POLITECNICO DI MILANO

Facoltà di Ingegneria Industriale

Corso di Laurea in
Ingegneria Spaziale

An Indirect Optimal Control Approach for Propellantless Orbital Rendez-Vous

Relatore: Prof. Franco Bernelli Zazzera

Tesi di Laurea di:

Brunella Carlomagno Matr: 765994

Anno Accademico 2012-2013

Acknowledgements

I would like to thank M. Franco Bernelli Zazzera for accepting to be my supervisor at Politecnico di Milano.

A special thanks to M. Lamberto dell'Elce, for his precious help during this work and his time spent to answer to all my questions.

I would like also to thank M. Pierluigi Di Lizia, for the simple idea that changes the fate of this work.

En Belgique...
Je tiens tout d'abord à remercier Vladi, Sarah, Dave, François, Céline, Thibault et tout le Master Aéro, pour l'accueil et l'affection de ces deux merveilleuses années à Liège.
Je remercie chaleureusement Marco et Pablo, pour la familiarité du temps passé ensemble: c'était comme avoir deux frères, qui m'embêtaient tout le temps, mais que j'adore.
Et enfin, Romain, pour être la meilleure surprise que la Belgique pouvait me réserver.

In Italia...
Un grazie speciale a tutti gli amici di Milano: senza di voi, l'università e la vita oltre il Poli non sarebbero state le stesse.
Agli amici di Matera, con i quali sono cresciuta (orsù, si fa per dire) e che mi hanno sempre incoraggiata.
A Cinzia, per essere stata una guida, accademicamente e umanamente.
Alle fantastiche 3: Anna, Barbara e Marianna, perchè è semplicemente naturale ritrovarsi anche dopo un anno come se fosse passato solo un giorno.
Ad Anto, per essere la mia ventata di gioia, di allegria, di amicizia incondizionata.
Infine, la mia famiglia: mia mamma, per essere la vera roccia della mia esistenza, e mia sorella, per essere la mia coperta di Linus. Grazie per l'infinita pazienza nel sopportare le mie nevrosi e il supporto nei momenti cruciali.

Un ultimo pensiero è per mio padre, che non è qui con me, ma che io sento sempre vicino. Alla fine, vorrei essere ricordata come lui: semplicemente con amore.

Contents

List of Figures

List of Tables

Politecnico di Milano

Abstract

Facoltà di Ingegneria Industriale

Corso di Laurea in Ingegneria Spaziale

An Indirect Optimal Control Approach for Propellantless Orbital Rendez-Vous

Author: Brunella Carlomagno

The implementation of an indirect method for the optimal trajectory of a rendez-vous manoeuvre using only the differential drag as actuator is presented. This propellantless technique exploits the exposed surface of the satellite in along-track direction to control the manoeuvre. The limit is in the small value of control, taking to the necessity of the trajectory optimization, which is obtained trough the implementation of an indirect method. Despite its great advantage of fast convergence and reduced number of variables, two important drawbacks have limited its diffusion: the necessity of the derivation of the optimality conditions and the costate initialization. The first one can be overcome with simpler models, when possible, or numerical differentiation; the second one has a much more complex solution. For this purpose, a "homotopic"-based approach has been implemented. The strategy has been tested through four models, with increasing complication, and several cases. The tests have given good results for simpler models, while for the more complex ones the strategy fails. Several changes have been proposed, without a remarkable improvement.

Keywords Differential Drag, Rendez-Vous, Optimal Control, Indirect Methods

Politecnico di Milano

Sommario

Facoltà di Ingegneria Industriale

Corso di Laurea in Ingegneria Spaziale

An Indirect Optimal Control Approach for Propellantless Orbital Rendez-Vous

Autore: Brunella Carlomagno

La dissertazione presenta l'implementazione di un metodo indiretto per il controllo ottimo del rendez-vous, utilizzando la resistenza differenziale come attuatore. Questa tecnica sfrutta la superficie esposta dal satellite nella direzione della velocità per controllare la manovra. Il limite è nel valore del controllo, molto piccolo, che quindi porta alla necessità di un'ottimizzazione della traiettoria, ottenuta attraverso l'implementazione del metodo indiretto. Nonostante i grandi vantaggi legati ad una convergenza molto veloce e a un ridotto numero di variabili, la diffusione di questi metodi è stata limitata da due inconvenienti: la necessità di derivare le condizioni di ottimalità e l'inizializzazione dei costati. Il primo può essere superato usando modelli più semplici, quando possibile, o la differenziazione numerica; il secondo necessita una soluzione più complessa. Per questo, un approccio "omotopico" è stato implementato. La strategia è stata testata attraverso quattro modelli, con complicazione crescente, e diversi casi. I test hanno dato buoni risultati per i modelli più semplici, mentre per i più complessi il metodo non ha funzionato. Nonostante i diversi cambiamenti proposti, nessun miglioramento è stato ottenuto.

Parole Chiave Resistenza Differenziale, Rendez-Vous, Controllo Ottimo, Metodi Indiretti

Chapter 1

Introduction

The objective of this thesis is to provide the numerical implementation of an indirect method for the optimal control problem of the rendez-vous manoeuvre.

The particular problem has been extensively studied over the years, starting from the "pioneering" [6] work of Leonard [3]. With her Master thesis at the MIT, she gave birth to a totally new way of thinking the propulsion for the rendez-vous manoeuvres. As for the cars in our towns, the fuel has a cost, but in space applications, this cost is normally linked to its weight: when more fuel is necessary, less instruments can be put on board. Especially nowadays, several propellantless techniques have been introduced, as the famous solar sails or the less famous geomagnetic systems. The idea of exploiting the atmospheric drag has the originality of changing what normally is considered an "enemy" into an helping friend. Originally, drag-plates were added to the satellite to operate the change in the differential drag; lately the system has been reduced to the simple attitude control, exploiting directly the exposed surface of the satellite. Lower values of control have been obtained, but smaller weight is considered. The first practical implementation is the $JC2Sat$ formation flying mission. It represents a collaboration between the Canadian and the Japanes Space Agencies with validation of this new actuator as main objective. Unfortunately, the mission has not yet been launched, preventing from data analysis. The interest is still growing and the *Université de Liège* has in charge to validate it with the $QB50$ nanosatellites constellation.

The second aspect of this dissertation is the numerical implementation of the previous idea. The trajectory optimization has a much longer history than the differential drag as actuator, implying much more information and

possibilities. The indirect methods are part of these. With the great advantage of the optimal solution and the smaller computational cost with respect to direct methods, they have been less developed: the cause is in their great disadvantage, the initialization. The main difference between the two typologies is in the definition of the control: the direct methods guess its structure, modifying it at each iteration; the indirect methods find it algebraically, through the action of the costate variables. These are the hearth of the matter: they do not have a real physical meaning, preventing from a definite initialization. Several approaches have been implemented to overcome this issue, the most famous is the *hybrid* one: mixing direct and indirect methods. The objective of this thesis is the indirect method implementation, without the participation of the direct ones. For this purpose, new strategies have been tried: the most promising is the *homotopic* approach, or *continuation* method. Its basic idea has been exploited, with a small modification.
The methodology followed during this thesis can be summarized in two words: reading and coding. The acquisition of information through books, articles, papers, Internet sites has been long and interesting; the numerical implementation often failing. The idea was to get as much information as possible on a particular strategy and implement it in a code. Often, this part has been stopped before getting any results, because of the advent of new issues. This trying methodology has the great advantage of considering a huge knowledge, easily spendable in the future. However, this also implies a slowdown of the work.

The numerical instrument is the *Matlab*® program. The reason is in its huge *global* diffusion, allowing the consultation of several tutorials and forums, and its *academic* diffusion, it represents one of the most used in trajectory optimization, together with Fortran.

Outline of the thesis

This dissertation is organized as follows: Chapter 2 and 3 are dedicated to the theoretical explication of the differential drag as actuator (Chapter 2) and the optimal control theory (Chapter 3). This last one contains also a compendium of methods, with a small description of each. The real implementation starts with Chapter 4, where the analysed models for the problem are presented. Chapter 5 considers the analytical solution. The hearth of the thesis is Chapter 6: the strategy is explained and implemented, analysing the results. Finally, the conclusions and future perspectives are presented in Chapter 7.

Chapter 2

"... Propellantless Orbital Rendez-Vous"

The rendez-vous consists in a series of manoeuvres done by a space object to approach another one. They can be two satellites, a spacecraft and a space station (as for the ATV and ISS), a satellite and a planet, a satellite and a point, even if in these last two cases, the definition is not completely exact. In this thesis, the case of two satellites is taken into account. By convention, the active control satellite is called *Chaser* and the followed one is the *Target*.

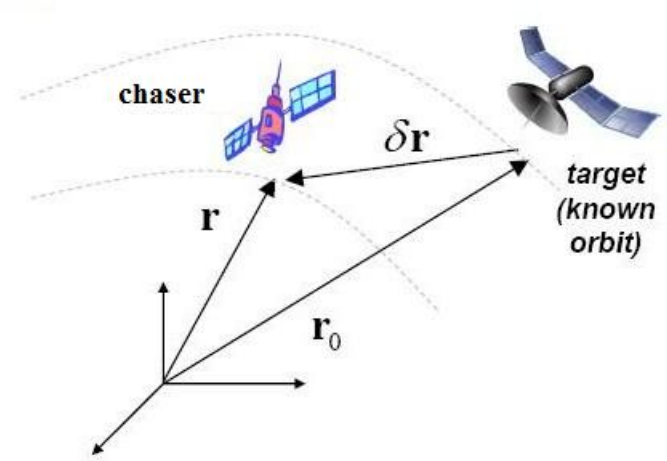


Figure 2.1: Relative motion between target and chaser (from [1])

2.1 Differential Drag

The control of orbits in a rendez-vous manoeuvre is normally operated trough a propulsion system, as mono-propellant with hydrazine, electric systems. The most important advantage of this type of propulsion is the precision, while the disadvantages concern two different aspects: the weight and the contamination. Any propellant system needs several components to operate: a tank for the propellant, tubes, a transformation system. All these increase the weight: it is evident that the more weight is used for the satellite operating systems, the less can be dedicated to the scientific instruments. The second disadvantage is the contamination: some instruments - as telescopes, solar panels - are very sensitive with respect to pollutant particles, that can alter the optical properties. The disadvantages of this type of propulsion become the advantages for a different type: propellanteless technique. The result is a simpler satellite where the absence of propellant avoids the contamination.
The propellantless technique examined in this thesis is the *Differential Drag*. Directly from the name, the physical principle is understandable: a difference in drag acting on each satellite creates a change in relative positions. This technique requires attitude or geometry changes to maximize or minimize the amount of atmospheric drag a satellite experiences, in order to maintain or speed up its orbital velocity to change the orbit itself and operate the rendez-vous manoeuvre.
The aerodynamic force acting on a flying object is typically decomposed into two components: lift, acting perpendicularly to the velocity direction, and drag, acting in this direction. For a satellite in LEO, the first is negligible, the second is the largest nonconservative force, as shown in figure 2.4. The drag acceleration is function of several parameters:

$$\mathbf{a}_{drag} = -\frac{1}{2}\rho BC v_r^2 \frac{\mathbf{v}_r}{v_r} \tag{2.1}$$

where:

- ρ is the atmospheric density;
- BC is the ballistic coefficient: $BC = \frac{C_D S}{m}$;
  - C_D is the drag coefficient: a dimensionless quantity that represents the extents to which the satellite is susceptible to atmospheric drag;
  - S is the cross-sectional area in velocity direction;
  - m is the mass of the satellite;

- v_r is the velocity of the spacecraft relative to the atmosphere;
- $\frac{\mathbf{v}_r}{v_r}$ indicates the velocity direction.

The differential drag is a difference in the atmospheric drag experienced by target and chaser, this means that one or several of these parameters are different between the two satellites:

- the atmospheric density: actually, this parameter is the most difficult to model, since altitude, temperature, solar and geomagnetic activities can significantly change it. Several models exist, as the Jacchia-Roberts, Harris-Priester, but it is still a challenging problem. For this reason, as first approximation, it has been considered constant: if the altitude variations remain in certain limits, this assumption is legitimate.
- the velocity of the spacecraft: it is considered constant, since the target's orbit is circular.
- the ballistic coefficient: it is in this parameter that the difference is created, in particular the easiest way is the change in cross-sectional area. The drag coefficient is function of several elements, as the shape, the material, the Mach number, which are quite difficult to modify on orbit. The mass is considered constant, since no propellant is used. The last element is the cross-sectional area: this difference can be accomplished by attitude or geometrical change, as shown in section 4 and Chapter 4.

The differential drag is then expressed as:

$$\mathbf{a}_{DiffDrag} = -\frac{1}{2}\rho\Delta BC v_r^2 \frac{\mathbf{v}_r}{v_r} \tag{2.2}$$

or in scalar form (since the only effect is in the velocity direction, which coincides with the y axis of the LVLH reference frame):

$$a_{DiffDrag} = -\frac{1}{2}\rho\Delta BC v_r^2 \tag{2.3}$$

where ΔBC represents the difference in ballistic coefficients between target and chaser. In particular, if the satellites are similar, the drag coefficients can be considered constant:

$$\Delta BC = BC_{Chaser} - BC_{Target} = \frac{C_{D_C} S_C}{m_C} - \frac{C_{D_T} S_T}{m_T} = C_D \left(\frac{S_C}{m_C} - \frac{S_T}{m_T} \right) \tag{2.4}$$

where m_C, m_T are respectively the mass of the chaser and the target and S_C, S_T are the surfaces.

2.2 Equations

In order to derive the set of equations for the rendez-vous manoeuvre, several assumptions have to be taken into account:

- orbit dimensions are far higher than the relative distance: $\delta r \ll r, r_0$
- the target's orbit is circular : the orbital rate ω is constant

The first assumption is quite easy to understand: a rendez-vous manoeuvre starts when the distance between the two satellites is relatively small (<100 km), while the orbit dimensions - as the semi-major axis - are much bigger. This aspect takes to a linearization procedure and the concern to the only relative motion dynamics.
The second one is a simplification: the target's orbit is considered known and circular. In this way, less variables have to be taken into account and the set of equations is simplified.

The derivation [1] of the equations starts from the classical "two-body problem" dynamics and, considering the assumptions, a set of linear ordinary differential equations is obtained:

$$\delta\ddot{\mathbf{r}} = -\frac{\mu}{r_0^3}\left(\delta\mathbf{r} - \frac{3\mathbf{r}_0.\delta\mathbf{r}}{r_0^2}\mathbf{r}_0\right) \tag{2.5}$$

Until now, the considered reference frame is still the inertial one; while the already mentioned relative motion calls for a different one: a switch to a local frame is then operated. The reference frame commonly used for this type of dynamics is the *Local Vertical Local Horizontal* (LVLH) reference frame (figure 2.2)

As the figure 2.2 shows, the center of the reference frame is the center of the target; the x ($\hat{i}$) axis points from the center of the Earth to the center of the target, the y ($\hat{j}$) axis points the direction of the motion, with positive versus as positive velocity direction; and z ($\hat{k}$) axis is normal to the orbit, to create a right-handed reference frame.
The coordinate transformation takes to the well-known Hill-Clohessy-Wiltshire equations:

$$\begin{cases} \ddot{x} - 2\omega\dot{y} - 3\omega^2 x = u_x \\ \ddot{y} + 2\omega\dot{x} = u_y \\ \ddot{z} + \omega^2 z = u_z \end{cases} \tag{2.6}$$

where x, y, z are the positions of the chaser with respect to the target; ω is the orbital rate of the target and u_x, u_y, u_z are the non-gravitational

[1] The complete derivation can be found in Appendix A

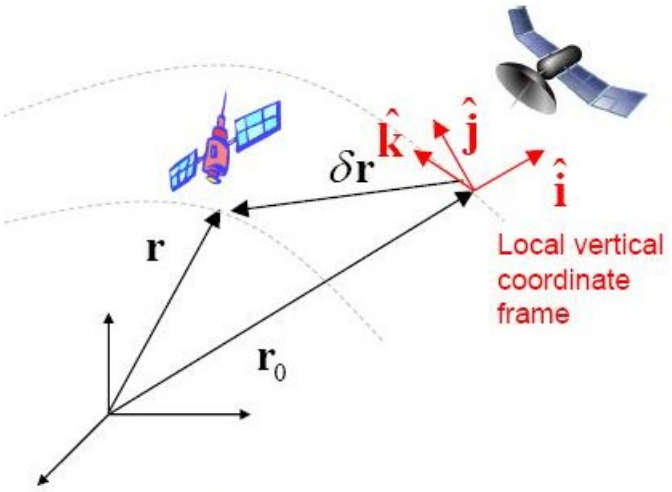


Figure 2.2: Local Reference Frame (from [1])

forces per unit mass, acting on the chaser.
Two simplifications can still be done:

- only differential drag: the x and z components of the non-gravitational forces are null, $u_x = u_z = 0$
- out-of-plane motion cannot be controlled only by differential drag: the z dynamics is separated from the other two, so no changes in inclination are considered;

The set of equations becomes:

$$\begin{cases} \ddot{x} - 2\omega\dot{y} - 3\omega^2 x = 0 \\ \ddot{y} + 2\omega\dot{x} = u_y \end{cases} \tag{2.7}$$

The unforced system has a simple solution :

$$\begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 4 - 3cos(\omega t) & 0 & \frac{sin(\omega t)}{\omega} & \frac{2}{\omega}\left(-cos(\omega t) + 1\right) \\ 6(sin(\omega t) - \omega t) & 1 & \frac{2}{\omega}\left(cos(\omega t) - 1\right) & \frac{4sin(\omega t)}{\omega} - 3t \\ 3\omega sin(\omega t) & 0 & cos(\omega t) & 2sin(\omega t) \\ 6\omega(cos(\omega t) - 1) & 0 & -2sin(\omega t) & 4cos(\omega t) - 3 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ \dot{x}_0 \\ \dot{y}_0 \end{bmatrix} \tag{2.8}$$

The relative motion is then an ellipse, where the center is constant in x-direction and moves away from the target in the y-direction, because of the secular terms, as shown in the figure 2.3.

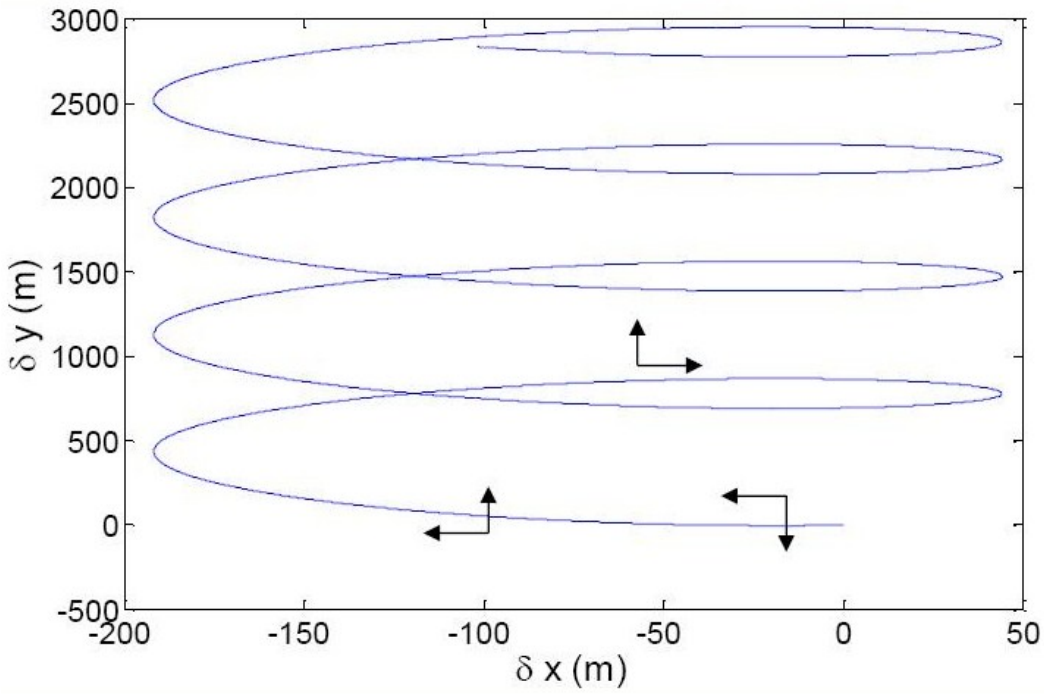


Figure 2.3: Relative position of the HST after releasing from Space Shuttle with not null initial velocity (from [1])

2.3 Including J_2 perturbation

In real life, satellites experience much more effects than the only gravitational one and the differential drag, as considered so far. In particular, the most important are:

- oblateness of the Earth;
- the solar pressure;
- the third body gravitational effect.

The last two effects become more important at higher altitude, where the atmospheric drag is less effective (figure 2.4).

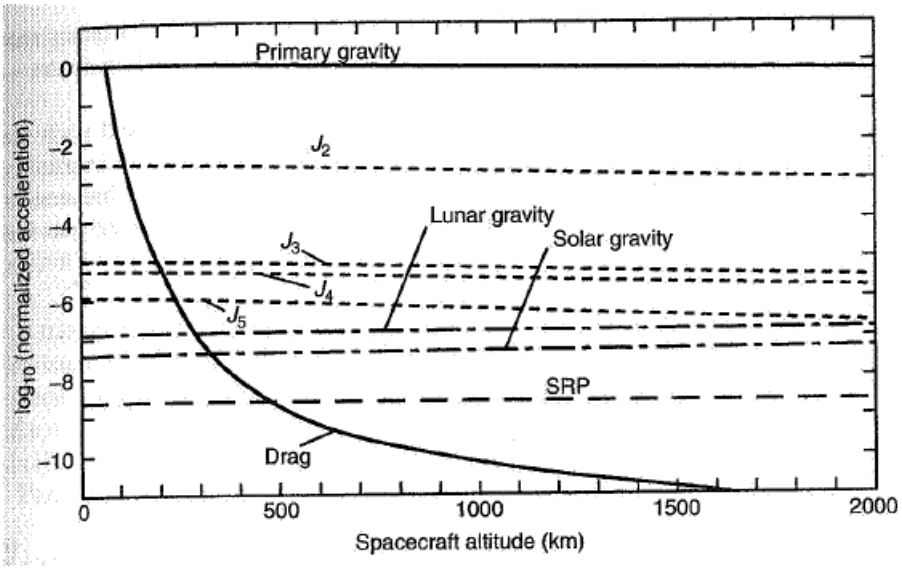


Figure 2.4: Relative importance of different perturbations with respect to the altitude (from [1])

It is then legitimate to neglect these two effects.
The real form of the Earth is more complicate than a sphere, because of the centrifugal force of the rotation that tends to increase the material at the Equator, but also the mass distribution on the surface (oceans, mountains) and in the inside (core, crust). The gravitational potential can be expressed in terms of the real mass distribution (which considers almost 5 millions parameters), taking to a very accurate but too expensive representation. Traditionally, these parameters are divided into two types: zonal harmonics and sectorial harmonics coefficients. The first ones dependent on the latitude, the second ones on the longitude. At a first approximation, the Earth can be considered an ellipsoid, in this way only the first zonal harmonic is taken into account. Its value is a thousand time larger than any other coefficients: $J_{2,0} = 0.001082$
The J_2 gravity perturbation has been incorporated in the rendez-vous set of equations by Scheweighart and Sedwick[2]. The linearized dynamic model is:

$$\begin{cases} \ddot{x} = 2(\omega c)\dot{y} + (5c^2 - 2)\omega^2 x + u_x \\ \ddot{y} = -2(\omega c)\dot{x} + u_y \\ \ddot{z} = -q^2 z + 2lqcos(qt + \phi) + u_z \end{cases} \tag{2.9}$$

where c is the Scheweighart-Sedwick coefficient that consider the J_2 perturbation, ω is still the orbital rate of the target (constant) and q, l and ϕ

are coefficients defined in Scheweighart-Sedwick work [2]

$$c = \sqrt{1 + \frac{3J_2R_\oplus^2}{8r_T^2}\left[1 + 3cos(2i_T)\right]} \tag{2.10}$$

where $R_\oplus$ is the Earth's mean radius ($R_\oplus$ = 6378 km), r_T is the target position vector in the inertial reference frame (ECI: Earth Centered Inertial), i_T is the inclination of the target orbit.
As for the Hill-Clohessy-Wiltshire equations, the only non gravitational force is the differential drag, which cannot control the out-of-plane motion. The set of equations is reduced to:

$$\begin{cases} \ddot{x} = 2\omega c\dot{y} + (5c^2 - 2)\omega^2 x \\ \ddot{y} = -2\omega c\dot{x} + u_y \end{cases} \tag{2.11}$$

2.4 State of the art

This concept of orbit controlling exploiting the differential drag has been analysed for the first time by Leonard [3]. In her work, the objective was the formationkeeping and the model was the simplified Hill-Clohessy-Wiltshire's. The differential drag was created by geometrically changing the satellite, in particular by adding drag plates, as shown in figure 2.5.

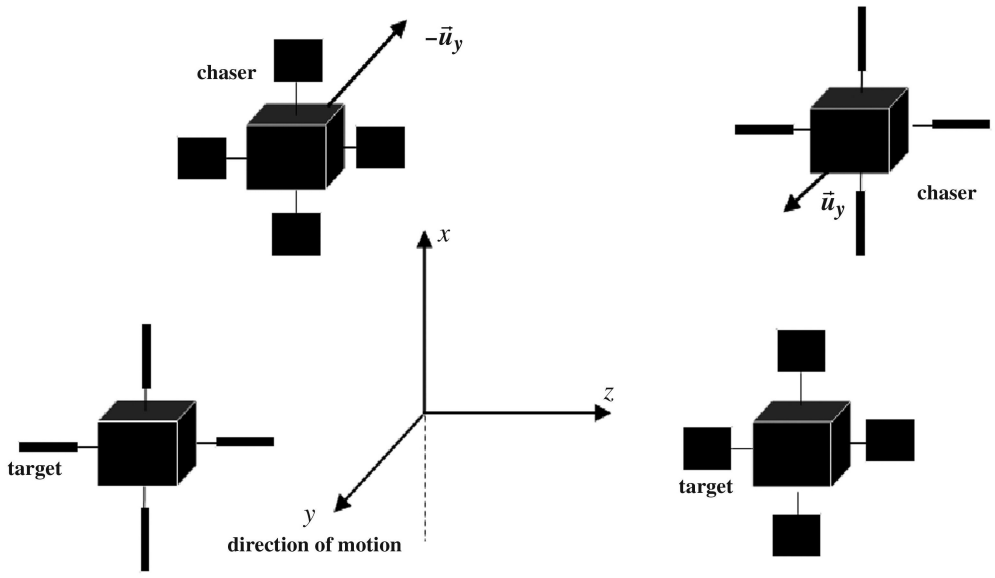


Figure 2.5: Differential Drag as actuator through drag plates (from [4])

These plates are present on both satellites and the angle of attack of the drag plate can assume only two values, 0 and 90 degrees. This results in four cases:

1. On both satellites, the angle is 0: no differential drag is created;
2. The drag plate on the target is at 90 degrees and the one on the chaser at 0: positive differential drag is created. It acts as a thruster for the chaser in the velocity direction;
3. The drag plate on the target is at 0 degrees and the one on the chaser at 90: negative differential drag is created. It acts as a thruster for the chaser in the direction opposite to the velocity;
4. On both satellites, the angle is 90: no differential drag is created. The difference with the first case is that this configuration makes the orbit decay more rapidly than before, because of the increment of the drag for both satellites.

Three assumptions are considered:

- The control law is time optimal;
- The change in differential drag is instantaneous and it does not affect the attitude of the satellite;
- The control law is of bang-bang type.

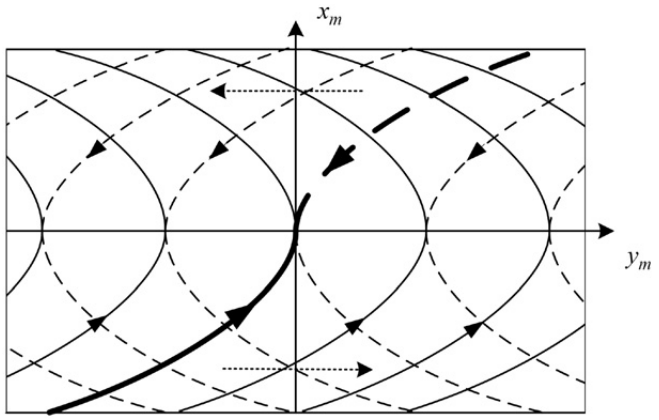
Introducing a separation of variables between mean (x_m, y_m) and oscillatory coordinates (α, β, with $\alpha = x - x_m$ and $\beta = y - y_m$), she decoupled the dynamics into two linear systems: a double integrator and an harmonic oscillator. The first describes the average position of the chaser with respect to the target; the second the eccentricity of the ellipse.

$$\begin{cases} \ddot{y}_m = -3u_y \\ \ddot{\beta} + \omega^2 \beta = 4u_y \end{cases} \tag{2.12}$$

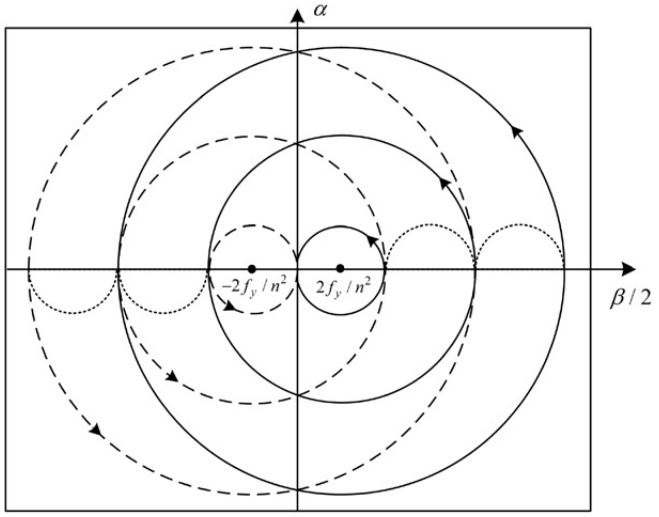
The control law is built in order to solve the double integrator and the harmonic oscillator simultaneously and dependently: in this way, once the average position is at the target's, bringing the periodic one to its target position does not move the first away from the designed point. In particular, it consists in two parts: the main control law and the gamma control scheme. The first drives the average position to the target's without an excessive increase in the eccentricity: a first switch between positive and negative

differential drag takes the states to the switching curve, where a second switch drives them to the origin through the switching curve itself (figure 2.6(a)).

The gamma control scheme reduces the eccentricity of the chaser with respect to the target, without driving away the average position. It starts when the main control law ends. It involves a series of switches between positive and negative differential drag twice per orbit until one of the small circles is reached, from there a final switch drives the states to the origin (figure 2.6(b)).



(a) Control trajectory in $x_m - y_m$ plane (from 5)



(b) Control trajectory in $\alpha - \beta/2$ plane (from 5)

Figure 2.6: Analytical solution (from [3])

Bevilacqua and Romano [4] improved the model including the J_2 gravitational perturbation, using the Scheweighart-Sedwick set of equations.

A slightly different decomposition takes anyway to a linear system: a double integrator and an harmonic oscillator:

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & -\frac{A}{A^2-B} & 0 \\ -\frac{AB}{A^2-B} & 0 & 0 & -\frac{B}{A^2-B} \\ 0 & 0 & \frac{A^2}{2(A^2-B)^{3/2}} & 0 \\ \frac{A^2B}{2(A^2-B)^{3/2}} & 0 & 0 & \frac{A^3}{2(A^2-B)^{3/2}} \end{bmatrix} \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} \tag{2.13}$$

with $A = 2\omega c$ and $B = (5c^2 - 2)\omega^2$.

The final set of equations is :

$$\begin{bmatrix} \dot{z_1} \\ \dot{z_2} \\ \dot{z_3} \\ \dot{z_4} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sqrt{A^2 - B} \\ 0 & 0 & -(A^2 - B) & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{B}{A^2-B} \\ 0 \\ \frac{A^3}{2(A^2-B)^2} \end{bmatrix} u_y \tag{2.14}$$

The use of this type of equations takes to a closed form solution for constant differential drag:

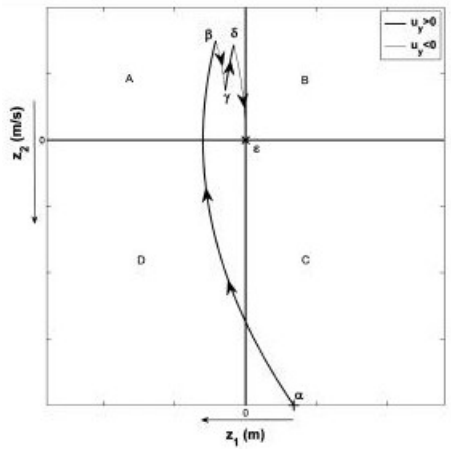
$$\begin{cases} z_1 = z_{1_0} + z_{2_0} t - \frac{Bt^2}{2(A^2-B)} u_y \\ z_2 = z_{2_0} - \frac{Bt}{(A^2-B)} u_y \\ z_3 = cos(\sqrt{A^2 - B}t) z_{3_0} + \frac{sin(\sqrt{A^2-B}t)}{\sqrt{A^2-B}} z_{4_0} + \frac{A^3[1-cos(\sqrt{A^2-B}t)]}{2(A^2-B)^{5/2}} u_y \\ z_4 = -\sqrt{A^2 - B} sin(\sqrt{A^2 - B}t) z_{3_0} + cos(\sqrt{A^2 - B}t) z_{4_0} + \frac{A^3 sin(\sqrt{A^2-B}t)}{2(A^2-B)^2} u_y \end{cases} \tag{2.15}$$

Assuming the same geometrical and inertial properties for target and chaser, they elaborated a two-phases control law, where states are driven to zero separately:

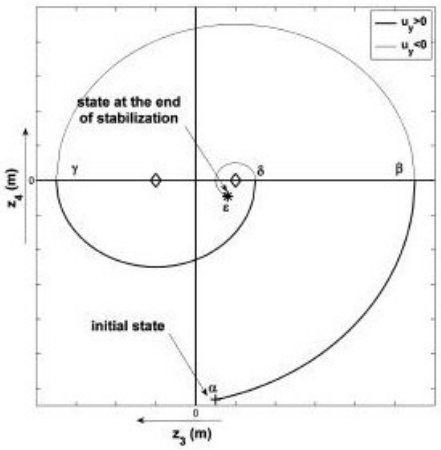
- Stabilization phase: states z_1 and z_2 are taken to the origin by a series of switches: once the switching curve is reached, the states follows the same curve to zero. At the end of this phase, the chaser is orbiting about the target with a stable closed relative orbit and the differential drag is set to zero (figures 2.7(a), 2.7(b)).

- Rendez-vous phase: this second phases counts a waiting phase, when the differential drag is null, and three switches to drive the semi-major axis of the relative orbit to zero (figures 2.7(c), 2.7(d)).

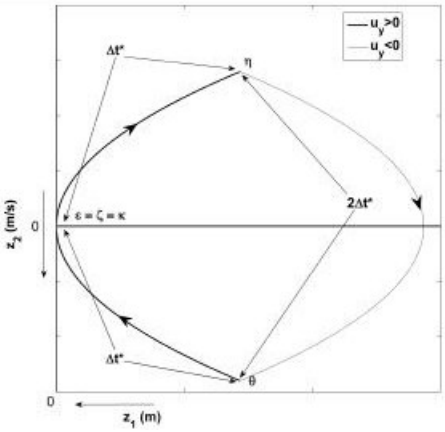
The most important improvement is the analytical character of the solution, which eliminates the numerical optimization. On the contrary, the control law is of bang-bang type, which is more difficult to deal with in real-life satellites. Especially in this case, being the control the differential drag,



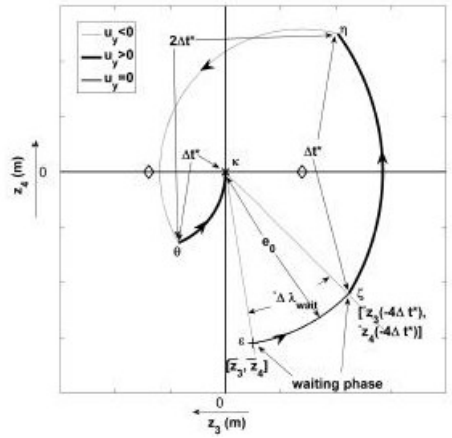
(a) Stabilization trajectory in z_1-z_2 plane (from 4)



(b) Stabilization trajectory in z_3-z_4 plane (from 4)



(c) Rendez-Vous trajectory in z_1-z_2 plane (from 4)



(d) Rendez-Vous trajectory in z_3-z_4 plane (from 4)

Figure 2.7: Two phase trajectory (from [4])

it is function of the position of the chaser with respect to the target, which makes it impossible an instantaneous variation of surface, preferring a continuous control law.

Further improvements come from Lambert, Kumar, Hamel and Hg [6], who implement this technique in a high precision propagator, using mean orbital elements for a better estimation of mean states.

Finally, at *Université de Liège*, Dell'Elce and Kerschen [7, 8] have implemented this technique on a nanosatellites constellation, the *QB50*, in order to validate it.

Chapter 3

"An Indirect Optimal Control Approach..."

3.1 A brief history of the theory

The optimal control theory has a long history, with a great improvement coming from the trajectory optimization in aerospace applications in the 1950s thanks to the advent of new technologies: computers. The first problems concerned the optimization of the rocket thrust profile for space launches.
In 1638, Galileo posed two shape problems: the catenary and the brachistochrone. The first consists in the research of the shape of a heavy chain constrained at its ends; the second concerns the shortest time path for a particle sliding without friction between two fixed points in a vertical plane (figure 3.1). Similarly, in 1662 Fermat postulated the principle that light always traverses a path through a sequence of optical media in minimum time.

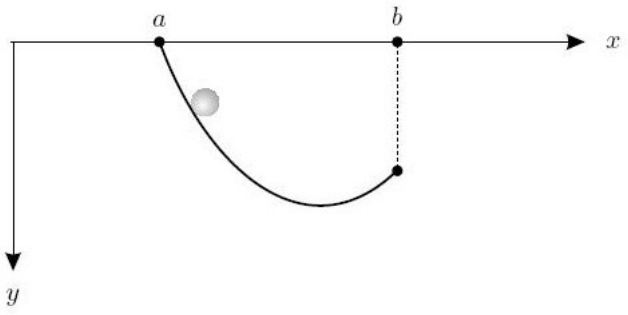


Figure 3.1: The brachistochrone problem

The name "brachistochrone" was given by Bernoulli, from ancient Greek: $\beta\rho\acute{\alpha}\chi\iota\sigma\tau o\zeta$, brachistos, "shortest" and $\chi\rho\acute{o}\nu o\zeta$, chronos, "time". In 1696, the scientist challenged his contemporaries to solve this problem and the next year he published the results of the five who responded, his brother Jakob, Liebnitz, l'Hopital, Tschirnhaus and Newton, with its own solution. Since the interest grew, Euler, a student of Bernoulli, published another book in 1744, collecting all the ideas of the epoch. He formulated the problem as: finding the curve $x(t)$ over the interval $a \leq t \leq b$ with given values $x(a), x(b)$, which minimizes

$$J = \int_b^a L(x(t), \dot{x}, t)dt \tag{3.1}$$

with $\dot{x} = \frac{dx}{dt}$ finding an optimality condition:

$$\frac{d}{dx}L_{\dot{x}}(t, x(t), \dot{x}(t)) = L_x(t, x(t), \dot{x}(t)) \tag{3.2}$$

Up to now, the solutions were only geometric, Lagrange introduced an analytical approach based on perturbations or "variations" of the optimal curve. Euler used this approach to reformulate its necessary conditions: the *Euler-Lagrange* equations. He renamed this approach as: the "calculus of variations".

3.2 Definitions

The objective of an optimal control problem is to determine a sequence of control variables to minimize a cost function, respecting some constraints. So the constitutive elements of the problem are:

- A mathematical model: a set of equations describing the system.
- A cost function, that expresses with a real number a global judgement on the behaviour of the system when stimulated by control variables.
- The constraints on states and control variables.

System dynamics The mathematical model is a set of differential equations, characterizing the dynamics of the system. It can be seen as a dynamical constraint and expressed as:

$$\frac{d\mathbf{x}}{dt} = \dot{\mathbf{x}}(t) = f(t, \mathbf{x}(t), \mathbf{u}(t)) \tag{3.3}$$

where $\mathbf{x}(t)\ \epsilon\ R^n$ are the states of the system and n is their number; $\mathbf{u}(t)\ \epsilon\ R^m$ are the control variables in number m. The function f is a vector of length

n and it can be expressed in matrix form, with the classical state-space representation:

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) \tag{3.4}$$

where $\mathbf{x}(t)$ and $\mathbf{u}(t)$ are the state and the control vectors; $\mathbf{A}(t)$ and $\mathbf{B}(t)$ are the state and the control matrix.

Cost Function The cost function in Bolza form is expressed as:

$$J = \Phi(x(t_f), t_f) + \int_{t_0}^{t_f} l(t, x(t), u(t))dt \tag{3.5}$$

where $\Phi(x(t_f), t_f) \ : \ R^n \times R \to R$ is the terminal cost and $l(t, x(t), u(t)) \ : \ R^n \times R^m \times R \to R$ is the integrated cost. In the special case of $\Phi = 0$, the cost function is in Lagrange form, if $l = 0$ the form is of Meyer's. By introducing an additional variable, $z(t)$, the Bolza problem can be reduced to the Meyer one:

$$y(t) = \begin{bmatrix} x(t) \\ z(t) \end{bmatrix}, \dot{y}(t) = \begin{bmatrix} f(t, x(t), u(t)) \\ l(t, x(t), u(t)) \end{bmatrix} = f(t, x(t), u(t)) \tag{3.6}$$

$$J = \Phi(t_f, x(t_f)) + (z(t_f) - z(t_0)) \tag{3.7}$$

The cost function determines the kind of problem to deal with, the most used expressions are:

1. Minimum time control problem: drive a state from x_0 to x_f in minimum time;

$$J = \int_{t_0}^{t_f} dt = t_f - t_0 = t^\star \tag{3.8}$$

2. Minimum fuel control problem: drive a state from an initial position to a final one minimizing the consumption of fuel;

$$J = \int_{t_0}^{t_f} |u(t)|dt \tag{3.9}$$

3. Minimum energy control problem: drive a state from x_0 to x_f minimizing the dissipated energy;

$$J = \int_{t_0}^{t_f} u(t)^2 dt \tag{3.10}$$

4. Minimum error on the final point:

$$J = x(t_f)^T S_f x(t_f) \tag{3.11}$$

Constraints A general form of constraints is:

$$\phi(t_0, x(t_0), t_f, x(t_f)) = 0 \tag{3.12}$$

where $\phi(t_0, x(t_0), t_f, x(t_f)) : R^n \times R \times R^n \times R \rightarrow R^q$ are the boundary conditions and q is their number. The most used expression counts two groups:

- Initial condition:
$$\mathbf{x}(t_0) = \mathbf{x}_0 \tag{3.13}$$
- Boundary Conditions:
$$\mathbf{\Psi}(t_f, \mathbf{x}(t_f)) = 0 \tag{3.14}$$

The most simple expression for Ψ is when the final states must reach a given value:

$$\mathbf{\Psi} = \mathbf{x}(t_f) - \mathbf{x}_f = 0 \tag{3.15}$$

Up to now, the only constraints are imposed at the ends of the time interval, but the real complication is when the solution must satisfy constraints along the path, on the state or on the control variables. Often, the states have only bounds to respect

$$\mathbf{x}_l \leq \mathbf{x}(t) \leq \mathbf{x}_u \tag{3.16}$$

while control variables can also have equality and inequality path constraints:

- Bounds:
$$\mathbf{u}_l \leq \mathbf{u}(t) \leq \mathbf{u}_u \tag{3.17}$$
- Equality constraints:
$$\mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t) = 0 \tag{3.18}$$
- Inequality constraints (as a generalization of Equality constraints):
$$\mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t) \leq 0 \tag{3.19}$$

The presence of these elements can really complicate the search for the solution and each solving method has implemented different approaches to deal with it.

General Expression The optimal control problem is then posed as: minimize

$$J = \Phi(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} l(t, \mathbf{x}(t), \mathbf{u}(t))dt \tag{3.20}$$

subjected to

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)) \tag{3.21}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0 \tag{3.22}$$

$$\mathbf{\Psi}(t_f, \mathbf{x}(t_f)) = 0 \tag{3.23}$$

$$\mathbf{x}_l \leq \mathbf{x}(t) \leq \mathbf{x}_u \tag{3.24}$$

$$\mathbf{u}_l \leq \mathbf{u}(t) \leq \mathbf{u}_u \tag{3.25}$$

$$\mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t) \leq 0 \tag{3.26}$$

Solution approaches The solution of an optimal control problem can be very tough and several methods have been studied during the years. There are three basic solving approaches:

I. Hamilton-Jacobi-Bellman (HJB) partial differential equations (PDE) and Dynamic Programming;

II. Direct methods based on a finite dimensional parametrization of the control;

III. Calculus of Variations, Euler-Lagrange differential equations and the Maximum Principle (indirect methods).

3.3 Hamilton-Jacobi-Bellman PDE and Dynamic Programming

The basic idea of Dynamic Programming is to compute the optimal cost function $J^\star(x, t)$ recursively backwards, starting from a known value at its end and applying the Principle of Optimality[1].

A classical example is the Shortest Path Problem [11] in figure 3.3. Here the J is the total travel time and each point of the grid constitutes a possible state.

[1] Any subarc of an optimal trajectory is also optimal: the subarc on $[t + \Delta t, t_f]$ is the optimal solution for the initial condition $\mathbf{x}(t + \Delta t)$ (figure 3.2)

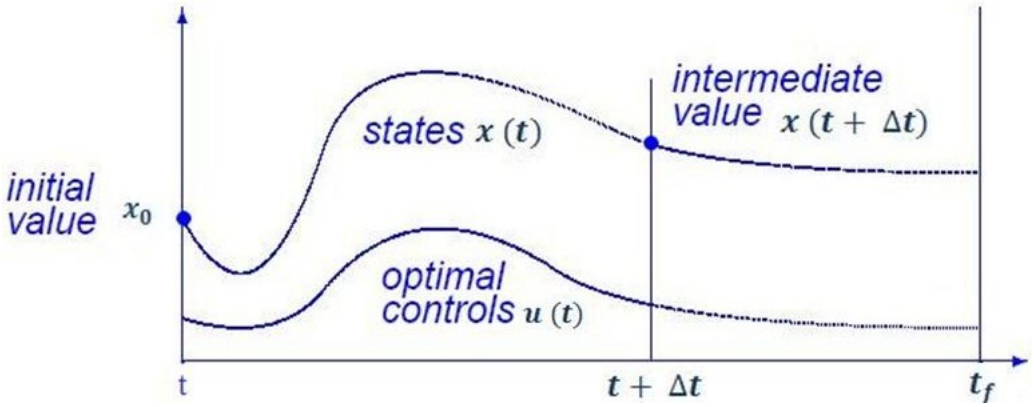


Figure 3.2: The Principle of Optimality (from [12])

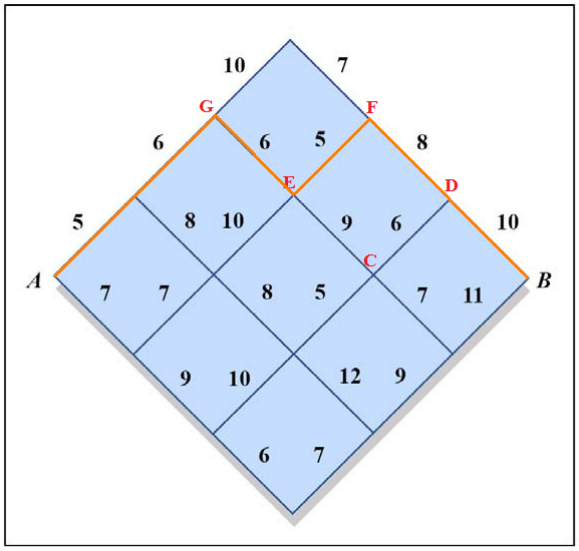


Figure 3.3: The Shortest Path Problem

In order to reach B from A in the minimum time (travel times are shown on each segment), first the grid has to be applied, then the backward approach is followed on that grid, exploiting the principle of Optimality.
From B, there are two ways: one with 10 and the other with 11. To apply the optimality principle, the same intermediate state has to be considered: point C. For the upper possibility, the travel time is 16, for the lower one is 18: the segment BD is chosen. At this point, the new "final condition" is point D: applying the same procedure (6+9 vs 8+5), the segment DF is chosen. This segment allows a consideration: the importance of using the same intermediate point, because in this case, it could be immediate the choice of 6 with respect to 8, but this would drive to a non optimum solution, as it can be easily calculated. The path to follow is then the orange one on the figure 3.3.
The previous example represents a discrete system, a continuous time solution takes to the famous $Hamilton - Jacobi - Bellman$ equation.
The first step is the time interval grid $[t, \Delta t]$ and $[t + \Delta t, t_f]$, with $\Delta t \rightarrow 0$, in order to have a continuous time solution.
The optimal cost function can be expressed as:

$$\begin{aligned} J^\star(\mathbf{x}(t), t) &= \min_{u(\tau)\in U, t\leq\tau\leq t_f} \left\{ \Phi(\mathbf{x}(t_f), t_f) + \int_t^{t_f} l(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau \right\} \\ &= \min_{u(\tau)\in U, t\leq\tau\leq t+\Delta t} \left\{ \Phi(\mathbf{x}(t_f), t_f) + \int_t^{t+\Delta t} l(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau \right\} \\ &\qquad \left\{ + \int_{t+\Delta t}^{t_f} l(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau \right\} \end{aligned} \tag{3.27}$$

The principle of Optimality allows to write the optimal cost-to-go function of the state at the time instant $t + \Delta t$:

$$J^\star(\mathbf{x}(t+\Delta t), t+\Delta t) = \min_{u(\tau)\in U, t\leq\tau\leq t+\Delta t} \left\{ \Phi(\mathbf{x}(t_f), t_f) + \int_{t+\Delta t}^{t_f} l(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau \right\} \tag{3.28}$$

The cost function can then be re-written as:

$$J^\star(\mathbf{x}(t), t) = \min_{u(\tau)\in U, t\leq\tau\leq t_f} \left\{ \int_t^{t+\Delta t} l(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau + J^\star(\mathbf{x}(t + \Delta t), t + \Delta t) \right\} \tag{3.29}$$

The cost-to-go cost function is expressed in Taylor series:

$$\begin{aligned} J^\star(\mathbf{x}(t + \Delta t), t + \Delta t) \approx J^\star(\mathbf{x}(t), t) &+ \left[\frac{\partial J^\star}{\partial t}(\mathbf{x}(t), t) \right] \Delta t \\ &\left[\frac{\partial J^\star}{\partial \mathbf{x}}(\mathbf{x}(t), t) \right] (\mathbf{x}(t + \Delta t) - \mathbf{x}(t)) \end{aligned} \tag{3.30}$$

For small Δt :

$$J^\star(\mathbf{x}(t+\Delta t), t+\Delta t) \approx J^\star(\mathbf{x}(t), t) + J_t^\star(\mathbf{x}(t), t)\Delta t + J_{\mathbf{x}}^\star(\mathbf{x}(t), t) f(\mathbf{x}(t), \mathbf{u}(t), t)\Delta t \tag{3.31}$$

Substituting in the cost function:

$$\begin{aligned} J^\star(\mathbf{x}(t), t) = \min_{u(\tau)\in U} \{ & l(\mathbf{x}(t), \mathbf{u}(t), t)\Delta t + J^\star(\mathbf{x}(t), t) \\ & + J_t^\star(\mathbf{x}(t), t)\Delta t + J_{\mathbf{x}}^\star(\mathbf{x}(t), t) f(\mathbf{x}(t), \mathbf{u}(t), t)\Delta t \} \end{aligned} \tag{3.32}$$

Extracting the terms independent of $\mathbf{u}(t)$:

$$0 = J_t^\star(\mathbf{x}(t), t) + \min_{u(\tau)\in U} \{ l(\mathbf{x}(t), \mathbf{u}(t), t) + J_{\mathbf{x}}^\star(\mathbf{x}(t), t) f(\mathbf{x}(t), \mathbf{u}(t), t) \} \tag{3.33}$$

This expression represents a partial differential equation in $J^\star(\mathbf{x}(t), t)$, solvable with the Dynamic Programming: backwards in time, with as initial value $J^\star(\mathbf{x}(t_f), t_f) = \Phi(\mathbf{x}(t_f), t_f)$ The **Hamilton −Jacobi −Bellman equation** is typically expressed as :

$$-J_t^\star(\mathbf{x}(t), t) = \min_{u(\tau)\in U} \{ l(\mathbf{x}(t), \mathbf{u}(t), t) + J_{\mathbf{x}}^\star(\mathbf{x}(t), t) f(\mathbf{x}(t), \mathbf{u}(t), t) \} \tag{3.34}$$

The method is greatly advantageous because this equation is both a necessary and sufficient condition, it is able to reach the global optimum of the function, even with a non convex problem. For the linear quadratic regulator, the HJB equation can be solved analytically or numerically by solving an algebraic or dynamic Ricatti equation. For quite general nonlinear problems, the solution is found by numerically approximating the value function, solving a first-order PDE: Lions (1992) found the famous *viscosity solution*. Besides this powerful advantage, its use is quite limited because of the huge amount of storage space that is needed even for a quite small number of variables: "If we want only one optimal path from a known initial point, it is wasteful and tedious to find a whole field of extremals" (from [11]). As Bellman himself said, the method suffers from the "*curse of dimensionality*": so normally the solution is restricted to low dimension ($n \leq 3$) problems. Several remedies have been implemented over the years, as a neural network approximation (*neuro − dynamic programming* of Bertsekas and Tsitsiklis (1996)), but it is still restricted to small dimension problems.

3.4 Direct Methods

The basic idea of direct methods is to first discretize the continuous control problem into a discrete one using a transcription technique, and then solve

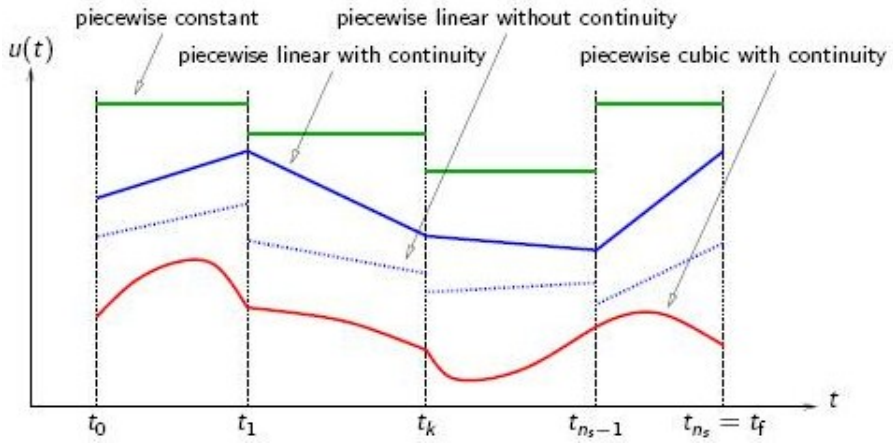


Figure 3.4: The control parametrization(from [13])

the resulting NLP (Non Linear Programming) problem through a parameter optimization method, based on Newton approaches. The presence of path constraints is treated in two possible ways: an active set strategy or an interior-point technique. In the first, the solution is sought moving in the feasible domain, searching for an active constraint. When found, it is followed until a new constraint becomes active, hoping in deactivating the first. The path to obtain the solution is at the limits of the feasible region (figure 3.5).

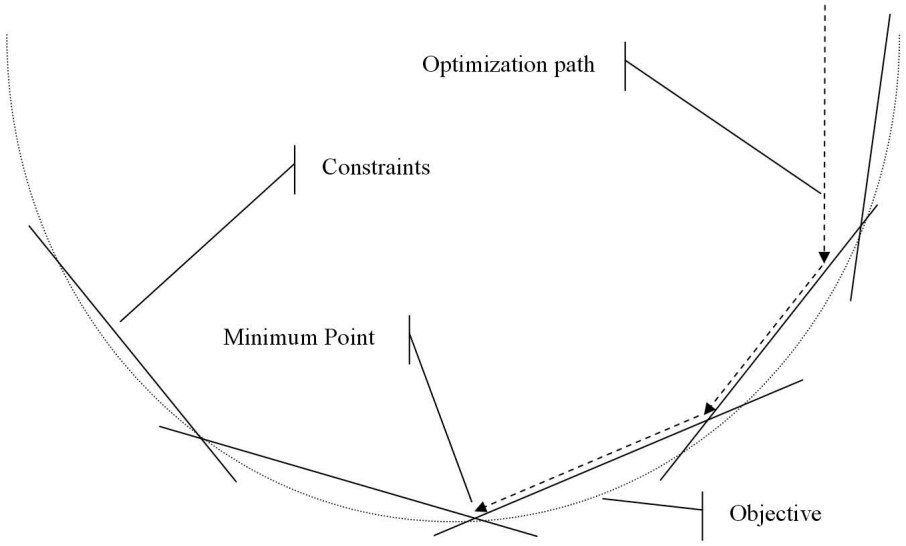


Figure 3.5: The active set strategy(from [14])

On the contrary, the interior-point method looks for the solution inside the feasible domain, by introducing a barrier in the objective function, that keeps the solution searching path away from the constraints. The solution is then obtained by iterating on the barrier value, optimizing an equality constrained problem and using its solution as initial guess for the successive optimization problem. The search is concluded when the barrier value is sufficiently small (figure 3.6).

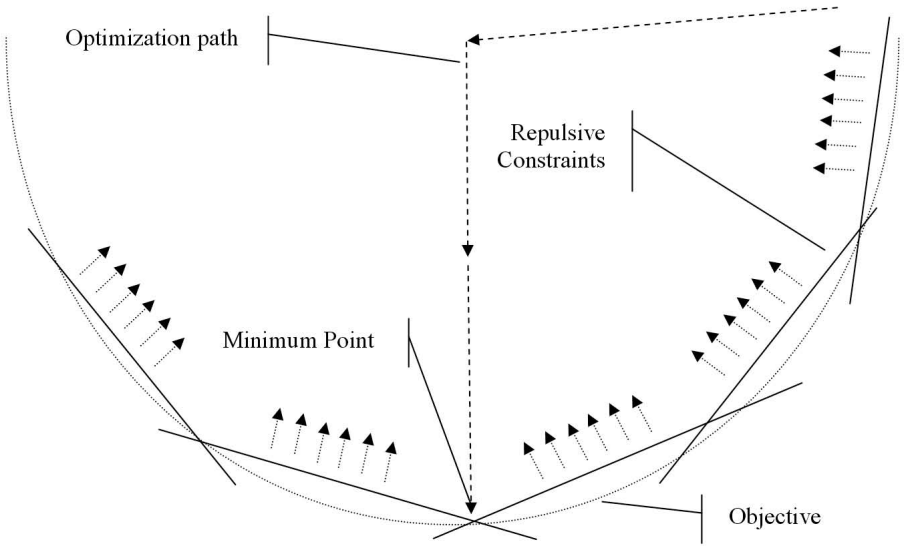


Figure 3.6: The interior point method(from [14])

The direct methods are the most studied, especially in the last 30 years and many complex applications have been solved with them. Two great advantages make these methods quite attractive: the possibility of using NLP solvers and a much easier way to deal with path and inequality constraints with respect to Dynamic Programming and Indirect Methods. On the other hand, the solution is only suboptimal and, since Newton methods are used, Jacobian and Hessian calculation can be difficult.
Three main strategies exist:

- Direct sequential approach;
- Direct simultaneous approach;
- Multiple Shooting Method.

3.4.1 Direct sequential approach or Direct Single Shooting Method

This method consists in the transcription of the optimal control problem through the discretization of the only control variables[2], considered as optimization parameters. The dynamic equations are solved "exactly", so the states are kept as dependent variables. The procedure counts four phases:

I subdivide the time interval into n_s control stages, called *collocation points* : $t_0 < t_1 < ... < t_{n_s} = t_f$;

II approximate the control in each time interval $[t_k, t_{k+1}]$ with different type of parametrization (linear, quadratic, BSpline, Lagrange polynomials: figure 3.4): $\tilde{u}(t) = U^k(t, q^k)$;

III integrate numerically the ODE with a classic IVP (Initial Value Problem) solver, as *Euler* (the implicit backward or the explicit forward), $Runge - Kutta$ (from the explicit RK4 to the most used non-stiff-problem solver $ode45$ to the stiff-problem-solver $ode15s$, both in $Matlab$®), *Multistep* (the Predictor-Corrector formula, as $ode113$, also in $Matlab$®), to compute the states as function of finitely many control parameters $\mathbf{q}$: $\tilde{x}(t, \mathbf{q})$;

IV the NLP is then obtained :

$$\tilde{J}(\mathbf{q}) = \min_{\mathbf{q}} \Phi(\tilde{\mathbf{x}}(t_f, \mathbf{q}), t_f) + \int_{t_0}^{t_f} l(t, \tilde{\mathbf{x}}(t, \mathbf{q}), \tilde{\mathbf{u}}(t, \mathbf{q}))dt \tag{3.35}$$

subjected to

- discretized path constraints (i = 0 : n_s):

$$g(\tilde{\mathbf{x}}(t_f, \mathbf{q})) \leq 0 \tag{3.36}$$

- terminal constraints:

$$\Psi(\tilde{x}(t_i, \mathbf{q}), \tilde{u}(t_i, \mathbf{q}), t) = 0 \tag{3.37}$$

and solved by a classical NLP solver.

A standard solver is the SQP, Sequential Quadratic Programming:

$$\min_q F(q) \ s.t. \ H(q) \leq 0 \tag{3.38}$$

[2]The method is often called *Control Vector Parametrization* (CVP)

1. guess for q^0, $k = 0$;
2. evaluate $F(q^0)$ and $H(q^0)$ by the ODE solution;
3. compute the solution Δq^k with the Quadratic Programming (by a Newton method):

$$\min_{\Delta q} \nabla F(q_k)^T \Delta q + \frac{1}{2} \Delta q^T A^k \Delta q \ s.t. \ \nabla H(q^k)^T \Delta q \leq 0 \tag{3.39}$$

4. compute $q^{k+1} = q^k + \alpha_k \Delta q^k$, with α_k the step length, obtained by line search

The direct single shooting method has the advantage of using very few degrees of freedom even for large ODE system, the active set changes are easily treated and the only guesses are on the control parameters q^k. However, it has serious difficulty in treating unstable problems and the states can depend very non-linearly on q^k, without the possibility of using information on x in initialization. Another important source of inaccuracy is the Jacobian and Hessian calculation in the Newton method , commonly obtained via finite difference approximations, that can be very expensive. But the very numerical effort is determined by the complexity of the parametrization of control variables: a compromise is sough between accuracy (more complex control parametrization with non uniform mesh) and computational cost (simpler parametrization). However, it is very difficult to know a priori the right control parametrization.
Several packages have been developed over the years that implement this method, as gOPT, DYOS, POST [3], GTS [4]. In space applications, these methods have been extensively used for launch and orbit transfer problems, because of the small number of NLP variables. For example, a two-burn orbit transfer problem can be expressed by eight variables (time of ignition and velocity increment for each burn) and four or five constraints.

3.4.2 Direct simultaneous approach or Direct Collocation Method

In this method, the transcription is obtained through the discretization of both state and control variables (*full discretization*). The procedure can be summarized as :

[3]The program to simulate trajectories of launch vehicles, by Martin Marietta
[4]Generalized Trajectory Simulation, developed by The Aerospace Corporation

I discretize the time interval into a fine grid, made by n_s collocation points, as for the Direct Shooting;

II approximate control (with parameters q^k) and state (with parameters s^k) variables on collocation intervals $[t_k, t_{k+1}]$, with a specific representation, as the Lagrange polynomials, the Monomial Basis or more complex, as for pseudospectal methods. For example, the Lagrange polynomials of degree N representation for the states is expressed as:

$$\tilde{\mathbf{x}}(t) = \mathbb{X}(t, s^k) \tag{3.40}$$

$$\mathbb{X}_j(t, s_j^k) = \sum_{i=0}^{N} s_{ij}^k L_i(t) = \sum_{i=0}^{N} s_{ij}^k \phi_i^{(N)} \left(\frac{t - t_{k-1}}{t_k - t_{k-1}} \right) \tag{3.41}$$

with $\phi_i^{(N)}(\tau_q) = \delta_{i,q}$, $q = 0 : N$

III the system dynamics

$$\dot{x}(t) - f(x(t), u(t), t) = 0 \tag{3.42}$$

are replaced by finite equality constraints with $k = 1 : n_s$

$$c^k(q^k, s^k, s^{k+1}) := \frac{s^{k+1} - s^k}{t_{k+1} - t_k} - f\left(\frac{s^k + s^{k+1}}{2}, q^k \right) = 0 \tag{3.43}$$

IV the integral is also approximated:

$$\int_{t_k}^{t_{k+1}} l(\mathbf{x}(t), \mathbf{u}(t)) dt \approx L^k(q^k, s^k, s^{k+1}) := L\left(\frac{s^k + s^{k+1}}{2}, q^k \right) (t_{k+1} - t_k) \tag{3.44}$$

V a large scale, but sparse NLP is obtained :

$$\min_{s,q} \Phi(\mathbf{s}_{n_s}) + \sum_{k=0}^{n_s - 1} L^k(q^k, s^k, s^{k+1}) \tag{3.45}$$

subjected to

- fixed initial conditions :

$$s_0 - x_0 = 0 \tag{3.46}$$

- discretized ODE model, with $k = 0 : n_s - 1$

$$c^k(q^k, s^k, s^{k+1}) = 0 \tag{3.47}$$

– discretized path constraints, with $k = 0 : n_s$

$$g(s^k, q^k) \leq 0 \tag{3.48}$$

– boundary conditions

$$\Psi(s_{n_s}) = 0 \tag{3.49}$$

One of the most important advantage of this method is the possibility of using NLP solver, as SQP, for sparse problems, that is much cheaper to solve. In this way, even very large problems are easily treated in sparse NLP. With respect to the Direct Shooting Method, it can use the information on x for initialization, since discretization is operated on both, states and control, variables, which allows also to naturally handle path and inequality constraints. With this method, also unstable systems find a solution. However, an important issue exists: a change in the problem creates a great change in the resolution, because a new grid has to be obtained, which can greatly modify the number of NLP variables. A second disadvantage is the requirement of a good initial guess for solution profile in the NLP.
The number of packages using this type of method is quite important: OTIS, SOCS, MISER, GESOP, NTG, IPOPT (Biegler, Wätcher), OCPRSQP (Betts, Bock, Schulz), DIRCOL (von Stryk), SOCS (Betts, Huffmann).

Pseudospectral Methods

A special look has to be granted to these methods, because of the extreme development in the last years. The pseudospectral methods are a particular type of direct collocation methods, where collocation nodes are obtained via different types of quadrature. The most common are three: Chebyshev pseudospectral method (CPM), Legendre pseudospectral method (LPM) and Gauss pseudospectral method (GPM). The first uses Chebyshev polynomials on Chebyshev-Gauss-Lobatto collocation points, the second implements Lagrange polynomials for state and control approximation and Legendre-Gauss-Lobatto collocation points and the last uses a Gaussian quadrature on Gauss collocation points. This method, in particular, has been extensively studied and several versions exist, depending on the set of collocation nodes [15]: $Legendre - Gauss$ (LG), $Legendre - Gauss - Radau$ (LGR) and $Legendre - Gauss - Lobatto$ (LGL). A complete description of these methods is beyond the scope of this dissertation, anyway some properties are presented.

All the three methods consider the domain [-1, 1], but each one includes different points: the LG doesn't include any of the endpoints, the LGR includes one of the endpoints (the initial or the final one, creating a non-unique asymmetric points distribution) and LGL includes both endpoints

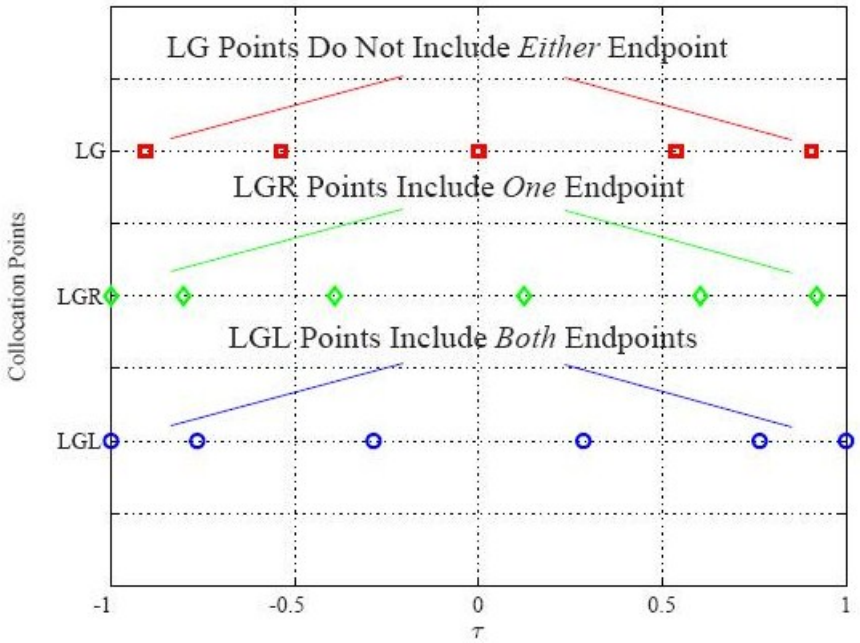


Figure 3.7: Differences between LG, LGR and LGL collocation points(from [15])

(figure 3.7). As example, the collocation at LGR points is presented.
As shown on figure 3.7, the LGR method includes only one endpoint: for this derivation, the final endpoint is considered. Each component of the state is approximated through a Lagrange polynomial expansion (equation 3.40, 3.41). The state differentiation is then expressed as:

$$\dot{x}_j^N(t_k) = \sum_{i=0}^{N} s_{ij}^k \dot{L}_i(t_k) = \sum_{i=0}^{N} D_{ki} s_{ij}^k \tag{3.50}$$

where $\mathbf{D}$ is the Radau Pseudospectral Differentiation Matrix of size $N \times (N+1)$, non-square, because $N+1$ points for the state approximation and N LGR collocation points. Denoting by $\mathbf{F}(\mathbf{X}^{\mathbf{LGR}}, \mathbf{U}^{\mathbf{LGR}})$ the $N \times n_s$ matrix containing the discretized dynamics, the Radau pseudospectral approxima-

tion to the continuous time optimal control problem is expressed as [16]:

$$minimize\ J = \Phi(\mathbf{X}_{N+1}, t_{N+1}) + \frac{t_f - t_0}{2} \sum_{i=1}^{N} w_i l(\mathbf{X}_i, \mathbf{U}_i, t_i, t_0, t_f) \tag{3.51}$$

$$subjected\ to\ \mathbf{D}\mathbf{X}^{LGR} - \frac{t_f - t_0}{2}\mathbf{f}(\mathbf{X}, \mathbf{U}, t, t_0, t_f) = \mathbf{0} \tag{3.52}$$

$$\Psi(\mathbf{X}_{N+1}, t_{N+1}) = \mathbf{0} \tag{3.53}$$

$$\frac{t_f - t_0}{2}\mathbf{C}(\mathbf{X}, \mathbf{U}, t, t_0, t_f) \leq \mathbf{0} \tag{3.54}$$

where the fraction $\frac{t_f - t_0}{2}$ is used to deal with time domain of the form $[-1, 1]$ and the NLP variables are $(\mathbf{X}_1, \ldots, \mathbf{X}_{N+1})$, $(\mathbf{U}_1, \ldots, \mathbf{U}_N)$, t_0 and t_f.
In LG and LGR methods, the degree of the polynomials of the state approximation is the same as the number of collocation points, while the LGL has a one degree lower approximation with respect to the nodes number. In reference 15, the three methods are compared and two examples are presented: the results show a better behaviour for Gauss and Radau with respect to Lobatto, especially in the costate estimation.
The great interest in these methods has developed several packages, one of the most famous is *GPOPS*, in *Matlab*® environment, that has been used to compare with indirect methods (section 6.5).

3.4.3 Direct Multiple Shooting Method

The Direct Multiple Shooting Method can be considered an *hybrid* method, since it includes characteristics of both sequential and simultaneous methods. In particular, the transcription into a NLP problem starts similarly to the single shooting method: a grid is created on time interval and control is parametrized. Then, the state trajectories are discretized by the same grid points: the initial values of state trajectories on each subinterval are also parametrized (figure 3.8).

The continuous dynamic equations are transformed into discretized equality constraints where the terminal value of the state at each subinterval is equal to the initial value of the state in the next subinterval.
The procedure can be summarized as:

I divide the time horizon into equal subintervals : $t_0 < t_1 < \cdots < t_N$;

II approximate control with a specific parametrization (linear, quadratic, BSpline) $\tilde{u}(t) = U^i(t, \mathbf{q}^i)$

III parametrize the initial conditions on the states on each subinterval $\tilde{x}(t^i) = X^i(t^i, \mathbf{s}^i)$;

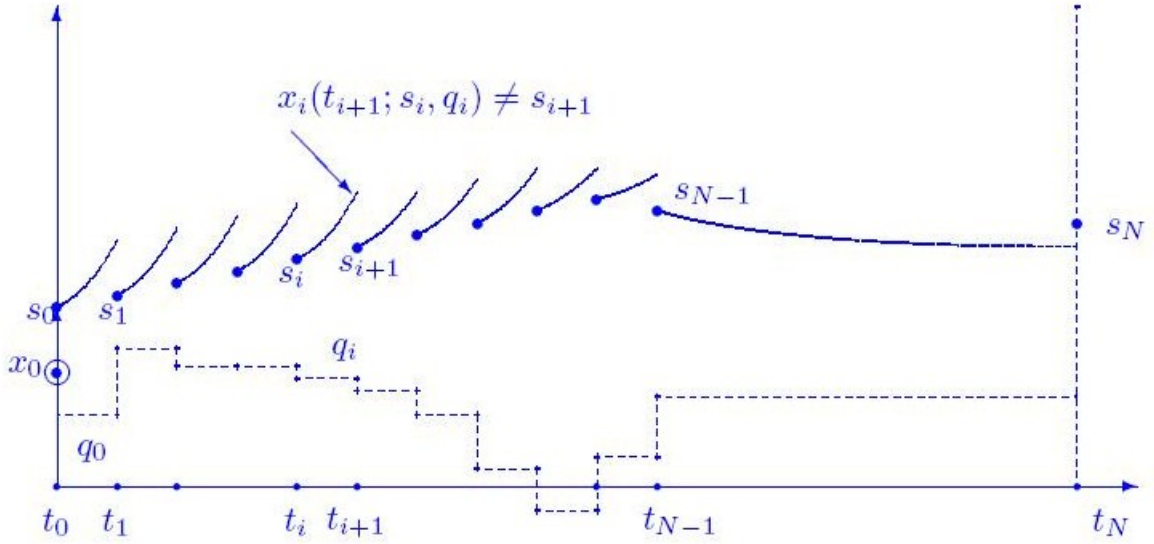


Figure 3.8: The Multiple Shooting Method(from [10])

IV solve IVPs at each subinterval to obtain states values at the end of each subinterval;

V define matching conditions to maintain continuity of states between subintervals;

VI define the NLP and solve it, through a Newton-based method.

The NLP in Direct Multiple Shooting Method is then expressed as :

$$\min_{s,q} \Phi(\mathbf{s}_N) + \sum_{k=0}^{N-1} L^i(q^i, s^i, s^{i+1}) \tag{3.55}$$

subjected to

- initial value:

$$\mathbf{s}_0 - \mathbf{x}_0 = 0 \tag{3.56}$$

- continuity constraints $i = 0 : N - 1$:

$$s_{i+1} - \tilde{x}^i(t_{i+1}, s^i, q^i) = 0 \tag{3.57}$$

- discretized path constraints $i = 0 : N$:

$$\tilde{g}(s^i, q^i) \leq 0 \tag{3.58}$$

- terminal constraints

$$\Psi(s_N) = 0 \tag{3.59}$$

Since an hybrid method, it shares advantages of both sequential and simultaneous methods. Since the states are parametrized in the initial conditions of each subproblem, an a priori knowledge on x can be used for the initialization (as collocation methods) and they can handle quite well path and inequality constraints. As collocation's, the optimization of unstable or even chaotic systems can be treated. Because each DAE partition can be treated independently, so IVP solution and derivatives computations are decoupled on each subinterval, this method is well suited for parallel computation. Its size can be placed between the previous two methods: it considers much more optimal parameters than sequential, but less than simultaneous. This takes to a less sparse structure, that can increase the computational cost, since the sparse NLP solvers are no longer a possibility. Despite the increased size, this method represents an improvement with respect to the Single Shooting, because the sensitivity to errors in the unknown initial conditions is reduced, since the integration is performed over significantly smaller time intervals.
Several packages have been developed over the years, the most known are *MUSCOD* and *HQP*.

3.5 Indirect Methods: Calculus of Variations, Euler-Lagrange differential equations and the Maximum Principle

In an optimal control problem, the Calculus of Variations allows to derive the necessary conditions for a local minimum [18].
As already mentioned, the goal is to minimize a functional, the cost function J. The same procedure of the minimization of a function can be implemented: find the gradient, set it to zero to obtain the stationary points and then analyse the higher order derivatives to determine if minimum or maximum points. A functional $J(\mathbf{x}(t))$ has a local minimum at $\mathbf{x}^\star(t)$ if

$$J(\mathbf{x}(t)) \geq J(\mathbf{x}^\star(t)) \tag{3.60}$$

The minimization is obtained by vanishing the variation of the cost function on the path $\mathbf{x}^\star(t)$ for all admissible variations $\delta\mathbf{x}$:

$$\delta J(\mathbf{x}^\star, \delta\mathbf{x}) = 0 \tag{3.61}$$

The Calculus of Variations is applied on the augmented cost function or Lagrangian, defined by adding the Lagrange Multipliers, static and dynamic, to terminal and dynamic constraints:

$$J_a = \Phi(\mathbf{x}(t_f), t_f) - \nu^T \Psi(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} \left[l(\mathbf{x}(t), \mathbf{u}(t), t) - \lambda^T(t) \left(\dot{\mathbf{x}} - \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \right) \right] dt \quad (3.62)$$

where $\nu \in \mathbb{R}^q$ are the Lagrangian Multipliers associated to the terminal constraints and $\lambda(t) \in \mathbb{R}^n$ are the Lagrangian Multipliers associated to dynamic constraints, called *costate* or *adjoint variables*. The Lagrangian is derived with respect to all the free variables:

$$\begin{aligned} \delta J_a = & \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} \delta \mathbf{x}_f + \frac{\partial \Phi}{\partial t_f} \delta t_f - \delta \nu^T \Psi - \nu^T \frac{\partial \Psi}{\partial \mathbf{x}(t_0)} \delta \mathbf{x}_0 - \nu^T \frac{\partial \Psi}{\partial t_0} \delta t_0 \\ & - \nu^T \frac{\partial \Psi}{\partial \mathbf{x}(t_f)} \delta \mathbf{x}_f - \nu^T \frac{\partial \Psi}{\partial t_f} \delta t_f + (l - \lambda^T(\dot{\mathbf{x}} - \mathbf{f}))|_{t=t_f} \delta t_f - (l - \lambda^T(\dot{\mathbf{x}} - \mathbf{f}))|_{t=t_0} \delta t_0 \\ & + \int_{t_0}^{t_f} \left[\frac{\partial l}{\partial \mathbf{x}} \delta \mathbf{x} + \frac{\partial l}{\partial \mathbf{u}} \delta \mathbf{u} - \delta \lambda^T (\dot{\mathbf{x}} - \mathbf{f}) + \lambda^T \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \delta \mathbf{x} + \lambda^T \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \delta \mathbf{u} - \lambda^T \delta \dot{\mathbf{x}} \right] dt \end{aligned} \quad (3.63)$$

Since final and initial time are free, $\delta \mathbf{x}(t_f(t_0))$ is different from $\delta \mathbf{x}_{f(0)}$ (figure 3.9). This comes from the definition of the variable itself: $\delta \mathbf{x}(t_f)$ is the difference between two admissible trajectories, the optimal one $\mathbf{x}^\star$, and the candidate one $\mathbf{x}$, for t_f fixed; while $\delta \mathbf{x}_f$ is the difference of the same variables, but when the final time is not fixed, so the candidate trajectory adds a part due to δt_f.

Integrating by parts, the variation of the Lagrangian is re-written as :

$$\begin{aligned} \delta J_a = & \left(\frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \nu^T \frac{\partial \Psi}{\partial \mathbf{x}(t_f)} - \lambda^T(t_f) \right) \delta \mathbf{x}_f + \left(-\nu^T \frac{\partial \Psi}{\partial \mathbf{x}(t_0)} + \lambda^T(t_0) \right) \delta \mathbf{x}_0 \\ & - \delta \nu^T \Psi + \left(-\nu^T \frac{\partial \Psi}{\partial t_0} - l(t_0) - \lambda^T(t_0) \mathbf{f}(t_0) \right) \delta t_0 \\ & \left(\frac{\partial \Phi}{\partial t_f} - \nu^T \frac{\partial \Psi}{\partial t_f} + l(t_f) + \lambda^T(t_f) \mathbf{f}(t_f) \right) \delta t_f \\ & \int_{t_0}^{t_f} \left[-\delta \lambda^T (\dot{\mathbf{x}} - \mathbf{f}) + \left(\frac{\partial l}{\partial \mathbf{x}} + \lambda^T \frac{\partial \mathbf{f}}{\partial \mathbf{x}} + \dot{\lambda} \right) \delta \mathbf{x} + \left(\frac{\partial l}{\partial \mathbf{u}} + \lambda^T \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right) \delta \mathbf{u} \right] dt \end{aligned} \quad (3.64)$$

The Hamiltonian is defined to simplify the expression of the optimality conditions:

$$H(\mathbf{x}(t), \mathbf{u}(t), \lambda(t), t) = l(\mathbf{x}(t), \mathbf{u}(t), t) + \lambda^T \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \quad (3.65)$$

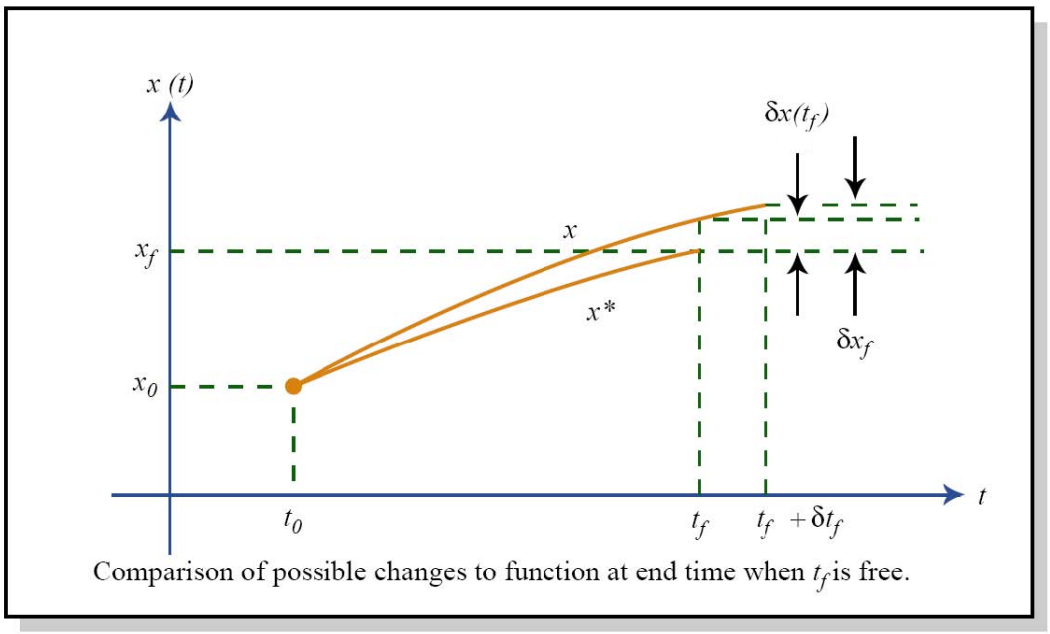


Figure 3.9: Differences between $\delta\mathbf{x}(t_f)$ and $\delta\mathbf{x}_f$ (from [12])

Because of the arbitrariness of the variations, the optimality first-order necessary conditions are obtained:

I *Euler-Lagrange* equations :

- $\forall\delta\lambda$:

$$\dot{\mathbf{x}} = \frac{\partial H}{\partial \lambda} \tag{3.66}$$

- $\forall\delta\mathbf{x}$:

$$\dot{\lambda} = -\frac{\partial H}{\partial \mathbf{x}} \tag{3.67}$$

- $\forall\delta\mathbf{u}$

$$\frac{\partial H}{\partial \mathbf{u}} = 0 \tag{3.68}$$

II the *Transversality Conditions* :

- $\forall\delta t_0$:

$$-H(t_0) - \nu^T \frac{\partial \Psi}{\partial t_0} = 0 \tag{3.69}$$

- $\forall \delta t_f$:

$$H(t_f) - \nu^T \frac{\partial \Psi}{\partial t_f} + \frac{\partial \Phi}{\partial t_f} = 0 \tag{3.70}$$

- $\forall \delta \mathbf{x}_0$:

$$\lambda^T(t_0) = \nu^T \frac{\partial \Psi}{\partial \mathbf{x}(t_0)} \tag{3.71}$$

- $\forall \delta \mathbf{x}_f$:

$$\lambda^T(t_f) = \frac{\partial \Phi}{\partial \mathbf{x}(t_f)} - \nu^T \frac{\partial \Psi}{\partial \mathbf{x}(t_f)} \tag{3.72}$$

III The *Boundary Conditions* : $\forall \delta \nu$:

$$\Psi(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) = 0 \tag{3.73}$$

This set of equations is composed of $2n$ differential equations (for state $\mathbf{x}(t)$ and costate $\lambda(t)$ variables, eq. 3.66, 3.67), m algebraic equations (for control $\mathbf{u}(t)$, eq. 3.68), $2n$ initial conditions (eq. 3.71, 3.73) and $2n$ terminal conditions (eq. 3.72, 3.73) and 2 boundary conditions for the time variables (eq. 3.69, 3.70). It is then clear that this set of equations forms a **Two-Point Boundary Value Problem** (TPBVP).
Up to now, no path restrictions have been introduced on control and state variables. The activation or deactivation of a state or a control constraint generally leads to a jump in the adjoint variables. These constraints prevent to obtain the control (eq. 3.68). In these cases, the **Pontryagin Maximum Principle** (PMP) allows to derive more general necessary conditions for optimality. The advent of this principle in '50s can be considered the birth of the mathematical theory of optimal control.
A path constraint for the control of the form

$$\mathbf{g}(\mathbf{u}(t), t) \leq 0 \tag{3.74}$$

defines the region of the feasible control, $\mathbf{u}(t) \in \mathbb{U}$. Perturbing the control

$$\mathbf{u} = \mathbf{u}^\star + \delta \mathbf{u} \tag{3.75}$$

the change in the cost function is expressed as:

$$\Delta J(\mathbf{u}^\star, \delta \mathbf{u}) = \delta J(\mathbf{u}^\star, \delta \mathbf{u}) + \textit{higher order terms} \tag{3.76}$$

Since, by definition, the cost function evaluated at the optimal control is at its minimum, it is valid that:

$$J(\mathbf{u}) - J(\mathbf{u}^\star) = \Delta J(\mathbf{u}^\star, \mathbf{u}) \geq 0 \tag{3.77}$$

and so, for sufficiently small perturbation $\delta\mathbf{u}$, the cost function has a local minimum if :

$$\delta J(\mathbf{u}^\star, \delta\mathbf{u}) \geq 0 \tag{3.78}$$

Considering the definition of the Hamiltonian and of the Lagrangian, at the optimal solution, the only term left is:

$$\begin{aligned} J_a &= \int_{t_0}^{t_f} \frac{\partial H}{\partial \mathbf{u}}(\mathbf{x}^\star(t), \mathbf{u}^\star(t), \lambda^\star(t), t)\delta\mathbf{u}dt \\ &= \int_{t_0}^{t_f} H(\mathbf{x}^\star(t), \mathbf{u}^\star(t) + \delta\mathbf{u}(t), \lambda^\star(t), t) - H(\mathbf{x}^\star(t), \mathbf{u}^\star(t), \lambda^\star(t), t)dt \end{aligned} \tag{3.79}$$

Since the variation of the cost function has been found to be non-negative, it is valid that :

$$H(\mathbf{x}^\star(t), \mathbf{u}^\star(t) + \delta\mathbf{u}(t), \lambda^\star(t), t) \geq H(\mathbf{x}^\star(t), \mathbf{u}^\star(t), \lambda^\star(t), t) \tag{3.80}$$

The Pontryagin Maximum Principle sets that the optimal control is the admissible function that minimizes the Hamiltonian:

$$\mathbf{u}^\star(t) = arg\min_{\mathbf{u}\in\mathbb{U}}[H(\mathbf{x}^\star(t), \mathbf{u}^\star(t), \lambda^\star(t), t)] \tag{3.81}$$

In the original paper of Pontryagin, the Hamiltonian has a different sign: for this reason, he found a *Maximum* principle, while this derivation takes to a *Minimum* principle.
Thanks to this expression for the control, an analysis can be done on the structure of it. Considering the cost functions previous in the chapter and hypothesizing that the system dynamics does not depend more than linearly on the control, two minimum problems with constraints bounds ($|u(t)| \leq 1$) are analysed in particular: minimum fuel/time and minimum energy.
The first is characterized by a **bang-bang** control structure: the control law allows the control variables to assume only extreme values. Considering the simple example of the control problem of a material point moving in linear motion:

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = u(t) \end{cases} \tag{3.82}$$

with fixed initial and final conditions. The costate equations are:

$$\begin{cases} \dot{\lambda}_1(t) = 0 \\ \dot{\lambda}_2(t) = \lambda_1(t) \end{cases} \tag{3.83}$$

In the case of minimum time problem, the control law is expressed as:

$$u(t) = -sign(\lambda_2(t)) \tag{3.84}$$

In the case of minimum fuel:

$$u(t) = -sign(1 + \lambda_2(t)) \tag{3.85}$$

The trajectories are shown in figure 3.10.

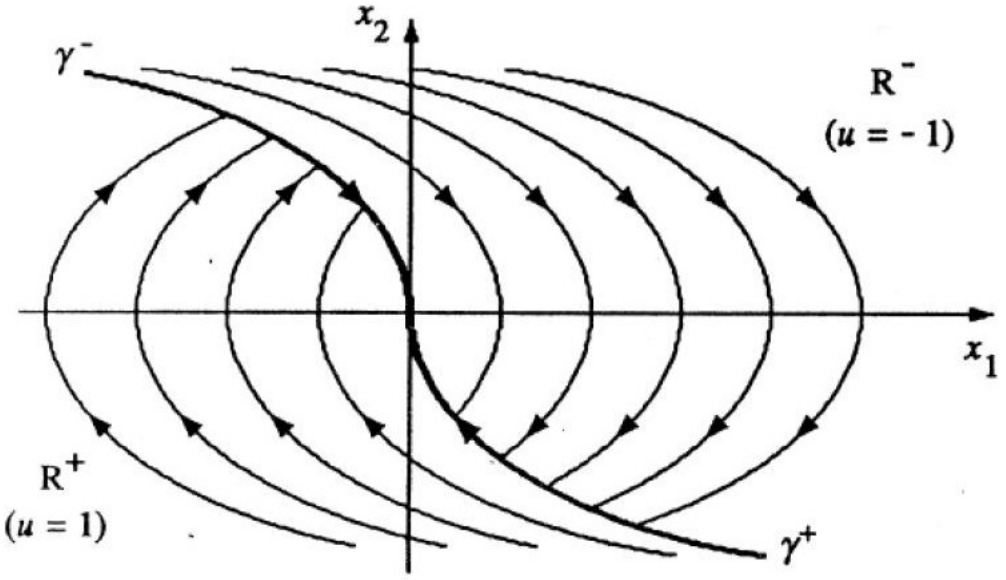


Figure 3.10: Trajectories for *bang-bang* control law

Analysing the transversality condition for not specified final time and the costate equations, it is then possible to obtain the number of switching times. Since λ_2 is a linear function of t and, in the case of minimum time, its value at the final time is not null (since $H(t_f) = 1 + \lambda_2(t_f)u(t_f) = 0$), only one commutation of the control value can be possible, that can be obtained by integrating the dynamic equations with specific value of the control.
The case of minimum energy gives a continuous control law:

$$u(t) = \begin{cases} -\lambda_2(t) & |\lambda_2(t)| \leq 1 \\ -sign(\lambda_2(t)) & |\lambda_2(t)| \geq 1 \end{cases} \tag{3.86}$$

It is evident that a *bang-bang* control law can produce a bigger difficulty, due to its discontinuous nature.
An important characteristic of the Hamiltonian arises for autonomous sys-

tems [5], by applying the time derivative :

$$\begin{aligned}\dot{H}(\mathbf{x}(t),\mathbf{u}(t),\lambda(t)) &= H_{\mathbf{x}}\dot{\mathbf{x}} + H_{\lambda}\dot{\lambda} + H_{\mathbf{u}}\dot{\mathbf{u}} \\ &= (-\dot{\lambda})^T\dot{\mathbf{x}} + \dot{\mathbf{x}}^T\dot{\lambda} + 0\dot{\mathbf{u}} \\ &= 0\end{aligned} \tag{3.87}$$

It is then obtained that the Hamiltonian is constant for autonomous systems. The numerical methods are called *Indirect*, because the solution for the optimal problem is obtained *indirectly* through the costate variables.

Costate variables These variables, called also adjoint variables, are the real source of issues for these methods. First of all, they do not have a real physical meaning with respect to the states. In order to obtain one, we consider the variation in the cost function J in control $\mathbf{u}(t)$ for *fixed times* t_0 and t_f and no terminal constraints (Ψ is empty):

$$\delta J_a = \left[\left(\frac{\partial\Phi}{\partial\mathbf{x}} - \lambda^T\right)\delta\mathbf{x}\right]_{t=t_f} + [\lambda^T\delta\mathbf{x}]_{t=t_0} + \int_{t_0}^{t_f}\left[\left(\frac{\partial H}{\partial\mathbf{x}} + \dot{\mathbf{x}}^T\right)\delta\mathbf{x} + \frac{\partial H}{\partial\mathbf{u}}\delta\mathbf{u}\right]dt \tag{3.88}$$

Considering the costate equation (eq. 3.67) and the boundary conditions (eq. 3.72), the cost function variation becomes:

$$\delta J_a = \lambda^T(t_0)\delta\mathbf{x}(t_0) + \int_{t_0}^{t_f}\frac{\partial H}{\partial\mathbf{u}}\delta\mathbf{u}dt \tag{3.89}$$

$\lambda^T(t_0)$ is the gradient of cost function J with respect to variations in initial conditions, keeping constant the control $\mathbf{u}(t)$. The costate variables are also called $influence\ functions$ on J of variations in $\mathbf{x}(t)$ with arbitrary t_0. The functions $\frac{\partial H}{\partial\mathbf{u}}$ represent the variation of the cost function due to a unit impulse of control $\delta\mathbf{u}$, keeping $\mathbf{x}(t_0)$ constant: they are called *impulse response functions* [11].

The great advantage of these methods is the fact that the obtained solution is the optimal one, while the direct methods develop a suboptimal solution, due to the control parametrization. The second reason for an increasing interest is the accuracy: since the procedure of these methods is "*optimize, then discretize*", a better solution is obtained with respect to direct methods, characterized by an opposite idea. A third aspect is linked to the implemented Newton approach: a very fast numerical convergence in the neighbourhood of the optimal solution. A last advantage can be found

[5] In an autonomous systems, the integrand cost $l(\mathbf{x}(t),\mathbf{u}(t))$ and the dynamic system $\mathbf{f}(\mathbf{x}(t),\mathbf{u}(t))$ do not depend explicitly on time.

in the number of variables, very small even for large scale systems with respect to direct strategies, that makes these methods very attractive. Three are the most important disadvantages: the first is linked to the derivation of Euler-Lagrange equations, the second concerns the initial conditions and the third the constraints.
The derivation of the **optimality conditions** can be very difficult when the optimal control problem is highly non-linear. This makes also mandatory to re-derive the complete set of equations in case of a change in the original optimal control problem formulation. It is then not very convenient. Linked to this, the resulting ODE are strongly nonlinear and unstable.
The second aspect is the extreme sensitiveness to **initial conditions**: as Bryson and Ho explained in [11], "The main difficulty with these methods is *getting started*". Since the methods are based on Newton strategies, the region of convergence is quite small, obliging to use other methods to obtain a good estimate for these variables. Several methods are possible: use a simpler problem, more likely to converge, as initial conditions; implement a direct or another method, creating an hybrid method (section 3.6.2). The great issue is for adjoint variables: it is very difficult to find a first estimate of the conditions at one end producing a solution reasonably close to specified conditions at the other end. Since they have no real physical meaning, they are very difficult to initialize. This great difficulty depends on the extreme sensitiveness of extremal solutions with respect to small changes in unspecified boundary conditions. The reason is in the nature of the Euler-Lagrange equations: they are influence functions equations. The costate functions are "adjoint differential equations to the linear perturbation system equations"[11]. If the fundamental solution $\mathbf{x}(t)$ decrease in magnitude as time increases, the solution of the adjoint equations $\lambda(t)$ increases in magnitude: they tend to become extremely different in magnitude as integration goes on. This difference becomes important in computer calculation and increases the loss of accuracy. Small errors in initial estimate can create big errors in influence functions at final time. This is more remarkable for highly dissipative systems, as the one with friction or drag.
The last disadvantage is linked to the presence of **constraints**: it is necessary to know a priori the solution structure (active and non-active constraints, singular arcs, switching times), if not it is very difficult to solve the problem, because of discontinuous integrated functions and singular Jacobian matrix. There exists several strategies:

I Gradient methods;

II Indirect Single Shooting Method;

III Indirect Multiple Shooting Method;

IV Homotopic Approach.

3.5.1 Gradient Methods or Backward-Sweep Method

The basic idea is to integrate each set of equations in the stable direction. In particular, integrate the state equations forward in time, while the adjoint equations are integrated backward from t_f to t_0. A simple procedure can be presented:

1. Make an initial guess for the control $\mathbf{u}(t)$ over the interval and store the initial guess for it;
2. Using initial conditions $\mathbf{x}(t_0) = \mathbf{x}_0$ and stored value of control, solve forward in time the dynamic equations;
3. Using the transversality conditions on adjoint variables and stored value of control and state trajectories, solve backward in time the costate equations;
4. Update the new control with its equation by entering the new state and costate trajectories;
5. Check the convergence: if variations with respect to last iteration is small, the output is the solution, if not, return to step 2.

To increase the convergence, a complex combination of previous and actual control can be implemented to update the control: for example, the mean between the two values.
An important issue is the final conditions of the costate variables: when the final states are not defined and the integrand cost is not directly dependent on the final states, the terminal conditions for adjoint variables are very simple, making the method much simpler to implement. When these conditions are not met, different adjustments have to be created, usually exploiting shooting method.

3.5.2 Indirect Single Shooting Method

The idea of this method is to exploit the Newton methods to iteratively modify the initial estimates for the adjoint variables, the Lagrangian Multipliers and the terminal time, that constitute the optimization parameters, to meet the transversality conditions (eq. 3.70, 3.71. 3.72), that represents the function to vanish with the Newton method. The main difference between the indirect and the direct single shooting method is the definition of the control

function $\mathbf{u}(t)$: the direct approach parametrizes it, while the indirect defines it by maximum principle.
A simple procedure is:

1. Initialization: choose initial estimates for costate variables λ_0^0, Lagrangian multipliers ν^0, terminal time t_f^0, impose the iteration to 0: k=0;

2. Calculate the defect $\mathbf{F}(\lambda_0^0, \nu^0, t_f^0)$, integrating the Euler-Lagrange equations:

$$\mathbf{F}(\lambda_0^0, \nu^0, t_f^0) = \begin{bmatrix} \lambda - \Phi_{\mathbf{x}} + \nu^{k,T}\Psi_{\mathbf{x}} \\ \Psi \\ l + \lambda^T \mathbf{f} + \Phi_t + \nu^{k,T}\Psi_t \end{bmatrix}_{t=t_f^k} \rightarrow 0 \tag{3.90}$$

   if $\| \mathbf{F} \| < \epsilon$, STOP;

3. Calculate the defect gradients $\nabla_{\lambda_0}\mathbf{F}$, $\nabla_{\nu}\mathbf{F}$ and $\nabla_{t_f}\mathbf{F}$;

4. Calculate the search direction via solution of the linear system:

$$\begin{bmatrix} \nabla_{\lambda_0}\mathbf{F} \\ \nabla_{\nu}\mathbf{F} \\ \nabla_{t_f}\mathbf{F} \end{bmatrix}^T \begin{bmatrix} \mathbf{d}_{\lambda}^k \\ \mathbf{d}_{\nu}^k \\ d_{t_f}^k \end{bmatrix} = -\mathbf{F}(\lambda_0^0, \nu^0, t_f^0) \tag{3.91}$$

5. Update the estimates:

$$\lambda_0^{k+1} = \lambda_0^k + \mathbf{d}_{\lambda}^k \tag{3.92}$$
$$\nu^{k+1} = \nu^k + \mathbf{d}_{\nu}^k \tag{3.93}$$
$$t_f^{k+1} = t_f^k + d_{t_f}^k \tag{3.94}$$

6. Increment k: k=k+1 and return to step 2.

An improvement can be obtained using a Damped Newton Method, in step 5:

$$x_{k+1} = x_k + \lambda_k d_k \tag{3.95}$$

The great advantages of this method are its simplicity in numerical implementation and the very small number of variables.
The main issues are four: the defect gradient, the initial conditions, constraints and small changes at the beginning of the trajectory.
Since a Newton method is implemented (step 3), the **defect gradients** with

respect to the optimization parameters have to be computed. A first strategy is the numerical differentiation:

$$J_{ij} = \frac{\partial f_i}{\partial x_j}(x_1, \ldots, x_n) \approx \frac{f_i(x_1, \ldots, x_j + h, \ldots, x_n) - f_i(x_1, \ldots, x_n)}{h} \quad (3.96)$$

A good choice for h is a balance between the round-off error and the computational ϵ:

$$h = \sqrt{macheps(1 + |x|)} \quad (3.97)$$

A second strategy is the Broyden update scheme for the Jacobian. This strategy is based on a "secant method" :

$$f'(x^k) \approx \frac{f(x^k) - f(x^{k-1})}{x^k - x^{k-1}} \quad (3.98)$$

$$\mathbf{J}_k(x^k - x^{k-1}) = f(x^k) - f(x^{k-1}) \quad (3.99)$$

The idea is then to obtain the current Jacobian by a modification of the previous one:

$$\mathbf{J}_{k+1} = \mathbf{J}_k + \frac{f(x^{k-1})(\Delta x^k)^T}{\| \Delta x^k \|_2^2} \quad (3.100)$$

with $\Delta x^k = x^k - x^{k-1}$. The second issue concerns the **initial conditions** for the optimization parameters, due to the small convergence radius. The third issue is linked to **constraints**: in order to deal with them, the switching structure has to be known a priori. These problems are treated as a sequence of constrained and unconstrained arcs.
A last difficulty is in the fact that a **small change** early in the trajectory can propagate into very nonlinear variations at the end of the trajectory.
Software as ADIFOR and OCCAL implement this method.

3.5.3 The Indirect Multiple Shooting Method

The idea is quite simple: subdivide the time interval and apply the single shooting method on each subinterval. The procedure can be summarized as:

1. Subdivide the time interval into m-1 fixed subintervals: $t_0 = t_1 < \cdots < t_m = t_f$
2. Guess the initial values for costate variables and for each subinterval, also the state variables: $\mathbf{Z}_j^0$, set k=0;
3. Compute the solution of the IVP with these initial guess, define with $\mathbf{z}_j$ the terminal values of this solution at each subinterval;

4. Compute the jumps: the final conditions obtained by the integration in one subinterval have to coincide with the guessed initial conditions of the next subinterval:

   - for internal subintervals, matching conditions have to be satisfied:

   $$F_j(Z_1, \ldots, Z_{m-1}) = z_j - Z_{j+1} \tag{3.101}$$

   - for the terminal subinterval, the transversality conditions must hold:

   $$F_{m-1}(Z_1, \ldots, Z_{m-1}) = \begin{bmatrix} \lambda - \Phi_{\mathbf{x}} + \nu^{k,T}\Psi_{\mathbf{x}} \\ \Psi \\ l + \lambda^T \mathbf{f} + \Phi_t + \nu^{k,T}\Psi_t \end{bmatrix}_{t=t_f^k} \tag{3.102}$$

   $$\mathbf{F} = \begin{bmatrix} z_1 - Z_2 \\ \cdots \\ z_{m-2} - Z_{m-1} \\ \lambda - \Phi_{\mathbf{x}} + \nu^{k,T}\Psi_{\mathbf{x}} \\ \Psi \\ l + \lambda^T \mathbf{f} + \Phi_t + \nu^{k,T}\Psi_t \end{bmatrix} \tag{3.103}$$

5. Numerical approximation of the Jacobian Matrix $\mathbf{J}$ of $\mathbf{F}(\mathbf{Z})$ via numerical differentiation or Broyden update of the previous matrix;

6. Compute the Newton correction $\Delta\mathbf{Z}$:

   $$\mathbf{J}\Delta\mathbf{Z} = -\mathbf{F}(\mathbf{Z}) \tag{3.104}$$

7. Determine an appropriate relaxation factor $\lambda \in [0, 1]$ and compute the new initial guesses:

   $$\mathbf{Z}^{new} = \mathbf{Z} + \lambda\Delta\mathbf{Z} \tag{3.105}$$

8. If convergence ($\| \mathbf{F} \| < \epsilon$) is not met, return to step 3, set k=k+1.

The main advantage is the reason of its birth: since in single shooting methods, a small change at the beginning of the time interval creates an important variation at the terminal end, subdividing the horizon domain in subintervals, it should reduce this inconvenience. As for direct multiple shooting, parallel processor can be exploited.
The main disadvantages are the same as single shooting, but another important one is peculiar to this method: the number of variables can be very high. But as for direct methods, matrices present a strongly sparse character, that

has to be exploited. The presence of constraints can be less problematic if the time intervals coincides with the singular arcs and the switching times. Still, the switching structure of constrained and unconstrained arcs has to be known a priori. The state and costate variables at the switching times represent new optimization parameters and the fulfilment of the constraints constitutes the defect function.
In Germany in 1989, a software implementing this method has been developed: BNDSCO, by Oberle and Grimm [20].

3.5.4 Homotopic Approach

The principle of a homotopic approach, known also as continuation method, is to solve a difficult problem, by solving a sequence of simpler ones and using the solution of one problem as initial guess for slightly modified one [36, 37, 38, 39]: for example, solve the orbit transfer with a two-body dynamics as initial guess for a more accurate dynamic problem. The connection is operated through an application, H, called homotopy, with the right properties linking the two problems. If r is the simpler problem and f is the original one, H is constructed so that:

$$H(-,0) = r \tag{3.106}$$

$$H(-,1) = f \tag{3.107}$$

A classic implementation for this method is to initialize a minimum fuel problem, presenting normally a bang-bang control structure, with a minimum energy problem, characterized by a continuous control structure. The homotopy connection needs a homotopic criterion to link the problems, between the various possibility, two are presented:

- convex criterion :

$$\int_{t_0}^{t_f} p \parallel u(t) \parallel + (1-p) \parallel u(t) \parallel^2 dt \tag{3.108}$$

- power criterion:

$$\int_{t_0}^{t_f} \parallel u(t) \parallel^{2-p} dt \tag{3.109}$$

with $p = 0$, the fuel problem and $p = 1$ the energy problem.
To solve the homotopy problem, the homotopic parameter has to be modified from 0 to 1. Three possibilities exist:

I the simplest way is to manually increase the parameter p in N steps;

II *differential homotopy*: dynamically follow the homotopy path with a predictor-corrector scheme;

III *simplicial homotopy*: subdivide the search space into subintervals by piecewise linear approximations.

The main advantage of this method is the fact that the control structure does not have to be known a priori, but it is calculated automatically.
The great disadvantage is linked to the case-dependent nature of the approach: a strong experience is requested to understand which simpler problem and how to link it to the original one. The computational effort is evidently a consequence of these choices.
Often this approach is used as initial condition for a shooting method, especially for the costate variables and the Lagrangian Multipliers.

3.6 Other Methods

Besides these methods, other exist, as the Genetic Algorithms and the Hybrid ones.

3.6.1 Genetic Algorithms

The *Genetic Algorithms* (GA) are one class of the *heuristic optimization methods* [17]. This technique is a global one, with respect to gradient methods, that are local. The main idea of an heuristic method is that the search is performed in a stochastic way, differently from the deterministic way of the other methods.
A Genetic Algorithm is an evolutionary algorithm approach: it emulates the evolutionary process of genetics on a computer. Firstly, an initial population of possible solutions is chosen. Each solution is characterized by a particular gene, whose quality is reflected in a *fitness*, typically the objective function to be minimized in the optimization problem. Genes are then recombined, *mutated*, to generate new populations. The genes with the highest fitness survive: it is a selection mechanism based on the concept of "survival-of-the-fittest".
Several approaches exist, the most famous are two: **Simulated Annealing** and **Particle Swarm Optimization** (PSO). In the first, the fitness coincides to the internal energy of the system. With a probabilistic approach, it decides to stay on the current state or move to the neighbouring one considering the fitness value and a global *temperature* parameter.
The PSO is a particular type of GA: it is a population-based stochastic

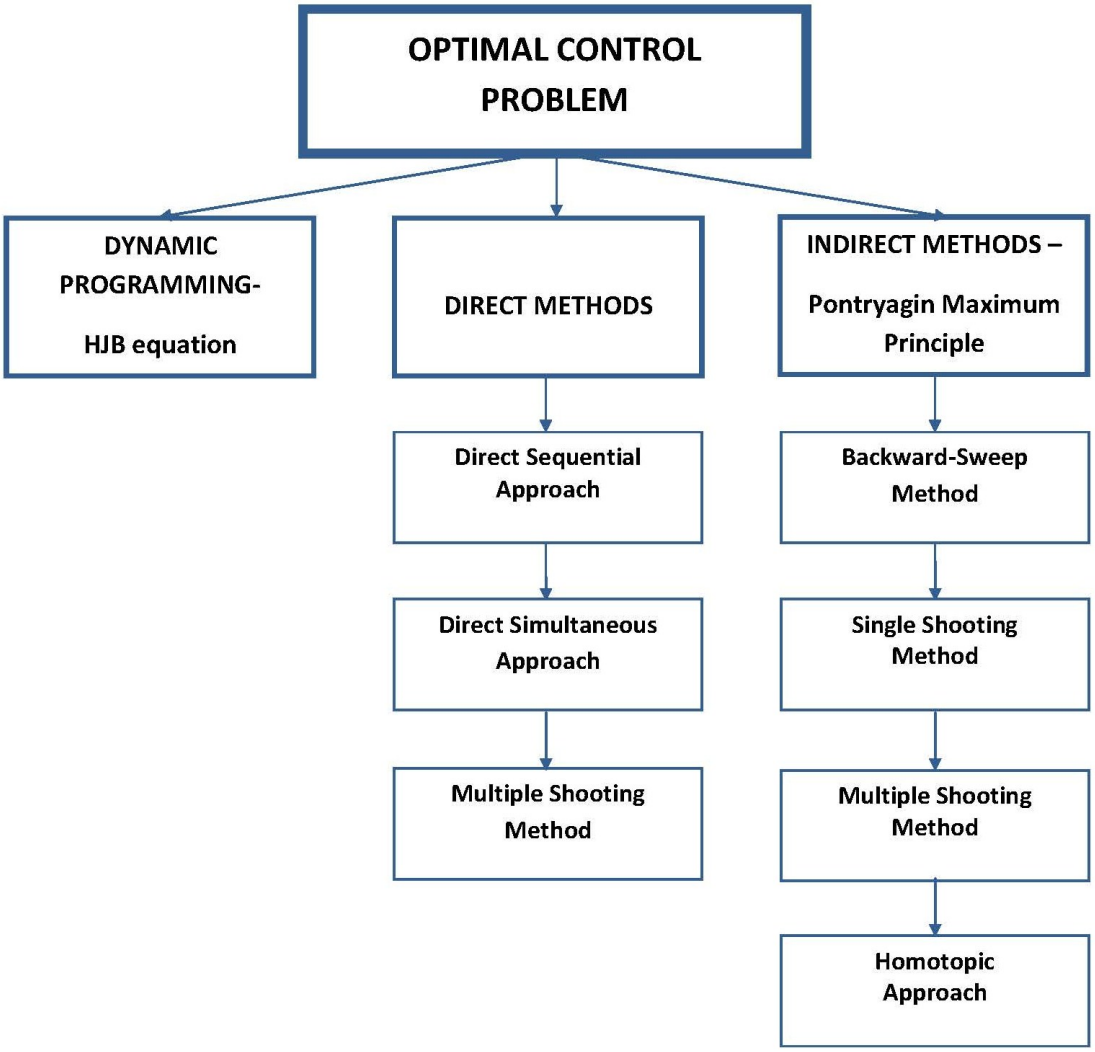


Figure 3.11: The Optimal Control Solution Approaches

method, based on the idea of swarms of animals. The candidate solutions, the *particles*, follow the particles with the lowest cost, recording the best solutions for the particles and for the neighbours. The PSO moves in the direction of the particles and neighbours best solution.
Despite several improvements, these methods keep a higher computational cost with respect to gradient methods.

3.6.2 Hybrid Methods

These methods are the most general, in the sense that they are typically composed of two different methods, to compensate the disadvantages of one with the advantages of the other. The most common is to use a direct method to initialize an indirect one: in this way, the fast convergence of the indirect is obtained, without the initialization issue, compensated by the direct one. Often, a pseudospectral method is used to determine the costate variables, that are the most difficult to initialize, through the discretized Lagrangian Multipliers [21]. Recently, also the PSO is used to initialize the indirect methods.
The idea is generally to exploit the great convergence properties of the indirect methods, eliminating the initialization issue with a simpler initialized method.

Chapter 4

Models

The Hill-Clohessy-Wiltshire equations are defined on the Hill's reference frame, that is a Cartesian frame. However, the equations are often interpreted as curvilinear [6]: x as radial variation ($x = \Delta r$) and y as the curved along-track position ($y = r\Delta\theta$). Even though these equations are rigorous only for circular orbits, small distances and Hill's reference frame, Schaub [22] has demonstrated that they are valid also for curvilinear coordinates.
As already implemented by Leonard [3], the Hill-Clohessy-Wiltshire equations are transformed through a state transformation into two decoupled linear systems composed by a double integrator and an harmonic oscillator:

$$\begin{cases} x_o = x - x_m \\ y_o = y - y_m \end{cases} \tag{4.1}$$

where x, y are the global states, x_m, y_m are the mean position of the chaser with respect to the target and x_o, y_o are the oscillatory components (expressed as "eccentricity" in Leonard's work). The transformation is simply [6]:

$$\begin{cases} x_m = 4x + \frac{2}{\omega}\dot{y} \\ y_m = y - \frac{2}{\omega}\dot{x} \end{cases} \tag{4.2}$$

The introduction of the J_2 gravity perturbation by Schweighart and Sedwick takes to a new state transformation [6]:

$$\begin{cases} x_m = \frac{4c^2}{2-c^2}x + \frac{2c}{(2-c^2)\omega}\dot{y} \\ y_m = y - \frac{2c}{(2-c^2)\omega}\dot{x} \end{cases} \tag{4.3}$$

where c is the Schweighart-Sedwick coefficient, already defined in Chapter 2, and ω is the target constant orbital rate. It is evident from the definition

of the Schweighart-Sedwick coefficient that its value is 1 when J_2 is not considered ($J_2 = 0$):

$$c = \sqrt{1 + \frac{3J_2R_\oplus^2}{8r_T^2}\left[1 + 3cos(2i_T)\right]} \tag{4.4}$$

and the first state transformation is recovered from the second when $c = 1$. Applying this state transformation to the Schweighart-Sedwick set of equations, the new system is:

$$\begin{cases} \dot{x}_m = \frac{2c}{(2-c^2)\omega}u_y \\ \dot{y}_m = \frac{(2-5c^2)\omega}{2c}x_m \\ \dot{x}_o = \frac{(2-c^2)\omega}{2c}y_o - \frac{2c}{(2-c^2)\omega}u_y \\ \dot{y}_o = -2\omega c x_o \end{cases} \tag{4.5}$$

The decoupled nature of the system is evident.

Optimal Control Problem Formulation

As explained in the previous chapter, the optimal control problem needs the formulation of a cost function:

$$J = \Phi(x(t_f), t_f) + \int_{t_0}^{t_f} l(t, x(t), u(t))dt \tag{4.6}$$

Three cost functions are considered: minimum time, minimum energy and a combination of the two:

1. Minimum time problem (with $t_0 = 0$):

$$J = \int_{t_0}^{t_f} 1 \; dt = t_f - t_0 = t_f \tag{4.7}$$

2. Minimum energy problem:

$$J = \int_{t_0}^{t_f} \frac{1}{2}u^2(t)dt \tag{4.8}$$

3. Combination of minimum time and minimum energy:

$$J = w_f \; t_f + \int_{t_0}^{t_f} \frac{1}{2}u^2(t)dt \tag{4.9}$$

For the last problem, a weight coefficient is necessary for the final time to obtain the same order of magnitude with respect to the control part of the cost function.
The system dynamics are defined by the Schweighart-Sedwick set of equations.
The last component of the optimal control problem formulation are the constraints:

1. Initial Conditions :
$$\mathbf{x}(t_0) = \mathbf{x}_0 \tag{4.10}$$

2. Boundary Conditions :
$$\Psi(\mathbf{x}(t_f), t_f) = 0 \quad \rightarrow \quad \mathbf{x}(t_f) = \mathbf{0} \tag{4.11}$$

3. Control bounds :
$$|u(t)| \leq u_{Max} \tag{4.12}$$

In this case, no path constraints or state constraints are considered.
The first condition is simply a fixed initial position with a fixed initial time: this implies that the transversality conditions deriving from the variations on these variables have no sense. The equations 3.69 and 3.71 are not considered.
The second condition is the rendez-vous condition: state variables to zero imply that target and chaser have the same position. As before, the transversality condition related to the variation of the terminal states (eq. 3.72) is not considered.
Finally, the control bounds. These are evidently linked to the nature of the control: the differential drag. In Chapter 2, it has been analysed how this can be implemented. In particular, two possibilities exist: geometrical or attitude changes. Leonard [3] and Bevilacqua-Romano [4,5] have considered the first, by adding drag plates; Dell'Elce-Kerschen [7,8] the second: varying the attitude, the exposed surface is modified (figure 4.1). This last idea can be more attractive for small size satellites, as CubeSats, because it does not add weight for the extra-system, keeping free space for scientific instruments. However, it has a disadvantage: the maximum value of the control can be very small, due to the limits on the surface values. An optimal trajectory becomes even more important.

A last aspect is the way these bounds are expressed:

$$|u(t)| \leq u_{Max} \tag{4.13}$$

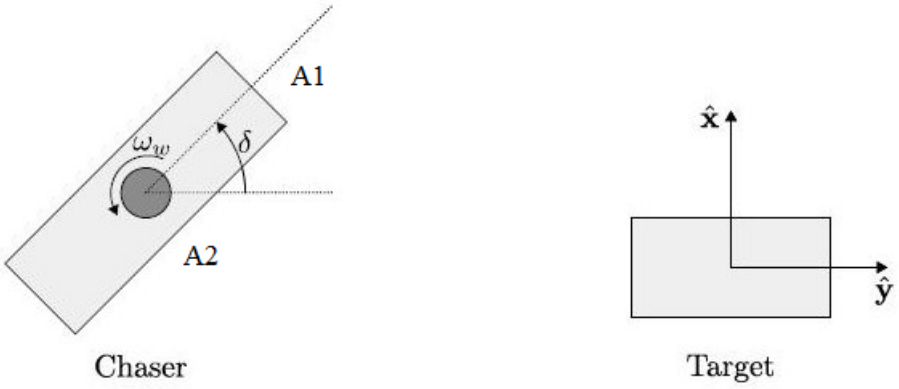


Figure 4.1: System attitude (from [8])

This equation implies that the control law is symmetric with respect to 0, because of the presence of the norm operator. In fact, for the minimum time problem, it is valid that:

$$\begin{cases} u(t) = u_{Max} & if\ u(\mathbf{x}(t), \lambda(t), t) > 0 \\ u(t) = -u_{Max} & if\ u(\mathbf{x}(t), \lambda(t), t) < 0 \end{cases} \tag{4.14}$$

where $u(\mathbf{x}(t), \lambda(t), t)$ represents the control law. This considerably simplifies the notation and the numerical implementation.
In the next sections of the chapter, the different models would be presented. As it is evident, the "original" system -classical Schweighart-Sedwick equations- has not been considered: the reason is in the extreme difficulty for the convergence. This is due to the necessary of coupled integration of mean and oscillatory components together, whose separate movements are characterized by different order of magnitude and expressions.

4.1 First Model: Mean with Differential Drag

In this first case, the simplest model is considered: the only mean component of the movement. Since the uncoupled nature of the dynamic system, this does not influence the global trajectory, in the sense that it can be seen as a first phase of a two-phase optimization [1]. Actually, a second phase is not implemented, the reason is simple: this first model is used to *try* the numerical method in order to implement the best version of it.

[1] In the same way as Leonard [3] and Bevilacqua-Romano [4,5]

As already explained, the optimal control problem has a dynamic system:

$$\begin{cases} \dot{x}_m = \frac{2c}{(2-c^2)\omega} u \\ \dot{y}_m = \frac{(2-5c^2)\omega}{2c} x_m \end{cases} \tag{4.15}$$

initial conditions, terminal conditions, control bounds:

$$\begin{cases} \mathbf{x}(0) = \mathbf{x}_0 & \begin{cases} x_m(0) = x_{m0} \\ y_m(0) = y_{m0} \end{cases} \\ \mathbf{x}(t_f) = \mathbf{0} & \begin{cases} x_m(t_f) = 0 \\ y_m(t_f) = 0 \end{cases} \\ |u(t)| \leq u_{Max} \end{cases} \tag{4.16}$$

and a cost function, which is different for each type of problem. The Calculus of Variations together with the Pontryagin Maximum Principle is applied. The Hamiltonian changes with different cost functions, but the costate equations are the same, since the cost functions are not related to state variables:

$$\begin{cases} \dot{\lambda}_1 = -\frac{(2-5c^2)\omega}{2c} \lambda_2 \\ \dot{\lambda}_2 = 0 \end{cases} \tag{4.17}$$

Finally, the transversality conditions. Since initial time, initial conditions and terminal conditions are fixed, the only transversality condition is the one on the final time:

$$\begin{aligned} H(t_f) - \nu^T \frac{\partial \Psi}{\partial t_f} + \frac{\partial \Phi}{\partial t_f} &= 0 \\ H(t_f) + \frac{\partial \Phi}{\partial t_f} &= 0 \\ l(u(t_f), t_f) + \lambda_1(t_f) \frac{2c}{(2-c^2)\omega} u(t_f) + \lambda_2 \frac{(2-5c^2)\omega}{2c} x_m(t_f) + \frac{\partial \Phi}{\partial t_f} &= 0 \end{aligned} \tag{4.18}$$

Actually, the transversality conditions related to the terminal conditions should be considered, because the final time is not fixed. Since the states are fixed at the final time ($\frac{\partial \Psi}{\partial \mathbf{x}_f} = 1$) and the three terminal costs do not depend on state variable, but only on final time, the expression for these conditions is quite simple:

$$\lambda(t_f) = -\nu^T \tag{4.19}$$

The Lagrangian Multipliers ν are also unknown variables, but constant in time. Since they do not really add information, the costate variables are just kept free at final time.
The different cost functions determine different control law and different transversality conditions.

4.1.1 Minimum Time Problem

The cost function is the one on equation 4.7.
The Hamiltonian is defined as:

$$H = 1 + \lambda_1(t)\frac{2c}{(2-c^2)\omega}u(t) + \lambda_2(t)\frac{(2-5c^2)\omega}{2c}x_m(t) \tag{4.20}$$

The transversality condition is :

$$\begin{aligned} H(t_f) - \nu^T\frac{\partial\Psi}{\partial t_f} + \frac{\partial\Phi}{\partial t_f} &= 0 \\ H(t_f) &= 0 \\ 1 + \lambda_1(t_f)\frac{2c}{(2-c^2)\omega}u(t_f) + \lambda_2(t_f)\frac{(2-5c^2)\omega}{2c}x_m(t_f) &= 0 \end{aligned} \tag{4.21}$$

The control law is :

$$H_u = 0 \quad \rightarrow \quad \lambda_1(t)\frac{2c}{(2-c^2)\omega} \quad \rightarrow \quad u(t) = -sign\left(\lambda_1\frac{2c}{(2-c^2)\omega}\right)u_{Max} \tag{4.22}$$

As typically for minimum time problem, the control law is of bang-bang type: the control can assume only extreme values.

4.1.2 Minimum Energy Problem

The cost function is the one on equation 4.8.
The Hamiltonian is defined as:

$$H = \frac{1}{2}u^2(t) + \lambda_1(t)\frac{2c}{(2-c^2)\omega}u(t) + \lambda_2(t)\frac{(2-5c^2)\omega}{2c}x_m(t) \tag{4.23}$$

The transversality condition is :

$$\begin{aligned} H(t_f) - \nu^T\frac{\partial\Psi}{\partial t_f} + \frac{\partial\Phi}{\partial t_f} &= 0 \\ H(t_f) &= 0 \\ \frac{1}{2}u^2(t_f) + \lambda_1(t_f)\frac{2c}{(2-c^2)\omega}u(t_f) + \lambda_2(t_f)\frac{(2-5c^2)\omega}{2c}x_m(t_f) &= 0 \end{aligned} \tag{4.24}$$

The control law is :

$$H_u = 0 \;\rightarrow\; u(t) + \lambda_1(t)\frac{2c}{(2-c^2)\omega} = 0 \;\rightarrow$$

$$u(t) = \begin{cases} -\lambda_1(t)\frac{2c}{(2-c^2)\omega} & \left|\lambda_1(t)\frac{2c}{(2-c^2)\omega}\right| \leq u_{Max} \\ -sign\left(\lambda_1(t)\frac{2c}{(2-c^2)\omega}\right)u_{Max} & \left|\lambda_1(t)\frac{2c}{(2-c^2)\omega}\right| \geq u_{Max} \end{cases} \tag{4.25}$$

For the minimum energy problem, the control law is of continuous type.

4.1.3 Minimum Time and Energy Problem

In this case, the cost function is a combination of the two, as the one on equation 4.9.
The Hamiltonian is defined as:

$$H = \frac{1}{2}u^2(t) + \lambda_1(t)\frac{2c}{(2-c^2)\omega}u(t) + \lambda_2(t)\frac{(2-5c^2)\omega}{2c}x_m(t) \tag{4.26}$$

The transversality condition is :

$$\begin{aligned} H(t_f) - \nu^T\frac{\partial\Psi}{\partial t_f} + \frac{\partial\Phi}{\partial t_f} &= 0 \\ H(t_f) + \frac{\partial\Phi}{\partial t_f} &= 0 \\ \frac{1}{2}u^2(t_f) + \lambda_1(t_f)\frac{2c}{(2-c^2)\omega}u(t_f) + \lambda_2(t_f)\frac{(2-5c^2)\omega}{2c}x_m(t_f) + w_f &= 0 \end{aligned} \tag{4.27}$$

The control law is the same as the previous problem:

$$H_u = 0 \;\rightarrow\; u(t) + \lambda_1(t)\frac{2c}{(2-c^2)\omega} = 0 \;\rightarrow$$

$$u(t) = \begin{cases} -\lambda_1(t)\frac{2c}{(2-c^2)\omega} & \left|\lambda_1(t)\frac{2c}{(2-c^2)\omega}\right| \leq u_{Max} \\ -sign\left(\lambda_1(t)\frac{2c}{(2-c^2)\omega}\right)u_{Max} & \left|\lambda_1(t)\frac{2c}{(2-c^2)\omega}\right| \geq u_{Max} \end{cases} \tag{4.28}$$

4.2 Second Model: Mean-Oscillatory with Differential Drag

This second model considers both the dynamics, the double integrator and the harmonic oscillator: it should represent the global trajectory. The control is still the differential drag, so the dynamic equations are the ones already presented (equation 4.5). The Hamiltonian is expressed as:

$$\begin{aligned} H = l(u(t)) &+ \lambda_1(t)\frac{2c}{(2-c^2)\omega}u(t) + \lambda_2(t)\frac{(2-5c^2)\omega}{2c}x_m(t) \\ &+ \lambda_3(t)\frac{(2-c^2)\omega}{2c}y_o(t) - \lambda_3 u(t)\frac{2c}{(2-c^2)\omega} - \lambda_4(t)2\omega c x_o(t) \end{aligned} \tag{4.29}$$

The costate equations are :

$$\begin{cases} \dot{\lambda}_1 = -\frac{(2-5c^2)\omega}{2c}\lambda_2 \\ \dot{\lambda}_2 = 0 \\ \dot{\lambda}_3 = 2\omega c\lambda_4 \\ \dot{\lambda}_4 = -\frac{(2-c^2)\omega}{2c}\lambda_3 \end{cases} \tag{4.30}$$

The expression of the transversality condition is the same as before:

$$\begin{aligned} & l(u(t_f),t_f) + \lambda_1(t_f)\frac{2c}{(2-c^2)\omega}u(t_f) + \lambda_2(t_f)\frac{(2-5c^2)\omega}{2c}x_m(t_f) \\ & +\lambda_3(t_f)\frac{(2-c^2)\omega}{2c}y_o(t_f) - \lambda_3(t_f)\frac{2c}{(2-c^2)\omega}u(t_f) - \lambda_4(t_f)2\omega c x_o(t_f) + \frac{\partial\Phi}{\partial t_f} = 0 \end{aligned} \tag{4.31}$$

4.2.1 Minimum Time Problem

The Hamiltonian is :

$$\begin{aligned} H = 1 &+ \lambda_1(t)\frac{2c}{(2-c^2)\omega}u(t) + \lambda_2(t)\frac{(2-5c^2)\omega}{2c}x_m(t) \\ &+ \lambda_3(t)\frac{(2-c^2)\omega}{2c}y_o(t) - \lambda_3(t)\frac{2c}{(2-c^2)\omega}u(t) - \lambda_4(t)2\omega c x_o(t) \end{aligned} \tag{4.32}$$

The transversality condition :

$$\begin{aligned} 1 &+ \lambda_1(t_f)\frac{2c}{(2-c^2)\omega}u(t_f) + \lambda_2(t_f)\frac{(2-5c^2)\omega}{2c}x_m(t_f) \\ &+ \lambda_3(t_f)\frac{(2-c^2)\omega}{2c}y_o(t_f) - \lambda_3(t_f)\frac{2c}{(2-c^2)\omega}u(t_f) - \lambda_4(t_f)2\omega c x_o(t_f) = 0 \end{aligned} \tag{4.33}$$

The control law:

$$\begin{aligned} H_u = 0 \;\rightarrow\; (-\lambda_1(t) + \lambda_3(t))\frac{2c}{(2-c^2)\omega} \;\rightarrow \\ u(t) = -sign\left((\lambda_1(t) - \lambda_3(t))\frac{2c}{(2-c^2)\omega}\right)u_{Max} \end{aligned} \tag{4.34}$$

Evidently, the control law is still a bang-bang one.

4.2.2 Minimum Energy Problem

The Hamiltonian is :

$$H = \frac{1}{2}u^2(t) + \lambda_1(t)\frac{2c}{(2-c^2)\omega}u(t) + \lambda_2(t)\frac{(2-5c^2)\omega}{2c}x_m(t) + \lambda_3(t)\frac{(2-c^2)\omega}{2c}y_o(t) - \lambda_3(t)\frac{2c}{(2-c^2)\omega}u(t) - \lambda_4(t)2\omega c x_o(t) \quad (4.35)$$

The transversality condition :

$$\frac{1}{2}u^2(t_f) + \lambda_1(t_f)\frac{2c}{(2-c^2)\omega}u(t_f) + \lambda_2(t_f)\frac{(2-5c^2)\omega}{2c}x_m(t_f) + \lambda_3(t_f)\frac{(2-c^2)\omega}{2c}y_o(t_f) - \lambda_3(t_f)\frac{2c}{(2-c^2)\omega}u(t_f) - \lambda_4(t_f)2\omega c x_o(t_f) = 0 \quad (4.36)$$

The control law:

$$H_u = 0 \;\rightarrow\; u(t) + (\lambda_1(t) - \lambda_3(t))\frac{2c}{(2-c^2)\omega} = 0$$

$$\rightarrow\; u(t) = \begin{cases} (-\lambda_1(t) + \lambda_3(t))\frac{2c}{(2-c^2)\omega} & \left|(\lambda_1(t) - \lambda_3(t))\frac{2c}{(2-c^2)\omega}\right| \leq u_{Max} \\ -sign\left((\lambda_1(t) - \lambda_3(t))\frac{2c}{(2-c^2)\omega}\right)u_{Max} & \left|(\lambda_1(t) - \lambda_3(t))\frac{2c}{(2-c^2)\omega}\right| \geq u_{Max} \end{cases} \quad (4.37)$$

As before, it is continuous.

4.2.3 Minimum Time and Energy Problem

The Hamiltonian and the control law are the same as in the minimum energy case. The transversality condition is slightly different:

$$\frac{1}{2}u^2(t_f) + \lambda_1(t_f)\frac{2c}{(2-c^2)\omega}u(t_f) + \lambda_2(t_f)\frac{(2-5c^2)\omega}{2c}x_m(t_f) + \lambda_3(t_f)\frac{(2-c^2)\omega}{2c}y_o(t_f) - \lambda_3(t_f)\frac{2c}{(2-c^2)\omega}u(t_f) - \lambda_4(t_f)2\omega c x_o(t_f) + w_f = 0 \quad (4.38)$$

4.3 Third Model : Mean-Oscillatory with Attitude Angle

This third model is the same as the previous one, but in this case the control is no more the differential drag, but the angle of the chaser with respect to

the target in the velocity direction: $u = \delta$ in figure 4.1. In this way, the real physics of the system is approached, since it is a mechanical system that, by changing the angle, modifies the differential drag, creating the control on the satellite. The expression for the differential drag with respect to the new control variable is:

$$a_{DiffDrag} = \frac{2c}{(2-c^2)\omega}|F_{dd,T}|\left[1 - \frac{C_{D,C}}{BC_T m_C}\left(A_1|cos(u)| + A_2|sin(u)|\right)\right] \tag{4.39}$$

where $|F_{dd,T}|$ is the drag acting on the target and BC_T is the ballistic coefficient of the target.

$$|F_{dd,T}| = \frac{1}{2}\rho BC_T v_r^2 \tag{4.40}$$

In this case, the control bounds could even not be imposed, because of the norm operator. This would imply the possibility for the chaser to make a complete tour, which is not a good strategy. A second reason is a pure convenient one: the control law is obtained by deriving the Hamiltonian with respect to the control variable. The derivation of the norm operator creates a numerical slowdown: to avoid it, the control is imposed to be between 0 deg and 90 deg. In this way, the norm operator can be eliminated and the derivation is simpler. But even with this trick, the control law is implicit, which creates a quite important numerical slowdown.
The system dynamics for this model are very similar to the previous one, except for the differential drag:

$$\begin{cases} \dot{x}_m = \frac{2c}{(2-c^2)\omega}|F_{dd,T}|\left[1 - \frac{C_{D,C}}{BC_T m_C}\left(A_1 cos(u) + A_2 sin(u)\right)\right] \\ \dot{y}_m = \frac{(2-5c^2)\omega}{2c}x_m \\ \dot{x}_o = \frac{(2-c^2)\omega}{2c}y_o - \frac{2c}{(2-c^2)\omega}|F_{dd,T}|\left[1 - \frac{C_{D,C}}{BC_T m_C}\left(A_1 cos(u) + A_2 sin(u)\right)\right] \\ \dot{y}_o = -2\omega c x_o \end{cases} \tag{4.41}$$

The Hamiltonian is :

$$\begin{aligned} H = l(u(t)) + \lambda_1(t)\frac{2c}{(2-c^2)\omega}|F_{dd,T}|\left[1 - \frac{C_{D,C}}{BC_T m_C}\left(A_1 cos(u(t)) + A_2 sin(u(t))\right)\right] \\ + \lambda_2(t)\frac{(2-5c^2)\omega}{2c}x_m(t) + \lambda_3(t)\frac{(2-c^2)\omega}{2c}y_o(t) - \\ \lambda_3(t)\frac{2c}{(2-c^2)\omega}|F_{dd,T}|\left[1 - \frac{C_{D,C}}{BC_T m_C}\left(A_1 cos(u(t)) + A_2 sin(u(t))\right)\right] - \lambda_4(t)2\omega c x_o(t) \end{aligned} \tag{4.42}$$

Keeping in mind that the cost function does not depend on the state variables, but only on final time and control, the costate equations are the same

as the previous case: the change in control definition affects only the control law.
The transversality condition has the same expression as before, except for the control definition:

$$l(u(t_f),t_f)+\lambda_1(t_f)\frac{2c}{(2-c^2)\omega}|F_{dd,T}|\left[1-\frac{C_{D,C}}{BC_Tm_C}\left(A_1cos(u(t_f))+A_2sin(u(t_f))\right)\right] + \lambda_2(t_f)\frac{(2-5c^2)\omega}{2c}x_m(t_f)+\lambda_3(t_f)\frac{(2-c^2)\omega}{2c}y_o(t_f) - \lambda_3(t_f)\frac{2c}{(2-c^2)\omega}|F_{dd,T}|\left[1-\frac{C_{D,C}}{BC_Tm_C}\left(A_1cos(u(t_f))+A_2sin(u(t_f))\right)\right] - \lambda_4(t_f)2\omega cx_o(t_f)+\frac{\partial\Phi}{\partial t_f}=0 \quad (4.43)$$

4.3.1 Minimum Time Problem

Keeping in mind the cost function expression for this type of problem, the Hamiltonian is simply defined as:

$$H=1+\lambda_1(t)\frac{2c}{(2-c^2)\omega}|F_{dd,T}|\left[1-\frac{C_{D,C}}{BC_Tm_C}\left(A_1cos(u(t))+A_2sin(u(t))\right)\right] + \lambda_2(t)\frac{(2-5c^2)\omega}{2c}x_m(t)+\lambda_3(t)\frac{(2-c^2)\omega}{2c}y_o(t)- \lambda_3(t)\frac{2c}{(2-c^2)\omega}|F_{dd,T}|\left[1-\frac{C_{D,C}}{BC_Tm_C}\left(A_1cos(u(t))+A_2sin(u(t))\right)\right]-\lambda_4(t)2\omega cx_o(t) \quad (4.44)$$

taking to a control law of bang-bang type:

$$H_u=0 \rightarrow (-\lambda_1(t)+\lambda_3(t))\frac{2c}{(2-c^2)\omega}|F_{dd,T}|\frac{C_{D,C}}{BC_Tm_C}(-A_1sin(u(t))-A_2cos(u(t)))=0$$
$$(-\lambda_1(t)+\lambda_3(t)\frac{2c}{(2-c^2)\omega}|F_{dd,T}|\frac{C_{D,C}}{BC_Tm_C}cos(u(t))(-A_1tan(u(t))-A_2)=0$$
$$\rightarrow u(t)=\begin{cases}0 & (\lambda_1(t)-\lambda_3(t))\leq 0\\ atan\left(\frac{A_2}{A_1}\right) & (\lambda_1(t)-\lambda_3(t))\geq 0\end{cases} \quad (4.45)$$

The transversality condition is simply the vanishing of the Hamiltonian.

4.3.2 Minimum Energy Problem

The transversality condition is :

$$\frac{1}{2}u^2(t_f)+\lambda_1(t_f)\frac{2c}{(2-c^2)\omega}|F_{dd,T}|\left[1-\frac{C_{D,C}}{BC_Tm_C}\left(A_1cos(u(t_f))+A_2sin(u(t_f))\right)\right] + \lambda_2(t_f)\frac{(2-5c^2)\omega}{2c}x_m(t_f)+\lambda_3(t_f)\frac{(2-c^2)\omega}{2c}y_o(t_f) - \lambda_3(t_f)\frac{2c}{(2-c^2)\omega}|F_{dd,T}|\left[1-\frac{C_{D,C}}{BC_Tm_C}\left(A_1cos(u(t_f))+A_2sin(u(t_f))\right)\right] - \lambda_4(t_f)2\omega cx_o(t_f)=0 \quad (4.46)$$

The presence in the cost function of the control variable makes the expression of the control law implicit:

$$H_u=0 \;\rightarrow$$
$$u(t)+(-\lambda_1(t)+\lambda_3(t))\frac{2c}{(2-c^2)\omega}|F_{dd,T}|\frac{C_{D,C}}{BC_Tm_C}(-A_1sin(u(t))-A_2cos(u(t)))=0 \quad (4.47)$$

4.3.3 Minimum Time and Energy Problem

The control law is the same as in the energy case, while the transversality condition is slightly modified by the weight coefficient of the final time:

$$\frac{1}{2}u^2(t_f)+\lambda_1(t_f)\frac{2c}{(2-c^2)\omega}|F_{dd,T}|\left[1-\frac{C_{D,C}}{BC_Tm_C}\left(A_1cos(u(t_f))+A_2sin(u(t_f))\right)\right] + \lambda_2(t_f)\frac{(2-5c^2)\omega}{2c}x_m(t_f)+\lambda_3(t_f)\frac{(2-c^2)\omega}{2c}y_o(t_f) - \lambda_3(t_f)\frac{2c}{(2-c^2)\omega}|F_{dd,T}|\left[1-\frac{C_{D,C}}{BC_Tm_C}\left(A_1cos(u(t_f))+A_2sin(u(t_f))\right)\right] - \lambda_4(t_f)2\omega cx_o(t_f)+w_f=0 \quad (4.48)$$

4.4 Fourth Model : Mean-Oscillatory with Angular Acceleration of Reaction Wheels

This fourth model is the most complex: the mechanical system interactions are considered [8]. In particular, the control is the angular acceleration of the reaction wheels in the inertial reference frame. This takes to a system composed of seven states:

- 4 classical states, mean and oscillatory components: x_m, y_m, x_o, y_o;
- angular variation of the satellite: δ;
- angular velocity of the satellite : $\dot{\delta}$;
- angular velocity of the reaction wheels : ω_w

By applying a simple equilibrium at the satellite level, the dynamics for the angular velocity of the satellite is derived:

$$I_{sat}\ddot{\delta} = M_{ext} - T_{int} = M_{ext} - I_w u \tag{4.49}$$

$$\rightarrow \ \ddot{\delta} = \frac{M_{ext} - I_w u}{I_{sat}} \tag{4.50}$$

where M_{ext} is the external torque applied on the satellite, T_{int} is the internal torque due to the reaction wheels, I_{sat} and I_w are the rotational inertia of the satellite and of the reaction wheels.
With the same reasoning, the dynamics for the angular velocity of the reaction wheels is obtained :

$$\dot{\omega}_w = \left(\frac{I_w}{I_{sat}} + 1\right) u - \frac{M_{ext}}{I_{sat}} \tag{4.51}$$

The final set of equations is:

$$\begin{cases} \dot{x}_m = \frac{2c}{(2-c^2)\omega}|F_{dd,T}| \left[1 - \frac{C_{D,C}}{BC_T m_C}\left(A_1 cos\delta + A_2 sin\delta\right)\right] \\ \dot{y}_m = \frac{(2-5c^2)\omega}{2c} x_m \\ \dot{x}_o = \frac{(2-c^2)\omega}{2c} y_o - \frac{2c}{(2-c^2)\omega}|F_{dd,T}| \left[1 - \frac{C_{D,C}}{BC_T m_C}\left(A_1 cos\delta + A_2 sin\delta\right)\right] \\ \dot{y}_o = -2\omega c x_o \\ \dot{\delta} = \dot{\delta} \\ \ddot{\delta} = \frac{M_{ext} - I_w u}{I_{sat}} \\ \dot{\omega}_w = \left(\frac{I_w}{I_{sat}} + 1\right) u - \frac{M_{ext}}{I_{sat}} \end{cases} \tag{4.52}$$

Remembering the expression of the Hamiltonian:

$$\begin{aligned} H = l(u(t)) + \lambda_1(t)\frac{2c}{(2-c^2)\omega}|F_{dd,T}|\left[1 - \frac{C_{D,C}}{BC_T m_C}\left(A_1 cos\delta(t) + A_2 sin\delta(t)\right)\right] \\ + \lambda_2(t)\frac{(2-5c^2)\omega}{2c}x_m + \lambda_3(t)\frac{(2-c^2)\omega}{2c}y_o(t) - \\ \lambda_3(t)\frac{2c}{(2-c^2)\omega}|F_{dd,T}|\left[1 - \frac{C_{D,C}}{BC_T m_C}\left(A_1 cos\delta(t) + A_2 sin\delta(t)\right)\right] - \lambda_4(t)2\omega c x_o(t) \\ + \lambda_5(t)\dot{\delta}(t) + \lambda_6(t)\frac{M_{ext} - I_w u(t)}{I_{sat}} + \lambda_7(t)\left(\frac{I_w}{I_{sat}} + 1\right)u(t) - \lambda_7(t)\frac{M_{ext}}{I_{sat}} \end{aligned} \tag{4.53}$$

The adjoint equations are:

$$\begin{cases} \dot{\lambda}_1 = -\frac{(2-5c^2)\omega}{2c}\lambda_2 \\ \dot{\lambda}_2 = 0 \\ \dot{\lambda}_3 = 2\omega c\lambda_4 \\ \dot{\lambda}_4 = -\frac{(2-c^2)\omega}{2c}\lambda_3 \\ \dot{\lambda}_5 = -\frac{2c}{(2-c^2)\omega}|F_{dd,T}|\left(\lambda_1 - \lambda_3\right)\left(-A_1 sin\delta + A_2 cos\delta\right) \\ \dot{\lambda}_6 = -\lambda_5 \\ \dot{\lambda}_7 = 0 \end{cases} \tag{4.54}$$

4.4.1 Minimum Time Problem

With the Hamiltonian independent on the control variable, the control law is the classic bang-bang:

$$\begin{aligned} H_u = 0 \ \rightarrow \ \left(-\lambda_6(t)\frac{I_w}{I_{sat}} + \lambda_7(t)\left(\frac{I_w}{I_{sat}} + 1\right)\right) \ \rightarrow \\ u(t) = -sign\left(\lambda_6(t)\frac{I_w}{I_{sat}} - \lambda_7(t)\left(\frac{I_w}{I_{sat}} + 1\right)\right)u_{Max} \end{aligned} \tag{4.55}$$

The transversality condition is simply the vanishing of the Hamiltonian.

4.4.2 Minimum Energy Problem

The Hamiltonian depends on the control variable, the control law is a continuous one:

$$H_u = 0 \;\rightarrow\; u(t) - \lambda_6(t)\frac{I_w}{I_{sat}} + \lambda_7(t)\left(\frac{I_w}{I_{sat}} + 1\right) = 0 \;\rightarrow$$

$$u(t) = \begin{cases} \lambda_6(t)\frac{I_w}{I_{sat}} - \lambda_7(t)\left(\frac{I_w}{I_{sat}} + 1\right) & |u(t)| \leq u_{Max} \\ -sign\left(-\lambda_6(t)\frac{I_w}{I_{sat}} + \lambda_7(t)\left(\frac{I_w}{I_{sat}} + 1\right)\right) u_{Max} & |u(t)| \geq u_{Max} \end{cases} \tag{4.56}$$

The transversality condition remains the simple vanishing of the Hamiltonian.

4.4.3 Minimum Time and Energy Problem

Since the Hamiltonian has no variations, the control law is the same as the previous problem. The transversality condition is the same as before, but adding the weight coefficient of the final time.

Chapter 5

Analytical solution

The decoupled nature of the equations are a valid help in case an analytic solution wants to be found. The simplest problem to solve analytically is the Minimum Time Problem, because of its bang-bang control law.

5.1 Minimum Time Problem

Considering the formulation of the optimal control problem proposed in the previous chapter, the analytical procedure is a mix of the first two models: a first manoeuvre takes the mean coordinates to the desired positions, then a second manoeuvre deals with the oscillatory ones. It is the same reasoning of Leonard [3] and Bevilacqua and Romano [4,5]. To simplify the development, some constants are defined:

$$C_1 = \frac{(2 - 5c^2)\omega}{2\omega} \tag{5.1}$$

$$C_2 = \frac{(2 - c^2)\omega}{2\omega} \tag{5.2}$$

$$C_3 = 2c\omega \tag{5.3}$$

Considering that the Scheweighart-Sedwick coefficient has a positive value, slightly bigger than 1, C_1 is a negative constant, while C_2 and C_3 are positive. The first manoeuvre takes into account only the mean coordinates: the system is the one on the previous chapter (4.15, 4.16, 4.17, 4.21, 4.22). From the transversality condition (4.21) and the fixed final condition on the states:

$$\begin{aligned} 0 =& 1 + \lambda_1(t_f)\frac{u(t_f)}{C_2} + \lambda_2(t_f)C_1 x_m(t_f) \\ 0 =& 1 + \lambda_1(t_f)\frac{u(t_f)}{C_2} \end{aligned} \tag{5.4}$$

an expression for the final costate variable can be found:

$$\lambda_1(t_f) = -\frac{C_2}{u(t_f)} \tag{5.5}$$

Since the control can assume only extreme values and never 0 or ∞, the final costate is surely not null: $\lambda_1(t_f) \neq 0$.
The dynamics of the adjoint variables is easily obtained from 4.17:

$$\begin{cases} \dot{\lambda_1} = -C_1\lambda_1 \\ \dot{\lambda_2} = 0 \end{cases} \rightarrow \begin{cases} \lambda_1(t) = \lambda_{10} - C_1\lambda_{20}t \\ \lambda_2(t) = \lambda_{20} \end{cases} \tag{5.6}$$

Since λ_1 is linear in time, it can cross the time axis only once: the control will have just one commutation.

$u = u_{Max}$

The state dynamics is:

$$\begin{cases} \dot{x_m}(t) = \frac{u(t)}{C_2} \\ \dot{y_m}(t) = C_1 x_m(t) \end{cases} \rightarrow \begin{cases} x_m(t) = x_{m0} + \frac{u_{Max}}{C_2}t \\ y_m(t) = \frac{C_1}{C_2}u_{Max}\frac{t^2}{2} + C_1 x_{m0}t + y_{m0} \end{cases} \tag{5.7}$$

from the first equation, the time variable is expressed as:

$$t = \frac{C_2(x_m - x_{m0})}{u_{Max}} \tag{5.8}$$

and the dependence between the two variables is obtained:

$$y_m(t) = y_{m0} - \frac{C_1C_2}{2u_{Max}}x_{m0}^2 + \frac{C_1C_2}{2u_{Max}}x_m^2 \tag{5.9}$$

The resulting trajectory is a decreasing parabola, because of the sign of the two constants ($C_1 < 0$, $C_2 > 0 \rightarrow C_1C_2 < 0$)(figure 5.1)

$u = -u_{Max}$

The change in sign takes to a change in the direction of the parabola:

$$\begin{cases} \dot{x_m}(t) = \frac{u(t)}{C_2} \\ \dot{y_m}(t) = C_1 x_m(t) \end{cases} \rightarrow \begin{cases} x_m(t) = x_{m0} - \frac{u_{Max}}{C_2}t \\ y_m(t) = -\frac{C_1}{C_2}u_{Max}\frac{t^2}{2} + C_1 x_{m0}t + y_{m0} \end{cases} \tag{5.10}$$

The time variable expression is opposite with respect to the previous one:

$$t = \frac{C_2(x_{m0} - x_m)}{u_{Max}} \tag{5.11}$$

and the parabola is obtained (figure 5.1)

$$y_m(t) = y_{m0} + \frac{C_1 C_2}{2u_{Max}} x_{m0}^2 - \frac{C_1 C_2}{2u_{Max}} x_m^2 \tag{5.12}$$

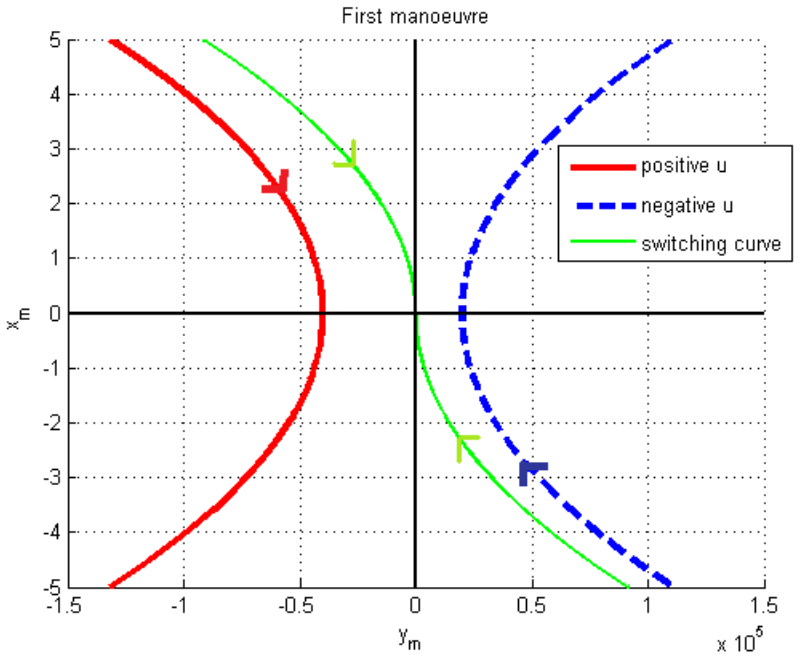


Figure 5.1: Control law for the first manoeuvre

The second manoeuvre considers only the oscillatory coordinates. The system dynamics is expressed by:

$$\begin{cases} \dot{x_o}(t) = C_2 y_o(t) - \frac{u(t)}{C_2} \\ \dot{y_m}(t) = -C_3 x_o(t) \end{cases} \tag{5.13}$$

The Hamiltonian is:

$$H = 1 + \lambda_3(t) C_2 y_o(t) - \lambda_3(t) \frac{u(t)}{C_2} - \lambda_4(t) C_3 x_o(t) \tag{5.14}$$

The control law is still a bang-bang one:

$$u(t) = sign\left(\frac{\lambda_3(t)}{C_2}\right) u_{Max} \tag{5.15}$$

The constate dynamics is slightly more difficult than before:

$$\begin{cases} \dot{\lambda}_3(t) = C_3\lambda_4(t) \\ \dot{\lambda}_4(t) = -C_2\lambda_3(t) \end{cases} \tag{5.16}$$

Taking λ_3 from the second equation, a second order differential equation is obtained for the second costate:

$$\ddot{\lambda}_4(t) + C_2C_3\lambda_4(t) = 0 \tag{5.17}$$

Since both constants are positive:

$$C_2C_3 = \alpha^2 \tag{5.18}$$

The solution for this costate is:

$$\lambda_4(t) = A\cos(\alpha t) + B\sin(\alpha t) \tag{5.19}$$

Considering the initial conditions, the final expression for the two costates is:

$$\begin{cases} \lambda_3(t) = \frac{\lambda_{40}\alpha}{C_2}\sin(\alpha t) + \lambda_{30}\cos(\alpha t) \\ \lambda_4(t) = -\frac{\lambda_{30}C_2}{\alpha}\sin(\alpha t) + \lambda_{40}\cos(\alpha t) \end{cases} \tag{5.20}$$

As for the previous manoeuvre, the control can assume only extreme values:

$u = u_{Max}$

Evidently, as for the costate variables, also the states form a second order differential equation, but, in this case, it is non homogeneous one:

$$\ddot{y_o}(t) + C_2C_3y_o - \frac{C_3}{C_2}u_{Max} = 0 \tag{5.21}$$

The solution for the homogeneous associated equation:

$$\begin{cases} y_o(t) = y_{o0}\cos(\alpha t) - \frac{C_3x_{o0}}{\alpha}\sin(\alpha t) \\ x_o(t) = x_{o0}\cos(\alpha t) + \frac{\alpha y_{o0}}{C_3}\sin(\alpha t) \end{cases} \tag{5.22}$$

The particular integral has the simple expression:

$$\bar{y}_o(t) = D \tag{5.23}$$

where D is a constant: $D = \frac{u_{Max}}{C_2^2}$. The final expression for the states is:

$$\begin{cases} y_o(t) = y_{o0}\cos(\alpha t) - \frac{C_3x_{o0}}{\alpha}\sin(\alpha t) + \frac{u_{Max}}{C_2^2} \\ x_o(t) = x_{o0}\cos(\alpha t) + \frac{\alpha y_{o0}}{C_3}\sin(\alpha t) \end{cases} \tag{5.24}$$

The resulting trajectory is circular (figure 5.2)

$u = -u_{Max}$

As for the previous case, the solution is still of circular type, with an opposite-sign centre:

$$\begin{cases} y_o(t) = y_{o0}\cos(\alpha t) - \frac{C_3 x_{o0}}{\alpha}\sin(\alpha t) - \frac{u_{Max}}{C_2^2} \\ x_o(t) = x_{o0}\cos(\alpha t) + \frac{\alpha y_{o0}}{C_3}\sin(\alpha t) \end{cases} \tag{5.25}$$

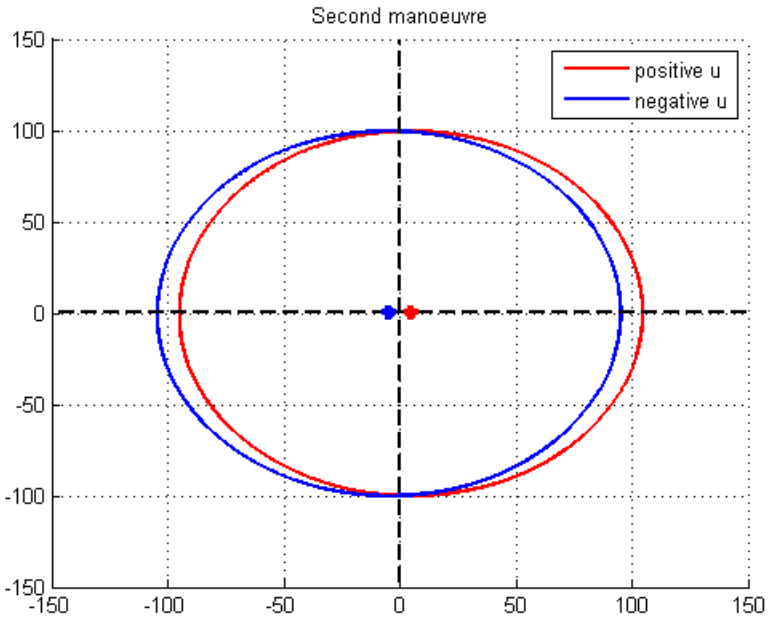


Figure 5.2: Control law for the second manoeuvre

The solution strategy is quite simple: follow the curves up to the origin, as already implemented by Leonard [3] and Bevilacqua and Romano [4,5]. It is then evident that the values of the initial conditions of the costates are not necessary.

5.2 Minimum Time and Energy Problem

Contrarily to the Minimum Time Problem, this composite problem needs the definition of the initial costates or, as normally implemented, the switching structure of the control law.
The manoeuvre strategy is still a separated one. Starting from the first couple

of coordinates, the control law is defined as in 4.28. The costate dynamics takes to the same equations for the first two costate variables. Evidently, as for the Minimum Time Problem, the λ_1 has a linear behaviour with respect to time: the control can cross the time axis only once. In particular, $\lambda_1(t)$ can be used to obtain the value of final time:

$$\lambda_1(t_f) = \lambda_{10} - \lambda_{20} C_1 t_f \;\rightarrow\; t_f = \frac{\lambda_{10} - \lambda_1(t_f)}{\lambda_{20} C_1} \tag{5.26}$$

The transversality condition takes to the definition of the final costate:

$$\lambda_1(t_f) = -\frac{C_2 w_f}{u(t_f)} - \frac{C_2}{2} u(t_f) \tag{5.27}$$

At this point, the extreme conditions on the control can be applied:

$\frac{|-\lambda_1|}{C_2} < u_{Max}$

The 5.27 together with the control law under exam takes to the definition of a limit for the final costate: after it, the control is at its maximum (or minimum).

$$\lambda_1(t_f) = -\sqrt{2 w_f} C_2 \tag{5.28}$$

The weighting factor on the terminal cost determines the slope of the control trajectory.
The control expression is:

$$u(t) = -\frac{\lambda_1(t)}{C_2} = -\frac{\lambda_{10}}{C_2} + \frac{\lambda_{20} C_1}{C_2} t \tag{5.29}$$

The state solution are simply:

$$\begin{cases} x_m(t) = x_{m0} - \frac{\lambda_{10}}{C_2^2} t + \frac{\lambda_{20} C_1}{C_2^2} \frac{t^2}{2} \\ y_m(t) = y_{m0} + x_{m0} C_1 t - \frac{C_1 \lambda_{10}}{C_2^2} \frac{t^2}{2} + \frac{\lambda_{20} C_1^2}{C_2^3} \frac{t^3}{6} \end{cases} \tag{5.30}$$

$\frac{|-\lambda_1|}{C_2} = u_{Max}$

The final costate is expressed as:

$$\lambda_1(t_f) = -\frac{w_f C_2}{u_{Max}} - \frac{C_2 u_{Max}}{2} \tag{5.31}$$

The state equations take the simple forms:

$$\begin{cases} x_m(t) = \pm \frac{u_{Max}}{C_2} t + x_{m0} \\ y_m(t) = \pm \frac{u_{Max} C_1}{C_2} \frac{t^2}{2} + x_{m0} C_1 t + y_{m0} \end{cases} \tag{5.32}$$

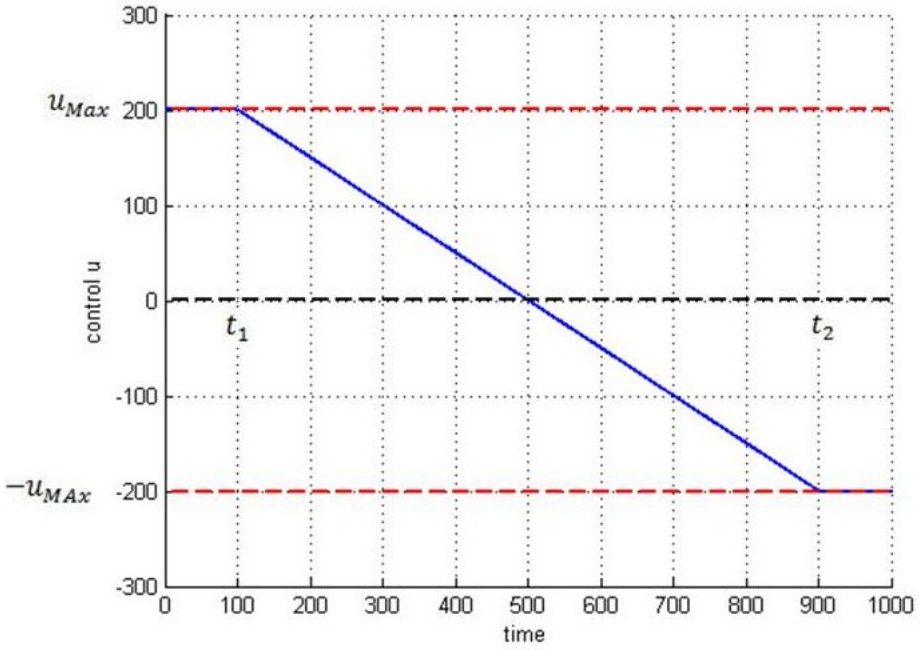


Figure 5.3: Control law for the first manoeuvre

The figure 5.3 shows the control law for this first manoeuvre. The two switching times can be used to obtain the initial costates:

$$\begin{cases} t_1 \rightarrow u = u_{Max} \rightarrow \frac{-\lambda_1}{C_2} = u_{Max} \\ t_2 \rightarrow u = -u_{Max} \rightarrow \frac{-\lambda_1}{C_2} = -u_{Max} \end{cases} \rightarrow \begin{cases} \lambda_{10} = u_{Max} C_2 \frac{t_1+t_2}{t_1-t_2} \\ \lambda_{20} = \frac{2u_{Max}C_2}{C_1(t_1-t_2)} \end{cases} \tag{5.33}$$

The second manoeuvre has the same structure of the first, with the same costate expressions of the Minimum Time Problem. The control depends on a costate that has a sinusoidal behaviour: the same behaviour is expected (figure 5.4)

Considering that this manoeuvre is successive with respect to the mean-trajectory one, the initial control value is known ($u_0 = u_{t_f}$), that allows to evaluate the initial costate λ_{30}:

$$\lambda_3(t = t_0) = \lambda_{30} = u_0 C_2 \tag{5.34}$$

The other costate can be obtained from the vanishing of the control:

$$u = 0 \ \rightarrow \ \frac{\lambda_3}{C_2} = 0 \ \rightarrow \ \frac{\alpha\lambda_{40}}{C_2}\sin(\alpha\bar{t}) + \lambda_{30}\cos(\alpha\bar{t}) = 0 \ \rightarrow$$

$$\lambda_{40} = -\frac{\lambda_{30}C_2}{\alpha tg(\alpha\bar{t})} \tag{5.35}$$

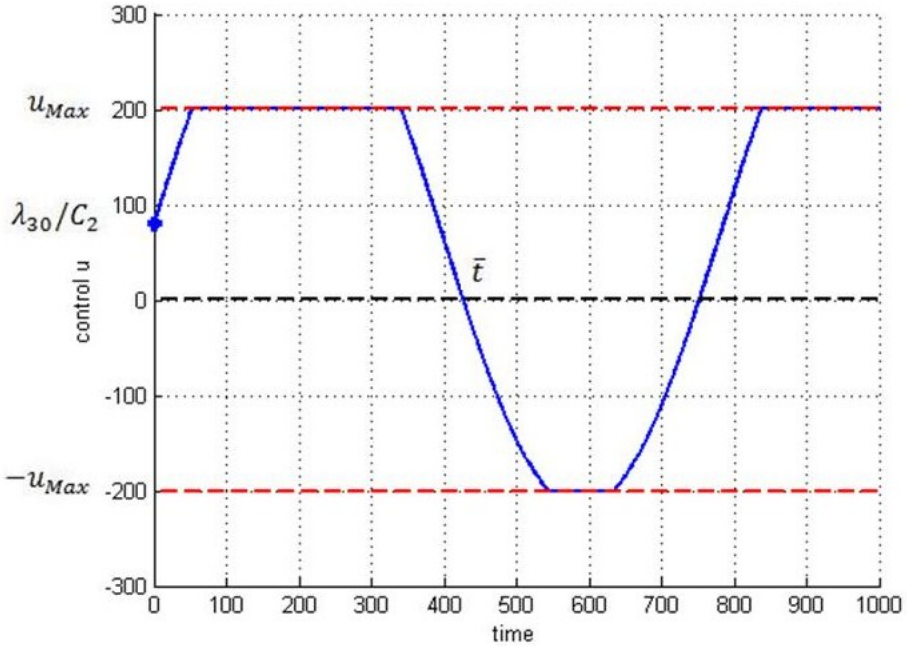


Figure 5.4: Control law for the first manoeuvre

This small demonstration shows that, in case of Minimum Time and Energy Problem, so when a continuous control law is expected, the switching structure is necessary to define the initial costates.

Chapter 6

Numerical Implementation

The simulation tests have been operated on two satellites of the $QB50$ constellation of nanosatellites [7,8]. This project has been proposed by the Von Karman Institute For Fluid Dynamics to study temporal and spatial variations of constituents of lower atmosphere and the re-entry process. In particular, the chaser is the satellite $QARMAN$, a triple-unit CubeSat, with on board a payload realized by the Université de Liège to validate the propellantless technique, as the differential drag one. The most important parameters for the simulation are shown on the tables 6.1, 6.2 , 6.3.

Altitude	350 km
Eccentricity	0
Inclination	98 deg

Table 6.1: Orbit definition

	Target	Chaser
Mass [kg]	2	4
Dimensions	-	0.1 x 0.1 x 0.3 m^3
Exposed surface	0.015 m^2	-
Drag Coefficient	2.8	2.8

Table 6.2: Characteristics of the satellites

These tables are useful to introduce an aspect: the symmetry of the control. As already explained in the previous chapters, the control is the Differential Drag, or something linked to it, as the angle δ of the chaser with respect to the target in the along-track direction - Third Model- or the angular acceleration of the reaction wheels in inertial reference frame -

Chaser Inertia	3.33 10^{-2} kg m^2
Reaction Wheels Maximum Torque	0.08 10^{-3} N m
Reaction Wheels Dynamic Operating Range	[-5500, 5500] rpm
Reaction Wheels Inertia	5.2 10^{-6} kg m^2

Table 6.3: Characteristics of the chaser

Fourth Model-. Analysing the differential drag formula:

$$a_{DiffDrag} = -\frac{1}{2}\rho C_D \left(\frac{S_C}{m_C} - \frac{S_T}{m_T}\right) v_r^2 \tag{6.1}$$

considering that the mass of the chaser is twice the one of the target:

$$a_{DiffDrag} = -\frac{1}{2}\rho \frac{C_D}{m_T} \left(\frac{S_C}{2} - S_T\right) v_r^2 \tag{6.2}$$

and the values for the exposed surfaces, with chaser maximum surface - A_2 in figure 4.1- is $0.1m\ x\ 0.3m = 0.03m^2$ and minimum one - A_1 - is $0.1m\ x\ 0.1m = 0.01m^2$; and target exposed surface of $0.01m^2$, we find that positive and negative differential drag have the same value, but opposite sign:

- Positive Differential Drag :

$$\begin{aligned} a_{DiffDrag} &= \frac{1}{2}\rho \frac{C_D}{m_T} \left(S_T - \frac{S_{C,min}}{2}\right) v_r^2 = \frac{1}{2}\rho \frac{C_D}{m_T} v_r^2 \left(0.01 - \frac{0.01}{2}\right) \\ &= Cost * (0.005) \end{aligned} \tag{6.3}$$

- Negative Differential Drag :

$$\begin{aligned} a_{DiffDrag} &= \frac{1}{2}\rho \frac{C_D}{m_T} \left(S_T - \frac{S_{C,max}}{2}\right) v_r^2 = \frac{1}{2}\rho \frac{C_D}{m_T} v_r^2 \left(0.01 - \frac{0.03}{2}\right) \\ &= Cost * (-0.005) \end{aligned} \tag{6.4}$$

The physical meaning of the differential drag, a thrust acting in opposite direction of velocity vector, explains this particularity: a positive differential drag is obtained via a minimum surface, because the ballistic coefficient of the chaser is lower than the one of the target, creating a positive thrust; while the maximum surface implies maximum drag for the chaser, so a negative differential drag, decelerating the chaser with respect to the target.

This symmetric nature of the differential drag is quite helpful in implementation, because the control bounds can simply be expressed as:

$$|u(t)| \leq u_{Max} \tag{6.5}$$

applied as:

$$\begin{cases} u(u > u_{Max}) = u_{Max} \\ u(u < -u_{Max}) = -u_{Max} \end{cases} \tag{6.6}$$

This implies, in terms of angle δ:

$$\delta = \begin{cases} 0\,\text{deg} & u = -u_{Max} \\ 90\,\text{deg} & u = u_{Max} \end{cases} \tag{6.7}$$

Actually, this is not completely true: the figure 6.1.

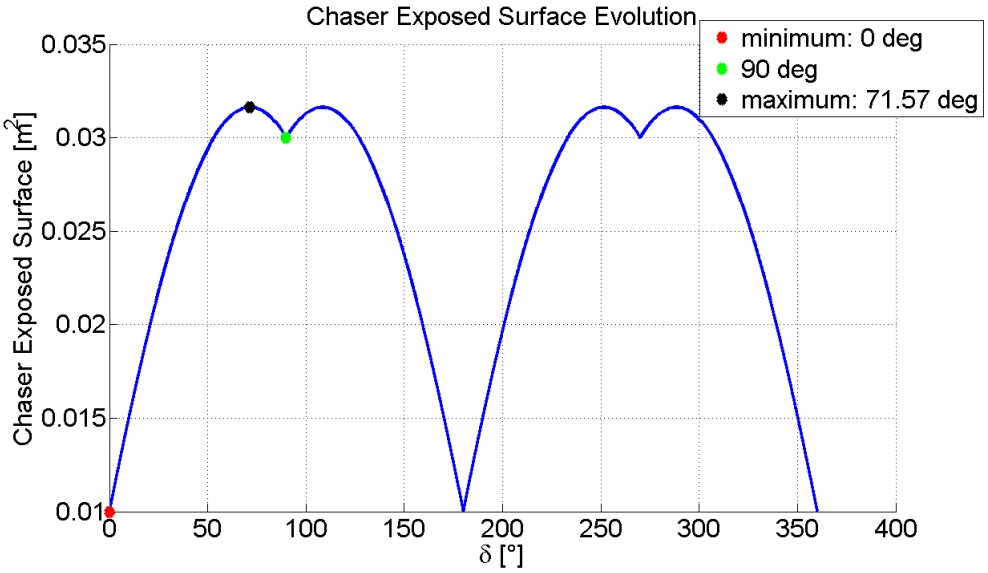


Figure 6.1: Chaser Exposed Surface Evolution

It is evident that the real maximum exposed surface for the chaser is $0.0316m^2$ obtained for an angle of 71.57 deg. This destroys the symmetry of the control, since the minimum surface is always the same. A comparison for the two control bounds is presented on the next sections, especially on the minimum time for the manoeuvre.

6.1 Choosing a Method

In Chapter 3, a compendium of optimal control solving methods have been proposed, but, as the title says, this thesis is focused on the **Indirect Methods**. In this way, the choice is limited to four strategies: each one is going to be studied for coding.

6.1.1 Gradient Methods or Backward-Sweep Method

As explained, the best implementation for this method is when the final states are not fixed and the terminal cost of the cost function does not depend on the terminal states. In this case, in fact, the terminal costates are simply null.
In the analysed problem, in any of its forms, the terminal states - the four positions - are always fixed, because of the definition of rendez-vous problem. For the last model, the last three costates are null at final time, since no constraints are imposed to angle δ, angular velocity of the satellite $\dot{\delta}$ and angular velocity of the reaction wheels ω_w. This could help, but still it is necessary a strategy to overcome the states issue. No valid strategy has been found during the year, so this method has been rejected.

6.1.2 Indirect Single Shooting Method

This method is the simplest in terms of hypothesis: the "only" problem is the initialization of the costates. No constraints are imposed elsewhere and no particular implementation seems to be used. Actually, this is not true. The big issue of this strategy is the Newton method inside it, especially in the Jacobian calculation. As already said, there exist two ways: numerical differentiation (ND) and Broyden update (BU), even with a Damped-Newton method. Both need several coefficients, as the h for the ND and the first iteration for BU, but often these strongly depend on the problem under analysis. In fact, it has to be remarked the meaning of this Jacobian: it represents the variation of the terminal values and constraints (through the transversality conditions) with respect to a variation on the initial conditions; often, with other variables that have a part on it too. The idea of using the ND as initial iteration for BU can be valid, but it needs several adjustments. First of all, the ND needs a h, that depends on the *macheps*, that can be fixed to a very small value, and on $|x|$, the norm of the variable. In this case, the variables are the initial conditions of what is unknown: costates, Lagrangian Multiplier ant final time. The ν can be forgotten for the problem under analysis, as already explained in Chapter 4, the remaining ones have

to be found. Once these values are obtained, the system dynamics has to be integrated in order to evaluate the defect function at final time: $\mathbf{F}^0$ is ready. Normally, it is not null. At this point, for each variable, the h is calculated and the Jacobian is formed: $\mathbf{J}^0$. This is the first iteration obtained via numerical differentiation, that is going to be used as initial condition for the second iteration, that implements a Broyden update. Actually, several opinions exist in this case: for some authors the best way is to keep the NU for all the code, for others Broyden is better. In this dissertation, the second idea is followed. The code is written in *Matlab*® environment and is composed of four functions: a main, with all the data, a $SSM.m$ implements the Single Shooting Method, a *propagation.m* with the system dynamics and a $defect.m$ establishes the satisfaction of the constraints. Evidently, the loop to iterate the initial conditions needs to be stopped: the stopping criterion is simply a maximum number of iterations and an error evaluation based on the norm of the defect. In particular, if the maximum number of iteration can be kept quite big, the tolerance for the norm error on the constraints should not be lowered too much. Since the most part of this error comes from the positions instead of the transversality condition for the final time, a percentage on the initial distance can be considered, or simply fix it to a value: 1.
This method seems work quite well for very simple test cases, but needs some adjustments for the more complicate ones, as the ones under analysis. For this reason, a trick has been used: finding the zero of the defect function can be seen also as finding the minimum of the norm of function. In this way, a well-known *Matlab* function can be exploited: $fminsearch$. This code counts three parts: a main with the data and the loop, a $dxfct.m$ with the system dynamics and a $defect.m$ with the integration and the defect calculations. It presents a better behaviour with respect to the previous one. However, the real big issue is still not be discussed: the costate initialization. The discussion would be faced in the next section.

6.1.3 Indirect Multiple Shooting Method

This method is very similar to the previous one, having a better convergence, since the time interval of the shooting is smaller. It has also more variables. These should be guessed in the radius of convergence of the problem: this could be even more difficult than find only the initial ones. In addition, since the $fminsearch$ (minimum of the function, not the zero) code is used, the matching conditions are not perfect, creating a segmented trajectory while implementing and a complete wrong one at the last. The reason is simple: the initial conditions for each segment can be right, but when the

integration is done on the entire time horizon with the initial conditions of the only first segment, the slightly differences in the matching conditions become important, because of the unstable nature of the dynamics, and the total trajectory is completely wrong. For this reason, this method has been abandoned.

6.1.4 Homotopic Approach

This method seems the most promising: with a simpler problem and a parameter going from 0 to 1, the original problem is obtained. Actually, in this simplicity, the difficulty is hidden: which simpler problem? As already explained, normally, this simpler problem is the minimum energy one to solve the minimum fuel one. In the case under analysis, the original problem is already the minimum energy one and with the classic BVP solver of *Matlab*, *bvp4c*, the solution for it has not been found. Several authors have successfully used this method, but for this problem, we can not. The most part of these scientists explains that a big experience is necessary to deal with this method, because many choices have to be made (simpler problem, homotopy function, parameter evolution). For this reason, after several tries, it has been rejected. However, as shown in the next section, its basic idea is kept.

6.2 Initialization

As already explained, these indirect methods are strongly dependent on costate variables and strongly unstable. The simplest way to initialize them is to guess them and try. Evidently, there are very few possibility to obtain the right ones, because of the very small radius of convergence of Newton-based methods. The best way to initialize this problem is to first understand why it is so unstable to wrong initial conditions. As already said, the Newton-method factor is very important for the convergence region, but it is not the only cause. A great source of instability is due to the bounds or the constraints on control and state variables. Fortunately, in this thesis the only control bounds are imposed. These create a discontinuity in the control trajectory, affecting the state one too.
The idea is to use a slightly modified homotopic principle. In particular, a first problem is solved: the minimum energy one without control bounds, the *unconstrained* problem. In this way, the control has a continuous trajectory, giving a very good final states. The second step is to use this information as initial guess for still a minimum energy problem, considering, this time, the control bounds. This first part has already been proposed by Bevilacqua and Romano [5], for the optimal low-thrust close proximity manoeuvres.

Actually, the homotopic principle counts, besides the link between two problems, a homotopic parameter. In this case, this parameter is applied to the control bounds, but, differently from the homotopic approach, it varies from a certain value bigger than 1 to 1: it means that the control bounds are progressively imposed. In particular, the control bounds are at first imposed with a quite big value, keeping the control strategy inside these bounds and the final states at a good position. They are progressively lowered, and the resulting optimized initial conditions for each problem become the initial guesses for the next problem.

$$u_{Max} = p\ u_{Max} \quad p = 3, ..., 1 \tag{6.8}$$

A second factor linked to the homotopic parameter is how it is lowered: in the classic homotopic approach, three strategies exist (manually, differential, simplicial), in this case, the first one is chosen. Evidently, it is not the best option, but the simplest in numerical implementation.

6.3 Minimum Time Problem

When the final time is a variable, a further problem arises in the propagation, because a fixed value is necessary for the classical IVP solver, as Runge-Kutta. To overcome this issue, a trick is used: the final time becomes a variable, whose dynamics is set to 0:

$$\mathbf{x}_{new} = \begin{bmatrix} \mathbf{x} \\ t_f \end{bmatrix} \ \rightarrow \ \dot{\mathbf{x}}_{new} = \begin{bmatrix} \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \\ 0 \end{bmatrix} = \bar{\mathbf{f}} \tag{6.9}$$

The new time interval is simply $\tau \in [0, 1]$, with $t = \tau\ t_f$ taking to a new dynamic system:

$$\frac{d\mathbf{x}}{d\tau} = \frac{d\mathbf{x}}{dt}\frac{dt}{d\tau} = \bar{\mathbf{f}} t_f \tag{6.10}$$

It is then sufficient to pre-multiply the system dynamics for the new variable.

6.4 Tests Procedure

The idea is to evaluate the accuracy of the method, considering three aspects: the initialization procedure, the variations in initial guess for costates and the variations of initial conditions in terms of state variables.
As explained in the previous chapter, the simplest problem to solve is the unconstrained Minimum Energy one, keeping the final time fixed. For this

purpose, the analytic final time is considered. Then, following the homotopic procedure, the initialization is tested: the resulting optimized initial conditions are used as initial conditions for the less simple problems, as the constrained minimum energy, the constrained minimum time and energy and the constrained minimum time. The second aspect to analyse implies a variation in guessed initial conditions for the costate variables: in this case, the idea is to evaluate how the method interacts with very bad initial conditions and the numerical cost. In particular, six cases have been considered:

	λ_1	λ_2	λ_3	λ_4
case 1	0	0	0	0
case 2	1e-7	1e-7	1e-7	1e-7
case 3	1e-3	1e-3	1e-3	1e-3
case 4	1	1e-7	1e-3	1e-7
case 5	1	1	1	1
case 6	5	5	5	5
case 7	10	10	10	10

Table 6.4: Costates cases

	x [m]	y [m]	$\dot{x}$ [m/s]	$\dot{y}$ [m/s]	x_m [m]	y_m [m]	x_o [m]	y_o [m]
case 1	0	-50	0	0	0	-50	0	0
case 2	0	-500	0	0	0	-500	0	0
case 3	0	-5000	0	0	0	-5000	0	0
case 4	0	-50000	0	0	0	-50000	0	0
case 5	5	-50	0	0	19.9725	-50	-14.9725	0
case 6	-10	-10	0	0	-39.9451	-10	29.9451	0
case 7	20	0	0	0	79.8901	0	-598901	0
case 8	20	-50000	0	0	79.8901	-50000	-59.8901	0
case 9	3	10	-0.001	0.001	13.73	11.7464	-10.73	-1.7464

Table 6.5: States cases

When the number of costate is bigger, as for the fourth model, the remaining adjoint variables are kept to 0.
Finally, the variations in state initial conditions: this is useful to evaluate the robustness of the method. In this case, 9 cases have been analysed (tables 6.5, 6.6).
Almost all the test cases have null initial velocity. For the last case, these

	Analytical time [s]
case 1	4560
case 2	14520
case 3	47940
case 4	151260
case 5	10560
case 6	17400
case 7	35460
case 8	168120
case 9	5220

Table 6.6: States cases: Analytical times

velocities are added. Despite the small values, these create a complication, that ends in a bigger computational time. For this reason, the most part of the test cases have been left without initial velocity.

6.4.1 First Model: Mean with Differential Drag

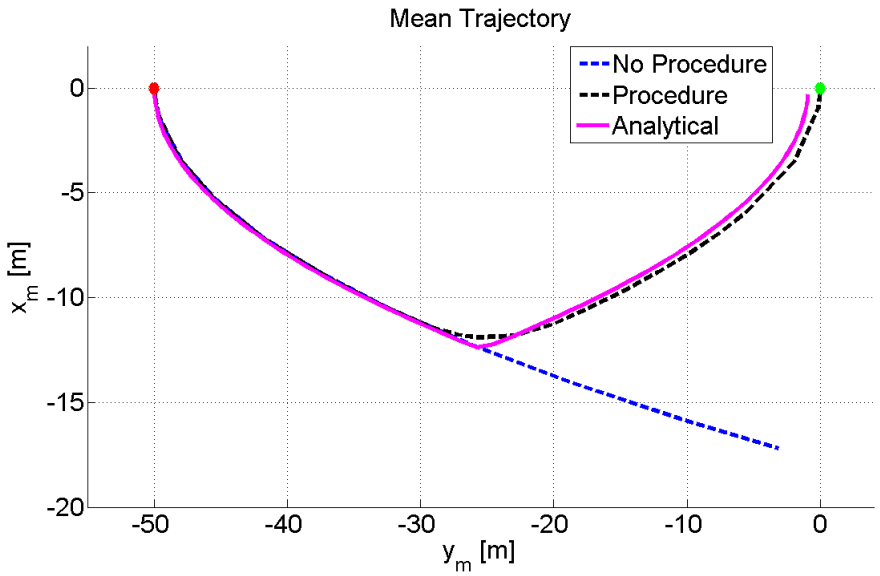


Figure 6.2: The resulting trajectory without the initialization procedure for case 1

The first aspect to analyse with this model is the validation of the **initialization procedure**. Starting with the closest test case (*case* 1), a Single

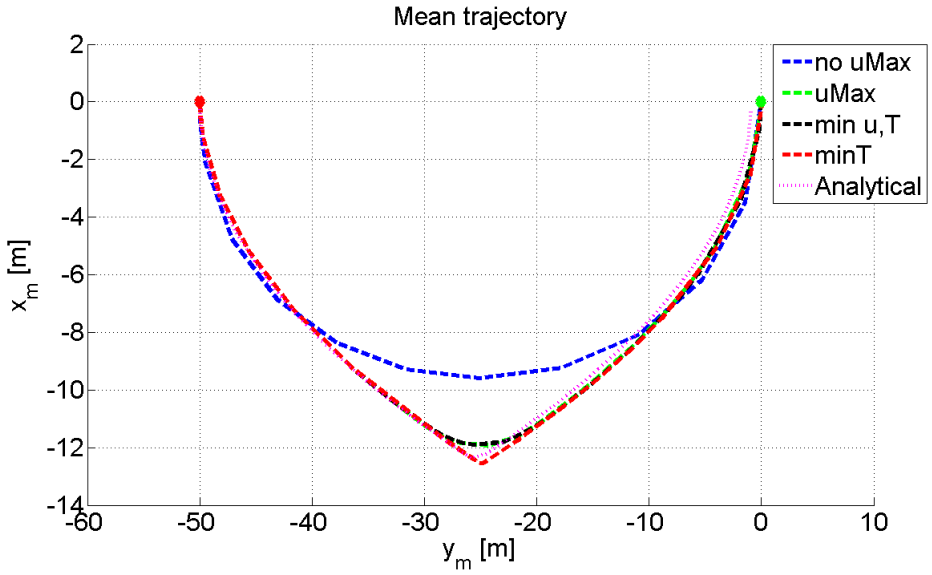


Figure 6.3: Mean trajectory for case 1

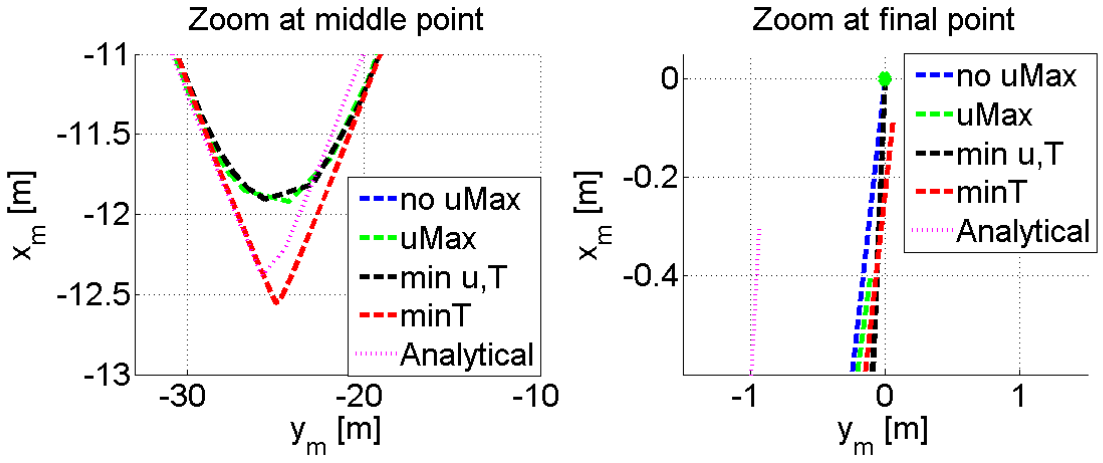


Figure 6.4: Mean trajectory for case 1: zooms

Shooting Method without initialization procedure does not work, taking to a very bad trajectory (figure 6.2, the error is 17.488). The strategy is then applied: the results are quite good (figures 6.3, 6.4), especially for smaller distances, as the table 6.8 shows. The trajectories for the various problems are ,evidently, slightly different. In particular, the absence of constraints (the blue curve in figure 6.3) allows a more "continuous" trajectory, while the control bounds make the trajectory more "pointed".

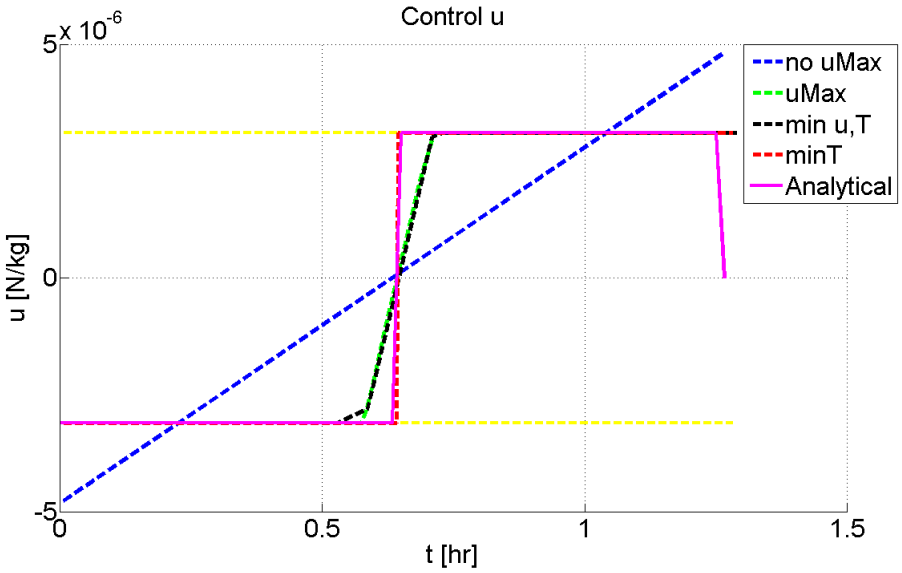


Figure 6.5: The control trajectory for case 1

The analytical solution is a time optimal one, for this reason it presents the same "pointed" behaviour of the red curve. The black one is the coupled problem (minimum time and energy) trajectory: since the initial guess for the final time is the analytical one, which is a time optimal solution, the only adding aspect with respect to the minimum energy problem is to allow a further iteration, improving the final positions.
The control trajectories follow the predicted ones (figure 6.5): the minimum energy problem has a continuous linear behaviour, the minimum time one a step behaviour. Evidently, the coupled one has both the behaviours: a "discontinuous linear" trajectory, with a higher slope. Again, the curves overlie: being the red one a minimum time problem solution, it has a step, exactly as the analytical solution; the minimum energy problems share a certain slope, lower than the previous problem, but bigger than the unconstrained one.

The figures on the state (6.3) and control (6.5) trajectory have a particular behaviour: it looks like a symmetric one. Actually, this is the result of the initial conditions on the state variable: this case has a not null initial condition only on the y_m. The same for the cases 2,3,4. The case 7 has also only one initial condition, but on x_m. As the figures 6.6 and 6.7 show, this case has not a symmetric behaviour. The difference between these two solution shapes is in the different initial conditions, linked to the system dynamics.

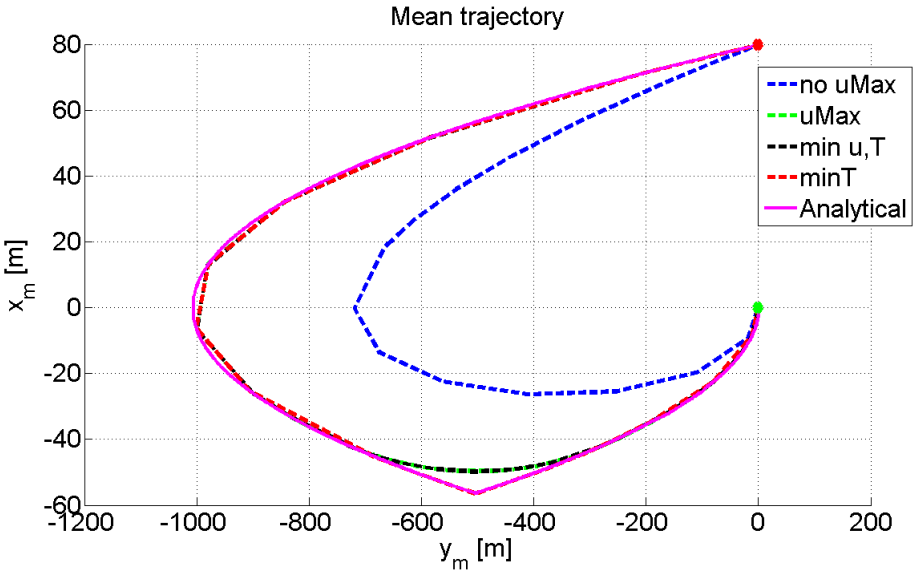


Figure 6.6: The resulting trajectory with the initialization procedure for case 7

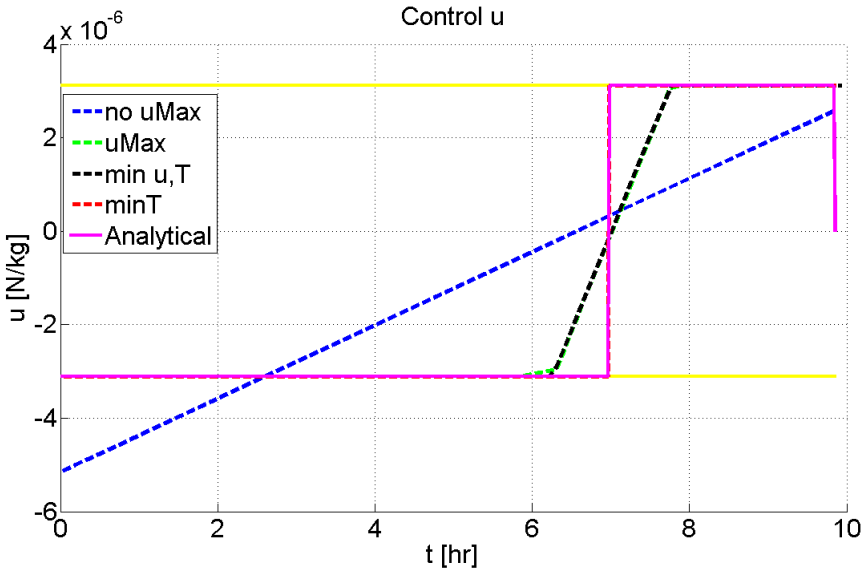


Figure 6.7: The control trajectory for case 7

As illustrated in Chapter 5, the analytical solution (a time optimal solution) implies a series of switches between curves to arrive on the switching curve, that takes directly to the origin (figure 5.1). Evidently, because of the shape

of these curves (y_m depends quadratically on x_m: they form a parabola) and the obliged movement (they can move only on these parabolas), having a zero x_m creates a perfect "pointed" parabola, while a problem with initial x_m not null obliges the satellite to follow the first curve slightly more than the switching curve, vanishing the symmetry. The figure 6.8 illustrates the minimum time trajectory for case 2 [1] and case 7: the switching curve can be guessed when the two curves overlie and the difference in the path is evident.

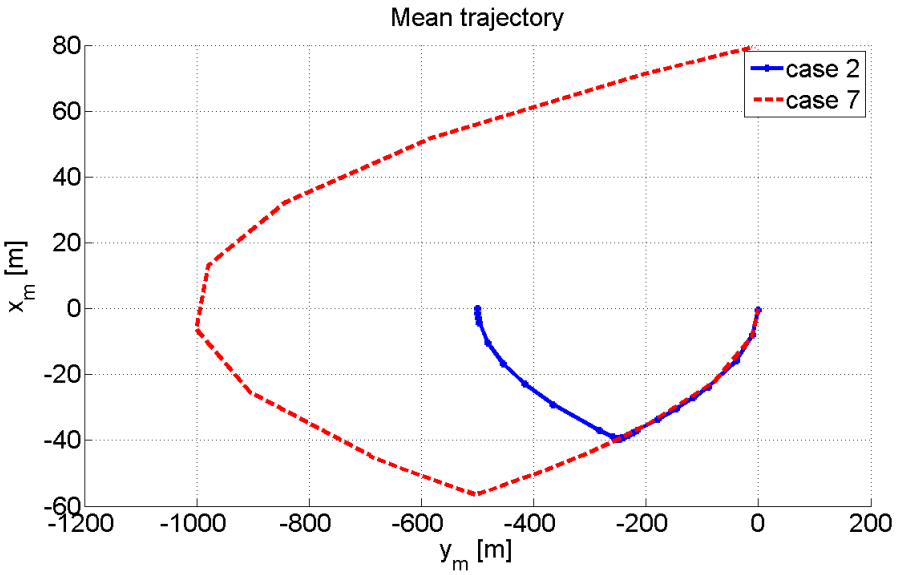


Figure 6.8: The difference between case 2 and case 7

The last variables to analyse are the costates. The figures 6.9 illustrate the evolution of these parameters for case 8: λ_1 has a linear behaviour, while λ_2 is continuous, as expected from the analytical solution.

The blue curve (the unconstrained problem) has a quite different slope for the first costate and a different value for the second. This depends on the optimized initial conditions that the method finds. In fact, the constrained solutions have almost the same initial conditions, while for the unconstrained one they are different (table 6.7). The first problem allows a first optimization, that is successively improved by the constrained problems. As for the trajectories, these problems show similar behaviours [2].

The next aspect to analyse is the error. The table 6.8 has some red values: the errors are bigger than the tolerance (1). Firstly, the error of

[1] This case has been chosen over case 1 only for graphic purpose, since case 1 has too

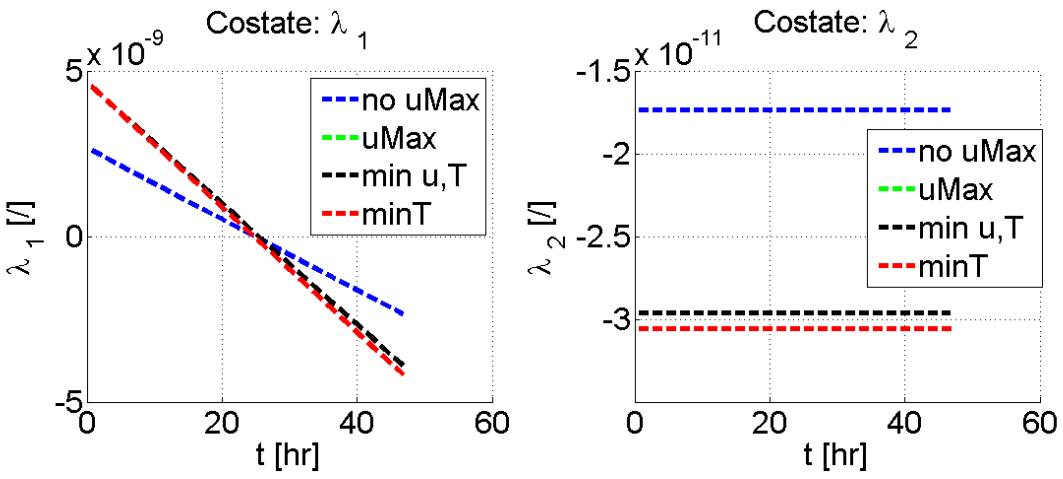


Figure 6.9: The costate evolution for case 8

	unconstrained min energy		constrained min energy		constrained min energy and time		constrained min time	
	λ_1	λ_2	λ_1	λ_2	λ_1	λ_2	λ_1	λ_2
case 1	2.7588e-9	-7.057e-10	1.7746e-8	-4.4702e-9	1.7471e-8	-4.3832e-9	1.7529e.8	-4.4142e-9
case 2	2.7209e-9	-2.1858e-10	1.6658e-8	-1.3217e-9	1.6694e-8	-1.3256e-9	1.6725e-8	-1.3342e-9
case 3	2.4979e-9	-6.0777e-11	4.0083e-9	-9.7492e-11	4.0122e-9	-9.7592e-11	3.9789e-9	-1.0022e-10
case 4	2.5076e-9	-1.9337e-11	4.0885e-9	-3.1487e-11	4.0908e-9	-3.1505e-11	4.0623e-9	-3.2233e-11
case 5	2.9957e-9	-2.6237e-10	2.3912e-8	-1.9451e-9	2.3992e-8	-1.9514e-9	2.4082e-8	-1.9620e-9
case 6	-2.9736e-9	1.4888e-10	-5.3164e-8	2.4889e-9	-5.3310e-8	2.4959e-9	-5.3468e-8	2.5051e-9
case 7	2.9546e-9	-7.2894e-11	1.6681e-8	-3.8506e-10	1.6694e-8	-3.8590e-10	1.6723e-8	-3.8861e-10
case 8	2.6599e-9	-1.7372e-11	4.6421e-9	-2.9614e-11	4.6423e-9	-2.9615e-11	4.6553e-9	-3.0581e-11
case 9	2.9551e-9	-4.6758e-10	1.2822e-7	-1.9124e-8	1.2889e-7	-1.9190e-8	1.2974e-7	-1.9310e-8

Table 6.7: The optimized costates for variations in state initial conditions

small values with respect to case 7.

[2]The green curve on the figures almost coincide with the black one, as for the trajectories.

	unconstrained min energy	constrained min energy	constrained min energy and time	constrained min time
case 1	1.3222e-4	0.4214	9.9584e-5	0.1032
case 2	3.1957e-4	0.8664	9.8742e-5	0.3950
case 3	1.7895e-4	0.0475	9.9188e-5	3.4084
case 4	1.1295e-4	0.8568	9.7370e-5	41.2523
case 5	1.1671e-4	0.6012	9.9373e-5	0.0209
case 6	1.1632e-4	0.8764	0.0356	0.0356
case 7	1.8546e-4	1.7074	9.8822e-5	0.1085
case 8	1.2793e-4	6.3356e-5	9.7257e-5	31.0266
case 9	1.0674e-4	0.4492	9.8707e-5	0.0624

Table 6.8: Error for variations in state initial conditions

the constrained minimum energy problem for case 7 has been analysed. The cause is the optimization parameter, whose evolution creates some issues. As explained in the previous chapter, the chosen strategy is the simplest one: manually lower this parameter. It is clear that it is not the best option, taking to a very long simulation, because the step size for the control bounds has to be reduced when the initial conditions on the states are bigger: already from the second test cases, the step sized is reduced to 0.005 (table 6.9). However, this strategy has been kept, because of a lack of information about the other two methods. The reason for the bigger error for the constrained minimum

	initial value	step size
case 1	1.7	0.01
case 2	1.7	0.005
case 3	1.7	0.005
case 4	1.7	0.005
case 5	1.7	0.005
case 6	1.7 / 1.01	0.002 / 0.001
case 7	1.7 / 1.008	0.002 / 0.00001
case 8	2.3	0.001
case 9	1.7	0.001

Table 6.9: Step size for the constrained min energy problem for variations in state initial conditions

energy problem for case 7 is then in table 6.9: a more reduced step size could help, but the computational cost was already quite big (2575.3 sec $\approx$ 42 min, with respect to the lower 27 min for case 8 and 7 for case 9).

A first cause for this necessity of lowering the step size can be found in the initial distance: the more the satellite is far away, the longer is the rendez-vous time, the bigger are the instabilities, due to the integration. As explained, these equations are particularly unstable: when the initial distance is too big, these instabilities sum themselves, taking to a less good final position. A second, probably more important, cause is another one: the initial costates. Even for this simple model, the variation in guessed initial adjoints variables creates a variation in final positions, as it will be illustrated in the next section.
The second group of red numbers is in the minimum time problem column: up to the minimum time and energy problem, the final positions are good; when its optimized initial conditions are imposed as guessed initial conditions for the minimum time problem, the error increases (table 6.8). The reason could be found in the bang-bang control law, that adds instabilities. These propagate with unstable equations, taking to a bad trajectory (figures 6.10).

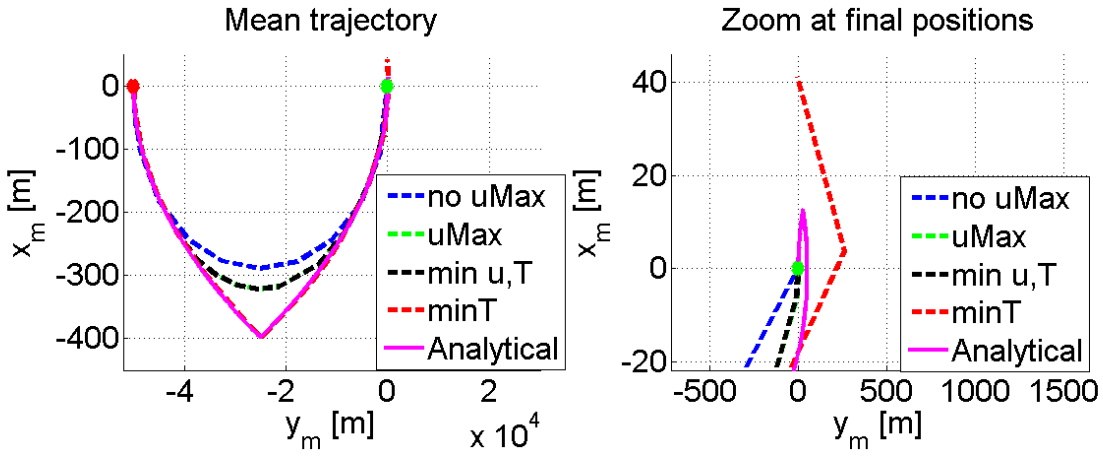


Figure 6.10: Mean trajectory for case 4

A second, *reduced*, initialization procedure has been tried: the resulting optimized initial costate variables from the unconstrained minimum energy problem are implemented in the constrained minimum time problem as initial guesses. The results are very similar to the global procedure, with a general bigger error. In some cases (1, 2, 4, 5), the errors are the same; while only in one case, the last one, the error is slightly lower. Actually, this last case represents the closest position of the chaser with respect to the target: probably, the evolution of the costates from the "homotopic" approach is not the optimal one.

This introduces the second aspect to analyse: the **costates variations**.

	Error
case 1	0.1032
case 2	0.3950
case 3	12.9686
case 4	41.2523
case 5	0.0209
case 6	0.7045
case 7	0.4157
case 8	49.1382
case 9	0.0496

Table 6.10: Error for the *reduced* initialization procedure

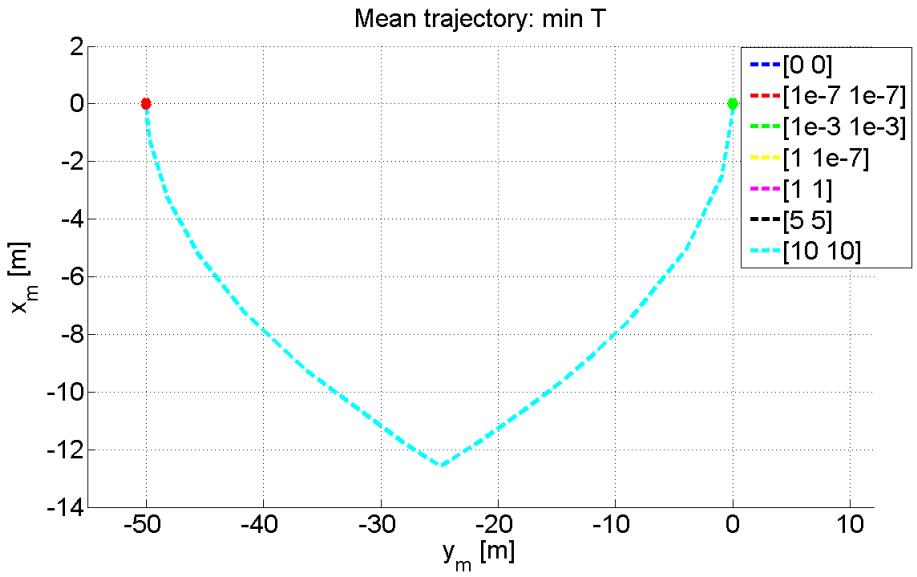


Figure 6.11: Mean trajectory for case 1

As already said, six cases have been studied. In order to reduce the calculation time, the *reduced* initialization procedure have been implemented. Actually, the first eight cases show only very small differences in final error, of the order of magnitude of $10^{-15} - 10^{-16}$, that can be easily caused by numerical calculations, than real differences, especially because the trajectories coincide (figure 6.11).

The only case showing a slightly bigger difference is case 9, still with val-

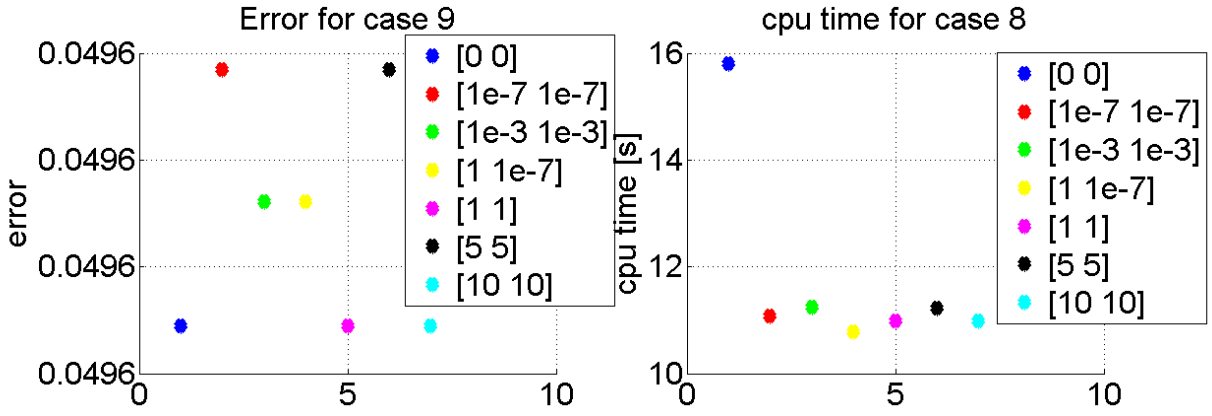


Figure 6.12: Variation of the initial conditions on the costate variables

ues of order of magnitude of 10^{-8}. This can explain the different behaviour with respect to other cases when the *reduced* initialization procedure is implemented: the first guess [0 0] is the best one (figure 6.12(a), with a difference with respect to the other two good conditions of the order of magnitude of 10^{-16}).
Actually, the most important difference in changing the initial costates is in the computational cost: the figure 6.12(b) shows for the case 8, the different computational times. This case has the biggest difference, still keeping it under 5 seconds, while the other cases have differences lower than 1 second. The simple results can be explained by the simplicity of the system: the mean coordinates have a quite simple dynamics, with quite simple control bounds and initial and final conditions. Moreover, the presence of only two variables adds further simplifications to the problem. The next Model will introduce complications.

The final aspect to study with respect to this first Model is the **symmetry of the control**. As explained previously in this chapter, since the differential drag is linked to the exposed surface of the chaser, a slightly bigger exposed surface ($0.0316m^2$ for an angle $\delta = 71.57$) exists, vanishing the symmetry. The difference between these two control bounds is actually more pronounced for the constrained minimum energy problem and the minimum time one (figures 6.13, 6.14). Evidently, the imposition of different control level has an impact on those problems that insert more issues (discontinuities, bounds). The figure 6.14(b) shows something expected: the final optimal time for the non-symmetric control bounds is lower than the symmetric case. Since the orbits are the same, so same energy, less time is sufficient when the control is bigger.

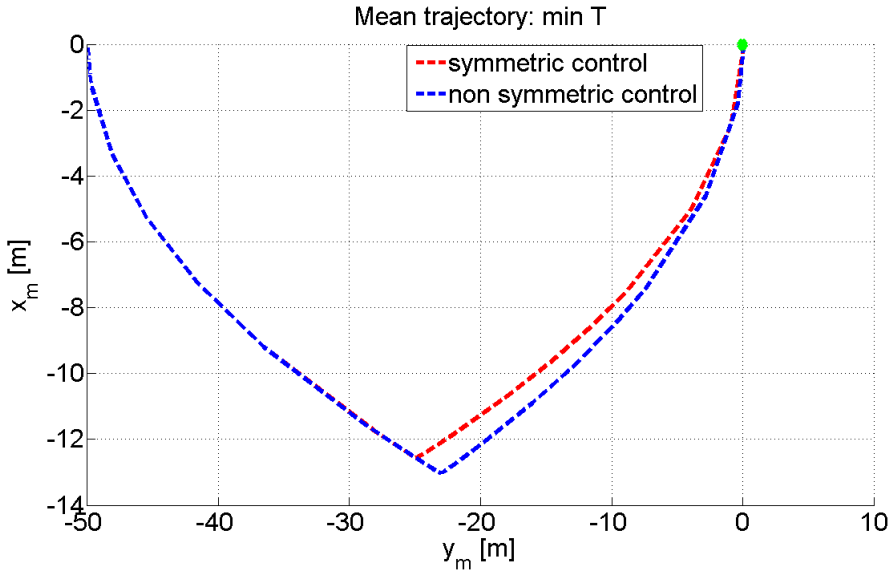


Figure 6.13: Mean trajectory for case 1

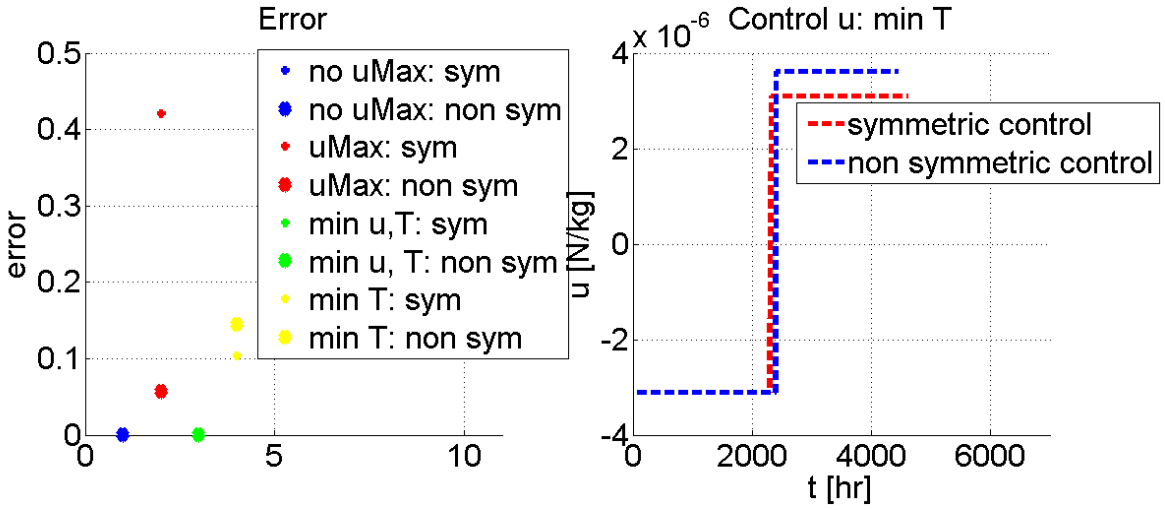


Figure 6.14: Error and Control for case 1

6.4.2 Second Model: Mean-Oscillatory with Differential Drag

This second model introduces the oscillatory coordinates, increasing the number of variables and the complexity of the calculations. In fact, this second group of variables has, as the name says, an oscillatory behaviour, that is more difficult to deal with.

	Factor
case 1	1.4
case 2	1.08
case 3	1.08
case 4	1.25
case 5	1.3
case 6	1.2
case 7	1.3
case 8	1.2
case 9	1

Table 6.11: Factor for the time interval definition

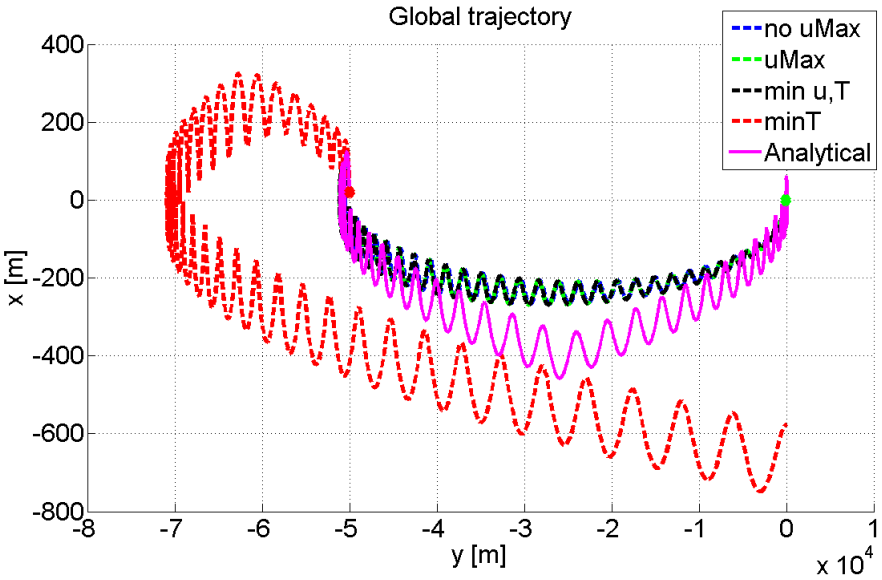


Figure 6.15: Difference in global trajectory with the analytical solution for case 8

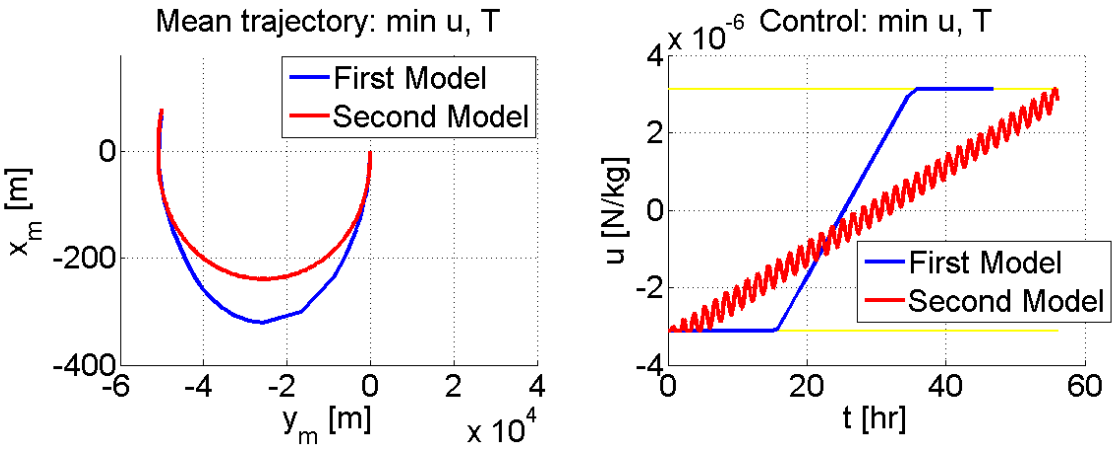


Figure 6.16: Differences between First and Second Model for case 8

The first complication is the time interval. The analytical solution is a two-phase manoeuvre, while this model takes to a global solution: this difference is quite important, because the control law is deeply different. Because of this, the time interval has been chosen differently than the previous case: it has been obtained by pre-multiplying the first-model optimal time by a factor (table 6.11). The choice of the factor depends on the final error. Differently from before, this final error is not the minimum time problem one, but the previous minimum energy and time one. The reason is that, for the first problem, it seems that none of the factors was good. For this reason, also the *reduced* initialization procedure is implemented with the minimum energy and time problem.
As explained, the control law is a global one, not divided into two manoeuvres. This implies a different trajectory with respect to the analytical solution (figure 6.15) and also a different mean trajectory with respect to the previous Model (figure 6.16(a)).

The figure 6.16(b) illustrates an important element: the completely different control law for the two Models. As shown in Chapter 5, the analytical solution for the control law of the oscillatory coordinates has a sinusoidal shape, that can be easily seen in this figure. The case 9 has no change in time interval with respect to the First Model, but the consequences are the same (figures 6.17, 6.18). It can be remarked that, for case 9, even with the same initial time interval, the minimum time and energy problem gives different optimized final time (figure 6.18(b)). Evidently, being the Models

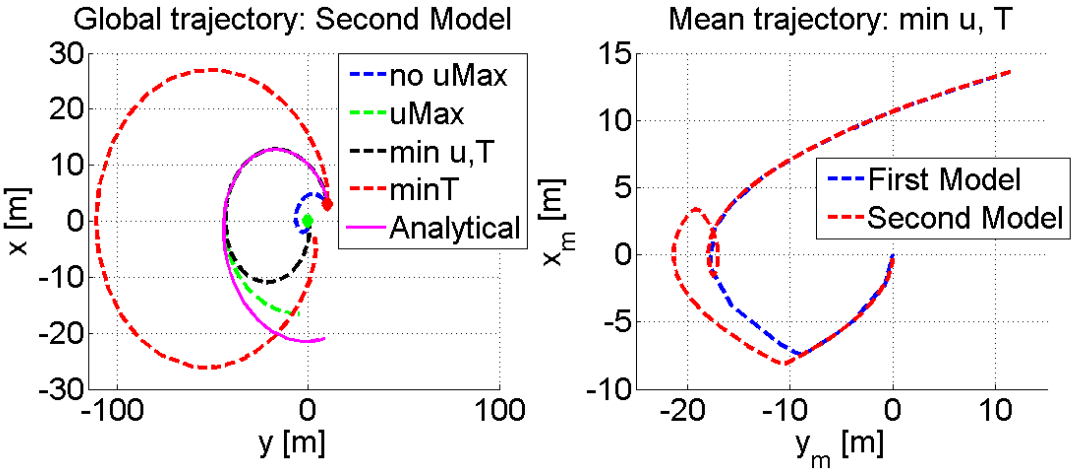


Figure 6.17: Differences between First and Second Model for case 9: trajectories

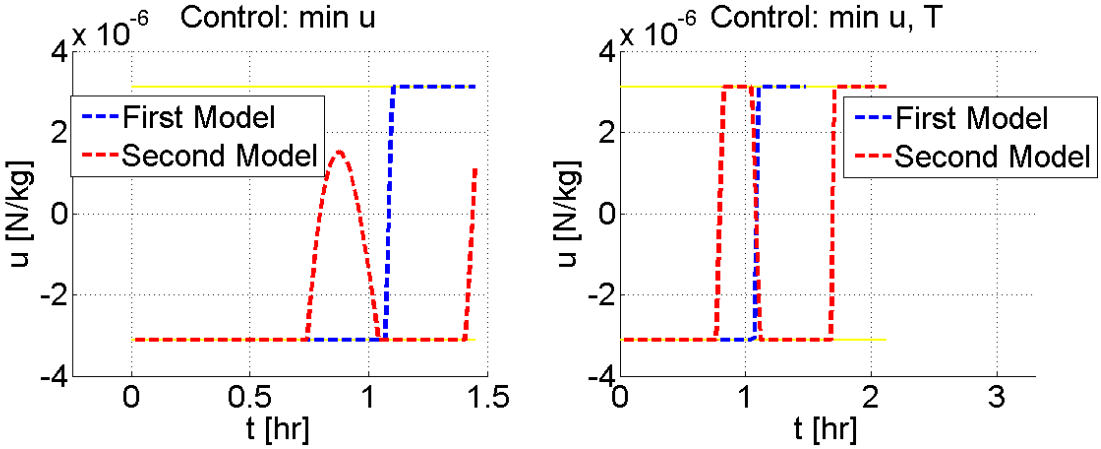


Figure 6.18: Differences between First and Second Model for case 9: control

so different, it was expected. The figures 6.15 and 6.17(a) show what has been already announced: the minimum time solution is completely wrong. It is an unexpected complication: the probable cause is in the bang-bang control law of the minimum time problem. If the linear control law of the First Model can be represented in a step shape, the same thing can not be said for the Second Model, whose control law is more complicated, being a combination of a linear one and a sinusoidal one.

As for the previous Model, the next element to analyse is the costate evolution. As illustrate in the previous chapters, the first two adjoint variables

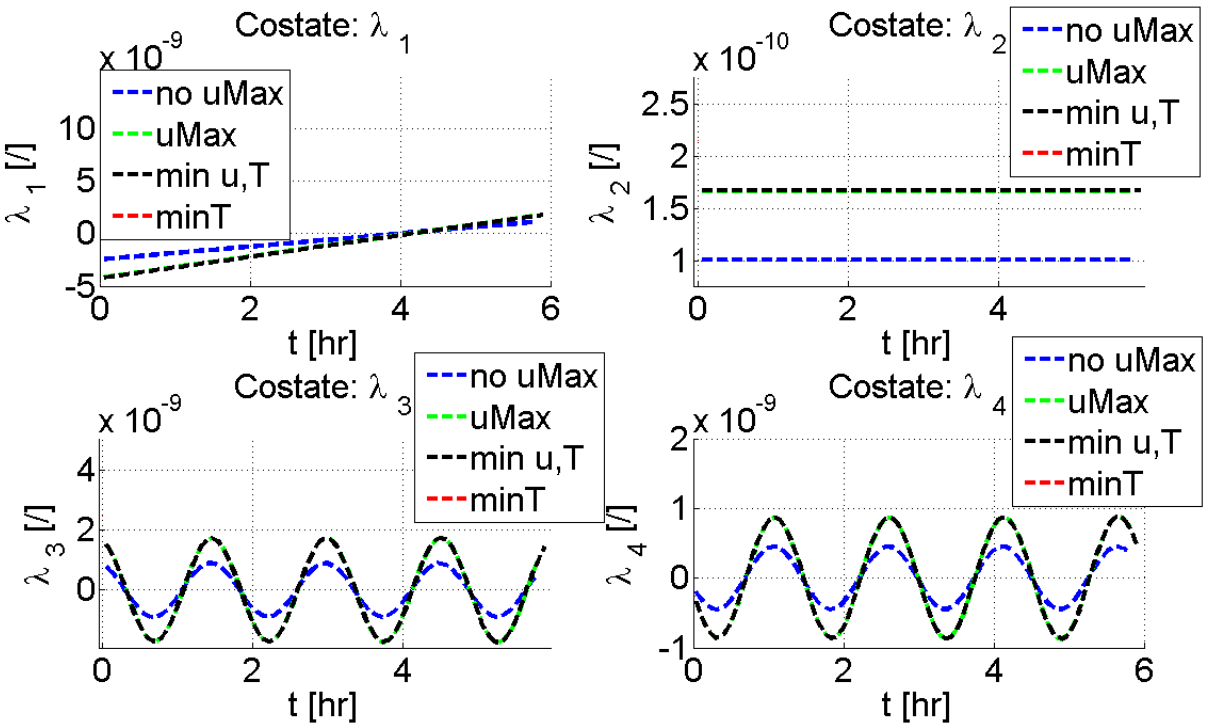


Figure 6.19: Costate evolution for case 6

keep the same shape with respect to the First Model, while the last two show a sinusoidal behaviour (figures 6.19).
From these figures, it seems that a colour is missing: the red curve is not present. Actually, this is not true (figures 6.20(b), 6.21): for the case 6, the minimum time problem converges to a very small trajectory, because the optimized final time is 37.0698 sec, with respect to the 21216 sec of the minimum time and energy problem.

As for the previous Model, the next element to consider is the error, to evaluate the precision of the method (table 6.12).
Evidently, the column of the minimum time problem is completely red. The red value on the second column can be explained with the "homotopic" optimization parameter evolution, that could be not the best one (table 6.13). For this Model, this evolution has been kept as simple as possible: because of the complexity of the system dynamics, the computational cost increases a lot with respect to the First Model.

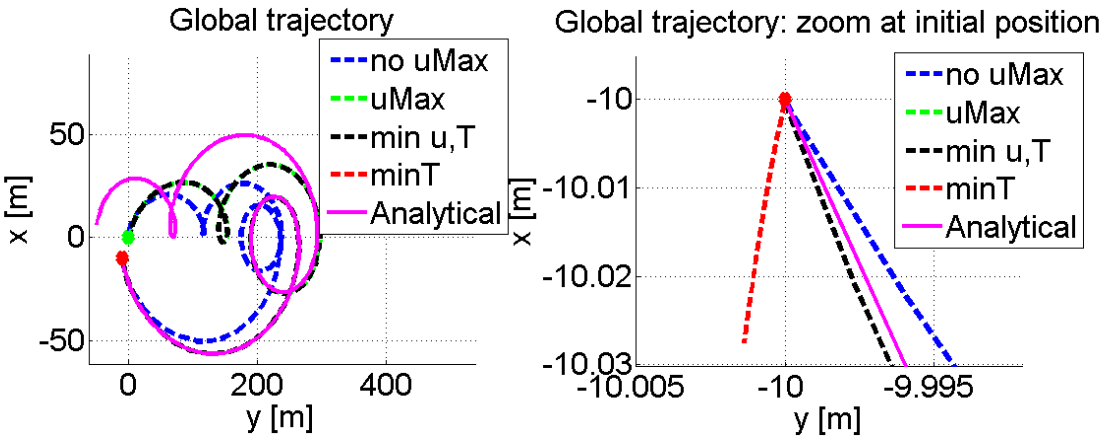


Figure 6.20: Global trajectory for case 6

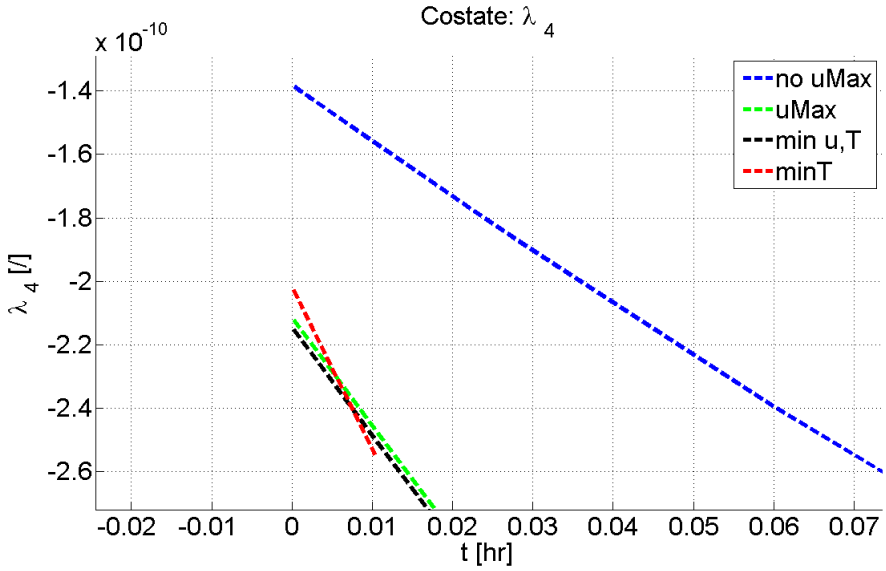


Figure 6.21: Costate evolution for case 6: $\lambda_4(t)$, zoom at initial position

The *reduced* initialization procedure is implemented with the minimum time and energy problem: the results are very similar to the global procedure, except for cases 6 and 7(table 6.14). The cause, as in the previous model, can be found in the costate initial conditions: the case 6 has not in [0 0 0 0] its best initial adjoint variables (figure 6.23(a)).
Differently from the First Model, the costate initial condition variation modifies not only the computational cost (figure 6.23(b), but also the trajectory (6.22). The reason is in the already mentioned extreme sensitivity of the

	unconstrained min energy	constrained min energy	constrained min energy and time	constrained min time
case 1	1.2859e-4	9.9701e-5	9.9942e-5	22.0132
case 2	1.5918e-4	0.1592	0.059	63.7670
case 3	8.8813e-5	0.1076	0.0354	196.8469
case 4	2.2056e-4	1.1322e-4	0.0057	653.6407
case 5	2.1554e-44	0.0602	0.0196	17.0568
case 6	1.0699e-4	1.3218	0.9987	50.8030
case 7	1.5744e-4	1.3941	0.5468	99.7579
case 8	3.7481e-4	0.9046	0.5203	675.9286
case 9	1.3423e-4	12.4547	0.0421	26.8215

Table 6.12: Error for variations in state initial conditions

	initial value	step size	increment in computational cost [s]
case 1	1.5	0.001	1102
case 2	1.5	0.001	1953
case 3	1.5	0.001	6586
case 4	1.5	0.01	1371
case 5	1.4	0.001	1638
case 6	1.5	0.001	4463
case 7	1.5	0.001	6135
case 8	1.5	0.001	16067
case 9	2.8	0.005	1476

Table 6.13: Step size for the constrained min energy problem for variations in state initial conditions

method with respect to the initial conditions, augmented by the complexity of the system dynamics.

Finally, the symmetry of the control law. The figure 6.24(a) shows a smaller error with respect to the previous Model. Actually, this behaviour is not common to all the cases: the case 6, for example, has a bigger error (figure 6.24(b)).

	Error
case 1	0.1244
case 2	0.0988
case 3	0.2514
case 4	6.411e-4
case 5	0.0466
case 6	2.5089
case 7	4.6655
case 8	0.2455
case 9	0.0162

Table 6.14: Error for the *reduced* initialization procedure

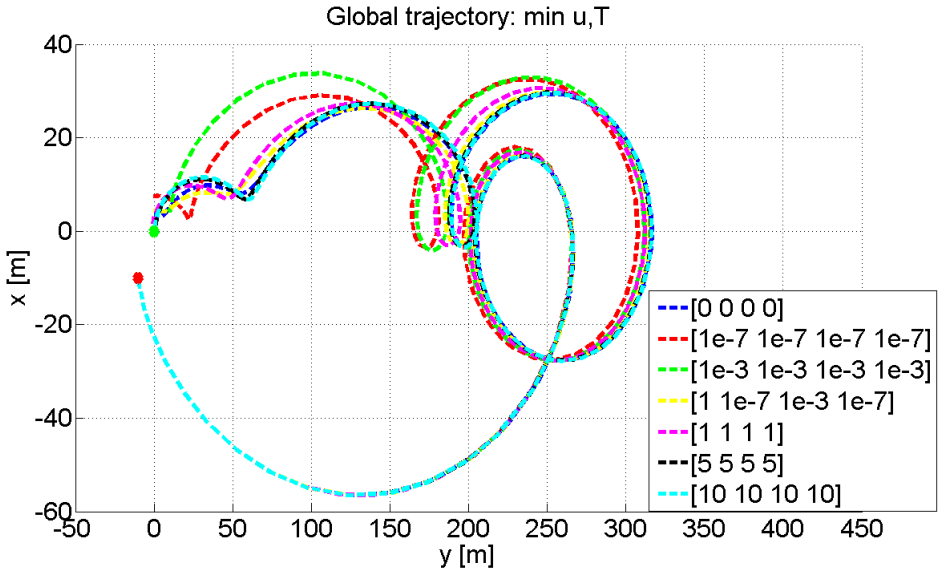


Figure 6.22: Costate evolution for case 6: minimum time and energy problem, global trajectory

6.4.3 Third Model : Mean-Oscillatory with Attitude Angle

This Model, together with the Fourth one, has given much more problems. Only one case has been analysed, the simplest case (case 1), but still no good solution has been found. Several strategies have been used, modifying the possible parameters: homotopic optimization parameter, time interval, initial costate from the Second Model.

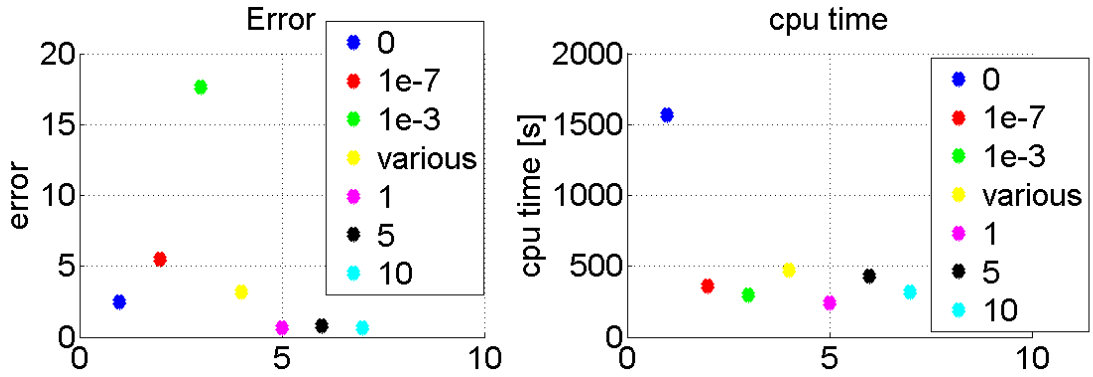


Figure 6.23: Costate evolution for case 6: minimum time and energy problem, error and computational time

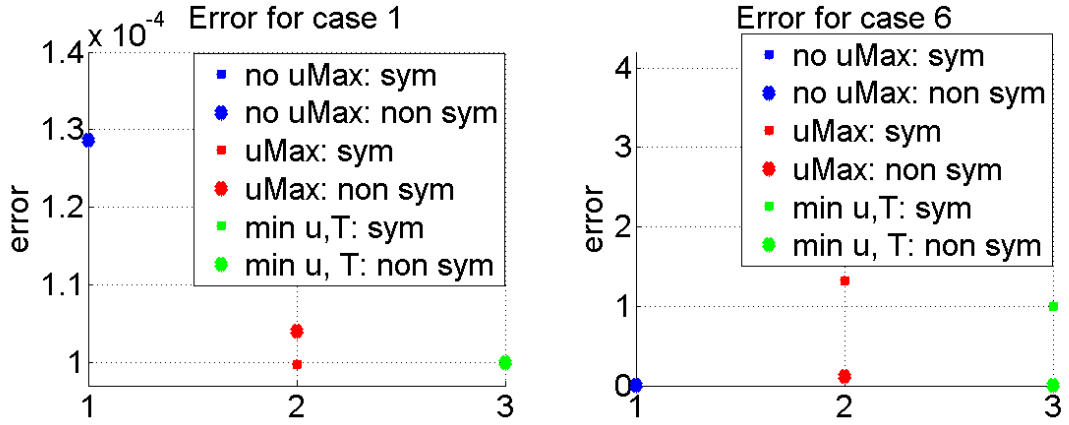


Figure 6.24: Differences in symmetric and non-symmetric control for case 1 and 6

The figures 6.25, 6.26(a), 6.26(b) show a comparison in the trajectories (global, mean and oscillatory) between this Model, the previous one and the analytical solution. Actually, the curves are only the ones related to the first two types of problem (unconstrained minimum energy and constrained minimum energy), because the problem related to the minimization of the time gives wrong solution: the minimum time problem has an optimized final time of the order of 10^{-13}! The cause can be found in the combination of bang-bang solution and the implicit control law, typical of this Model.
The minimization of the energy seems work well: the constrained one is very similar to the solution of the Second Model. As for the previous Model, the

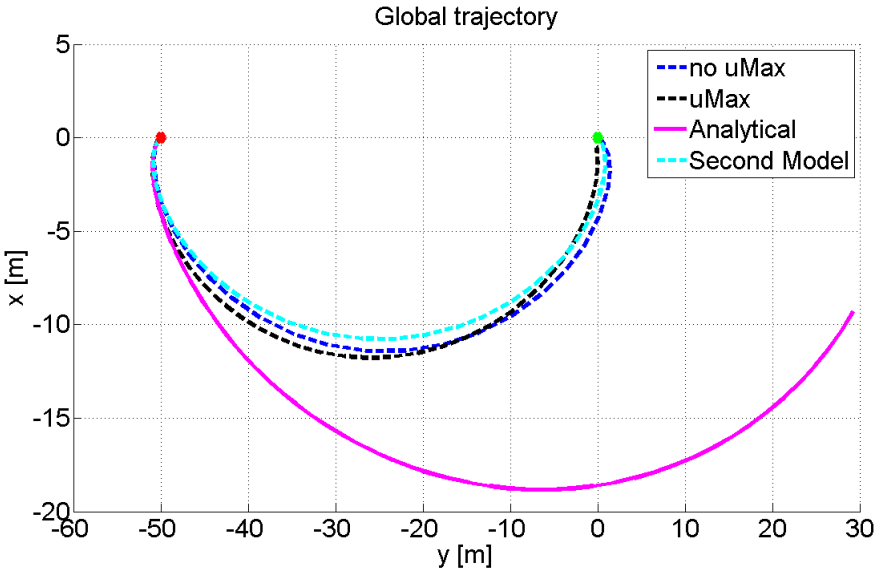


Figure 6.25: Global trajectory

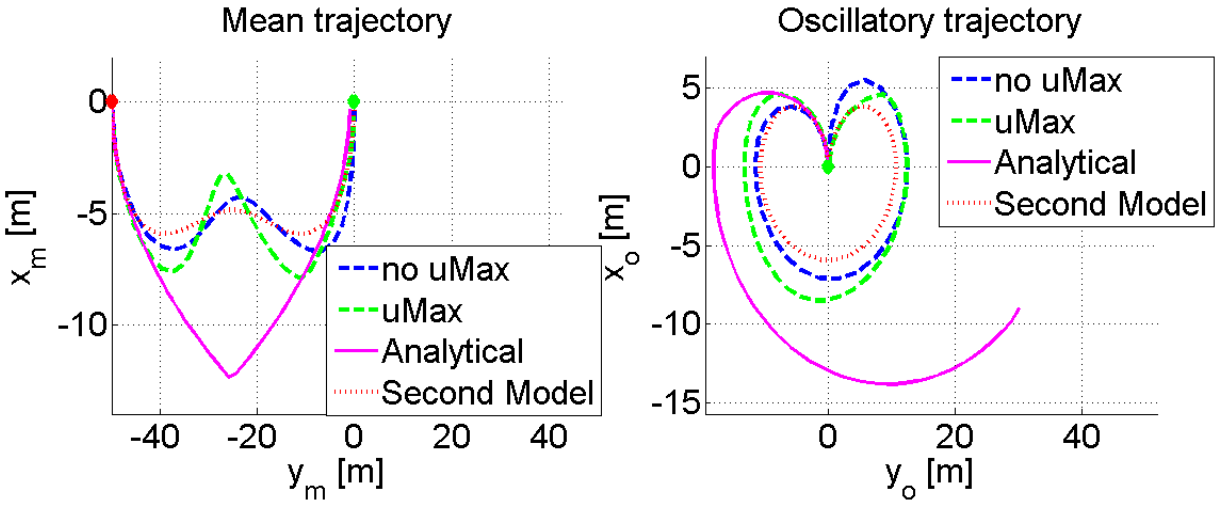


Figure 6.26: Mean and oscillatory trajectories

difference with the analytical solution is in the two-phase manoeuvre that characterizes this solution, while the method looks for a global manoeuvre, combining the two movements.

As announced, the control bounds are, in this case, angle values: 0 deg and 90 deg, because of the norm operator that creates some issues in the implicit

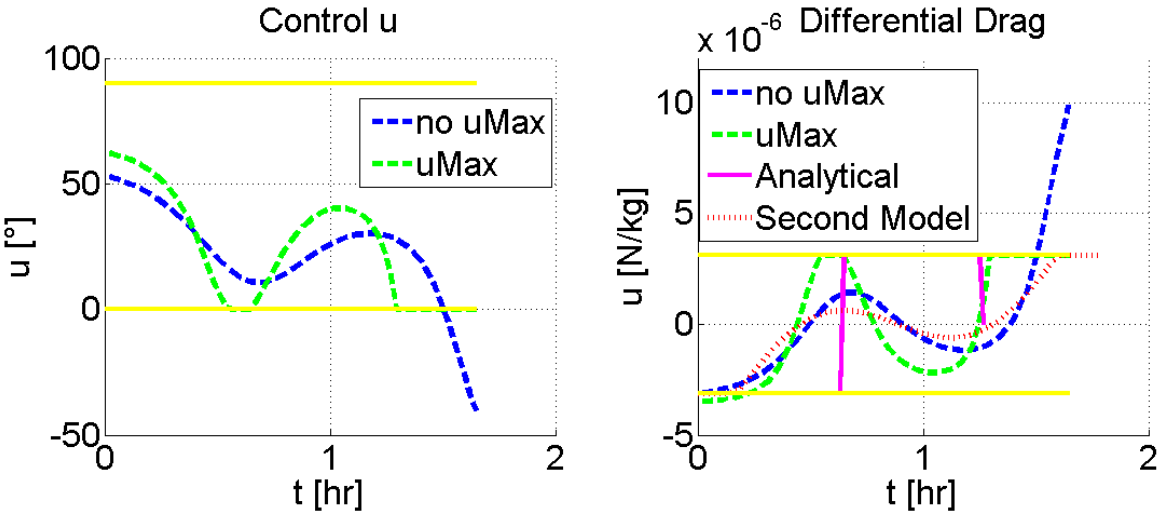


Figure 6.27: Control and Differential Drag

solution of the control.

When the optimized initial conditions of the minimum time and energy problem of the Second Model are used on the minimum time of the Third Model, an acceptable solution is found, with an error of 4.3839. Actually, this takes to the next issue: the costates. The figure 6.28 shows the costate evolution for the energy minimization problems: if the shapes are the right ones, very similar to the Second Model, the values are completely different. In the previous two models, the adjoint variables keep very small values, below 10^{-7}; this Model has values 14 orders of magnitude bigger. This could be the cause for the failure of the time minimization problems. Even when the initial conditions of the previous Model are used for the unconstrained problem, the results are not so different than these. This represents the realization of the most important issue for the Indirect Methods: the initialization.

6.4.4 Fourth Model : Mean-Oscillatory with Angular Acceleration of Reaction Wheels

This final Model is the most complex in terms of system dynamics and number of variables. Because of this, only the simple case 1 has been studied: after three days of simulation, the case 8 has not yet given a solution.

As the figures 6.29(a), 6.29(b) show, the solution is not found also with the initial conditions of the Second Model. Actually, these conditions are only

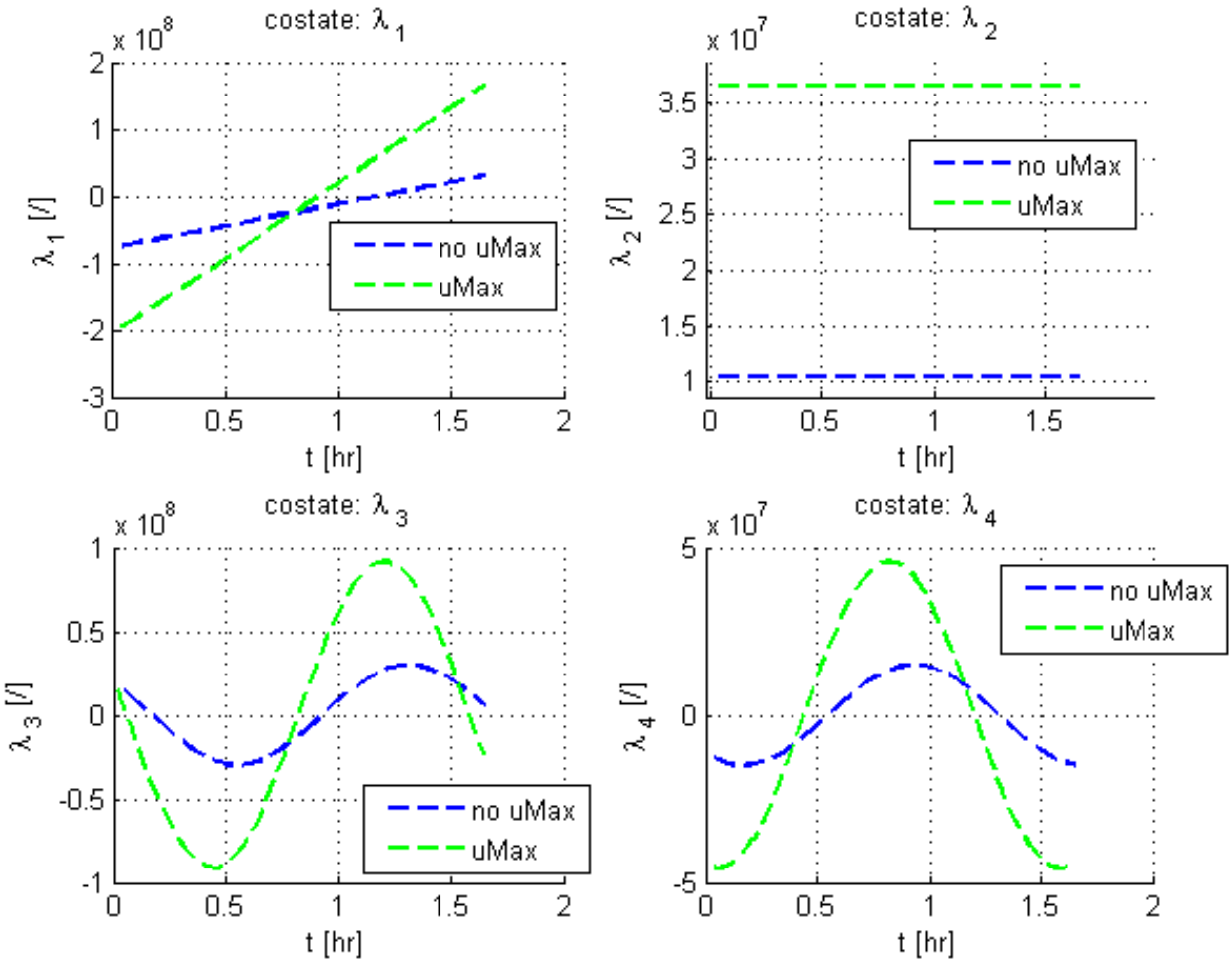


Figure 6.28: The costates evolution for case 1

four, for the four state variables, while this Model counts seven variables (the four states, the angle δ, the angular velocity of the satellite $\dot{\delta}$ and the angular velocity of the reaction wheels ω_w). These last ones are simply set to zero, because no information can be obtained from the previous Model. The "homotopic" strategy has failed for this Model. The causes can be different: a wrong numerical implementation and an extreme sensibility of the Model with respect to the initial conditions, even in the unconstrained problem, are the most probable.

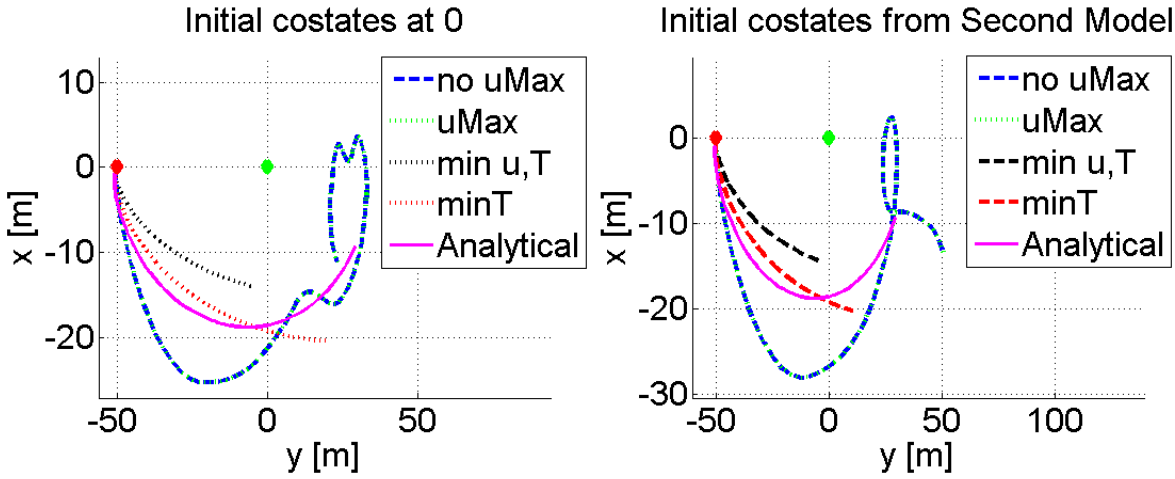


Figure 6.29: Global trajectories

6.5 Comparison with a direct method

The direct methods are the most used in trajectory optimization, because of the generality and the limited sensitivity to initial conditions. A comparison with indirect methods can be interesting. For this purpose, the presudospectral methods are used, in particular the program *GPOPS* [19]. The reason is quite simple: the program already exists and a version for the Fourth Model has been provided [8]. The program is very adaptable and different tutorials exist [19, 43].

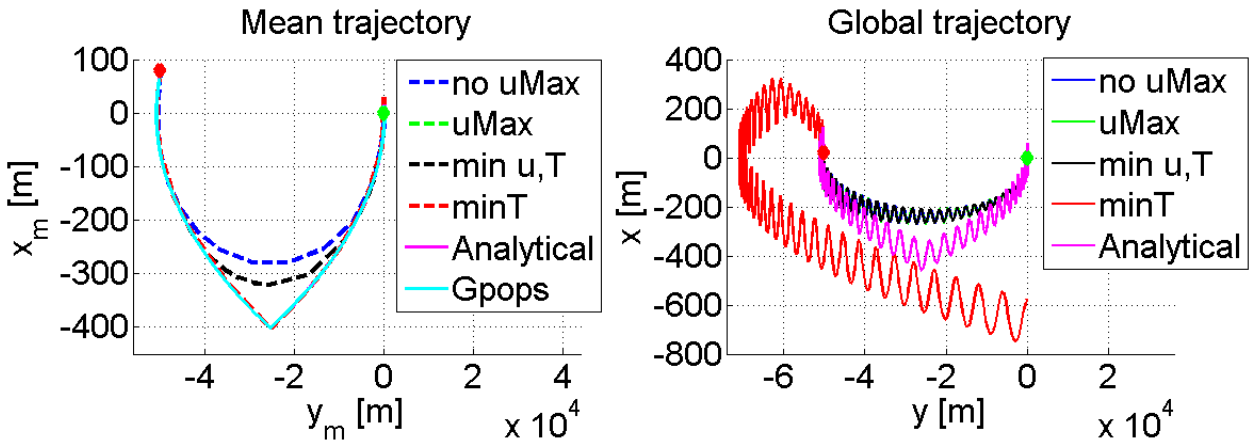


Figure 6.30: Trajectories for case 8

As the figures 6.30(a), 6.30(b) show, the trajectories are very similar to the

analytical one. This represents a good and a bad point for these methods: when a good initial trajectory is used, the results are impressive; but when this solution is not provided, more work has to be done. Fortunately, the presence of the analytical solution has simplified the numerical implementation. During the tests for the different cases, a second important aspect arises: the scaling. A small change in the scaling factors can really modify the entire trajectory. Each test case needs a peculiar scaling factor, increasing the global computational time, because several tries are necessary. Finally, the costates. As a pseudospectral method, the adjoint variables can be used to check the control of the direct method [21]. Not in all tests cases, these variables follow the right behaviour (figure 6.31), without any consequences on the trajectory [3].

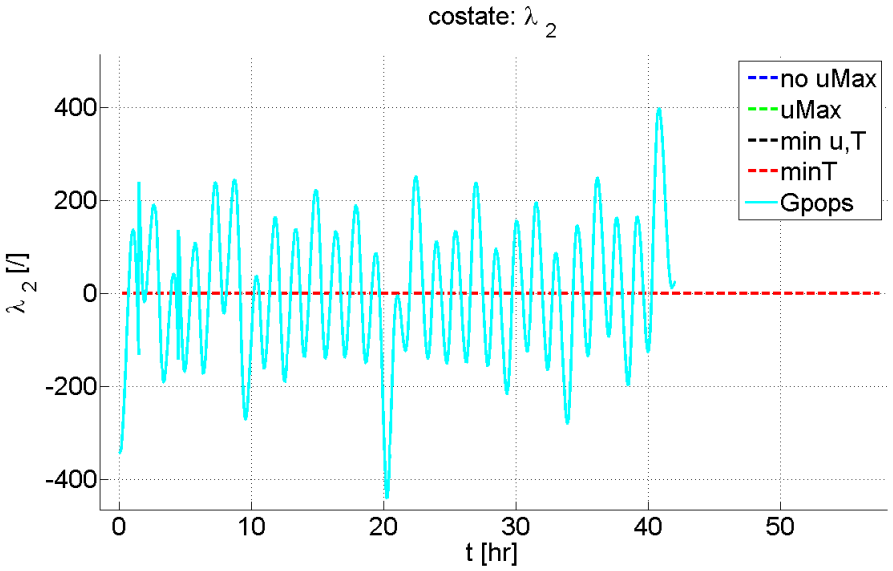


Figure 6.31: The λ_2 evolution for case 4

As already well known, the second costate should be constant. Actually, a real cause for this has not been found: the most probable is in a wrong implementation of the Model, despite the right trajectory.

[3] This depends on the fact that the costates, in the direct methods, are computed *a posteriori*, from the discretized Lagrangian Multipliers.

Chapter 7

Conclusions and Future Developments

The main objective of the thesis was to exploit an indirect method to evaluate the optimal trajectory for a rendez-vous manoeuvre, using the differential drag as actuator.
The implemented indirect method is the simplest one: the Single Shooting Method. The idea is to search for the zero of a defect function, composed by final and transversality conditions, modifying the initial conditions. The choice has fallen on this, because its simplicity implies also smaller computational cost, when it converges. Evidently, when the convergence is not reached, the other methods are more suitable. However, even in these cases, the same method has been kept. Practically, this method has been implemented in a code, taking to good results for simpler problems. The more complex ones require some adjustments (Numerical Differentiation or Broyden update for the Jacobian, definition of several parameters). For this reason, the already implemented $fminsearch$ code in $Matlab$® has been used: the approach changes from searching the zero of a function (classical SSM) to searching the minimum value of its norm ($fminsearch$). The already definite code has considerably reduced the computational work.
The second step is the "homotopic" type approach: link a simple problem to a more complex one, exploiting its solutions. In particular, the first problem is the unconstrained minimum energy, normally presenting a quite simple solution. Its optimized initial conditions on the costate variables are then used as initial guesses for the more complicated constrained problems, the minimum energy, minimum time and a combination of the two. The control bounds are progressively added through an optimization parameter, similar to the homotopic one. The main difference with the classical continuation approach is in this parameter: the homotopic one links the two problems,

varying from 0 to 1; this one changes from a certain value, bigger control bounds, more similar to the unconstrained problem, to 1, the right control bounds.
The need for this approach comes from the initialization issue, typical of these methods. The causes are in two directions: on one side, these methods use a Newton-based approach, characterized by a global solution, but with a very small radius of convergence. The second aspect is linked to the intrinsic problems of these methods: the instabilities. The dynamic equations are unstable, because of the presence of the costate variables; and the imposition of state and/or control constraints adds complications, invoking the knowledge of the switching structure of the control problem. Fortunately, in this thesis, the only maximum and minimum control bounds are imposed. The presence of these has requested the implementation of the "homotopic" approach: even simple constraints prevent from a solution in indirect methods.
The four problems (the unconstrained one and the three constrained ones) have been tested with four different dynamic models and several test cases. Very good results have been obtained for the simplest Model -the First Model- and a more complex one- the Second Model-. Despite its simplicity, this last model has a good representation of the reality. In order to increase its realistic basis, two other Models have been derived. The difference is in the control variable: the Second Model considers directly the differential drag, the Third Model reduces it to the along-track direction angle of the satellite,and the last model includes a mechanical representation of the chaser, defining the angular acceleration of the reaction wheels in an inertial reference frame as its actuator. The increase in complication of the Model has taken to often quite poor solutions. The causes are several: from a possible wrong implementation, to the necessity of good initial conditions also for the unconstrained minimum energy problem.

Future perspectives exist in the improvement of the realistic aspect of the model. Firstly, a more realistic atmosphere: in this thesis, density has been considered constant. A temperature model is envisaged, together with the inclusion of the change in density due to light and dark side of the earth. The classical two-body problem dynamics has been considered with the only J_2 gravity perturbation. The other disturbances can be analysed: solar pressure and third body gravitational effect.
The presented control law approach focuses on the in-plane motion, because the differential drag has no effect on the out-of-plane one. The control of this part has to be considered, with a different actuator or a new implementation of the differential drag.
Finally, a complete comparison with a propellant technique, as chemical

thrusters, to underline the limits of both. The differential drag has a limited value, depending on the exposed surface of the satellite: very long distance initial conditions take to very long trajectory. The intrinsic disturbance nature of this force has to be considered: the satellite needs more often to be reboosted. This should be realized with propellant techniques: a comparison in fuel savings between a complete propellant system and an "hybrid" one is necessary.
The second aspect for future developments is in the numerical implementation. The indirect methods have great possibilities: the convergence is straightforward when the right initial conditions are used and the solution is an optimal one, with respect to the suboptimal of the direct methods. However, the necessity of deriving the optimality conditions, that can be very tough, and the initialization issue have limited their developments. Lately, a new interest is born on these methods, taking to possible future improvements. Two possible ways exist: implement a completely new strategy, based on indirect methods, or exploit the "hybrid" approach. This last one has already a certain number of followers, because it represents the possibility of summing the advantages of the two methods, eliminating the disadvantages.

Bibliography

[1] Astrodynamics course, Mr. Kerschen, Université de Liège, Belgium, 2012-2013.

[2] Schweighart S.A., Sedwick R.J., "High-fidelity linearized J_2 model for satellite formation flight", *Journal of Guidance, Control, and Dynamics*, Vol 25, 2002.

[3] Leonard C.L., "Formationkeeping of Spacecraft Via Differential Drag", M.Sc. Thesis, Massachusetts Institute of Technology, 1986.

[4] Bevilacqua R., Romano M., "Rendezvous Maneuvers of Multiple Spacecraft Using Differential Drag Under J_2 Perturbation", *Journal of Guidance, Control, and Dynamics*, Vol 31, 2008.

[5] Bevilacqua R., Hall J.S., Romano M., "Multiple spacecraft assembly maneuvers by differential drag and low thrust engines", *Celestial Mech. Dyn. Astron.*, Vol 106, 2010;

[6] Lambert C., Kumar B.S., Hamel J-F., Ng A., "Implementation and performance of formation flying using differential drag", *Acta Astronautica*, Vol 71, 2011.

[7] Dell'Elce L., Kerschen G., "Orbital rendez-vous using differential drag in the QB50 constellation", *AIAA-AAS Astrodynamics Specialists Conference*, Minneapolis, 2012.

[8] Dell'Elce L., Kerschen G., "Comparison between analytical and optimal control techniques in the differential drag based rendez-vous", 2013.

[9] Pontryagin L., Boltjanskiy V. G., Gamkrelidze R. V., Mishenko E. F., "The Mathematical theory of Optimal Control Processes", Wiley-Interscience, New York, 1969.

[10] Diehl M., Numerical Optimal Control course, K.U. Leuven, Belgium.

[11] Bryson , Ho, "Applied Optimal Control", Wiley, New York, 1975.

[12] Principle of Optimal Control, MIT OpenCourse, Spring 2008.

[13] Chachuat B., Optimal Control, McMaster University, Spring 2008.

[14] Massari M., Bernelli Zazzera F., "Optimization of Low-Thrust Trajectories for Formation Flying with Parallel Multiple Shooting".

[15] Garg D., Patterson M. A., Hager W. W., Rao A. V., Benson D., Huntington G. T., "An overview of three pseudospectral methods for the numerical solution of optimal control problem", AAS 09-332.

[16] Garg D., "Advances in Global Pseudospectral Methods for optimal control", PhD Thesis, University of Florida, 2011.

[17] Rao A. V., "A Survey of Numerical Methods for Optimal Control", AAS 09-334.

[18] Benson D., "A Gauss Pseudospectral transcription for Optimal Control", PhD Thesis, MIT, 2005.

[19] Patterson M. A., Rao A. V., "GPOPS-II Version 1.0: A General-Purpose MATLAB Toolbox for Solving Optimal Control Problems Using the Radau Pseudospectral Method", University of Florida, 2013.

[20] Oberle H. J., Grimm W., "BNDSCO A Program for the Numerical Solution of Optimal Control Problems".

[21] Fahroo F., Ross I. M.,"Costate Estimation by a Legendre Pseudospectral Method", *Journal of Guidance, Control, and Dynamics*, Vol 24, 2001.

[22] Schaub H., "Incorporating the secular drifts into the orbit element difference description of relative orbits", *AAS/AIAA Space Flight Mechanics Meeting*, Ponce, 2003.

[23] Sargent T.W.H., "Optimal Control", *Journal of Computational and Applied Mathematics*, Vol 124, 2000.

[24] Harris M.W, Açikmese B., "Minimum Time Rendezvous of Multiple Spacecraft using Differential Drag", University of Texas at Austin at Austin, Texas.

[25] Binder T., Blank L., Bock H.G., Bulirsch R., Dahmen W., Diehl M., Kronseder T., Marquardt W., Schlöder J.P., von Stryk O., "Introduction to Model Based Optimization of Chemical Processes on Moving Horizons", in "Online Optimization of Large Scale Systems", Springer, 2001.

[26] Burlisch R., Nerz E., Persch H.J., von Stryk O., "Combining Direct and Indirect Methods in Optimal Control: Range Maximization of a Hang Glider", *Optimal Control*, Vol 111 of International Series of Numerical Mathematics, 1991.

[27] Trélat E., "Optimal control and applications to aerospace: some results and challenges", *Journal of Optimization Theory and Applications*, Vol 154, 2012.

[28] Pesch H.J., "A practical guide to the solution of real-life optimal control problems", Mathematisches Institut, Technische Universität München, 1994.

[29] Pesch H.J., "Optimal and Nearly Optimal Guidance by Multiple Shooting", *Proceedings of the International Symposium*, 1989.

[30] Cots O., "Contrôle Optimal Géométrique: Méthodes Homotopiques et Applications", PhD Thesis, Université de Bourgogne, 2012.

[31] von Stryk O., Burlisch R., "Direct and Indirect Methods for Trajectory Optimizations", *Annals of Operations Research*, Vol 37, 1992.

[32] Hannemann R., Marquardt W., "Combining Direct and Indirect Methods for Optimal Control- a Case Study", *Proceedings of the 9th International Symposium on Dynamics and Control of Process Systems*, Leuven, 2010.

[33] Betts J.T., "Survey of Numerical Methods for Trajectory Optimization", *Journal of Guidance, Control, and Dynamics*, Vol 21, 1998.

[34] Betts J.T., "Practical Methods for Optimal Control and Estimation Using Nonlinear Programming", Advances in Design and Control, 2010.

[35] Holsapple R.,"A Modified Simple Shooting Method for Solving Two-Point Boundary Value Problems", Master thesis, Texas Tech University, 2003.

[36] Jiang F., Baoyin H., Li J., "Practical Techniques for Low-Thrust Trajectory Optimization with Homotopic Approach", *Journal of Guidance, Control, and Dynamics*, Vol 35, 2012.

[37] Rasotto M., "Optimal Low-Thrust Transfers in Two-Body and Three-Body Dynamics", *64th International Astronautical Congress*, 2013.

[38] Di Lizia P., Armellin R., Morselli A., Bernelli-Zazzera F., "High Order Optimal Feedback Control of Space Trajectories with Bounded Control", *Acta Astronautica*, 2012.

[39] Graichen K., Petit N., "Constructive Methods for Initialization and Handling Mixed State-Input Constraints in Optimal Control", *Journal of Guidance, Control, and Dynamics*, Vol 31, 2008.

[40] Cho D.H., Vadali S.R., "The successive Backward Sweep Method for Optimal Control of Nonlinear Systems with Constraints", AAS 12-610.

[41] McReynolds S.R., "The successive Sweep Method and Dynamic Programming", *Journal of Mathematical Analysis ans Applications*, Vol 19, 1967.

[42] The Mathworks, *Matlab* help.

[43] www.gpops.org

Appendix A

The derivation of Hill-Clohessy-Wiltshire equations: from [1].
The starting point is the dynamics of the classical "two-body problem":

$$\ddot{\mathbf{r}} = -\mu \frac{\mathbf{r}}{r^3} \tag{A.1}$$

The position of one satellite, the chaser, can be expressed in function of the target's one (figure 2.1):

$$\mathbf{r} = \mathbf{r}_0 + \delta\mathbf{r} \tag{A.2}$$

The hypothesis of small distances with respect to the orbit dimensions are exploited:

$$\frac{\delta r}{r_0} \ll 1 \tag{A.3}$$

The system dynamics can then be derived as:

$$\ddot{\mathbf{r}} = -\mu \frac{\mathbf{r}}{r^3} \quad \rightarrow \quad \ddot{\mathbf{r}}_0 + \delta\ddot{\mathbf{r}} = -\mu \frac{\mathbf{r}_0 + \delta\mathbf{r}}{r^3} \tag{A.4}$$

The r^3 is obtained as:

$$r^2 = (\mathbf{r}_0 + \delta\mathbf{r})(\mathbf{r}_0 + \delta\mathbf{r}) = r_0^2 + 2\delta\mathbf{r}.\mathbf{r}_0 + \delta r^2 \tag{A.5}$$

The higher order terms can be eliminated:

$$r^2 = r_0^2 \left(1 + \frac{2\delta\mathbf{r}.\mathbf{r}_0}{r_0^2} + \frac{\delta r^2}{r_0^2}\right) = r_0^2 \left(1 + \frac{2\delta\mathbf{r}.\mathbf{r}_0}{r_0^2}\right) \tag{A.6}$$

The result is then obtained:

$$r^{-3} = r^{-2\frac{3}{2}} \quad \rightarrow \quad r^{-3} = \left[r_0^{-2}\left(1 + \frac{2\delta\mathbf{r}\mathbf{r}_0}{r_0^2}\right)^{-1}\right]^{\frac{3}{2}} \tag{A.7}$$

$$r^{-3} = r_0^{-3}\left(1 + \frac{2\delta\mathbf{r}.\mathbf{r}_0}{r_0^2}\right)^{\frac{-3}{2}} \tag{A.8}$$

Through the implement of the binomial theorem:

$$r^{-3} = r_0^{-3}\left(1 - \frac{3\delta\mathbf{r}.\mathbf{r}_0}{r_0^2}\right) \tag{A.9}$$

The dynamics become:

$$\begin{aligned}\delta\ddot{\mathbf{r}} &= -\ddot{\mathbf{r}}_0 - \mu\frac{\mathbf{r}_0 + \delta\mathbf{r}}{r_0^3\left(1 - \frac{3\delta\mathbf{r}.\mathbf{r}_0}{r_0^2}\right)^{-1}} \\ &= -\ddot{\mathbf{r}}_0 - \mu\left(1 - \frac{3\delta\mathbf{r}.\mathbf{r}_0}{r_0^2}\right)\frac{\mathbf{r}_0 + \delta\mathbf{r}}{r_0^3} \\ &= -\ddot{\mathbf{r}}_0 - \mu\frac{\mathbf{r}_0}{r_0^3} + \mu\frac{3\delta\mathbf{r}.\mathbf{r}_0}{r_0^5}\mathbf{r}_0 - \mu\frac{\delta\mathbf{r}}{r_0^3} + \mu\frac{3\delta\mathbf{r}.\mathbf{r}_0}{r_0^5}\delta\mathbf{r}\end{aligned} \tag{A.10}$$

The first two terms represent the classical two-body problem for the target and the last term can be neglected because of the higher terms presented. The final form is:

$$\delta\ddot{\mathbf{r}} = -\frac{\mu}{r_0^3}\left(\delta\mathbf{r} - \frac{3\delta\mathbf{r}.\mathbf{r}_0}{r_0^2}\mathbf{r}_0\right) \tag{A.11}$$

This represents a set of linear ordinary differential equation, still in the inertial frame.
A switch to the local reference frame allows to consider only the relative motion. The reference system is the one presented in Chapter 2, figure 2.2.

$$\begin{aligned}\ddot{\mathbf{r}} &= \ddot{\mathbf{r}}_0 + \frac{d\boldsymbol{\Omega}}{dt} \times \delta\mathbf{r} + \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \delta\mathbf{r}) + 2\boldsymbol{\Omega} \times \delta\mathbf{v}_{rel} + \delta\mathbf{a}_{rel} \\ &= \ddot{\mathbf{r}}_0 + \delta\ddot{\mathbf{r}}\end{aligned} \tag{A.12}$$

The angular velocity $\boldsymbol{\Omega}$ is considered constant:

$$\delta\ddot{\mathbf{r}} = \boldsymbol{\Omega}.(\boldsymbol{\Omega}.\delta\mathbf{r}) - \Omega^2\delta\mathbf{r} + 2\boldsymbol{\Omega} \times \delta\mathbf{v}_{rel} + \delta\mathbf{a}_{rel} \tag{A.13}$$

The following definitions are then used:

$$\delta\mathbf{r} = x\hat{\mathbf{i}} + y\hat{\mathbf{j}} + z\hat{\mathbf{k}} \tag{A.14}$$

$$\delta\mathbf{v}_{rel} = \dot{x}\hat{\mathbf{i}} + \dot{y}\hat{\mathbf{j}} + \dot{z}\hat{\mathbf{k}} \tag{A.15}$$

$$\delta\mathbf{a}_{rel} = \ddot{x}\hat{\mathbf{i}} + \ddot{y}\hat{\mathbf{j}} + \ddot{z}\hat{\mathbf{k}} \tag{A.16}$$

$$\boldsymbol{\Omega} = \omega\hat{\mathbf{k}} \tag{A.17}$$

The resulting expression is:

$$\delta\ddot{\mathbf{r}} = (-\omega^2 x - 2\omega\dot{y} + \ddot{x})\hat{\mathbf{i}} + (-\omega^2 y + 2\omega\dot{x} + \ddot{y})\hat{\mathbf{j}} + \ddot{z}\hat{\mathbf{k}} \tag{A.18}$$

The inertial form of the Linear ODE (equation A. 11) are put in the local reference frame:

$$\delta\ddot{\mathbf{r}} = -\omega^2 \left(x\hat{\mathbf{i}} + y\hat{\mathbf{j}} + z\hat{\mathbf{k}} - \frac{3r_0.\delta x}{r_0^2} r_0\hat{\mathbf{i}} \right) \tag{A.19}$$

The final expression of the Hill-Clohessy-Wiltshire equations in the local reference frame is then obtained:

I The force free case:

$$\begin{cases} \ddot{x} - 2\omega\dot{y} - 3\omega^2 x = 0 \\ \ddot{y} + 2\omega\dot{x} = 0 \\ \ddot{z} + \omega^2 z = 0 \end{cases} \tag{A.20}$$

II The forced case, with only the differential drag [1]:

$$\begin{cases} \ddot{x} - 2\omega\dot{y} - 3\omega^2 x = 0 \\ \ddot{y} + 2\omega\dot{x} = u \\ \ddot{z} + \omega^2 z = 0 \end{cases} \tag{A.21}$$

Having the differential drag no effect on the out-of-plane motion and being the z dynamics separated from the other two, the system is reduced to the only first two equations.
The J_2 gravity perturbation can be introduced [2], taking to a new set of equations:

$$\begin{cases} \ddot{x} = 2\omega c\dot{y} + (5c^2 - 2)\omega^2 x \\ \ddot{y} = -2\omega c\dot{x} + u_y \end{cases} \tag{A.22}$$

The next step is to derive the mean-oscillatory form of the equations. Through a state transformation,

$$\begin{cases} x = x_m + x_o \\ y = y_m + y_o \end{cases} \tag{A.23}$$

the system is decomposed into a double integrator and an harmonic oscillator (x, y are the global coordinates, x_m, y_m the mean ones, x_o, y_o the oscillatory ones). The real transformation is operated through:

$$\begin{cases} x_m = 4x + \frac{2}{\omega}\dot{y} \\ y_m = y - \frac{2}{\omega}\dot{x} \end{cases} \tag{A.24}$$

[1] In this thesis, no other forces are considered.

that is slightly changed with the introduction of the J_2 gravitational perturbation effect by Scheweighart-Sedwick [2]:

$$\begin{cases} x_m = \frac{4c^2}{2-c^2}x + \frac{2c}{(2-c^2)\omega}\dot{y} \\ y_m = y - \frac{2c}{(2-c^2)\omega}\dot{x} \end{cases} \tag{A.25}$$

where c is the Scheweighart-Sedwick coefficient, already defined.
The system dynamics can then be derived (considering A.22 and A.24):

$$\begin{aligned} \dot{x}_m &= \frac{4c^2}{2-c^2}\dot{x} + \frac{2c}{(2-c^2)\omega}\ddot{y} \\ &= \frac{4c^2}{2-c^2}\dot{x} + \frac{2c}{(2-c^2)\omega}(-2\omega c\dot{x} + u_y) \\ &= \left(\frac{4c^2}{2-c^2} - \frac{2c}{(2-c^2)\omega}2\omega c\right)\dot{x} + \frac{2c}{(2-c^2)\omega}u_y \\ &= \left(\frac{4c^2-4c^2}{2-c^2}\right)\dot{x} + + \frac{2c}{(2-c^2)\omega}u_y \\ &= \frac{2c}{(2-c^2)\omega}u_y \end{aligned} \tag{A.26}$$

$$\begin{aligned} \dot{y}_m &= \dot{y} - \frac{2c}{(2-c^2)\omega}\ddot{x} \\ &= \dot{y} - \frac{2c}{(2-c^2)\omega}\left(2\omega c\dot{y} + (5c^2-2)\omega^2 x\right) \\ &= \dot{y} - \frac{2c}{(2-c^2)\omega}2\omega c\dot{y} - \frac{2c}{(2-c^2)\omega}(5c^2-2)\omega^2 x \\ &= \left(1 - \frac{2c}{(2-c^2)\omega}2\omega c\right)\dot{y} + \frac{2-5c^2}{(2-c^2)\omega}2\omega c x \\ &= \frac{2-c^2-4c^2}{2-c^2}\dot{y} + \frac{2-5c^2}{(2-c^2)\omega}2\omega c x \\ &= \frac{2-5c^2}{2-c^2}\dot{y} + \frac{2-5c^2}{(2-c^2)\omega}2\omega c x \\ &= \frac{2c}{(2-c^2)\omega}\frac{2-5c^2}{2-c^2}\frac{(2-c^2)\omega}{2c}\dot{y} + \frac{2-5c^2}{(2-c^2)\omega}2\omega c\frac{4c^2}{2-c^2}\frac{2-c^2}{4c^2}x \\ &= \frac{(2-5c^2)\omega}{2c}\frac{2c}{(2-c^2)\omega}\dot{y} + \frac{(2-5c^2)\omega}{2c}\frac{4c^2}{2-c^2}x \\ &= \frac{(2-5c^2)\omega}{2c}\left(\frac{2c}{(2-c^2)\omega}\dot{y} + \frac{4c^2}{2-c^2}x\right) \\ &= \frac{(2-5c^2)\omega}{2c}x_m \end{aligned} \tag{A.27}$$

$$\begin{aligned}\dot{x}_o &= \dot{x} - \dot{x}_m \\ &= \frac{(2-c^2)\omega}{2c}(y - y_m) - \frac{2c}{(2-c^2)\omega}u_y \\ &= \frac{(2-c^2)\omega}{2c}y_o - \frac{2c}{(2-c^2)\omega}u_y\end{aligned} \tag{A.28}$$

$$\begin{aligned}\dot{y}_o = \dot{y} - \dot{y}_m &= \left(x_m - \frac{4c^2}{2-c^2}x\right)\frac{(2-c^2)\omega}{2c} - \frac{(2-5c^2)\omega}{2c}x_m \\ &= \frac{(2-c^2)\omega}{2c}x_m - \frac{4c^2}{2-c^2}\frac{(2-c^2)\omega}{2c}x - \frac{(2-5c^2)\omega}{2c}x_m \\ &= \frac{2\omega - c^2\omega - 2\omega + 5c^2\omega}{2c}x_m - 2\omega c x \\ &= \frac{4c^2\omega}{2c}x_m - 2\omega c x \\ &= 2\omega c(x_m - x) \\ &= -2\omega c x_o\end{aligned} \tag{A.29}$$