Development of a battery of tests for a numerical optimal control library

Bachelor thesis

Author
José Fernández Pérez

Tutor
Daniel González Arribas

Co-Tutor
Manuel Soler Arnedo

Bachelor in Aerospace Engineering

Spring, 2016

Contents

List of Figures

List of Tables

Acknowledgments

A mis padres, por apoyarme en todo momento durante estos cuatro años.

1 Introduction

1.1 Motivation

Optimal control theory has revolutionized the aerospace field since the beginnings of Cold War, when Lev Pontryagin derived its famous Maximum Principle in the 50s. It was used in the Space Race optimizing the weight of Sputnik 1 and 2, and later used in the launching of several spacecraft. In that time, the problem to solve was the steering of an spacecraft from Earth to a certain point in space consuming the minimum amount of fuel and in the minimum time possible, that is, finding optimum trajectories in a three-dimensional space, which could not be treated with traditional optimization techniques [1]. Thanks to the effort of North American and Soviet scientists this theory was developed.

Nowadays, this theory is fundamental to deal with the complex challenges of the actual aerospace industry, since it can be applied to many practical problems.

In the aeronautical field, airlines interest in saving costs drives to the development of new routes, based on forecast reports of weather and traffic, in order to minimize fuel consumption, and consequently, apart from the clear economic benefits, allows a reduction in the environmental impact. (The reader is invited to read about the SESAR project [2]).

Regarding the space field, optimal control theory is even more important. In space, it costs around 10000 $ to put a kilogram of payload into orbit, so weight reduction and fuel consumption minimization is the main objective in this sector [3]. There is an increase of satellite-based services, whose trajectories must be optimized (GPS, communications). Also, there is still an interest in space exploration (Rosetta mission, from the European Space Agency [4]) and upcoming travels to Mars (SpaceX), where optimal control theory will play a crucial role in finding trajectories for these new missions.

When comparing current procedures and trajectories with the optimal ones, it can be seen that in many cases they are far from each other, and there is still a lot of work to do in order to achieve a better performance.

1.2 Objectives

One of the two objectives of this bachelor thesis is the development of a set of problems to be solved with an optimal control numerical method and evaluate the obtained results. The problems to be solved have been selected from the literature, which from now on are going to be named 'canonical problems'. These problems are used by authors in the field of optimal control in order to compare their results between them and assess the strength of the optimization method they are using.

The second objective of this thesis is the development of four problems, related with climb, cruise and descend performance of a general twin turbofan airliner, applying the numerical method in question, and evaluating the results. This is then a manner to apply optimal control into more common problems in the aerospace field.

2 Methodology

2.1 The optimal control problem

Optimal control theory is a mathematical discipline consisting on finding the control policy for given inputs (controls) in order to maximize or minimize a certain cost function fulfilling a number of constraints within a system governed by a set of algebraic and differential equations.

The cost function can be defined as the addition of two contributions, one depending only on the final state of the system (Mayer term), and other defined as an integral of a certain function over the time domain (Lagrangian term).

The optimal control problem can be defined as:

Determine the state $x(t) \in \mathbb{R}^n$, the control $u(t) \in \mathbb{R}^m$, the vector of static parameters $p \in \mathbb{R}^q$, the initial time, t_0, and final time, t_f, that maximizes or minimizes the cost functional:

$$J(t, x(t), u(t), p) = \phi(t_f, x(t_f)) + \int_{t_0}^{t_f} \mathcal{L}(t, x(t), u(t), p)dt \tag{2.1}$$

Subjected to dynamic equations;

$$\dot{x}(t) = f(x(t), u(t), p) \tag{2.2}$$

algebraic equations;

$$g(x(t), u(t), p) = 0 \tag{2.3}$$

initial boundary conditions;

$$x(t_0) = x_0 \tag{2.4}$$

final boundary conditions;

$$\psi(x(t_f)) = 0 \tag{2.5}$$

and path constraints

$$\phi_i \leq \phi(t, x(t), u(t), p) \leq \phi_u \tag{2.6}$$

The dynamics of the system are governed by equation 2.2 and equation 2.1 can be thought as a measure of the 'quality' of the trajectory chosen. Thus, optimal control consists on finding the trajectory with best quality, that is, the one which maximizes/minimizes equation 2.1. Of course, changing this equation means changing the 'quality criteria' and thus, another trajectory could be more suitable than the original one, although the dynamics of the system remain unchanged.

2.2 Numerical methods

Most of optimal control problems are highly complex. Solving them analytically, even the most simple ones, whenever possible it is an arduous task and highly impractical. This is why numerical methods have been developed, since they provide reasonably accurate solutions to problems whose analytical treatment is unfeasible.

Numerical methods can be classified in dynamic programming, direct and indirect methods. The differences between them are explained below:

2.2.1 Dynamic programming

Dynamic programming is a technique consisting on solving a complex problem by dividing it on a set of simpler sub-problems. The sub-problems are linked to each other by a recurrence relation. When applying dynamic programming to optimal control theory, time is discretized and dynamic programming is used to find a solution to the Hamilton-Jacobi-Bellman equation, which establishes the sufficient conditions for a trajectory to be optimum [7]. The main drawback is that for continuous systems with a large number of variables the computational cost grows exponentially and dynamic programming becomes impractical for the majority of optimal control problems applied to the aerospace field.

2.2.2 Indirect methods

Indirect methods are based on the first optimize and then discretize philosophy. First, Pontryagin's maximum principle is used in order to derive the necessary conditions for a trajectory to be optimum in the evaluated optimal control problem [8]. So, the objective is to find a solution by means of calculus of variations of the necessary conditions for the trajectory to be optimum. Once the conditions of optimality are derived, the problem is converted into a 2-point boundary value problem and is discretized and solved as a parameter optimization problem. Examples of indirect methods are the indirect shooting method, indirect local collocation or indirect global collocation.

Indirect methods main drawbacks are the requirement of high experience in optimal control in order to be able to derive the optimality conditions, the formulation of accurate initial guesses (Also for the costates) since these methods are highly sensible to the initial guess [5] and difficulties arising on the handling of active constraints, since their switching structure must be known or derived.

2.2.3 Direct methods

Direct methods consist on discretizing first, transforming it on a constrained parameter optimization problem, and after that, solving it using optimization methods (Typical strategy is solving with non-linear programming methods). In this thesis, the problem will be transcribed into a non-linear programming problem which will be solved using IPOPT (Interior Point OPTimizer) algorithm [9]. Examples of direct methods are the direct shooting method, local collocation method and pseudoespectral method (global collocation).

Collocation methods enforce the dynamic equations through quadrature rules or interpolation. Time is discretized in many sub-intervals. The interpolant must be chosen such that it passes through the state nodes and maintains the state derivatives at the points delimiting one sub-interval of time (grid points). The interpolant is then evaluated at points between these sub-intervals, called collocation points. At each collocation point, a constraint equating the interpolant derivative to the state derivative function is introduced to ensure that differential equations are approximately satisfied across each sub-interval of time [17].

Collocation methods are divided into local collocation and global collocation, also called pseudo-spectral methods. Local collocation methods use Hermite-Legendre-Gauss-Lobatto polynomials as interpolants, and just within each sub-interval of time. Pseudo-spectral methods use high order global polynomials in order to represent globally the solution, and the collocation points are determined by a set of quadrature rules. More information about pseudo-spectral methods can be found in [11].

Direct methods are easier to implement since optimality conditions are not required although precision could be in some cases lower than their alternatives. Nowadays these methods are considered the best option when solving aerospace optimal control problems.

2.2.4 Local collocation

In this thesis a local collocation method is used to solve the optimal control problems. In local direct collocation, the objetive is to replace the original set of differential equations by a set of constraints, called defect constraints. Thus, the problem is transcribed into a non-linear programming problem (NLP) and solved using non-linear programming optimization methods.

First, as in all direct methods, the time span is divided into many sub-intervals. The end-points of these sub-intervals are denoted by:

$$\{t_0, t_1 ... t_N\}$$

The NLP variables become the values of the state and control at the grid points:

$$\{x_0, u_0, x_1, u_1, ..., x_n, u_n\}$$

In order to ensure continuity in the state variables across the sub-intervals, the following compatibility constraint is enforced at the interface of each sub-interval:

$$x(t_i^-) = x(t_i^+)$$

The integration is performed as follows. Considering:

$$\frac{dx}{dt} = f(t)$$

Then, it can be stated that:

$$x_{i+1} = x_i + \int_{t_i}^{t_{i+1}} f(t) dt$$

The above integral can be numerically approximated with different methods. In this collocation method the resulting polynomial interpolants take the form of a family of modified-Gaussian quadrature rules known as the Gauss-Lobatto rules.

One of the most basic methods, and the one used in this thesis, is the trapezoidal rule (Second order Gauss-Lobatto integrating polynomial). In this method, the integrand is replaced by a linear function connecting the state values at the two grid points. Below, this rule is expressed mathematically:

$$x_{i+1} - x_i = \int_{t_i}^{t_{i+1}} f(x)dx \approx (t_{i+1} - t_i)\frac{f(t_{i+1}) + f(t_i)}{2}$$

Below is presented a figure giving a visual insight of the trapezoidal rule.

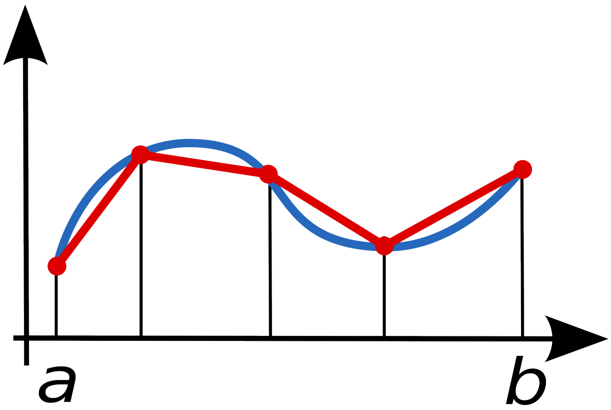


Figure 1: Graphic clarification of the trapezoidal rule

Other integration methods replaces the interpolant with a higher order polynomial than in the trapezoidal rule. For an example of a different collocation scheme the Simpson's method (Third order Gauss-Lobatto integrating polynomial) is presented below:

$$x_{i+1} - x_i = \int_{t_i}^{t_{i+1}} f(x)dx \approx (t_{i+1} - t_i)\frac{f(t_i) + 4f(t_{i,c}) + f(t_{i+1})}{6}$$

Where $t_{i,c}$ is a collocation point situated at the middle of the sub-interval, $[t_{i+1}+t_i]/2$. Higher order polynomials will present higher order collocation points, and thus, higher accuracy. When more than three collocation points are used, it is not so obvious where they must be placed in order to reduce the error as much as possible, so a mathematical method based on quadrature procedures must be followed to determine the optimum position of these points. Since this thesis only deals with two collocation points (trapezoidal rule), placed at the grid points, the explanation of this method is not under the scope of this project. For more information, the reader is referred to [17].

When evaluating the differential equation defining a dynamic system (Equation 2.2):

$$\frac{dx}{dt} = f(x)$$

A defect constraint must be created at each sub-interval. This defect constraint is expressed as follows (Trapezoidal defect constraint in this case):

$$\zeta_i(x_i, x_{i+1})^T = x_i - x_{i+1} + \frac{t_{i+1} - t_i}{2}[f(x_i) + f(x_{i+1})] = 0$$

In Simpson's rule the constraint would take the following expression:

$$\zeta_i(x_i, x_{i+1})^S = x_i - x_{i+1} + \frac{t_{i+1} - t_i}{6}[f(x_i) + 4f(x_{i,c}) + f(x_{i+1})] = 0$$

In higher order polynomials constraints acquire more complicated expressions.

Coming back to the trapezoidal rule, used in this thesis, the original problem is now transcribed into a NLP problem. The parameter to minimize is now approximated as:

$$\phi(x_n, t_n) + \sum_{i=0}^{n-1} \int_{t_i}^{t_{i+1}} \mathcal{L}(x_i, x_{i+1}, u_i, u_{i+1}) \cdot dt$$

Subjected to the constraints:

$$\zeta_i(x_i, x_{i+1}, u_i, u_{i+1}) = 0$$

Apart from the constraints stated in the problem (Equation 2.6), which are expressed in terms of the non-linear programming variables.

The NLP problem is solved with a NLP algorithm, which iterates from an initial point seeking a solution that is locally optimum. The optimality conditions of a NLP problem are called the KKT (Karush-Kuhn-Tucker) conditions, which are a generalization of Lagrange multipliers method. More information of this conditions can be found in [12].

2.3 The optimal control library

An optimal control library being developed in the Department of Bioengineering and Aerospace Engineering will be employed in this work, and the examples that have been implemented will become test cases for the library.

This library is written in *Python* and is built on top of *Pyomo* [13], a library for the solution of NLP problems providing an interface to the most popular NLP solvers (as well as solvers for other kind of optimization problems) and *Numpy/Scipy*.

It supports multi-phase optimal control problems with any boundary conditions, interior point constraints or path constraints. It implements both local collocation and pseudo-spectral methods. In the future, using the implemented problems, it will feature adaptive methods [11].

3 Canonical problems

3.1 Brachistochrone problem

This problem is the classic example in calculus of variations. It consists on finding the curve between two points such that an ideal point particle under the only action of its own weight, starting at rest at the initial point and constrained to move around the curve, reaches the final point in the shortest time.

The advantage of this problem, apart of its simplicity, is the existence of an analytical solution.

The control in this problem will be considered to be the angle formed by the curve tangent with the vertical. For simplicity, the coordinate system is selected such that gravity is directed in the direction of the vertical coordinate:

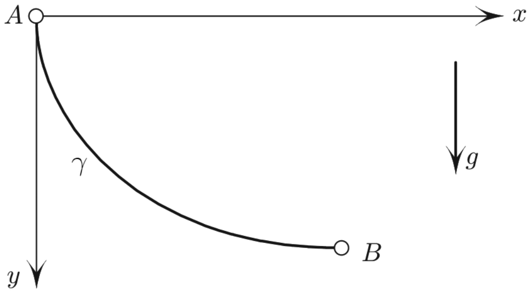


Figure 2: Reference system used in Brachistochrone problem

Thus, the system dynamics are modeled as:

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ v \end{bmatrix} = \begin{bmatrix} v \cdot sin(u) \\ v \cdot cos(u) \\ g \cdot cos(u) \end{bmatrix} \tag{3.1}$$

Minimizing the cost functional:

$$J = t_f$$

With boundary conditions:

$t_0 = 0$ [*s*]	$t_f = Free$ [*s*]
$x(0) = 0$ [*m*]	$x(t_f) = 2$ [*m*]
$y(0) = 0$ [*m*]	$y(t_f) = 2$ [*m*]
$v(0) = 0$ [*m/s*]	$v(t_f) = Free$ [*m/s*]

Subjected to the inequality constraint:

$$-\frac{\pi}{2} <= u <= \frac{3\pi}{2}$$

In order to make the control unique.

Finally, the gravity acceleration is $g = 9.80665\ [m/s^2]$.

As explained before, the NLP solver requires an initial guess to start the iteration, so it is necessary to provide an inital trajectory.

The initial guess selected is the straight line connecting the initial and final points, as this is the most simple path satisfying the conditions of the problem. Of course, it is not the optimum solution, but still is a valid trajectory from which the solver will compute the optimum one. Knowing the inital and final points, it can be seen that the control angle in the initial guess will have a constant value of $45°$.

Parametrizing all the state and control variables of the initial guess as a function of time:
$x^*(t) = sin(\pi/4)cos(\pi/4)gt^2$
$y^*(t) = sin^2(\pi/4)gt^2$
$v^*(t) = sin(\pi/4)gt$
$\theta^*(t) = \pi/4$

Below are presented the results obtained. First of all, the trajectory the particle describes is introduced below:

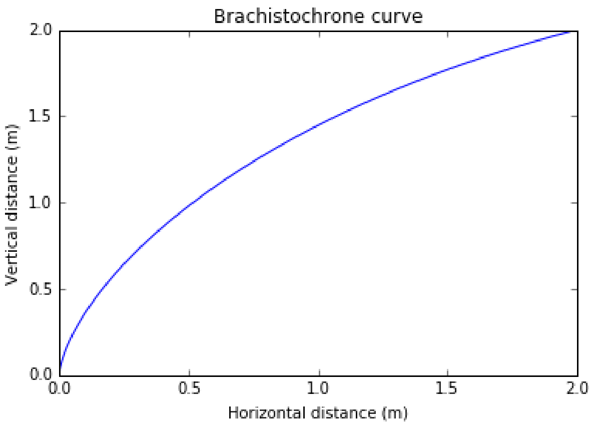
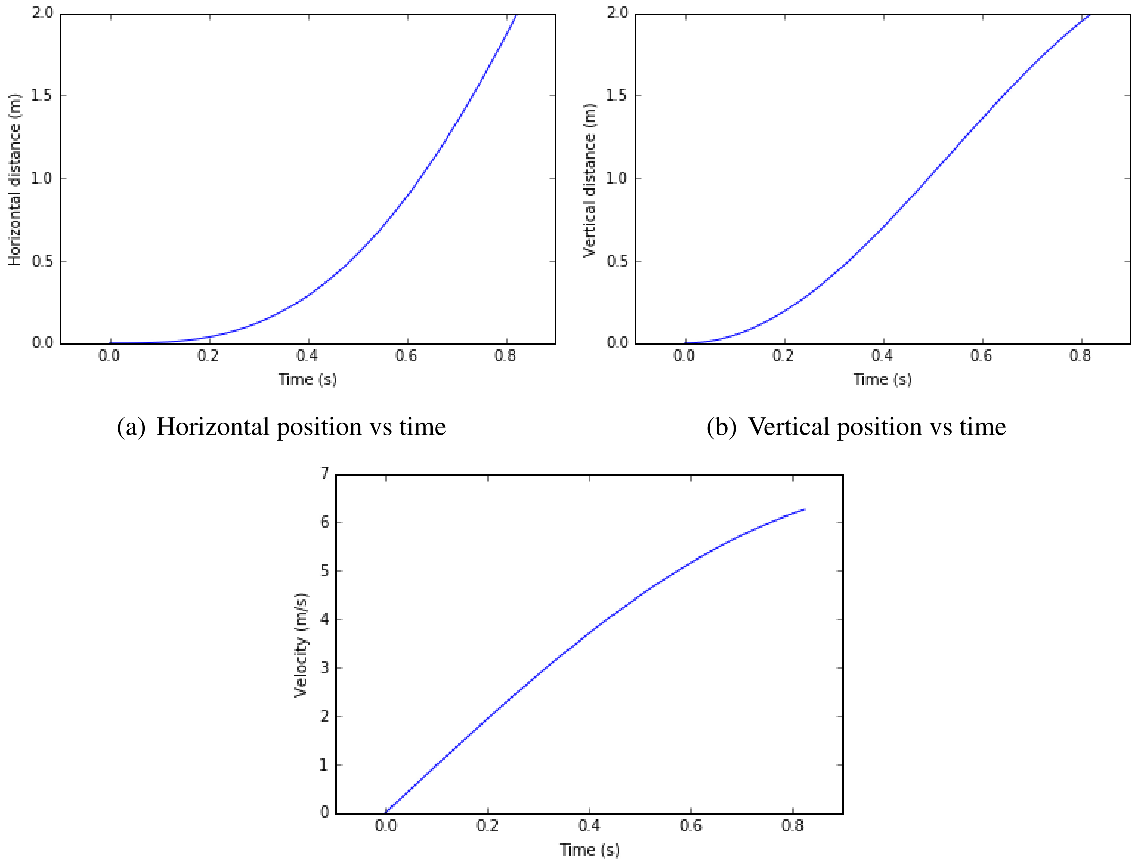


Figure 3: Brachistochrone curve

This trajectory coincides with the expected cycloid corresponding with the analytical solution.

In the next page plots of the state and control variables as a function of time are presented.



(a) Horizontal position vs time

(b) Vertical position vs time

(c) Velocity vs time

Figure 4: State variables evolution of Brachistochrone problem

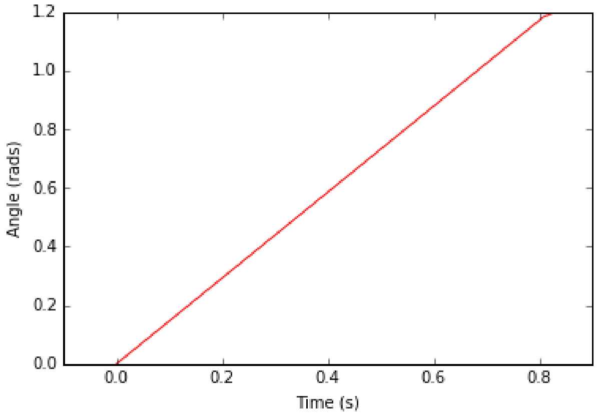


Figure 5: Control variable evolution of Brachistochrone problem

3.2 Moon lander

This problem is taken from [14]. It consists about the minimization of fuel consumption during a moon landing. For simplicity, it is analyzed in 1D. The moon lander starts at a given altitude and vertical velocity and the objective is doing an adequate use of the landing rockets in order to end up at zero altitude and at rest. As said before, the parameter to minimize is the integral over the time domain of the exerted acceleration.

The control of this problem will be the acceleration derived from the thrust force executed by the landing rockets.

The dynamics of the problem are relatively simple:

$$\frac{d}{dt}\begin{bmatrix} h \\ v \end{bmatrix} = \begin{bmatrix} v \\ -g+u \end{bmatrix} \tag{3.2}$$

Minimizing the cost functional:

$$J = \int_{t_0}^{t_f} u \cdot dt$$

With boundary conditions:

$t_0 = 0$ [*s*] $\quad$ $t_f = Free$ [*s*]
$h(0) = 10$ [*m*] $\quad$ $h(t_f) = 0$ [*m*]
$v(0) = -2$ [*m/s*] $\quad$ $v(t_f) = 0$ [*m/s*]

Subjected to the inequality constraint:

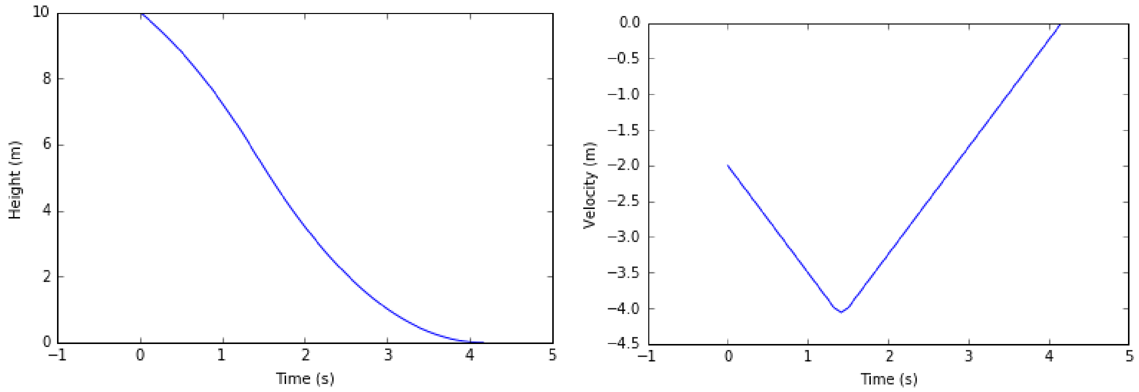
$$0 <= u <= 3$$

Where $g = 1.5[m/s^2]$.

The initial guess proposed by the author to solve this problem consists on setting the acceleration produced by the rockets linearly increasing with time. Integrating the total acceleration, the velocity is obtained as a function of time, and integrating again the height as a function of time is finally obtained. The proportionality constant of the rocket acceleration is selected such that all the state variables fulfill the boundary conditions:

$h^*(t) = 10 - 2t + 0.5/3 \cdot 0.9419t^3 - 0.5gt^2$
$v^*(t) = -2 + 0.5 \cdot 0.9419t^2 - gt$
$u^*(t) = 0.9149t$

Plots of the evolution of state and control variables are introduced below. The height presents a monotonically decreasing behavior, and the velocity, as expected, starts at a negative value and finish at rest. With respect to the control variable, it exhibits a bang-bang behavior, that is, the control only switches between the extreme values (One example of this control is boiling of water in the shortest time, first, full heat is applied and once the water boils, it is suddenly switched off). It is a behavior that appears on a certain class of optimal control problems.



(a) Height vs time

(b) Velocity vs time

Figure 6: State variables evolution of Moon lander problem

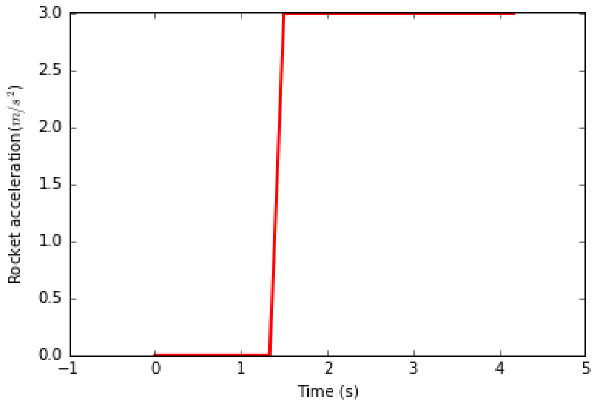


Figure 7: Control variable evolution of Moon lander problem

3.3 Orbit raising

This problem is extracted from [15]. It consists on maximizing a satellite orbit radius with a given amount of fuel. The boundary condition states that the satellite will start burning fuel and at the moment the firing is completed, the satellite must end up immersed on a circular orbit. (It would be much more efficient two firings, one the start, and one at the end, to circularize the orbit (See Hohmann transfer), but that will inevitably convert the problem in a multi-phase problem, not single-phase).

Below are presented the dynamic equations of the system (Since the problem consists about orbits, it is much more interesting using polar coordinates).

$$\frac{d}{dt}\begin{bmatrix} r \\ \theta \\ v_r \\ v_\theta \end{bmatrix} = \begin{bmatrix} v_r \\ \frac{v_\theta}{r} \\ \frac{v_\theta^2}{r} - \frac{\mu}{r^2} + a \cdot u_1 \\ -\frac{v_r \cdot v_\theta}{r} + a \cdot u_2 \end{bmatrix} \tag{3.3}$$

Minimizing the cost functional: (This is the same as maximize the final orbital radius)

$$J = -r(t_f)$$

With boundary conditions (Note that arbitrary units to simplify the problem are used):

$t_0 = 0$ [*s*]	$t_f = 3.32$ [*s*]
$r(0) = 1$ [*a.u.*]	$r(t_f) = Objective$ [*a.u.*]
$\theta(0) = 0$ [*rads*]	$\theta(t_f) = Free$ [*rads*]
$v_r(0) = 0$ [*a.u.*]	$v_r(t_f) = 0$ [*a.u.*]
$v_\theta(0) = 1$ [*a.u.*]	$v_\theta(t_f) = \sqrt{\frac{\mu}{r(t_f)}}$ [*a.u.*]

Subjected to the equality path constraint (Trigonometric restriction):

$$u_1^2 + u_2^2 = 1$$

Where $\mu = 1$, $m_0 = 1$ and vector thrust controls are u_1 and u_2. Note that the controls u_1 and u_2 are the radial and tangential components of the thrust vector. Thrust intensity can not be controlled and has a constant value of 0.145. Fuel flow rate has a constant value of 0.0749. Thus, the acceleration, present in the dynamic system, can be expressed as:

$$a(t) = \frac{T}{m_0 - \dot{m}_f t}$$

It can be seen that the above expression depends explicitly on time. Since the time variable currently can not be implemented in the algorithm, in the script a new state variable emulating time (With time derivative equal one in the dynamic system) was implemented and used as 'time'.

The initial guess is formulated as forcing the thrust to act only in the azimuthal direction and the values of the state variables are parametrized as a linear interpolation between the initial and final state of the system. This initial guess does not satisfy the dynamics of the system but it is not indispensable as NLP solvers can start from an infeasible point.

$r^*(t) = 1 + 2/9t$
$\theta^*(t) = \pi/9t$
$v_r^*(t) = -1/4t^2 + 9/4t$
$v_\theta^*(t) = 1 + \frac{\sqrt{1/3}-1}{9}t$
$u_1^*(t) = 0$
$u_2^*(t) = 1$

Solving the problem, a final radius of 1.5184 arbitrary units is obtained. This may not seems much but in fact it corresponds with an 52% increase from the original one.

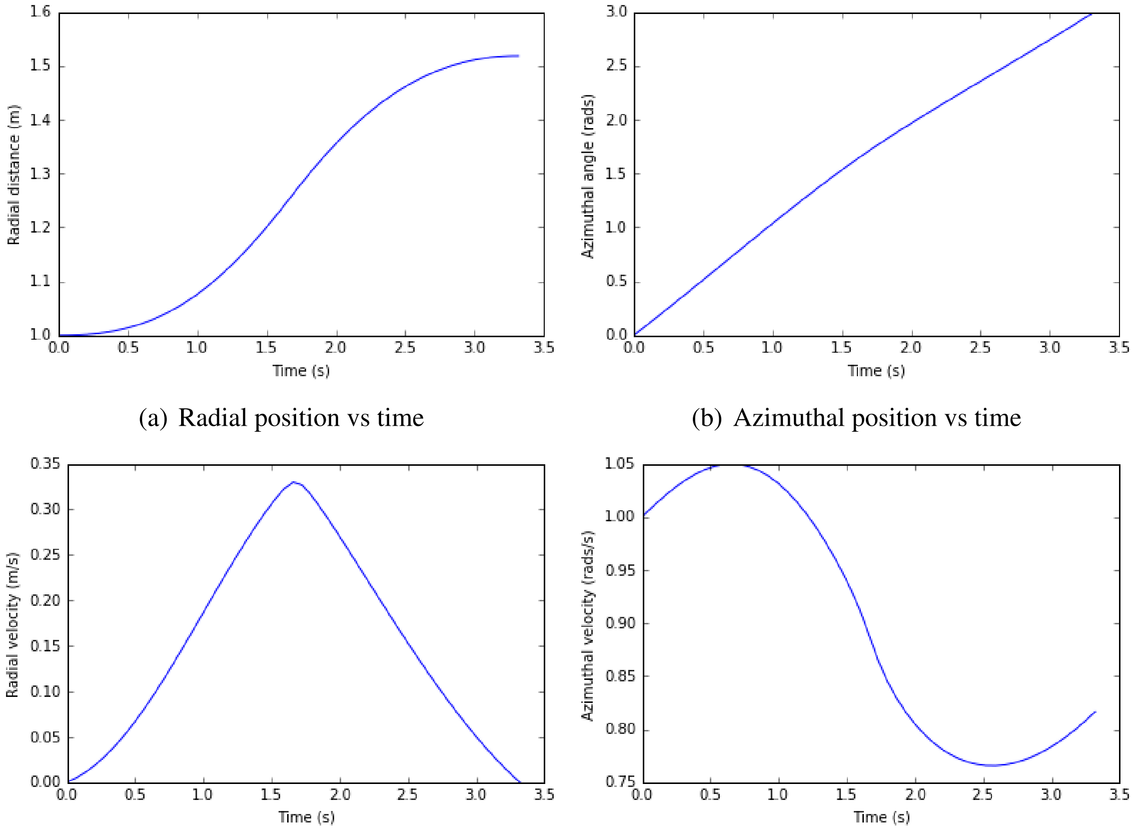
Plots of the obtained results are presented in the next page.

Regarding the radial coordinate, it can be appreciated how the first half of the time, thrust in the radial direction points outwards the orbit in order to increase the radius but in the second half, it points inwards in order to end with zero radial velocity and thus, in a circular orbit.

As well, it can be seen that the azimuthal velocity is always positive but ends with a lower value than at the start, because of the circular orbit characteristic, where the gravity acceleration exerted on the satellite must be equal to the centripetal force in order to keep the circular path:

$$G\frac{Mm}{r^2} = m\frac{v_\theta^2}{r} \Rightarrow GM = \mu = v_\theta^2 r \tag{3.4}$$

Since the gravitational parameter is constant, the above equation must be balanced, thus, a higher orbit radius means a lower azimuthal velocity.



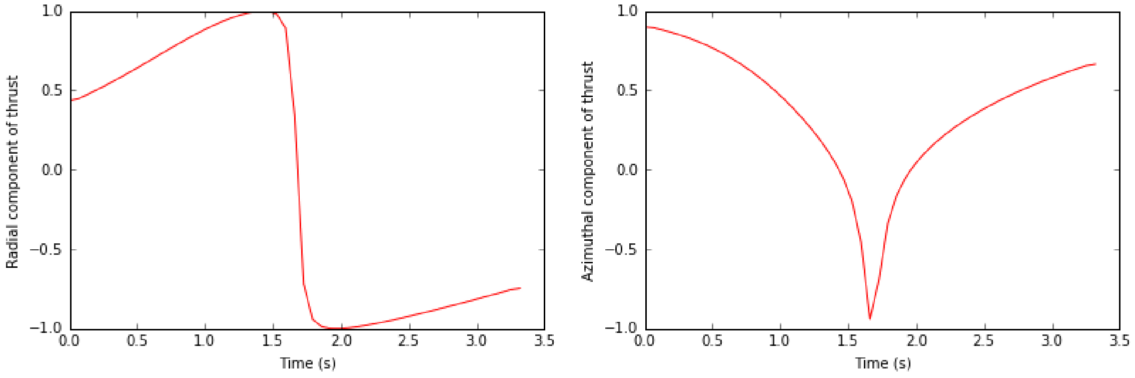
(a) Radial position vs time

(b) Azimuthal position vs time

(c) Radial velocity vs time

(d) Azimuthal velocity vs time

Figure 8: State variables evolution of Orbit rising problem



(a) Radial component of thrust vs time

(b) Azimuthal component of thrust vs time

Figure 9: Control variables evolution of Orbit rising problem

3.4 Range maximization of a hang glider

This problem is treated in [5]. It consists about range optimization of a hang glider in the presence of thermal updrafts. Glider pilots use to 'dolphin'. This is a technique consisting on reducing the airspeed when the aircraft is inside a thermal updraft in order to maximize the time they are acquiring positive vertical velocity and maximize the altitude gain. As well, when they are immersed in a thermal downdraft, the opposite action is taken, airspeed is increased in order to 'escape' from the descending stream as fast as possible.

In this problem the glider will surpass a simple-modeled thermal updraft and the control will be the lift coefficient. Below are presented the dynamic equations governing the aircraft motion in the longitudinal plane:

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ \frac{1}{m}(-Lsin(\eta) - Dcos(\eta)) \\ \frac{1}{m}(Lcos(\eta) - Dsin(\eta) - W) \end{bmatrix} \tag{3.5}$$

Range must be maximized, thus, the following cost functional must be minimized:

$$J = -x(t_f)$$

With boundary conditions:

$t_0 = 0$ [*s*]
$x(0) = 0$ [*m*]
$y(0) = 1000$ [*m*]
$v_x(0) = 13.2275675$ [*m/s*]
$v_y(0) = -1.28750052$ [*m/s*]

$t_f = Free$ [*s*]
$x(t_f) = Objective$ [*m*]
$y(t_f) = 900$ [*m*]
$v_x(t_f) = 13.2275675$ [*m/s*]
$v_y(t_f) = -1.28750052$ [*m/s*]

Subjected to the constraint:

$$0 \leq C_L \leq 1.4$$

The thermal updraft intensity, $u_a(x)$, is modeled as:

$$X(x) = \left(\frac{x}{R} - 2.5\right)^2$$
$$u_a(x) = u_M(1 - X)exp(-X)$$

The following auxiliary functions are involved in the dynamics:

$$V_y = v_y - u_a(x)$$

$$sin(\eta) = \frac{V_y}{v_r}$$

$$v_r = \sqrt{v_x^2 + V_y^2}$$

$$L = \frac{1}{2}\rho S v_r^2 C_L$$

$$D = \frac{1}{2}\rho S v_r^2 (C_{D0} + kC_L^2)$$

Finally, the following constants are necessary to completely define the problem:

- $u_M = 2.5$
- $R = 100$ $[m]$
- $C_{D0} = 0.034$
- $k = 0.069662$
- $m = 100$ $[kg]$
- $S = 14$ $[m^2]$
- $\rho = 1.13$ $[kg/m^3]$
- $g = 9.80665$ $[m/s^2]$

The initial guess used is the trajectory obtained where the velocity (both horizontal and vertical) components are assumed to be constant. Thus, vertical and horizontal position are just a linear function of time and lift coefficient is set to fulfill these conditions (No resultant force in the plane-vertical direction). The effects of the thermal updraft are ignored, which makes this initial guess incompatible with the dynamics of the system, but the solver is sufficiently robust to use this simplified path as initial guess.

$$x^*(t) = v_x(0)t \qquad v_x^*(t) = v_x(0)$$
$$y^*(t) = y(0) + v_y(0)t \qquad v_y^*(t) = v_y(0)$$
$$C_L^*(t) = \frac{2mgcos(\gamma(0))}{\rho S(v_x^2(0)+v_y^2(0))}$$

Below is presented the trajectory described by the glider where range is maximized:

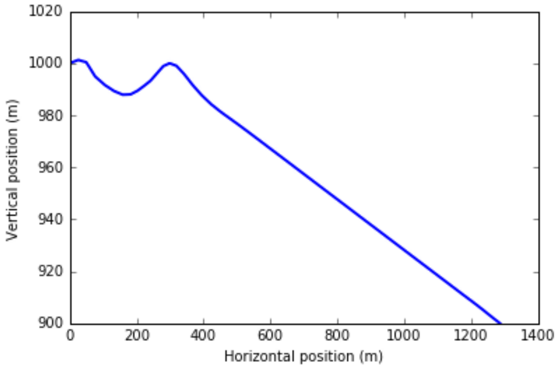


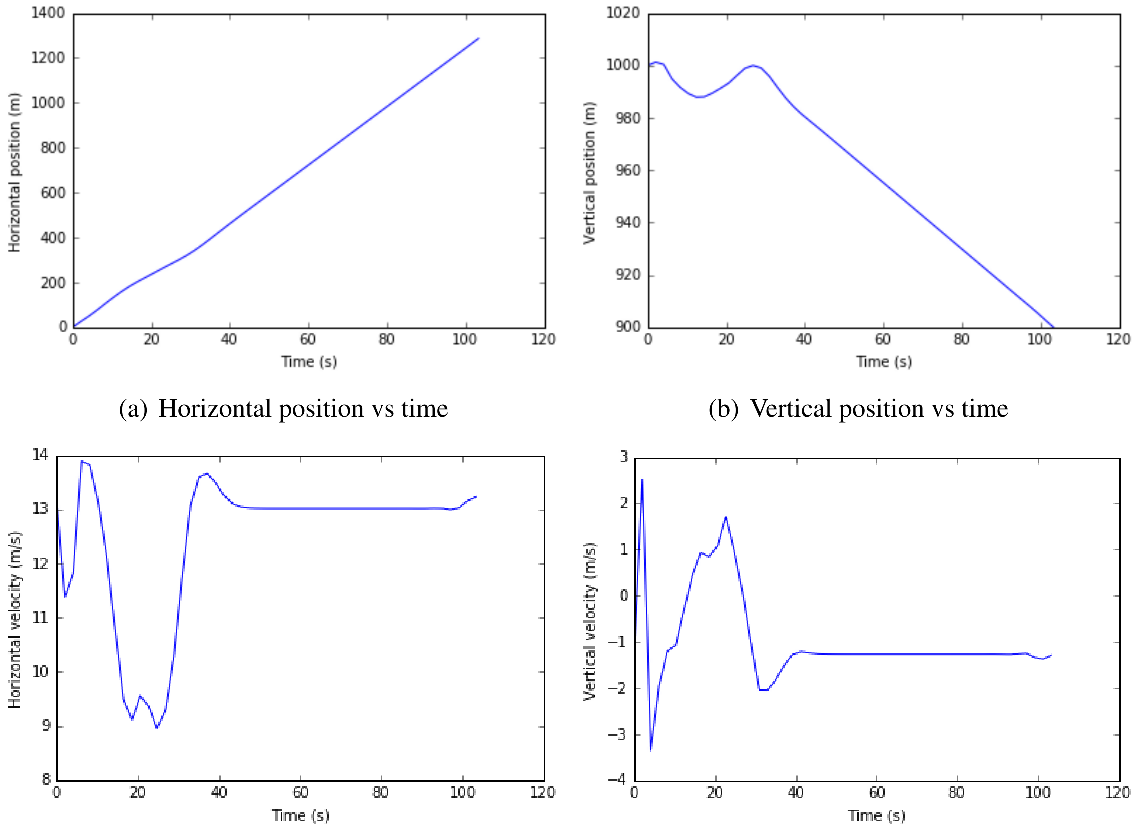
Figure 10: Trajectory of Range maximization problem

It can be seen where the thermal updraft is located (Between the 200-400 meters gap), which corresponds with the zone where an altitude gain has been produced. Once outside the thermal, the glide path is rectilinear, with constant velocity.

Plots corresponding with the evolution in time of state and the control variable can be found in the next page.

It can be seen the reduction in horizontal velocity when surpassing the thermal (Dolphin technique) and then the recovery. With respect the vertical velocity, there is a mild descent prior to the thermal, and later it reaches a peak value almost of 2 m/s of ascending velocity, on the thermal core. Then, it keeps a constant descent velocity of around -1 m/s until the end of the trajectory.

With respect to the lift coefficient, it goes to its maximum allowed value inside the thermal (1.4), to be at the lower velocity possible, in order to maximize the time inside the ascending stream.



(a) Horizontal position vs time

(b) Vertical position vs time

(c) Horizontal velocity vs time

(d) Vertical velocity vs time

Figure 11: State variables evolution of Range maximization problem

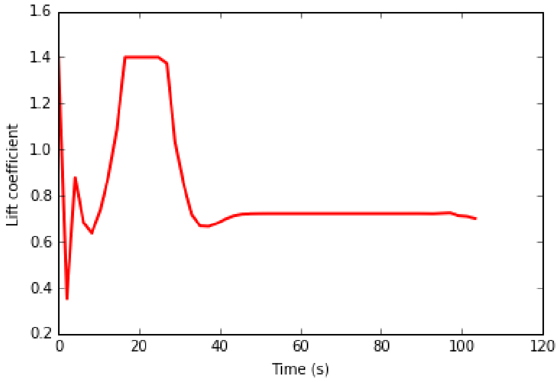


Figure 12: Control variables evolution of Range maximization problem

3.5 Free flying robot

This problem is extracted from [5]. The problem describes the motion of a simple vehicle equipped with a propulsion system. The robot moves in a 2-D horizontal plane and the controls will be the thrust exerted by the left and right engines, T_1 and T_2. In this problem thrust can be positive or negative. The robot can change its orientation by producing a torque with an asymmetric thrust profile, as it will be seen in the dynamic equations. The objective is departing at rest from an initial point and reach a final point at a given state minimizing the thrust exerted in the trajectory.

The dynamics of the system are stated as:

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ v_x \\ v_y \\ \theta \\ \omega \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ (T_1+T_2)cos(\theta) \\ (T_1+T_2)sin(\theta) \\ \omega \\ \alpha T_1 - \beta T_2 \end{bmatrix} \tag{3.6}$$

Where $\alpha = \beta = 0.2$

The objective is to minimize the cost function:

$$J = \int_{t_0}^{t_f} (|T_1| + |T_2|) \cdot dt$$

Unfortunately, the above expression presents discontinuous derivatives due to the absolute value function so a change of variable will be made (Non-linear programming solvers can not deal with discontinuous derivatives): Each thrust contribution will be splitted between their positive and negative contribution, that is:

$$T_1 = u_1 - u_2$$

$$T_2 = u_3 - u_4$$

Where $(u_1, u_2, u_3, u_4) \geq 0$

This reformulation of variables is a usual practice in non-linear programming.

Note that when one of this new variables is being used, the optimal solution will set its pair to be zero, and vice-versa, in order to avoid unnecessary fuel consumption (It is like maintaining a certain speed on a car pushing the throttle and braking pedals at the same time). Thus, the absolute values of left and right thrust can be expressed as (Without altering the solution):

$$|T_1| = u_1 + u_2$$

$$|T_2| = u_3 + u_4$$

And the cost function can be transformed into the following expression, where the absolute values disappear:

$$J = \int_{t_0}^{t_f} (u_1 + u_2 + u_3 + u_4) \cdot dt$$

The problem has the following boundary conditions. Basically the robot starts at a certain point facing North, and must end up at at the origin, at rest, and facing East:

$$\begin{array}{ll}
t_0 = 0\ [s] & t_f = Free\ [s] \\
x(0) = -10\ [a.u.] & x(t_f) = 0\ [a.u.] \\
y(0) = -10\ [a.u.] & y(t_f) = 0\ [a.u.] \\
v_x(0) = 0\ [a.u.] & v_x(t_f) = 0\ [a.u.] \\
v_y(0) = 0\ [a.u.] & v_y(t_f) = 0\ [a.u.] \\
\theta(0) = \pi/2\ [rads] & \theta(t_f) = 0\ [rads] \\
\omega(0) = 0\ [rads/s] & \omega(t_f) = 0\ [rads/s]
\end{array}$$

The robot is subjected to the following control inequality path constraints:

$$0 \leq u_i \leq 1000, \quad (i = 1, 2, 3, 4)$$

$$0 \leq T_i \leq 1, \quad (i = 1, 2, 3, 4)$$

The initial guess is taken as the trajectory consisting on the linear path between initial and final point. Thrust is selected such that the attitude is not modified during the trajectory and velocity is set constant. This initial guess do not fulfill neither the boundary conditions (The robot starts facing North and ends facing East, and in this trajectory it takes the intermediate value) and the dynamics of the system (With a constant amount of thrust, velocity can not be constant) but it is good enough for algorithm to solve the problem:

$$\begin{array}{ll}
x^*(t) = -10 + 10/12t & \omega^*(t) = 0 \\
y^*(t) = -10 + 10/12t & u_1^*(t) = 1 \\
v_x^*(t) = 10/12 & u_2^*(t) = 0 \\
v_y^*(t) = 10/12 & u_3^*(t) = 1 \\
\theta^*(t) = \pi/4 & u_4^*(t) = 0
\end{array}$$

Next two pages contain plots of state and control variables over time, and the trajectory of the robot. Note that, as explained before, when one control is acting, its complementary remains at rest.

It can be seen that the trajectory is similar to the straight line but it curves at the end in order to finish at the desired orientation.

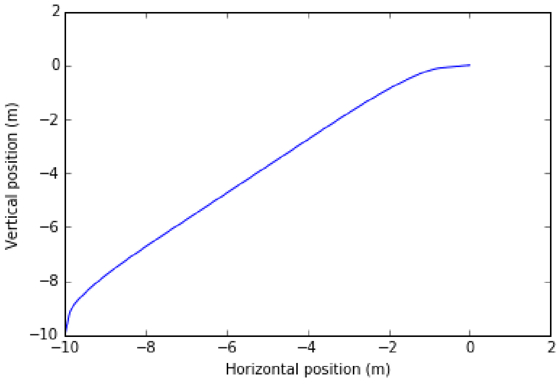
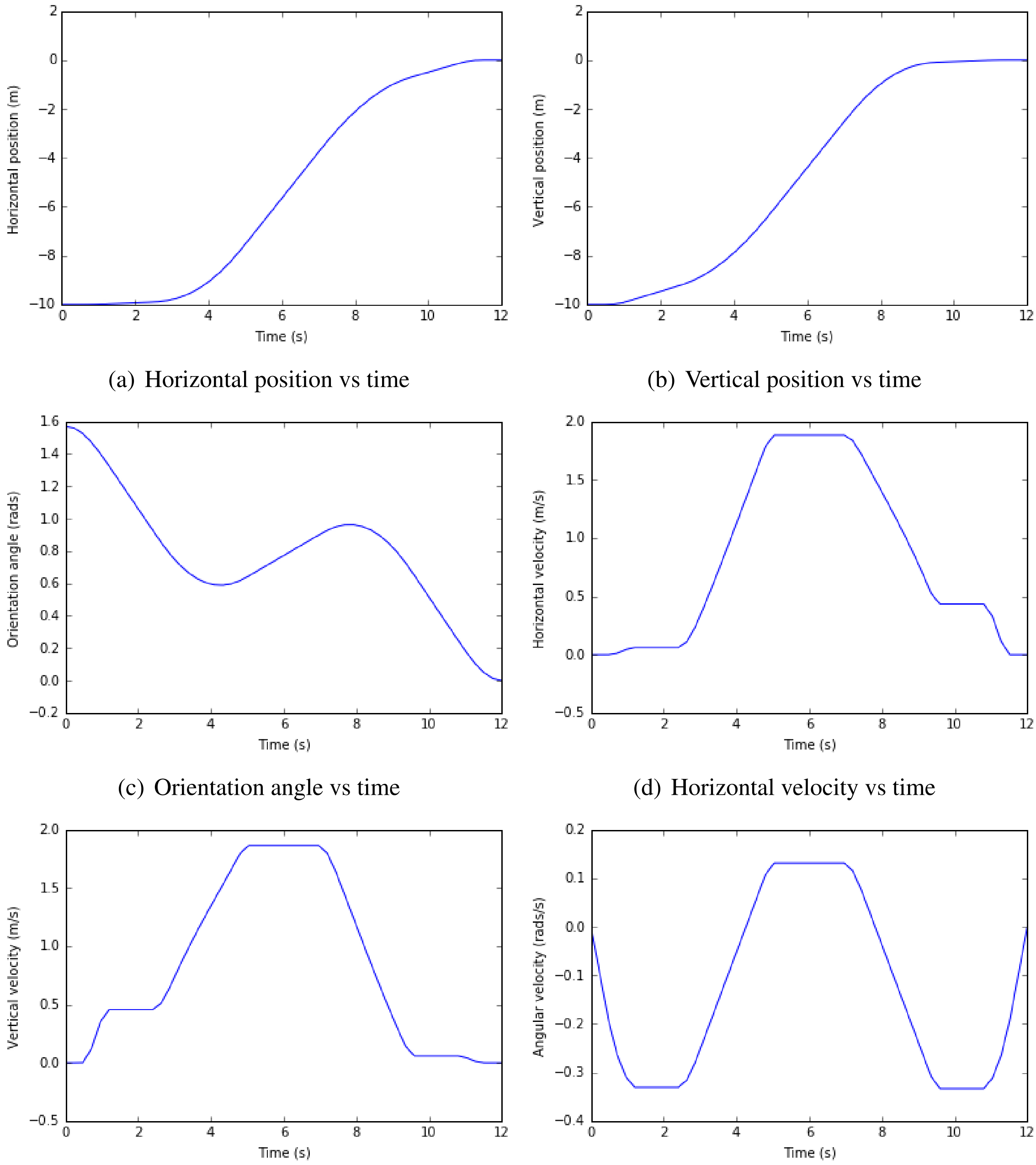


Figure 13: Trajectory of Free flying robot problem



(a) Horizontal position vs time

(b) Vertical position vs time

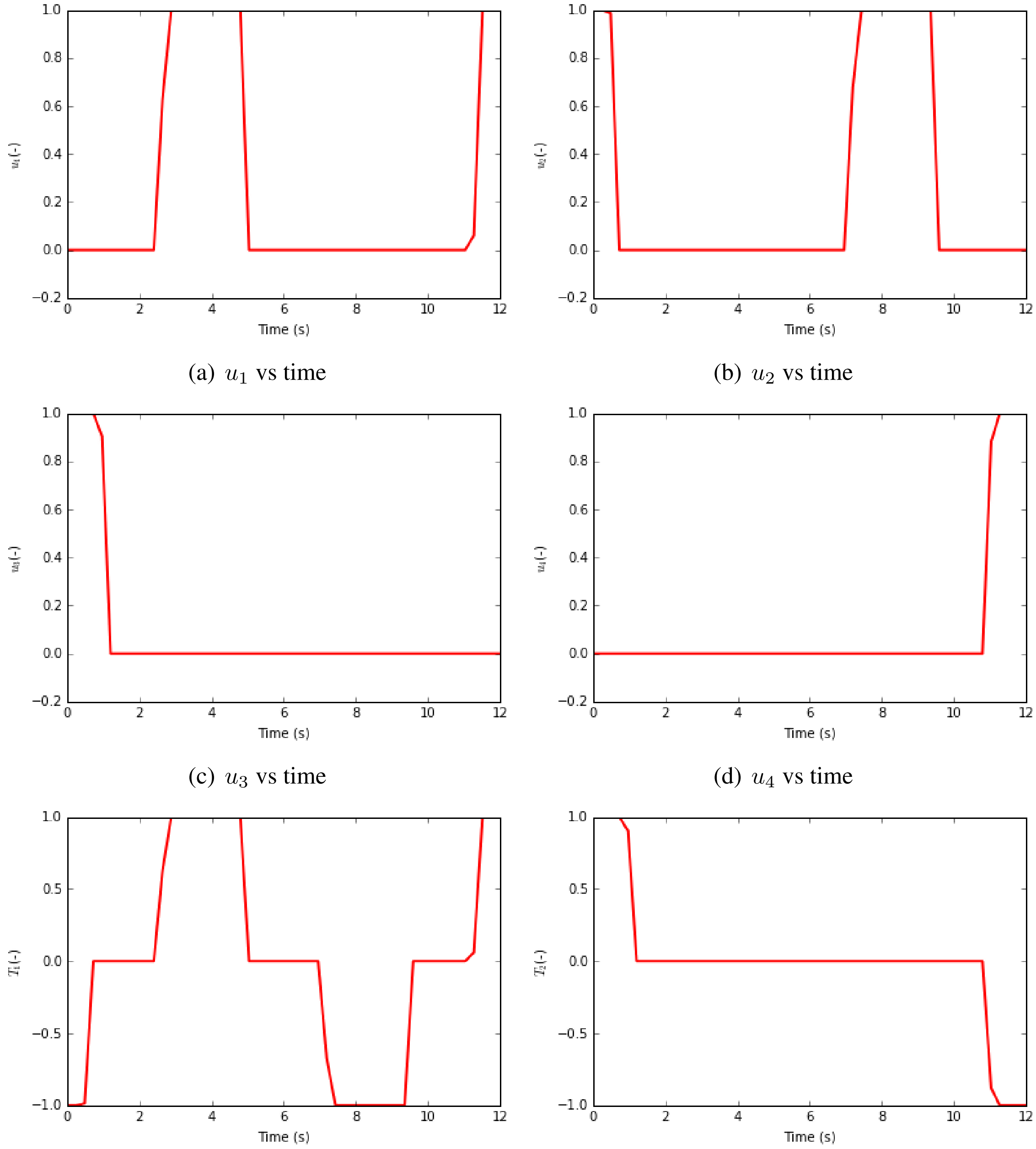
(c) Orientation angle vs time

(d) Horizontal velocity vs time

(e) Vertical velocity vs time

(f) Angular velocity vs time

Figure 14: State variables evolution of Free flying robot problem



(a) u_1 vs time

(b) u_2 vs time

(c) u_3 vs time

(d) u_4 vs time

(e) T_1 vs time

(f) T_2 vs time

Figure 15: Control variables evolution of Free flying robot problem

3.6 Dynamic soaring

This problem is seen in [14]. Dynamic soaring is a complex technique consisting on maintaining or gaining airspeed taking advantage of wind gradients with respect to altitude. It is a cyclic motion; glider starts at a given altitude and airspeed. The glider then faces wind and climbs, where wind intensity becomes larger due to the wind gradients with altitude, then gaining airspeed and keeping groundspeed due to inertia. At a given height won, the aircraft starts turning with the obtained airspeed while descending and thus obtaining an increase in groundspeed keeping airspeed constant. When the aircraft completes the turn, it faces wind again with same or even higher airspeed than in the previous cycle.

The optimal control problem is formulated as finding the minimum wind gradient such that the hang glider finish the cycle at exactly the same conditions where it started. For simplicity, the wind is coming from West and glider starts facing North.

The dynamics of the system are just the equations of motion of a flight vehicle flying in wind conditions. Since wind comes from West, it only takes action in the x-direction.

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ h \\ V \\ \gamma \\ \psi \end{bmatrix} = \begin{bmatrix} V \cdot cos(\gamma) \cdot sin(\psi) + W_x \\ V \cdot cos(\gamma) \cdot cos(\psi) \\ V \cdot sin(\gamma) \\ \left[-D - mg \cdot sin(\gamma) - m\dot{W}_x \cdot cos(\gamma) \cdot sin(\psi)\right] /m \\ \left[L \cdot cos(\sigma) - mg \cdot cos(\gamma) + m\dot{W}_x \cdot sin(\gamma) \cdot sin(\psi)\right] /mV \\ \left[L \cdot sin(\sigma) - m\dot{W}_x \cdot cos(\psi)\right] /mVcos(\gamma) \end{bmatrix} \tag{3.7}$$

The objective is finding the minimum wind gradient such that the cycle can be sustained, that is, minimize:

$$J = \beta$$

The boundary conditions are the following: The glider starts at the origin, facing North, where airspeed and flight path angle are free, and must finish the cycle at exactly the same state as in the initial conditions. This is expressed as:

$t_0 = 0$ [*s*]	$t_f = Free$ [*s*]
$x(0) = 0$ [*ft*]	$x(t_f) = x(0)$ [*ft*]
$y(0) = 0$ [*ft*]	$y(t_f) = y(0)$ [*ft*]
$h(0) = 0$ [*ft*]	$h(t_f) = h(0)$ [*ft*]
$V(0) = Free$ [*fps*]	$V(t_f) = V(0)$ [*fps*]
$\gamma(0) = Free$ [*rads*]	$\gamma(t_f) = \gamma(0)$ [*rads*]
$\psi(0) = 0$ [*rads*]	$\psi(t_f) = -2\pi$ [*rads*]

Glider's load factor is constrained to be within the range:

$$-2 \leq \frac{L}{mg} \leq 5$$

The following data complete the definition of the problem:

- $m = 5.6$ $[slugs]$
- $S = 45.09703$ $[ft^2]$
- $g = 32.2$ $[ft/s^2]$
- $\rho = 0.002378$ $[slug/ft^3]$
- $C_{D0} = 0.00873$
- $k = 0.045$

Obtaining the initial guess for this problem is a complex task due to the nature of the dynamical system. (Highly non-linear and coupled differential equations). Thus, it is not feasible to obtain an analytical expression for each state variable and approximations must be performed. At a first glance, it is known the motion will present a periodic behavior, so trigonometric functions must be used in the initial guess. Results of this problem are evaluated from [14] in order to have a physical insight of the order of magnitude of the variables of the problem. One control, the bank angle, is set to has a constant value of -0.5rads (Turning leftwards) since the initial guess is that the resulting motion will be approximated as an steady coordinated turn. Based on this, the heading angle, ψ, is linearly interpolated between its initial and final value. Rest of the parameters are defined as having a reference mean value plus the contribution of a periodic function with period corresponding to 25 seconds, which is the expected time spent of the turning, as seen in [14], and an appropriate amplitude in order to reach their maximum or minimum value expected during the corresponding phases of the turn. Thus, the initial guess is finally defined as:

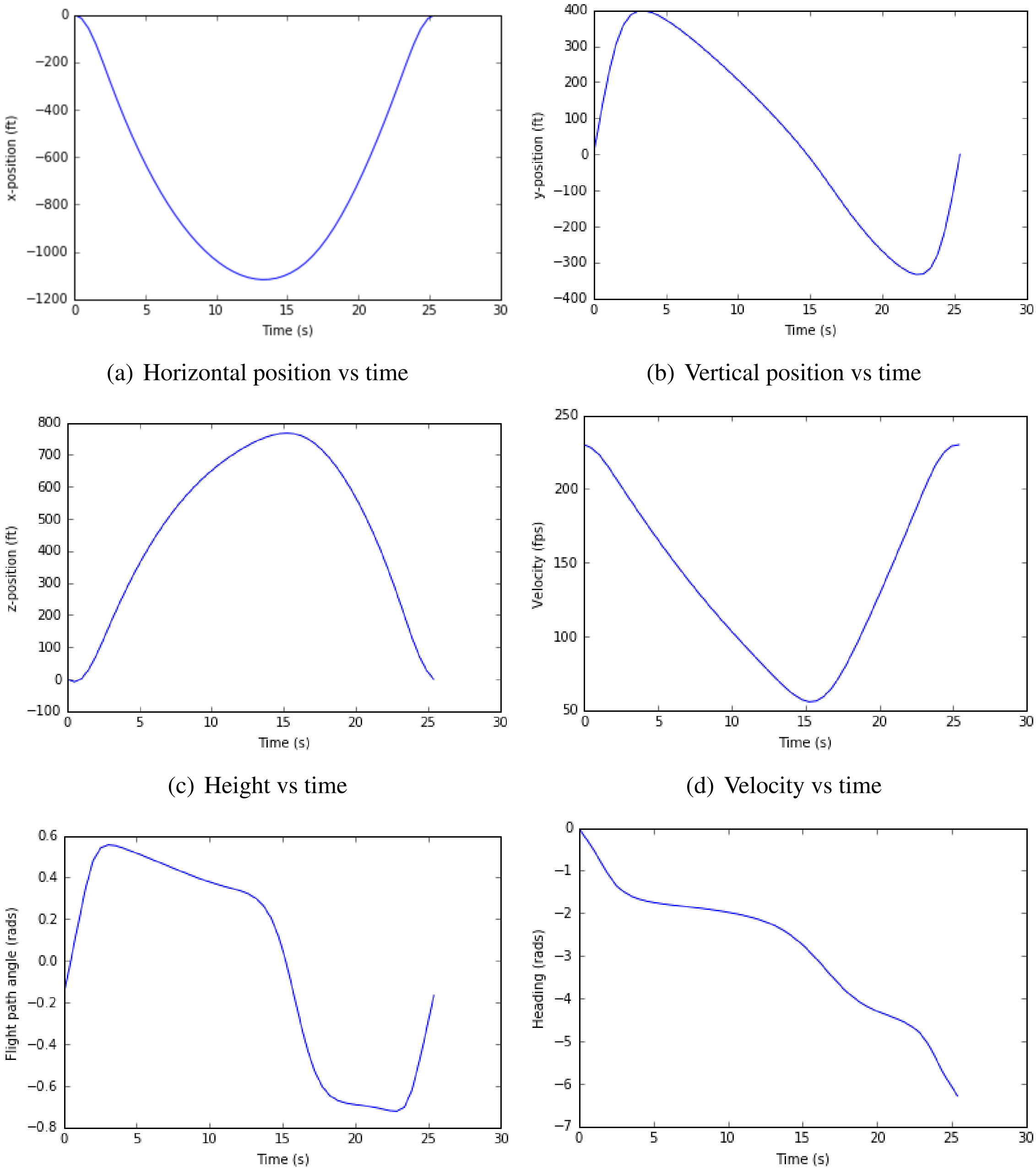
$$x^*(t) = -884.94/2 + 884.94/2cos(2\pi/25t)$$
$$y^*(t) = 884.94/2 + 884.94/2sin(2\pi/25t + 3/2\pi)$$
$$h^*(t) = 884.94/sin(5\pi/180)sin(t/25)$$
$$V^*(t) = 140 + 85cos(2\pi t/25)$$
$$\gamma^*(t) = 0.5sin(2\pi t/25)$$
$$\psi^*(t) = -2\pi/25t$$
$$C_L^*(t) = 0.5 + 0.25cos(2\pi t/16.67 + \pi/2)$$
$$\sigma^*(t) = -0.5$$

In the next two pages plots of the state and control variables evolution in time can be found, as well as the trajectory described by the hang glider.

With respect to the state variables, it can be seen that all of them are periodic (Except the heading angle, because the radians surpassed in previous orbits are kept in the variable).

In the trajectory plot, it has been plotted the projection of the flight path on the z=0 plane in order to have a visual reference and have a better perception of the trajectory described. As expected, the trajectory is a closed curve due to its periodic behavior and the maximum altitude acquired exceeds the 700 meters.

Finally, the minimum wind gradient such that the cycle can be sustained is $\beta = 0.06352$.



(a) Horizontal position vs time

(b) Vertical position vs time

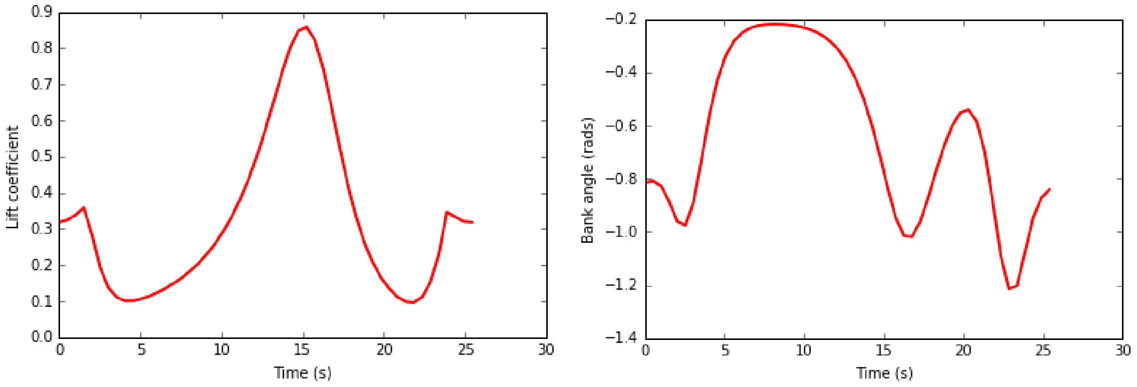
(c) Height vs time

(d) Velocity vs time

(e) Flight path angle vs time

(f) Heading angle vs time

Figure 16: State variables evolution of Dynamic soaring problem



(a) Lift coefficient vs time

(b) Bank angle vs time

Figure 17: Control variables evolution of Dynamic soaring problem

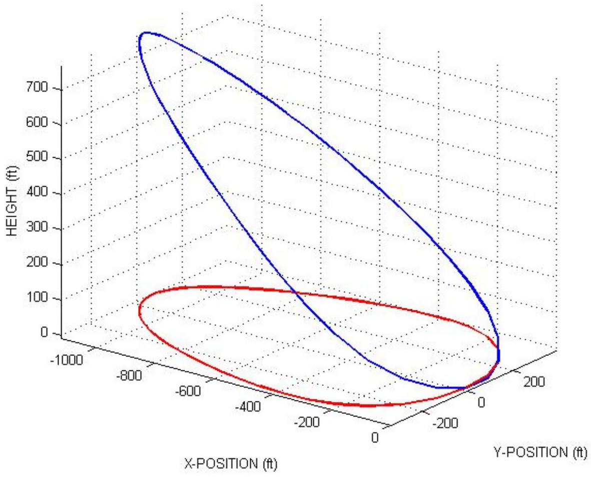


Figure 18: 3D Trajectory of Dynamic soaring problem

3.7 Minimum time to climb of a supersonic aircraft

This problem is extracted from [5]. It consists on finding the optimum trajectory for a supersonic aircraft in order to reach a determinate altitude as fast as possible. As the aircraft flies both in subsonic and supersonic conditions, lift and drag coefficients, as well as maximum thrust, will depend on the Mach number, since compressibility effects can not be neglected. One difficulty of this problem arises in the fact that these coefficients are given in tables, thus not having an analytical expression to implement in the system. Unfortunately, the software used is not prepared to work with tabular data, so analytic functions fitting the data on the tables must be developed. In the original reference the measuring system is the imperial one, so all initial conditions are expressed in this system, but they have been converted to metric system along the problem and results will be expressed in metric units.

For simplicity, an Earth-fixed reference frame has been chosen, and Earth rotation has been neglected. Thus, system dynamics remain to:

$$\frac{d}{dt}\begin{bmatrix} h \\ V \\ \gamma \\ m \end{bmatrix} = \begin{bmatrix} V \cdot sin(\gamma) \\ (T - D - W sin(\gamma))/m \\ \left(\frac{L}{W} - cos(\gamma)\right) g/V \\ -\eta T \end{bmatrix} \tag{3.8}$$

The objective is minimize the time until a given height is reached, thus, the cost function is:

$$J = t_f$$

With boundary conditions:

$t_0 = 0$ [*s*]	$t_f = Objective$ [*s*]
$h(0) = 0$ [*ft*]	$h(t_f) = 65600$ [*ft*]
$V(0) = 424.260$ [*fps*]	$V(t_f) = 968.148$ [*fps*]
$\gamma(0) = 0$ [*rads*]	$\gamma(t_f) = 0$ [*rads*]
$W(0) = 42000$ [*lb*]	$W(t_f) = Free$ [*lb*]

Subjected to the inequality path constraints:

$$0 \leq h \leq 69000 ft$$

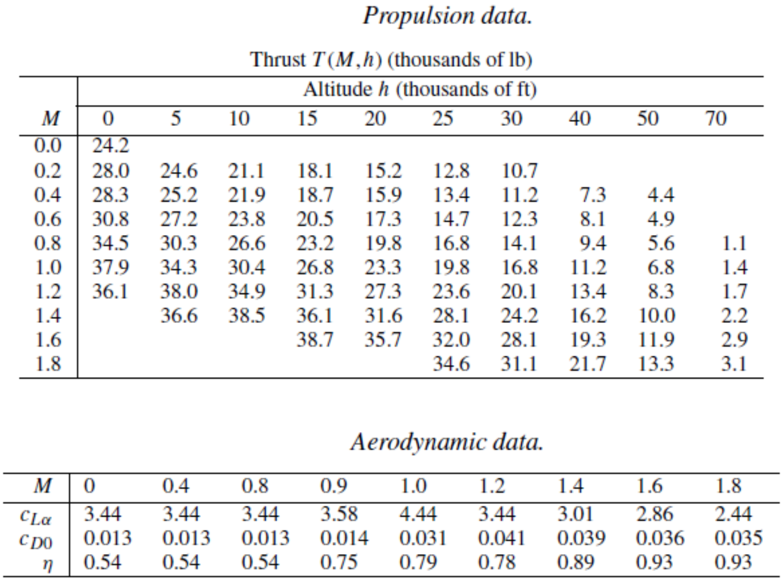
$$1 \leq V \leq 2000 fps$$

$$-89^\circ \leq \gamma \leq 89^\circ$$

$$0 \leq W \leq 45000 lb$$

$$-20^\circ \leq \alpha \leq 20^\circ$$

Below is presented the propulsive and aerodynamic data as stated in [5].



Propulsion data.

Thrust $T(M,h)$ (thousands of lb)

	Altitude h (thousands of ft)									
M	0	5	10	15	20	25	30	40	50	70
0.0	24.2									
0.2	28.0	24.6	21.1	18.1	15.2	12.8	10.7			
0.4	28.3	25.2	21.9	18.7	15.9	13.4	11.2	7.3	4.4	
0.6	30.8	27.2	23.8	20.5	17.3	14.7	12.3	8.1	4.9	
0.8	34.5	30.3	26.6	23.2	19.8	16.8	14.1	9.4	5.6	1.1
1.0	37.9	34.3	30.4	26.8	23.3	19.8	16.8	11.2	6.8	1.4
1.2	36.1	38.0	34.9	31.3	27.3	23.6	20.1	13.4	8.3	1.7
1.4		36.6	38.5	36.1	31.6	28.1	24.2	16.2	10.0	2.2
1.6				38.7	35.7	32.0	28.1	19.3	11.9	2.9
1.8						34.6	31.1	21.7	13.3	3.1

Aerodynamic data.

M	0	0.4	0.8	0.9	1.0	1.2	1.4	1.6	1.8
$c_{L\alpha}$	3.44	3.44	3.44	3.58	4.44	3.44	3.01	2.86	2.44
c_{D0}	0.013	0.013	0.013	0.014	0.031	0.041	0.039	0.036	0.035
η	0.54	0.54	0.54	0.75	0.79	0.78	0.89	0.93	0.93

Figure 19: Propulsion and aerodynamic data from [5].

The propulsive data is adjusted using a polynomial surface fitting which later will be constrained up to a maximum value. Below is expressed the surface-fitting:

$$T(h, M) \approx 25.05 - 0.5144h + 5.309M + 0.002132h^2 - 0.1774hM + 5.758M^2$$

Where h is the altitude, in kilo-feet, M is the Mach number, and the T is the thrust, expressed in kilo-pounds.

Regarding the aerodynamic data, since subsonic and supersonic theories use different equations, and both present an asymptotic behavior as approaching the sonic region (Singularity at Mach = 1), an analytic function emulating a step-function but being continuous and derivable over the whole domain is needed. That function is the hyperbolic tangent function, which manipulated appropriately, can fit reasonably well to the aerodynamic data for the parasitic and induced drag coefficients. The lift slope is approximated as a simple polynomial:

$$C_{D0}(M) \approx 0.013 + 0.0115 \cdot [1 + tanh(100 \cdot (M - 1))]$$

$$\eta(M) \approx 0.545 + 0.1925 \cdot [1 + tanh(5 \cdot (M - 1))]$$

$$C_{L_\alpha}(M) \approx 3.3856 + 0.3177M + 0.5481M^2 - 0.5834M^3$$

The initial guess is simple. The trajectory is linear (constant flight path angle) and the velocity is linearly interpolated between initial and final values stated in boundary conditions. The total time, t_F is assumed to be in the order of 350 seconds. The flight path angle is obtained from the average speed, the total height gain and the total time, and a value of $0.272 rads$ is obtained. Aircraft velocity times the sine of the flight path angle is the vertical speed, which integrated over time gives the altitude as a function of time. Regarding the mass, it is approximated as a linear decrease with time where the final mass is 85% of the initial mass. The angle of attack is assumed constant with a value of $2°$. This initial guess does not fulfill boundary conditions and system dynamics, but it is sufficiently accurate in order to reach a convergent solution.

$$h^*(t) = 0.3048 \cdot sin(\gamma^*) \cdot (424.26t + 0.5(968.148 - 424.26)t^2/t_F)$$
$$V^*(t) = 0.3048(424.26 + (968.148 - 424.26) \cdot t/t_F)$$
$$\gamma^*(t) = 0.272577$$
$$m^*(t) = m_0 - (m_0 - 0.85m_0) \cdot t/t_F$$
$$\alpha^*(t) = \pi/90$$

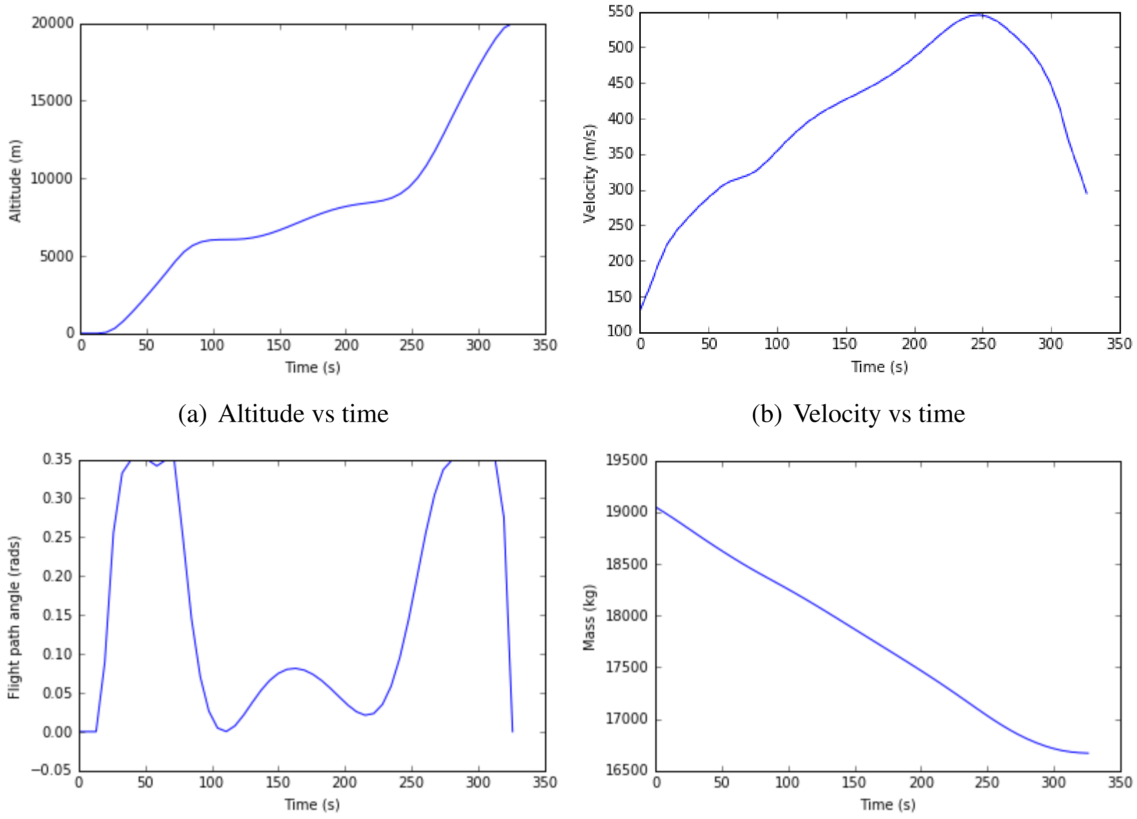
Plots of state and control variables are introduced in the next page.

The velocity keeps increasing all the climb flight until the plane has enough inertia to reach the final altitude and brake using drag and gravity to the desired final velocity.

It is interesting to remark the behavior of the flight path angle. First, it increases fastly up to a maximum angle of 20 degrees approximately, then sonic conditions are reached (Around 75 seconds) and the flight path angle is reduced abruptly (But always kept positive) and finally it increases again to the previous value. In the final part is reduced for the reason explained in the velocity profile. This non-intuitive behavior may be due to the asymptotic values of the aerodynamic and propulsive data, which changes suddenly when passing from subsonic to supersonic conditions.

With respect to mass, it has a linear profile until the braking phase is reached, where fuel consumption is reduced, and the angle of attack has approximately a constant value again until the braking phase, where it acquires negative values in order to reach the final state.

The final time obtained is 326 seconds, which is slightly less than 5 minutes and a half.



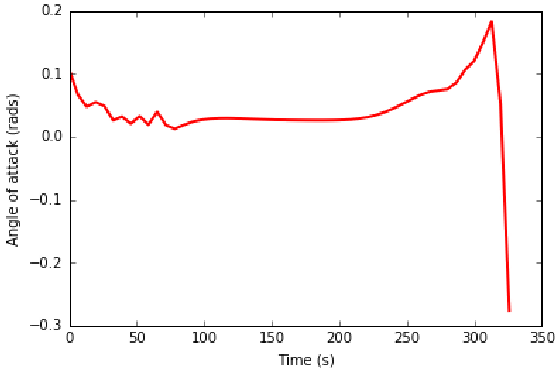
(a) Altitude vs time

(b) Velocity vs time

(c) Flight path angle vs time

(d) Mass vs time

Figure 20: State variables evolution of Supersonic climb problem



(a) Angle of attack vs time

Figure 21: Control variable evolution of Supersonic climb problem

4 Developed tests

In this section, more day-to-day problems of the aerospace field are presented. This set consists on four problems, one concerning the minimization of fuel consumption in cruise flight, the second concerns fuel consumption minimization during the climb phase, the third in an aircraft descent, and the final one is a combination of the previous ones, where the aircraft is supposed to flight a certain distance in cruise conditions at 11000 meters followed by a descend up to 10000 feet (FL100), where the ground distance is prefixed.

The multi-phase problem tries to recreate an actual flight profile of an aircraft from cruise to descent and apply the optimal control theory to it.

The aircraft studied on this problems will be an Airbus A-320, and it is modeled, as well as the dynamics of the system where the aircraft will be operating, with Eurocontrol's BADA file (3.11 Version). This document provides all the necessary data to characterize the aircraft. However, the script is prepared to intake any aircraft present in the BADA file and solve these problems with the specified aircraft data.

4.1 Aircraft model

As said before, the aicraft studied will be an Airbus A320, and it will be modeled according to BADA user manual [18]. The BADA file provides a series of data corresponding to airplane variables and must be used appropriately in order to have a good approximation of the airplane performance.

Since BADA offers information of the aircraft in all different phases of flight and configurations, for simplicity the aircraft is assumed to be in clean configuration (No flaps nor landing gear extended) and only the relevant corrections will be explained.

4.1.1 Atmospheric modeling

Ambient thermodynamic variables will depend only on the altitude according to ISA atmosphere. At sea level, the thermodynamic variables are assumed to be:

$T_0 = 288.15\ [K]$
$\rho_0 = 1.225\ [kg/m^3]$
$P_0 = 101325\ [Pa]$
$a_0 = 340.294\ [m/s]$

Temperature is set to decrease linearly with altitude proportional to the temperature gradient, λ, which corresponds with a decrease of $6.5C^\circ$ per kilometer of altitude. Thus, non-dimensionalizing the following expressions are obtained:

$$\theta(h) = \frac{T(h)}{T_0} = 1 - \frac{\lambda}{T_0}h \tag{4.1}$$

$$\sigma(h) = \frac{\rho(h)}{\rho_0} = \theta(h)^{4.2561} \tag{4.2}$$

$$\delta(h) = \frac{P(h)}{P_0} = \theta(h)^{5.2561} \tag{4.3}$$

$$a(h) = a_0\sqrt{\theta(h)} \tag{4.4}$$

These expressions are only valid below the tropopause (11000 m), but since this altitude is not going to be surpassed, there is no need in implementing a model for the stratosphere.

4.1.2 Mass and flight envelope

Aircraft operating speeds vary with the aircraft mass. The variation is calculated with the formula below:

$$V = V_{ref}\sqrt{\frac{m}{m_{ref}}} \tag{4.5}$$

Where the values with the subscript 'ref' means reference values.

This expression is useful to compute the stall speed for different aircraft masses, for example.

Aircraft maximum velocity is restricted in two different ways. First, it should not exceed the value 'V_{MO}', stated in the BADA file, expressed in calibrated airspeed and in knots (KCAS). Also, it must not exceed the maximum operating Mach number, 'M_{MO}', also specified in the BADA file.

Minimum speed of the aircraft is expressed in terms of stall speed:

$$V_{min} = C_{V,min}V_{stall} \tag{4.6}$$

Where $C_{V,min}$ is a safety factor. Note that stall velocity from BADA file must be corrected to the current aircraft mass and current flight conditions since it is expressed in calibrated airspeed.

BADA presents also limitations in altitude but are not relevant for this thesis.

4.1.3 Engine model

In this tesis only turbofan engines are being considered, although BADA also provides a model for turboprop engines.

$$(Thr_{maxclimb})_{ISA} = C_{T,c1}\left(1 - \frac{h}{C_{T,c2}} + C_{T,c3}h^2\right) \tag{4.7}$$

Where altitude is expressed in feet.

BADA offers a correction in thrust for temperature deviations from standard atmosphere, but this condition is not considered in this thesis.

In cruise conditions maximum thrust is expressed as follows:

$$(Thr_{cruise})_{max} = C_{T,cR} \cdot Thr_{maxclimb} \tag{4.8}$$

4.1.4 Fuel consumption model

In jet engines, the specific fuel consumption, η $[\frac{kg}{min \cdot kN}]$ is defined as a function of the true airspeed:

$$\eta = C_{f1}\left(1 + \frac{KTAS}{C_{f2}}\right) \tag{4.9}$$

Thus, the fuel mass flow is simply defined as:

$$\dot{m}_f = \eta T \tag{4.10}$$

Finally, a minimum fuel flow rate must be stated corresponding with engine idle, that is, the minimum thrust the engine can deliver:

$$\dot{m}_{f,min} = C_{f3}\left(1 - \frac{h}{C_{f4}}\right) \tag{4.11}$$

4.1.5 Aircraft data

Below are enlisted the aircraft data from the BADA file corresponding with the A-320 needed to solve the proposed problems. BADA file contains much more information but since that data is not going to be used it is not necessary to be included in the following list.

- $m_{ref} = 64000\ [kg]$
- $V_{max} = 350\ [KCAS]$
- $C_{f1} = 0.75882\ [kg/(min \cdot kN)]$
- $C_{f2} = 2938.5\ [kts]$
- $C_{D0,cruise} = 0.026659\ [-]$
- $V_{stall,cruise} = 140.5\ [KCAS]$
- $S = 122.6\ [m^2]$
- $C_{fcr} = 0.96358\ [-]$
- $M_{max} = 0.82\ [-]$
- $C_{t,c1} = 142310.0\ [N]$
- $C_{t,c2} = 51680.0\ [ft]$
- $C_{t,c3} = 5.6809 \cdot 10^{-11}\ [1/ft^2]$
- $C_{D2,cruise} = 0.038726\ [-]$

4.2 Cruise 1D

This simple problem consists on an aircraft flying at cruise conditions and constant altitude, and it is expected to travel a determined range within a RTA (Requested time of arrival, one hour in this problem) minimizing the fuel consumption.

A priori, the controls in this problem are the throttle setting and the lift coefficient, but since there exists the constraint of constant altitude, the lift coefficient will be forced to satisfy that condition, ceasing to be a free-control. Thus, the only control will be the throttle setting.

The dynamics of the system for cruise flight in 1D are:

$$\frac{d}{dt}\begin{bmatrix} x \\ V \\ m \end{bmatrix} = \begin{bmatrix} V \\ (T-D)/m \\ -\eta T \end{bmatrix} \tag{4.12}$$

Where $T = \pi T_{max}$, D keeps the usual formulation and the expressions for T_{max} and η are obtained from BADA.

The objective is to maximize the final mass, that is, minimize the cost functional:

$$J = -m(t_f)$$

Since relatively high oscillations are obtained in the throttle setting, the cost function is reformulated and a penalty in the throttle setting is introduced, in order to disencourage the oscillatory behavior. This oscillatory behavior is caused by a numerical phenomena in non-adaptive methods due to the size of the time-step at the start and end of the problem.

$$J = \int_{t_0}^{t_f} (\dot{m}_f + 0.0001\pi^2) \cdot dt$$

Where $\dot{m}_f$ is the mass flow of fuel and π is the throttle setting.

Below are presented the boundary conditions of the problem:

$t_0 = 0$ [*s*]	$t_f = 3600$ [*s*]
$x(0) = 0$ [*m*]	$x(t_f) = 420$ [*nm*]
$V(0) = 420$ [*kts*]	$V(t_f) = Free$ [*kts*]
$m(0) = 64000$ [*kg*]	$m(t_f) = Objective$ [*kg*]

The aircraft is subjected, as explained before, to the control constraint:

$$C_L = \frac{2mg}{\rho S V^2}$$

Apart from the constraints already established in the BADA file (Velocities, fuel flow...)

The initial guess is defined as the aircraft is flying at constant velocity, thus distance traveled is just velocity times time, mass evolution with time is computed analytically, and an expression involved arc tangents is obtained (Theoretical expression for mass evolution for cruise at constant velocity and altitude, varying C_L) and throttle setting is set to fulfill the thrust equals drag condition:

$$
\begin{aligned}
&x^*(t) = R_{cruise}/RTA \cdot t \\
&V^*(t) = R_{cruise}/RTA \\
&m^*(t) = \sqrt{A_1/B_1} \cdot \tan\left[\arctan(\sqrt{B_1/A_1} \cdot m_0) - \sqrt{A_1/B_1} \cdot t\right] \\
&\pi^*(t) = \left[A + B \cdot m^*(t)^2\right]/T_{max}(h_{cruise})
\end{aligned}
$$

Where:

$$
\begin{cases}
A = 1/2\rho S(cruise_{range}/RTA)^2 C_{D0} \\
B = 2kg^2/[\rho S(cruise_{range}/RTA)^2] \\
A_1 = \eta_{cruise} A \\
B_1 = \eta_{cruise} B
\end{cases}
$$

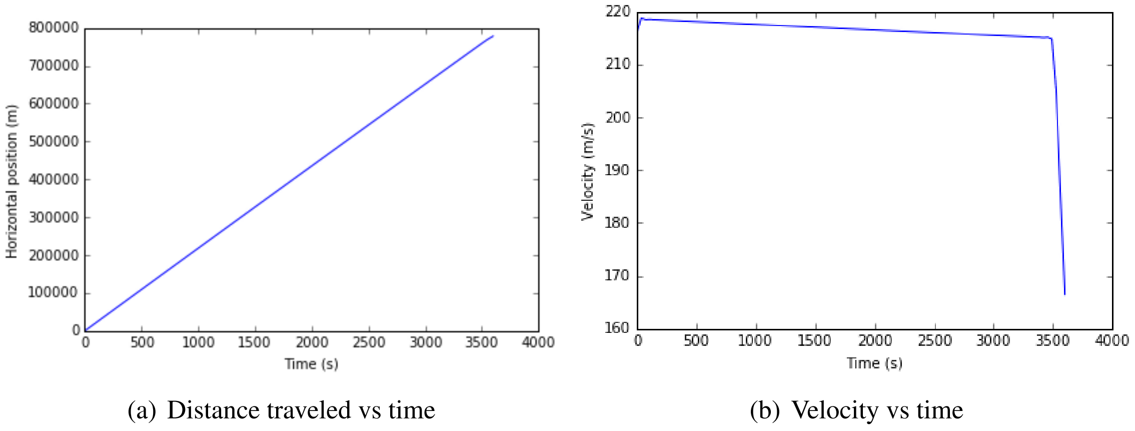
Evolution of state and control variables are presented in the next page.

The velocity is kept almost constant, slightly reducing over the cruise as fuel is being burned and airplane mass changing, and presents a drastic decrease at the end. Since velocity is almost constant, the evolution of the distance traveled over time is linear, as well as the mass decreasing due to fuel consumption.

Also, the throttle setting, as velocity, is almost constant, with a slight decrease over the overall part of the flight, and again, an abrupt decrease at the ending where the throttle setting goes to the minimum fuel consumption allowed. Clearly, the aircraft is flying at a given cruise speed until it has sufficient inertia to reach the final point with engines at idle conditions. Of course, this means a drop in the flight velocity, but since the aircraft is not required to finish with a certain speed, this technique is valid.

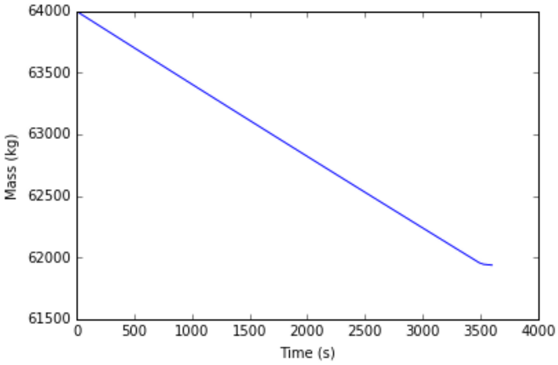
Theoretically, the aircraft should fly at its optimum velocity in order to minimize fuel consumption, but since final distance and time are fixed, aircraft should fly faster, stepping away from optimum conditions.

Flying at constant velocity, analytically the final mass can be obtained and corresponds with a value of 61897.38 kg. Applying optimal control, the final mass is 61937.67 kg. Thus, there has been a fuel economy of more than 40 kilograms, which in fact is not very significative. Also, flying at constant velocity means ending with this velocity, so in order to make a fair comparison, a boundary condition in the velocity should be imposed in the optimal control problem. In this case, savings drops to 0.087 kilograms, which is meaningless in 1 hour cruise. This is because theoretically, flying at constant airspeed corresponds with the optimum performance.



(a) Distance traveled vs time

(b) Velocity vs time



(c) Mass vs time

Figure 22: State variables evolution of 1D-cruise problem

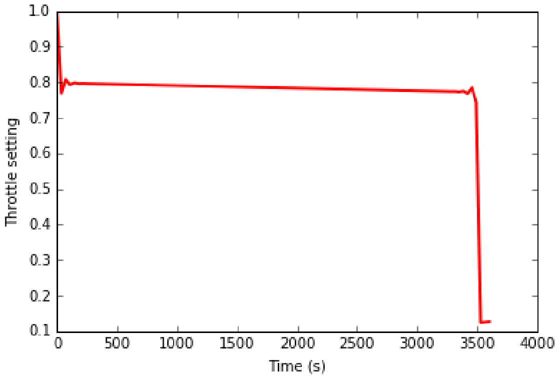


Figure 23: Control variable evolution of 1D-cruise

4.3 Climb

In aeronautical engineering, climb is the operation of altitude increasing of an aircraft. It is the phase following take-off and preceding cruise.

In this section the climb problem is stated as the aircraft, in clean configuration, departing from FL100 and finishing at 11000 meters (Tropopause).

System dynamics are similar as those used in Supersonic aircraft climb problem, where a Flat-Earth approximation is implemented:

$$\frac{d}{dt}\begin{bmatrix} x \\ h \\ V \\ \gamma \\ m \end{bmatrix} = \begin{bmatrix} V \cdot cos(\gamma) \\ V \cdot sin(\gamma) \\ (T - D - Wsin(\gamma))/m \\ \left(\frac{L}{W} - cos(\gamma)\right) g/V \\ -\eta T \end{bmatrix} \tag{4.13}$$

Respecting the boundary conditions, aircraft starts climb at an altitude of 10000 ft (At this altitude the aircraft is already in clean configuration), with a given flight speed and flight path angle, and must finish at 11000 meters with an specified velocity, totally prepared to start cruise:

$t_0 = 0$ [*s*]	$t_f = Free$ [*s*]
$x(0) = 0$ [*m*]	$x(t_f) = Free$ [*m*]
$h(0) = 10000$ [*ft*]	$h(t_f) = 11000$ [*m*]
$V(0) = 370$ [*kts*]	$V(t_f) = 420$ [*kts*]
$\gamma(0) = 2°$ [*rads*]	$\gamma(t_f) = Free$ [*rads*]
$m(0) = 64000$ [*kg*]	$m(t_f) = Objective$ [*kg*]

The initial guess is defined as the aircraft is flying at constant velocity, with constant flight path angle, so the variables velocity and gamma will be constant, and horizontal and vertical position will vary linearly with time. The mass is set to vary linearly with time also, and with respect the controls, lift coefficient is chosen constant, and throttle setting is expressed as the ratio of thrust and maximum thrust at an intermediate altitude:

$$\begin{aligned}
&x^*(t) = V^*cos(\gamma^*)t \\
&h^*(t) = h(0) + V^*sin(\gamma^*)t \\
&V^*(t) = V(0) \\
&\gamma^*(t) = 2^\circ \\
&m^*(t) = m(0) - \eta_{climb}A[C_{D0} + kC_L(0)^2]t \\
&C_L^*(t) = gcos(\gamma^*)m(0)/A \\
&\pi^*(t) = A(C_{D0} + kC_L(0)^2)/T_{max}(h_{avg})
\end{aligned}$$

Where:

$$\begin{cases} A = 1/2\rho_{avg}SV^{*2} \\ \rho_{avg} = \rho(h_{avg}) \\ h_{avg} = 20000ft \end{cases}$$

In this page is presented the climb path and the evolution in time of the controls of the problem, and next page shows the evolution in time of the state variables.

The control policy consists on setting the throttle to its maximum value and ascend as fast as possible, in order to reduce as much as possible the flying time, and thus, fuel consumption.

In this problem final distance is not enforced to be a given value, but if this was a requirment, the control policy again would be the throttle setting set to one and once the final altitude is reached, begin a cruise flight at optimum conditions. This will be seen later in the multiphase problem.

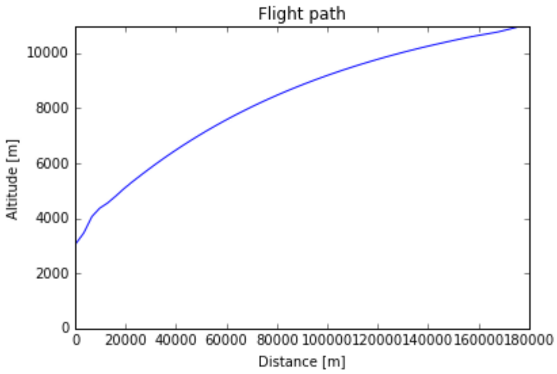
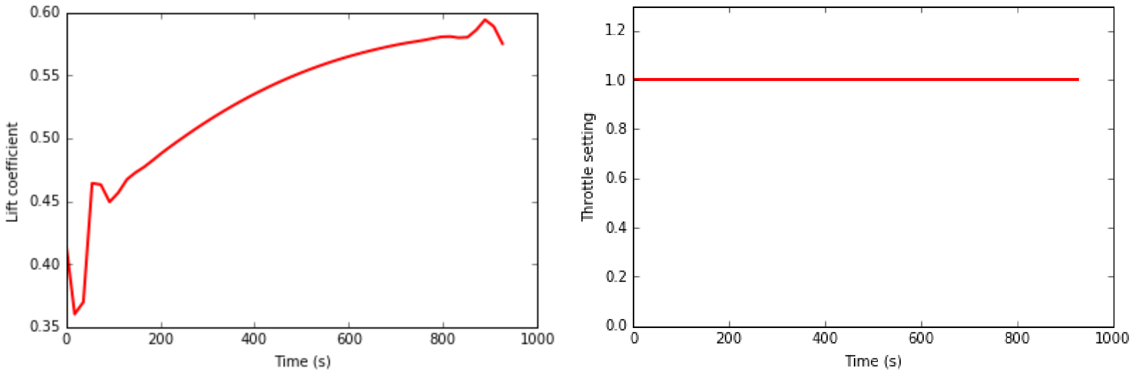


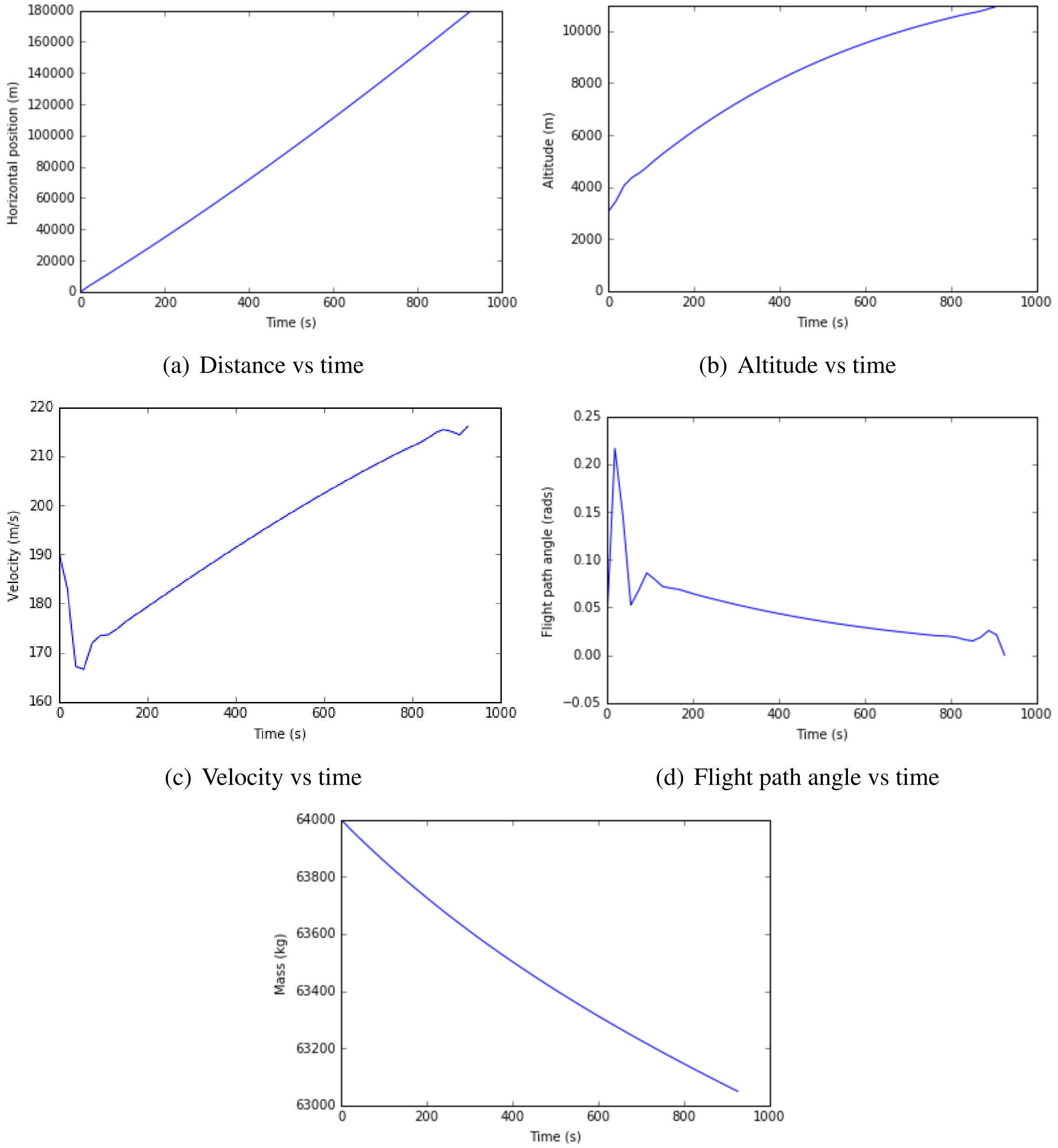
Figure 24: Trajectory of Climb problem



(a) Lift coefficient vs time

(b) Throttle setting vs time

Figure 25: Control variables evolution with time of Climb problem



(a) Distance vs time

(b) Altitude vs time

(c) Velocity vs time

(d) Flight path angle vs time

(e) Mass vs time

Figure 26: State variables evolution with time of Climb problem

4.4 Descent

This problem concerns about an aircraft descent from cruise to the flight level preliminary to the approach. In this tesis, the aircraft starts in cruise conditions at 11000 meters and will finish at 10000 feet. The objective is minimize fuel consumption during the descent.

The aircraft starts with an average cruise velocity for that kind of plane and must end up at 250KEAS due to ATM regulations. (This procedure is reasonably similar in real ATM airspace).

System dynamics are exactly the same as in the climb problem:

$$\frac{d}{dt}\begin{bmatrix} x \\ h \\ V \\ \gamma \\ m \end{bmatrix} = \begin{bmatrix} V \cdot cos(\gamma) \\ V \cdot sin(\gamma) \\ (T - D - W sin(\gamma))/m \\ \left(\frac{L}{W} - cos(\gamma)\right) g/V \\ -\eta T \end{bmatrix} \tag{4.14}$$

The boundary conditions are the following: Aircraft now begins at 11000m, with certain velocity and flight path angle, and must finish at 10000 ft, with an specified velocity, as opposed to climb problem. One major difference is that now, the distance traveled by the aircraft must be fixed, because if not, the aircraft will simply set engines to idle and descend with minimum fuel consumption, since final time is not required. This required distance is computed assuming a descend angle of 2 degrees. Thus, boundary conditions remain as follows:

$t_0 = 0$ [*s*]	$t_f = Free$ [*s*]
$x(0) = 0$ [*m*]	$x(t_f) = \Delta h/tan(2°)$ [*m*]
$h(0) = 11000$ [*m*]	$h(t_f) = 10000$ [*ft*]
$V(0) = 420$ [*kts*]	$V(t_f) = 250$ [*KEAS*]
$\gamma(0) = -2°$ [*rads*]	$\gamma(t_f) = Free$ [*rads*]
$m(0) = 64000$ [*kg*]	$m(t_f) = Objective$ [*kg*]

Since this problem presents the same dynamics as the climb problem but the only change is the value of the flight path angle, the initial guess taken is the same but with a negative γ:

$x^*(t) = V^* cos(\gamma^*)t$
$h^*(t) = h(0) + V^* sin(\gamma^*)t$
$V^*(t) = V(0)$
$\gamma^*(t) = -2°$
$m^*(t) = m(0) - \eta_{climb} A[C_{D0} + kC_L(0)^2]t$
$C_L^*(t) = gcos(\gamma^*)m(0)/A$
$\pi^*(t) = A(C_{D0} + kC_L(0)^2)/T_{max}(h_{avg})$

Where:

$$\begin{cases} A = 1/2\rho_{avg} S V^{*2} \\ \rho_{avg} = \rho(h_{avg}) \\ h_{avg} = 20000ft \end{cases}$$

In this page is presented the descent path and the evolution in time of the controls of the problem, and next page shows the evolution in time of the state variables.

The strategy followed by the aircraft consists on setting engines to idle as soon as possible, and descend gliding with minimum fuel consumption and at the final point perform an acceleration in order to finish with the required velocity.

In problems where the required distance was much larger, the control policy was the symmetric to the climb problem. The aircraft kept the initial altitude, flying at cruise conditions at that altitude, where drag is much smaller, until it was possible to descend at idle with minimum fuel consumption and reaching the target distance.

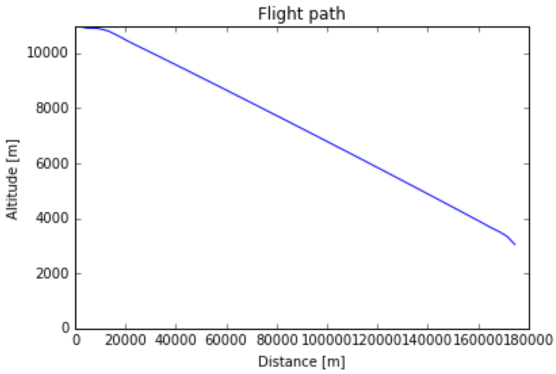
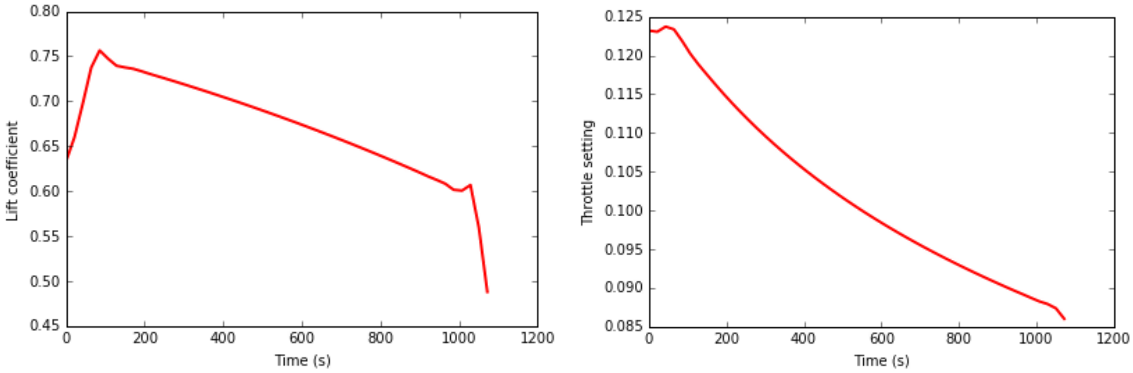


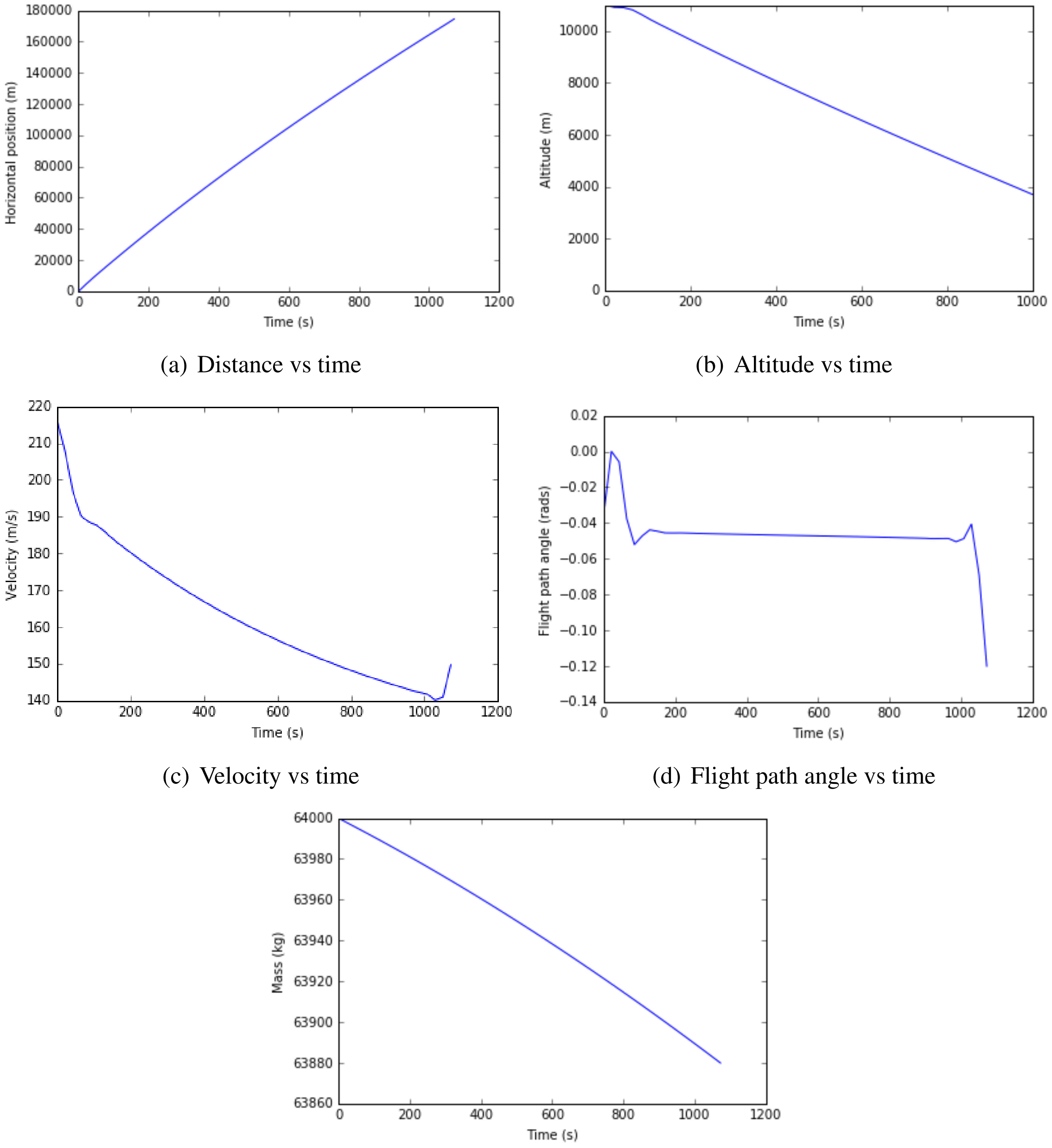
Figure 27: Trajectory of Descent problem



(a) Lift coefficient vs time

(b) Throttle setting vs time

Figure 28: Control variables evolution with time of Descent problem



(a) Distance vs time

(b) Altitude vs time

(c) Velocity vs time

(d) Flight path angle vs time

(e) Mass vs time

Figure 29: State variables evolution with time of Descent problem

4.5 Multi-phase entire flight simulation

This problem consists on three different phases: First, climb from 10000ft up to 11000 m, followed by a cruise, consisting on 2000 nautical miles, and finishing with a descent from 11000m to 10000 ft. The particularity of this problem is that it is multiphase, that is, the three problems must be optimally solved at the same time, not independently (The final state of the optimal solution of the first phase may penalize the second one obtaining a non-optimal global solution).

System dynamics are the same as in previous problems, that is, the longitudinal equations of motion of an aircraft:

$$\frac{d}{dt}\begin{bmatrix} x \\ h \\ V \\ \gamma \\ m \end{bmatrix} = \begin{bmatrix} V \cdot cos(\gamma) \\ V \cdot sin(\gamma) \\ (T - D - Wsin(\gamma))/m \\ \left(\frac{L}{W} - cos(\gamma)\right) g/V \\ -\eta T \end{bmatrix} \tag{4.15}$$

Regarding boundary conditions, the initial ones correspond with the initial ones from climb problem, and the final ones, with the final ones of the descent problem. Also, intermediate boundary conditions must be introduced, which are responsible for determining when a phase ends and the next one starts. In this problems there are two inter-phases, so intermediate boundary conditions must be defined two times. The first phase ends when an altitude of 11000m is reached, and also there is imposed a final flight path angle of 0 degrees in order to avoid discontinuities between one phase and the next one (Remember in cruise the flight path angle is 0), and the second phase ends when a distance equivalent to the cruise range plus the distance traveled during an standard climb, approximated as climbing at 2 degrees with constant velocity, is reached.

$t_0 = 0$ [*s*]	$t_f = Free$ [*s*]
$x(0) = 0$ [*m*]	$x(t_f) = R_{climb} + R_{cruise} + R_{descent} = 2246$ [*nm*]
$h(0) = 10000$ [*ft*]	$h(t_f) = 10000$ [*ft*]
$V(0) = 370$ [*kts*]	$V(t_f) = 250$ [*KEAS*]
$\gamma(0) = 2$ [*degrees*]	$\gamma(t_f) = Free$ [*rads*]
$m(0) = 64000$ [*kg*]	$m(t_f) = Objective$ [*kg*]

Conditions at the inter-phases:

$h(t_1) = 11000$ [*m*]	$x(t_2) = R_{climb} + R_{cruise} = 2123$ [*nm*]
$\gamma(t_1) = 0$ [*rads*]	

The cost function has contributions of all the different phases of the flight:

$$J = \int_{t_0}^{t_1} (\dot{m}_f + 0.01\pi^2) \cdot dt + \int_{t_1}^{t_2} (\dot{m}_f + 0.01\pi^2) \cdot dt + \int_{t_2}^{t_f} (\dot{m}_f + 0.01\pi^2) \cdot dt$$

Where t_1 is the time where climb ends and cruise starts and t_2 is the time where cruise ends and descent starts. It can be seen that the objective is minimize the overall fuel mass flow. Again, a penalty is introduced in the throttle setting in order to eliminate oscillations.

The restrictions are those specified in the BADA file and apart, flight path angle has been constrained to be within -5 and 5 degrees in order to avoid too steep performances, impractical in commercial aviation.

With respect to the initial guess, since the problem is divided in three phases, three different initial guesses must be proposed, one for each phase. Since each problem have been solved indidually before, the initial guesses are reused but treating accordingly the value of the state variables:

For $0 < t < t_1$:

$x^*(t) = V^* \cos(\gamma^*)t$
$h^*(t) = h(0) + V^* \sin(\gamma^*)t$
$V^*(t) = V(0)$
$\gamma^*(t) = -2^\circ$
$m^*(t) = m(0) - \eta_{climb}A[C_{D0} + kC_L(0)^2]t$
$C_L^*(t) = g\cos(\gamma^*)m(0)/A$
$\pi^*(t) = A(C_{D0} + kC_L(0)^2)/T_{max}(h_{avg})$

For $t_1 < t < t_2$:

$x^*(t) = R_{climb} + V^*(t - t_1)$
$h^*(t) = h_{cruise}$
$V^*(t) = V(t_1)$
$\gamma^*(t) = 0$
$m^*(t) = m(t_1) - \eta_{cruise}A[C_{D0} + kC_L(t_1)^2](t - t_1)$
$C_L^*(t) = m(t_1)g/A$
$\pi^*(t) = A(C_{D0} + kC_L(t_1)^2)/T_{max}(h_{cruise})$

For $t_2 < t < t_f$:

$x^*(t) = R_{climb} + R_{cruise} + V^* \cos(\gamma^*)(t - t_2)$
$h^*(t) = h_{cruise} + V^* \sin(\gamma^*)(t - t_2)$
$V^*(t) = V(t_2)$
$\gamma^*(t) = -2^\circ$
$m^*(t) = m(t_2) - \eta_{descent}A[C_{D0} + kC_L(t_2)^2](t - t_2)$
$C_L^*(t) = gcos(\gamma^*)m(t_2)/A$
$\pi^*(t) = A(C_{D0} + kC_L(t_2)^2)/T_{max}(h_{avg})$

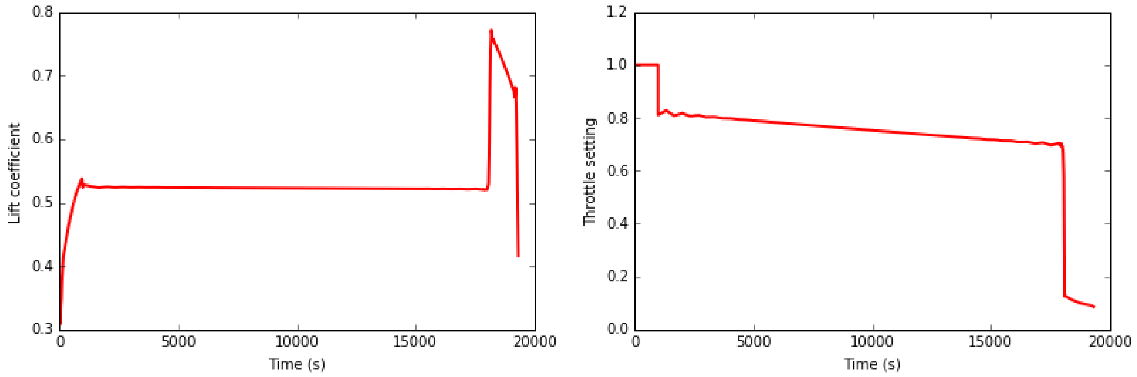
Where:

$$\begin{cases} A = 1/2\rho_{avg}SV^{*2} \\ h_{avg} = 20000ft \\ h_{cruise} = 11000m \end{cases}$$

In this page is presented the flight path and the evolution in time of the controls of the problem, and next page shows the evolution in time of the state variables.

As explained in the climb problem, in this case the control policy is the same. It can be seen that throttle setting is set to maximum until the final height is reached, and keeps cruise conditions with the throttle setting at a lower value until the phase-change occurs.

With respect to descent, aircraft descends first with engines close to idle, and then at idle conditions, with a flight path angle remaining nearly constant over the descend, with a value of 5 degrees, that is the maximum allowed in this descend.



(a) Lift coefficient vs time

(b) Throttle setting vs time

Figure 30: Control variables evolution with time of Multiphase problem

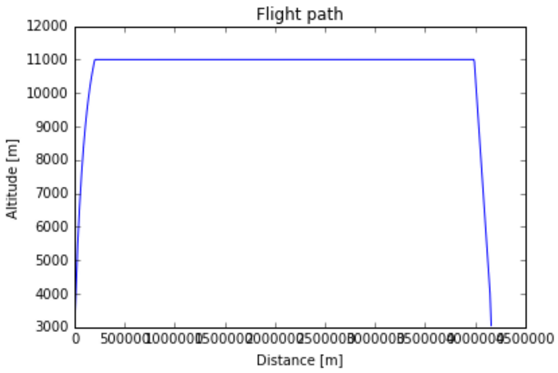
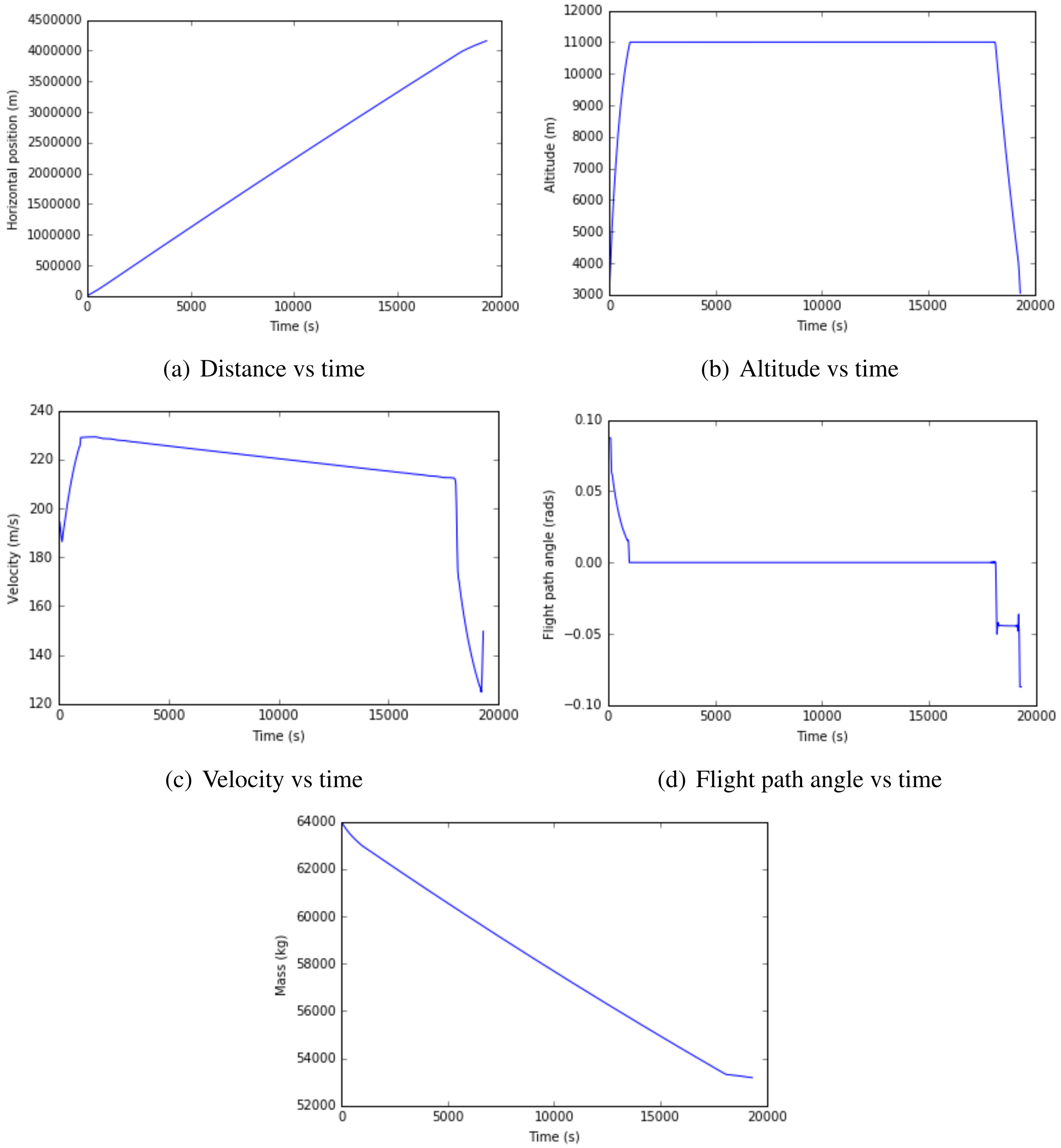


Figure 31: Trajectory of Multiphase problem



(a) Distance vs time

(b) Altitude vs time

(c) Velocity vs time

(d) Flight path angle vs time

(e) Mass vs time

Figure 32: State variables evolution with time of Multiphase problem

5 Conclusions

One conclusion is the great advantages of direct methods, due to its ease of implementation and robustness. A great knowledge in optimal control is not required and thanks to its robustness, the initial guess may not be so accurate or even infeasible and complex problems can be treated with these methods.

One disadvantage of the trapezoidal method is the tendency to present oscillations in the variables, and penalties must be introduced, modifying the desired objective function, although the penalties introduced did not exert much influence in the solution.

In collocation methods, trapezoidal collocation is one of the less accurate ones, since it is just of order 2 in the Gauss-Lobatto integration rules, and higher orders such as Hermite-Simpson would present more accurate solutions. Nevertheless, for simple problems where a rapid convergence is reached, trapezoidal collocation is sufficiently good and precise solutions can be obtained. The problem is when dealing with systems with a high number of variables and interrelated between them in a complex system of differential equations, where accuracy is reduced and it becomes necessary to consider other collocation methods, if a very accurate solution is desired.

Evaluating the canonical problems, it can be stated that the results obtained with the used algorithm were satisfactory, since they were compared with literature results and were similar.

5.1 Future work

With respect to the canonical tests, in this thesis they have been solved using only 2nd order Gauss-Lobatto quadratures, so an important step could be the implementation of higher order methods, as well as other numerical methods, such as the pseudo-spectral. The algorithm allows the use of other numerical methods which can be used to solve the problems. Currently, adaptive methods are being developed and will be implemented in the library in the future.

Another possible task is the implementation in the algorithm the ability to work with tabular data, since many aircraft parameters are specified in this form and might facilitate the user's experience as well as improve the solution's quality, since up to now the only method of dealing with it is creating an analytical fitting function to represent fairly the data inside the tables (See Supersonic climb problem in Section 4.7). Other possible improvement could be the including and use of the time variable within the equations of the dynamic system, since there might be functions which depend explicitly on this variable. In this thesis this problem occurred in the orbit raising problem, and a new state variable emulating time (In the dynamic system, just setting the time derivative equal to one) was implemented, so this improvement also would facilitate user's experience.

Regarding the new tests, complexity may be increased, for example adding 3D trajectories, studying the inertial forces effects of our spherical and rotating planet, as well as introducing the effect of wind, which is a major factor in the design of an aircraft trajectory and changes completely the control policies in order to achieve an optimum solution. Other possible improvement could be the development of a model simulating ATM restrictions, delays... in order to obtain a more approximate model of a real commercial flight.

6 Project management

6.1 Project planning

The project planning consisted on 6 different parts: Familiarization, tests selection, programming, testing, thesis writing and thesis revision.

First part, familiarization, consisted on the reading of papers and other literature concerning optimal control theory and related topics necessary for the construction of the theoretical basis applied on this thesis. Also, another task of this block consisted on learning *Python* programming language, as well as the installation and setting up of the software workspace.

Second part, selection, covered an exhaustive reading of literature in order to extract the candidate problems to be included in this thesis and implemented in the library.

Third part dealt with the programming of this tests into *Python*.

Fourth part, testing, consisted about solving the problems and obtaining results.

The fifth part covers the thesis writing.

Last part consisted on the revision of the writing and last-minute changes.

Project is assumed to start on February 15 and end on June 20.

Below is presented a comparative Gantt diagram including the original planning and the actual planning.

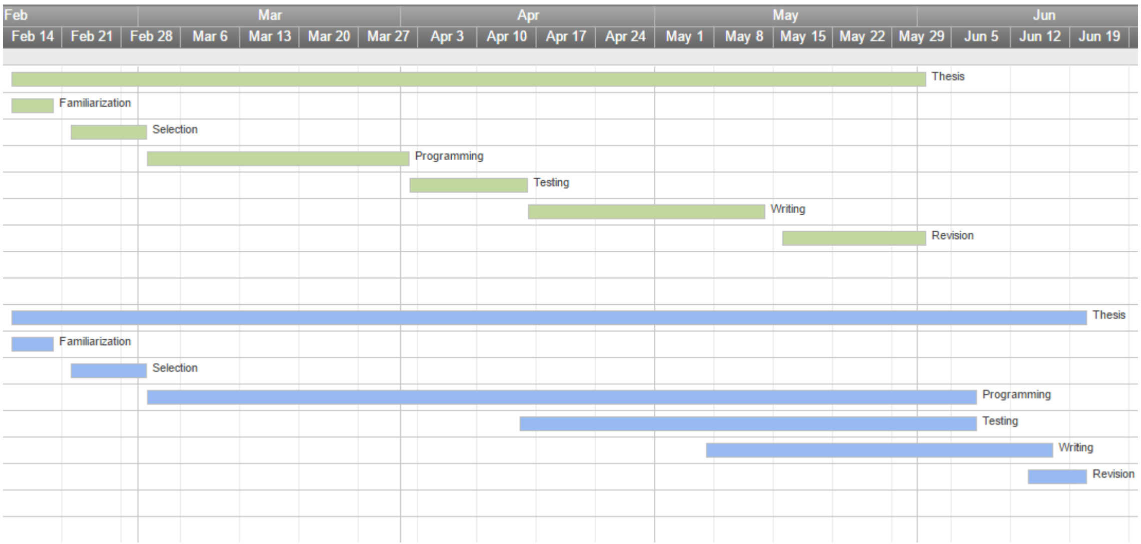


Figure 33: Gantt diagram comparison

In green are presented the activities from the original project schedule and in blue the actual dates of the development of these activities.

Differences between original and actual plans can be observed. First problem was the selection of suitable initial guesses, since in many case the ones selected did not work and certain problems could not be solved, apart from other errors committed during scripts coding. The second issue was the inclusion of additional problems which were considered of relevant interest for this thesis, since originally only 1D cruise and descent were selected, but climb and multi-phase problems were included, wit the additional difficulties inherent to multi-phase problems. The third issue was the author's final exams in May which delayed the progression of the thesis writing and programming of the newly included problems. Finally, the real project planning did not present a linear progression since many activities were performed at the same time, but at the end the project could be concluded on the specified dates.

6.2 Regulatory framework

This project can be categorized as a software development project.

Since the software used, *Spyder*, an integrated development environment for scientific programming on *Python* language, enters within the definition of free and open source software, the regulations arising this kind of software are explained below.

Free and open source software can be classified as both free software and open source software. This means that '*anyone is freely licensed to use, copy, study, and change the software in any way, and the source code is openly shared so that people are encouraged to voluntarily improve the design of the software*' (Extracted from the Free Software Foundation web page). This is in contrast to proprietary software, where the source code is usually hidden from the users.

Note that free software is not similar to freeware, the first one enable the community to access the source code while the second one is just free, although the software could be hidden to the user.

The library as well as the scripts developed in this thesis will be published under free license.

Thus, no restrictions apply to the development of this project.

6.3 Socio-economic framework

In this project, cost is composed by three contributors: Software costs, hardware costs and personnel costs.

The software used is *Python 3.5.1*. The main advantage of this software, is that is free and open source software, that is, it is open to the whole community. Also, there is no cost in acquiring it. The use of this software is more and more frequent in the scientific community due to its free condition and also because of the community surrounding it, where users can download and improve other users' work.

Below is introduced a table containing the software costs:

Software	Unitary cost (€)	Unities	Total cost (€)
Python 3.5.1	0	1	0

Table 1: Software costs

The hardware used in the development of this thesis was the author's personal computer. The model is '*ASUS N750JV*' and its market price was 1100 €.

Hardware	Unitary cost (€)	Unities	Total cost (€)
Personal computer	1100	1	1100

Table 2: Hardware costs

Personnel costs are calculated assuming an hourly pay for a software developer of 32 €per hour. The number of hours employed is 250.

Below is presented a table with the disaggregated personnel costs.

Personnel	Hourly salary (€)	Hours employed	Total cost (€)
Aerospace Engineer	32	250	8000

Table 3: Personnel costs

Thus, total cost ascends to 9100 €.

References

[1] Pesch, H. J., & Plail, M. (2012). The Cold War and the Maximum Principle of Optimal Control. *Optimization Stories. Documenta Mathematica.*

[2] SESAR Consortium. (2012). European ATM master plan.

[3] Larson, W. J., & Wertz, J. R. (1992). *Space mission analysis and design* (No. DOE/NE/32145–T1). Microcosm, Inc., Torrance, CA (US).

[4] Glassmeier, K. H., Boehnhardt, H., Koschny, D., Khrt, E., & Richter, I. (2007). The Rosetta mission: flying towards the origin of the solar system. *Space Science Reviews, 128*(1-4), 1-21.

[5] Betts, J. T. (2010). *Practical methods for optimal control and estimation using non linear programming* (Vol. 19). Siam.

[6] Rao, A. V. (2009). A survey of numerical methods for optimal control. *Advances in the Astronautical Sciences*, 135(1), 497-528.

[7] Bertsekas, D. P., & Bertsekas, D. P. (1995). *Dynamic programming and optimal control* (Vol. 1, No. 2). Belmont, MA: Athena Scientific.

[8] Pontryagin, L. S. (1987). *Mathematical theory of optimal processes*. CRC Press.

[9] Wächter, A., & Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1), 25-57.

[10] Betts, J. T. (1998). Survey of numerical methods for trajectory optimization. *Journal of guidance, control, and dynamics, 21*(2), 193-207.

[11] Garg, D., Patterson, M., Hager, W. W., Rao, A. V., Benson, D. A., & Huntington, G. T. (2010). A unified framework for the numerical solution of optimal control problems using pseudospectral methods. *Automatica*, 46(11), 1843-1851.

[12] Gill, P. E., Murray, W., & Wright, M. H. (1981). *Practical optimization.*

[13] Hart, W. E., Laird, C., Watson, J. P., & Woodruff, D. L. (2012). *Pyomooptimization modeling in python* (Vol. 67). Springer Science & Business Media.

[14] Darby, C. L., Hager, W. W., & Rao, A. V. (2011). An hp-adaptive pseudospectral method for solving optimal control problems. *Optimal Control Applications and Methods, 32*(4), 476-502.

[15] Patterson, M. (2013). *Efficient solutions to nonlinear optimal control problems using adaptive mesh orthogonal collocation methods*. University of Florida.

[16] García-Heras, J., Soler, M., & Sáez, F. J. *Collocation methods to minimum-fuel trajectory problems with required time of arrival in ATM.*

[17] Soler, M. (2013). *Comercial aircraft trajectory planning based on multiphase mixed-integer optimal control*. Omm Editorial.

[18] Eurocontrol. (2013). *User manual for the base of aircraft data (BADA)*. Revision 3.11. *http://www.eurocontrol.int/services/bada/*