) + \dot{\omega}_1 \times \mathbf{r} \tag{2.18}$$

Where $\dot{\mathbf{v}}_h$ and $\mathbf{v}_h$ are the acceleration and velocity vectors in the Hill frame, $\dot{\omega}_1$ and ω_1 are the acceleration and angular velocity vectors of P_1 seen from the generic inertial frame R. Considering that the Hill frame is constructed as

$R_h = \{P_1, \mathbf{h}_1 = \mathbf{r}_1/r_1, \mathbf{h}_2, \mathbf{h}_3 = \omega_1/|\omega_1|\}$, it can be written:

$$\begin{aligned}
&\mathbf{r}_1 = r_1\mathbf{h}_1, \quad \mathbf{r}_h = \mathbf{r} = x\mathbf{h}_1 + y\mathbf{h}_2 + z\mathbf{h}_3 = \mathbf{R}\mathbf{r}' \\
&\mathbf{v}_h = \dot{x}\mathbf{h}_1 + \dot{y}\mathbf{h}_2 + \dot{z}\mathbf{h}_3 = \mathbf{R}\dot{\mathbf{r}}' \\
&\dot{\mathbf{v}}_h = \ddot{x}\mathbf{h}_1 + \ddot{y}\mathbf{h}_2 + \ddot{z}\mathbf{h}_3 = \mathbf{R}\ddot{\mathbf{r}}' \\
&\mathbf{F}_h = F_1\mathbf{h}_1 + F_2\mathbf{h}_2 + F_3\mathbf{h}_3 = \mathbf{R}\mathbf{F}'
\end{aligned} \tag{2.19}$$

It is worth to note that the same physical vector (linear combination of the unit vectors) has different representations (coordinate vectors) depending in which frame it is written. The second members of the previous equations are the products between $\mathbf{r}' = [x, y, z]$ which are the coordinates of P_2 seen in the Hill frame, while $\mathbf{R} = [\mathbf{h}_1\ \mathbf{h}_2\ \mathbf{h}_3]$ is the matrix that contains the unit vectors of the Hill frame. This can be seen as a matrix that can be constructed from the P_1 position components in the inertial frame, and it can be used as DCM to transform the coordinates from a frame to another (this will be used in the next chapter 4). Indeed, the i-th unit vector $\mathbf{h}_i$ can be seen has the linear combination of the inertial frame unit vectors.

Considering that the chief orbit is described by the FR2B equation ($\mathbf{h}$ is constant and it lays along the orbit pole $\mathbf{h}_3$). Then, taking the equations 2.16, 2.17, and the previous ones. It is possible to write:

$$\begin{aligned}
&\mathbf{r}_1 = r_1\mathbf{h}_1, \quad \dot{\mathbf{v}}_1 = \dot{v}_1\mathbf{h}_1 = -\frac{\mu}{r_1^2}\mathbf{h}_1 \\
&\dot{\mathbf{v}}_2 = -\frac{\mu}{r_2^3}\mathbf{r}_2 + \frac{\mathbf{F}}{m_2} = -\frac{\mu}{r_2^3}(\mathbf{r}_1 + \mathbf{r}) + \frac{\mathbf{F}}{m_2} = \\
&= -\frac{\mu}{r_2^3}((r_1 + x)\mathbf{h}_1 + y\mathbf{h}_2 + z\mathbf{h}_3) + \frac{F_1}{m_2}\mathbf{h}_1 + \frac{F_2}{m_2}\mathbf{h}_2 + \frac{F_3}{m_2}\mathbf{h}_3 \\
&\omega_1 = |\omega_1|\mathbf{h}_3, \quad \dot{\omega}_1 = |\dot{\omega}_1|\mathbf{h}_3
\end{aligned} \tag{2.20}$$

Now, replacing the results in 2.18 and after doing some simplifications, the nonlinear differential equations of the relative motion are found to be (considering that $\mathbf{h}_1 \times \mathbf{h}_2 = \mathbf{h}_3$, $\mathbf{h}_2 \times \mathbf{h}_1 = -\mathbf{h}_3$, and so on) :

$$\begin{aligned}
&\dot{\mathbf{v}} = \dot{\mathbf{v}}_2 - \dot{\mathbf{v}}_1 = -\frac{\mu}{r_2^3}((r_1 + x)\mathbf{h}_1 + y\mathbf{h}_2 + z\mathbf{h}_3) \\
&+ \frac{F_1}{m_2}\mathbf{h}_1 + \frac{F_2}{m_2}\mathbf{h}_2 + \frac{F_3}{m_2}\mathbf{h}_3 + \frac{\mu}{r_1^3}r_1\mathbf{h}_1 = \\
&= \dot{\mathbf{v}}_h + 2\ \omega_1 \times \mathbf{v}_h + \omega_1 \times (\omega_1 \times \mathbf{r}) + \dot{\omega}_1 \times \mathbf{r} = \\
&= \ddot{x}\mathbf{h}_1 + \ddot{y}\mathbf{h}_2 + \ddot{z}\mathbf{h}_3 + 2|\omega_1|\dot{x}\mathbf{h}_2 - 2|\omega_1|\dot{y}\mathbf{h}_1 + |\omega_1|\mathbf{h}_3 \times (|\omega_1|x\mathbf{h}_2 - |\omega_1|y\mathbf{h}_1) \\
&+ |\dot{\omega}_1|x\mathbf{h}_2 - |\dot{\omega}_1|x\mathbf{h}_1 = \\
&= \ddot{x}\mathbf{h}_1 + \ddot{y}\mathbf{h}_2 + \ddot{z}\mathbf{h}_3 + 2|\omega_1|\dot{x}\mathbf{h}_2 - 2|\omega_1|\dot{y}\mathbf{h}_1 \\
&- |\omega_1|^2x\mathbf{h}_1 - |\omega_1|^2y\mathbf{h}_2 + |\dot{\omega}_1|x\mathbf{h}_2 - |\dot{\omega}_1|x\mathbf{h}_1
\end{aligned} \tag{2.21}$$

$$\begin{aligned}
&\textit{Rearranging along the unit vectors directions and writing in function}\\
&\textit{of the relative accelerations:}\\
&\ddot{x} = 2\omega_1\dot{y} + \dot{\omega}_1 y + \omega_1^2 x + \frac{\mu}{r_1^3}r_1 - \frac{\mu}{r_2^3}(r_1 + x) + \frac{F_1}{m_2}\\
&\ddot{y} = -2\omega_1\dot{x} - \dot{\omega}_1 x + \omega_1^2 y - \frac{\mu}{r_2^3}y + \frac{F_2}{m_2}\\
&\ddot{z} = -\frac{\mu}{r_2^3}z + \frac{F_3}{m_2}
\end{aligned} \tag{2.22}$$

They can only be solved by numerical integration.

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

$$\begin{aligned}
&\frac{\mu}{r_2^3} \approx \frac{\mu}{r_1^3}\left(1 - 3\frac{x}{r_1}\right)\\
&\textit{Now, Considering:}\\
&\mathbf{v}_1 = \omega_1 \times \mathbf{r}_1 \textit{ at the apsides: } \mathbf{v}_1 = \sqrt{\mu/a}, \quad \mathbf{r}_1 = a\\
&\quad \Longrightarrow \omega_1 = \sqrt{\mu/a^3} \approx \sqrt{\mu/r_1^3} = \textit{constant}\\
&\omega = \omega_1, \quad \dot{\omega} = 0
\end{aligned} \tag{2.23}$$

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 it is possible to obtain the HCW equations:

$$\begin{aligned}
\ddot{x} &= 2\omega\dot{y} + 3\omega^2 x + \frac{F_1}{m_2}\\
\ddot{y} &= -2\omega\dot{x} + \frac{F_2}{m_2}\\
\ddot{z} &= -2\omega^2 z + \frac{F_3}{m_2}
\end{aligned} \tag{2.24}$$

2.2 Controller Comparison

There are different types of control methods. In this section will be mentioned, without entering in details, some of them in order to select the better controller (for the writer) for the considered space mission.

Since the equation that describe the dynamics of the system are nonlinear (note that also the HCW equations become nonlinear if the S/C mass is not constant), the research is concentrated in those methods used for controlling nonlinear systems.

2.2.1 General Requirements

When a controller has to be designed there are some important requirements that it has to satisfy:

- *Stability:* In many application it is required to obtain a stable system (either in a local or in global sense). Indeed, if some state variable diverges to infinity it can damage the plant to control (zero dynamics problem). In general, the stability is studied with: Lyapunov *Direct Method* and *Linearization Method*. In the second one, the system is linearized around a given Equilibrium Point and it is possible to obtain only local results. The first one is more general, but it requires to find a Lyapunov function. In some control methods it is not possible to obtain analytical solution for understanding if the stability properties are satisfied globally.

- *Accuracy and speed response:* The output should converge in a quick way to the reference and it must track very precisely the reference. In this way, it is possible to understand whether the system works properly.

- *Constraints fulfillment:* In many applications there can be present external constraints on the state variables or on the outputs. There can be also internal constraints e.g. for the input. They can be static or varying in time. In any case, what is important is that the controller must be take into account that constraints.

- *Robustness:* It is a very important skill that the control system has to have in order to behave correctly also when there are system variations. In real system there are always present: disturbances, noises, neglected dynamics, and uncertainties of various nature. The control system must work even if all all these factors are not considered, but they are present.

2.2.2 Nonlinear Control Design Methods

Most common methods used for nonlinear system are:

1. *Methods based on linearization e.g. Gain Scheduling:* These methods are based on the linearization of the system around a given working point (made with the jacobians). In this way, it is possible to control the system in a neighborhood of the considered point with a linear controller. In the Gain Scheduling approach, it is possible to linearize the system in different points and, thanks to the scheduling of a variable, to control it by different linear controllers (one for each point in which the linearization is made).

2. *Feedback Linearization:* The nonlinear plant is transformed into a linear system by means of some (nonlinear) state transformation realized via feedback. This linearization is not local and it is exact.

3. *Sliding Mode Control:* This is based on the concept of *sliding surface.* This is a subset of the state space, with some features, where the trajectory of the plant is desired to lay. The sliding surface is designed in order to obtain the desired behavior of the plant.

4. *Nonlinear Model Predictive Control:* This technique will be analyzed in the next chapter 3.

5. *Embedded Model Control:* This approach is based on linear control, but the nonlinearities, uncertainties, and noises are treated as disturbances. The idea is to build an *extended state observer* able to estimate all the quantities considered as disturbances. Then, a control law is designed considering a state feedback linear term plus another term that is used to cancel the estimated disturbances. In this way, the sources that create problems are eliminated.

6. *Adaptive Control:* The key point of this method is parameter estimation. An adaptive controller is a controller with adjustment parameters and a mechanism for adjusting the parameters. It does not need apriori information about the bounds on uncertain or time-varying parameters. It is concerned with control law changing itself.

2.2.3 Controller Choice

Among the mentioned methodologies it is chosen an adaptive version of the NMPC. This choice is done for several reasons:

1. The system equations are not in *affine form* in u. This form is needed in the Feedback Linearization (FBL) and in the Sliding Mode Control (SMC);

2. There are constraint that must be accounted for.

3. It is not possible to use a Gain Scheduling (GS) approach because the predictable variation of the parameters is unknown.

4. The NMPC works in closed-loop.

5. The NMPC is able to do *Guidance* and *Control* in the same time. This is due to the fact that the controller calculates an optimal trajectory on the basis of the constraints accounted for. In the same time it gives to the system the control law needed to follow this optimal trajectory.

Thanks to these features the NMPC can be seen as a step forward in autonomous guidance and control. In order to make adaptive the NMPC an estimation algorithm is needed.

2.2.4 Adaptive NMPC

The construction of an adaptive controller contains the following steps:

1. Characterize the desired behavior of the closed-loop system;
2. Determine a suitable control law with adjustable parameters;
3. Finding a mechanism for adjusting the parameters;
4. Implement the control law;

The general structure is composed by two loops: one loop is a normal feedback with the process and the controller the other one is a parameter adjusting loop [16]. Adaptive control is different from robust control in that it does not need a priori information about the bounds on uncertain or time-varying parameters; robust control guarantees that if the changes are within given bounds the control law need not be changed, while adaptive control is concerned with control law changing itself.

The key idea in this work is to combine the robust Model Predictive Control (MPC) method with an adaptive parameter estimation method. The development of adaptive-type MPC schemes is one of the research issues for control of constrained systems. One of the main reasons is the difficulty to guarantee the fulfillment of constraints in the presence of an adaptive mechanism. In order to overcome this problem, the future behavior of the real system must be predicted while updating the system parameters on-line. In addition, it seems extremely difficult to guarantee both feasibility and stability theoretically whenever an adaptive approach to MPC is adopted. Anyway there are works that propose attractive ways to handle with this controller [17]-[20].

2.3 Estimation Process

The estimation problem refers to the empirical evaluation of an uncertain variable, like an unknown characteristic parameter or a remote signal, on the basis of observations and experimental measurements of the phenomenon under investigation. There are two main kind of approaches to estimate variable:

- Using mathematical formulations in order to obtain an estimated variable that recursively converges to the real value. An algorithm based on this is, for example, the *Recursive Least Square Estimator.*
- Estimate variables with an observer (filter). The parameters are treated as states in these filters. The identification problem becomes an estimation problem of the state in this way. An example of this is the *Extended Kalman Filter (EKF).*

Among the huge number of possible estimation algorithm, in this work it will be considered only two methods. One for each of the two main methods. The algorithms will be discussed deeply later, in the proper sections. The first algorithm is based on a recursive estimation. While the second one is based on the EKF.

Chapter 3

Model Predictive Control

This chapter deals with on one of the most powerful and flexible control approach available today: the MPC and its nonlinear version NMPC, more general than the previous one.

The MPC could be compared to a chess player. A good chess player chooses a move by projecting in the future the game scenario and by trying to predict how the opponent will answer to a sequence of moves. If the opponent replies in an unexpected way, to counteract the effect of the new move a game scenario rescheduling is needed. In the same way the MPC adjusts its behavior to perform in the best way under a varying scenario. As bibliographic references, see [7], [11], [12], [13], [14], and [15].

3.1 Introduction

The most general case to discuss is the NMPC. This flexible approach can be employed to control nonlinear systems and it allows to deal with constraints.

These can be on the command input u, on the state x, on the output y. Moreover they can be used to systematically manage the trade-off between performance and command activity.

In order to increase performances, in general, it is needed an high command activity. This, in real applications, implies high energy consumption (imagine to require from a car an high acceleration, this high acceleration implies an high fuel consumption). The MPC is able to do an efficient management of the trade-off between the energy consumption and the performances finding the best command activity to this task.

The MPC approach is based on two fundamental operations, that are computed at each sampling time:

- Prediction over a finite time horizon;
- Optimization.

In the Prediction step the Controller uses a model of the plant to predict the future behavior of the system. On the basis of the predicted behavior, the algorithm chooses the best input sequence to be provided to the system (on-line Optimization).

3.2 Theoretical formulation

To formulate the NMPC, in a theoretical sense, a Multiple Input - Multiple-Output (MIMO) nonlinear system is considered:

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

where: $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^{n_u}$ is the command input vector, and $y =\in \mathbb{R}^{n_y}$ is the output vector.

Differently to other formulations (like FBL and SMC) here a more general case is considered. Indeed, the system is a MIMO system, and the state equations are not written in affine form.

For formulation purposes, it is assumed that the state of the system is measured (in real applications if it isn't possible to measure all the states, then observers can be used to accomplish this task). The measurements are collected at each *sampling time* (T_s), so they are updated in real-time:

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

with: $k = 0, 1, 2, \ldots$

It is worth to note that $x(t_k)$ is the measurement of the state vector taken at t_k time, that is a discrete time.

3.2.1 Prediction

At each time $t = t_k$ the NMPC uses a model to make a prediction of the system behavior over a given time interval $[t, t + T_p]$ (so from the present time t to a future time $t + T_p$). This T_p is the *prediction horizon* and it must be always bigger (or at least equal) than the sampling time $T_p \geq T_s$.

The used model, that is in the form 3.1, is composed by the *state equation* (that is in differential form) and by the *output equation.* To make the prediction, the controller integrate the state equations, that describe the system, starting from the initial conditions given by the current state at time t. Then, the output y is obtained by considering the output equation. The following notation is used for the *predicted output*:

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

where: $\tau \in [t, t + T_p]$

$\hat{y}(\tau)$ is a function of the time τ, of the initial state $x(t)$, and also of the command signal $u(t:\tau)$. When the prediction is made, the initial state is the state at time $t = t_k$, this time is the starting time for the prediction that is used to define the initial conditions for the state vector. The notation used for the command input signal indicates that it is the signal applied from time t to time τ . So, $u(t:\tau)$ denotes the input signal in the interval $[t,\tau]$.

One important observation is that in the time interval $[t, t+T_p]$, $u(\tau)$ is an open-loop input in the sense that it doesn't depend on $x(\tau)$ (please note that $u(\tau)$ is a vector, while $u(t:\tau)$ is a signal). At the general time instant τ (inside the whole time interval), the predicted state $\hat{x}$ does not depend on the value of u at the same time. While, it depends on the $u(t:\tau)$ signal and on the initial conditions at time t. To explain better this concept the following picture 3.1 is presented:

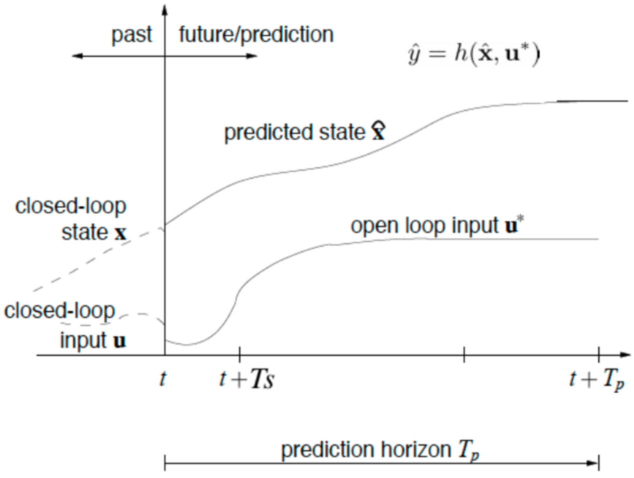

Figure 3.1: (x,t) and (u,t) plots

At time t the time interval $[t, t+T_p]$ is defined. The general input $u(t:\tau)$ is a signal in this interval. Starting from the model a given predicted state signal $\hat{x}$ is obtained (for simplicity the behavior of just one state variable is shown in 3.1) for a given input signal u. Clearly, if u changes in the time interval, also the behavior of the $\hat{x}$ changes, and in cascade also the behavior of $\hat{y}$ changes. So, anytime u changes also $\hat{x}$ and $\hat{y}$ signals change over the whole time interval, because they depend on the signal u.

The idea of Predictive Control is to search, thanks to an optimization algorithm, for an input signal such that the output will have the desired behavior. Among the various input signals that can be applied to the system, the optimal input signal that gives the optimal desired behavior of the output is denoted with $u^*(t:\tau)$.

At each sampling time, the NMPC researches the optimal input signal (applied in open-loop), such that the prediction

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

has the desired behavior for $\tau \in [t, t + T_p]$.

Differently to the MPC, where the optimal input is always global (because the optimization algorithm works with linear functions and can work only with *convex constraints*), in the NMPC case, the optimal input can be local. This will be discussed deeply later.

3.2.2 Optimization

In order to formalize the concept of desired behavior for the output, it is defined what is called *Objective Function* (or *Cost Function*):

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

Where: $\tilde{y_p}(\tau) \doteq r(\tau) - y\hat{(\tau)}$ is the *predicted tracking error*, in which $r(\tau) \in \mathbb{R}^{n_y}$ is the reference to track.

The first two norms inside the cost function are weighted vector norms and theirs integral are (square) signal norms. Indeed, if one signal is considered at a given time this can be seen as a vector. The last term is the predicted tracking error at the final time of the interval (that is a square weighted vector norm).

It is important to note that J is a function only of the whole signal $u(t : t + T_p)$, because the all terms inside the cost function depend only by $u(\tau)$, with τ that goes from t to $t + T_p$.

The optimization problem is to minimize the cost function w.r.t. the command input u. The signal $u^*(t : t + T_p)$ is chosen as the *minimizer* of objective function $J(u(t : t + T_p))$. Minimize J means minimizing the tracking error, the norm of the input signal (so the command activity), and also the tracking error at the end of the time interval. In this way, the desired behavior is obtained. The NMPC repeat the optimization at each time t_k (look 3.2) finding the best open-loop input signal for the whole interval $[t_k, t_k + T_p]$.

- The fist term inside J is the integral, in the whole time interval, of the predicted tracking error square weighted norm $\|\tilde{y_p}(\tau)\|_Q^2$. This means that this quantity is minimized not only at a given time, but along the complete time interval.
- The same considerations hold for the second term $\|u(\tau)\|_R^2$. This term allows to manage the trade-off between performance and command activity.
- The last term gives further importance to the final tracking error.

In order to obtain the desired behavior the designer chooses (by trial and error procedures) the coefficients of the weight matrices Q, R, and P.

Constraints

Once the optimization problem is defined, it needs to be determined not only the cost function, but also the constraints for this problem.

One important constraint is that $\hat{y}(\tau)$ must satisfy the model equations. Indeed, it is obtained from that equations by integration. $\hat{y}$ depends on $\hat{x}$, that depends on u, so $\hat{y}$ is obtained from u through the model equations. The minimization of J is thus subject to this constraint:

$$\dot{\hat{x}}(\tau) = f(\hat{x}(\tau), u(\tau)), \quad \hat{x}(t) = x(t), \quad \tau \in [t, t + T_p], \quad \hat{y}(\tau) = h(\hat{x}(\tau), u(\tau)) \tag{3.6}$$

Where the second one tells that the initial condition of the predicted state is the measured state at the initial time t.

Imagine to be at time t, the state is measured ad updated as initial condition. The NMPC integrates the differential equation from t to $t + T_p$ and find the signal $\hat{x(\tau)}$. Then, by using the output equation, also the value of $\hat{y}(\tau)$ is obtained and used to construct the predicted tracking error inside the cost function.

When the optimization problem is formulated a relevant constraint to be taken into account is that the differential and the output equations of the model describing the system must be satisfied. In other words, $\hat{y}(\tau)$ and $u(\tau)$ are not independent. These two variables must be consistent with the system dynamics.

In the optimization problem there may be other constraints. For example can be of interest to have constraints on the state, on the output, and on the input:

- $\hat{x}(\tau) \in X_c, \quad \hat{y}(\tau) \in Y_c, \quad \tau \in [t, t+T_p]$ (example: collision avoidance)
- $u(\tau) \in U_c, \quad \tau \in [t, t + T_p]$ (example: input saturation)

Where X_c, Y_c, and U_c are sets that provide the constraints of the variables (typically these sets are described by inequalities).

Feasibility

An optimization problem is *feasible* if there exist at least one solution satisfying all the constraints. Sometimes, when there are a lot of constraints it may be difficult to see if they are all feasible together. Testing the feasibility of the problem is significant in many situations.

Local Vs Global Minima

Only in simple situations it is possible to find analytical solution of the optimization problem: if there is a large number of variables the problem may be extremely difficult, in this situation an analytical solution is impossible to be found and the alternative is to proceed with a numerical solution.

There are many algorithms that can be used on this purpose. The working principle is almost the same for all of them. Thanks to an *iterative process* the algorithm start from an initial point x_0. Then, by using the gradient function (that gives the direction of maximum variation), it elaborates the points in the neighborhood of x_0 until a minimum is found. There are several stopping criteria, but the algorithm is able to find them when there is no more significant variation. In fact, it stops to work and provides the value of the minimum.

A fundamental problem of this kind of solution is that typically it is not possible to be sure that the found minimum is local or global. This depend on the starting point x_0. This issue does not depend on the employed algorithm, but it is a mathematical problem. Finding the solution of a multivariate non-linear function is difficult and usually the complexity of the problem grows exponentially with the dimension of the decision variable vector.

A particular class of functions, for which is always possible to find a global minimum, is the *class of convex functions*. For these functions every local minimum is global and if a suitable numerical algorithm is used it is possible to be sure that the found minimum is the global one. In these functions whatever is the starting point the algorithm always converges to the global minimum.

A function is convex if its levels curves define a convex set.

The convex set can be defined starting from a closed level curve. Taken any couple of points inside the curve if the linear connection between them remains inside the curve the set is convex otherwise it is a *Non-convex set.*

It is possible that some constraints are defined with non-convex sets. In this way, although the starting problem is convex it becomes non-convex.

Anyway, also in the case of non-convex functions with local minima it is possible to find a good solution for the optimization problem.

3.2.3 Receding Horizon Principle

The NMPC works in real-time and in closed-loop and also it can be work with varying constraint. This thanks to the *Receding Horizon (RH) Principle* that makes the controller flexible and able to adapt to many working conditions.

As said before, the result of the optimization problem is the optimal input signal $u^*(t : t + T_p)$, that minimize the cost function. This signal is applied in open-loop to the system, because it is computed by using only the information about the state at the time in which the optimization is made as initial conditions $x(t)$ (it doesn't use $x(\tau)$ with $\tau > t$).

The optimal input is calculated in the *whole time interval* $[t, t + T_p]$ and applied to the system, but after a sampling time the optimization is repeated without waiting for the whole time horizon T_p.

This repeating procedure (that is the RH principle) produces a closed-loop system that is able to increase precision, to reduce errors and disturbances, and to adapt to the varying scenario.

The NMPC calculates at each sampling time the optimal input and in this way, with the new information on the the state (taken always as initial conditions), it decides again what is the best input that solve the new optimization problem. In summary:

1. At a given time $t = t_k$ the optimal input signal is computed via optimization in the whole time interval;
2. The optimal input signal is applied in open-loop to the system;
3. The same operation is repeated for $t = t_{k+1}, t_{k+2}, \dots$, so after each sample time;
4. The controller in this way works in closed-loop and is able to react to any possible change;

3.3 Mathematical Formulation

In this section a general mathematical formulation of the NMPC is given. This will be the basis for the algorithm's design.

At each time step $t = t_k$ for $\tau \in [t, t + T_p]$ the NMPC solves the following optimization problem:

$$\begin{aligned} & u^*(t : t + T_p) = \arg\min_{u(\cdot)} \; J(u(t : t + T_p)) \\ & \text{Subject to the following constraints:} \\ & \dot{\hat{x}}(\tau) = f(\hat{x}(\tau), u(\tau)), \quad \hat{x}(t) = x(t) \\ & \hat{y}(\tau) = h(\hat{x}(\tau), u(\tau)) \\ & \hat{x}(\tau) \in X_c, \quad \hat{y}(\tau) \in Y_c, \quad u(\tau) \in U_c \\ & u(\tau) = u(t + T_c), \quad \tau \in [t + T_c, t + T_p] \end{aligned} \tag{3.7}$$

The time is denoted with t, while t_k indicates the time instant at which the measurement of the state is collected. As seen before, for each $t = t_k$ the optimization problem is solved, in this way the cost function J, subjected to some constraints, is minimized.

The first constraints tells that the model equations must be satisfied, then additional constraints can be on the state, on the output, and on the input.

The last constraint $u(\tau) = u(t+T_c)$ is used to reduce computational effort. Indeed, the command input is assumed to be constant for all the times that go from $t + T_c$ to $t + T_p$ with $\tau \in [t + T_c, t + T_p]$.

The T_c is called *control time* and $T_p \geq T_c \geq T_s \geq 0$. In this way from t to $t + T_c$ the controller optimize the input, then from $t + T_c$ onwards the input is constant.

3.3.1 Features of the Optimization Problem

The optimization problem is *non-convex*, in general. The main reason is that, in general, J is a non-convex function. This means that there can be several minima and the NMPC optimization algorithm cannot guarantee if the found minimum is global or not.

The other reason (for the non-convexity) is that the sets: X_c, Y_c, and U_c can be also non-convex.

In general, in the end the NMPC woks with a non-convex optimization problem.

Another feature of the optimization problem is that it must be solved *on-line* at each sampling time.

The key point in the NMPC is to have an efficient optimization algorithm in order to solve the problem in a time that is not too large. Moreover, for the non-convexity, the algorithm must be good enough to guarantee that the found minimum is quite good to accomplish the control task (local minimum value close to the global one).

3.3.2 Features of the Objective Function

The cost function J is a *functional* (a function of function), since it depends on the input signal u, that is defined in the prediction time interval.

An important point is that the decision variable of the optimization problem is a signal. A signal can be seen as a vector with an infinite number of elements and, as a consequence, the optimization is made w.r.t. an infinite number of decision variables. The idea to overcome this hard problem is to do a signal *parameterization.*

3.3.3 Input Parameterization

The input signal can be parameterized in such a way that the infinite dimensional optimization problem becomes a finite dimensional problem. Here two solutions are proposed:

(a) *Piece-wise constant input:*

$$u(\tau) = u_p(\tau) = c_i \qquad \text{for} \quad \begin{cases} \tau \in (t + (i-1)T_s, t + iT_s] \\ i = 1, \dots, m \equiv \frac{T_c}{T_s} \end{cases} \tag{3.8}$$

The input signal is assumed to be constant and equal to c_i for each sub-interval indicated in the curly bracket. In this way, the interval $[t, t + T_p]$ is partitioned in sub-intervals and it is assumed that in each of them the input is constant.

(b) *Polynomial parameterization:*

$$u(\tau) = u_p(\tau) = \sum_{i=1}^{m} c_i(\tau - t)^{i-1} \tag{3.9}$$

The input is written as a linear combination of polynomials. The result is a polynomial representation of the input. The c_i coefficients are polynomial coefficients.

When $i = 1$, c_1 is constant, when $i = 2$, $c_2(\tau - t)$ is a linear term, and so on. By selecting a given m (must be an integer number) the parameterization is stopped and in this way the optimization is made not over the complete signal (that is of infinite dimension), but it is performed only w.r.t. these coefficients c_i.

In both parameterizations the optimization is performed w.r.t. the finite dimension matrix $c = [c_1, \ldots, c_m] \in \mathbb{R}^{n_u \times m}$.

The designer has to choose m. He has to decide when stopping the parameterization. Typically no large values of m are needed to have good results. A strategy is to begin with $m = 1$ and then growing this number if it is necessary.

With $m = 1$ the command input is constant for the whole prediction interval.

In the first case where $m = T_c/T_s$ the designer can select the near integer number to this ratio, or (better) can select T_c as a multiple of T_s.

3.4 MPC Algorithm and Design

In this section a general NMPC algorithm with a possible control scheme is analyzed, in the first part. In the second one, the design parameters are treated in order to have a global point of view on this control approach.

3.4.1 MPC Algorithm

The general working principle of the NMPC algorithm can be divided into two consequently steps:

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

$$\begin{aligned}
&c^* = \arg \min_{c \in \mathbb{R}^{n_u \times m}} \; J(c) \\
&\text{Subject to:} \\
&\dot{\hat{x}}(\tau) = f(\hat{x}(\tau), u(\tau)), \quad \hat{x}(t) = x(t) \\
&\hat{y}(\tau) = h(\hat{x}(\tau), u(\tau)) \\
&u(\tau) = u_p(\tau) \\
&\hat{x}(\tau) \in X_c, \quad \hat{y}(\tau) \in Y_c, \quad u(\tau) \in U_c \\
&u(\tau) = u(t + T_c), \quad \tau \in [t + T_c, t + T_p]
\end{aligned} \tag{3.10}$$

Where c is the parameterized input.

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

2. Repeat step 1 for $t = t_{k+1}, t_{k+2}, \ldots$

Control Scheme

In the following picture the general control scheme made with MATLAB is showed:

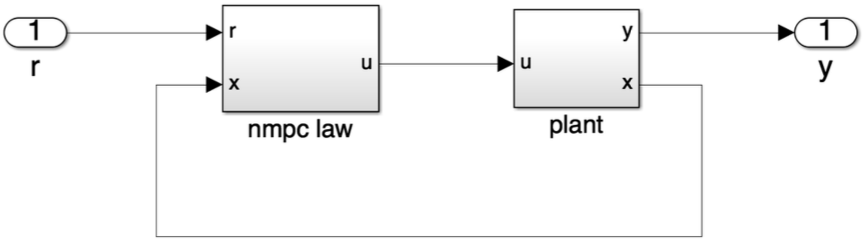

Figure 3.2: Control Scheme

- In the *Plant* block there are the equations that describe the system in the form:
$$\dot{x} = f(x,u), \quad y = h(x,u)$$
- In the *NMPC law* block there is the algorithm seen previously. It contains a model of the plant that it is used for prediction. The prediction model is in the form:
$$\dot{\hat{x}} = \hat{f}(\hat{x},u), \quad \hat{y} = \hat{h}(\hat{x},u)$$
If the model uses the true functions: $\hat{f} = f$ and $\hat{h} = h$
- r and y are respectively the reference signal and the output signal.

3.4.2 MPC Design

The design of a good NMPC algorithm stand on the specific application in which it must be used. The proper parameters are selected by the designer, almost the time, by trial and error procedure. In the following, some indications about the choice of the fundamental parameters is given.

Choice of Parameters

1. *Sampling Time T_s:* In many situations, it is given by the specific application in which the NMPC has to be employed. Otherwise, it is chosen by trial and error procedure, considering that it should be: sufficiently small to deal with plant dynamics (Nyquist-Shannon sampling theorem), and not too small to avoid numerical problems and slow computation.

2. *Prediction Horizon T_p:* It can be chosen through trial and error procedure by considering that: a large T_p increases the robustness, increases the closed-loop stability proprieties, and a too large T_p may reduce the short-time tracking accuracy.

3. *Control Time T_c:* Small values of T_c reduce the computational time, without often reducing the performance. In many application T_c is set equal to T_s (this means $m = 1$), in that applications satisfactory performance can be obtained.

4. *m:* In many cases a low number of parameters is enough to obtain a satisfactory control performance. Choosing $m = 1$ works in many situations.

Choice of Weighted Matrices

As seen in the optimization problem there are three weighted matrices to be chosen: Q, P, and R. Where the first is related to the predicted tracking error $\hat{y_p}$ in the whole time interval, the second one is also related on the predicted tracking error, but this time on the value at the final time of the prediction, and the last one is related to the input signal.

It is always convenient to choose these matrices as diagonal. The idea to choose their diagonal coefficients is to start with a reasonable choice and then by using trial and error procedure adjust the values in order to optimize the behavior of the controller.

A possible initial choice can be:

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

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

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

It is worth to note that the matrices are related to the signal norms and that the norms are related to the energy. Changing the diagonal coefficients means changing the energy of the associated signals. If the weight of a coefficient is increased the energy associated with that signal will be decreased. Increasing the weight means giving more importance to the terms that contain these weights. The algorithm minimize these variables more than the others because the weight is large.

Increasing Q_{ii} and P_{ii} reduce oscillations and converging time. Increasing R_{ii} reducing command effort and fuel consumption.

3.5 MPC Properties

In the following section *Stability* and *Robustness* properties are treated.

Stability

A typical approach to enforce closed-loop stability in the NMPC controller is to add what is called *terminal region constraint* in the optimization problem:

$$\hat{x}(t + T_p) \in \Omega \quad \text{with: } \Omega \in \mathbb{R}^n \tag{3.11}$$

Basically, this constraint enforce the final predicted state $\hat{x}$ (taken at time $t + T_p$) to belong to a given region Ω that is a bounded, closed, and connected set.

If this constraint is added to the optimization problem, then the stability of the closed-loop is guaranteed. Typically, it is not required, because if the designer choose correctly the parameters then what is obtained is a controller that stabilizes by itself the closed-loop. Hence, only in particular situations this can be useful.

Under some particular conditions, it is possible to give a theoretical proof of stability and also convergence for the NMPC, but it is more difficult than other control methods. The terminal region approach is more general and flexible but it is difficult to provide theoretical results. In any case these kind of results can be found in literature and they are not really useful from a practical point of view.

Robustness

In many applications the exact model of the plant is seldom known. What is used as model, for control design purposes, is an approximated one. Instead of the true model f and h the controller uses $\hat{f}$ and $\hat{h}$. Several robust version of the NMPC have been proposed to guarantee stability and/or performance when the model is not exactly known. Assuming that $\|f - \hat{f}\|$ is bounded in some norm the following NMPC robust methods can be used:

- *min-max NMPC*
- *H_∞-like NMPC*
- *parametrized controller NMPC*

Can be shown that if the difference between the system and the model functions is bounded in some norm, then stability or other kind of performance can be guarantee.

The first two techniques may be too conservative. While all the three, typically, require an high computational effort. Thus, they cannot be applied to problems where a small T_s is required.

In any case, thanks to the RH strategy, standard NMPC is in general characterized by good robustness properties.

3.6 Advantages and Drawbacks

In this section is given a list of advantages and drawbacks of the NMPC.

- **Advantages:**

  1. *General and flexible approach:* The formulation is given starting from a complex MIMO system, without considering the affine form (needed in other approaches);
  2. *Intuitive formulation:* That is based on optimal concepts;
  3. *Constraints and input saturation accounted for:* These can be also time-varying;
  4. *Efficient management of performance/input activity trade-off*;
  5. *Optimal trajectories* (over a finite time interval);
  6. *Unified computation of optimal trajectory and control law:* The controller is able to do both these operations together. Indeed, it not only gives the control law, but also designs a some optimal trajectory;

- **Drawbacks:**

  1. *High on-line computational cost:* At each sampling time the NMPC has to solve the optimization problem. In order to work properly, it is needed to have an hardware and enough amount of energy that are more expensive than other controllers;
  2. *Possible local minima in the optimization problem*;
  3. *Problems in the case of unstable zero-dynamics* (like all methods);

3.7 Comparison between NMPC and linear MPC

In general the MPC is easier than its nonlinear version and it requires a lower computational effort. If the system to be controlled is Linear Time Invariant (LTI), with convex constraints, the MPC gives an optimal solution that is always global. While, with the NMPC only local optima can be found in general.

The MPC can be used to control nonlinear systems only if a linearization of the plant is made around a given working point along the trajectory. This, in case of tracking problems, can be very complicated and obviously, the computational effort increases.

The MPC requires that all the constraints are convex sets. If they are not a convexification can be made, but also this is a quite complicated.

On the contrary, the NMPC can be used directly to control nonlinear plants and the used constraints can be simply written by inequalities.

Chapter 4

Space Mission and Control Algorithm

A very important sector in space engineering is called GNC. These are three operations that must be execute in space missions. The goal is to control the motion of the S/C in order to accomplish given tasks of a mission. *Guidance* consists in planning an optimal trajectory that must be tracked by the S/C. *Navigation* consists in measuring and/or estimating all the state variables of the S/C that are important for the control task. *Control* is finalized to bring the S/C state close to the planned trajectory.

In this chapter, first the maneuvers needed for the mission are treated. Then, the specific control algorithm is designed in order to accomplish the various tasks.

4.1 Space Mission

As seen in the previous chapter (2), there are several reference frames that can be used to specify the position of an object in the space. Depending on the specific mission/maneuver is convenient to choose properly these frames. Since, another important think (always related to the used frame) is that the equations of motion are different.

In the years, humanity have launched a lot of satellites, that orbit around the Earth. Lot of them, especially the oldest ones, become obsolete, or worst they broke after collisions or because technical damages. In these last cases, there be an unpredicted waste of money.

In this way, these satellites become debris an they (or parts of them) can damage others. Other times, since in the space near to Earth surface the atmosphere is not so dense to burn little rocks, they can be attracted by the gravity and remain in orbit. In both cases, a *collision orbit* with other satellites can be very dangerous.

By considering this problem, the selected space mission has to solve in an autonomous way the problem of debris around the Earth, that will become very serious because the increasing number of satellites (currently 2787 satellites orbiting Earth [8]).

4.1.1 Mission Maneuvers

This *debris removal mission* is composed by several orbital maneuver. By considering the starting position of the debris and of the S/C, the following maneuvers have to be done in order to complete that mission:

1. *Rendezvous maneuver:* the S/C approaches the debris;
2. *Docking maneuver:* the S/C gets in contact with the debris;
3. *Estimation maneuver:* the unknown mass of the debris is estimated;
4. *Second Rendezvous maneuver:* the system (composed by S/C and debris) goes near to a specific starting point for the other maneuver. This maneuver is not necessary, but may be convenient to have a known starting point, or also to start the orbit-change maneuver from an *empty* orbit;
5. *Orbit-Change maneuver:* the S/C brings the debris in an another orbit to avoid collisions problem.

These maneuver are treated deeply in the following.

Rendezvous Maneuver

In this maneuver two objects (most of the time S/Cs) approach to a very close distance. The S/C, that has to move in order to enter in contact with the second object, is called *Chaser*. While, the object that is moving in its own orbit (it has not propulsion) is called *Target* [10].

The Chaser moves by considering a specific final point near to the Target. The side on which the two bodies have to be, at the final, is very important. In order to considering this think and to avoid collision problems between them, specific constraints on the state (or on the output) must be accounted for.

Another important constraint is on the input. Since this maneuver is precision, too high values of the input signal may cause wrong approach trajectories, or worst collisions. For this reason, also a saturation constraint on the input must be considered.

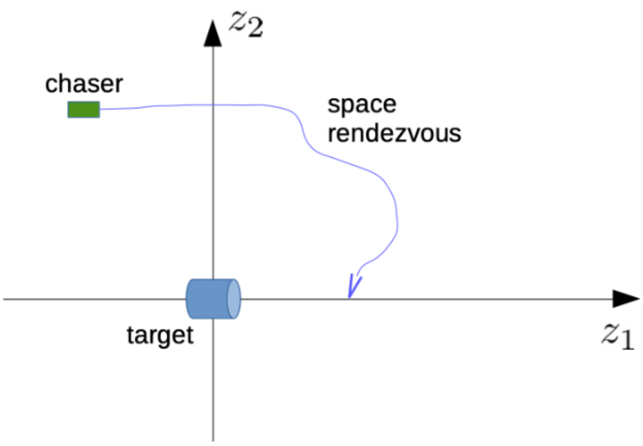

Figure 4.1: Rendezvous 2D trajectory

The equations used to describe the relative motion between the two objects are the HCW equations (see section 2.1.8). They describe the dynamics of the Chaser in a neighborhood of the Target:

$$\begin{aligned} \ddot{z}_1 &= 3\omega^2 z_1 + 2\omega\dot{z}_2 + \frac{u_1}{m_1} \\ \ddot{z}_2 &= -2\omega\dot{z}_1 + \frac{u_2}{m_1} \\ \ddot{z}_3 &= -\omega^2 z_3 + \frac{u_3}{m_1} \end{aligned} \tag{4.1}$$

Where z_i are the coordinates of the Chaser in a Target-centered reference frame, u_i are the components of the input along the reference axes (they represent the components of force given by the thrusters), m_1 is the Chaser mass, ω is the angular velocity of the Target (so of the relative frame) w.r.t. an inertial frame (note that in a neighborhood of the target the angular velocities of the two bodies can be considered equal $\omega_1 \approx \omega_2 \approx \omega$).

An important observation is that if the variation of the Chaser mass (due to the fuel consumption) is not considered, these equation are LTI. Since, ω and m_1 are constant parameters.

However, the control problem becomes nonlinear and non-convex because the constraints.

If the LTI equations are rewritten in matrix form, it can be seen that the equations are describing an unstable system. Indeed, the state matrix A contains a double null eigenvalue.

Docking Maneuver

Similarly to the Rendezvous, the Docking is a precision maneuver. It consists in a rigid and stable contact with two objects. After this maneuver the two bodies can be seen and treated as an unique object.

Figure 4.2: Demo-1 Crew Dragon Docking with the ISS

This maneuver always starts after the Rendezvous, but it has more stringent constraints. While the Chaser approaches the Target in the docking point, the maneuver space is reducing. This constraint is implemented by means of a conic constraint on the state (or also in this case, as output constraint). The hardness of the constraint w.r.t. the Rendezvous is also on the input: the relative velocity between the two bodies must converge to zero in order to avoid collisions and damages.

Also in this case the equations that describe the motion are the HCW equations.

Estimation Maneuver

This maneuver is execute in this specific mission in order to estimate the mass of the debris. In [21],[22], and [23] can be seen that an higher estimate precision is obtained by moving the system randomly in a given region. More the behavior of the system diverges from the expected one, more the estimation will be efficient.

By means of some estimation algorithm the mass is estimated and then the maneuver is completed.

Orbit-Change Maneuver

Differently from the precision maneuvers, that use the HCW equations, this kind of maneuvers are done by considering the inertial GE frame. In this way, the used equations are the ones given by the 2B problem:

$$\dot{r} = v, \quad \dot{v} = -\mu \frac{r}{|r|^3} + \frac{1}{m} u, \tag{4.2}$$

This maneuver consists to change the orbit of the debris. There are different types of maneuvers of this kind. The principal ones can be execute by considering some orbit elements. The changing can be: on the *eccentricity* e, on the *inclination* i angle, on the *semi-major axis* a, or on a combination of them.

Considering this specific mission, it is propose an orbit-change maneuver by increasing the semi-major axis. The S/C brings the debris faraway, in another orbit that is empty, in order to avoid collision problems with other satellites. Another possibility is to bring the debris in a very low orbit, near to the Earth surface. By the effect of the *drag force*, the debris slows down its velocity (consequently the semi-major axis is reduced) and then it falls and burns in the atmosphere.

4.2 Control Algorithm

Once the mission and the various maneuvers are specified, an efficient control algorithm is needed in order to accomplish the tasks.

In this work, the control method used is the NMPC in an adaptive way. This is chosen for several reasons:

- The modified equations of the motion appear in a form that is not affine;
- It can be used to account for static and varying constraints;
- It is flexible and robust to system variation. In this way, an optimal trajectory can be almost always found;
- The adaptability is in the estimation maneuver. Thanks to a specific estimation algorithm, the estimated value of the mass is update on-line to the controller. The controller is able to adjust its behaviour at each step time and to find a different value of the mass that converge, towards infinity, to the real value. In this way, the controller is able to bring the behavior of the system near to the planned one.

In the following the control algorithm and setting are specified for each maneuver. It will see that a scheduling is needed in order to understand when a specific maneuver has to start. But first, the system model and the control model are treated.

4.2.1 System Model

The equations describing the motion of the system are not the same of the real system: they are always an approximation. Indeed, when a model is designed there can be:

- *Dynamics approximations:* They don't consider (or consider in an approximate way) some aspect that changes not considerably the dynamics of the system;

- *Parameter approximations:* They are based on the fact that is not possible to have an exact value for a parameter. Indeed, there will be a tolerance range in which the real value is present.

In the space field, there are a lot of disturbances/perturbations that can affect the dynamics of the system. Examples of them are: drag force, different Earth's mass distribution that corresponds to a different gravity potential, effects due to the gravity of other celestial bodies (like the Moon, and the Sun), solar/cosmic radiations, thermal radiations, and so on.

To construct the system model is considered only the biggest perturbation effect: the *Drag Force* due to residual of atmosphere in LEO. Indeed, the contribution of the *Atmospheric Drag* is significant in the motion of a body on a LEO. The following approximate formulation is considered to taking into account this effect:

$$F_d = -\frac{1}{2}\rho \; C_D S |v_{rel}| v_{rel} \tag{4.3}$$

Where ρ is the local atmospheric density, C_D is the drag coefficient, S is the S/C area projected along the direction of motion, v_{rel} is the relative velocity of the S/C w.r.t. the atmosphere.

Assuming a negligible atmospheric velocity: $v_{rel} = v$. F_d has the same direction of v_{rel}, while its intensity is proportional to the square of v_{rel}. C_D is a coefficient that depends on the shape of the S/C. A possible model for ρ is the following:

$$\rho(r) = \rho_0 e^{-\frac{r-r_0}{H}} \tag{4.4}$$

In this model the density is a function of r (distance from the Earth's CoM, that corresponds to the altitude), ρ_0 and r_0 are reference density and height, while H is a scale height coefficient.

An important observation is that this density is described by a negative exponential model. The density decrease exponentially with the altitude $r - r_0$.

This model is very simple, but for simulation purposes it is good enough. The other perturbations are neglected, because some of them don't have a quite strong effect, and they are not so easy to model. However, by using a random noise generator, it is possible to taking into account also them in an approximate way.

The equations that describe the motion of the S/C are 2B equations (see section 2.1.3). In this way the *S/C orbital dynamics equations* are:

$$\begin{aligned}
&\dot{r} = v, \quad \dot{v} = -\mu \frac{r}{|r|^3} + \frac{1}{m}(F_d + d + u), \\
&F_d = -\frac{1}{2}\rho \; C_D S |v| v, \quad \rho = \rho_0 e^{-\frac{r-r_0}{H}} \\
&u = 0 \quad \text{if } m \leq m_b \\
&\dot{m} = \begin{cases} 0 & \text{if } u = 0 \\ -\frac{|u|}{v_e} & \text{if } u \neq 0 \end{cases}
\end{aligned} \tag{4.5}$$

Where d is a disturbance that takes into account all the small perturbation effects, m_b is the S/C mass without the fuel, m is the total mass considering also the fuel, $\dot{m}$ is the mass variation due to the fuel consumption and it is related to the command input u by the Tsiolkovsky rocket equation, v_e is the *engine exhaust velocity.*

By considering the components of r and v along the reference frame chosen, the state vector can be written as:

$$x = (x_1, x_2, x_3, x_4, x_5, x_6, x_7) = (r_1, r_2, r_3, v_1, v_2, v_3, m)$$

In this way it is possible to write the state equations. The subscripts specify the components along the GE frame.

The equations that describe the motion of the debris are the FR2B equation (see section 2.1.3):

$$\dot{r} = v, \quad \dot{v} = -\mu \frac{r}{|r|^3} \tag{4.6}$$

Also in this case by considering $x = (x_1, x_2, x_3, x_4, x_5, x_6) = (r_1, r_2, r_3, v_1, v_2, v_3)$ the equations can be rewritten in state form.

In the debris orbit, it is not considered any perturbation. This is done because two principal reasons:

1. If it is considered the atmospheric drag, it can be seen that it is proportional to S. In the debris case this is assumed to be more lower than the S/C case;

2. It is supposed that the first two maneuvers (Rendezvous and Docking) happen too quick to appreciate the effect of small perturbations on the debris orbit.

Inside the plant there are both the equations that describe the motion of the S/C and of the debris. The controller uses the information given by the solutions of these equations in order to accomplish the mission tasks.

The following simulink scheme is used to describe the system plant:

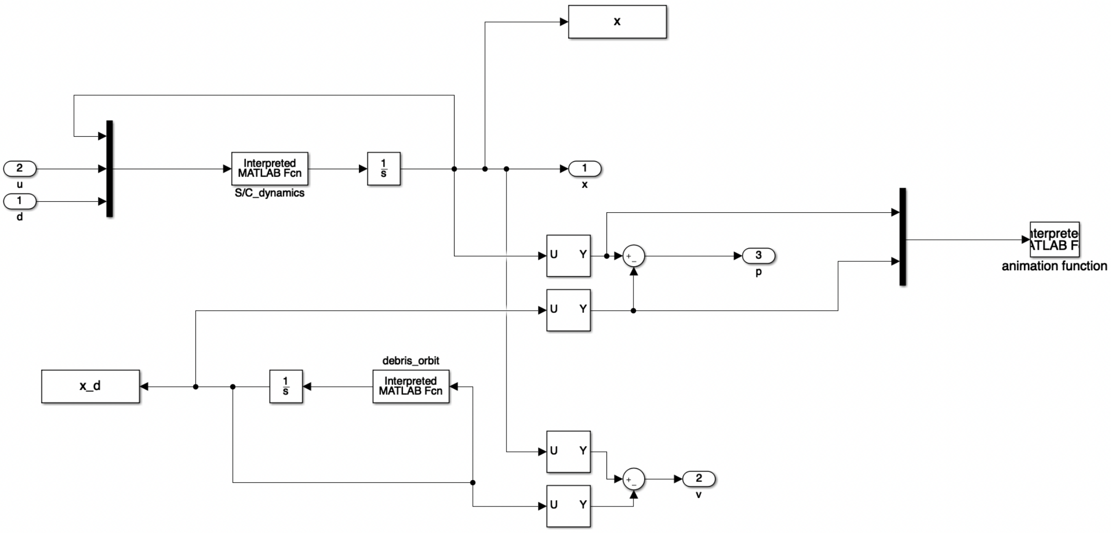

Figure 4.3: System plant scheme

Where inside *S/C dynamics* and *debris orbit* blocks there are the the previous equations, *animation function* is a function used to plot the two trajectories, while, the use of the outputs p and v will be explained later.

However, it is worth to note that the system model is not the same used inside the NMPC.

4.2.2 Prediction Models

Depending on the specific maneuver, that has to be done, the model used by the NMPC changes.

As seen in the previous chapter (see subsection 3.4.1) the controller uses a prediction model that most of the times is not the same of the plant. Indeed, the behavior of the system is not exactly known and, for this reason, the functions that describe the system behavior are approximated.

The precision maneuvers required a model that takes into account the relative motion between the S/C and the debris. For this purpose the HCW equations are used. While, the model used for the Orbit-Change maneuver is the one given by the 2B equations. In this work the 2B equations are also used for describing the model in the estimation maneuver, but this is a choice of the designer.

The model based on the HCW equation is:

$$\begin{aligned}
&\dot{x}_1 = x_4, \quad \dot{x}_2 = x_5, \quad \dot{x}_3 = x_6\\
&\dot{x}_4 = 3\omega^2 x_1 + 2\omega x_5 + \frac{u_1}{x_7}\\
&\dot{x}_5 = -2\omega x_4 + \frac{u_2}{x_7}\\
&\dot{x}_6 = -\omega^2 x_3 + \frac{u_3}{x_7}\\
&\dot{x}_7 = 0\\
&h = x
\end{aligned} \tag{4.7}$$

Where x is the state: $x = (x_1, x_2, x_3, x_4, x_5, x_6, x_7) = (z_1, z_2, z_3, \dot{z}_1, \dot{z}_2, \dot{z}_3, m)$. While, the function h is used to define the output equation. In this case, the output coincide with the state.

In this modified version the total mass m is supposed to remain constant in the prediction interval. Indeed, its derivative is equal to zero. It will see that after the estimation maneuver, for finding the debris mass, the estimated value will enter in this model.

The model based on the 2B equation is:

$$\begin{aligned}
&\dot{x}_1 = x_4, \quad \dot{x}_2 = x_5, \quad \dot{x}_3 = x_6\\
&\dot{x}_4 = -\mu\frac{x_1}{r^3} + \frac{u_1}{x_7 + m_{2\ est}}\\
&\dot{x}_5 = -\mu\frac{x_2}{r^3} + \frac{u_2}{x_7 + m_{2\ est}}\\
&\dot{x}_6 = -\mu\frac{x_3}{r^3} + \frac{u_3}{x_7 + m_{2\ est}}\\
&\dot{x}_7 = 0\\
&h = x
\end{aligned} \tag{4.8}$$

Where r^3 is the euclidean norm of the first three state components (it correspond to cube of the distance from the CoM of the Earth), $m_{2\ est}$ is the estimated value of the debris mass.

This model, as said before, is used in the last maneuver once the mass is estimated. It will see later that a similar model is considered in the estimation process by adding as state the estimate value of the debris mass.

4.2.3 General Settings

In order to accomplish the various mission maneuvers a schedule method is needed. The controller is always the same, but what is changing are: the used setting parameters (like x_0, u_0, T_p, T_s, Q and so on), the constraints (on the input and on the state), the final points given by the reference values, and the approximate model used by the controller.

To obtain an algorithm that changes its setting is considered a *for-cycle*, that uses the time as variable. In this way, the total simulation time (that coincides with the total mission time) is divided in small period. For each of them, thanks to specific flags and conditions, it is possible to run only the part related to a given maneuver.

A specific maneuver starts when the previous one is completed. By considering the reference signals, that give the relative final points in the first two maneuvers and in the third one, it is possible to run the algorithm until a certain precision is achieved. This precision has the meaning of a tolerance on the S/C final point. In other words, for each condition, the algorithm run a specific part until the specific condition remains satisfied. The conditions for these maneuvers are in the following form:

$$\begin{gathered} |(|r_{rel}| - r_0)| - tol_1 \geq 0 \\ |(|v_{rel}| - v_0)| - tol_2 \geq 0 \end{gathered} \tag{4.9}$$

Where r_{rel} and v_{rel} are the relative position and velocity of the S/C seen from the relative LVLH frame centered with the debris CoM, r_0 and v_0 are the position and the velocity given the reference signal, while tol_1 and tol_2 are two tolerances.

When both these two conditions are no more satisfied, the algorithm sets the parameters needed for the next maneuver that will start.

To obtain a good algorithm, some flags are inserted to understand when a maneuver is completed. Thanks to them, the algorithm automatically avoids to repeat paths (so to restart maneuvers already done) inside the cycle.

Another important consideration is on the unit measures that, obviously must be consistent in the whole algorithm.

To initialize the algorithm the following parameters are considered:

- *Gravitational Parameter:* $\mu = 0.3986e15$ $[\frac{m^3}{s^2}]$;
- *Equatorial Radius:* $r_e = 6.38e6$ $[m]$;
- *Partial Simulation Time:* $t_{step} = 50$ $[s]$;
- *Simulation Time:* $t_{total} = 8e3$ $[s]$;
- *S/C Mass:* $m_1 = 1000$ $[kg]$;
- *Fuel Mass:* $m_{fuel} = 10e3$ $[kg]$;
- *Debris Mass:* $m_2 = 550$ $[kg]$;
- *Engine Exhaust Velocity:* $v_e = 4.4e3$ $[m/s]$;
- *Debris Radius:* $a = r_e + 400e3$ $[m]$;

- *Debris Angular velocity:* $\omega = \sqrt{\frac{\mu}{a^3}}\ [\frac{rad}{s}]$;
- *Debris Starting Conditions:* $x_{d0} = [a; 0; 0; 0; \sqrt{\frac{\mu}{a}}; 0; m_2]$;
- *S/C Starting Conditions:* $x_0 = [200 + a; 0; 0; 0; \sqrt{\frac{\mu}{200+a}}; 0; m_1 + m_{fuel}]$;
- *Input Initial Conditions:* $u_0 = [0; 0; 0]\ [N]$;

It is worth to note that both the S/C and the debris at the initial time are at the *perigee* and in an equatorial circular orbit ($|e| = 0$, $\cos i = 1$). Indeed, the initial conditions are written w.r.t. the GE frame.

4.2.4 Maneuvers Settings

In this section are analyzed the various maneuvers settings. It will be seen that, depending on the specific task, the controller adapts itself to bring the system in the optimal (or almost optimal) trajectory needed.

At each partial simulation time, the controller reads the output values (given by simulation) and on the basis of them, and on the basis of the imposed previous conditions and flags, it understand what is the maneuver that has to be executed.

First Rendezvous Maneuver

The first maneuver is the Rendezvous. If the S/C is too far from the debris may be useful to have another maneuver (of the Orbit-Change kind) before this one, in order to bring the S/C near to the debris (in a field where the HCW equations are valid).

In this work this maneuver is already considered. For this reason the first one of interest is the Rendezvous.

In order to adapt the two models (system and control models), that work with two different reference frames, a function that calculates the components of the relative frame is needed. The axes of the LVLH frame are constructed by considering:

- The debris CoM as origin (O');
- The *Local-Vertical* axis is defined along the direction $\bar{OO'}$ on the orbit plane (where O is the Earth CoM);
- The *Local-Horizontal* axis is perpendicular to the local vertical, it is on the orbit plane, and the sign is concordant with the orbital velocity;
- The *Orbit Pole* axis is given by the vector product of the first two axes and it is perpendicular to the orbit plane;

On the basis of these considerations, it is possible to write the unit vectors of the three axes and to find the relation between the two frames (by means of rotation matrix):

$$\begin{aligned}
e_x &= \frac{r_0}{\|r_0\|} \\
e_z &= \frac{r_0 \times v_0}{\|r_0 \times v_0\|} \\
e_y &= e_z \times e_x \\
R &= [e_x\ e_y\ e_z]^T
\end{aligned} \tag{4.10}$$

Where r_0 and v_0 are respectively the position vector and the velocity vector of the debris seen in the GE frame.

In this way, by considering the differences p (for the position vectors) and v (for the velocity vectors) between the S/C and the of the debris, it is possible to write the S/C components seen in the LVLH frame, and to write the components of the command input vector in the GE frame:

$$\begin{aligned}
&p = x(1:3) - x_d(1:3), \quad v = x(4:6) - x_d(4:6), \\
&x_{rel}(1:3) = Rp, \quad x_{rel}(4:6) = Rv + x_{rel}(1:3) \times \omega, \\
&u = R' u_{rel}
\end{aligned} \tag{4.11}$$

Where x is the state vector of the S/C, x_d is the state vector of the debris, ω is the angular velocity of the debris, x_{rel} is the state of the S/C in the LVLH frame.

It is worth to note that R is the DCM. It is an orthonormal matrix: $R' = R^T$.

Considering a final position point for the S/C, near to the debris, a constant reference signal is need. It is specified on the LVLH frame:

$$r_1 = [20; 0; 0; 0; 0; 0]$$

Where the first three components are the reference values for the relative position, while the last three are the references values for the relative velocity.

The algorithm conditions for this maneuver are:

$$\begin{aligned}
&tol = tol_1 = tol_2 \\
&|(\|x_{rel}(end, 1:3)^T\| - \|r_1(1:3)\|)| - tol \geq 0 \\
&|(\|x_{rel}(end, 4:6)^T\| - \|r_1(4:6)\|)| - tol \geq 0
\end{aligned} \tag{4.12}$$

By means of simulation, the algorithm collect at each partial simulation time the values of the relative state x_{rel} (it is composed by N samples with $N = (t_{step}/t_{sampling})$). The last collected values are used to understand if the final point is reached with a given tolerance. If this point is not reached, the algorithm rerun the piece of code relative to this maneuver until the previous conditions are no more satisfied.

The NMPC uses an inequality that gives to it the constraints on the state (see section 3.3). In this maneuver a collision path with the debris must be avoid. For this reason a spherical constraint debris-centered, with radius equal to 10, is considered. It is important to note that if it is seen from the inertial GE frame, it is a varying constraint, because it moves with the debris. Instead, considering the LVLH frame the same constraint becomes static. The mathematical formulation is the following:

$$F = 10 - \|x(1:3,:)\| \tag{4.13}$$

Where F is interpreted by the NMPC as $F \leq 0$. The norm on the S/C position (indices from 1 to 3) is a *L2* signal norm.

It can be seen that the signal, that contains the values of the components of a general vector at each sampling time, if it is written as a vector, resulting as output from simulation (see equation 4.12 for example), it is transpose w.r.t. the same signal used inside the matlab functions for simulation purposes (see equation 4.13 for example).

The following NMPC settings are considered for this maneuver (they were found by means of trial and error procedure as seen in the section 3.4.2):

- *Model Functions:* It is in the same form of the model seen in 4.7. Since, it is written as a matlab function (then used inside simulink), this is rewritten as follows:

$$\begin{aligned}
&\dot{x}(1:3,:) = x(4:6,:)\\
&\dot{x}(4,:) = 3\omega^2 x(1,:) + 2\omega x(5,:) + \frac{u(1,:)}{x(7,:)}\\
&\dot{x}(5,:) = -2\omega x(4,:) + \frac{u(2,:)}{x(7,:)}\\
&\dot{x}(6,:) = -\omega^2 x(3,:) + \frac{u(3,:)}{x(7,:)}\\
&\dot{x}(7,:) = 0\\
&y = x(1:6,:)
\end{aligned} \tag{4.14}$$

- *State Constraints:* It is the function seen previously;
- *Sampling Time:* $T_s = 1$ $[s]$;
- *Prediction Time:* $T_p = 200$ $[s]$;
- *System Order:* $n = 7$;
- *Q-matrix:* $Q = zeros(6)$;

- *P-matrix:* $P = 10\ eye(6)$;
- *R-matrix:* $R = 0.1\ ones(3, 1)$;
- *Input Saturation Constraints:* $|u| \leq 10e3\ ones(3, 1)\ [N]$;
- *Used Flags:* $j = 0$, $g = 0$, and $q = 0$. They are used as adding conditions, as said before, to avoid wrong repeating paths inside the for-cycle;

The used simulink control scheme for this maneuver is the following:

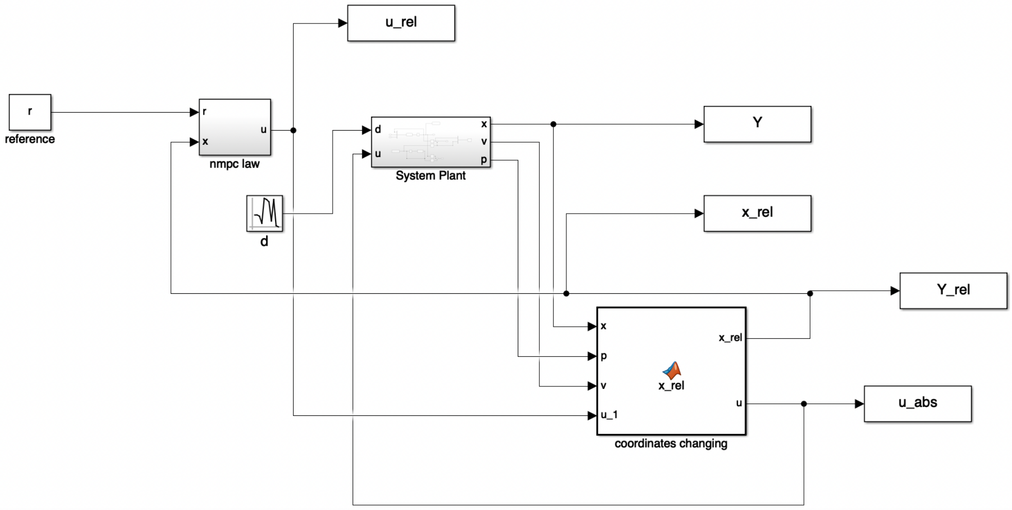

Figure 4.4: First Rendezvous maneuver control scheme

Where inside the *System Plant* block there are the system equation (written in the GE frame), inside the *NMPC law* block there is the control law that uses the previous settings, while inside the *Coordinate changing* block there are the equation needed to change the coordinate from a frame to the other, while the reference signal is: $r = r_1$.

Docking Maneuver

The Docking maneuver takes place when the Rendezvous is completed. The controller, also in this case, uses as model the HCW equations. The only things that change are the setting used, the reference signal, and the constraint function. Since this maneuver is more precise than the first one, the controller must have more stringent constraints on the state and on the input. In order to give more importance to the final reached point, higher coefficients of the weight matrix P are chosen (as seen in 3.4.2 this matrix is related to the final predicted error, if P coefficients increase the algorithm takes care to minimize more these variables than others).

The final point that the S/C has to reach is the following, given by the constant reference signal:

$$r_2 = [0; 0; 0; 0; 0; 0]$$

The algorithm condition are in the same form of the ones seen in 4.12. The only thing that changes is the reference that in this case is r_2:

$$\begin{aligned} & tol = tol_1 = tol_2 \\ & |(\|x_{rel}(end, 1:3)^T\| - \|r_2(1:3)\|)| - tol \geq 0 \\ & |(\|x_{rel}(end, 4:6)^T\| - \|r_2(4:6)\|)| - tol \geq 0 \end{aligned} \tag{4.15}$$

The constraint function in this case has the shape of a cone. Indeed, while the S/C approaches the debris the space in which it can move is reducing. The considered function is a cone with vertex coincident with the debris CoM, height equal to the first component of the final point given by the Rendezvous maneuver and equal to the base radius. In other words, at the initial point, the height of cone is coincident with the distances between the S/C and the debris, the radius of the base circumference is equal to that distance. While the S/C approaches to the debris, the distance is reducing (hence the height) as well as the base radius (that remains equal to the height in the considered point). The mathematical formulation is the following (remember that $F \leq 0$ as seen before):

$$\begin{aligned} & \text{if} \quad 0 <= x(1,:) <= r_1(1) + tol \\ & \qquad F = \|x(2:3,:)\| - \|x(1,:)\| \\ & \text{else} \\ & \qquad F = [\,] \\ & \text{end} \end{aligned} \tag{4.16}$$

As can be seen this constraint is implemented by considering an initial condition. If the distance between the S/C and the debris is inside the cone the constraint is given, otherwise there are not constraints. The *else* condition is not really needed, because the initial point of the docking maneuver is inside the *if* condition and the S/C moves towards the docking point. In order to avoid possible collision problems (maybe due to some computational trajectory error), it is possible to give a constraint also in this *else* condition (a possibility can be to use the constraint considered in the Rendezvous maneuver).

The cone constraint is simple, the S/C has to move inside the space described by the function. The radius of the base circumference $\|x(2:3,:)\|$ is equal to the distance $\|x(1,:)\|$. In this way, rewriting this as inequality it is possible to obtain the constraint.

The following NMPC settings are considered for this maneuver (they were found by means of trial and error procedure as seen in the section 3.4.2):

- *Model Functions:* It is the same model seen in the Rendezvous maneuver;
- *State Constraints:* It is the function seen previously;
- *Sampling Time:* $T_s = 1$ $[s]$;
- *Prediction Time:* $T_p = 100$ $[s]$;

- *System Order:* $n = 7$;
- *Q-matrix:* $Q = 10\ eye(6)$;
- *P-matrix:* $P = 1e3\ eye(6)$;
- *R-matrix:* $R = 0\ ones(3, 1)$;
- *Input Saturation Constraints:* $|u| \leq 5e3\ ones(3, 1)\ [N]$;
- *Used Flags:* $g = 0$, and $q = 0$;
- *Setted Flags:* $j = 1$. The first time when the algorithm run this piece of code it set $j = 1$ in order to avoid to repeat the (already done) Rendezvous maneuver;

The used simulink control scheme for this maneuver is the same of before. The only things that change are: the used reference signal $r = r_2$, and the NMPC control law, that this time uses the seen settings.

Estimation Maneuver

The estimation maneuver is the *heart* of this work. Thanks to the *on-line* estimation of the debris mass it is possible to complete optimally the GNC problem. The NMPC adapts its behavior on the basis of this estimation.

As seen in section 2.3, there are two main methods to estimate some variable: *recursive algorithms* and *filters (observers)*. In this work three algorithms are tested to accomplish this task. The first two are recursive algorithms, while the third one is an observer:

- *Recursive Least Squares (RLS) Algorithm:* It will see that it is not possible to use this method in order to obtain a good estimation;
- *Recursive Average Algorithm:* It works by collecting the solutions of some equations, at each sampling time. Then, in a recursive way, it makes an average on these values and gives the estimated mass value to the controller;
- *EKF:* It is an observer and by means of an internal mathematical model it is able to give the values of the state of the system. Contrary to a simple model (that uses only the information of the inputs), it works with both the inputs and the outputs. The outputs are used to correct the estimated values of the state;

These work on-line, hence, at each sampling time they updates the estimated debris-mass variable.

Once the Docking maneuver is complete, there is a rigid connection between the S/C and the debris. The system that has to be controlled becomes the fusion of the two bodies. In this way, the previous model that describe the dynamics of the S/C is not more valid. The *real value* of the debris mass must be considered in these equations. The new model, that takes into account this connection, is a modified model of the one seen before (see the equation 4.5):

$$
\begin{aligned}
&\dot{x}(1:3,:) = x(4:6,:)\\
&\dot{x}(4:6,:) = -\mu\frac{x(1:3,:)}{\|x(1:3,:)\|^3} + \frac{1}{x(7,:)+m_2}(F_d + d + u)\\
&u = 0 \quad \text{if } x(7,:) \leq m_b\\
&\dot{x}(7,:) = \begin{cases} 0 & \text{if } u = 0\\ -\frac{|u|}{v_e} & \text{if } u \neq 0 \end{cases}\\
&F_d = -\frac{1}{2}\rho\ C_D S|v|v, \quad \rho = \rho_0 e^{-\frac{r-r_0}{H}}
\end{aligned}
\tag{4.17}
$$

In this case also the model used by the NMPC is changed. Indeed, the used equations are the ones written according to the 2B problem, and the estimated value of the mass enter as a state variable. In this way, it is possible to obtain an adaptive NMPC:

$$
\begin{aligned}
&\dot{x}(1:3,:) = x(4:6,:)\\
&\dot{x}(4:6,:) = -\mu\frac{x(1:6,:)}{\|x(1:3,:)\|^3} + \frac{u}{x(7,:)+x(8,:)}\\
&\dot{x}(7,:) = 0\\
&\dot{x}(8,:) = 0\\
&y = x(1:6,:)
\end{aligned}
\tag{4.18}
$$

From now on, the two recursive algorithms are discussed. Then, it is presented the EKF.

An important consideration, that must be done, is that the controller gives to the system a command input u that it is calculated, as the optimal one for the planned trajectory, on the basis of the estimated value (x_8) of the real debris mass (m_2). The system, by using this command input, reacts according to the equations that describe its dynamics. If u is too small it is not possible to appreciate a divergent trajectory from the planned one. Considering this point, it is needed a condition that reject estimated values when u is below a certain limit.

The average recursive algorithm is build by considering the equations that describe the system motion. Indeed, it is possible to rewrite them in function of the unknown mass. In this way, it is possible to have three equations (one for each direction) on which collect data:

$$
\begin{aligned}
&\text{if} \quad |u(1,:)| \geq u_t \\
&m_{2\ 1} = \frac{u(1,:)}{\dot{x}(4,:) + (\mu x(1,:)/\|x(1:3,:)\|^3)} - x(7,:) \\
&\text{else} \\
&m_{2\ 1} = 0 \\
&\text{end} \\
&\text{if} \quad |u(2,:)| \geq u_t \\
&m_{2\ 2} = \frac{u(2,:)}{\dot{x}(5,:) + (\mu x(2,:)/\|x(1:3,:)\|^3)} - x(7,:) \\
&\text{else} \\
&m_{2\ 2} = 0 \\
&\text{end} \\
&\text{if} \quad |u(3,:)| \geq u_t \\
&m_{2\ 3} = \frac{u(3,:)}{\dot{x}(6,:) + (\mu x(3,:)/\|x(1:3,:)\|^3)} - x(7,:) \\
&\text{else} \\
&m_{2\ 3} = 0 \\
&\text{end}
\end{aligned}
\tag{4.19}
$$

Where u_t is a given threshold used to discard wrong estimated values (if $u \to 0 \implies m_{2\ i} \to 0$).

How it is possible to see, only the estimated values of the mass, corresponding to a given command input (greater than a given threshold), are considered.

The algorithm first, on the basis of the results of these three equations, calculates the average of them at the specific time. Then, it takes all these average values (one for each sampling time) and starting from the first until the value given at the present time, recursively, it calculates the average of them. The output value (that is updated at each sampling time), given by this last step, is applied to the controller as state variable (x_8):

```
if $m_{2\ 1} > 0 \wedge m_{2\ 2} > 0 \wedge m_{2\ 3} > 0)$
$m_{2\ partial} = \frac{1}{3}(m_{2\ 1} + m_{2\ 2} + m_{2\ 3})$
elseif $(m_{2\ 1} \neq m_{2\ 2} \wedge m_{2\ 2} \neq m_{2\ 1}) \wedge (m_{2\ 1} = 0 \vee m_{2\ 2} = 0 \vee m_{2\ 3} = 0)$
$m_{2\ partial} = \frac{1}{2}(m_{2\ 1} + m_{2\ 2} + m_{2\ 3})$
else
$m_{2\ partial} = (m_{2\ 1} + m_{2\ 2} + m_{2\ 3})$
end
for $i = 1 : size(m_{2\ partial})$
$m_{2\ est}(i,:) = \frac{1}{i}\sum m_{2\ partial}(i,:)$
end
```

(4.20)

Where i is the *i-th* present time. It contains all the previous collected values. At the same time the following relation is true: $m_{2\ est}(i,:) = x(8,:)$.

Another approach based on RLS method for the estimation problem is attempted. But, it will be seen that this method is not applicable. However, the mathematical formulation (found only for the 4-th equation of the dynamics model) of the problem is the following:

- *The problem is to find θ from:*

$$y = A\theta \implies \theta = (A^TA)^-1A^Ty$$

- *Measured Output:*

$$y(1) = \dot{x}(4,1) + \frac{\mu x(1,1)}{\|x(1:3,1)\|^3} = x(4,2) + \frac{\mu x(1,1)}{\|x(1:3,1)\|^3}$$
$$y(2) = \dot{x}(4,2) + \frac{\mu x(1,2)}{\|x(1:3,2)\|^3} = x(4,3) + \frac{\mu x(1,2)}{\|x(1:3,2)\|^3}$$
$$\vdots$$
$$y(N-1) = \dot{x}(4,N-1) + \frac{\mu x(1,N-1)}{\|x(1:3,N-1)\|^3} = x(4,N) + \frac{\mu x(1,N-1)}{\|x(1:3,N-1)\|^3}$$
$$y = [y(1);y(2);\ldots;y(N-1)]$$

- *Regressor A:*

$$A(1) = u(1,1), \quad A(2) = u(1,2), \quad \ldots \quad A(N-1) = u(1,N-1)$$
$$A = [A(1);A(2);\ldots;A(N-1)]$$

- *Solution:*

$$\theta = constant = (A^T A)^{-}1 A^T y = \frac{y(1)}{u(1,1)} + \frac{y(2)}{u(1,2)} + \cdots + \frac{y(N-1)}{u(1,N-1)}$$

- *Since in θ there is $x(7,:)$ (that depends on the fuel consumption) it can not be a constant and diverges at each step-time:*

$$\theta(1) = \frac{1}{x(7,1)+m_2}, \; \theta(2) = \frac{1}{x(7,2)+m_2}, \; \ldots \; \theta(N-1) = \frac{1}{x(7,N-1)+m_2}$$
$$\theta = [\theta(1); \theta(2); \ldots; \theta(N-1)]$$

As it can be seen, it is not possible to estimate the debris mass with the RLS method. However, a possible solution is to consider constant $x(7,:)$ in a certain range and then apply the algorithm in sufficiently small number of samples. By repeating this procedure, every N samples (in which $x(7,:) = constant$ it is considered), it is possible to find different values of θ (by a simple equation $m_{2\ est} = (1 - m_{const}\theta)/\theta$) and then by considering a recursive average it is possible to refine the estimated value of the mass. Another possibility is to reduce the input saturation constraint and obtain a small mass consumption. In this case, the small divergence can be seen as error on the measured output. Also in this case a sufficiently small number of samples are needed in order to avoid strong variations. Moreover, as seen before, if u is too small there can be wrong estimated values.

The EKF works in discrete time. In order to formalize the theoretical problem, general system equations (that describe the system plant) are considered and they have the following form:

$$x_{k+1} = f(x_k, u_k) + d_k, \quad y_k = h(x_k) + d_k^y$$

Where k is the time index, d_k is a disturbance, and d_k^y is a measurement noise. The measurements are only on the input u_k and on the output y_y. The goal is to using current and past values of these measurement to obtain an accurate estimate $\hat{x}_k$ of x_k.

The formulation used starts by defining two fundamental steps:

1. *Prediction:* The algorithm computes a prediction x_k^p on the state x_k using a model of the system:

$$x_k^p = \hat{f}(\hat{x}_{k-1}, u_{k-1}) \tag{4.21}$$

   Where $\hat{f}$ is a known model of the system. f is the exact model used to describe the dynamics of the plant and in general it is different from $\hat{f}$ (that it can be more general).

2. *Correction:* The measurements on the output are used to improve the prediction and obtain a better state estimate:

$$\hat{x}_k = x_k^p + K_k \Delta y_k \tag{4.22}$$

Where x_k is the estimate of the state at time k. It is an improvement of the prediction obtained in the previous step. K_k is a suitable matrix multiplied by the *innovation* Δy_k. The innovation is defined as the measured output minus the predicted output: $\Delta y_k = y_k - h(x_k^p)$.

The EKF works with the jacobians of the system computed along the trajectory:

$$F_k = \frac{\partial \hat{f}}{\partial x}(x_k, u_k), \quad H_k = \frac{\partial h}{\partial x}(x_k)$$

Other relevant quantities, needed for formulation purposes, are the following covariance matrices:

$$P_k = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T], \quad R^d = E[d_k^y(d_k^y)^T], \quad Q^d = E[d_k(d_k)^T]$$

Where the P_k is the covarince matrix of the estimation error. In the linear Kalman filter the matrix K_k is chosen in order to minimize this error. R^d and Q^d are respectively the covariance matrices of d_k^y and d_k. These last two matrices are chosen by trial and error procedure. While P_k at the initial time P_0 is chosen on the information of the initial state x_0. The on-line operations that are made by the algorithm are:

1. *Prediction:*

$$x_k^p = \hat{f}(\hat{x}_{k-1}, u_{k-1}), \quad P_k^p = F_{k-1}Pk - 1Fk - 1^T + Q^d \tag{4.23}$$

2. *Correction:*

$$\begin{aligned} &S_k = H_k P_k^p H_k^T + R^d, \quad K_k = P_k^p H_k^T S_k^- 1, \quad \Delta y_k = y_k - h(x_k) \\ &\hat{x}_k = x_k^p + K_k \Delta y_k, \quad P_k = (I - K_k H_k) P_k^p \end{aligned} \tag{4.24}$$

In the prediction step the covarince matrix P must be also predicted.

In this application the following mathematical model and settings are chosen:

- *Used EKF model:* It corresponds to a discretization of the model used by the controller. The discretization is made using Euler forward method:

$$\begin{aligned}
&x_p(1,:) = x(4,:)\tau + x(1,:) \\
&x_p(2,:) = x(5,:)\tau + x(2,:) \\
&x_p(3,:) = x(6,:)\tau + x(3,:) \\
&x_p(4,:) = \left(\frac{-x(1,:)\mu}{r^3} + \frac{u(1,:)}{x(7,:) + x(8,:)}\right)\tau + x(4,:) \\
&x_p(5,:) = \left(\frac{-x(2,:)\mu}{r^3} + \frac{u(2,:)}{x(7,:) + x(8,:)}\right)\tau + x(5,:) \\
&x_p(6,:) = \left(\frac{-x(3,:)\mu}{r^3} + \frac{u(3,:)}{x(7,:) + x(8,:)}\right)\tau + x(6,:) \\
&x_p(7,:) = x(7,:) \\
&x_p(8,:) = x(8,:) \\
&y = x(1:7,:)
\end{aligned} \tag{4.25}$$

- *Covarince Matrices:*

$$\begin{aligned} P_0 &= eye(8)[1e-6\ ones(7,1); 5e6] \\ Q^d &= eye(8)[1e-3\ ones(7,1); 1e-9] \\ R^d &= eye(7)1e2 \end{aligned} \tag{4.26}$$

  It can be seen that the values of P_0 take care of having good information on the first seven state variables at the initial time, while the information on the eighth state variable (that corresponds to the debris mass) are not so good. Taking R^d high means that the measured outputs that enter in the equations have an high spread. While, Q^d is considering that the disturbances given by the solutions of the model equations are not so high (the spread is almost equal to the one setted for the disturbances, coming from the sensors, in the simulink model). Since m_2 is a constant value the corresponding coefficient in Q^d is considered more stringent.

In this work are considered only the estimation processes done with the recursive average method and with the EKF. The first one is simple, it don't require any additive computational effort, or to modify the system, the convergence towards the real value is higher than the one given by the EKF, and the estimation result is good enough.

The EKF can be used in whose applications that require an high precision on the measurements. Indeed, it is possible to use multiple sensors combined with the EKF in order to increase the precision. This is called *sensor fusion.*

The time, in which the estimation takes place, is not so high. This is done in order to avoid fuel consumption. If a long time interval is considered the EKF will be the best choice. Because in recursive average the errors coming from the first solutions always will affect the last estimated value of the debris-mass. While, with the EKF the estimation error will converge exactly to zero thanks to the correction step.

Another important consideration is if one of the two estimation algorithm is maintained also in the future maneuvers (without having thrusters saturation) a better estimation can be obtained. It will be seen that this is not required since the mass estimation will be near to the real value. Anyway, as seen before, in order to increase the accuracy of the sensors the EKF could be used.

To obtain the saturation of the command input u, in order to avoid bad estimated values, a far point as reference is considered. If it is considered the previous average function, it can be seen that there isn't a rejecting condition to discard the values of $m_{2\ partial} = 0$. Considering the recursion of the problem this can cause estimation problem. A possible choice, for the discard condition, can be implemented holding the precedent value of $m_{2\ partial} \neq 0$ in all the subsequent samples in which $m_{2\ partial} = 0$. In this work, this condition is not considered, because the thrusters work in saturation. In this way, $m_{2\ partial} \neq 0$ is always satisfied. Indeed, before starting, the algorithm of this maneuver sets the inputs equal to the saturated values.

This time the constant reference signal is specified in the GE frame. In this way, the problem has the meaning of *set-point* maneuver.

It is worth to note that the system doesn't reach this reference. Since, the estimation maneuver takes place only in a finite period equal to the *Partial Simulation Time*. For this reason the algorithm conditions, to repeat this maneuver, are not needed. There are only flags to understand when the maneuver is made.

$$r_3 = [1e10; 1e10; 1e10; 0; 0; 0]$$

The considered state constraint is related to the Earth. The planned trajectory can not pass near to the Earth surface, or worst it can not be planned as a collision trajectory. The constraint has the following form:

$$F = 1.05\ r_e - \|x(1:3,:)\| \tag{4.27}$$

Where r_e is the, already seen, Earth's radius.

The following NMPC settings are considered for this maneuver (they were found by means of trial and error procedure as seen in the section 3.4.2):

- *Model Functions:* They are the ones described in 4.18;
- *State Constraints:* It is the function seen previously;
- *Sampling Time:* $T_s = 0.01$ $[s]$;
- *Prediction Time:* $T_p = 100$ $[s]$;
- *System Order:* $n = 8$;
- *Q-matrix:* $Q = zeros(6)$;
- *P-matrix:* $P = zeros(6)$;
- *R-matrix:* $R = 0.1\ ones(3,1)$;
- *Input Saturation Constraints:* $|u| \leq 10e3\ ones(3,1)$ $[N]$;
- *Used Flags:* $oc = 0$, $q = 0$;
- *Setted Flags:* $g = 1$, $q = 1$. The first is used to avoid to repeat the Docking maneuver. The second one is used to compute the estimation value only in a partial simulation time. In this way, the number of samples are $N = (t_{step}/T_s) + 1 = 5001$, with $t_{step} = 50$ $[s]$, $T_s = 0.01$ $[s]$, while the $+1$ takes into account the initial value.

The used simulink control scheme is the following:

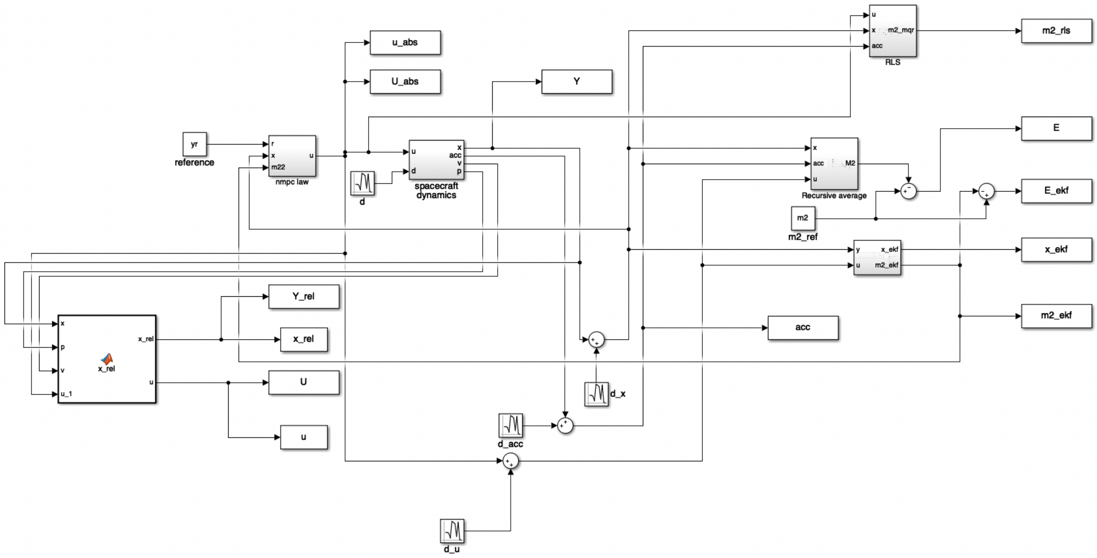

Figure 4.5: Estimation maneuver control scheme

As it can be seen, there is also the *RLS* block to calculate $m_{2\ est}$ with the RLS method (applied only in the forth system equation). But, the estimation, as already seen, is not correct. The *Random number* blocks are disturbance that take into account measurement errors and small perturbations. The *Recursive average* block contains the equations seen before and written with simulink blocks and functions. While, inside the *EKF* block there is the algorithm that works with the previous model and matrices. In the following there are the three estimation blocks:

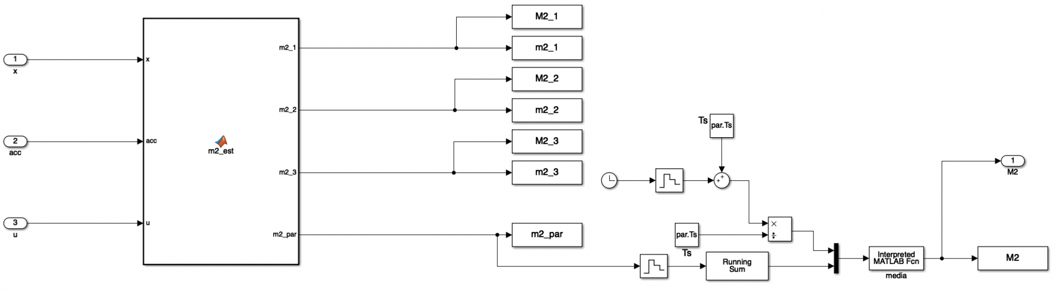

Figure 4.6: Recursive average block

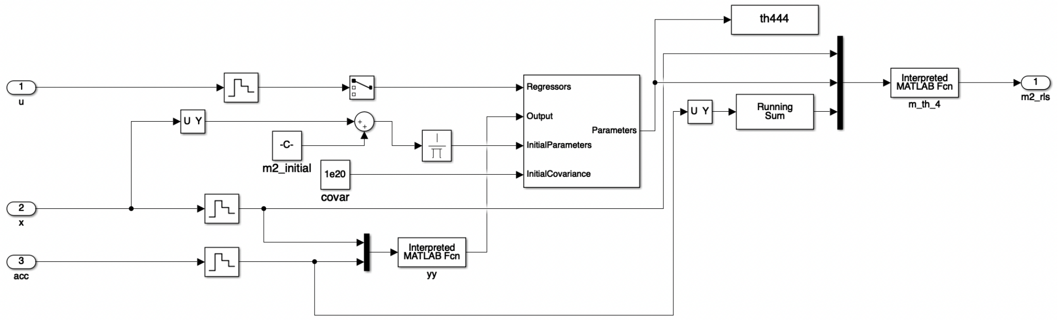

Figure 4.7: RLS block

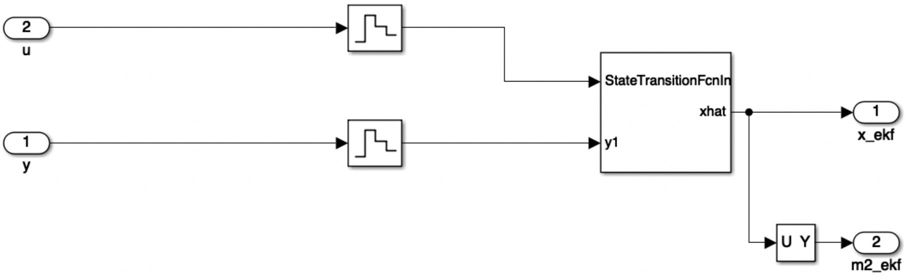

Figure 4.8: EKF block

Second Rendezvous Maneuver

This maneuver is not really needed. It is considered only to have a convenient initial starting point, for the Orbit-Change maneuver. The mission task of this work is to bring the debris faraway in another orbit. It maybe convenient to start from a point that is on an *empty* orbit, or that minimizes the planned trajectory without considering constraints due to the presence of other satellites.

This type of maneuver is Rendezvous/Docking-like. In the sense, that a final point in a relative frame is considered as reference. This time, there are not objects to approach and for this reason the constraints can be considered less stringent than in the first two precision maneuvers.

The system plant remains the same of before. While, once the estimation process is completed, the final value of the debris mass is applied as a constant parameter into the controller model:

$$\begin{aligned}
&\dot{x}(1:3,:) = x(4:6,:)\\
&\dot{x}(4,:) = 3\omega^2 x(1,:) + 2\omega x(5,:) + \frac{u(1,:)}{x(7,:) + m_{2\ est}}\\
&\dot{x}(5,:) = -2\omega x(4,:) + \frac{u(2,:)}{x(7,:) + m_{2\ est}}\\
&\dot{x}(6,:) = -\omega^2 x(3,:) + \frac{u(3,:)}{x(7,:) + m_{2\ est}}\\
&\dot{x}(7,:) = 0\\
&y = x(1:6,:)
\end{aligned} \tag{4.28}$$

For simplicity reasons, the final point of this maneuver is considered equal to the one that would have the CoM of the debris in its previous orbit. The constant reference signal and the conditions algorithm are equal to the second one:

$$\begin{aligned}
&r_4 = r_2 = [0;0;0;0;0;0]\\
&tol = tol_1 = tol_2\\
&|(\|x_{rel}(end, 1:3)^T\| - \|r_4(1:3)\|)| - tol \geq 0\\
&|(\|x_{rel}(end, 4:6)^T\| - \|r_4(4:6)\|)| - tol \geq 0
\end{aligned}$$

The considered state constraint function takes into account that the planned trajectory can not pass near to the Earth's surface, like the 4.27, but this time it is written in the LVLH frame:

$$F = 1.05\ r_e - \|[x(1,:) - r_e + a, x(2,:), x(3,:)]\| \tag{4.29}$$

Where a is the previous *debris radius*.

The following NMPC settings are considered for this maneuver (they were found by means of trial and error procedure as seen in the section 3.4.2):

- *Model Functions:* They are the ones described in 4.28;
- *State Constraints:* It is the function seen previously;
- *Sampling Time:* $T_s = 1\ [s]$;
- *Prediction Time:* $T_p = 100\ [s]$;
- *System Order:* $n = 7$;
- *Q-matrix:* $Q = zeros(6)$;
- *P-matrix:* $P = 1\ eye(6)$;
- *R-matrix:* $R = zeros(3,1)$;

- *Input Saturation Constraints:* $|u| \leq 10e3\ ones(3,1)\ [N]$;

- *Used Flags:* $oc = 0$, $q = 1$;

The used simulink control scheme is the same of the one seen in the first two maneuvers (see figure:4.4). The only things that change are: the used functions for the model and the controller, the new reference, and the new settings used by the NMPC.

Orbit-Change Maneuver

The last maneuver, to accomplish the mission task, is the Orbit-Change maneuver. As seen before, it consists to increase the *major-semi axis* of the debris orbit.

Also in this case, the system plant remains the same. While, the model used by the controller is written in the GE frame:

$$\begin{aligned}
&\dot{x}(1:3,:) = x(4:6,:)\\
&\dot{x}(4:6,:) = -\mu\frac{x(1:6,:)}{\|x(1:3,:)\|^3} + \frac{u}{x(7,:) + m_{2\ est}}\\
&\dot{x}(7,:) = 0\\
&y = rv2oe(x)
\end{aligned} \tag{4.30}$$

Where the output y is expressed in *orbital-elements* thanks to the *rv2oe* function.

Since y is expressed in orbital-elements, also the constant reference signal to be tracked is in the same form:

$$r = [a_r; e_{r1}; e_{r2}; e_{r3}; (\cos i)_r] = [r_e + 900e3; 0; 0; 0; 0; 1]$$

Where a_r is the semi-major axis reference, e_{ri} are the components of the eccentricity vector, and $(\cos i)_r$ is the cosine of the inclination angle. The reference orbit, in this way, is circular, equatorial (since $e_r = 0$ and $(\cos i)_r = 1$), and it is at distance a_r from the Earth's CoM.

The state constraint in this case is equal to the one seen in the estimation maneuver (see equation 4.27). If there are other objects (like satellites or debris) their position must be take into account as state constraint functions. For simplicity reasons, the consider space between the initial orbit to the final one is considered empty. The considered constraint is the only one due to the Earth.

The following NMPC settings are considered for this maneuver (they were found by means of trial and error procedure as seen in the section 3.4.2):

- *Model Functions:* They are the ones described in 4.30;

- *State Constraints:* It is the function 4.27;

- *Sampling Time:* $T_s = 5$ $[s]$;
- *Prediction Time:* $T_p = 100$ $[s]$;
- *System Order:* $n = 7$;
- *Q-matrix:* $Q = zeros(5)$;
- *P-matrix:* $P = diag([1, 1e6, 1e6, 1e6, 1e7])$;
- *R-matrix:* $R = zeros(3, 1)$;
- *Input Saturation Constraints:* $|u| \leq 132e3\ ones(3, 1)$ $[N]$;
- *Setted Flags:* $oc = 1$. It is used to avoid to repeat the second Rendezvous maneuver;

The simulink block diagram is the following:

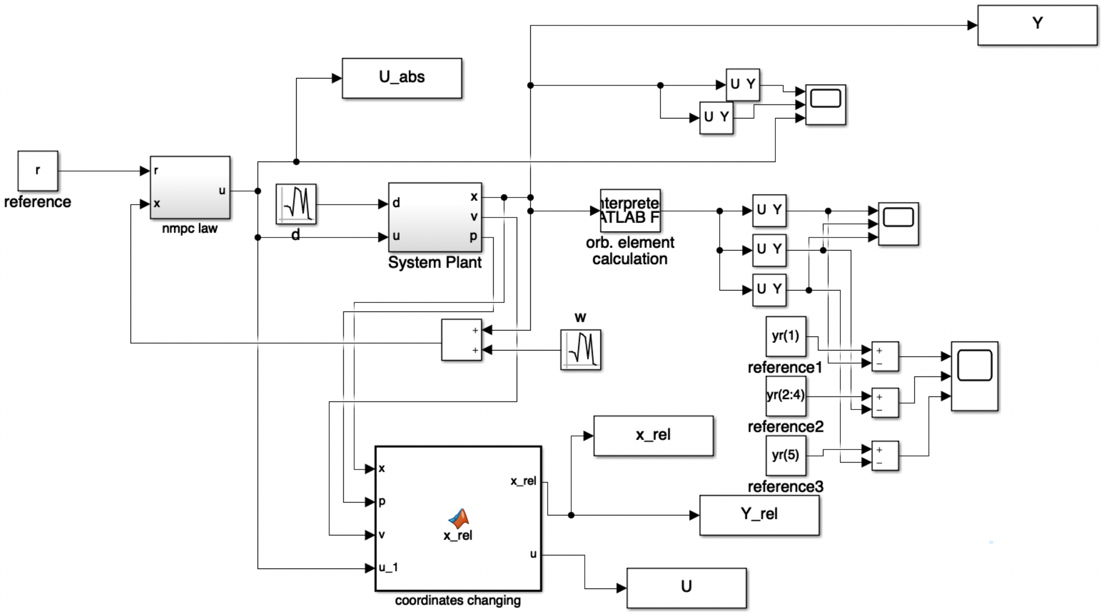

Figure 4.9: Orbit-Change control scheme

Chapter 5

Simulation Results

5.1 Simulation Results

By the means of matlab and simulink simulation, in this chapter will be seen the obtained results. First, the solutions of each maneuver are treated in terms of: obtained trajectories, fuel consumption, and if the constraints are satisfied. Then, It will make a comparison without estimating the debris mass.

The simulink solvers used, in all the first four maneuvers, are a variable-step solver *ODE-45*. While, the solver used in the Orbit-Change is *ODE-3*, that is a fixed-step solver (with fixed-step size equal to $st_{size} = 5$).

5.2 Rendezvous Maneuver

The considered simulation parameters are:

- *Drag force parameters:* $C_D = 1$, $S = 12$ $[m^2]$, $\rho_0 = 1.22$ $[kg/m^3]$, $H = 8e3$ $[m]$;
- *Other Perturbations/Disturbances:* $std(d(t)) = 1$ $[N]$;
- *Condition tolerances:* $tol_1 = 0.5$ $[m]$, $tol_2 = 0.05$ $[m/s]$;

5.2.1 Time and Delta-V

The reference point is reached after:

$$t = 1100 \; [s] \approx 18 \; [min]$$

While the Delta-V is:

$$\Delta V_1 = |V_{final} - V_{start}| = 0.1203 \; [m/s]$$

5.2.2 Command Input, Position, and Velocity

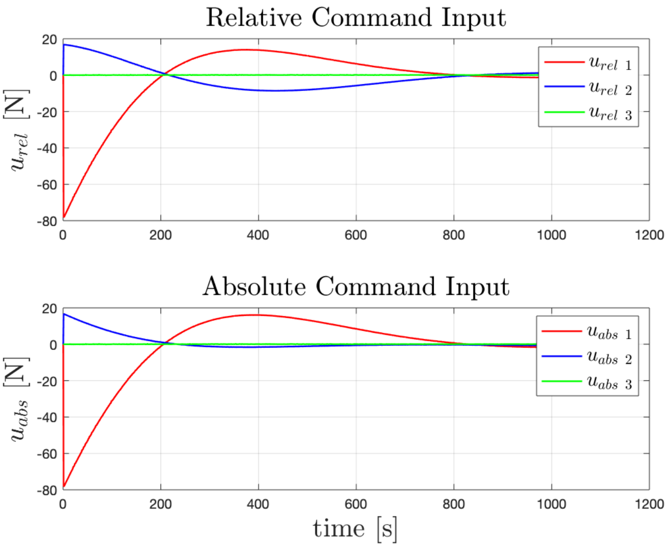

Figure 5.1: Relative and Absolute Command Input Vs Time

The constraints on the input are expressed in the LVLH frame. As it can be seen they are satisfied.

In the following the relative and absolute plots of the position and velocity are given:

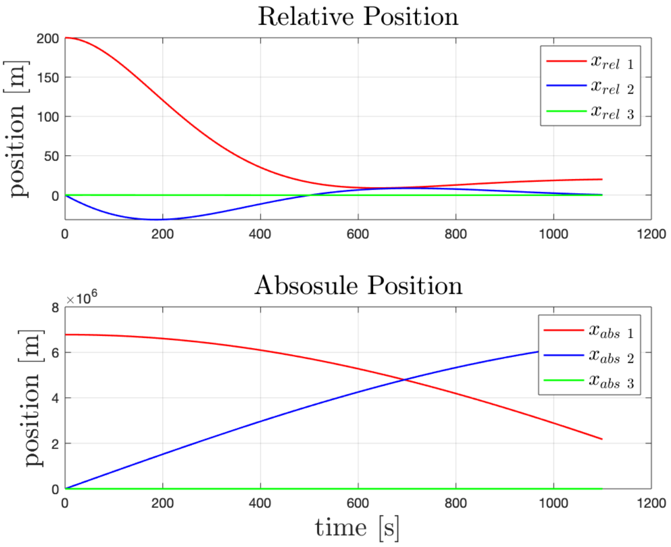

Figure 5.2: Relative and Absolute Position Vs Time

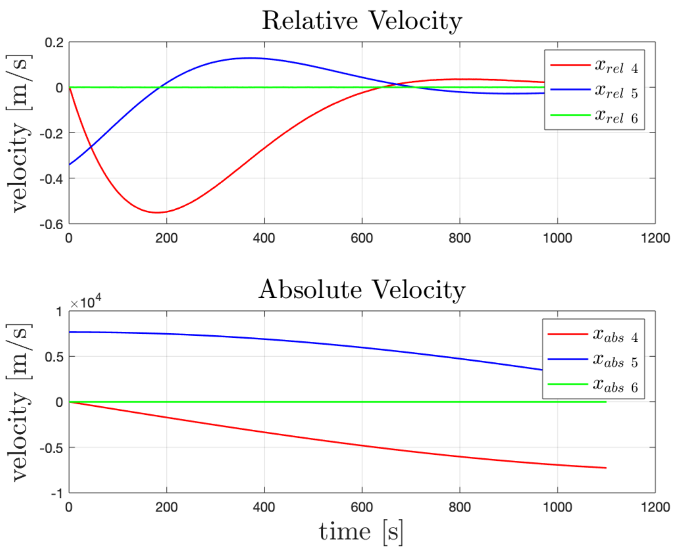

Figure 5.3: Relative and Absolute Velocity Vs Time

5.2.3 Trajectory

The relative and absolute trajectories, for this maneuver, are now presented. In the relative ones, it can be seen that, the constraints on the state are satisfied.

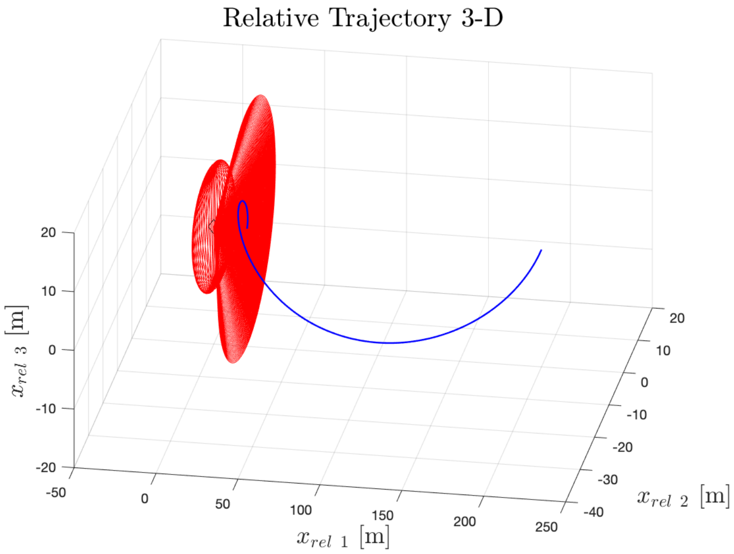

Figure 5.4: Relative position between the S/C and the debris

The red ball indicates the constraint function used in this maneuver (the red cone is used for the next Docking maneuver, but for simplicity reasons it is plotted together with the rendezvous constraint).

In the following is reported the trajectory in a *2-D* plane:

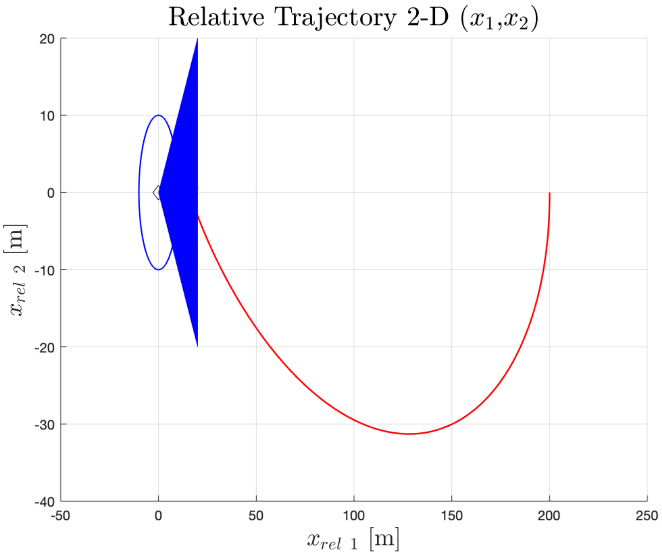

Figure 5.5: Relative position between the S/C and the debris in a 2D plane

Here, the absolute trajectory seen in a *3-D* space and in a *2-D* plane:

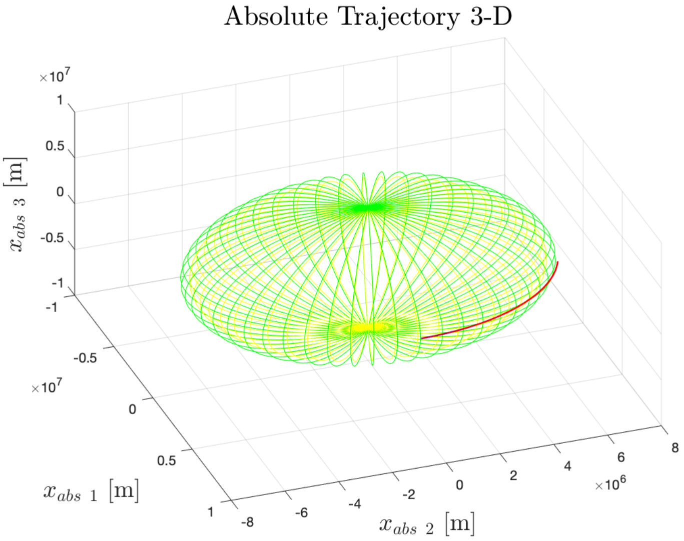

Figure 5.6: Absolute position between the S/C and the debris in a 3D space

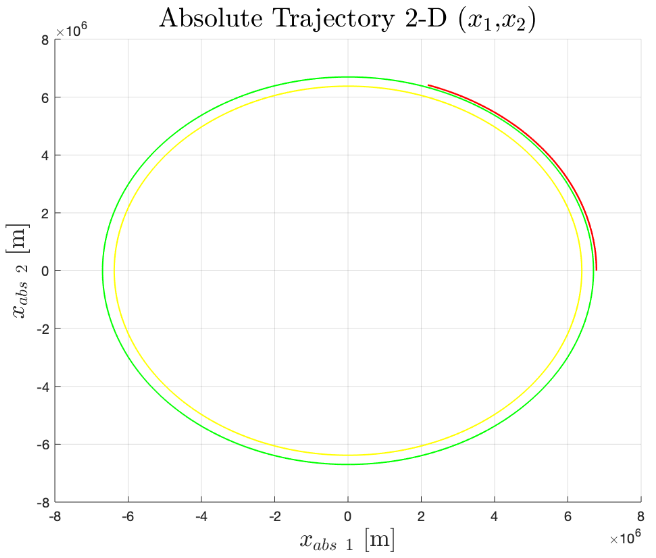

Figure 5.7: Absolute position between the S/C and the debris in a 2D plane

The initial position between the S/C and the debris is too small to appreciate a difference between the two trajectories. However, in the end, when a comparison without the estimation process it will make, it is presented a zoomed plot in a specific point. In that case it is possible to see that the two trajectories are separated.

5.2.4 Fuel Consumption

The total fuel's mass used is:

$$m_{lost} = 2.96 \approx 3 \ [kg]$$

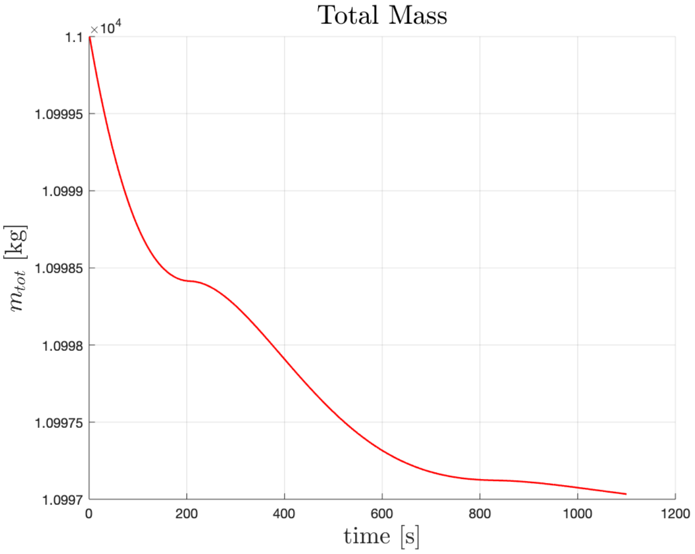

Figure 5.8: Total Mass Vs Time

It is not too much. Indeed, this maneuver is a precision one. There is not needed to have high command activity.

5.3 Docking Maneuver

The considered simulation parameters are:

- *Drag force parameters:* $C_D = 1$, $S = 12\ [m^2]$, $\rho_0 = 1.22\ [kg/m^3]$, $H = 8e3\ [m]$;
- *Other Perturbations/Disturbances:* $std(d(t)) = 1\ [N]$;
- *Condition tolerances:* $tol_1 = 0.5\ [m]$, $tol_2 = 0.05\ [m/s]$. In order to have a smooth docking, a stringent condition is considered on the relative velocity;

5.3.1 Time and Delta-V

The reference point is reached after:

$$t = 1550\ [s] \approx 26\ [min]$$

While the maneuver time is:

$$t = 1550 - 1100 = 450\ [s] = 7.5\ [min]$$

The Delta-V is:

$$\Delta V_2 = 0.0080\ [m/s]$$

5.3.2 Command Input, Position, and Velocity

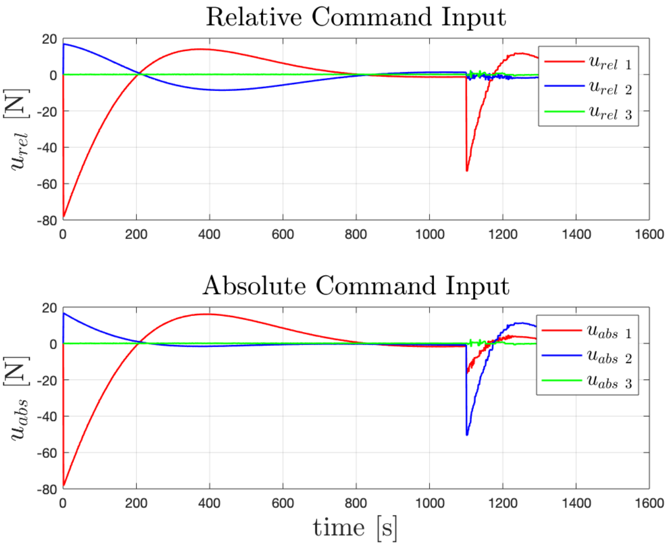

Figure 5.9: Relative and Absolute Command Input Vs Time

The constraints on the input are expressed in the LVLH frame. As it can be seen they are satisfied.

In the following the relative and absolute plots of the position and velocity are given:

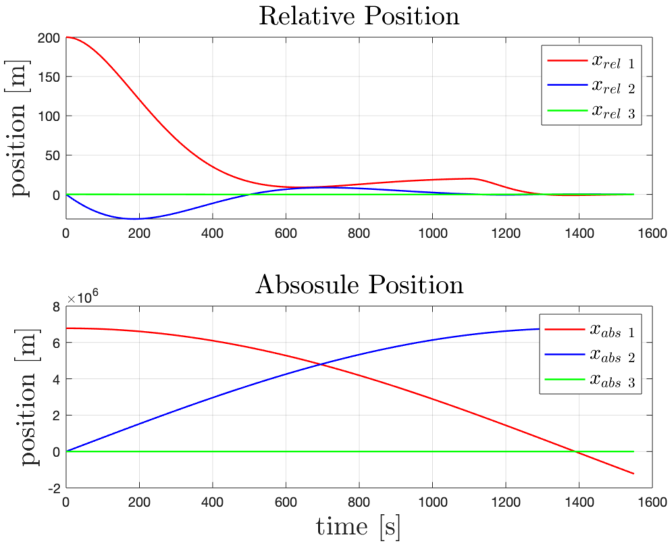

Figure 5.10: Relative and Absolute Position Vs Time

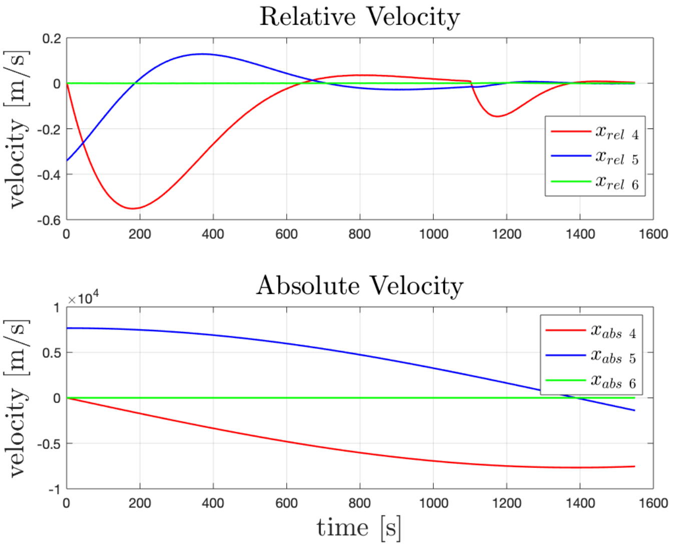

Figure 5.11: Relative and Absolute Velocity Vs Time

5.3.3 Trajectory

The relative and absolute trajectories, for this maneuver, are now presented. In the relative ones, it can be seen that, the constraints on the state are satisfied.

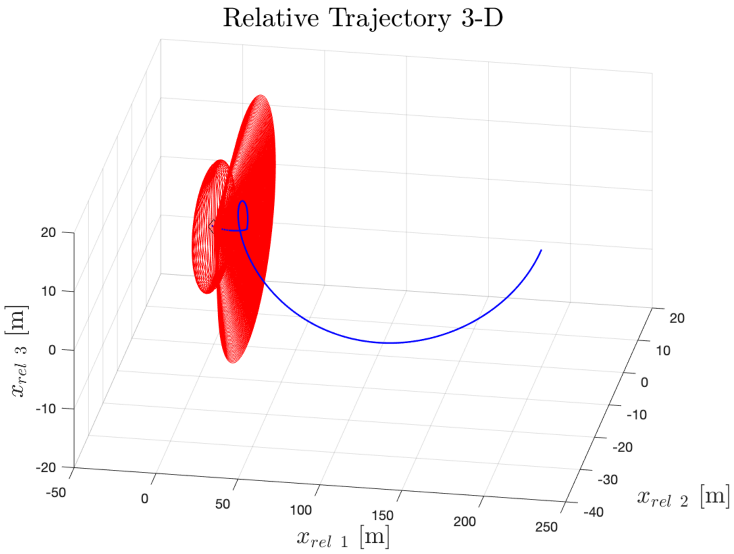

Figure 5.12: Relative position between the S/C and the debris

the red cone is used to indicate the constraint in this maneuver.

In the following is reported the trajectory in a *2-D* plane:

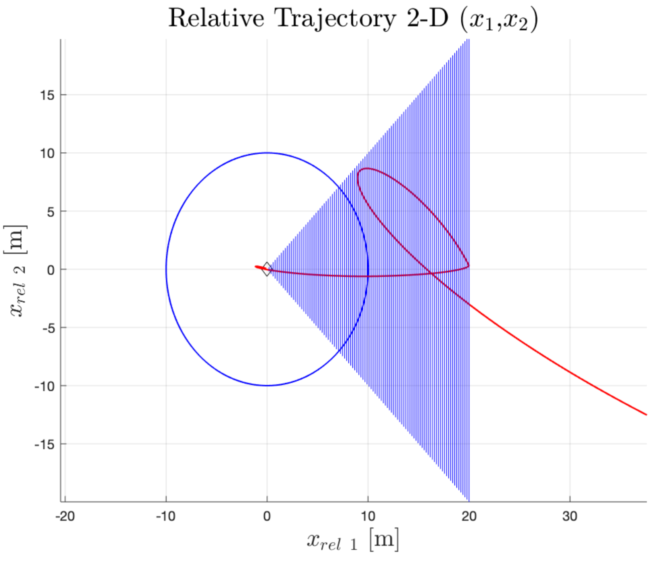

Figure 5.13: Relative position between the S/C and the debris in a 2D plane

Here, the absolute trajectory seen in a *3-D* space and in a *2-D* plane:

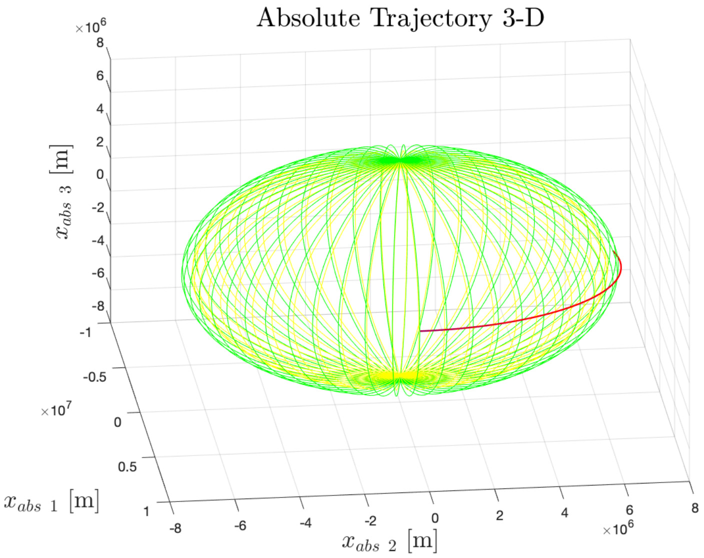

Figure 5.14: Absolute position between the S/C and the debris in a 3D space

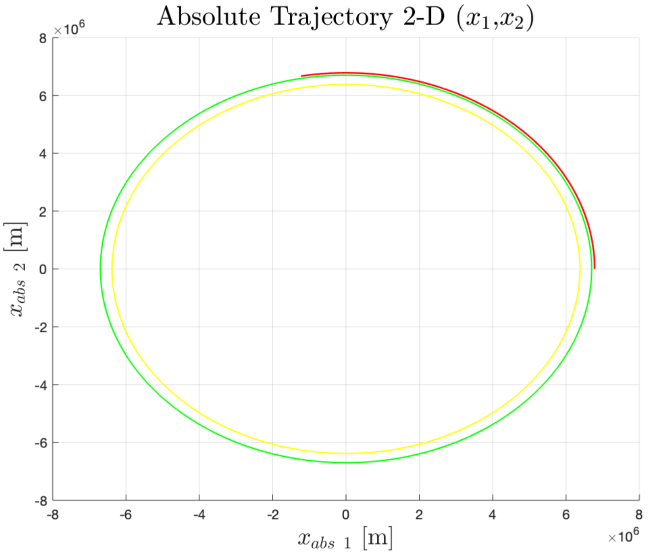

Figure 5.15: Absolute position between the S/C and the debris in a 2D plane

5.3.4 Fuel Consumption

As it can be seen from the following plot, the total fuel's mass used is:

$$m_{lost} = 3.76 \approx 3.80\ [kg]$$

While, the fuel used in this maneuver is:

$$m_{lost} = 3.76 - 2.96 = 0.80\ [kg]$$

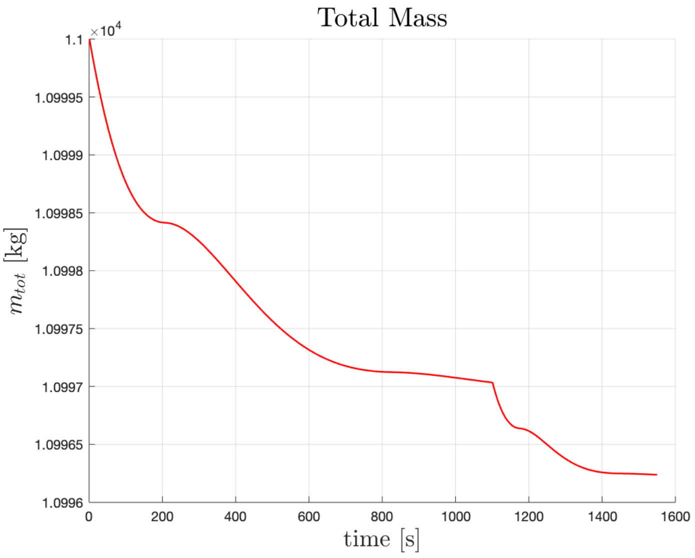

Figure 5.16: Total Mass Vs Time

It is not too much. Indeed, this maneuver is a precision one. There is not needed to have high command activity.

5.4 Estimation Maneuver

The considered simulation parameters are:

- *Drag force parameters:* $C_D = 1$, $S = 12\ [m^2]$, $\rho_0 = 1.22\ [kg/m^3]$, $H = 8e3\ [m]$;
- *Perturbations/Disturbances:* $std(d(t)) = 1\ [N]$;
- *Measurements Errors:*
  $std(w(t)) = [ones(3,1)\ [m]; 1e-1*ones(3,1)\ [m/s]; 1e-3\ [kg]]$,
  $std(d_x(t)) = [ones(3,1)\ [m]; 1e-1*ones(3,1)\ [m/s]; 1e-3\ [kg]]$,
  $std(d_{\dot{x}}(t)) = 1e-4*ones(7,1)\ [ones(3,1)[m/s]; ones(3,1)[m/s^2]; [kg/s]]$,
  $std(d_u(t)) = 1e-1*ones(3,1)\ [N]$
  They are considered in order to see how much good is the estimation process;

5.4.1 Time and Delta-V

The estimation process is complete after 50 $[s]$. The total time is:

$$t = 1600\ [s] \approx 26.5\ [min]$$

As seen before, in this maneuver, the reference is not reached. It is used only to have thrusters saturation.
The Delta-V is:

$$\Delta V_3 = 52.5443\ [m/s]$$

5.4.2 Command Input, Position, and Velocity

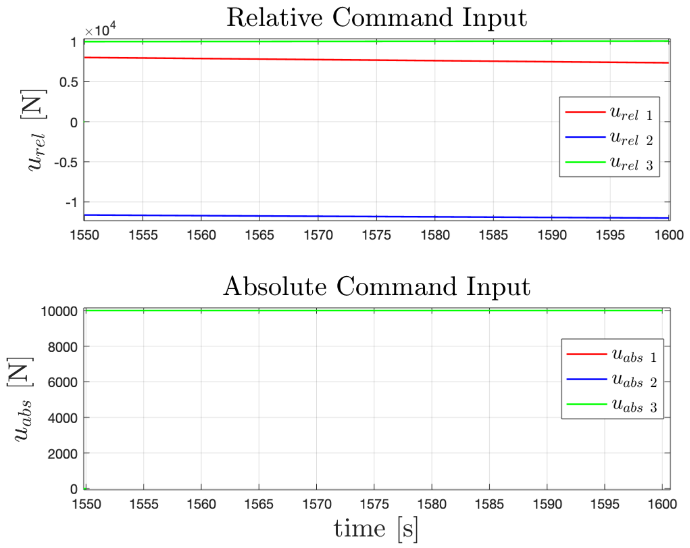

Figure 5.17: Relative and Absolute Command Input Vs Time

The constraints on the input, this time, are expressed in the GE frame. As it can be seen, the thrusters are in saturation conditions.

In the following the relative and absolute plots of the position and velocity are given:

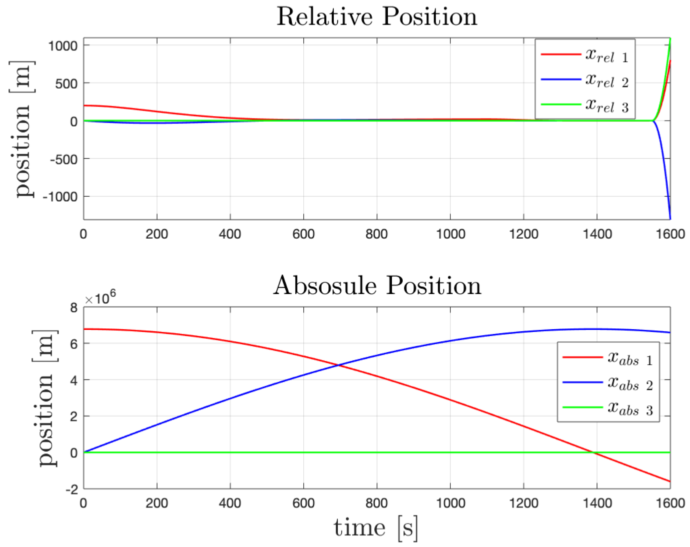

Figure 5.18: Relative and Absolute Position Vs Time

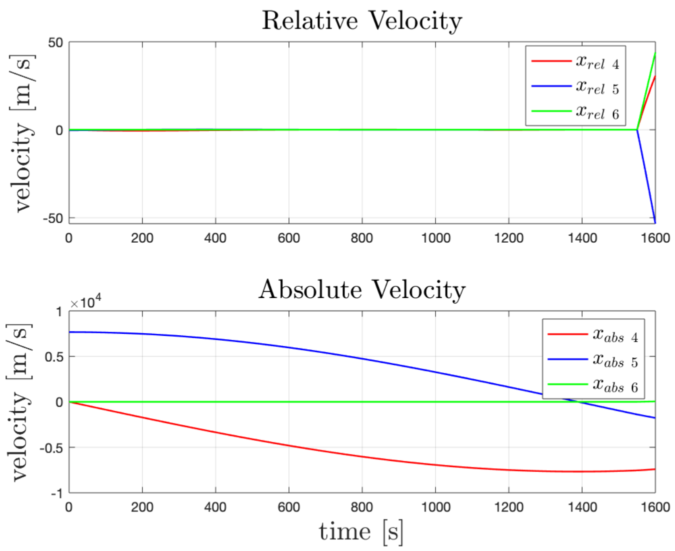

Figure 5.19: Relative and Absolute Velocity Vs Time

5.4.3 Trajectory

The relative and absolute trajectories, for this maneuver, are now presented. In the absolute ones, it can be seen that, the constraint on the state are satisfied.

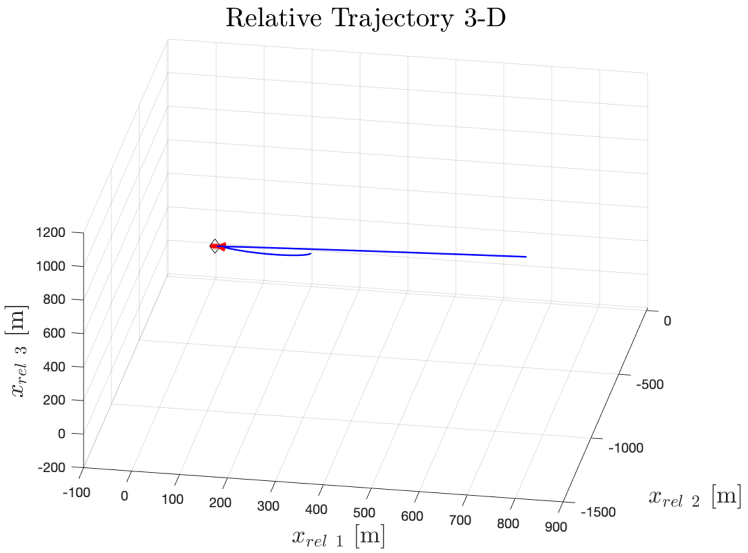

Figure 5.20: Relative position between the S/C and the debris

In the following is reported the trajectory in a *2-D* plane:

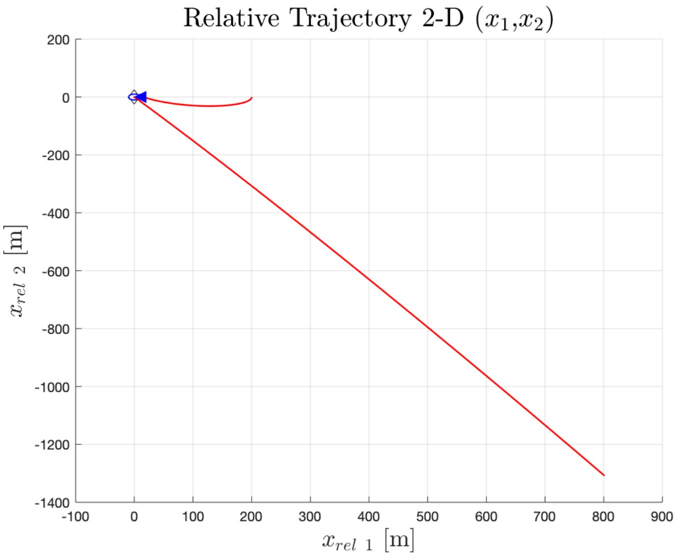

Figure 5.21: Relative position between the S/C and the debris in a 2D plane

Here, the absolute trajectory seen in a *3-D* space and in a *2-D* plane:

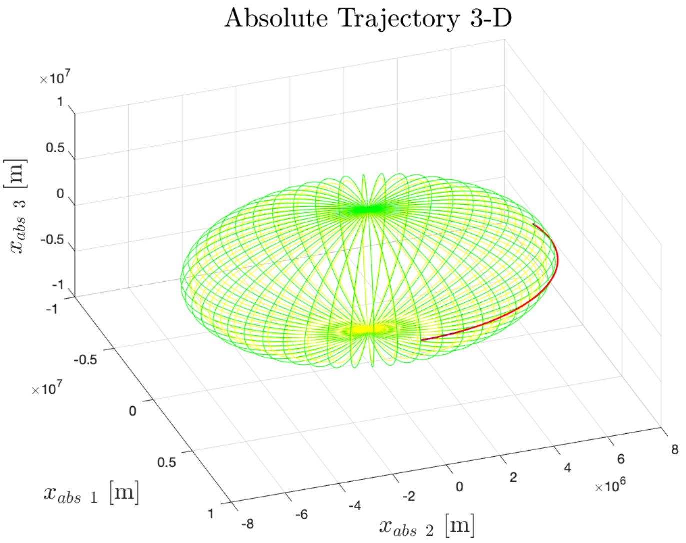

Figure 5.22: Absolute position between the S/C and the debris in a 3D space

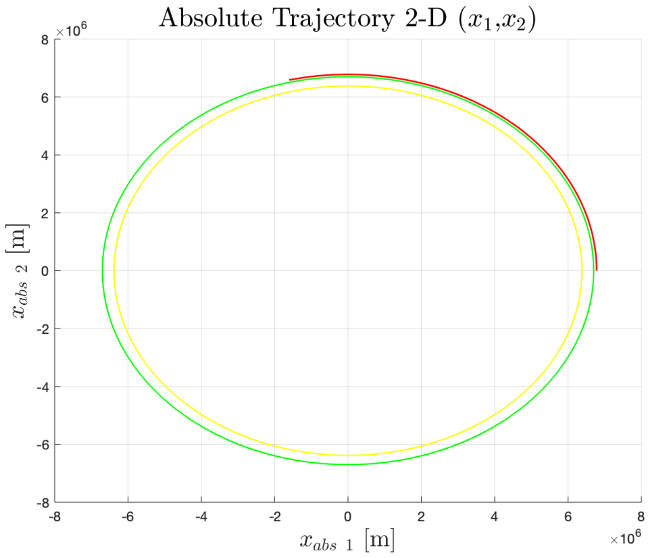

Figure 5.23: Absolute position between the S/C and the debris in a 2D plane

5.4.4 Fuel Consumption

After the Docking maneuver, the system becomes the fusion of the two bodies. For this reason the total mass of the system is increased. The total fuel used is:

$$m_{lost} = 200.58 \approx 200.6\ [kg]$$

While, the fuel used in this maneuver is:

$$m_{lost} = 200.58 - 3.76 = 196.82\ [kg]$$

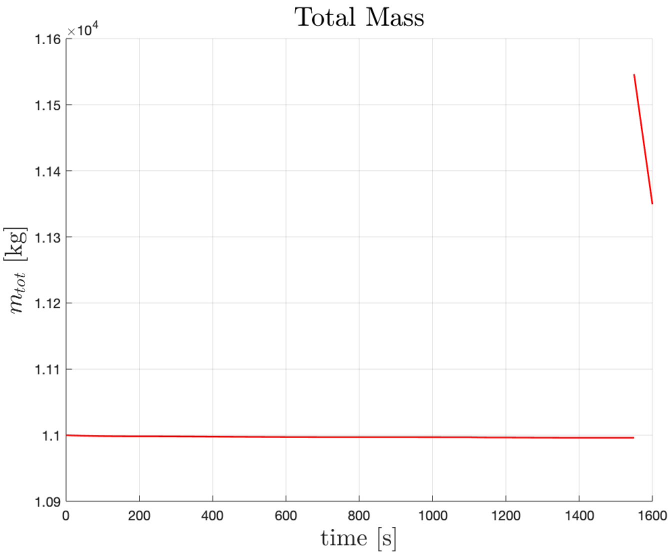

Figure 5.24: Total Mass Vs Time

5.4.5 Estimation Process

In the following plots can be seen:

- The results given at each sampling time of the three equations used to estimate the debris mass (one for each direction);

- The result of the recursive average;

- The result of the EKF estimation;

- The *Estimation errors*;

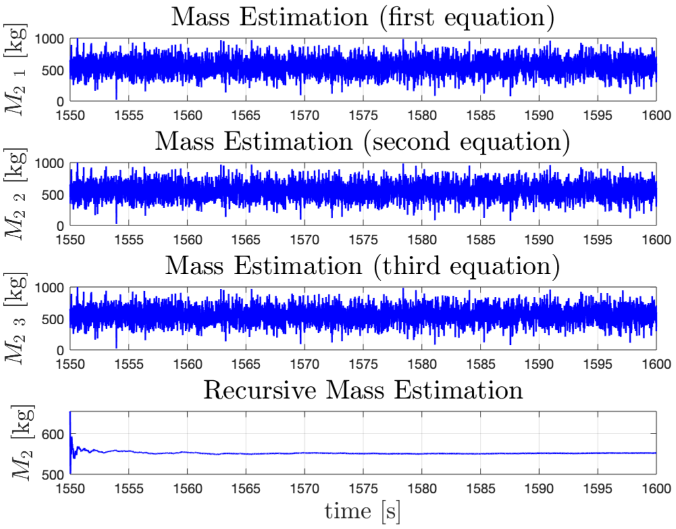

Figure 5.25: Estimated Mass Vs Time

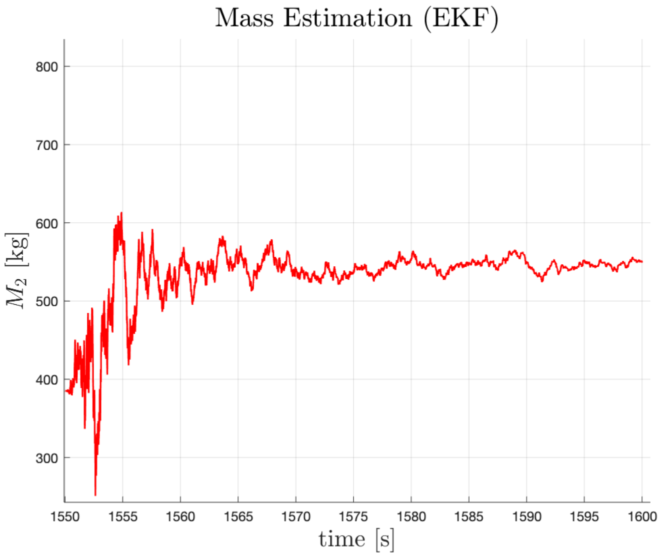

Figure 5.26: Estimated Mass Vs Time

The final estimated values of the mass are:

$$m_{2\ av\ est} = 552.03\ [kg], \quad m_{2\ ekf\ est} = 549.07\ [kg], \quad m_{2\ est} = m_{2\ av\ est},$$

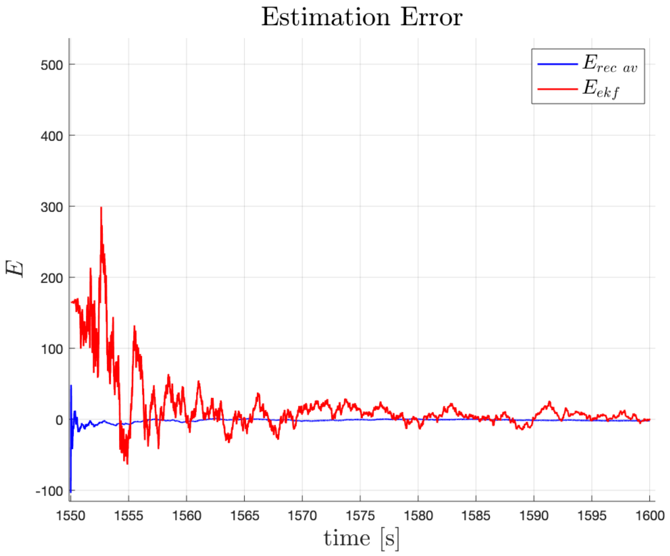

Figure 5.27: Estimation Error Vs Time

The final errors are:

$$|e_{av\ final}| = |m_2 - m_{2\ av\ est}| = 2.03\ [kg], \quad |e_{ekf\ final}| = 0.92\ [kg]$$

It can be also seen that, the errors slowly converges to zero.

In this work are considered only the estimation processes done via the recursive average method and via the EKF. The first one is simple, the convergence towards the real value is higher than the one given by the EKF, and the estimation result is good enough.

The EKF can be used in whose applications that require an high precision on the measurements. Indeed, it is possible to use multiple sensors combined with the EKF in order to increase the precision. This is called *sensor fusion*.

The time, in which the estimation takes place, is not so high. This is done in order to avoid fuel consumption. If a long time interval is considered the EKF will be the best choice. Because in recursive average the errors coming from the first solutions always will affect the last estimated value of the debris-mass. While, with the EKF the estimation error will converge exactly to zero thanks to the correction step.

Another important consideration is if one of the two estimation algorithm is maintained also in the future maneuvers (without having thrusters saturation) a better estimation can be obtained. This is not really required since the mass estimation is near to the real value. Anyway, as seen before, in order to increase the accuracy of the sensors and of the estimation the EKF could be considered in the whole mission.

5.5 Second Rendezvous Maneuver

The considered simulation parameters are:

- *Drag force parameters:* $C_D = 1$, $S = 12\ [m^2]$, $\rho_0 = 1.22\ [kg/m^3]$, $H = 8e3\ [m]$;
- *Other Perturbations/Disturbances:* $std(d(t)) = 1\ [N]$;
- *Condition tolerances:* $tol_1 = 0.5\ [m]$, $tol_2 = 0.5\ [m/s]$. This time the condition on the velocity is less stringent than before. In the point to be reached there are not objects. As seen before, this maneuver is done only to have a good starting point for the Orbit-Change maneuver.

5.5.1 Time and Delta-V

The reference point is reached after:

$$t = 2500\ [s] \approx 41.5\ [min]$$

While, the maneuver time is:

$$t = 2500 - 1600 = 900\ [s] = 15\ [min]$$

The Delta-V is:

$$\Delta V_4 = 52.5663\ [m/s]$$

5.5.2 Command Input, Position, and Velocity

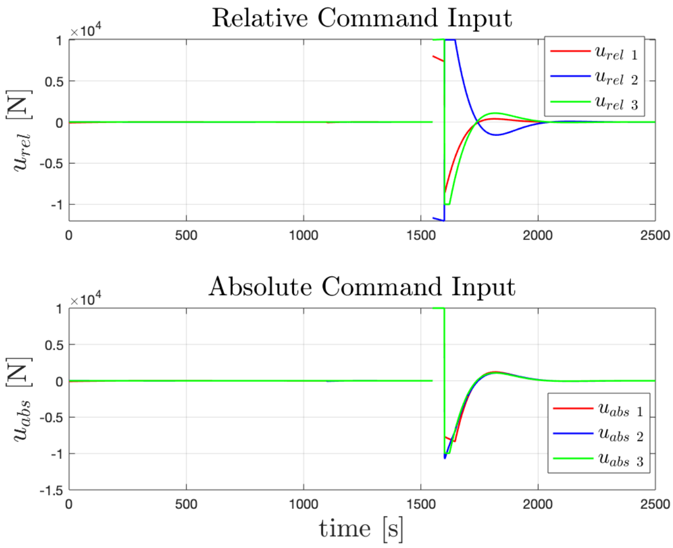

Figure 5.28: Relative and Absolute Command Input Vs Time

The constraints on the input are expressed in the LVLH frame. As it can be seen they are satisfied.

In the following the relative and absolute plots of the position and velocity are given:

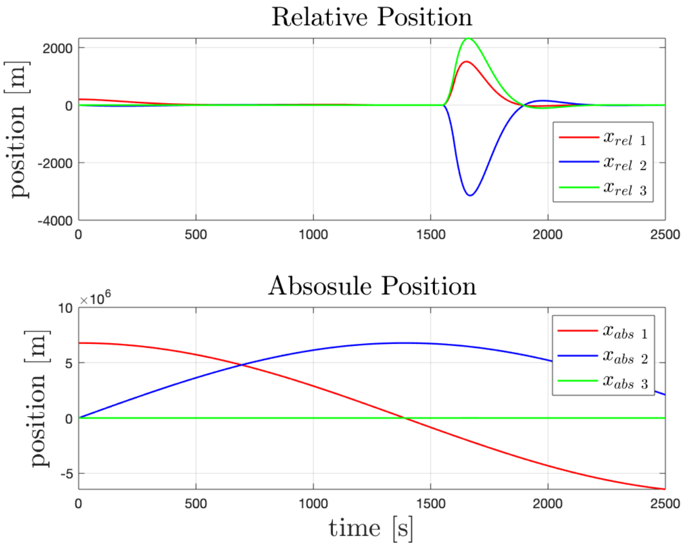

Figure 5.29: Relative and Absolute Position Vs Time

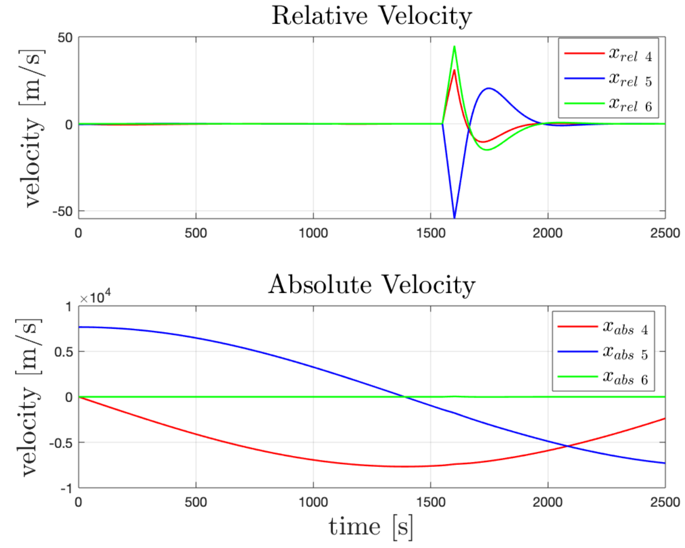

Figure 5.30: Relative and Absolute Velocity Vs Time

5.5.3 Trajectory

The relative and absolute trajectories, for this maneuver, are now presented. In the absolute ones, it can be seen that, the constraints on the state are satisfied.

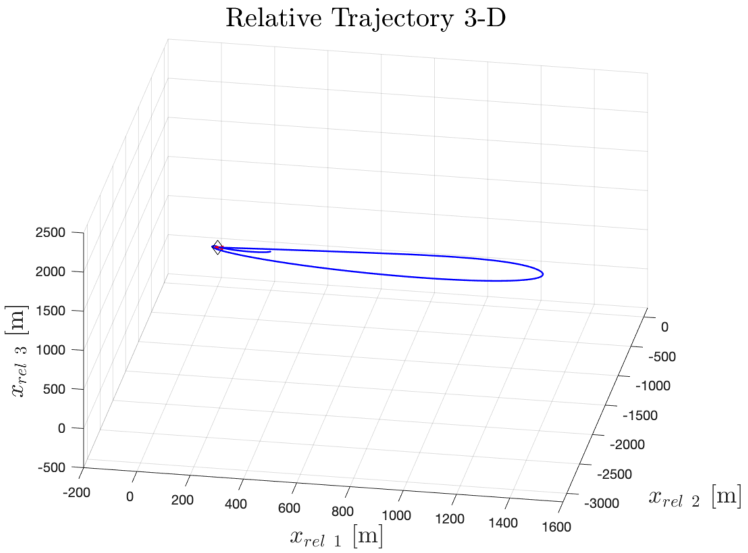

Figure 5.31: Relative position between the S/C and the debris

the red cone is used to indicate the constraint in this maneuver.

In the following is reported the trajectory in a *2-D* plane:

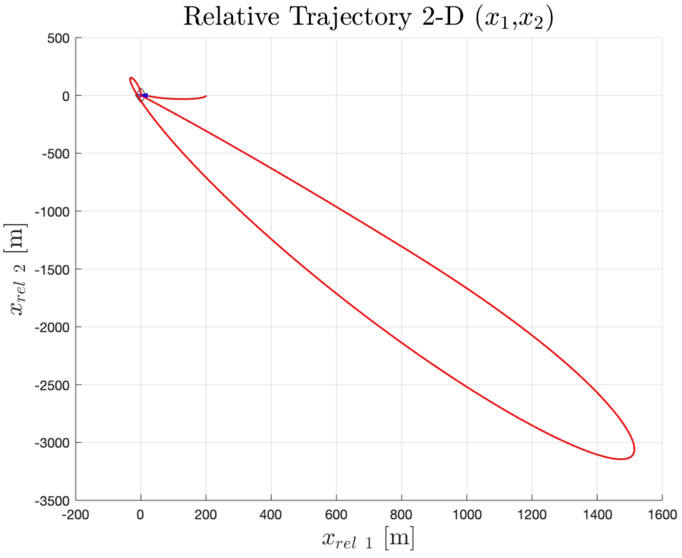

Figure 5.32: Relative position between the S/C and the debris in a 2D plane

Here, the absolute trajectory seen in a *3-D* space and in a *2-D* plane:

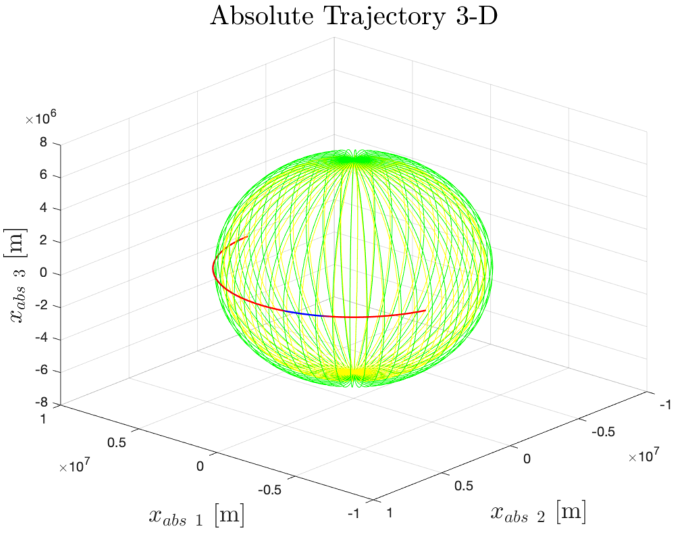

Figure 5.33: Absolute position between the S/C and the debris in a 3D space

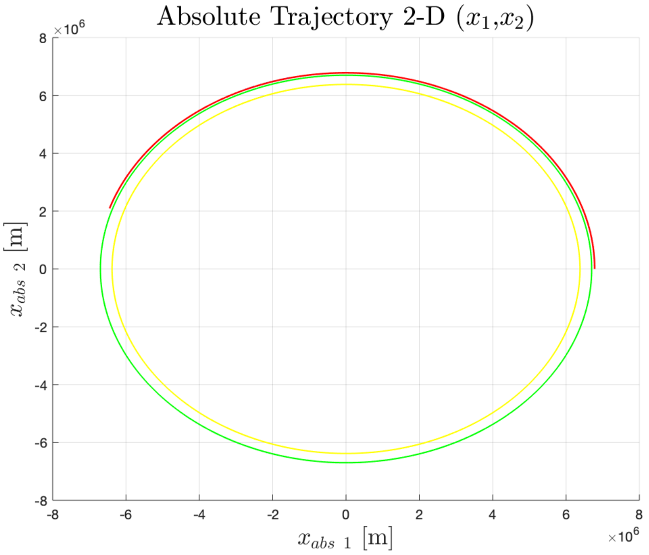

Figure 5.34: Absolute position between the S/C and the debris in a 2D plane

5.5.4 Fuel Consumption

As it can be seen from the following plot, the total fuel's mass used is:

$$m_{lost} = 545.01 \approx 545\ [kg]$$

While, the fuel used in this maneuver is:

$$m_{lost} = 545.01 - 200.58 = 344.43\ [kg]$$

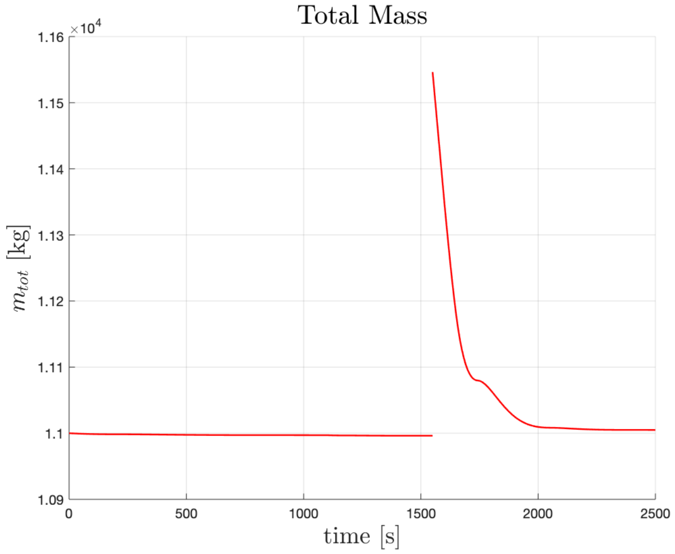

Figure 5.35: Total Mass Vs Time

5.6 Orbit-Change Maneuver

The considered simulation parameters are:

- *Drag force parameters:* $C_D = 1$, $S = 12$ $[m^2]$, $\rho_0 = 1.22$ $[kg/m^3]$, $H = 8e3$ $[m]$;
- *Other Perturbations/Disturbances:* $std(d(t)) = 1$ $[N]$;

5.6.1 Time and Delta-V

The reference point is reached after:

$$t = 5000 \ [s] \approx 1 : 23 \ [h : min]$$

While, the maneuver time is:

$$t = 5000 - 2500 = 2500 \ [s] = 41.5 \ [min]$$

$$\Delta V_5 = 267.6127 \ [m/s]$$

While the total Delta-V needed for the complete mission is:

$$\Delta V = \sum \Delta V_i = 372.8515 \ [m/s] = 0.3728 \ [km/s]$$

5.6.2 Command Input, Position, and Velocity

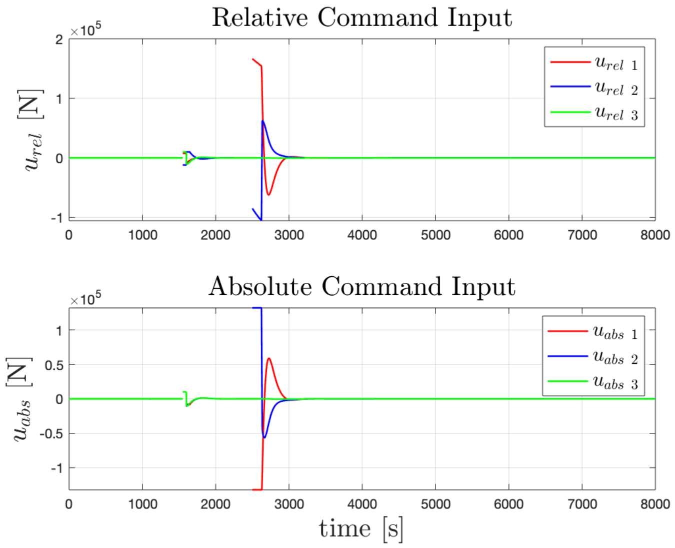

Figure 5.36: Relative and Absolute Command Input Vs Time

The constraints on the input are expressed in the GE frame. As it can be seen they are satisfied.

In the following the relative and absolute plots of the position and velocity are given:

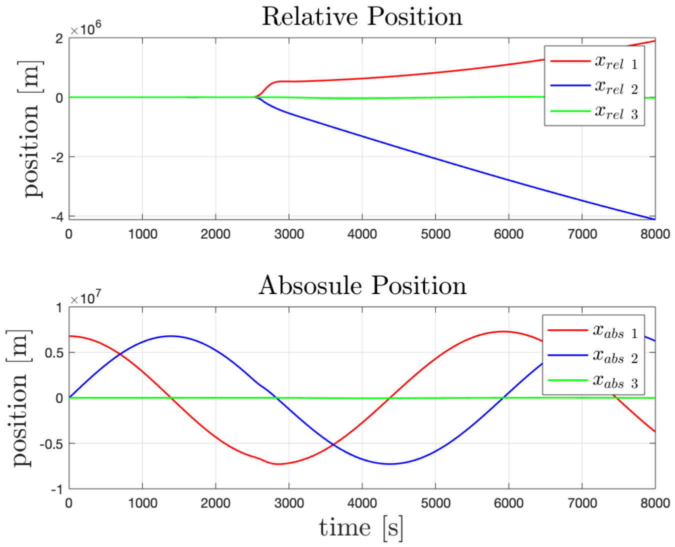

Figure 5.37: Relative and Absolute Position Vs Time

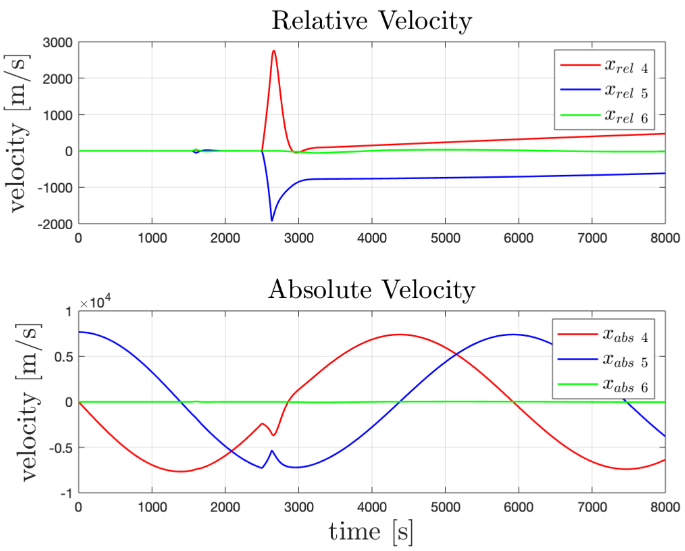

Figure 5.38: Relative and Absolute Velocity Vs Time

5.6.3 Trajectory

The relative and absolute trajectories, for this maneuver, are now presented. In the absolute ones, it can be seen that, the constraints on the state are satisfied.

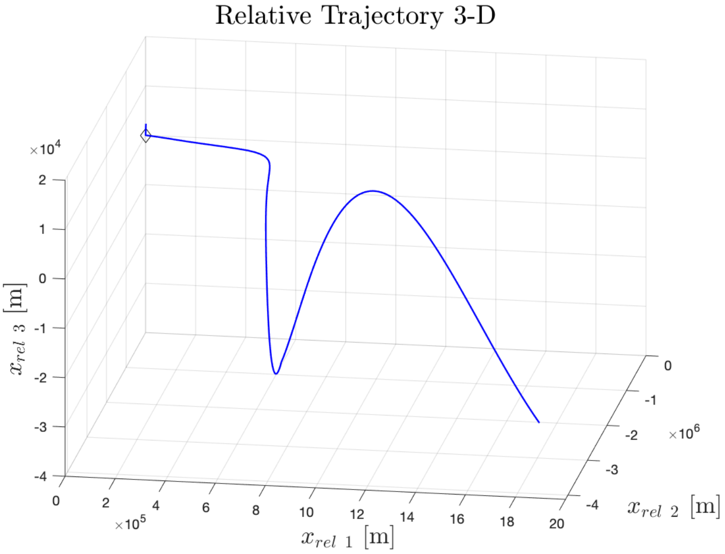

Figure 5.39: Relative position between the S/C and the debris

In the following is reported the trajectory in a *2-D* plane:

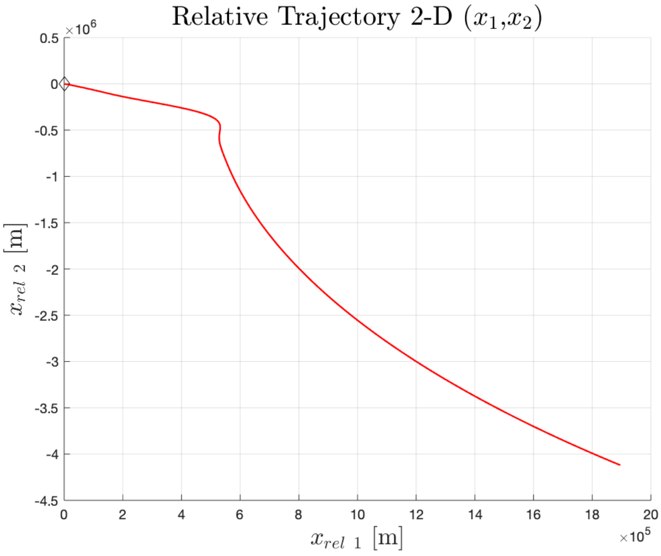

Figure 5.40: Relative position between the S/C and the debris in a 2D plane

Here, the absolute trajectory seen in a *3-D* space and in a *2-D* plane:

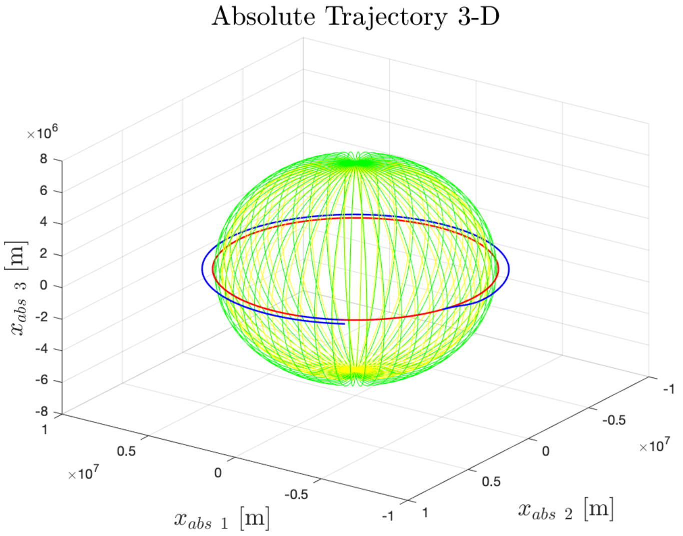

Figure 5.41: Absolute position between the S/C and the debris in a 3D space

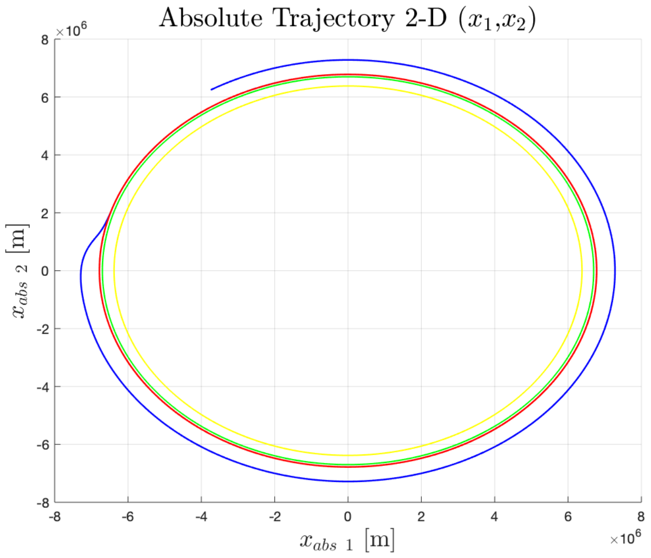

Figure 5.42: Absolute position between the S/C and the debris in a 2D plane

5.6.4 Fuel Consumption

As it can be seen from the following plot, the total fuel's mass used is:

$$m_{lost} = 9312.1 \approx 9312\ [kg]$$

While, the fuel used in this maneuver is:

$$m_{lost} = 9312 - 545 = 8767\ [kg]$$

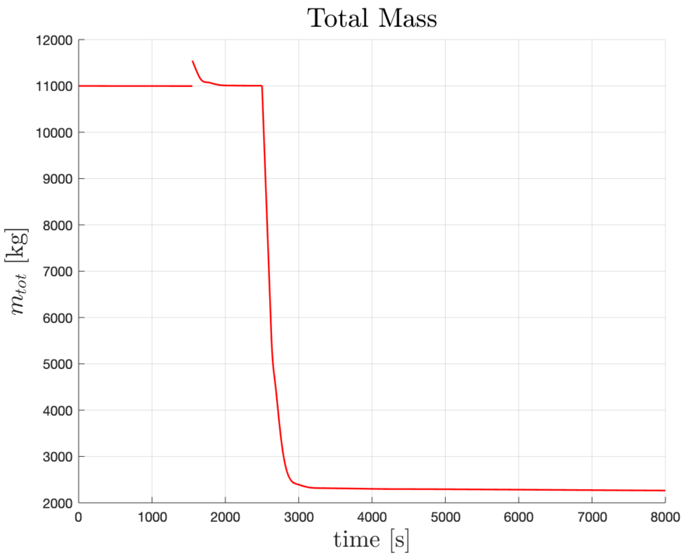

Figure 5.43: Total Mass Vs Time

5.7 Comparison Without Estimating m_2

Thanks to its flexibility, the NMPC is able to find a solution (very near to the optimal one) also in the presence of unknown estimation of the debris mass. As discussed in the previous chapter (3), the NMPC is very robust to system variations, though it is not simple to make an analytical analysis. The results taking into account are the ones related to the trajectories of the maneuver that takes place after the estimation process.

The final point is reached at the same time $t = 2500$ $[s]$.

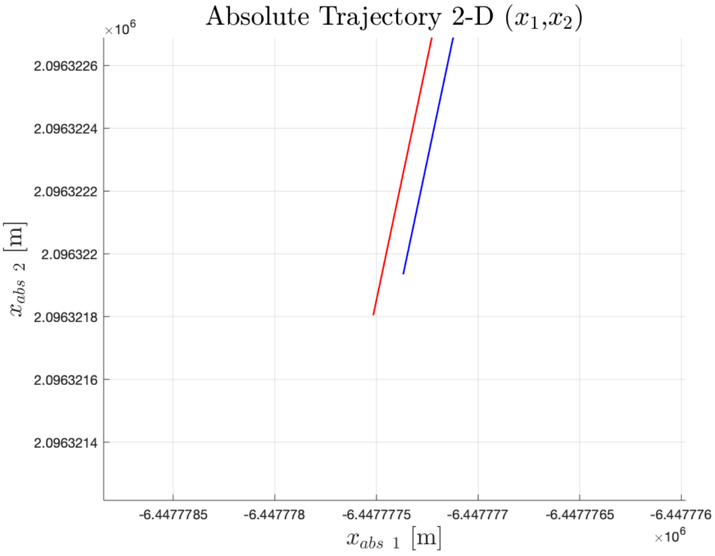

Figure 5.44: Zoomed Absolute Trajectory with m_2 Estimation

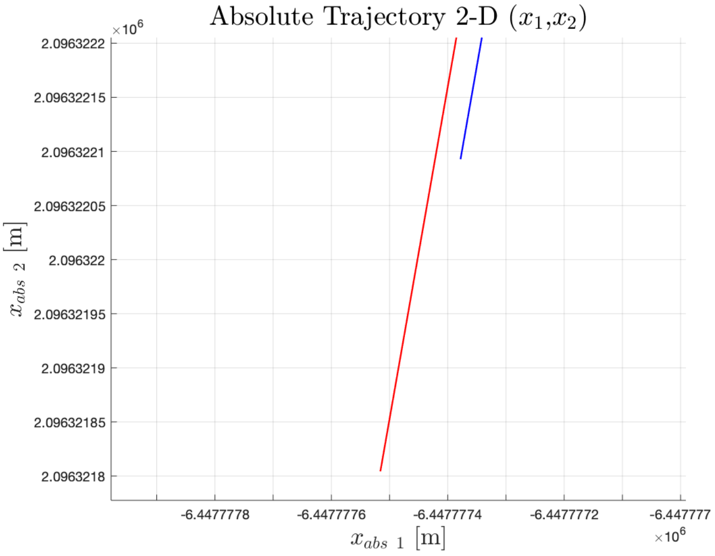

Figure 5.45: Zoomed Absolute Trajectory without m_2 Estimation

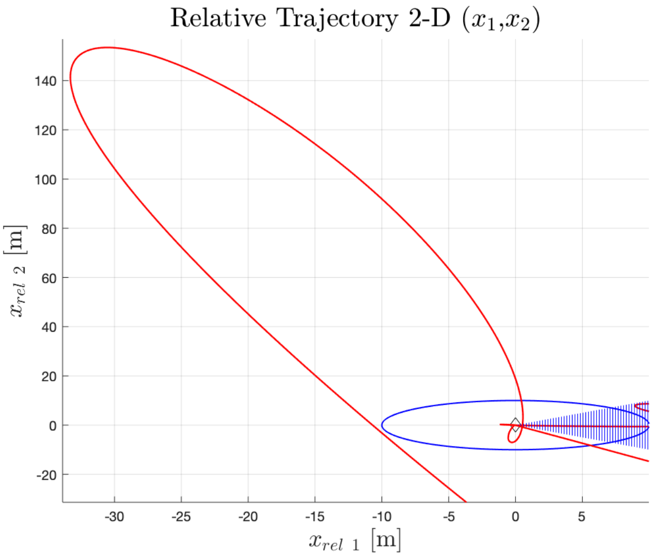

Figure 5.46: Zoomed Relative Trajectory with m_2 Estimation

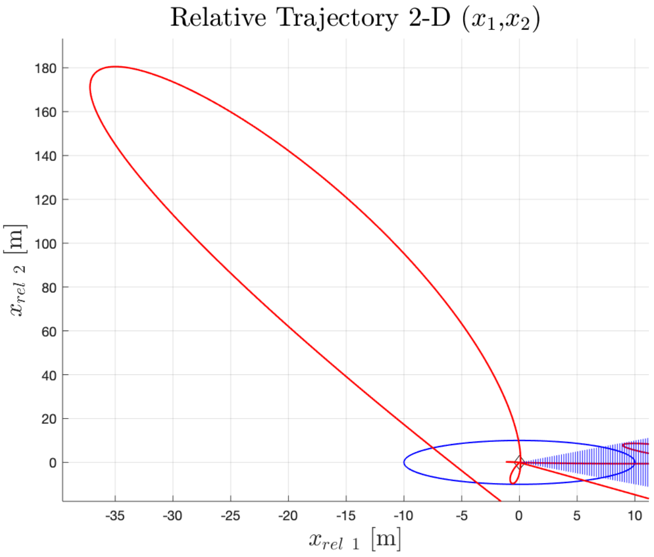

Figure 5.47: Zoomed Relative Trajectory without m_2 Estimation

The solution without the estimate value of the debris mass, inside the control model, doesn't diverges significantly from the one in which it is is considered.

Thanks to this robustness feature, the controller can be applied also when it is not possible to have a precise mass estimation. This is very important in that debris removal mission where the debris masses are too small ($m_2 \approx 0.1 \div 100\ [kg]$).

In any case, if an estimation algorithm is considered the solutions converge faster and there is a better management of the trade-off between performance and command activity. In all the cases where an high precision is needed, it is almost mandatory to use an estimation process in order to increase it and consequently avoid possible collisions with other objects. It is also possible, thanks to these considerations, to save fuel and consequently money.

Chapter 6

Conclusion

The used control methodology is seen to be good enough. Thanks to the obtained results in the space mission simulation, it is possible to see that the controller works also without an estimation process. In this way, it is always possible to obtain good trajectories. This last result shows that the robustness of this control method is high. Anyway, the mass estimation could be useful in order to increase precision and performances. The Delta-V calculated for each maneuver are found to be equal to: $\Delta V_1 = 0.1203$ $[m/s]$, $\Delta V_2 = 0.0080$ $[m/s]$, $\Delta V_3 = 52.5443$ $[m/s]$, $\Delta V_4 = 52.5663$ $[m/s]$, $\Delta V_5 = 267.6127$ $[m/s]$. While the total Delta-V needed for the complete mission is: $\Delta V = 372.8515$ $[m/s] = 0.3728$ $[km/s]$. It is possible to note that the Delta-V associated to the obit-change from the debris position to an orbit with altitude at 900 $[km]$ is not high if compared with other Space missions. Anyway, this value could be reduced if it is considered a de-orbit maneuver for the debris.

If an estimation algorithm is considered the solutions converge faster and there is a better management of the trade-off between performance and command activity. In all the cases where an high precision is needed, it is almost mandatory to use an estimation process in order to increase it. It is also possible, thanks to these considerations, to save fuel and consequently money. The analyzed estimation solutions are similar, but depending on the specific mission in which the controller has to be used, it is possible to select an *ad hoc* estimation algorithm in order to minimize possible estimation errors (coming from different type of estimators). Indeed, sometimes the estimation can be execute for interval much grater than the one used in the estimation maneuver of this mission. For this reason a more efficient estimator, that works well in long time intervals, can be chosen.

The EKF has the advantage to be an optimal observer. Indeed, it uses the measured outputs to correct all the estimated values. By using it, it is possible to increase the precision of the sensors used to collects all the state measurements. Another important consideration is that when the EKF is used, the debris mass enters in the problem as a state variable. In this way, the whole algorithm is also able to fully accomplish the GNC. While with the recursive average the mass is estimated as parameter (it is not considered as state variable).

Many approximations are made in order to construct the considered mission problem. Some of them are in the model used as plant by means of neglecting dynamics coming from different perturbation sources. The debris orbit was considered *kleperian* (described by the FR2B equations) without any type of perturbations. While, the equations used inside the controller for the precision maneuvers are an approximate version of the HCW equations. However, in this last case, the approximation can be considered good enough, since the the debris orbit was considered circular $e = 0$ (see section 2.1.8). A possible future work can be the one to modify these models and study the controller results.

6.1 Further Works

Further works can be on the using of this adaptive NMPC on different missions or also in different engineering fields. By considering both the debris mass and how it is distributed inside the volume, it is possible completely describe the system (composed by debris and S/C) in the space. In this way, an estimate of the inertial matrix is needed in order to track a desired attitude this problem is related to the so called Attitude and Orbit Control Systems (AOCS). Considering that a common problem with the debris may be on the rotation of the debris itself around a rotation axis, possible works could be done on the non-cooperative maneuvers. A possible solution could be to estimate the position of the rotation axis and keeping the debris with a robotic arm on this direction (possibly with the same, or almost the same, angular velocity of the debris). Another possible mission can be the one in which the atmospheric density is not exactly known. By means of estimation processes this value can be implemented inside the controller in order to develop a good *De-Orbiting* mission. In these years studies, like [24], are made forward detection algorithm and optimal trajectories to multiple targets. These studies can be used for debris detection purposes and to find the optimal path in order to remove multiple debris.

However, the scope of this thesis is reached. The found control algorithm can be extended in flexible way in a great number of applications.

Bibliography

[1] Wade Pulliam, *Catcher's Mitt Final Report*, DARPA technical report, 2011;

[2] C. Priyant Mark, Surekha Kamath, *Review of Active Space Debris Removal Methods*, Space Policy 47, 194-206, 2019;

[3] Jason L. Forshawa, Guglielmo S. Agliettia, Simon Fellowesa, Thierry Salmonb, Ingo Retatc, Alexander Halld, Thomas Chabote, Aurélien Pisseloupe, Daniel Tyef, Cesar Bernalg, François Chaumetteh, Alexandre Pollinii, Willem H. Steynj, *The active space debris removal mission RemoveDebris*, Acta Astronautica 168, 293-309, 2020;

[4] Weilin Wang, Lei Chen, Kebo Li, Yongjun Lei, *One active debris removal control system design and error analysis*, Acta Astronautica 128, 499–512, 2016;

[5] Camille Pirat, Muriel Richard-Noca, Christophe Paccolat, Federico Belloni, Reto Wiesendanger, Daniel Courtney, Roger Walker, Volker Gass, *Mission design and GNC for In-Orbit Demonstration of Active Debris Removal technologies with CubeSats*, Acta Astronautica 130, 114-127, 2017;

[6] Marko Jankovic, Jan Paul, Frank Kirchner, *GNC architecture for autonomous robotic capture of a non-cooperative target: Preliminary concept design*, Advances in Space Research 57, 1715-1736, 2016;

[7] Alberto Bemporad, Francesco Borrelli, Manfred Morari *Predictive Control for Linear and Hybrid Systems*, Cambridge University Press, 2017;

[8] UCS Satellite Database, https://www.ucsusa.org/resources/satellite-database;

[9] Mazzoldi, Nigro, Voci, *Fisica 1*, Edises, Moti Relativi, 72-90;

[10] E. Canuto, C. Novara, L. Massotti, C. Perez Montenegro and D. Carlucci, *Spacecraft dynamics and control. The embedded model control approach*, Butterworth-Heinemann, 2018;

[11] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control*, Springer;

[12] M. Morari, J. H. Lee, *Model predictive control. Past, present and future*, Computers and Chemical Engineering 23, 667-682, 1999;

[13] Yang Wang, Stephen Boyd, *Fast Model Predictive Control Using Online Optimization*, IEEE Transactions on Control Systems Technology 18, 267-278, 2010;

[14] Hiroaki Fukushima, Tae-Hyoung Kim, Toshiharu Sugie, *Adaptive model predictive control for a class of constrained linear systems based on the comparison model*, Automatica 43, 301–308, 2007;

[15] James Rowlings, David Mayne, *Model Predictive Control Theory and Design*, Nob Hill Pub, 2009;

[16] Karl J. Astrom, Bjorn Wittenmark, *Adaptive Control*, Second edition, Dover Publications, 2008;

[17] Girish Chowdhary, Maximilian Mühlegg, Jonathan P. How, Florian Holzapfel *Concurrent Learning Adaptive Model Predictive Control*, Advances in Aerospace Guidance, Navigation and Control Selected Papers of the Second CEAS Specialist Conference on Guidance, Navigation and Control, Springer, 29-47, 2013;

[18] James P. Nelson, Mark J. Balas, Richard S. Erwin *Model Reference Adaptive Control of Mildly Non-Linear Systems with Time Varying Input Delays*, Advances in Aerospace Guidance, Navigation and Control Selected Papers of the Second CEAS Specialist Conference on Guidance, Navigation and Control, Springer, 49-59, 2013;

[19] Veronica Adetola, Darryl DeHaan, Martin Guay *Adaptive model predictive control for constrained nonlinear systems*, Systems & Control Letters 58, 320-326, 2009;

[20] Kwangjin Yang, Yeonsik Kang, Salah Sukkarieh *Adaptive nonlinear model predictive path-following control for a fixed-wing unmanned aerial vehicle*, International Journal of Control, Automation and Systems 11, 65–74, 2013;

[21] Boguslaw Dolega, Robert Glebocki, Damian Kordos, Marcin Zugaj, *Advances in Aerospace Guidance, Navigation and Control - Selected Papers of the Fourth CEAS*, Springer, 2018;

[22] Joël Bordeneuve-Guibé, Antoine Drouin, Clément Roos, *Advances in Aerospace Guidance, Navigation and Control - Selected Papers of the third CEAS*, Springer, 2015;

[23] Raziye Tekin, Harald Pfifer, Qiping Chu, Bob Mulder, Daniel Choukroun, Erik-Jan van Kampen, Coen de Visser, Gertjan Looye, *Advances in Aerospace Guidance, Navigation and Control - Selected Papers of the Second CEAS*, Springer, 2013;;

[24] Hong-Xin Shen, Tian-Jiao Zhang, Lorenzo Casalino, Dario Pastrone, *Optimization of ADR Missions with Multiple Targhets*, Journal of Spacecraft and Rockets 55, 1-9, 2017;