Augusto José, Tam Tapia

Space Craft Reliable Trajectory Tracking and Landing using Model Predictive Control with Chance Constraints

Master Thesis in Mechatronics

02 May 2017

Please cite as:
Augusto José, Tam Tapia, "Space Craft Reliable Trajectory Tracking and Landing using Model Predictive Control with Chance Constraints", Master Thesis (Masterarbeit), Technische Universität Ilmenau, Computer Science and Automation Faculty, May 2017.

Pontifical Catholic University of Peru
Graduate School
MSc in Mechatronics Engineering

Av. Universitaria 1801-San Miguel · 15088 Lima · Perú

Technische Universität Ilmenau
Computer Science and Automation Faculty
Institute for Automation and Systems Engineering
Simulation and Optimal Processes Group

Helmholtzplatz 5 · 98693 Ilmenau · Germany

Pontificia Universidad Católica del Perú

Escuela de Posgrado

Space Craft Reliable Trajectory Tracking and Landing using Model Predictive Control with Chance Constraints

Master Thesis in Mechatronics

submitted by

Augusto José, Tam Tapia

born on 05. March 1990
in Trujillo-Perú

in the

Simulation and Optimal Processes Group

Computer Science and Automation Faculty
Technische Universität Ilmenau

Supervisors: **Dr.rer.nat. Abebe Geletu W.Selassie (TUI)**
PhD.Ing. Julio C. Tafur (PUCP)

Submission Date: **02 May 2017**

Declaration

I declare that the work is entirely my own and was produced with no assistance from third parties.
I certify that the work has not been submitted in the same or any similar form for assessment to any other examining body and all references, direct and indirect, are indicated as such and have been cited accordingly.

(Augusto José, Tam Tapia)
Ilmenau, 02 May 2017

Abstract

This work considers the study of chance constrained Model Predictive Control (MPC) for reliable spacecraft trajectory tracking and landing.
Objectives of the master thesis:

- To identify and study mathematical dynamic models of a spacecraft.
- To study the trajectory design and landing schemes for a given mission.
- To study the source of uncertainty in the model parameters and external disturbances.
- To study the chance constrained MPC scheme for the reliable and optimal trajectory tracking and landing.
- To testing the new analytic approximation approaches, Inner and Outer, for chance constraints.
- To study appropriate MPC algorithms and implement on case-studies.

In the first part of the thesis considers deterministic dynamical models of spacecraft are discussed.
The first example is about the tracking of trajectory and soft landing on the surface of an asteroid EROS433, this model uses Cartesian coordinates.
In the second example, in a similar way to the first example, the trajectory and soft landing is performed on the surface of a celestial body. It is assumed that the celestial body is a perfect sphere, something that does not happen in the first example. Thus, the second example uses a Spherical coordinate system.
The third example is about a Lander that enters the Martian atmosphere. This Lander follows a designed trajectory until reaching a certain altitude over the Martian surface. At this altitude the Lander deploys a parachute to make the landing.
To solve the deterministic examples described above, the following sequence of steps are:

- pose the deterministic Nonlinear Optimal Control Problem (NOCP),
- convert the infinite Optimal Control Problem (OCP) to a finite Nonlinear Programming Problem (NLP), applying the Runge-Kutta 4th order discretization method,
- apply the Quasi-sequential method to the deterministic NLP obtained from the previous step,
- solution of the reduced NLP obtained from the previous step using IpOpt software.

The steps outlined above are also part of the Nonlinear Model Predictive Control (NMPC) approach.
In the second part of the thesis, the same examples of the first part are used but now with stochastic variables. To find the control law in each model, the stochastic NMPC was used. The above mentioned approach begins with a chance constrained OCP. The latter is discretized obtaining an NLP. The problem with this NLP, with chance constraints, is that is very difficult to solve in analytic form. So these chance constraints are approached by a different method that exist in the state of the art. This thesis work is focused on approaching the chance constraints through Analytic Approximation Strategies, specifically by the recent: Inner and Outer Approximation methods.
The chance constrained MPC is expensive from a computational point of view, but it allows to find a control law for a more reliable trajectory-tracking and soft landing . That is suitable for applications with random disturbances, model inaccuracies, and measurement errors.

Kurzfassung

Diese Arbeit betrachtet das Studium der wahrscheinlichkeitsbeeschränkten modellprädiktiven Regelung (MPC) für zuverlässige Raumfahrzeug-Trajektorienverfolgung und -Landung.
Die Ziele der Masterarbeit sind,

- mathematische, dynamische Modelle eines Raumfahrzeugs zu identifizieren und zu studieren,
- der Trajektorienverfolgung und Landungspläne für eine bestimmte Mission zu untersuchen,
- die Unsicherheitsquellen bei den Modellparametern und externen Störungen zu untersuchen,
- das wahrscheinlichkeitsbeschränkte MPC-Schema für die zuverlässige und optimaleTrajektorienverfolgung und Landung zu studieren,
- die neuen, analytischen inneren und äußeren Approximationsansätze für Wahrscheinlichkeitsnebenbedingungen zu testen und
- geeignete MPC-Algorithmen zu untersuchen und auf Fallstudien umzusetzen. Im ersten Teil der Arbeit werden deterministische, dynamische Modelle von Raumfahrzeugen diskutiert.

Das erste Modell beschreibt die Verfolgung der Trajektorie und die weiche Landung auf der Oberfläche des Asteroiden EROS433 und verwendet kartesische Koordinaten.
Im zweiten Beispiel wird in ähnlicher Weise wie beim ersten die Trajektorie und die weiche Landung auf der Oberfläche eines Himmelskörpers durchgeführt. Es wird davon ausgegangen, dass der Himmelskörper eine perfekte sphärische Form hat, was im ersten Beispiel nicht gegeben ist. Das zweite Beispiel verwendet ein sphärisches Koordinatensystem.
Das dritte Beispiel handelt von einer Landungssonde, die in die Marsatmosphäre eintritt. Diese Sonde folgt einer entworfenen Bahn, bis sie eine gewisse Höhe über der Marsoberfläche erreicht hat. Auf dieser Höhe entfaltet die Sonde einen Fallschirm, um die Landung durchzuführen.
Um die oben beschriebenen deterministischen Beispiele zu lösen, sind die folgenden Schritte nötig:

- Aufstellen des deterministischen, nichtlinearen optimalen Steuerungsproblem (NOCP) Umwandlung des unendlich-dimensionalen, optimalen Steuerungsproblems (OCP) zu einem endlich-dimensionalen, nichtlinearen Programmierungsproblem (NLP), wobei die Runge-Kutta- Methode 4. Ordnung als Diskretisierungsmethode angewendet wird

- Anwendung des Quasi-Sequentiellen Ansatzes auf das deterministische NLP , das aus dem vorherigen Schritt gewonnen wurde,

- Lösen des reduzierten NLP aus dem vorherigen Schritt mit Ipopt-Software, Die oben beschriebenen Schritte sind Teil des "Nonlinear Model Predictive Control" (NMPC)-Ansatzes.

Im zweiten Teil der Arbeit werden die gleichen Modelle des ersten Teils, aber jetzt mit stochastischen Variablen angewendet. Um das Steuergesetz in jedem Modell zu finden, wurde die stochastische NMPC verwendet. Der oben genannte Ansatz beginnt mit einem wahrscheinlichkeitsbeschränkten OCP. Letzteres ist diskretisiert worden, um ein NLP zu erhalten. Das wahrscheinlichkeitsbeschränkte NLP ist sehr schwierig in analytischer Form zu lösen. So werden diese Wahrscheinlichkeitsbeschränkungen durch eine andere Methode behandelt, , die dem Stand der Technik entsprechen. Diese Arbeit beschäftigt sich mit der Annäherung an die Wahrscheinlichkeitsbeschränkungen durch analytische Approximationsstrategien, insbesondere durch die jüngsten Methoden der inneren und äußeren Approximation.
Die wahrscheinlichkeitsbeschränkte MPC ist aus rechnerischen Sicht aufwändig, aber sie erlaubt, ein Steuergesetz für eine zuverlässigere Trajektorienverfolgung und eine weiche Landung zu finden. Das eignet sich für Anwendungen mit zufälligen Störungen, Modell-Ungenauigkeiten und Messfehlern.

Acknowledgements

First, I would like to thank God for allowing me to reach this point, for giving me life to get my goals and put the right people in my way. And my parents, for your unconditional support and continuous motivation that I should be a good person and for your unconditional love.
And my appreciation also extend to my supervisor Dr.rer.nat Abebe Geletu W. Selassie, for his continuous guide and valuable support, and his early insights launched the greater part of this dissertation.
Finally, my deeply thanks to CONCYTEC for the financial support that has made this experience in my life possible, and contribute to my professional development.

Contents

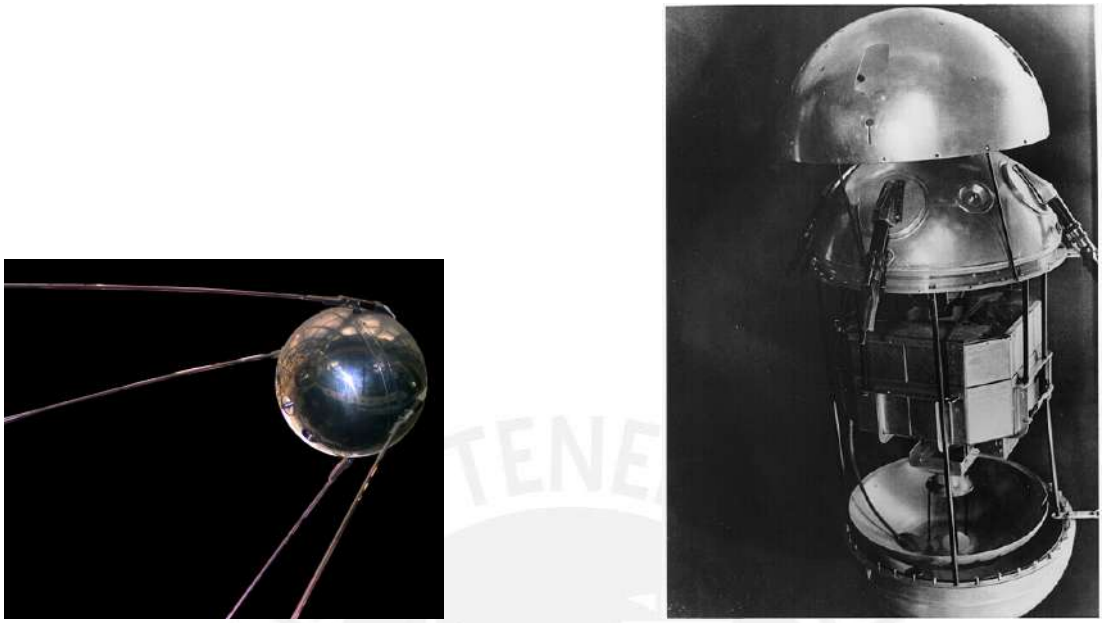
Chapter 1

Introduction

Space exploration is the continuous discovery and exploration in the deep space of celestial bodies, this requires the continuous development and improvement of the technology to fulfill the exploration purposes.
Today astronomers study deep space using powerful telescope systems located on our planet, the physical exploration of deep space is carried out by unmanned robotic space probes, but also use manned spaceships to explorations not so distant to our planet.

Many years ago the scientist Galileo Galilei studied the surface of the Moon, the cluster of stars of the Pleiades to cite some of his astronomical studies, making use of a simple telescope, of limited scope, and collecting data of what he observed (e.g. The movement of the celestial bodies).
It was not until the 20th century, when the first long-range missiles were developed that for the first time in the history of mankind allowed the physical exploration of outer space. The importance of exploring space is varied, we will state some reasons.
Advances in scientific research are achieved, the teamwork of several nations allows technological developments such as the International Space Station (ISS) (see Figure 1.3), ensure the future survival of humanity.
Finally, the least noble of the reasons is the development of military power and military strategies.

The Space Age (understood as a period of mankind) in its beginnings was ruled by the race to reach space, a kind of competition, rivalry between the United States of America and the USSR (today Russia), because of the "Cold War".
The launch of the first man-made unmanned spacecraft whose purpose was to orbit the earth was first achieved by the USSR. That spacecraft was called Sputnik 1 (see Figures 1.1a and 1.1b).
Then it happened that mankind got to send a man to the moon and bring him back to earth safe and sound, inside a spaceship called Apollo 11 (see Figure 1.2) owned by the United States of America.
These two events mentioned above are considered milestones in the space age.



(a) Assembled Sputnik 1 **(b)** Explode view of Sputnik 1

Figure 1.1 – Sputnik 1 [1]

After 20 years since the beginning of the physical exploration of outer space, there were very significant changes such as:

1. now the development of reusable spaceships that could be used in several missions was looked for, and not only in a unique mission like before it happened,
2. the strong spirit of competition between nations was changed by a cooperation between them, joining efforts was achieved to build the ISS(see Figure 1.3).

In 2000 the People's Republic of China initiated a successful space program for a manned spacecraft. Japan, China, India and Russia have opted for manned space missions to the moon in the present century, on the other hand the European community has opted for manned missions to both the Moon and Mars.

We mentioned above the importance of space exploration, we will give more reasons of the importance of the same in more detail [2]:

1. The need to explore outer space in search of answers to the following questions: How old is our Solar System? Know about the history of our Solar System. The collateral effects of this space exploration that seeks to answer these questions have been the immense development, expansion of technology, creation of new industries.

2. Space probes whose mission is to visit asteroids near our planet, try to answer the questions that humanity is asking: Are there more forms of life in outer space?

Figure 1.2 – Apollo 11 [1]

Also the probes collect data of scientific interest about the gravitational field of these asteroids, to achieve that a space probe orbits during a period of time around the asteroid, sends all the collected data to the Earth where it is processed and a three-dimensional image is obtained of the gravitational field. The probes can also land on asteroids, as the previous step is done with respect to the gravitational field, the landing is planned, if this is done successfully then it can collect more data of the asteroid related to its chemical composition, and even return with samples to the Earth. It is also important to mention that missions with space probes is not only important because of the data collected, but also because it allows to acquire more experience, training to in the future to achieve greater challenges such as sending a man to Mars.

3. Space missions to the translunar space (a space beyond the orbit of the Moon, where Earth and Moon gravitation predominate) allow to study the cosmic radiations of the Galaxy, the latter are very harmful to the human health, the data collection is important to develop mitigation techniques, and thus be able to travel in deep space in the future without fear of the harmful effects of these radiations. Another advantage of translunar missions is that it provides the opportunity to develop tools, techniques for future space explorations, without the need to move far from the Earth.

In order to carry out the aforementioned space missions, it is necessary to develop robust controllers that can effectively follow a desired path, to land in a celestial body of interest (e.g. The moon) and also to handle uncertainties present in outer space as for example:

- The pressure exerted by the radiation, solar winds, the drag of outer space, space junk.
- The dynamics not modeled in the model of the real plant, is sometimes treated as a stochastic disturbance.
- Inaccurately modelled gravitational fields surrounding celestial bodies.

- Sensor errors(e.g. radar, accelerometers, etc) leading to inaccuracies in navigation and landing.
- Misalignment of the propellers.

The robust controllers adequate to handle these uncertainties and to fulfill the objectives of the missions are the "chance constrained Stochastic Model Predictive Control (SMPC) controllers", that will be developed in this thesis for some space missions in later chapters.

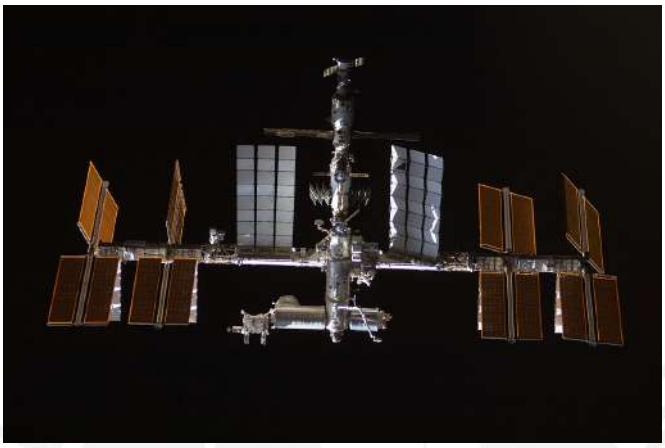


Figure 1.3 – ISS [1]

1.1 Motivation

Spacecraft trajectory tracking and soft landing requires an efficient control strategy that provides robust performance and fault-tolerant navigation in the face of uncertainties arising from aerodynamics, atmospheric density, gravitational field modeling, plant model mismatch, etc. For such mission-critical applications, efficient trajectory tracking and precision landing are central issues. The Stochastic Model Predictive Control provides the most convenient scheme for the control of a Spacecraft on moving horizons, where chance constraints are to be used to guarantee reliable trajectory tracking as well as safe, stable, accurate localized landing with high reliability.

1.2 Objectives of the Thesis

The objectives of the master thesis research work are:

- To identify and study mathematical dynamic models of a spacecraft.
- To study the trajectory design and landing schemes for a given mission.
- To study the source of uncertainty in the model parameters and external disturbances.
- To study the chance constrained MPC scheme for the reliable and optimal trajectory tracking and landing.

- To study appropriate MPC algorithms and implement on case-studies.

1.3 Structure of the thesis

The following will briefly explain each of the remaining chapters:

Chapter 2 sets out theoretical foundations for the operation of the MPC approach, the general formulation of an OCP is also presented, followed by an exposition of the different methods of the state of the art that allow the resolution of the OCP, of these methods, the most used today are the Direct Methods. The latter are subclassified in Sequential, Simultaneous and Quasi-Sequential Methods. Later the main methods of these subclassifications are outlined (Direct Single Shooting, Direct Multiple Shooting and Quasi-sequential method). Finally this chapter ends briefly explaining the main numerical methods for the solution of NLP: SQP and Interior Point(IP) Methods.

In Chapter 3, the chance constrained MPC is formulated.
Then the NLP Problem is presented which results from the discretization of the first one mentioned. The problem with this NLP is that it has chance constraints that are very difficult to evaluate, for that reason are exposed the different methods of the state of the art that serve to approximate these chance constraints, and thus obtain a Deterministic Optimization Problem.
This thesis-work applies recent analytic approximation approaches such as Inner and Outer Approximations, which are also explained in this chapter.

Chapter 4 presents the dynamic equations, the designed trajectories, chance constrained OCPs, chance constrained NLPs, parameters for the simulations, of the three study cases: Tracking trajectory and landing on the asteroid Eros433, Guidance law in three dimensions for a Soft landing on a celestial body, and Longitudinal model of lifting entry of a Mars Lander.

Chapter 6 presents the results of the simulations that are done to find the control laws in the deterministic case as well as the stochastic case, for the 3 study cases presented in Chapter 4.

Chapter 7 concludes with a summary of obtained results and by suggesting possible future work.

Chapter 2

State of the art

The development of this chapter will be started, exposing current works related to the tracking trajectory and landing of Spacecraft using control techniques based on MPC.

In [3] an MPC controller was developed for the orientation of a spacecraft, the spacecraft consists of actuators like flywheels that allow torque to be applied.
In addition, there are restrictions on the control variable, e.g., Torque.
The MPC controller takes advantage of the term of the objective function related to the reference system.
The addition of a low-complexity integral action to eliminate the offset in the tracking of the orientation set-points, and an online optimization algorithm for the solution of a quadratic programming problem set to fixed-point arithmetic.
The simulation made with a nonlinear model for the spacecraft shows that the MPC controller achieves a very good tracking of the set-points, besides that the control restrictions (the torque of the flywheels) are satisfied. The controller developed has low computational complexity, and is also suitable for implementation in spacecraft with fixed point processors. Comparing explicit MPC solutions to MPC online solutions, the latter require small Read-only memorys (ROMs), which is an advantage since ROMs may be limited in size on-board a spacecraft.
Furthermore, large ROMs may get damaged by outer space radiation.

In [4] a robust MPC controller was used to solve a problem of spacecraft rendezvous, in order to achieve the above, it was necessary to apply the Clohessy-Wiltshire-Hill (CWH) [5, 6] model with perturbations and Line of Sight (LOS) constraints.
A robust MPC controller was designed using a chance constrained approach for the satisfaction of probabilistic constraints.
In a space vehicle navigation there are multiple sources of disturbances such as error in the measurement of position or speed, thruster misalignments, atmospheric drag. Hence the need to design robust control schemes to deal with these disturbances.
The perturbations are modeled as Gaussian distributions. This may allow to convert the probabilistic restrictions into simple algebraic constraints for linear plant models.
In the end, the robust MPC controller is compared with the non-robust MPC controller using the Monte Carlo method. Finally, the results demonstrate the superiority of the

robust MPC controller. In [4], an online estimator was used to find the statistical properties of the perturbations.

In [7], the author has developed in previous works an algorithm of path planning for spacecraft rendezvous that calculates the optimum control signal by Pulse Width Modulation (PWM).
A realistic model was used in relation to the spacecraft propellers. Commonly the propellers actuators are ON/OFF type.
These impellers can only toggle their state of "produce maximum force" (ON) to "no force" (OFF) and vice versa. We can only control these ON-OFF changes of the impellers by generating pulse trains as shown in Figure 2.1.

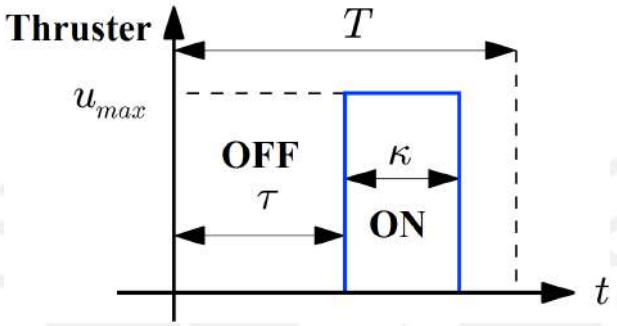


Figure 2.1 – Thruster ON-OFF change, modified from [7].

Planning the trajectory for spacecraft rendezvous with the actuators generating pulse trains, poses a great challenge because the system is nonlinear when the impellers alternate their state (ON-OFF).

The work done in [8,9], proposes a path-planning algorithm for spacecraft rendezvous that can handle PWM control signals. In [8] was used the Clohessy-Wiltshire model [6]. In [9] the time-varying linear model of Tschaune and Hempel [10] was used.
However, the above methods based on path planning cannot handle orbit perturbations, uncertainties or errors in the model.
In order to overcome these difficulties, an MPC algorithm based on the open-loop PWM planer [9], was developed, and this algorithm was tested for elliptical trajectories of arbitrary eccentricity [10]. The operation of the proposed algorithm is as follows. The MPC is initialized by solving the open loop problem with the path planning PWM algorithm. Then at each subsequent step, the MPC saves time by recomputing the path by applying an iterative scheme of linearization of the path planning algorithm.

In [11] an approach is developed for the control of the relative movement of a spacecraft based on the application of Linear Quadratic MPC with reconfigurable dynamic constraints. The MPC controller was designed to produce Δv impulsive velocity changes rather than a constant piecewise thrust profile. The restrictions on the positioning control system of the spacecraft, in addition to the restrictions concerning the directions (orientation), are taken into account.

The MPC controller is validated by making use of a spacecraft motion model that is highly nonlinear.
It is also augmented with an Extended Kalman Filter (EKF) which is used to estimate spacecraft states. It proved:

1. The robustness of the proposed approach with respect to unmeasured disturbances. This is achieved through corrections made through feedback with MPC.
2. Feedback MPC can only be implemented based on measures of relative angles as well as relative ranges.
3. Capacity to switch between MPC guidance in the spacecraft rendezvous phase and MPC guidance in the spacecraft docking phase, with the specific requirements, restrictions and sample rates for each phase [11].

In [12] an MPC approach is used to calculate the optimal control strategy for a spacecraft to be able to mate (rendezvous) to a spacecraft that orbits our planet. The life of spacecraft is limited by the fuel they carry on-board and by the rate of fuel consumption. If the rendezvous maneuvers are carried out optimally, the life of the missions with the spacecraft will be extended. Usually the planning of the two spacecraft's rendezvous path is carried out in open loop in the mission planning phase, after which a closed loop controller is used to follow the pre-planned path.
In addition, the MPC approach has to recalculate the optimal path every time obstacles occur (e.g. other spacecraft, garbage from outer space). This is done in real time.
Obstacles are approached by ellipsoids for:

1. simple evaluation of the restrictions and for
2. a better representation of the position of obstacles.

SQP was used to solve this quadratic optimization problem with nonlinear constraints to avoid obstacles. In situations where multiple obstacles are in motion, this MPC approach is adequate because the formulation of nonlinear constraints is flexible.

In the paper [13], an MPC system was implemented that is capable of guiding, controlling a tracking spacecraft during the rendezvous process with another target spacecraft that is orbiting in a circle or ellipse.
To achieve an efficient system design, the rendezvous maneuver is partitioned into three phases. In addition, to a maneuver to avoid collisions in case of failure. Each of the three phases has its own MPC controller.
Figure 2.2 shows the three phases as well as the maneuver in case of failure mentioned in the previous paragraph.
The first phase is called Orbit Synchronization Translational Guidance (OSTG), the second phase is called Impulsive Nominal Translational Orientation (INTG), the third phase is called Forced Terminal Translation Guide (FTTG), the maneuver in case of failure is called Collision Avoidance Maneuver (CAM). Finally the acronym TAP means "target approach point".
Linear time-varying systems were used that allowed for trajectory predictions in elliptic as well as circular orbits, and a norm in the objective function over cost in the change of velocity allowed to minimize the propulsion fuel consumption.

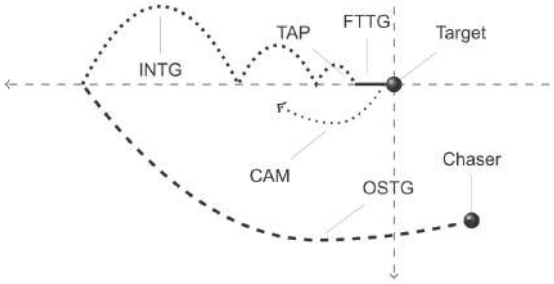


Figure 2.2 – Rendezvous phases [13].

A very important characteristic in the design proposed in this work was that nonconvex constraints can be shifted to convex constraints, which allowed the use of linear and convex quadratic programming.

There is a significant reduction in total fuel consumption in the propulsion compared to that obtained in a baseline benchmark solution.

In [14] an MPC based scheme is built to control a spacecraft when it performs maneuvers such as the rendezvous and splices with another spacecraft platform that does not rotate or tumble(see Figure 2.3).

Restrictions were imposed to restrict the spacecraft within the LOS cone during the docking maneuver as well as to make the spacecraft speed as close to or equal to the docking port speed.

Finally the restrictions are also to limit the control inputs subject to the physical limits of the propellers.

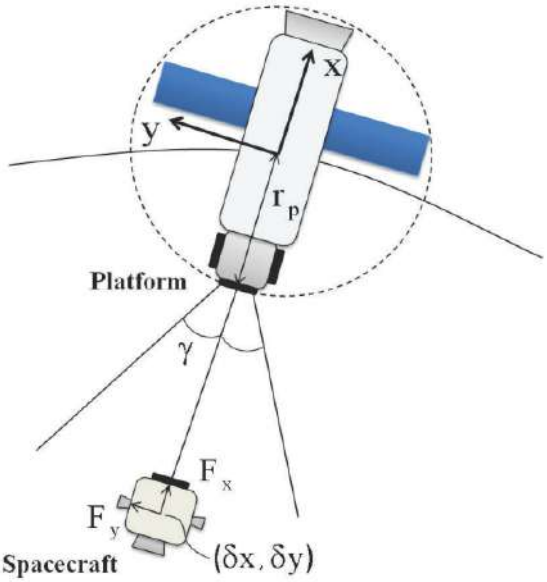


Figure 2.3 – Target platform and spacecraft [14].

The approach combines Linear Quadratic MPC with restrictions that change in real time and that are also dynamically reconfigurable, also this approach approximates the original restrictions. An explicit solution was constructed in the form of a piecewise control law.
Some advantages of the approach presented in this paper:

- The MPC optimization problem can be solved using conventional Quadratic Programming (QP) optimization solvers.
- An explicit solution to the MPC optimization problem can be obtained off-line and be applied online. This gives us the advantage of not embedding the numerical solver for the optimization problem within the spacecraft software package.

Also note that, the MPC controller proposed in this paper includes feedback to compensate for non-measured perturbations such as:

- Error in the thrusters.
- Drag of the air when the spacecraft orbits very close to the terrestrial atmosphere.

The paper [15] applies the MPC approach to Rendezvous and proximity Operations (RPO), based on the extended use of Linear Quadratic MPC scheme with constant horizon, real-type optimization variables, and dynamically reconfigurable linear constraints. The cost function of the MPC problem is based on a scenario and terminal costs defined by Lyapunov stability considerations. To apply the approach proposed in this paper to the following application: "maneuvers approaching a spacecraft to a disk-shaped platform (this platform orbits in circles the planet Earth)".
The considerations necessary to address the above problem are:

1. the mass of the spacecraft is concentrated in its center of mass,
2. circular orbits, and
3. movement of the spaceship limited to a plane.

The three previous considerations are accepted in many maneuvers of this same type. The model used in the MPC strategy was the CWH [5, 6] of the relative spacecraft movement.
It was shown that the multiple constraints that arise in RPO problems can be handled with the MPC approach.
Some of the constraints are:

- The impellers have a maximum capacity due to physical limits.
- Restrictions on the spacecraft to keep it inside the LOS cone while performing the operations necessary for a successful docking with the platform, on the platform is mounted the port for the docking.
- We also have speed restrictions that the spacecraft has a speed equal to or very close to the docking port of the platform (soft docking).

The platform was studied in two states of motion:

1. When the platform does not rotate through its axis, in addition the port for the docking that is in the platform is fixed in relation to the reference system used in the analysis. An off-line solution to the MPC optimization problem is feasible. The solution takes the form of a piecewise constant control law. This solution is suitable for on-line implementation without the need to package a numerical solver for the optimization problem within the spacecraft software package.
2. When the platform rotates through its own axis, it is observed that, it is necessary to predict the rotation of the platform in order to execute the maneuvers and reduce the fuel consumption.

In paper [16] a method based on Constrained MPC was proposed for the autonomous optical Guidance, navigation and control (GNC) problem applied to smooth landings in asteroids. There is a growing interest in the study of asteroids, but to study them is cumbersome because they are very far from the earth, have irregular shapes, are also small size.
To overcome the difficulties mentioned above, a GNC system is necessary for the soft, safe and accurate landing on the asteroid.
In Figure 2.4 the different systems of references are shown, that were used to develop the dynamic model of the spacecraft. In paper [17] a detailed development of the dynamic model of the spacecraft is given.
The design strategy of the MPC controller was as follows. First a path was designed from the initial position of the spacecraft to the desired landing site, it was taken into account that the total time to travel this path is T , second, we apply the MPC method with constraints to follow the path designed in the previous step.
Due to the inherent danger in the maneuvers that are to avoid collisions, in addition to the scientific information of investigations related to this subject, the spacecraft must use the shortest possible time to arrive at a point above the place of landing starting from the initial position , therefore, most of the time T is used for the descent from the point above the landing site towards the aforementioned same.
The MPC controller was used together with an EKF, the EKF is a standard technology used in nonlinear estimation, it is the most suitable for navigation, detailed information of the EKF is found in paper [18].

In the paper [19] a design was proposed to control soft landing, for example in a spacecraft, the approach was based on invariant control sequences as well as on Receding Horizon Control (RHC)(also known as MPC). The control of the soft landing is of great interest in many practical applications, such control is intended to ensure that a mobile vehicle is positioned with very high precision at a target location, simultaneously, as it is closer to the target location, the vehicle speed will decrease. The advantages of a soft landing is that it allows to avoid mechanical damages, as well as the wear of parts. The trajectories obtained by this approach achieve soft landing, regardless of the cost function, the finite time horizon, and even when there are uncertainties. To generate the above-mentioned trajectories the following was done, an invariable control sequence was calculated and used as additional constraints on the MPC problem.
In this paper [20] was developed onboard, real-time, robust GNC algorithms that allow maneuvering near celestial bodies (e.g. planets, moons, asteroids, etc.) that have

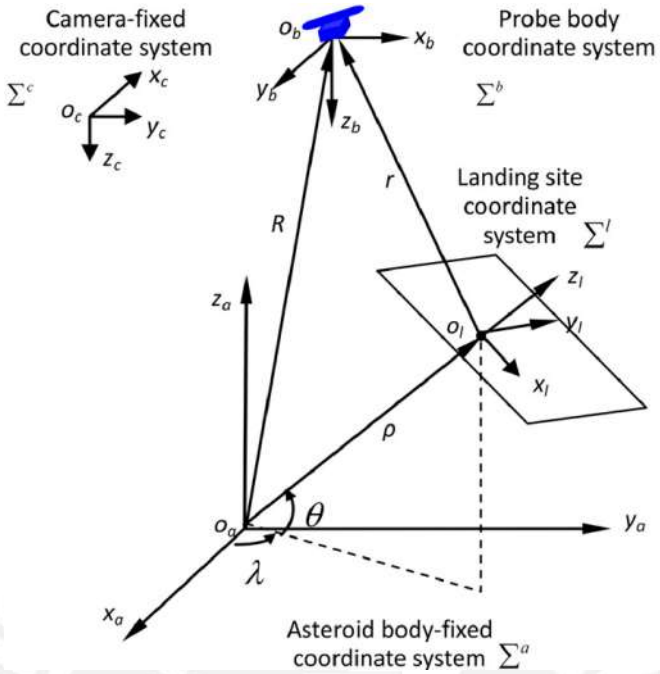


Figure 2.4 – Geometrical relationship of coordinate systems [17].

stable and constant rotation speeds. For the development of such algorithms MPC techniques are used. These algorithms are the feedforward and feedback approaches, a brief explanation of them will be given below:

Feedfordward approach Also known as open-loop guided, based on the PWG (Pseudo-way-Generation) algorithm, this algorithm makes use of a discrete, linear, time-varying dynamic model (include thruster silent times). If a gravitational model is added to the PWG algorithm, the performance of the controller increases dramatically. From the PWG algorithm it is obtained a series of trajectories to guide the spacecraft towards the celestial body of interest, these trajectories are polynomials that are adjusted based on the present state, as well as the necessary states in the future to carry out a maneuver (e.g. The landing).

The PWG algorithm generates a sequence of states, if a gravitational model of a celestial body has been included, this sequence will include the effect of that gravitational field. The generation of a sequence of states with the PWG algorithm is given simultaneously to the determination of the control inputs required for that sequence, this simultaneous work is possible due to the convexification of the nonlinear dynamics governing, control constraints , and the constraints on both trajectory and states, this allows the problem to be modeled as a SOCP (Second-Order Cone Program) to optimize fuel consumption or total energy consumed by the propellers. The resulting SOCP can be solved through Interior Point algorithms. The PWG algorithm is very accurate when there are no perturbations present.

Feedback approach The objective of this algorithm is to guarantee the solution of the trajectories that are obtained from the PWG algorithm, in other words, it is obtained a control signal for each one of the trajectories generated by the PWG algorithm, which allows the proper tracking. The design of this approach is carried out off-line.

2.1 Introduction to the MPC approach.

Proportional-Integral-Differential (PID) controller is a classic control strategy, suitable for linear systems.

There are many real-world problems that cannot be solved or addressed with this technique. For example, if we have a system where it is used the PID controller and also presents disturbances, then the PID controller cannot guarantee the system stability.

Given the limitation of the PID controller, more advanced techniques are developed that have enabled the development of robust controllers such as Linear Quadratic Regulator (LQR), H_2, H_∞ which are the well known. The above provide a law of robust feedback control which optimizes a performance criterion. Although, the LQR and H_∞ can address dynamical systems with constraints.

The problem they present is when the system has great nonlinearities and critical constraints, which may cause stability problems.

To overcome the limitation of these controllers, the concept of MPC arises, which overcomes the limitations presented by the previous controllers mentioned.

2.1.1 General operation of the MPC

The MPC [21–24] is based on the RHC [25]. The MPC has the advantage that introduces a feedback controller, with that, system stability is achieved, in addition to good performance. At each sampling time (Δt) the following is done:

1. The current state that corresponds to the sampling time is measured by sensors or estimated.
2. Predict the $\boldsymbol{u}^*$ (optimal control variable) in a finite time horizon ("N" points), this $\boldsymbol{u}^*$ minimizes an objective function "J", in addition to driving the dynamic of the system towards the desired behavior.
3. The actuator applies only the first data of sequence $\boldsymbol{u}^*$, until the next sampling time.
4. The prediction horizon is receding to the next sampling time.
5. This is iteratively repeated while the system operates in real time.

These set of steps constitute Algorithm 1.

Algorithm 1: MPC Strategy [26]

Input : Initial value of the states
Output : Optimal control variables

1 **repeat**
2 Measure or observe the process state $\boldsymbol{x}$ at time t_k;
3 Formulate the Optimization problem for $t \in [t_k, t_{k+N}]$;
4 Compute the optimal control sequence $\boldsymbol{u}^*(t), t \in [t_k, t_{k+N}]$ by solving the Optimization problem;
5 Apply optimal control $\boldsymbol{u}(t) = \boldsymbol{u}^*(t_k), t \in [t_k, t_{k+1}]$ to the system until $t = t_{k+1}$;
6 Set $k = k + 1$;
7 **until** *stopped by user;*;

The complexity of "J" goes according to the application, it should preferably be as simple as possible. There is no standard rule for choosing "N", but it is advisable to use a value that includes beyond the transient dynamic of the system, also must be considered that depending on the taken by "N" value will increase or decrease the complexity in solving the optimal problem, when we solve this problem will get $\boldsymbol{u}^* = \{u_k^*, u_{k+1}^* ... u_{k+N-1}^*\}$. The MPC approach requires a model of the plant.

This model should capture the significant dynamic, should not be too complex, nor pursue an extremely high accuracy.

Because this causes higher order models which sometimes have a very similar dynamic to simple models with lower order. The MPC approach operation is illustrated in Figure 2.5.

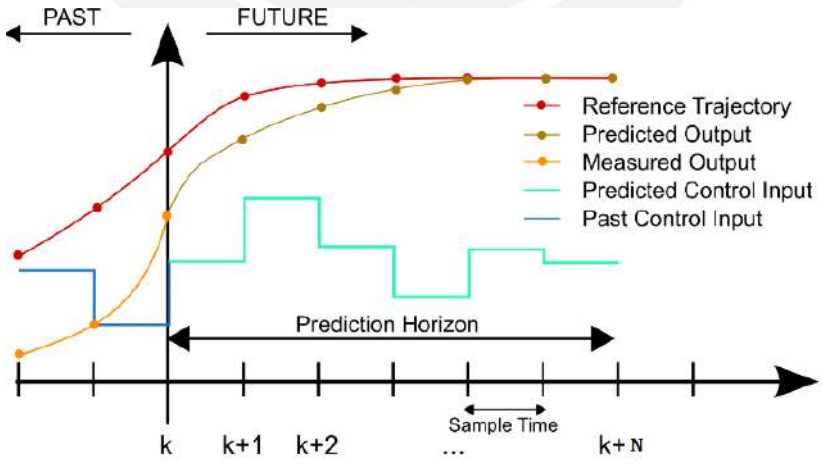


Figure 2.5 – A basic working principle of model predictive control [1]

2.1.2 Advantages and disadvantages of MPC

1. You can specify restrictions for the state vector $\boldsymbol{x}$ and the control variables $\boldsymbol{u}$, besides the "J" which depends on the dynamic of the system.
2. Manage Multiple inputs-Multiple outputs (MIMO) of large-scale systems (linear or nonlinear).
3. Minimization of trajectory tracking error.
4. Noise propagation errors are reduced, also dead time is compensated.
5. Compensation of disturbances is achieved by the feedback feature that is characteristic of the receding horizon approach [25].
6. The MPC controller can be used to control complex processes (e.g. processes in the chemical industry), regardless of whether there are large delays in the processes or they are unstable.
7. The complexity in solving the optimization problem causes the sampling time to be a critical factor.

   In the best case, this time is enough to solve the optimization problem. In the adverse case, when the time is insufficient, auxiliary numerical methods must be applied.

   An example where the sampling time becomes a critical factor is autonomous vehicle navigation.

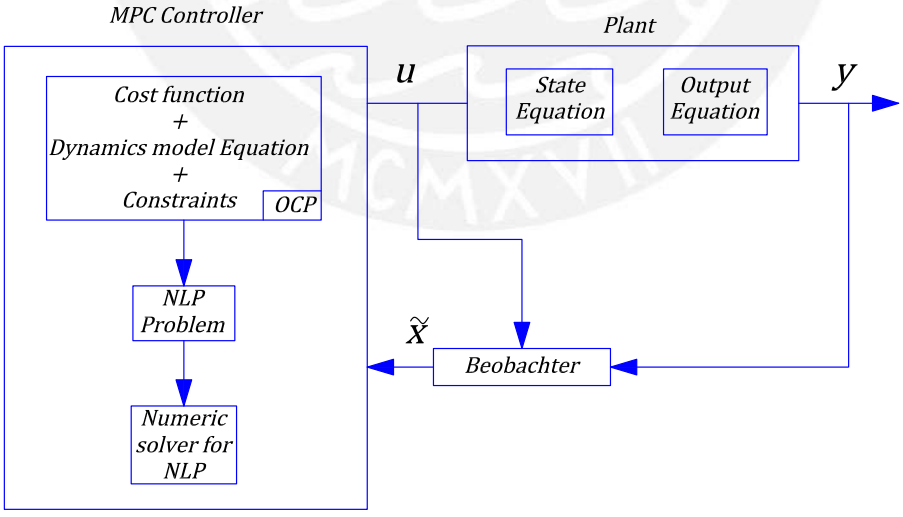


Figure 2.6 – Basic structure of MPC, modified from [27].

2.2 Nonlinear Model Predictive Control

In the study and resolution of various engineering problems, the first step is to obtain a model of that part of the reality that is intended to study, a model is an approximation of reality usually represented by a set of differential equations, sometimes the models obtained from reality are highly nonlinear. When designing a control strategy, if the model of the plant is nonlinear, the controller can be designed based on the linearized model of the plant.

Sometimes the controller so designed has a good performance, sometimes not. Then it is necessary to design a control strategy that can handle highly nonlinear dynamic models such as the Nonlinear Model Predictive Control (NMPC), which will be used in this thesis.

The NMPC not only handles dynamic models with complex nonlinearities, it is also very robust against stochastic perturbations.

Stochastic NMPC will be discussed in Chapter 3.

The NMPC is a case of the general MPC, which is characterized because the equation of the dynamic model is nonlinear or the constraints are nonlinear or the objective function is non-quadratic.

The NMPC is very useful in solving the infinite time horizon Nonlinear Optimal Control Problem (NOCP).

As shown in the Section 2.1.1, the MPC is a finite time horizon that slides in each iteration.

If this finite time horizon is sufficiently long, the control and state variables thus calculated will have a good accuracy, with respect to the exact solution [27, 28].

It may be the case that an NLP problem associated with a NMPC has a nonconvex objective function.

This would make real-time optimization difficult.

2.2.1 Formulation of OCP

The OCPs that will be found in Chapter 3 of this thesis, are chance constrained NOCPs, that is to say in its formulation we find stochastic restrictions (chance constraints), in this chapter we will approach the NOCP without taking into account the restrictions before mentioned, then only the initial condition in the state vector, the dynamic model equation, trajectory constraints and finally a final condition for the state vector will be taken into account, mathematically this is expressed in Equation (2.1).

$$J = \min_{\boldsymbol{x},\boldsymbol{u}} \left\{ \int_0^T L(\boldsymbol{x}(t), \boldsymbol{u}(t))dt + E(\boldsymbol{x}(T)) \right\} \tag{2.1a}$$

subject to

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)),\ t \in [0, T] \tag{2.1b}$$
$$\boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{2.1c}$$
$$\boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \geq \boldsymbol{0},\ t \in [0, T] \tag{2.1d}$$
$$\boldsymbol{r}(\boldsymbol{x}(T)) = \boldsymbol{0} \tag{2.1e}$$

It must be understood by a NOCP, such as that optimal control problem whose dynamic model equation is nonlinear or the constraints are nonlinear or the objective function is non-quadratic. In the Figure 2.7 we can see a schematic of an OCP that has the mathematical formulation of the Equation (2.1), but that counts on a single variable of state and of control, this scheme serves for illustrative purposes, and serves to have an idea of the NOCP with many control variables and states. The solution of the OCP of the Equation (2.1) , will be those variables of state and control that satisfy with all the restrictions, the dynamic equation of the model, and that they make that the objective function takes its minimum value.

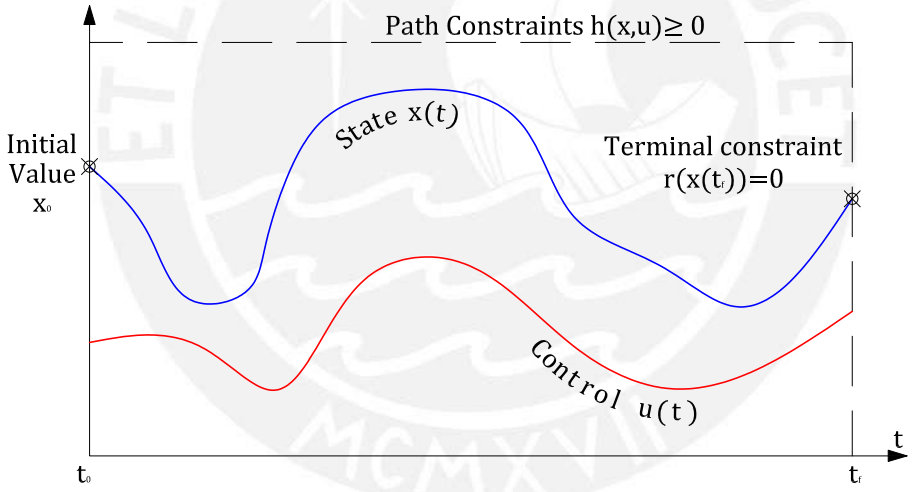


Figure 2.7 – OCP, modified from [29].

2.2.2 OCP Solutions

The interest to solve an OCP arises because these kinds of problems are frequent to find them in industrial processes, in the following fields: chemistry, automotive industry, systems of distribution of energy, economic systems, etc. In the state of the art are several methods for the resolution of OCPs, we can classify them as shown in the figure Figure 2.8.

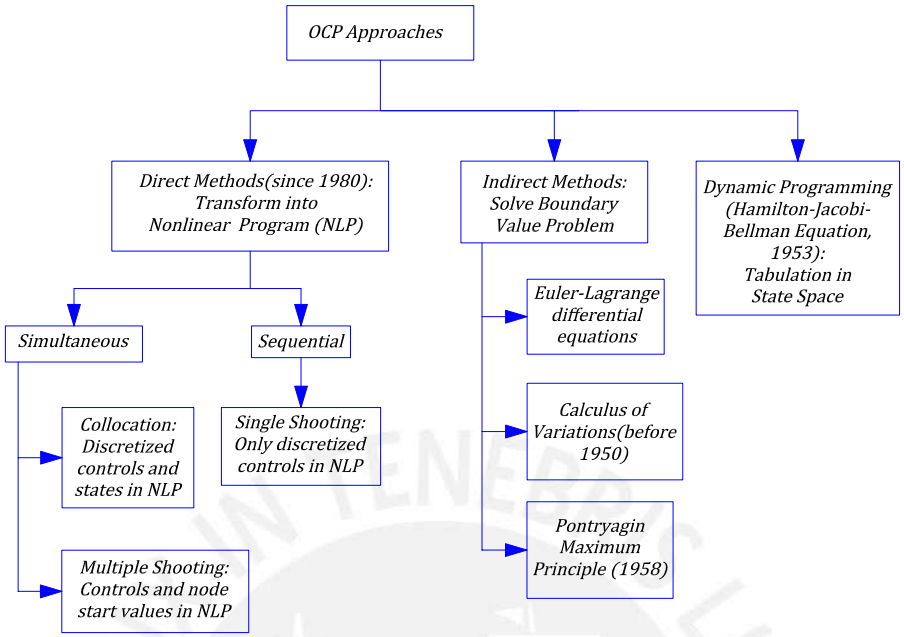


Figure 2.8 – Classification of methods to solve OCPs, modified from [26, 29].

1. **Dynamic Programming** [29–31]: It is based on the principle of optimality of sub-arcs to calculate iteratively a feedback control for the entire time horizon $[t_0, t_f]$ and x_0. The foregoing leads to the Hamilton-Jacobi-Bellman (HJB) equation, which is a partial differential equation in state space. The solution of the HJB equation is a sufficient condition for optimality. Applying numerical methods to solve the HJB equation is not always easy, in addition this method cannot support constraints of the type inequality.

2. **Indirect methods**: They use the necessary optimality conditions of an infinite time problem, to obtain a Boundary Value Problem (BVP) in Ordinary Differential Equations (ODEs), then the BVP has to be solved by a numerical method. They are known as "first optimize, then discretize" because that is precisely the sequence of steps for resolution. Usually the collocation methods are used for BVP resolution [32]. In the scenario where the BVP has multiple local solutions, the solution obtained has to comply with HJB conditions to ensure it is optimal. This class of methods has as main disadvantages that it is very difficult to solve the differential equations due to the large nonlinearities, besides that they cannot handle constraints of the type inequality.

3. **Direct methods** [27, 29]: What these methods do is to transform the infinite OCP into a finite NLP problem, then through the use of the various numerical solvers of the state of the art, we proceed to solve the NLP, for the above are known as "first discretize, the optimize". This class of methods is better than the previous ones because they can handle constraints of the type inequality. The direct methods are based on the parameterization of the control path (piecewise

constant), but differ in the way they handle the state variables. In the state of the art the direct methods are sub-classified into two groups [33]: Sequential Methods and Simultaneous Methods.
In the sequential methods, the control variables are discretized over the entire time horizon (piecewise constant), in each iteration of the NLP solver the dynamic equation of the model will be solved, making use of the initial states $\boldsymbol{x}_0$, and of the variable of control piecewise constant in each subinterval of a time interval, the foregoing implies that the optimization variables are the discretized control variables, and that, therefore, the extended NLP(the optimization variables are the control and states) has to be transformed to a reduced NLP (only the control variables are considered as optimization variables), the numerical methods of Runge-Kutta [34, 35] can be used to solve the dynamic equation of the model.
In the simultaneous methods, both the control and state variables will be discretized in a time interval, then we will have as NLP problem optimization variables to the discretized state and control variables, additionally we add equations representing the dynamics of the model in discrete time. The simulation and optimization are given in parallel, in the Figure 2.8 you can see the main direct methods.

2.2.3 Direct methods for the discretization of OCPs

2.2.3.1 Direct Single Shooting Methods

This method first discretizes the control variables in the interval $[t_0, t_f]$, for this purpose we will use the following partition:

$$t_0 < t_1 < t_2 < \ldots < t_N = t_f \tag{2.2}$$

In each interval of this partition the control variables will be piecewise constant, mathematically this is expressed as follows, $\boldsymbol{u}(t) = \boldsymbol{q}_k$, $\forall t \in [t_k, t_{k+1}]$, with this $\boldsymbol{u}(t)$ only depends of the parameters $\boldsymbol{q} = \{\boldsymbol{q}_0, \boldsymbol{q}_1, ..., \boldsymbol{q}_{N-1}\}$, we can denote this dependence in the expression $\boldsymbol{u}(t; \boldsymbol{q})$, see Figure 2.9.
We will need a numerical method to solve the following Initial Value Problem (IVP):

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0,\ \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t; \boldsymbol{q})),\ \forall t \in [t_0, t_f] \tag{2.3}$$

Once the IVP is solved, the state variables $\boldsymbol{x}(t)$, $\forall t \in [t_0, t_f]$, are dependent variables, this is denoted as $\boldsymbol{x}(t; \boldsymbol{q})$, see Figure 2.9. Depending on the dynamic model of the plant will choose the numerical method to solve the IVP, this numerical solver must properly handle the sensitivity of the dynamic model. Then we proceed to discretize the constraints of the trajectory Equation (2.1d), in the points of the partition Equation (2.2), thus we obtain the following finite NLP:

$$J = \min_{\boldsymbol{q}} \left\{ \int_0^T L(\boldsymbol{x}(t; \boldsymbol{q}), \boldsymbol{u}(t, \boldsymbol{q}))dt + E(\boldsymbol{x}(T, \boldsymbol{q})) \right\} \tag{2.4a}$$

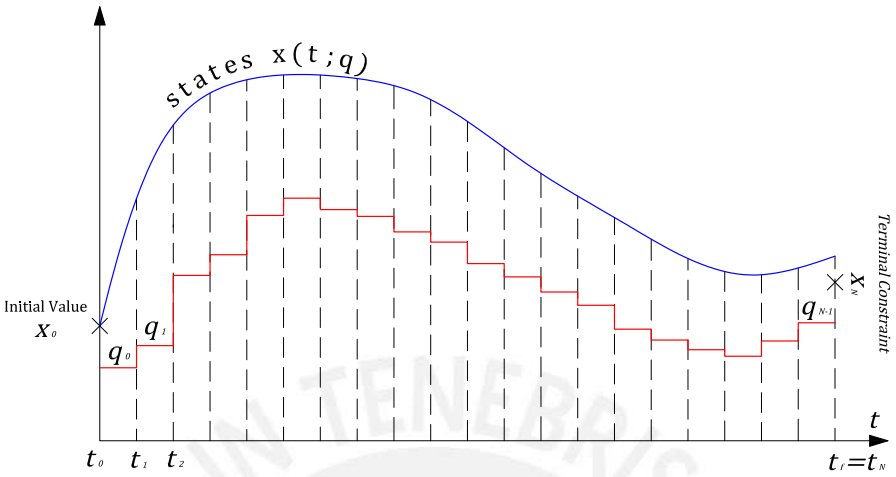


Figure 2.9 – Direct Single Shooting scheme, this graph illustrates a $u \in \Re$, $x \in \Re$, in the more general case we have that $\boldsymbol{x} \in \Re^p$, $\boldsymbol{q} \in \Re^m$.

subject to

$$\boldsymbol{h}(\boldsymbol{x}(t_k;\boldsymbol{q}),\boldsymbol{u}(t_k;\boldsymbol{q})) \geq 0,\ k = 0,1,2,\ldots,N \tag{2.4b}$$

$$\boldsymbol{r}(\boldsymbol{x}(T;\boldsymbol{q})) = 0 \tag{2.4c}$$

The Equation (2.4) is solved by a numerical solver based on SQP or IP (Interior Point) methods. The advantages of direct single shooting method are:

- Numerical solver can be used for Differential Algebraic Equations (DAEs) or ODEs, full adaptative, and that control the error.
- Low number of optimization variables even when working with large scale DAEs or ODEs.
- It is only necessary to assume the value of the discretized control variables at the beginning.
- You can easily handle "Active set changes".

Some disadvantages of this method:

- Handling constraints on the trajectory of state variables is complicated.
- It is complicated to work unstable systems with this method [36].
- The resolution of the dynamic model of the plant $\boldsymbol{x}(t;\boldsymbol{q})$, in occasions has a highly nonlinear dependence with $\mathbf{q}$.

This type of direct methods is implemented in engineering software packages such as gOPT([37]), DYOS([38]).

2.2.3.2 Collocation Methods

In this method [39], the control and state variables are first discretized in a partition like Equation (2.2), in each one of the intervals of the partition, the control variable is piecewise constant, therefore its values are the sequence $\boldsymbol{q} = \{\boldsymbol{q}_0, \boldsymbol{q}_1, ...\boldsymbol{q}_{N-1}\}$, $\boldsymbol{q}_k$ for the interval $[t_k, t_{k+1}]$, the optimization variables of the NLP are the sequence $\boldsymbol{q}$. The values of the state variables at each point in the partition are expressed by $\boldsymbol{m}_k \approx \boldsymbol{x}(t_k)$. In the collocation methods the dynamic equation of the model is replaced by a finite set of constraints type equation:

$$\boldsymbol{c}_k(\boldsymbol{q}_k, \boldsymbol{m}_k, \bar{\boldsymbol{m}}_k, \boldsymbol{m}_{k+1}) = 0 \tag{2.5}$$

In Equation (2.5) the term $\bar{\boldsymbol{m}}_k$ refers to possible additional intermediate collocation points in the interval $[t_k, t_{k+1}]$, in the case these additional collocation points will be used, the appropriate selection of such points will provide a much greater approximation, they are usually chosen to be equal to the roots of orthogonal polynomials. It also approaches the objective function (Equation (2.1a)) in each interval of the partition (Equation (2.2)):

$$L_k(\boldsymbol{q}_k, \boldsymbol{m}_k, \bar{\boldsymbol{m}}_k, \boldsymbol{m}_{k+1}) \tag{2.6}$$

The large-scale and sparse NLP problem is:

$$J = \min_{\boldsymbol{q}, \boldsymbol{m}, \bar{\boldsymbol{m}}} \left\{ \sum_{k=0}^{N-1} L_k(\boldsymbol{q}_k, \boldsymbol{m}_k, \bar{\boldsymbol{m}}_k, \boldsymbol{m}_{k+1}) + E(\boldsymbol{m}_N) \right\} \tag{2.7a}$$

$$\boldsymbol{c}_k(\boldsymbol{q}_k, \boldsymbol{m}_k, \bar{\boldsymbol{m}}_k, \boldsymbol{m}_{k+1}) = 0,\ k = 0, 1, 2, 3 \ldots N-1 \tag{2.7b}$$

$$\boldsymbol{h}_k(\boldsymbol{q}_k, \boldsymbol{m}_k, \bar{\boldsymbol{m}}_k, \boldsymbol{m}_{k+1}) \geq 0,\ k = 0, 1, 2, 3 \ldots N-1 \tag{2.7c}$$

$$\boldsymbol{m}_0 = \boldsymbol{x}_0 \tag{2.7d}$$

$$\boldsymbol{r}(\boldsymbol{m}_N) = 0 \tag{2.7e}$$

The NLP of Equation (2.7) is solved iteratively by a numerical solver, in each iteration the Equation (2.5) are not necessarily satisfied, when the numerical solver converges, the Equation (2.7) are satisfied and are obtained as outputs to $\boldsymbol{q}_k^*, \boldsymbol{m}_k^*, \bar{\boldsymbol{m}}_k^*$. Like the previous method of Section 2.2.3.1, the problem of Equation (2.7) is solved by a numeric solver based on SQP [40] or Interior Point methods [41].
Some advantages of this method are [29]:

- A large-scale, but widely dispersed (sparse) NLP is obtained.
- Unlike direct single shooting, this method can use information concerning the trajectory of the states in the initialization.
- Very fast local convergence.
- Efficient management of systems that are unstable.
- Robust handling of path constraints and terminal constraints.

Some of its disadvantages:

- The adaptive control of the discretization error can change to the partition Equation (2.2) (e.g. greater number of collocation points), this causes the dimensions of the NLP to change.

Some of the software packages used for its implementation are the IpOpt [41], DIRCOL [42].

2.2.3.3 Direct Multiple Shooting Methods

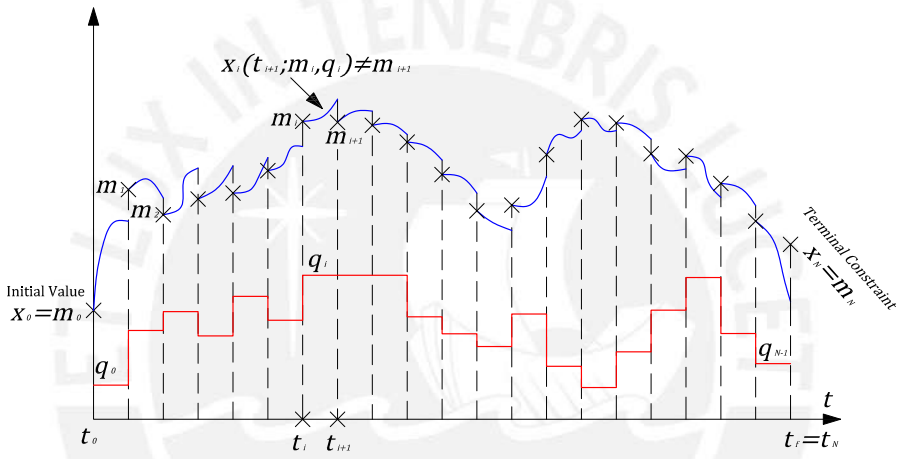


Figure 2.10 – Direct Multiple Shooting scheme when the solver still does not converge.

This method [43], combines the advantages of a simultaneous method like the one of collocation(Section 2.2.3.2), with the advantages of direct single shooting methods(Section 2.2.3.1), therefore, this method can make an adaptive control of the error in the ODE solvers .
The first step of this method is to discretize the control variables in each interval of the partition Equation (2.2), in this way the control variable will be piecewise constant in each of these intervals, mathematically this is expressed in $\boldsymbol{u}(t) = \boldsymbol{q}_k$, for $t \in [t_k, t_{k+1}]$, $i = 0, 1 \dots N-1$, then the dynamic equation of the model in each interval $[t_k, t_{k+1}]$ will be solved using an artificial initial state $\boldsymbol{m}_k$, in addition to the control variable $\boldsymbol{q}_k$.

$$\begin{aligned} \dot{\boldsymbol{x}}_k(t) &= \boldsymbol{f}(\boldsymbol{x}_k(t), \boldsymbol{q}_k), \ t \in [t_k, t_{k+1}] \\ \boldsymbol{x}_k(t_k) &= \boldsymbol{m}_k \end{aligned}$$

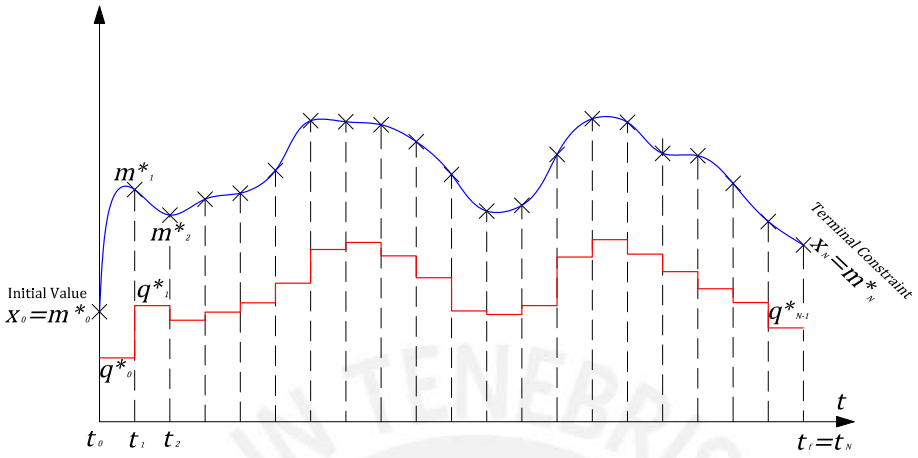


Figure 2.11 – Direct Multiple Shooting scheme when the solver converge.

A set of trajectories that are dependent on the time, the artificial initial state $\boldsymbol{m}_k$, and the control variable $\boldsymbol{q}_k$, $\boldsymbol{x}_k(t;\boldsymbol{m}_k,\boldsymbol{q}_k)$ will be obtained, see Figure 2.10. Simultaneously with the resolution of the dynamic equation of the model in each interval of the partition, the following integrals will be calculated numerically:

$$l_k(\boldsymbol{m}_k,\boldsymbol{q}_k) := \int_{t_k}^{t_{k+1}} L(\boldsymbol{x}_k(t_k;\boldsymbol{m}_k,\boldsymbol{q}_k))dt$$

It will be necessary additionally to impose constraints on the artificial initial states $\boldsymbol{m}_k$, $\boldsymbol{m}_{k+1} = \boldsymbol{x}_k(t_{k+1};\boldsymbol{m}_k,\boldsymbol{q}_k)$, this ensures that when the solver of the NLP converges, a continuous path in the states will be obtained, see Figure 2.11.
After the steps explained above, we get the following sparse structure NLP:

$$J = \min_{\boldsymbol{q},\boldsymbol{m}} \left\{ \sum_{k=0}^{N-1} l_k(\boldsymbol{q}_k,\boldsymbol{m}_k) + E(\boldsymbol{m}_N) \right\} \tag{2.9a}$$

$$\boldsymbol{x}_k(t_{k+1};\boldsymbol{q}_k,\boldsymbol{m}_k) = \boldsymbol{m}_{k+1},\ k = 0,1,2,3\ldots N-1 \tag{2.9b}$$

$$\boldsymbol{h}_k(\boldsymbol{q}_k,\boldsymbol{m}_k) \geq 0,\ k = 0,1,2,3\ldots N \tag{2.9c}$$

$$\boldsymbol{x}_0 = \boldsymbol{m}_0 \tag{2.9d}$$

$$\boldsymbol{r}(\boldsymbol{m}_N) = 0 \tag{2.9e}$$

This method has been used in the resolution of practical off-line problems such as [44], it is also widely used in real-time optimization [45, 46].
Some of the advantages of this method are:

- State trajectory information can be used at initialization.

- Uses adaptive numerical solver for ODEs or DAEs.
- The resulting NLP has a fixed dimension.
- Can handle unstable systems, path constraints, final conditions.
- In each of the solver iterations to solve the NLP, the part that most demands computational effort is the solution of the ODEs, this can easily be parallelized.

Some disadvantages:

- The NLP that is obtained with this method is of smaller dimension with respect to the one obtained in the collocation method, but less sparse.

In this thesis the OCP in each prediction horizon of the MPC is discretized using Runge-Kutta 4th order.

2.3 Solution methods for NLP

After applying some method of discretization to the NOCP, we will obtain an NLP, the resolution of the optimization problem is to find the value of the optimization variables that satisfy the equations, inequalities, and that minimize or maximize (depending on the case) the value of the objective function. The NLP has the following structure:

$$\min_{\boldsymbol{v}} F(\boldsymbol{v}) \tag{2.10a}$$

subject to

$$\boldsymbol{G}(\boldsymbol{v}) = \boldsymbol{0} \tag{2.10b}$$
$$\boldsymbol{H}(\boldsymbol{v}) \leq \boldsymbol{0} \tag{2.10c}$$

The terms of Equation (2.10) are:

- $\boldsymbol{v}$ is a vector conformed by the discretized control and state variables in the partition Equation (2.2).
- $F(\boldsymbol{v})$ is the objective function, dependent on the elements that make up the vector $\boldsymbol{v}$.
- $\boldsymbol{H}(\boldsymbol{v})$ group of equations that are derived from the dynamic equation of the model, is a function of the elements of vector $\boldsymbol{v}$.
- $\boldsymbol{G}(\boldsymbol{v})$, a group of inequalities derived from constraints in the trajectory of states, constraints on control variables, is a function of the elements of vector $\boldsymbol{v}$.

The three substructures of the NLP are the equations(Equation (2.10b)), inequalities(Equation (2.10c)), and the objective function(Equation (2.10a)). If the previous substructures were all linear, then the optimization problem is a Linear Programming(LP) problem, otherwise it is called NLP. The optimization problem of Equation (2.10) is usually nonconvex, so more sophisticated numerical methods are required than are used for the convex case. It will be mentioned three numerical methods for this class of NLP.

- The quasi-sequential method.
- The sequential-quadratic-programming methods.
- The interior-point methods.

The choice of which method to use depends on the properties of the aforementioned as well as the characteristics of the optimization problem that is intended to be solved. A brief explanation of each of these methods will be given, with an emphasis on the most important of each of them.

2.3.1 Quasi-Sequential Method

This method discretizes control and state variables using collocation on finite elements [47] or some Runge-Kutta method [48, 49], and can handle constraints in the state trajectory.
The approach has to solve the set of algebraic equations $\boldsymbol{C}(\boldsymbol{X},\boldsymbol{U}) = \boldsymbol{0}$ (Figure 2.12) resulting from the discretization of DAEs or ODEs, making use of a numerical solver, usually a Newton method [48, 50, 51], in a simulation layer. The above allows to reduce the dimension of the optimization problem, this is because the equality constraints and the state variables are eliminated.
The resulting NLP (Figure 2.12, third block) consists only of inequality constraints $\begin{bmatrix} \boldsymbol{h}(\boldsymbol{X}(\boldsymbol{U}),\boldsymbol{U}) \\ \boldsymbol{X}(\boldsymbol{U}) \end{bmatrix}$ and control variables $\boldsymbol{U}$, the NLP stated in this way greatly improves the perfomance of the Line Search [50–52]. The solution of this NLP is done in the optimization layer.
This approach, compared to the Simultaneous Approach (Sections 2.2.3.2 and 2.2.3.3), requires a greater computational effort because it solves the set of equations $\boldsymbol{C}(\boldsymbol{X},\boldsymbol{U}) = \boldsymbol{0}$, in each iteration, if the solution of the set of equations by some numerical solver converges quickly, then the Quasi-sequential method [53–55] can be more efficient than the Simultaneous Approach. This approach is appropriate for highly nonlinear large-scale OCPs.

This thesis uses this method implemented in IpOpt (more details about IpOpt software in Section 2.3.3), below a brief explanation of the implementation:
The IpOpt in each iteration k provides an $\boldsymbol{U}_k$, with this $\boldsymbol{U}_k$ will solve the system of equations $\boldsymbol{C}(\boldsymbol{X},\boldsymbol{U}) = \boldsymbol{0}$ obtaining an $\boldsymbol{X}_k$.
Then we derive with respect to $\boldsymbol{U}$ the system of equations $\boldsymbol{C}(\boldsymbol{X},\boldsymbol{U}) = \boldsymbol{0}$, with what we would have:

$$\frac{d\boldsymbol{C}(\boldsymbol{X},\boldsymbol{U})}{d\boldsymbol{U}} = \frac{\partial \boldsymbol{C}(\boldsymbol{X},\boldsymbol{U})}{\partial \boldsymbol{X}} \frac{d\boldsymbol{X}}{d\boldsymbol{U}} + \frac{\partial \boldsymbol{C}(\boldsymbol{X},\boldsymbol{U})}{\partial \boldsymbol{U}} = \boldsymbol{0} \tag{2.11}$$

Then the matrix $\left.\frac{d\boldsymbol{X}}{d\boldsymbol{U}}\right|_k$ is found by applying some linear algebra package to Equation (2.11):

$$\left.\frac{\partial \boldsymbol{C}(\boldsymbol{X},\boldsymbol{U})}{\partial \boldsymbol{X}}\right|_{\substack{\boldsymbol{X}_k \\ \boldsymbol{U}_k}} \left(\frac{d\boldsymbol{X}}{d\boldsymbol{U}}\right) + \left.\frac{\partial \boldsymbol{C}(\boldsymbol{X},\boldsymbol{U})}{\partial \boldsymbol{U}}\right|_{\substack{\boldsymbol{X}_k \\ \boldsymbol{U}_k}} = \boldsymbol{0} \rightarrow \left.\frac{d\boldsymbol{X}}{d\boldsymbol{U}}\right|_k$$

With the previous matrix we can find the gradient of the objective function and the Jacobian of the constraints (there are only inequalities):

$$\frac{d\varphi(\boldsymbol{X}(\boldsymbol{U}),\boldsymbol{U})}{d\boldsymbol{U}} = \frac{d\varphi(\boldsymbol{X},\boldsymbol{U})}{d\boldsymbol{U}} = \frac{\partial\varphi(\boldsymbol{X},\boldsymbol{U})}{\partial\boldsymbol{X}}\frac{d\boldsymbol{X}}{d\boldsymbol{U}} + \frac{\partial\varphi(\boldsymbol{X},\boldsymbol{U})}{\partial\boldsymbol{U}} \rightarrow$$
$$\left.\frac{d\varphi(\boldsymbol{X}(\boldsymbol{U}),\boldsymbol{U})}{d\boldsymbol{U}}\right|_{\boldsymbol{U}_k} = \left.\frac{d\varphi(\boldsymbol{X},\boldsymbol{U})}{d\boldsymbol{U}}\right|_{\boldsymbol{U}_k} = \left.\frac{\partial\varphi(\boldsymbol{X},\boldsymbol{U})}{\partial\boldsymbol{X}}\right|_{\substack{\boldsymbol{X}_k\\\boldsymbol{U}_k}} \left.\frac{d\boldsymbol{X}}{d\boldsymbol{U}}\right|_k + \left.\frac{\partial\varphi(\boldsymbol{X},\boldsymbol{U})}{\partial\boldsymbol{U}}\right|_{\substack{\boldsymbol{X}_k\\\boldsymbol{U}_k}} \tag{2.12a}$$

$$\frac{d}{d\boldsymbol{U}}\begin{bmatrix}\boldsymbol{h}(\boldsymbol{X}(\boldsymbol{U}),\boldsymbol{U})\\ \boldsymbol{X}(\boldsymbol{U})\end{bmatrix} = \frac{d}{d\boldsymbol{U}}\begin{bmatrix}\boldsymbol{h}(\boldsymbol{X},\boldsymbol{U})\\ \boldsymbol{X}\end{bmatrix} = \begin{bmatrix}\frac{\partial\boldsymbol{h}(\boldsymbol{X},\boldsymbol{U})}{\partial\boldsymbol{X}}\frac{d\boldsymbol{X}}{d\boldsymbol{U}} + \frac{\partial\boldsymbol{h}(\boldsymbol{X},\boldsymbol{U})}{\partial\boldsymbol{U}}\\ \frac{d\boldsymbol{X}}{d\boldsymbol{U}}\end{bmatrix} \rightarrow$$
$$\left.\frac{d}{d\boldsymbol{U}}\begin{bmatrix}\boldsymbol{h}(\boldsymbol{X}(\boldsymbol{U}),\boldsymbol{U})\\ \boldsymbol{X}(\boldsymbol{U})\end{bmatrix}\right|_{\boldsymbol{U}_k} = \left.\frac{d}{d\boldsymbol{U}}\begin{bmatrix}\boldsymbol{h}(\boldsymbol{X},\boldsymbol{U})\\ \boldsymbol{X}\end{bmatrix}\right|_{\boldsymbol{U}_k} = \begin{bmatrix}\left.\frac{\partial\boldsymbol{h}(\boldsymbol{X},\boldsymbol{U})}{\partial\boldsymbol{X}}\right|_{\substack{\boldsymbol{X}_k\\\boldsymbol{U}_k}} \left.\frac{d\boldsymbol{X}}{d\boldsymbol{U}}\right|_k + \left.\frac{\partial\boldsymbol{h}(\boldsymbol{X},\boldsymbol{U})}{\partial\boldsymbol{U}}\right|_{\substack{\boldsymbol{X}_k\\\boldsymbol{U}_k}}\\ \left.\frac{d\boldsymbol{X}}{d\boldsymbol{U}}\right|_k\end{bmatrix} \tag{2.12b}$$

Then it is necessary to find the objective function, the constraints evaluated in $\boldsymbol{U}_k$, $\boldsymbol{X}_k$:

$$\left.\varphi(\boldsymbol{X}(\boldsymbol{U}),\boldsymbol{U})\right|_{\boldsymbol{U}_k} = \left.\varphi(\boldsymbol{X},\boldsymbol{U})\right|_{\substack{\boldsymbol{X}_k\\\boldsymbol{U}_k}} \tag{2.13a}$$

$$\left.\begin{bmatrix}\boldsymbol{h}(\boldsymbol{X}(\boldsymbol{U}),\boldsymbol{U})\\ \boldsymbol{X}(\boldsymbol{U})\end{bmatrix}\right|_{\boldsymbol{U}_k} = \left.\begin{bmatrix}\boldsymbol{h}(\boldsymbol{X},\boldsymbol{U})\\ \boldsymbol{X}\end{bmatrix}\right|_{\substack{\boldsymbol{X}_k\\\boldsymbol{U}_k}} \tag{2.13b}$$

After it, is necessary to find the Hessian of the Lagrangian, this will be approximated numerically by the L-BFGS algorithm, which comes implemented in the IpOpt.
The IpOpt will iterate until finding the optimal solution $\boldsymbol{U}^*$.

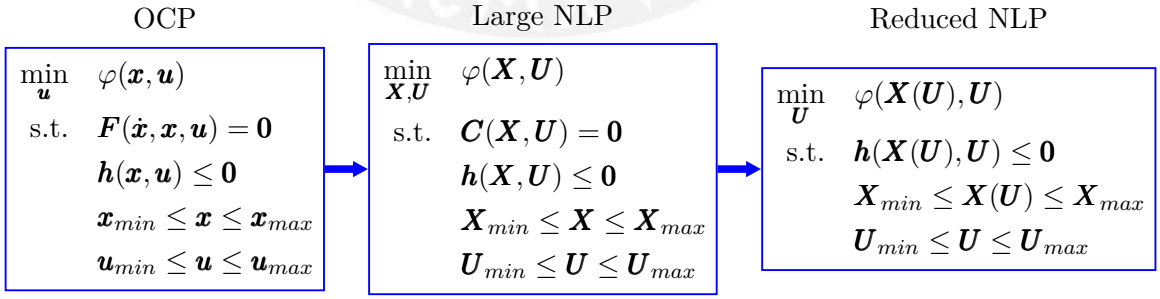


Figure 2.12 – Illustrative scheme of how to convert an OCP to a reduced NLP by applying the Quasi-sequential method.

2.3.2 Sequential Quadratic Programming Method [26]

SQP [52, 56] is one of the most commonly used numerical methods for solving NLPs, because it is very robust and effective [57]. Its operation is as follows: At each iteration the SQP algorithm solves a QP problem [36] , which is obtained through the linearization of the NLP in the vicinity of the point $\boldsymbol{v}_k$ corresponding to the iteration k ($k = 0, 1, 2 \ldots$). The solution of the QP problem at iteration k provides a descent vector $\vec{\boldsymbol{d}}$, to the next point $\boldsymbol{v}_{k+1}$. The next step is to minimize a merit function in the direction and sense of the vector $\vec{\boldsymbol{d}}$, with this determining the length to the next point $\boldsymbol{v}_{k+1}$. The process is repeated iteratively, the result of these iterations is a sequence $\{\boldsymbol{v}_0, \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \ldots, \boldsymbol{v}^*\}$ that converges to $\boldsymbol{v}^*$.
Next, the resolution of the optimization problem Equation (2.10) will be explained using this method. Starting from an initial point $\boldsymbol{v}_0$, the SQP algorithm iterates as follows:

$$\boldsymbol{v}_{k+1} = \boldsymbol{v}_k + \alpha_k \, \vec{\boldsymbol{d}}_k, \; k = 0, 1, \ldots \tag{2.14}$$

In Equation (2.14) α_k is the length of the step that is determined by the merit function, $\vec{\boldsymbol{d}}_k$ is the result of solving the following QP problem Equation (2.15), obtained from the linearization of the NLP at point $\boldsymbol{v}_k$.

$$\min_{\vec{\boldsymbol{d}}_k} \{(\nabla_{\boldsymbol{v}} F|_{\boldsymbol{v}_k})^\top \vec{\boldsymbol{d}}_k + 0.5(\vec{\boldsymbol{d}}_k)^\top \boldsymbol{P}_k \, \vec{\boldsymbol{d}}_k\} \tag{2.15a}$$

subject to

$$\boldsymbol{G}(\boldsymbol{v}_k) + \nabla_{\boldsymbol{v}} \boldsymbol{G}|_{\boldsymbol{v}_k} \vec{\boldsymbol{d}}_k = \boldsymbol{0} \tag{2.15b}$$

$$\boldsymbol{H}(\boldsymbol{v}_k) + \nabla_{\boldsymbol{v}} \boldsymbol{H}|_{\boldsymbol{v}_k} \vec{\boldsymbol{d}}_k \leq \boldsymbol{0} \tag{2.15c}$$

From Equation (2.15a), $\boldsymbol{P}_k$ is an approximation of the Hessian of the Lagrangian:

$$\boldsymbol{P}_k \approx \nabla^2_{\boldsymbol{v}} L(\boldsymbol{v}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \Big|_{\substack{\boldsymbol{v}_k \\ \boldsymbol{\lambda}_k \\ \boldsymbol{\mu}_k}} \tag{2.16a}$$

$$L(\boldsymbol{v}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = F(\boldsymbol{v}) + \boldsymbol{\lambda}^\top \boldsymbol{G}(\boldsymbol{v}) + \boldsymbol{\mu}^\top \boldsymbol{H}(\boldsymbol{v}) \tag{2.16b}$$

To approximate the Hessian matrix of the Lagrangian(Equation (2.16a)) one can use Quasi-Newton methods [50–52, 58], or the method of Gauss-Newton [52, 59]. For the solution of the QP problem Equation (2.15), the most well-known convex optimization methods such as Interior Point Methods [60] or Active Set Methods [52] can be used. For the implementation of the SQP method, Active Set Methods are used commonly, and not the Interior Point Methods, because in using the latter, the computational complexity is considerably increased.
NPSO [61] was one of the first implementations of SQP methods, based on the FORTRAN programming language, used for solving smooth problems. For large-scale problems solving there are efficient commercial programs such as SNOPT [62], GAMS [63], the first of which uses a reduced Hessian active-set method. Among the free software packages

is NLPQL [64], based on FORTRAN.
The SQP methods prove to be very robust when both the gradient and the jacobian do not approximate in numerical form, but rather are calculated analytically, they are also robust in the scenario when the objective function, and the NLP constraints are smooth nonlinear functions [65]. It is usual that of the multiple iterations used by the SQP method, some of them do not satisfy the nonlinear constraints, for this reason and in addition to the above, this method is applied in those problems that present low nonlinearities.

2.3.3 Interior Point Method

This method is one of the most efficient algorithms for solving large-scale NLPs. There are a number of OCPs that discrete them are obtained NLPs whose objective functions are nonconvex, in that context it is no longer possible to apply the interior-point methods that are for convex optimization problems, but rather will use much complex interior-point algorithms that need a series of additional resources such as linear algebra solvers, efficient methods to compute the gradient and hessian, regularization methods, efficient use of the Line-search strategy, to mention a few.
To solve the NLPs of this thesis will be used the IpOpt solver [41], this numerical solver is an implementation of the interior-point method specially designed to address large-scale optimization problems and containing many high nonlinearities.

IpOpt solver [26, 41, 66]

The IpOpt (Interior Point Optimizer) solver is a free software designed for solving large-scale optimization problems, based on a primal-dual interior-point line-search filter method [41]. This solver in its beginnings was developed with FORTRAN [67], currently it can find the software implementing in several languages of programming like C ++, C, Python, Matlab.
If we apply this solver to an optimization problem whose objective function is nonconvex, and we consider that within the admissible region for optimization variables there is more than a local minimum, if we start from an arbitrary initial point $\boldsymbol{v}_0$, we will find a local minimum which cannot be guaranteed to be the global minimum, for more details about the mathematical fundamentals of this solver see [41].
For the realization of this thesis was used the IpOpt code in C++ programming language [68], in addition worked under the Linux Ubuntu operating system . In order to solve an NLP problem using the IpOpt solver, it is necessary first to define some parameters, functions with which this solver works, such as: The number of optimization variables (n), the total number of equations plus inequations (m) , the objective function, the constraints, the upper and lower limits of the optimization variables must be defined. In addition, since the IpOpt solver is based on an interior-point method, it will be necessary to provide information of the first and second derivatives.
For NMPC problems whose finite time horizon is relatively large, and which also have a high number of state variables, it is practical to approximate the Hessians using a C++ Automatic Differentiation Solver [69] such as CASADI [70], or using the approximation method of the IpOpt, L-BFGS Hessian approximation [71, 72]. The Figure 2.13 shows a

flow diagram of how the IpOpt solves an NLP.
When we define the Jacobian of the constrains and the Hessian of the Lagrangian in the corresponding functions in the IpOpt C++ code, previously, in the function where the initial parameters are defined, we should have indicated the number of nonzero elements of both the Jacobian as Hessian, therefore, we must only define the elements of each of these matrices that are nonzero, this is advantageous in those matrices very dispersed with a large number of zeros.
The IpOpt can work efficiently with large dispersed matrices, this is achieved through the use of efficient sparse linear algebra solvers, then the linear algebra solvers that come within the package will be listed:

HSL-MA27 [73] is the solver that uses IpOpt by default, uses a sequential algorithm. It manages to solve a system of linear equations $Ax = b$ of large scale by Gaussian elimination.

HSL-MA57 [74] it is a version after the MA27, of very similar operation, its operation differs with respect to the first one exposed because it uses the aggregation of scattered elements, with this improves the performance.

HSL-MA86 solves a system of symmetric linear equations through LDLT factorization, which is specially designed for multi-core processors.

MUMPS [75] solves linear equations system by LU factorization when matrices are square, in the case of symmetric matrices it uses the LDLT factorization.

PARDISO [76] multithreaded parallel processing algorithm, using OpenMP [77], the algorithm allows the Cholesky or LDLT factorization for symmetric matrices, in the case of non-symmetric matrices it uses a QR factorization.

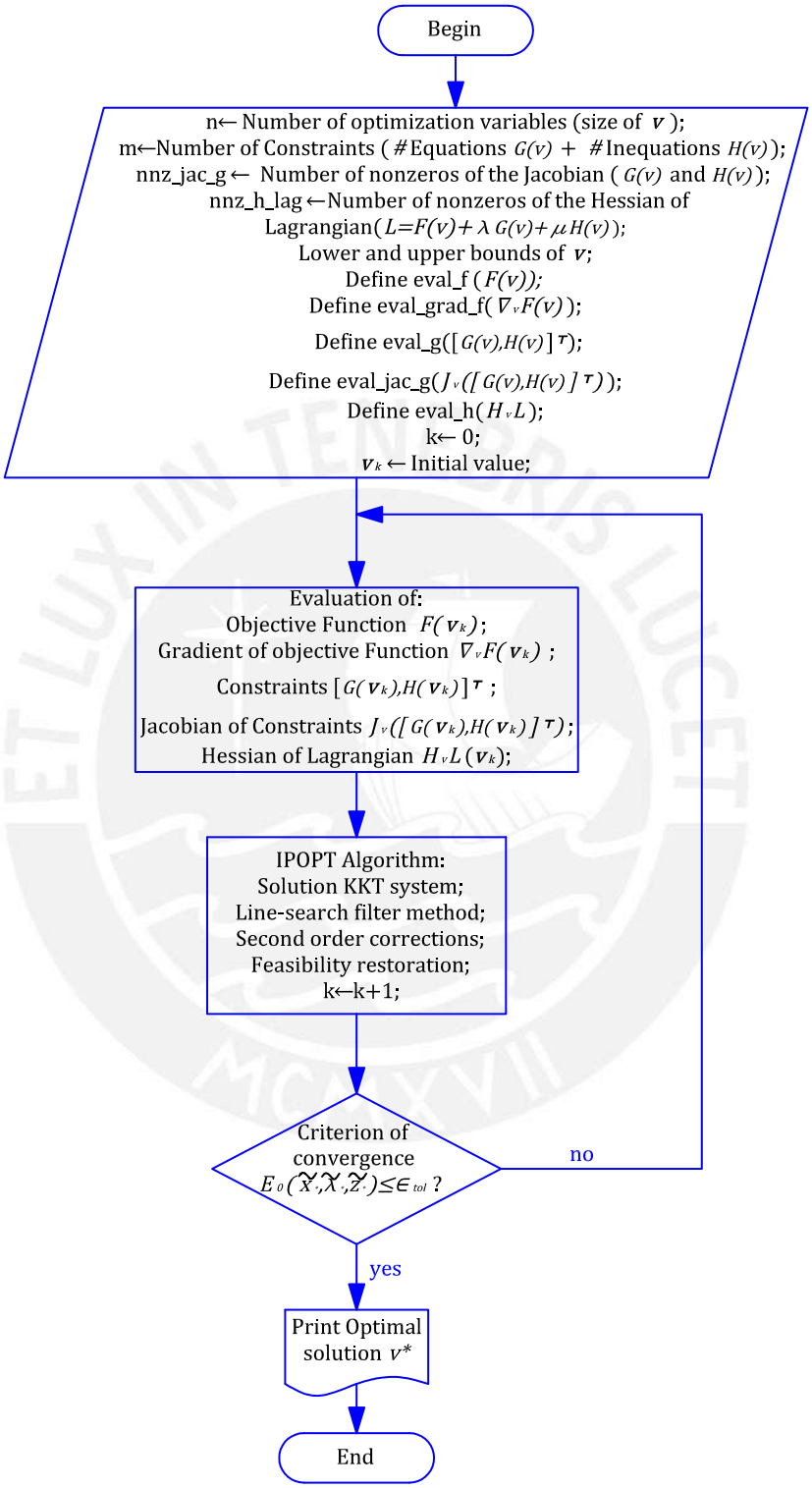


Figure 2.13 – Flowchart of the IpOpt, modified from [26].

Chapter 3

Chance Constrained MPC

The operation of the chance constrained MPC is the same as explained in Section 2.1.1, the difference with respect to the deterministic MPC is in the formulation of the OCP, NLP, now in the latter are present chance constraints, it will be explained that it is a Chance Constraint later.
Determining the value of chance constraints is not easy, and it can demand a lot of computational effort, which results in more calculation time by iteration, this can be very critical for systems with high sampling frequencies, fortunately in the state of the art there are several methods of approaching chance constraints such as: Back Mapping, Robust Optimization, Sample Average Approximation, Analytic Approximation; these will be explained later in the Chapter. Approximation methods transform the chance constrained NLP into a Deterministic Optimization Problem that can be solved by the numerical solvers discussed in Section 2.3.
There are engineering applications with operational restrictions that sometimes cannot be satisfied due to unexpected events, extreme events, the presence of measurement errors, etc. In such situations, what is done in practice is to establish a limit or tolerance of violation of restrictions. This tolerance of violation of the restrictions is expressed mathematically as follows:

$$\lim_{\#Experiment \to \infty} \left(\frac{\#Event\ \boldsymbol{A}}{\#Experiment} \right) = Pr\{\boldsymbol{A}\} \geq \alpha \tag{3.1}$$

Where $\boldsymbol{A}$ is the study event, in this context a operational restriction, $\alpha \in [0, 1]$ is the probability level. The Equation (3.1) is known as Chance Constraint, its interpretation [78] is that if you perform a number "n" of times the experiment associated with event $\boldsymbol{A}$, and this "n" is large enough, then event $\boldsymbol{A}$ is expected to occur a number "$\alpha \times n$" of times, at least.

3.1 Chance Constrained OCP

The mathematical formulation of chance constrained OCP [79] is as follows:

$$J = \min_{\boldsymbol{u}} \{\gamma_1 E\left[J_1(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\xi})\right] + \gamma_2 J_2(\boldsymbol{u})\} \tag{3.2a}$$

subject to

$$\boldsymbol{G}(\dot{\boldsymbol{x}}(t), \boldsymbol{x}(t), \boldsymbol{u}, \boldsymbol{\xi}, t) = 0 \tag{3.2b}$$

$$Pr\left\{x_i^{min} \leq x_i(t) \leq x_i^{max}\right\} \geq \alpha_i;\ i = 1, 2, 3, \ldots, n \tag{3.2c}$$

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{3.2d}$$

$$u_j^{min} \leq u_j(t) \leq u_j^{max};\ j = 1, 2, 3, \ldots, m \tag{3.2e}$$

$$t_0 \leq t \leq t_f \tag{3.2f}$$

From Equation (3.2):

- $\boldsymbol{x}$: are the State variables. $\boldsymbol{x}$ is related to $\boldsymbol{\xi}$ through Equation (3.2b), so it is expected that they have a behavior similar to that of the Stochastic variables.
- $\boldsymbol{\xi} \in \chi \subset \Re^q$: are the Stochastic variables, they may be dependent or independent of time, it is normally assumed that the Stochastic variables have a known distribution.
- The two functions that are part of the objective function J are J_1 and J_2, which represent the error of the tracking to the designed path, the fuel consumption, respectively.
- $\boldsymbol{G} : \Re^n \times \Re^n \times \Re^m \times \Re^q \to \Re^n$: defines the dynamic model equation.
- $Pr\{\bullet\},\ E[\bullet]$: they are operators that represent the probability, the average or expected value, respectively.
- $\gamma_1,\ \gamma_2 \geq 0$: are weighting factors.

In this work of thesis will be taken to control variables $\boldsymbol{u}$ as deterministic, this is what is normally done in the state of the art.
There are two ways in which chance constraints can be expressed: Single Chance Constraint (SCC) and Joint Chance Constraint (JCC). Here is a brief explanation.

Single Chance Constraint

The Equation (3.2c) equation is an example of SCCs because it represents each Chance Constraint separately:

$$\begin{aligned} Pr\{x_1^{min} \leq x_1(t) \leq x_1^{max}\} &\geq \alpha_1 \\ Pr\{x_2^{min} \leq x_2(t) \leq x_2^{max}\} &\geq \alpha_2 \\ &\vdots \\ Pr\{x_n^{min} \leq x_n(t) \leq x_n^{max}\} &\geq \alpha_n \end{aligned}$$

The interpretation is that in each Chance Constraint a probability level is established to keep the corresponding State variable within a lower and upper limit.

Joint Chance Constraint

Unlike the SCCs, the JCCs groups many chance constraints establishing a same level of probability for the whole set, mathematically this is expressed:

$$Pr\left\{\begin{array}{c} x_1^{min} \leq x_1(t) \leq x_1^{max} \\ x_2^{min} \leq x_2(t) \leq x_2^{max} \\ \vdots \\ x_n^{min} \leq x_n(t) \leq x_n^{max} \end{array}\right\} \geq \alpha$$

In this thesis work will be used SCCs, the use of JCCs is usually more complex, and will be left as future work.
Continuing with the exposure of the chance constraints, below are additional notes about them.
From Probability Theory [78], it is true that:

$$Pr\{A\} + Pr\{A^c\} = 1;\ A\text{: is an event} \tag{3.4}$$

Using Equation (3.4) in Equation (3.2c), it is obtain the following:

$$Pr\{x_i > x_i^{max} \ \vee \ x_i < x_i^{min}\} \leq 1 - \alpha_i;\ i = 1, 2, 3 \ldots n \tag{3.5}$$

The Equation (3.5) is interpreted as the risk of states exceeding the upper and lower limits, the maximum risk is $1 - \alpha_i,\ i = 1, 2, \ldots, n$, for each state.
The property of convexity in the Chance Constraint is dependent on the function that defines them, also of the Probability Density Function (PDF) of the stochastic variables.

Nonlinear chance constraints

Mathematically a Nonlinear Chance Constraint is expressed in generic form as follows:

$$Pr\{\text{Lower bound} \leq c(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\xi}) \leq \text{Upper bound}\} \geq \alpha$$

A simple example of a state, control, and stochastic variable is: $c(x, u, \xi) = x^3 \sin(u)\xi$.
Nonlinear chance constraints are very difficult to work with.
It is not common for them to have the convexity property.
It is also rare a direct deterministic representation, like the one given in the previous example.

3.2 Chance Constrained NLP

After formulating the chance constrained OCP, as in the deterministic case.
This will be solved by using Direct Methods.
Therefore the infinite OCP is transformed to a finite NLP that has chance constraints.
Henceforth the latter will be called the chance constrained NLP.

The evaluation of the chance constraints is not an easy task, Approximation Methods will be used which will be explained in detail in the following section, to approximate the SCCs of chance constrained NLP.
In this way the Stochastic Optimization Problem becomes a deterministic one. The quasi-sequential method explained in Section 2.3.1 is applied to the obtained deterministic NLP.
Finally the obtained reduced NLP , will be solved by IpOpt.
Mathematically the chance constrained NLP is the following expression.

$$J = \min_{\boldsymbol{X},\boldsymbol{u}} \left\{ \gamma_1 E\left[f_1(\boldsymbol{x}^i, \boldsymbol{u}, \boldsymbol{\xi}) \right] + \gamma_2 f_2(\boldsymbol{u}) \right\}; \tag{3.6a}$$

$i = 1, 2, 3, \ldots, M;$ M is the number of samples,

$$\boldsymbol{u} = [\boldsymbol{u}_0, \boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_{N-1}],\ \boldsymbol{u}_j = \begin{bmatrix} u_{j,1} \\ u_{j,2} \\ \vdots \\ u_{j,m} \end{bmatrix},$$

where the second index after "j" indicates the number of Control variable, it is obvious that the total number of Control variables is "m",

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}^1 \\ \boldsymbol{x}^2 \\ \vdots \\ \boldsymbol{x}^M \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_1^1 & \boldsymbol{x}_2^1 & \boldsymbol{x}_3^1 & \ldots & \boldsymbol{x}_N^1 \\ \boldsymbol{x}_1^2 & \boldsymbol{x}_2^2 & \boldsymbol{x}_3^2 & \ldots & \boldsymbol{x}_N^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x}_1^M & \boldsymbol{x}_2^M & \boldsymbol{x}_3^M & \ldots & \boldsymbol{x}_N^M \end{bmatrix},\ \boldsymbol{x}_j^i = \begin{bmatrix} x_{j,1}^i \\ x_{j,2}^i \\ \vdots \\ x_{j,n}^i \end{bmatrix},$$

where the second index after "j" indicates the State variable number, the total of State variables is "n".

Subject to

$$\boldsymbol{x}_{j+1}^i = \boldsymbol{x}_j^i + \boldsymbol{F}(\boldsymbol{x}_j^i, \boldsymbol{u}_j, \boldsymbol{\xi});\ j = 0, 1, 2, \ldots, (N-1);\ i = 1, 2, 3, \ldots, M; \tag{3.6b}$$

the above discretization method for example can be Runge-Kutta 4th Order,

$$\boldsymbol{X}(t_0) = \boldsymbol{X}_0; \tag{3.6c}$$

$$Pr\{(x_{j,k})^{min} \leq x_{j,k}(\boldsymbol{u}, \boldsymbol{\xi}) \leq (x_{j,k})^{max}\} \geq \alpha_{j,k};\ j = 1, 2, \ldots, N;\ k = 1, 2, \ldots, n; \tag{3.6d}$$

where "j" indicates the time, "k" the state variable number, this expression does not depend on the Sample number.

$$\boldsymbol{u}_j \in \boldsymbol{U} \subset \Re^m;\ j = 0, 1, 2, \ldots (N-1) \tag{3.6e}$$

Below is a brief explanation of the terms of Equation (3.6):

- γ_1, γ_2: are weighting factors.

- $E[\bullet]$: is the expected value.
- $f_1,\ f_2$: are the discretized functions corresponding to the tracking error of the designed path, the fuel consumption, respectively.
- $\boldsymbol{\xi} \in \chi \subset \Re^q$: are the Stochastic variables, they may be dependent or independent of time.

3.3 Approximation Methods for chance constraints

As discussed in the previous section, approximation methods are used for SCCs, in this way a Deterministic Optimization Problem will be obtained. This section discusses some methods of approximation existing in the state of the art.

3.3.1 Sampling Approaches

3.3.1.1 Robust Optimization Techniques [80]

This method aims at minimizing the worst-case. The following is a generic mathematical formulation:

$$J = \min_{\boldsymbol{x}} \{E[f(\boldsymbol{x}, \boldsymbol{\xi})]\} \tag{3.7a}$$

subject to

$$\boldsymbol{c}(\boldsymbol{x}, \boldsymbol{\xi}) \leq \mathbf{0};\ \boldsymbol{\xi} \in \Omega \tag{3.7b}$$

$$\boldsymbol{x} \in \chi \tag{3.7c}$$

Then, it is necessary to generate a sequence of vectors of stochastic variables $\{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_N\}$; $\boldsymbol{\xi}_k \in \Omega$, the method looks for the constraints to be fulfilled as much as possible for each element of the sequence shown above. The sequence is generated by the Quasi Monte-Carlo (QSM) [81] algorithm.
With this method it is not necessary to calculate integrals, the convexity is preserved in the mathematical expressions of the deterministic NLP, the latter is simple to implement and solve. On the other hand, it has drawbacks such as that the deterministic NLP solution may not be feasible for the chance constrained NLP; in order to increase the reliability, a greater number of elements in the sequence of stochastic variables is necessary, which increases the computational effort.

3.3.1.2 SAA [82]

In this method the following function is first defined:

$$\boldsymbol{I}(c(x, \xi)) := \begin{cases} 0, \text{ if } c(x, \xi) > 0 \\ 1, \text{ if } c(x, \xi) \leq 0 \end{cases}$$

The next step is to generate sequences of stochastic variables ξ of low discrepancy such as the SOBOL [83] sequence , with these, chance constraints are approximated as follows.

$$Pr\{c(x,\xi) \leq 0\} \geq \alpha \equiv \frac{1}{N}\sum_{j=1}^{N} \boldsymbol{I}(c(x,\xi^j)) \geq \alpha$$

The advantages of using this method are that it is avoided to calculate multidimensional integrals, the mathematical expressions of the chance constrained NLP preserve their convexity even after the method has been applied. On the other hand, the use of this method leads to the resolution of a non-smooth deterministic NLP, the obtained solution will be feasible when "N" tends to infinity or is a very large number.

3.3.2 Back Mapping Method [84]

This method looks for a monotonic relation between $z = c(x, \boldsymbol{\xi})$ and a stochastic variable ξ_i, for it has to verify theoretically [85] or experimentally that there exists a Real function φ in which it is fulfilled:

$$z = \varphi_x(\xi_i),\ \exists x \in \chi, \tag{3.8}$$

where $\varphi_x(\bullet)$ is strictly increasing or decreasing, then applying the inverse of φ_x to two sides of Equation (3.8), it will be obtained:

$$\begin{aligned} \xi_i &= \varphi_x^{-1}(z) \\ \xi_i \uparrow z \longrightarrow Pr\{c(x,\boldsymbol{\xi}) \leq 0\} &= Pr\{\xi_i \leq \varphi_x^{-1}(0)\} \\ \xi_i \downarrow z \longrightarrow Pr\{c(x,\boldsymbol{\xi}) \leq 0\} &= Pr\{\xi_i \geq \varphi_x^{-1}(0)\} \end{aligned}$$

Then in a chance constrained Optimization Problem, if $\xi_i \uparrow z$, chance constraints can be calculated as follows:

$$Pr\{\xi_i \leq \varphi_x^{-1}(0)\} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\cdots\int_{-\infty}^{\varphi_x^{-1}(0)} \rho(\xi_1,\xi_2,\ldots)d\xi_1 d\xi_2 \ldots\ ,$$

the gradients of chance constraints are mathematically expressed as follows:

$$\nabla Pr\{\xi_i \leq \varphi_x^{-1}(0)\} = \underbrace{\int_{-\infty}^{+\infty}\cdots\int_{-\infty}^{+\infty}}_{(q-1),\,\boldsymbol{\xi}\in\Re^q} \nabla_x\varphi_x^{-1}(0)\rho(\xi_1,\xi_2,\ldots)d\xi_1 d\xi_2 \ldots$$

This method is used if it is easy to find the function φ, the chance constraints are expressed directly, the method has the disadvantage that the function φ does not always exist, or that it actually exists but it is extremely difficult to prove its existence.

3.3.3 Analytic Approximation Strategies [79, 86]

To understand the operation of this kind of methods, it will be started with the following Chance Constraint:

$$Pr\{c(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi}) \leq 0\} \geq \alpha \equiv Pr\{c(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi}) > 0\} \leq (1-\alpha)$$

Then the following function is defined:

$$h(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi}) := \begin{cases} 0, \text{ if } c(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi}) < 0 \\ 1, \text{ if } c(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi}) \geq 0, \end{cases}$$

then it is true that $Pr\{c(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi}) > 0\} = E[h(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi})]$, the expected value approximates the Chance Constraint exactly when the experiment associated with event $c(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi}) > 0$ is performed a sufficiently large number of times. The disadvantage of approaching chance constraints with the function $h(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi})$ is that the latter is not continuous, which makes it not suitable for computational calculations.
The Analytic Approximation Approaches uses instead of the $h(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi})$ function, the $\psi(\tau,\boldsymbol{x},\boldsymbol{u})$ function, which depends on the parameter $\tau > 0$, is continuous and possibly smooth.

$$E[h(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi})] \leq \psi(\tau,\boldsymbol{x},\boldsymbol{u}), \forall \tau > 0, \tag{3.11}$$

A suitable $\psi(\tau,\boldsymbol{x},\boldsymbol{u})$ function will be one in which Equation (3.12) is satisfied.

$$\lim_{\tau \to 0^+} \psi(\tau,\boldsymbol{x},\boldsymbol{u}) = E[h(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi})] \tag{3.12}$$

Define $\psi(\tau,\boldsymbol{x},\boldsymbol{u})$ as:

$$\psi(\tau,\boldsymbol{x},\boldsymbol{u}) := E[\Theta(\tau, c(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi}))],$$

where Θ is a function such that $\Theta : \Re^+ \times \Re \to \Re$. In the state of the art are different proposals for Θ, here are some of them:

- $\Theta(\tau, c(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi})) = \Theta(\tau, s) = \exp(\tau \times s),\ \tau > 0,$ [87]
- $\Theta(\tau, c(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi})) = \Theta(\tau, s) = \exp(\tau^{-1} \times s),\ \tau > 0,$ [88]
- $\Theta(\tau, c(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi})) = \Theta(\tau, s) = \tau + \left(\frac{1}{1-\alpha}\right) \max\{s-\tau, 0\},\ \tau > 0,$ [89]

New Analytic Approximation Strategies [79, 86]

These new analytical approximation methods are those that are used in this work of thesis.

Inner Approximation

$$\Theta(\tau, c(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi})) = \Theta(\tau, s) = \frac{1 + m_1\tau}{1 + m_2\tau \exp\left(\frac{-s}{\tau}\right)}, \tag{3.13}$$

where $m_1 \geq m_2 > 0,\ \tau \in < 0, 1 >$, Equation (3.13) gives a smooth approximation to the unit step function:

$$u_{step}(s) = \begin{cases} 0, & s < 0 \\ 1, & s \geq 0 \end{cases}$$

A chance constrained NLP has a Feasible Set "$\boldsymbol{P}$", if the chance constraints are approximated by this method, it will be obtained a deterministic $\text{NLP}_{\tau}^{\text{Inner}}$ whose Feasible Set approaches "$\boldsymbol{P}$" when $\tau \to 0^+$, this idea is illustrated in Figure 3.1. The Inner Approximation Method is usually accompanied by another method known as Outer Approximation, for corroboration. A Chance Constraint approximated by this method would be expressed as follows:

$$\lim_{\tau \to 0^+} \psi(\tau, \boldsymbol{x}, \boldsymbol{u}) = \lim_{\tau \to 0^+} E[\Theta(\tau, c(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\xi}))] = E[h(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\xi})] \leq (1 - \alpha) \tag{3.14}$$

The Inner Approximation has the advantage that the deterministic $\text{NLP}_{\tau}^{\text{Inner}}$ solution will always be a point in the chance constrained NLP Feasible Set, in contrast, the chance constrained NLP mathematical structures, it may be the case that they lose the property of convexity, obviously those structures that had said property before the application of the method.

Outer Approximation

The following function is defined:

$$\varphi(\tau, \boldsymbol{x}, \boldsymbol{u}) := E[\Theta(\tau, c(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\xi}))],$$

where Θ is the next function:

$$\Theta(\tau, c(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\xi})) = \Theta(\tau, s) = \frac{1 + m_1 \tau}{1 + m_2 \tau \exp\left(\frac{s}{\tau}\right)}, \tag{3.15}$$

in addition the following properties are fulfilled, $\varphi(\tau, \boldsymbol{x}, \boldsymbol{u}) \geq Pr\{c(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\xi}) \leq 0\}$, $\forall \tau \in < 0, 1 >$; φ is not a decreasing function; and finally $\lim_{\tau \to 0^+}[\varphi(\tau, \boldsymbol{x}, \boldsymbol{u})] = Pr\{c(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\xi}) \leq 0\}$.

A chance constrained NLP has a Feasible Set "$\boldsymbol{P}$", if the chance constraints are approximated by this method, it will be obtained a deterministic $\text{NLP}_{\tau}^{\text{Outer}}$ whose Feasible Set approaches "$\boldsymbol{P}$" when $\tau \to 0^+$, this idea is illustrated in Figure 3.1.

A Chance Constraint approximated by this method would be expressed as follows:

$$\lim_{\tau \to 0^+} \varphi(\tau, \boldsymbol{x}, \boldsymbol{u}) = \lim_{\tau \to 0^+} E[\Theta(\tau, c(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\xi}))] = Pr\{c(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\xi}) \leq 0\} \geq \alpha \tag{3.16}$$

If the $\text{NLP}_{\tau}^{\text{Outer}}$, $\text{NLP}_{\tau}^{\text{Inner}}$, for $\tau \to 0^+$ is solved, the corresponding feasible sets would have the graphical behavior of Figure 3.1.

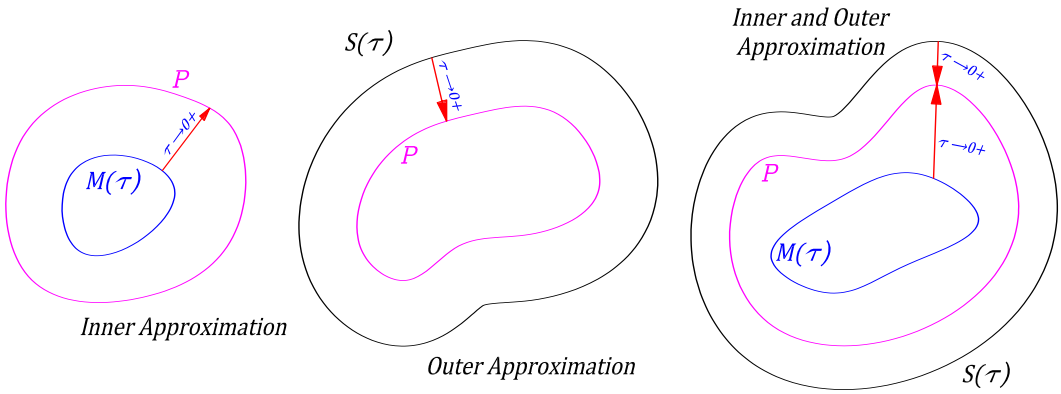


Figure 3.1 – Behaviour of the Feasible Sets of Inner and Outer Approximations.

From Figure 3.1:

- $M(\tau)$: Feasible Set of $\text{NLP}_{\tau}^{\text{Inner}}$.
- $S(\tau)$: Feasible Set of $\text{NLP}_{\tau}^{\text{Outer}}$.
- $\boldsymbol{P}$: Feasible Set of chance constrained NLP.

At the limit $\tau \to 0^{+}$, the Feasible Sets of $\text{NLP}_{\tau}^{\text{Inner}}$, $\text{NLP}_{\tau}^{\text{Outer}}$ are the same as the chance constrained NLP, i.e. "$\boldsymbol{P}$", then the Control Law $\boldsymbol{u}^{*}$ (or J^{*}) obtained for the Inner will be the same as for the Outer Approximation.
To conclude this chapter, in the study cases of Chapter 4, you will see chance constraints as follows:

$$\begin{aligned} Pr\{a \leq c_p(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi}) \leq b\} \geq \alpha \equiv \\ Pr\{\underbrace{k_2[\ln(\exp(k_1(-c_p(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi})+a))+\exp(k_1(c_p(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi})-b)))-k_3]}_{c(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi})} \leq 0\} \geq \alpha \equiv \\ Pr\{c(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi}) \leq 0\} \geq \alpha, \end{aligned} \tag{3.17}$$

then the last equivalence can be approximated by Inner or Outer Approximations.

Chapter 4

Study cases

In this chapter the characteristics, considerations, dynamic equations, design of trajectories, parameters for the simulations, of the study cases of this thesis will be exposed.

4.1 Trajectory tracking and landing on the asteroid Eros433

This study case is based on [16].
For a detailed development of this dynamic model see [17]. The dynamic model of the spacecraft expressed as a state equation, and with respect to the fixed coordinate system Σ_a in the center of mass of the asteroid Eros433 is:

$$\boldsymbol{x} = [x_1, x_2, x_3, x_4, x_5, x_6] = [x, y, z, \dot{x}, \dot{y}, \dot{z}]$$

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \\ \dot{x_5} \\ \dot{x_6} \end{bmatrix} = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ 2\omega_a x_5 + \omega_a^2 x_1 + U_{x_1} + u_{x_1} \\ -2\omega_a x_4 + \omega_a^2 x_2 + U_{x_2} + u_{x_2} \\ U_{x_3} + u_{x_3} \end{bmatrix} + \begin{bmatrix} \delta_{x_1} \\ \delta_{x_2} \\ \delta_{x_3} \\ \delta_{x_4} \\ \delta_{x_5} \\ \delta_{x_6} \end{bmatrix} \tag{4.1}$$

where,

- $\boldsymbol{\omega} = [0, 0, \omega_a]^\top$: is the rotational velocity of the Asteroid in the coordinate system Σ_a on $\frac{rad}{s}$, it is assumed to be constant, see Figure 4.1.
- $[U_{x_1}, U_{x_2}, U_{x_3}]$: are the components of the gradient of the gravitational potential U(see references [90], [17]), on $\frac{m}{s^2}$.
- $[x_1, x_2, x_3]$: are the position of the Spacecraft with respect the coordinate system Σ_a on the axes x, y, z respectively, on meters.
- $[x_4, x_5, x_6]$: are the velocities of the Spacecraft with respect the coordinate system Σ_a on the axes x, y, z respectively, on m/s.

- $[u_{x_1}, u_{x_2}, u_{x_3}]$: are the control accelerations with respect the coordinate system Σ_a, on m/s^2.
- $[\delta_{x_1}, \delta_{x_2}, \delta_{x_3}, \delta_{x_4}, \delta_{x_5}, \delta_{x_6}]$: errors of observation and perturbation accelerations on axes x, y, z (Σ_a) .

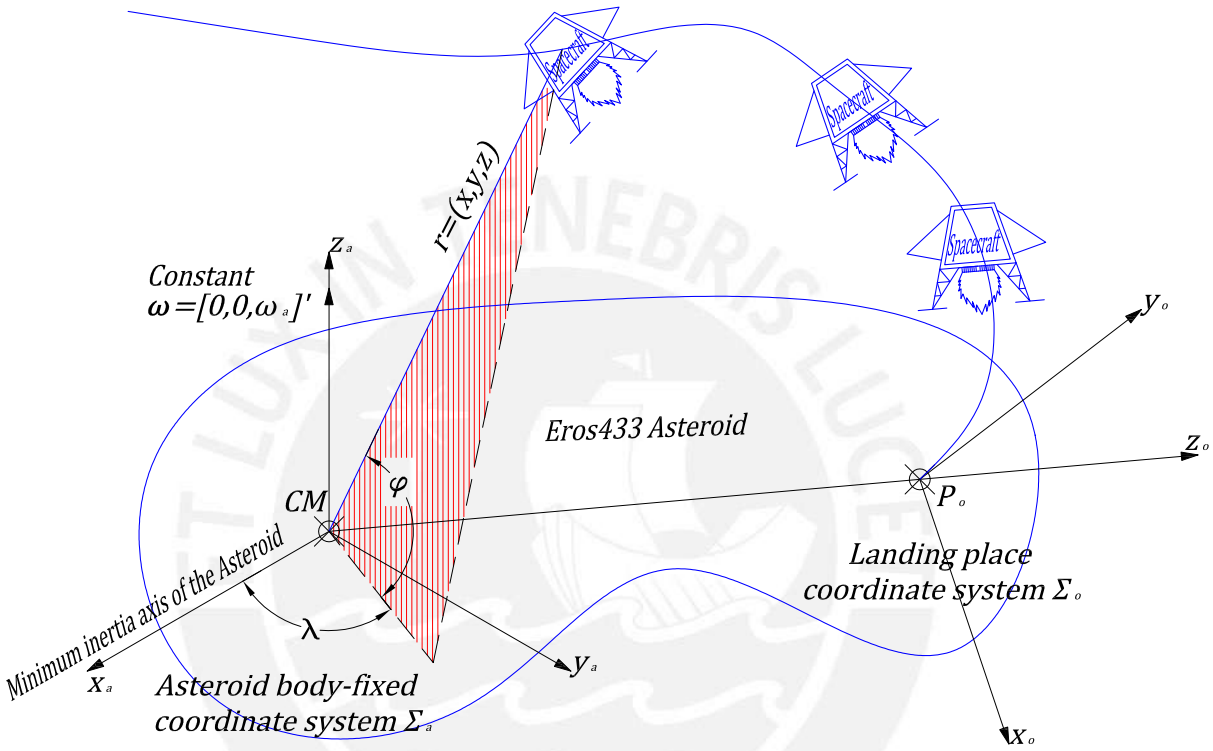


Figure 4.1 – Scheme of Case 1

In Figure 4.1 we can see the scheme from which the Equation (4.1) of state was derived. Mathematically the gravitational potential U of the asteroid is expressed as an expansion of a series of spherical harmonics [17], the expression is as follows:

$$U = \frac{GM}{r} \sum_{n=0}^{\infty} \sum_{m=0}^{n} \left(\frac{r_0}{r}\right)^n P_{nm}(\sin\varphi) \left[C_{nm}\cos(m\lambda) + S_{nm}\sin(m\lambda)\right] \tag{4.2}$$

where,

- GM: is the product of the mass of the asteroid(M) by the gravitational constant(G).
- n: is the degree.
- m: is the order.
- P_{nm}: is the fully Legendre polynomials.

- C_{nm}, S_{nm}: are the coefficients of the potential determined by the distribution of asteroid's mass inside itself.
- r_0: is the largest equatorial radius of the asteroid.
- φ, λ: are the latitude and longitude respectively of the Spacecraft with respect the coordinate system Σ_a(see Figure 4.1).
- r: is the distance to the Spacecraft from the origin of the coordinate system Σ_a, see Figure 4.1.

A very brief explanation will be given of how the asteroid is modeled to obtain the gravitational potential gradient [17].

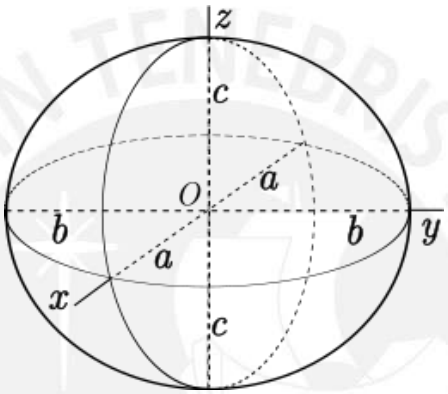


Figure 4.2 – Triaxial Ellipsoid [1]

The Asteroid was approximated using a triaxial ellipsoid with axes a, b, c (see Figure 4.2). The coefficients for this symmetric body are:

- $C_{nm} = 0$ for n or m odd.
- $S_{nm} = 0$ for all n or m.

For simplicity we used Equation (4.2) through 4 order, then the coefficients are equal to:

$$C_{20} = \frac{2c^2 - (a^2 + b^2)}{10r_0^2} \tag{4.3a}$$

$$C_{22} = \frac{a^2 - b^2}{20r_0^2} \tag{4.3b}$$

$$C_{40} = \frac{3}{140}\left[\frac{3(a^4 + b^4) + 8c^4 + 2a^2b^2 - 8(a^2 + b^2)c^2}{r_0^4}\right] \tag{4.3c}$$

$$C_{42} = \frac{(a^2 - b^2)(2c^2 - a^2 - b^2)}{280r_0^4} \tag{4.3d}$$

$$C_{44} = \frac{(a^2 - b^2)^2}{2240r_0^4} \tag{4.3e}$$

Therefore, Equation (4.2) is of fourth order with the mathematical expression:

$$U = \frac{GM}{r}\left[1 + \left(\frac{r_0}{r}\right)^2\left[\left(\frac{1}{2}\right)C_{20}(3\sin^2\varphi - 1) + 3C_{22}\cos^2\varphi\cos 2\lambda\right] + \left(\frac{r_0}{r}\right)^4\left[\left(\frac{1}{8}\right)C_{40}(35\sin^4\varphi - 30\sin^2\varphi + 3) + \left(\frac{15}{2}\right)C_{42}\cos^2\varphi\left(7\sin^2\varphi - 1\right)\cos 2\lambda + 105C_{44}\cos^2\varphi\cos 4\lambda\right] + O\left(r^{-5}\right)\right] \quad (4.4)$$

$$r = \sqrt{x^2 + y^2 + z^2} \quad (4.5a)$$

$$\varphi = \arctan\left(\frac{z}{\sqrt{x^2 + y^2}}\right) \quad (4.5b)$$

$$\lambda = \arctan\left(\frac{y}{x}\right) \quad (4.5c)$$

Replacing Equation (4.5) in Equation (4.4) and finding the gradient($\vec{\nabla} = \left[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right]^\top$) of the resulting function U.

The gradient of the gravitational potential $[U_{x_1}, U_{x_2}, U_{x_3}]$ was obtained.

The resulting mathematical expression will not be shown in this chapter. The complete mathematical expression is shown in the appendices, Section A..

4.1.1 Trajectory planning x_d, y_d, z_d

The strategy of planning the trajectory consists of having a time T to carry out the whole maneuver until landing.

The spacecraft will start from an initial position at $t = 0$, after a period of time $t = \zeta(\zeta < T)$, the spacecraft has to be found above the position, where it is intended to land, then proceed to the soft vertical landing (seen from the Σ_a system). We can see a general scheme of the above explained in Figure 4.3.

It will begin with the design of the trajectory in z_d:

Initial Position	At landing place
$z_d(0) = z_0$	$z_d(T) = z_n$
$\dot{z}_d(0) = \dot{z}_0$	$\dot{z}_d(T) = 0,$ to ensure soft landing

Table 4.1 – Design conditions for $z_d(t)$

It is sought that the trajectory in $z_d(t)$ is a polynomial of third degree (see [16], [17]):

$$z_d(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \quad (4.6)$$

Using the initial and final conditions of the Table 4.1:

$$z_d(t) = z_0 + \dot{z}_0 t + (3z_n - 3z_0 - 2\dot{z}_0 T)\left(\frac{t}{T}\right)^2 + (2z_0 + \dot{z}_0 T - 2z_n)\left(\frac{t}{T}\right)^3 \tag{4.7}$$

The next step is to design the trajectories for $x_d(t)$, $y_d(t)$ over time.

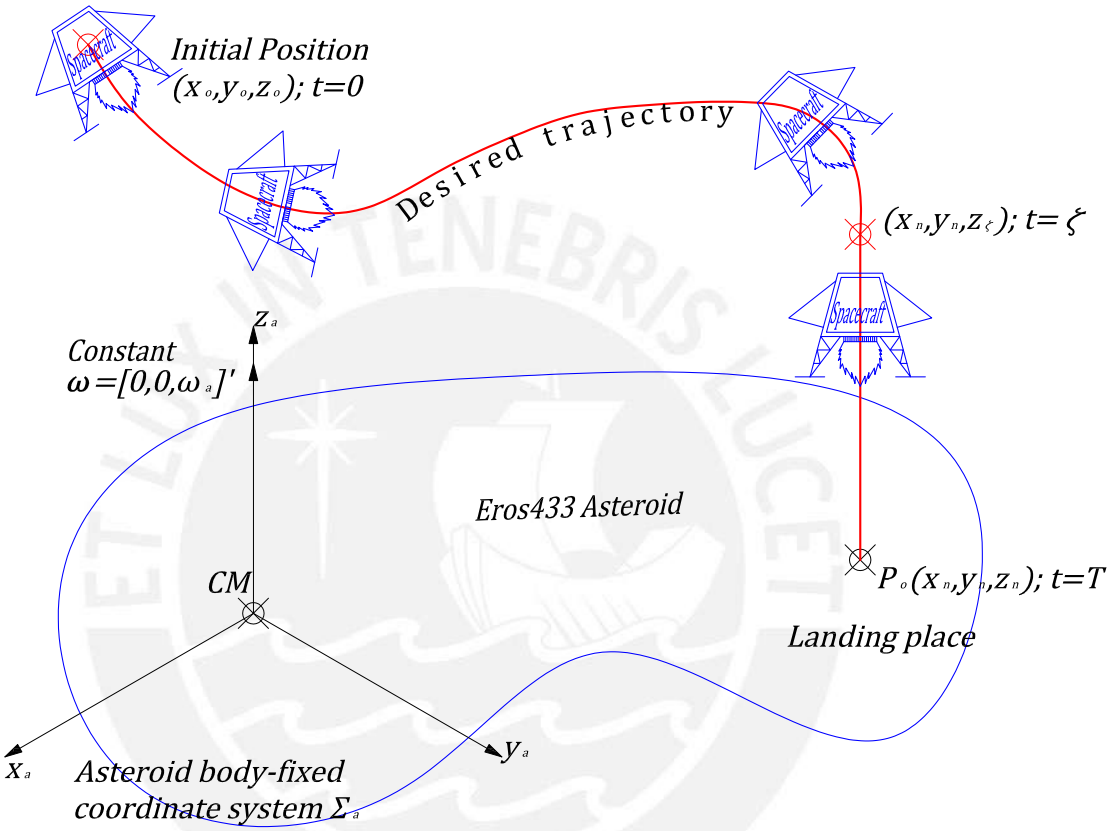


Figure 4.3 – Planning the trajectory of the Spacecraft

It must be considered that at time $t = \zeta (\zeta < T)$ the spacecraft has to be above the landing point (see Figure 4.3). The design is done this way for the following reasons [16]:

- Avoid the hazards of this type of operations.
- The accumulated scientific experience on this type of operations.

Initial Position	At landing place
$x_d(0) = x_0$	$x_d(T) = x_n$
$\dot{x}_d(0) = \dot{x}_0$	$\dot{x}_d(T) = 0$, to ensure soft landing

Table 4.2 – Design conditions for $x_d(t)$

Initial Position	At landing place
$y_d(0) = y_0$	$y_d(T) = y_n$
$\dot{y}_d(0) = \dot{y}_0$	$\dot{y}_d(T) = 0$, to ensure soft landing

Table 4.3 – Design conditions for $y_d(t)$

Using the initial and final conditions of the Tables 4.2 and 4.3, we obtain the following:

$$x_d(t) = \begin{cases} x_0 + \dot{x}_0 t + (3x_n - 3x_0 - 2\dot{x}_0\zeta)\left(\frac{t}{\zeta}\right)^2 + (2x_0 + \dot{x}_0\zeta - 2x_n)\left(\frac{t}{\zeta}\right)^3; & t \leq \zeta \\ x_n; \quad t > \zeta \end{cases} \tag{4.8}$$

$$y_d(t) = \begin{cases} y_0 + \dot{y}_0 t + (3y_n - 3y_0 - 2\dot{y}_0\zeta)\left(\frac{t}{\zeta}\right)^2 + (2y_0 + \dot{y}_0\zeta - 2y_n)\left(\frac{t}{\zeta}\right)^3; & t \leq \zeta \\ y_n; \quad t > \zeta \end{cases} \tag{4.9}$$

Using the Matlab software [91], together with the parameters of Table 4.4, it was obtained the designed position(Figures 4.5a and 4.5b), and velocity(Figure 4.5c) graphs. It is also shown the graph of the components of the gradient of the gravitational potential Figure 4.4.

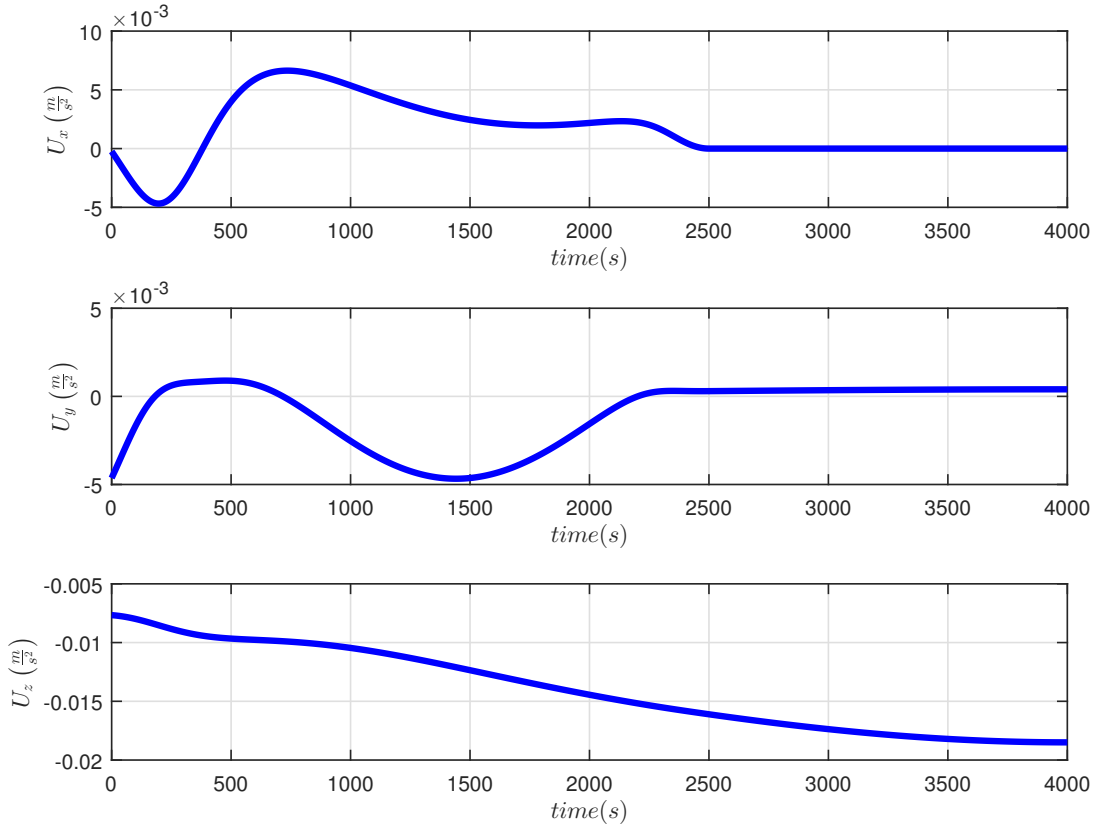
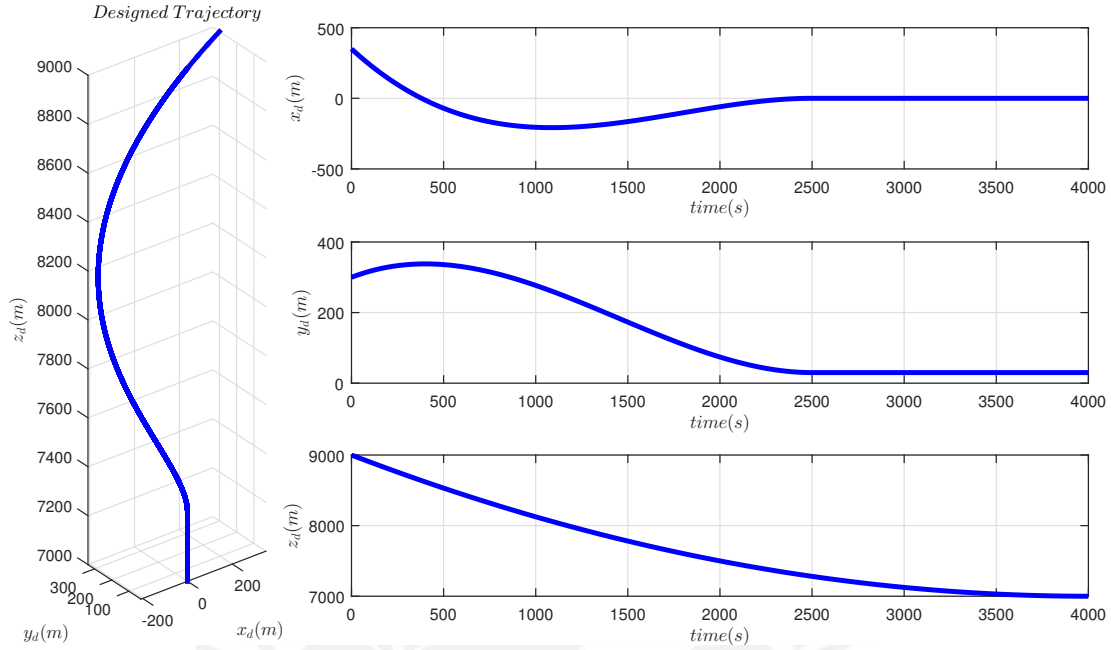
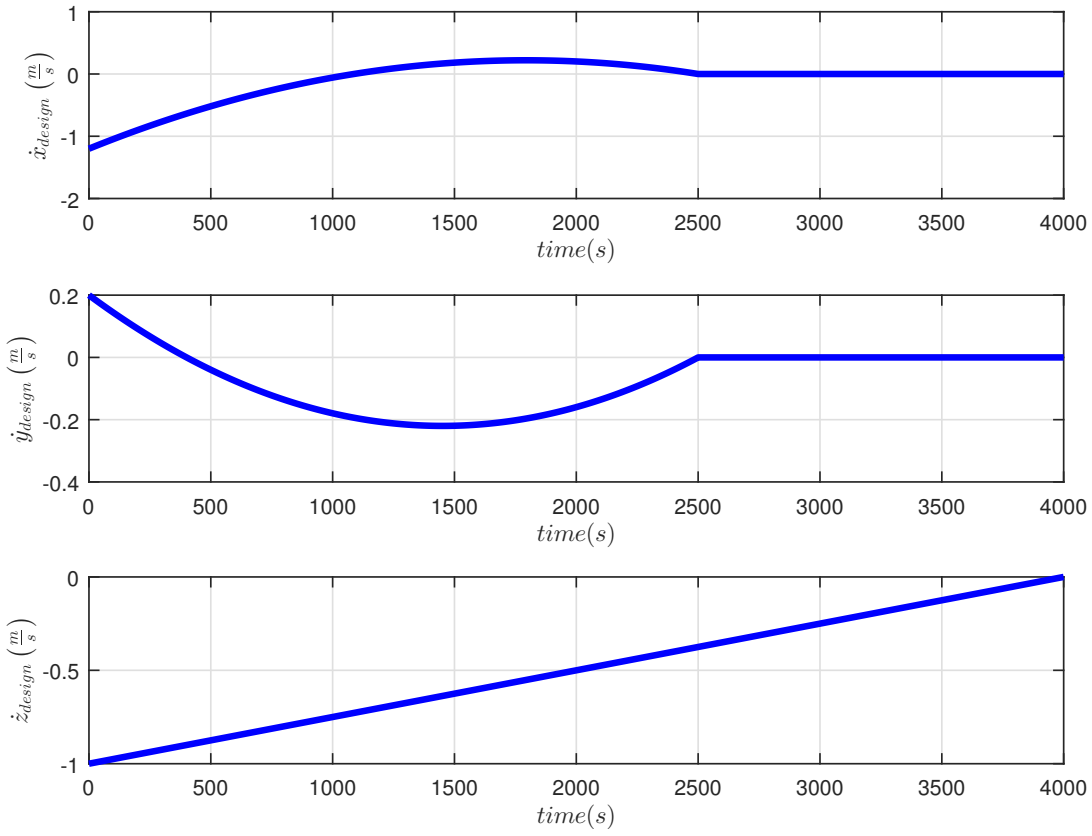


Figure 4.4 – Components of the gradient of the gravitational potential(U) along the time.



(a) Designed Trajectory for Case 1

(b) Components of the designed trajectory along the time.



(c) Components of designed speed over the time.

Figure 4.5 – Designed Trajectories for the Case 1

4.1.2 Parameters for the simulation

433 Eros Physics Parameters	
$GM\left(\frac{m^3}{s^2}\right)$	446300
$\omega_a\left(\frac{rad}{s}\right)$	0.000331182021
Reference Radius $r_o(m)$	16000
$[C_{20}, C_{22}, C_{40}, C_{42}, C_{44}]$	[3e−2,3.8e−3,4.1e−3,6.2e−3,5.1e−3]
Spacecraft Parameters	
$m(kg)$	100
Spacecraft Initial Parameters	
$[x_0, y_0, z_0]$ (meters)	[350,300,9000]
$[\dot{x}_0, \dot{y}_0, \dot{x}_0]\ (\frac{m}{s})$	[-1.2,0.2,-1]
Spacecraft Final Parameters	
$[x_n, y_n, z_n]$ (meters)	[0,30,7000]
$[\dot{x}_n, \dot{y}_n, \dot{x}_n]\ (\frac{m}{s})$	[0,0,0]
Time Simulation Parameters	
T (seconds)	4000
ζ (seconds)	2500
MPC Parameters	
$u_{x_i}^{min},\ i = 1,2,3 \quad \left(\frac{m}{s^2}\right)$	-50×10^{-3}
$u_{x_i}^{max},\ i = 1,2,3 \quad \left(\frac{m}{s^2}\right)$	50×10^{-3}

Table 4.4 – Simulation Parameters for Case 1 [92], [16].

4.2 Guidance law in three dimensions for a soft landing on a celestial body

This model is based on the papers presented by [93, 94]. The "Rendezvous" maneuver of a Spacecraft in a "Celestial Body", consists of three phases:

- Approach to the vicinity of the celestial body.
- A closer approach to the celestial body.
- Maneuvers performed on the surface of the celestial body.

The model presented in the papers of [93,94], takes into account the effect of the gravity of the celestial body, in addition to the drag force of the atmosphere. Effects such as solar radiation pressure, non-spherical gravitational effects, are treated as perturbations, the authors of these papers [93,94] model these perturbations as trigonometric functions, instead , the QSM algorithm will be used to simulate the effect of these perturbations on the model. We will use this model for the following:

- Ensure the landing of the spacecraft on the surface of the celestial body in a finite time.
- Ensure that the landing is smooth, this means that the speed an instant before touching the ground is zero or almost zero.
- Design of a suitable trajectory to carry out the tracking, see Figure 4.6.

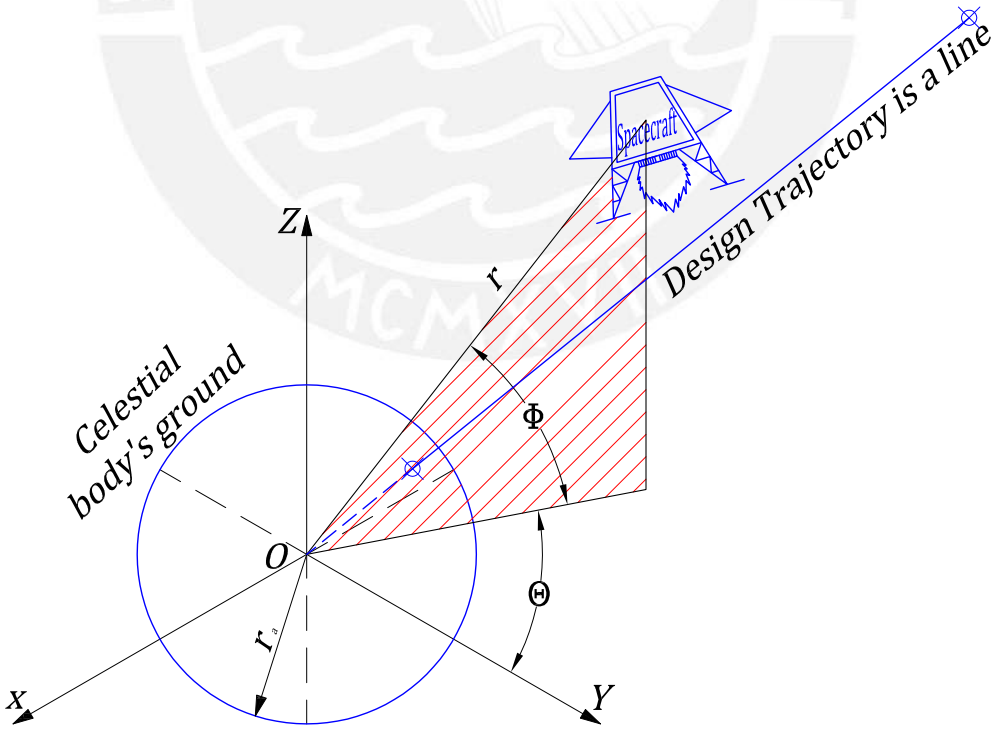


Figure 4.6 – Scheme for Case 2

In Figure 4.6 we have:

- r: Distance from the center of the celestial body to the spacecraft in km.
- Θ: Azimuth angle with respect to the fixed spherical coordinate system in the celestial body in rad .
- Φ: Pitch angle with respect to the spherical coordinate system fixed in the celestial body in rad.
- r_a: Celestial body radius in km.

The dynamic equation of the model expressed in state space has the following form:

$$\boldsymbol{x} = [x_1, x_2, x_3, x_4, x_5, x_6] = [r, \Theta, \Phi, \dot{r}, \dot{\Theta}, \dot{\Phi}]$$

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \end{bmatrix} = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \end{bmatrix} \tag{4.10a}$$

$$\begin{bmatrix} \dot{x_4} \\ \dot{x_5} \\ \dot{x_6} \end{bmatrix} = \begin{bmatrix} x_1 x_6^2 + x_1 x_5^2 \cos x_3{}^2 - \frac{\mu}{x_1^2} \\ 2x_5 x_6 \tan x_3 - 2\frac{x_4 x_5}{x_1} \\ -2\frac{x_4 x_6}{x_1} - x_5^2 \cos x_3 \sin x_3 \end{bmatrix}$$
$$- \beta \sqrt{x_4^2 + (x_1 x_5 \cos x_3)^2 + (x_1 x_6)^2} \times \left(w_1 {x_1}^{-2} + w_2 \exp(-k \times x_1) \right) \times \begin{bmatrix} x_4 \\ x_1 x_5 \cos x_3 \\ x_1 x_6 \end{bmatrix}$$
$$+ \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{x_1 \cos x_3} & 0 \\ 0 & 0 & \frac{1}{x_1} \end{bmatrix} \times \begin{bmatrix} u_r \\ u_\Theta \\ u_\Phi \end{bmatrix} + \begin{bmatrix} \delta_r \\ \delta_\Theta \\ \delta_\Phi \end{bmatrix} \tag{4.10b}$$

The terms of Equation (4.10) are:

- $\mu = G \times M_{Celestial\ Body}$; Standard gravitational parameter.
- $w_1,\ w_2$; are constants.
- $\beta > 0$; Drag coefficient.
- $[u_r, u_\Theta, u_\Phi]$; are the control inputs.
- $[\delta_u, \delta_\Theta, \delta_\Phi]$; are the coupling effects and external disturbances.

4.2.1 Designed trajectory

Then the designed path given in [93, 94] will be exposed.

$$r_d = r_a + (r_0 - r_a) \times \exp\left(\frac{-t}{10}\right) - \delta \tag{4.11a}$$

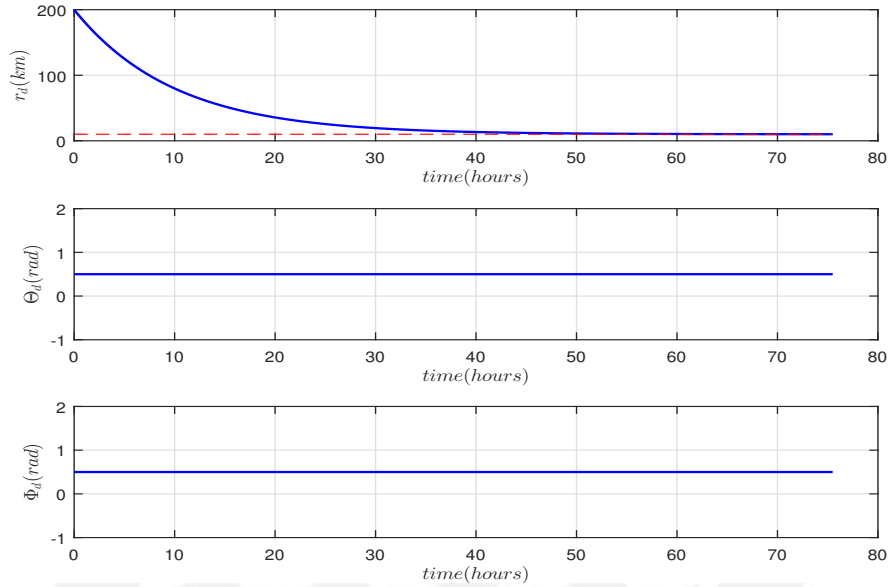
$$\Theta_d = \Theta_{final} \tag{4.11b}$$

$$\Phi_d = \Phi_{final} \tag{4.11c}$$

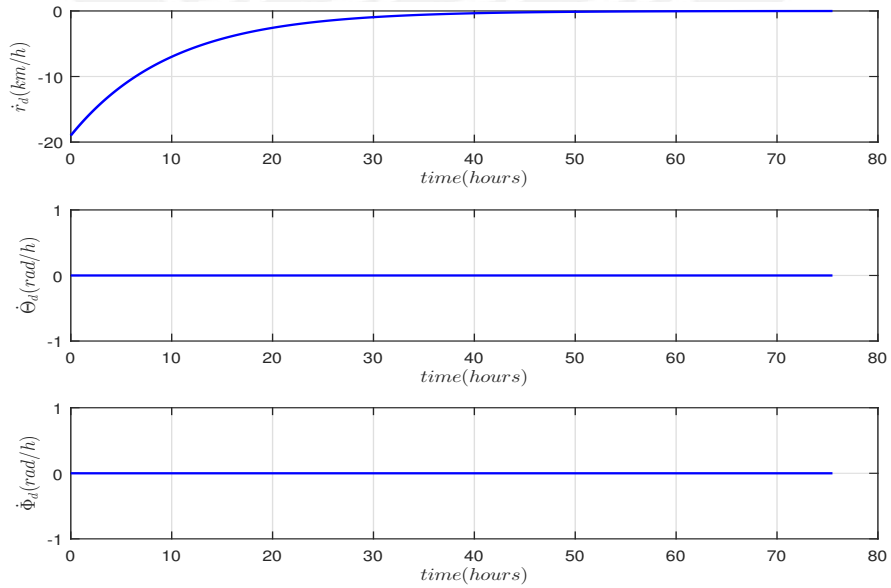
Where;

- r_a: Celestial body radius in km.
- δ: Parameter that assures to land in the celestial body in a finite time, in km.
- Θ_{final}, Φ_{final}: Desired final angles of Θ and Φ, in rad.

The components across time in r, Θ, Φ of the position and speed can be seen in Figures 4.7a and 4.7b respectively.



(a) Components of the trajectory designed in time.



(b) Components of speed designed in time.

Figure 4.7 – Position and velocity components designed for Case 2.

4.2.2 Parameters for the Simulation

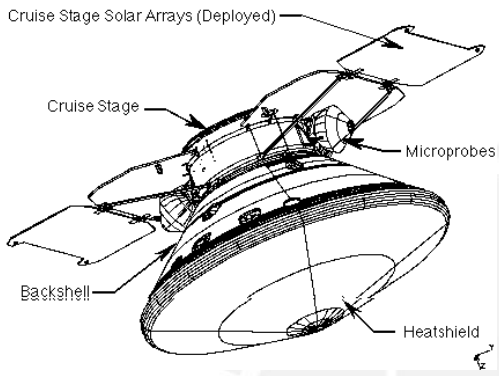
The parameters for the simulation are as follows:

Parameter	*Value*	*Units*	*Parameter*	*Value*	*Units*	*Parameter*	*Value*	*Units*
r_a	10	km	r_0	200	km	$\dot{r}_0$	-20	$\frac{km}{h}$
Θ_0	0.5	rad	Φ_0	0.5	rad	k	0.1	$-$
$\dot{\Phi}_0$	0.2	$\frac{rad}{h}$	μ	4000	$\frac{km^3}{h^2}$	Φ_{final}	0.5	rad
w_1	1	$-$	δ	0.1	km	$\dot{\Theta}_0$	0.2	$\frac{rad}{h}$
w_2	1	$-$	Θ_{final}	0.5	rad	β	0.02	$-$
u_r^{min}	-1000	$\left(\frac{km}{h^2}\right)$	u_r^{max}	1000	$\left(\frac{km}{h^2}\right)$	u_Θ^{min}	-1000	$\left(\frac{rad}{h^2}\right)$
u_Θ^{max}	1000	$\left(\frac{rad}{h^2}\right)$	u_Φ^{min}	-1000	$\left(\frac{rad}{h^2}\right)$	u_Φ^{max}	1000	$\left(\frac{rad}{h^2}\right)$
σ_{δ_r}	10^{-4}	$\left(\frac{km}{h^2}\right)$	σ_{δ_Θ}	10^{-4}	$\left(\frac{rad}{h^2}\right)$	σ_{δ_Φ}	10^{-4}	$\left(\frac{rad}{h^2}\right)$

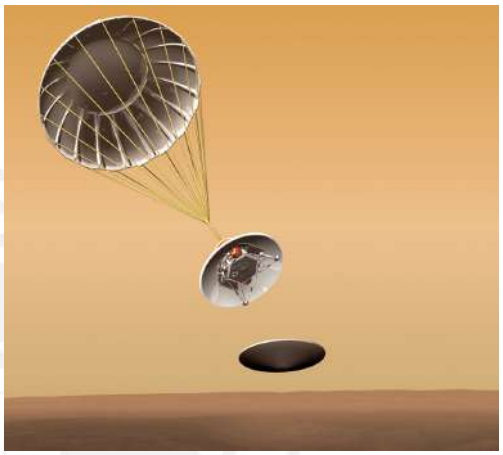
Table 4.5 – Simulation parameters for Case 2

4.3 Longitudinal model of lifting entry of a Mars Lander

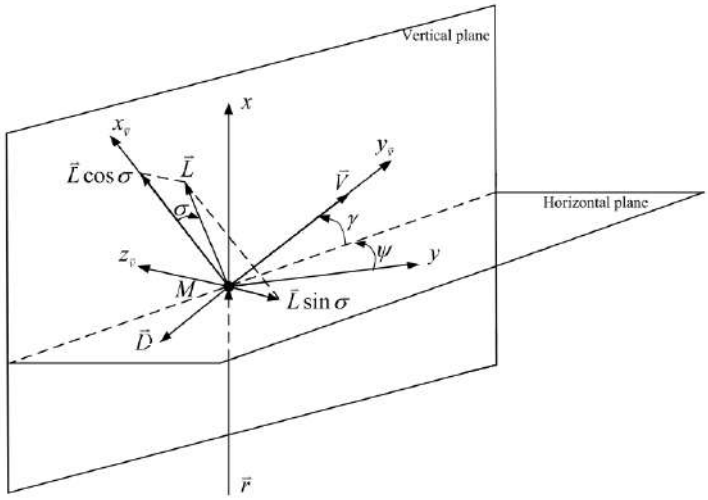
The case 3 model is based on papers by [95, 96].
With the model of case 3 we look for a control law to follow a designed path, to ensure an accurate landing, we have to consider in this model the nonlinear dynamics, uncertainties, and input saturation constraints.



(a) Mars Polar Lander Cruise Configuration [1]



(b) Mars Polar Lander parachute descent illustration [1]



(c) MSL type Vehicle Coordinate System [96]

Figure 4.8 – Mars Lander

Assumptions of this model:

- The atmosphere of Mars remains static.
- The physical effects of the Mars rotation will not be considered.

- The dynamic equation of the model was deduced in an MSL type vehicle coordinate system, see Figure 4.8c.

The dynamic equation of this model is as follows:

$$\dot{h} = v \sin\gamma + \xi_h \tag{4.12a}$$
$$\dot{v} = -D - g \sin\gamma + \xi_v \tag{4.12b}$$
$$\dot{\gamma} = \left(\frac{v}{r} - \frac{g}{v}\right)\cos\gamma + \frac{1}{v} L \cos\sigma + \xi_\gamma \tag{4.12c}$$
$$\dot{s} = v \cos\gamma \tag{4.12d}$$

In addition to Equation (4.12) we have the following algebraic equations.

$$g = \frac{\mu}{(h + r_{Mars})^2} \tag{4.13a}$$
$$L = \frac{1}{2}\rho v^2 \left(\frac{C_L S_r}{m}\right) \tag{4.13b}$$
$$D = \frac{1}{2}\rho v^2 \left(\frac{C_D S_r}{m}\right) \tag{4.13c}$$
$$\rho = \rho_0 \exp\left(-\left(\frac{r - r_{Mars}}{h_s}\right)\right) \tag{4.13d}$$
$$r = h + r_{Mars} \tag{4.13e}$$

In Equations (4.12) and (4.13) the following terms are:

- h: It is the height of the Mars Lander with respect to the Martian surface.
- v: It is the magnitude of Mars Lander's speed.
- γ: Is the flight path angle.
- s: Is the downrange.
- σ: Is the bank angle.
- g: Gravity equation of Mars.
- μ: Mars gravitational parameter.
- r_{Mars}: Is the radius of Mars.
- L: Lift acceleration.
- D: Drag acceleration.
- C_L: Lift coefficient.
- C_D: Drag coefficient.
- S_r: Vehicle reference surface Area.
- m: Vehicle mass.
- ρ: Atmospheric density.
- ρ_0: Density at Sea level.

- h_s: Constant scale height.
- $[\xi_h, \xi_v, \xi_\gamma]$: Stochastic variables.

Some of the above terms can be seen in Figure 4.8c.

4.3.1 Designed Trajectory

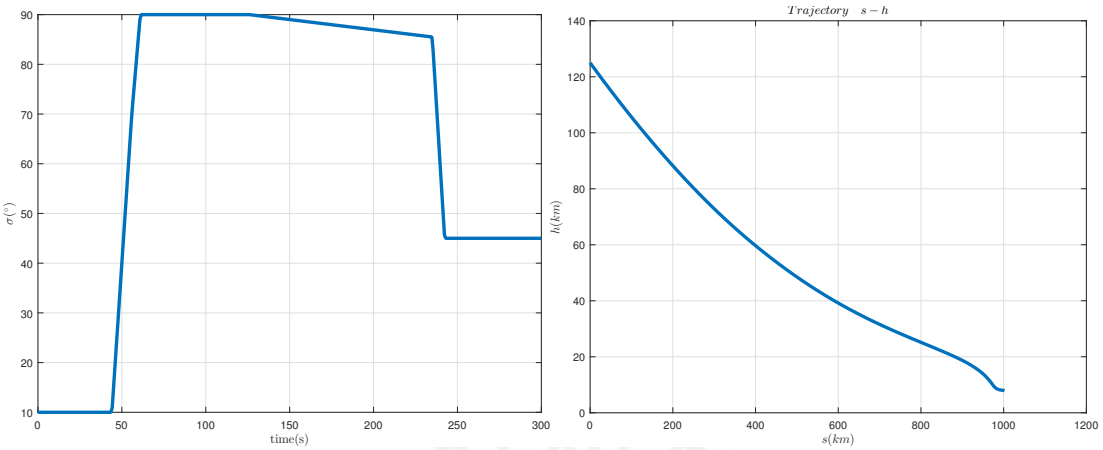
For the design of the trajectory that the Mars Lander must follow, one has to consider the initial states as well as the final states of the Lander, these states are seen in Table 4.6. The Mars Lander starts from its initial position, at a certain height from the Martian surface, at a certain speed, and has to arrive at a position described by the final states, at this point it deploys a parachute that it has incorporated, and begins to decelerate until it touches the Martian soil, we can see a scheme of the Lander in Figure 4.8a, as well as a scheme of descent with the parachute in Figure 4.8b. The methodology for the design of the trajectory consisted of the following:

- A graph has to be found as a function of time for σ, up to a time of 300 seconds.
- With this graph, we solve Equation (4.12) by Runge-Kutta [48, 49] of $4th$ order, with a constant step of 0.75 seconds, obtaining the graphs shown in Figure 4.9c.
- The graph of σ as a function of time must be such that it complies with the values of the initial and final states.

To find the appropriate σ plot as a function of time, was used as reference the σ plot used in paper [96], finally the σ plot obtained is shown in Figure 4.9a.
In Figure 4.9b we can see the trajectory designed in a vertical plane $s-h$, in Figure 4.9c the resulting designed graphs for $h,\ v,\ \gamma,\ s,\ \dot{h},\ \dot{s}$ are shown, the last two were obtained by deriving once with respect to time Equations (4.12a) and (4.12d), and replacing Equations (4.12b) and (4.12c).

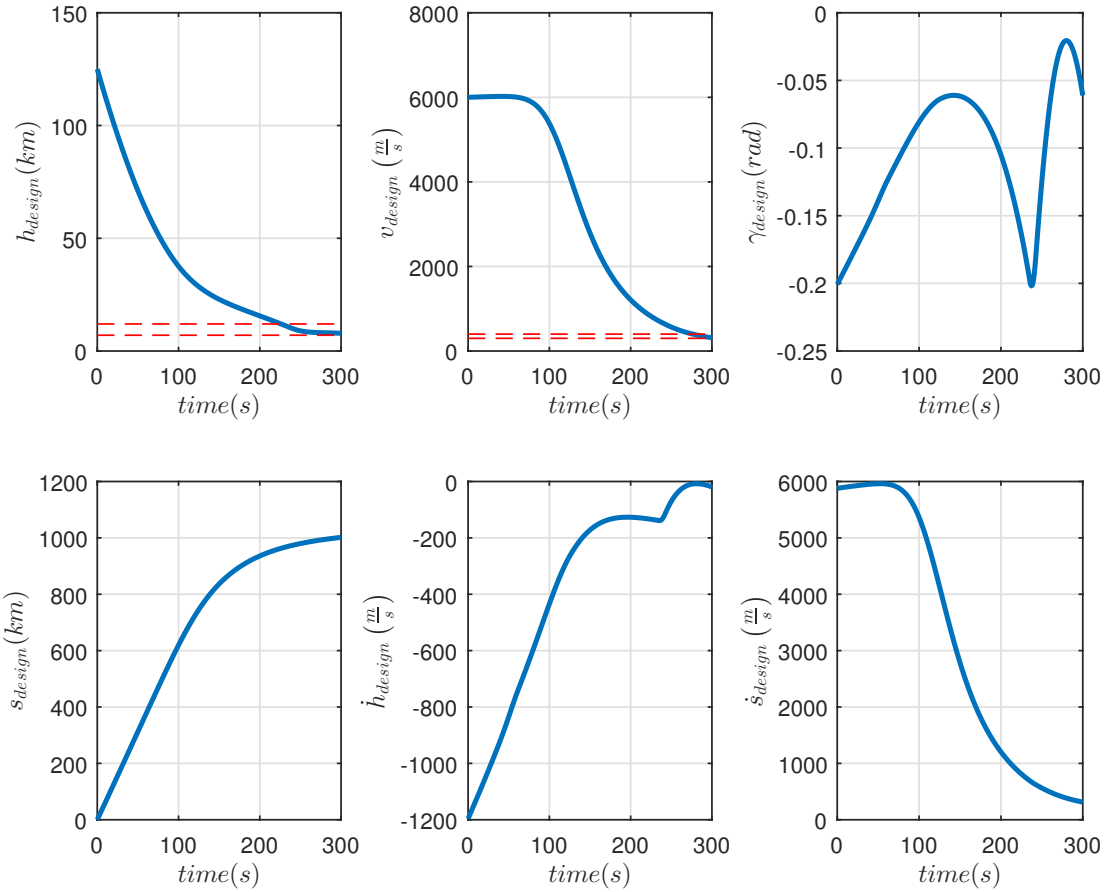
Initial States	*Final States*
$h = 125km$	$h \in [7, 12]km$
$v = 6000\frac{m}{s}$	$v \leq 400\frac{m}{s}$
$\gamma = -11.5°$	$\gamma = any$
$s = 0km$	$s = any$

Table 4.6 – Initial and final states for the trajectory design.



(a) Graph of σ vs time, with which the trajectories are designed.

(b) Trajectory designed in the $s-h$ plane that the Mars Lander has to follow.



(c) Designed Trajectories

Figure 4.9 – Designed Trajectories for the Case 3

4.3.2 Parameters for the Simulation

The parameters for the simulation are as follows:

Parameter	*Value*	*Units*	*Parameter*	*Value*	*Units*
r_{Mars}	3397	km	h_s	9353.5	m
μ	4.282837e13	$\frac{m^3}{s^2}$	ρ	0.0158	$\frac{kg}{m^3}$
m	2200	kg	S_r	12.8825	m^2
C_D	1.4499	$-$	C_L	1.7979	$-$
u_{min}	10	$\circ$	u_{max}	90	$\circ$
σ_{ξ_h}	0.25	$\frac{m}{s}$	σ_{ξ_v}	0.25	$\frac{m}{s^2}$
σ_{ξ_γ}	10^{-6}	$\frac{rad}{s}$	$-$	$-$	$-$

Table 4.7 – Simulation parameters for Case 3

Chapter 5

Deterministic and Stochastic MPC Formulation

In this chapter the deterministic and stochastic MPC will be formulated for the study cases of Chapter 4.

5.1 Deterministic MPC Formulation

The operation of the deterministic MPC is explained in the flowchart of Figure 5.1.
The deterministic MPC was formulated for each of the study cases, making use of this flowchart.
Some of the steps in the flowchart require additional precisions:

- The discretization method used to transform the deterministic NOCP to a deterministic NLP was the Runge-Kutta 4th order method.
- The deterministic OCP and NLP of each of the study cases is presented later in this section.
- The GNU Scientific Library [97] was used to implement the Newton solver, and the linear algebra solver in C++ [68].
- In order to find the first derivatives, the objective function and the constraints (equations + inequations), Casadi [98] software was used, as well as MATLAB [91]. The second derivative, that is, the Hessian, was approximated by the algorithm L-BFGS, which is part of the IpOpt software package
- IpOpt is an optimization software based on gradients, that is why it is necessary to calculate the first and second derivatives. For more detailed information on the IpOpt, see Section 2.3.3.
- The program of the deterministic MPC was implemented in C++, under the operating system Linux Ubuntu.

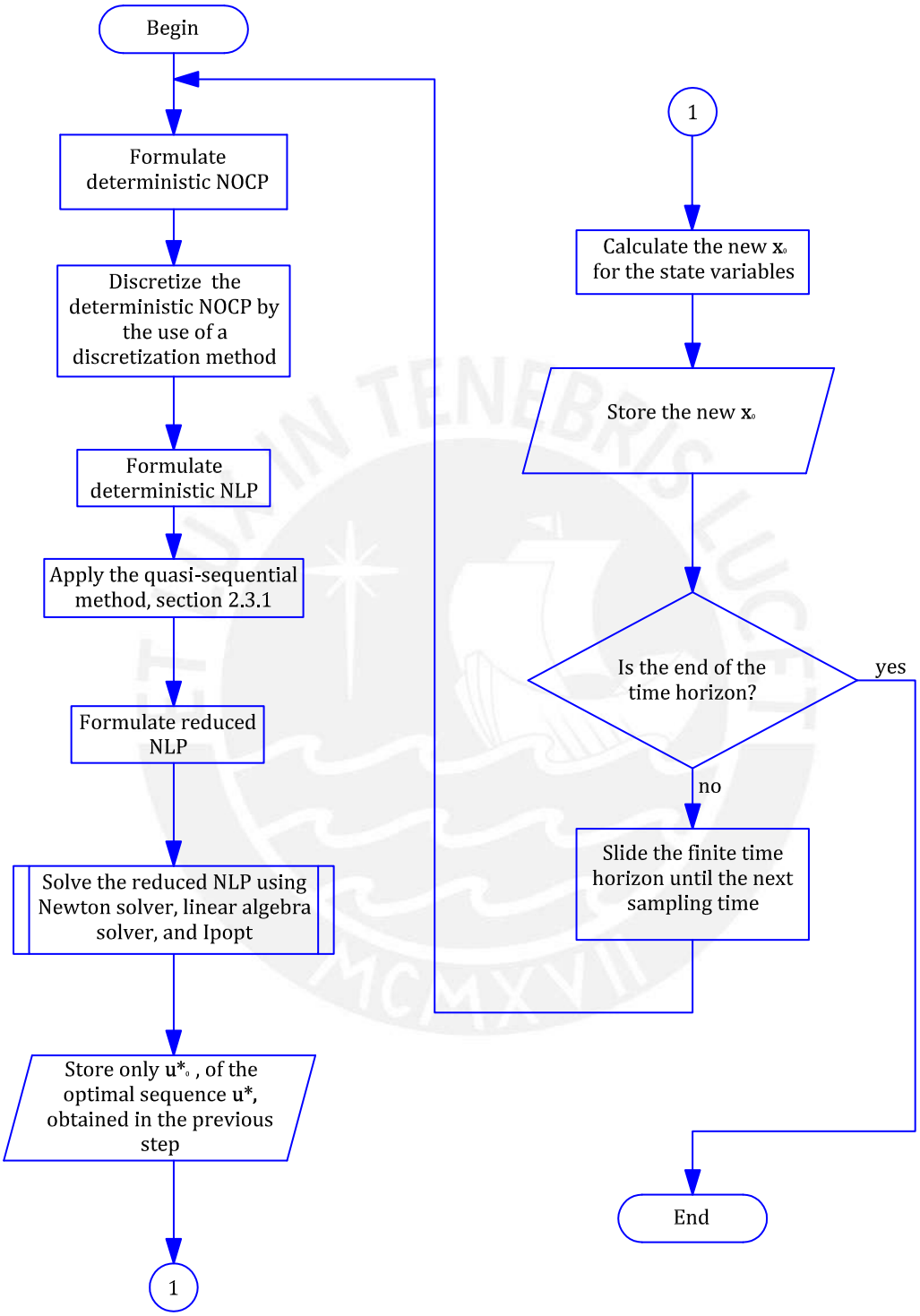


Figure 5.1 – Flowchart of the deterministic MPC.

5.1.1 Trajectory tracking and landing on the asteroid Eros433

OCP formulation

The deterministic OCP is expressed by the following equations:

$$[x_1, x_2, x_3, x_4, x_5, x_6] = [x, y, z, \dot{x}, \dot{y}, \dot{z}]$$

$$J = \min_{u_{x_1}, u_{x_2}, u_{x_3}} \left\{ \int_0^T \left((x_1(t) - x_d(t))^2 + (x_2(t) - y_d(t))^2 + (x_3(t) - z_d(t))^2 \right) dt + \int_0^T \left(u_{x_1}^2 + u_{x_2}^2 + u_{x_3}^2 \right) dt \right\} \quad (5.1a)$$

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \\ \dot{x_5} \\ \dot{x_6} \end{bmatrix} = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ 2\omega_a x_5 + \omega_a^2 x_1 + U_{x_1} + u_{x_1} \\ -2\omega_a x_4 + \omega_a^2 x_2 + U_{x_2} + u_{x_2} \\ U_{x_3} + u_{x_3} \end{bmatrix} \quad (5.1b)$$

$$0 \leq \left((x_1(t) - x_d(t))^2 + (x_2(t) - y_d(t))^2 + (x_3(t) - z_d(t))^2 \right) \leq 1m^2 \quad (5.1c)$$

$$u_{x_1}^{min} \leq u_{x_1} \leq u_{x_1}^{max} \quad (5.1d)$$

$$u_{x_2}^{min} \leq u_{x_2} \leq u_{x_2}^{max} \quad (5.1e)$$

$$u_{x_3}^{min} \leq u_{x_3} \leq u_{x_3}^{max} \quad (5.1f)$$

NLP formulation

The NLP obtained from the deterministic OCP is as follows:

$$J = \min_{(x_1^k, x_2^k, x_3^k, x_4^k, x_5^k, x_6^k, u_{x_1}^k, u_{x_2}^k, u_{x_3}^k)} \left\{ \sum_{k=0}^{N-1} a_k \Delta t \left[(x_1^k - x_d^k)^2 + (x_2^k - y_d^k)^2 + (x_3^k - z_d^k)^2 \right] + \sum_{k=0}^{N-1} b_k \Delta t \left({u_{x_1}^k}^2 + {u_{x_2}^k}^2 + {u_{x_3}^k}^2 \right) \right\} \quad (5.2a)$$

$a_k = 1;\ k = 0, 1, 2, 3 \ldots (N-1)$
$b_k = 2.5;\ k = 0, 1, 2, 3 \ldots (N-1)$

subject to

$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{F}^k(\boldsymbol{x}_k, \boldsymbol{u}_k, t_k)$; for some Runge-Kutta method(Euler,Heun,Runge Kutta 4th order)

Assuming it is the Runge-Kutta 4th order method, one would have:

$$\begin{aligned}
\boldsymbol{k}_1^k &= \boldsymbol{F}(\boldsymbol{x}_k, \boldsymbol{u}_k, t_k) \\
\boldsymbol{k}_2^k &= \Delta t \boldsymbol{F}(\boldsymbol{x}_k + \frac{\boldsymbol{k}_1^k}{2}, \boldsymbol{u}_k, t_k + \frac{\Delta t}{2}) \\
\boldsymbol{k}_3^k &= \Delta t \boldsymbol{F}(\boldsymbol{x}_k + \frac{\boldsymbol{k}_2^k}{2}, \boldsymbol{u}_k, t_k + \frac{\Delta t}{2}) \\
\boldsymbol{k}_4^k &= \Delta t \boldsymbol{F}(\boldsymbol{x}_k + \boldsymbol{k}_3^k, \boldsymbol{u}_k, t_k + \Delta t) \\
\boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \underbrace{\left(\frac{\boldsymbol{k}_1^k}{6} + \frac{\boldsymbol{k}_2^k}{3} + \frac{\boldsymbol{k}_3^k}{3} + \frac{\boldsymbol{k}_4^k}{6}\right)}_{\boldsymbol{F}^k(\boldsymbol{x}_k, \boldsymbol{u}_k, t_k)};\ k = 0, 1, 2 \ldots (N-1)
\end{aligned} \tag{5.2b}$$

Where $\boldsymbol{F}(\boldsymbol{x}_k, \boldsymbol{u}_k, t_k)$ is the right part of the Equation (5.1b).

The discretized trajectory constraints are:

$$0 \leq (x_1^k - x_d^k)^2 + (x_2^k - y_d^k)^2 + (x_3^k - z_d^k)^2 \leq 1m^2;\ k = 1, 2, 3, \ldots N \tag{5.2c}$$

The upper and lower limits for the discretized control variables are:

$$-50 \times 10^{-3} \left(\frac{m}{s^2}\right) \leq u_{x_i}^k \leq 50 \times 10^{-3} \left(\frac{m}{s^2}\right);\ i = 1, 2, 3;\ k = 0, 1, 2, \ldots (N-1) \tag{5.2d}$$

5.1.2 Guidance law in three dimensions for a soft landing on a celestial body

OCP formulation

The deterministic OCP is expressed in the following formulas:

$$[x_1, x_2, x_3, x_4, x_5, x_6] = [r, \Theta, \Phi, \dot{r}, \dot{\Theta}, \dot{\Phi}]$$

$$J = \min_{u_r, u_\Theta, u_\Phi} \left\{ \int_0^T \left((x_1 - r_d(t))^2 + (x_2 - \Theta_d(t))^2 + (x_3 - \Phi_d(t))^2 \right) dt + \int_0^T \left(u_r^2 + u_\Theta^2 + u_\Phi^2 \right) dt \right\} \quad (5.3a)$$

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \end{bmatrix} = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \end{bmatrix} \quad (5.3b)$$

$$\begin{bmatrix} \dot{x_4} \\ \dot{x_5} \\ \dot{x_6} \end{bmatrix} = \begin{bmatrix} x_1 x_6^2 + x_1 x_5^2 \cos x_3{}^2 - \frac{\mu}{x_1^2} \\ 2x_5 x_6 \tan x_3 - 2\frac{x_4 x_5}{x_1} \\ -2\frac{x_4 x_6}{x_1} - x_5^2 \cos x_3 \sin x_3 \end{bmatrix} - \beta \sqrt{x_4^2 + (x_1 x_5 \cos x_3)^2 + (x_1 x_6)^2} \times \left(w_1 {x_1}^{-2} + w_2 \exp(-k \times x_1) \right) \times \begin{bmatrix} x_4 \\ x_1 x_5 \cos x_3 \\ x_1 x_6 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{x_1 \cos x_3} & 0 \\ 0 & 0 & \frac{1}{x_1} \end{bmatrix} \times \begin{bmatrix} u_r \\ u_\Theta \\ u_\Phi \end{bmatrix} \quad (5.3c)$$

$$0 \leq \left((x_1 \cos x_3 \cos x_2 - r_d \cos \Phi_d \cos \Theta_d)^2 + (x_1 \cos x_3 \sin x_2 - r_d \cos \Phi_d \sin \Theta_d)^2 + (x_1 \sin x_3 - r_d \sin \Phi_d)^2 \right) \leq 25km^2 \quad (5.3d)$$

$$u_r^{min} \leq u_r \leq u_r^{max} \quad (5.3e)$$

$$u_\Theta^{min} \leq u_\Theta \leq u_\Theta^{max} \quad (5.3f)$$

$$u_\Phi^{min} \leq u_\Phi \leq u_\Phi^{max} \quad (5.3g)$$

NLP formulation

The NLP that is obtained from the deterministic OCP is expressed in the following formulas:

$$J = \min_{(x_1^k, x_2^k, x_3^k, x_4^k, x_5^k, x_6^k, u_r^k, u_\Theta^k, u_\Phi^k)} \left\{ \sum_{k=0}^{N-1} a_k \Delta t \left[(x_1^k - r_d^k)^2 + (x_2^k - \Theta_d^k)^2 + (x_3^k - \Phi_d^k)^2 \right] + \sum_{k=0}^{N-1} b_k \Delta t \left({u_r^k}^2 + {u_\Theta^k}^2 + {u_\Phi^k}^2 \right) \right\} \quad (5.4a)$$

$$a_k = 1;\ k = 0, 1, 2, 3 \ldots (N-1)$$
$$b_k = 3;\ k = 0, 1, 2, 3 \ldots (N-1)$$

subject to

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{F}^k(\boldsymbol{x}_k, \boldsymbol{u}_k, t_k);\ k = 0, 1, 2 \ldots (N-1) \quad (5.4b)$$

For a better understanding of Equation (5.4b) see Equation (5.2b).

The discretized trajectory constraints are:

$$0 \leq \Big((x_1^k \cos x_3^k \cos x_2^k - r_d^k \cos \Phi_d^k \cos \Theta_d^k)^2 + (x_1^k \cos x_3^k \sin x_2^k - r_d^k \cos \Phi_d^k \sin \Theta_d^k)^2 + (x_1^k \sin x_3^k - r_d^k \sin \Phi_d^k)^2 \Big) \leq 25km^2;\ k = 1, 2, 3, \ldots N \quad (5.4c)$$

The upper and lower limits for the discretized control variables are:

$$-1000 \left(\frac{km}{h^2} \right) \leq u_r^k \leq 1000 \left(\frac{km}{h^2} \right);\ k = 0, 1, 2, \ldots (N-1) \quad (5.4d)$$

$$-1000 \left(\frac{rad}{h^2} \right) \leq u_\Theta^k \leq 1000 \left(\frac{rad}{h^2} \right);\ k = 0, 1, 2, \ldots (N-1) \quad (5.4e)$$

$$-1000 \left(\frac{rad}{h^2} \right) \leq u_\Phi^k \leq 1000 \left(\frac{rad}{h^2} \right);\ k = 0, 1, 2, \ldots (N-1) \quad (5.4f)$$

5.1.3 Longitudinal model of lifting entry of a Mars Lander

Before formulating the OCP, a change of variable will be made as follows:

$$e_h = h(t) - h_{design}(t) \tag{5.5a}$$
$$e_s = s(t) - s_{design}(t) \tag{5.5b}$$

Replacing Equation (5.5) in Equations (4.12) and (4.13), and reformulating in the space of states, we have:

$$\boldsymbol{x} = [x_1, x_2, x_3, x_4] = [e_h, v, \gamma, e_s]; \quad u = \sigma$$
$$\dot{x}_1 = x_2 \sin x_3 - \dot{h}_{design}(t) + \xi_h \tag{5.6a}$$
$$\dot{x}_2 = -D - g \sin x_3 + \xi_v \tag{5.6b}$$
$$\dot{x}_3 = \left(\frac{x_2}{r} - \frac{g}{x_2}\right)\cos x_3 + \frac{1}{x_2} L \cos u + \xi_\gamma \tag{5.6c}$$
$$\dot{x}_4 = x_2 \cos x_3 - \dot{s}_{design}(t) \tag{5.6d}$$

And the algebraic equations are now:

$$g = \frac{\mu}{(x_1 + h_{design}(t) + r_{Mars})^2} \tag{5.7a}$$
$$L = \frac{1}{2}\rho {x_2}^2 \left(\frac{C_L S_r}{m}\right) \tag{5.7b}$$
$$D = \frac{1}{2}\rho {x_2}^2 \left(\frac{C_D S_r}{m}\right) \tag{5.7c}$$
$$\rho = \rho_0 \exp\left(-\left(\frac{r - r_{Mars}}{h_s}\right)\right) \tag{5.7d}$$
$$r = x_1 + h_{design}(t) + r_{Mars} \tag{5.7e}$$

OCP formulation

The deterministic OCP is expressed in the following formulas:

$$\boldsymbol{x} = [x_1, x_2, x_3, x_4] = [e_h, v, \gamma, e_s]; \quad u = \sigma$$
$$J = \min_{u(t)} \left\{ \int_0^T \left({x_1}^2 + {x_4}^2\right) dt + \int_0^T u^2 dt \right\} \tag{5.8a}$$

subject to

$$\dot{x}_1 = x_2 \sin x_3 - \dot{h}_{design}(t) \tag{5.8b}$$

$$\dot{x}_2 = -\frac{1}{2}\left[\rho_0 \exp\left(-\frac{x_1 + h_{design}(t)}{h_s}\right)\right] {x_2}^2 \left(\frac{C_D S_r}{m}\right) - \left[\frac{\mu}{(x_1 + h_{design}(t) + r_{Mars})^2}\right] \sin x_3 \tag{5.8c}$$

$$\dot{x}_3 = \left[\frac{x_2}{x_1 + h_{design}(t) + r_{Mars}} - \frac{\left(\frac{\mu}{(x_1 + h_{design}(t) + r_{Mars})^2}\right)}{x_2}\right] \cos x_3 + \frac{1}{x_2}\left[\frac{1}{2}\rho_0 \exp\left(-\frac{x_1 + h_{design}(t)}{h_s}\right) {x_2}^2 \left(\frac{C_L S_r}{m}\right)\right] \cos u \tag{5.8d}$$

$$\dot{x}_4 = x_2 \cos x_3 - \dot{s}_{design}(t) \tag{5.8e}$$

$$0 \leq x_1^2 + x_4^2 \leq R^2 \tag{5.8f}$$

$$u_{min} \leq u \leq u_{max} \tag{5.8g}$$

NLP formulation

The NLP that is obtained from the deterministic OCP is expressed in the following formulas:

$$J = \min_{(x_1^k, x_2^k, x_3^k, x_4^k, u^k)} \left\{ \sum_{k=0}^{N-1} a_k \Delta t \left[(x_1^k)^2 + (x_4^k)^2\right] + \sum_{k=0}^{N-2} b_k \Delta t \left(u_{k+1} - u_k\right)^2 \right\} \tag{5.9a}$$

$$a_k = 2;\ k = 0, 1, 2, 3 \ldots (N-1)$$
$$b_k = 0.1;\ k = 0, 1, 2, 3 \ldots (N-2)$$

subject to

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{F}^k(\boldsymbol{x}_k, \boldsymbol{u}_k, t_k);\ k = 0, 1, 2 \ldots (N-1) \tag{5.9b}$$

For a better understanding of Equation (5.9b) see Equation (5.2b).

The discretized trajectory constraints are:

$$0 \leq \left((x_1^k)^2 + (x_4^k)^2\right) \leq R^2;\ k = 1, 2, \ldots N \tag{5.9c}$$

The upper and lower limits for the discretized control variable is:

$$10^\circ \leq u^k \leq 90^\circ;\ k = 0, 1, 2, \ldots (N-1) \tag{5.9d}$$

5.2 Stochastic MPC Formulation

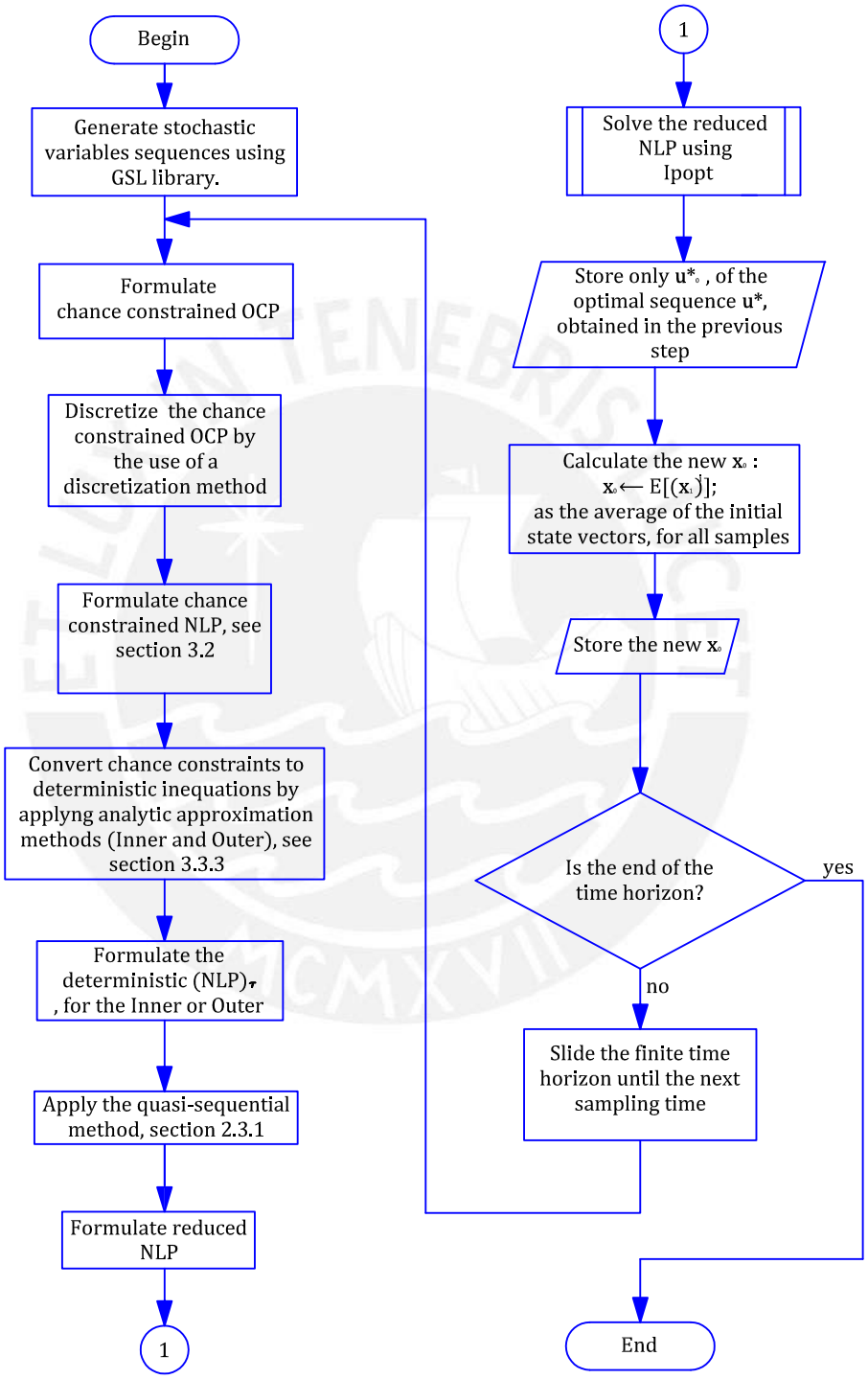


Figure 5.2 – Flowchart of the stochastic MPC.

The operation of the stochastic MPC is explained step by step in the flowchart of Figure 5.2. The stochastic MPC was formulated, for each of the study cases, based on this flowchart.
Some of the steps in the flowchart require additional precisions:

- It was used the discretization method of Runge-Kutta 4th order.
- The chance constrained OCP, chance constrained NLP, and the deterministic NLP_τ (inner or outer) are presented later in this section, for each study case.
- The GNU Scientific Library [97] was used in C++, to generate sequences of random numbers of low discrepancy (Sobol sequences).
- The first and second derivatives were calculated, making use of the same software that was used for the deterministic MPC, Section 5.1.
- The program of the stochastic MPC was implemented in C++, under the operating system Linux Ubuntu.

5.2.1 Trajectory tracking and landing on the asteroid Eros433

OCP formulation

The chance constrained OCP formulation is:

$$J = \min_{u_{x_1}, u_{x_2}, u_{x_3}} \left\{ E\left[\int_0^T \| \left(x_1(t) - x_d(t), x_2(t) - y_d(t), x_3(t) - z_d(t)\right) \|^2 dt \right] + \int_0^T \| \left(u_{x_1}, u_{x_2}, u_{x_3}\right) \|^2 dt \right\} \tag{5.10a}$$

subject to

Dynamics model Equation (4.1)

$$u_{x_1}^{min} \leq u_{x_1} \leq u_{x_1}^{max} \tag{5.10b}$$

$$u_{x_2}^{min} \leq u_{x_2} \leq u_{x_2}^{max} \tag{5.10c}$$

$$u_{x_3}^{min} \leq u_{x_3} \leq u_{x_3}^{max} \tag{5.10d}$$

$$Pr\left\{ \| \left(x_1(t) - x_d(t), x_2(t) - y_d(t), x_3(t) - z_d(t)\right) \|^2 \leq 1m^2 \right\} \geq 0.97 \tag{5.10e}$$

NLP formulation

After discretizing the chance constrained OCP, we will have a chance constrained NLP as explained in Section 3.2.
This NLP has the following objective function:

$$J = \min_{\boldsymbol{X}, \boldsymbol{u}} \left\{ \left(\frac{1}{M}\right) \sum_{i=1}^{M} \left(\sum_{j=0}^{N-1} a_j^i \Delta t \left[(x_{j,1}^i - x_{dj})^2 + (x_{j,2}^i - y_{dj})^2 + (x_{j,3}^i - z_{dj})^2 \right] \right) + \sum_{j=0}^{N-1} b_j \Delta t \left({u_{j,x_1}}^2 + {u_{j,x_2}}^2 + {u_{j,x_3}}^2 \right) \right\}, \tag{5.11a}$$

Runge Kutta 4th Order is used to discretize the Dynamic Equation (4.1). A set of equations will be obtained for each sample, this is explained in Equation (3.6b)

$$Pr\left\{ 0 \leq (x_{j,1} - x_{dj})^2 + (x_{j,2} - y_{dj})^2 + (x_{j,3} - z_{dj})^2 \leq 1m^2 \right\} \geq 0.97;\ j = 1, \ldots, N \tag{5.11b}$$

$$u_{j,x_k}^{min} \leq u_{j,x_k} \leq u_{j,x_k}^{max};\ j = 0, 1, \ldots, (N-1);\ k = 1, 2, 3 \tag{5.11c}$$

Equation (3.17) will be used to express the chance constraints of Equation (5.11b) in its equivalent form,

$$Pr\{c_j(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\xi}) \leq 0\} \geq 0.97;\ j = 1, 2, \ldots N, \tag{5.12}$$

where,

$$c_j(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi}) = p_2[\ln(\exp(p_1(-((x_{j,1}-x_{dj})^2+(x_{j,2}-y_{dj})^2+(x_{j,3}-z_{dj})^2)))+ \\ \exp(p_1(((x_{j,1}-x_{dj})^2+(x_{j,2}-y_{dj})^2+(x_{j,3}-z_{dj})^2)-1)))-p_3],$$

where p_1, p_2, p_3 are coefficients that have to be adjusted. The Inner Approximation(Section 3.3.3) is applied to Equation (5.12), obtaining the following group of deterministic inequalities.

$$\left(\frac{1}{M}\right)\sum_{i=1}^{M}\left(\frac{1+m_1\tau}{1+m_2\tau\exp\left(\frac{-c_j^i(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi})}{\tau}\right)}\right) \leq 0.03;\ j=1,2,\dots N, \tag{5.13}$$

If the Outer Approximation(Section 3.3.3) is applied, the following deterministic inequalities are obtained.

$$\left(\frac{1}{M}\right)\sum_{i=1}^{M}\left(\frac{1+m_1\tau}{1+m_2\tau\exp\left(\frac{c_j^i(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi})}{\tau}\right)}\right) \geq 0.97;\ j=1,2,\dots N, \tag{5.14}$$

Finally a deterministic NLP is obtained that will be solved by IpOpt. The equations corresponding to the $\text{NLP}_\tau^{\text{Inner}}$ are Equations (5.11a), (5.11c) and (5.13), and the discretized Dynamic Equation (4.1) for all samples, the corresponding ones to the $\text{NLP}_\tau^{\text{Outer}}$ are the same as the latter but instead of Equation (5.13), it will be Equation (5.14).

5.2.2 Guidance law in three dimensions for a soft landing on a celestial body

OCP formulation

The chance constrained OCP formulation is:

$$J = \min_{u_r, u_\Theta, u_\Phi} \left\{ E\left[\int_0^T \left((r(t) - r_d(t))^2 + (\Theta(t) - \Theta_d(t))^2 + (\Phi(t) - \Phi_d(t))^2 \right) dt \right] + \int_0^T \left(u_r^2 + u_\Theta^2 + u_\Phi^2 \right) dt \right\} \quad (5.15a)$$

subject to

Dynamics model Equation (4.10)

$$u_r^{min} \leq u_r \leq u_r^{max} \quad (5.15b)$$

$$u_\Theta^{min} \leq u_\Theta \leq u_\Theta^{max} \quad (5.15c)$$

$$u_\Phi^{min} \leq u_\Phi \leq u_\Phi^{max} \quad (5.15d)$$

$$Pr\left\{ 0 \leq \left((x(t) - x_d(t))^2 + (y(t) - y_d(t))^2 + (z(t) - z_d(t))^2 \right) \leq 25km^2 \right\} \geq 0.97 \quad (5.15e)$$

Equation (5.15e) is expressed in cartesian coordinates, so we must transform it into spherical coordinates.

$$x(t) = r \cos\Phi \cos\Theta \quad (5.16a)$$

$$y(t) = r \cos\Phi \sin\Theta \quad (5.16b)$$

$$z(t) = r \sin\Phi \quad (5.16c)$$

$$x_d(t) = r_d \cos\Phi_d \cos\Theta_d \quad (5.16d)$$

$$y_d(t) = r_d \cos\Phi_d \sin\Theta_d \quad (5.16e)$$

$$z_d(t) = r_d \sin\Phi_d \quad (5.16f)$$

Replacing Equation (5.16) in Equation (5.15e), we obtain the following Chance Constraint.

$$Pr\left\{ 0 \leq \left((r\cos\Phi\cos\Theta - r_d\cos\Phi_d\cos\Theta_d)^2 + (r\cos\Phi\sin\Theta - r_d\cos\Phi_d\sin\Theta_d)^2 + (r\sin\Phi - r_d\sin\Phi_d)^2 \right) \leq 25km^2 \right\} \geq 0.97$$

NLP formulation

After discretizing the chance constrained OCP, we will have a chance constrained NLP as explained in Section 3.2.

This NLP has the following objective function:

$$J = \min_{\boldsymbol{X},\boldsymbol{u}} \left\{ \left(\frac{1}{M}\right) \sum_{i=1}^{M} \left(\sum_{j=0}^{N-1} a_j^i \Delta t \left[(x_{j,1}^i - r_{dj})^2 + (x_{j,2}^i - \Theta_{dj})^2 + (x_{j,3}^i - \Phi_{dj})^2 \right] \right) + \sum_{j=0}^{N-1} b_j \Delta t \left({u_{j,r}}^2 + {u_{j,\Theta}}^2 + {u_{j,\Phi}}^2 \right) \right\}, \tag{5.17a}$$

Runge Kutta 4th Order is used to discretize the Dynamic Equation (4.10). A set of equations will be obtained for each sample, this is explained in Equation (3.6b)

$$\begin{aligned} Pr \Big\{ 0 \leq & (x_{j,1} \cos x_{j,3} \cos x_{j,2} - r_{dj} \cos \Phi_{dj} \cos \Theta_{dj})^2 + \\ & (x_{j,1} \cos x_{j,3} \sin x_{j,2} - r_{dj} \cos \Phi_{dj} \sin \Theta_{dj})^2 + \\ & (x_{j,1} \sin x_{j,3} - r_{dj} \sin \Phi_{dj})^2 \leq 25km^2 \Big\} \geq 0.97;\ j = 1, \ldots, N \end{aligned} \tag{5.17b}$$

$$u_{j,k}^{min} \leq u_{j,k} \leq u_{j,k}^{max};\ j = 0, 1, \ldots, (N-1);\ k = r, \Theta, \Phi \tag{5.17c}$$

Equation (3.17) will be used to express the chance constraints of Equation (5.17b) in its equivalent form,

$$Pr\{c_j(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\xi}) \leq 0\} \geq 0.97;\ j = 1, 2, \ldots N, \tag{5.18}$$

where,

$$\begin{aligned} Expression = & (x_{j,1} \cos x_{j,3} \cos x_{j,2} - r_{dj} \cos \Phi_{dj} \cos \Theta_{dj})^2 + \\ & (x_{j,1} \cos x_{j,3} \sin x_{j,2} - r_{dj} \cos \Phi_{dj} \sin \Theta_{dj})^2 + \\ & (x_{j,1} \sin x_{j,3} - r_{dj} \sin \Phi_{dj})^2, \\ c_j(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\xi}) = & p_2[\ln(\exp(p_1(-Expression)) + \exp(p_1(Expression - 25))) - p_3], \end{aligned}$$

where p_1, p_2, p_3 are coefficients that have to be adjusted. The Inner Approximation(Section 3.3.3) is applied to Equation (5.18), obtaining the following group of deterministic inequalities.

$$\left(\frac{1}{M}\right) \sum_{i=1}^{M} \left(\frac{1 + m_1 \tau}{1 + m_2 \tau \exp\left(\frac{-c_j^i(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\xi})}{\tau}\right)} \right) \leq 0.03;\ j = 1, 2, \ldots N, \tag{5.20}$$

If the Outer Approximation(Section 3.3.3) is applied, the following deterministic inequalities are obtained.

$$\left(\frac{1}{M}\right) \sum_{i=1}^{M} \left(\frac{1 + m_1 \tau}{1 + m_2 \tau \exp\left(\frac{c_j^i(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\xi})}{\tau}\right)} \right) \geq 0.97;\ j = 1, 2, \ldots N, \tag{5.21}$$

Finally a deterministic NLP is obtained that will be solved by IpOpt. The equations corresponding to the $\text{NLP}_{\tau}^{\text{Inner}}$ are Equations (5.17a), (5.17c) and (5.20), and the discretized Dynamic Equation (4.10) for all samples, the corresponding ones to the $\text{NLP}_{\tau}^{\text{Outer}}$ are the same as the latter but instead of Equation (5.20), it will be Equation (5.21).

5.2.3 Longitudinal model of lifting entry of a Mars Lander

OCP Formulation

The formulation of the OCP is as follows:

$$J = \min_{u(t)} \left\{ E\left[\int_0^T \left({x_1}^2 + {x_4}^2 \right) dt \right] + \int_0^T u^2 dt \right\} \tag{5.22a}$$

subject to

Equations (5.6) and (5.7)

$$u_{min} \leq u \leq u_{max} \tag{5.22b}$$

$$Pr\left\{ 0 \leq x_1^2 + x_4^2 \leq R^2 \right\} \geq 0.97 \tag{5.22c}$$

NLP Formulation

After discretizing the chance constrained OCP, we will have a chance constrained NLP as explained in Section 3.2.

This NLP has the following objective function:

$$J = \min_{\boldsymbol{X},\boldsymbol{u}} \left\{ \left(\frac{1}{M} \right) \sum_{i=1}^{M} \left(\sum_{j=0}^{N-1} a_j^i \Delta t \left[(x_{j,1}^i)^2 + (x_{j,4}^i)^2 \right] \right) + \sum_{j=0}^{N-2} b_j \Delta t \left(u_{j+1} - u_j \right)^2 \right\}, \tag{5.23a}$$

Runge Kutta 4th Order is used to discretize the Dynamic Equation (5.6). A set of equations will be obtained for each sample, this is explained in Equation (3.6b)

$$Pr\left\{ 0 \leq x_{j,1}^2 + x_{j,4}^2 \leq R^2 \right\} \geq 0.97;\ j = 1, \ldots, N \tag{5.23b}$$

$$u_j^{min} \leq u_j \leq u_j^{max};\ j = 0, 1, \ldots, (N-1) \tag{5.23c}$$

Equation (3.17) will be used to express the chance constraints of Equation (5.23b) in its equivalent form,

$$Pr\{ c_j(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\xi}) \leq 0 \} \geq 0.97;\ j = 1, 2, \ldots N, \tag{5.24}$$

where,

$$c_j(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\xi}) = p_2[\ln(\exp(p_1(-(x_{j,1}^2 + x_{j,4}^2))) + \exp(p_1((x_{j,1}^2 + x_{j,4}^2) - R^2))) - p_3],$$

where p_1, p_2, p_3 are coefficients that have to be adjusted. The Inner Approximation(Section 3.3.3) is applied to Equation (5.24), obtaining the following group of

deterministic inequalities.

$$\left(\frac{1}{M}\right)\sum_{i=1}^{M}\left(\frac{1+m_1\tau}{1+m_2\tau\exp\left(\frac{-c_j^i(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi})}{\tau}\right)}\right)\leq 0.03;\ j=1,2,\dots N, \tag{5.25}$$

If the Outer Approximation(Section 3.3.3) is applied, the following deterministic inequalities are obtained.

$$\left(\frac{1}{M}\right)\sum_{i=1}^{M}\left(\frac{1+m_1\tau}{1+m_2\tau\exp\left(\frac{c_j^i(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\xi})}{\tau}\right)}\right)\geq 0.97;\ j=1,2,\dots N, \tag{5.26}$$

Finally a deterministic NLP is obtained that will be solved by IpOpt. The equations corresponding to the $\text{NLP}_\tau^{\text{Inner}}$ are Equations (5.23a), (5.23c) and (5.25), and the discretized Dynamic Equation (5.6) for all samples, the corresponding ones to the $\text{NLP}_\tau^{\text{Outer}}$ are the same as the latter but instead of Equation (5.25), it will be Equation (5.26).

Chapter 6

Computational results

In this chapter we show the computational results, from the simulations of the study cases, in the deterministic and stochastic MPC.
All simulations were carried out on a desktop computer with the following features: Intel Core i7 CPU X980, 6-core, with 3.33Ghz of speed per core, and 6GB of RAM.

6.1 Deterministic results

In this section we show the results of the simulations, of solving the case studies with the deterministic MPC.

6.1.1 Trajectory tracking and landing on the asteroid Eros433

This study case was solved with the deterministic MPC approach, discussed in Section 5.1, we also used the equations in Section 5.1.1 and the data in Table 4.4. The method of discretization was Runge-Kutta 4th order, the finite time horizon was $N = 10$, the sampling time was $\Delta t = 1s$.
The iteration time reported by IpOpt, to solve the optimization problem, is in the interval $[0.09, 0.5]$ seconds.
In Figures 6.1 to 6.3, the data obtained by IpOpt is observed in visual form.

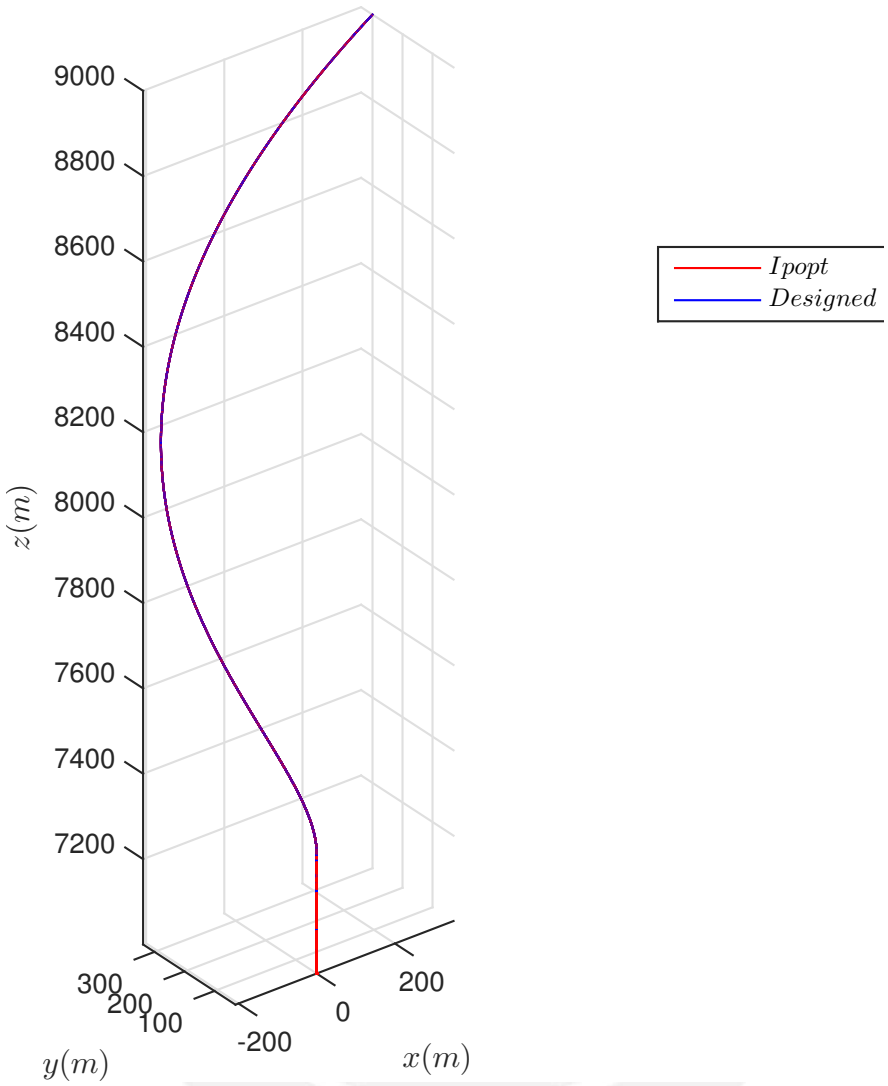
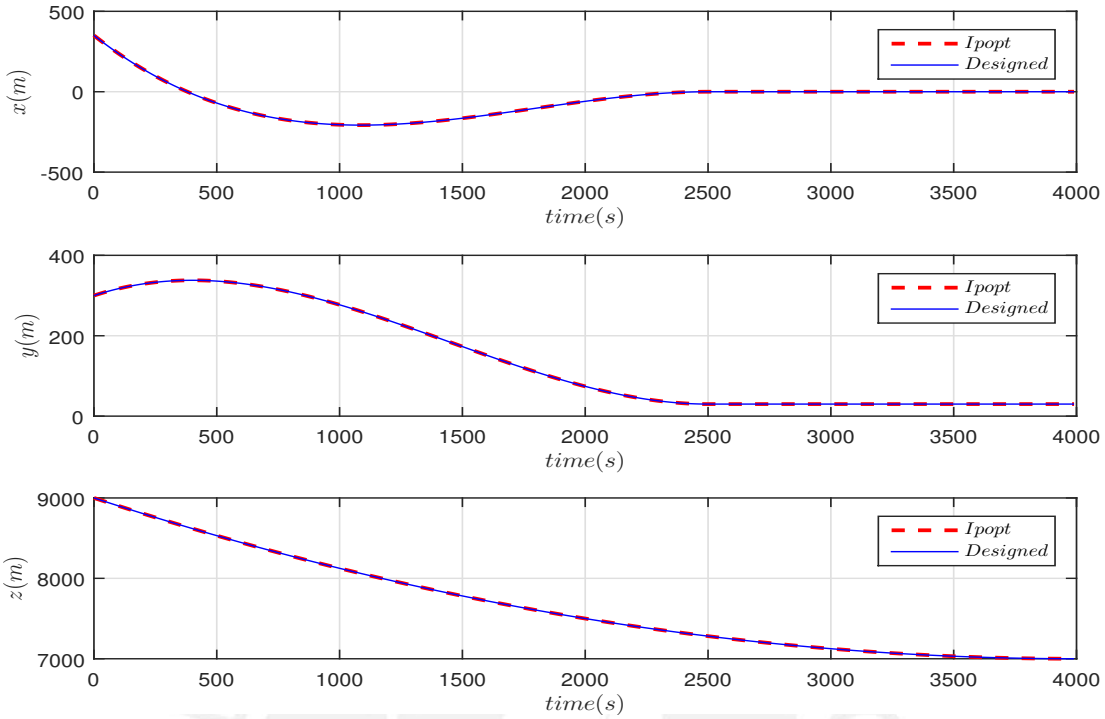
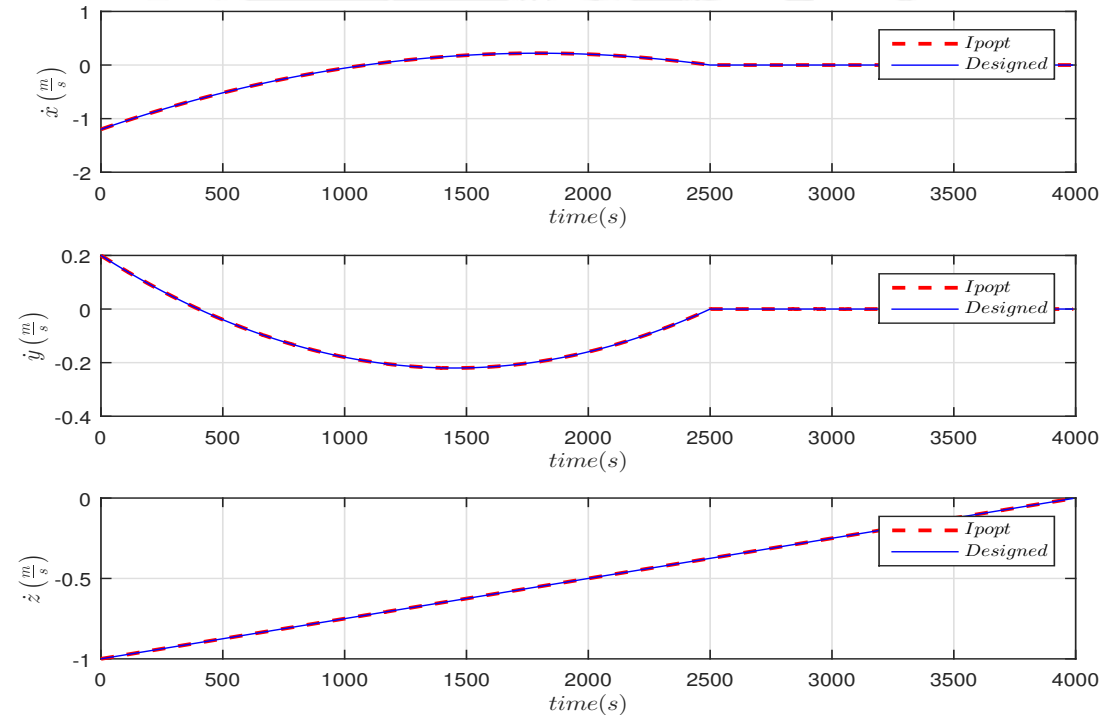


Figure 6.1 – Trajectory calculated by IpOpt and trajectory designed, you can see that both trajectories are very close, the error of the tracking of the position is small. Calculated using Runge-Kutta 4th Order discretization method, finite time horizon $N = 10$, $\Delta t = 1s$, for Case 1.

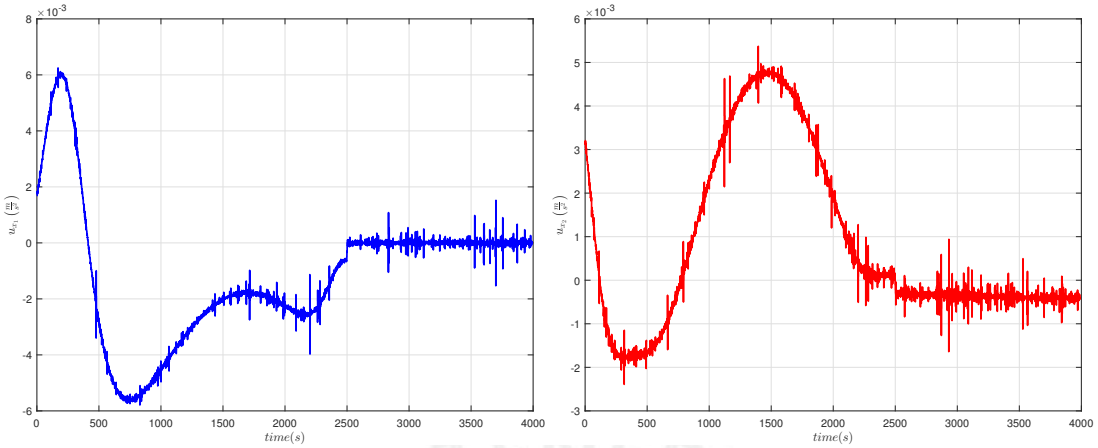


(a) Tracking the trajectory designed over time in each of its components, it can be noted that the error of tracking the position is small, almost imperceptible.



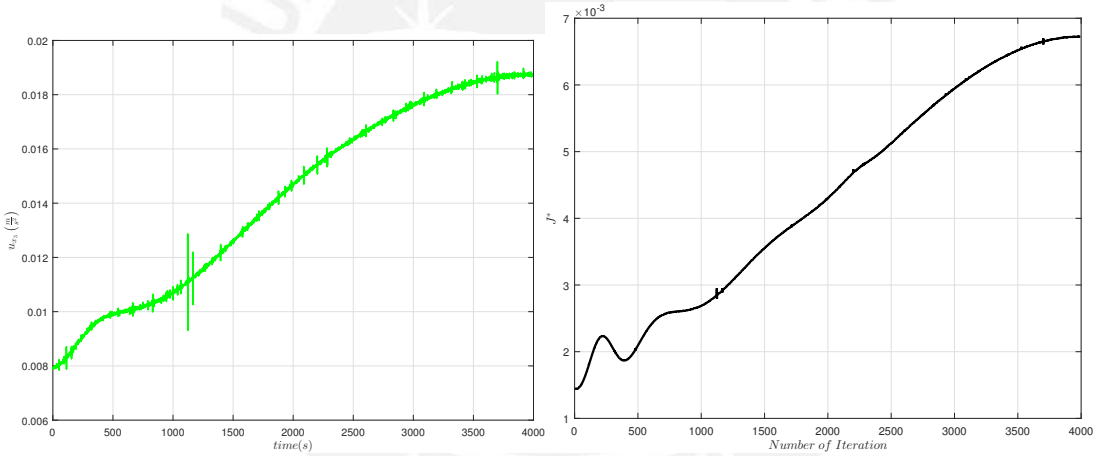
(b) This graph shows that the velocities calculated by the IpOpt follow with an error almost imperceptible to the speeds designed.

Figure 6.2 – Calculated using the Runge-Kutta 4th order method for discretization, a finite time horizon of $N = 10$, $\Delta t = 1s$, for Case 1.



(a) It can be seen that the graph of u_{x_1} has the shape of the graph Figure 4.4 but inverted with respect to the time axis, this is logical since the controller has to exert a force or effect contrary to that produced by gravity, plus an effect or force that allows to follow the path designed with low error, that is the reason why the graphs have the same shape, but do not match their values at every instant in time.

(b) Analogous to the analysis of Figure 6.3a but with respect to the corresponding graph of Figure 4.4.



(c) Analogous to the analysis of Figure 6.3a but with respect to the corresponding graph of Figure 4.4.

(d) The tendency of this graph to increase its value in each iteration is because as the Spacecraft is closer to the asteroid, the intensity of the gravitational field increases in the component "z", as shown in Figure 4.4, the controller has to increase the effort applied to counteract the intensity of the gravitational field that increases in the component "z", this is seen in Figure 6.3c.

Figure 6.3 – Graph of control variables computed by IpOpt, using the Runge-Kutta 4th order discretization method, a finite time horizon of $N = 10$, $\Delta t = 1s$, for Case 1. A graph of the value of the objective function in each iteration is also shown.

6.1.2 Guidance law in three dimensions for a soft landing on a celestial body

This case study was solved with the deterministic MPC approach, discussed in Section 5.1, we also used the equations in Section 5.1.2 and the data in Table 4.5. The method of discretization was Runge-Kutta 4th order, the finite time horizon was $N = 10$, the sampling time was $\Delta t = 0.07h$.

The iteration time reported by IpOpt, to solve the optimization problem, is in the interval $]0, 0.144]$ seconds.

In Figures 6.4 to 6.6, the data obtained by IpOpt is observed in visual form.

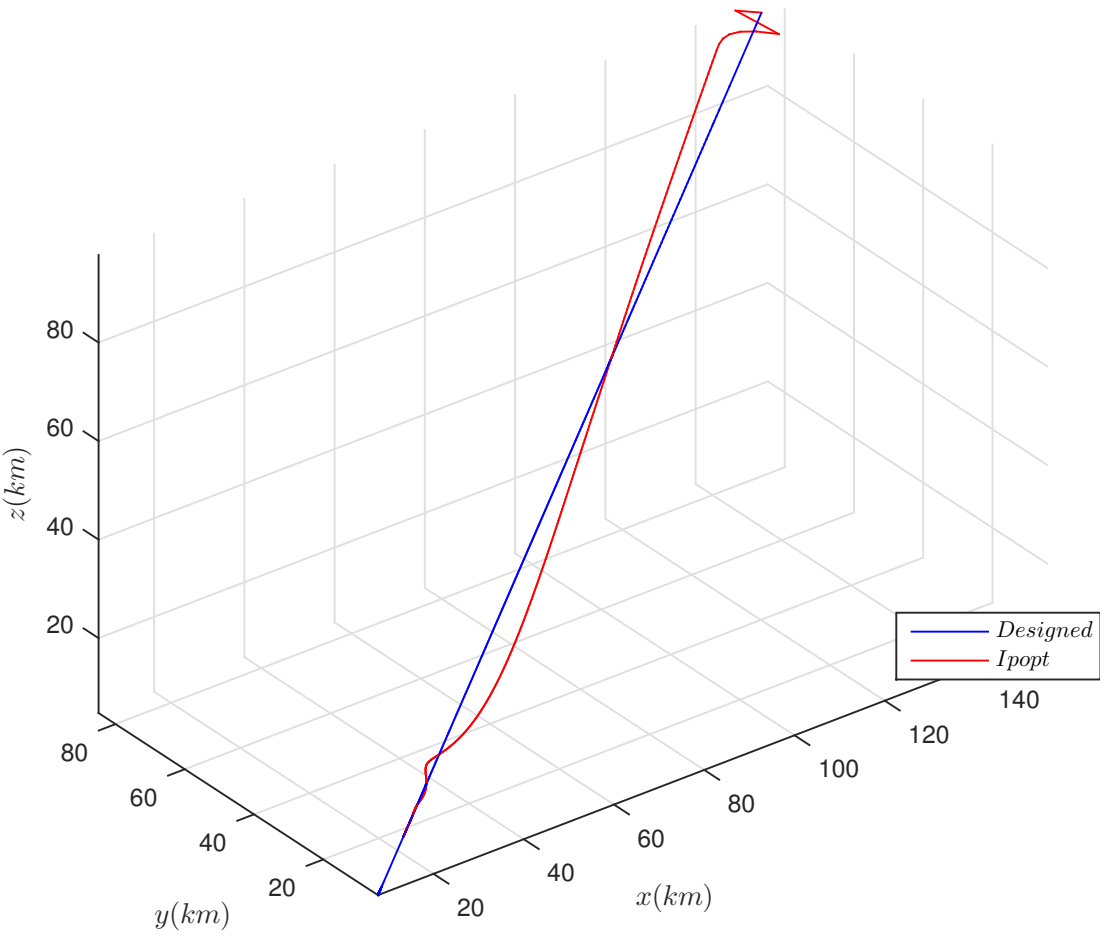
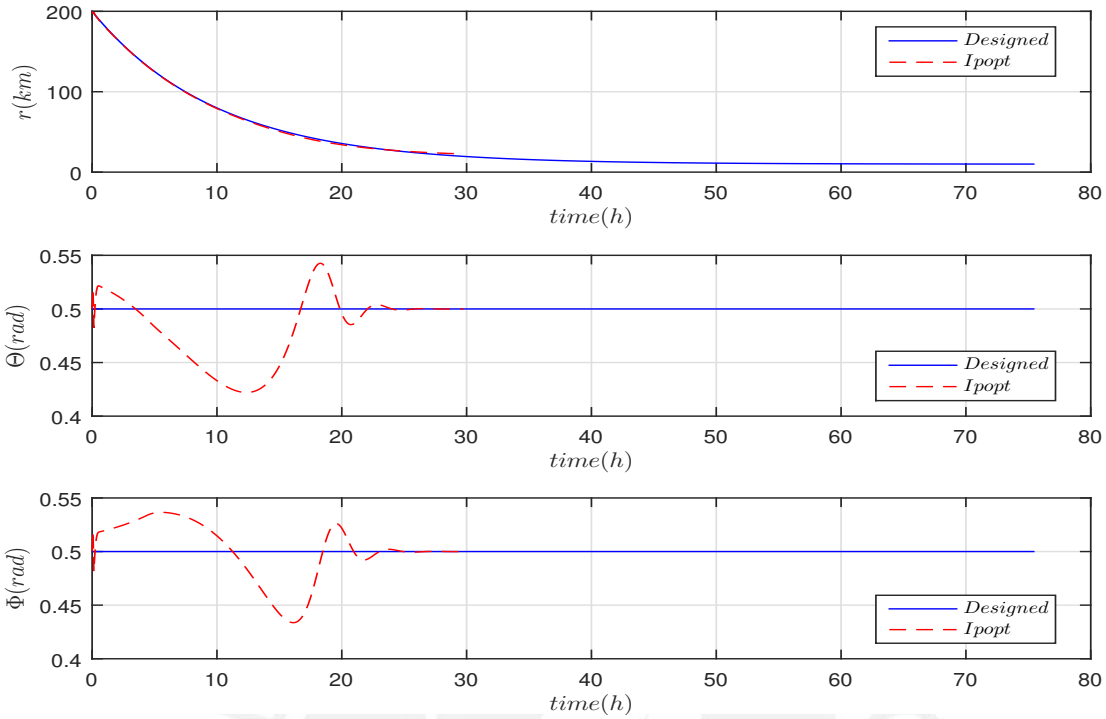
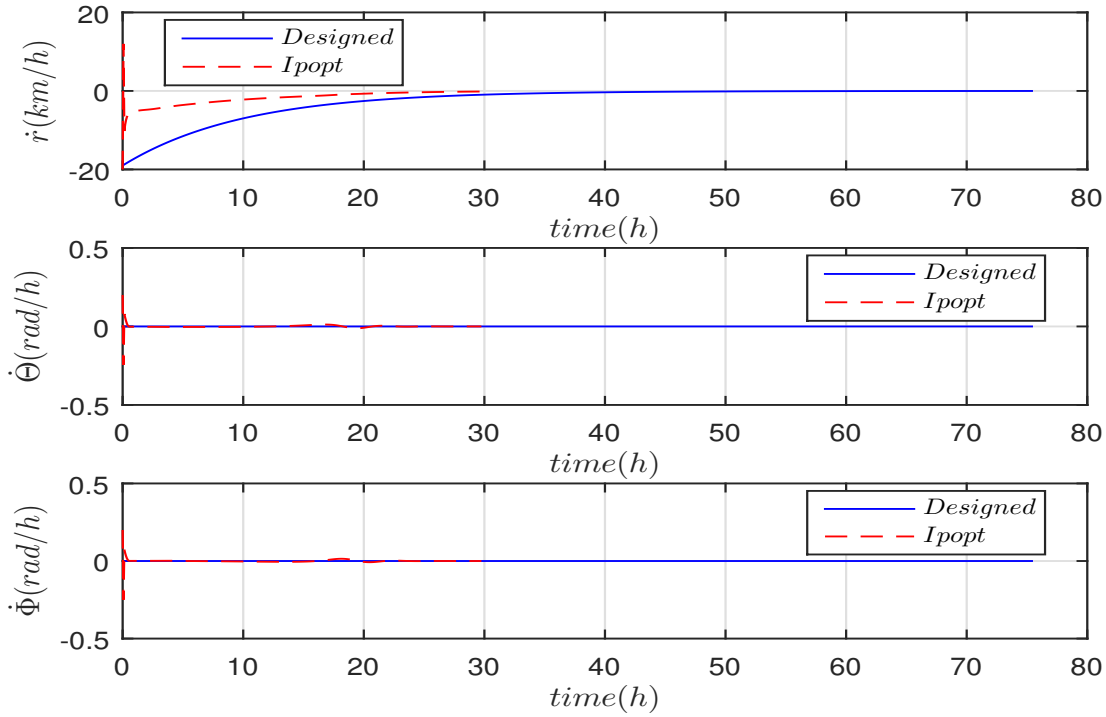


Figure 6.4 – Trajectory designed and trajectory calculated by IpOpt, for Case 2, with a finite time horizon $N = 10$, sampling time $\Delta t = 0.07h$, and using Runge-Kutta 4th order discretization method. It is observed that at the beginning there is a great error to follow the path designed, after a time the controller manages to follow the path with low error.

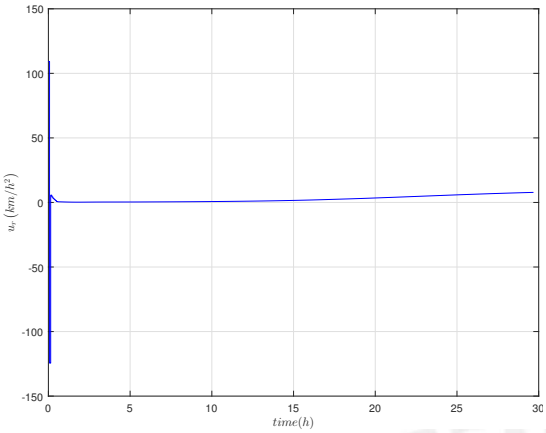


(a) Tracking the trajectory designed over time in each of its components, it has to be remarked that only the simulation could be carried out until the time $t \approx 30h$, after that instant of time the IpOpt software reports problems of precision and aborts the simulation.

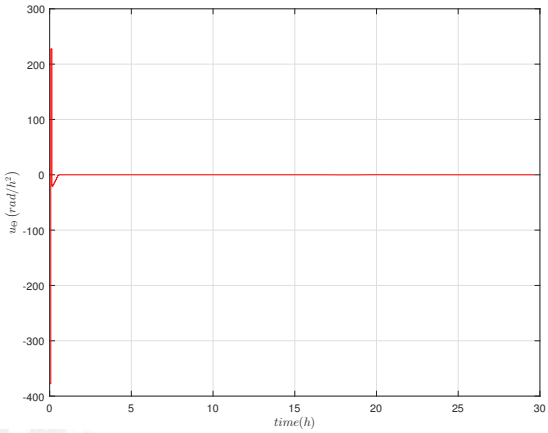


(b) In this graph it is seen how the speed calculated by the IpOpt follows the designed speed, in its three components. It is very remarkable the error in the component $\dot{r}$, besides converging to zero relatively fast, at time $t \approx 30h$, it is possible that this premature convergence to zero, is the cause by which IpOpt fails.

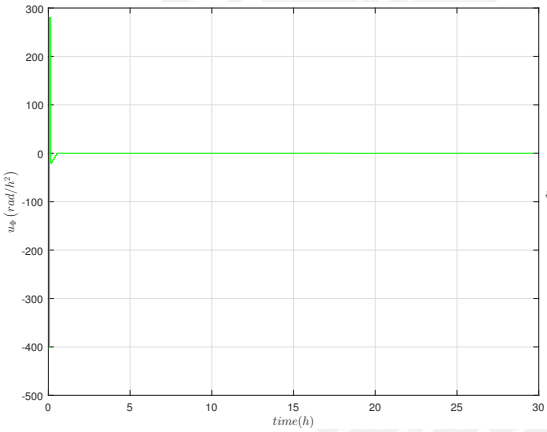
Figure 6.5 – Calculated using the Runge-Kutta 4th Order discretization method, a finite time horizon of $N = 10$, $\Delta t = 0.07h$, for Case 2.



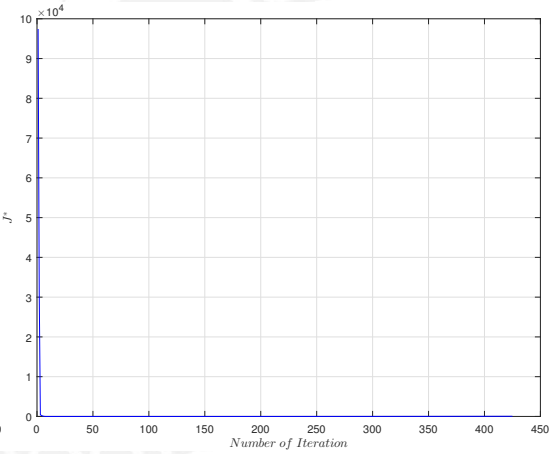
(a) The oscillatory behavior observed in the initial instants is of short duration, after it is observed that the effort of the controller increases in time, this is because as the Spacecraft approaches the celestial body, the gravitational field becomes more intense, which translates into a greater effort of the controller in component r, so that the landing is smooth.



(b) In this graph it is observed at the beginning a sudden change of the effort applied by the controller, in a relatively short time, the reason of this abrupt change is because in the tracking of the designed speed in this component Θ, Figure 6.5b, in the instant of time 0, there is a noticeable difference in the magnitude of the velocity of the IpOpt, with respect to the one designed. After the above explained, the controller effort in this component is almost constant in the value of 0, this steady behavior is because the gravitational field only acts in the component r, but not in Θ or Φ.



(c) The analysis of this graph is analogous to that of Figure 6.6b.



(d) In this graph the objective function takes after a few iterations a value of almost 0.

Figure 6.6 – Control variables calculated by IpOpt, using the Runge-Kutta 4th order discretization method, a finite time horizon of $N = 10$, $\Delta t = 0.07h$, for Case 2.

The same case was also solved, using the Heun discretization method (Runge-Kutta 2nd order), a finite time horizon $N = 10$, a sampling time $\Delta t = 0.05h$.

The reason why this case is approached using Heun discretization method is because, when attempting to solve it using the Runge-Kutta 4th order discretization method, with the stochastic MPC approach, there were many problems of precision.

The time IpOpt uses to solve the optimization problem varies from iteration to iteration, this time is in the interval $]0, 0.208]$ seconds.

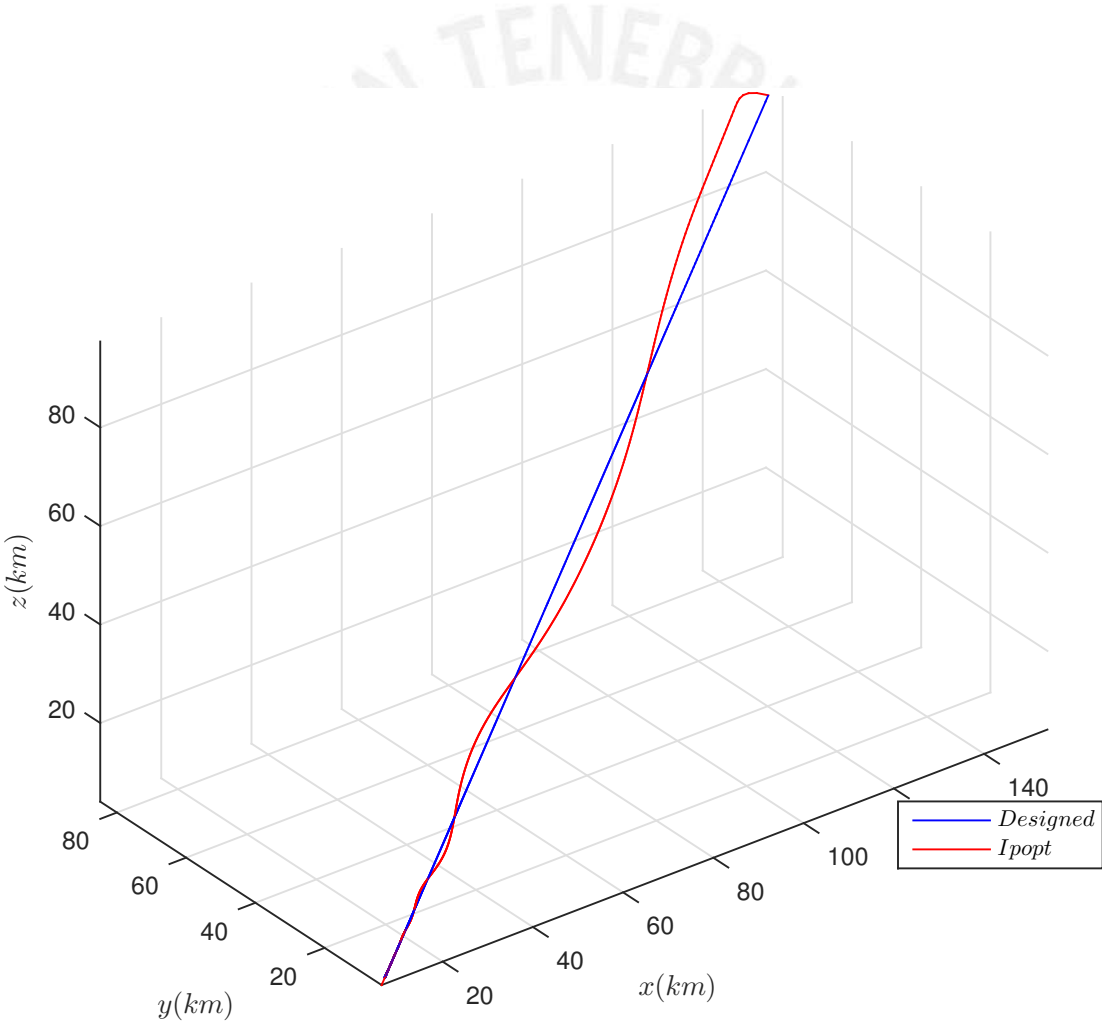
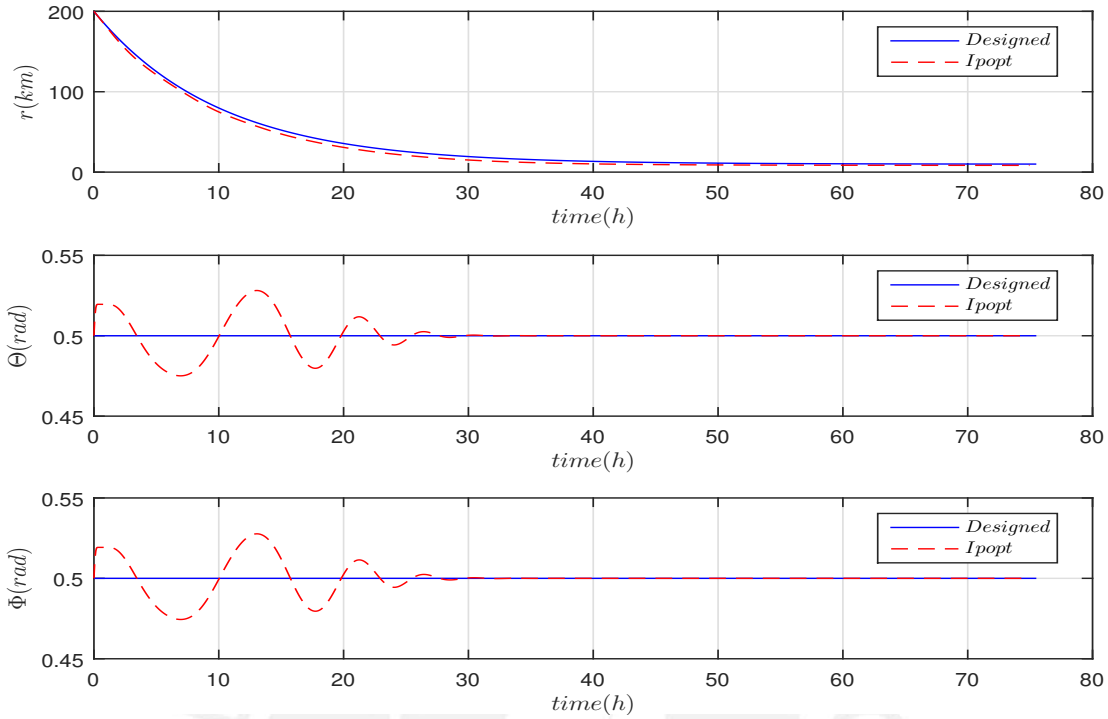
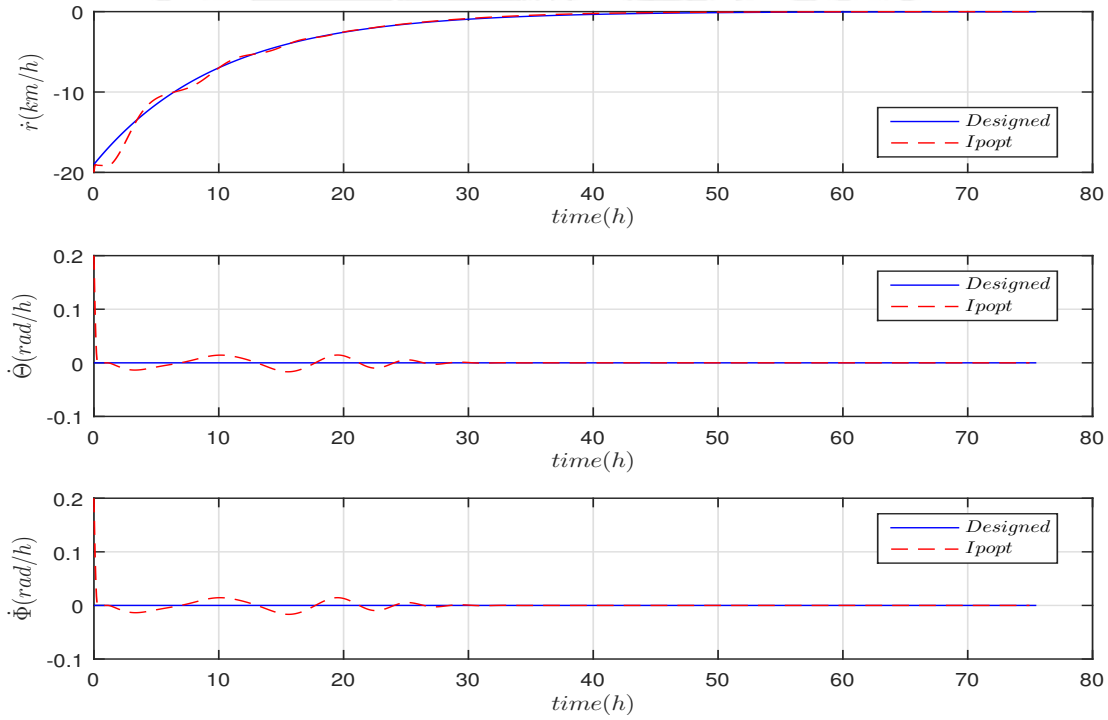


Figure 6.7 – Trajectory designed and trajectory calculated by IpOpt, for Case 2, with a finite time horizon $N = 10$, sampling time $\Delta t = 0.05h$, and using Heun discretization method. It is observed that at the beginning there is a great error to follow the path designed, after a time the controller manages to follow the path with low error.

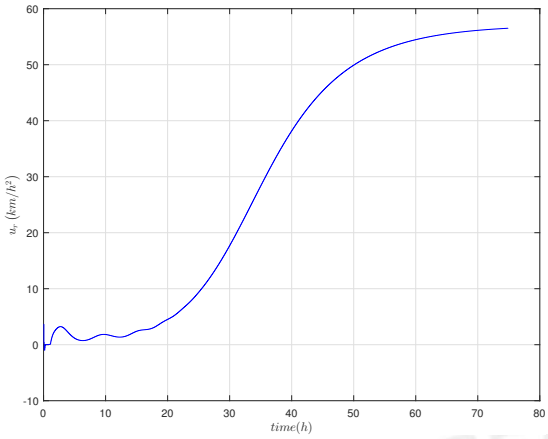


(a) Tracking the trajectory designed over time in each of its components, it is appreciated that until the time instant $t \approx 30h$, there is an error in the tracking of the designed path, that is relatively small, in each component.

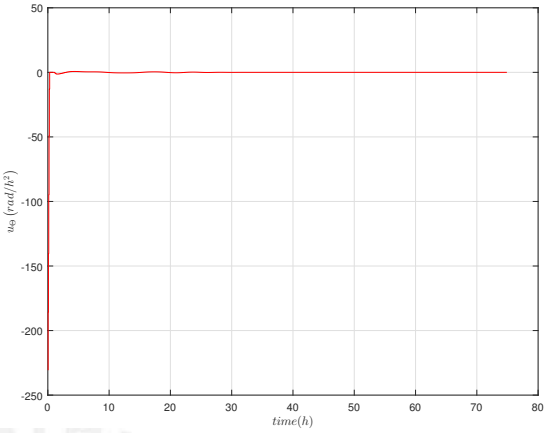


(b) In this graph it is seen how the speed calculated by the IpOpt follows the designed speed, in its three components.

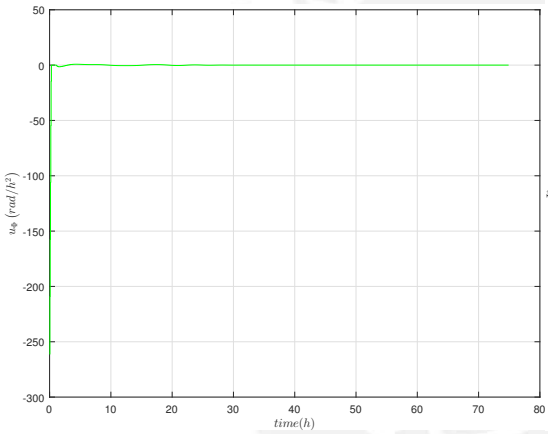
Figure 6.8 – Calculated using the Heun method for discretization, a finite time horizon of $N = 10$, $\Delta t = 0.05h$, for Case 2.



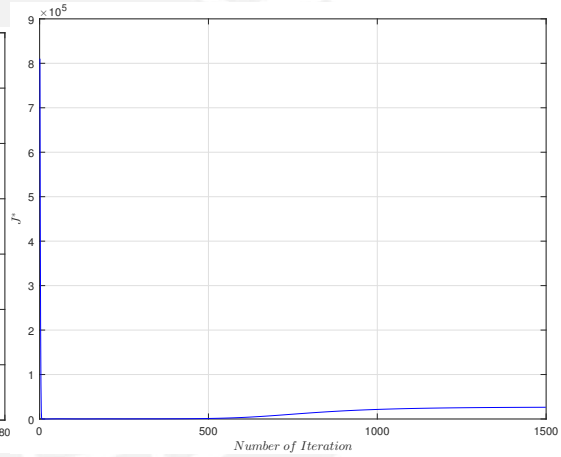
(a) It is observed that the effort of the controller increases in time, this is because as the Spacecraft approaches the celestial body, the gravitational field becomes more intense, which translates into a greater effort of the controller in component r, so that the landing is smooth.



(b) The controller effort in this component is almost constant in the value of 0, with imperceptible oscillations (it is necessary to zoom), this steady behavior, because the gravitational field only acts in the component r, but not in Θ or Φ.



(c) The analysis of this graph is analogous to that of Figure 6.9b.

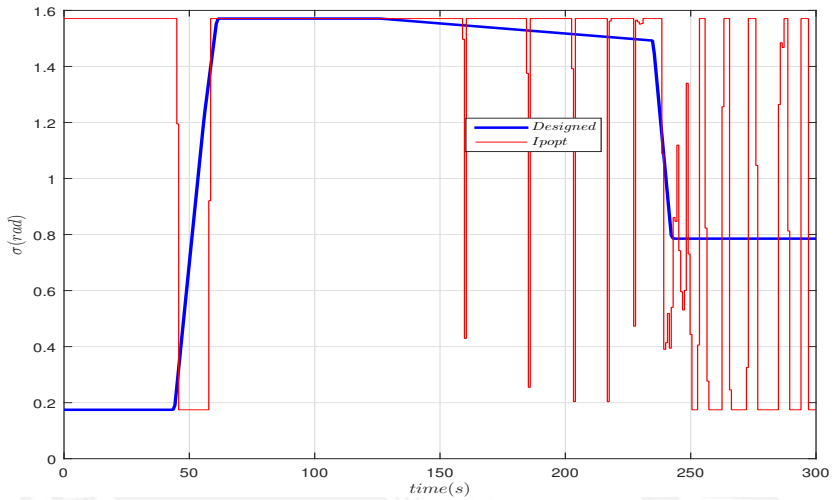


(d) In this graph the objective function takes after a few iterations a value of almost 0, but then begins to increase slightly because as the Spacecraft approaches the celestial body, the intensity of the gravitational field in the component r increases, see Figure 6.9a.

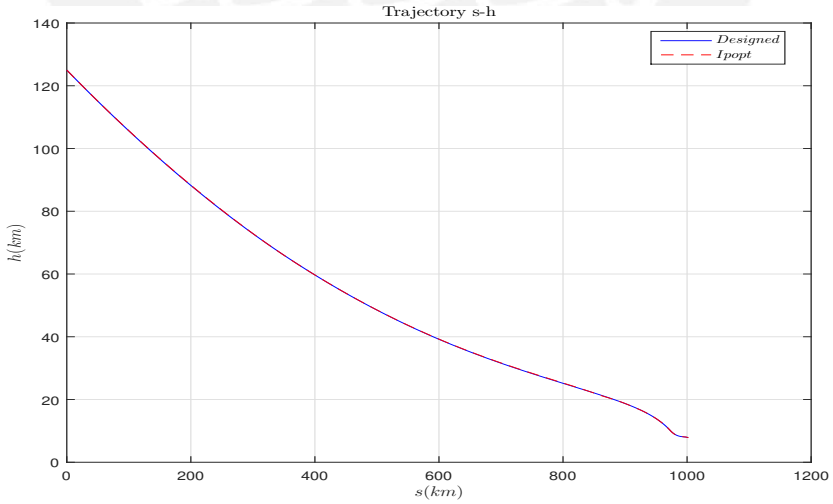
Figure 6.9 – Control variables calculated by IpOpt, using the Heun discretization method, a finite time horizon of $N = 10$, $\Delta t = 0.05h$, for Case 2.

6.1.3 Longitudinal model of lifting entry of a Mars Lander

This case study was solved with the deterministic MPC approach, discussed in Section 5.1, we also used the equations in Section 5.1.3 and the data in Table 4.7.
The method of discretization was Runge-Kutta 4th order, the finite time horizon was $N = 15$, the sampling time was $\Delta t = 0.75s$.
The iteration time reported by IpOpt, to solve the optimization problem, is in the interval $]0, 0.16]$ seconds.
In Figures 6.10 and 6.11, the data obtained by IpOpt is observed in visual form.

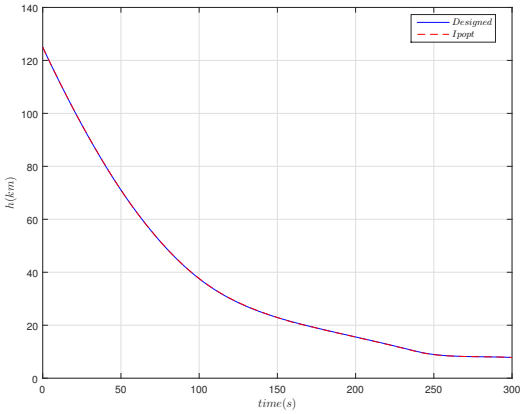


(a) The control variable calculated by the IpOpt in contrast to the one that was designed, shows sudden changes in its value over time, it was expected that its graph is similar to the one designed, a possible cause of this result is that of the 4 states of the dynamic equation of the model, we have 3 whose rate of change in time is high, whereas only the state γ has a very slow rate of change over time.

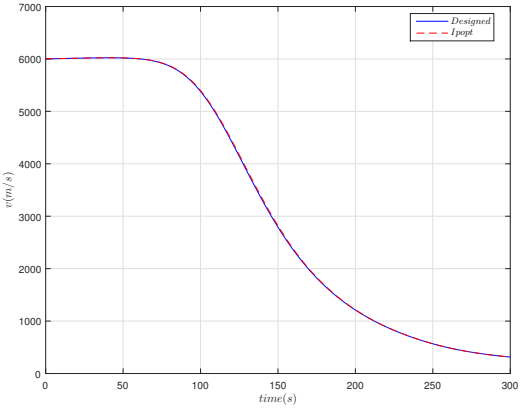


(b) Trajectory designed in the $s-h$ plane, also the trajectory calculated by IpOpt, it is observed that the controller manages to follow the path designed with low error.

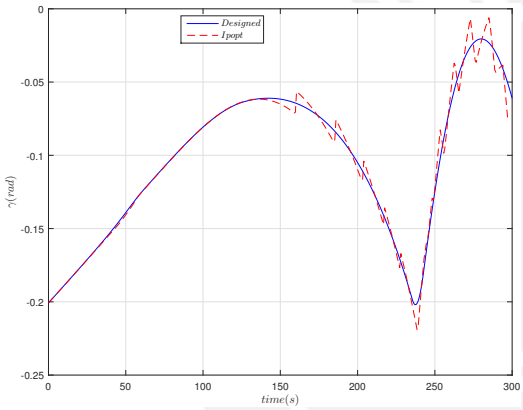
Figure 6.10 – Graph of σ, graph of the trajectory designed in the $s-h$ plane.



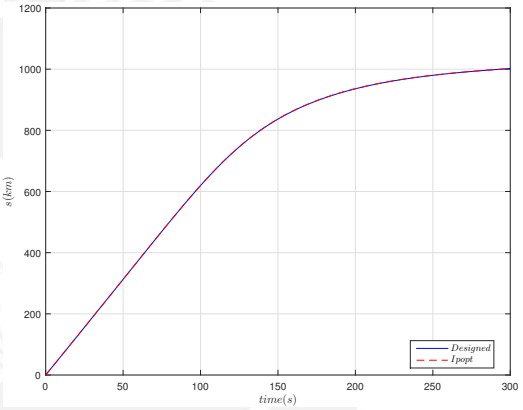
(a) The controller does a tracking of this state with almost imperceptible error.



(b) The controller does a tracking of this state with almost imperceptible error.

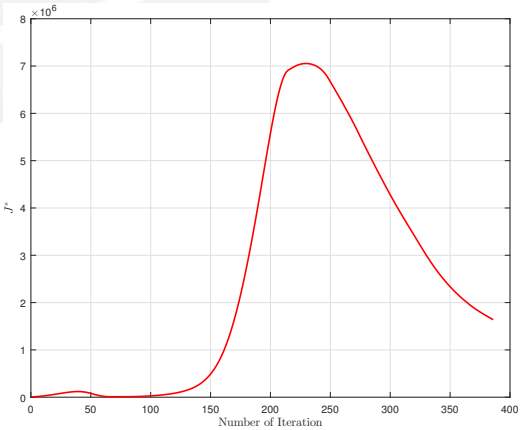


(c) The controller fails to perform a very good tracking of this state, it is seen in the graph that there is a relatively large error from the time $t \approx 150s$, a possible cause because the controller is unable to follow this state with a low error, is because the rate of change of said state with respect to time is low.



(d) The controller does a tracking of this state with almost imperceptible error.

(e) Error at every instant of time.



(f) The graph of the objective function has the same form as the error graph, Figure 6.11e, the interpretation is that as the Mars Lander is diverted from the designed path, the error increases, and consequently the controller will try harder to reduce said error of tracking.

Figure 6.11 – Graphs of the tracking of the 4 states $[h, v, \gamma, s]$, also the Objective function.

6.2 Stochastic results

In this section we show the results of the simulations, of solving the case studies with the stochastic MPC.

6.2.1 Trajectory tracking and landing on the asteroid Eros433

This case was solved using the stochastic MPC approach, explained in Section 5.2, also the equations of Section 5.2.1, and the data of Table 4.4.
Different set of parameters were tested to solve the $\text{NLP}_{\tau}^{Outer}$ and $\text{NLP}_{\tau}^{Inner}$, below a summary of the obtained results (Tables 6.1 to 6.3).

	Number of iterations	Cause of failure
$\tau_{\text{Inner}} = 0.9$	475	Problem may be infeasible
$\tau_{\text{Inner}} = 0.8$	478	Problem may be infeasible
$\tau_{\text{Inner}} = 0.7$	478	Problem may be infeasible
$\tau_{\text{Inner}} = 0.6$	487	Problem may be infeasible
$\tau_{\text{Inner}} = 0.5$	498	Problem may be infeasible
$\tau_{\text{Inner}} = 0.4$	497	Problem may be infeasible
$\tau_{\text{Inner}} = 0.3$	0	Restoration failed
$\tau_{\text{Inner}} = 0.2$	0	Restoration failed
$\tau_{\text{Inner}} = 0.1$	0	Restoration failed
$\tau_{\text{Outer}} = 0.9$	0	Restoration failed
$\tau_{\text{Outer}} = 0.8$	0	Restoration failed
$\tau_{\text{Outer}} = 0.7$	0	Restoration failed
$\tau_{\text{Outer}} = 0.6$	0	Restoration failed
$\tau_{\text{Outer}} = 0.5$	0	Restoration failed
$\tau_{\text{Outer}} = 0.4$	0	Restoration failed
$\tau_{\text{Outer}} = 0.3$	0	Restoration failed
$\tau_{\text{Outer}} = 0.2$	0	Restoration failed
$\tau_{\text{Outer}} = 0.1$	0	Restoration failed

Table 6.1 – $m_1 = m_2 = 0.01$, $p_1 = 2$, $p_2 = 500$, $p_3 = 0.25$, $a = 1$, $b = 2.5$, $\alpha = 0.95$, $M = 1000$ samples, $N = 10$ (Finite time horizon), $\Delta t = 1s$, Runge-Kutta 4th order as discretization method, $\sigma_{\delta_{x_1}} = \sigma_{\delta_{x_2}} = \sigma_{\delta_{x_3}} = 10^{-4}\left(\frac{m}{s}\right)$, $\sigma_{\delta_{x_4}} = \sigma_{\delta_{x_5}} = \sigma_{\delta_{x_6}} = 10^{-4}\left(\frac{m}{s^2}\right)$

It was also tried to simulate with $M = 10000$ samples, using the same data of Table 6.1, but with different standard deviation for the stochastic variables $\sigma_{\delta_{x_1}} = \sigma_{\delta_{x_2}} = \sigma_{\delta_{x_3}} = 10^{-5}\left(\frac{m}{s}\right)$, $\sigma_{\delta_{x_4}} = \sigma_{\delta_{x_5}} = \sigma_{\delta_{x_6}} = 10^{-5}\left(\frac{m}{s^2}\right)$.
It was tested for $\tau = \{0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2\}$, the results were:

- For the $\text{NLP}_{\tau}^{Outer}$, in all cases IpOpt showed the message "Restoration Failed", it was not iterated even once,

- For the $\text{NLP}_{\tau}^{Inner}$, IpOpt invests a lot of time per iteration, after a certain number of iterations fails, cannot find the control law for the entire time horizon ($T = 4000$).

	Number of iterations	Cause of failure
$\tau_{\text{Inner}} = 0.9$	0	Problem may be infeasible
$\tau_{\text{Inner}} = 0.8$	0	Problem may be infeasible
$\tau_{\text{Inner}} = 0.7$	2	Problem may be infeasible
$\tau_{\text{Inner}} = 0.6$	3	Problem may be infeasible
$\tau_{\text{Inner}} = 0.5$	1	Problem may be infeasible
$\tau_{\text{Inner}} = 0.4$	0	Problem may be infeasible
$\tau_{\text{Outer}} = 0.9$	0	Restoration failed
$\tau_{\text{Outer}} = 0.8$	0	Restoration failed
$\tau_{\text{Outer}} = 0.7$	0	Restoration failed
$\tau_{\text{Outer}} = 0.6$	0	Restoration failed
$\tau_{\text{Outer}} = 0.5$	0	Restoration failed
$\tau_{\text{Outer}} = 0.4$	0	Restoration failed

Table 6.2 – $m_1 = m_2 = 0.01,\ p_1 = 2,\ p_2 = 60,\ p_3 = 0.25,\ a = 1,\ b = 2.5,\ \alpha = 0.95,\ M = 1000$ samples, $N = 10$ (Finite time horizon), $\Delta t = 1s$, Runge-Kutta 4th order as discretization method, $\sigma_{\delta_{x_1}} = \sigma_{\delta_{x_2}} = \sigma_{\delta_{x_3}} = 10^{-6}\left(\frac{m}{s}\right),\ \sigma_{\delta_{x_4}} = \sigma_{\delta_{x_5}} = \sigma_{\delta_{x_6}} = 2.5 \times 10^{-3}\left(\frac{m}{s^2}\right)$

	Number of iterations	Cause of failure
$\tau_{\text{Inner}} = 0.9$	481	Problem may be infeasible
$\tau_{\text{Inner}} = 0.8$	491	Problem may be infeasible
$\tau_{\text{Inner}} = 0.7$	493	Problem may be infeasible
$\tau_{\text{Inner}} = 0.6$	517	Problem may be infeasible
$\tau_{\text{Inner}} = 0.5$	514	Problem may be infeasible
$\tau_{\text{Inner}} = 0.4$	534	Problem may be infeasible
$\tau_{\text{Outer}} = 0.9$	0	Restoration failed
$\tau_{\text{Outer}} = 0.8$	0	Restoration failed
$\tau_{\text{Outer}} = 0.7$	0	Restoration failed
$\tau_{\text{Outer}} = 0.6$	0	Restoration failed
$\tau_{\text{Outer}} = 0.5$	0	Restoration failed
$\tau_{\text{Outer}} = 0.4$	0	Restoration failed

Table 6.3 – $m_1 = m_2 = 0.01,\ p_1 = 2,\ p_2 = 500,\ p_3 = 0.25,\ a = 1,\ b = 2.5,\ \alpha = 0.95,\ M = 100$ samples, $N = 10$ (Finite time horizon), $\Delta t = 1s$, Runge-Kutta 4th order as discretization method, $\sigma_{\delta_{x_1}} = \sigma_{\delta_{x_2}} = \sigma_{\delta_{x_3}} = 10^{-5}\left(\frac{m}{s}\right),\ \sigma_{\delta_{x_4}} = \sigma_{\delta_{x_5}} = \sigma_{\delta_{x_6}} = 10^{-5}\left(\frac{m}{s^2}\right)$

Next, we graph the data obtained by Ipopt, for some cases in Table 6.1.
For a $\tau_{\text{Inner}} = 0.9$, the time that IpOpt invests to solve the optimization problem, is in the interval $[1.724, 2.752]$ seconds, for each iteration. In Figures 6.12 to 6.15, the data obtained with the IpOpt is shown in visual form.

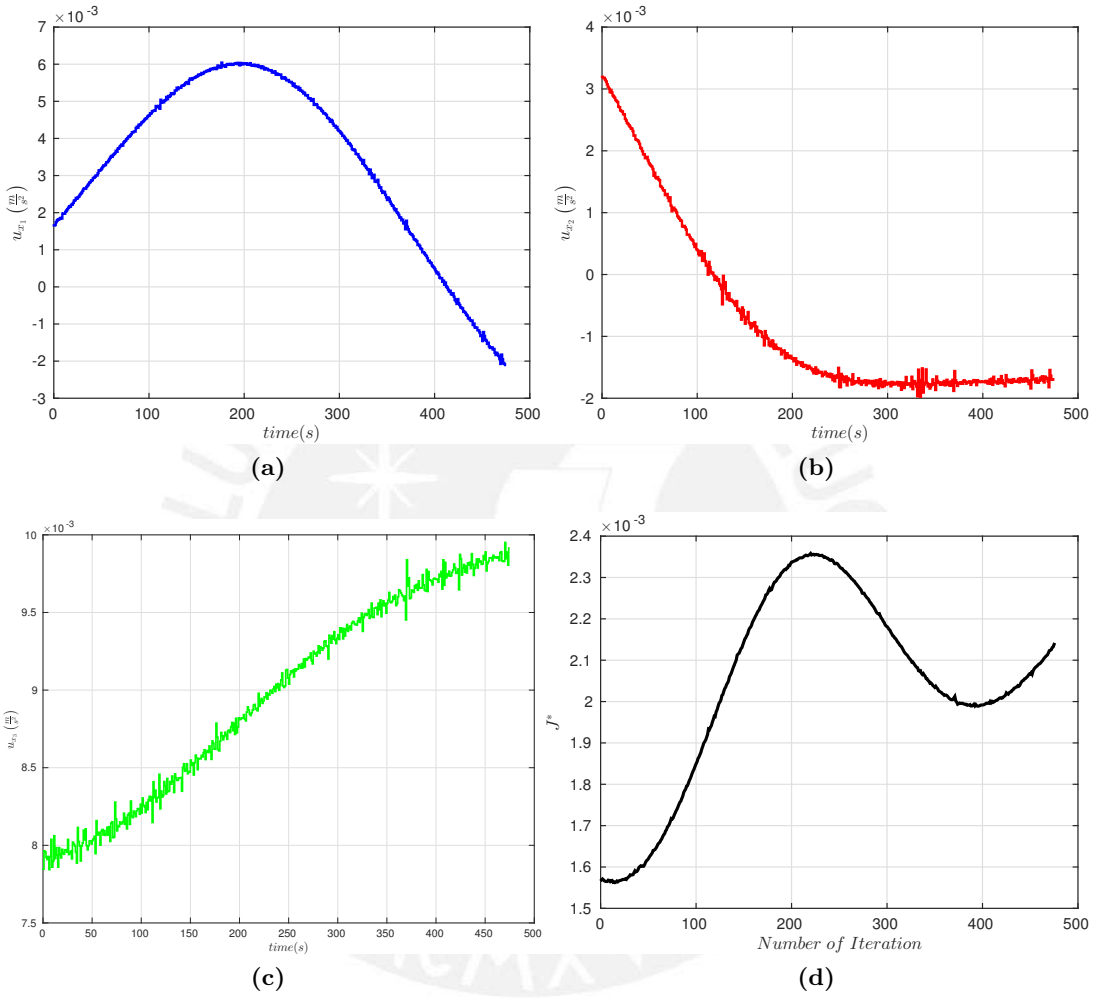
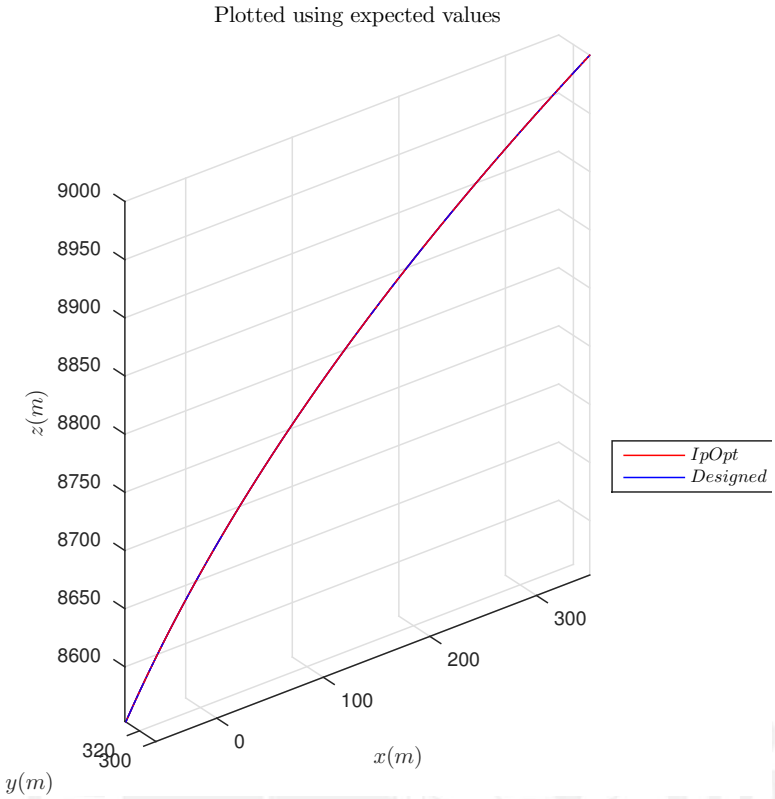
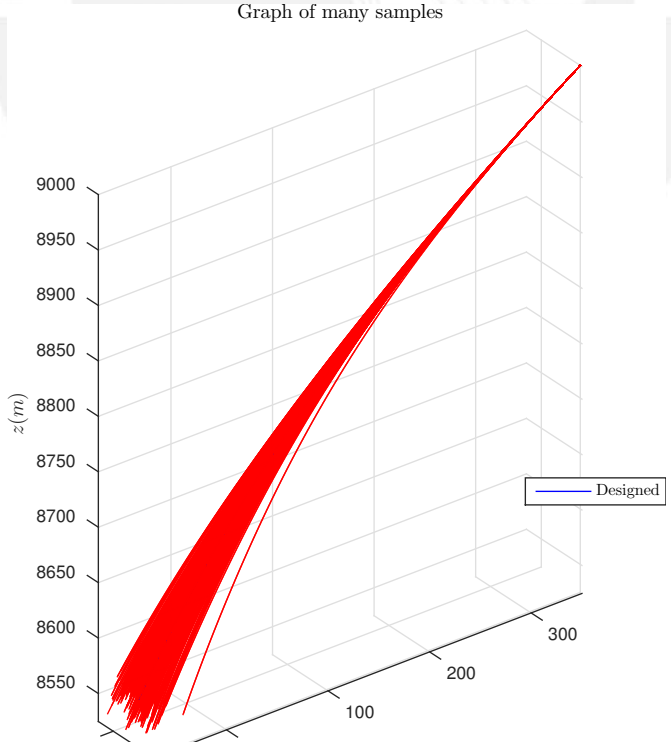


Figure 6.12 – Graph of control variables computed by IpOpt. A graph of the value of the objective function in each iteration is also shown. Note that it is only possible to iterate until time $t \approx 500$.

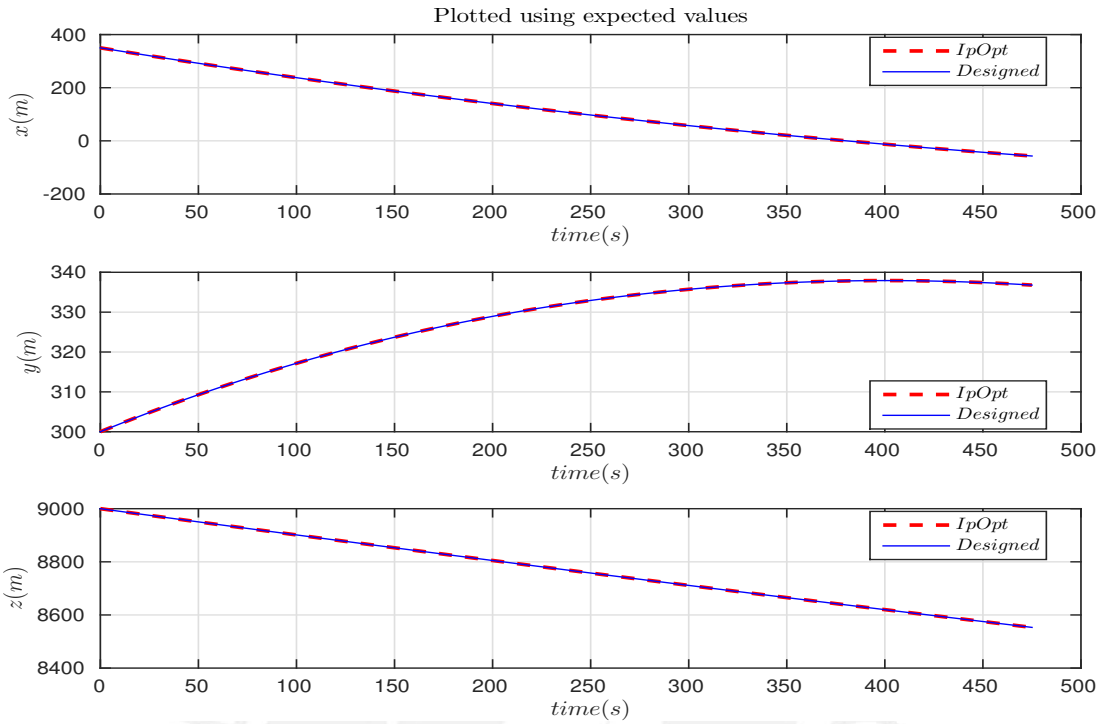


(a)

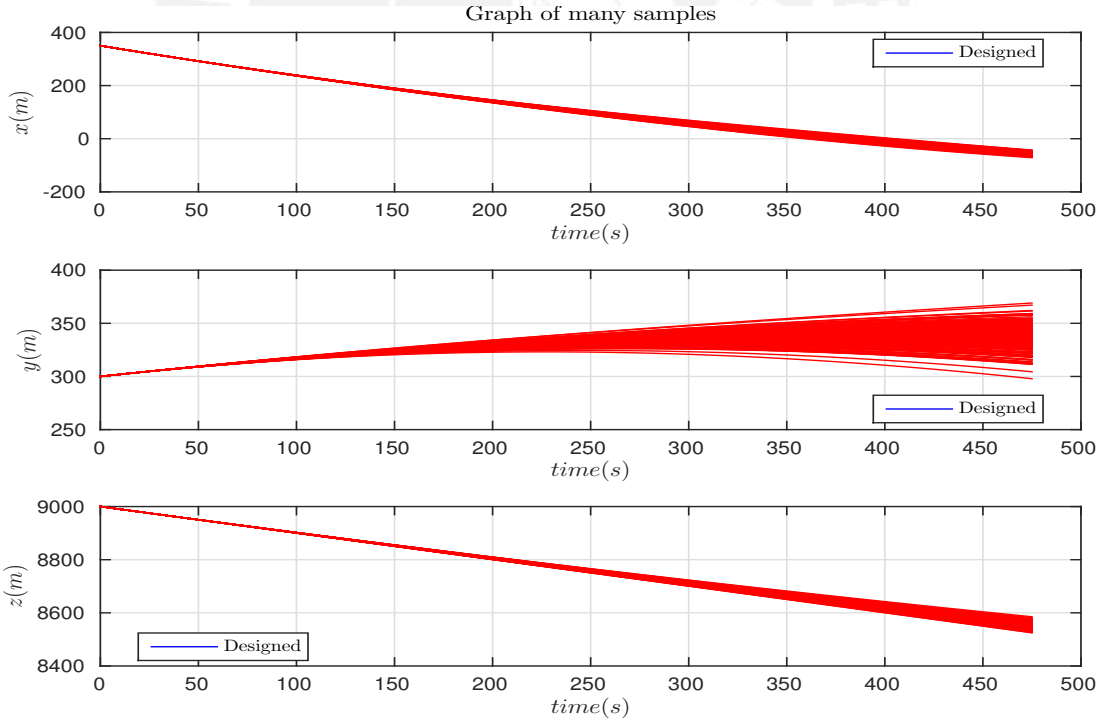


(b)

Figure 6.13 – Trajectory calculated by IpOpt and trajectory designed. Note that it is only possible to iterate until time $t \approx 500$.

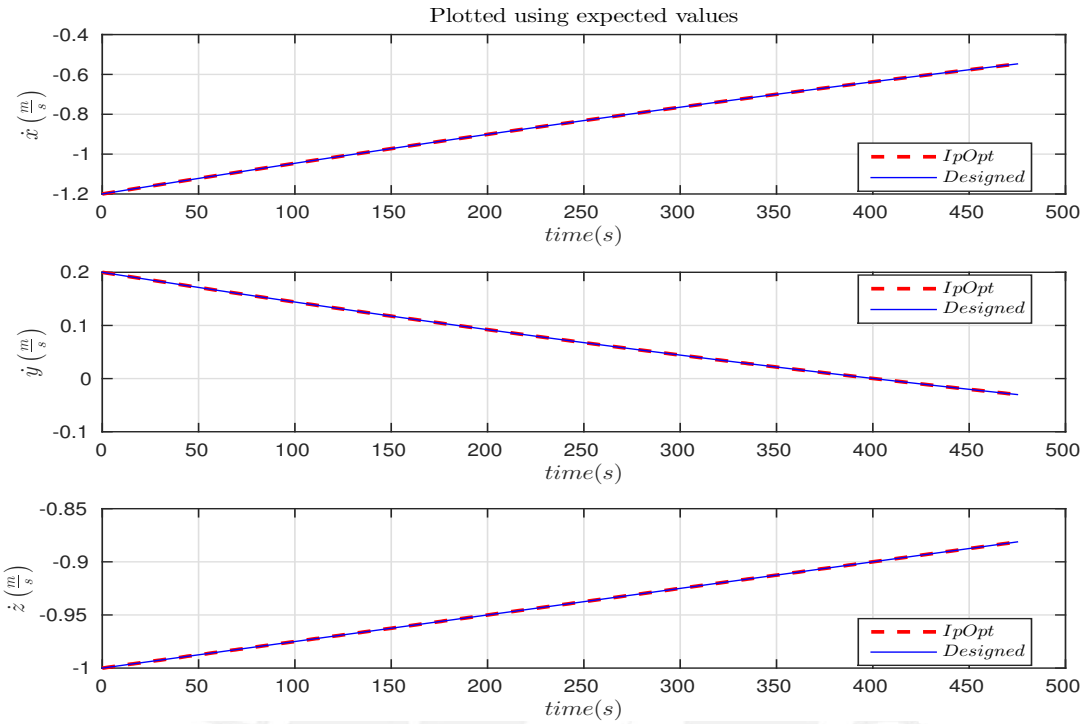


(a)

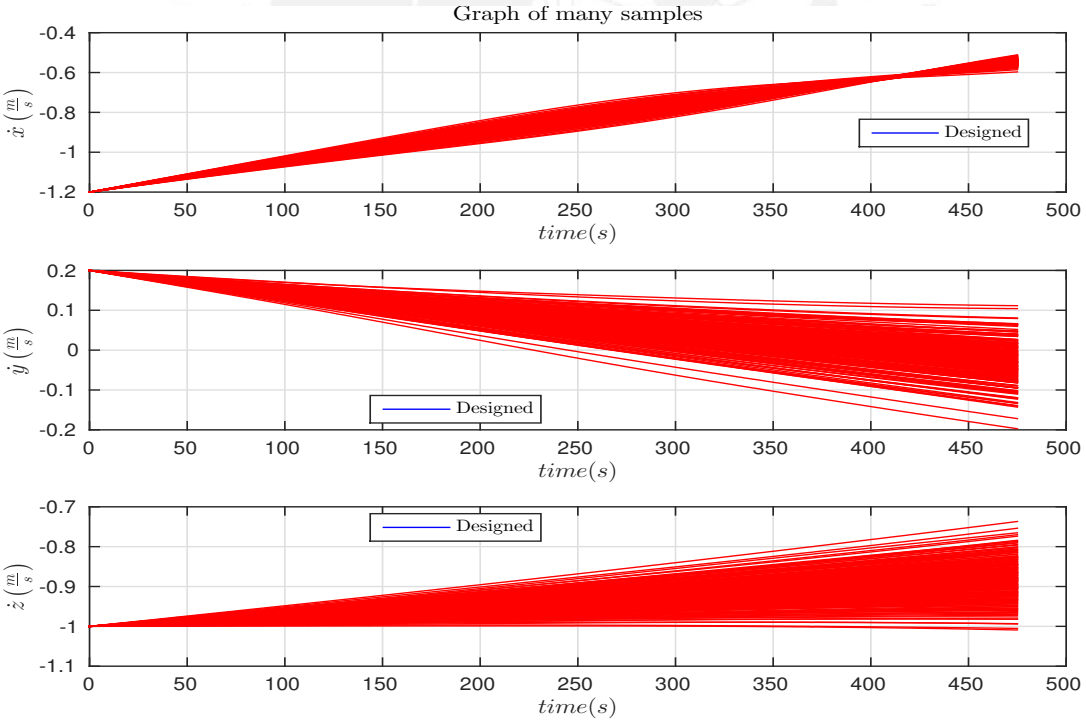


(b)

Figure 6.14 – Tracking of the position.



(a) This graph shows that the velocities calculated by the IpOpt follow with an error almost imperceptible to the speeds designed.



(b)

Figure 6.15 – Tacking of the velocities.

For a $\tau_{\text{Inner}} = 0.6$, the time that IpOpt invests to solve the optimization problem, is in the interval $[1.676, 2.176]$ seconds, for each iteration. In Figures 6.16 to 6.19, the data obtained with the IpOpt is shown in visual form.

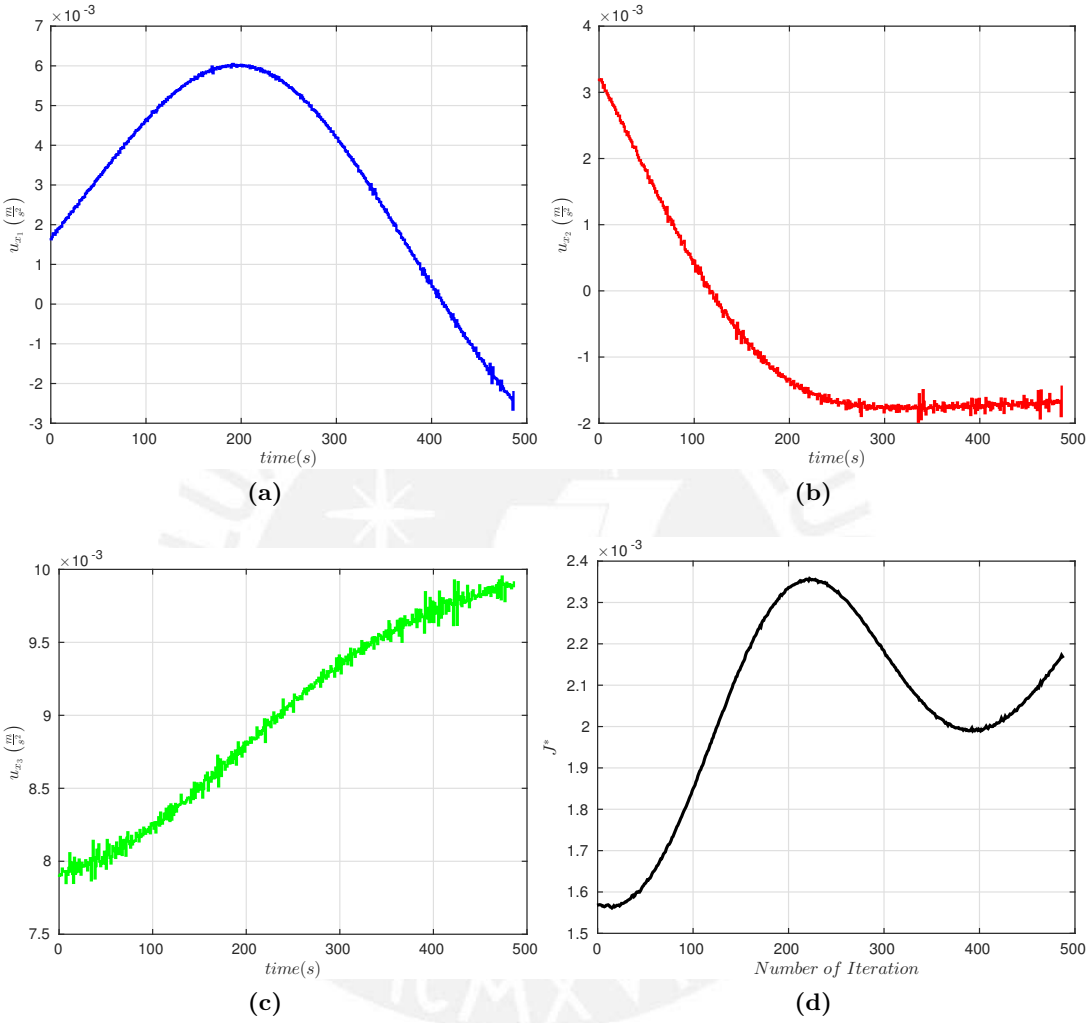
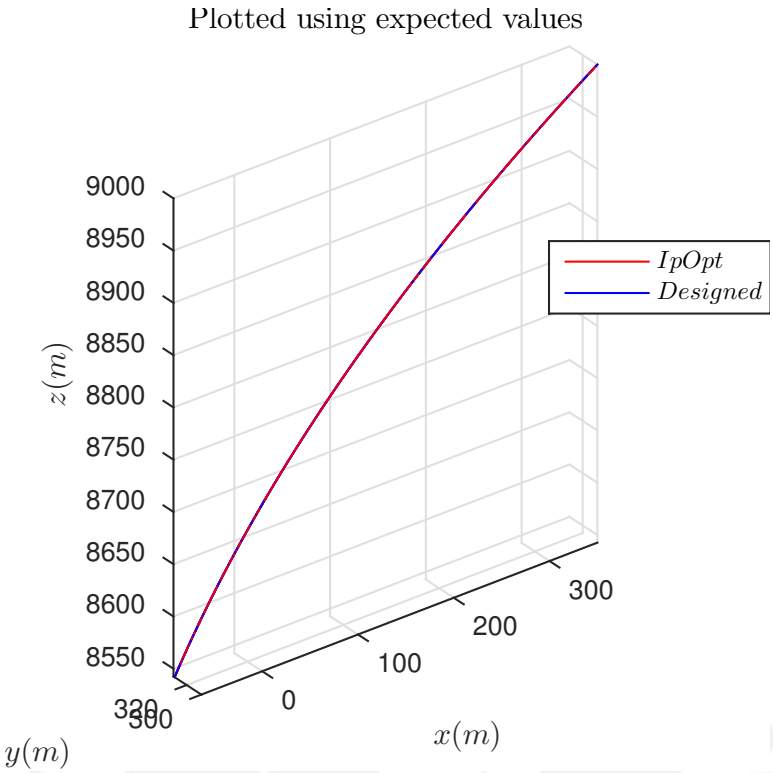
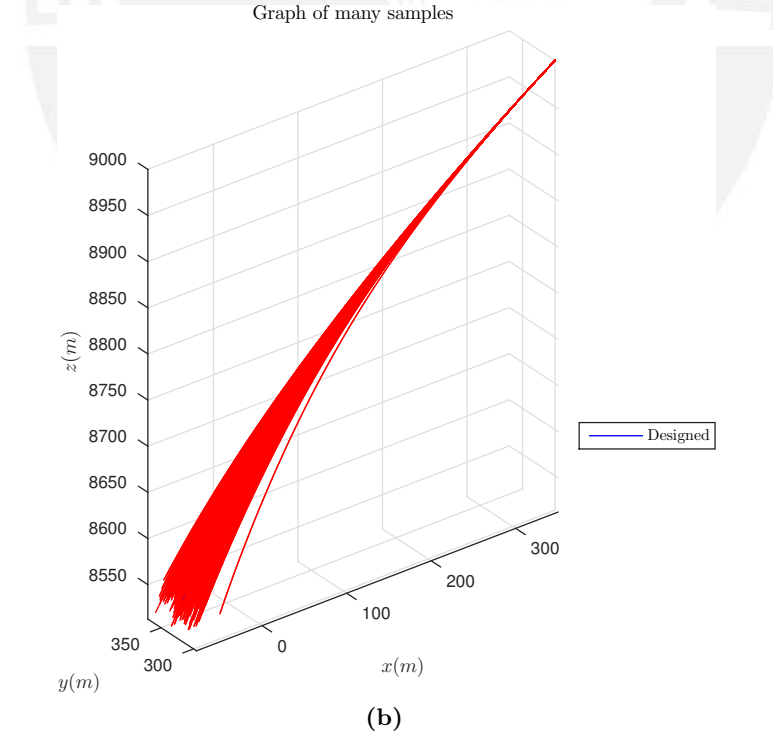


Figure 6.16 – Graph of control variables computed by IpOpt. A graph of the value of the objective function in each iteration is also shown. Note that it is only possible to iterate until time $t \approx 500$.

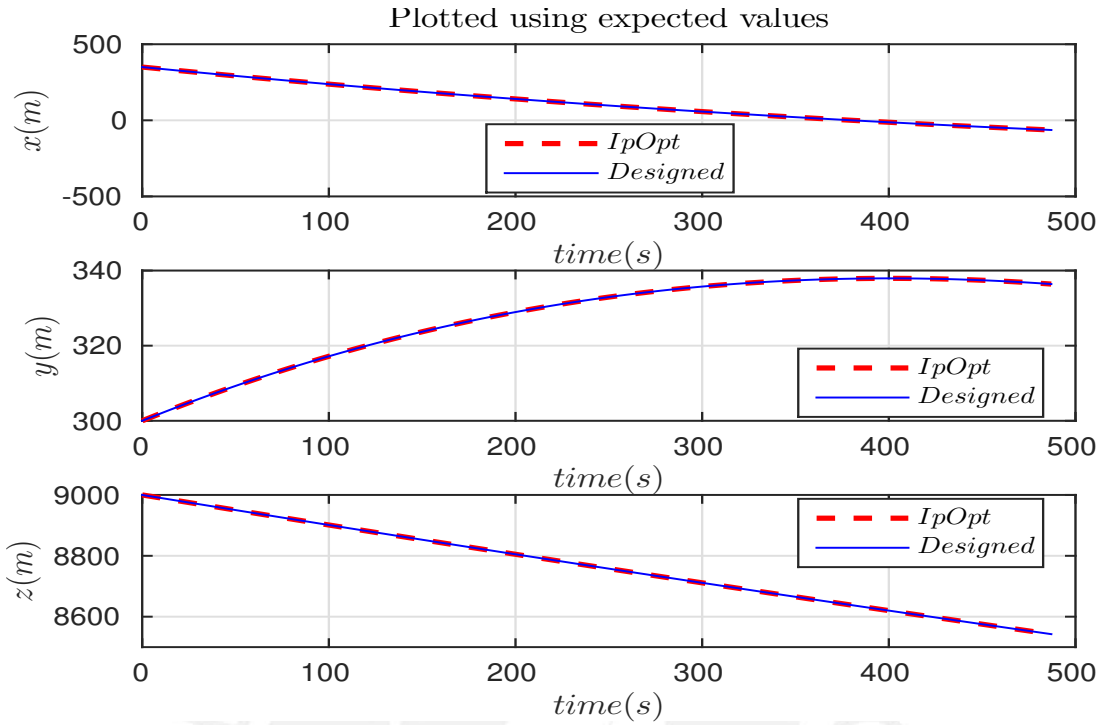


(a)

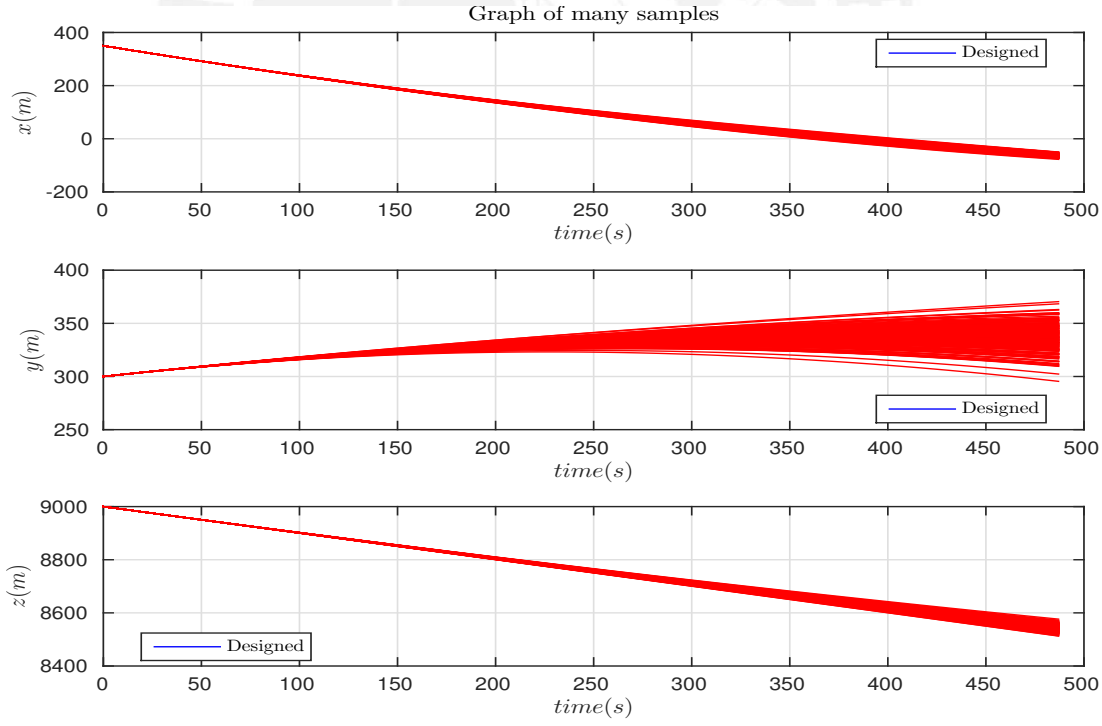


(b)

Figure 6.17 – Trajectory calculated by IpOpt and trajectory designed. Note that it is only possible to iterate until time $t \approx 500$.

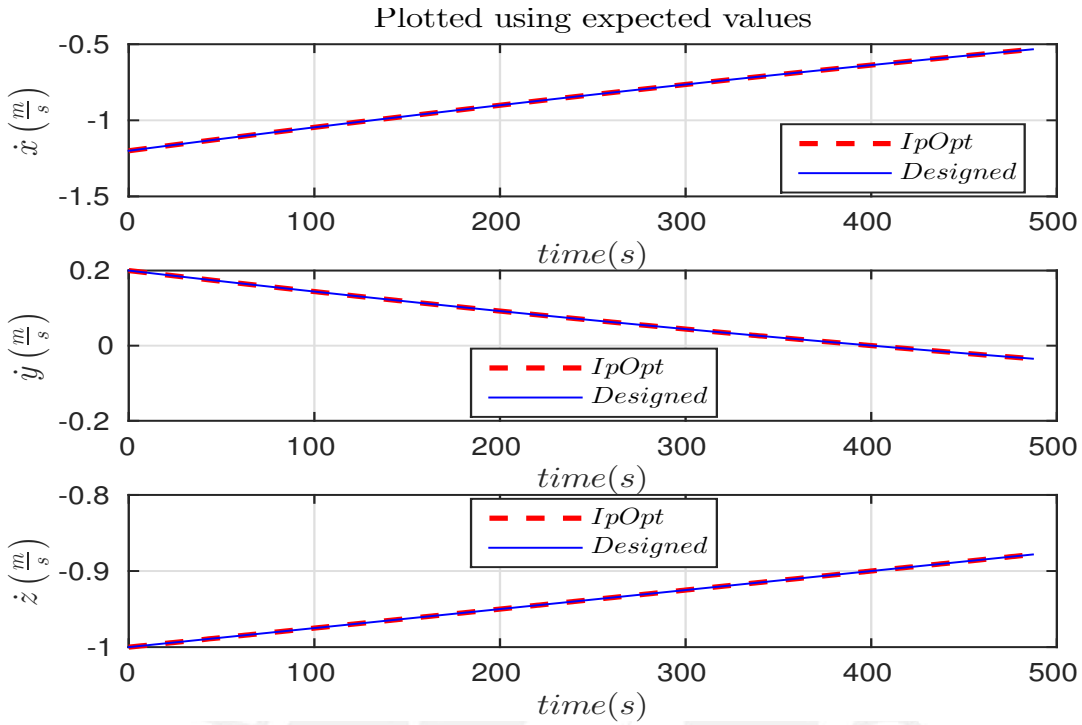


(a)

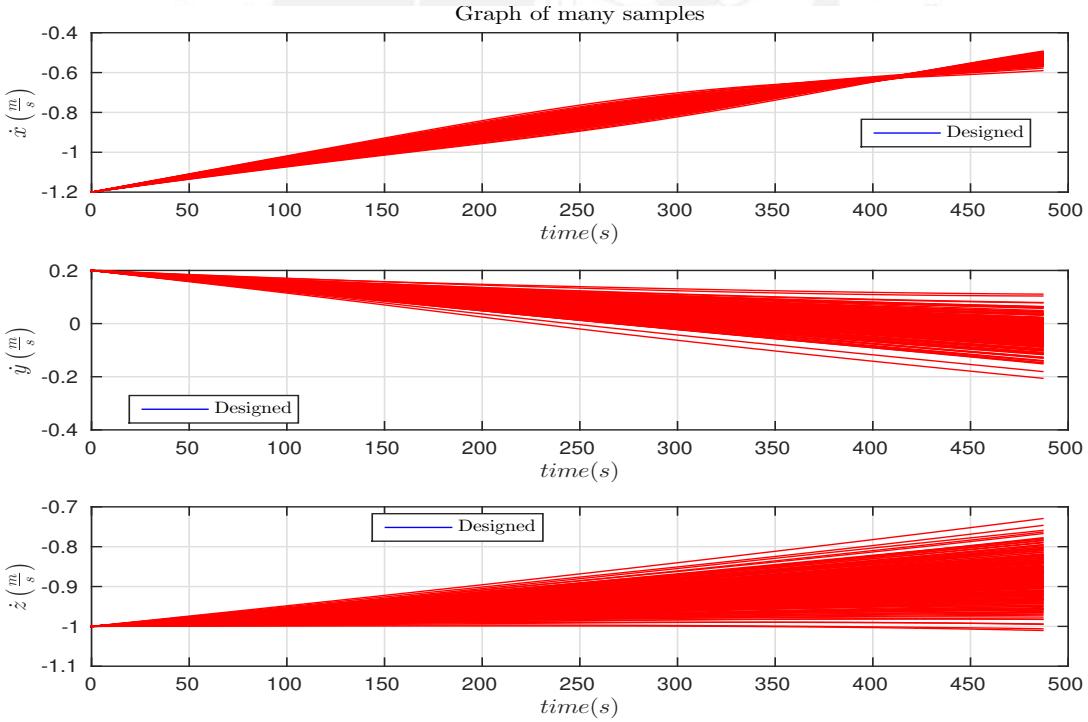


(b)

Figure 6.18 – Tracking of the position.



(a) This graph shows that the velocities calculated by the IpOpt follow with an error almost imperceptible to the speeds designed.



(b)

Figure 6.19 – Tacking of the velocities.

For a $\tau_{\text{Inner}} = 0.4$, the time that IpOpt invests to solve the optimization problem, is in the interval $[1.636, 1.968]$ seconds, for each iteration. In Figures 6.20 to 6.23 the data obtained with the IpOpt is shown in visual form.

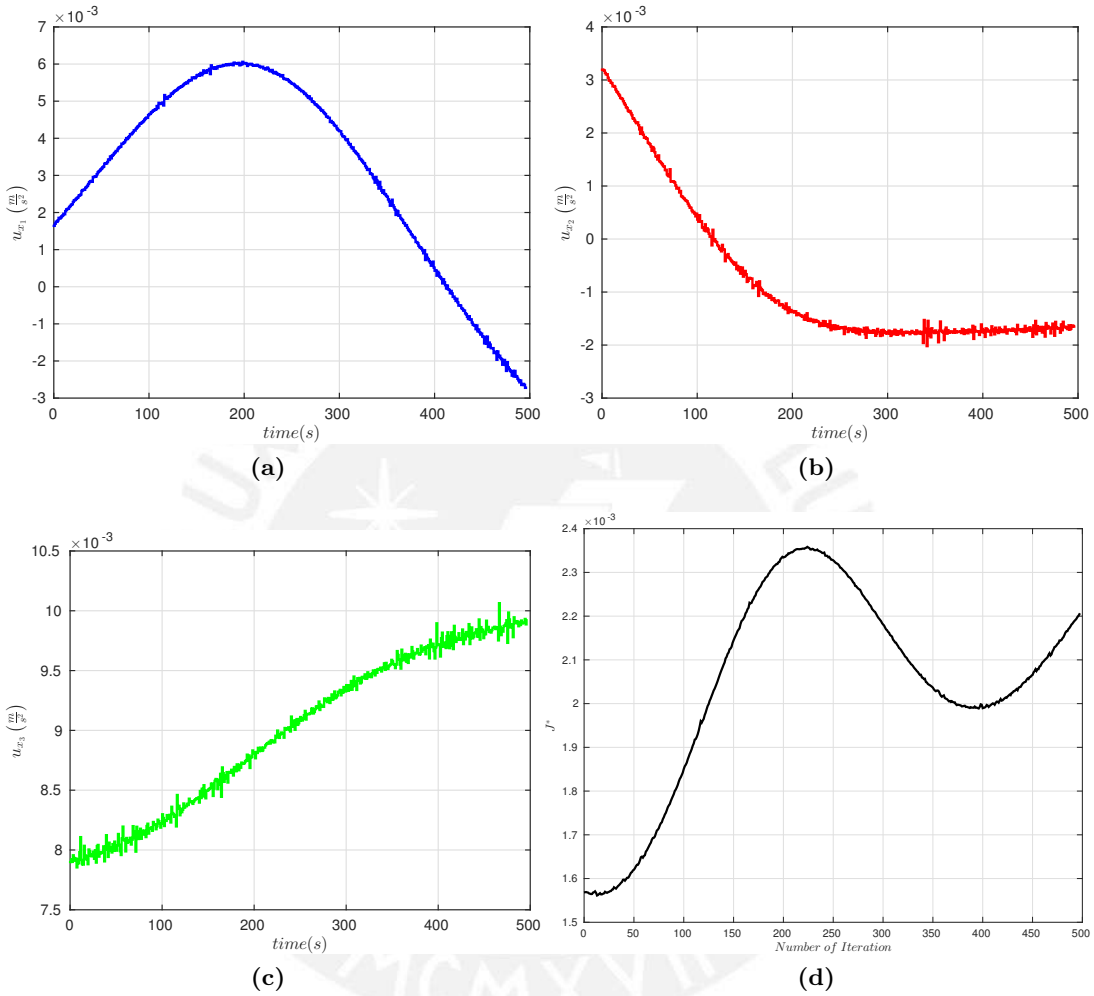
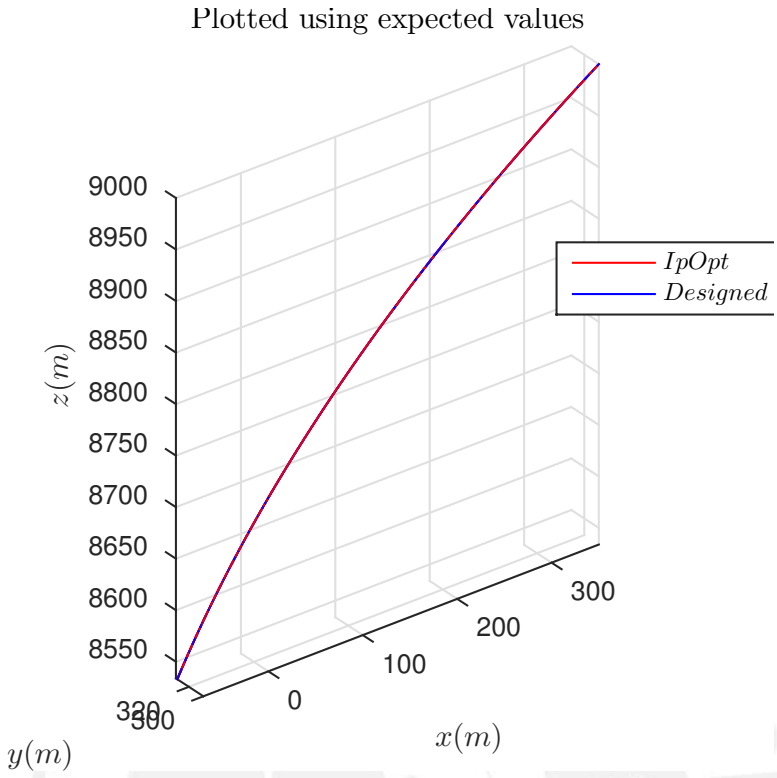
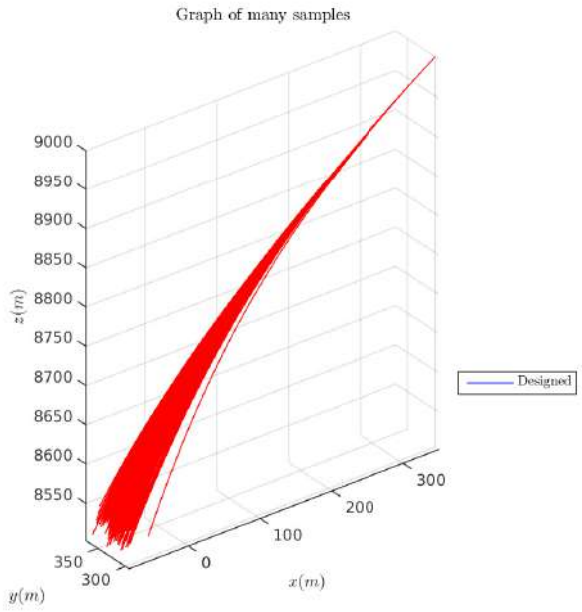


Figure 6.20 – Graph of control variables computed by IpOpt. A graph of the value of the objective function in each iteration is also shown. Note that it is only possible to iterate until time $t \approx 500$.

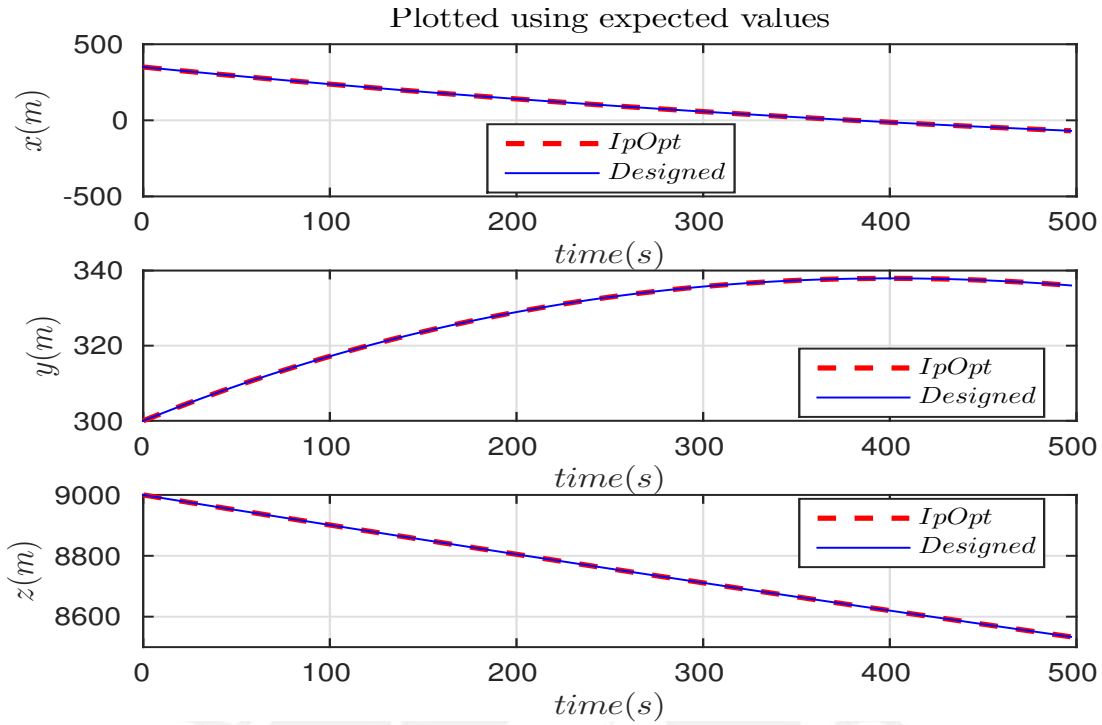


(a)

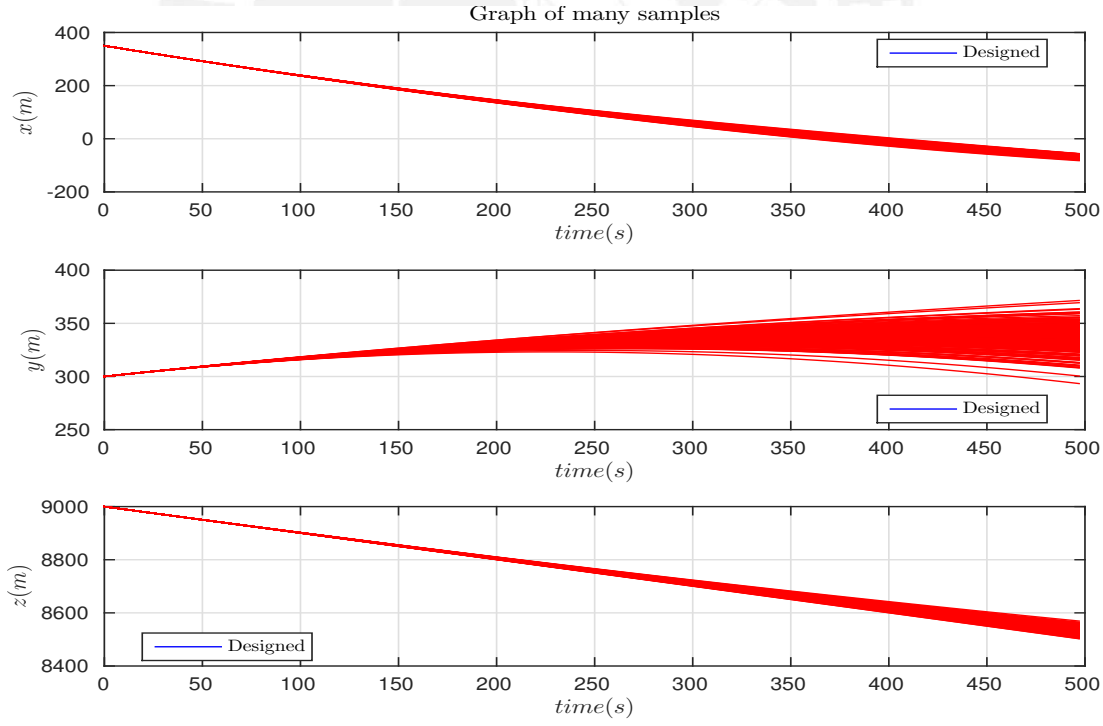


(b)

Figure 6.21 – Trajectory calculated by IpOpt and trajectory designed. Note that it is only possible to iterate until time $t \approx 500$.

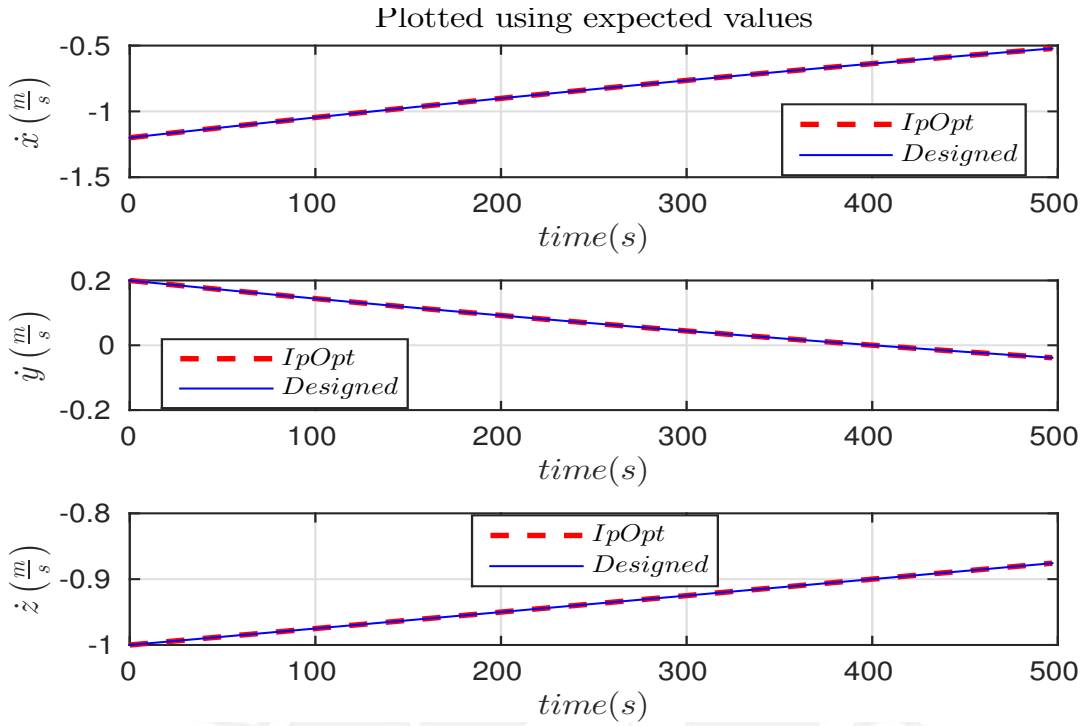


(a)

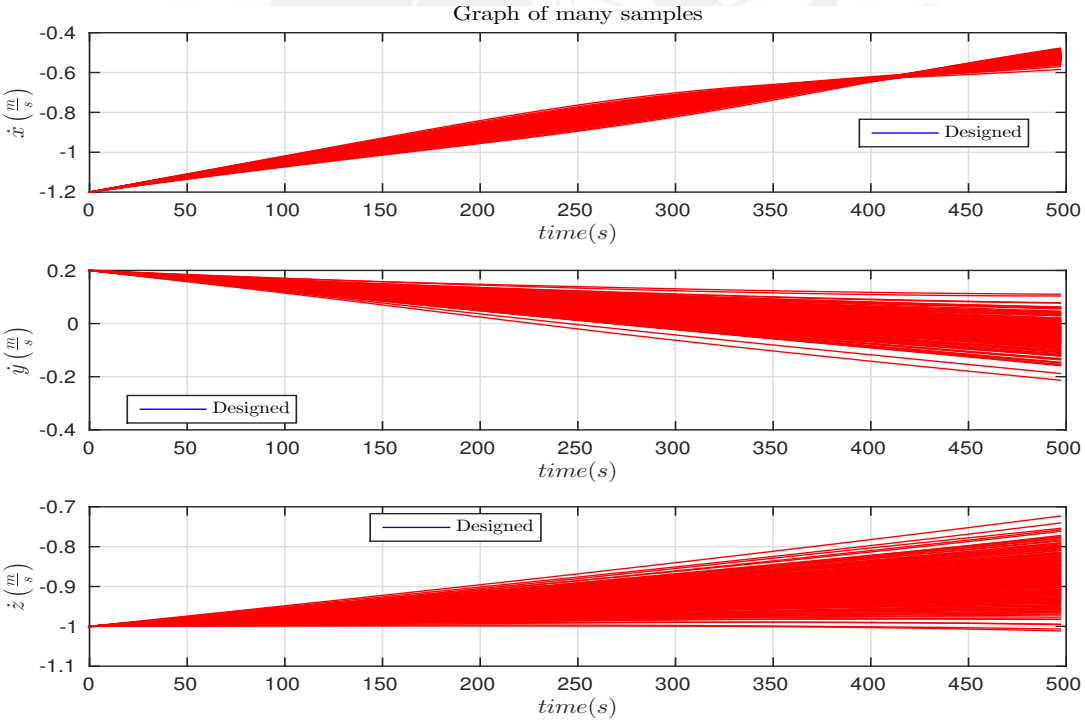


(b)

Figure 6.22 – Tracking of the position.



(a) This graph shows that the velocities calculated by the IpOpt follow with an error almost imperceptible to the speeds designed.



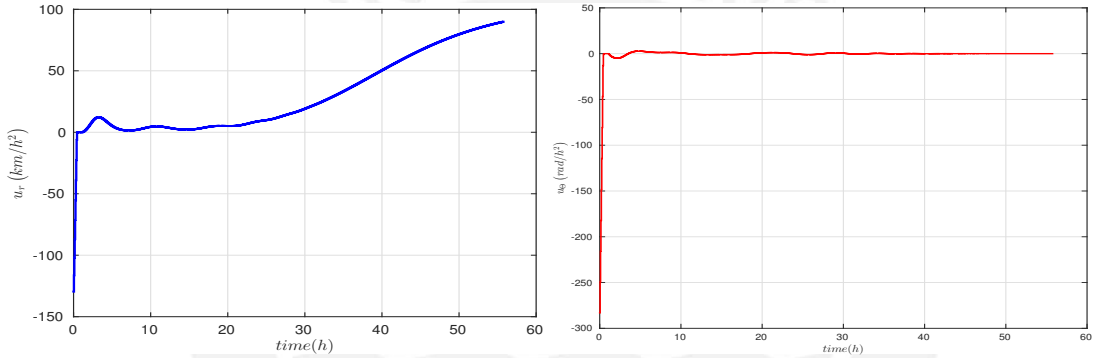
(b)

Figure 6.23 – Tacking of the velocities.

Based on the theory of Inner and Outer methods (Section 3.3.3). If $\tau \to 0^+$, then it is to be expected that the $\boldsymbol{u}^*$, J^* of both the $\text{NLP}_\tau^{\text{Inner}}$, $\text{NLP}_\tau^{\text{outer}}$, are the same. There has to be a trend.
Looking at Figures 6.12, 6.16 and 6.20 it is observed that there is no trend in the $\boldsymbol{u}^*$, J^*, in fact, there are no appreciable changes between those of one graph with respect to another.

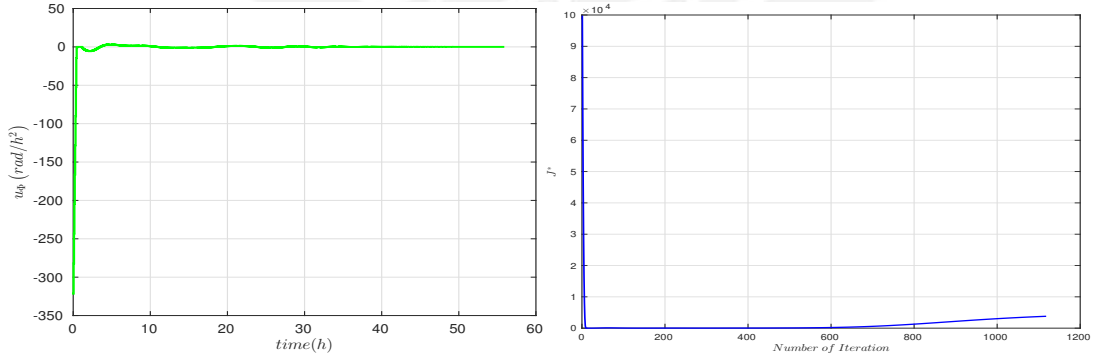
6.2.2 Guidance law in three dimensions for a soft landing on a celestial body

This case was solved using the stochastic MPC approach, explained in Section 5.2, also the equations of Section 5.2.2, and the data of Table 4.5.
We attempted to solve the deterministic NLP_τ^{Outer}, with different sets of parameters $m_1,\ m_2,\ p_1,\ p_2,\ p_3,\ a,\ b,\ \alpha$, without any success.
However, it was possible to solve the deterministic NLP_τ^{Inner}, with the parameters: $m_1 = 0.1,\ m_2 = 0.1,\ p_1 = 0.08,\ p_2 = 5,\ p_3 = 1,\ a = 1,\ b = 3,\ \alpha = 0.95$
In addition, the total number of samples used was $M = 10000$, the finite time horizon was $N = 10$, the Heun discretization method was used, the sampling time was $\Delta t = 0.05h$. For a $\tau_{\text{Inner}} = 0.8$, the time that IpOpt invests to solve the optimization problem, is in the interval $[2.336, 2.932]$ seconds, for each iteration. In Figures 6.24 to 6.27, the data obtained with the IpOpt is shown in visual form.



(a) It is observed that the effort of the controller increases in time, this is because as the Spacecraft approaches the celestial body, the gravitational field becomes more intense, which translates into a greater effort of the controller in component r, so that the landing is smooth.

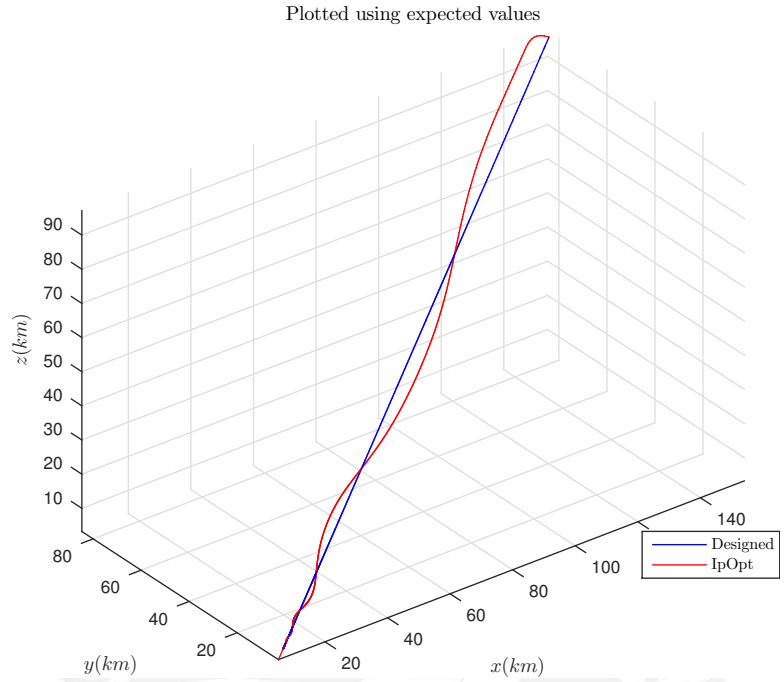
(b) The controller effort in this component is almost constant in the value of 0, with imperceptible oscillations (it is necessary to zoom), this steady behavior, because the gravitational field only acts in the component r, but not in Θ or Φ.



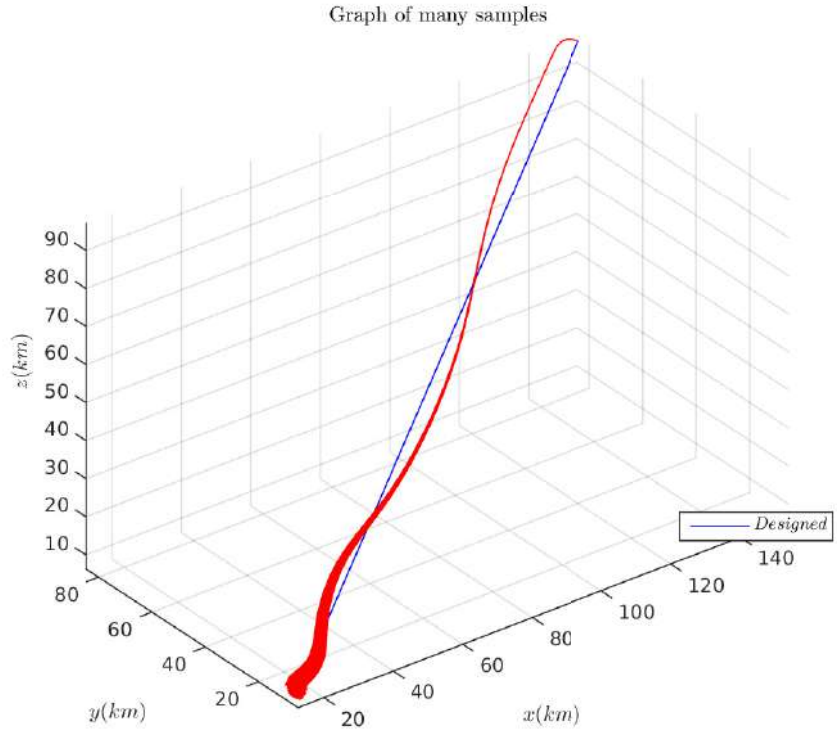
(c) The analysis of this graph is analogous to that of Figure 6.24b.

(d) In this graph the objective function takes after a few iterations a value of almost 0, but then begins to increase slightly because as the Spacecraft approaches the celestial body, the intensity of the gravitational field in the component r increases, see Figure 6.24a.

Figure 6.24 – Control variables calculated by IpOpt, using the Heun discretization method, a finite time horizon of $N = 10$, $\Delta t = 0.05h$, for Case 2.

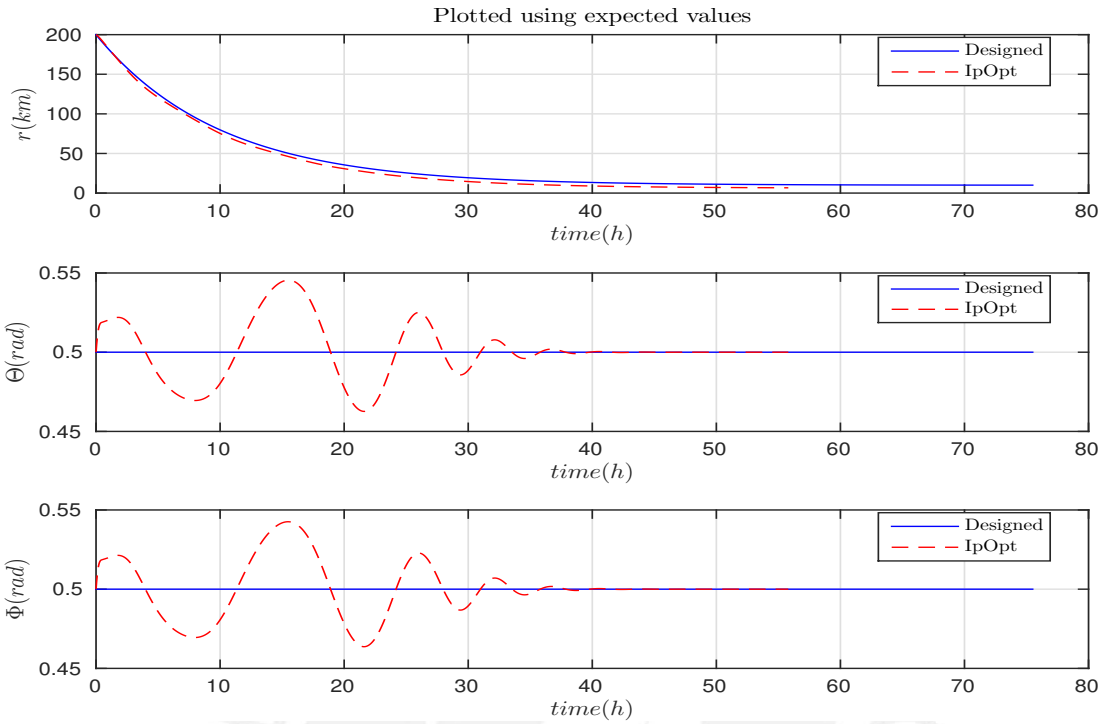


(a)

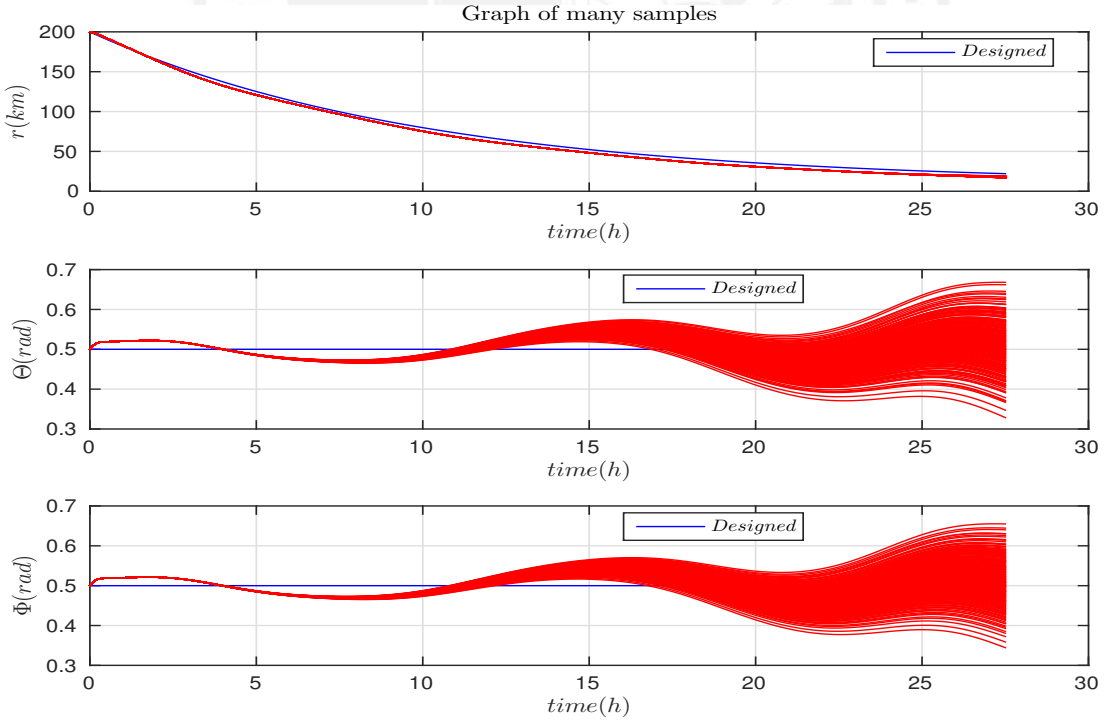


(b)

Figure 6.25 – Trajectory designed and trajectory calculated by IpOpt, for Case 2, with a finite time horizon $N = 10$, sampling time $\Delta t = 0.05h$, and using Heun discretization method.

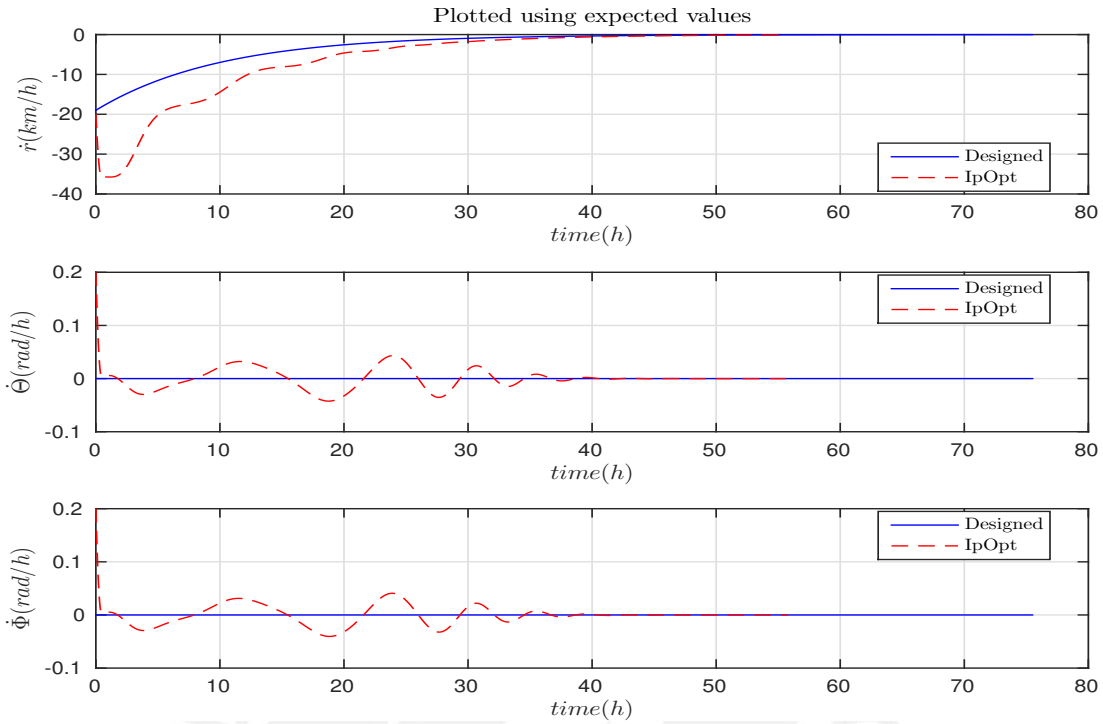


(a) Tracking the trajectory designed over time in each of its components, it is appreciated that until the time instant $t \approx 40h$, there is an error in the tracking of the designed path, that is relatively small, in each component.

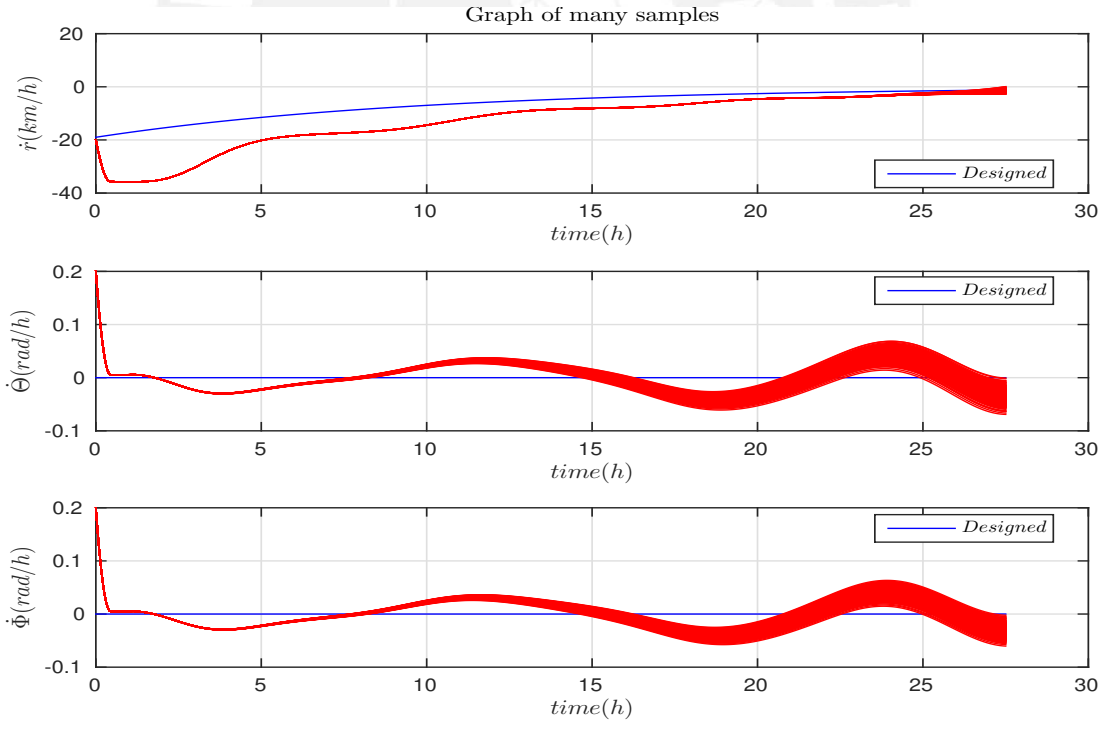


(b)

Figure 6.26 – Calculated using the Heun discretization method, a finite time horizon of $N = 10$, $\Delta t = 0.05h$, for Case 2.



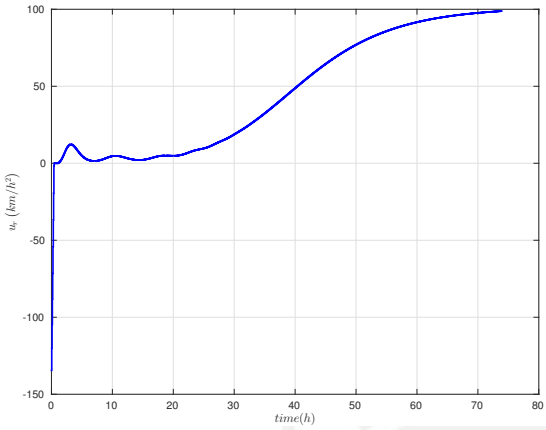
(a) In this graph it is seen how the speed calculated by the IpOpt follows the designed speed, in its three components.



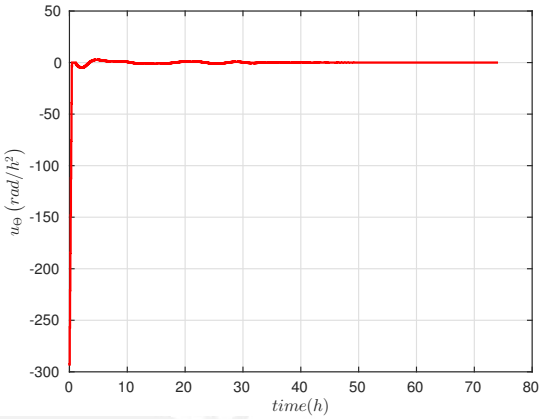
(b)

Figure 6.27 – Calculated using the Heun discretization method, a finite time horizon of $N = 10$, $\Delta t = 0.05h$, for Case 2.

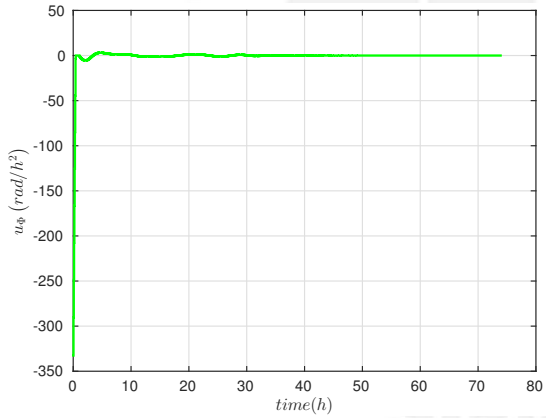
For a $\tau_{\text{Inner}} = 0.9$, the time that IpOpt invests to solve the optimization problem, is in the interval $[2.396, 3.116]$ seconds, for each iteration. In Figures 6.28 to 6.31, the data obtained with the IpOpt is shown in visual form.



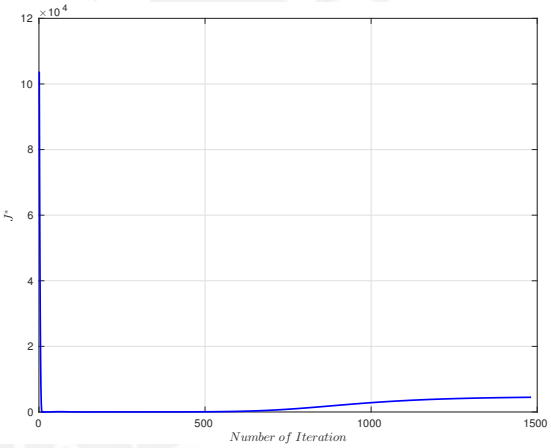
(a) It is observed that the effort of the controller increases in time, this is because as the Spacecraft approaches the celestial body, the gravitational field becomes more intense, which translates into a greater effort of the controller in component r, so that the landing is smooth.



(b) The controller effort in this component is almost constant in the value of 0, with imperceptible oscillations (it is necessary to zoom), this steady behavior, because the gravitational field only acts in the component r, but not in Θ or Φ.

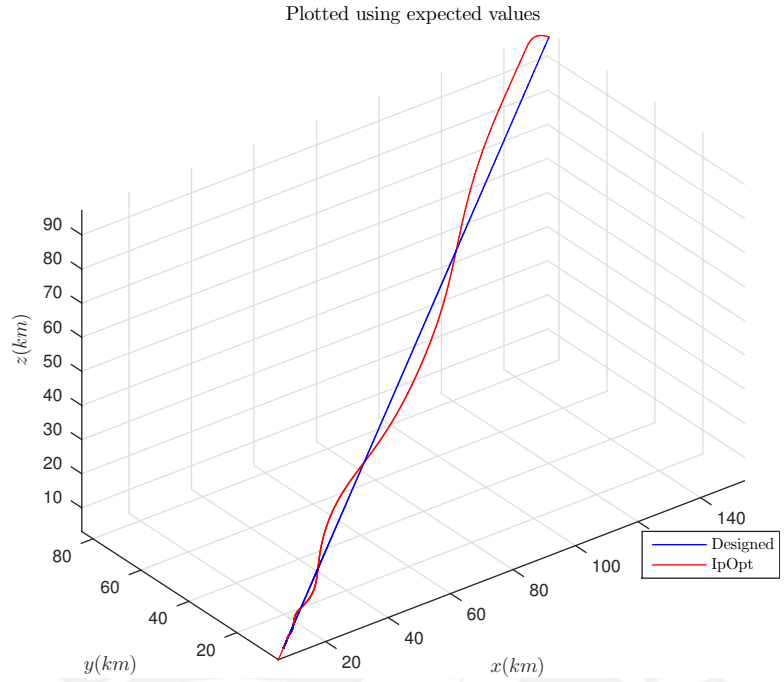


(c) The analysis of this graph is analogous to that of Figure 6.28b.

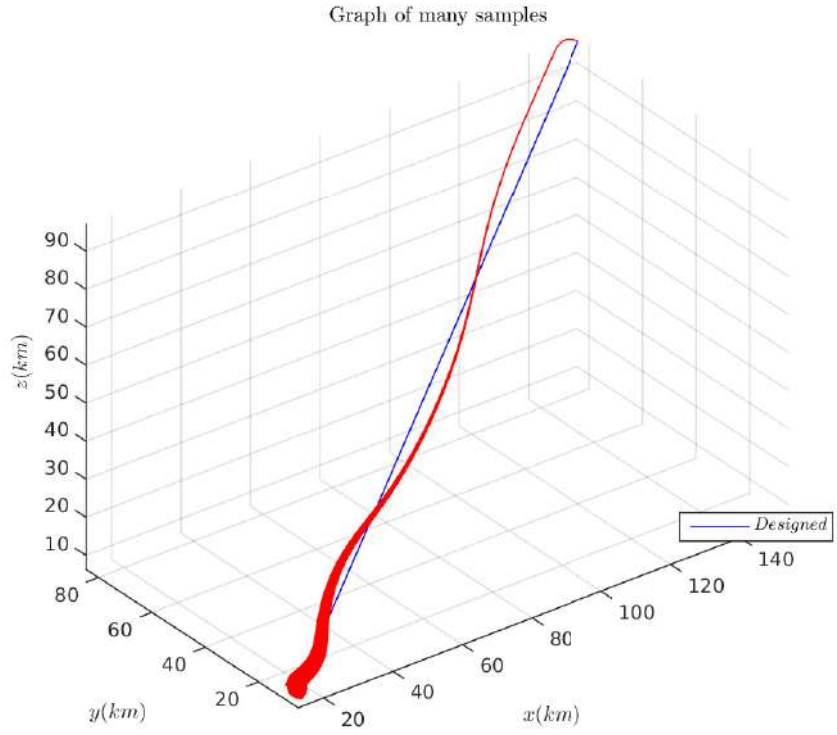


(d) In this graph the objective function takes after a few iterations a value of almost 0, but then begins to increase slightly because as the Spacecraft approaches the celestial body, the intensity of the gravitational field in the component r increases, see Figure 6.28a.

Figure 6.28 – Control variables calculated by IpOpt, using the Heun discretization method, a finite time horizon of $N = 10$, $\Delta t = 0.05h$, for Case 2.

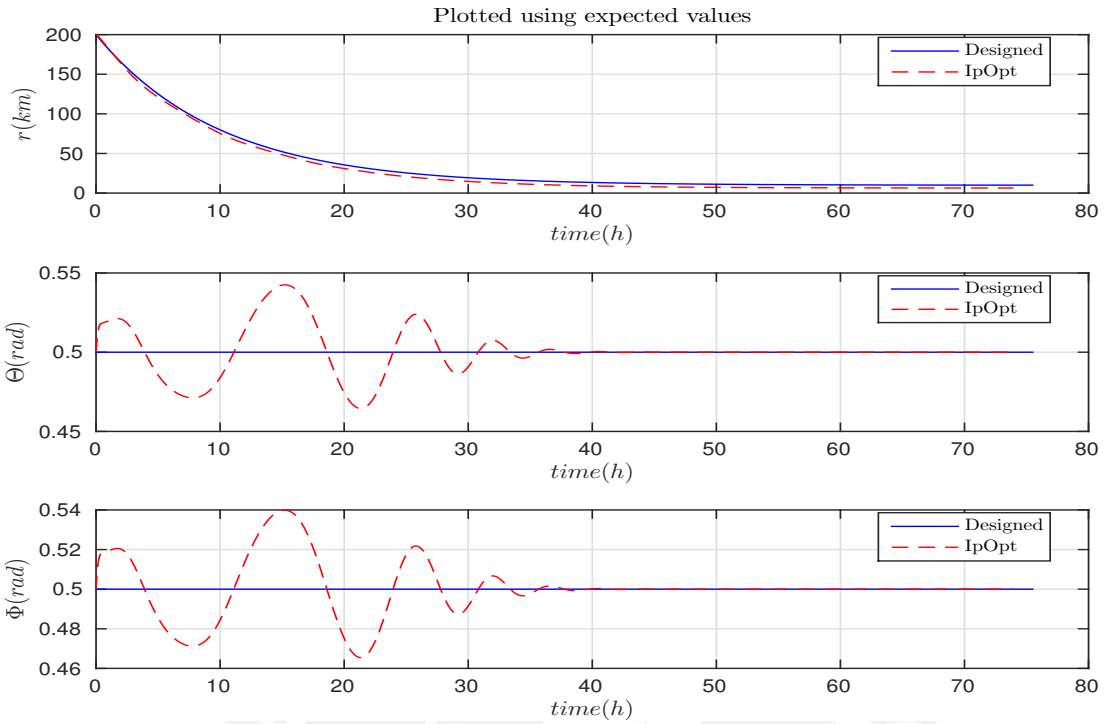


(a)

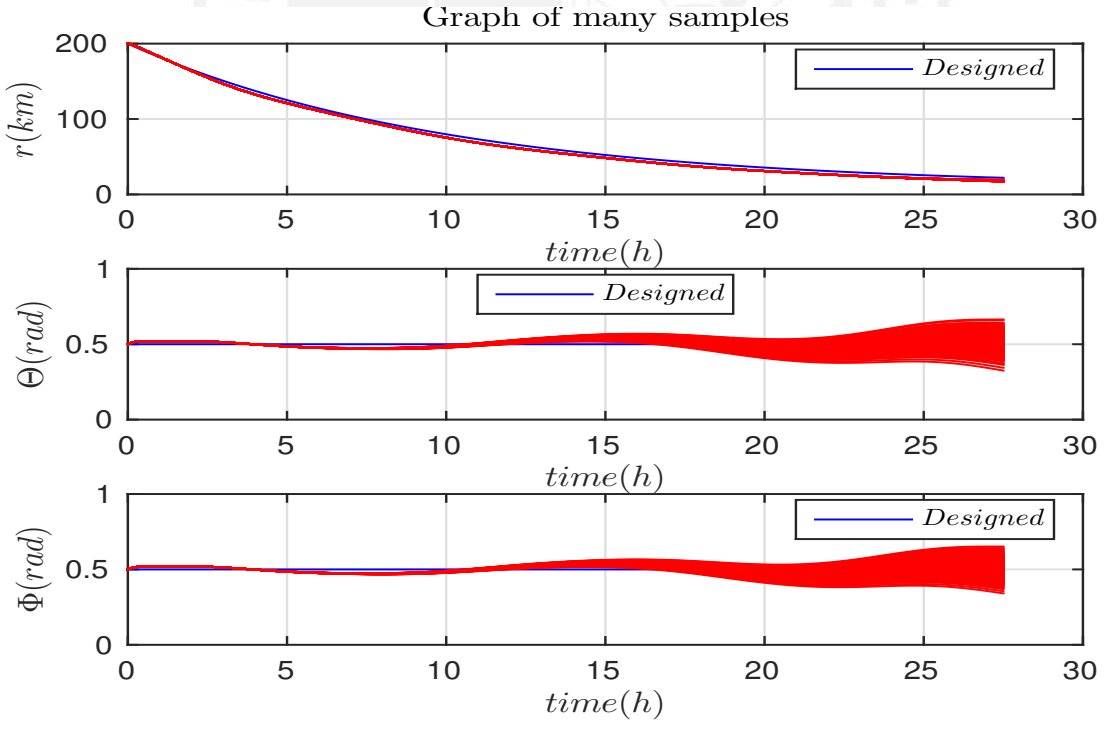


(b)

Figure 6.29 – Trajectory designed and trajectory calculated by IpOpt, for Case 2, with a finite time horizon $N = 10$, sampling time $\Delta t = 0.05h$, and using Heun discretization method.

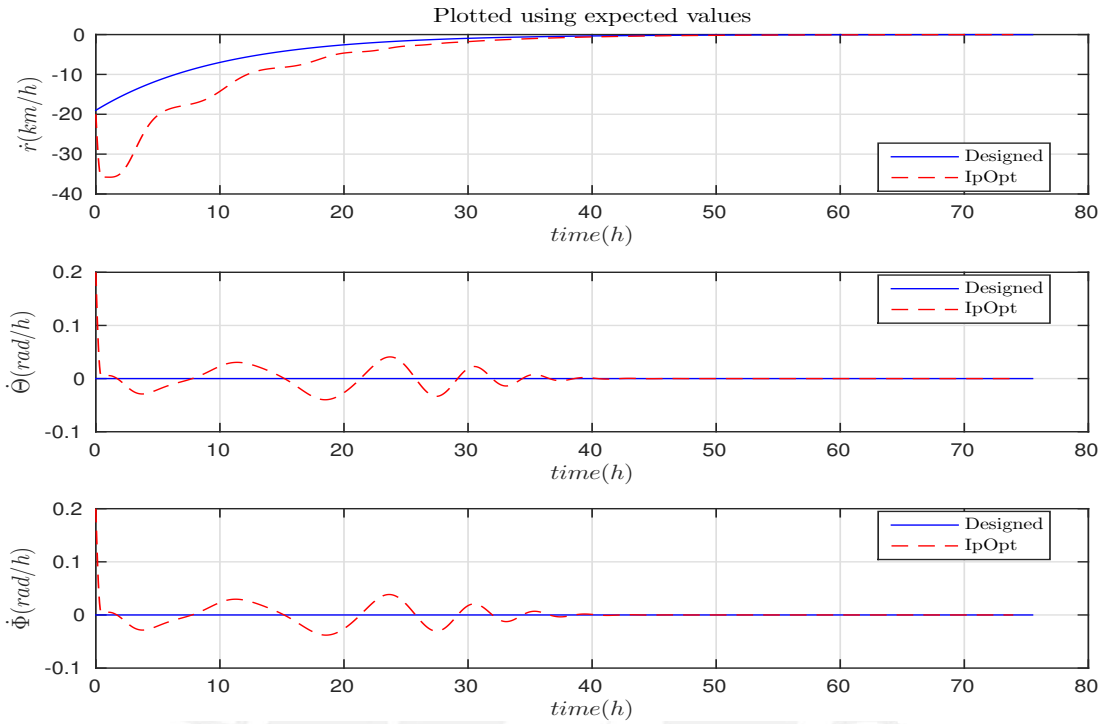


(a) Tracking the trajectory designed over time in each of its components, it is appreciated that until the time instant $t \approx 40h$, there is an error in the tracking of the designed path, that is relatively small, in each component.

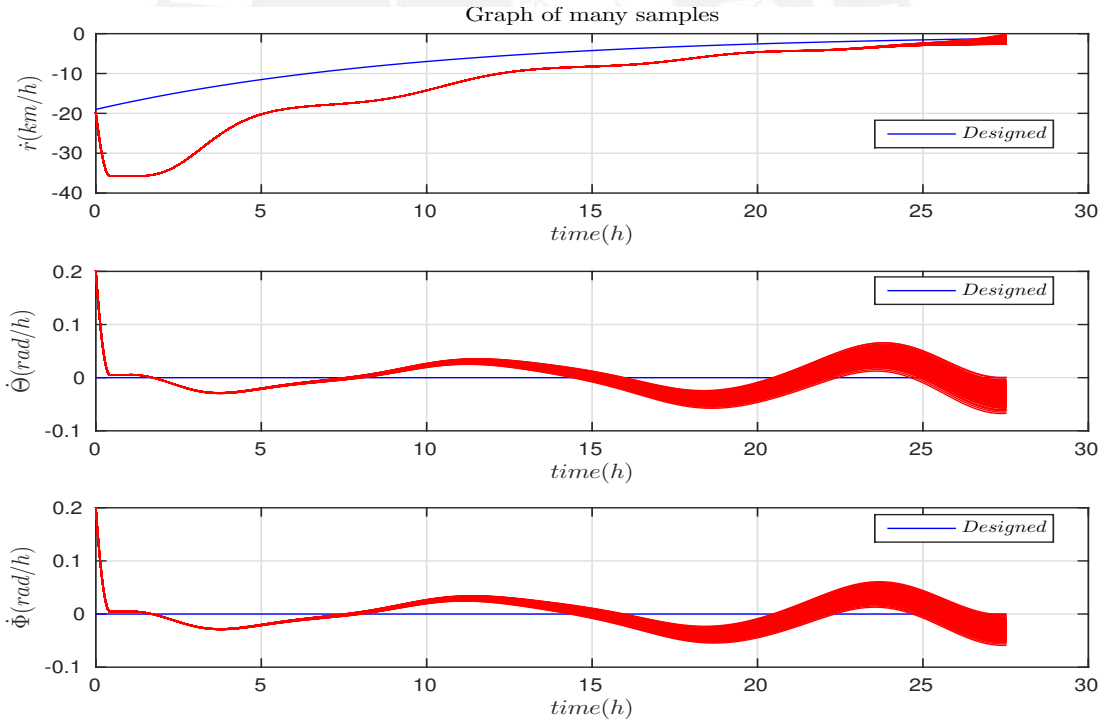


(b)

Figure 6.30 – Calculated using the Heun discretization method, a finite time horizon of $N = 10$, $\Delta t = 0.05h$, for Case 2.



(a) In this graph it is seen how the speed calculated by the IpOpt follows the designed speed, in its three components.



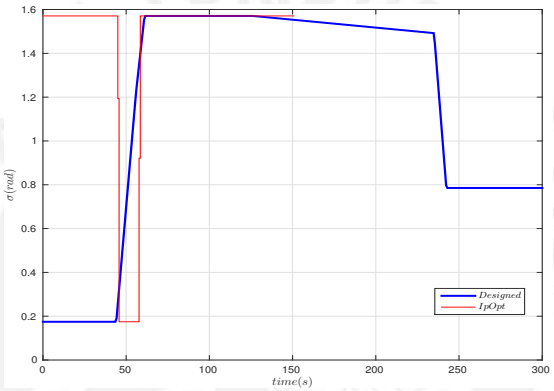
(b)

Figure 6.31 – Calculated using the Heun discretization method, a finite time horizon of $N = 10$, $\Delta t = 0.05h$, for Case 2.

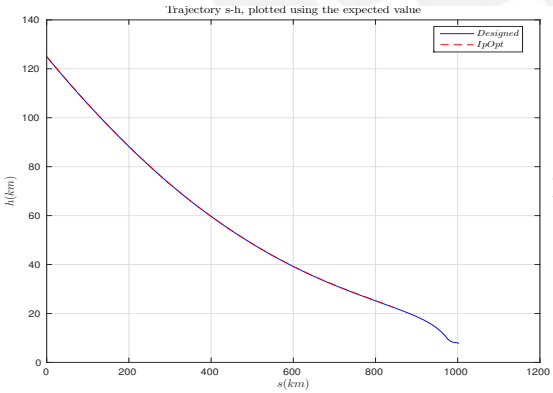
We tried to do the simulation with smaller τ , but it was not successful, the IpOpt software iterates a few times and then it aborts, or it does not manage to iterate even once.
The recurring message that IpOpt displays when aborting is "Restoration Failed".
Contrasting the results obtained with $\tau_{\text{Inner}} = 0.8$, $\tau_{\text{Inner}} = 0.9$, it is observed that there is a slight change in the control variable u_r, and J^*.
It remains as future work, to investigate why this case study does not work when the chance constraints are approximated by the Outer approximation approach.

6.2.3 Longitudinal model of lifting entry of a Mars Lander

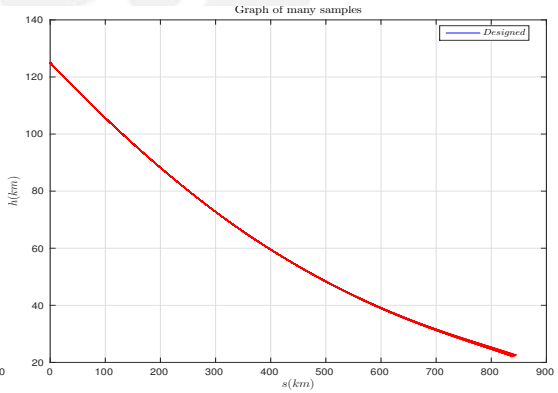
This case study was solved with the stochastic MPC approach, discussed in Section 5.2, we also used the equations in Section 5.2.3 and the data in Table 4.7.
The deterministic $\text{NLP}_\tau^{\text{Inner}}$, $\text{NLP}_\tau^{\text{Outer}}$ were solved for $\tau = 0.5,\ 0.25,\ 0.0625$, the parameters used for the simulation were: $m_1 = 0.01,\ m_2 = m_1,\ p_1 = 3.125 \times 10^{-6},\ p_2 = 100,\ p_3 = 0.25,\ a = 2,\ b = 0.1,\ \alpha = 0.97$.
Additionally, we used a total number of samples $M = 10000$, the finite time horizon was $N = 15$, the Runge-Kutta 4th order method was used for the discretization, the sampling time was $\Delta t = 0.75s$.
For $\tau_{\text{Inner}} = 0.5$, IpOpt solves the optimization problem, for each iteration, in a time that oscillates in the interval $[3.144, 3.476]$ seconds.
In Figures 6.32 to 6.34, the data obtained with the IpOpt is shown in visual form.



(a) It can be seen in the figure that the IpOpt data has been graphed to the middle of the time horizon ($T = 300s$). The reason is that IpOpt only manages to iterate until half of the time horizon T, at that time $t \approx 150s$, IpOpt shows the message: " Converged to a point of local infeasibility. Problem may be infeasible ".

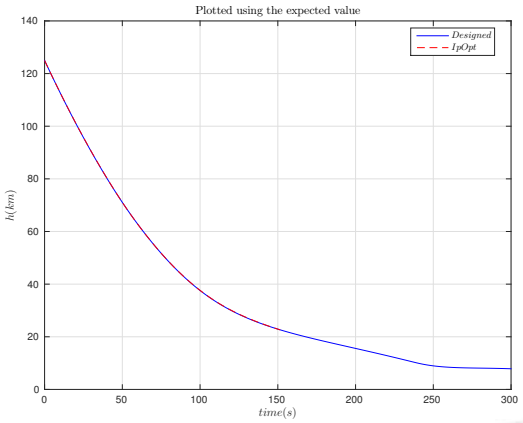


(b) Trajectory designed in the $s - h$ plane, also the trajectory calculated by IpOpt.

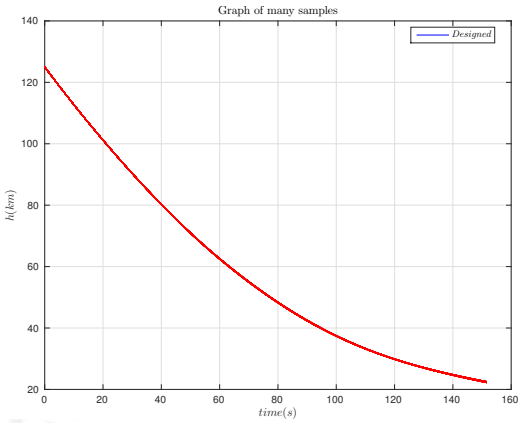


(c)

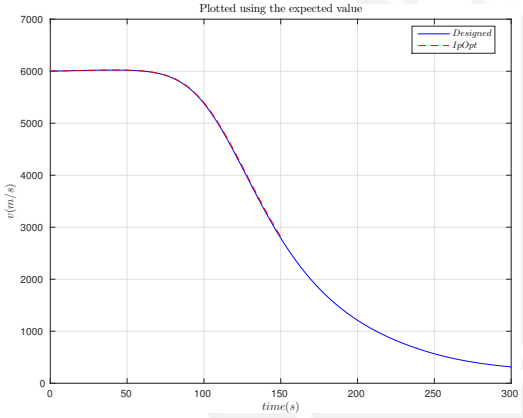
Figure 6.32 – Graph of σ, graph of the trajectory designed in the $s - h$ plane.



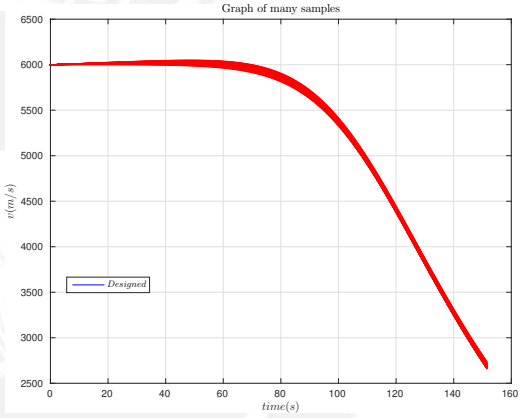
(a) The controller does a tracking of this state with almost imperceptible error.



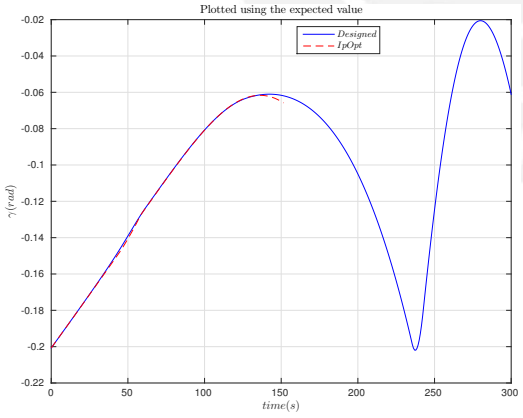
(b)



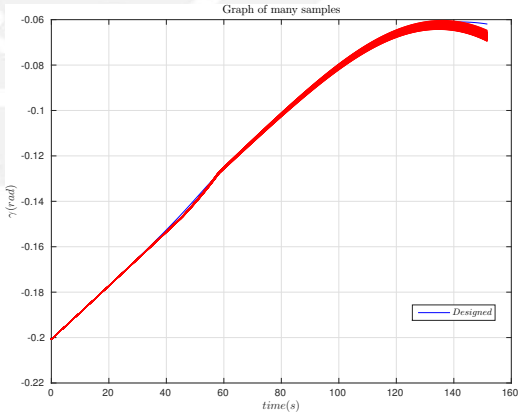
(c) The controller does a tracking of this state with almost imperceptible error.



(d)

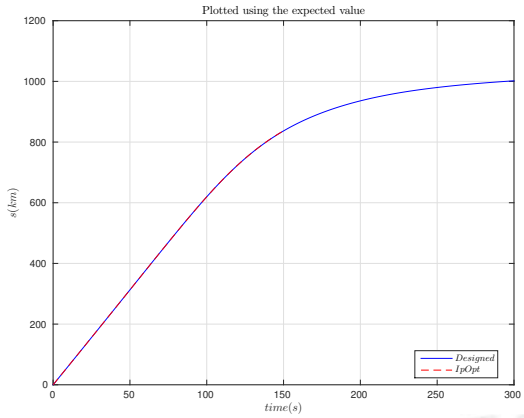


(e) It is observed that there is a slight difference between what was designed and what is calculated by IpOpt at time $t \approx 50s$. It is also seen that at time $t \approx 150s$, the difference becomes relatively large, it begins to increase.

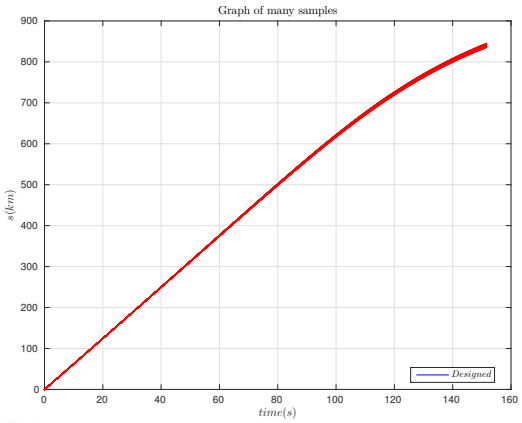


(f)

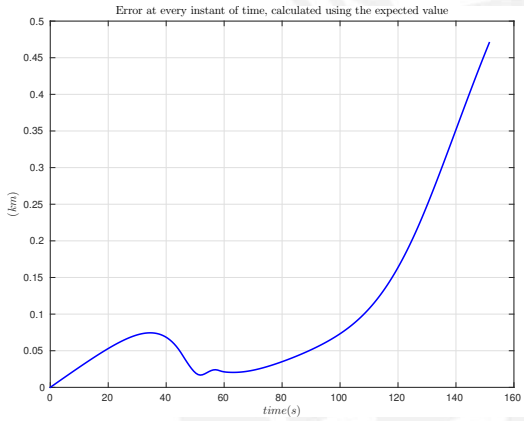
Figure 6.33 – Graphs of the tracking of the states $[h, v, \gamma]$.



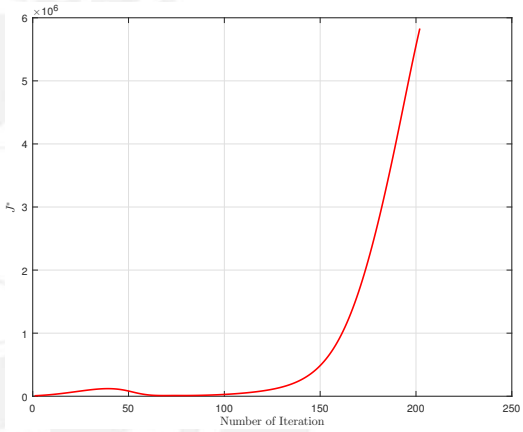
(a) The controller does a tracking of this state with almost imperceptible error.



(b)



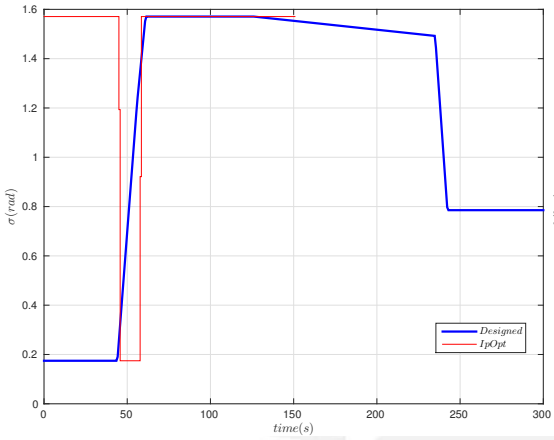
(c) Error at every instant of time.



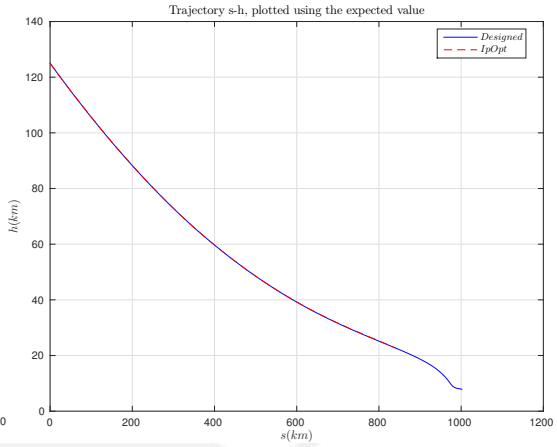
(d) The graph of the objective function has the same form as the error graph, Figure 6.34c, the interpretation is that as the Mars Lander is diverted from the designed path, the error increases, and consequently the controller will try harder to reduce said error of tracking.

Figure 6.34 – Graph of the tracking of the state s, also the Objective function.

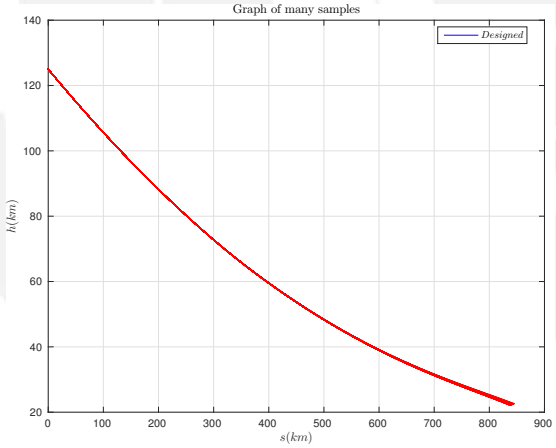
For $\tau_{\text{Outer}} = 0.5$, IpOpt solves the optimization problem, for each iteration, in a time that oscillates in the interval $[3.088, 3.380]$ seconds.
In Figures 6.35 to 6.37, the data obtained with the IpOpt is shown in visual form.



(a) It can be seen in the figure that the IpOpt data has been graphed to the middle of the time horizon ($T = 300s$). The reason is that IpOpt only manages to iterate until half of the time horizon T, at that time $t \approx 150s$, IpOpt shows the message: " Converged to a point of local infeasibility. Problem may be infeasible ".

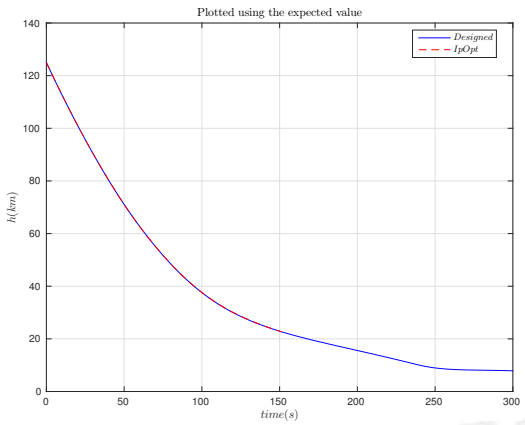


(b) Trajectory designed in the $s - h$ plane, also the trajectory calculated by IpOpt.

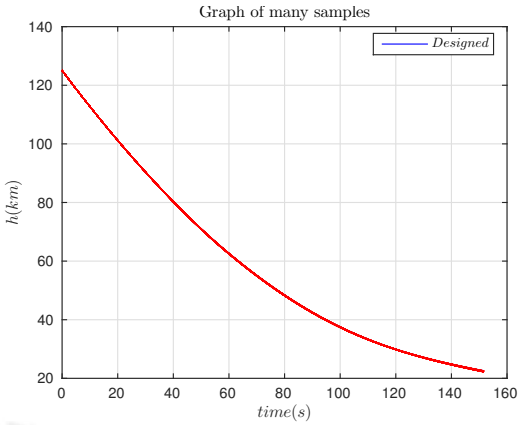


(c)

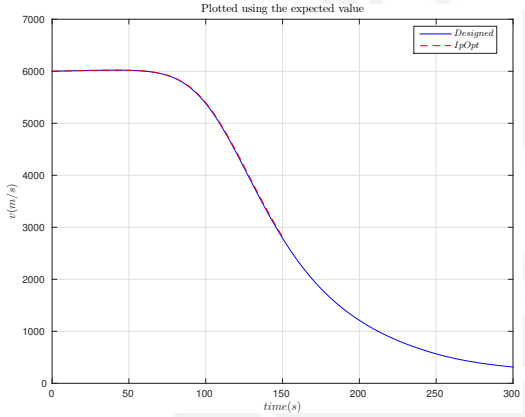
Figure 6.35 – Graph of σ, graph of the trajectory designed in the $s - h$ plane.



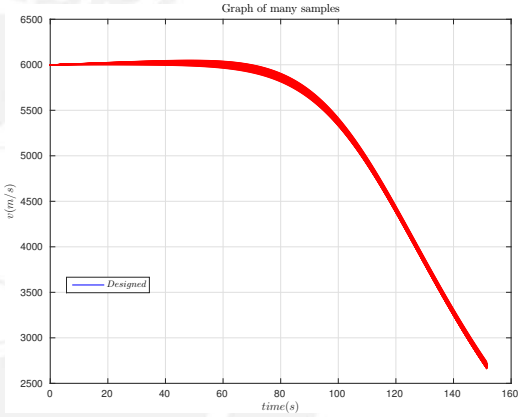
(a) The controller does a tracking of this state with almost imperceptible error.



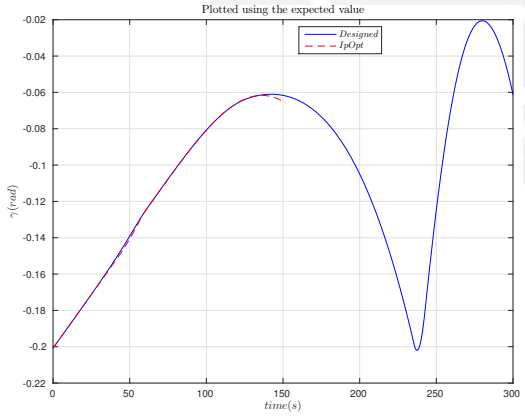
(b)



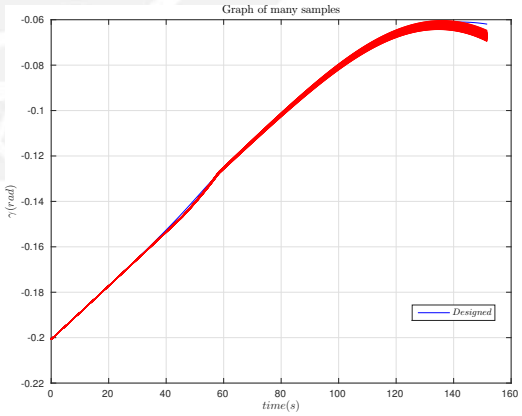
(c) The controller does a tracking of this state with almost imperceptible error.



(d)

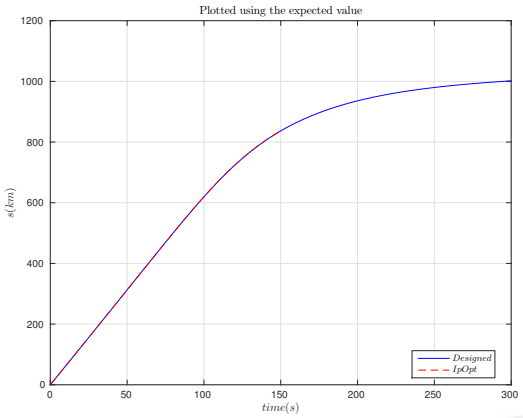


(e) It is observed that there is a slight difference between what was designed and what is calculated by IpOpt at time $t \approx 50s$. It is also seen that at time $t \approx 150s$, the difference becomes relatively large, it begins to increase.

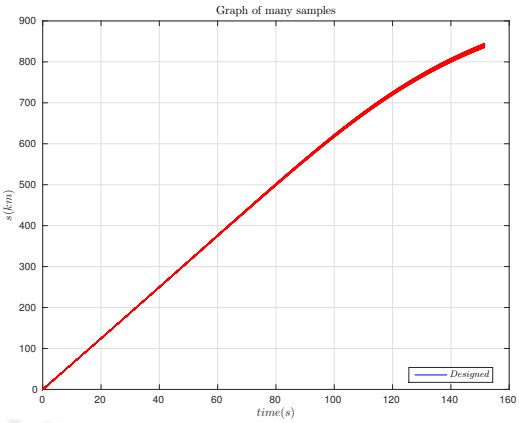


(f)

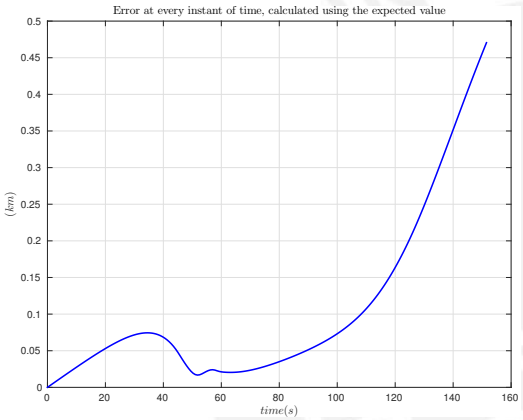
Figure 6.36 – Graphs of the tracking of the states $[h, v, \gamma]$.



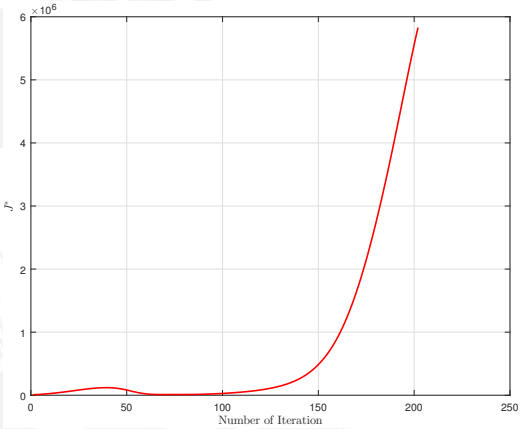
(a) The controller does a tracking of this state with almost imperceptible error.



(b)



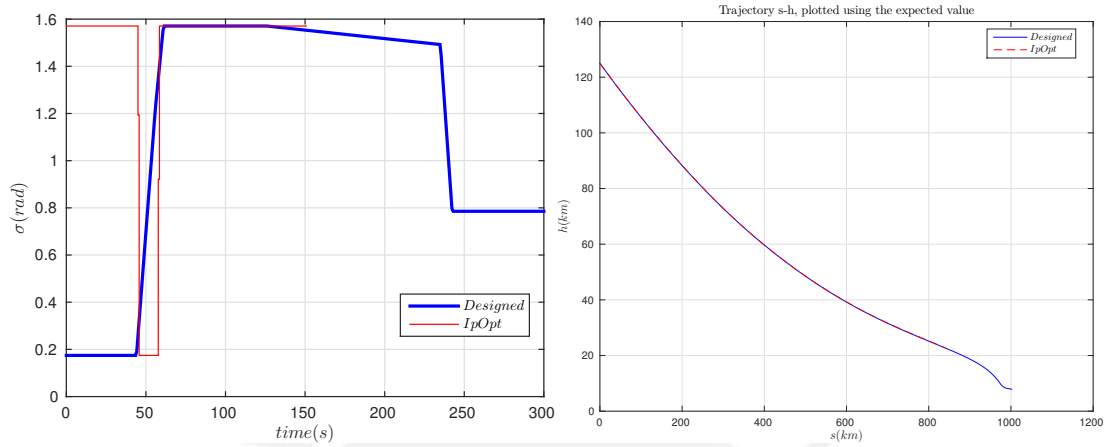
(c) Error at every instant of time.



(d) The graph of the objective function has the same form as the error graph, Figure 6.37c, the interpretation is that as the Mars Lander is diverted from the designed path, the error increases, and consequently the controller will try harder to reduce said error of tracking.

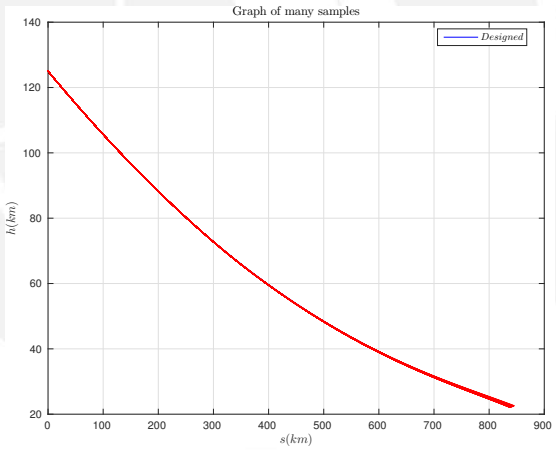
Figure 6.37 – Graph of the tracking of the state s, also the Objective function.

For $\tau_{\text{Inner}} = 0.25$, IpOpt solves the optimization problem, for each iteration, in a time that oscillates in the interval $[3.172, 3.692]$ seconds.
In Figures 6.38 to 6.40, the data obtained with the IpOpt is shown in visual form.



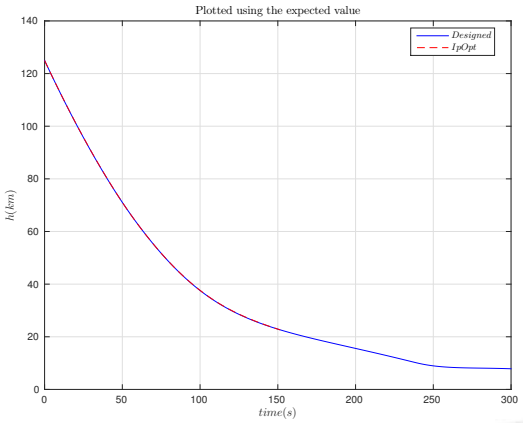
(a) It can be seen in the figure that the IpOpt data has been graphed to the middle of the time horizon ($T = 300s$). The reason is that IpOpt only manages to iterate until half of the time horizon T, at that time $t \approx 150s$, IpOpt shows the message: " Converged to a point of local infeasibility. Problem may be infeasible ".

(b) Trajectory designed in the $s - h$ plane, also the trajectory calculated by IpOpt.

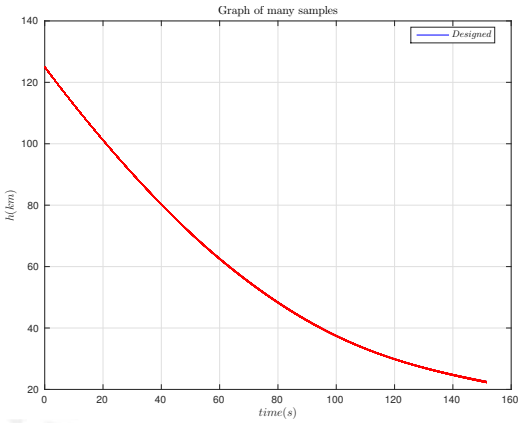


(c)

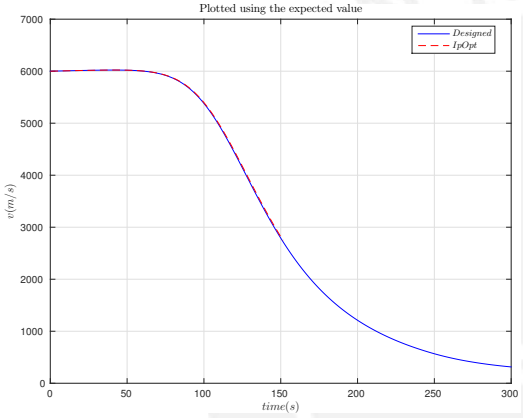
Figure 6.38 – Graph of σ, graph of the trajectory designed in the $s - h$ plane.



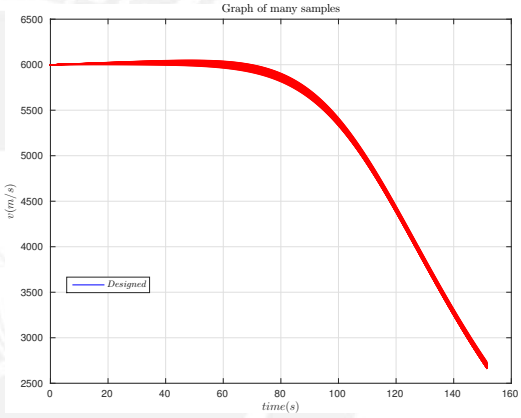
(a) The controller does a tracking of this state with almost imperceptible error.



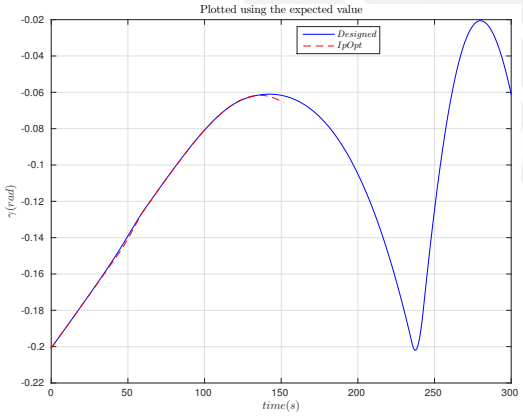
(b)



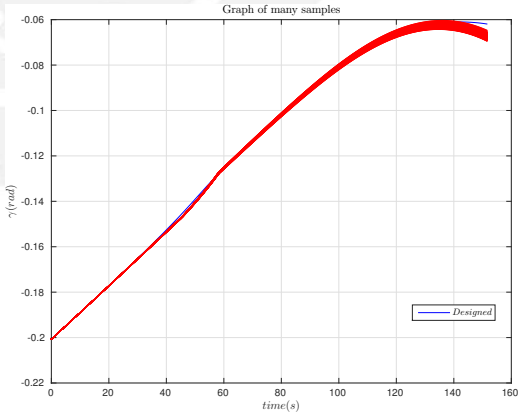
(c) The controller does a tracking of this state with almost imperceptible error.



(d)

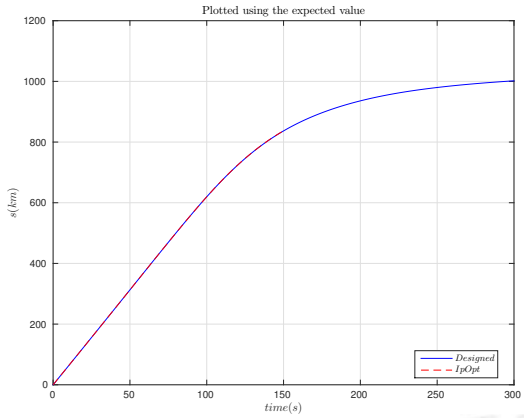


(e) It is observed that there is a slight difference between what was designed and what is calculated by IpOpt at time $t \approx 50s$. It is also seen that at time $t \approx 150s$, the difference becomes relatively large, it begins to increase.

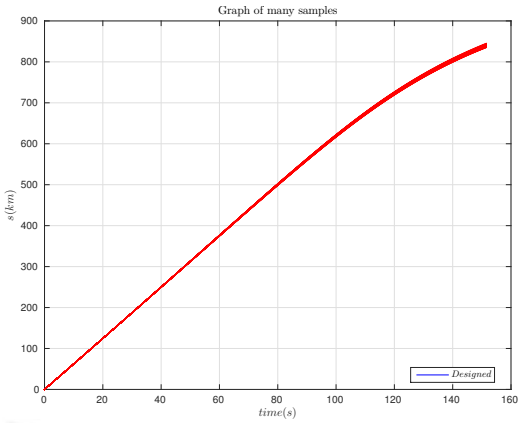


(f)

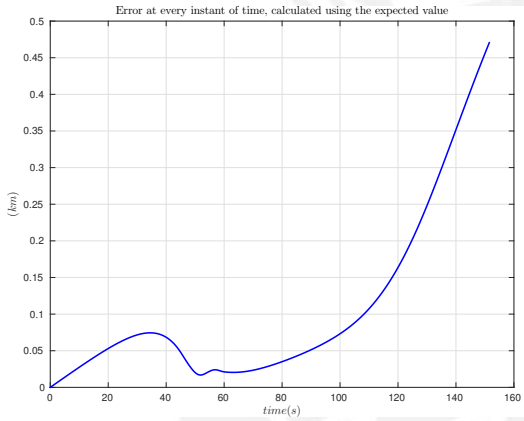
Figure 6.39 – Graphs of the tracking of the states $[h, v, \gamma]$.



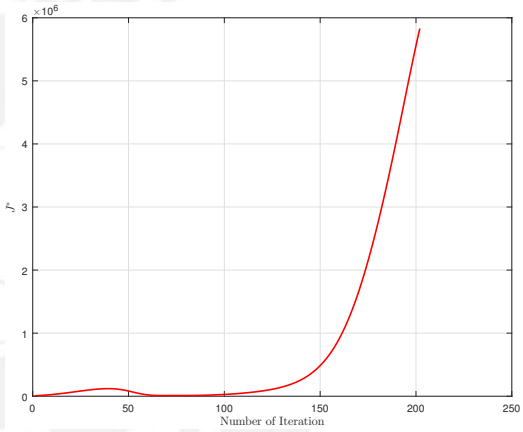
(a) The controller does a tracking of this state with almost imperceptible error.



(b)



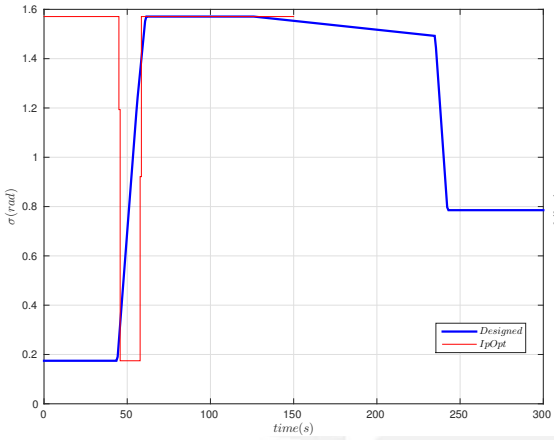
(c) Error at every instant of time.



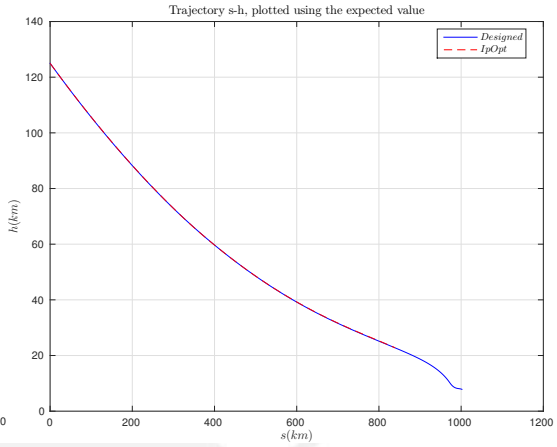
(d) The graph of the objective function has the same form as the error graph, Figure 6.40c, the interpretation is that as the Mars Lander is diverted from the designed path, the error increases, and consequently the controller will try harder to reduce said error of tracking.

Figure 6.40 – Graph of the tracking of the state s, also the Objective function.

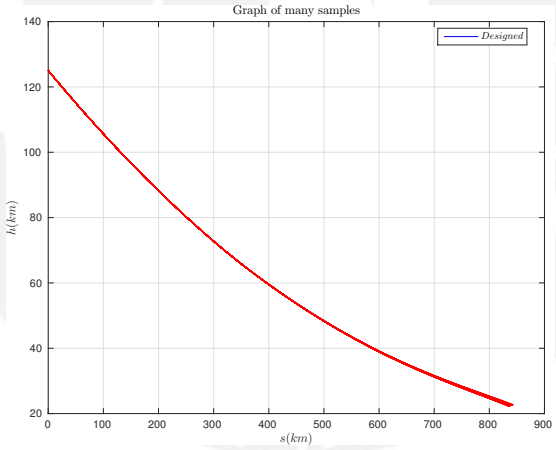
For $\tau_{\text{Outer}} = 0.25$, IpOpt solves the optimization problem, for each iteration, in a time that oscillates in the interval $[3.088, 3.7]$ seconds.
In Figures 6.41 to 6.43, the data obtained with the IpOpt is shown in visual form.



(a) It can be seen in the figure that the IpOpt data has been graphed to the middle of the time horizon ($T = 300s$). The reason is that IpOpt only manages to iterate until half of the time horizon T, at that time $t \approx 150s$, IpOpt shows the message: "Restoration Failed".

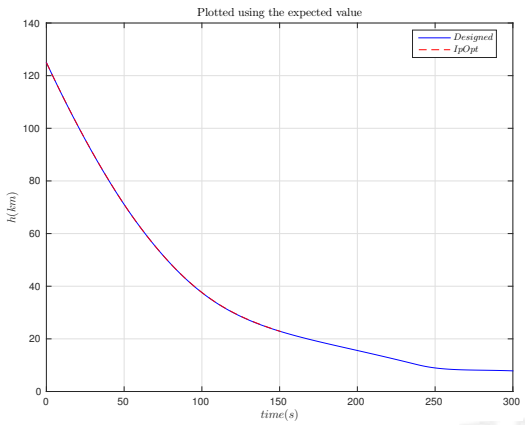


(b) Trajectory designed in the $s - h$ plane, also the trajectory calculated by IpOpt.

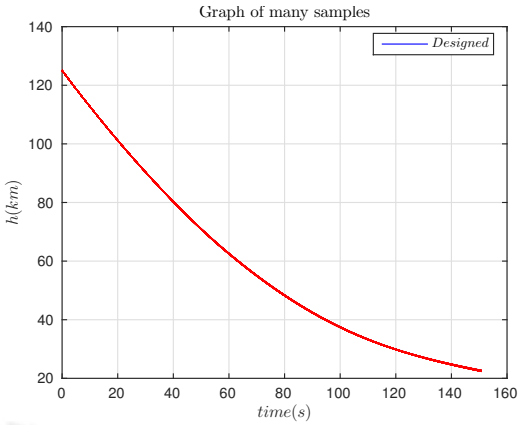


(c)

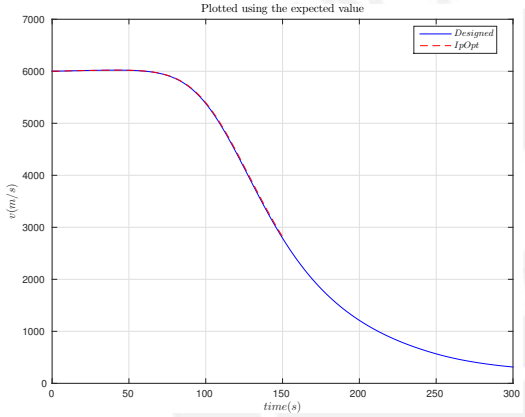
Figure 6.41 – Graph of σ, graph of the trajectory designed in the $s - h$ plane.



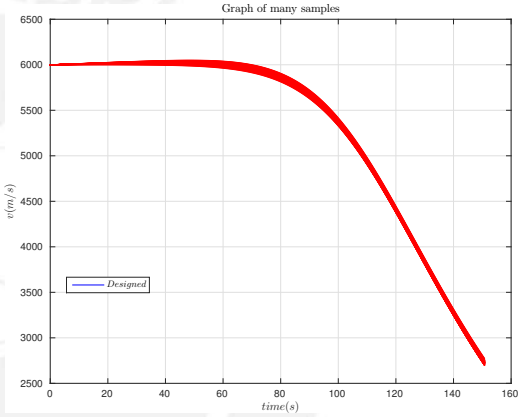
(a) The controller does a tracking of this state with almost imperceptible error.



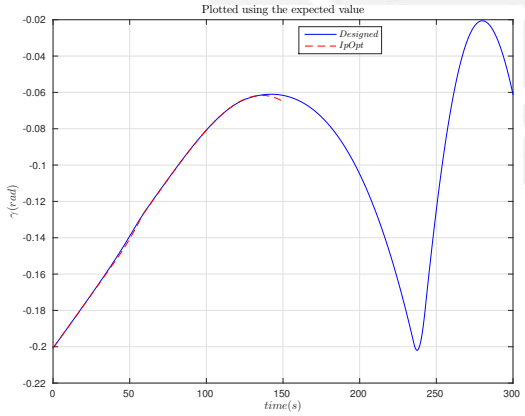
(b)



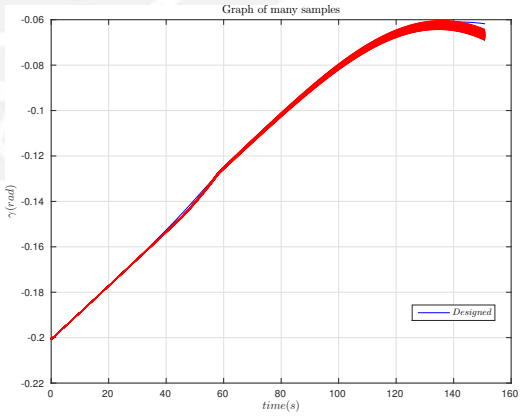
(c) The controller does a tracking of this state with almost imperceptible error.



(d)

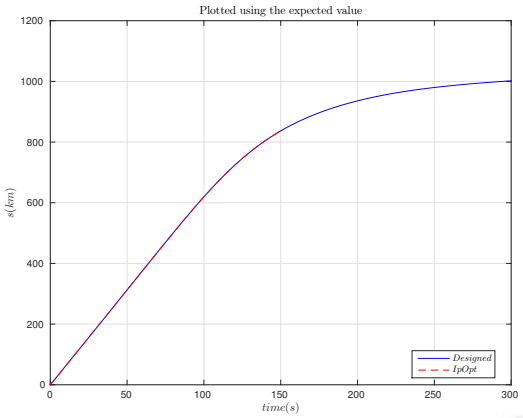


(e) It is observed that there is a slight difference between what was designed and what is calculated by IpOpt at time $t \approx 50s$. It is also seen that at time $t \approx 150s$, the difference becomes relatively large, it begins to increase.

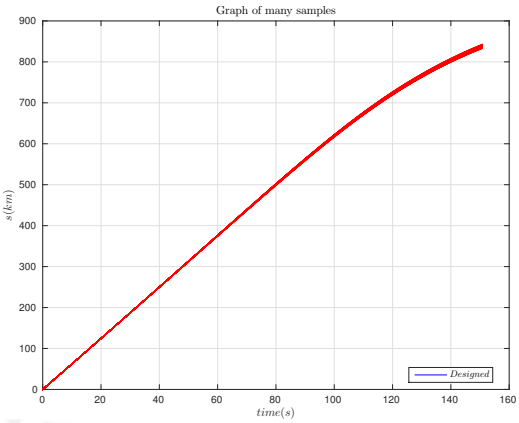


(f)

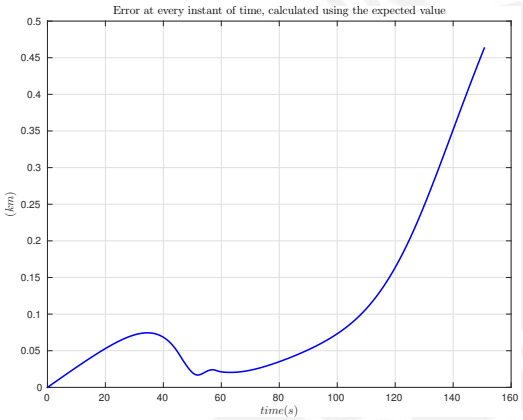
Figure 6.42 – Graphs of the tracking of the states $[h, v, \gamma]$.



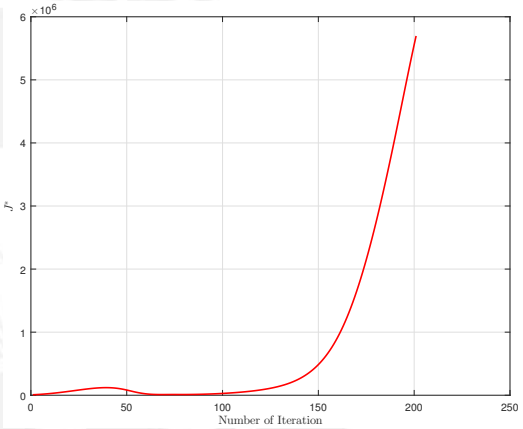
(a) The controller does a tracking of this state with almost imperceptible error.



(b)



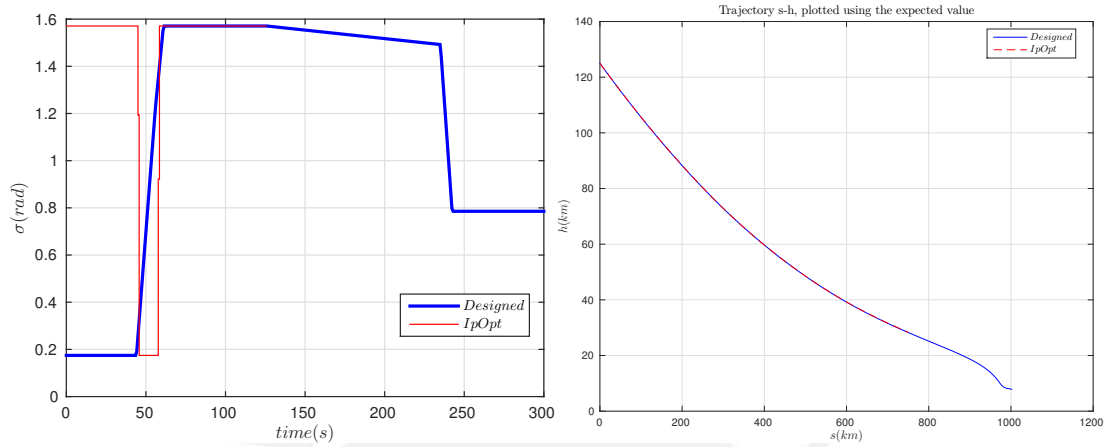
(c) Error at every instant of time.



(d) The graph of the objective function has the same form as the error graph, Figure 6.43c, the interpretation is that as the Mars Lander is diverted from the designed path, the error increases, and consequently the controller will try harder to reduce said error of tracking.

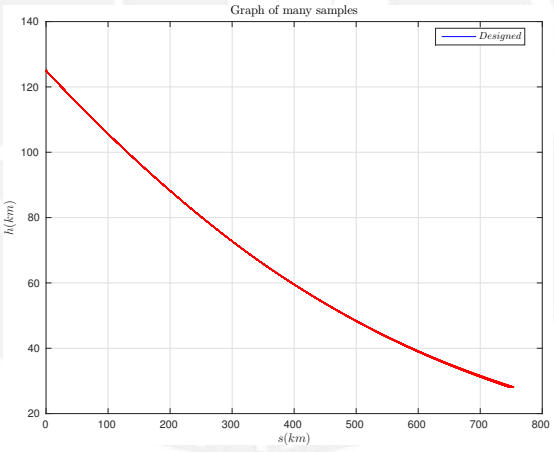
Figure 6.43 – Graph of the tracking of the state s, also the Objective function.

For $\tau_{\text{Inner}} = 0.0625$, IpOpt solves the optimization problem, for each iteration, in a time that oscillates in the interval $[3.184, 3.624]$ seconds.
In Figures 6.44 to 6.46, the data obtained with the IpOpt is shown in visual form.



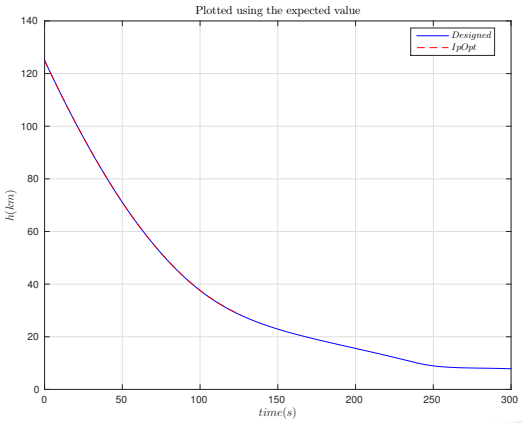
(a) It can be seen in the figure that the IpOpt data has been graphed to the middle of the time horizon ($T = 300s$). The reason is that IpOpt only manages to iterate until half of the time horizon T, at that time $t \approx 130s$, IpOpt shows the message: "Restoration Failed".

(b) Trajectory designed in the $s - h$ plane, also the trajectory calculated by IpOpt.

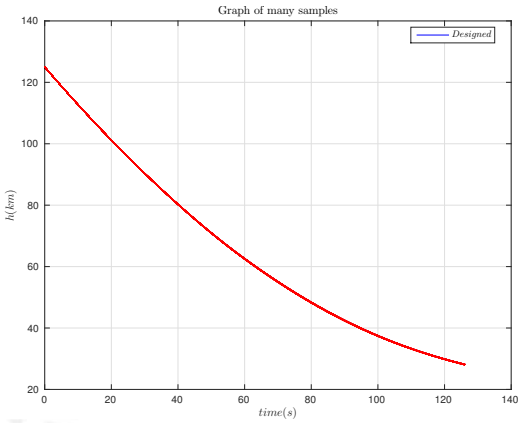


(c)

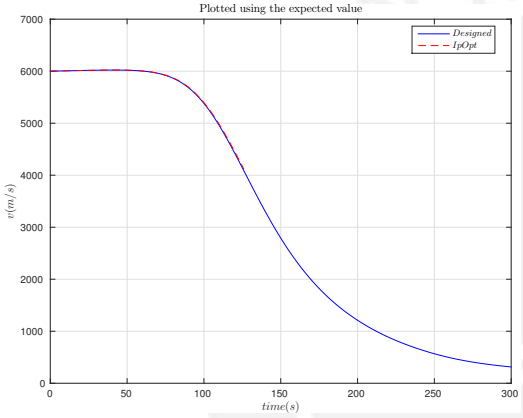
Figure 6.44 – Graph of σ, graph of the trajectory designed in the $s - h$ plane.



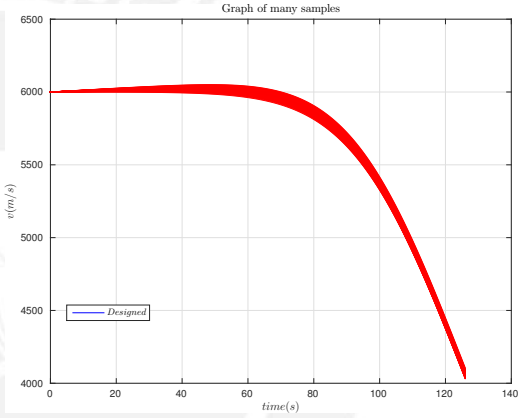
(a) The controller does a tracking of this state with almost imperceptible error.



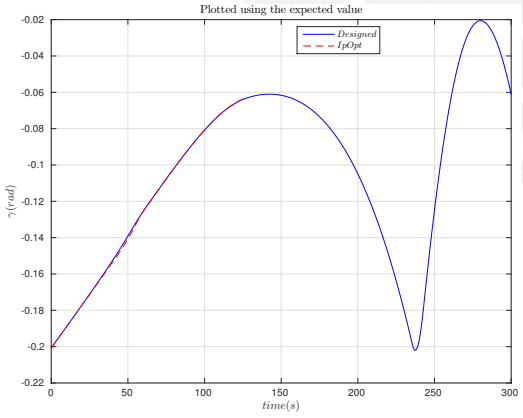
(b)



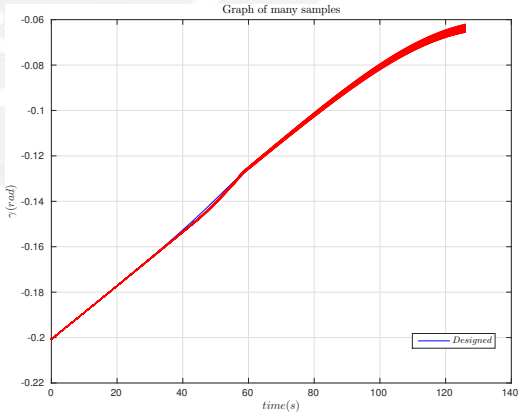
(c) The controller does a tracking of this state with almost imperceptible error.



(d)

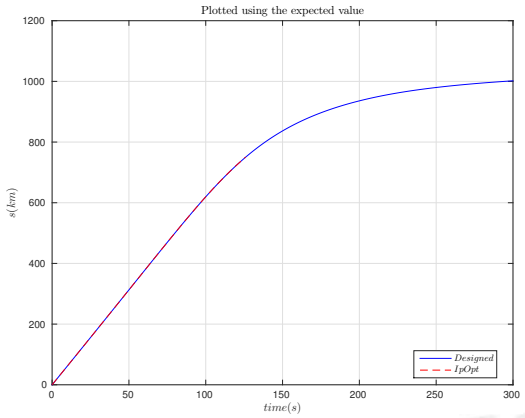


(e) It is observed that there is a slight difference between what was designed and what is calculated by IpOpt at time $t \approx 50s$.

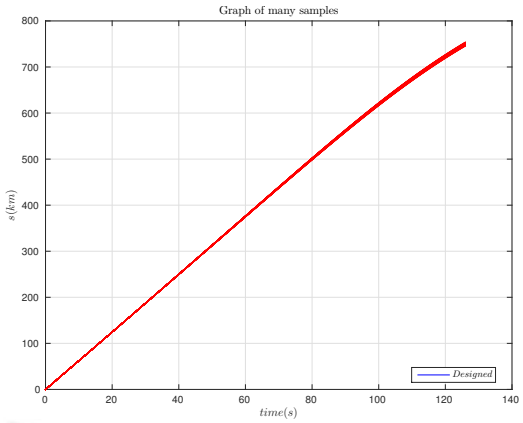


(f)

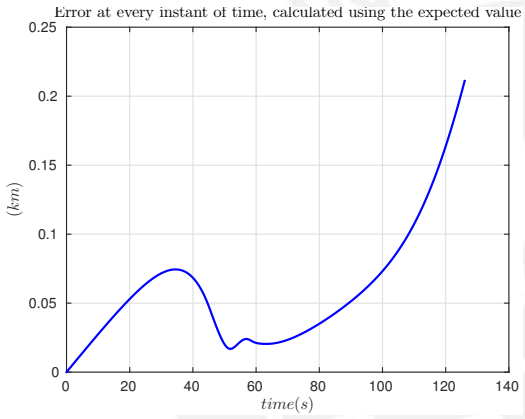
Figure 6.45 – Graphs of the tracking of the states $[h, v, \gamma]$.



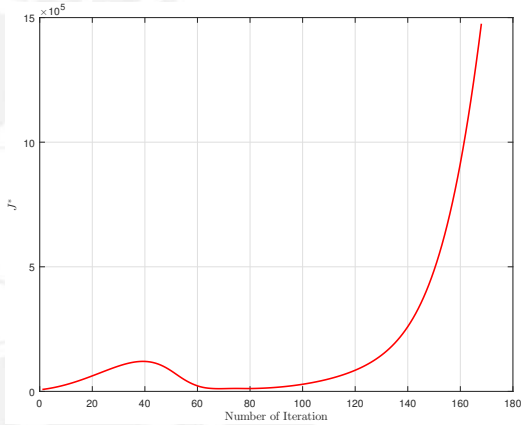
(a) The controller does a tracking of this state with almost imperceptible error.



(b)



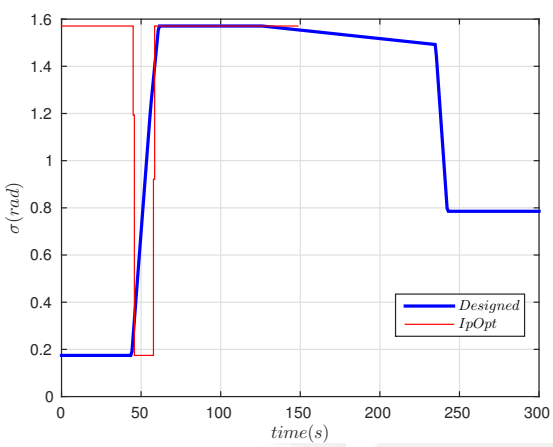
(c) Error at every instant of time.



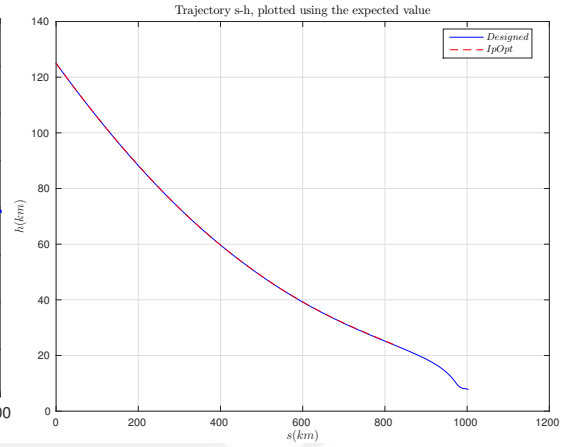
(d) The graph of the objective function has the same form as the error graph, Figure 6.46c, the interpretation is that as the Mars Lander is diverted from the designed path, the error increases, and consequently the controller will try harder to reduce said error of tracking.

Figure 6.46 – Graph of the tracking of the state s, also the Objective function.

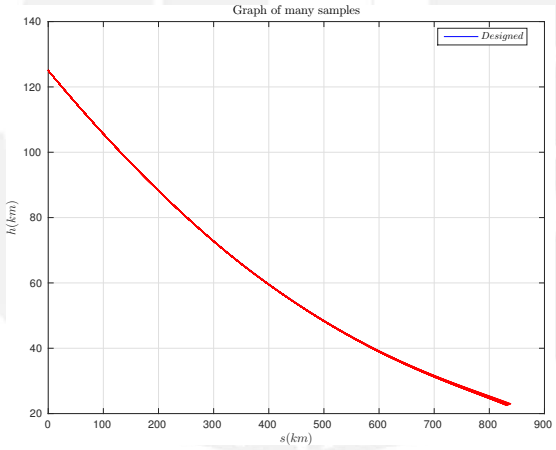
For $\tau_{\text{Outer}} = 0.0625$, IpOpt solves the optimization problem, for each iteration, in a time that oscillates in the interval $[3.068, 3.592]$ seconds.
In Figures 6.47 to 6.49, the data obtained with the IpOpt is shown in visual form.



(a) It can be seen in the figure that the IpOpt data has been graphed to the middle of the time horizon ($T = 300s$). The reason is that IpOpt only manages to iterate until half of the time horizon T, at that time $t \approx 150s$, IpOpt shows the message: "Restoration Failed".

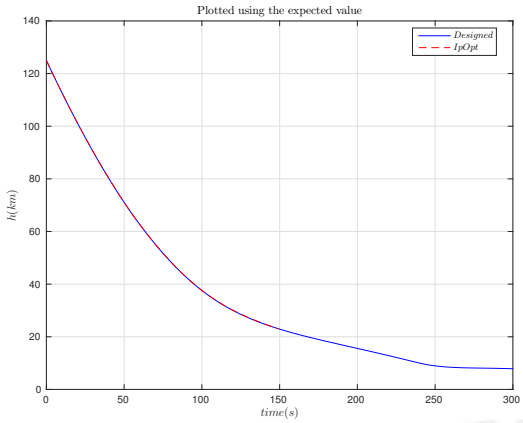


(b) Trajectory designed in the $s - h$ plane, also the trajectory calculated by IpOpt.

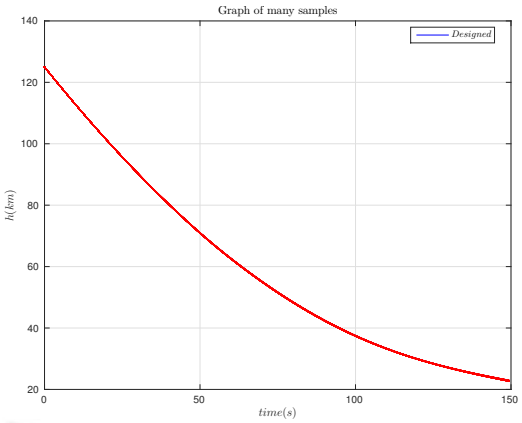


(c)

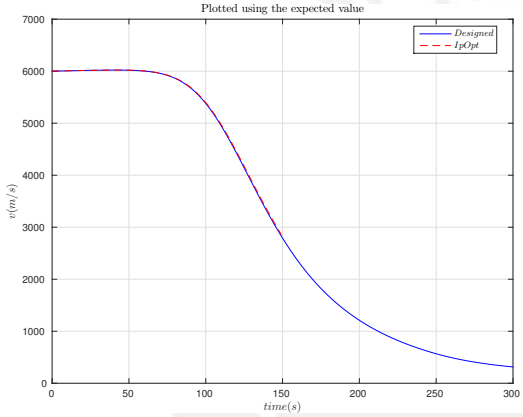
Figure 6.47 – Graph of σ, graph of the trajectory designed in the $s - h$ plane.



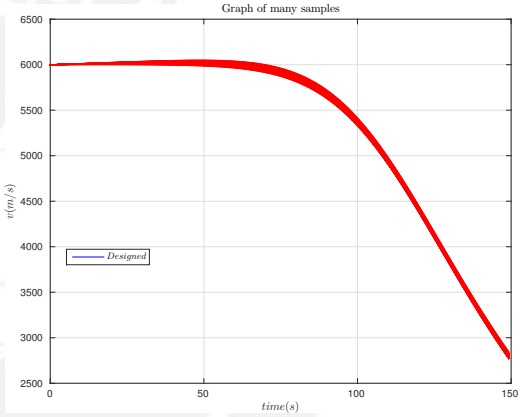
(a) The controller does a tracking of this state with almost imperceptible error.



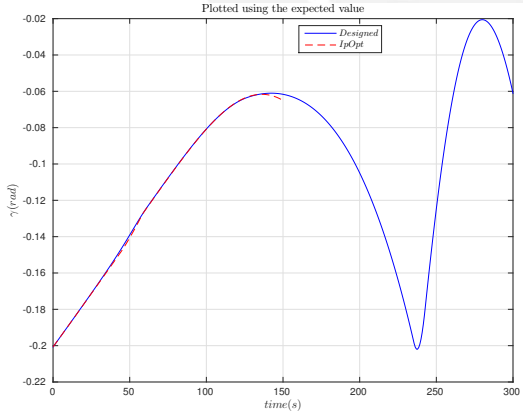
(b)



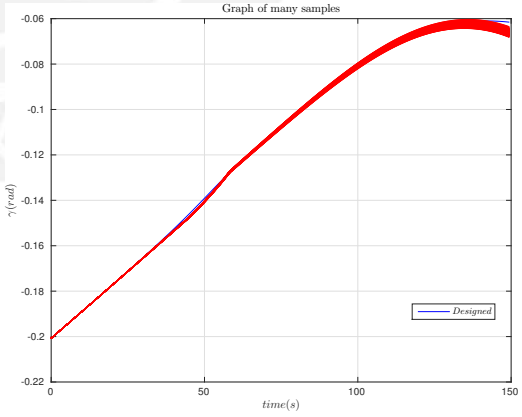
(c) The controller does a tracking of this state with almost imperceptible error.



(d)

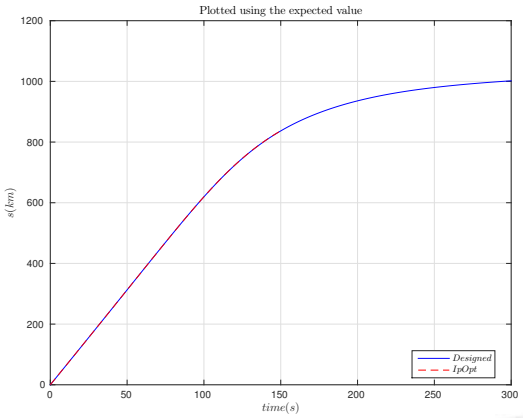


(e) It is observed that there is a slight difference between what was designed and what is calculated by IpOpt at time $t \approx 50s$. It is also seen that at time $t \approx 150s$, the difference becomes relatively large, it begins to increase.

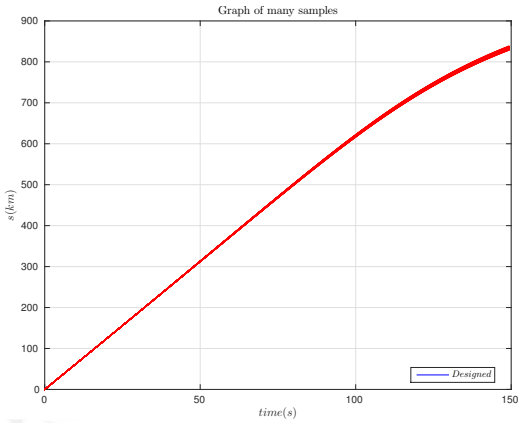


(f)

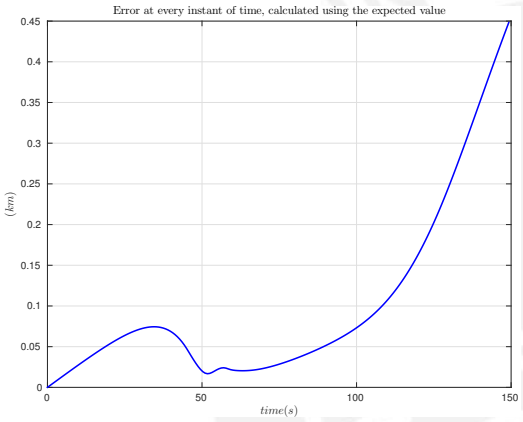
Figure 6.48 – Graphs of the tracking of the states $[h, v, \gamma]$.



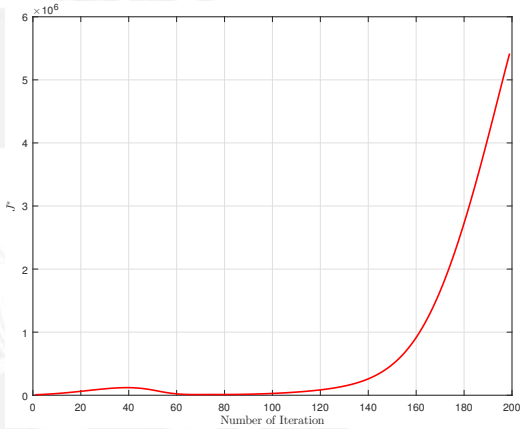
(a) The controller does a tracking of this state with almost imperceptible error.



(b)



(c) Error at every instant of time.



(d) The graph of the objective function has the same form as the error graph, Figure 6.49c, the interpretation is that as the Mars Lander is diverted from the designed path, the error increases, and consequently the controller will try harder to reduce said error of tracking.

Figure 6.49 – Graph of the tracking of the state s, also the Objective function.

According to the theory of the Inner and Outer approximations (Section 3.3.3), when τ tends to zero, the solution of $\text{NLP}_\tau^{\text{Inner}}$ and $\text{NLP}_\tau^{\text{outer}}$ must be the same, the same should occur with the value of their respective objective functions J^*.

From the above, then a trend should be observed for both the control law $\boldsymbol{u}^*$, and the value of the objective function J^* , as the value of τ decreases.

It can be seen that the control laws $\boldsymbol{u}^*$ obtained from Figures 6.32a, 6.35a, 6.38a, 6.41a, 6.44a and 6.47a are very similar, there are no appreciable changes.

With respect to the objective function J^*, we see in Figures 6.34d, 6.37d, 6.40d, 6.43d, 6.46d and 6.49d, those are very similar, no substantial changes are observed.

No trends are observed in $\boldsymbol{u}^*$, J^*, as τ tends to zero, it is left as future work to investigate because this trend is not appreciated.

It is re-emphasized that it was not possible to obtain the control laws $\boldsymbol{u}^*$ for the whole time horizon $T = 300s$, because the IpOpt aborts the simulations at time $t \approx 150$, and reports on the failure: "Converged to a point of local infeasibility. Problem may be infeasible" or "Restoration Failed".

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this thesis, we study and implement (by simulation) the chance constrained MPC for reliable spacecraft trajectory tracking and landing, this control strategy is adequate in those applications where there are disturbances in both the model parameters and external.

It was identified that in space missions there are many sources of disturbances, such as the pressure exerted by solar winds, space debris, the drag force of outer space, the dynamics not modeled in the model used, errors in sensor measurements, misalignment of propellants, etc.

This control strategy is also adequate when the dynamics of the model is highly nonlinear, this is precisely the case with the models addressed in this thesis.

In order to solve the chance constrained MPC, it was first necessary to discretize the chance constrained OCP to a chance constrained NLP, the discretization method used was the Runge-Kutta 4th order, then the chance constraints are approximated in such a way that they become a group of deterministic inequalities. The above is achieved by applying the new analytic approximation methods, inner and outer.

After applying the new analytic approximation methods, the chance constrained NLP is converted to a deterministic $\text{NLP}_{\tau}^{\text{Inner}}$, or $\text{NLP}_{\tau}^{\text{Outer}}$.

This deterministic NLP is applied the Quasi-sequential approach, in this way, a reduced NLP is obtained. The optimization variables of the reduced NLP are the elements of the sequence $\boldsymbol{u}^* = \{\boldsymbol{u}_0^*, \boldsymbol{u}_1^*, \ldots, \boldsymbol{u}_{N-1}^*\}$.

The reduced NLP can be solved by some numerical method for optimization problems.

After the first element of the sequence $\boldsymbol{u}^*$ is preserved, the next step is to find the new initial state vector $\boldsymbol{x}_0$, finally the finite time horizon slides to the next sampling time.

The exposed sequence of steps is repeated until a stop condition is satisfied.

It is very important to generate sequences of random numbers of low discrepancy (Sobol sequences), without the previous sequences, cannot carry out the simulation.

The generation of sequences of random numbers of low discrepancy (Sobol sequences), the implementation of a solver of Newton, the implementation of a linear algebra solver

for sparse linear system equations, was realized by means of the implementation of libraries in C++, also in these C++ libraries , the GNU scientific library was used.
IpOpt software was used in C++ to solve the reduced NLPs obtained in each case study. IpOpt is an optimization solver based on gradients, so it is necessary to provide information about first and second derivatives.
To find the first and second derivatives, we used Matlab and Cassadi.
The second derivative, that is the Hessian of the Lagrangian, was approximated by the algorithm L-BFGS, which is part of the IpOpt software package.
The use of the new analytic approximation methods (Inner and Outer), leads to the solution of a deterministic NLP, whose resolution is computationally expensive, but whose results are much more realistic, if reliable statistical information is available of the stochastic variables.
In the simulation of the chance constrained MPC, for Case 1, the following problems occurred:

- It was possible to solve the $\text{NLP}_{\tau}^{\text{Inner}}$ for several cases mentioned in Tables 6.1 to 6.3, but this was not the case for $\text{NLP}_{\tau}^{\text{Outer}}$.
- It was not possible to find the stochastic control law $\boldsymbol{u}^*$ for the entire time horizon $T = 4000s$.

In the simulation of the chance constrained MPC, for Case 2, the following problems occurred:

- It was possible to solve the deterministic $\text{NLP}_{\tau}^{\text{Inner}}$, for $\tau = 0.8,\ 0.9$, and determined set of parameters. The deterministic $\text{NLP}_{\tau}^{\text{Inner}}$ could not be resolved with an smaller τ.
- It was not possible to resolve the $\text{NLP}_{\tau}^{\text{Outer}}$, for any τ, even when tried with several sets of parameters.

From the above, it will be left as future work to investigate in detail the dynamics and properties of the model of Case 2.
In the simulation of the chance constrained MPC , for Case 3, the following events occurred:

- It was possible to solve the deterministic $\text{NLP}_{\tau}^{\text{Inner}}$, $\text{NLP}_{\tau}^{\text{Outer}}$, for $\tau = 0.5,\ 0.25,\ 0.0625$ and certain set of parameters.
- There is no trend in $\boldsymbol{u}^*$, J^*, as for the $\text{NLP}_{\tau}^{\text{Inner}}$ and $\text{NLP}_{\tau}^{\text{Outer}}$, as τ tends to zero.
- IpOpt fails to find the control law $\boldsymbol{u}^*$, for the entire time horizon $T = 300s$, it is only possible to find the control law until the time $t \approx 150s$. At that instant IpOpt aborts the program, and reports the cause of the failure: “Converged to a point of local infeasibility. Problem may be infeasible” or “Restoration Failed”.

7.2 Future Work

- To investigate why the $\text{NLP}_{\tau}^{\text{Outer}}$ of case 1 and 2, cannot be solved by IpOpt, this implies a detailed study of the dynamic model and its properties.

- Design a solution that allows to solve the stochastic MPC of case 2, making use of the discretization method of Runge-Kutta 4th order.
- To investigate why in case 3, we cannot see the trend in $\boldsymbol{u}^*$ and J^*, as the value of τ tends to zero.
- Address the three case studies of the thesis, using JCC, which are more realistic restrictions.
- Application of parallelization techniques, to reduce the time invested in the evaluation of NLP functions.

Appendices

A. Gradient of the Gravitational Potential U

$$
\begin{aligned}
U_x = -\left(\frac{\mathrm{GM}}{8(x^2+y^2+z^2)^{\frac{13}{2}}}\right)\Big[&40\,x^3\,y^8+80\,x^5\,y^6+80\,x^7\,y^4+40\,x^9\,y^2+40\,x^3\,z^8\\
&+80\,x^5\,z^6+80\,x^7\,z^4+40\,x^9\,z^2+8\,x\,y^{10}+8\,x\,z^{10}+8\,x^{11}+15\,\mathrm{C}_{40}\,{\mathrm{r}_0}^4\,x^7\\
&+40\,x\,y^2\,z^8+80\,x\,y^4\,z^6+80\,x\,y^6\,z^4+40\,x\,y^8\,z^2+160\,x^3\,y^2\,z^6+240\,x^3\,y^4\,z^4+160\,x^3\,y^6\,z^2\\
&+240\,x^5\,y^2\,z^4+240\,x^5\,y^4\,z^2+160\,x^7\,y^2\,z^2-300\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,x^7\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&+4200\,\mathrm{C}_{44}\,{\mathrm{r}_0}^4\,x^7\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right)+45\,\mathrm{C}_{40}\,{\mathrm{r}_0}^4\,x^3\,y^4+45\,\mathrm{C}_{40}\,{\mathrm{r}_0}^4\,x^5\,y^2\\
&+72\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,x^3\,z^6+72\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,x^5\,z^4+24\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,x^7\,z^2\\
&-80\,\mathrm{C}_{40}\,{\mathrm{r}_0}^4\,x^3\,z^4-105\,\mathrm{C}_{40}\,{\mathrm{r}_0}^4\,x^5\,z^2-48\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,y^9\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&-12\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,x^7\,\left(x^2+y^2+z^2\right)+15\,\mathrm{C}_{40}\,{\mathrm{r}_0}^4\,x\,y^6+24\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,x\,z^8\\
&+40\,\mathrm{C}_{40}\,{\mathrm{r}_0}^4\,x\,z^6-192\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^2\,y^7\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&-288\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^4\,y^5\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)-192\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^6\,y^3\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&+72\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^7\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\left(x^2+y^2+z^2\right)-192\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,y^3\,z^6\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&-288\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,y^5\,z^4\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)-192\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,y^7\,z^2\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&-36\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,x^3\,y^4\,\left(x^2+y^2+z^2\right)-36\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,x^5\,y^2\,\left(x^2+y^2+z^2\right)\\
&+36\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,x^3\,z^4\,\left(x^2+y^2+z^2\right)-60\,\mathrm{C}_{40}\,{\mathrm{r}_0}^4\,x^3\,z^2\,\left(x^2+y^2+z^2\right)\\
&+120\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,y^5\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\left(x^2+y^2+z^2\right)
\end{aligned}
$$

Continuation of Equation U_x

$$
\begin{aligned}
&- 48\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x\,z^8\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) + 72\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,x\,y^2\,z^6 + 72\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,x\,y^4\,z^4 \\
&+ 24\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,x\,y^6\,z^2 - 80\,\mathrm{C_{40}}\,{\mathrm{r_0}}^4\,x\,y^2\,z^4 - 105\,\mathrm{C_{40}}\,{\mathrm{r_0}}^4\,x\,y^4\,z^2 \\
&- 48\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^8\,y\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right) - 48\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,y\,z^8\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right) \\
&- 12\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,x\,y^6\,\left(x^2+y^2+z^2\right) + 24\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,x\,z^6\,\left(x^2+y^2+z^2\right) + 80\,\mathrm{C_{40}}\,{\mathrm{r_0}}^4\,x\,z^4\,\left(x^2+y^2+z^2\right) \\
&- 900\,\mathrm{C_{42}}\,{\mathrm{r_0}}^4\,x^3\,y^4\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) - 900\,\mathrm{C_{42}}\,{\mathrm{r_0}}^4\,x^5\,y^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) \\
&+ 12600\,\mathrm{C_{44}}\,{\mathrm{r_0}}^4\,x^3\,y^4\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right) + 12600\,\mathrm{C_{44}}\,{\mathrm{r_0}}^4\,x^5\,y^2\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right) \\
&- 144\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^3\,z^6\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) - 144\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^5\,z^4\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) \\
&- 48\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^7\,z^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) + 1800\,\mathrm{C_{42}}\,{\mathrm{r_0}}^4\,x^3\,z^4\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) \\
&+ 1500\,\mathrm{C_{42}}\,{\mathrm{r_0}}^4\,x^5\,z^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) + 4200\,\mathrm{C_{44}}\,{\mathrm{r_0}}^4\,x^3\,z^4\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right) \\
&+ 8400\,\mathrm{C_{44}}\,{\mathrm{r_0}}^4\,x^5\,z^2\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right) + 144\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,x^3\,y^2\,z^4 + 72\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,x^3\,y^4\,z^2 \\
&+ 72\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,x^5\,y^2\,z^2 - 210\,\mathrm{C_{40}}\,{\mathrm{r_0}}^4\,x^3\,y^2\,z^2 - 144\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x\,y^2\,z^6\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) \\
&- 144\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x\,y^4\,z^4\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) - 48\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x\,y^6\,z^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) \\
&+ 1800\,\mathrm{C_{42}}\,{\mathrm{r_0}}^4\,x\,y^2\,z^4\,\mathrm{cos!}\left(2\,\arctan\left(\frac{y}{x}\right)\right) + 1500\,\mathrm{C_{42}}\,{\mathrm{r_0}}^4\,x\,y^4\,z^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) \\
&+ 4200\,\mathrm{C_{44}}\,{\mathrm{r_0}}^4\,x\,y^2\,z^4\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right) - 60\,\mathrm{C_{40}}\,{\mathrm{r_0}}^4\,x\,y^2\,z^2\,\left(x^2+y^2+z^2\right) \\
&+ 8400\,\mathrm{C_{44}}\,{\mathrm{r_0}}^4\,x\,y^4\,z^2\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right) + 72\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x\,y^6\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right) \\
&- 192\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^2\,y\,z^6\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right) - 288\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^4\,y\,z^4\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right) \\
&- 192\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^6\,y\,z^2\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right) - 720\,\mathrm{C_{42}}\,{\mathrm{r_0}}^4\,x\,z^4\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right) \\
&- 1680\,\mathrm{C_{44}}\,{\mathrm{r_0}}^4\,x\,z^4\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right) + 36\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,x\,y^2\,z^4\,\left(x^2+y^2+z^2\right) \\
&+ 120\,\mathrm{C_{42}}\,{\mathrm{r_0}}^4\,x^4\,y\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right) \\
&- 3360\,\mathrm{C_{44}}\,{\mathrm{r_0}}^4\,x^4\,y\,\sin\left(4\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right) \\
&- 720\,\mathrm{C_{42}}\,{\mathrm{r_0}}^4\,y\,z^4\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right) \\
&- 3360\,\mathrm{C_{44}}\,{\mathrm{r_0}}^4\,y\,z^4\,\sin\left(4\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right) \\
&- 288\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^3\,y^2\,z^4\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) - 144\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^3\,y^4\,z^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)
\end{aligned}
$$

Continuation of Equation U_x

$$
\begin{aligned}
&- 144\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^5\,y^2\,z^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) + 3000\,\mathrm{C_{42}}\,{\mathrm{r_0}}^4\,x^3\,y^2\,z^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&+ 16800\,\mathrm{C_{44}}\,{\mathrm{r_0}}^4\,x^3\,y^2\,z^2\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right) + 216\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^3\,y^4\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 216\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^5\,y^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right) - 576\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^2\,y^3\,z^4\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&- 576\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^2\,y^5\,z^2\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right) - 576\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^4\,y^3\,z^2\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&+ 72\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^3\,z^4\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 144\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^5\,z^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 960\,\mathrm{C_{42}}\,{\mathrm{r_0}}^4\,x^3\,z^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&- 1680\,\mathrm{C_{44}}\,{\mathrm{r_0}}^4\,x^3\,z^2\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 240\,\mathrm{C_{42}}\,{\mathrm{r_0}}^4\,x^2\,y^3\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&- 6720\,\mathrm{C_{44}}\,{\mathrm{r_0}}^4\,x^2\,y^3\,\sin\left(4\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&- 600\,\mathrm{C_{42}}\,{\mathrm{r_0}}^4\,y^3\,z^2\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&- 6720\,\mathrm{C_{44}}\,{\mathrm{r_0}}^4\,y^3\,z^2\,\sin\left(4\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 72\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x\,y^2\,z^4\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 144\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x\,y^4\,z^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 960\,\mathrm{C_{42}}\,{\mathrm{r_0}}^4\,x\,y^2\,z^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&- 1680\,\mathrm{C_{44}}\,{\mathrm{r_0}}^4\,x\,y^2\,z^2\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&- 600\,\mathrm{C_{42}}\,{\mathrm{r_0}}^4\,x^2\,y\,z^2\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&- 6720\,\mathrm{C_{44}}\,{\mathrm{r_0}}^4\,x^2\,y\,z^2\,\sin\left(4\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&- 3360\,\mathrm{C_{44}}\,{\mathrm{r_0}}^4\,y^5\,\sin\left(4\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&- 300\,\mathrm{C_{42}}\,{\mathrm{r_0}}^4\,x\,y^6\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) + 4200\,\mathrm{C_{44}}\,{\mathrm{r_0}}^4\,x\,y^6\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right)\\
&+288\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^3\,y^2\,z^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\Bigg]
\end{aligned}
$$

$$
\begin{aligned}
U_y = &-\left(\frac{\mathrm{GM}}{8(x^2+y^2+z^2)^{\frac{13}{2}}}\right)\Big[40\,x^2\,y^9+80\,x^4\,y^7+80\,x^6\,y^5+40\,x^8\,y^3+40\,y^3\,z^8+80\,y^5\,z^6\\
&+80\,y^7\,z^4+40\,y^9\,z^2+8\,x^{10}\,y+8\,y\,z^{10}+8\,y^{11}+15\,\mathrm{C}_{40}\,{\mathrm{r}_0}^4\,y^7+40\,x^2\,y\,z^8+80\,x^4\,y\,z^6+80\,x^6\,y\,z^4\\
&+160\,x^2\,y^3\,z^6+240\,x^2\,y^5\,z^4+160\,x^2\,y^7\,z^2+240\,x^4\,y^3\,z^4+240\,x^4\,y^5\,z^2+160\,x^6\,y^3\,z^2\\
&+4200\,\mathrm{C}_{44}\,{\mathrm{r}_0}^4\,y^7\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right)+45\,\mathrm{C}_{40}\,{\mathrm{r}_0}^4\,x^2\,y^5+45\,\mathrm{C}_{40}\,{\mathrm{r}_0}^4\,x^4\,y^3+40\,x^8\,y\,z^2\\
&+72\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,y^3\,z^6+72\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,y^5\,z^4+24\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,y^7\,z^2-80\,\mathrm{C}_{40}\,{\mathrm{r}_0}^4\,y^3\,z^4-105\,\mathrm{C}_{40}\,{\mathrm{r}_0}^4\,y^5\,z^2\\
&+48\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^9\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)-12\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,y^7\,\left(x^2+y^2+z^2\right)+15\,\mathrm{C}_{40}\,{\mathrm{r}_0}^4\,x^6\,y\\
&+24\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,y\,z^8+40\,\mathrm{C}_{40}\,{\mathrm{r}_0}^4\,y\,z^6+192\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^3\,y^6\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&+288\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^5\,y^4\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)+192\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^7\,y^2\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&+192\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^3\,z^6\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)+288\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^5\,z^4\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&+192\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^7\,z^2\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)+72\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,y^7\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&-36\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,x^2\,y^5\,\left(x^2+y^2+z^2\right)-36\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,x^4\,y^3\,\left(x^2+y^2+z^2\right)+36\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,y^3\,z^4\,\left(x^2+y^2+z^2\right)\\
&-300\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,y^7\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)-60\,\mathrm{C}_{40}\,{\mathrm{r}_0}^4\,y^3\,z^2\,\left(x^2+y^2+z^2\right)\\
&-120\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,x^5\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+3360\,\mathrm{C}_{44}\,{\mathrm{r}_0}^4\,x^5\,\sin\left(4\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&-300\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,x^6\,y\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)+4200\,\mathrm{C}_{44}\,{\mathrm{r}_0}^4\,x^6\,y\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right)\\
&-48\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,y\,z^8\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)+72\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,x^2\,y\,z^6+72\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,x^4\,y\,z^4\\
&+24\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,x^6\,y\,z^2-80\,\mathrm{C}_{40}\,{\mathrm{r}_0}^4\,x^2\,y\,z^4-105\,\mathrm{C}_{40}\,{\mathrm{r}_0}^4\,x^4\,y\,z^2\\
&+48\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x\,y^8\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)+48\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x\,z^8\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&-12\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,x^6\,y\,\left(x^2+y^2+z^2\right)+24\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,y\,z^6\,\left(x^2+y^2+z^2\right)\\
&+80\,\mathrm{C}_{40}\,{\mathrm{r}_0}^4\,y\,z^4\,\left(x^2+y^2+z^2\right)-900\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,x^2\,y^5\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&-900\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,x^4\,y^3\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)+12600\,\mathrm{C}_{44}\,{\mathrm{r}_0}^4\,x^2\,y^5\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right)\\
&+12600\,\mathrm{C}_{44}\,{\mathrm{r}_0}^4\,x^4\,y^3\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right)-144\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,y^3\,z^6\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&-144\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,y^5\,z^4\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)-48\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,y^7\,z^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&+1800\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,y^3\,z^4\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)+1500\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,y^5\,z^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)
\end{aligned}
$$

Continuation of Equation U_y

$$
\begin{aligned}
&+ 4200\,\mathrm{C}_{44}\,{\mathrm{r}_0}^4\,y^3\,z^4\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right) + 8400\,\mathrm{C}_{44}\,{\mathrm{r}_0}^4\,y^5\,z^2\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right)\\
&+ 144\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,x^2\,y^3\,z^4 + 72\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,x^2\,y^5\,z^2 + 72\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,x^4\,y^3\,z^2\\
&- 210\,\mathrm{C}_{40}\,{\mathrm{r}_0}^4\,x^2\,y^3\,z^2 - 144\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^2\,y\,z^6\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&- 144\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^4\,y\,z^4\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) - 48\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^6\,y\,z^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&+ 1800\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,x^2\,y\,z^4\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) + 1500\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,x^4\,y\,z^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&+ 4200\,\mathrm{C}_{44}\,{\mathrm{r}_0}^4\,x^2\,y\,z^4\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right) + 8400\,\mathrm{C}_{44}\,{\mathrm{r}_0}^4\,x^4\,y\,z^2\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right)\\
&+ 72\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^6\,y\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\left(x^2+y^2+z^2\right) + 192\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x\,y^2\,z^6\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&+ 288\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x\,y^4\,z^4\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right) + 192\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x\,y^6\,z^2\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&- 720\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,y\,z^4\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\left(x^2+y^2+z^2\right)\\
&- 1680\,\mathrm{C}_{44}\,{\mathrm{r}_0}^4\,y\,z^4\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right)\left(x^2+y^2+z^2\right) + 36\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,x^2\,y\,z^4\,\left(x^2+y^2+z^2\right)\\
&- 60\,\mathrm{C}_{40}\,{\mathrm{r}_0}^4\,x^2\,y\,z^2\,\left(x^2+y^2+z^2\right) - 120\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,x\,y^4\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\left(x^2+y^2+z^2\right)\\
&+ 3360\,\mathrm{C}_{44}\,{\mathrm{r}_0}^4\,x\,y^4\,\sin\left(4\,\arctan\left(\frac{y}{x}\right)\right)\left(x^2+y^2+z^2\right)\\
&+ 720\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,x\,z^4\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\left(x^2+y^2+z^2\right)\\
&+ 3360\,\mathrm{C}_{44}\,{\mathrm{r}_0}^4\,x\,z^4\,\sin\left(4\,\arctan\left(\frac{y}{x}\right)\right)\left(x^2+y^2+z^2\right) - 288\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^2\,y^3\,z^4\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&- 144\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^2\,y^5\,z^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) - 144\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^4\,y^3\,z^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&+ 3000\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,x^2\,y^3\,z^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) + 16800\,\mathrm{C}_{44}\,{\mathrm{r}_0}^4\,x^2\,y^3\,z^2\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right)\\
&+ 216\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^2\,y^5\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\left(x^2+y^2+z^2\right)\\
&+ 216\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^4\,y^3\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\left(x^2+y^2+z^2\right) + 576\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^3\,y^2\,z^4\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&+ 576\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^3\,y^4\,z^2\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right) + 576\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^5\,y^2\,z^2\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&+ 72\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,y^3\,z^4\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\left(x^2+y^2+z^2\right)\\
&+ 144\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,y^5\,z^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\left(x^2+y^2+z^2\right)\\
&+ 960\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,y^3\,z^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\left(x^2+y^2+z^2\right)
\end{aligned}
$$

Continuation of Equation U_y

$$
\begin{aligned}
&- 1680\,\mathrm{C}_{44}\,{\mathrm{r}_0}^4\,y^3\,z^2\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&- 240\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,x^3\,y^2\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 6720\,\mathrm{C}_{44}\,{\mathrm{r}_0}^4\,x^3\,y^2\,\sin\left(4\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 600\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,x^3\,z^2\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 6720\,\mathrm{C}_{44}\,{\mathrm{r}_0}^4\,x^3\,z^2\,\sin\left(4\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 72\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^2\,y\,z^4\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 144\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^4\,y\,z^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 960\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,x^2\,y\,z^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&- 1680\,\mathrm{C}_{44}\,{\mathrm{r}_0}^4\,x^2\,y\,z^2\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 600\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,x\,y^2\,z^2\,\sin\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 6720\,\mathrm{C}_{44}\,{\mathrm{r}_0}^4\,x\,y^2\,z^2\,\sin\left(4\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 288\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^2\,y^3\,z^2\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\Bigg]
\end{aligned}
$$

$$
\begin{aligned}
U_z = -\left(\frac{\mathrm{GM}}{(x^2+y^2+z^2)^{\frac{13}{2}}}\right)\Big[
&40\,x^2\,z^9 + 80\,x^4\,z^7 + 80\,x^6\,z^5 + 40\,x^8\,z^3 + 40\,y^2\,z^9 + 80\,y^4\,z^7 \\
&+ 80\,y^6\,z^5 + 40\,y^8\,z^3 + 8\,x^{10}\,z + 8\,y^{10}\,z + 8\,z^{11} + 40\,\mathrm{C_{40}}\,{\mathrm{r_0}}^4\,z^7 + 40\,x^2\,y^8\,z + 80\,x^4\,y^6\,z \\
&+ 80\,x^6\,y^4\,z + 40\,x^8\,y^2\,z + 160\,x^2\,y^2\,z^7 + 240\,x^2\,y^4\,z^5 + 160\,x^2\,y^6\,z^3 + 240\,x^4\,y^2\,z^5 + 240\,x^4\,y^4\,z^3 \\
&+ 160\,x^6\,y^2\,z^3 - 24\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,x^2\,z^7 - 72\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,x^4\,z^5 - 72\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,x^6\,z^3 - 80\,\mathrm{C_{40}}\,{\mathrm{r_0}}^4\,x^2\,z^5 \\
&- 105\,\mathrm{C_{40}}\,{\mathrm{r_0}}^4\,x^4\,z^3 - 24\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,y^2\,z^7 - 72\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,y^4\,z^5 - 72\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,y^6\,z^3 - 80\,\mathrm{C_{40}}\,{\mathrm{r_0}}^4\,y^2\,z^5 \\
&- 105\,\mathrm{C_{40}}\,{\mathrm{r_0}}^4\,y^4\,z^3 + 24\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,z^7\,\left(x^2+y^2+z^2\right) - 24\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,x^8\,z + 15\,\mathrm{C_{40}}\,{\mathrm{r_0}}^4\,x^6\,z \\
&- 24\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,y^8\,z + 15\,\mathrm{C_{40}}\,{\mathrm{r_0}}^4\,y^6\,z + 36\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,x^2\,z^5\,\left(x^2+y^2+z^2\right) \\
&- 80\,\mathrm{C_{40}}\,{\mathrm{r_0}}^4\,x^2\,z^3\,\left(x^2+y^2+z^2\right) + 36\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,y^2\,z^5\,\left(x^2+y^2+z^2\right) \\
&+ 48\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^8\,z\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) - 300\,\mathrm{C_{42}}\,{\mathrm{r_0}}^4\,x^6\,z\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) \\
&+ 4200\,\mathrm{C_{44}}\,{\mathrm{r_0}}^4\,x^6\,z\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right) + 48\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,y^8\,z\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) \\
&- 300\,\mathrm{C_{42}}\,{\mathrm{r_0}}^4\,y^6\,z\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) + 4200\,\mathrm{C_{44}}\,{\mathrm{r_0}}^4\,y^6\,z\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right) \\
&- 96\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,x^2\,y^6\,z - 144\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,x^4\,y^4\,z - 96\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,x^6\,y^2\,z \\
&+ 45\,\mathrm{C_{40}}\,{\mathrm{r_0}}^4\,x^2\,y^4\,z + 45\,\mathrm{C_{40}}\,{\mathrm{r_0}}^4\,x^4\,y^2\,z - 12\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,x^6\,z\,\left(x^2+y^2+z^2\right) \\
&+ 60\,\mathrm{C_{40}}\,{\mathrm{r_0}}^4\,x^4\,z\,\left(x^2+y^2+z^2\right) - 12\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,y^6\,z\,\left(x^2+y^2+z^2\right) \\
&+ 48\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^2\,z^7\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) + 144\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^4\,z^5\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) \\
&+ 144\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^6\,z^3\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) + 1800\,\mathrm{C_{42}}\,{\mathrm{r_0}}^4\,x^2\,z^5\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) \\
&+ 1500\,\mathrm{C_{42}}\,{\mathrm{r_0}}^4\,x^4\,z^3\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) + 4200\,\mathrm{C_{44}}\,{\mathrm{r_0}}^4\,x^2\,z^5\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right) \\
&+ 8400\,\mathrm{C_{44}}\,{\mathrm{r_0}}^4\,x^4\,z^3\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right) + 48\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,y^2\,z^7\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) \\
&+ 144\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,y^4\,z^5\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) + 144\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,y^6\,z^3\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) \\
&+ 1800\,\mathrm{C_{42}}\,{\mathrm{r_0}}^4\,y^2\,z^5\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) + 1500\,\mathrm{C_{42}}\,{\mathrm{r_0}}^4\,y^4\,z^3\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) \\
&+ 4200\,\mathrm{C_{44}}\,{\mathrm{r_0}}^4\,y^2\,z^5\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right) + 8400\,\mathrm{C_{44}}\,{\mathrm{r_0}}^4\,y^4\,z^3\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right) \\
&- 144\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,x^2\,y^2\,z^5 - 216\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,x^2\,y^4\,z^3 - 216\,\mathrm{C_{20}}\,{\mathrm{r_0}}^2\,x^4\,y^2\,z^3 - 210\,\mathrm{C_{40}}\,{\mathrm{r_0}}^4\,x^2\,y^2\,z^3 \\
&+ 192\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^2\,y^6\,z\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) + 288\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^4\,y^4\,z\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) \\
&+ 192\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^6\,y^2\,z\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) - 900\,\mathrm{C_{42}}\,{\mathrm{r_0}}^4\,x^2\,y^4\,z\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) \\
&- 900\,\mathrm{C_{42}}\,{\mathrm{r_0}}^4\,x^4\,y^2\,z\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) + 12600\,\mathrm{C_{44}}\,{\mathrm{r_0}}^4\,x^2\,y^4\,z\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right) \\
&+ 12600\,\mathrm{C_{44}}\,{\mathrm{r_0}}^4\,x^4\,y^2\,z\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right) + 72\,\mathrm{C_{22}}\,{\mathrm{r_0}}^2\,x^6\,z\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\left(x^2+y^2+z^2\right)
\end{aligned}
$$

Continuation of Equation U_z

$$
\begin{aligned}
&- 960\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,x^4\,z\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 1680\,\mathrm{C}_{44}\,{\mathrm{r}_0}^4\,x^4\,z\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 72\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,y^6\,z\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&- 960\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,y^4\,z\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right) - 80\,\mathrm{C}_{40}\,{\mathrm{r}_0}^4\,y^2\,z^3\,\left(x^2+y^2+z^2\right)\\
&+ 1680\,\mathrm{C}_{44}\,{\mathrm{r}_0}^4\,y^4\,z\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right) + 120\,\mathrm{C}_{40}\,{\mathrm{r}_0}^4\,x^2\,y^2\,z\,\left(x^2+y^2+z^2\right)\\
&- 36\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,x^2\,y^4\,z\,\left(x^2+y^2+z^2\right) - 36\,\mathrm{C}_{20}\,{\mathrm{r}_0}^2\,x^4\,y^2\,z\,\left(x^2+y^2+z^2\right)\\
&+ 288\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^2\,y^2\,z^5\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) + 432\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^2\,y^4\,z^3\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&+ 432\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^4\,y^2\,z^3\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right) + 3000\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,x^2\,y^2\,z^3\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\\
&+ 16800\,\mathrm{C}_{44}\,{\mathrm{r}_0}^4\,x^2\,y^2\,z^3\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right) + 72\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^2\,z^5\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 144\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^4\,z^3\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right) + 60\,\mathrm{C}_{40}\,{\mathrm{r}_0}^4\,y^4\,z\,\left(x^2+y^2+z^2\right)\\
&+ 720\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,x^2\,z^3\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 1680\,\mathrm{C}_{44}\,{\mathrm{r}_0}^4\,x^2\,z^3\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 72\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,y^2\,z^5\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 144\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,y^4\,z^3\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 720\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,y^2\,z^3\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 1680\,\mathrm{C}_{44}\,{\mathrm{r}_0}^4\,y^2\,z^3\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 216\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^2\,y^4\,z\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 216\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^4\,y^2\,z\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&- 1920\,\mathrm{C}_{42}\,{\mathrm{r}_0}^4\,x^2\,y^2\,z\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 3360\,\mathrm{C}_{44}\,{\mathrm{r}_0}^4\,x^2\,y^2\,z\,\cos\left(4\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\\
&+ 288\,\mathrm{C}_{22}\,{\mathrm{r}_0}^2\,x^2\,y^2\,z^3\,\cos\left(2\,\arctan\left(\frac{y}{x}\right)\right)\,\left(x^2+y^2+z^2\right)\Big]
\end{aligned}
$$

List of Acronyms

CAM	Collision Avoidance Maneuver
FTTG	Forced Terminal Translation Guide
INTG	Impulsive Nominal Translational Orientation
OSTG	Orbit Synchronization Translational Guidance
ISS	International Space Station
SAA	Sample Average Approximation
MPC	Model Predictive Control
ROM	Read-only memory
PWM	Pulse Width Modulation
EKF	Extended Kalman Filter
SQP	Sequential Quadratic Programming
OCP	Optimal Control Problem
LOS	Line of Sight
QP	Quadratic Programming
RPO	Rendezvous and proximity Operations
CWH	Clohessy-Wiltshire-Hill
GNC	Guidance, navigation and control
RHC	Receding Horizon Control
PID	Proportional-Integral-Differential
LQR	Linear Quadratic Regulator
MIMO	Multiple inputs-Multiple outputs
SMPC	Stochastic Model Predictive Control

NLP	Nonlinear Programming Problem
QSM	Quasi Monte-Carlo
MSL	Mars Science Laboratory
NMPC	Nonlinear Model Predictive Control
NOCP	Nonlinear Optimal Control Problem
HJB	Hamilton-Jacobi-Bellman
BVP	Boundary Value Problem
ODE	Ordinary Differential Equation
IVP	Initial Value Problem
DAE	Differential Algebraic Equation
SCC	Single Chance Constraint
JCC	Joint Chance Constraint
PDF	Probability Density Function

List of Figures

List of Tables

Bibliography

[1] "Freely usable media files," https://commons.wikimedia.org, accessed: 2017-02-02.

[2] "The importance of exploration," https://www.nasa.gov/missions/solarsystem/Why_We_01pt1.html, accessed: 2017-02-02.

[3] A. Guiggiani, I. Kolmanovsky, P. Patrinos, and A. Bemporad, "Fixed-point constrained model predictive control of spacecraft attitude," in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 2317–2322.

[4] F. Gavilan, R. Vazquez, and E. F. Camacho, "Chance-constrained model predictive control for spacecraft rendezvous with disturbance estimation," *Control Engineering Practice*, vol. 20, no. 2, pp. 111–122, 2012.

[5] G. W. Hill, "Researches in the lunar theory," *American Journal of Mathematics*, vol. 1, no. 1, pp. 5–26, 1878. [Online]. Available: http://www.jstor.org/stable/2369430

[6] W. Clohessy and R. Wiltshire, "Terminal guidance system for satellite rendezvous." *J. Aerosp. Sci.*, vol. 27, pp. 653–658, 674, 1960.

[7] R. Vazquez, F. Gavilan, and E. F. Camacho, "Model predictive control for spacecraft rendezvous in elliptical orbits with on/off thrusters," *IFAC-PapersOnLine*, vol. 48, no. 9, pp. 251–256, 2015.

[8] ——, "Trajectory planning for spacecraft rendezvous with on/off thrusters," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 8473–8478, 2011.

[9] ——, "Trajectory planning for spacecraft rendezvous in elliptical orbits with on/off thrusters," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 9703–9708, 2014.

[10] T. J and P. Hempel, "Rendezvous zu einem in elliptischer bahn umlaufenden ziel," *Astronautica Acta*, vol. 11, no. 2, p. 104, 1965.

[11] A. Weiss, I. Kolmanovsky, M. Baldwin, and R. S. Erwin, "Model predictive control of three dimensional spacecraft relative motion," in *2012 American Control Conference (ACC)*. IEEE, 2012, pp. 173–178.

[12] C. Jewison, R. S. Erwin, and A. Saenz-Otero, "Model predictive control with ellipsoid obstacle constraints for spacecraft rendezvous," *IFAC-PapersOnLine*,

vol. 48, no. 9, pp. 257 – 262, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S240589631500960X

[13] E. N. Hartley, P. A. Trodden, A. G. Richards, and J. M. Maciejowski, "Model predictive control system design and implementation for spacecraft rendezvous," *Control Engineering Practice*, vol. 20, no. 7, pp. 695 – 713, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066112000664

[14] H. Park, S. D. Cairano, and I. Kolmanovsky, "Model predictive control of spacecraft docking with a non-rotating platform," *{IFAC} Proceedings Volumes*, vol. 44, no. 1, pp. 8485 – 8490, 2011, 18th {IFAC} World Congress. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667016449736

[15] S. Di Cairano, H. Park, and I. Kolmanovsky, "Model predictive control approach for guidance of spacecraft rendezvous and proximity maneuvering," *International Journal of Robust and Nonlinear Control*, vol. 22, no. 12, pp. 1398–1427, 2012.

[16] W. Ruoyan, R. Xiaogang, Z. Xiaoqing, O. Sie, and X. Yao, "A method of guidance, navigation and control for soft landing on asteroid based on constrained mpc," in *Proceeding of the 11th World Congress on Intelligent Control and Automation*, June 2014, pp. 4213–4217.

[17] Z. Zexu, W. Weidong, L. Litao, H. Xiangyu, C. Hutao, L. Shuang, and C. Pingyuan, "Robust sliding mode guidance and control for soft landing on small bodies," *Journal of the Franklin Institute*, vol. 349, no. 2, pp. 493 – 509, 2012, advances in Guidance and Control of Aerospace Vehicles using Sliding Mode Control and Observation Techniques. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0016003211001797

[18] S. S. Haykin, *Kalman Filtering and Neural Networks*. New York, NY, USA: John Wiley & Sons, Inc., 2001.

[19] S. D. Cairano, A. Ulusoy, and S. Haghighat, "Soft-landing control by control invariance and receding horizon control," in *2014 American Control Conference*, June 2014, pp. 784–789.

[20] J. Carson and A. Acikmese, "A model-predictive control technique with guaranteed resolvability and required thruster silent times for small-body proximity operations," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006, p. 6780.

[21] J. M. Maciejowski, *Predictive control: with constraints*. Pearson education, 2002.

[22] D. Q. Mayne, "Control of constrained dynamic systems," *European Journal of Control*, vol. 7, no. 2-3, pp. 87–99, 2001.

[23] J. B. Rawlings, "Tutorial overview of model predictive control," *IEEE Control Systems*, vol. 20, no. 3, pp. 38–52, Jun 2000.

[24] J. Tamimi and P. Li, "Nonlinear model predictive control using multiple shooting combined with collocation on finite elements," *{IFAC} Proceedings Volumes*, vol. 42, no. 11, pp. 703 – 708, 2009, 7th {IFAC} Symposium on Advanced Control of Chemical Processes. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667015303578

[25] J. B. Rawlings and D. Q. Mayne, *Model predictive control: Theory and design*. Nob Hill Pub., 2009.

[26] M. L. Correa Cordova, "High performance implementation of MPC schemes for fast systems," Masterarbeit, Technische Universität Ilmenau, März 2016.

[27] J. Aburajabaltamimi, "Development of efficient algorithms for model predictive control of fast systems," Ph.D. dissertation, Ilmenau, Technische Universität Ilmenau, Diss., 2011, 2011.

[28] D. Q. Mayne and H. Michalska, "Receding horizon control of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 35, no. 7, pp. 814–824, Jul 1990.

[29] M. Diehl, H. Bock, H. Diedam, and P.-B. Wieber, *Fast Direct Multiple Shooting Algorithms for Optimal Robot Control*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 65–93. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-36119-0_4

[30] R. Bellman, *Dynamic Programming*, 1st ed. Princeton, NJ, USA: Princeton University Press, 1957. [Online]. Available: http://books.google.com/books?id=fyVtp3EMxasC&pg=PR5&dq=dynamic+programming+richard+e+bellman&client=firefox-a#v=onepage&q=dynamic%20programming%20richard%20e%20bellman&f=false

[31] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Athena Scientific, 2000.

[32] B. Passenberg, "Theory and algorithms for indirect methods in optimal control of hybrid systems," Ph.D. dissertation.

[33] R. Findeisen and F. Allgöwer, "An introduction to nonlinear model predictive control," in *21st Benelux Meeting on Systems and Control*, vol. 11. Technische Universiteit Eindhoven Veldhoven Eindhoven, The Netherlands, 2002, pp. 119–141.

[34] D. Kraft, *On Converting Optimal Control Problems into Nonlinear Programming Problems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1985, pp. 261–280. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-82450-0_9

[35] R. Sargent and G. Sullivan, "The development of an efficient optimal control package," in *Optimization Techniques*. Springer, 1978, pp. 158–168.

[36] M. Diehl and K. ESAT-SCD, "Numerical optimal control draft," 2011.

[37] Process Systems Enterprise , "gproms," 1997-2015. [Online]. Available: https://www.psenterprise.com/gproms

[38] Lehrstuhl für Prozesstechnik, RWTH Aachen, "User manual," 2001.

[39] L. Grüne, D. Nešić, and J. Pannek, *Model Predictive Control for Nonlinear Sampled-Data Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 105–113. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-72699-9_8

[40] L. T. Biegler, "Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation," *Computers and Chemical Engineering*, vol. 8, no. 3, pp. 243–247, 1984. [Online]. Available: http://www.sciencedirect.com/science/article/pii/009813548487012X

[41] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006. [Online]. Available: http://dx.doi.org/10.1007/s10107-004-0559-y

[42] O. von Stryk, "User's Guide for DIRCOL. A Direct Collocation Method for the Numerical Solution of Optimal Control Problems," 1999. [Online]. Available: https://www.sim.informatik.tu-darmstadt.de/en/res/sw/dircol/info1/

[43] H. G. Bock and K.-J. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems." PROCEEDINGS OF THE IFAC WORLD CONGRESS, 1984.

[44] D. B. Leineweber, A. Schäfer, H. G. Bock, and J. P. Schlöder, "An efficient multiple shooting based reduced {SQP} strategy for large-scale dynamic process optimization: Part ii: Software aspects and applications," *Computers and Chemical Engineering*, vol. 27, no. 2, pp. 167 – 174, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0098135402001953

[45] L. O. Santos, P. A. Afonso, J. A. Castro, N. M. Oliveira, and L. T. Biegler, "On-line implementation of nonlinear mpc: an experimental case study," *Control Engineering Practice*, vol. 9, no. 8, pp. 847 – 857, 2001, advanced Control of Chemical Processes. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066101000491

[46] M. Diehl, L. Magni, and G. D. Nicolao, "Efficient {NMPC} of unstable periodic systems using approximate infinite horizon closed loop costing," *Annual Reviews in Control*, vol. 28, no. 1, pp. 37 – 45, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1367578804000057

[47] F. Topputo and C. Zhang, "Survey of direct transcription for low-thrust space trajectory optimization with applications," in *Abstract and Applied Analysis*, vol. 2014. Hindawi Publishing Corporation, 2014.

[48] R. Burden and J. Faires, *Numerical Analysis*. Cengage Learning, 2010. [Online]. Available: https://books.google.de/books?id=zXnSxY9G2JgC

[49] S. C. Chapra and R. Canale, *Numerical Methods for Engineers*, 5th ed. New York, NY, USA: McGraw-Hill, Inc., 2006.

[50] W. Sun and Y.-X. Yuan, *Optimization theory and methods: nonlinear programming*. Springer Science & Business Media, 2006, vol. 1.

[51] L. T. Biegler, *Nonlinear programming: concepts, algorithms, and applications to chemical processes*. SIAM, 2010.

[52] S. Wright and J. Nocedal, "Numerical optimization," *Springer Science*, vol. 35, pp. 67–68, 1999.

[53] Q. D. Vu and P. Li, "A reduced-space interior-point quasi-sequential approach to nonlinear optimization of large-scale dynamic systems," in *2010 IEEE RIVF International Conference on Computing Communication Technologies, Research, Innovation, and Vision for the Future (RIVF)*, Nov 2010, pp. 1–6.

[54] W. Hong, S. Wang, P. Li, G. Wozny, and L. T. Biegler, "A quasi-sequential approach to large-scale dynamic optimization problems," *AIChE Journal*, vol. 52, no. 1, pp. 255–268, 2006.

[55] M. Bartl, P. Li, and L. T. Biegler, "Improvement of state profile accuracy in nonlinear dynamic optimization with the quasi-sequential approach," *AIChE Journal*, vol. 57, no. 8, pp. 2185–2197, 2011.

[56] N. I. M. Gould and P. L. Toint, *SQP Methods for Large-Scale Nonlinear Programming*. Boston, MA: Springer US, 2000, pp. 149–178. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-35514-6_7

[57] C. Büskens and H. Maurer, "Sqp-methods for solving optimal control problems with control and state constraints: adjoint variables, sensitivity analysis and real-time control," *Journal of Computational and Applied Mathematics*, vol. 120, no. 1–2, pp. 85 – 108, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377042700003058

[58] A. R. Conn, N. I. M. Gould, and P. L. Toint, "Convergence of quasi-newton matrices generated by the symmetric rank one update," *Mathematical Programming*, vol. 50, no. 1, pp. 177–195, 1991. [Online]. Available: http://dx.doi.org/10.1007/BF01594934

[59] M. Loke and T. Dahlin, "A comparison of the gauss–newton and quasi-newton methods in resistivity imaging inversion," *Journal of Applied Geophysics*, vol. 49, no. 3, pp. 149–162, 2002.

[60] N. Karmarkar, "A new polynomial-time algorithm for linear programming," in *Proceedings of the sixteenth annual ACM symposium on Theory of computing*. ACM, 1984, pp. 302–311.

[61] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, "User's guide for npsol (version 4.0): A fortran package for nonlinear programming." DTIC Document, Tech. Rep., 1986.

[62] P. E. Gill, W. Murray, and M. A. Saunders, "Snopt: An sqp algorithm for large-scale constrained optimization," *SIAM review*, vol. 47, no. 1, pp. 99–131, 2005.

[63] M. R. Bussieck and A. Meeraus, *General Algebraic Modeling System (GAMS)*. Boston, MA: Springer US, 2004, pp. 137–157. [Online]. Available: http://dx.doi.org/10.1007/978-1-4613-0215-5_8

[64] K. Schittkowski, "Nlpql: A fortran subroutine solving constrained nonlinear programming problems," *Annals of Operations Research*, vol. 5, no. 2, pp. 485–500, 1986. [Online]. Available: http://dx.doi.org/10.1007/BF02022087

[65] A. Barclay, P. E. Gill, and J. Ben Rosen, *SQP Methods and their Application to Numerical Optimal Control*. Basel: Birkhäuser Basel, 1998, pp. 207–222. [Online]. Available: http://dx.doi.org/10.1007/978-3-0348-8802-8_21

[66] A. Wächter, "Short tutorial: getting started with ipopt in 90 minutes," in *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2009.

[67] A. Wächter and L. T. Biegler, "Line search filter methods for nonlinear programming: Local convergence," *SIAM J. on Optimization*, vol. 16, no. 1, pp. 32–48, May 2005. [Online]. Available: http://dx.doi.org/10.1137/S1052623403426544

[68] "ISO/IEC. (2014). ISO International Standard ISO/IEC 14882:2014(E) – Programming Language C++. [Working Draft]. Geneva, Switzerland: International Organization for Standardization (ISO)." 2014. [Online]. Available: https://isocpp.org/std/the-standard

[69] A. Griewank and A. Walther, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, 2nd ed. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2008.

[70] J. Andersson, "A general-purpose software framework for dynamic optimization," 2013.

[71] R. H. Byrd, J. Nocedal, and R. B. Schnabel, "Representations of quasi-newton matrices and their use in limited memory methods," *Mathematical Programming*, vol. 63, no. 1, pp. 129–156, 1994. [Online]. Available: http://dx.doi.org/10.1007/BF01582063

[72] J. Nocedal, "Updating quasi-newton matrices with limited storage," *Mathematics of Computation*, vol. 35, no. 151, pp. 773–782, 1980. [Online]. Available: http://www.jstor.org/stable/2006193

[73] I. S. Duff and J. K. Reid, *MA27-a set of Fortran subroutines for solving sparse symmetric sets of linear equations*. UKAEA Atomic Energy Research Establishment, 1982.

[74] I. S. Duff, "Ma57—a code for the solution of sparse symmetric definite and indefinite systems," *ACM Trans. Math. Softw.*, vol. 30, no. 2, pp. 118–144, Jun. 2004. [Online]. Available: http://doi.acm.org/10.1145/992200.992202

[75] P. Amestoy, I. Duff, and J.-Y. L'Excellent, "Multifrontal parallel distributed symmetric and unsymmetric solvers," *Computer Methods in Applied Mechanics and Engineering*, vol. 184, no. 2–4, pp. 501 – 520, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S004578259900242X

[76] O. Schenk, K. Gärtner, W. Fichtner, and A. Stricker, "Pardiso: a high-performance serial and parallel sparse linear solver in semiconductor device simulation," *Future Generation Computer Systems*, vol. 18, no. 1, pp. 69 – 78, 2001, i. High Performance Numerical Methods and Applications. II. Performance Data Mining: Automated Diagnosis, Adaption, and Optimization. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X00000765

[77] R. Chandra, L. Dagum, D. Kohr, D. Maydan, J. McDonald, and R. Menon, *Parallel Programming in OpenMP*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001.

[78] W. C. Navidi, *Statistics for engineers and scientists*. McGraw-Hill New York, 2006, vol. 2.

[79] A. Geletu, M. Klöppel, H. Zhang, and P. Li, "Advances and applications of chance-constrained approaches to systems optimisation under uncertainty," *International Journal of Systems Science*, vol. 44, no. 7, pp. 1209–1232, 2013. [Online]. Available: http://dx.doi.org/10.1080/00207721.2012.670310

[80] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust optimization*. Princeton University Press, 2009.

[81] S. Asmussen and P. W. Glynn, *Stochastic simulation: algorithms and analysis*. Springer Science & Business Media, 2007, vol. 57.

[82] B. K. Pagnoncelli, S. Ahmed, and A. Shapiro, "Sample average approximation method for chance constrained programming: Theory and applications," *Journal of Optimization Theory and Applications*, vol. 142, no. 2, pp. 399–416, 2009. [Online]. Available: http://dx.doi.org/10.1007/s10957-009-9523-6

[83] I. M. Sobol, D. Asotsky, A. Kreinin, and S. Kucherenko, "Construction and comparison of high-dimensional sobol'generators," *Wilmott*, vol. 2011, no. 56, pp. 64–79, 2011.

[84] M. Wendt, P. Li, and G. Wozny, "Nonlinear chance-constrained process optimization under uncertainty," *Industrial & engineering chemistry research*, vol. 41, no. 15, pp. 3621–3629, 2002.

[85] A. Geletu, A. Hoffmann, M. Klöppel, and P. Li, "Monotony analysis and sparse-grid integration for nonlinear chance constrained process optimization," *Engineering Optimization*, vol. 43, no. 10, pp. 1019–1041, 2011. [Online]. Available: http://dx.doi.org/10.1080/0305215X.2010.532552

[86] A. Geletu, M. Klöppel, A. Hoffmann, and P. Li, "A tractable approximation of non-convex chance constrained optimization with non-gaussian uncertainties," *Engineering Optimization*, vol. 47, no. 4, pp. 495–520, 2015.

[87] J. Pinter, "Deterministic Approximation of Probability Inequalities," *Operations Research*, vol. 33, p. 219–239, 1989.

[88] A. Nemirovski and A. Shapiro, "Convex approximations of chance constrained programs," *SIAM Journal on Optimization*, vol. 17, no. 4, pp. 969–996, 2007. [Online]. Available: http://dx.doi.org/10.1137/050622328

[89] R. T. Rockafellar and S. Uryasev, "Optimization of conditional value-at-risk."

[90] Z. Zhang, "Determining the epipolar geometry and its uncertainty: A review," *International Journal of Computer Vision*, vol. 27, no. 2, pp. 161–195, 1998. [Online]. Available: http://dx.doi.org/10.1023/A:1007941100561

[91] "MATLAB and Symbolic Toolbox Release 2014a, The Mathworks, Inc., Natick, Massachusetts, United States." 2014. [Online]. Available: http://www.mathworks.com

[92] S. Li, P. Cui, and H. Cui, "Autonomous navigation and guidance for landing on asteroids," *Aerospace Science and Technology*, vol. 10, no. 3, pp. 239 – 247, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1270963805001690

[93] R. Zhang, L. Weng, W. Cai, M. Zhang, and Y. Song, "Neuro robust adaptive descending control of space vehicles," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 6525 – 6529, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S147466701639992X

[94] D.-C. Liaw, C.-C. Cheng, and Y.-W. Liang, "Three-dimensional guidance law for landing on a celestial object," *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 5, pp. 890–892, 2000.

[95] A. I. Kozynchenko, "Analysis of predictive entry guidance for a mars lander under high model uncertainties," *Acta Astronautica*, vol. 68, no. 1–2, pp. 121 – 132, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0094576510002936

[96] C. Wu, S. Li, J. Yang, and L. Guo, "Mars entry trajectory tracking via constrained multi-model predictive control," in *Proceedings of the 33rd Chinese Control Conference*, July 2014, pp. 7805–7810.

[97] M. Galassi and et al, "GNU Scientific Library Reference Manual (3rd ed.)." [Online]. Available: http://www.gnu.org/software/gsl/

[98] J. Andersson, "A General-Purpose Software Framework for Dynamic Optimization," PhD thesis, Arenberg Doctoral School, KU Leuven, Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium, October 2013.