Optimization based trajectory planning with adaptive SMC for space applications

Politecnico di Torino

Master course in

Aerospace Engineering

Candidate:
Pierantonio Bertuccio

Supervisors:
Prof. Elisa Capello
(Politecnico di Torino)
Prof. Jaemyung Ahn
(Korean Advanced Institute of technology)

December 13, 2023

I would like to deeply thank my supervisors.
I thank Prof. 안재명 *(Ahn Jaemyung) for his generosity in lending me his time, his patience, his experience and advice, for his warm welcome inside his labs seminaries among his students. I also thank him for inspiring and supporting throughout all this work always with a smile.*
I also thank prof. Elisa Capello, for her time, patience and lending me her expertise, for giving me the support in starting this collaboration. I also thank her for introducing me to Dott. Mancini, who kindly provided his advice and help as well.
I thank them both for making this work possible, despite the contingencies and the logistic limitations of having to work from distant parts of the world.
I would like to thank all my friends, from the most to the least important, from the oldest to the youngest. Everyone of you had a contribution to the man I am today and the man I would like to become. I'm lucky someone of you is still walking with me and I'll treasure in my heart the ones who had to part, joyfully or sadly.
I thank all my family, who gave roots to my heart and still provide love to fill it with. I thank my mother, for always being present and offering a chance for good talk. I thank my father for his unconditional support and reminding what's important. I thank my brother for being so different from me, giving chance to grow together in life and in wisdom.

Abstract

In the history of space exploration, the ability to navigate, rendezvous and interact with a target in orbit always had profound importance. As our understanding of the cosmos deepens, the demand for advanced satellite systems capable of undertaking more complex missions has grown exponentially with newer challenges upfront. The number of unmanned missions and satellites has recently risen more than ever, driving a potential revolution in satellite operations thanks to topics like space debris removal, on-orbit servicing and formation flying. These activities have recently taken some spare of the space market as they are now supported and adopted inside the space companies. ESA is developing the "Clean Space" program and has already expressed interest in starting on-orbit servicing missions to extend the life of the already orbiting satellites instead of having them replaced (with every kind of waste inferred) or having to launch more. Mission planning is a key topic of interest, including the selection and sequencing of targets, along with the optimization of transfer trajectories. The objective of this thesis is the design of a guidance algorithm, able to generate an onboard trajectory, and of an adaptive control system. This thesis is a comprehensive exploration of a 3DOF orbital simulator developed in MATLAB Simulink which aims to replicate the dynamics of a satellite engaged in a complex multi-target rendezvous mission. It consists of several key components: a system, which models the satellite's behaviour in orbit, a guidance model for path determination, a controller model for decision-making, an actuator model, and a disturbance model for orbital perturbations. The specific scenario considered involves a satellite in Low Earth Orbit (LEO) pursuing a target within the valid range of the Hills Equation. The control system's primary objective is to navigate the satellite into proximity with the target and execute rendezvous manoeuvres while accounting for uncertainties and variations common in complex missions. The results obtained show the system successfully completing the rendezvous with different mass properties and in different starting positions. An investigation was conducted to determine the influence of the guidance parameters on the resulting trajectories and some correlations were found with the time and propellant involved in the mission.

Contents

List of Figures

List of Tables

Introduction

In unmanned missions, the ability to make real-time decisions on board is paramount, ensuring autonomy and adaptability in dynamic environments. This capability not only enhances operational efficiency but also mitigates the impact of communication delays, enabling spacecraft to respond swiftly to unforeseen challenges and optimize mission success. As consequence of this, automation and robotics have experienced significant advancements and the technological developments have expanded the capabilities, the efficiency, and the safety of space missions. This rapid development of automated satellites is also due to the rapid development of autonomous and efficient control systems. They are vital for missions that require spacecraft to autonomously approach and connect with other modules, such as the International Space Station or future space habitats. The recent NASA mission "CAPSTONE" is one of the last efforts to test a GNC system that doesn't rely on the ground support but will exchange information with the orbiter that monitors lunar position [26]. Several contemporary studies [3, 4, 22] have also been conducted on the investigation of the multi-target rendezvous mission concept. Multi-target rendezvous missions are for sure a strong option when it comes to space debris removal, due to their efficiency and cost effectiveness. Mission planning is a central area of interest, including the selection and sequencing of targets, along with the optimization of transfer trajectories.

Spacecraft's autonomy is mostly given by his control system. When it comes to autonomous path planning there are several works that take in consideration the use of path generation algorithms such as A* and RRT* algorithms. The concept behind these methods is to explore and sample the state space of the controlled

(a) Esa Clean Space

(b) Nasa Capstone

system and to estimate the reaching and remaining cost in the approach toward the target. Based on these calculations the "optimal path" is then traced. In [12] the A* search algorithm is used and implemented for spacecraft rendezvous missions using an optimization control as heuristic/criteria for the optimal path selection. Another study [14] uses RRT* instead and couples it with a genetic algorithm to have the optimal path obtained. The study in [2] couples the idea of the APF (artificial potential fields) with an already generated path, exploiting the APF nature of obstacle avoidance and using an additional force definition based on the distance from the path. All these studies centre their focus on the optimization and constraints definition and subsequent application on already established path generation algorithms. Most of the other studies reviewed have the same framework, with a focus on the efficiency of the sampling. Sometimes the path is calculated and then used as the initial guess for some nonlinear programming. Some techniques involve the use of convex optimization. A lot of optimization-based frameworks are reliable, don't need a starting guess, and have been tested in real-world situations. Lately, they've been used and studied for planning paths during orbital rendezvous and close approaches in space.
In [24] and [27], the trajectory planning problem is transformed in a SOCP problem, by taking advantage of the linear dynamics provided by CW equations. The problem is posed without the need for an initial guess, and it is easy to add constraints regarding the cone of approach, keep out zones and control bounds without escaping the SOCP formulation. Ref.[27] also proposes the same technique used in an arbitrary rotating frame, more suitable for tumbling and uncontrolled satellites and objects. It should be noted that this formulation requires the problem to be time fixed, so the total time required for the rendezvous and for the obtained trajectory is considered given a priori. There are also some examples, as in [32], where problems are posed in a general NLP formulation and then solved through genetical algorithms.
Finally, another approach reviewed and considered is based on model predictive concept. Model Predictive Control (MPC) operates by repeatedly predicting a system's future behaviour using a mathematical model, then optimizing control inputs to steer the system towards desired outcomes. It iteratively refines predictions and adjustments to achieve optimal performance, inspiring to the model predictive applications, where the concept of "horizon" is used, which means that there is a limited and determined amount of time for the prediction, . The horizon keeps moving together with the system and it's also useful to balance the computational burden of the algorithm. Refences [9] and [1] take advantage of some predefined trajectory to be followed and develop a technique based on model predictive methods to formulate an optimal trajectory to follow and reach the predefined one. To predict the system behaviour, since in this case no necessary control is being computed, it is necessary to consider a closed-loop system for the system prediction. This allows for different initial states to be take in account, but once again, a predefined path must be already defined to run the algorithm.
At last, another most critical design topics when it comes to on board computing is the computational burden for the on-board computer. Practical situations lead to

design trade-offs and choices to make, unless there is a way to overcome them.

Sliding Mode Control (SMC) is a widely adopted control strategy in the space industry due to its robustness, precision, and adaptability. SMC's ability to handle uncertainties, provide accurate control, respond rapidly to changing conditions, and accommodate nonlinear systems makes it suitable for various space missions. As the SMC controllers are not new, different types of algorithms based on it have been developed in the years, trying to reduce the limits of this control method. Some algorithms try to eliminate the problem of the discontinuity [28], others are focused on the time needed to the system for converging to the desired state [6, 30, 25]. Even though SMC is already a versatile controller, some of the latest investigations also include solutions and methods for erasing the reaching phase. This means reducing the time frame inside which the system is the most vulnerable to uncertainties and disturbs and thus making it more adaptable to situations. For the purpose of this work, the implementation of the control algorithm has been made under the assumption of a multi-target rendezvous mission engaged satellite and so with the intention to provide the satellite of a controller able to withstand changes in the satellites mass properties, consequence of a serviced satellite or captured debris.

This thesis aims to create a guidance algorithm, capable of generating an onboard trajectory, and an adaptive control system. The focus of the development is to enhance the autonomy of the controlled system during its mission. The proposed guidance algorithm is adaptable and open to future refinements for real-world applications. The goal is to establish a versatile and adaptable framework for trajectory planning across various design environments. The algorithm, created and tested in Simulink simulations, reflects ideas gathered from the literature review and it is designed to be flexible.

Overview

This report is divided in sections, each sections illustrating a different aspect of this work. In the first chapter the main mathematical models and relationships are illustrated and reported as well as the main assumptions behind them. The second chapter will explain the starting points and the development of the guidance algorithm, the main challenges and steps taken to front them. The third chapter will be a discussion on the control strategy and the two main pieces composing it will be explored. The fourth chapter will provide the results and the observations made in this study, the most interesting correlations found will be found and the main objectives will be given a verification. In the appendix it is possible to find every parameter used in the simulations.

Chapter 1

Models and reference system

In this chapter all the components of the simulator not related to the control system are illustrated with the main hypothesis and assumptions stated and explored.

1.1 Reference System

The satellite's dynamics is described in the LVLH frame, also known as the Local Vertical Local Horizontal frame. It is one of the most used coordinate systems used in aerospace engineering and spaceflight dynamics to describe the orientation and position of a spacecraft or satellite relative to another body under the influence of a much greater celestial body. This system is linear approximation of the space near the target and it's fixed on the target itself, moving with it along the orbit.
In the LVLH frame, the "Local Vertical" axis points radially outward from the central body (typically Earth) toward the spacecraft. The "Local Horizontal" axis is perpendicular to the local vertical axis and lies in the plane tangent to the central body's surface. The third axis, often called the "Local Nadir" axis, completes the right-handed coordinate system.
To better visualize it, in the figure above it is possible to see the orientation of the axes. The X axis is also called the V-bar axis, it represents the along track direction of motion or, simpler, the direction towards the satellite moves. The Z axis is also called R-bar and the Y axis completes the reference system, representing the across track orbital direction, it's also called H-bar.

1.2 System Dynamics

The satellite's dynamics is described with the Clohessy-Wiltshire equations, also known as the Hill-Clohessy-Wiltshire equations or the Clohessy-Wiltshire-Hill equations, which are a set of linearized differential equations used to describe the relative motion of two objects in space, typically a spacecraft (chaser) and a target object (e.g., satellite or space station) in a near-circular orbit. These equations are widely used in orbital mechanics and relative navigation for tasks like rendezvous and docking. They are derived from the linearization of the equations for the relative motion

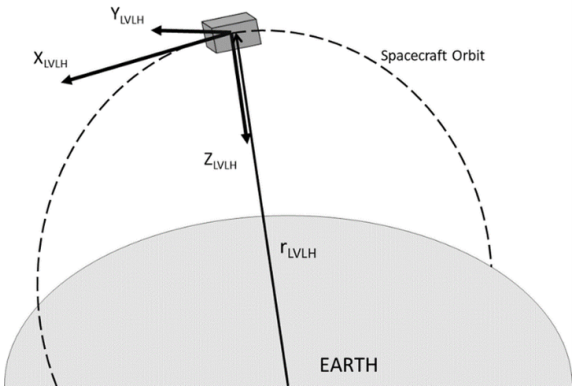


Figure 1.1: LVLH frame

of two objects in space. The equations are named after their developers:

- James R. Wertz - who is credited with the earliest work on these equations.
- George S. Clohessy - who, in 1960, published an important paper introducing these equations.
- Robert S. Wiltshire - who, in 1960, also published a similar paper independently from Clohessy, and his work is often combined with Clohessy's when referring to the equations.

The Clohessy-Wiltshire equations are a linear approximation that simplifies the problem of relative motion in space. They are applicable when the relative distances between the chaser and the target are small compared to the radius of their orbital paths, and when both objects are in near-circular orbits. The equations describe how the relative position and velocity of the chaser change with time as it manoeuvres to reach or maintain a desired relative position with respect to the target.
These equations provide a valuable tool for mission planning and control in scenarios where precise relative motion between two objects in orbit needs to be managed, such as during rendezvous and docking operations, space inspections, or satellite servicing missions. They are also foundational for the development of control algorithms for automated systems used in space missions. The set of differential equations can be expressed as follows[20]:

$$\ddot{x} = 2n\dot{z} + \frac{F_x}{m_c} + f_x \tag{1.1a}$$

$$\ddot{y} = -n^2\dot{y} + \frac{F_y}{m_c} + f_y \tag{1.1b}$$

$$\ddot{z} = -2n\dot{x} + 3nz + \frac{F_z}{m_c} + f_z \tag{1.1c}$$

Where the $\ddot{x}, \ddot{y}, \ddot{z}$ are the acceleration in the LVLH frame, $\dot{x}, \dot{y}, \dot{z}$ are the speed and x,y,z are the position terms. The mass of the system is represented by m_c and

n is the orbital rate for circular orbits $\sqrt{\frac{\mu}{a^3}}$. The F terms represents external forces applied on the system, these include the thrusters' actuation and the disturbs that affect the satellite and its orbit, the f terms represents accelerations due to external disturbances.
It shall be noted that the set of linear time varying differential equations in (1.1) represents the general system suitable for]any arbitrary relative trajectory between a chaser spacecraft and a target spacecraft, both under the influence of a central gravity field.
The motion represented by (1.1) can be actually split in two different dynamics: the in-plane dynamics and the out-of-plane dynamics.
This is easy to notice when looking at $(1.1a, 1.1c)$ and $(1.1b)$ separately since the latter just represents the differential equation describing the harmonic oscillator behaviour in the absence of a dumping term while along x and z axis the dynamics are coupled, as they influence each other. If a constant input force is considered they can be integrated to obtain a state transition matrix of the system (here assuming no force applied for simplicity)[20]:

$$\Phi(t) = \begin{bmatrix} 1 & 0 & 6(\omega t - \sin(\omega t) & -3t + \frac{4}{\omega}\sin(\omega t) & 0 & \frac{2}{\omega}(1 - \cos(\omega t) \\ 0 & \cos(\omega t) & 0 & 0 & \frac{1}{\omega}\sin(\omega t) & 0 \\ 0 & 0 & 6\omega(1 - \cos(\omega t) & -3 + 4\cos(\omega t) & 0 & 2\sin(\omega t) \\ 0 & -\omega\sin(\omega t) & 0 & 0 & \cos(\omega t) & 0 \\ 0 & 0 & 3\omega\sin(\omega t) & -2\sin(\omega t) & 0 & \cos(\omega t) \end{bmatrix} \quad (1.2)$$

In the simulator, the system 1.1 have been implemented by assuming that the mass is constant even if thrusters actuation would normally deprive the system of some propellant mass. This is because the variation of mass taken into account when dealing with the capability of the system to adapt to inertial variations is higher than the one consumed in a single approach.

1.3 Satellite Model

To simulate a plausible system, the DEMETER satellite, from CNES, was taken as reference. DEMETER (Detection of Electro-Magnetic Emissions Transmitted from Earthquake Regions) is a satellite mission operated by CNES (Centre National d'Études Spatiales), the French space agency. Launched on June 29, 2004. It is a scientific satellite designed to study the electromagnetic and plasma phenomena in Earth's ionosphere and their possible connection to seismic activity and natural phenomena on the Earth's surface, particularly earthquakes. In this study, the system will be assumed a micro-satellite of mass about 130kg [10], the chassis shape is simple like the original, its dimensions are 60x85x110 cm.

The satellite's propulsion system is composed by hydrazine fuelled on/off monodirectional thrusters. Each thruster has a constant output and can only be activated or deactivated, with no option for modulating the thrust. In other words, each thruster can deliver either its full thrust capacity when turned on or produce no

Figure 1.2: The DEMETER Satellite photo in flight [10]

force when turned off.
Usually, thruster's location on the chassis influences the coupling between the position dynamics and attitude dynamics. This is often an important design topic as it includes some trade-offs, for the purpose of this study the thrusters will be assumed installed in such location and manner that no resulting torque is produced during an actuation and the whole system is able to provide accelerations in both directions on every axis. To better visualize this, figure 2.31.3 shows an idea of how this can be accomplished theoretically. Inside the simulator, a Pulse-Width Pulse-Frequency (PWPF) modulator is implemented as a model for the thrusters, given the discontinuous nature of their control action. Details about the parameters used can be found in the appendix.

1.4 Disturbances

This section explains about external factors that influences and can impact the relative motion of spacecraft during proximity operations. These disturbances can arise from a multitude of sources, including gravitational variations, radiation pressure, micrometeoroids, and other external forces. Managing these disturbances is a fundamental aspect of ensuring the safe and precise execution of rendezvous maneuvers, as they can affect the spacecraft's ability to attain and sustain the desired relative position with respect to the target object. Consequently, a comprehensive understanding of disturbances and the development of effective strategies for their control are pivotal for the success of rendezvous missions in space.

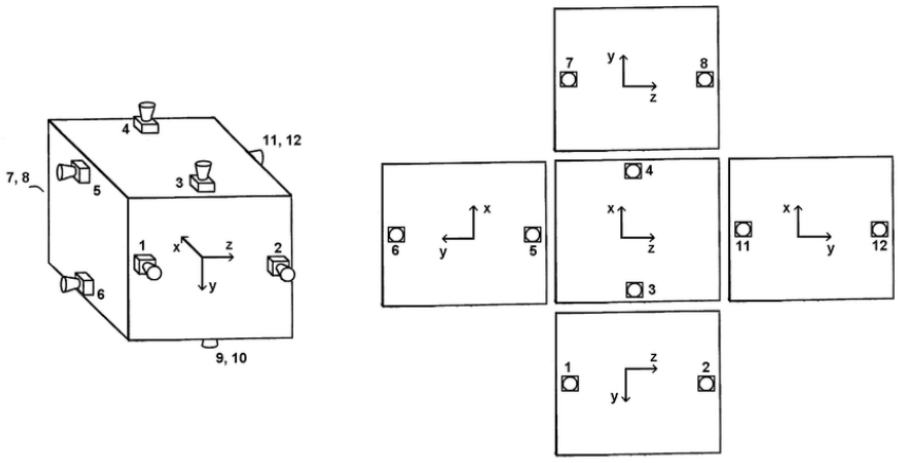


Figure 1.3: Thrusters' locations example

The fig.(1.4) represent the variation of the disturbance magnitude with the altitude of the satellite. After the primary gravitational forces, the zonal harmonic potential terms are the most dominant. The J2, or Earth's oblateness, effects are approximately three orders of magnitude smaller than the primary gravitational forces. However, these effects still have a notable impact on the perigee and nodal positions of an orbit. When it comes to planning operations in low Earth orbit (LEO), such as rendezvous strategies, the influence of higher-order zonal harmonics is somewhat less significant, but it can result in noticeable long-term variations in the orbital elements [19].
The forces acting on a spacecraft due to its interaction with the Earth's surface, such as drag and solar radiation pressure (SRP), depend on the spacecraft's area-to-mass ratio. Additionally, it's important to note that the behavior of the drag force is influenced by the level of solar activity. In this context, a moderate solar activity level was chosen. However, it's worth mentioning that, especially at altitudes around 500 km, the drag acceleration during periods of high solar activity can be up to an order of magnitude greater than during solar minimum (see [1.4.1]). In this study, only the atmospheric drag and the *J2 effect* where considered and implemented in the simulator.

1.4.1 Atmospheric drag

Space is for definition something that stays outside the atmosphere but it is well known that some remaining molecules still remains although very rarefied. Atmospheric drag, primarily composed of interactions with this traces of particles and gases, introduces resistance that can alter the trajectory and orbital parameters of satellites and spacecraft. The drag disturbance can be represented by

$$F_D = \frac{1}{2}\rho V_x^2 C_D S$$

where the $V_x = \omega r$ is the orbital velocity, C_D is a drag coefficient and S is the cross-section area of the body. The accuracy of the atmospheric density (ρ) is the most

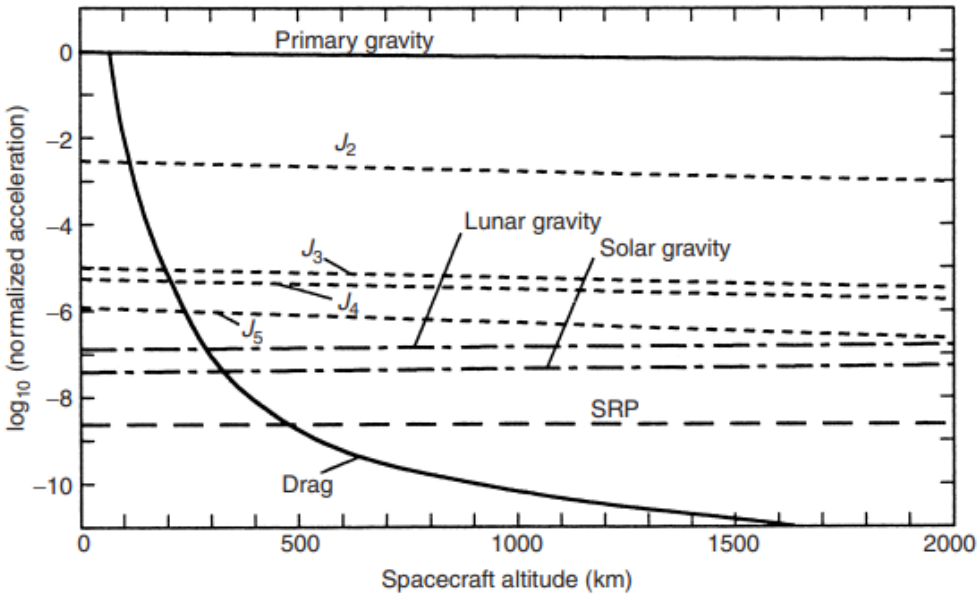


Figure 1.4: Altitude vs. Disturbance Magnitude [19]

challenging aspect to determine in this equation. The density of the atmosphere at a specific altitude is closely tied to the atmospheric temperature. On the Sun-illuminated side, the atmosphere tends to expand, causing denser regions to ascend to higher altitudes. As a result, the density at a given orbital height is not uniform. It increases on the illuminated side of the orbit due to the "solar bulge," and conversely, it decreases on the opposite side. However, the impact of the solar bulge on the relative motion between the chaser and target spacecraft is relatively small, primarily because both spacecraft are subject to the same effect.
Additionally, as this effect follows a periodic pattern with each orbit, for manoeuvres with a transfer time spanning one orbit, the influence of the solar bulge is largely averaged out, mitigating its impact. For the purposes of this study the atmospheric density will be kept constant and with a value from [18]. The drag coefficient will be taken constant [18] as well and the front section S of the satellite will be assumed constant and determined.

1.4.2 J2 Pertubation

Another source of disturbance for objects in orbit is due to the Earth's gravitational field. Geopotential effects are a consequence of the planet's non-spherical shape and uneven mass distribution. The gravitational forces acting on objects in space are not solely directed towards the center of Earth; they can have components both within and outside the orbital plane. Consequently, these forces vary over one orbital revolution, leading to changes in the orbital parameters.
To model these gravitational forces, scientists approximate Earth's gravitational potential with a mathematical function that includes various terms. The key com-

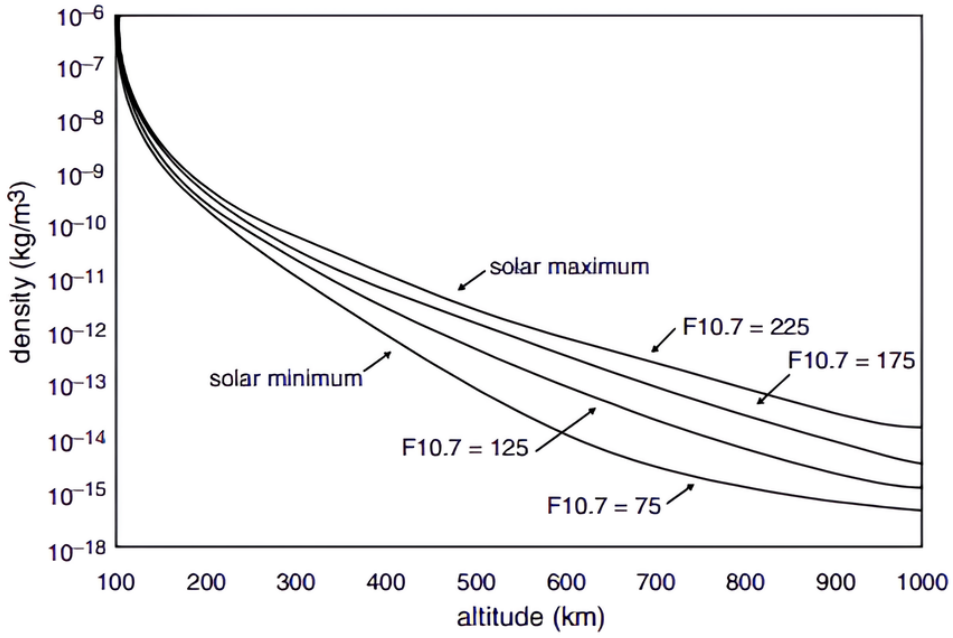


Figure 1.5: Density vs. altitude at various levels of solar flux

ponent is the Earth's oblateness, primarily represented by the second harmonic of the Earth potential, denoted as J2. The J2 coefficient is exceptionally significant, being over two orders of magnitude larger than all other coefficients. This pronounced effect is often referred to as the J2 effect and plays a fundamental role in understanding and predicting the behaviour of objects in near-Earth orbits. While the gravitational potential function is approximated using harmonic coefficients, such as J2, and Legendre polynomials, more comprehensive models can incorporate additional terms like sectoral and tesseral, which depend on longitude and are generally of lower importance, especially for shorter-duration rendezvous missions.
From [20]:

$$\begin{aligned} f_x &= \frac{3\mu J_2 R_e^2}{2r^4}\left(3\sin^2 i \sin^2(\omega+\theta) - 1\right) \\ f_y &= -\frac{3\mu J_2 R_e^2}{2r^4}\sin 2i \sin(\omega+\theta) \\ f_z &= -\frac{3\mu J_2 R_e^2}{2r^4}\sin^2 i \sin(2(\omega+\theta)) \end{aligned} \tag{1.3}$$

Where J_2 is the module of the second geopotential harmonic, R_e is the Earth Radius, μ is Earth's gravitatinal constant and r is the distance of the satellite from the center of the planet. There are some other terms involving the true anomaly θ and the orbital inclination i that makes the forces periodic and constantly varying. The trigonometry terms can be considered higher order terms when the inclination is small enough, so it is possible to neglect the R-bar and H-bar components, leading to a constant force of disturb along the orbit motion.

In this study the J2 effect will be assumed producing an acceleration like

$$F_x = -3\mu J_2 \frac{R_e^2}{r^4}$$

Chapter 2

Optimization-Based Guidance Algorithm

The guidance algorithm is developed and implemented based on the optimization. It solves an optimal trajectory problem and generates a trajectory inside a time range called *horizon*. The development of the guidance algorithm aimed to the combination of some common ideas (see *Introduction*). In most of the formulations revised in the literature, fixed time problems are formulated in a convex way with a rendezvous time given a priori. In the aim to update this feature, the idea of *discretizing* the mission in multiple little *horizons* was tested. In this way, the system gains autonomy and chooses its own pace to reach the target based on the optimal trajectories generated. Another advantage of using horizons (which is a typical concept of Model Predictive Control, MPC) is the reduction of computational effort when more precision is required and a whole trajectory has to be computed, giving higher reaction time to eventual obstacles or when in proximity of the target. The formulation is a quadratic programming type. The cost function is quadratic and represents a certain potential function for minimizing the algorithm. Although the actual system is equipped with on/off thrusters, the formulation is relaxed and assumes the input variables are bounded but continuous. This way, we can address the problem without using the integer program and a more complex solution procedure. The trajectory tracking task will be given to the controller, that also can take in account system uncertainties and external disturbances unlike in the optimization problem, saving unwanted and unnecessary complexities.

2.1 Problem Formulation

The problem is solved as a parameter optimization problem as this method keeps getting more success in the recent research. This approach permits easier comprehension and visualization of the optimization problems, unlike the indirect methods. The trajectory is discretized in the time, the number of steps and the step length can be tuned, allowing to change the horizon length. The problem is formulated as a quadratic programming, which brings assurances to the problem convergence and global optimality. It also helps to reduce the computational effort and the erasing

need for an initial guess. However, the formulation is easily convertible to a Second Order Cone Programming, as in the ref.[24, 27].

$$\min_{s,u} J = \frac{1}{2} \begin{bmatrix} \boldsymbol{s} \\ \boldsymbol{u} \end{bmatrix}^T H \begin{bmatrix} \boldsymbol{s} \\ \boldsymbol{u} \end{bmatrix} \tag{2.1}$$

$$s.t. \quad \begin{bmatrix} \boldsymbol{x}_{i+1} \\ \dot{\boldsymbol{x}}_{i+1} \end{bmatrix} = \boldsymbol{\Phi}(\Delta t, 0) \begin{bmatrix} \boldsymbol{x}_i \\ \dot{\boldsymbol{x}}_i \end{bmatrix} + \int_0^{\Delta t} \boldsymbol{\Phi}(\Delta t, \tau) \boldsymbol{B} \boldsymbol{u}_i d\tau, \quad \forall i \in [0, N] \tag{2.2}$$

$$\boldsymbol{s_0} = [x_{live}, y_{live}, z_{live}, \dot{x}_{live}, \dot{y}_{live}, \dot{z}_{live}] \tag{2.3}$$

$$|\boldsymbol{u_i}| \leq U_{max} \tag{2.4}$$

$$|\dot{\boldsymbol{x}}|, |\dot{\boldsymbol{y}}|, |\dot{\boldsymbol{z}}| \leq MaxSpeed \tag{2.5}$$

where $i \in [0, N]$ and represents the current time step, N represent the total number of time steps in the problem and $\boldsymbol{s}_i$ represents the state vector in a determined time step i

$$s = [s_0, s_1, s_2 ... s_{N-1}, s_N]^T, \quad s_i = [x_i, y_i, z_i, \dot{x}_i, \dot{y}_i, \dot{z}_i]^T$$

so that 2.2 can be rewritten as

$$\boldsymbol{s}_{i+1} = \boldsymbol{\Phi}(\Delta t, 0) \boldsymbol{s}_i + \int_0^{\Delta t} \boldsymbol{\Phi}(\Delta t, \tau) \boldsymbol{B} \boldsymbol{u}_i d\tau, \quad \forall i \in [0, N] \tag{2.6}$$

In 2.3, $\boldsymbol{s}_0$ represents the first state vector, which is to be equally constrained to the current live position, the start of the generated trajectory. Here $[x_{live}, y_{live}, z_{live}, \dot{x}_{live}, \dot{y}_{live}, \dot{z}_{live}]$ represents the current state vector passed to the guidance algorithm. Finally, $\boldsymbol{u}_i$ is used to indicate the control vector in a determined time step

$$\boldsymbol{u} = [\boldsymbol{u_1}, \boldsymbol{u_2} ... \boldsymbol{u_{N-1}}, \boldsymbol{u_N}]^T, \quad \boldsymbol{u_i} = [u_{xi}, u_{yi}, u_{zi}]^T$$

and so the vector $\begin{bmatrix} s \\ u \end{bmatrix}$ in 2.1 represent the whole design variables vector to be optimized.

The last constraint (2.5) can be added arbitrarily inside a certain operative range, in this case it actually limits the system speed when it is inside the cone of approach range. As a matter of fact, using optimization allows to add constraints whenever needed, so that the algorithm can be easily updated.

2.2 Cost Function

Despite the simple structure of the cost function a sort of weight scheduling can be done by changing them according to the situation. The whole design is reduced to the definition of the H matrix, if a diagonal matrix is considered, weights determination is a straightforward result of some tuning and intuition. Using this kind of approach assures every variable can be minimised independently and versatility, and different importance in the can be assessed as well. For example, it could be required to allow more motion along a certain axis in the first seconds of the trajectory, but

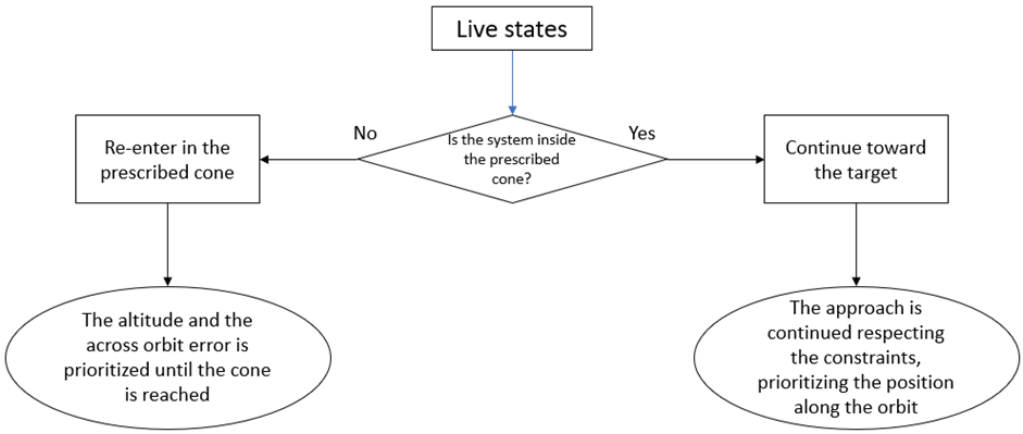


Figure 2.1: Cost function decision making example

then the system is asked to penalize that motion the more the end of the horizon is near.

$$H = diag([k_{x1}, k_{y1}, k_{z1}, k_{\dot{x}1}, k_{\dot{y}1}, k_{\dot{z}1}, k_{x2}, k_{y2}, k_{z2}, k_{\dot{x}2}, k_{\dot{y}2}, k_{\dot{z}2} \dots k_{xN}, k_{yN}, k_{zN}, k_{\dot{x}N}, k_{\dot{y}N}, k_{\dot{z}N}])$$

The weighting of the cost function is tuned and scheduled based on the current distance from the target and considering the desired type of manoeuvre, the system dynamics and actuation power. This means that according to the current situation different behaviours will be adopted by the system to achieve the mission, using the cost function as guide.
For example, if the system is near enough to must be inside the cone of approach, but for any reason this is not the current situation, the cost function has been designed to lead to the system inside the cone of approach before letting the system continue the approach.
Some commonly used solvers assume a classical formulation (i.e., $\frac{1}{2}xH^Tx + f^Tx$), requiring H and f as inputs. For simplicity, to overcome the limit of only having the origin vector as the minimum point of the function and so as the target of the algorithm, a simple method was applied to modify the desired ending states. It is possible to traslate the state space as much as the desired final states for the algorithm to generate a feasibile trajectory by solving the same problem.

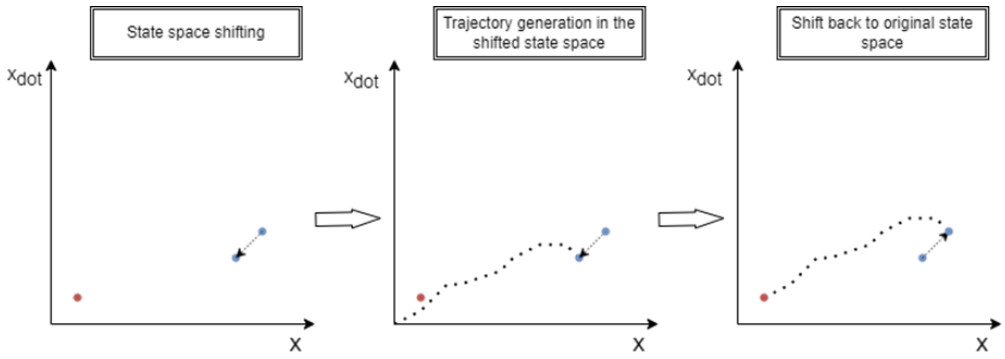


Figure 2.2: State space shifting example: the red is the target state, the blue dot is the current one. If we traslate the current state of the exact target's distance from the origin the trajectory generated will be ending perfectly on the spot once shifted back

Once the problem has been solved the state space can be shifted back to obtain the actual resulting trajectory. To help with the visualization of this, it is possible to look at this method as an error based optimization, since now the design variables are the error of the current states with respect to the desired ones.

2.3 Constraints

Since the algorithm is based on an optimization problem, it is also possible to plan how and when the constraints are going to be active and so modify his behaviour according to arbitrary requirements. The control variable is bound due to the limits of the thrusters and this constraint is active throughout all the mission. Something different can be done with constraints on the approaching speed, which can be added when desired (e.g. inside a certain range) to assure safety and making the system more easy to handle. This is exactly the case shown in the results, were the speed of the system is restricted once the the system is in the cone of approach range.
A common way is also to add some conical constraints during the cone of approach phase, this surely guarantees the computed trajectory to be feasible (if a correct solution is found) but at the same time that introduce a quadratic constraint to the formulation making it a little more complex than it was before.

$$||x_i, y_i, z_i|| \cos(\alpha_{cone}) \leq [-1, 0, 0][x_i, y_i, z_i]^T \tag{2.7}$$

with α_{cone} being the cone aperture and $[-1, 0, 0]$ being here an example of semicone axis versor, which in general is dipendent from the mission requirements and also the coordinate rappresentation used.

2.3.1 Dynamical constraints

The Clohessy-Wiltshire equations 1.1 are a set of linear differential equation, so linear dynamical constraints can be derived instead of using some transcription

method like trapezoidal rule or Simpson rule. Let's consider the definition of the state transition matrix (1.2) and the relationships

$$\Phi_B = \int_0^{\Delta t} \Phi(\Delta t, \tau) \boldsymbol{B}\boldsymbol{u} d\tau \tag{2.8}$$

$$x(t) = \boldsymbol{\Phi}(\Delta t, t_0) x(t_0) + \int_0^{\Delta t} \Phi(\Delta t, \tau) \boldsymbol{B}\boldsymbol{u}(\tau) d\tau \tag{2.9}$$

if the control variable is assumed constant inside each time step duration we can write

$$x(t) = \boldsymbol{\Phi}(\Delta t, t_0) x(t_0) + \Phi_B \boldsymbol{u} \tag{2.10}$$

and since the time is fixed the time steps duration Δt is also fixed, the expression in (2.10) can be converted to linear dynamical constraints for every time step, as in (2.2). Finally, it also possible to have a matrix form of them and build a matrix and a vector representing linear inequality constraints (as requested for *quad_write*, see next section) allowing for specialized solver to operate faster.

2.3.2 Final Horizon

When the system is near enough to the target position, a criterion is applied to add an additional constraint and make sure the target position is reached. The criterion is once again arbitrary in general, inside the simulation a simple one is used considering the actual speed and the remaining distance was implemented in such a way that if the system is near enough and fast enough the final point of the computed trajectory is constrained to be the desired one. This is of course more effective when the horizon is not so long and the situational awareness of the system is restricted. When longer horizons are considered the algorithm has more "time" to see in the future and realise a target or an object is near, making the right adjustments.

2.4 Solving & Output

Once the optimization problem is solved the trajectory generated is discretized in points in every time steps. Once the trajectory is generated, it must be passed to the controller but the shape of the solution is not convenient. One way to solve this is interpolating the points of the trajectory and passing it to the controller continously, so that the update rate can be arbitrary and the whole process is not influenced. In the simulator the interpolation is conducted using a spline algorithm built-in in Simulink.
The solver's choice can be really meaningful when dealing with optimization problems inside real and practical situations. The time needed, the computational burden and the accuracy of the solution are all topics of reference. One of the main effort of this study is to give extended tunability to the horizon length (and so number of optimization variables) and the guidance update rate trying to allow the best

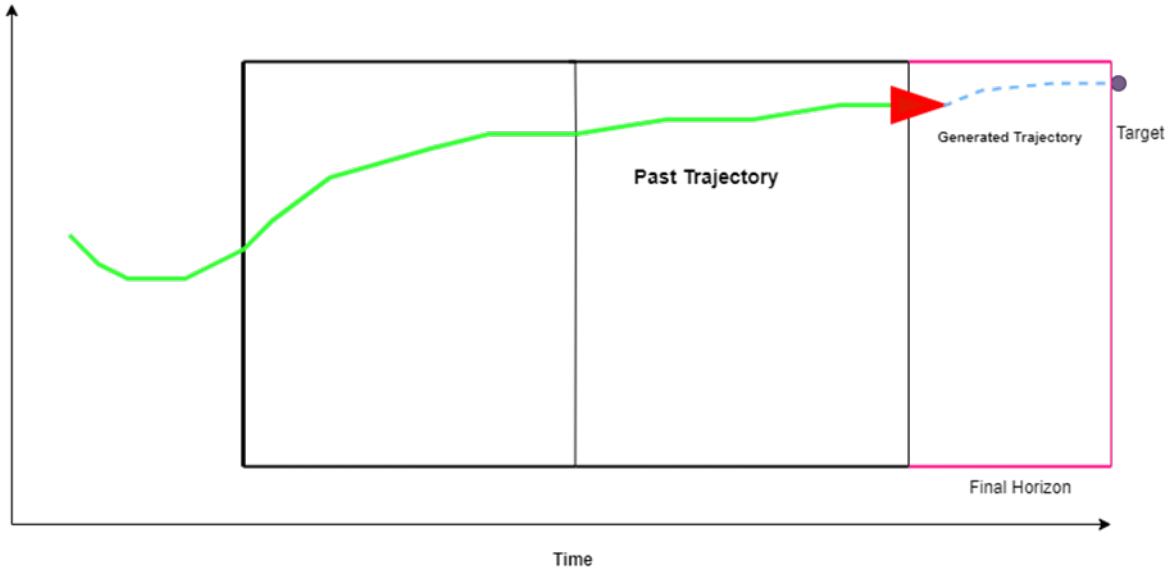


Figure 2.3: Example of approach

practical resolution for the most of the cases.
In this study the *quad_wright* script[13] was used and implemented inside the algorithm. In the code, Wright's method is used to solve the problem but doesn't allow for quadratic constraints to be applied. While it could be possible to surrogate some sort of constraints using some linear relationships, the algorithm implemented and presented does not include any conical constraint while still obtaining good results.

Chapter 3

Adaptive Sliding Mode Control Algorithm

The control algorithm implemented is the combination of two different concepts belonging to the SMC theory and methodology. For the sake of comprehensiveness, in this chapter, the main principles of SMC and the basic concepts and ideas behind these two different methods will be explored with an effort to reduce the mathematical complexities when not extremely required.

3.1 Sliding Mode Control - General Overview

Since they were first introduced, sliding mode methods showed great robustness and versatility for a wide range of applications while presenting some potential issues. The main idea of the method is to have a control law capable of driving the system towards a certain desired equilibrium (that could also be a trajectory), a *sliding surface*, and then keeping it in such state until desired, without suffering of uncertainties and disturbances. It is intuitive to deconstruct the whole control action in two different moments, the first is the reaching phase, the system is driven towards the sliding surface and it's vulnerable to uncertainties and disturbs, the controller drives the system until it reaches the sliding surface, where the second moments starts, the sliding motion, the system is now now nonsensitive to uncertainties and slides on the sliding surface. The ideal design is the one that can make the reaching phase as shortest as possible.
Another common issue of this methods is the phenomenon of the "chattering". Since the system staying perfectly upon the sliding surface is ideal and almost not realisable in practical situations, what happens is that the system finds itself oscillating back and forth with a certain frequency due to the discrete nature of the control law. Even though this guarantees the system to stay near enough to sliding surface and in the desired equilibrium, the "chattering" is often considered an issue, because it leads to decreased control accuracy, fatigue of the moving mechanical parts involved in the control action and high heat losses in power circuits.
The *sliding surface* is created by usually defining a sliding variable as a function of the system output variables. For example, if σ is the sliding variable, λ is a constant

and s, v are respectively the position and the velocity of a particle moving along a certain axis, than if we define:

$$\sigma(x) = v + \lambda s \tag{3.1}$$

$$\sigma(x) = 0; \tag{3.2}$$

The expression above is representing a surface (in this case a single line) in the state space (s, v). When the system is exactly located on the surface the sliding variable is a value of zero. To guarantee that the sliding motion will happen it is sufficient to satisfy a condition [31], often called reachability condition, which can be expressed:

$$\sigma\dot{\sigma} \leq 0 \tag{3.3}$$

This condition assures the stability of the system which will be driven towards the surface during the reaching phase. This is intuitive since the surface must be attractive for the trajectories to be directed towards it.
Here is an example of sliding surface, for simplicity of exposition, a single input single output system will be considered, but the concepts provided in the following lines don't lose of generality when multiple variables are considered.
If a linear time invariant system with position represented by x and speed represented by $\dot{x}$ is moving along the sliding surface

$$\sigma(x, \dot{x}) = \dot{x} + \lambda x = 0 \quad \text{with } \lambda > 0 \tag{3.4}$$

then it means the system undergoes a motion that satisfies a decreasing exponential law that is the solution of the differential equation (3.4)

$$x(t) = e^{-\lambda t} \tag{3.5}$$

which clearly identifies a motion asymptotically converging to the origin of the axis. Eq.(3.5) represents the position evolution in time when the system undergoes the sliding motion, it is typically said the sliding motion reduce the system dynamics by one order [31].
The same process can be repeated considering the errors from some desired states

$$\begin{gathered} e = x - x_d \quad \dot{e} = \dot{x} - \dot{x}_d \\ \sigma(e, \dot{e}) = \dot{e} + \lambda e \\ \textit{if } \sigma(e, \dot{e}) \quad \textit{than} \quad e(t) = e^{-\lambda t} \end{gathered} \tag{3.6}$$

so that now the errors from desired position x and speed $\dot{x}$ are asymptotically driven towards specified values.
The most classical version of the sliding mode control has a discontinue control law

$$u(x, \dot{x}) = -k \operatorname{sgn}(\sigma(x, \dot{x})) = \tag{3.7}$$

$$\operatorname{sgn}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases} \tag{3.8}$$

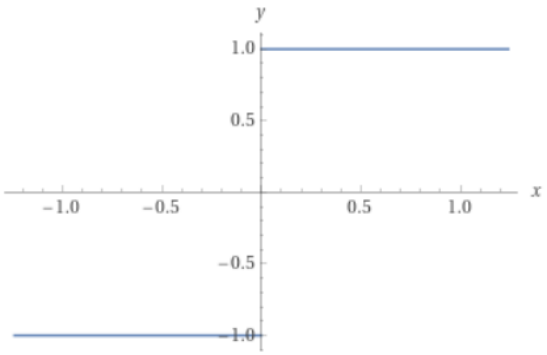


Figure 3.1: Sign function

The control input is function of the sliding variable which is some sort of function of the system states, so when in proximity of the sliding surface, as the reachability conditions guarantees, this control law will maintain the system on the sliding surface ideally. If from one side this solution let us have strong guaranties that the system will be staying in the immediate surroundings of the sliding surface, it is appropriate to remember the presence of the discontinuity in the control represents the most prominent source of the chattering phenomenon.

3.2 Chattering reduction methods

In literature there are several methods that try to reduce the chattering. In the following section some of them will be briefly explored and than some focus will be given to the super twisting algorithm, implemented during this work.

3.2.1 Removing the discontinuity

In certain applications, when dealing with the control of electrical motors or power converters, the control action required has a discontinuous nature so the sliding mode technique offers very good performance. When the high frequency effects of the chattering must be avoid, one straightforward and simple manner to reduce the chattering is of course removing the source itself. The sign function can indeed be substituted with some continuos approximation but at some loss, the sliding motion will be no longer guaranteed.
Without going in the mathematical details, what is crucial to know is that there is a condition that guarantees the reaching phase will be ending in a limited and defined time. This condition is met no more when a different function is substituted to the sign function,for example a continuous one, and the control law will drive the system only *asymptotically* into the sliding surface, meaning it is theoretically never reached.
In literature this is often referred as *pseudo-sliding motion* and solution as *Boundary layer sliding modes* have been explored. "Boundary layer" comes from the intuitive

idea that now the system will be staying in the neighbourhood of the sliding surface, which acts as an attractive layer of a finite thickness, instead of a simple surface (or line). The sliding motion is no longer guaranteed but it is possible to prove that the controller is capable of driving the system near enough the sliding surface in finite time.

One option for approximating the sign function is the hyperbolic tangent (3.9). When inside the hyperbolic tangent a coefficient is applied (α), it is possible to change the approximation fidelity, as the coefficient gets higher the function gets sharper and the approximation is better.

$$u = -k \tanh(\alpha\sigma) \tag{3.9}$$

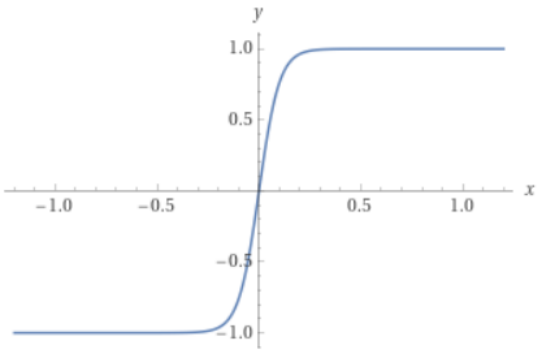


Figure 3.2: Hyperbolic tangent function with $\alpha = 10$

Some goes with another common solution, usually referred as *sigmoid function* (3.10), here the ϵ plays a really similar role to α in the previous example but in the opposite way, yielding a better approximation when being lower.

$$S_\sigma(\sigma) = \frac{\sigma}{|\sigma| + \epsilon} \tag{3.10}$$

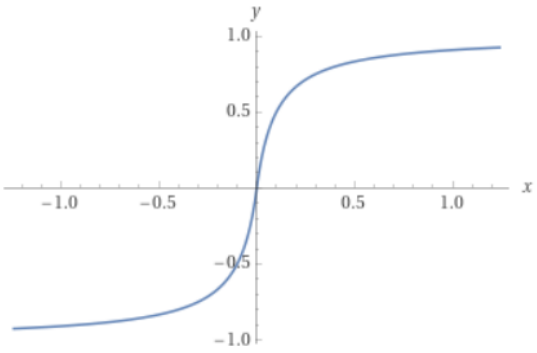


Figure 3.3: Sigmoid function with $\epsilon = 0.1$

3.2.2 Super Twisting Algorithm

The Super Twisting Algorithm (STA) is an advanced control technique that addresses some limitations of traditional sliding mode controllers. It is an algorithm that realizes higher order sliding modes (HOSM) [23]. STA significantly reduces the rapid and high-frequency switching between control modes, leading to smoother control signals and less mechanical wear also allowing more precision and tracking accuracy. Unlike traditional SMCs, which achieve asymptotic convergence, STA guarantees that the system reaches the desired state in a finite amount of time, regardless of initial conditions. The control law presents a term that is integrated, hence removing the discontinuity, while only preserving the need of knowledge of the sliding variable derivative only [28].
The control law implemented inside the algorithm is the following

$$u = \sqrt{|\sigma|}\tanh(k_\sigma \sigma) + \nu + k_v \dot{x} \tag{3.11}$$

$$\dot{\nu} = \begin{cases} -k\tanh(k_\sigma \sigma) & \text{if } u \leq U_{max} \\ u & \text{if } u > U_{max} \end{cases} \tag{3.12}$$

with $k_\sigma = 15$.
It is possible to recognise the implementation of a super twisting algorithm with an additional term where $\dot{x}$ represents the speed and k_v is a constant coefficient. This terms is purely meant as a sort of dumping term or as the derivative term of PID controller.

3.3 Time variant sliding surface

The SMCs algorithms are known for their robustness against uncertainties, but this is only valid when the reaching phase is ended, and the system *slides* on the sliding surface. This means, shortening or erasing the reaching phase can virtually lead to a controller capable of making the system totally invulnerable to uncertainties. Some work has been done previously to build an adaptive SMC algorithm, often proposing to use classic adaption mechanisms like parameter estimation or fuzzy logics to modify the control gains as the system is being driven. Successively the focus has moved to the sliding surface, the reaching phase lasts the time needed to the controller to drive the system on the sliding surface, where the sliding motion activates.
Some published solutions start from a sliding surface that includes initial conditions and slowly transpose or rotate it towards the desired one, with the system never experiencing a reaching phase and its downsides. Most of these solutions utilize polynomial time laws [5, 11, 29] where the coeffients are derived with methods based on parameter estimation and mission requirements. Other works propose the sliding surfaces coefficients to computed by some optimization framework which takes in account the system and mission requirements to determine them[8].

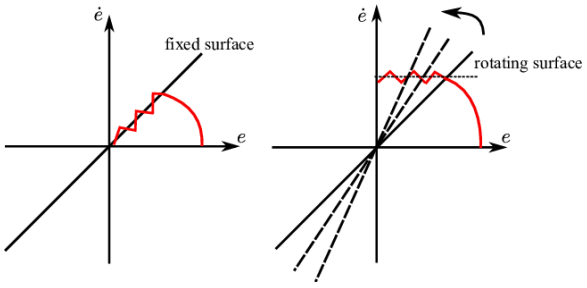


Figure 3.4: Example of a fixed and adaptive sliding surfaces in the state space [17]

In the references [25, 21], a time varying sliding surface rotates in the state space according to a law that takes in account the sliding variable and the states error as well, the proof of the stability of this law can be found in [25]. The sliding surface here applied guarantees the reduction of the reaching phase and improving of the tracking performance. This mechanism has been implemented, presented and proven twice in the mentioned previous studies, in this work an adapted version for the position dynamics is proposed.

Since the main task is tracking the trajectory generated and provided by the guidance algorithm, it was chosen to implement three different sliding surfaces (lines), one for each axis of motion, treating them as they were three different single input single output systems.
The sliding surfaces have been defined as

$$\sigma_x(x,t) = \dot{e}_x + \lambda(t)e_x \tag{3.13}$$
$$\sigma_y(y,t) = \dot{e}_y + \lambda(t)e_y \tag{3.14}$$
$$\sigma_z(z,t) = \dot{e}_z + \lambda(t)e_z \tag{3.15}$$

with $\dot{e}$ representing the velocity error and e the position error. The λ coefficient is no longer constant now, but changes with time according to

$$\dot{\lambda} = \text{proj}_{[\underline{\lambda},\overline{\lambda}]}(\lambda, h), \quad \lambda \in [\underline{\lambda},\overline{\lambda}] \tag{3.16}$$

$$\text{proj}_{[\underline{\lambda},\overline{\lambda}]} = \begin{cases} max\{0,h\} & \text{if } \lambda = \underline{\lambda} \\ h & \text{if } \underline{\lambda} \leq \lambda \leq \overline{\lambda} \\ min\{0,h\} & \text{if } \lambda = \overline{\lambda} \end{cases} \tag{3.17}$$

The proj function is designed to not let the coefficient λ outside the interval $[\underline{\lambda},\overline{\lambda}]$. The parameter h is also variable and it is function of the state error and the sliding variable

$$h = G\zeta_{\bar{\sigma}}(\sigma)\,\text{sign}(e_1) - c(\lambda - \bar{\lambda}), \tag{3.18}$$

$$G = \frac{c(\underline{\lambda} - \overline{\lambda})}{\overline{\sigma}} < 0 \tag{3.19}$$

$$\zeta_{\bar{\sigma}} = \begin{cases} 0 & \text{if } |\sigma| \geq \overline{\sigma} \\ \sigma & \text{if } |\sigma| < \overline{\sigma} \end{cases} \tag{3.20}$$

Where $\overline{\sigma}$ is the thickness of the boundary layer around the sliding surface inside which we want to maintain our system.
Finally, for this mechanism to work a condition shall be met regarding the highest admissible deviation from the desired position(or speed), the highest λ coefficient possible and the boundary layer thickness[25]

$$\overline{\lambda} = \frac{\overline{\sigma}}{E} \tag{3.21}$$

where E is the deviation.

Chapter 4

Results

In this chapter the most significant results will be discussed along with the sensitivity analysis of the main parameters and their influence on the final results. The parameters used in the simulation can be found in the appendix when they are not specified.

4.1 Complete Rendezvous

One of the main focuses of this work is to allow a satellite to be able to reach every eventual target during a multiple rendezvous mission while sustaining mass variations due to the mission purpose (debris removal, on orbit servicing). During every simulation the system is place in an arbitrary initial position with the ending point set about 4 meters away from the target's position. The guide block has also been designed to let the system reach the beginning of the final cone before starting with the approach, in this way it is easier to accomplish the rendezvous without violating the approach cone. The nominal mass of the DEMETER is about $130kg$, during this work variations up to 50% have been considered, making $65kg$ the lighter configuration and $195kg$ the heavier configuration. The results will be shown looking at the LVLH plane, with the V-bar as horizontal axis and R-bar as the vertical one, the R-bar axis has been reversed for better visualization.
Manoeuvres originating from distant initial positions have been tested and are now presented. The selected starting positions are intended to identify limit situations, as the system will be placed just within the range of validity for (1.1) and the LVLH frame. These results aim to validate the system's capability to successfully execute rendezvous operations, even when provided with the farthest initial positions.

In the figures (4.1,4.2,4.3) a manoeuvre starting from a lower altitude is showed, each one of them presents a different system weight during the simulation. The system completes the approach sufficiently well in each of the cases, showing some minor issues during the final stop when it is lighter. This is probably due to the thrusters, now providing more acceleration, and a more responsive dynamics because of the less amount of inertia. This of course reflects also on the shape of the trajectory in the LVLH frame, the trajectory is sharper when the system is lighter,

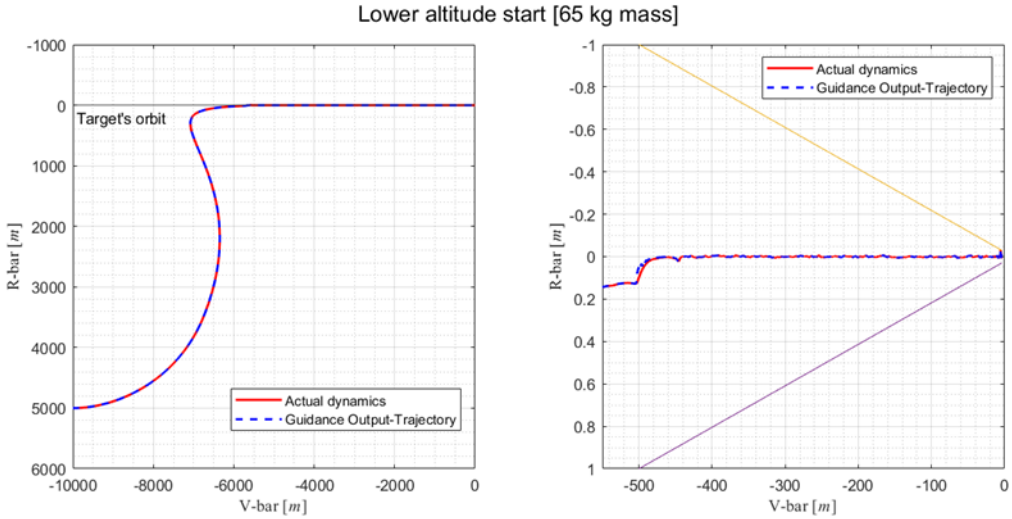


Figure 4.1: Rendezvous starting from lower altitude with lighter configuration. Complete manoeuvre(left),Cone of approach(right)

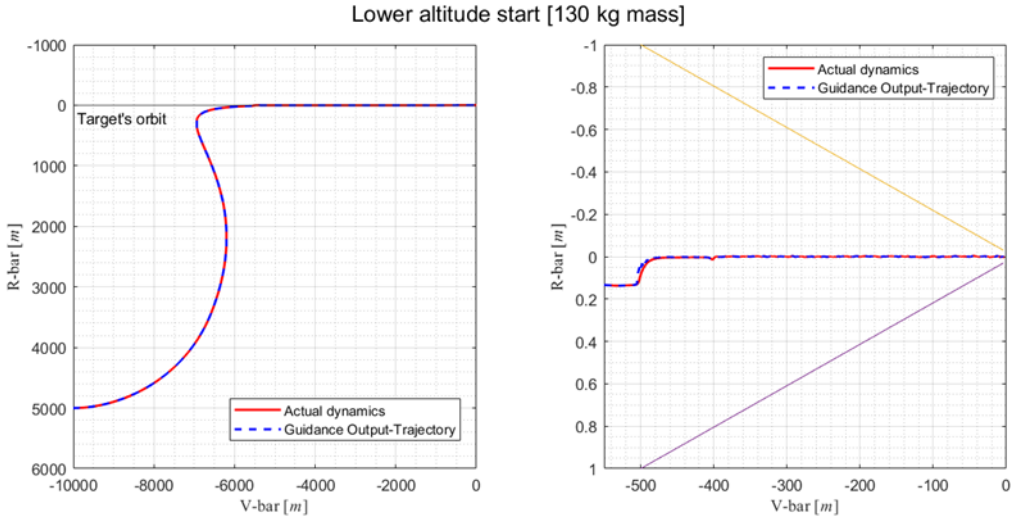


Figure 4.2: Rendezvous starting from lower altitude with nominal configuration. Complete manoeuvre(left),Cone of approach(right)

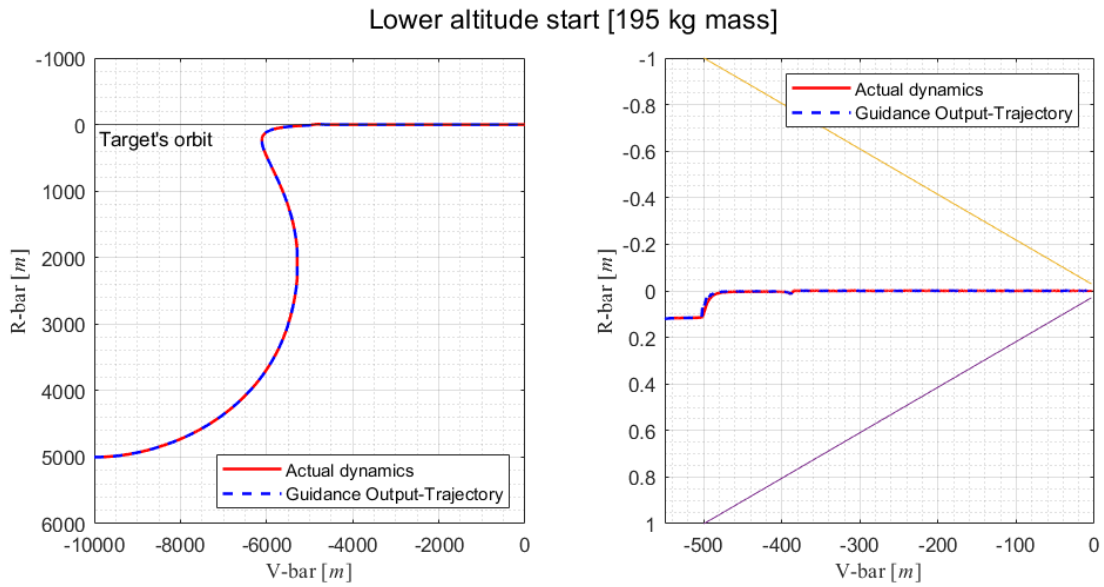


Figure 4.3: Rendezvous starting from lower altitude with heavier configuration. Complete manoeuvre(left),Cone of approach(right)

(there is more control authority and less inertia), and on the also on the amount of propellant consumed, which is higher when the system is heavier, as expected.
The guidance algorithm is not designed to adapt to the system weight, despite that, it manages to create admissible and successful trajectories for the system as we can see there is no difficulty for the controller to track the trajectory in any configuration presented. The time needed to complete the rendezvous safely was around 5200 and 5400 seconds, less than an orbital period.

In the figures (4.4,4.5,4.6) an higher altitude starting point is considered. The results again show the system is successful with a very smooth trajectory when the weight configuration is lighter. The heavier configuration managed to complete the rendez-vous but it was the one who took the most time, not only because of the less acceleration provided but also becacuse of some adjustment that was required when first approached to the target orbit. In any case the system was able to complete the mission in less than two hours, with a range of 5800 seconds to 6400 seconds respectively considering the lighter and the heavier configuration.

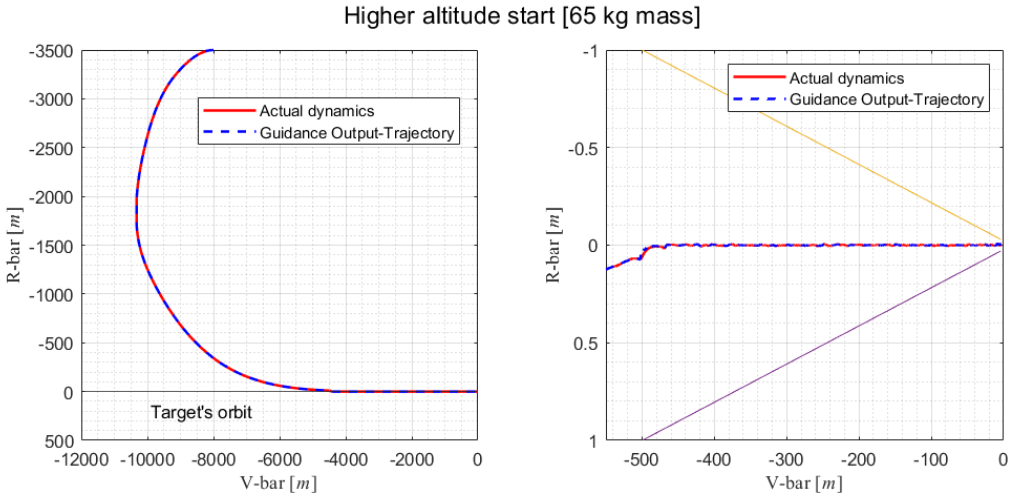


Figure 4.4: Rendezvous starting from higher altitude with lighter configuration. Complete manoeuvre(left),Cone of approach(right)

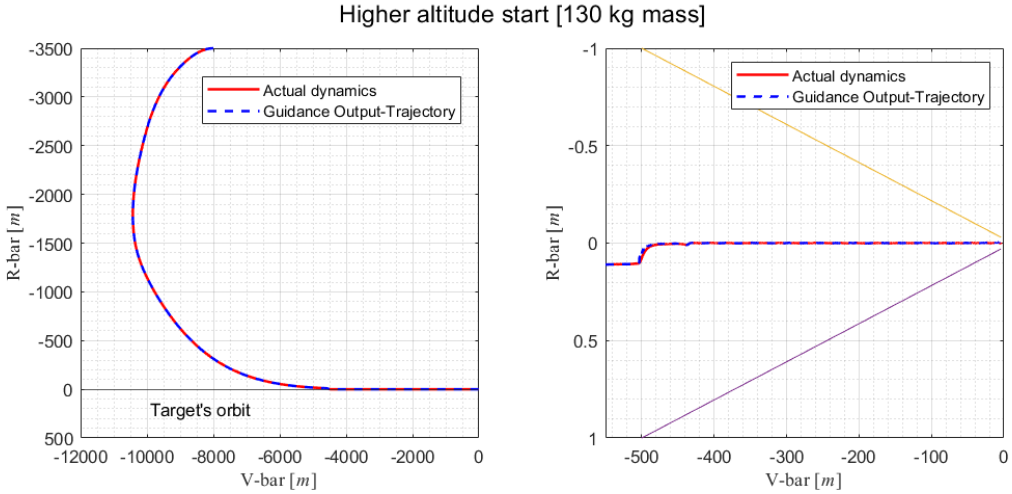


Figure 4.5: Rendezvous starting from higher altitude with nominal configuration. Complete manoeuvre(left),Cone of approach(right)

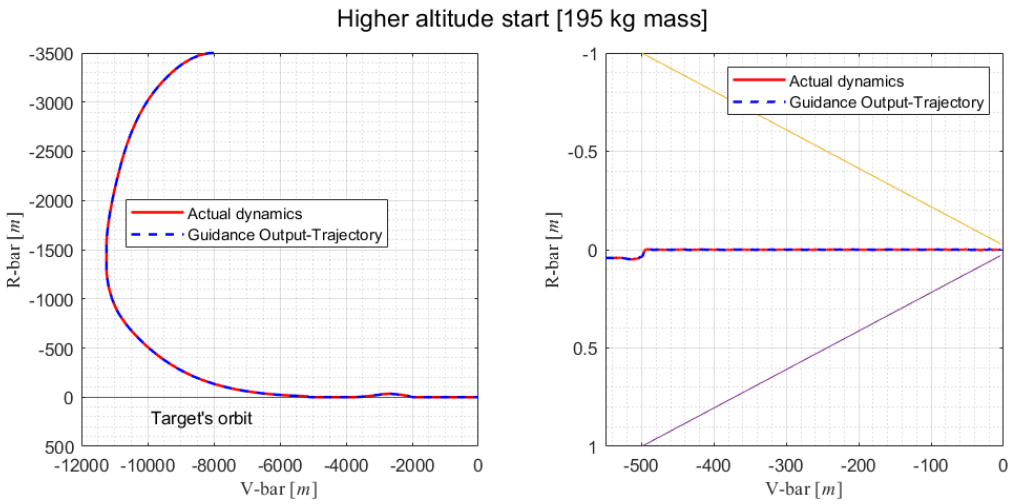


Figure 4.6: Rendezvous starting from higher altitude with nominal configuration. Complete manoeuvre(left),Cone of approach(right)

4.2 Actual Control vs Optimal Control

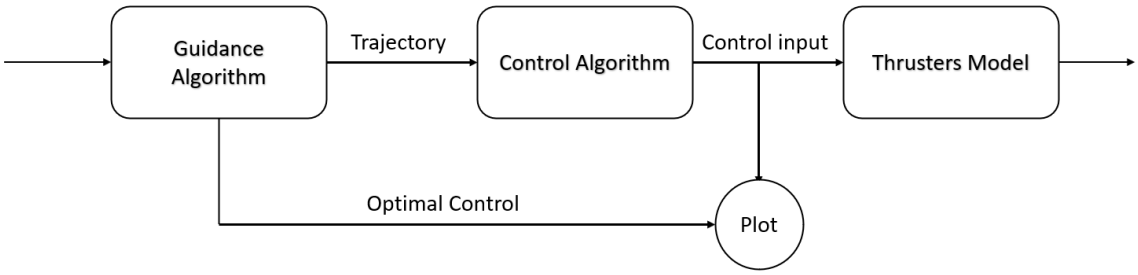


Figure 4.7: Section of GNC flowchart,from left there is navigation/system dynamics output, forward right there is the actual thrusters force.

Keeping in mind that the guidance algorithm does not have any adaptation mechanism to respond to weight variations, it is possible to make a confrontation between the controller output and the control variable the guidance algorithm computes for the optimal trajectory. The two results are often really similar even though no information about the control is exchanged between the two blocks (see fig.4.7); the first one is what the control algorithm determines to be the fittest control to follow the current desired trajectory, the second one is the result of the optimization process needed to compute that trajectory itself.
The manoeuvres represented in fig.(4.8) starts from , 200 meters, and with a 100 meters altitude less than the target, only the first 600 seconds are represented. The control sequence in fig.(4.94.10) is along the R-bar axis. The first manoeuvre is performed with nominal weight and fully operational thrusters, the second one is

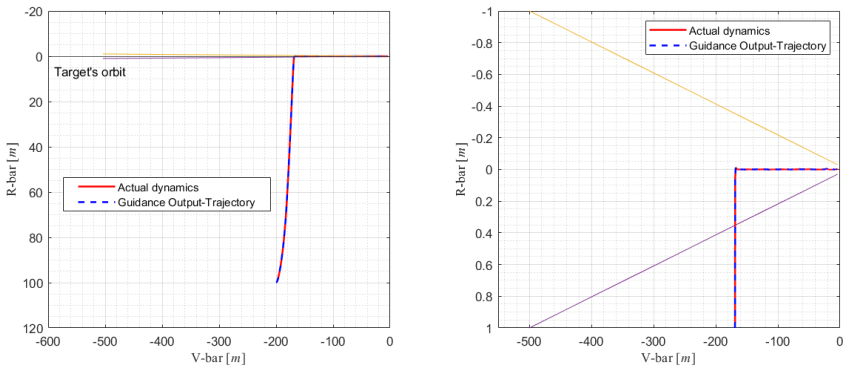


Figure 4.8: Manoeuvre considered for the study

performed with the thrust along the R-bar axis halved due to an hypothetical malfunction. The same has been done for the V-bar axis, the results can be found in the fig.(4.114.12).
To better visualize this, the controller output has been filtered with a simple low pass filter and reported in the graph along with its unfiltered version and the guidance optimal control. In this way the "noise" due to the sliding mode method is evened out (in a very similar fashion to SMC expositions when trying to visualize the equivalent control[31]) and allows to better visualize the confront. The results with the nominal configuration show great similarities between the plots and, when the high frequency variations are even out for sliding mode output, this is even more clear. When the configuration with halved is considered, the two plots retain a similar shapes but the values are slightly different. This can of course be attributed to the adaptive nature of the controller, whose action is here clearly visible.
It is worth remembering that the final control input the system will receive is given by the thrusters which have limited magnitude of thrust and on/off actuation. This means the actual control input received by the system will also be discontinue even though inside the guidance algorithm the control variable is bounded but continue and constant during each time step. Here the control provided by thrusters have been omitted for a cleaner visualization of the graphs since their model is composed by PWPF modulator. The time step length for the trajectory computation in the guidance block is set to 2 seconds and it small enough to provide a good approximation of an optimal control sequence when long enough horizons are considered. This also proves the choice to assume constant the thrust during each time step more reasonable, while maintaining a great simplification in the problem formulation. Further details about this can be found in the next section, where the effects of increasing the time step duration are investigated.

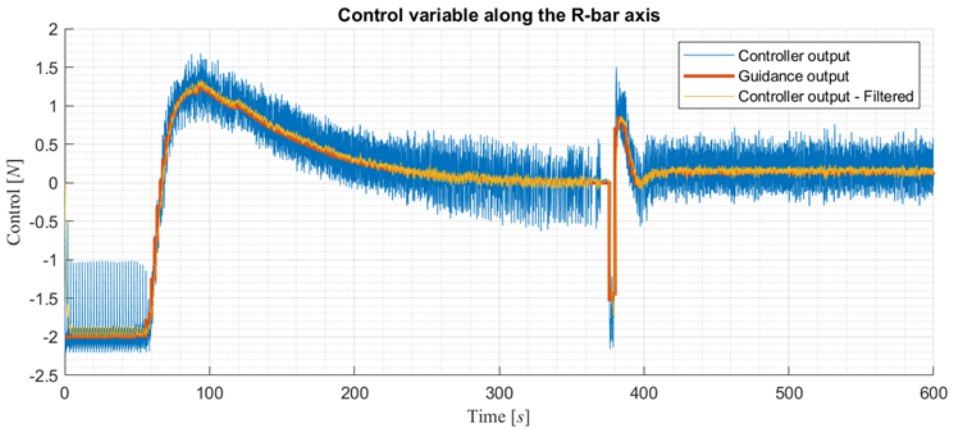


Figure 4.9: Confront between the guidance output and the controller output in a nominal configuration.

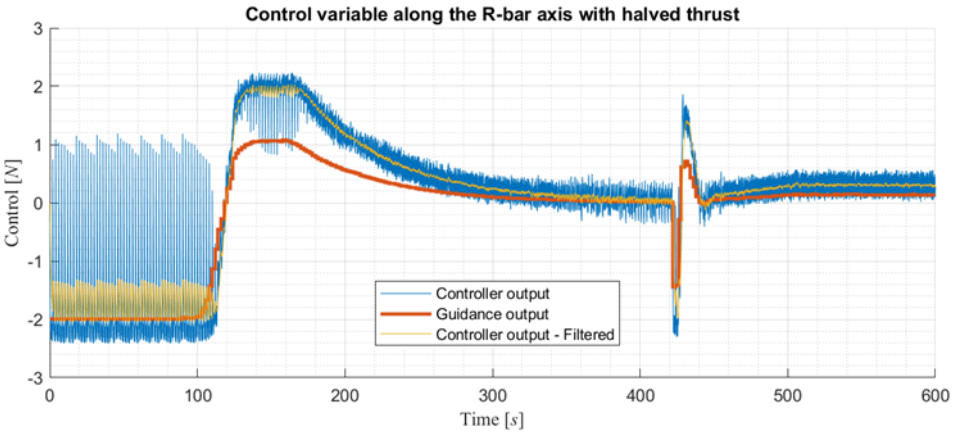


Figure 4.10: Confront between the guidance output and the controller output with halved thrust provided by the the reaction control system.

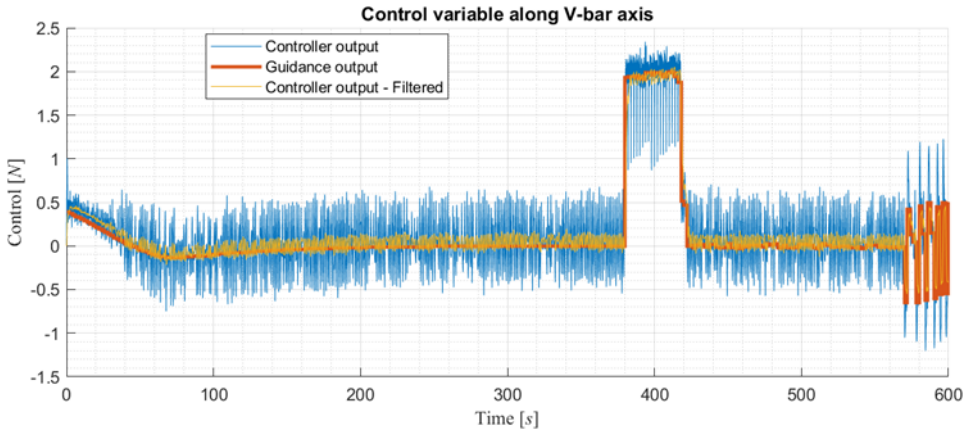


Figure 4.11: Confront between the guidance output and the controller output in a nominal configuration.

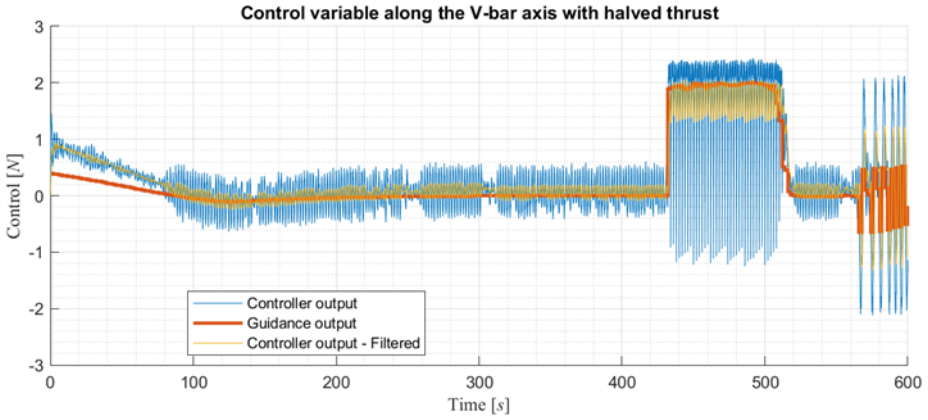
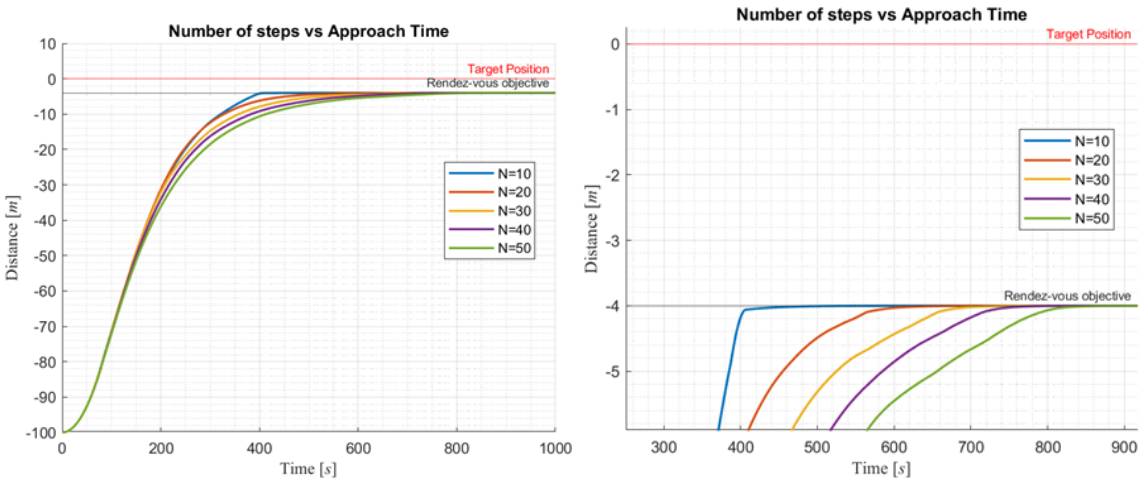


Figure 4.12: Confront between the guidance output and the controller output with halved thrust provided by the the reaction control system.

4.3 Horizon time influence

4.3.1 Number of steps variation

The final task to be verified is the guidance ability to allow autonomy to the system in terms of time of approach. In the work [27] a 10 minutes approach starting from a 100 meters distance is considered and then investigated about topics like propellant consumption and starting position. In any case presented, this is the time required for the approach since inside that framework it is implicitly defined a priori. This is because even though the definition of the number of steps and the step length is carried in the same way, the ending point for the computed trajectory are always imposed as constraint, computing a whole trajectory from start to end every iteration.
The guidance algorithm implemented provides the possibility to change the number of steps and their duration inside the optimization process, furthermore the update rate of the guidance block can be varied as well. In the following section, we explore the impact of varying the horizon length. This investigation involves keeping the update rate and step length constant while manipulating the number of steps, and conversely, keeping the number of steps constant while adjusting the time step length. It's important to note that the update rate itself does not directly affect the horizon length, its significance becomes more apparent in terms of computational effort and computational time. The system performs a manoeuvre starting from 100 meters of distance from the target along the orbit considered, all results from the simulations are reported in a single graph having time as horizontal axis and the distance along V-bar in vertical axis.



(a) Complete (b) Zoom

Figure 4.13: Approach time when the number of steps varies

As it is possible to see in fig.(4.13), there's seem to be a bond between the number of steps and the time required to approach the target. The higher the number of steps considered the more it takes for the system to reach the final position. This can be

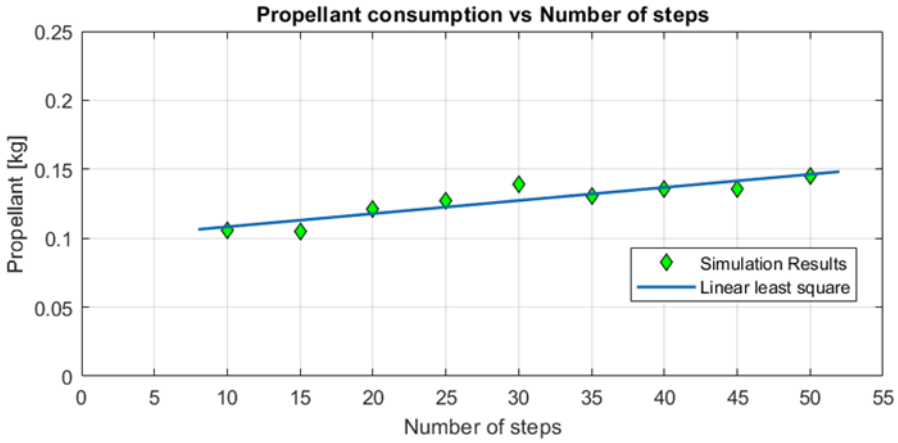


Figure 4.14: Linear fitting of the propellant consumed to accomplish the rendezvous while varying number of steps.

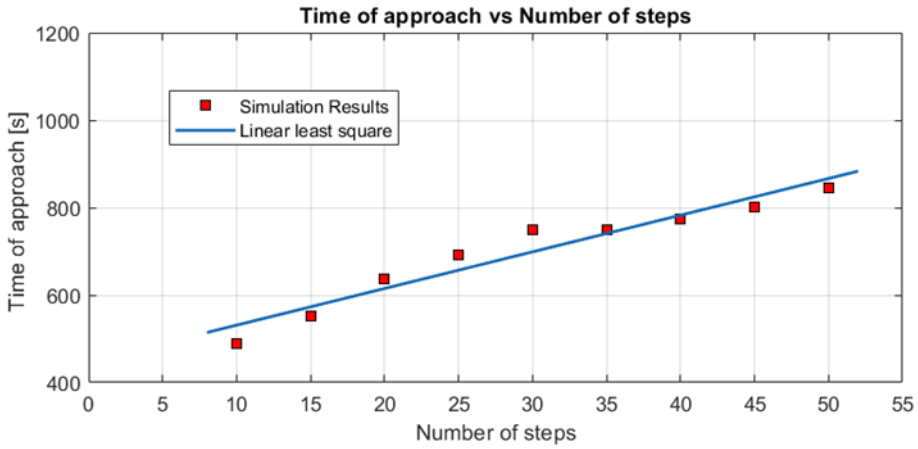
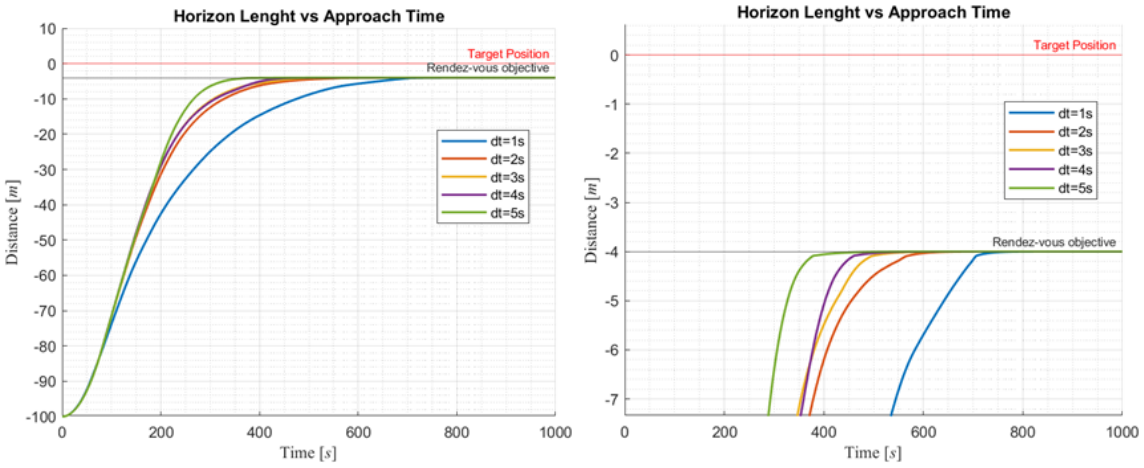


Figure 4.15: Linear fitting of the time required to accomplish the rendezvous while varying number of steps.

easily interpreted and explained if the very nature of an optimization process is attentioned. If the horizon time is thought as the amount of time that the optimization algorithm has to optimize the cost function, when the number of steps is lower, the process has less *time* to optimize the current position of the system, additionally the number of control variables is generally less than the states variable inside the the design variable vector. The result is that with this setup the optimization process prefer faster and sharper trajectories. When the number of step becomes higher this is no more valid, more time spent in idling and free-drifting is possible while waiting for the dynamics' outcome. This can also be found in fig.(4.14), which shows that the propellant consumption is not influenced so much and almost remain constant despite the more time spent to reach the target. The system is not manoeuvring more but smarter, slower and more efficiently.

4.3.2 Δt variation

Figure (4.16) shows the results if the time-step duration is varied. It is possible



(a) Complete (b) Zoom

Figure 4.16: Approach time when Δt varies.

to see that quite the opposite of earlier happens. The systems seems to become faster as the horizon time becomes longer due to longer time-steps. If the value of is allowed to increase furthermore, something different can be observed.
As shown in fig.(4.18), making the horizon longer allows the system to be faster until a certain point beyond which returns to be slower. The difference is that even though the time available is longer the system is faster and this is due to the fact that the number of variable is now fixed, since the number of steps doesn't change, so here a longer horizon seems to be of real benefit.
Nevertheless, when the step duration increase too far a certain point it is no longer beneficial, instead it makes the system less responsive and less agile, making it slower. This also is in line with the fact the a too wide step duration would make a poor continuous approximation of a control sequence. In this case the propellant consumption seems again to be not influenced by the horizon parameters, this is also due to the fact that, despite the changes, the system is consistent in its paths, that remain similar, and so almost the same amount of propellant is always used. At the same time it must be remembered that every trajectory followed by the system is an optimal one and since no cost function parameters were changed, it makes sense the resulting propellant consumption to be consistent as well.

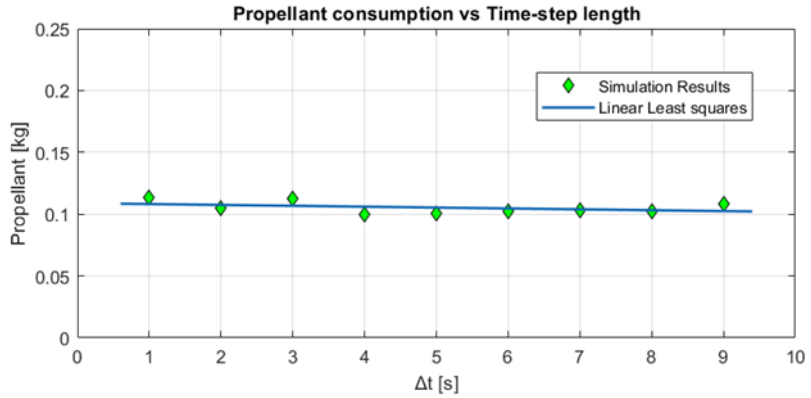


Figure 4.17: Linear fitting of the propellant consumed to accomplish the rendezvous while varying the time step duration.

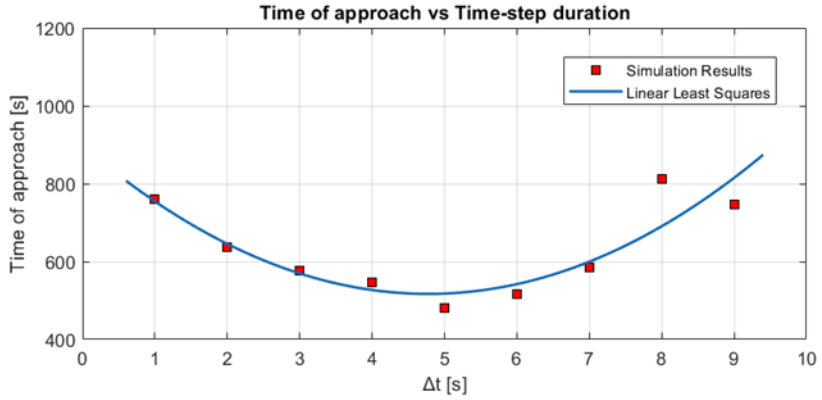


Figure 4.18: Quadratic fitting of the time required to accomplish the rendezvous while varying the time step duration.

Conclusions

The work presented in this thesis is meant to address the practical challenges related to debris removal and on orbit servicing missions inside the scenario of a multi-target rendezvous mission. The inevitable change of inertial properties inside a mission of this kind have been fronted developing a guidance algorithm and an adaptive control algorithm for a linearized relative motion dynamic.
The devised guidance algorithm is meant for on-board trajectory planning, employing convex optimization with quadratic programming for parameter optimization, solved through Wright's method. The optimization procedure operates within temporal frames or horizons, where a trajectory is generated and then tracked until the subsequent update. This methodology presents an innovative approach, differing from the conventional method that generates a trajectory from start to end in a singular process. The generated trajectory is then interpolated and transmitted to the controller for effective tracking.
The control algorithm is designed to be able to track the generated trajectory adapting to widest range possible of different conditions. To do so, a sliding mode controller has been used. The control law implemented is inspired to Super-Twisting algorithms and it works alongside a time varying sliding surface, which rotates in the state space according to an adaptive mechanism that takes in account the state error and the sliding variable. The first feature helps mitigating chattering and addresses issues associated with the typical control discontinuity often observed in Sliding Mode Control (SMC) methods. By doing so, it contributes to the overall stability and smoothness of the control process. The second feature is needed to minimize the reaching phase, the period during which the system is not fully capitalizing on the sliding motion and is particularly susceptible to uncertainties. This dual-feature approach enhances the robustness and efficiency of the control algorithm, making it able to adapt to different conditions.
The synergic action of guidance and control algorithm has demonstrated to be successful within the scope of assumptions outlined throughout the entire study. The results show the system was successfully driven to the desired destination and successively kept the position even when different weights have been used to run the simulations. A further investigation showed an example of how the controller acts to compensate for the different weight/thrust conditions of the system when the guidance is operating assuming the nominal configuration.
The investigation into the influence of horizon time length involved systematically varying the step time length and step time number. The results reveal that larger horizons generally enable faster trajectories, given that the vector of parameter vari-

ables remains constant (i.e., the number of steps remains fixed). However, when this is not the case, enlarging the horizon time tends to result in the system taking more time to manoeuvre.
Throughout this investigation, propellant consumption remained nearly constant. This observation proves the algorithm's capability to consistently compute optimal trajectories, irrespective of changes in horizon configurations. This adaptability allows the system to modify its maneuvering approach without significantly altering the overall effort, showcasing the algorithm's robustness and reliability.
There are few ideas for future expansions of this work, the first one being the realisation of the same work for attitude guidance and tracking. Deeper investigations can be done by analysing the horizon influence in a broader number of scenarios with the objective of refining the algorithm and make it the most versatile possible. Alongside these investigations, it can be important to determine the true potentials and limitations of this framework, working on the efficiency of the code and the solving methods. Some real hardware test can be part of this process, observing how much long and detailed (e.g. more steps inside a certain time frame) can the horizon become without exceeding computational time requirements. Throughout this refining process, a concerted effort should be made to relax the strongest assumptions made during this work. This approach aims to achieve higher fidelity results, ensuring the algorithm's applicability in real-world scenarios.
The other main perspectives that has driven the basis of this work is the obstacle detection and avoidance applications objective. The use of limited temporal frame to compute the trajectory faster and on board and is also meant as a starting tool to give awareness to the system about potential obstacles before being dangerously near to them, taking advantage of the fact that the trajectory is constantly updating and there is no need to know about the obstacle from the start of the mission. In this way, the system could be able to avoid collisions with any obstacle detected inside certain ranges by simply generating a new trajectory to follow once the obstacle information is inserted inside the optimization process. The convex programming approach allows for multiple solutions to be explored, starting from the insertion of additional constraints inside the problem formulation when appropriate to designing a quadratric barrier/penalty element inside the cost function to discourage the system to move inside certain restricted spaces.

Appendix A

Simulation parameters

NOTE: All the simulations are run using the ode4 solver with a step of $0.01s$.

Table A.1: Script Parameters

Parameter	**Value**
μ	$3.986\,012 \times 10^{14}$ m^3 s^{-2}
r_E	6378.145×10^3 m
Altitude	715×10^3 m

Table A.2: Thrusters and Controller Parameters

Parameter	**Value**
KF	12.5
τ	1
U_{on}	0.75
U_{off}	0.2
U_{max}	2
$\overline{\lambda}$	0.2
$\underline{\lambda}$	0.0001
σ_0	5×10^{-4}
c	1×10^{-4}
G	1×10^{-4}
k_x	2
k_y	0.2
k_z	2
K_v	20
α_x	2
α_y	0.05
α_z	2

Table A.3: Aerodynamic Drag Parameters

Parameter	**Value**
ρ	4×10^{-13} kg m^{-3}
C_D	2.2

Table A.4: Horizon Guidance Parameters

Parameter	**Value**
f	Zero vector of size $9N - 3$
H	Diagonal matrix with objectives for inside cone
H_{out}	Diagonal matrix with objectives for outside cone
$H_{\text{far\_altitude}}$	Diagonal matrix with objectives for lower altitude maneuvers
$H_{\text{far\_altitude\_off}}$	Diagonal matrix with objectives for higher altitude maneuvers
$H_{\text{far\_distance}}$	Diagonal matrix with objectives for same altitude maneuvers

Table A.5: Horizon Guidance Matrices

Matrix	**Expression**
H	diag ([repmat([0 0 1e3 1e2 1e2 1e1]′, $N-1$, 1); [1 1e1 1e3 1e3 1e2 1e1]′; ones($3(N-1)$, 1)])
H_{out}	diag ([repmat([0 0 1e1 1 5e3 1e4]′, $N-1$, 1); [0 5e1 1e1 1 5e3 1e4]′; 1e2 ones($3(N-1)$, 1)])
$H_{\text{far\_altitude}}$	diag ([repmat([1e − 1 0 5e1 1e4 0 5e5]′, $N-1$, 1); [1 0 1e2 1e4 0 1e6]′; 1e4 ones($3(N-1)$, 1)])
$H_{\text{far\_altitude\_off}}$	diag ([repmat([1e − 1 0 1e1 1e4 0 1e6]′, $N-1$, 1); [1 0 1e3 1e4 0 1e6]′; 1e3 ones($3(N-1)$, 1)])
$H_{\text{far\_distance}}$	diag ([repmat([0 1 1e3 1e3 1e3 7.5e2]′, $N-1$, 1); [1 1e1 1e3 2e3 1e3 1e3]′; 1e3 ones($3(N-1)$, 1)])

Table A.6: List of Symbols

Symbol	**Explanation**
μ	Earth gravitational parameter
r_E	Earth Radius
N	Optimization problem, number of discretization steps
KF	Filter gain for thrusters
τ	Filter time constant for thrusters
U_{on}	Trigger upper threshold for thrusters
U_{off}	Trigger lower threshold for thrusters
U_{max}	Maximum control input for thrusters
$\overline{\lambda}$ $\underline{\lambda}$	Upper bound for λ variable
$\overline{\lambda}$ $\underline{\lambda}$	Lower bound for λ variable
σ_0	Boundary layer thickness
c	Controller parameter
G	Controller parameter
k_x, k_y, k_z	Control law coefficients
K_v	Dumping term coefficient
α_x, α_y, α_z	Integrated term in control law coefficients
ρ	Air density for aerodynamic drag
CD	Drag coefficient for satellite

Bibliography

[1] Younes Al Younes and Martin Barczyk. Nonlinear model predictive horizon for optimal trajectory generation. *Robotics*, 10(3), 2021.

[2] Javad Amiryan and Mansour Jamzad. Adaptive motion planning with artificial potential fields using a prior path. In *2015 3rd RSI International Conference on Robotics and Mechatronics (ICROM)*, pages 731–736, 2015.

[3] Jun Bang and Jaemyung Ahn. Two-phase framework for near-optimal multi-target lambert rendezvous. *Advances in Space Research*, 61(5):1273–1285, 2018.

[4] Jun Bang and Jaemyung Ahn. Multitarget rendezvous for active debris removal using multiple spacecraft. *Journal of Spacecraft and Rockets*, 56(4):1237–1247, 2019.

[5] Andrzej Bartoszewicz. A comment on 'a time-varying sliding surface for fast and robust tracking control of second-order uncertain systems'. *Automatica*, 31(12):1893–1895, 1995. Trends in System Identification.

[6] Andrzej Bartoszewicz and Aleksandra Nowacka-Leverton. *Time-Varying Sliding Modes for the Second Order Systems*, pages 17–65. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[7] Andrzej Bartoszewicz and Aleksandra Nowacka-Leverton. *Time-Varying Sliding Modes for the Second Order Systems*, pages 17–65. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[8] Andrzej Bartoszewicz and Justyna Zuk. Time-varying switching lines for vsc of robot manipulators. 2006.

[9] Kristoffer Bergman, Oskar Ljungqvist, Torkel Glad, and Daniel Axehill. An optimization-based receding horizon trajectory planning algorithm. *IFAC-PapersOnLine*, 53(2):15550–15557, 2020. 21st IFAC World Congress.

[10] François Buisson. The demeter program: A pathfinder to a high performance micro satellite line. 2003.

[11] Seung-Bok Choi, Dong-Won Park, and Suhada Jayasuriya. A time-varying sliding surface for fast and robust tracking control of second-order uncertain systems. *Automatica*, 30(5):899–904, 1994.

[12] Connerney. In-space inspection maneuver analysis using trajectory optimization. *www.amostech.com*, 2021.

[13] Jonathan Currie. Practical applications of industrial optimization: from high-speed embedded controllers to large discrete utility systems. 2014.

[14] Andrew H. Digirolamo, Lawrence J.and Hoskins, Kurt A. Hacker, and David Bradley Spencer. "a hybrid motion planning algorithm for safe and efficient close proximity, autonomous spacecraft missions". 2014.

[15] Christopher Edwards and Sarah K. Spurgeon. Sliding mode control : theory and applications. 1998.

[16] ESA. https://www.esa.int/space_safety/clean_space.

[17] Bita Fallahi, Carlos Rossa, Ron Sloboda, Nawaid Usmani, and Mahdi Tavakoli. Sliding-based image-guided 3d needle steering in soft tissue. *Control Engineering Practice*, 63:34–43, 06 2017.

[18] W. Fehse. *Automated Rendezvous and Docking of Spacecraft*. Cambridge Aerospace Series. Cambridge University Press, 2008.

[19] Peter W. Fortescue, J. P. Stark, and Graham G. Swinerd. Spacecraft systems engineering. 1995.

[20] De Ruiter Anton H J., Christopher Damaren, and James R. Forbes. *Spacecraft Dynamics and control an introduction*. Wiley, 2013.

[21] Yoni Lahana, Mauro Mancini, Dimitri Peaucelle, Elisa Capello, and Hélène Evain. Comparison of adaptive control laws on a satellite attitude control benchmark*. In *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 1–6, 2023.

[22] DongUk Lee and Jaemyung Ahn. Optimal multitarget rendezvous using hybrid propulsion system. *Journal of Spacecraft and Rockets*, 60(2):689–698, 2023.

[23] ARIE LEVANT. Sliding order and sliding accuracy in sliding mode control. *International Journal of Control*, 58(6):1247–1263, 1993.

[24] Ping Lu and Xinfu Liu. Autonomous trajectory planning for rendezvous and proximity operations by conic optimization. *Journal of Guidance, Control, and Dynamics*, 36(2):375–389, 2013.

[25] Mancini M. *Adaptive Variable Structure Control System for Attitude Spacecraft Applications*. PhD thesis, Politecnico di Torino, 2023.

[26] Nasa. https://www.nasa.gov/mission/capstone/.

[27] Nicholas Ortolano, David Geller, and Aaron Avery. Autonomous optimal trajectory planning for orbital rendezvous, satellite inspection, and final approach based on convex optimization. *The Journal of the Astronautical Sciences*, 68, 05 2021.

[28] Jorge Rivera Dominguez, Luis Garcia, Christian Mora, Juan Panduro, and Susana Cisneros. *Super-Twisting Sliding Mode in Motion Control Systems*. 04 2011.

[29] A. Sivert, F. Betin, A. Faqir, and G.A. Capolino. Robust control of an induction machine drive using a time-varying sliding surface. In *2004 IEEE International Symposium on Industrial Electronics*, volume 2, pages 1369–1374 vol. 2, 2004.

[30] S Tokat, I Eksin, and M Güzelkaya. A new design method for sliding mode controllers using a linear time-varying sliding surface. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 216(6):455–466, 2002.

[31] Vadim I. Utkin. Sliding mode control. 2004.

[32] JingRui Zhang, ShuGe Zhao, Yao Zhang, and Ying Li. Optimal planning approaches with multiple impulses for rendezvous based on hybrid genetic algorithm and control method. *Advances in Mechanical Engineering*, 7(3):1687814015573783, 2015.