POLITECNICO
MILANO 1863

Tube Model Predictive Control for Robust Close-Range Rendezvous with Tumbling Target

Tesi di Laurea Magistrale in
Space Engineering - Ingegneria Spaziale

Author: **Davide Zamblera**

Student ID: 10612897
Advisor: Prof. Mauro Massari
Academic Year: 2022-23

A mio zio Andrea.

Abstract

The increasing threat of collision with space debris for commercial and scientific missions in Earth orbit has sparked interest in GNC methods that can successfully rendezvous with these uncontrolled objects for de-orbiting. An important area of research deals with assuring a successful manoeuver despite the uncertainties in the estimation of the target state. The current work uses Tube Model Predictive Control to deal with the disturbances acting on the system and track the reference trajectory for each possible realization of the stochastic dynamics. Differently from previous works in the literature, the method is here applied to a realistic setting considering a rendezvous with the Envisat satellite with uncertainty on the inertia of the target. The controller robustness guarantees are verified with a Monte Carlo analysis.

Keywords: Space debris; GNC methods; Tube Model Predictive Control; Envisat satellite; Monte Carlo analysis

Abstract in lingua italiana

La crescente minaccia di collisione con detriti spaziali per missioni commerciali e scientifiche in orbita terrestre ha suscitato interesse nei metodi GNC che possono effettuare rendezvous con questi oggetti non controllati in un ottica di de-orbiting. Un'importante area di ricerca si occupa di garantire il successo della manovra nonostante le incertezze nella stima dello stato del bersaglio. Il presente lavoro utilizza il controllo tube-MPC per gestire le perturbazioni che agiscono sul sistema e seguire la traiettoria di riferimento per ogni possibile realizzazione della dinamica stocastica. Diversamente dai precedenti lavori in letteratura, il metodo è qui applicato in un contesto realistico considerando un rendezvous con il satellite Envisat con incertezza sull'inerzia del bersaglio. Le garanzie di robustezza del controller sono verificate con un'analisi di Monte Carlo.

Parole chiave: Detriti spaziali; Metodi GNC; Tube Model Predictive Control; satellite Envisat; analisi Monte Carlo

Contents

1 | Introduction

1.1. Motivation

In recent times the number of space debris in Earth orbit is growing at an increasingly faster rate. This is a fact that is reported by a large number of sources. Research on the main composition and causes of this growth are here discussed to motivate the need for new methods capable of dealing with this phenomenon.

The idea that one day the orbital space around Earth would become too crowded with debris and uncontrollable satellites was born right at the beginning of the space exploration era. Scientific writer Willy Ley stated in 1960 that “In time, a number of such accidentally too-lucky shots will accumulate in space and will have to be removed when the era of manned space flight arrives”[19].

In the following decades, the work of Kessler led to one of the first models that described the satellite population, such models provide a mathematical description of the distribution of objects in space, their movement and flux based on the physical characteristics of such objects (e.g. size, mass, density, reflection properties and intrinsic motion)[31]. Using publicly available databases such as NORAD the model could predict the evolution of the debris population. In particular, the phenomena of generation of debris can be subject to runoff and the time required to generate an asteroid belt that would impact human commercial space application was quantified in decades. Although the term Kessler syndrome was coined after his first work the current meaning of the expression, indicating a chain reaction of collision ignited when reaching a certain critical population density of objects, was obtained only after the publication of the paper “Collisional cascading: The limits of population growth in low Earth orbit’. The main driver in the production of debris was identified as the collision of debris with satellites of 1000 kg or more, such collisions tend to result in the generation of hundreds of additional fragments larger than 10 cm[15].

The situation requires active measures to deal with the increasing number of space residents in Earth orbit, the effect of continuing with the policy of business as usual would lead to the rapid deterioration of access to space in the current century, as visible in fig. 1.1

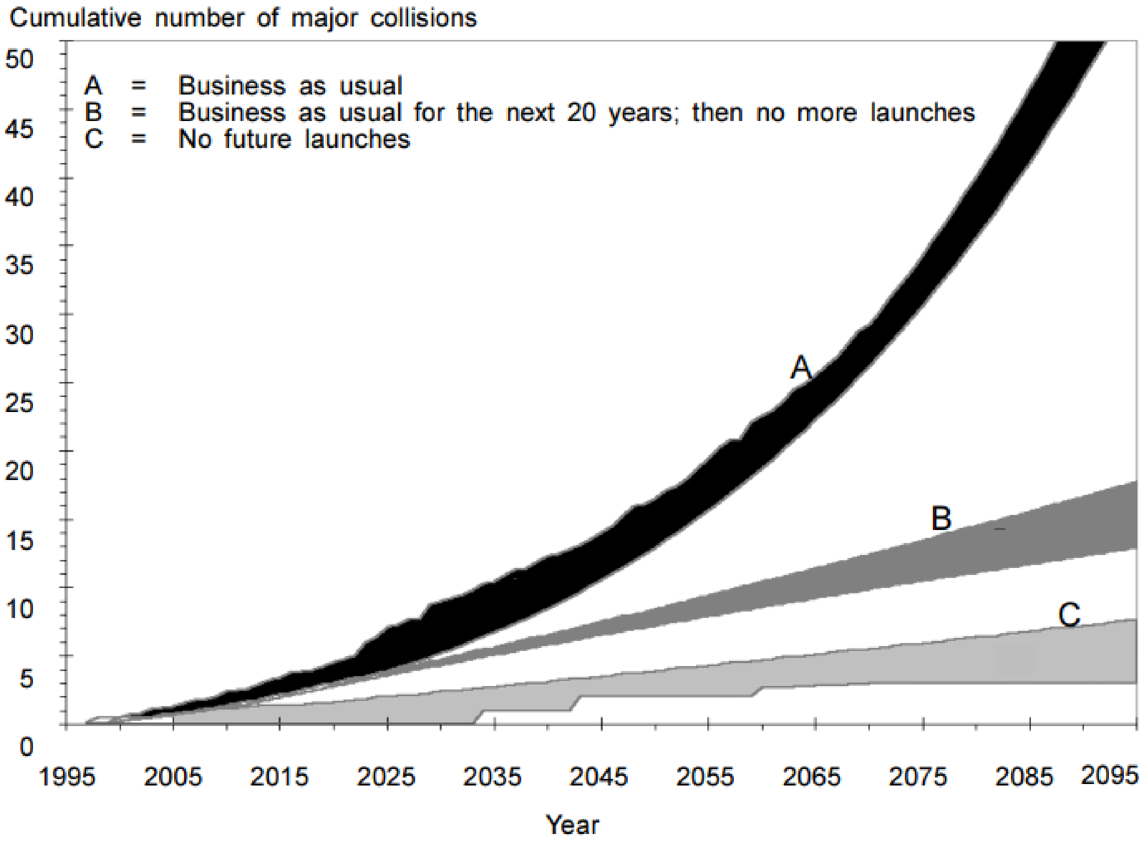


Figure 1.1: Expected number of collisions, from UN Technical Report (1999)[31]

These results motivated the space science community to research ways to limit the approach to the critical population density. Even though the great majority of objects over 1000 kg is capable of taking active countermeasure from an impending collision, for example, the ISS which has reached a total of 32 collision avoidance maneuvers since 1999[1], many others are uncontrollable, due to failures or not having a control system to begin with, such as rocket upper stages.

1.2. Active Debris Removal

The only way to reduce the number of uncontrolled space objects is to actively remove them from their current orbit, either by placing them in a "graveyard" orbit, an option most suitable for GEO satellites, or by controlled atmosphere de-orbit. During the rest of the work, the uncontrolled spacecraft is referred to as target or tumbler, as it will rotate in an initially unknown tumbling motion that results from the action of perturbations until the time of the rendezvous. The spacecraft that is tasked with the rendezvous and that provides the control authority is referred to as the chaser. The outline of the mission is depicted in fig. 1.2. The mission is composed of four main phases[12]:

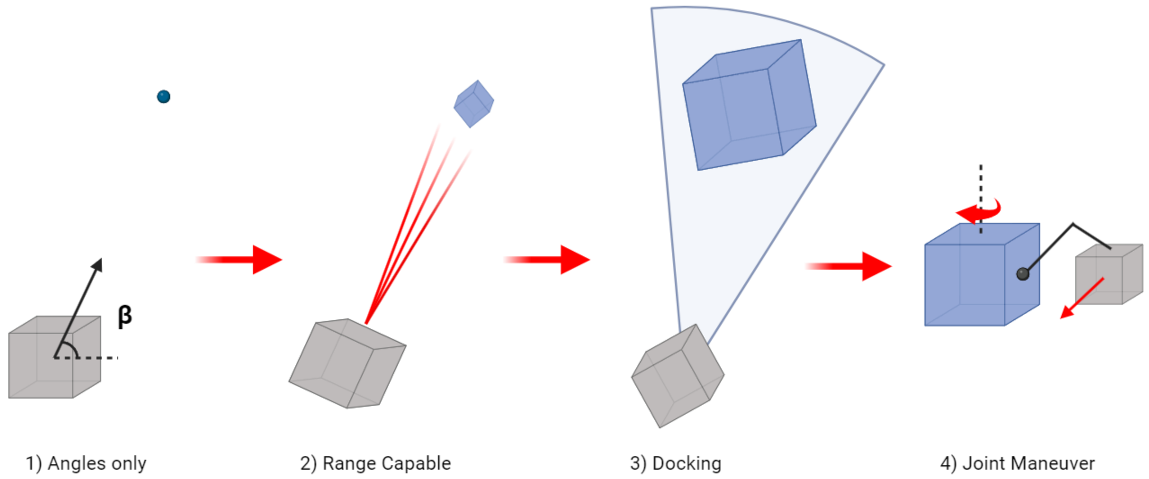


Figure 1.2: Phases of the ADR mission

1. Angles only rendezvous phase: This phase is carried out in the range from 10 km to 1 km from the target. It consists of a motion towards the target using only relative angular measurements. The objective is to lower the distance such that additional sensors can be used.
2. Range-capable rendezvous: This phase begins when range-based measurements become available and terminates at 100 m from the target. Thanks to the use of LIDAR or RADAR the chaser is capable of much more accurate measurements, although still insufficient for a rendezvous.
3. Docking: in this phase all the measurement suite of the chaser is available, this includes 2D visual cameras, 3D time of flight cameras and an inertial measurement unit. This phase requires the chaser to move the workspace of the robotic arm such that it contains the mating point of the target, identified as a feasible and easy-to-reach point on the target for the robotic arm to grasp. A very important part of this phase is the grasp itself of the robotic arm which requires modeling of the physical contact by use of non-smooth dynamics.
4. Joint maneuver: during this last phase the chaser uses its control authority to detumble the target and then moves it to a new relocation position using relative angle measurements.

This thesis will focus on the Docking phase, in particular on the rendezvous with the target, an analysis of the previous works and the available literature was carried out to understand what has already been achieved and the future directions of the research field.

1.3. State of the art and Development Trends

The field of ADR is relatively young when compared to other aerospace technologies. In this thesis only the advancements toward missions with robotic arms are considered since they are the most relevant to this work and also the most mature of the technologies, compared to tethered nets, electrostatic or gravity tractors. Even though no operative missions have been carried out as of the publication of this work, a certain number of demonstrative missions were conducted. These missions although not in itself ADR are important advancements towards autonomous and robust rendezvous capabilities. Notable examples of demonstrations of autonomous rendezvous are the ETS-VII[21] and DART missions. While both of these missions encountered various anomalies during their execution ETS-VII terminated in a success. The main task of ETS-VII was to use its robotic arm to place its smaller target satellite Orihime and perform experiments on it. Orbital Express, a more recent mission from DARPA, demonstrated multiple critical technologies in autonomous rendezvous. It must be highlighted that all these missions operated on known and to some extent cooperative targets, these conditions are never respected when operating on space debris. The most recent trends in the research community reflect the need to obtain technologies and algorithms capable of operating in these difficult conditions.

A recent snapshot of this research branch shows that the main research effort has been split into two categories [34]. On one side there is a need to produce incrementally better estimation methods such that the uncertainty that the system has to deal with is reduced, the most notable works in this sense revolved around the application of EK filters in various contexts and the development of novel dynamics models in term of minimum set inertia parameters [10]. Since the uncertainty can for sure be reduced but cannot be completely removed a second important research thrust consists of developing new guidance and control laws that allow the chaser spacecraft to deal with these uncertainties and complete the rendezvous successfully in all the possible realization of the mission.

In the following, various advanced control methods are highlighted, in particular those that contributed the most to the development of this research area. These approaches play a crucial role in providing context for the current state of research and insight into how to control complex systems in diverse environments with uncertainties.

Slotine and Li [27] apply sliding mode control (SMC) to the close-rendezvous problem. SMC consists of a control system with multiple structures that can be switched depending on the current state. This approach offers a robust solution, but it can result in excessive control effort and state oscillations around reference values. The paper provides insights into SMC's strengths and limitations, contributing to the overall understanding of control under uncertainty.

Rekleitis and Papadopoulos [24] propose a linear approximation of uncertainties and utilize model-based proportional-derivative (PD) control to achieve better control performance. They emphasize the importance of managing linear systems with robustness and parametric sensitivity tools, demonstrating how simpler control methods can be successful when reasonable approximation can be made to the stochastic quantities affecting the system.

Seddaoui and Saaj [26] explore the application of nonlinear H-infinity controllers to achieve robustness when handling different internal and external perturbations. Their research shows that while the controller effectively deals with the disturbances, it introduces complexity and computational overhead as trade-offs. This highlights the importance of optimizing robust control approaches for practical applications.

Ulrich et al. [30] present an adaptive control strategy designed to achieve mission objectives in real-time contexts, outperforming traditional PD control methods. However, the approach has limitations, in particular in the use applied to free-floating satellites, where it has restricted applicability.

1.4. Model Predictive Control and Tube-MPC

An important control method that was applied only recently to this problem is Model Predictive Control and its variants. Model predictive control (MPC) is a control method initially used to manage the process of chemical plants and oil refineries since the 80s[32]. It relies on a model of the system for predicting its behavior thus allowing for an optimization that takes into account the system evolution at future times. Only the first input of the optimized control sequence is applied and the process is repeated by shifting the time horizon.

In recent times the application in other fields and the need to have safety guarantees where random disturbances are important forces acting on the system has generated multiple derived methods [23]. If a certain bound can be computed for the magnitude of the disturbance, then the family of methods called Robust Control can be applied and the states and control constraints can be satisfied for all control sequences. Alternatively, for unbounded disturbances, Stochastic MPC control is used, which satisfies these constraints on average or with a certain probability. The disturbance for this specific problem was already verified to be bounded in a previous work [2]. The first Robust MPC control method is min-max robust MPC, which is based on Dynamic programming but employs simple parametrization to reduce the computational cost. The second alternative is the Tube-Based MPC. This controller shifts the idea of controlling a single trajectory to controlling the whole realization which can be imagined as being bounded in a tube. Theoretical results on the invariance and robustness of the trajectories under any sequence of the bounded disturbance are well-consolidated [20]. The method was recently applied to the rendezvous problem, showing good performances. The work of Buckner [6] verified the method, although only for a final angular disturbance of the target orientation, which is not similar to the one acting on the chaser in practice.

1.5. Thesis outline

In recent times complete pipelines for the rendezvous part of an ADR mission have been developed. In particular, this thesis is built upon the work done in the paper “A Robust Observation, Planning, and Control Pipeline for Autonomous Rendezvous with Tumbling Targets” by Keenan Albee et al. The paper uses innovations in the two main regions of research on ADR to build a software system that was to be ultimately tested on Astrobee robots, free-flying robots operating aboard the ISS. The autonomy pipeline is defined here as it is useful to have a holistic view of the system and how all the pieces interact:

1. A Graph-based SLAM estimation incorporates all measurements from depth-based odometry and IMU-based odometry to produce low uncertainty knowledge of the target tumbling motion. The motion is estimated for a long time before the beginning of the maneuver itself. In this thesis, this part of the pipeline is not treated, but simulated, as the primary focus is on the guidance and control of the chaser.

2. The most recent estimate of the tumbler attitude is passed to a motion planning unit. This unit uses the nominal tumbling motion to generate a collision-free and optimized trajectory in the tumbler frame. This part of the pipeline is treated in chapter 3.

3. Due to the mismatch between the nominal and real angular dynamics of the tumbler, the reference trajectory and the error dynamics of the chaser will be affected by a disturbance. The maximum values of this disturbance are computed in the uncertainty-bound approximation unit. This unit is described in chapter 4.

4. The uncertainty bound is used together with information about the system, such as constraints and dynamics, to produce a robust invariant set that guarantees correct trajectory tracking when the system is subjected to any uncertainty in the bound. Along with these, this unit produces tightened constraints that account for the fact that part of the control authority is used to deal with the disturbance. This part is treated in chapter 5.

5. The results from the robust set approximation and the nominal trajectory, along with the most recent measurements are then used by the control unit to generate a control action at each time instant. The controller uses tube guarantees to deal with the uncertainty and track the nominal MPC trajectory. This unit is described in detail in chapter 6.

Even though the original paper treated the problem as a whole, for reasons related to the feasibility and easiness of implementation on real hardware it implemented the method for very short rendezvous distances, in the order of 1 meter. This thesis intends to expand the results of the implementation using the case of Envisat, with a realistic application scenario. It must be noted that in Buckner [6] a similar scenario is developed but there are two main differences with this work. The disturbance in this case comes from a continuous update of the nominal trajectories when new data are available, instead of a perturbation of the final rendezvous point position. Additionally, the effect of inertia uncertainties of the target is considered, on top of the uncertainty on the initial state.

2 | Problem Definition

As explained in the introductory chapter the current work is limited to the first part of the Docking phase, this means that the environment considered will be bounded in the vicinity of the target, in the order of hundreds of meters, which allows for simplified dynamics. Additionally, the maneuver takes place in a short time horizon (minutes), such that orbital and attitude perturbations can be ignored.

Section 2.1 contains a description of the rendezvous and the reference frames used in the rest of the thesis. A dynamical model of the linear and angular relative dynamics is required for the generation of the reference trajectory and the prediction capabilities of the MPC. The dynamics of the system include the Clohessy-Wiltshire equations for the relative linear dynamics, treated in section 2.2.1 for continuous time and in section 2.2.2 for the discretized version used in the MPC; the Euler equations of a rigid body for the chaser and the target satellite are presented in section 2.3.

2.1. Setting and Reference frames

The problem setting is that of close rendezvous operations between a chaser and a target. The chaser is considered a free-flying object, with full controllability both in linear and angular motion, constrained solely by the thruster's force limitations. The target is instead a free-floating object, on which no external forces are exerted and as such its angular motion will not be altered during the maneuver. The frames used in this work are represented in fig. 2.1, in detail, these are:

1. LVLH frame (L) with center in the CoM of the target and x-axis in the radial direction of the orbit, while z perpendicular to the plane of the orbit.

2. Inertial frame (I) taken arbitrarily equal to L frame at t_0, its orientation is not important, but it is necessary to express correctly the dynamics of the spacecraft.

3. Target frame (T), attached to the space debris, with the center in the CoM of the target and with an arbitrary orientation, specified in chapter 7 here for the Envisat case.

4. Chaser frame (C), as for the Target frame but for the Chaser.

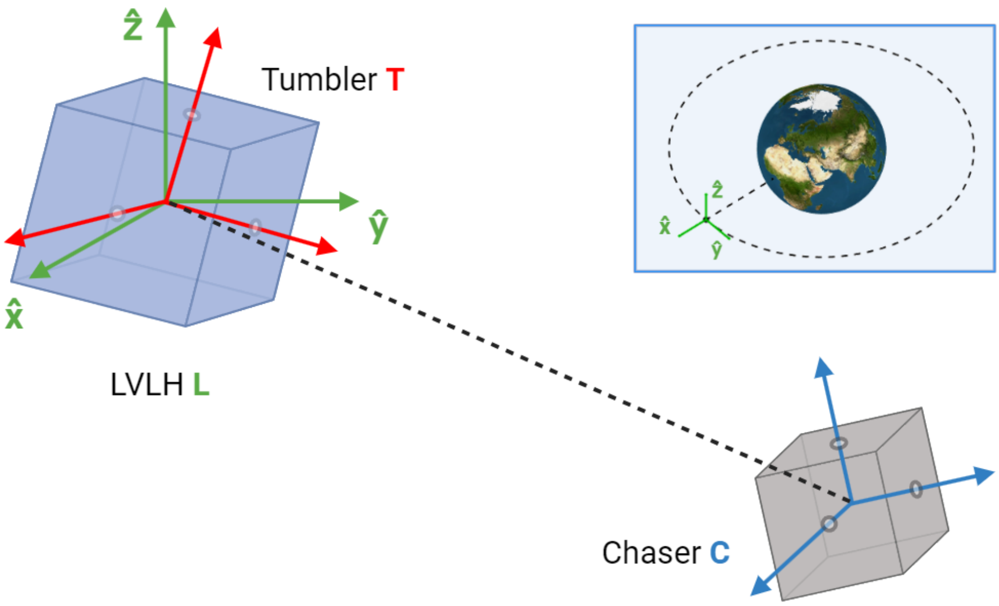


Figure 2.1: Depiction of the problem frames of reference

2.2. Translation dynamics

2.2.1. Continuous Linearized Dynamics

The translation dynamics govern the motion of the chaser relative to the target center of mass in LVLH frame. The main simplifying assumptions used are:

1. The chaser is in the final approach phase, meaning it will remain in an area of at most hundreds of meters from the target.
2. The target's orbit is considered approximately circular

The linearized dynamics in a non-inertial LVLH frame, known as the Clohessy-Wiltshire equations, can then be utilized. These equations can be expressed in state space form as shown in eq. (2.1). The state matrix **A** and input matrix **B** are provided in eq. (2.2), where Ω is the orbital angular velocity and m_c the mass of the chaser. These equations adequately model the spacecraft's dynamics, offering reduced computational costs compared to nonlinear dynamics.

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{2.1}$$

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3\Omega^2 & 0 & 0 & 0 & 2\Omega & 0 \\ 0 & 0 & 0 & -2\Omega & 0 & 0 \\ 0 & 0 & -\Omega^2 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \frac{1}{m_c} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

2.2.2. Discrete time linear dynamics

Since the control pipeline uses an MPC controller which is a discrete-time controller the translation dynamics must be discretized. This is done by operating a Zero-Order Hold (ZOH) discretization, which does not introduce errors compared to the continuous one if the inputs are staircase inputs. The dynamics are then expressed as a difference system of equations, reported in eq. (2.3) and where $\mathbf{A}_d$ and $\mathbf{B}_d$ are obtained from $\mathbf{A}$ and $\mathbf{B}$ as detailed in eq. (2.4), where T is the discrete time step.

$$\mathbf{x}_{k+1} = \mathbf{A}_d \mathbf{x}_k + \mathbf{B}_d \mathbf{u}_k \tag{2.3}$$

$$\begin{aligned} \mathbf{A}_d &= e^{\mathbf{A}T} \\ \mathbf{B}_d &= \int_0^T e^{\mathbf{A}\tau} d\tau \mathbf{B} \end{aligned} \tag{2.4}$$

2.3. Angular dynamics

The angular dynamics of both the target and the chaser spacecraft can be represented in a first approximation as the Euler equations for a rigid body, expressed in the respective frames of the bodies. It is important to note that the dynamics of the target spacecraft, reported in eq. (2.5), is not controllable since it is assumed that the target is uncooperative, the worst case condition in these kinds of maneuvers.

$$\mathbf{J}_T {}^I\dot{\boldsymbol{\omega}}_T^T + {}^I\boldsymbol{\omega}_T^T \times \mathbf{J}_T {}^I\boldsymbol{\omega}_T^T = \mathbf{0} \tag{2.5}$$

Instead, the chaser is considered fully actuated and the angular dynamics are described in eq. (2.6).

$$\mathbf{J}_C{}^I\dot{\boldsymbol{\omega}}_C^C + {}^I\boldsymbol{\omega}_C^C \times \mathbf{J}_C{}^I\boldsymbol{\omega}_C^C = \boldsymbol{\tau} \tag{2.6}$$

The kinematics depend upon the parametrization chosen for the orientation of the rigid bodies. It was decided to use quaternions as they are more computationally efficient than the alternatives and they don't have mathematical singularities. The unit norm condition is maintained by a normalization operation that is applied at every time step. Then the kinematics are represented as eq. (2.7). It is reported only the target case as the chaser case is identical.

$${}^L\dot{\mathbf{q}}^T = \frac{1}{2}\mathbf{A}_T{}^L\mathbf{q}^T \tag{2.7}$$

Where the matrix $\mathbf{A}_T$ is the kinematics update matrix of the quaternions shown in eq. (2.8).

$$\begin{bmatrix} 0 & {}^L\omega_{L_Z}^T & -{}^L\omega_{L_Y}^T & {}^L\omega_{L_X}^T \\ -{}^L\omega_{L_Z}^T & 0 & {}^L\omega_{L_X}^T & {}^L\omega_{L_Y}^T \\ {}^L\omega_{L_Y}^T & -{}^L\omega_{L_X}^T & 0 & {}^L\omega_{L_Z}^T \\ -{}^L\omega_{L_X}^T & -{}^L\omega_{L_Y}^T & -{}^L\omega_{L_Z}^T & 0 \end{bmatrix} \tag{2.8}$$

The angular velocity of the satellite expressed in the LVLH frame is obtained as eq. (2.9), and the direction cosine matrix can be constructed from the quaternions.

$$\begin{bmatrix} {}^L\omega_{L_X}^T \\ {}^L\omega_{L_Y}^T \\ {}^L\omega_{L_Z}^T \end{bmatrix} = {}^L\mathbf{R}^T \begin{bmatrix} {}^I\omega_{T_X}^T \\ {}^I\omega_{T_Y}^T \\ {}^I\omega_{T_Z}^T \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \Omega \end{bmatrix} \tag{2.9}$$

3 | Trajectory Generation

The first step in the guidance and control pipeline is the generation of an optimal trajectory which the controller can track to reach the rendezvous point and end the maneuver successfully.

The resolution of an optimal control problem such as a trajectory optimization problem can be done through two main families of methods:

1. In the indirect approach, the differential optimization problem is treated considering first-order necessary conditions, and expanded by a costate. The problem then becomes a two-point boundary value problem (TPBVP), and thus it can be solved with a single shooting method.

2. The direct approach consists of transforming the continuous optimization into an approximated discrete one, by using a discretized trajectory and an initial guess that on multiple iterations will possibly converge to the optimal solution.

Both methods are dependent on unknown and problem-specific quantities. For the indirect methods, the initial costate consists of variables with no physical meaning that can have values of very different magnitudes. Their search can be automated but it consists of a trial and error process which can be rather expensive, although when they are found the values will be similar for small variations of the problem. For the direct methods, instead an initial guess trajectory is necessary, if this is very far away from the optimal solution the iterative process will be more difficult, a tradeoff is also present, as a finer discretization is more computationally expensive but allows for more expressive solutions.

For this thesis the direct approach is chosen, the first part of the implementation will consist of determining a good initial guess to pass to the optimal control solver. The following sections are organized as follows: in section 3.1 the end condition is generated assuming nominal motion; in section 3.2 the initial guess for the optimization problem is created; while in section 3.3 the optimal control problem that generates the reference trajectory is defined and the numerical solution described.

3.1. Determination of End State in Orbital frame

The motion planning unit receives the estimator's best estimate of the current tumbler angular state. This estimate serves as the initial state for propagating the dynamics outlined in eq. (2.5) over a specified maneuver duration. This operation is permissible because the MPC controller manages the impact of measurement uncertainty on the generated trajectory. Then the conditions at the rendezvous point require the chaser satellite to be at a certain distance from one of the tumbler faces and to have a specific velocity computable with the formulae in eq. (3.1)[5].

$$\begin{cases} \mathbf{p}_f = {}^{L}\mathbf{R}^{T}\mathbf{p}_{berth} & (3.1a) \\ \mathbf{v}_f = {}^{L}\boldsymbol{\omega}_{L}^{T} \times {}^{L}\mathbf{R}^{T}\mathbf{p}_{berth} & (3.1b) \end{cases}$$

The propagation was implemented with `ode45` the Matlab implementation of the Runge-Kutta integration method with variable integration time step. It is important to choose a high accuracy such that the normalization of the quaternions happens with a high enough frequency and the unit norm is assured.

3.2. Determination of a Guess Trajectory

After the establishment of the initial and terminal points, a preliminary trajectory for optimal control can be constructed. The methodology employed for this generation ensures that the initial and terminal conditions of both position and velocity are either adhered to or closely approximated. Furthermore, the trajectory is designed to preserve a certain level of continuity. For these reasons, it was decided to use a spline composed of Bezier cubic curves as the initial guess.

3.2.1. Bezier curves

A spline is simply a succession of curves that are joined to respect a certain degree of continuity, in this case, Bezier curves are chosen as the building blocks of the spline. These curves are C^2 continuous and their position in space and tangent can be computed at every point in between the 2 endpoints, once the control points $\mathbf{P}_1$ and $\mathbf{P}_2$ are determined. The mathematical formula that is used to obtain the position and its derivative are reported in eq. (3.2)

$$\begin{cases} \mathbf{B}(t) = (1-t)^3\mathbf{P}_0 + 3(1-t)^2 t\mathbf{P}_1 + 3(1-t)t^2\mathbf{P}_2 + t^3\mathbf{P}_3, & (3.2a) \\ \mathbf{B}'(t) = -3(1-t)^2\mathbf{P}_0 + 3(1-4t+3t^2)\mathbf{P}_1 + 3(2t-3t^2)\mathbf{P}_2 + 3t^2\mathbf{P}_3, & (3.2b) \end{cases}$$

While the knot points for each segment are placeable, the control points $\mathbf{P}_1$ and $\mathbf{P}_2$ must be set for each segment such that the following condition of continuity and two natural boundary conditions, as shown in eq. (3.3).

$$\begin{cases} \mathbf{B}'_i(0) = \mathbf{B}'_{i-1}(1), & (3.3a) \\ \mathbf{B}''_0(0) = 0, & (3.3b) \\ \mathbf{B}''_{n-1}(1) = 0, & (3.3c) \end{cases}$$

By substituting eq. (3.2) in eq. (3.3) a linear system is obtained. This system is tri-diagonal and can be efficiently solved using the Thomas Algorithm. Once the control points are set, arbitrary points on the guess trajectory can be generated.

The spline is constituted by a first curve, passing through the initial position and the rendezvous point, and a second curve added such that the termination of the first curve is nearly parallel to the final velocity required for a successful rendezvous. This is achieved practically by selecting the third point far enough along the final velocity direction to ensure parallelism. Then the position and velocity guess values can be obtained by using equations eq. (3.2), where the velocity norm is fixed by linearly interpolating the magnitude between the values of the two extremes. A guess for the action should also be generated, this is chosen as zero for all the times, since the objective is to minimize the action this seems like a reasonable choice.

3.3. Optimal Control for motion planning

In this section, the optimal control problem is formulated as a specific realization of the general problem structure with:

1. the cost function J, which can be decomposed in a terminal part depending on initial and final times and the time integral of a time-dependent part, called running cost, as reported in eq. (3.4a).
2. the dynamical system that controls the evolution of the state, reported in its most general form in eq. (3.4b), and already specified in the previous chapter as eq. (2.1)
3. the constraint equations which can take the form of inequality constraints eq. (3.4c)

or terminal constraints eq. (3.4d)

$$\begin{cases} J\left(\mathbf{x}(\cdot),\mathbf{u}(\cdot),t_0,t_f\right) := E\left(t_0,\mathbf{x}_0,t_f,\mathbf{x}_f\right) + \displaystyle\int_{t_0}^{t_f} F\left(t,\mathbf{x}(t),\mathbf{u}(t)\right)\mathrm{d}t, & (3.4a)\\ \dot{\mathbf{x}}(t) = \mathbf{f}\left(t,\mathbf{x}(t),\mathbf{u}(t)\right), & (3.4b)\\ \mathbf{g}\left(t,\mathbf{x}(t),\mathbf{u}(t)\right) \leq \mathbf{0}, & (3.4c)\\ \mathbf{h}\left(t_0,\mathbf{x}_0,t_f,\mathbf{x}_f\right) = 0. & (3.4d) \end{cases}$$

It is important to note that the problem can have a fixed end time or a moving one, in this case, the end time is constrained to a fixed maneuver time chosen beforehand.

3.3.1. Objective Function

In the specific case of this problem, there is no endpoint cost, but only a running cost taken as the L2 norm squared of the actions eq. (3.5), this identifies the need for the solution to minimize the propellant used during the maneuver. This objective function is chosen instead of the sum of the absolute values of the control action itself because it has better continuity properties.

$$F\left(t,\mathbf{x}(t),\mathbf{u}(t)\right) = \frac{1}{2}\mathbf{u}^T\mathbf{u} \tag{3.5}$$

The analytical gradients of the objective function are provided as eq. (3.6). By providing the gradients the optimization solver is not required to compute the derivatives numerically by finite differences. This decreases the computational cost and allows the use of more accurate information on the direction of descent.

$$\frac{\partial F}{\partial \mathbf{u}} = \mathbf{u} \tag{3.6}$$

3.3.2. Constraints

The constraints of the problem are of two kinds. The first kind requires that the initial and end conditions on position and velocity must be respected, then these are set as in eq. (3.7)

$$\begin{cases} \mathbf{p}\left(t_0\right) = \mathbf{p}_0 & (3.7a) \\ \mathbf{v}\left(t_0\right) = \mathbf{v}_0 & (3.7b) \\ \mathbf{p}\left(t_f\right) = \mathbf{p}_f & (3.7c) \\ \mathbf{v}\left(t_f\right) = \mathbf{v}_f & (3.7d) \end{cases}$$

Moreover, the reference track that is created should include a guarantee for collision avoidance. This means that at all times, considering the tumbler motion that originates from the nominal initial tumbler state, the chaser remains outside of the geometry of the target. This collision avoidance constraint must be implemented on a case-by-case basis considering the shape of the target and a certain safety factor that augments the non-accessible space. This augmentation is done to avoid solutions in which the chaser's center of gravity is out of the ellipsoid but its geometry, approximated as a small sphere, can collide with the tumbler's body.

Once a geometrical model of the target is known, a keep-out zone can be represented as an ellipsoid. This ellipsoid is defined via a matrix $\mathbf{P}$ and a point $\mathbf{x}_c$. The matrix $\mathbf{P}$ has diagonal elements with values equal to the reciprocals of the squares of the semi-axes, so the extension in each semi-axes can be controlled by changing these elements. The point $\mathbf{x}_c$ is used to set the center of the ellipsoid, which in general will not be zero if the model is not symmetric about the center of gravity. The inequality that requires the chaser to be at all times outside of this zone is reported in eq. (3.8). Since the target is tumbling the position of the chaser in CW frame $\mathbf{p}$ must be rotated with an appropriate rotation matrix, variable in time but which can be pre-computed since the nominal expected target motion is used.

$$g_k(\mathbf{p}) = 1 - \left({}^{T}\mathbf{R}^{L}(k)\mathbf{p} - \mathbf{x}_c\right)^T \mathbf{P} \left({}^{T}\mathbf{R}^{L}(k)\mathbf{p} - \mathbf{x}_c\right) < 0 \quad k = k_0 \ldots k_N \tag{3.8}$$

Similarly to what was done for the objective function, analytical derivatives that define the Jacobian of the constraints are provided to the optimal solver. The only non-zero derivatives are represented in eq. (3.9), where the rotation matrix notation is reduced to $\mathbf{R}$ to avoid confusion with the transpose operation.

$$\frac{\partial g_k}{\partial \mathbf{p}} = -2\mathbf{R}^T(k)\mathbf{P}\left(\mathbf{R}(k)\mathbf{p} - \mathbf{x}_c\right) \quad k = k_0 \ldots k_N \tag{3.9}$$

Since these collision avoidance constraints can only be assigned at the specific times that are generated by the discretization of the optimal control it is important to have a grid dense enough such that situations such as those depicted in fig. 3.1 are avoided. The issue is reported again in the next section.

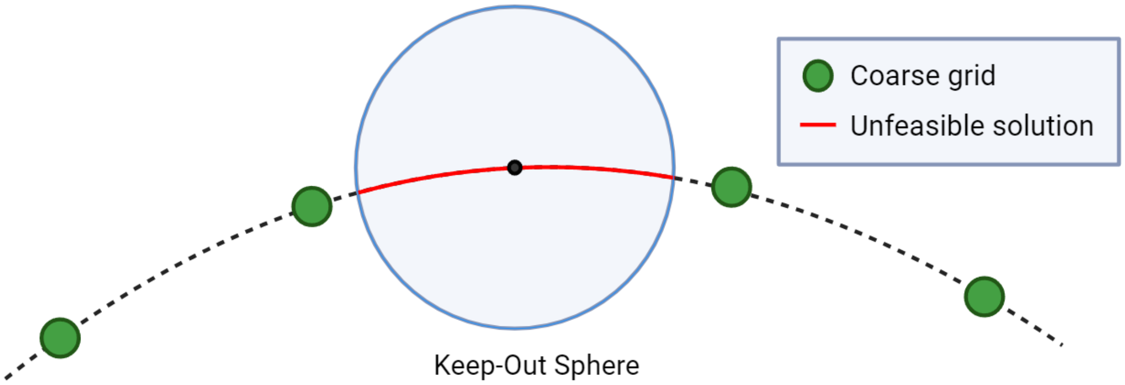


Figure 3.1: Importance of a finer grid for collision avoidance

The last constraint that is considered is the limitation of the control action, presented in eq. (3.10), the $\mathbf{u}_{lim}$ is the maximum control action that the chaser spacecraft can use at a certain time.

$$I_{3\times3}\mathbf{u} \leq \mathbf{u}_{lim} \quad \text{and} \quad -I_{3\times3}\mathbf{u} \leq \mathbf{u}_{lim} \tag{3.10}$$

3.3.3. Numerical solution of the optimal control

Once the mathematical problem has been defined, it is solved by application of the so-called direct methods. For this work, the Matlab OptimTraj[14] library was used. The OptimTraj offers implementations of trapezoidal, Hermite-Simpson and Chebyshev-Lobatto collocation methods, these differ by the way they provide a discretized approximation of the state and action solution trajectory. Due to the simplicity of the optimization problem, the trapezoidal direct collocation was used in this work. The trapezoidal method represents the system dynamics as constraints as shown in eq. (3.11a). The path constraints, such as the collision avoidance constraint of this problem can be discretized at the collocation points as in eq. (3.11b). Then the action and the state can be approximated by low-order splines as in eq. (3.11c) and eq. (3.11d). This completes the discretization of the optimal control problem that can be now solved with nonlinear programming solvers such as `fmincon`.

$$\begin{cases} x_{k+1} - x_k = \dfrac{1}{2} h_k \left(f_{k+1} + f_k \right), \quad k = 0, \dots, N-1 & (3.11a) \\ g(t_k, x_k, u_k) < 0 \quad \forall k & (3.11b) \\ u(t) \approx u_k + \dfrac{\tau}{h_k}(u_{k+1} - u_k) & (3.11c) \\ x(t) = \displaystyle\int_0^t \dot{x}(t)\, dt \approx c_1 + f_k \cdot \tau + \dfrac{\tau^2}{2h_k} \left(f_{k+1} - f_k \right) & (3.11d) \end{cases}$$

As introduced before due to the discretization the collision avoidance constraints are assured only at the instant of time of the discretization grid. Then a balance must be found, between the safety level provided by these constraints and the computational cost, which increases with the number of grid points. For this problem, since the constraints themselves are not that computationally expensive the number of grid points can be increased while the computational time remains in the order of minutes.

4 | Generation of Disturbance bounds

The Tube-MPC method is a robust controller that provides a robustness guarantee only for a bounded level of disturbance. It is then necessary to provide a good guess of the bounds of the disturbance that acts on the system. In this chapter, the implementation of the method that provides these bounds is described: in section 4.1 the origin of the disturbance is described in detail; in section 4.2 the algorithm for the production of the bounds is laid out; section 4.3 consists of a discussion on the alternatives in computing the bounds.

4.1. Description of the disturbance

The disturbance that affects this problem is not as simple as an unmodeled environmental force that disturbs the nominal dynamics. Instead, it is a consequence of the uncertainty on the initial target state and its inertial parameters. The trajectory generator computes a reference with collision avoidance guarantees in the tumbler frame. This trajectory can then be transformed to the orbital frame by a rotation, using rotation matrixes that are affected by errors due to the uncertainties at the initial time. This operation is illustrated in eq. (4.1a). As the system executes the maneuver, new estimates of the current orientation of the tumbler in the orbital frame become available. These can be utilized as in eq. (4.1b) to obtain the actual reference trajectory in the orbital frame. Every time new estimates about the tumbler motion are obtained the chaser will be required to track a new updated trajectory in the orbital frame, the disturbance that will act upon the system can then be computed as in eq. (4.1c).

$$\begin{cases} {}^{L}\mathbf{x}_{nom}(i) = {}^{L}\mathbf{R}_{nom}^{T}\mathbf{x}_{ref}(i) & \text{(4.1a)} \\ {}^{L}\mathbf{x}_{est}(i) = {}^{L}\mathbf{R}_{est}^{T}\mathbf{x}_{ref}(i) & \text{(4.1b)} \\ \mathbf{w}(i) = {}^{L}\mathbf{x}_{est}(i) - {}^{L}\mathbf{x}_{nom}(i) & \text{(4.1c)} \end{cases}$$

The error dynamics equation, reported in eq. (4.2), is a difference equation that describes the evolution of the error during tracking of the reference trajectory by the MPC controller. Then this will be disturbed by $\mathbf{w}$ each time an update is received from the estimator because that updates the system knowledge of the right trajectory to track.

$$\mathbf{x}_{err}^{+} = \mathbf{A}_d\mathbf{x}_{err} + \mathbf{B}_d\mathbf{u}_{err} + \mathbf{w} \tag{4.2}$$

A stochastic dynamic equation for the robust controller to track has been derived. In the next section, a bound on the value of $\mathbf{w}$ is computed by using a Monte Carlo approach.

4.2. Monte Carlo method for bound generation

The disturbances bounds that are required by the tube-MPC can be generated by using a Monte Carlo approach, as detailed in algorithm 4.1. The algorithm takes as input the current best estimates of the tumbler state and inertial parameters, these are provided as statistics containing the mean value μ and covariance matrixes Σ. Also, the algorithm takes as input the reference trajectory in the target frame from the motion planning unit. Then for N_{trials} a random initial state and random inertia matrix are generated. This is done by assuming a Gaussian multivariate distribution for the rotation matrix, the angular velocity and the inertia matrix. The probability density function of these distributions is written as in eq. (4.3).

$$y = f(x, \mu, \Sigma) = \frac{1}{\sqrt{|\Sigma|(2\pi)^d}} \exp\left(-\frac{1}{2}(x-\mu)\Sigma^{-1}(x-\mu)'\right) \tag{4.3}$$

Given the randomly generated initial tumbler state and inertia matrix, the initial state can be propagated generating an alternative possible sequence of orientation for the tumbler. The propagation is done for a period T_{tube}, which defines the time before the invariants and the tube are recomputed. The orientations are used to bring the reference trajectory from the target frame to the orbital frame. Subtracting the nominal position and velocities from the alternative ones results in one possible realization of $\mathbf{w}$. This can be repeated by N_{trials} and all the various $\mathbf{w}$ found are the result of the Monte Carlo simulation. The bounds will tend to match the true ones as N_{trials} is increased.

Algorithm 4.1 Algorithm for the computation of bounds on the disturbance [2]

```
**Require:** output estimator ($\hat{\mathbf{q}}_0, \hat{\omega}_0, \Sigma_q, \Sigma_\omega, \sigma_J$), output reference ${}^T x_{ref,0:N}$
  propagate $\hat{\mathbf{q}}_0, \hat{\omega}_0$
  rotate the reference ${}^T x_{ref,0:N}$, obtain ${}^L x_{ref,0:N}$
  **for** $i = 1$ to $nTrials$ **do**
    using $\Sigma_q, \Sigma_\omega$ obtain realizations $\mathbf{q}_0, \omega_0$
    using $\sigma_J$ obtain possible realizations of $\mathbf{J}$
    propagate $\mathbf{q}_0, \omega_0$
    **for** $j = 1$ to $nTime$ **do**
      ${}^L z_{ref}(j) = {}^L\mathbf{R}^T(j) {}^T x_{ref}(j)$
    **end for**
    $\mathbf{w}(i) = {}^L x_{ref,0:N} - {}^L z_{ref,0:N}$
  **end for**
  $\mathbf{w}_{bound} \leftarrow$computed using max or a sigma level
  **return** $\mathbf{w}_{bound}$
```

4.3. Bound and alternatives

From these sequences of disturbances, useful bounds such as $\mathbf{w}_{\max}$ or $\mathbf{w}_\sigma$ can be obtained, here described as:

1. Taking the absolute values of the disturbance first, and then determining the maximum among all the realizations, allows for a conservative estimate of the disturbance $\mathbf{w}_{\max}$ that will act on the dynamical system at each time.

2. By determining the standard deviation over the realizations and the times a less conservative estimate of the bounds $\mathbf{w}_\sigma$ can be obtained.

In general, the conservative estimates tend to be too strong for the robust controller to keep up with them, even with a high control authority from the mission definition. This possibility is a characteristic of the tube control that is reported also in the original work[2]. The solution is typically to postpone the maneuver until more accurate estimates are available or to use a $\mathbf{w}$ based on a certain σ level.

5 | Tube Robustness and Generation of the Robust Set

Before introducing and implementing the MPC control itself it is necessary to use the previous results, in particular, the bound on $\mathbf{w}$, to determine the tube that guarantees the robustness of the tracking. In this chapter then: a description of the tube and some results about the guaranteed properties of the method are presented in section 5.1 and section 5.2; the approximation of the invariant set is reported in section 5.3; the computation of the tracking set is discussed in section 5.4; in section 5.5 the determination of tightened constraints that will act on the nominal MPC is presented.

5.1. Theoretical background Tube-MPC

The objective of this section is to define the uncertainties, the dynamic system and the theoretical results that allow for the control of the system under any uncertainty in the bounds.

5.1.1. Representation of the uncertainty

The deterministic dynamics of the rendezvous are perturbed by random disturbances such as:

1. **Additive Disturbance**: it is a random or unpredictable interference that is added to the system dynamics. This is the case for the disturbance $\mathbf{w}$ described in the previous chapter.
2. **Unmodeled dynamics**: they are present because the model used to describe the system is not a perfect representation of the real system. In this case they arise due to the discretization of the system dynamics and the CW approximation.
3. **Measurement noise**: it is a random disturbance that affects the measurements of the system. They are present as Gaussian white noise in the simulation.

In this work, the main disturbance is a specific additive uncertainty that derives from the repeated update of the estimate of the target angular motion state. The other disturbances are considered secondary in magnitude and assumed to be zero in the following. The bounds on this particular disturbance are computable as reported in chapter 4.

5.1.2. Representation of the system under uncertainty

The dynamical system in its most basic form is described as a difference equation such as the one in eq. (5.1), where ϕ represent the solution at time k given the control sequence $\mathbf{U} = (\mathbf{u}(0), \mathbf{u}(1), ...)$. In this case, the system is linear, so it can be rewritten in terms of state matrix and input matrix.

$$\begin{cases} \mathbf{x}^+ = f(\mathbf{x}, \mathbf{u}) = A\mathbf{x} + B\mathbf{u} & \text{(5.1a)} \\ \mathbf{x}(k) = \phi(k; \mathbf{x}, \mathbf{U}) & \text{(5.1b)} \end{cases}$$

Once a system with disturbances is considered as the one in eq. (5.2), then the solution ϕ doesn't depend only on $\mathbf{U}$ but also on the sequence $\mathbf{W} = (\mathbf{w}(0), \mathbf{w}(1), ...)$, where each $\mathbf{w}(k)$ takes their value from a specific set $\mathbb{W}$.

$$\begin{cases} \mathbf{x}^+ = f(\mathbf{x}, \mathbf{u}, \mathbf{w}) = A\mathbf{x} + B\mathbf{u} + \mathbf{w} & \text{(5.2a)} \\ \mathbf{x}(k) = \phi(k; \mathbf{x}, \mathbf{U}, \mathbf{W}) & \text{(5.2b)} \end{cases}$$

Then the dynamics can also be seen as a differential inclusion in the form of eq. (5.3a), where F is a set-valued map. Knowing the set of the disturbances $\mathbb{W}$, the set-valued map can be written with the system dynamics right-hand side as eq. (5.3b).

$$\begin{cases} \mathbf{x}^+ \in F(\mathbf{x}, \mathbf{u}) & \text{(5.3a)} \\ F(\mathbf{x}, \mathbf{u}) = f(\mathbf{x}, \mathbf{u}, \mathbb{W}) = \{f(\mathbf{x}, \mathbf{u}, \mathbf{w}) | \mathbf{w} \in \mathbb{W}\} & \text{(5.3b)} \end{cases}$$

This set map takes a state, a control action generated by a control law or a series of control laws with input from the state itself and returns a successor state $\mathbf{x}^+$. It becomes clear that the dynamical system in question cannot possess a singular solution. Instead for every realization $\mathbf{W}$ given the same initial state and control law a multitude of solution trajectories $S(\cdot)$ can be generated, one of these $\phi(\cdot)^* \in S(x)$ generated by a control law μ is reported in eq. (5.4). The definition of a tube containing these solutions is the base of Tube-MPC.

$$\phi^*(t) = \phi(t, x_0, \mu, \mathbf{W}) \tag{5.4}$$

5.2. Definition of the tube

To understand why a tube that controls the multitude of trajectories is a necessary technique, it is useful to quantify the computational requirements for a brute force alternative[23]. If classic MPC control were applied in this uncertain setting, it would require the control system to solve online an optimal control problem for all possible disturbances. This results in a control problem with a computational complexity proportional to the number of possible disturbances. In the most simple case where $\mathbb{W}$ is a polytope and the problem is linear, this number is equal to the number of vertexes of the bound polytope $\mathbb{W}$ elevated to N, the length of the MPC horizon. The prediction horizon controls the optimality of the method, so only low-performance controllers are admissible for real-time usage when considering this method.

The tube-based approach instead shows that all trajectories of the uncertain system lie in a bounded neighborhood of the nominal MPC trajectory[23]. It is specifically this bounded neighborhood that is called a Tube. Once the tube is determined, by controlling the nominal trajectory to respect some tightened constraint all the possible solutions are guaranteed to respect the original constraints.

Assuming that $\mathbf{w} \in \mathbb{W}$, and the set $\mathbb{W}$ is a convex subset of $\mathbb{R}^n$ containing the origin, a set containing all the possible points deviating from the nominal trajectory after a certain time i can be obtained. This is achieved by utilizing eq. (5.1) and eq. (5.3), which represent the nominal and perturbed dynamics respectively. By subtracting these, a difference equation representing the evolution of the error for the nominal dynamics is obtained, as shown in eq. (5.5).

$$\mathbf{e}^+ = A\mathbf{e} + \mathbf{w} \tag{5.5}$$

Considering a certain sequence of $\mathbf{w}(j)$ then the error at time i is computed as in eq. (5.6).

$$\mathbf{e}(i) = A^i\mathbf{e}(0) + \sum_{j=0}^{i-1} A^j\mathbf{w}(j) \tag{5.6}$$

For our problem $\mathbf{e}$ is considered zero at time t_0, then a sequence of sets $S(i)$ for which $e(i) \in S(i)$ can be obtained, meaning that $S(i)$ contains at each time i any possible realization of the error from the nominal trajectory. This is equivalent to the set sum in eq. (5.7), where $\oplus$ is the Minkowsky sum, defined as $A \oplus B = \{a + b | a \in A, b \in B\}$, additional information about its properties and application on bounded polyhedron can be found in the references[25].

$$S(i) = \sum_{j=0}^{i-1} A^j \mathbb{W} = \mathbb{W} \oplus A\mathbb{W} \oplus \cdots \oplus A^{i-1}\mathbb{W} \tag{5.7}$$

A tube $\mathbf{X}$ for the open-loop system can then be derived as the affine mapping of sets $X(i, x)$, expressed in terms of S_i as shown in eq. (5.8). This tube is centered in the nominal trajectory and it contains all the possible realizations of the dynamics under uncertainty given that it is bounded with the sets S_i.

$$\begin{cases} \mathbf{X} = (X(0; x_0), X(1; x_0, \mathbf{U}), \ldots, X(N; x_0, \mathbf{U})) & (5.8a) \\ X(i, x_0) = \{\hat{x}(i)\} \oplus S_i & (5.8b) \end{cases}$$

Not all tubes that can be chosen are equal, in particular in the following, tubes that are composed of robust positive invariant sets are considered. The mathematical definition of these invariants is:

Definition 1. *A robust positively invariant set is a set S such that $Ax + w \in S, \forall x \in S$ and $\forall w \in \mathbb{W}$*

In other words, if a trajectory starts or enters the set it cannot exit it under any realization of the disturbance. Given that the size of the invariant will impact the reduction of the constraints of the nominal MPC, it is useful to choose an invariant that has a small dimension. The minimum RPI can be defined as:

Definition 2. *The set $\mathbb{F}_\infty$ is the minimum RPI (mRPI) if it is RPI and it is contained by every closed RPI set of the system dynamics.*

In a paper by Kolmanovsky and Gilbert [16] the minimum RPI $\mathbb{F}_\infty$ was found to be equal to the infinite limit of eq. (5.7). Because of this, in the following sections $S(\infty)$ and $\mathbb{F}_\infty$ are used interchangeably. An algorithm for the computation of its approximation can be found in the same paper, and it will be described in the next section. Then a tube can be defined as bounded by this set around the nominal trajectory, and in practice it is often used this constant outer bounding set, as shown in eq. (5.9), as it is easier to compute than the tube with variable cross-section S_i.

$$X(i, x_0) = \{\hat{\mathbf{x}}(i)\} \oplus S(\infty) \tag{5.9}$$

In practice, the resulting set $S(\infty)$ usually becomes too large, this is because the tube was generated using an open-loop dynamics $\mathbf{A}$. To control the size of the tube a feedback law must be implemented, and it takes the form in eq. (5.10), where $\hat{\mathbf{x}}$ and $\hat{\mathbf{u}}$ are the nominal MPC states and actions.

$$\mathbf{u} = \hat{\mathbf{u}} + \mathbf{K}_{dr}(\mathbf{x} - \hat{\mathbf{x}}) \tag{5.10}$$

In the current work, $\mathbf{K}_{dr}$ is chosen by LQR design, such that stability of $\mathbf{A}_k = \mathbf{A} + \mathbf{B}\mathbf{K}_{dr}$ is always assured, the parameters $\mathbf{Q}$ and $\mathbf{R}$ can be used to obtain a mRPI set of reduced dimensions and obtain feasible tightened constraints. Then the error dynamics are governed by eq. (5.11a) and a more compact mRPI and tube can be defined, as reported in eq. (5.11b) and eq. (5.11c).

$$\begin{cases} \mathbf{e}^+ = A_K\mathbf{e} + \mathbf{w} & \text{(5.11a)} \\ S_K(i) = \sum_{j=0}^{i-1} A_K^j \mathbb{W} & \text{(5.11b)} \\ X(i, x_0) = \{\hat{\mathbf{x}}(i)\} \oplus S_K(i) & \text{(5.11c)} \end{cases}$$

Since $\mathbf{A}_K$ is for sure stable then $S_K(\infty)$ exists and it is a robust positive invariant [22], which means that if $\mathbf{e}(0) \in S_K(\infty)$ then $e(i) \in S_K(\infty)$ for all i and for all $\mathbf{W} \in \mathbb{W}^i$. This is the robustness guarantee used in the rest of the thesis. The computation of $S_K(\infty)$ is not easy, thus in the next session some results valid for this specific problem are presented such that the tube and tightened constraints can be more easily computed.

5.3. Computation of the tube

In the previous section, a suitable invariant set was identified as the mRPI. Since its definition in eq. (5.7) involves an infinite number of Pontryagin sums such that it is in general impossible to compute this quantity. This means that an approximation of the mRPI must be computed, the algorithm for this computation was presented in a paper by Raković [22]. The advantage of the algorithm in the paper is that it computes an approximation with guarantees regarding its accuracy. The algorithm takes as input a certain error bound ϵ, then the approximation is guaranteed to be in between of $\mathbb{F}_\infty$ and

$\mathbb{F}_\infty \oplus \mathbb{B}_n(\epsilon)$ where $\mathbb{B}_n(\epsilon)$ is a hypercube of dimension n and "radius" ϵ. The algorithm can be described as computing a sequence of $F(\alpha, s)$ sets which are an outer approximation of the mRPI and then scaling the final one by a certain amount dependent on α.

Even if $F(\alpha, s)$ can be a poor approximation, it is guaranteed that for s sufficiently large or α sufficiently small it will tend to $\mathbb{F}_\infty$. The smallest α for which the approximation reaches the desired level of accuracy is defined in eq. (5.12). A version in which the assumption of $\mathbb{W}$ being a bounded polytope is also provided as eq. (5.12b). In particular $\mathbb{W}$ takes the form $\mathbb{W} = \{w \in \mathbb{R}^n | f_i^T w \leq g_i, i \in \mathbb{I}\}$ and h_W is the support function of $\mathbb{W}$, which is computable by solving a LP.

$$\begin{cases} \alpha^\circ(s) \triangleq \min \{\alpha \in \mathbb{R} \mid A^s W \subseteq \alpha W\} & \text{(5.12a)} \\ \alpha^\circ(s) = \max\limits_{i \in \mathcal{I}} \dfrac{h_W\left(\left(A^s\right)^T f_i\right)}{g_i} & \text{(5.12b)} \end{cases}$$

To decide when to terminate the iterative algorithm a priori an error bound on the approximation $F(\alpha, s)$ is determined. In particular the intermediate quantity $M(s)$ is calculated as in eq. (5.13b), while the condition to achieve a ϵ error bound is eq. (5.13c).

$$\begin{cases} M(s) \triangleq \min\limits_{\gamma} \{\gamma \mid F_s \subseteq \mathbf{B}_\infty^n(\gamma)\} & \text{(5.13a)} \\ M(s) = \max\limits_{j \in \{1,\ldots,n\}} \left\{ \sum\limits_{i=0}^{s-1} h_W\left(\left(A^i\right)^T e_j\right), \sum\limits_{i=0}^{s-1} h_W\left(-\left(A^i\right)^T e_j\right) \right\} & \text{(5.13b)} \\ \quad \alpha(1-\alpha)^{-1} F_s \subseteq B_\infty^n(\varepsilon) \Longleftrightarrow \alpha \leq \dfrac{\varepsilon}{(\varepsilon + M(s))} & \text{(5.13c)} \end{cases}$$

The algorithm that computes a ϵ-approximation of the mRPI is reported in algorithm 5.1.

Algorithm 5.1 Algorithm for the computation of a ϵ approximation of mRPI [22]

```
Require: A and W and $\epsilon > 0$
  Choose any s (ideally, set s=0).
  repeat
    Increment s by one.
    Compute $\alpha^\circ(s)$ as in eq. (5.12b) and set $\alpha = \alpha^\circ(s)$.
    Compute M(s) as in eq. (5.13b).
  until $\alpha \leq \epsilon/(\epsilon + M(s))$
  Compute $F_s$ as the Minkowski sum eq. (5.11b) and scale it to give $F(\alpha, s) = (1-\alpha)^{-1} F_s$.

  return $F(\alpha, s)$
```

This algorithm requires the computation of a series of Minkowski sums. If the required error bound is very low the number of sums might become too great and the computation will require an unreasonable amount of time. Since the bounded uncertainty $\mathbb{W}$ is defined as a hypercube, it is possible to use alternative methods that take advantage of this fact. A hypercube is a member of the family of zonotopes [9], centrally symmetric polytopes that can be defined in terms of a center c and generators g_i as reported in eq. (5.14).

$$Z = \{x \in \mathbb{R}^n : x = c + \sum_{i=1}^{p} \xi_i g_i, \xi_i \in [-1, 1] \forall i = 1, \dots, p\} \tag{5.14}$$

Recently fast algorithms for the computation of the mRPI under zonotopical disturbances were developed[17], this thesis takes advantage of the one-step optimization algorithm developed in the paper by Trodden[29]. In appendix A validation of the ϵ-mRPI algorithm and a comparison with the zonotope algorithm in terms of computation time is presented.

5.4. Computation of the terminal set

When MPC is applied to a tracking problem usually additional constraints must be applied as end conditions. To guarantee the recursive feasibility of the problem a robust invariant should be used as a tracking set for the current optimization window. It is convenient to choose a large tracking set, a large set increases the applicability of the controller as the controller will be less strict about the choice of the initial conditions that allow for the end constraint to be met[18]. Following the literature guidelines the maximum RPI set is chosen as the end constraint, this is defined as:

Definition 3. *The set $\mathbb{O}_\infty$ is the maximum RPI (MRPI) if it is RPI and it contains every RPI set under the state constraints.*

A recursive algorithm for the computation of the MRPI set is implemented, the algorithm is specialized for the case in which the constraints Y and the disturbance bound $\mathbb{W}$ are polyhedrons[16]. The algorithm then considers the constraint set $Y = \{y \in \mathbb{R}^p : Sy \leq r\}$ which gets updated as $Y_t = \{y \in \mathbb{R}^p : Sy \leq r_t\}$. The update is defined as eq. (5.15), where h_W is the support function of the disturbance polyhedron, s_i are columns of S.

$$\begin{cases} r_0^i = r^i - h_W\left(D^T s_i\right) & \text{(5.15a)} \\ r_{t+1}^i = r_t^i - h_W\left(\left(CA^T B\right)^T s_i\right) & \text{(5.15b)} \end{cases}$$

The approximation of the MRPI is defined as $O_t = \{\phi \in \mathbb{R}^n : H_t\phi \leq g_t\}$. The initial approximation of the MRPI and the update expression are reported respectively in eq. (5.16) and in eq. (5.17).

$$H_0 = SC, \qquad g_0 = r_0, \tag{5.16}$$

$$H_{t+1} = \begin{bmatrix} H_t \\ SCA^{t+1} \end{bmatrix}, \qquad g_{t+1} = \begin{bmatrix} g_t \\ r_{t+1} \end{bmatrix} \tag{5.17}$$

The algorithm 5.2 computes the MRPI and apart from some rare conditions is such that $\mathbb{O}_\infty$ is finitely determinable. In appendix B a validation of the implementation of this algorithm against the original paper results can be found.

Algorithm 5.2 Algorithm for the computation of the MRPI [16]

```
Require: A, B, C, D, $\mathbb{W}$, Y
  Set t = 0
  Compute $r_0$ by eq. (5.15a)
  if $r_0 < 0$ then
    set $\mathbb{O}_\infty = \varnothing$, terminate
  end if
  repeat
    Compute $r_{t+1}$ using eq. (5.15b)
    if $r_{t+1} < 0$ then
      set $\mathbb{O}_\infty = \varnothing$, terminate
    end if
    Compute $H_{t+1}$ and $g_{t+1}$ using eq. (5.17)
    set t = t+1
  until $\mathbb{O}^i_\infty = \mathbb{O}^{i+1}_\infty$
  return $\mathbb{O}_\infty$
```

5.5. Computation of tightened constraints

Once the mRPI has been computed, then the MPC constraints on the state and action can be expressed in terms of the size of the invariant and sent to the controller. In its most general form, it can be written that the tightened constraints are related to the original constraints by eq. (5.18a) for the state and eq. (5.18b) for the action. Where $\ominus$ is the Pontryagin difference, defined as $A \ominus B = \{x | x + b \in A, \forall b \in B\}$.$\mathbf{K}_{dr}$ is the matrix that stabilizes the dynamical system, computed by solving an LQR on the nominal dynamics.

$$\begin{cases} \overline{\mathbb{X}} = \mathbb{X} \ominus S_\infty & (5.18a) \\ \overline{\mathbb{U}} = \mathbb{U} \ominus \mathbf{K}_{dr} S_\infty & (5.18b) \end{cases}$$

Considering that both $\mathbb{X}$ and $\mathbb{U}$ can be represented as a system of inequalities as in eq. (5.19), the formula for the tightened constraints can be simplified.

$$\begin{cases} \mathbf{C}_1 \mathbf{x} \leq \mathbf{d}_1 & (5.19a) \\ \mathbf{C}_2 \mathbf{u} \leq \mathbf{d}_2 & (5.19b) \end{cases}$$

The tightened constraints then can be represented as eq. (5.20), in a form ready to be used in the MPC control problem. The Pontryagin difference can be substituted by a simple difference by considering the extreme points of the invariant set.

$$\begin{cases} \mathbf{C}_1 \mathbf{x} \leq \mathbf{d}_1 - F_K(\infty) & (5.20a) \\ \mathbf{C}_2 \mathbf{u} \leq \mathbf{d}_2 - \mathbf{K}_{dr} F_K(\infty) & (5.20b) \end{cases}$$

6 | Tube Model Predictive Control

In this chapter, the controller that is tasked to track the results from the trajectory generation unit is defined. A description of the nominal MPC is presented in section 6.1; the robustness guarantee constraints typical of the tube variant are explained in section 6.2; in section 6.3 the Matlab implementation of the complete pipeline is described.

6.1. Nominal MPC

The need for a controller to complete the GNC pipeline arises from the fact that the trajectory generator is responsible for collision avoidance with the tumbling satellite and it relies on uncertain measures of the satellite attitude at the time of computation. This means that tracking the reference trajectory with the original tumbling motion does not offer reliable safety guarantees. As shown in fig. 6.1 the controller will receive both a reference trajectory in the target frame and the current best estimate of the tumbling motion of the target. This estimate is then propagated and used to rotate the trajectory in LVLH frame. The other inputs of the control unit come from the robust set generator and as such they are treated in a later section.

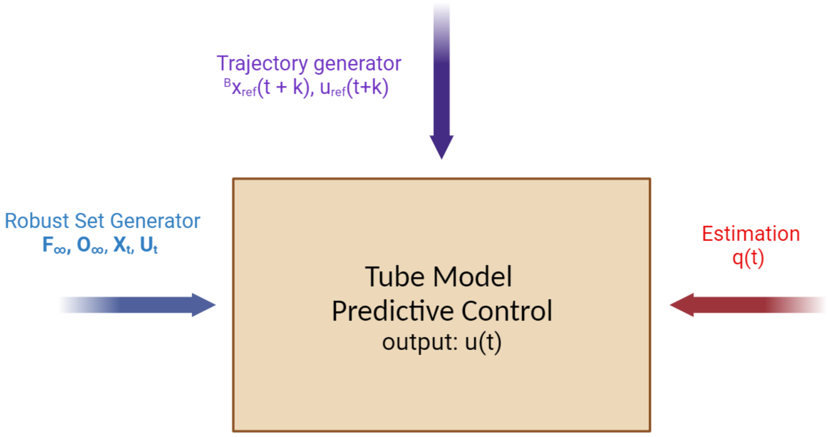


Figure 6.1: Input and outputs of the control unit

The MPC controller is based on the solution of an optimal control problem. This type of controller is unique because it performs optimization in real-time at every time step, which sets it apart from other traditional controllers. This is enabled by the recent implementation of fast nonlinear optimization solvers. Considerations about the applicability in real time of this particular controller can be found in chapter 7.

The mode of operation for the MPC controller is depicted in fig. 6.2. The controller operates using a discretized version of the model dynamics. It then solves an optimization problem which has as variables the state at the current time step and future actions $u(k+i)$ with $i \in [1 \cdots N]$. Using the discretized dynamics and these actions is possible to obtain a prediction of the system state for a certain time horizon N. The objective is to minimize the difference between the prediction and the reference trajectory, then the optimal action at the current time $u(k)$ is actuated for the whole time step T. The reliance of the method on a model of the real system underlines the importance of using a high-performance but also high-fidelity model, in the current work the relevant dynamics acting on the system are linear, enabling fast solutions to the optimization problem.

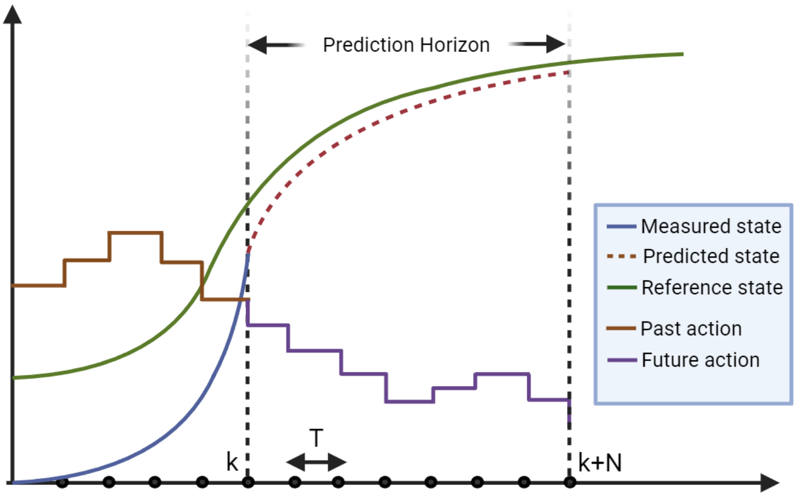


Figure 6.2: Example of MPC operation

A description of the various parts of the optimization problem that define the MPC can be found in the following.

6.1.1. Objective function

In the context of the current problem the optimization is implemented to solve a tracking problem, the reference trajectory is composed of nominal state and actuation which are rotated with the most recent estimates of the target orientation as reported in eq. (6.1).

$$\begin{cases} {}^{I}\mathbf{p}_{ref}(k) = {}^{I}R^{T}(k)^{T}\mathbf{p}_{ref}(k) & (6.1a) \\ {}^{I}\mathbf{v}_{ref}(k) = {}^{I}R^{T}(k)^{T}\mathbf{v}_{ref}(k) & (6.1b) \\ {}^{I}\mathbf{u}_{ref}(k) = {}^{I}R^{T}(k)^{T}\mathbf{u}_{ref}(k) & (6.1c) \end{cases}$$

Once these are computed the objective function can be written as eq. (6.2). The objective function is parametrized with 3 matrixes: the state weight matrix $\mathbf{Q}$ which controls the degree of penalty given to the tracking errors of the state reference; the action matrix $\mathbf{R}$ does the same for errors in the actions; the matrix $\mathbf{P}$ penalizes the system if the final state reference is not accurately reached.

$$\begin{aligned} \min_{\mathbf{u}(i),\mathbf{x}(i)} J = \sum_{i=0}^{N-1} \left(\mathbf{x}(i) - \mathbf{x}^{i}_{ref}\right)^{T} \mathbf{Q} \left(\mathbf{x}(i) - \mathbf{x}^{i}_{ref}\right) + \\ + \left(\mathbf{u}(i) - \mathbf{u}^{i}_{ref}\right)^{T} \mathbf{R} \left(\mathbf{u}(i) - \mathbf{u}^{i}_{ref}\right) + \left(\mathbf{x}(N) - \mathbf{x}^{N}_{ref}\right)^{T} \mathbf{P} \left(\mathbf{x}(N) - \mathbf{x}^{N}_{ref}\right) \end{aligned} \tag{6.2}$$

Equation (6.2) shows that the optimization variables are the sequence of discrete actions $\mathbf{u}(i)$ and the sequence of states $\mathbf{x}(i)$. This is the form in which the problem is implemented, it should be noted that as reported in the next section due to the dynamics constraints the only state variable that can be varied freely is $\mathbf{x}_0$.

6.1.2. Constraints

Another peculiarity of MPC control is that it allows for the implementation of constraints on the state and action variables. A first set of equality constraints is needed to guarantee that the discrete dynamics are respected at all future prediction times, this set is represented by eq. (6.3), the matrixes $\mathbf{A}_d$ and $\mathbf{B}_d$ can be found in eq. (2.2).

$$\mathbf{x}(i+1) = \mathbf{A}_d\mathbf{x}(i) + \mathbf{B}_d\mathbf{u}(i) \quad \forall i = 0 \cdots N-1 \tag{6.3}$$

Other constraints typical of nominal MPC are the upper and lower bounds of state and action variables. For a close-range rendezvous, the state constraints are: limitation on the distance from the target, since the close-range sensors, such as LIDAR, that allow for the reduction of uncertainty in the target states have a maximum range[13]. Even if a sensor suite with a sufficient range is considered, the validity of the Clohessy-Wiltshire approximation is another factor for which this limitation must be imposed, due to this reason the position is limited in a 100 m range box; The velocity is limited in a sensible range of ±5 m/s; The thrusters of the chaser spacecraft have limitations on the maximal thrust that can be employed at any moment, in this work the thrust is assumed to be limited in a ±100 N range; The whole set of upper and lower bound constraints is written in eq. (6.4) where i goes from 0 to N for the states and N-1 for the actions.

$$\begin{cases} -\mathbf{p}_{bound} \leq \mathbf{p}(i) \leq \mathbf{p}_{bound} & \text{(6.4a)} \\ -\mathbf{v}_{bound} \leq \mathbf{v}(i) \leq \mathbf{v}_{bound} & \text{(6.4b)} \\ -\mathbf{u}_{bound} \leq \mathbf{u}(i) \leq \mathbf{u}_{bound} & \text{(6.4c)} \end{cases}$$

Due to the robustness requirements, the actual bounds will be smaller than the ones initially allocated. As it was already reported in section 5.5 the bounds initially quantified in terms of sets operations can be rewritten in the box form of eq. (6.4).

Since this is a tracking problem an additional constraint on the end state is added. The final state is required to be inside a tracking set, for this problem it was chosen the maximum RPI set. The constraint is shown in eq. (6.5) and calculated as described in section 5.4. The end constraint is associated with the recursive feasibility over an infinite horizon, a property that the basic MPC lacks since it optimizes over a finite horizon.

$$\hat{\mathbf{x}} \in \{\mathbf{x}_{ref}(N)\} \oplus \mathbb{O}_\infty \tag{6.5}$$

6.2. Robust tube MPC

Tube MPC is one of the Robust implementations of model predictive control. The main changes made in addition to the nominal version of the controller are necessary to deal with the uncertainty affecting the system. The first variation is the addition of a disturbance rejection component to the nominal actuation. Since the additional component is proportional to the current estimated state it is more correct to say that the controller determines a series of control laws, a policy, instead of control actions. Equation (6.6) shows the control law implemented at step i until step $i + 1$. The matrix $\mathbf{K}_{dr}$ is the disturbance rejection matrix, it is chosen as the matrix that stabilizes the nominal linear system using LQR, this is a common choice in the literature[2].

$$\hat{\mathbf{u}}(i) = \mathbf{u}(i) + \mathbf{K}_{dr}\left(\hat{\mathbf{x}} - \mathbf{x}(i)\right) \tag{6.6}$$

In addition to the tightening of the box constraints defined in section 5.5, the linear inequalities constraints must be also expanded to account for the robustness guarantees.

Equation (6.7) constraints the real state to be within the minimal RPI centered around the initial predicted state. Once this is guaranteed then all successive states must continue to be within this tube due to the properties of the invariant set.

$$\hat{\mathbf{x}} \in \{\mathbf{x}(0)\} \oplus \mathbb{F}_\infty \tag{6.7}$$

The robust MPC was implemented using the CasaDi framework[3]. The details of the implementation are not crucial to the current dissertation, this is also because the framework itself has a symbolic abstraction layer which makes the actual code very similar to the mathematical counterpart. It is useful though to point out that the problem was set up using a vector of parameters containing the current state and the reference trajectory, this speeds up the computation as the initialization of the solver is done only once and at each time instant setting the parameters allows the fast resolution of the receding horizon optimization problem.

6.3. Full control pipeline

To effectively use the various components presented in the previous chapters, together with the Tube-MPC controller a simulation loop has to be implemented. Since the current work focused on the guidance and control part of the rendezvous problem it is necessary to implement a substitute of the estimation unit that provides measures and estimates of the states with a reasonable amount of uncertainty. A simulator that implemented eq. (2.1), the dynamics of the chaser in continuous time, and eq. (2.5), the angular dynamics of the tumbling target, is developed. On top of this, the measures coming from the simulator are returned to the control system after being varied by suitable amounts to provide an equivalent of the uncertainty of real estimates. These uncertainties consist of:

1. Gaussian white noise on the components of the chaser state.
2. Uncertainty on the quaternions describing the tumbler orientation, generated using a multivariate distribution and a covariance matrix Σ_q.
3. Uncertainty on the angular velocity of the target, generated using a multivariate distribution and a covariance matrix Σ_ω.
4. Uncertainty on the knowledge of the target inertia matrix J_T given by $\sigma_{i,j}$ for each component and contained in matrix $\delta\mathbf{J}$.

The algorithm that defines the simulation loop for the whole control pipeline is reported in algorithm 6.1. In it only the main steps of the Matlab code are reported, it is worth the effort to comment on some peculiarities. First of all the control law obtained at time t_i is determined as a handle function and it will use the most recent estimation of the state to determine a control action. The time step ΔT quantifies the step used to discretize the dynamics and the model predictive controller, while t_{est} is the time between an estimation step and the next one. Usually $\Delta T \gg t_{est}$, lowering ΔT can improve the performance of the model at the expense of computational cost.

The loop will also repeatedly compute disturbance bounds and the invariant sets. The amount of time between updates is given by T_{tube}. This quantity is chosen with a tradeoff between computational cost and the size of the disturbance from the trajectory update. T_{tube} is lower bounded by the computation time required to generate the bounds and the invariants, this is necessary to allow the algorithm to run in real-time. A lower T_{tube} means also that the robust set generator will consume more computational resources during the maneuver. On the other hand, if T_{tube} is increased too much the disturbances from a reference update will be too large and the tightened constraints will be empty, meaning that the controller cannot maintain the robustness guarantees. The computation of the invariant sets requires several seconds but it can be bound by a T_c duration. Once the time from the last update t_{last} becomes too close to T_{tube} the bounds and invariants are computed once again.

Algorithm 6.1 Algorithm for the simulation of the whole control pipeline

```
Require: $T_{\max}$ maneuver time, controller settings, dynamics settings.
  Setup and discretize dynamic system
  Compute $\mathbf{K}_{dr}$ using LQR and setup controller
  Determine a feasible trajectory in tumbler frame ${}^{T}x_{ref}(t)$
  Initialize simulator, obtain a first estimate of the state $x_{meas}$
  while $t < T_{\max}$ do
    Generate ${}^{T}x_{ref}(t_{0:N})$
    Rotate the reference to LVLH frame using $q_{meas}(i)$
    if $t_{last} + T_c \geq T_{tube}$ then
      Compute a bound on the disturbance $\mathbf{w}_{bound}$
      Compute minRPI $F_\infty$ and maxRPI $\mathbb{O}_\infty$
      Compute tightened constraints $\overline{\mathbb{X}}$ and $\overline{\mathbb{U}}$
    end if
    Solve the nominal part of Tube MPC, obtain $\mathbf{x}(i)$ and $\mathbf{u}(i)$
    $\hat{\mathbf{u}}(t) = \mathbf{u}(i) + \mathbf{K}_{dr}\left(\hat{\mathbf{x}} - \mathbf{x}(i)\right)$
    while $t < t_i + \Delta T$ do
      Obtain current action $\hat{\mathbf{u}}(t)$
      Apply action, simulate dynamics for $t_{est}$
      $t = t + t_{est}$
    end while
  end while
```

7 | Results

This chapter contains the results of the application of the guidance and control pipeline. The control system is applied to the realistic rendezvous scenario with the Envisat satellite in section 7.1. Ultimately a Monte Carlo analysis that verifies the robustness of the controller tracking is considered in section 7.2

7.1. Envisat mission scenario

7.1.1. Background and motivation

The mission scenario that was chosen for the application of the control pipeline is a rendezvous with the uncontrolled Envisat satellite. Envisat is an ESA Earth observation satellite. It was launched into orbit the 1 March 2002, to obtain data from a wide range of instruments, and improve the knowledge of Earth's physical and chemical phenomena. On 8 April 2012, ESA lost contact with the spacecraft, all attempts to regain control over the satellite were unsuccessful and ESA declared the end of Envisat's mission on 9 May 2012. The mission scenario was chosen primarily for the availability of data regarding the properties of the Envisat satellite. The satellite has received a lot of attention after ESA decided to fund a mission for its removal called e.Deorbit. The mission was then stopped in 2018 but reached the detailed design phase (B1), generating accurate information on the target and its presumed tumbling state[4]. The satellite poses a considerable threat to the space environment, it orbits a zone where close encounters happen yearly. Additionally, due to its high mass a violent hit would generate an extended cloud of debris potentially triggering further collisions.

7.1.2. Envisat model

The first element of the simulation is the Envisat satellite. The satellite mass and inertia properties are known within a certain uncertainty and are reported in table 7.2 and in table 7.3[7]. The inertia is expressed in the target body frame defined as the following:

1. **x axis**: positioned centrally to the service module body, towards the payload adapter.
2. **z axis**: parallel to the parabolic antenna, with the same direction.
3. **y axis**: completes the right handed coordinate system.

The coordinate system has the center of gravity of the whole spacecraft, including the solar panels, as the origin. A depiction is provided as fig. 7.1.

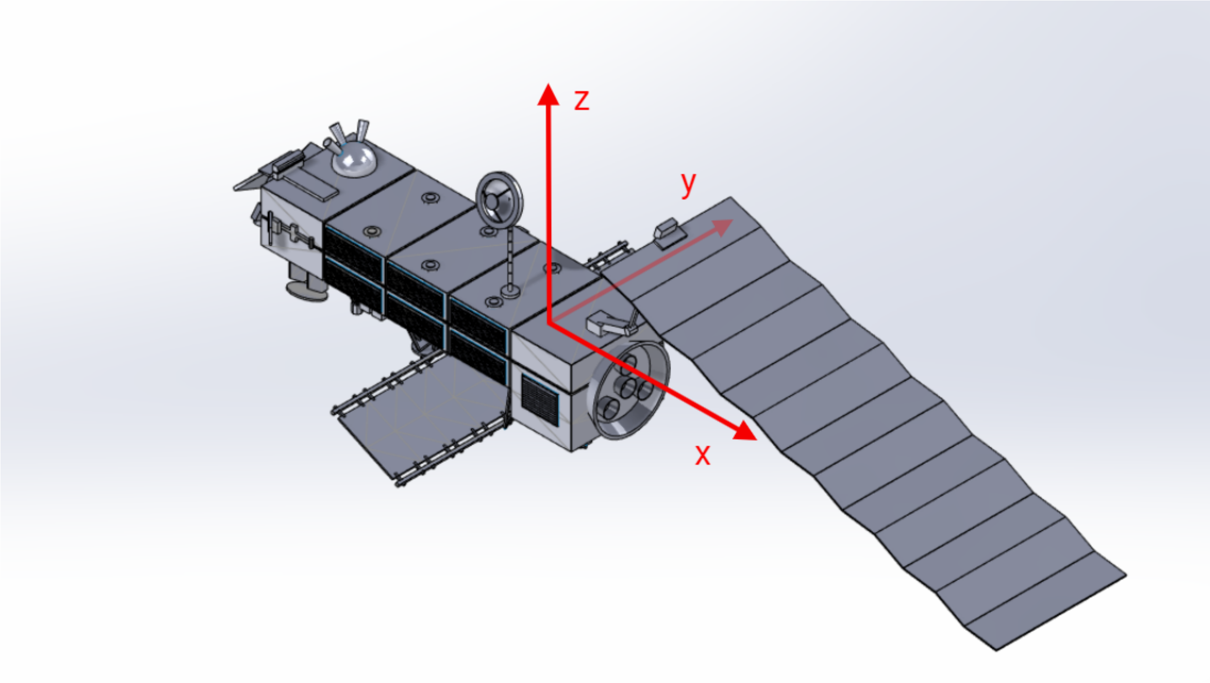


Figure 7.1: Target frame orientation

	x	**y**	**z**
x	17023	397.1	-2171
y	397.1	124826	344.2
z	-2171	344.2	129112

Table 7.1: Inertia matrix [kg m^2]

	Value	**Uncertainty**
Mass	7828	30

Table 7.2: Mass [kg]

Table 7.3 contains the uncertainty bound on the inertia. These are the ones that are known on-ground when no estimation process in the near range of the satellite has even begun. It is reasonable to assume that once the initial standoff estimation is completed these uncertainties will be reduced. Because of this assumption in the following some uncertainty values are considered 10 times lower than the standoff bounds, as they would have generated disturbances with magnitudes impossible to counteract.

	x	**y**	**z**
x	350	100	250
y	100	3000	150
z	250	150	3000

Table 7.3: Inertia uncertainty 1σ [kg m^2]

The attitude is not known accurately but it is believed to be bounded and inferior to 3.5 deg/s[28]. Due to multiple perturbation forces acting on the attitude dynamics, the rotation behavior is chosen as a generic tumble and not in a specific mode such as flat spin.

It is also known that the satellite orbits the Earth in an almost circular orbit, with the relevant orbital parameters reported in table 7.4. The low eccentricity means that the CW dynamics are a good model of the relative dynamics in the near vicinity of the spacecraft.

Orbital parameter	**Value**
Semi-major axis a	7144.8 km
Eccentricity e	0.0000982

Table 7.4: Relevant orbital parameters at mission end

Using the semi-major axis, the value of the orbital angular velocity Ω can be computed using eq. (7.1), where μ is the Earth's gravitational parameter.

$$\Omega = \sqrt{\frac{\mu}{a^3}} = 1.0454 \times 10^{-3} rad/s \tag{7.1}$$

7.1.3. Chaser model

The chaser object can be described with a cubic main body which is composed of subsystems for maintaining a nominal operational status, communicating with ground control and providing angular and linear motion capabilities. Additionally, it has a robotic arm which is considered to be in a stowed configuration for the entirety of the rendezvous. The whole spacecraft can be considered as bounded by a sphere and as such the control of the attitude can be decoupled from the control of the position. It will be sufficient to avoid the sphere entering contact with the target model, independently of the orientation of the chaser. To achieve a good close-range estimation it is assumed that an attitude control law which keeps the target in the FOV of the chaser instrument is employed.

The numerical data describing the chaser that is considered for the following work are reported in table 7.5.

Parameter	**Value**
Chaser mass	850 kg
Chaser box side	2.5 m
Bound sphere radius	$1.5\sqrt{2}$ m
Stretched arm length	4.5 m
Thruster max thrust	100 N

Table 7.5: Relevant chaser parameters

7.1.4. Reference generation

The first step in the solution of the rendezvous consists of generating a safe trajectory in the target frame, which can be tracked by the robust controller. The settings used for this particular rendezvous are reported in table 7.6.

Parameter	Value
Initial chaser position	[40, 0, 0] m
Initial chaser velocity	[0, 0, 0] m/s
Target orientation $\mathbf{q}$	[0.3826834, 0, 0, 0.9238795]
Target angular velocity	[1, 2, 1] °/s
Maneuver duration	300 s
Rendezvous point	[0, 0, -5.5] m

Table 7.6: Reference generation settings

The setup of the reference generator unit is completed once the collision avoidance is defined. The CAD replica of the Envisat satellite is too complicated to be used as a keep-out zone, then a Delaunay triangulation[33] of the model is generated such that is easier to visualize the convex hull of the object. The capability of the Multi Parametric Toolbox[11] is used again to generate an equivalent polyhedron. Since the chaser can assume any orientation it can be identified with an equivalent sphere, the Minkowski sum of this sphere with the satellite's convex hull generates a keep-out region. Then an equivalent ellipsoid is determined such that it envelopes the expanded hull with a certain degree of accuracy. The various objects and the final ellipsoidal keep-out zone are visible in fig. 7.2. The data of the ellipsoid defined by eq. (7.2) is reported in table 7.7.

$$(\mathbf{x}-\mathbf{x_c})^T\mathbf{P}(\mathbf{x}-\mathbf{x_c}) = 1 \tag{7.2}$$

Parameter	Value
Ellipsoid center $\mathbf{x}_c$	[1.5, 0, 0.75] m
Semi-major axis $1/\sqrt{P_{ii}}$	[17, 8, 6] m

Table 7.7: Ellipsoid definition data

The rendezvous point is right outside of the keep-out zone. Even if technically the fully stretched arm should reach the target main body, the ellipsoid is a conservative keep-out zone and it is assumed that after the maneuver the chaser would move closer to improve the workspace of the arm.

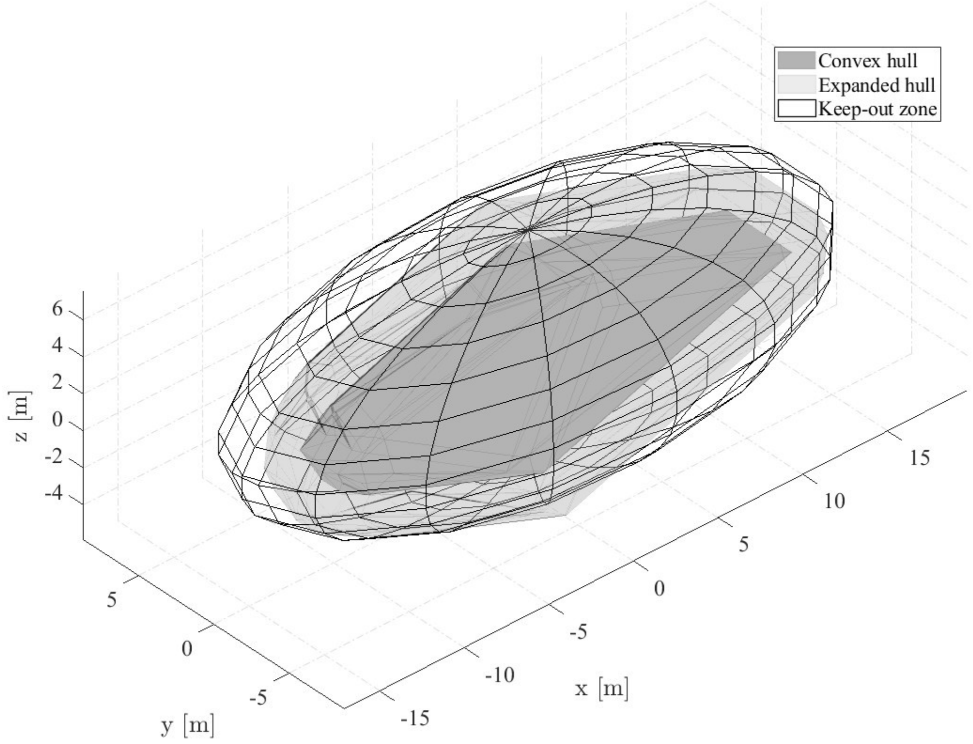


Figure 7.2: Keep-out zone, target frame

The optimal control problem is solved with the trapezoidal direct collocation method. At first, the solver applies a rough optimization with a grid of 40 points, and then a more precise one of 60 points is used to achieve a more accurate minimum. Given that the maneuver lasts 300 seconds this means that a collision constraint is active every 5 seconds, in the context of a real mission the constraints can be made more frequent at the cost of a higher computational time.

The result of the trajectory optimization is depicted in fig. 7.3. A simplified representation of the target is used to not obfuscate the image, the arrows represent the velocities at various points in the trajectories. The guess trajectory does not respect the system dynamics but it is good enough to allow the optimization process to obtain an optimal trajectory that respects the constraints. The optimization times with and without the analytical derivatives are reported in table 7.8, the method is fast enough that new trajectories could be generated online during the maneuver but this property is not exploited in the current work.

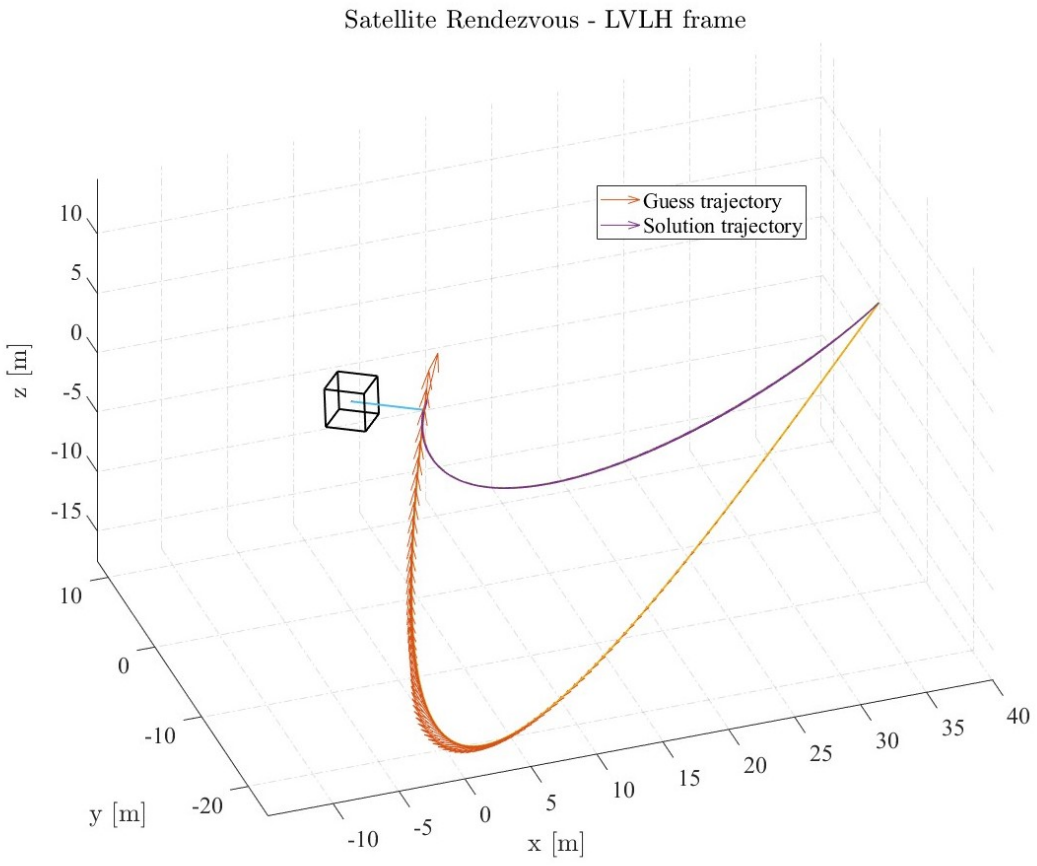


Figure 7.3: Trajectory generation, LVLH frame

	Mean value	Standard deviation
Analytical all	2.7061 s	0.0197 s
Analytical objective	2.9378 s	0.0138 s
Analytical constraint	4.4178 s	0.0456 s
Numerical all	4.9147 s	0.0410 s

Table 7.8: Time of trajectory generation

7.1.5. Uncertainty bound

The lack of an estimation unit, which is out of the scope of this work, means that the level of confidence in the simulated measures from the systems must be set a priori. For this work, they are chosen as table 7.9, ulterior experimental trials should verify that these assumptions hold also in a realistic setting. In the table `randn` is a procedure that generates a square matrix of the specified dimension with elements taken from a normal distribution, `symrandn` does the same but only for the upper triangular part of the matrix, the rest is mirrored.

	Helper quantity	**Uncertainty measure**
Covariance quaternion	$\mathbf{b}_q = 0.001\mathtt{randn}(4)$	$\mathbf{\Sigma}_q = \mathbf{b}_q^T\mathbf{b}_q$
Covariance angular velocity	$\mathbf{b}_\omega = 0.0005\mathtt{randn}(3)$	$\mathbf{\Sigma}_\omega = \mathbf{b}_\omega^T\mathbf{b}_\omega$
Uncertainty inertia	–	$\delta\mathbf{J}_T = 100\mathtt{symrandn}(3)$

Table 7.9: Definition of the uncertainty acting on the system

Algorithm 4.1 can be used to find a bound on the disturbance. In fig. 7.4 the sum of the standard deviations on the estimates of the various w_i is plotted against the number of trials. The average computation time of the procedure is also plotted. The reliability of the bounds plateaus after a certain number of trials while the time increases linearly. The bounds will be then generated for $n_{trials} = 500$.

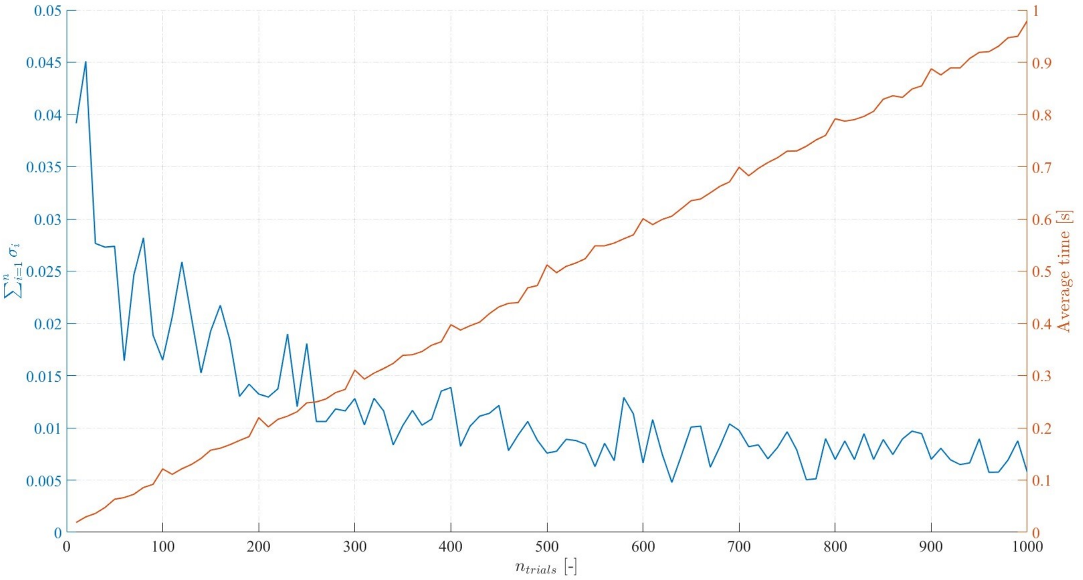


Figure 7.4: Information about bound generation

A sample of the Monte Carlo perturbed trajectories is visible in fig. 7.5. These are confronted with the nominal trajectory, as expected the maximal disturbance in a reference trajectory update will tend to happen toward the end of the prediction horizon.

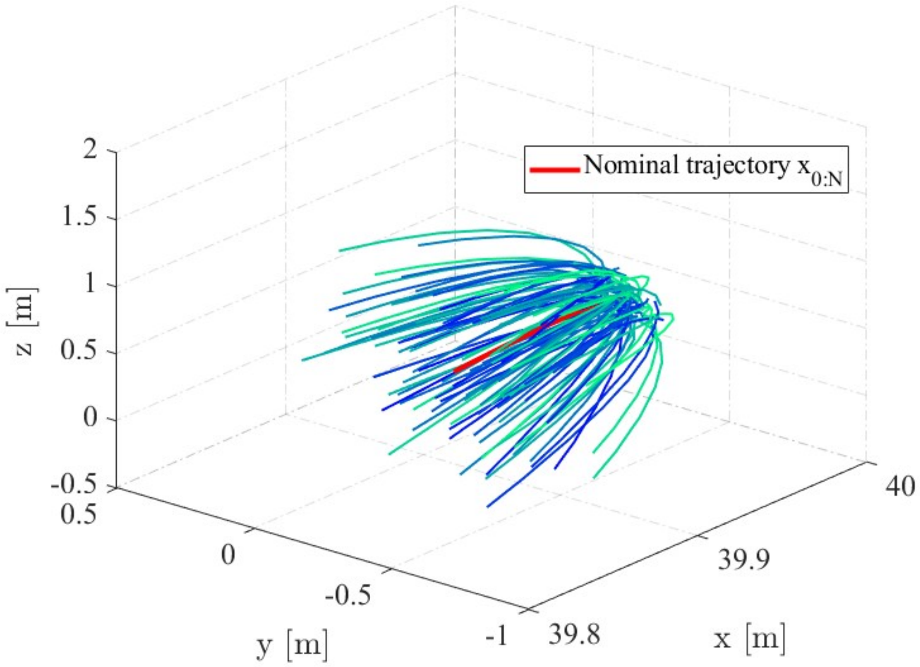


Figure 7.5: Nominal and sample of perturbed trajectories

The resulting bounds calculated both with a max approach and σ levels are reported in table 7.10.

	max bound	**σ bound**	**3σ bound**
x	0.03582	0.00295	0.00885
y	0.75931	0.10433	0.31299
z	1.50039	0.18279	0.54835
v_x	0.00031	0.00004	0.00012
v_y	0.00047	0.00005	0.00016
v_z	0.00091	0.00010	0.00031

Table 7.10: Uncertainty bounds at t_0

7.1.6. Generation of the RPI sets

The uncertainty bounds can be used to determine the maximum and minimum RPI sets and to tighten the constraints. The bounds obtained for the max and 3σ strategy generate empty tightened constraints. This means that this kind of uncertainty is too high for the robust controller to maintain its guarantees in all cases. Then in the following report, the results are limited to the case of 1σ bound with a more limited uncertainty. The need to consider a decreased uncertainty bound was already reported in the paper by Albee et al.[2] and does not compromise the applicability of the controller.

Using the algorithm 5.1 and algorithm 5.2 the mRPI for tube robustness and the MRPI for set tracking are obtained in the case for 1σ uncertainty bound. The invariant sets calculated at initial time t_0 are here reported in fig. 7.6 as lower dimensionality phase space plots in the 3 cartesian directions.

Before obtaining the tightened constraints the disturbance rejection gain matrix $\mathbf{K}_{dr}$ should be determined by an LQR procedure. By choosing $\mathbf{Q} = 1 \times 10^3 \mathbb{I}_6$ and $\mathbf{R} = \mathbb{I}_3$ the $\mathbf{K}_{dr}$ is obtained as shown in eq. (7.3).

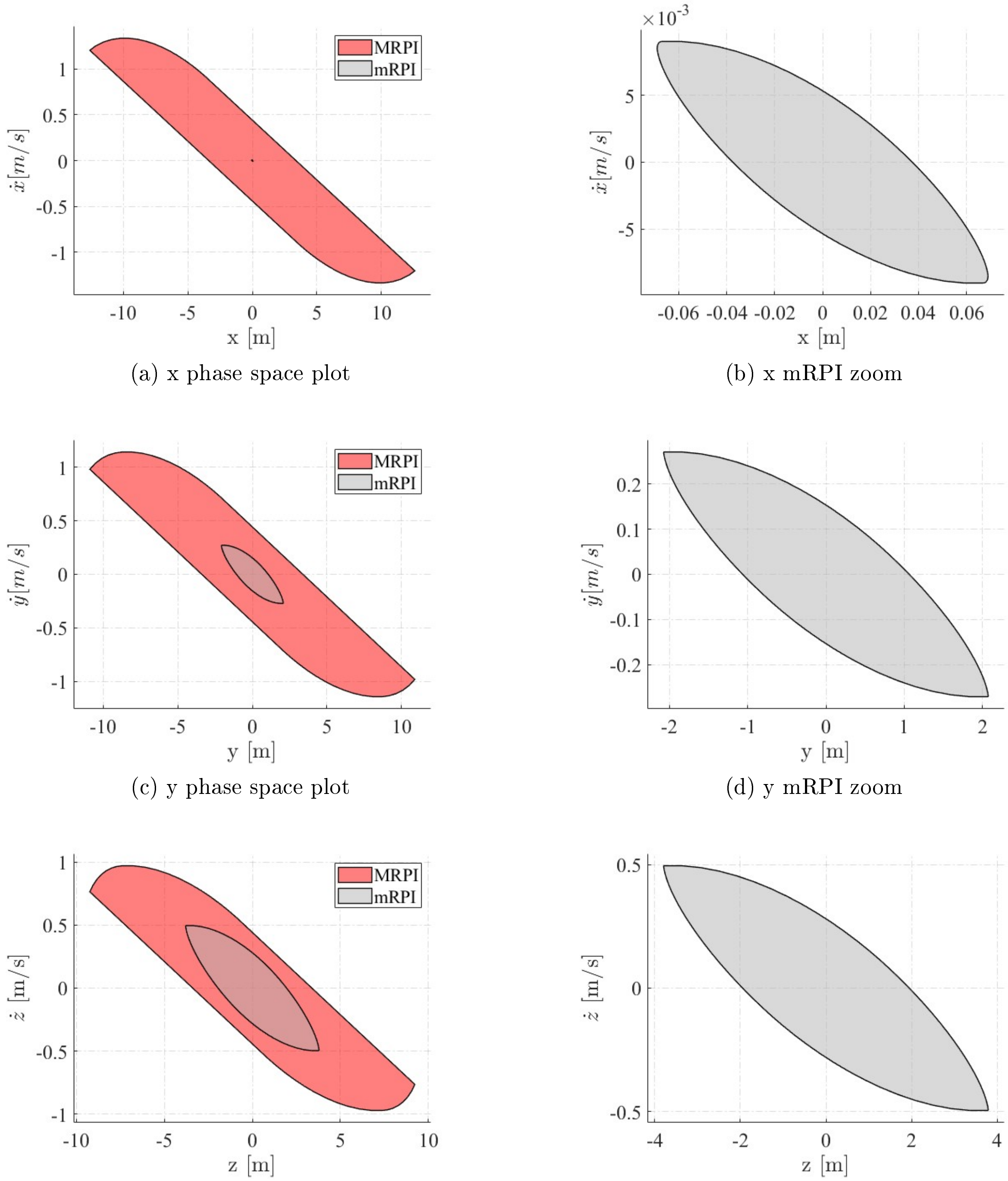
$$\mathbf{K}_{dr} = \begin{bmatrix} -29.523 & -0.015 & 0.000 & -225.970 & 1.660 & 0.000 \\ -0.015 & -29.520 & 0.001 & 1.660 & -225.960 & 0.006 \\ 0.000 & 0.000 & -29.519 & 0.001 & 0.001 & -225.951 \end{bmatrix} \tag{7.3}$$

Then the new constraints can be obtained from eq. (5.20) and are reported in table 7.11 along with the original ones.

	Original constraints	**Tightened constraints**
$\mathbf{x}$	±100 m	±99.931 m
$\mathbf{y}$	±100 m	±97.925 m
$\mathbf{z}$	±100 m	±96.205 m
$\mathbf{v}_x$	±5 m/s	±4.9909 m/s
$\mathbf{v}_y$	±5 m/s	±4.7289 m/s
$\mathbf{v}_z$	±5 m/s	±4.5041 m/s
$\mathbf{u}_x$	±100 N	±98.762 N
$\mathbf{u}_y$	±100 N	±64.651 N
$\mathbf{u}_z$	±100 N	±35.316 N

Table 7.11: Tightening of the constraints at t_0

The tightening of the constraints is particularly evident in the action space where there are drops greater than 50%. This could be due to the disturbance rejection gain which was chosen by a simple LQR procedure. This result is not unexpected as similar variations were reported in previous works[2][6].



(a) x phase space plot

(b) x mRPI zoom

(c) y phase space plot

(d) y mRPI zoom

(e) z phase space plot

(f) z mRPI zoom

Figure 7.6: Visualization of robust sets for Tube-MPC at t_0

The results for the uncertainty bounds and the minimum RPI set are reported for the initial time. The difference in magnitude of the bounds can be explained by the different magnitude of the Envisat inertia matrix values, although due to the nonlinear dynamics, the effect cannot be described by a simple law, this is why the bounds are computed with a Monte Carlo simulation. As the target orientation changes the bounds magnitudes and invariant set dimensions will change as well. The conditions for the updates of the bounds are reported in the next section.

Another peculiarity of the invariants is that the greater the disturbances in a dimension the bigger the mRPI and the smaller the MRPI set are when plotted in the same dimension. As the disturbance increases the MRPI might become empty while the mRPI is still computable. This might seem to be an impossibility but it is explained by the fact that the MRPI set is computed considering the constraints on the state space, while the mRPI by Minkowski sum considers only the system dynamics. When this happens a lower disturbance must be considered similarly to when the tightened constraints become empty sets.

7.1.7. Tube MPC controller settings

Before running the whole pipeline it is necessary to tune the MPC controller, the controller settings are then explored in their possible variations to obtain good tracking properties. The main parameters of the controller are reported in table 7.12, additionally the controller uses constraints obtained from the computed invariant sets and as such their application is also commented in the following.

Parameter	**Value**
T	0.5 s
N	20 [-]
$\mathbf{Q}$	$\texttt{diag}([1 \times 10^4, 1 \times 10^4, 1 \times 10^4, 1 \times 10^{-3}, 1 \times 10^{-3}, 1 \times 10^{-3}])$
$\mathbf{R}$	$\texttt{diag}([1 \times 10^3, 1 \times 10^3, 1 \times 10^3])$
$\mathbf{P}$	Obtained from stabilizing LQR
T_{tube}	10 s

Table 7.12: Parameters of MPC control

The discretization time T is the first parameter that was fixed. This parameter controls the frequency at which the nominal MPC action is updated, as such it is usually decreased to improve the controller behavior against unmodeled disturbances. A tradeoff arises since lower values increase the computational time of the controller during the maneuver. In this case, the disturbances are also handled by the ancillary controller, so the value of 0.5 s, common in the aerospace industry[8] is considered adequate.

The prediction horizon N is set by exploring a limited range of values. The performance of the tracking of the position shows increasingly better results in the range from 10 to 20, more details will be reported later in this section. Even for N chosen at the end of the explored range, the computational time remains minimal. Thus, a prediction horizon of N=20 is selected. In particular, the solution of the MPC takes an average of 0.02726 seconds, with σ of 0.0042 seconds.

The matrixes $\mathbf{Q}$ and $\mathbf{R}$ are kept diagonal for simplicity. The increase of one of the diagonal values means that the optimization process will focus more on the tracking of that particular variable. The chosen values are the result of a trial and error process.

The matrix $\mathbf{P}$ controls the weights for the last state in the optimization. Its value is chosen as the solution of the Riccati equation for the LQR that stabilizes the nominal system. This choice is made following the literature implementation[2].

The T_{tube} controls the time for the update of the uncertainty bounds and invariant sets. The computation was run multiple times to obtain a lower bound of the variable for real-time computation, the results of this study are reported in table 7.13. The computation time T_c can be upper bounded by 3 s and T_{tube} can be set to a 10-second value for which the tightened constraints are always non-null. In the MATLAB simulation, the computations are done sequentially for simplicity, though a real-time version with parallel computations for the update and MPC optimization must be implemented for the real-time application.

	Uncertainty	**Invariants**	**Total**
Maximum T_c	0.4583	2.1614	2.6197

Table 7.13: Maximum computation times for tube update, 100 trials

7.1.8. Tube MPC results

The whole pipeline is then initialized as described in the previous sections and run for a demonstrative rendezvous. The real tumbling conditions and inertia matrix of the target are hidden from the control pipeline for obvious reasons. To evaluate the performance of the controller the trajectory brought to the LVLH frame by considering the real tumbling motion of the target is computed. This trajectory is referred to as $\mathbf{x}_{real}$ and will be compared with the trajectory that was achieved in practice by the use of the MPC controller $\mathbf{x}_{mpc}$. The nominal trajectory $\mathbf{x}_{nom}$ is the reference trajectory in LVLH frame considering the knowledge of the target motion at t_0. It is shown to visualize the differences between the real trajectory to follow and highlight the importance of the control system in avoiding collisions with the target.

At first, the tracking performances of the controller are shown in fig. 7.7 for the position and in fig. 7.8 for the velocities. The position tracking performs exceptionally well, staying on target for the entirety of the maneuver. The tracking of the velocity instead is mostly unable to maintain the tracking, in particular when the real trajectory deviates the most from the nominal one, as evidenced at the 150-second mark. The tracking of the velocity probably could be improved by choosing other parameters for the MPC optimization. During the tuning of the controller, it was seen that partially the tracking of the velocity can be improved, at the expense of the tracking of the position. Since the main objective of the pipeline is to robustly follow a trajectory that has collision avoidance capabilities the tracking of the velocities is seen as secondary and the settings of the MPC are chosen to have good tracking capabilities for the positions. The final velocity constraints are also approximately respected and the rendezvous can be considered successful.

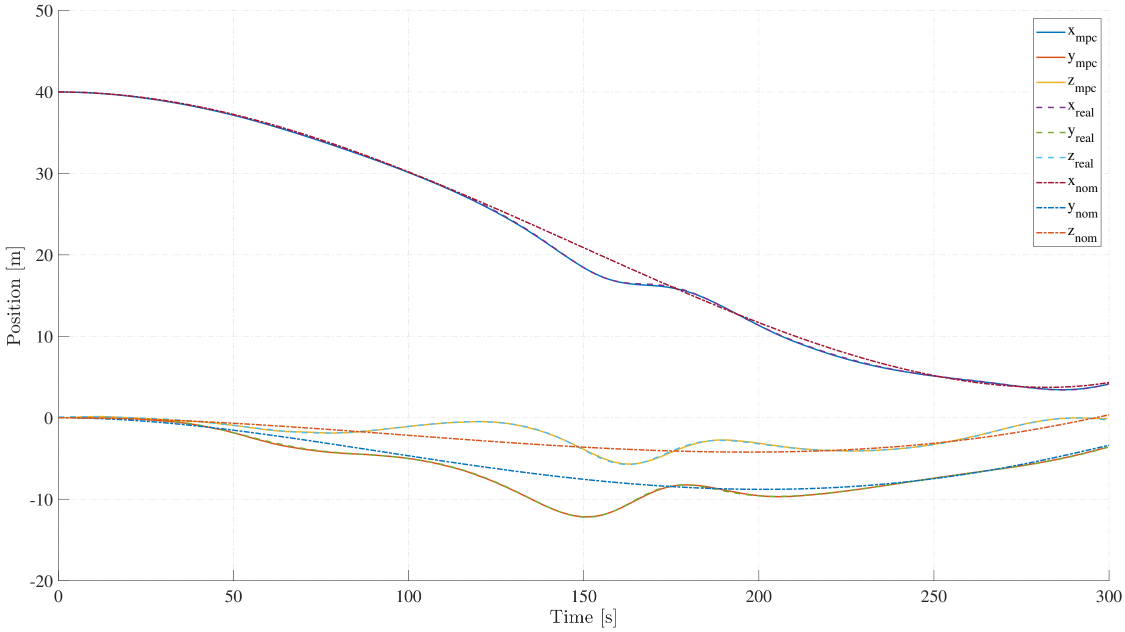


Figure 7.7: Tracking performance on position, LVLH frame

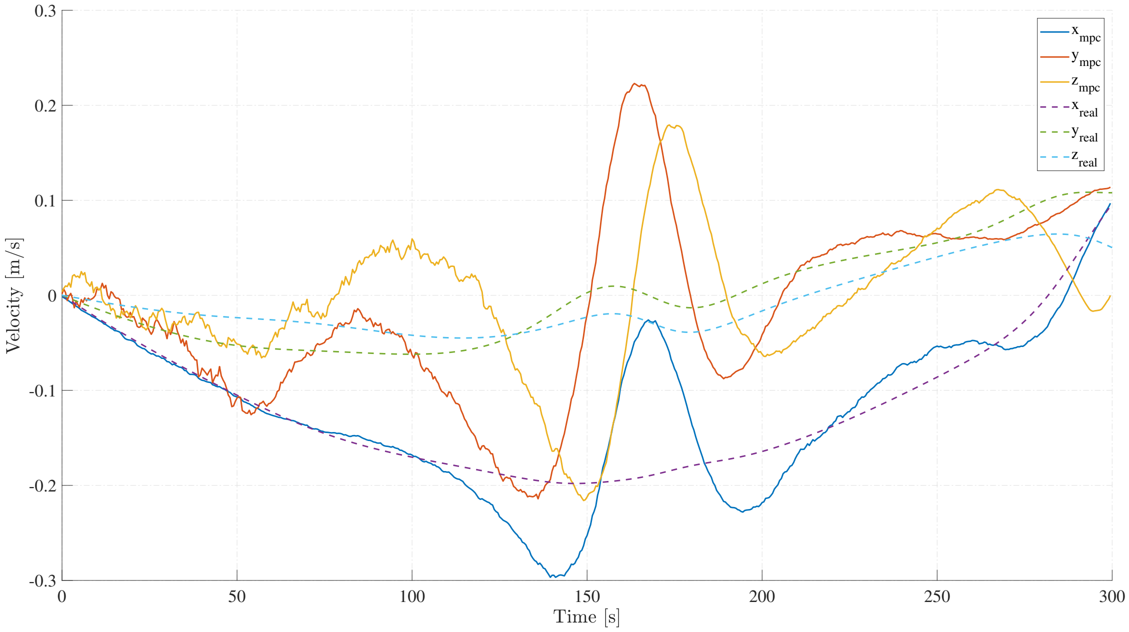


Figure 7.8: Tracking performance on velocity, LVLH frame

The good performance of the controller on the position is also shown in fig. 7.9, where it can be seen that the deviation from the real trajectory is always inferior to 25 cm, error curves for other values of N are also reported, as comparison data. The image shows another deficiency of the controller, as the reference trajectory to track comes to an end it is necessary to reduce the horizon as there is no more additional data available. This leads to a rapid degradation of the capability of the controller. Thus in a real mission scenario, there must be a switch in the control law that changes the objective from the tracking problem to maintaining the chaser at the right relative position as the target continues its tumbling.

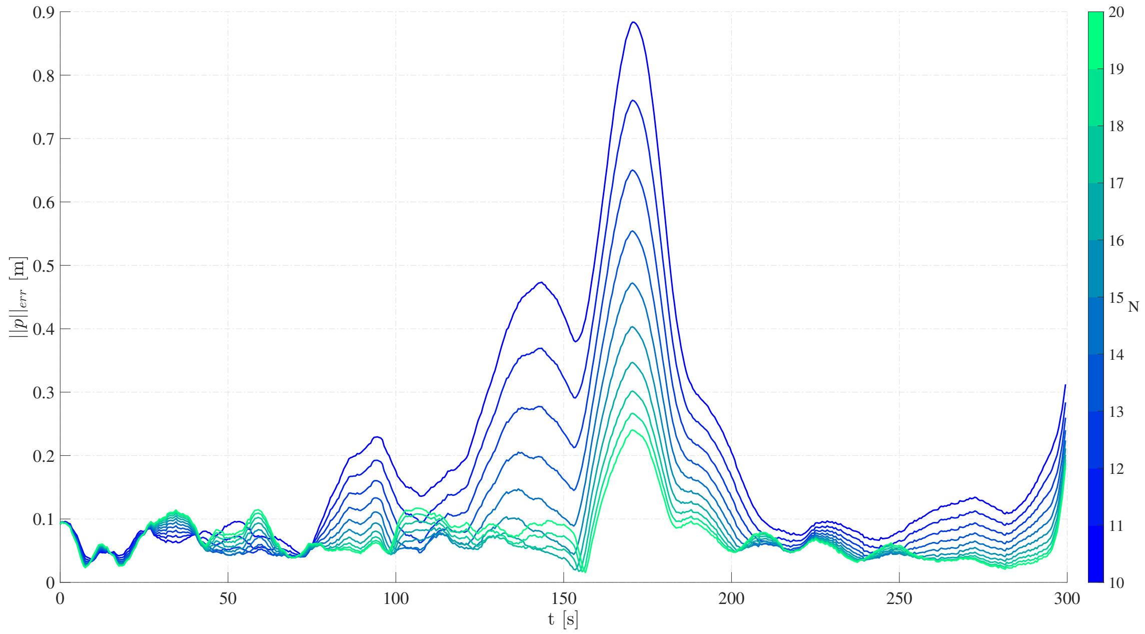


Figure 7.9: Error between real and effective solution, for various horizons N

7.2. Verification of robustness

The controller's ability to track the real trajectory in an inertial frame despite the disturbance coming from the trajectory updates was verified on a single case in the previous section. Since the robustness of the controller means that the tracking must be successful against any possible realization of the disturbances a Monte Carlo simulation is run to show this capability. The disturbances come from the uncertainty of the measurements and as such the simulation consists of a repeated use of the control loop against a simulator initialized every time with a different random seed. The simulation is run for 100 trials, the results are shown in fig. 7.10 and in fig. 7.11. As expected no matter the disturbance the controller maintains a tracking performance similar to the one obtained in the demonstrative example.

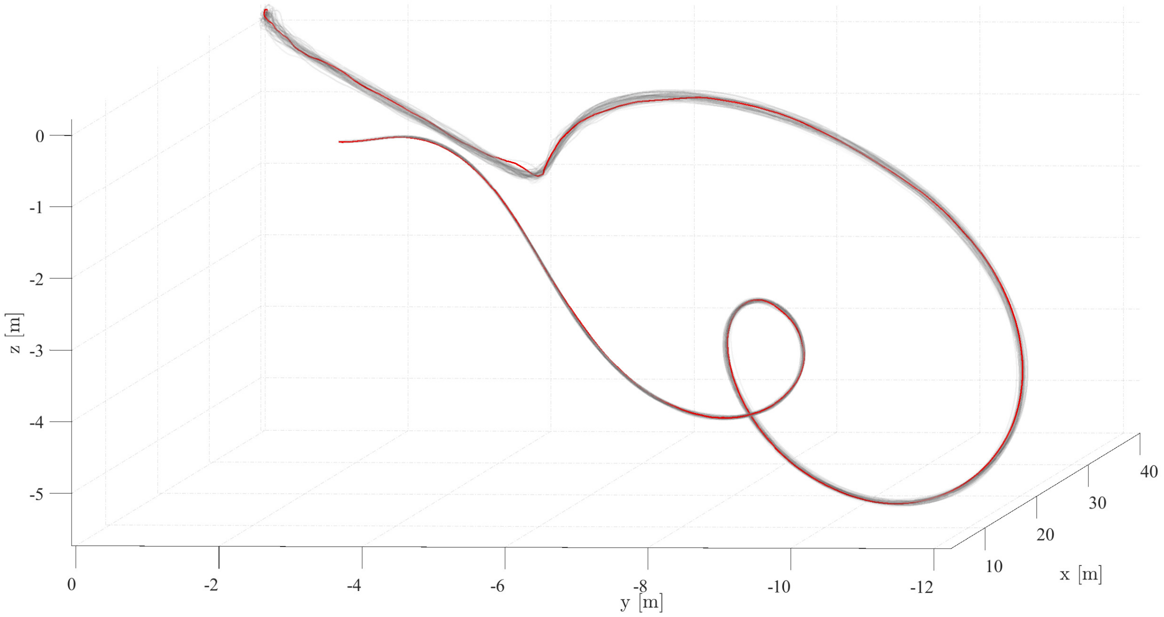


Figure 7.10: Robustness verification on position, LVLH frame

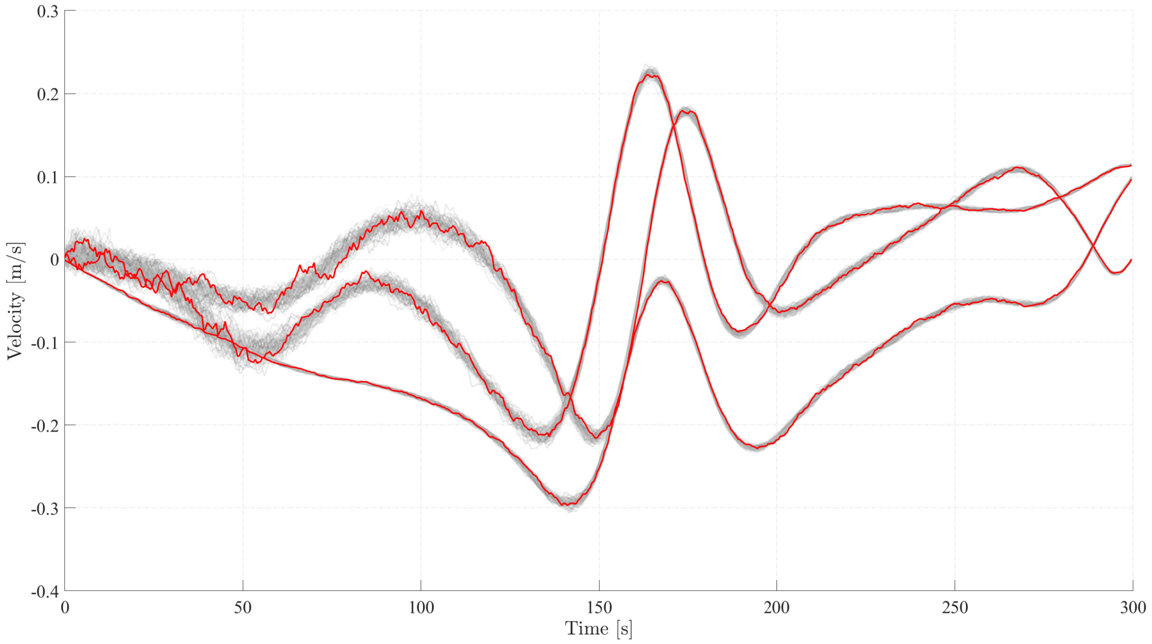


Figure 7.11: Robustness verification on velocity, LVLH frame

8 | Conclusions and future developments

In this thesis the guidance and control pipeline presented in Albee et al.[2] was applied to a new simulation setting to verify that the method is capable of dealing with an environment more similar to a real rendezvous mission. In particular, the rendezvous with Envisat was taken as a mission scenario. This required to use of the Clohessy-Wiltshire instead of the double integrator dynamics. An additional difference from the original work was to consider the inertia of the target spacecraft as uncertain, and as such it becomes another factor that contributes to the disturbance acting on the system.

The results have shown that the method is successful and maintains its robustness properties for the Envisat rendezvous scenario. Although the method has been proven to work even in this setting, some additional considerations have arisen for real-time applicability. The high and non-spherical inertia of Envisat has shown that even for moderate uncertainties in the initial target state and in the inertia itself, the disturbance acting on the system can be considerable. This could be problematic for a real mission since the only way to lower the disturbance bounds is to increase the frequency at which the robust invariants are computed, and the computational burden.

The next steps necessary before application to a real mission scenario are:

1. The uncertainties assumed in this work must be confronted with the ones achievable by state-of-the-art estimation units, to prove that the assumptions made in this thesis are compatible with uncertainty levels attainable in practice.

2. The degradation of the controller performance at the end of the reference trajectory must be investigated in more detail. A potentially viable solution could be to transition to an alternate controller when close enough to the rendezvous point.

Bibliography

[1] Orbital debris, quarterly news. 26, 2022.

[2] K. Albee, C. Oestreich, C. Specht, A. Terán Espinoza, J. Todd, I. Hokaj, R. Lampariello, and R. Linares. A robust observation, planning, and control pipeline for autonomous rendezvous with tumbling targets. *Frontiers in Robotics and AI*, 8: 641338, 2021. doi: 10.3389/frobt.2021.641338.

[3] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019. doi: 10.1007/s12532-018-0139-4.

[4] B. Bastida Virgili, S. Lemmens, and H. Krag. Investigation on envisat attitude motion. ESA/ESOC Space Debris Office, e.Deorbit Workshop, 5 2014.

[5] G. Boyarko, O. Yakimenko, and M. Romano. Optimal rendezvous trajectories of a controlled spacecraft and a tumbling object. *Journal of Guidance, Control, and Dynamics*, 34(4), 2011.

[6] C. Buckner and R. Lampariello. Tube-based model predictive control for the approach maneuver of a spacecraft to a free-tumbling target satellite. In *2018 Annual American Control Conference (ACC)*, Wisconsin Center, Milwaukee, USA, June 27–29 2018.

[7] M. Corso. Modelling, dynamic analysis and robotic control strategies for the deorbiting operations of the esa satellite envisat. Master's thesis, Master in Space & Astronautical Engineering, 5 2016.

[8] D. W. Ebdon. Model predictive control of aerospace systems. Master's thesis, Air Education and Training Command, December 1996.

[9] K. Fukuda. Lecture: Polyhedral computation. Department of Mathematics, and Institute of Theoretical Computer Science, ETH Zurich, Switzerland, 2 2015.

[10] F. Ghili. Pre and post-grasping robot motion planning to capture and stabilize a tumbling/drifting free-floater with uncertain dynamics. In *IEEE International Con-*

ference on Robotics and Automation (ICRA), pages 5441–5448, Karlsruhe, Germany, 2013.

[11] M. Herceg, M. Kvasnica, C. Jones, and M. Morari. Multi-Parametric Toolbox 3.0. In *Proc. of the European Control Conference*, pages 502–510, Zurich, Switzerland, July 17–19 2013. `http://control.ee.ethz.ch/~mpt`.

[12] C. M. Jewison. *Guidance and Control for Multi-stage Rendezvous and Docking Operations in the Presence of Uncertainty.* PhD thesis, Massachusetts Institute of Technology, 6 2017.

[13] S. C. John A. Christian. A survey of lidar technology and its use in spacecraft relative navigation. 8 2013.

[14] M. Kelly. An introduction to trajectory optimization: How to do your own direct collocation. *SIAM Review*, 59(4):849–904, 2017.

[15] D. J. Kessler. Collisional cascading: The limits of population growth in low earth orbit. *Advances in Space Research*, 11(12):63–66, 1991.

[16] I. Kolmanovsky and E. G. Gilbert. Theory and computation of disturbance invariant sets for discrete-time linear systems. *Mathematical Problems in Engineering*, 4:317–367, 1998.

[17] A.-K. Kopetzki, B. Schurmann, and M. Althoff. Methods for order reduction of zonotopes. pages 5626–5633, 12 2017. doi: 10.1109/CDC.2017.8264508.

[18] B. Kouvaritakis and M. Cannon. *Model Predictive Control: Classical, Robust and Stochastic.* Advanced Textbooks in Control and Signal Processing. Springer International Publishing, 2015. ISBN 9783319248530.

[19] W. Ley. *How to Slay Dragons.* August 1960.

[20] D. Limon, I. Alvarado, T. Alamo, and E. F. Camacho. On the design of robust tube-based mpc for tracking. In *Proceedings of the 17th World Congress The International Federation of Automatic Control*, Seoul, Korea, July 6–11 2008.

[21] M. Oda. Ets-vii: Achievements, troubles and future. In *Proceedings of the 6th International Symposium on Artificial Intelligence and Robotics & Automation in Space: i-SAIRAS 2001*, Canadian Space Agency, St-Hubert, Quebec, Canada, June 18–22 2001.

[22] S. V. Rakovic, E. C. Kerrigan, K. I. Kouramas, and D. Q. Mayne. Invariant ap-

proximations of the minimal robust positively invariant set. *IEEE Transactions on Automatic Control*, 50(3), March 2005.

[23] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl. *Model Predictive Control: Theory, Computation, and Design.* Nob Hill Publishing, LLC, Santa Barbara, CA, USA; London, England; Freiburg, Germany, 2nd edition, 2022.

[24] G. Rekleitis and E. Papadopoulos. On controller parametric sensitivity of passive object handling in space by robotic servicers. In *International Conference on Intelligent Robots and Systems*, pages 14–18, Chicago, IL, USA, 2014.

[25] R. Schneider. *Minkowski addition*, pages 139–207. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2013. doi: 10.1017/CBO9781139003858.006.

[26] A. Seddaoui and C. M. Saaj. Combined nonlinear h-inf controller for a controlled-floating space robot. *Journal of Guidance, Control, and Dynamics*, 42(8):1878–1885, 2019. doi: 10.2514/1.g003811.

[27] J. Slotine and W. Li. *Applied Nonlinear Control.* Prentice-Hall, 1991.

[28] M. Sánchez Nogales. Elecnor deimos participation in e.deorbit and adr activities. In *e.Deorbit Symposium*, Leeuwenhorst, Netherlands, 2014.

[29] P. Trodden. A one-step approach to computing a polytopic robust positively invariant set. *IEEE Transactions on Automatic Control*, 61(12):4100–4105, 2016.

[30] S. Ulrich, A. Saenz-Otero, and I. Barkana. Passivity-based adaptive control of robotic spacecraft for proximity operations under uncertainties. *Journal of Guidance, Control, and Dynamics*, 39(6):1444–1453, 2016. doi: 10.2514/1.g001491.

[31] C. o. t. U. N. United Nations. Technical report space debris, 1999.

[32] Q. Zhan and V. G. Grassi. An industrial approach to implementing mpc in speciality chemical plant. *IFAC Proceedings Volumes*, 32(2):6603–6608, 1999. ISSN 1474-6670. doi: https://doi.org/10.1016/S1474-6670(17)57128-1. URL `https://www.sciencedirect.com/science/article/pii/S1474667017571281`. 14th IFAC World Congress 1999, Beijing, Chia, 5-9 July.

[33] L. Zhang and J.-c. Xu. Optimal delaunay triangulations. *Journal of Computational Mathematics*, 22(2):299–308, 2004.

[34] W. Zhang, F. Li, J. Li, and Q. Cheng. Review of on-orbit robotic arm active debris

capture removal methods. *Aerospace*, 10(13), 2023. URL `https://doi.org/10.3390/aerospace10010013`.

A | Computation of the mRPI set

In this appendix, the implementation for the mRPI computation is verified against an example problem solved for the first time in [22]. Additionally, the performance increase of the zonotopic approach is commented on.

Given the system in eq. (A.1), the additive disturbance $\mathbb{W} = \{w \in \mathbb{R}^2 \mid ||w||_\infty < 1\}$ and the control law $u = -\left[1.171.03\right]x$, the mRPI can be calculated with algorithm 5.1. The results are equivalent to the ones found in the paper as shown in fig. A.1. In the image are also shown the iterative approximations of the mRPI for various s and the final set.

$$x = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u + w \tag{A.1}$$

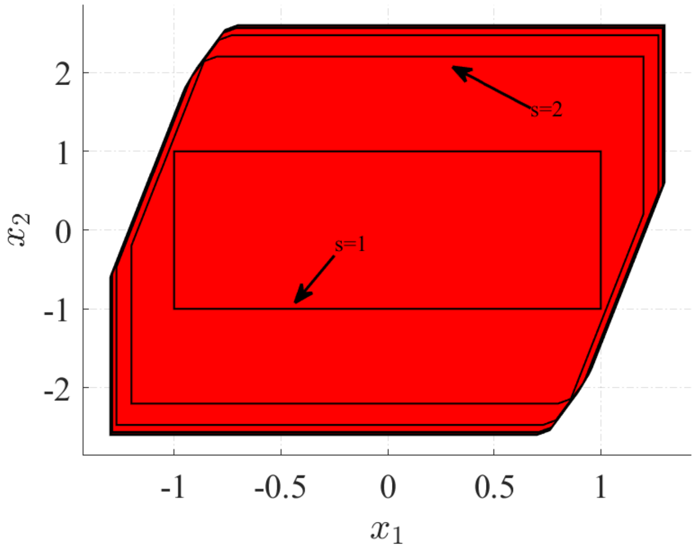


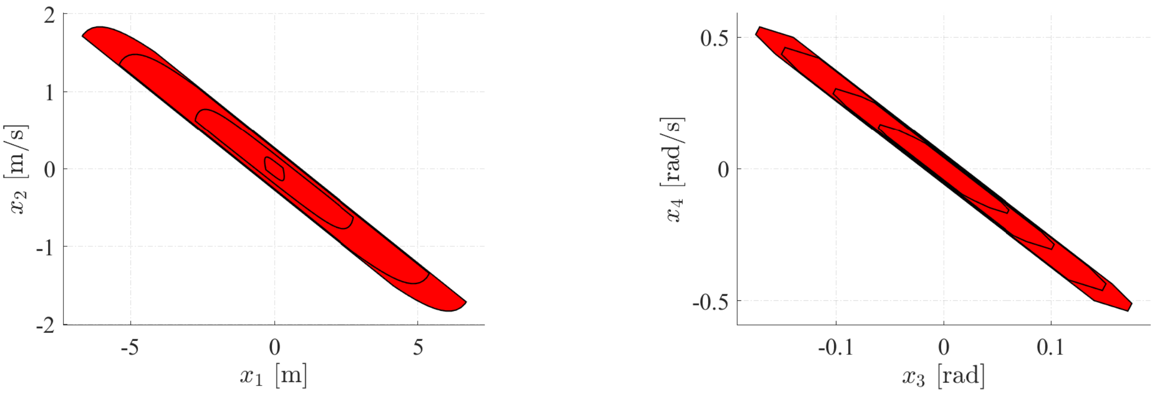
Figure A.1: Results mRPI set

The algorithm based on zonotopes gives an identical mRPI, but requires only 0.0161 seconds, while the traditional one requires 0.288 seconds.

B | Computation of the MRPI set

In this appendix, the implementation for the MRPI computation is verified against an example problem solved for the first time in [16]. The dynamical system consists of a cart with an inverted pendulum, as described by eq. (B.1). Further details can be found in the paper. As shown in fig. B.1 the results obtained match the invariant sets obtained in the paper for various levels of uncertainty.

$$\frac{d}{dt}\begin{pmatrix}x_1\\x_2\\x_3\\x_4\end{pmatrix} = \begin{pmatrix}0 & 1 & 0 & 0\\0 & 0 & -\frac{mg}{M} & 0\\0 & 0 & 0 & 1\\0 & 0 & \frac{(M+m)g}{Ml} & 0\end{pmatrix}\begin{pmatrix}x_1\\x_2\\x_3\\x_4\end{pmatrix} + \begin{pmatrix}0\\\dfrac{1}{M}\\0\\-\dfrac{1}{Ml}\end{pmatrix}u \tag{B.1}$$



(a) Slice at x_3=0 and x_4=0

(b) Slice at x_1=0 and x_2=0

Figure B.1: Results MRPI set

List of Figures

List of Tables

List of Symbols

Variable	Description	SI unit
$\mathbf{A}$	State matrix	[-]
$\mathbf{B}$	Input matrix	[-]
$\mathbf{x}$	State	[-]
$\mathbf{u}$	Input	[-]
μ	Earth gravitational parameter	$[km^3/s^2]$
m	Mass	[kg]
$\boldsymbol{\omega}$	Angular velocity	[rad/s]
$\mathbf{J}$	Inertia matrix	[kg m^2]
$\boldsymbol{\tau}$	Torque	[Nm]
Ω	Orbital angular velocity	[rad/s]
$\mathbf{q}$	Quaternion	[-]
${}^x\mathbf{R}^y$	Rotation matrix	[-]
$\mathbf{p}$	Position vector	[m]
$\mathbf{v}$	Velocity vector	[m/s]
$\mathbf{w}$	Disturbance vector	[-]
Σ	Covariance matrix	[-]
$\mathbb{W}$	Disturbance set	[-]
$\mathbb{F}_\infty$	Minimum RPI set	[-]
$\mathbb{O}_\infty$	Maximum RPI set	[-]
$\mathbf{K}_{dr}$	Disturbance rejection gain	[-]
$\mathbf{C}$	Output matrix	[-]
$\mathbf{D}$	Feed-forward matrix	[-]
Y	Output constraint set	[-]
$\mathbb{X}$	State constraint set	[-]
$\mathbb{U}$	Action constraint set	[-]
$\mathbf{Q}$	LQR State weight matrix	[-]
$\mathbf{R}$	LQR Action weight matrix	[-]
$\mathbf{P}$	LQR End State weight matrix	[-]
a	Orbital semi-major axis	[km]
e	Orbital eccentricity	[-]

Acronyms

GNC	Guidance, Navigation, and Control
MPC	Model Predictive Control
NORAD	North American Aerospace Defense Command
GEO	Geosynchronous Earth Orbit
LIDAR	Light Detection and Ranging
RADAR	Radio Detection And Ranging
ADR	Active Debris Removal
DARPA	Defense Advanced Research Projects Agency
EK	Extended Kalman
SMC	Sliding Mode Control
PD	Proportional Derivative
SLAM	Simultaneous Localization and Mapping
IMU	Inertial Measurement Unit
LVLH	Local Vertical Local Horizontal
CoM	Center of Mass
ZOH	Zero-Order Hold
TPBVP	Two Point Boundary Value Problem
CW	Clohessy-Wiltshire
RPI	Robust Positive Invariant
mRPI	Minimum RPI
MRPI	Maximum RPI
LQR	Linear Quadratic Regulator
ESA	European Space Agency
CAD	Computer Aided Design

Acknowledgements

I would like to express my deepest appreciation to all those who provided me with the possibility to complete this thesis. A special gratitude I give to my thesis advisor Prof. Mauro Massari, whose contribution in stimulating suggestions and insights, helped me to coordinate my project and enrich my perspective on the research topic.

Furthermore, I would also like to acknowledge with much appreciation the crucial role of my family, friends and colleagues, who gave the moral support and the motivation to stay focused on my studies.