Universidad Nacional de Educación a Distancia

(UNED)

E.T.S. Ingeniería Informática

Dpto. Informática y Automática

TESIS DOCTORAL

INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

Application of Intelligent Algorithms to Aerospace Problems

Fernando Alonso Zotes

Ingeniero Industrial

2011

Universidad Nacional de Educación a Distancia

(UNED)

E.T.S. Ingeniería Informática

Dpto. Informática y Automática

TESIS DOCTORAL

INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

Application of Intelligent Algorithms to Aerospace Problems

Author:

Fernando Alonso Zotes

Ingeniero Industrial (esp. Electrónica y Automática)

Universidad Carlos III de Madrid

Supervisor:

Matilde Santos Peñas

Dpto. Arquitectura de Computadores y Automática

Facultad de Informática, Universidad Complutense de Madrid

2011

Es mi deseo agradecer el apoyo prestado por mi familia durante el tiempo que han durado mis estudios.

Asimismo, quiero agradecer a mi directora Matilde Santos su ayuda desde que inicié mis estudios de doctorado, ya hace unos cuantos años, por compartir su interés por la inteligencia artificial y la automática, sus buenos consejos y sin quien los artículos llevados a congresos o publicados en revistas no habrían visto la luz.

También quiero agradecer al Dpto de Informática y Automática de la UNED el haberme brindado la oportunidad de haber realizado mis estudios de doctorado, compaginándolo con mis obligaciones laborales en Reino Unido y Alemania.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AI	Artificial Intelligence
AU	Astronomical Unit
CM	Centre of Mass
DEA	Diploma de Estudios Avanzados
DSM	Deep Space Manoeuvre
ESA	European Space Agency
GA	Genetic Algorithm
JPL	Jet Propulsion Laboratory
LEO	Low Earth Orbit
LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
MOPSO	Multi Output Particle Swarm Optimisation
NASA	Nacional Aeronautics and Space Administration
P controller	Proportional controller
PD controller	Proportional Derivative controller
PID controller	Proportional Integral Derivative controller
PSO	Particle Swarm Optimisation
TOF	Time of Flight
UCM	Universidad Complutense de Madrid
UNED	Universidad Nacional de Educación a Distancia

Chapter 1
Introduction

The purpose of this chapter is to explain the objectives and structure of the PhD thesis. This thesis is the culmination of the PhD studies at the Universidad Nacional de Educación a Distancia (UNED). An overview of the structure of the PhD studies is also given in this chapter.

1.1 Motivation

Aerospace missions often involve the optimisation and solution of complex problems, where the modelling of multivariable systems leads to sets of equations without analytical solution. While the power of modern computers has increased over the last years, and further development are foreseen in the next decades, the complexity of aerospace missions and its solutions cannot rely purely in the computational power of modern devices, as this is unaffordable by several orders of magnitude, surpassing the computational capabilities of any machine today or even in the near future. In particular, splitting each input variable of a complex aerospace system into a fine grid, and subsequently analysing any possible combination inside that grid in order to find the best possible solution, would lead to spending huge amounts of time, if the solution can ever be reached at all.

Therefore, some intelligent approach of this kind of complex problems, involving aerospace systems, is mandatory and definitely more efficient. Such heuristic approach should be based on the intelligent searching of solutions, often involving time-consuming computations but less demanding than an analytical and exhaustive exploration of the space of solutions.

The basic idea behind the application of computational intelligence is similar to that made by a human being, whose expertise and skills when designing and solving new problems cannot be replaced by any algorithm. An effort has been done, thought, in that direction, this is, to develop intelligent computational algorithms that may emulate (often quite successfully) some of the skills and experiences of human experts. Ideally, an artificial intelligent (AI) algorithm should combine the flexibility of human knowledge when challenged by new problems, with the power and speed of machines. Even if currently such ideal combination is far from reality, undoubtedly major steps have been made in the right direction. In particular, regarding aerospace problems, some areas of mission design and analysis may rely on automatic optimisation to provide sometimes a definitive optimal solution or, more commonly, to give an initial nearly-

optimal solution or set of solutions that can be further refined and tuned by scientists and engineers. Some examples of the state-of-the-art are provided in section 2.1.

Three of the most interesting aerospace problems today are: optimisation and control of trajectories around Lagrange points; optimisation of launch trajectories; and optimisation of interplanetary trajectories, involving fly-bys in several planets in the solar system.

1.2 Objectives

The general goal of this PhD thesis is to describe the research and discuss the conclusions drawn on the application of Artificial Intelligence (AI) to aerospace problems, covering related topics such as launch trajectory optimisation, control of satellites around the Earth-Moon Lagrange points, and optimisation of interplanetary trajectories with gravitational assistances. Further information can be found on the articles by F. Alonso (author of this thesis) and M. Santos (supervisor of this thesis), published in several conferences and scientific journals and listed in section 1.4.

1.3 Structure of this Document

This PhD thesis is organized as follows.

Chapter 1 describes the objectives, goals and structure of this PhD thesis.

Chapter 2 provides an overview of general aerospace and aerodynamics concepts, needed in order to understand this PhD thesis, and the state of the art.

Chapter 3 the application of fuzzy logic to the modelling, simulation and controlling of spacecrafts in the surroundings of Lagrange points are presented.

In Chapter 4 the usage of genetic algorithms (GA) is discussed in the context of the optimisation of trajectories of launchers.

Chapter 5 describes the optimisation of interplanetary trajectories, using fly-bys, with genetic algorithms (GA) and Particle Swarm Optimisation (PSO) and Multi-Output Particle Swarm Optimisation (MOPSO). Results are compared and discussed.

This thesis ends with Chapter 6, where the conclusions and future work on artificial intelligence applied to aerospace are presented.

Finally, at the end of the thesis a list of references is given.

1.4 List of Publications

1.4.1 Articles in Journals

Autores: F. Alonso Zotes, M. Santos

Título: Particle swarm optimisation of an interplanetary trajectory from Earth to Jupiter

Revista: *Engineering Application of Artificial Intelligence (EAAI).* Pergamon-Elsevier Science. Oxford, UK. ISSN: 0952-1976

Volumen, pág. (año): in press, 2011

Calidad: JCR 2010: 1.344

(2010) Engineering Multidisciplinary 21/87 (Q1)

Autores: F. Alonso Zotes, M. Santos

Título: Modelado, simulación y control de satélites situados en los puntos de Lagrange del sistema Tierra-Luna

Revista: *Revista Iberoamericana de Automática e Informática Industrial (RIAI).* Comité Español de Automática CEA-IFAC. ISSN: 1697-7912

Volumen, pág. (año): vol. 8, 204-215, 2011

Calidad: JCR 2010: 0.195

(2010) Automation & Control Systems 56/60 (Q4)

Autores: F. Alonso Zotes**,** M. Santos

Título Multi-criteria Genetic Optimisation of the Manoeuvres of a Two-Stage Launcher

Revista: *Information Science*s.

Elsevier Science Inc. NY, United States. ISSN: 0020-0255

Volumen, pág. (año): 180(6), 896-910, 2010

Calidad: JCR 2010: 2.833

(2010) Computer Science, Information Systems 9/126 (Q1)

1.4.2 Book Chapters

Autores: F. Alonso Zotes, M. Santos

Título libro: *Computational Intelligence. Foundations and Applications*

Capítulo: Genetic optimization of an interplanetary trajectory from Earth to Jupiter

Editor(es): D. Ruan, T. Li, Y. Xu, G. Chen, E. Kerre

Editorial (año): World Scientific, 1048-1053, 2010

ISBN: 978-981-4324-69-4

Autores: F. Alonso Zotes, M. Santos Peñas

Título libro: *Computational Intelligence in Decision and Control*

Capítulo: GA Optimization of the height of a low earth orbit

Editor(es): D. Ruan, J. Montero, J. Lu, L. Martínez, P. D'hondt, E. E. Kerre

Editorial (año): World Scientific, 719-724, 2008

ISBN: 978-981-279-946-3

1.4.3 Conferences

2010 IEEE ISKE International Conference on Intelligent System and Knowledge Engineering

Authors: F. Alonso, M. Santos

Title: Intelligent satellites control based on fuzzy logic in the Earth-Moon libration points

Date, location: 15-16/11/2010, Hangzhou, China

Journal: Proceedings of *2010 IEEE Int. Conference on Intelligent System and Knowledge Engineering ISKE 2010*, IEEE Press, ISBN: 978-1-4244-6792

Pages, year: 605-610, 2010

CORE: A

2010 IEEE World Congress on Computational Intelligence WCCI (IEEE CEC 2010)

Authors: F. Alonso Zotes, M. Santos

Title: Delta-V Genetic Optimization of a Trajectory from Earth to Saturn with Fly-by in Mars

Date, location: 18-23/07/2010, Barcelona, España.

Journal: Conference Proceedings of WCCI 2010, ISBN: 978-1-4244-6910-9

Pages, year: 1836-1841, 2010

CORE: B

ICATT 2010, 4th International Conference on Astrodynamics Tools and Techniques

Authors: F. Alonso Zotes, Matilde Santos Peñas

Title: From Earth to Kuiper belt: swarm optimisation algorithm applied to interplanetary missions

Date, location: Madrid, Spain, 3-6/05/2010.

Journal: Proceedings WPP-308, 4th International Conference on Astrodynamics Tools and Techniques

Year: 2010.

The 9th International FLINS Conference on Foundations and Applications of Computational Intelligence, 2010

Authors: F. Alonso Zotes, Matilde Santos Peñas

Title: Genetic Optimization of an Interplanetary Trajectory from Earth to Jupiter.

Date, location: Chengdu, China, 2-4/08/2010.

Journal: Computational Intelligence. Foundations and Applications. World Scientific, pp. 1048-1053, 2010. ISBN: 978-981-4324-69-4, D. Ruan, T. Li, Y. Xu, G. Chen, E. Kerre

Year: 2010.

The 8th International FLINS Conference on Computational Intelligence in Decision and Control, 2008

Authors: F. Alonso Zotes, Matilde Santos Peñas

Title: Optimization of the height of a Low Earth Orbit.

Date, location: Madrid, Spain, 21-24/09/2008.

Journal: Computational Intelligence in Decision and Control. World Scientific, pp. 719-724. 2008. ISBN: 978-981-279-946-3, D. Ruan, J. Montero, J. Lu, L. Martínez, P. D'hondt, E. E. Kerre

Year: 2008

JJAA 2008, XXIX Jornadas de Automática

Authors: F. Alonso-Zotes, M. Santos

Title: Modelado y simulación con Modelica y Dymola de cinco satélites situados en los puntos de Lagrange del sistema tierra-luna

Date, location: Universitat Rovira y Virgili, Tarragona, España, 3-5/09/2008.

Journal: Actas del Congreso. Ed. Universitat Rovira i Virgili,ISBN 978-84-691-6883-7

Year: 2008

JJAA 2007, XXVIII Jornadas de Automática

Authors: F. Alonso, M. Santos

Title: Optimización de lanzamiento vertical de dos fases utilizando un algoritmo genético multicriterio

Date, location: Huelva, España, 5-7/09/2007.

Journal: Actas del Congreso (CD-ROM). ISBN: 978-84-690-7497-8

Year: 2007.

2nd International Symposium on Formation Flying Missions and Technologies

Also presented in: 2nd ESA Workshop on Astrodynamics Tools and Techniques, *September 2004, ESTEC, Noordwijk, The Netherlands*

Authors: Bengoa, G.; Alonso, F.; García, D.; Graziano, M.; Beech, T.; Ortega, G

Title: FAMOS-V2: Formation Flying and Rendezvous and Docking Tool for Exploration Mission in Circular and Elliptical Orbits.

Date, location: September 2004, Washington D.C., USA

Journal: Proceedings 2nd International Symposium on Formation Flying Missions and Technologies

Year: 2004.

2nd ESA Workshop on Astrodynamics Tools and Techniques

Authors: Bengoa, G.; Alonso, F.; García, D.; Graziano, M.; Beech, T.; Ortega, G

Title: FAMOS-V2: Formation Flying and Rendezvous and Docking Tool for Exploration Mission in Circular and Elliptical Orbits.

Date, location: September 2004, ESTEC, Noordwijk, The Netherlands

Year: 2004

1.5 PhD on Systems Engineering and Automatics

As a result of a collaboration agreement between Universidad Nacional de Educación a Distancia (UNED) and Universidad Complutense de Madrid (UCM), the PhD program on Systems Engineering and Automatics was given by the Computing and Automatics Department of UNED (http://www.dia.uned.es) and the Computers Architecture and Automatics Department of UCM (http://www.dacya.ucm.es).

This doctorate program focuses on important aspects of Automatics and Computer Science, applied to Control Engineering, with the aim of providing the students with specific tools for further research in Systems Engineering and Automatics.

The structure of the doctoral studies is as follows:

1. First phase.

This phase consists of a learning period and the development of a research project. The latter is necessary to achieve the Diploma de Estudios Avanzados (DEA).

The following subjects were studied during the learning period:

- Dynamic Systems Simulation

  Modelling and simulation are nowadays relevant subjects in all engineering and science fields. Numerical simulation is, altogether with the study of experimental results from real systems, the only technique to accurately analyse general non-linear systems under variable conditions. In the framework of process modelling and control, dynamic hybrid models are especially relevant, which are described with continuous and discreet equations; the combination of these formulae is a powerful modelling technique that deals with continuous changes in parameters, as well as changes in the structure of the model itself. This course includes the following subjects: modelling principles for hybrid systems; symbolic handling of equations; algorithms for simulation; programming in Modelica. The teachers of this course are: Dr. Sebastián Dormido Bencomo, Dr. Mª Antonia Canto Díez, Dr. Alfonso Urquía Moraleda, Dr. Victorino Sanz Prat, Dr. Miguel Ángel Rubio González and Dr. Carla Martín Villalba.

- Intelligent control

  The aim of this subject is to provide a detailed overview of advanced control techniques, based on Artificial Intelligence. This subject includes the following topics: expert systems, neural networks, fuzzy logic, genetic algorithms, neurofuzzy applied to pattern recognition, artificial learning. The teacher of this course is Dr. Matilde Santos Peñas (supervisor of this thesis).

- Neural networks and applications

This subject covers the basis of the main types of neural networks, such as feed-forward, dynamic networks, competitive networks, etc, including a deep analysis of their structure, functionalities, goals, characteristics, etc. The aim of this analysis is to understand how they work, to apply this knowledge to any type of neural networks, and to acquire skills to design a neural network for specific problems or applications. An overview of the common characteristics and several classifications are provided. The following networks are described: perceptron, adaline, multi-lawyer perceptron and madelaine, backpropagation, dynamic network Hopfield and competitive network ART. The teachers of this subject are Dr. José Antonio López Orozco and Dr. Juan Jiménez Castellanos.

- Aerospace systems control

The syllabus of this course in aerospace systems and control includes: plane modelling and control, avionics, helicopter control, launchers, orbital motion, rendez-vous, attitude control, interplanetary missions and re-entry. The teacher of this course is Dr. José María Girón Sierra.

After the learning period, a research project was carried out on the topic of optimisation of launchers. The title of this project was “Genetic Algorithm applied to the Optimisation of a Two Stage Launcher”, which director was Dr. Matilde Santos Peñas.

2. Second phase.

This phase consists of the preparation and development of a research work, whose result is this PhD thesis. The aim of this thesis is to apply the knowledge and skills acquired during the first phase of the doctoral program to solve problems in the aerospace field.

Chapter 2

Introduction to Aerospace Problems

The purpose of this chapter is to justify the application of Artificial Intelligence (AI) to complex aerospace problems, and to summarize the state of the art of this fascinating subject. A description of the basics on astrophysics, astrodynamics and aeronautics is also provided, with the aim of refreshing the knowledge of such fundamental concepts, and in order to improve the understanding of this thesis in case the reader is not familiar with those particular areas of physics and engineering. A brief on artificial intelligence is provided at the end of this chapter.

2.1 Overview of related Works and State of the Art

Preliminary works [1][2][3][4][5][6][7][8][9][10] have shown the efficiency of intelligent computational techniques when dealing with complex systems such as aerospace applications. The results proved the value of fuzzy logic for controlling the position of satellites; genetic algorithms (GA) for the optimisation of ascent trajectories, given a set of constrains; genetic algorithms and particle swarm optimisation (PSO) for optimisation of interplanetary trajectories involving fly-bys in planets.

2.1.1 Fuzzy Logic, Lagrange Points and Satellite Control

Modelling and control are intensively studied and researched in multiple engineering fields, like for example the quite interesting investigations presented in [12] [15][16][17][18][74][75][88][94]. More specifically, fuzzy logic works can be found in [12][74][75][94]. In [12], a quite interesting and innovative approach is presented, combining fuzzy control and genetic algorithms. Whereas works based in fuzzy logic are common, in general works based on fuzzy logic with application to control of satellites in Lagrange points is quite scarce, and possibly [5] and [10] are pioneers. Some alternative approach to [5][10] can be found in [67], also related to PIDs. In [82] a PID controller (adaptive-predictive) is applied to a microsatellite. Several control strategies are developed in [24][35] for a varied set of aerospace applications. A mission related to Lagrange points was launched in 2010 by ESA, Herschel-Planck [58], however an impulsive control is applied in that case, with a low and irregular frequently, quite different from the continuous control applied in [5][10]. Related to continuous control, a LQG/LQR control with continuous propulsion is applied in [23] in order to achieve a stable position in the Lagrange points, while keeping the relative position of several satellites (formation flying and keeping). In [97] an analysis of the Lagrange points is presented, in a binary system different from the Earth-Moon (considered in this thesis). Also interesting are [55][57][60][59][95][54][78]. In [49] a quite innovative approach to the applications of Lagrange points is presented.

2.1.2 Genetic Algorithms and Launchers

GAs are non-deterministic search methods that implement some operators observed in the natural evolution of life forms, such as crossovers and mutations. After each generation, the next offspring is intended to improve the performance index provided by the parents. More specific information on GAs can be found in [52] and [72], and a brief summary in section 2.3.2.1 of this PhD thesis. Crossovers are performed between numerical string (chromosome) encoded individuals. A parents' random selection mechanism is introduced to bias the population replacement towards increasing the selected performance index. The higher the performance index of an individual is, the higher the probability of being chosen as a parent, so those good characteristics of the chosen parents shall hopefully pass to the generated offspring. The chromosomes of the parents are combined and a one-point crossover is performed. The best individual in the previous generation substitutes the worst individual of the new generation. Elitism prevents the best element of the population from being deleted to make room for a worse new individual. Once the GA generates the offspring, a mutation operator changes individuals' genes randomly. When the population is not centred on a maximum, a low mutation rate is applied, thus the GA performs the search for the optimum point mostly mixing and combining existing individuals into new ones, whereas raising the mutation rate switches to a broad search when a maximum has been reached and no local improvement can be expected. An outer calculation adjusts the mutation rate in response to the observed behaviour, and when a lot of individuals come up to a specific point of the searching space, the mutation probability automatically increases, avoiding premature convergence to a local function extreme.

A major difficulty when optimizing parameters in a launch mission is to provide algorithms that convergence towards an optimum. Continuation methods, in which a branch of the solution is followed, may be adequate in some cases, and the optimum of a nearby problem may be used as a good initial guess of the solution to the new problem. But even making wise use of the continuation method, a branch of the solution may bifurcate or disappear. And in any case it is difficult or impossible to prove that the found solution is the global optimum and not a local one. Furthermore, completely new missions may arise for which a desired neighbouring problem is absent. Since it is hard

to guess where the domain of convergence towards the global optimum is in a general case, an exploratory work may become recommendable. Exhaustive search, in which the space of optimization variables is explored with a fine grid, is unaffordable except in very simple cases.

Evolutive approaches have been used to optimize some types of trajectories of launchers [26][48] but not the parameters of the launcher and the parameters of the trajectory in order to reach a given trajectory profile in an optimum way. Fuzzy logic and genetic algorithms are put together in [12], following a quite interesting investigation. In [30] the authors present a brief review of the very limited prior uses of GAs to determine optimal spacecraft trajectories. It also applies a multi-objective GA to the optimization of low-thrust interplanetary spacecraft missions (Earth-Mars and Earth-Mercury missions) and so does [91]. In [84] a brief review is presented of the scarce prior uses of GAs to determine optimal spacecraft trajectories, where a guidance method for the ascent trajectory of a single stage to low-earth-orbit (LEO) launch vehicle is developed with the aid of a GA, maximizing the payload delivered to LEO. Evolutive approaches for RLV re-entry is presented in [26] and [96]. Multi-objective GA optimisations of low-thrust interplanetary spacecraft missions (Earth-Mars and Earth-Mercury missions) are presented in [48] and [14]. In most of the references gathered in [48] the objective is to minimize the propellant required for the ascent trajectory from the surface of the Moon, between Earth and Mars, and for developing comet sample return missions.

2.1.3 Genetic Algorithms, Particle Swarm Optimisation and Interplanetary Trajectories

Evolutive approaches have been used to optimise some types of trajectories of launchers, usually RLV re-entry [30][26][48]. Multi-objective GA optimisations of low-thrust interplanetary spacecraft missions are presented in [22][62][84][91][102][103][104]. In most of them the objective is to minimize the propellant required for the ascent trajectory from the surface of the Moon, between Earth and Mars, and for developing comet sample return missions. There is an extensive collection of single objective Particle Swarm Optimisation (PSO) algorithms [38][41][69]. The usage of Multi Objective Particle Swarm Optimisation (MOPSO)

algorithms [13][46][71][85][92] is quite promising; however the optimisation of interplanetary trajectories using MOPSO is quite scarce to date. There are some papers related to interplanetary missions and low thrust, non-keplerian orbits, reflecting an increasing interest during the last decade in evolutionary techniques, as an efficient optimisation tool for these aerospace applications [103][104]. An evolutionary technique is applied to unmanned air vehicles in [34]. Evolutionary neurocontrol is analysed in [33], while in [50] there is an interesting development of formulae to handle low thrust orbits using the Lagrange planetary equations. A Pareto GA is applied in [6][56] to the optimisation of interplanetary trajectories. Gravity assists and low-thrust trajectories are combined in [77][102]. Particularly interesting is the exponential sinusoid approach to optimize low-thrust trajectories as explained in [62][102], which is valid as an initial nearly-analytical guess for further refined solutions, saving time for otherwise time-consuming numerical simulations based on artificial intelligence (AI) optimisation algorithms. Following this approach, a shape-based algorithm is presented in [90]. More information on global optimisation for Mission Analysis can be found in [81].

There are some papers related to interplanetary missions and low thrust, non-keplerian orbits, reflecting an increasing interest during the last decade in evolutionary techniques, as an efficient optimization tool for these aerospace applications [30][52] [92]. Evolutionary neurocontrol is analysed in [33], while in [50] there is an interesting development of formulae to handle low thrust orbits using the Lagrange planetary equations. A Pareto GA is applied in [56] to the optimisation of interplanetary trajectories. Particularly interesting is the exponential sinusoid approach explained in [62][63], which is valid as an initial nearly-analytical guess for later refined solutions, saving time for otherwise time-consuming numerical simulations with others AI optimisation algorithms. Following this approach, a shape-based algorithm is presented in [90]. Whereas shape-based algorithms are highly interesting, a more generic approach is pursued in the present thesis, where impulsive manoeuvres and evolutive algorithms are combined in order to find optimum solutions, as in [6][7][8][9][11]. Some work on low-thrust manoeuvres is also presented, based on article [6], where propagation of orbital motion is minimized by applying a Lambert solver in small arcs of orbit, obtaining a quicker optimisation process that otherwise would be extremely time consuming and therefore unfeasible.

There is an extensive collection of single objective Particle Swarm Optimisation (PSO) algorithms [38][41][69]. The usage of Multi Objective Particle Swarm Optimisation (MOPSO) algorithms [13][46][71][85][92] is quite promising. However, the optimisation of interplanetary trajectories using MOPSO is quite scarce to date; still, it is possible to find some articles suggesting or actually combining both concepts [30][103][104][91]. In particular, gravity assists and low-thrust trajectories are combined in [77][102]. More examples on global optimisation of multiple gravity assist trajectories can be found in [65], focused on pruning techniques. More information on global optimisation for Mission Analysis can be found in [22][81].

2.2 Fundamental Aerospace Concepts

The aim of this section is to explain the mathematics and physics behind the models used in this thesis. This section provides an overview of basic concepts related to celestial mechanics, gravity, launchers, manoeuvres, Lagrange points and interplanetary trajectories.

2.2.1 Gravity field

According to physics today, four basic forces lead all processes in Universe: strong, weak, electromagnetism and gravity. The latter is responsible of the macroscopic structure of the Universe, and consists on the attraction between two punctual masses according to the Newton's Law of Universal Gravitation as expressed in (1):

$$F = G\frac{Mm}{r^2} \qquad (1)$$

where:
F is the gravitational force
G is the universal gravitational constant
M y m are the values of the masses of the particles that attract to each other
r is the distance between particles.

Taking into account that the two particles attract each other, it is possible to write the formula for the gravitational force in the following vector form (2):

$$\bar{F} = G\frac{Mm}{r^2}\frac{\bar{r}}{r} = G\frac{Mm}{r^3}\bar{r} \qquad (2)$$

where:

$\bar{F}$ is the vector of the force applied on particle m

$\bar{r}$ is the vector from m to M

$\frac{\bar{r}}{r}$ is a unitary vector in the direction of $\bar{r}$

A punctual particle is a mathematical concept that does not occur in reality. Indeed, particles have not only mass, but also volume. The gravitational force applied by a real body (such as a planet) over its surrounding can be computed according to the distribution of its mass in its entire volume. In case of a spherical planet with an homogenous distribution of mass (constant density), the gravitational force over any point outside the surface of that planet can be computed by solving an integral formula (3), leading to the conclusion that this force is equivalent to the attraction applied by a punctual particle with the same mass than the whole planet, placed in its centre of mass (which in such case is also the geometrical centre of the planet).

$$\bar{a}_g = G\int_V \frac{\bar{r}}{r^3}\rho(\bar{r})dV \qquad (3)$$

where:

$\bar{a}_g$ is the gravitational acceleration

ρ is the density in a given point inner to the planet

However, in general planets do not have a homogeneous distribution of mass. Hence, solving the integral formula (3) is possible only if the distribution of mass in the planet is exhaustively known. Gravitational force is derived from a gravitational potential [20], which in case of real planets is written as a series of Bessel functions, whose parameters are estimated from data of well-known trajectories of natural or artificial satellites. Such Bessel series are useful to compute the gravitational force taking into account the non-sphericity of the planet, as a function of the latitude, the longitude and the height respect

to that planet. In the case of the Earth and Mars, the knowledge of their gravity fields and non-sphericities is very exhaustive. A possible expression of the gravitational potential [20] is given in (4):

$$U = \frac{\mu}{r}\left[1 - \sum_{n=2}^{\infty} J_n \left(\frac{r_0}{r}\right)^n P_n \sin(L)\right] \qquad (4)$$

where:

U is the gravitational potential

μ is the gravitational parameter GM (M is the mass of the planet)

J_n are coefficients to be determined by experimental observation

r_0 is the equatorial radius of the planet

P_n are Legendre polynomials

L is the geocentric latitude

Alternative expressions for the harmonics of the gravitational field of a planet due to its non-sphericity can be found in many books, like for example [40]. Formula (4) yields to formula (5) when the harmonics are ignored and the planet is considered as if with homogeneous density (i.e. as if the gravitational field were created by a punctual mass):

$$U = \frac{\mu}{r} \qquad (5)$$

Non-sphericity has important long-term effects in the orbits of satellites, as well as in their attitude (orientation). The gravitational perturbations due to the non-sphericity of the planets are called tide forces. Whereas in the Earth the water of the oceans aligns in the direction of the Earth-Moon axis, a similar effect takes place when the natural satellite of a planet orientates in a certain direction respect to the planet, due to the non-sphericity of the satellite; this is the case of our Moon, whose visible surface never changes when seen from the Earth. Such effect is exploited in some artificial satellites, which achieve a certain attitude without the need of commanding any attitude manoeuvre.

2.2.2 Orbital Trajectories in an Isolated System

A system is said to be isolated when it does not interact with any other system. Let it be a system consisting only on two punctual masses, with no external forces acting over them. As explained in section 2.2.1, both masses shall attract each other according to (1) and (2). Depending on the initial position and velocity of both particles, they may follow converging trajectories and crash into each other; or they may approach each other but without crashing, and then travel away forever; or they may move respect to the other cyclically.

Hence, there are two types of cases: those when the relative motion between the two particles describes closed trajectories; and those when such trajectories are open. As said above, we are considering only two particles in our system, ignoring effects different from the gravitational force. Under these premises, such trajectories are called ideal or keplerian [20][40], and in general they consist on ellipses or hyperbolas, as well as straight lines, parabolas or circumferences in particular cases.

Ellipses are the trajectories of two particles when they describe closed curves. In particular, one particle shall describe an ellipse respect to the other, which shall occupy one of the foci of the ellipse. An ellipse may collapse into a straight line, in case of two particles that crash into each other. A circumference is also a particular case of an ellipse, when the two foci are in the same position.

Hyperbolas are open curves, when the two particles has a relative speed such that the gravitational force is not enough to keep them together, so both particles travel away from each other. A hyperbola may also collapse into a straight line.

Parabolas represent the borderline case between ellipses and hyperbolas, when the relative speed of the particles is just enough to take them apart, so both of them asymptotically reach a null speed after enough time. In ballistic, a common practice is to ignore air drag, in which case the trajectory of a projectile is elliptical; in turn, such ellipse is approached to a parabola, however this approach is not valid in case of long distances (such as missiles).

2.2.3 Orbital Energy

The specific orbital energy [20] of a body orbiting around a planet is the sum of the potential and kinetic energies of that body per unit of mass.

The specific orbital energy is an orbital invariant, due to the principle of conservation of energy: ignoring the influence of third bodies and other external effects, the total energy of the orbiting body shall not vary. During each orbit, there are exchanges between orbital and kinetic energy, while their sum remains constant. The kinetic energy of a body per unit of mass is shown in (6), where v is the velocity of the orbiting body at a given instant. The potential energy per unit of mass is shown in (7).

$$\widetilde{E}_c = \frac{1}{2} v^2 \tag{6}$$

$$\widetilde{E}_p = -\frac{GM}{r} \tag{7}$$

Note that equation (7), which is equivalent to (5), is the conservative field that yields to the calculation of the gravitational force in (1); the value of the gravitational field is null at infinite distance, and decreases towards negative values (i.e. increases in absolute value) when decreasing the distance to the planet. The spatial gradient of the potential energy in a radial direction (8) is the gravitational force, thus matching equation (1).

$$F = \frac{dE_p}{dr} = m\frac{d\widetilde{E}_p}{dr} = m\frac{GM}{r^2} \tag{8}$$

Summing (6) and (7) leads to (9), which is the specific orbital energy of the orbiting body:

$$\widetilde{E}_{orb} = \widetilde{E}_c + \widetilde{E}_p = \frac{v^2}{2} - \frac{GM}{r} \tag{9}$$

Figure 1 show some geometric parameters of an ellipse.

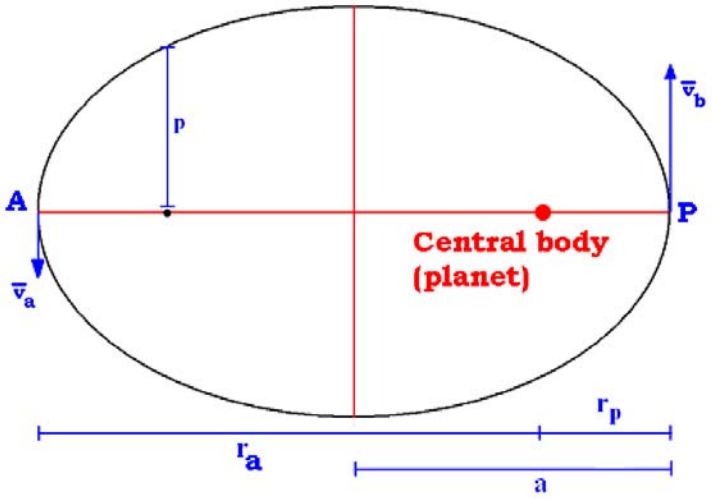

Figure 1: An elliptic orbit

Conservation of energy between the apoapsis A and the periapsis P (which are, respectively, the furthest and nearest points to the central body or planet) can be analyzed from equation (9), as shown in equation b), which yields to c).

$$\widetilde{E}_{orbA} = \frac{v_A^2}{2} - \frac{GM}{r_A} = \frac{v_P^2}{2} - \frac{GM}{r_P} = \widetilde{E}_{orbP} \qquad \text{b)}$$

$$v_P^2 - v_A^2 = 2GM\left(\frac{1}{r_P} - \frac{1}{r_A}\right) \qquad \text{c)}$$

where:

r_A is the distance from the apoapsis to the central body

r_P is the distance from the periapsis to the central body

v_A is the velocity of the orbiting body in the apoapsis

v_P is the velocity of the orbiting body in the periapsis

The angular momentum of the orbiting body respect to the central body is also invariant. Computing the value of the angular momentum is quite simple in the apoapsis and the periapsis d), as in such points of the orbit the position vector of the orbiting body respect to the central body is perpendicular to the speed.

$$h_A = r_A v_A = r_P v_P = h_P \qquad \text{d)}$$

where h_A and h_P are the angular momentum in the apoapsis and periapsis, respectively. Equation d) yields to e):

$$v_P = \frac{v_A r_A}{r_P} \qquad \text{e)}$$

Substituting equation e) in equation c) yields to f) and g):

$$v_A^2\left(\frac{r_A^2}{r_P^2} - 1\right) = 2GM\left(\frac{1}{r_P} - \frac{1}{r_A}\right) \qquad \text{f)}$$

$$v_A^2 = \frac{2GMr_P}{r_A\left(r_A + r_P\right)} \qquad \text{g)}$$

From Figure 1, it is trivia to obtain equation h):

$$2a = r_A + r_P \qquad \text{h)}$$

Equations g) and h) yield to equation i):

$$v_A^2 = \frac{GMr_P}{a\, r_A} \qquad \text{i)}$$

Replacing equation i) in equation a)i)(9) for apoapsis, we obtain equation j).

$$\widetilde{E}_{orbA} = \frac{v_A^2}{2} - \frac{GM}{r_A} = \frac{GMr_P}{2a\, r_A} - \frac{GM}{r_A} = \frac{GM}{2a}\left(\frac{r_P}{r_A} - \frac{2a}{r_A}\right) \qquad \text{j)}$$

From Figure 1 and equation h), it is trivia to obtain equation k):

$$2a = r_A + r_P \qquad \text{k)}$$

Considering j) and k), we obtain equation l):

$$\widetilde{E}_{orbA} = \frac{GM}{2a}\left(\frac{2a - r_A - 2a}{r_A}\right) = -\frac{GM}{2a} \qquad \text{l)}$$

The specific orbital energy is constant all over the orbit, thus equation l) can be written in general as in equation m):

$$\widetilde{E}_{orb} = -\frac{GM}{2a} \qquad \text{m)}$$

Equation m) proves that, apart from the mass of the central body and the universal gravitational constant (which values remain constant), the calculation of the specific

orbital energy depends purely on *a*, which is the semi major axis of the orbit. The specific energy of a body placed statically on the surface of a planet (i.e., without any velocity respect to the planet) can be computed considering only the potential energy, according to equation n), which is a particular case of equation a)i)(7).

$$\widetilde{E}_0 = -\frac{GM}{R} \qquad \text{n)}$$

where *R* is the distance from the surface to the centre of the planet.

Comparing equations m) and n), the additional energy required to place a body in orbit, departing from the surface of the planet, and ignoring the rotation of that planet, can be computed from equation o):

$$\Delta\widetilde{E} = GM\left(\frac{1}{R} - \frac{1}{2a}\right) \qquad \text{o)}$$

Equation o) is useful for calculations related to the energy per unit of mass that is required to launch an object for the surface of a planet in order to place it in orbit. Such calculus, however, is approximate and not accurate, due to the ignored effects of air drag, efficiency of motors, rotation of the planet, etc. Equation o) is useful when comparing the energy required launching objects into very different orbits, assuming that the dispersion of energy is similar for those orbits.

2.2.4 Orbital Elements

Five independent quantities are sufficient to completely describe the size, shape and orientation of an orbit, while a sixth quantity is required to define the position of a satellite in the orbit at a particular time. Such quantities can be expressed in several ways, perhaps being the most intuitive the three components of the position vector of the satellite plus the three components of its velocity.

A no so obvious way of defining an orbit and the position of an orbiting body in it is through the so called orbital elements. These are quantities directly related to the geometry of the ellipse. The exact set of elements may vary depending on the

application and the singularity of the orbit (i.e., if it is parabolic, circular or straight line), but the most classical set [20] (sometimes undefined for some of those singular orbits) is as follows:

- semi-major axis, a: See Figure 1. If this quantity is positive, the orbit is an ellipse, and then a is half the length of the major axis. If this quantity is negative, this quantity is half the distance between the two branches of an hyperbola. If this quantity is infinity, the resulting orbit is parabolic. Finally, if this value is zero, then the orbit is a straight line.

- eccentricity, e: This quantity represents the ratio between the foci distance of a conic and its major axis, and it is always defined as positive. Its value is between 0 and 1 in an ellipse. The eccentricity of a circle is 0, while in a parabola this is exactly 1. In an hyperbola, the eccentricity is greater than 1.

- inclination, i: This is the angle between the orbital plane and an arbitrary plane, normally defined as the equatorial plane of a planet, or the orbital plane of the Earth around the Sun in case of heliocentric orbits.

- longitude of ascending node, Ω: Let it be the line of nodes of an orbit, defined as the intersection between the orbital plane and a reference plane. The orbit passes through the line of nodes in two points or nodes, called ascending and descending node, depending on the direction of the probe in those two nodes. Let it be an arbitrary direction in the reference plane, normally defined as that of the vernal equinox. Then, the longitude of the ascending node is defined as the angle between the line of nodes and the vernal equinox.

- argument of periapsis, ω: This quantity represents the angle, in the orbital plane, between the ascending node and the periapsis.

- time of periapsis, t_p: This is the time at which the probe is in periapsis, respect to some specified reference initial time.

This list is not exhaustive, as it is possible to define a different set of variables that may include for instance:

- the semi-latus rectum, also called parameter of the conic, p: See Figure 1. The relation between p, e and a is given by (1):

$$p = a\left(1 - e^2\right) \tag{1}$$

- the true anomaly at epoch, ν_0: The angle of the probe, in the orbital plane, respect to the periapsis, is called true anomaly. Given this quantity in a certain instant of time, the position of the satellite is defined.

- the argument of latitude at epoch, u_0: The argument of latitude is the angle of the probe, in the orbital plane, respect to the ascending node.

- the true longitude at epoch, l_0: This is the sum of the longitude of the ascending node Ω and the argument of latitude at epoch u_0.

2.2.5 Lagrange Points

Given a binary system (two celestial bodies orbiting each other, with no relevant influence of third bodies), in which the two central bodies describe two perfect circular trajectories around each other, it is possible to define five points in the rotary reference system in which the gravitational forces from the two bodies perfectly compensates the centrifugal force.

Such points, called Lagrange points or libration points, are quite interesting, as it is possible to place a spacecraft in any of those points, and the spacecraft shall keep its position respect to the two central bodies without any need of manoeuvring or fuel consumption. In practice, only two of the five Lagrange points are stable, while strictly speaking the Lagrange points only exist when the two trajectories of the central bodies around each other are perfectly circular (no eccentricity at all), and there are not effects due to third bodies (other planets, Sun), or non-sphericity gravitational harmonics.

A representation of a binary system and its five Lagrange points is provided in Figure 2. The masses of the two planets, M1 and M2, are represented as red circles. The two planets rotate around each other describing perfectly circular orbits respect to the centre of masses CM, with a constant angular speed ω. If there are not gravitational fields from third bodies (i.e., ignoring the rest of the Universe), the position of CM remains fixed. Three of the five Lagrange points (L1, L2 and L3) are in the straight line that passes through M1 and M2. The other two Lagrange points (L4 and L5) rest on the intersection of two circles, centred in each planet and with radius equal to the distance between M1 and M2. Thus, it is possible to represent the two green equilateral triangles shown in Figure 2, whose vortexes rest on M1, M2, L4 and L5. When M1 has much more mass than M2, then CM is near M1, so the orbit of M2 passes near L4 and L5. Otherwise, L4 and L5 rest outside the orbit of M2 around CM, as M2 is a bit nearer to CM than M1, while the distance from M2 to L4 and L5 is the same than from M1 to M2. Segment from M1 to M2 has an angle of 60 degrees respect to the segments joining M1 and M2 with L4 and L5. In Figure 2 the most common notation has been used, so L1 is between M1 and M2, L2 is beyond M2, L3 is beyond M1, L4 is in the direction of the movement of the less massive body M2, and L5 is in the direction of the movement of M1.

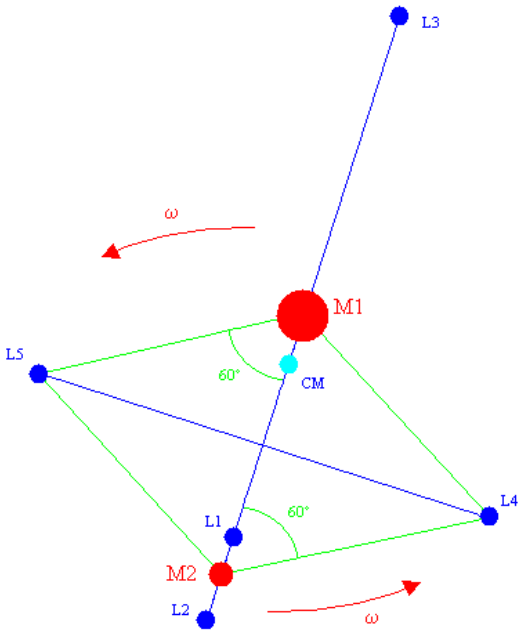

Figure 2: Representation of the Lagrange points in a binary system

It is possible to compute the position of the Lagrange points by finding those points in which the gravitational force compensates the centrifugal force of the binary system. Given the mass *m* of a particle placed in a Lagrange point (we will see later that the value of such mass is irrelevant for computations), given the position vector $\bar{r}$ of such particle respect to the CM in the rotatory reference system, given the masses M_1 and M_2 of central bodies M1 and M2, given their positions $\bar{R}_1$ and $\bar{R}_2$ respect to the CM in the rotary reference system, and given the rotary speed of the binary system ω, then centrifugal force on mass *m* is p):

$$F_{centr} = m\omega^2\bar{r} \qquad \text{p)}$$

The value of the two gravitational forces applied by each central body on the particle is q):

$$F_{grav,i} = G\frac{M_i m}{\left|\bar{R}_i - \bar{r}\right|^2}\frac{\bar{R}_i - \bar{r}}{\left|\bar{R}_i - \bar{r}\right|} \qquad \text{q)}$$

where $\frac{\bar{R}_i - \bar{r}}{\left|\bar{R}_i - \bar{r}\right|}$ is the unitary vector in the direction of each central body.

On the Lagrange points, the centrifugal force compensates the joint gravitational forces of the two central bodies, as shown in r):

$$m\omega^2\bar{r} = G\frac{M_1 m}{\left|\bar{R}_1 - \bar{r}\right|^3}\left(\bar{R}_1 - \bar{r}\right) + G\frac{M_2 m}{\left|\bar{R}_2 - \bar{r}\right|^3}\left(\bar{R}_2 - \bar{r}\right) = G\left[\frac{M_1 m}{\left|\bar{R}_1 - \bar{r}\right|^3}\left(\bar{R}_1 - \bar{r}\right) + \frac{M_2 m}{\left|\bar{R}_2 - \bar{r}\right|^3}\left(\bar{R}_2 - \bar{r}\right)\right] \qquad \text{r)}$$

As said above, mass *m* can be simplified in r), which yields to s):

$$\omega^2\bar{r} = G\left[\frac{M_1}{\left|\bar{R}_1 - \bar{r}\right|^3}\left(\bar{R}_1 - \bar{r}\right) + \frac{M_2}{\left|\bar{R}_2 - \bar{r}\right|^3}\left(\bar{R}_2 - \bar{r}\right)\right] \qquad \text{s)}$$

Whereas equation s) is valid, a more practical equation should be achieved for each of the five Lagrange points separately. Proving that there are only five Lagrange points is beyond the scope of this section.

Considering the geometry shown in Figure 2, scalar equations t), u) and v) are valid for L1, L2 and L3, respectively.

$$\omega^2 r_{L1} = G\left[\frac{M_1}{(R_1 + r_{L1})^2} - \frac{M_2}{(R_2 - r_{L1})^2}\right] \qquad \text{t)}$$

$$\omega^2 r_{L2} = G\left[\frac{M_1}{(R_1 + r_{L2})^2} + \frac{M_2}{(R_2 + r_{L2})^2}\right] \qquad \text{u)}$$

$$\omega^2 r_{L3} = -G\left[\frac{M_1}{(r_{L3} - R_1)^2} + \frac{M_2}{(r_{L3} + R_2)^2}\right] \qquad \text{v)}$$

Equations t), u) and v) can be written in vectorial form as in w), x) and y):

$$\omega^2 \bar{r}_{L1} = \frac{\bar{R}_2}{R_2} G\left[\frac{M_1}{(R_1 + r_{L1})^2} - \frac{M_2}{(R_2 - r_{L1})^2}\right] \qquad \text{w)}$$

$$\omega^2 \bar{r}_{L2} = \frac{\bar{R}_2}{R_2} G\left[\frac{M_1}{(r_{L2} + R_1)^2} + \frac{M_2}{(r_{L2} + R_2)^2}\right] \qquad \text{x)}$$

$$\omega^2 \bar{r}_{L3} = -\frac{\bar{R}_2}{R_2} G\left[\frac{M_1}{(r_{L3} - R_1)^2} + \frac{M_2}{(r_{L3} + R_2)^2}\right] \qquad \text{y)}$$

Regarding L4, considering the angle between the segment joining L4 to M1and segment joining M1 and M2, it is possible to solve for the position of L4 from z):

$$\bar{r}_{L4} - \bar{R}_1 = \frac{\bar{L}}{2} - \frac{\sqrt{3}\bar{L}^*}{2} \qquad \text{z)}$$

where:

$\bar{L} = [L_x, L_y] = \bar{R}_2 - \bar{R}_1$ is a vector from M1 to M2 (see Figure 2 and Figure 3)

$\bar{L}^* = [-L_y, L_x]$ is a vector perpendicular to $\bar{L}$ (see Figure 2 and Figure 3) and ending in L4 (so it begins in the middle point of the segment joining M1 to M2).

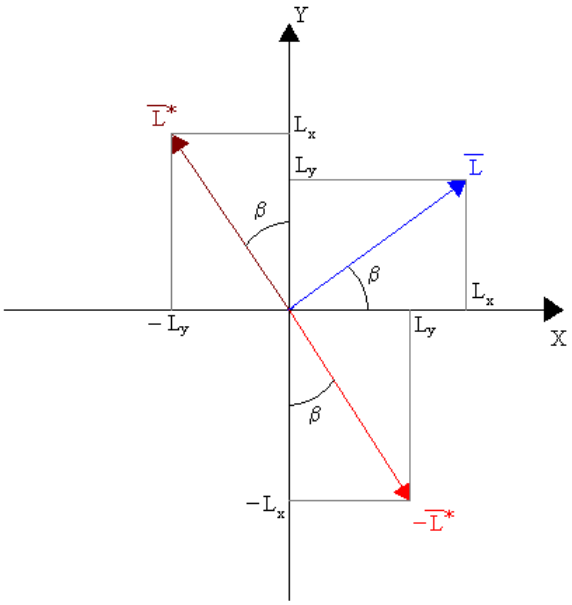

Figure 3: Relation between vectors $\overline{L}$, $\overline{L}^*$ and $-\overline{L}^*$

Analogously, the position of L5 can be solved from aa):

$$\bar{r}_{L5} - \overline{R}_1 = \frac{\overline{L}}{2} + \frac{\sqrt{3}\overline{L}^*}{2} \qquad \text{aa)}$$

Proving that L4 and L5 are equilibrium points is out of the scope of this section. Also, any formal stability analysis of the five Lagrange points is out of scope, however it may be useful to point out that only L4 and L5 are stable, while L1, L2 and L3 are "saddle" points. Nevertheless, all the Lagrange points are not stable in real systems, due to perturbations from third bodies, to non-sphericity of the central bodies and their gravity fields, and most important, to the eccentricity of the central bodies around each other. However, the Lagrange points (which strictly speaking do not exist in real systems) are still valid as the drift from equilibrium position is very slow, thus it is possible to place spacecrafts in the Lagrange points of the Earth-Moon or Earth-Sun binary systems for observation of the central bodies or the rest of the Universe, while keeping a fixed position to the Earth and avoiding fuel consumption, thus increasing the life of the missions. Temporarily placing probes near Lagrange points may become the base of longer interplanetary trips [49]. In practice, in Herschel-Planck [58] and other missions no satellite is placed exactly in the Lagrange points, but in orbits around them instead, involving manoeuvres to correct the drift in position from time to time, so the instability of the Lagrange points is compensated.

2.2.6 The Gauss' Problem and Orbit Transfers

The Gauss´ problem, also known as the Lambert problem, may be defined as follows [20]: given two positions $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ in an orbit, given the time of flight t from $\boldsymbol{r_1}$ to $\boldsymbol{r_2}$, and the direction of motion ("short way" or "long way", as the orbiting object may follow the orbit in two possible directions), find velocities at $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$, that shall be called $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ respectively. Solving this orbit determination problem leads to the definition of the arch of orbit connecting two different orbital positions. Also, this is particularly useful when $\boldsymbol{r_1}$ belong to the current orbit, and $\boldsymbol{r_2}$ belongs to a different orbit that the spacecraft should reach in a given time t. Solving the Gauss's problem leads, in this case, to defining the arch of orbit connecting the initial and final orbits, that the spacecraft should follow when transferring from the first to the second. The speeds $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ define, respectively, the desired speeds when initiating or finishing the transfer, which in turn define the additional speeds or delta-Vs that must be achieved with a combination of engines and, depending on the conditions, a fly-by around a planet (see section 2.2.6).

The Gauss' problem has not an analytical solution, thus it must be solved using iterative methods. There is an extensive literature [21][40][20] describing the various iterative strategies for this interesting problem (often relying in variables with not an obvious physical meaning). Some interesting works can also been found in [53][64][70]. It is not intended to present a exhaustive analysis of all the different methods available in literature, however a brief summary is introduced here regarding the main two branches or families of such approaches to solve the Gauss' problem: universal variables and p-iteration. For details on these methods, a better understanding of the heavy mathematics behind, a meaningful interpretation of the involved variables, suggestions on initial values and alternative iterations for those variables, and considerations on convergence, divergence and computational speed, please refer to the literature. An interpretation of variables f, g, x, y and z can be found in [21] and especially in [40]. A historical approach (quite interesting, from an academic point of view in order to understand the different methods in place today to solve the Gauss' problem) can be found in [20]. A combination and automatic selection of the two main algorithms (and variations) is

mandatory in order to improve convergence and speed, depending on the geometry of the problem; bearing this dynamic selection in mind, the approach followed in [64] has been used in Chapter 5.

2.2.6.1 Solution of the Gauss Problem using the p-iteration Method

First, constants *k*, *l* and *m* are calculated, as shown in equations (1), (1)(a) and (2), where ν is the true anomaly of the probe (see section 2.2.4) and $\Delta\nu$ is the angle between $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$. A first guess for *p* (normally called "parameter" or "semi-latus rectum") is calculated using (3), based on the mean values of the limits of this parameter as shown in equations (4) and (5), corresponding to two parabolic orbits passing through $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$. From (6) the semi-major axis *a* of the conic will be calculated, thus defining the type of conic (hyperbola if *a* is negative, ellipse if *a* is positive). Then, values for *f*, *g* and $\dot{f}$ are solved from equations (7), (8) and (9), so the elliptical (ΔE) or hyperbolic (ΔF) eccentric anomaly can be computed, from (10) and (11) in the elliptical case, or from (12) and (13) in the hyperbolic one (note that there is no ambiguity in determining ΔF, as this is always assumed to be positive, so only (12) or (13) is required). Using (14) for an elliptical trajectory or (15) for an hyperbolic one, it is possible to estimate the time of flight, which most probably will not match the desired time of flight as *p* has been guessed, thus *p* needs to be adjusted to a new value and the whole process must be repeated from (3) until the obtained time of flight equals that of the problem. In order to calculate a new estimation for *p*, the linear interpolation described in (16) may be used, however other alternatives are also valid as explained in literature. Finally, once convergence is achieved, $\dot{g}$ must be computed from (17), and then $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ are calculated from (18) and (19), respectively.

$$k = r_1 r_2 \left(1 - \cos\Delta\nu\right) \tag{1}$$

$$l = r_1 + r_2 \tag{a}$$

$$m = r_1 r_2 \left(1 + \cos\Delta\nu\right) \tag{2}$$

$$p = \left(\frac{k}{l + \sqrt{2m}} + \frac{k}{l - \sqrt{2m}} \right) \Big/ 2 \tag{3}$$

$$p_i = \frac{k}{l + \sqrt{2m}} \tag{4}$$

$$p_{ii} = \frac{k}{l - \sqrt{2m}} \tag{5}$$

$$a = \frac{mkp}{\left(2m - l^2\right)p^2 + 2klp - k^2} \tag{6}$$

$$f = 1 - \frac{r_2}{p}\left(1 - \cos\Delta\nu\right) \tag{7}$$

$$g = \frac{r_1 r_2 \sin\Delta\nu}{\sqrt{\mu\, p}} \tag{8}$$

$$\dot{f} = \sqrt{\frac{\mu}{p}} \tan\frac{\Delta\nu}{2}\left(\frac{1 - \cos\Delta\nu}{p} - \frac{1}{r_1} - \frac{1}{r_2}\right) \tag{9}$$

$$\cos\Delta E = 1 - \frac{r_1}{a}\left(1 - f\right) \tag{10}$$

$$\sin\Delta E = \frac{-r_1 r_2 \dot{f}}{\sqrt{\mu a}} \tag{11}$$

$$\cos\Delta F = 1 - \frac{r_1}{a}\left(1 - f\right) \tag{12}$$

$$\sin\Delta F = \sqrt{1 - \cos^2\Delta F} \tag{13}$$

$$t = g + \sqrt{\frac{a^3}{\mu}}\left(\Delta E - \sin\Delta E\right) \tag{14}$$

$$t = g + \sqrt{\frac{\left(-a\right)^3}{\mu}}\left(\sinh\Delta F - \Delta F\right) \tag{15}$$

$$p_{n+1} = p_n + \frac{\left(t - t_n\right)\left(p_n - p_{n-1}\right)}{t_n - t_{n-1}} \tag{16}$$

$$\dot{g} = 1 - \frac{r_1}{p}\left(1 - \cos\Delta\nu\right) \tag{17}$$

$$\bar{v}_1 = \frac{\bar{r}_2 - f\,\bar{r}_1}{g} \tag{18}$$

$$\bar{v}_2 = \dot{f}\,\bar{r}_1 + \dot{g}\,\bar{v}_1 \qquad (19)$$

The *p* iteration method converges in all cases except when $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ are exactly collinear, which is not a probable scenario as numerical methods are in place, thus the initial and final position vectors are subjected to some numerical accuracy. Nevertheless, the colinearity scenario is typically associated to a free fall ($\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ in the same direction) or a Hohmann transfer ($\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ in opposite directions), being the latter generally the optimum transfer manoeuvre between two circular orbits, which would be applicable when travelling from one position to another, but this specific manoeuvre is highly improbable in any of the patched conics of an analysed interplanetary trajectory, as it involves several planets with quite arbitrary positions (see section 2.2.7 and Chapter 5).

2.2.6.2 Solution of the Gauss Problem via Universal Variables

From $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$, and taking into account the direction of motion, constant *A* should be evaluated from equation (20). Using a trial value of variable *z*, equations (20)(a) and (21) should be evaluated, so then variable *y* can be computed from equation (22), and variable *x* from (23). The trial value of *z* is then checked based on the time of flight (TOF) calculated from equation (24). Procedure should be repeated until a stable value of *t* is achieved. Value of *z* may be interpolated based in equation (25), which is analogue to equation (16), however other more successful methods can be found in literature. Then, equations (26),(27) and (28) are evaluated and, from the values of *f*, *g* and $\dot{g}$, the initial and final speeds of the arch of orbit, $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$, are obtained from equations (18) and (29).

$$A = \frac{\sqrt{r_1 r_2}\,\sin\Delta\upsilon}{\sqrt{1-\cos\Delta\upsilon}} \qquad (20)$$

$$C = \frac{1-\cos\sqrt{z}}{z} \qquad (a)$$

$$S = \frac{\sqrt{z}-\sin\sqrt{z}}{\sqrt{z^3}} \qquad (21)$$

$$y = r_1 + r_2 - A\frac{1 - zS}{\sqrt{C}} \tag{22}$$

$$x = \sqrt{\frac{y}{C}} \tag{23}$$

$$\sqrt{\mu}t = x^3 S + A\sqrt{y} \tag{24}$$

$$z_{n+1} = p_n + \frac{(t - t_n)(z_n - z_{n-1})}{z_n - z_{n-1}} \tag{25}$$

$$f = 1 - \frac{y}{r_1} \tag{26}$$

$$g = A\sqrt{\frac{y}{\mu}} \tag{27}$$

$$\dot{g} = 1 - \frac{y}{r_2} \tag{28}$$

$$\bar{v}_2 = \frac{\dot{g}\bar{r}_2 - \bar{r}_1}{g} \tag{29}$$

Any computational algorithm for solving the Gauss problem via universal variables should consider the possibility of negative values of y, prior to the evaluation of x with equation (23). This check is necessary only for positive values of constant A ("short way" trajectories) [20].

2.2.7 Fly-by or Gravitational Assistance around Planets

The fly-by trajectory consists on using the gravity field of a planet to apply a delta-V to a probe, which may accelerate considerably, depending on the approaching and leaving directions respect to the planet (the kinetic energy gained by the probe comes from the negligible reduction of the planet's speed). Considering the size of the sphere of influence of a planet respect to the size of its orbit, and comparing the interplanetary times of flight with the duration of the fly-by, any planet may be considered as a

dimensionless point of mass at which the spacecraft changes its velocity instantly due to the fly-by.

Taking into account a safe radius of curvature of the fly-by is necessary in order to obtain practical results, otherwise considering a planet as a punctual mass would lead to a nearly-zero approach and an unrealistic high acceleration due to the gravity force. When using a geometrical analysis (see below), the lack of a defined minimum radius shall lead, under extreme conditions, to fly-by around, for example, a grain of dust (if such grain were included in the solar system model). While such extreme situation is not possible, it improves the understanding of the need of some defined radius of curvature for fly-bys (infinity, in the case of the grain). More likely but still unrealistic scenarios would involve fly-by through the atmosphere or even the surface of the planet. Such unrealistic situations are avoided by defining radius of curvature, which are directly related to the size of the planet, and indirectly to the mass; for instance, a heavy planet may be used to change the direction of a probe quite quickly, while a light planet may not be used for such demanding changes, if any fly-by is possible at all.

As shown in Figure 4, the probe approaches the planet with a velocity $\boldsymbol{v_a}$ respect to the heliocentric reference system, and $\boldsymbol{v_{\infty,a}}$ respect to the planet. When departing from the planet, the velocity of the probe respect to the heliocentric reference system is $\boldsymbol{v_b}$, while the velocity respect to the planet is $\boldsymbol{v_{\infty,b}}$. Since the trajectory of the probe respect to the planet is hyperbolic, the values of modules $v_{\infty,a}$ and $v_{\infty,b}$ should be equal to the same value v_∞.

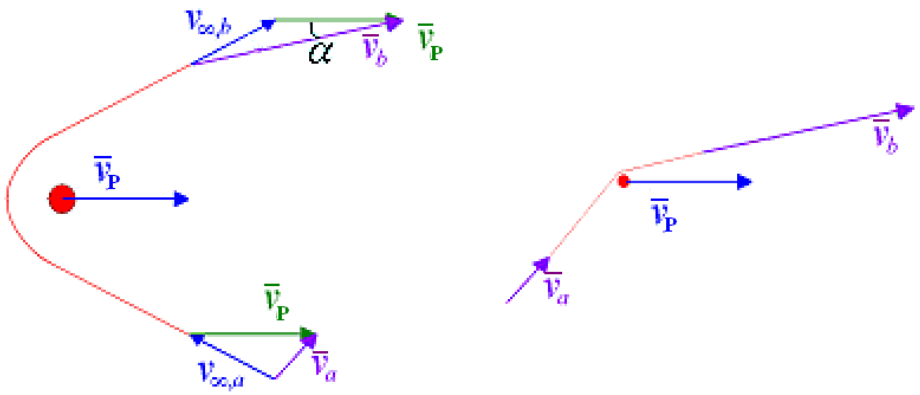

Figure 4: Fly-by in a planet P

The relations between $\boldsymbol{v_a}$, $\boldsymbol{v_b}$, $\boldsymbol{v_{\infty,a}}$, $\boldsymbol{v_{\infty,b}}$ and the velocity of the planet $\boldsymbol{v_P}$ are shown in equations bb) and cc):

$$\overline{v}_{\infty,a} = \overline{v}_a - \overline{v}_P \qquad \text{bb)}$$

$$\overline{v}_{\infty,b} = \overline{v}_b - \overline{v}_P \qquad \text{cc)}$$

Obtaining the value of v_∞ is trivial dd):

$$v_\infty = \sqrt{\overline{v}_{\infty,a} \cdot \overline{v}_{\infty,a}} \qquad \text{dd)}$$

Equation ee) shows how to compute the cosine of angle α between $\boldsymbol{v_b}$ and $\boldsymbol{v_P}$, from which v_b can be calculated using the theorem of the cosine ff), as represented in Figure 4.

$$\cos\alpha = \frac{\overline{v}_P \cdot \overline{v}_b}{v_P v_b} \qquad \text{ee)}$$

$$v_b^2 + v_P^2 - 2v_b v_P \cos\alpha = v_\infty^2 \qquad \text{ff)}$$

2.2.8 Propulsion Systems

The specific impulse I_{sp} of a propulsion system is defined as the provided force (momentum variation) per propellant unit [98]. Spent fuel is disposed and expelled through the rocket nozzle. The fuel budget may be given as weight (kg·m/s^2) or mass (kg). Thus, the specific impulse units shall be in seconds (s) or in meters per second (m/s). In this thesis, the specific impulse units are in meters per second. The force or thrust provided by the rocket is computed by multiplying the specific impulse of the propellant and the mass flow thought the active nozzles. The flow of mass thought the nozzles should be taken into account to calculate the current mass of the spacecraft during each stage, according to equation (1):

$$\frac{dm}{dt} = -\frac{T}{gI_{sp}} \qquad (1)$$

where T is the thrust and g is the gravity acceleration at the surface of the Earth.

2.2.9 Gravity Turn

Gravity turn refers to one of the phases of a launch. During that phase, as seen in Figure 5, the probe or launcher is pushed by the thrust of the propulsion system (section 2.2.8), while pulled downwards by the gravitational force of the Earth (section 2.2.1). Obviously, the thrust direction of any launcher or vehicle is applied in the same direction of the movement for optimal design (i.e., thrust in the direction of the aerodynamic axis). The equations governing the motion during the gravity turn manoeuvre [27] are equations (2)(3)(4)(5):

$$\frac{dx}{dt} = V\cos\gamma \tag{2}$$

$$\frac{dh}{dt} = V\sin\gamma \tag{3}$$

$$m\frac{dV}{dt} = T - D - \left(ma_g - \frac{m\left(\frac{dx}{dt}\right)^2}{R+h} \right)\sin\gamma \tag{4}$$

$$mV\frac{d\gamma}{dt} = -\left(ma_g - \frac{m\left(\frac{dx}{dt}\right)^2}{R+h} \right)\cos\gamma \tag{5}$$

where:

m is the mass of the spacecraft during each stage considering equation (1),

V is the spacecraft speed, T is the thrust,

D is the air drag (see section 2.2.10),

x is the horizontal downrange distance,

h is the vertical altitude distance,

a_g is the gravitational acceleration at each point,

R is the radius of the Earth, and

γ is the flight path angle of the vehicle, which is defined as the angle between the velocity vector of the launcher and the local horizon.

The calculation of equations (2)(3)(4)(5) is based in a rotating frame, which *y* axis is oriented in a positive altitude direction, and *x* axis in a local horizon forward direction, as represented in Figure 5. In particular, (2) and (3) have been calculated geometrically, while (4) and (5) were obtained by equilibrium of forces in vehicle along and transverse axes. These forces are the thrust *T*, the weight *P*, the air drag *D* and the centrifugal force F_c (see Figure 5), which appears in the specified non-inertial rotating frame.

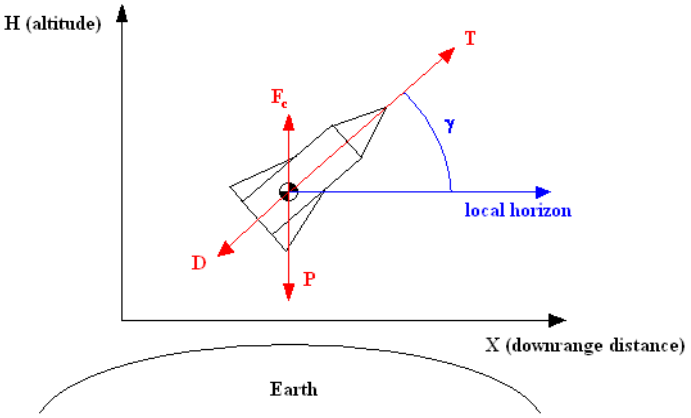

Figure 5: Local horizon rotating frame for gravity turn modelling

2.2.10 Air Drag

The air drag force is provided by equation (6):

$$D = \frac{1}{2}\rho A_{ef} V^2 \qquad (6)$$

where:

A_{ef} is the effective area for air drag calculations,

ρ is the air density, and

V is the speed of the spacecraft.

There are many different Earth atmospheric models that provide with useful information for air density calculation ρ, for an average day, a hot day, a cold day, a tropical day, etc. These models are updated every few years to include the latest atmospheric data. Models are generated by considering atmospheric measurements that are averaged and

matched a curve to produce estimated equations, which assume that the pressure and temperature change only with altitude [20]. One of the most important models is the U.S. Standard Atmosphere 1976 one [100]. In Figure 6 the air density is shown against the height (up to 200 km), highlighting that the Earth's atmosphere is an extremely thin sheet of air which density drops quickly with altitude. Therefore, the air drag effects are especially important near the Earth's surface, and almost negligible at high altitudes (but still important enough to be considered for accurate calculations in long LEO missions around the Earth).

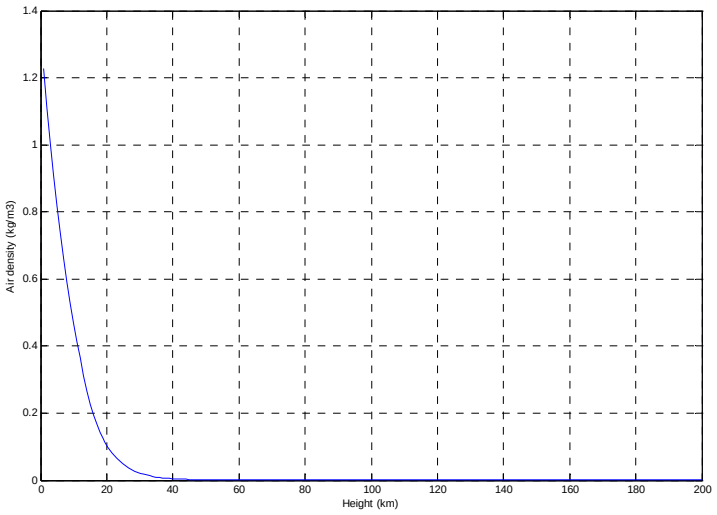

Figure 6: Air density evolution in altitude in Earth's atmosphere (U.S. Standard Atmosphere 1976)

2.3 Artificial Intelligence and Intelligent Strategies

Several concepts has been used and applied during the development of the PhD degree: neural networks, fuzzy logic, genetic algorithms, expert systems, particle swarm optimisation, etc (see section 1.5). All of these intelligent strategies are based on the processes of learning and solving problems by human beings, particularly when a mathematical formulation lead to complex models with no analytical solutions, or to variables that can be better expressed and understood in terms of qualitative descriptions rather than exact values. Combining such human skills with the power of computers is

the aim of the artificial intelligence. Whereas for sure all the intelligent strategies mentioned above are highly interesting, three of them have been intensively used during the research and research at this PhD degree: fuzzy logic, genetic algorithms and particle swarm optimisation. A brief summary of each technique is presented in this section.

2.3.1 Fuzzy Logic

While fuzzy logic [31][32][86] is mostly used for machine control nowadays, other applications out of that scope are also possible, such as implementation of controllers for satellites. Other applications for tuning PIDs are available in [67]. Fuzzy concepts are related to variables that are better described qualitatively, rather than with exact numbers. Genetic algorithms and neural networks may also deal with fuzzy variables, and some neural networks are exactly equivalent to fuzzy systems [66]. Fuzzy logic has the advantage of handling variables like human beings, so the human experience is easily modelled, especially in those cases and contexts when such experience is already applied successfully. Also, whenever modelling a complex system is not possible or too expensive, or processing and simulating a complex model is time-consuming, fuzzy controllers may become an interesting alternative.

The values of the input variables in a fuzzy control system are mapped into fuzzy sets. The process to associate values of the variables with fuzzy sets is called fuzzyfication, and is performed via membership functions that transform the value of the input variables into fuzzy values of the fuzzy sets. The output variables are also defined as fuzzy sets. For example, given a set of input variables, the first variable may belong to a fuzzy set which values are “high” and “low”, the fuzzy set for the second variable may have values “hot”, “warm”, “cold”, etc; analogously, the output variables of the system may belong to fuzzy sets with values such as “on” and “off”; “nominal”, “warning” and “stop”; “maximum power”, “nominal power” and “no power”, etc. In summary, some implemented rules of the fuzzy system are applied to the inputs and to combinations of inputs, and the results of the rules are mapped into fuzzy sets. Based on these results, the output variables are generated and converted into practical values through a process called defuzzyfication.

2.3.2 Evolutionary Strategies

A major difficulty when optimising parameters in a complex system is to provide algorithms that converge towards an optimum. Exhaustive search is unaffordable except for very simple cases. In general, using an algorithm based on an intelligent exploration of solutions such as genetic algorithms (GA) [52] or Particle Swarm Optimisation (PSO) [69] is necessary. These techniques are evolutionary: from an initial set of guesses, these are improved by combining information between them, as the exploration of solutions continues, and new solutions with better performances are found after each step, until ideally an optimum solution is achieved.

2.3.2.1 Genetic Algorithms (GA)

GAs are non-deterministic, evolutionary, search methods that implement some operators observed in the natural evolution of life forms, such as crossovers and mutations. After each generation, the next offspring is intended to improve the performance index provided by the parents. More specific information on GAs can be found in [52] and [72].

Crossovers are performed between numerical strings (chromosome) encoded individuals. A parents' random selection mechanism is introduced to bias the population replacement towards increasing the selected performance index. The higher the performance index of an individual is, the higher the probability of being chosen as a parent, so those good characteristics of the chosen parents shall hopefully pass to the next generation. The chromosomes of the parents are combined and a one-point crossover is normally performed, however performing crossovers in more points is also possible. The best individual in the previous generation substitutes the worst individual of the new generation. Elitism prevents the best element of the population from being deleted to make room for a worse new individual.

Once the GA generates the offspring, a mutation operator changes individuals' genes randomly. A possible technique is to apply a variable mutation [2][3][5][7][8][9][1]: when the population is not centred on a maximum, a low mutation rate is applied, thus the GA performs the search for the optimum point mostly mixing and combining existing individuals into new ones, whereas raising the mutation rate switches to a broad searching space when a maximum has been reached and no local improvement can be expected. An outer calculation adjusts the mutation rate in response to the observed behaviour, and when a lot of individuals come up to a specific point of the searching space, the mutation probability automatically increases, avoiding premature convergence to a local function extreme.

Given a complex system, a set of individuals is defined. Each individual consists on the concatenation of a representation of each of the input values of the system. Hence, given an individual, it represents a certain combination of values of the input variables. This representation is usually binary), as conceptually represented in Figure 7, (or "calibrated" binary, as explained below) but this is not mandatory and any other representation is also valid. In Figure 7, an individual is represented as an example, consisting in a certain combination of values for the three input variables of a system.

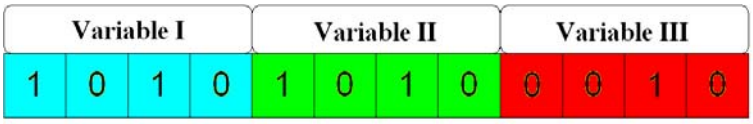

Figure 7: Example of individual in GAs

Let it be η one of the adjustable parameters (system input variable) of the GA, which maximum and minimum values are η_{max} y η_{min}, respectively. Let it be N the number of bits used for binary-coding the value of η. Then, it is possible to code up to 2^N different values of η in those N bits. A traditional binary interpretation of the bits would be based on equation gg):

$$\eta_{dec} = \sum_i b_i 2^{i-1} \qquad \text{gg)}$$

where η_{dec} is the value of η according to a traditional representation of N bits, i is the position of each bit (the first bit is assumed to refer to 2^0) and b_i is the value of each bit.

While representation in gg) is valid, it does not represent all possible values of η efficiently, in case η_{min} is much bigger than 0, or if η_{max} is much lower than the value represented when all bits are set to 1; in such cases, many possible representations shall never be used, thus wasting the potential of the *N* bits, while the valid values of η are confined altogether in a narrow set of representations, thus losing all the potential accuracy that the *N* bits may offer. Also, negative values would require some special representation, preferably continuous so jumps when executing the GA are avoided, but this issue is not trivia with the traditional binary coding, even if one of the bits is used as the sign of the represented value. Bearing these issues in mind, an alternative binary representation is desirable, applying a rescale and an offset, so a binary representation consisting in all 1s is associated to η_{max}, and a representation of all 0s is associated to η_{min}. Such alternative representation is based in equation hh), which permits the usage of the whole range of values of the *N* bits. The joint of these two operations, rescale and offset, is sometimes called "calibration", however this term is used in other contexts with different meanings, usually referring to the testing and initial set-up of a machine, an actuator or a sensor.

$$\eta = \frac{\eta_{dec}\eta_{\max}}{\left(2^N - 1\right)} + \eta_{\min} \qquad \text{hh)}$$

Crossing between individuals lead to an offspring that may take the characteristics of the parents. Thus, only the best individuals are chosen in order to generate the next set of individuals. There are many different possible implementations of the selection algorithm, which normally select parents randomly but with some favouritism for those individuals with better characteristics (better performance, according to some criteria function related to the optimisation of the output variables of the system). It is expected that, using this technique, the positive characteristics of the chosen parents are taken by the next generation, while those individuals with worse characteristics (worse performance, according to the criteria function) tend to disappear. A selection of parents exclusively based on performance and with no random selection may lead to homogeneous results in the population; this homogeneity decreases the success of the GA as it may miss better solutions far away in the space of input variables, thus some random factor is desirable when selecting parents, even if some individuals with bad

characteristics are chosen. Some elitism is also desirable, to ensure that the best individual of a given generation is not lost in the next generation.

The combination of parents, represented in the left part of Figure 8, consists on splitting the binary strings or chromosomes in a certain random position (one-point cross-over), or in several positions (several-point cross-over). In Figure 8 a one-point cross-over is represented. Once the strings or chromosomes are split, their parts are exchanged and a new individual is created. Finally, the mutation is applied randomly to one or more bits (some GA are implemented so a maximum of 1 bit can be modified by the mutation operator).

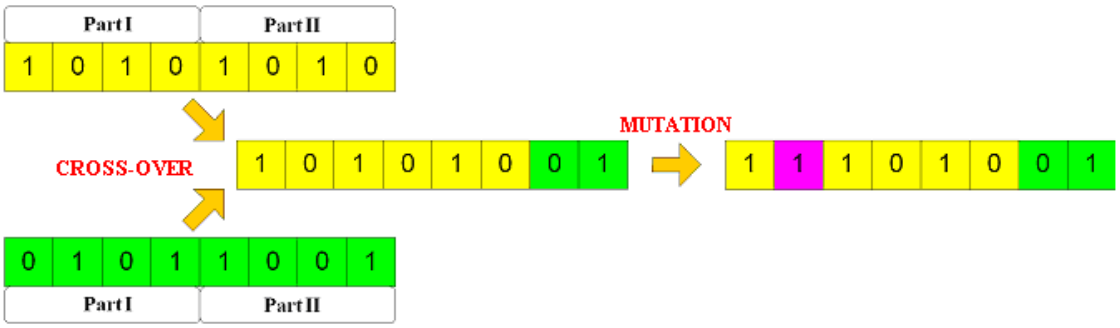

Figure 8: Representation of the cross-over and mutation operators in GA

2.3.2.2 Particle Swarm Optimisation (PSO)

PSO is a population-based, swarm intelligent, search algorithm based on the simulation of the social behaviour of birds within a flock or any other swarm organism [69], using physical movements of the individuals within the swarm. Many different techniques can be applied to implement PSO, so the evolutionary mechanism of the swarm algorithm adapts to the global and local exploration of the search space area. Whereas other evolutionary algorithms are based on parent representation and selection of individuals, PSO and Multi-Objective PSO (MOPSO) rely on leaders to guide the whole population. PSO refers to single-output analysis, while MOPSO refers to PSO when more than one output is available [13]. One of the first works on swarm algorithms is reported in [69]. A survey of the state-of-the-art of particle swarm optimization is summarized in [92]. A comparison of multiobjective evolutionary algorithms is presented in [109], while in

[71] some suggestions are explained for improving both spread and convergence in swarm algorithms and Pareto-fronts. Data structure and turbulence concepts are applied to swarm algorithms in [46].

An exhaustive and detailed explanation on PSO and MOPSO, and their different variants, is beyond the scope of this section, however a brief explanation on these techniques is provided with the purpose of improving the understanding of forthcoming chapters.

Given a complex system, let it be a population of individuals, each defined by its position in the space of input variables. The updated position of every *i-th* individual after the *k-th* step can be described with the general formulae (1) and (2):

$$\bar{v}_{i,k} = \bar{v}_{i,k-1} + \sum_{j=leaders} c_{i,j}\left(\bar{x}_{i,j} - \bar{x}_{i,k-1}\right) + c_0 \qquad (1)$$

$$\bar{x}_{i,k} = \bar{x}_{i,k-1} + \bar{v}_{i,k} \qquad (2)$$

where:

$\bar{v}_{i,k}$ is the velocity vector of the *i-th* individual in transition from step *k-1* to step *k*.

$c_{i,j}$ are random numbers

c_0 is a random number (sometimes called "turbulence factor") that introduces a random component into the updated final position of each individual, similar to mutation in GAs

$\bar{x}_{i,j}$ is the *j-th* leader for the *i-th* individual

$\bar{x}_{i,k}$ is the position (decision vector) of the *i-th* individual in *k-th* step.

As shown in (1) and (2), a leader is defined for each individual in the population, and the position of each individual is changed by weighting the current distance to each leader with some random numbers, and combining these weighted distances into a single vector that represents the change in the position of the individual in the space of input variables.

Whereas formulae (1) and (2) are quite intuitive, it is not obvious to define the leaders for each individual, and there are many different criteria in scientific articles, using local leaders, past history of each particle, etc. Nearly all PSO approaches use as one of the leaders the individual with the best global results in the population. Normally, a second leader is the best historical result of each individual (so this second leader is "personal"

to each individual). In order to identify and characterize the best individual of the population, a fitness function should be defined, based on the relevant output or outputs of the system. When the fitness function has only one output, then the selection of the best individuals is quite straight forward, proved that the fitness functions is wisely chosen. There are other interesting considerations when designing a PSO algorithm, such as distribution of random weights, dynamic assignment of leaders, turbulence factors, etc.

2.3.2.3 Multi Output Particle Swarm Optimisation (MOPSO)

When the fitness function of a PSO has several outputs (i.e., when several outputs of the system are relevant for optimisation), each output may conflict with each other and the selection of the leaders is not as obvious as in section 2.3.2.2, where the selection of the leader was based on a single output. Some strategies for selection are required for defining and identifying the PSO leaders when a multi-output criteria function is in place. Multiobjective optimisation problems are quite common in every area of knowledge [47][108][61]. In such problems, the objectives to be optimised are normally in conflict respect to each other, hence there is not a single solution for these problems. Traditionally, multiobjective optimisation problems are solved with multi-input, single-output target functions, combining all the objectives into just one formula to be optimised. This approach has been proved to be useful in many problems, but it is not always easy to find the best fitness function, being a more interesting approach to obtain a global view of all the different dominant solutions, presented as Pareto-fronts. Such global view may be particularly necessary when analysing the influence of the variations of some inputs in all the outputs. Whereas this analysis can be made for single-output functions (applying gradient methods, for instance), some difficulties may arise when there are many variables, local solutions or discontinuities. Swarm algorithms avoid these drawbacks and support the multi-output, Pareto-front concept better than other AI algorithms [29][28]. In such multi-objective problems, it is necessary to implement in the MOPSO algorithm how to choose the leaders (for instance, the nearest solution in the Pareto-front).

The function to optimise in a general multi-objective minimization problem may be written as in (3):

$$\bar{y} = f(\bar{x}) = [f_1(\bar{x}), f_2(\bar{x}), ..., f_k(\bar{x})] \quad (3)$$

where $\bar{x} = [x_1, x_2, ..., x_n]$ is the vector of decision variables and $\bar{y} = [y_1, y_2, ..., y_n]$ is the vector of outputs.

If (3) cannot be analytically evaluated, it is mandatory to use some numerical calculations and iterative processes in order to find the optimum solution. Given two different potential solutions, represented as vectors $\bar{y}_A$ and $\bar{y}_B$, the first one would be dominant respect to the second one if (4) is verified. If the inequality is strictly fulfilled, then $\bar{y}_B$ is called a dominated solution.

$$y_{A,i} \leq y_{B,i} \qquad \forall i \quad (4)$$

Note that minimization problems are considered here, without loss of generalization, as (3) and (4) can be easily adapted to maximization problems. Dominated solutions are not interesting in optimisation problems, as the interpretation of (4) is that solution $\bar{y}_A$ is better than $\bar{y}_B$, for any of their components. Therefore, $\bar{y}_B$ can be discarded. However, it may happen that $\bar{y}_A$ surpassed $\bar{y}_B$ in only some (but not all) outputs, and vice versa; in that case, $\bar{y}_A$ and $\bar{y}_B$ are both non-dominated solutions to each other, and both may offer interesting properties and results.

The aim of the MOPSO is to find a set of non-dominated solutions (Pareto-front), which have to be saved in some data structure during the execution of the algorithm. The size of this structure may grow in size very quickly, as more non-dominated solutions are found in every step. After each step, the new position of the individuals in the population must be compared to every solution in the Pareto-front, to perform some substitution if necessary. This increases the complexity and the computational time in every step, as the Pareto-front becomes bigger. In order to ensure that solutions are varied and heterogeneous, some mechanism must be used so the distribution of the decision vectors in the Pareto-front is smooth and uniform. Dealing with these three issues (size of the Pareto front, comparison of Pareto front to individuals in the population, and mechanism to ensure a smooth distribution) lead to establish a limit of

the number of solutions in the Pareto-front. Individuals with new positions in the space of input variables become candidates to join the Pareto-front, if they are non-dominated respect to any other Pareto-front solution. Before they finally join the Pareto-front (discarding some other solution already in the Pareto-front), a comparison is performed between the candidate to join and the candidate to be dismissed, in terms of distances to other solutions in the Pareto-front. The substitution is performed by considering only the highest distance, so the density in the Pareto-front is kept as low as possible. In spite of this technique, that ensures some variety of the solutions, sometimes the Pareto-front is not as broad as expected, due to isolated solutions and strong discontinuities in global, single-output optima.

Chapter 3

Modelling, Simulation and Fuzzy Control of Satellites in Lagrange Points

The aim of this section is to present the modelling and analysis of the behaviour of five satellites placed at the five Lagrange points of the Earth-Moon system, and to control their position with conventional and fuzzy PD controllers.

3.1 Introduction

The modelling of the five satellites placed at each of the five Lagrange points of the Earth-Moon system [10] has been developed including solar wind effects, non-spherical gravitational fields and the drift of the orbital elements of the Moon orbit. A continuous propulsion is applied to each satellite, computed with fuzzy PD controllers, and the results are compared to classic PD controllers. The aim of those controllers is to keep the position of the satellites in spite of the disturbances (solar wind, etc).

As explained in section 2.2.55, the Lagrange points (also known as libration points) of a system of two planets, one orbiting around the other in perfect circles, are those points where the centrifugal and gravitational forces cancel out, for the corresponding rotating reference frame [20][49]. Therefore, a satellite placed on those Lagrange points shall maintain its position respect to the two central bodies, without the need of applying any thrust or propulsion [98]. Two of the Lagrange points are stable, however a satellite near one of those two points would describe circles around the equilibrium position. The other three Lagrange points are unstable, with slow drifts away from the equilibrium position.

However, those theoretical Lagrange points are not possible in real systems, due to several effects (perturbations) that make impossible to exactly cancel the centrifugal and gravitational forces: each central body does not describe a perfect circumference around the other planet, effects from third bodies, effects from solar wind, non-spherical harmonics of the gravitational fields of the planets, drift of the gravitational parameters of the Moon, etc. In a real system, though, it is possible to keep the position of a satellite in a Lagrange point applying a low thrust, so fuel expenses are also low and the mission may last for a long time. In order to design such a mission, a control system is required that stabilizes the position of a spacecraft in each Lagrange point in spite of perturbations. Indeed, the five Lagrange points are interesting for observation mission with a long duration.

Satellite control based on PD controllers was presented in [4] and [10]; in the latest, most complex models were present, as described in this section. An adaptive-predictive controller is applied to a microsatellite, based on a PID, in [82]. Dynamic programming is presented in [80]. Some original aerospace control strategies are found in [35][24]. For the particular case of Lagrange points, in mission Herschel-Planck [58] an impulsive control is in place, where each impulsive manoeuvre is computed without any automation. A LQG strategy is applied to the simulations in [23] in order to achieve a stable position around the Lagrange points, in the context of formation flying. In [97] an analysis of the Lagrange points is presented, for the Neptune-Triton system. Also interesting are [55][57][60][59][95][54][78]. In [49] a quite innovative approach to the applications of Lagrange points is presented. Whereas those works are related to PD controllers, satellite control and Lagrange points, in [4] and [10], perhaps for the first time, a continuous fuzzy PD controller is analyzed for all the Lagrange points of the Earth-Moon system.

3.2 Description of the Model

The models of the Earth-Moon-satellites system were written in modelling language *Modelica* [79], and the simulations were run in *Dymola* [37]. *Modelica* is a non-proprietary, object-oriented, equation based language to model complex physical systems containing, e.g., mechanical, electrical, electronic, hydraulic, thermal, control, electric power or process-oriented subcomponents. *Dymola* is a software to simulate the dynamic behaviour and complex interactions between systems of many engineering fields, such as mechanical, electrical, thermodynamic, hydraulic, pneumatic, thermal and control systems. Aerospace applications of *Modelica* and *Dymola* like in [4] and [10] are scarce to date, and the approach presented in those articles is quite original. Indeed, *Modelica* is normally used for modelling complex physical systems with elements from mechanics, electricity, electronics, hydraulics, thermal engineering, etc [99]. Dymola provides an environment that supports the implementation and simulation of the Modelica models, including plots and 2D and 3D animations, normally for the engineering fields listed above. A 3D animation was introduced in the model described in this section, with the purpose of visualising the movement of the Moon, the Earth and

the satellites around the centre of mass of the whole system, which remains in the same position during the simulation.

The Earth and the Moon are implemented as a 3D system that includes the drift of the gravitational parameters of the Moon around the Earth [20]. As explained in section 2.2.1, the gravitational force is due to a gravitational field, which can be written a series of harmonic functions [68], whose parameters can be computed and estimated from the trajectories of known natural and artificial satellites. Based on equation (4), 50 harmonics has been considered for the Earth, and 20 for the Moon. The non-sphericity of the Earth and the Moon has important long-term effects in the motion of the satellites and their orientation (attitude), specially the harmonic J2 [40].

The solar wind has been modelled according to [44] and [45], based on (5):

$$P_{solar} = \alpha(r_{solar})\rho v_{solar}^2 \qquad (5)$$

where:
P_{solar} is the solar pressure,
α is a parameter that depends on the distance to the Sun (r_{solar}),
ρ is the ionic density, and
v_{solar} is the speed of the solar wind.

The value of the perturbation due to the solar wind is the result of multiplying P_{solar} and the equivalent area of the satellite (the area seen from the Sun, which depends on the orientation of the satellite in case the satellite has no spherical symmetry).

The third body effects (due to the gravitational forces of other bodies in the Solar system) are not relevant. The biggest third body effect corresponds to the Sun, which is around 0.0059 m/s^2, thus no relevant for the modelling and simulation of the Earth-Moon system.

Each satellite is modelled as a punctual mass. Three forces act on each satellite: solar wind, gravitational force and thrust from the propulsion system, as seen in (6):

$$\overline{F} = \overline{F}_{gravity} + \overline{F}_{solar} + \overline{F}_{thrust} \qquad (6)$$

where **F** is the sum of the three aforementioned forces.

Based on section 2.2.1, the gravitational force due to the joint effects of the Earth and the Moon can be computed according to (7):

$$\bar{F}_{gravity} = Gm\left[\frac{M_1}{\left|\bar{R}_1 - \bar{r}\right|^3}\left(\bar{R}_1 - \bar{r}\right) + \frac{M_2}{\left|\bar{R}_2 - \bar{r}\right|^3}\left(\bar{R}_2 - \bar{r}\right)\right] \quad (7)$$

where:

$\bar{F}_{gravity}$ is the gravitational force on each satellite, due to the two central bodies (Earth and Moon, in this section)

M_1 and M_2 are the mass values of the two central bodies

m is the mass of the satellite

$\bar{r}$ is the position of the satellite respect to the centre of mass of the Earth-Moon system

$\bar{R}_1$ and $\bar{R}_2$ are the positions of the two central bodies respect to the centre of mass of the Earth-Moon system

Whereas in [4] the trajectories of the satellites were simulated in free drift (i.e. without applying any controlled thrust), in [10] a PD control loop is applied as shown in Figure 9. The inputs of the controller are the three components of the difference between the position of the Lagrange point and the actual position of the satellite. Based on this error, the controller computes the thrust (control signal) of the propulsion system, which delivers a variable force with a very low mass flow and high specific impulse [20][93]. The product of these two values is the force of the propulsion system, as explained in section 2.2.8. In this case, while the thrust is low, this is enough to cancel the slow drift of the satellites away from the Lagrange points, while ensuring a quite low fuel expense that permits a long life of the mission [57].

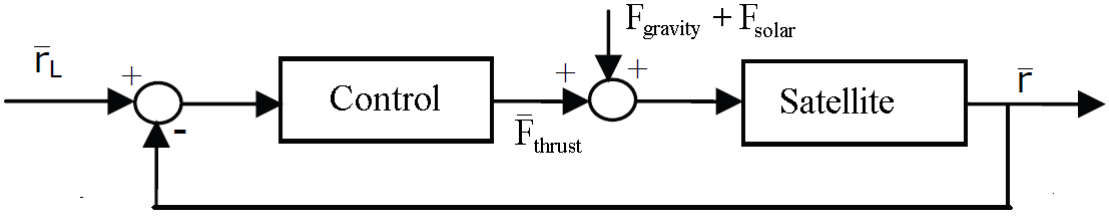

Figure 9: Control loop for satellites in the Lagrange points

The Lagrange points have been modelled as explained in section 2.2.5. The need of a controller is justified by the unstable nature of those points, particularly in real systems with external perturbations and where the movement of each of the planets around the other is not a circumference.

As *Modelica* is an object-oriented language, some objects have been created in [10] that could be reused to create new models without the need of programming from scratch. Indeed, the idea of using objects in *Modelica* is to create a library that can be used for modelling aerospace projects, and in this section. In general, some of the objects in *Modelica* can be reused directly, and some others need a small modification [99]; also in general, there are not too many objects in the standard *Modelica* libraries for aerospace modelling, thus [10] is interesting to cover such a gap. For details on these *Modelica* objects, please refer to [10]. The configuration of the system in the initial time of the simulation is shown in Figure 10. The reference system is inertial and its origin is the centre of mass of the Earth-Moon system; the Y axis passes through the Earth and the Moon in the perigee of the Moon; the X axis is contained in the plane of motion; the Z axis is perpendicular to the visual plane of Figure 10. In the initial time, the Moon is in the perigee, thus on the Y axis. The Earth-Moon system rotates clockwise in Figure 10. The initial positions and velocities of all the bodies can be defined from geometry, in order to let *Dymola* simulate the system from the initial time.

The model was validated as explained in [10]. The five satellites are first simulated without control (free drift) as in [4], and later a controller is added (see section 3.3). Figure 11, Figure 12, Figure 13, Figure 14 and Figure 15 show the drift of the satellites during 10^7 seconds (around 116 days). The red line corresponds to the x-axis, and the blue line to the y-axis. The z-axis is not interesting and has not been represented.

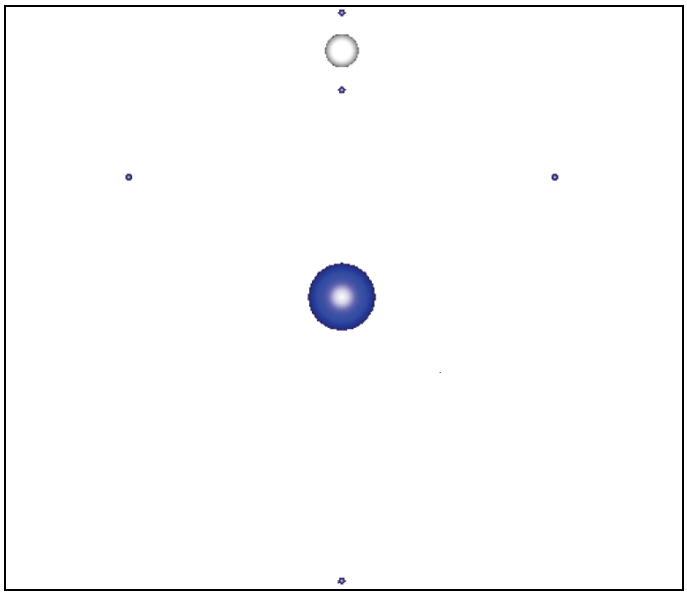

Figure 10: Initial time of the simulation of the satellites in the Earth-Moon Lagrange points

In Figure 11 the drift of the satellite in L1 is appreciated from the very beginning and in less than six days it is already several hundreds of thousands of kilometres away from the equilibrium position. The behaviour of the satellite seems cyclic but the initial position is never recovered.

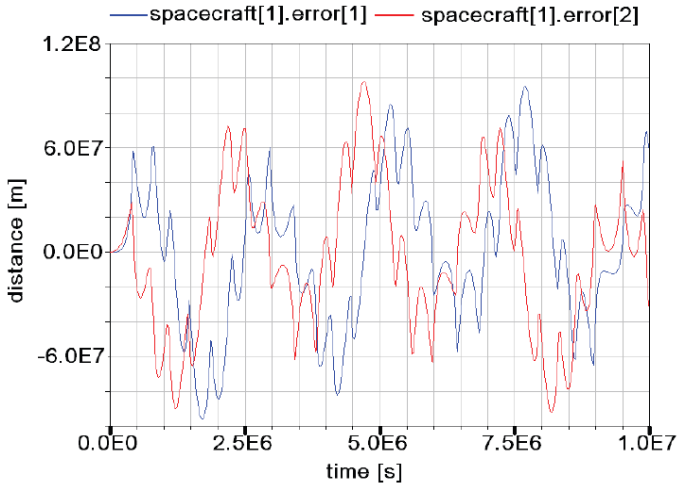

Figure 11: Position errors of a satellite in free drift near L1

The free drift of the satellites in L2 (Figure 12) and L3 (Figure 13) is initially slower than in the case of L1, however the differences between the centrifugal and gravitational

forces becomes important and after 10^7 seconds those satellites are around thirty times farer from the equilibrium positions than in the case of L1.

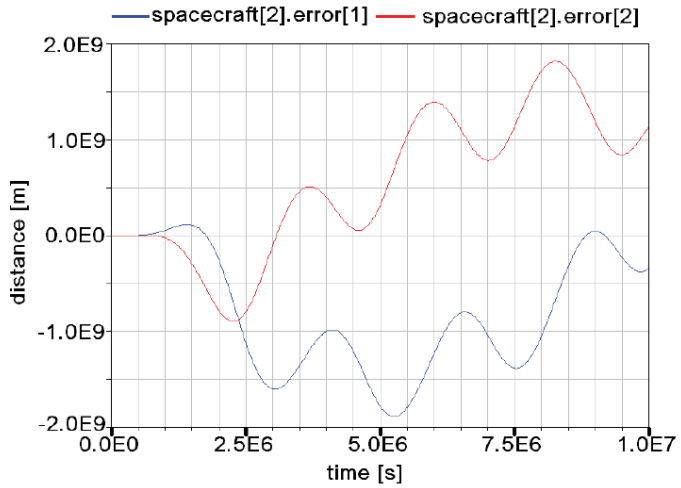

Figure 12: Position errors of a satellite in free drift near L2

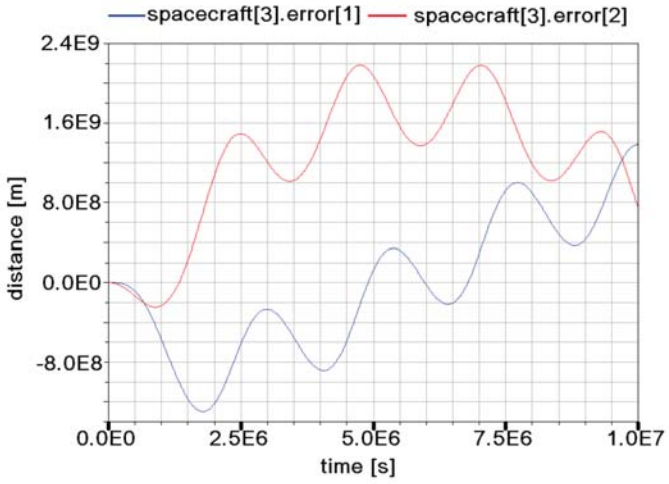

Figure 13: Position errors of a satellite in free drift near L3

The satellites in L4 (Figure 14) and L5 (Figure 15) show an unstable position as well, however the drift is dramatically slower (only 16 mm after 10^7 seconds). Such results confirm the stability analysis explained in section 2.2.5.

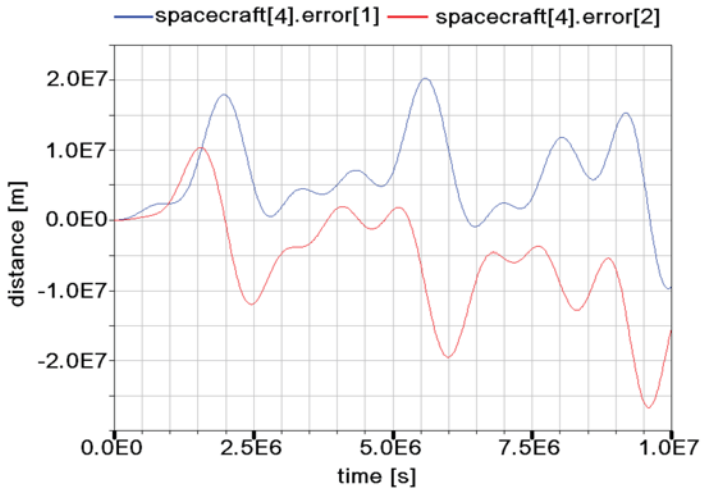

Figure 14: Position errors of a satellite in free drift near L4

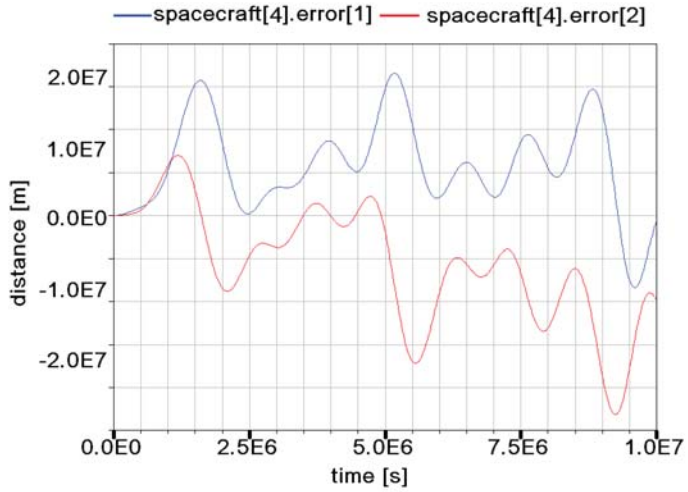

Figure 15: Position errors of a satellite in free drift near L5

3.3 Control Results

As shown in Figure 11, Figure 12, Figure 13, Figure 14 and Figure 15, a control strategy is necessary in order to ensure that the satellites remain in the Lagrange positions. As a continuation of [4], PD controllers (conventional and fuzzy) were implemented in [10]. No integrator has been implemented in the controller, due to the double-integrating action of the transfer function of the satellite in Figure 9 (in the control loop, the input to the satellite is force, and the output is the position). The parameters of the controller were adjusted based on the convenience of the results. The

purpose of the controllers is to analyze if placing satellites in Lagrange points and controlling their thrust automatically with a fuzzy algorithm is feasible. The most important plots shall be shown, following the approach of [10] and for the same reason: to avoid repeating again and again the same plots with not added value or information. A summary of the results in presented in Table 1.

3.3.1 Conventional P and PD Controllers

A proportional controller is applied to compute the thrust of the propulsion system of the satellites. In Figure 16 and Figure 17 the two components of the error of position in the *x* and *y* axis are shown, for L4 and L5. The position error is always within 6 meters for L1, within 25 cm for L2, and within 40 m for L3. Some initial transitory is observed in L1, L2 and L3, as initially the acceleration of the Lagrange points is different from that of the satellites, hence both the position and the velocity become different almost instantly, and only recovered after a few days. An initial transitory is also observed in the position error of satellites in L4 and L5, however the position error is within 10 cm afterwards, much smaller than in L1, L2 and L3. With the P controller, the applied thrust on each satellite is proportional to the position error. As the error is small, so is the propulsion, permitting low fuel expenses.

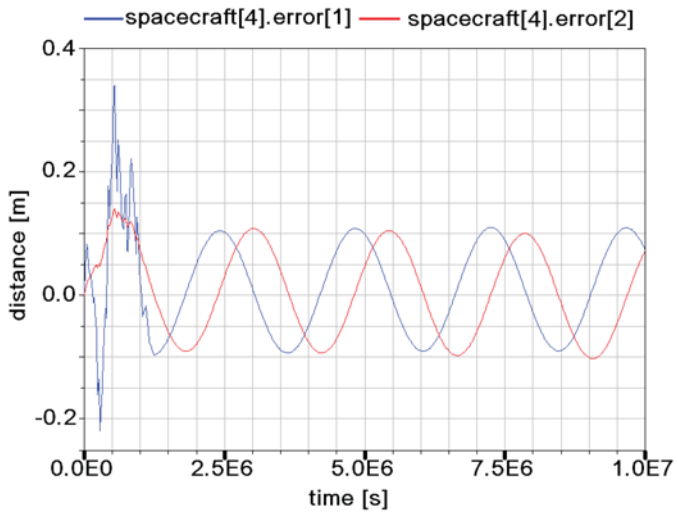

Figure 16: Position errors of a satellite in L1 (P controller)

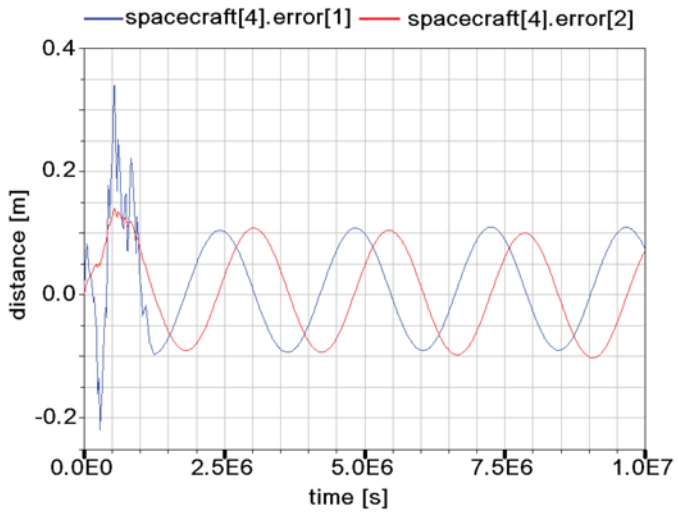

Figure 17: Position errors of a satellite in L4 (P controller)

In order to improve the behaviour of the satellites and to avoid the initial transitory error, the controller is improved [10] with a derivative action in the controller. As a result, the initial transitory has disappeared from Figure 18 and Figure 19, where the position error for L3 and L5 is shown. The steady error has also been improved: within 2 m for L3, being this the biggest error. The thrust is similar to that when the P controller was applied, without the initial transitory in propulsion.

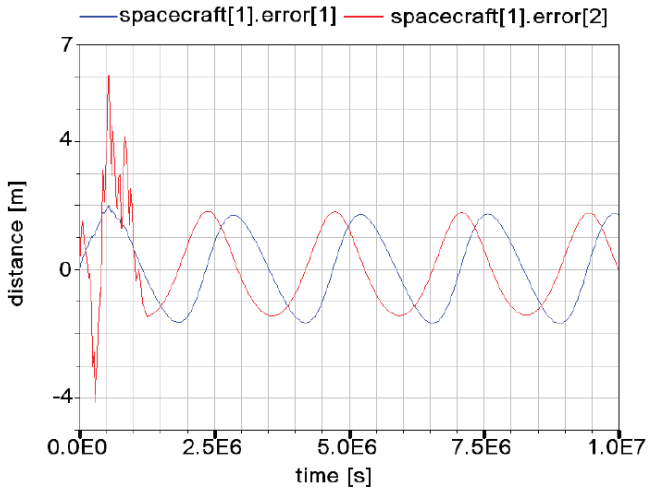

Figure 18: Position errors of a satellite in L3 (PD controller)

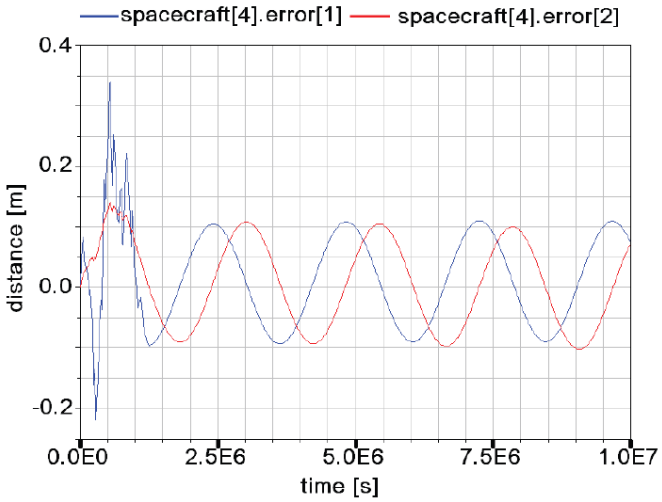

Figure 19: Position errors of a satellite in L5 (PD controller)

3.3.2 Intelligent Fuzzy P and PD Controllers

In order to improve the free drift behaviour, in [10] two additional control strategies, based on fuzzy logic, were implemented. For the fuzzy P controller, the inputs are the components of the position error vector, and the outputs are the components of the thrust (control signals). For the fuzzy PD controller, the inputs also include the components of the derivative of the position error.

Regarding the P controller, two fuzzy sets were considered, labelled as “positive error” and “negative error”. Both fuzzy sets were implemented with ramp functions between 0 and 1, reaching such values when the error was 10 km (see Figure 20). The output variable “thrust” was defined as two fuzzy sets: “no propulsion” and “propulsion”. The “no propulsion” covers the case in which the sum of the truth values of the inputs is below a certain threshold (i.e., small error implies no propulsion), whereas “propulsion” covers the case in which the sum of truth values of the inputs is above such threshold (i.e., big error implies propulsion). This strategy reduces the action of the controller, hence reducing the fuel budget, as the propulsion is no continuous like in section 3.3.1., and the control is applied only when necessary.

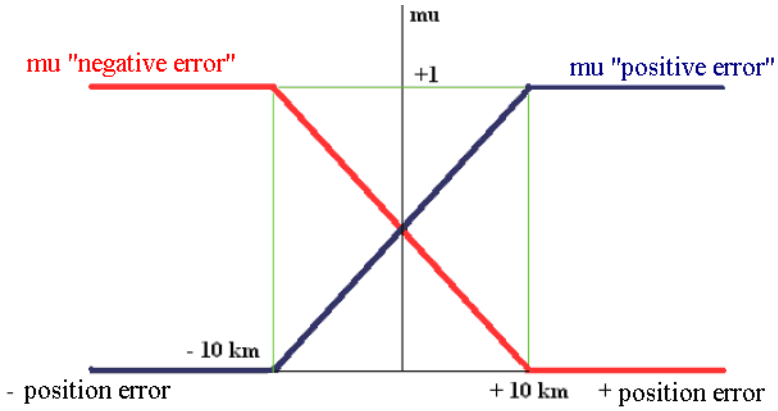

Figure 20: Fuzzy sets of the fuzzy P and PD controllers

The defuzzyfication formula ii) is used by the fuzzy P controller to compute the value of the thrust:

$$\overline{F}_{thrust} = k\left(\overline{\mu}^{+} - \overline{\mu}^{-}\right) \qquad \text{ii)}$$

where:

$\overline{\mu}^{+}$ is a 3-element vector of input truth values for the fuzzy set "positive error" (one component per spatial direction)

$\overline{\mu}^{-}$ is a 3-element vector of input truth values for the fuzzy set "negative error" (one component per spatial direction)

k is a parameter of the controller, which represents the maximum thrust that can be produced by the propulsion system of the satellites.

Equation ii) is a representation of the fuzzy rules that can be analyzed as follows: the value of the thrust (control signal for each dimension of the space) has a sign such that the satellite shall approach the Lagrange point, reducing the error, thus reducing the truth values of the input fuzzy sets. Indeed, the equilibrium is achieved if and only if the position error is null, in each direction. The value of the propulsion is bigger if any of the two μs increases.

The results of the P controller can be seen in Figure 21 (for L3, similar to L1 and L2) and Figure 22 (for L4, similar to L5), improving the results of the conventional P controllers. The position errors in L3 are stable in the long term. Errors in L1 and L2 are reduced to only 2 cm, while the biggest error is in L3 and still within 1.5 meters. The

reduction of errors is especially effective for the initial transitory. A slight improvement is appreciated for L4 and L5, as the fuzzy controller tends to smooth the output (thrust). In the five Lagrange points, the errors have been reduced to the order of magnitude of the solar wind and the non-spherical gravitational fields, while the effects due to the eccentricity of the orbit of the Moon has virtually disappeared.

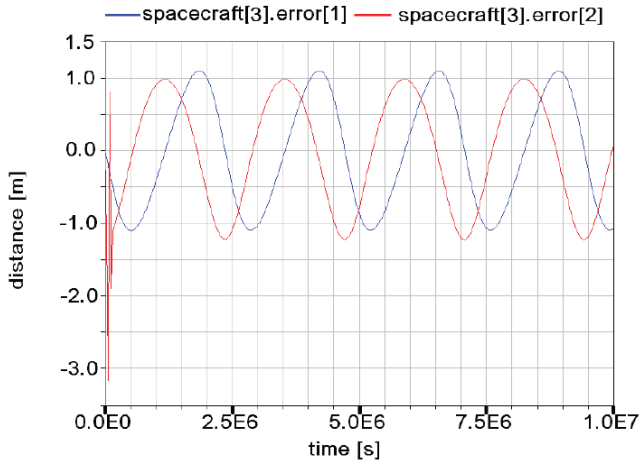

Figure 21: Position errors of a satellite in L3 (fuzzy P controller)

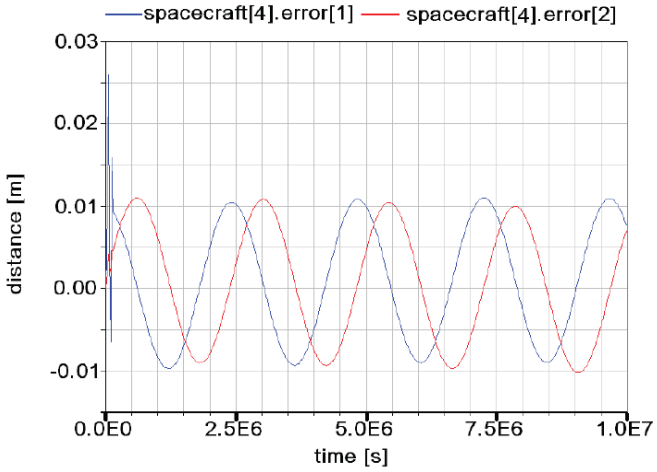

Figure 22: Position errors of a satellite in L4 (fuzzy P controller)

Similarly to the P controller, the PD controller in [10] is based in the fuzzy sets labelled as “positive error” and “negative error”. For the derivative of the error there are other two fuzzy sets labelled as “positive derivative error” and “negative derivative error”. Formula jj) is in use for the PD controller:

$$\overline{F}_{thrust} = k\left(\mu^{++} - \mu^{--}\right) \qquad \text{jj)}$$

where:

$\overline{\mu}^{++} = \min\left(\overline{\mu}^{+}, \overline{\mu}^{d+}\right)$, where $\boldsymbol{\mu}^{d+}$ contains the truth values of the fuzzy set for "positive derivative error" for each spatial component.

$\overline{\mu}^{--} = \min\left(\overline{\mu}^{-}, \overline{\mu}^{d-}\right)$, where $\boldsymbol{\mu}^{d-}$ contains the truth values of the fuzzy set for "negative derivative error" for each spatial component.

According to jj), the thrust (control signal, in each direction) is applied when the satellites is far from the satellite and moving away, thus the PD controller applies less often than in the case of the fuzzy P controller.

As appreciated in Figure 23 and Figure 24, the fuzzy PD controller improves the position error. The initial transitory has been removed, however the position error has increased slightly.

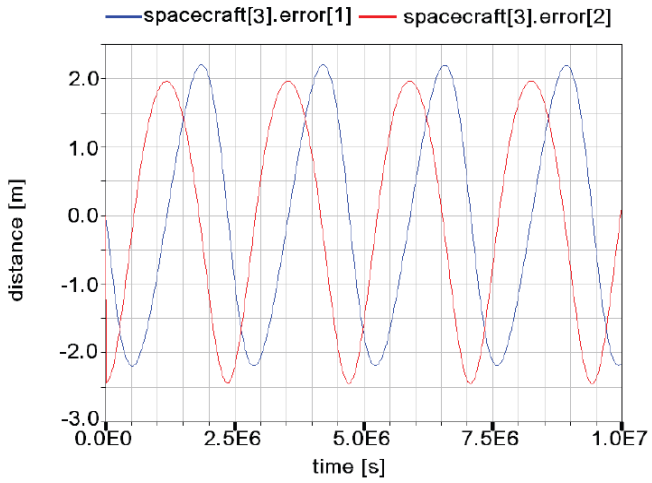

Figure 23: Position errors of a satellite in L3 (fuzzy PD controller)

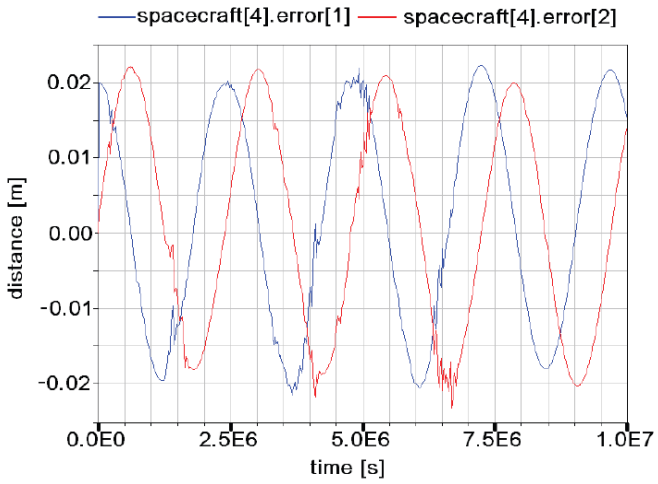

Figure 24: Position errors of a satellite in L4 (fuzzy PD controller)

3.4 Conclusions

In [4] and [10] several objects and models were implemented in Modelica with the purpose of simulating two planets and their libration points, and analyzing the behaviour of the satellites placed in their surroundings. Several effects have been introduced such that the system is quite realistic and definitively far from any theoretical or ideal condition: drift in gravitational parameters of the Moon, eccentricity of the orbit of the Moon, solar wind, non-spherical gravitational fields, etc. P and PD controllers, fuzzy and classic, were implemented in order to reduce or remove the position errors, with successful results. The models in [10] are developed with Modelica, which is useful for the modelling and simulation of dynamic systems, whose parameters vary in time [76]. Several objects are implemented, so they can be used and combined easily to produce all the models seen in this section. Those objects were design specifically for aerospace applications, filling a gap in the standard libraries of *Modelica*. It was an advantage to implement the models with a modelling language instead of using a traditional language, as the equations could be written and coded directly (quite simpler than directly solving for the positions of the Lagrange points and implementing the solution in another language). The results included plots and animations based on the software *Dymola*. A summary of the results is presented in Table 1.

	P	PD	Fuzzy P	Fuzzy PD
L1	2	2	0.02	0.4
L2	0.02	0.08	0.02	0.04
L3	40	12	1.5	2.5
L4	0.12	0.12	0.012	0.025
L5	0.12	0.12	0.012	0.025
Transitory	10^6 s	-	10^6 s	-

Table 1: Position errors (m) of the satellites in the Lagrange points, for conventional P, conventional PD, fuzzy P and fuzzy PD controllers

Four different controllers were implemented: P and PD, classic and fuzzy. All of them were useful to stabilize the position of the satellites in the Lagrange points. The behaviour of the satellites in L4 and L5 was especially good, and the position errors have virtually disappeared. For L1, L2 and L3, both P controllers conventional and fuzzy are quite feasible and successful. The PD controllers are better, as the initial transitory was removed and steady errors were reduced.

In general, the energy required by the propulsion system is an important parameter and a limitation to the length of space missions, due to the fuel budget. For all the satellites, the implemented controllers require a quite low propulsion, thus ensuring longer mission lives. Additionally, if the propulsion system requires an extremely low thrust (based on ions or plasma [107]), this energy does not depend on fuel budget and could be achieved using solar panels; in this case, the limiting parameter of the mission would not be the fuel budget (energy), but the power of the solar panels. As shown in Table 2, the maximum power for all satellites and controllers is quite low in general, and particularly low for the PD controllers. The values for L1 and L3 are especially better if a fuzzy PD is used instead of a classic PD. All these power levels can be obtained from solar energy with the modern electric propulsion systems, based on ions and plasma, such as the GOCE mission of ESA [36].

	P	PD	Fuzzy P	Fuzzy PD
L1	85	70	65	55
L2	1.2	0.40	1.0	0.40
L3	75	65	65	55
L4	0.5	0.35	0.5	0.35
L5	0.5	0.35	0.5	0.35

Table 2: Maximum power (W) of the propulsion systems of the satellites in Lagrange points, for conventional P, conventional PD, fuzzy P and fuzzy PD controllers

The results could improve with other types of controllers (for instance, non-linear controllers), however this is not strictly necessary, in sight of the results presented in this section. The approach followed in [10] is quite innovative and different from that used in other space missions. For example, in Herschel-Planck [58], the intention is not to exactly reach the Lagrange point L2 (the only one, in that mission), but to move around it, keeping an arbitrary variable distance and correcting drifts with manoeuvres every four or six weeks. The applied manoeuvres are impulsive, while in [10] the thrust is continuous in order to keep each satellite exactly in the Lagrange point. An amalgamation of the two approaches should be interesting, combining the aim of keeping a distance to the equilibrium position with a continuous control based on P and PD, conventional and fuzzy, and based on a continuous thrust provided by ion and plasma drives, which permit a low thrust for very long periods of time. Such mixed approach could also include other control strategies, based on PIDs, filters, non-linear control, etc, as well as other perturbations and models. The new lines of research, open by [10], are endless and interesting.

Chapter 4
Genetic Algorithms Applied to Launchers Optimisation

The aim of this chapter is to present the application of Genetic Algorithms (GAs) to the optimisation of launcher trajectories and launchers design. This is an interesting field in aerospace that requires the support of complex models and simulations, where analytical solutions are not possible. As in any other field of engineering where there is no obvious strategy in the search of a solution, heuristic methods such as GAs provide an automatic tool for finding the optima. Three cases are studied in this chapter. A first approach to the problem is given in [1] (see section 4.1), where a vertical launch is studied, focusing on the design of the ascent vehicle given the trajectory. Following that article, the subject was further analyzed in [2][3][5] (see sections 4.2 and 4.3), covering the design and the optimisation of two trajectories for a given launcher, using more realistic trajectories.

4.1 Fuel Optimisation of a Two-stage Launcher following a Vertical Trajectory

4.1.1 Introduction

The purpose of this section is to present and explain the application of a multi-criteria genetic algorithm (GA) to the optimisation of a vertical launch, with the aim of maximizing the lineal momentum [1]. The launcher has two stages, each one with a certain amount of propellant. The aim of the GA is to take a decision on the amount of propellant for each stage.

Such preliminary analysis is interesting from a theoretical point of view, while normally there are more complex launch strategies to place an ascent vehicle in orbit [2][3][5] (see sections 4.2 and 4.3 of this PhD thesis). The content of [1] may have some potential applications as well in far future [101], however [1] is a proof that, nowadays, a purely vertical launch is not feasible. The content of the article should be understood as a preliminary analysis of the potential applications of GA to launchers, as such experience eventually led to more sophisticated results in [3][5].

To solve for the optimum vertical launch, a GA was implemented in Matlab, and the model of the launcher was implemented in Simulink. In later works [2][3][5], Simulink was dismissed in order to improve performance. Each of the two stages of the launcher are implemented as different Simulink models, which compute and integrates the motion equations of the ascent vehicle, providing values for height, speed and acceleration continuously during the whole simulated mission. The GA tunes the input variables of the two models, until the optimum solution is found.

The aim of the problem is to define the amount of fuel in each stage of the launcher, so the linear momentum is maximized by the end of the mission. The initial mass of the ascent vehicle is 2.500.000 kg, including dry and fuel masses; as a result, the higher the fuel mass, the lower the payload and dry mass, and vice-versa. Once the fuel of the first

stage is over, the dry mass of that stage (including rockets, motors, fuel storage deposit) detaches from the main body, so the total weight of the spacecraft decreases. According to this criterion, the first stage should detach as soon as possible. However, attending to the fact that the first stage is more powerful, the detachment of such stage should occur as late as possible. Hence, there are two opposite criteria to design the amount of fuel of the first stage, and ultimately to produce the detachment sooner or later.

In both the first and second stages, more fuel leads to a longer impulse time, at the cost of increasing the initial weight of the spacecraft. If the amount of fuel is low, the weight decreases, but the fuel will run out quickly. With too much fuel, the spacecraft shall be too heavy to move at all.

The linear momentum is the product of the mass and the velocity. Thus, ideally the problem is solved when both quantities are as high as possible. However, a high mass results in a lower acceleration, thus the final speed is also lower, eventually decreasing the linear momentum in spite of the higher mass. If the total initial mass is fixed (as in this section), a big amount of fuel improves the final speed, but this implies a lower final mass, thus eventually a lower linear momentum shall be achieved.

Final mass, by itself, is important in aerospace missions, and it represents the payload (useful load) of the mission (scientific material, astronauts, etc). In other problems, this final mass may become the parameter to maximize, but this is not the case in this section. The purpose is to maximize the final linear momentum. This maximization problem is analogous to that when the objective is to target and impact space objects in order to modify their orbit, such as missiles [20][21], space debris [83][43] or meteors.

4.1.2 Implementation of the GA for Momentum Optimisation

Information on GAs can be found on [52][72]. The genetic algorithm shall maximize the final linear momentum of the spacecraft (1), by adjusting the initial fuel of the first and the second stages during the launch. Limit values of these design parameters are 0 and 1.000.000 kg.

$$z = m_f \cdot v_f \qquad (1)$$

where m_f is the final mass of the launcher and v_f is the final speed.

The figures representing the amounts of fuel has been coded into binary, and put together as the chromosomes of a chain of bits. There are 14 bits per parameter (first stage fuel, second stage fuel), so there are 268435456 possible combinations, hence confirming the problem cannot be solved by random "try and error" techniques. The optimal solution cannot be found with analytical methods, due to the complexity of the problem and of the involved models (see section 4.2.3).

Each GA offspring consists on a population of 10 individuals. Total number of offspring is 300. The initial population is created randomly. A valid method to select individuals for crossing is to choose those with the best performance index. This leads to improving the individuals of the next offspring, at the cost of falling into local minima as the population becomes homogeneous. Thus, a random selection has been implemented, with a higher probability of choosing the individuals with a higher performance index.

Once the individuals have been selected for a potential crossing, the probability of applying a crossing to each individual is 0.8; this means that 20 % of the chosen individuals will be copied into the next offspring without any modification, while the remaining 80 % will be crossed with each other to generate new individuals. The crossing is based on a one-point crossover, and the two parents disappear from the population.

After the crossing operation, the mutation operator is applied, with a probability of 0.001 for each bit; this means that some individuals may remain unchanged, some others may change in one bit; and some others may change in several bits (highly improbable, though).

4.1.3 Description of the Optimisation Problem

Final mass and speed are necessary to compute the performance index (1). The final mass depends on the fuel mass of the two launch stages, while the final speed depends on the thrust produced by the spent fuel in those two stages.

The final mass can be computed according to (2):

$$m_f = m_0 - m_{c1} - m_{i1} - m_{c2} \qquad (2)$$

where:

m_0 is the initial mass of the spacecraft

m_{c1} is the fuel mass of the first stage

m_{i1} is the dry mass of the first stage

m_{c2} is the fuel mass of the second stage

The dry mass of the first stage consists on motors, rockets and other structures that are useless once the fuel is over, thus this dry mass detaches so the total mass of the spacecraft is lower. The value of m_{i1} is 1.000.000 kg. The second stage has not dry mass, as the target of the criteria function (1) is to maximize the linear momentum, thus getting rid of this stage implies reducing the final mass and lowering the performance index. Hence, there is no reason to detach the second stage to decrease the weight, as there is not further acceleration of the spacecraft.

Final height of the spacecraft is not important in this problem. However, some reasonable considerations have been taken into account. If only one, very heavy stage is used, the final linear momentum shall be extremely high, even if the final speed is low. Such a design would maximize the output of equation (1), however, such a design is not reasonable and at least a height of 200-300 km is expected to consider that the spacecraft has a reasonable mission. A restriction has been implemented in the model, to consider valid only those launches that reach 250 km. Apart of this, there is no further interest in final height. Some additional restrictions have been imposed to ensure that the final mass of each stage is not null, otherwise this means that the whole

launcher is made of fuel, with no physical sense. A boundary condition ensures that the final mass of the first stage is not lower than the fuel mass of the second stage. Also, if the initial thrust at launch is not enough to accelerate the total initial mass of the spacecraft, the performance index is set to zero, so that solution is discarded for crossing. Analogously, only a positive acceleration at the beginning of the second stage firing is allowed.

Final speed of the spacecraft can be computed by solving for the dynamics equations of the launcher, including models for the air drag, the gravity and the thrust. Assuming that the positive forces on the spacecraft are upwards, the total force on the spacecraft is given by (3):

$$F_T = T - D - P \tag{3}$$

where:

F_T is the total force on the spacecraft

T is the thrust

D is the air drag

P is the gravitational force

The first stage uses solid fuel which specific impulse (see section 2.2.8) is 5.000 m/s, while the mass flow is 10.000 kg/s. The specific impulse of the second stage is 7.000 m/s, with a mass flow of 2.000 kg/s. The calculation of the thrust for the first and second stages is as in (4):

$$\begin{aligned} F_1 &= I_{sp1} \cdot \dot{m}_1 = 50MN \\ F_2 &= I_{sp2} \cdot \dot{m}_2 = 14MN \end{aligned} \tag{4}$$

where:

I_{spi} is the specific impulse for stage i

$\dot{m}_i$ is the mass flow through the nozzles in stage i

Air drag can be computed as explained in section 2.2.10. To calculate the air density, an exponential model was used in [1], based on equation (5):

$$\rho(h) = k_1 \exp(-k_2 h) \tag{5}$$

where:

k_1 and k_2 are constants

h is the height of the launcher

For $h = 0$, air density is that at sea level, which value is around 1.2 kg/m^3 as in equation (6):

$$\rho(h = 0) = k_1 = 1.2 kg / m^3 \tag{6}$$

Atmospheric pressure at sea level can be computed by integrating (5) in a "column" or air as in (7):

$$P = \frac{m_{air} g}{A} = \frac{g}{A} \int_{h=0}^{h=\infty} k_1 \exp(-k_2 h) dV \tag{7}$$

where:

P is the pressure at sea level

$m_{air}g$ is the weight of the "column" of air

g is the gravitational acceleration near the surface of the Earth (nearly constant)

A is an arbitrary area at sea level, base of the air "column"

dV is a differential volume of air

Formula (7) is valid only if the atmosphere is modelled as a thin lawyer in comparison to the radius of the Earth. This condition is necessary due to geometric considerations, but also in order to consider a constant gravitational acceleration. Taking into account that the atmosphere finishes at height of around 100 km (where air density is around 10.8 mg/m^3), and that the radius of Earth is around 6370 km, formula (7) should be valid. Nevertheless, errors from this estimation are smaller than those due to random factors: weather conditions, non-homogeneous atmosphere, etc. Solving for (7) yields to (8):

$$P = \frac{g k_1}{k_2} \tag{8}$$

where obviously $k_2 = 0.00011618 m^{-1}$.

As explained in section 2.2.10, the effective drag area is equal to the product of the cross area and a coefficient, which in this case has been considered as 0.25. For a cross area of 35 m^2, the effective drag area is around 10 m^2.

Gravitational force has been computed according to section 2.2.1, with some numerical considerations to avoid multiplying or dividing by large numbers, as explained in section 4.2.3 and in [2][3].

4.1.4 GA Optimisation Results

Figure 25 shows the evolution of the results in every new offspring (best value of each offspring, and best historical value). As shown in this figure, the initial random individuals are crossed and the performance index improves.

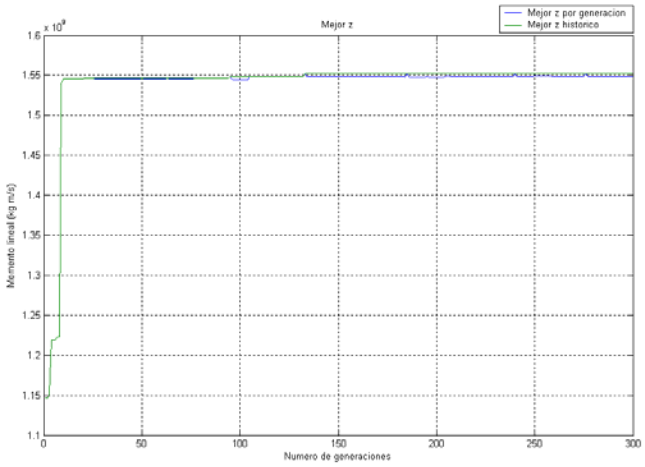

Figure 25: Evolution of individuals (GA applied to vertical launch)

In the first offspring (left side of Figure 25), the optimum solution improves very quickly. Afterwards, the improvement is much slower. The final best performance index (1) by the end of the execution of the GA is *1.5527 x 10^9 kg·m/s* (linear momentum) for m_{c1} *= 813.049 kg* and m_{c1} *= 432.128 kg*. The evolution of the height for this solution is represented in Figure 26, the speed evolution in Figure 27 and the acceleration evolution in Figure 28. A final height of around 700 km is achieved, before the second stage runs out of fuel. Final speed is around 6 km/s, which is a quite reasonable speed

for Low Earth Orbits (LEO), however note that in this section the problem consisted on a vertical launch, rather than on a final orbit as in [2][3][5]. Final acceleration is above 45 m/s^2. Acceleration increases as air drag and gravitational forces decreases. The latest decreases as the height increases (see section 2.2.1), and as the fuel is expelled through the nozzles, thus reducing mass. Acceleration decreases suddenly when the first stage is discarded and the second stage is fired, due to the lower thrust of the second stage in comparison to the first (4); mass also decreases as the first stage detaches, but this is not enough to compensate the lower thrust, thus acceleration jumps.

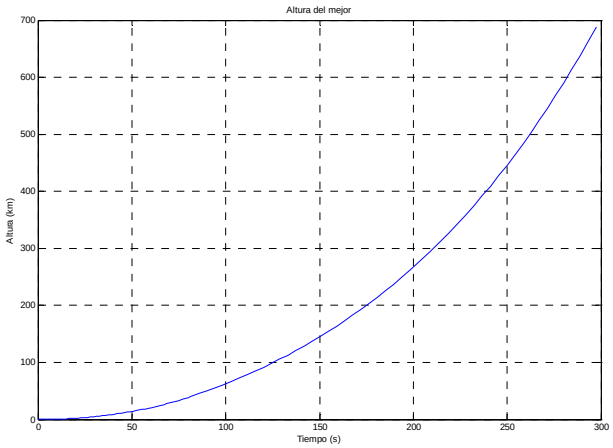

Figure 26: Height (GA applied to vertical launch)

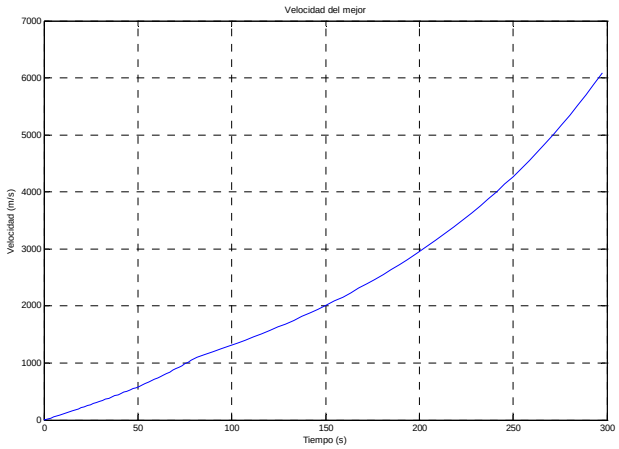

Figure 27: Speed (GA applied to vertical launch)

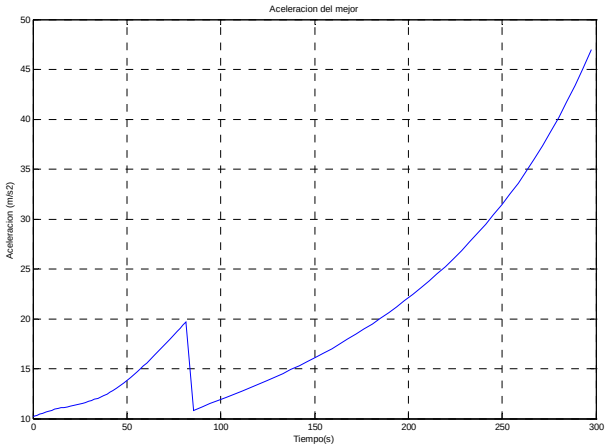

Figure 28: Acceleration (GA applied to vertical launch)

4.2 Design of Ascent Trajectory to a Final Stable Circular Orbit

4.2.1 Introduction

The aim of this section is to explain and analyze the results obtained after using a genetic algorithm (GA) to optimize the ascent trajectory of a conventional two-stage launcher [2][3]. The purpose of this optimization is to obtain a final stable circular Low Earth Orbit (LEO) with maximum height. A subset of parameterized variables are tuning and optimized by the GA in order to achieve the mission purpose while satisfying the constraints and maximizing the final height of the orbit, given the technical characteristics and limitations of the launcher design, and a fixed weight of the payload. Indeed, due to the complexity of the implemented models and the lack of analytical solutions to the required equations, an alternative non-traditional optimization algorithm such as GA is necessary to perform the optimization task, using a variable mutation rate that broadens the search area whenever the population becomes uniform, avoiding local maximums and exploring for global ones.

The preliminary analysis considered in the Automatics Workshop hold in Spain on September 2007 [1] led to a successful set of results that highlighted the added value of genetic algorithms utilization for the optimization of tuning parameters in complex

systems, where common optimization techniques are not applicable. In particular, genetic algorithms are useful for optimizing the ascent trajectory of launchers. A more ambitious research was performed in [3] based on a realistic approach to the launch optimization problem, which final purpose is, in this case, the injection of the spacecraft in a circular LEO with maximum height. While the GA in [1] focused in the spacecraft design, the article in [3] focused in the ascent trajectory parameters. An even more sophisticated approached, solving for two different trajectories, was later on presented in [5] (see section 4.3).

In [3] a realistic model has been implemented based on the Ariane 5 technical data [87]. This commercial launcher consists on two rocket-propelled stages plus a payload, which depending on the mission usually is nuclear power propelled and is not considered part of the launcher. The first stage provides thrust during the vertical trajectory, pitch-push over and gravity turn phases. It consists on two P238 engines attached to a main H158 engine. This set of engines is capable of providing a very high demanding thrust, in order to counteract both the gravity field of the Earth when it is at its maximum value, and the air drag of the atmosphere when the air density is the highest. The spacecraft relies on these extremely powerful engines during that first stage, when the ascent trajectory is mostly vertical. These engines are very heavy, therefore they are discarded once the thrust demand drops in higher, less dense atmosphere layers, when the vertical component of the speed is less important, and at that point a lighter and less powerful engine is used instead. The two P238 are discarded once their fuel is exhausted, still during the first stage, and then the launcher relies on the H158 alone, until all of its fuel is burnt and the second stage is switched on. The second stage, which includes an Aestus engine, provides the thrust necessary to reach a LEO orbit, applying a bilinear tangential manoeuvre. The launch mission is completed once the LEO is reached. Then, the second stage and other remains of the launcher are discarded altogether. The payload relies in its own propulsion system to perform the defined space mission.

A fixed payload weight is considered. The technical data of the Ariane 5 engines specify the thrusts and the burn times of the first and second stages of the mission. The tuning parameters of the ascent launch trajectory are the vertical ascent trajectory period, the pitch-push over period and the pitch-push over final angle. These three tuning parameters have a great impact in the final radius of the LEO and in the final

speed of the spacecraft. Thus, the target multi-criteria function of the GA considers both the LEO height and the similarity between the final speed and the circular orbit speed for the achieved orbit. Taking into consideration these mission specifications and constraints, the target function is implemented and maximized by the GA, tuning the three input parameters accordingly. The multi-criteria optimization algorithm refers in this case to an algorithm that combines different criteria into a single criterion, and it does not consider different criteria separately, which would result in non-dominated solutions [47] as explained in sections 5.2.3 and 5.4.

4.2.2 Implementation of GA for Height Optimisation

In [3] three variables in a ten bits codification define a net of some 1 billion nodes. Thus, using an algorithm based in an intelligent exploration of solutions such as GA is necessary. In this case, the GA explores 20 individuals during a maximum of 100 generations, with a mutation rate that increases as the individuals in the population become more similar between them. When no improvement occurs during 15 generations, convergence is assumed and the algorithm is interrupted.

The function to maximize (criteria function) is represented in equation (9):

$$z = \frac{h_f}{1 + \left| v_f - v_{orb} \right|} \tag{9}$$

Where:

h_f is the final altitude of the launcher at the end of the launch mission (i.e. the height of the stationary LEO),

v_f is the final launcher speed (which should be purely horizontal, after applying the bilinear tangential manoeuvre), and

v_{orb} is the necessary spacecraft speed to maintain the final circular LEO in the long-term [20], as calculated by equation (10), which guarantees the equilibrium between the gravity and centrifugal forces at the rotating reference frame:

$$v_{orb} = R\sqrt{\frac{g}{R+h}} \tag{10}$$

In the denominator of equation (9) a unitary term has been added in order to prevent improbable but still possible divisions by zero during the GA execution. The variables involved in equation (9) depend on the tuning parameters of the manoeuvres performed during the launch mission. Thus, the multi-criteria target function strongly depends on the vertical ascent trajectory period (duration of the first manoeuvre), the angle at the end of the pitch-push over manoeuvre (which is the angle at the beginning of the gravity turn manoeuvre), and the period of this pitch-push over manoeuvre.

The GA should be able to maximize the target function, so the altitude of the final LEO is high whereas the difference between the theoretical circular LEO speed and the actual final launcher speed are kept similar. The target function is complex and involves several numerical simulations per individual in the space of search.

4.2.3 Description of the Optimisation Problem

The first stage of the modelled launcher, based on Ariane 5 [87], consists on a cryogenic H158 [105] main stage which hosts a large tank with two compartments, one for 130 tonnes of liquid oxygen and one for 25 tonnes of liquid hydrogen. This stage also hosts a Vulcain engine at the base with thrust of more than one meganewton (MN). This stage weighs about 15 tonnes when empty. Attached to the sides are two solid propellant P238 boosters. The second stage of the launcher is on top of the main stage and below the payload. It consists on an Aestus engine [19] fuelled by high volatile liquids such as monomethylhydrazine and nitrogen tetroxide. A summary of the two stages technical specifications implemented in the model is presented in Table 3. The flow of mass thought the nozzles should be taken into account to calculate the current mass of the spacecraft during each stage, according to equation (1) in section 2.2.8 of this thesis.

	H158	P238	Aestus
I_{sp} (m/s)	4,300	2,750	3,400
Propellant mass (kg)	155,000	277,000	8,910
Empty stage weight (kg)	15,000	4,000	150
Burn time (s)	589	129	1,100
Mass flow (kg/s)	263	2,147	8.1
Provided force (N)	1,118,000	6,050,000	27,540

Table 3: Facts sheet for H158, P238 and Aestus engines

The launch mission consists on several manoeuvres [106], whose effects are simulated and analyzed well in advance before a launch takes place, during the design phase. However, normally dynamic corrections of the trajectory are mandatory during the launch mission, due to the presence of undesired effects such as atmospheric weather conditions. Nowadays, several on-board and on-site computers carry out the necessary calculations based on the sensors' data, in order to apply dynamic corrections of the ascent trajectory once the launcher mission has begun.

The first manoeuvre is the vertical launch ascent. The launcher ascent trajectory is purely vertical during some seconds. No gravity turn can be flown from the start in a conventional launch site. The duration of this manoeuvre is a tuning parameter, to be optimized by the GA.

The second manoeuvre is the pitch-push over. A gravity turn cannot be flown unless some flight angle is achieved, otherwise the launcher ascent trajectory shall remain purely vertical (see [1] and section 2.2.9 of this thesis). There must be some push over manoeuvre to get the trajectory out of the vertical direction and establish a defined flight-path angle, which value, a GA tuning parameter, is typically small. The dynamics involved in this manoeuvre are highly complex, and its duration is short. This duration is also a GA tuning parameter. Based in small duration and small final angle, the thrust and velocity of the launcher are assumed to remain nearly in the same direction, thus a valid approximation for this manoeuvre is a constant angle rate based in an average value.

The third manoeuvre is an elementary gravity turn. While rocket thrust serves for acceleration along the current velocity vector, the gravitational acceleration turns the direction of the speed, giving shape to a typical trajectory that is extremely sensitive with respect to the state vector at the beginning of the manoeuvre. The equations governing the motion during the gravity turn are presented in section 2.2.9 of this thesis.

During the vertical launch ascent, the pitch push-over manoeuvre and part of the gravity-turn manoeuvre, the full first stage of the launcher (H158 + 2xP238) provides the required very high thrust. During the gravity-turn manoeuvre, the two P238 boosters are discarded and the manoeuvre continues relying in the H158 engine alone, until all of its propellant is exhausted. At this point, the H158 engine and the remains of the first stage are discarded, then the second stage is ignited and a bilinear tangent manoeuvre is performed and the thrust is provided by the Aestus engine alone. This fourth manoeuvre is an optimal guidance law in presence of a uniform gravitational acceleration [106]. Indeed, in this case, the bilinear tangential guidance law is nearly optimal since the gravitational acceleration is nearly uniform. The formulation of the manoeuvre [25] is shown in equation (11):

$$\tan\gamma = \left(\tan\gamma_f - \tan\gamma_0\right)\frac{t}{t_f} + \tan\gamma_0 \qquad (11)$$

where γ_0 is the flight path angle at the beginning of the manoeuvre, γ_f is the flight path angle at the end (usually, like in this case, equals to zero, since the aim of the bilinear tangential manoeuvre is to place the spacecraft in a circular orbit), and t_f is the duration of the manoeuvre. This duration is based on the technical specifications of the Aestus engine [19], given the mass flow of burn propellant and the total amount of propellant available at the beginning of the second stage. Once the propellant is fully exhausted, the bilinear tangent manoeuvre should be completed and the second stage discarded. At this point, the payload mission begins.

The air drag force is provided by equation (6) in section 2.2.10.The effective area may be calculated as the multiplication of the transversal area and a non-dimensional air drag coefficient. This coefficient depends on the limit thermal layer at the launcher surface, and also on the launcher geometry, among other relevant factors. The effective air drag

area may be 25 % of the actual transversal area if the launcher is very aerodynamic. In this case, a 50 m^2 effective area has been considered for the full first stage, 30 m^2 for the H158 stage alone and 10 m^2 for the second stage. One of the most important models is the U.S. Standard Atmosphere 1976 one [100].

Earth's gravity field [20] pulls the spacecraft down with a value given by the Newton's Universal Gravity Law, as explained in section 2.2.1. The distance from the spacecraft to the centre of the Earth equals to (12):

$$r = R + h \tag{12}$$

where R is the radius of the Earth (6,370 kilometres approximately) and h is the height of the launcher respect to the Earth's surface. When $h = 0$ (at Earth's surface), the gravitational force equals mg, as shown in (13):

$$F(h=0) = mg = \frac{GMm}{R^2} \tag{13}$$

Thus, equation (14) is obtained:

$$GM = gR^2 \tag{14}$$

Substituting (12) and (14) in (1) yields equation (15), which is more convenient than (1); indeed, in equation (15) variables directly related to the launch mission are involved for the calculation of the gravity force on the launcher, whereas the multiplication of very large and small figures is avoided.

$$F(h) = gm\left(\frac{R}{R+h}\right)^2 \tag{15}$$

4.2.4 GA Optimisation Results

In Figure 29 the evolution of the performance index is shown for the simulated generations. Each n-th line points out the value of the n-th best solution per generation. Best six solutions per each generation are highlighted. The performance indexes of the randomly generated initial individuals are improved as the GA performs the crossing and mutating operations along the generations. In each generation, results are clearly

improved while the GA searches for the final optimum solution. It may be assumed that this final solution is the best possible one for this particular problem; otherwise the solution provided by the GA could be refined by using other optimization techniques. If this is the case, the solution provided by the GA may be used as an initial guess.

There are some erratic lines in Figure 29, which are associated to the worst solutions per generation. Indeed, bad solutions appear randomly at any moment during the GA execution, particularly due to the mutation operator, and sometimes also due to inconvenient crossings between better solutions. In general, bad individuals are discarded in the following generation since the probability of being chosen as a parent is low. It is possible to note in Figure 29 that mutations occur specially when the individuals have similar performance indexes and the population becomes uniform, usually when a local maximum has been reached. Rising dynamic variable mutation rate allows a broader search in the space of potential solutions, avoiding getting stuck in a local maximum of the target function. This is an added value to more traditional implementations of GAs.

Best solutions are clearly stable in Figure 29. The implemented algorithm guarantees that the best solution is never lost when generating a new offspring. This is not guaranteed for the second best and following solutions, but still Figure 29 shows a satisfactory tendency for them. Elitism for the very best solution not only improves the best performance index in the next generations, but also other individuals become better.

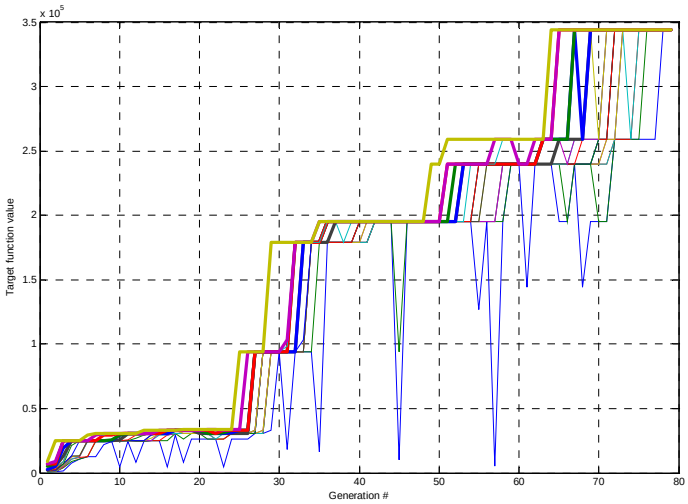

Figure 29: Evolution of performance indexes for every individual in the population along the generations (GA applied to launch trajectory optimisation, final circular orbit)

The best value of the GA target function has been obtained for the trajectory parameters shown in Table 4.

Vertical ascent duration	6.82 seconds
Pitch-push over duration	20 seconds
Pitch-push over final angle	84.784 degrees

Table 4: Values of the parameters for the best GA solution (GA applied to launch trajectory optimisation, final circular orbit)

Figure 30 shows the optimum solution of the altitude evolution of the launcher for different downrange distances. The altitude evolution in time is shown in Figure 31. The speed evolution is presented in Figure 32, whereas the acceleration evolution is shown in Figure 33. All the plots are divided in areas according to the usage of engines in each stage: first, H158 engine with two P238 boosters for vertical ascent, pitch-push over and beginning of the gravity turn manoeuvre; second, H158 engine alone for the rest of the gravity turn manoeuvre; third, Aestus engine for bilinear tangent manoeuvre; and finally, free drift, whose representation allows to check the successful stability of the achieved LEO. It is interesting to note that the H158 + 2xP238 stage duration is really

short in comparison to the other stages of the launch mission, since the two P238 boosters are discarded relatively soon.

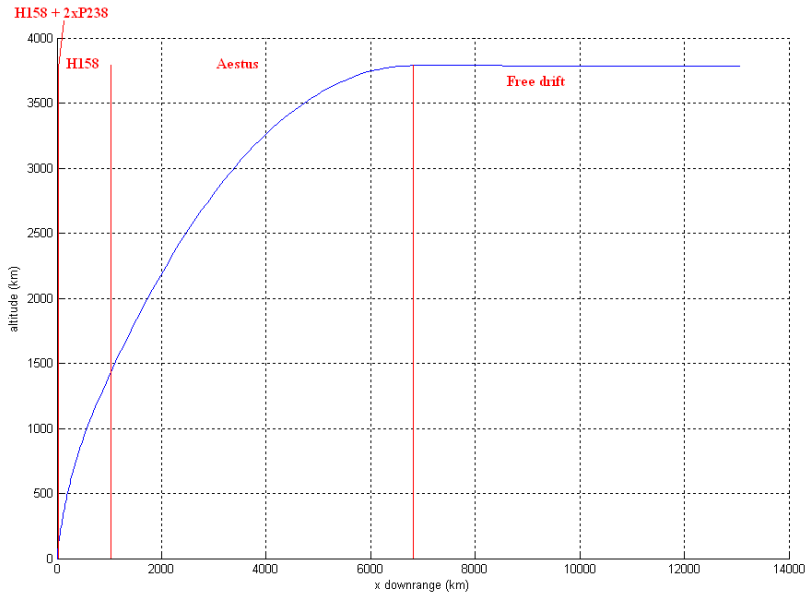

Figure 30: Launcher altitude evolution profile (GA applied to launch trajectory optimisation, final circular orbit)

Figure 30 provides a clear view of the optimum trajectory profile. This shape of the ascent trajectory is typical of Ariane 5 and other launchers. In the best solution obtained from the GA, 20 tonnes has been delivered to a circular LEO. Final altitude is around 3,790 km. Indeed, according to Ariane 5 specifications, it is possible to deliver 20 tonnes of payload to a LEO, which altitude is typically 500 – 4,000 km. This comparison confirms the results described in this work.

Final speed for the optimum solution is around 6.25 km/s according to Figure 32. This means only a 0.16% difference regarding the LEO circular speed for that gravitational radius. Furthermore, the final LEO circular orbit is really stable, as confirmed in Figure 30 and Figure 31. Indeed, the LEO is maintained in free drift after a reasonable number of seconds.

Gravitational corrective manoeuvres would be necessary in the long-term in order to compensate effects such as errors in measurements and in engines, non-spherical effects in Earth's gravitational field, solar wind, other bodies gravitational acceleration (mainly

Sun and Moon), etc [98]. The achieved LEO is not stable after a while due to these effects, of which only the air drag has been considered in the implemented model.

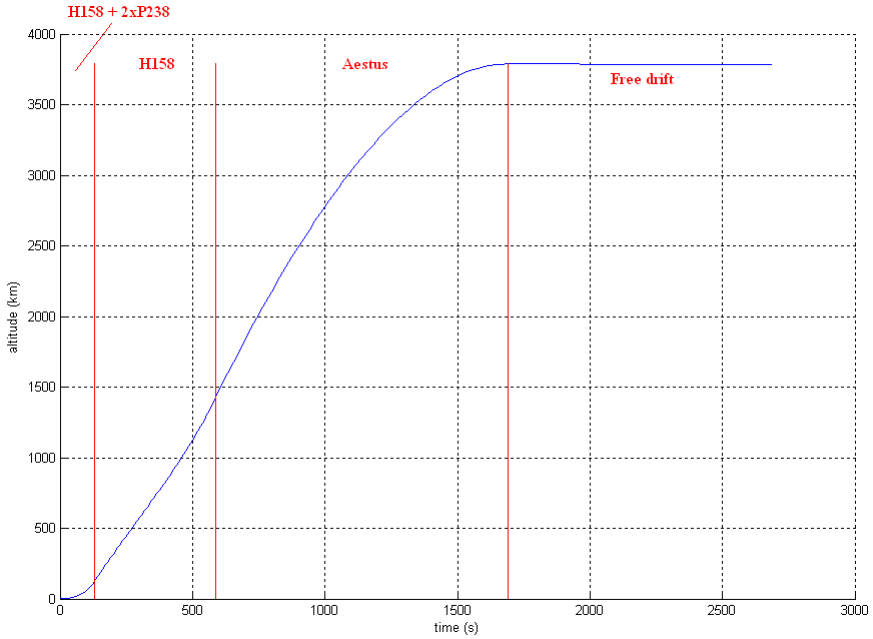

Figure 31: Launcher altitude evolution in time (GA applied to launch trajectory optimisation, final circular orbit)

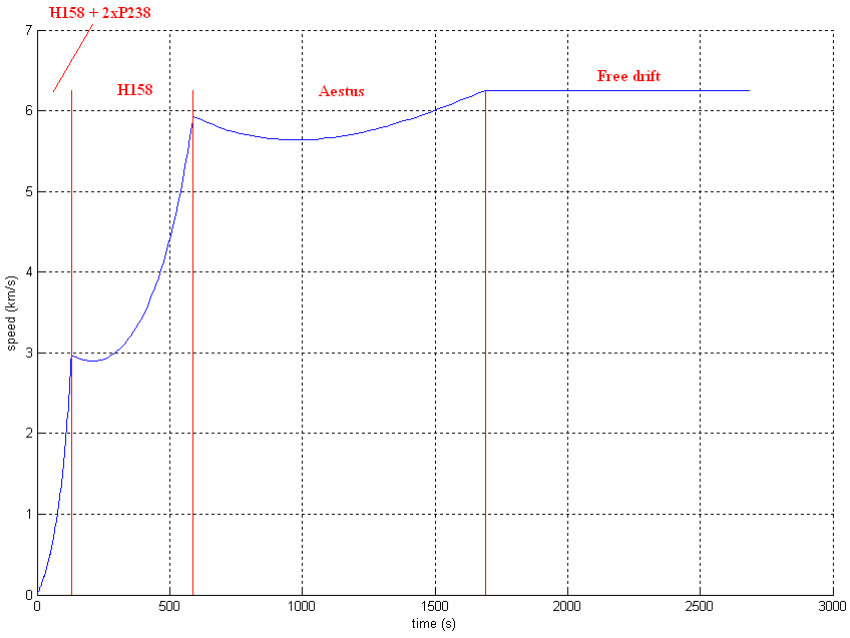

Figure 32: Launcher speed evolution in time (GA applied to launch trajectory optimisation, final circular orbit)

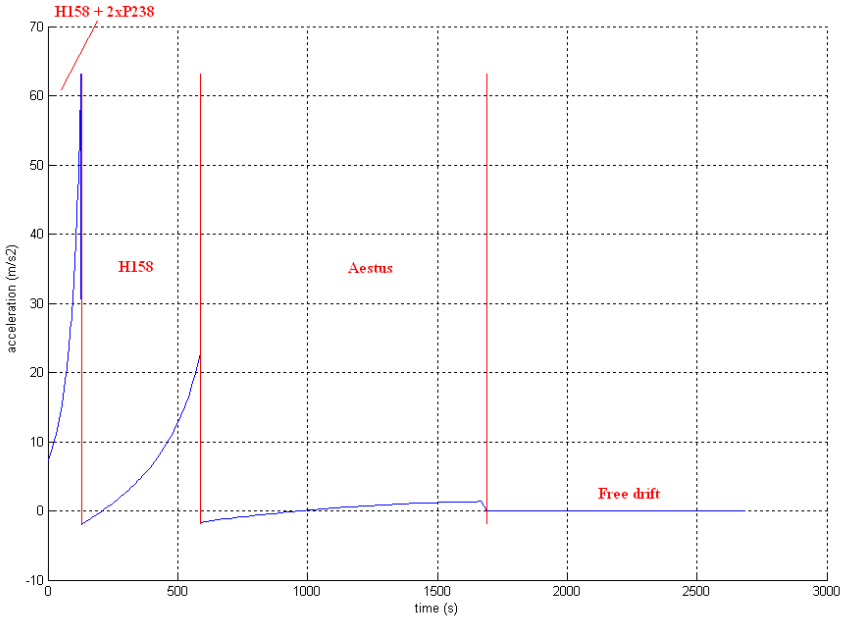

Figure 33: Launcher acceleration evolution in time (GA applied to launch trajectory optimisation, final circular orbit)

As shown in Figure 33, acceleration increases as air drag decreases, due to the rapid decrease of air density. This air drag also descends due to the smaller transversal area of the spacecraft as propulsion stages are discarded. Acceleration also increases when the weight decreases, as the gravitational acceleration decreases and the fuel is burnt and expelled thought the nozzles. Transitions in the acceleration are due to engine changes (with different impulse each) and to the spacecraft mass changes (when a stage is discarded). Indeed, each new stage provides less impulse than the previous one; that is why the acceleration decreases in spite of the discarded weight of the previous stage. The acceleration peaks bring sudden changes in the slope of the speed as shown in Figure 32, and concavity shape changes in the altitude evolution in Figure 33.

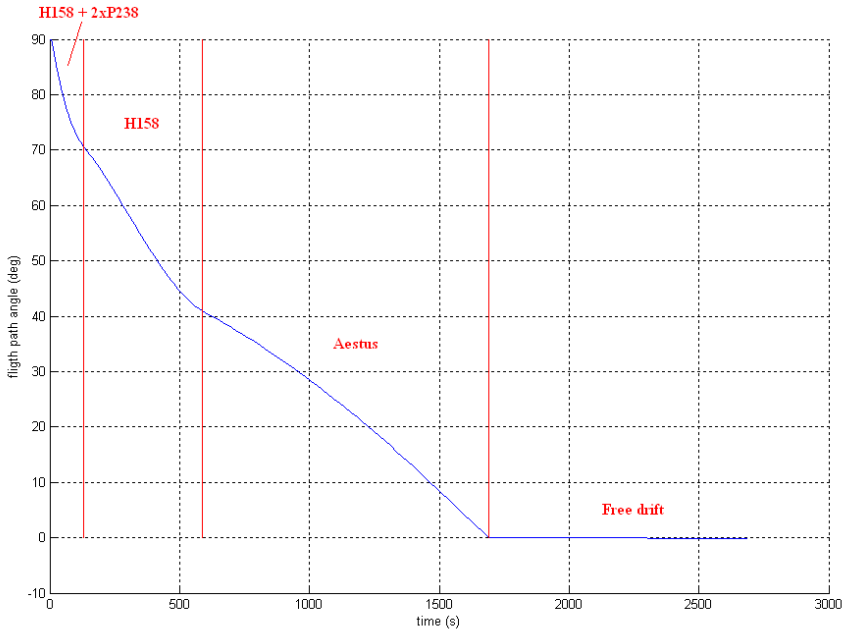

Figure 34: Launcher flight path angle evolution (GA applied to launch trajectory optimisation, final circular orbit)

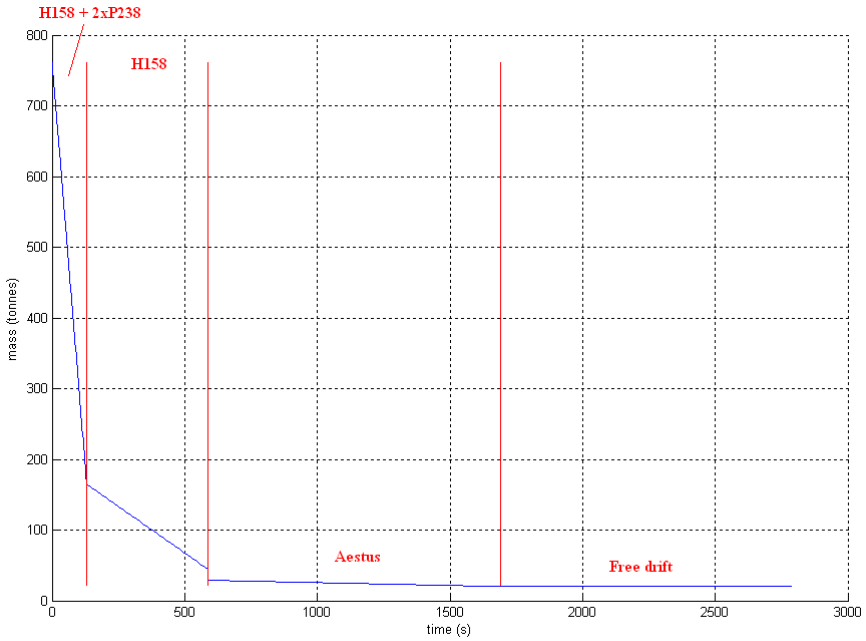

Figure 35: Launcher mass evolution (GA applied to launch trajectory optimisation, final circular orbit)

Figure 34 shows the launcher flight path angle, which evolves from 90 degrees at the beginning of the launch mission to 0 degrees at the end, and then remains constant

during the LEO free drift. The flight path angle also remains constant during the vertical ascent manoeuvre at the beginning of the launch mission, which is not well noticed in Figure 34 due to the short period of this manoeuvre in comparison to the duration of the whole launch mission. For the same reason, the constant angle rate during the pitch-push over manoeuvre is difficult to see either. The evolution of the flight path angle, and the effects of changing active engines, can be clearly seen during the gravity turn. The characteristic shape of the bilinear tangential guidance is appreciated during the Aestus stage, in which the tangent of the flight path angle changes in linear proportion to time.

Figure 35 shows the launcher mass evolution, whose decreasing rate varies depending on each stage: high mass decreasing rate at the beginning, when the H158 engine and the two P238 boosters are switched on; lower mass decreasing rate when only the H158 engine is burning fuel; and the lowest mass decreasing rate during the Aestus stage. At the end, the mass remains constant since no more fuel is burnt. Payload may spend some fuel for mission operations and corrective gravitational manoeuvres, which are not included in the launch mission and therefore have not been considered in the model.

4.3 Design of a Hyperbolical Trajectory for an Ascent Vehicle

4.3.1 Introduction

The aim of this section is to analyse the results obtained when using a genetic algorithm (GA) to optimise the ascent trajectory of a conventional two-stage launcher [5]. Due to the complexity of the implemented models and the lack of analytical solutions to the system equations, and in order to deal with the uncertainty, an alternative non-traditional algorithm provided by soft-computing techniques such as GA is necessary to perform this task. The purpose of the optimisation is to achieve an escape trajectory. A subset of variables are tuned and optimised by the GA in order to achieve the mission purpose while satisfying the constraints and maximizing the goals, i.e., the final height of the orbit in the first case, and the final speed in the second case. The technical characteristics and limitations of the launcher and a fixed payload weight have been

considered. In this case, a variable mutation rate has been implemented that broadens the search area whenever the population becomes uniform. The results are very promising in both cases and can help in the design of a space mission as an initial guess.

The preliminary analysis considered in a previous work [4] led to a successful set of results that highlighted the added value of genetic algorithms for the optimisation of tuning parameters in complex systems, where analytical optimisation techniques are not applicable. In particular, genetic algorithms have been proved to be useful for optimising the ascent trajectory of launchers.

In a later article [3], a more ambitious research was performed based on a new approach to the launch optimisation problem, which final purpose was, in that case, the injection of the spacecraft in a circular LEO with maximum height. While the GA in Chapter 6 was focused on the spacecraft design, in [3] the GA was focused on the ascent trajectory parameters.

A third case was considered in paper [5] as an extension of [2]. Models are improved to make them more realistic, and a GA is used in order to optimise the four manoeuvres of the ascent trajectory of a two-stage launcher in the first stages of an interplanetary mission, from the launching to the final escape trajectory. Due to the complexity of the implemented models and the lack of analytical solutions to the system equations, an alternative non-traditional algorithm provided by soft-computing techniques such as GA is used to perform this task.

As in [3] (see section 4.3), in [5] a realistic model has been implemented based on the Ariane 5 technical data [87], and a fixed payload weight is also considered. Again, the tuning parameters of the ascent launch trajectory are the vertical ascent trajectory period, the pitch-push over period and the pitch-push over final angle. These three parameters have a great impact on the final trajectory of the spacecraft. The target multi-criteria function of the GA considers the final height to ensure air drag free conditions and to maximize final speed, so the three tuning parameters are adjusted accordingly. The multi-criteria optimization algorithm refers in this case to an algorithm that combines different criteria into a single criterion, and it does not consider different

criteria separately, which would result in non-dominated solutions [47] as explained in sections 5.2.3 and 5.4.

4.3.2 Implementation of GA for Speed Optimisation

A major difficulty when optimising parameters in a launch mission is to provide algorithms that converge towards an optimum. Continuation methods, in which a branch of the solution is followed, may be adequate in some cases, and the optimum of a nearby problem may be used as a good initial guess of the solution of the new problem. But even making wise use of the continuation method, a branch of the solution may bifurcate or disappear. And in any case, it is difficult or impossible to prove that the obtained solution is the global optimum and not a local one. Furthermore, completely new missions may arise for which a desired neighbouring problem is absent. Since it is hard to guess where the domain of convergence towards the global optimum is, in a general case, an exploratory work may become recommendable [52][72]. Exhaustive search, in which the space of optimisation variables is explored with a fine grid, is unaffordable except in very simple cases. In the case covered by this article, three variables in a ten bits codification define a net of some 1 billion nodes. Thus, using an algorithm based in an intelligent exploration of solutions such as GA is necessary. In this article, the GA explores 20 individuals during a maximum of 100 generations, with a mutation rate that increases as the individuals in the population become more similar between them. Since there are no means to prove the solution obtained by the GA is the best possible optimisation, some criteria has to be established to stop the GA iterations. In this case, it can be assumed that enough good values have been reached when no improvement has occurred after a certain number of generations. After many simulation experiments, it was concluded that no improvement is likely to happen if the best solution does not evolve after being constant for 15 or more generations. This assumption is quite reasonable as it is very unlikely that the best solution per generation remains constant for so long, given the extremely high variability of the individuals and the broad and intensive searching of the solution space when mixing heterogeneous GA individuals with similar optimum performance indexes.

The target of the GA is to achieve maximum speed at a air drag free height (approximately 45,000 km), as represented in equation (16):

$$z = \frac{v_f}{1 + \left| h_f - h_p \right|} \tag{16}$$

where h_f is the height at the end of the simulation and h_p is the considered air drag free height. The value of this height is arbitrary. However the results do not depend on this value, as the maximization of speed at any height beyond the atmosphere also guarantees the maximization of the speed at any other height, as long as the spacecraft is inside a conservative, gravitational field. The two variables h_f and v_f again depend on the same parameters as in the first problem.

The GA should be able to maximize the target function (16), so the altitude of the final LEO is high whereas the difference between the theoretical circular LEO speed and the actual final launcher speed are kept similar. The target function is complex and involves several numerical simulations per individual in the space of searching.

To summarize, the launcher manoeuvres depend on the following input variables: vertical ascent duration, pitch-push over duration, and pitch-push over final angle. Since the manoeuvres are correlative, these input variables not only affect the shape and duration of the trajectories, but each manoeuvre has a strong influence on the others and on the launching mission as a whole. Since analytical analysis of the whole launching trajectory is not possible, simulation is necessary to calculate the parameters involved in the formulae of the performance indexes as given above (particularly, final speed and height), which therefore depend at the same time on how the input variables are tuned. GAs are non-deterministic search methods that implement the natural evolution of life forms by some genetic operators such as crossovers and mutations that are applied to an initial population of individuals (possible solutions). After each generation, the next offspring are intended to improve a performance index and therefore to be a better solution. More specific information on GAs can be found in [52] and [72], as well as in section 2.3.2.1 of this PhD thesis.

Once the GA generates the offspring, a mutation operator changes individuals' genes randomly. When the population is not centred on a maximum, a low mutation rate is

applied, thus the GA performs the search for the optimum point mostly mixing and combining existing individuals into new ones, whereas raising the mutation rate switches to a broad searching space when a maximum has been reached and no local improvement can be expected. An outer calculation adjusts the mutation rate in response to the observed behaviour, and when most of individuals come up to a specific point of the searching space, the mutation probability automatically increases, avoiding premature convergence to a local extreme. It has been observed (at least in this particular problem) that a quicker and more optimum convergence is achieved when a variable mutation rate is used rather than a fixed one.

4.3.3 Description of the Optimisation Problem

The first stage of the modelled launcher, based on Ariane 5 [87], consists of a cryogenic H158 main stage which hosts a large tank with two compartments, one for 130 tonnes of liquid oxygen and one for 25 tonnes of liquid hydrogen. This stage also hosts a Vulcain engine [105] at the base with thrust of more than one meganewton (1 MN). This stage weighs about 15 tonnes when empty. Attached to the sides are two solid propellant P238 boosters.

The second stage of the launcher is on top of the main stage and below the payload. It consists on an Aestus engine [19] fuelled by high volatile liquids such as monomethylhydrazine and nitrogen tetroxide.

A summary of the two stages technical specifications implemented in the model is presented in Table 3 (section 4.2.3). The flow of mass thought the nozzles should be taken into account to calculate the current mass of the spacecraft during each stage, according to equation ff)(1) in section 2.2.8 of this thesis.

The launch mission consists on several manoeuvres [106], as explained in section 4.2.3. During the vertical launch ascent, the pitch push-over manoeuvre and part of the gravity-turn manoeuvre, the full first stage of the launcher (H158 + 2xP238) provides the required very high thrust. During the gravity-turn manoeuvre, the two P238 boosters are discarded and the manoeuvre continues relying in the H158 engine alone, until all of

its propellant is exhausted. At this point, the H158 engine and the remains of the first stage are discarded. Then the second stage is ignited and a bilinear tangent manoeuvre is performed and the thrust is provided by the Aestus engine alone. This fourth manoeuvre is an optimal guidance law in presence of a uniform gravitational acceleration [106]. The formulation of the manoeuvre [25] was shown in equation (11). The air drag force is provided by equation (6) in section 2.2.10.The effective area may be calculated as the multiplication of the transversal area and a non-dimensional air drag coefficient. This coefficient depends on the limit thermal layer at the launcher surface, and also on the launcher geometry, among other relevant factors. As in section 4.2.3, the effective air drag area may be 25 % of the actual transversal area if the launcher is very aerodynamic. In this case, a 50 m^2 effective area has been considered for the full first stage, 30 m^2 for the H158 stage alone and 10 m^2 for the second stage. One of the most important models is the U.S. Standard Atmosphere 1976 one [100]. Modelling of the gravitational force is based on [20] and section 2.2.1, with the numerical considerations explained in section 4.2.3.

4.3.4 GA Optimisation Results

In Figure 36 the evolution of the performance indexes is shown for all the generations. Each n-th line points out the value of the n-th best solution per generation. Best six solutions per each generation have been highlighted. As in section 4.2, results clearly improved as the GA searches for the final optimum solution, finding the optimum quicker in this case (around 30 generations). This optimum may be used as an initial guess and refined if desired in order to achieve a better solution.

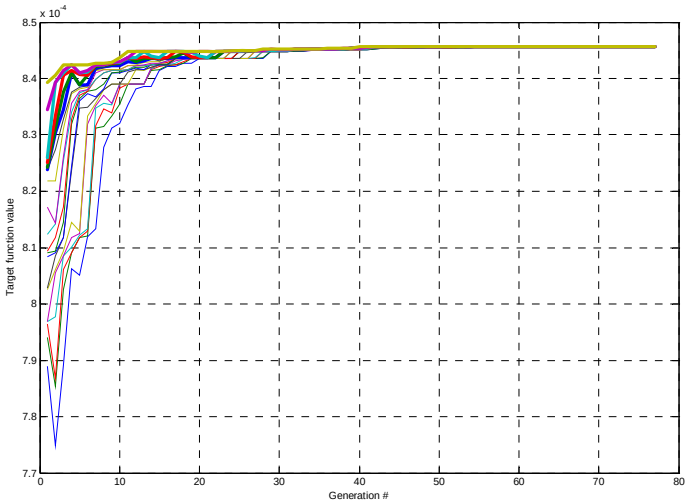

Figure 36: Performance indexes for every individual in the population (GA applied to launch trajectory optimisation, hyperbolical trajectory)

The best value of the GA target function has been obtained for the trajectory parameters shown in Table 5. While in Table 4 the results were more appropriate of a final circular LEO, the escape trajectory implies a vertical trajectory and the tuning parameters are adjusted in that way. A perfectly vertical trajectory beyond the initial seconds is however highly inadvisable, as proven in [4].

Vertical ascent duration	5 seconds
Pitch-push over duration	10.12 seconds
Pitch-push over final angle	1.5533 degrees

Table 5: Values of the parameters for the best GA solution (GA applied to launch trajectory optimisation, hyperbolical trajectory)

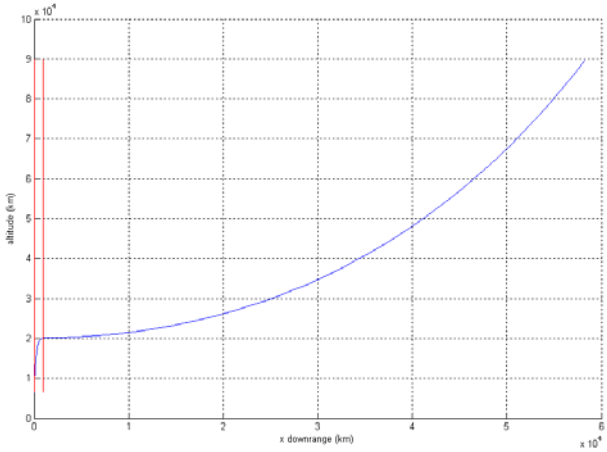

Figure 37: Launcher altitude evolution profile (GA applied to launch trajectory optimisation, hyperbolical trajectory)

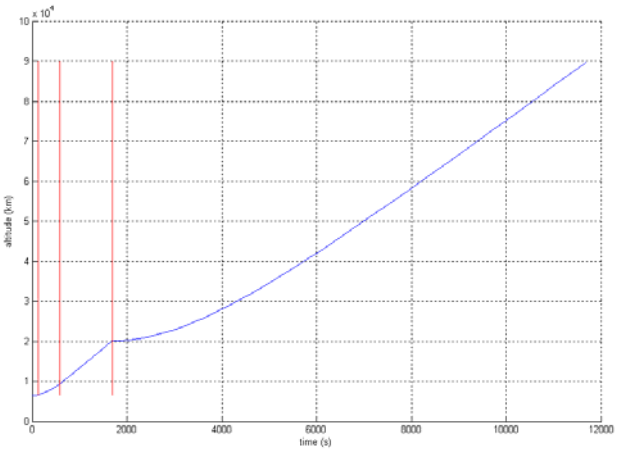

Figure 38: Launcher altitude evolution in time (GA applied to launch trajectory optimisation, hyperbolical trajectory)

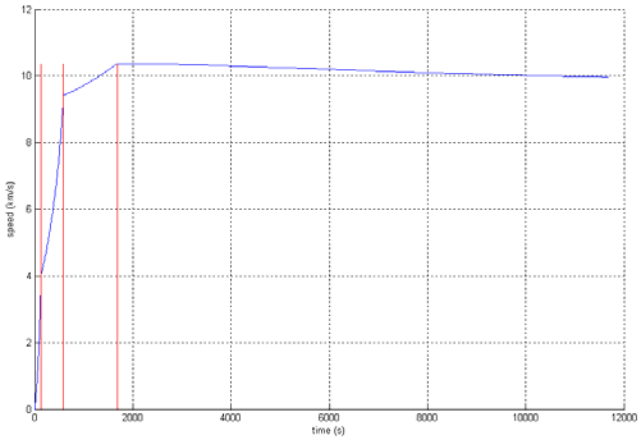

Figure 39: Launcher speed evolution in time (GA applied to launch trajectory optimisation, hyperbolical trajectory)

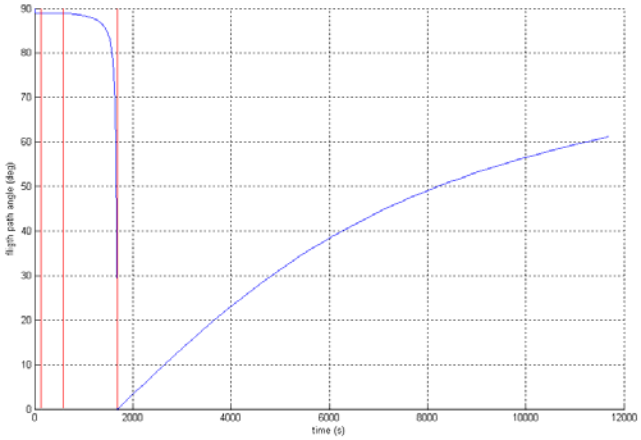

Figure 40: Launcher and satellite flight angle evolution in time (GA applied to launch trajectory optimisation, hyperbolical trajectory)

Figure 37, Figure 38, Figure 39 and Figure 40 show the evolution of the latitude, speed and flight angle of the escape hyperbolical trajectory of the launcher. The altitude increases (Figure 37, Figure 38) without any limit as speed decreases (Figure 39). The speed tends asymptotically to its stable final value. As the distance to Earth increases, the relative direction of the spacecraft respect to the Earth approaches 90 degrees (Figure 40) as it is expected.

4.4 Conclusions

In section 4.1 a very simple model of a launch, following a completely vertical path, has been optimised with the support of a genetic algorithm, proved useful to solve this case. The optimum solution, while interesting, is not enough to leave the Earth gravitational field, as the escape speed is not achieved, in spite of the powerful engines and the quite overestimated specific impulses of the fuel. Thus, this analysis confirms that vertical launching is not a good idea, and that is why normally other strategies are in place when designing launch missions [4][3]. Normally, instead of a purely vertical launch, the ascent trajectories are designed in such a way that the launcher departs from the ground in a vertical direction (or at some angle respect to the local horizon), and then it is successively turned into an horizontal direction of flight with the support of the gravitational force of the Earth (gravity turn).

In sections 4.2 and 4.3 a genetic algorithm has been used to optimise the manoeuvres involved in an ascent trajectory for a typical launcher such as Ariane 5. The results, similar to those given by the launcher vehicle manufacturer, have been proved to be successful and useful, not only for the calculation of optimal solutions, but also as a realistic initial guess for other methods that may refine the solution provided by the GA. The problem of obtaining an acceptable guess of the ascent trajectory is critical when a new launcher is considered or when the desired mission is too different from the ones already studied for that vehicle. Two different cases have been considered: the purpose of the first optimisation is to obtain a final stable circular Low Earth Orbit (LEO) with maximum height, while in the second the aim of the optimisation is to achieve an escape trajectory with maximum speed. As an added value to traditional GAs, a variable mutation rate has been implemented, with satisfactory results. Raising the dynamic mutation rate when the population becomes homogeneous broadens the exploration area in the space of potential solutions, avoiding local maximums, and improving the successful and efficient search for an optimum solution. The results are very promising in both cases and can help in the design of a space mission as an initial guess.

Chapter 5
Optimisation of Interplanetary Trajectories

The aim of this chapter is to provide a description of several evolutive techniques applied to the optimisation of interplanetary trajectories. Such trajectories consist on the departure from Earth and arrival to a planet [7][8][9] or some other destination (such as the Kuiper belt [6]), performing fly-bys in other planets, while the whole delta-V and total time of flight are minimized.

5.1 GA applied to Optimisation of Interplanetary Trajectories with Gravitational Assistance

5.1.1 Introduction

The aim of this section is to analyse the results obtained when using a genetic algorithm (GA) to optimise the interplanetary trajectory of a spacecraft. The desired trajectory should visit a planet, after performing a gravitational assistance or fly-by in planet Mars [7][8][9]. The GA tunes a set of variables, in order to achieve the mission purpose while satisfying the constraints and minimizing the delta-V of the mission. Due to the complexity of the implemented models and the lack of analytical solutions, an alternative non-traditional algorithm provided by soft-computing techniques such as GA is necessary to achieve an optimum solution. The positions of planets as provided by Jet Propulsion Laboratory (JPL) have been considered. A variable mutation rate has been implemented that broadens the search area whenever the population becomes uniform. The results are very useful from the point of view of mission analysis and therefore can be used as an initial guess for further optimizations. They can also be the first step for a more refined analysis and time-consuming simulations based on more complex models of orbital perturbations.

The preliminary astrodynamic mission analysis presented in [4][3][5] have led to a successful understanding of the processes involved when optimising complex systems such astrophysics applications by artificial intelligence techniques. The set of results highlighted the added value of genetic algorithms (GA) for the optimal tuning of parameters in dynamical systems, with the purpose of maximizing the performance and the ascent trajectory of launchers, given a set of constrains. In this section a genetic optimisation of a complex trajectory is presented, aimed to reach Jupiter or Saturn from a Low Earth Orbit (LEO) around the Earth, and performing a fly-by (gravitational assistance) in Mars, while minimizing the fuel budget or total delta-V of the whole mission. Due to the complexity of the implemented models and the lack of analytical

solutions to the equations of the system, the approach provided by the GA was required to perform this optimisation, with satisfactory results.

Two different interplanetary missions are considered and optimised with GA: a mission to Jupiter, and a mission to Saturn. Both analysis are based on the optimisation of the total delta-V of the interplanetary mission. A realistic model of the Solar System has been implemented using the data from the Jet Propulsion Laboratory JPL [96]. A first delta-V is applied when the spacecraft is in a LEO around the Earth, in order to reach planet Mars after a given time of flight. The gravity field of Mars is used to modify the direction and speed of the probe, with the support of an additional delta-V provided by the engines of the spacecraft, which finally reaches planet Jupiter or Saturn after leaving Mars. Once arrived at this final destination, the probe may continue its journey, leaving the Solar System in case of a hyperbolical trajectory; or turning around the Sun in case of an elliptical orbit. A third delta-V may be applied to reduce the speed and to achieve an orbit around Saturn.

The tuning parameters of the GA are the initial epoch for departing from the Earth, the time of flight from Earth to Mars, and the time of flight from Mars to the destination (Jupiter or Saturn). In order to calculate the required delta-V to travel from the Earth to Mars, initial values have to be given to calculate the position of the Earth at departure and the position of Mars at arrival. That is, defining the input data that are required for the solution of the Gauss' problem, also known as Lambert's problem [20][73]. The algorithms that solve the Gauss' problem (see section 2.2.6) provide as outputs the velocities at the tips of the arch of the orbit, thus fully defining the orbit itself and providing the required delta-V at origin, and the final velocity at the destination. In Mars, the calculation of the approaching velocity respect to the planet provides the information required to calculate the velocity at departure after the fly-by, using basic geometrical calculations. The calculation of the trajectory from Mars to Jupiter or Saturn is once again based on the Gauss' problem, given the position of Mars at departure, the position of the destination planet at arrival, and a given time of flight. Obviously, in order to ensure a proper patching of the two different trajectories (from Earth to Mars, and from Mars to the destination planet), the arrival position and time in Mars are equal to those at departure from that planet.

5.1.2 Implementation of the GA for Interplanetary Trajectory Optimisation

Exhaustive search, in which the space of solutions is explored with a fine grid, is unaffordable except for very simple cases. In this application, three variables (t_E, t_{EM} and t_{MS}) in a 10 bits codification define a net of some 1 billion nodes. Thus, using an algorithm based on an intelligent exploration of solutions such as GA is necessary. In this article, the GA explores an initial population of 100 individuals during a maximum of 1000 generations, with a mutation rate that increases as the individuals in the population become homogeneous. Since there are no means to prove the solution obtained by the GA is the best possible optimisation, some criteria is needed to stop the GA; in this case, it can be assumed that a good enough solution is found if no improvement has occurred after 50 generations. This assumption, taken after running many simulations with different number of individuals and generations, is reasonable as it is very unlikely that the best solution per generation remains constant for so long, given the extremely high variability of the individuals and the broad and intensive searching of the solution space when mixing heterogeneous GA individuals with similar optimum performance indexes.

The target of the GA is to achieve an interplanetary trajectory from the Earth to another planet (missions to Jupiter [8][9]and Saturn [7] have been studied) with a fly-by in Mars, while minimizing the fuel budget (delta-V). A function to maximize a performance index *z* has been considered (17):

$$z = 1/v_{\Delta Total} \tag{17}$$

As represented in equation (18), the total delta-V depends on the tuning parameters of the GA (t_E, t_{EM} and t_{MJ}). The exact formula is provided later, in (20).

$$v_{\Delta Total} = v_{\Delta Total}(t_E, t_{EM}, t_{MP}) \tag{18}$$

The performance index in equation (17) is maximized as the total delta-V decreases. Since the roulette method is used to select the parents in the GA, equation (17) encourages the elimination of bad solutions, whereas higher negative powers would force the disappearance of not-so-bad solutions, thus impoverishing the performance of the GA.

GAs are non-deterministic search methods that implement the natural evolution of life forms by some genetic operators such as crossovers and mutations that are applied to an initial population of individuals (possible solutions). After each generation, the next offspring is intended to improve a performance index (17) and therefore to lead to a better solution. More specific information on GAs can be found in [52][72]. In [7][8][9] each input variable (chromosome) is converted into bits, after rescaling according to hh). In that formula, η is the original value of the considered value, η_{res} is the rescaled value, η_{min} and η_{max} are the limits of the valid values, and N is the number of bits (10 in this case) for the binary representation of each variable. After converting η_{res} into bits, twenty zeros should represent the lowest possible value η_{min}, whereas twenty ones should represent the highest possible value η_{max}, thus ensuring that the whole range of bits is used, and increasing accuracy. For a fixed number of bits, a smaller range of binary numbers and a poorer accuracy would be achieved if the rescaling were suppressed. An individual is the concatenation of three chromosomes, i.e., the three input variables: t_E, t_{EM}, t_{MP} (see section 5.1.3), where t_{MP} refers to t_{MJ} or to t_{MS} depending on the considered destination planet. A parents' random selection mechanism (proportional to the performance index) is introduced to bias the population replacement towards increasing the selected fitness function. A one-point crossover is carried out at a randomly chosen bit of the two parents, which are split into two parts each, and then those parts are exchanged to assemble two new individuals. The best individual in the previous generation substitutes the worst individual of the new offspring; elitism prevents the best element of the population from being deleted to make room for a worse new individual.

Once the GA generates the offspring, a mutation operator changes individuals' genes randomly. When the population is not centred on a maximum, a low mutation rate is applied. Then the GA performs the searching for the optimum mostly mixing and combining existing individuals into new ones. On the other hand, raising the mutation rate switches to a broader searching space when a maximum has been reached and no local improvement can be expected. Variable mutation rates have been proved useful in [5]. An outer calculation (19) adjusts the mutation rate in response to the observed behaviour, and when most of the individuals come up to a specific point of the

searching space, the mutation probability automatically increases, avoiding premature convergence to a local maximum.

$$P_m = 0.01\left(1 - \frac{z_{best} - z_{worst}}{z_{best}}\right) \tag{19}$$

where z_{best} and z_{worst} are the best and worst solutions respectively in a generation and P_m is the mutation rate. The GA operators (cross-over and fixed mutation) could have been enough to sample the search space, however in [5] a quicker convergence and a better optimum solution were achieved when using a variable mutation rate.

5.1.3 Description of the Optimisation Problem

The models explained in this section are implemented in Matlab. With the purpose of modelling the departure, arrival and fly-by manoeuvres of the probe at each planet, these planets are modelled as dimensionless points of mass. In fact, the longest planetary radius in the Solar System corresponds to Jupiter (66,850 to 71,500 km), and this is much shorter than any of the major semi-axes of the patched conic orbits involved in the analysed interplanetary mission (for example, the perihelion of the Earth's orbit is around $147 \cdot 10^6$ km). As a result of the comparison of these dimensions, it is possible to conclude that, for a mission analysis optimisation, the dimensions of the planets are irrelevant. The ephemeris of the planets are calculated from tables and formulae provided by [96]. The departure from the Earth at epoch t_E, to reach Mars after a certain time t_{EM}, requires an arch of an orbit around the Sun, that is defined by the heliocentric velocities at departure from Earth, $\boldsymbol{v_{1,EM}}$ and arrival to Mars, $\boldsymbol{v_{2,EM}}$. The required delta-V at the Earth, $\boldsymbol{v_{\Delta E}}$ must achieve $\boldsymbol{v_{1,EM}}$ taking into account the velocity $\boldsymbol{v_{Earth}}$ of planet Earth. In order to calculate $\boldsymbol{v_{1,EM}}$ and $\boldsymbol{v_{2,EM}}$, the Gauss' problem is solved. Knowing the arrival velocity $\boldsymbol{v_{2,EM}}$ to planet Mars, it is possible to calculate the module of the departing speed v_∞ respect to Mars, after performing a fly-by. Solving the Gauss' problem for the trajectory from Mars to Jupiter given a time of flight t_{MJ}, or to Saturn given a time of flight t_{MS}, leads to knowing the velocities at the beginning and at the end of this trajectory, $\boldsymbol{v_{1,MJ}}$ and $\boldsymbol{v_{2,MJ}}$ respectively for the case of the trajectory from Mars to Jupiter, or $\boldsymbol{v_{1,MS}}$ and $\boldsymbol{v_{2,MS}}$ respectively for the case of destination in Saturn. For simplicity, from now on $\boldsymbol{v_{1,MP}}$ shall refer to both $\boldsymbol{v_{1,MJ}}$ and $\boldsymbol{v_{1,MS}}$, and $\boldsymbol{v_{2,MP}}$ shall refer to

$\boldsymbol{v_{2,MJ}}$ and $\boldsymbol{v_{2,MS}}$. Then, once the direction of $\boldsymbol{v_{1,MP}}$ and the departing speed v_∞ respect to Mars are known, it is possible to calculate the actual velocity of the probe after the fly-by respect to the heliocentric reference system, as it is parallel to $\boldsymbol{v_{1,MP}}$ (so the required delta-V after the fly-by is minimized). Then the additional delta-V $\boldsymbol{v_{\Delta M}}$ required to achieve $\boldsymbol{v_{2,MP}}$ may be calculated. A final delta-V $\boldsymbol{v_{\Delta x}}$ ($\boldsymbol{v_{\Delta J}}$ for Jupiter and $\boldsymbol{v_{\Delta S}}$ for Saturn) may be calculated to achieve an orbit around Saturn, if required. This is not necessary if the probe visits Saturn and then goes away. More details on fly-bys design can be found in section 2.2.7.

Values t_E, t_{EM} and t_{MP} (t_{MJ} for Jupiter and t_{MS} for Saturn) are tuning parameters of the GA, which purpose is to minimize the total delta-V $v_{\Delta Total}$ (20) spent in the whole interplanetary mission.

$$v_{\Delta Total} = v_{\Delta E} + v_{\Delta M} + v_{\Delta x} \tag{20}$$

Please also note that, due to the extremely slow variation of all the planetary orbital elements (except obviously the mean longitude), the effect of the variations in the velocity of the planet can be ignored [40][96].

The Keplerian elements for the position of the planets, obtained from [96], contain the Keplerian elements and their rates, with respect to the mean ecliptic and equinox of J2000, valid for the time-interval 1800 AD - 2050 AD, and could be easily extrapolated for a few years without significant loss of accuracy. Those orbital elements are semi-major axis a, eccentricity e, inclination I, mean longitude L of the planet, longitude of perihelion ϖ and longitude of the ascending node Ω (see section 2.2.4). Instructions for using those tables are provided by JPL. First, the Keplerian elements for a given epoch must be computed from the information of the table (values of the elements and their rates). Second, the argument of perihelion ω and the mean anomaly M are calculated. From M it is possible to compute the eccentric anomaly E, following an iterative algorithm to solve the Kepler's equation (21).

$$M = E - e\sin E \tag{21}$$

The iterative algorithm begins with an initial guess of E, E_0 (22), and a quick convergence is achieved following iterations n = 0,1,2,3… as shown in equations (23)(24)(25), until $|\Delta E| < 10^{-6}$ degrees (this tolerance is enough for the present context).

$$E_0 = M + e\sin M \tag{22}$$

$$\Delta M = M - \left(E_n - e\sin E_n\right) \tag{23}$$

$$\Delta E = \frac{\Delta M}{1 - e\cos E_n} \tag{24}$$

$$E_{n+1} = E_n + \Delta E \tag{25}$$

The planet's heliocentric coordinates in its orbital plane ***r'*** (with *x'*-axis aligned from the focus to the perihelion) can be computed from equations (26) and (27). Although not explicitly mentioned in [12], the velocity of the planet can be computed from (28) and (29) [40], where μ is the gravitational parameter of the Sun.

$$x' = a\left(\cos E - e\right) \tag{26}$$

$$y' = a\sqrt{1-e^2}\,\sin E \tag{27}$$

$$\dot{x}' = -\sqrt{\frac{\mu}{a}}\,\frac{\sin E}{1 - e\cos E} \tag{28}$$

$$\dot{y}' = -\sqrt{\frac{\mu\left(1-e^2\right)}{a}}\,\frac{\cos E}{1 - e\cos E} \tag{29}$$

To compute the coordinates ***r*** of the planet and its velocity in the J2000 ecliptic plane (with the *x'*- axis aligned toward the equinox) a rotation of the position [96] and velocity [40] in the orbital plane is necessary, as shown in (30) and (31), where the rotation matrix R_{3x2} from the 2-D coordinates in the orbital plane to those in the J2000 ecliptic plane – modelled in 3-D – is based in three successive rotations (argument of perihelion, inclination and longitude of ascending node), and computed as shown in (32).

$$\bar{r} = R_{3x2}\bar{r}' \tag{30}$$

$$\dot{\bar{r}} = R_{3x2}\dot{\bar{r}}' \tag{31}$$

$$R_{3x2} = R_z(-\Omega)R_x(-I)R_z(-\omega) = \begin{bmatrix} \cos\omega\cos\Omega - \sin\omega\sin\Omega\cos I & -\sin\omega\cos\Omega - \cos\omega\sin\Omega\cos I \\ \cos\omega\sin\Omega + \sin\omega\cos\Omega\cos I & -\sin\omega\sin\Omega + \cos\omega\cos\Omega\cos I \\ \sin\omega\sin I & \cos\omega\sin I \end{bmatrix} \tag{32}$$

Please note that $\bar{r}', \dot{\bar{r}}' \in \Re_{1x2}$, while $\bar{r}, \dot{\bar{r}} \in \Re_{1x3}$. Please also note that, due to the extremely slow variation of all the planetary orbital elements (except obviously the mean longitude), the effect of these variations in the velocity of the planet have been ignored in (31).

The Gauss´ problem may be defined as follows [7] (see section 2.2.6): given two positions $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ in an orbit, given the time of flight t from $\boldsymbol{r_1}$ to $\boldsymbol{r_2}$, and the direction of motion ("short way" or "long way"; we shall always use the "short way" solution in this article), find velocities at $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$, that shall be called $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ respectively. Solving this orbit determination problem leads to the definition of the arch of heliocentric orbit connecting two different planets, providing the required delta-Vs at the planet of departure and the planet of arrival. These delta-Vs can be achieved with a combination of engines and, depending on the conditions, a fly-by. Arches of two different orbits are required to implement the trajectory from Earth to Jupiter or Saturn: a first arch from Earth to Mars, and a second arch from Mars to the destination planet. In the Earth-to-Mars arch, $\boldsymbol{v_1}$ is the velocity necessary to reach Mars from the Earth, given a certain time of flight, and should be achieved by a delta-V provided by the engines of the spacecraft, taking into account that the spacecraft is already moving around the Sun as the Earth is orbiting (thus, part of $\boldsymbol{v_1}$ may be provided by the movement of the Earth, if the direction of the delta-V is wisely chosen). The calculation of the velocity $\boldsymbol{v_2}$ when arriving in Mars is useful to compute the departing velocity after the fly-by. In the Earth-to-Saturn arch, $\boldsymbol{v_1}$ is the velocity at Mars necessary to reach Saturn, given once again a time of flight t_{MJ}. Velocity $\boldsymbol{v_1}$ can be achieved with the fly-by at Mars plus the support of the probe's engines. The velocity at Jupiter $\boldsymbol{v_2}$ is irrelevant if the probe is intended to continue its journey, otherwise velocity $\boldsymbol{v_2}$ must be compensated with a new delta-V (provided this time by the engines of the probe only) if an orbit around Jupiter or Saturn is the final target of the interplanetary mission. The Gauss' problem has not an

analytical solution, thus it must be solved using iterative methods (see section 2.2.6). The *p* iteration method has been implemented and converges in all cases except when $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ are exactly collinear, which is not a probable scenario as numerical methods are in place. In fact, the initial and final position vectors are subjected to some numerical accuracy. In any case, the colinearity scenario is typically associated to a free fall ($\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ in the same direction) or a Hohmann transfer ($\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ in opposite directions), being the latter generally the optimum transfer manoeuvre between two circular orbits, which would be applicable when travelling from one planet to another. But this specific manoeuvre is highly improbable in any of the patched conics of the analysed interplanetary trajectory, as it involves several planets [73].

Given a certain time of flight $t_{EM,}$ and an initial epoch t_E for departure, it is possible to calculate the positions of planets Earth and Mars as explained above, thus the algorithms explained in section 2.2.6 can be used to solve the Gauss' problem for the Earth-to-Mars trajectory (first arch of the whole interplanetary mission from Earth to Saturn), leading to the computation of the departing velocity from Earth $\boldsymbol{v_{1,EM}}$, and the arriving velocity in Mars $\boldsymbol{v_{2,EM}}$, both respect to the heliocentric reference system. Considering that the probe departs from a given LEO around the Earth, the initial speed of the probe respect to the Earth is negligible, in comparison with the velocity $\boldsymbol{v_{Earth}}$ of the Earth around the Sun, and from the point of view of this Mission Analysis interplanetary study we may assume that the probe moves respect to the Sun at the same speed than the Earth. Thus, equation (33), based on equation bb), may be used to calculate the required delta-V $\boldsymbol{v_{\Delta E}}$ to achieve $\boldsymbol{v_{1,EM}}$ and to leave the Earth.

$$\overline{v}_{\Delta E} = \overline{v}_{1,EM} - \overline{v}_{Earth} \tag{33}$$

The fly-by trajectory (see section 2.2.7) consists on using the gravity field of a planet to apply a delta-V to a probe, which may accelerate considerably, depending on the approaching and leaving directions respect to the planet (the kinetic energy gained by the probe comes from the negligible reduction of the planet's speed). Considering the size of the sphere of influence of Mars respect to the size of its orbit, and comparing the interplanetary times of flight with the duration of the fly-by, Mars may be considered as a dimensionless point of mass at which the spacecraft changes its velocity instantly due

to the fly-by. The relations between $\boldsymbol{v_a}$, $\boldsymbol{v_b}$, $\boldsymbol{v_{\infty,a}}$, $\boldsymbol{v_{\infty,b}}$ (see Figure 4) and the velocity of the planet Mars $\boldsymbol{v_{Mars}}$ are shown in (34) and (35), respectively based on bb) and cc).

$$\overline{v}_{\infty,a} = \overline{v}_a - \overline{v}_{Mars} \tag{34}$$

$$\overline{v}_{\infty,b} = \overline{v}_b - \overline{v}_{Mars} \tag{35}$$

After solving the Gauss' problem for the first arch of the trajectory, the approaching heliocentric velocity $\boldsymbol{v_a}$ of the probe to Mars is known (36), therefore it is possible to calculate $\boldsymbol{v_{\infty,a}}$ (34). The value of v_∞ is trivial (37).

$$\overline{v}_a = \overline{v}_{2,EM} \tag{36}$$

$$v_\infty = \sqrt{\overline{v}_{\infty,a} \cdot \overline{v}_{\infty,a}} \tag{37}$$

Given a certain time of flight t_{MS} for the Mars-to-Saturn trajectory, it is possible to solve the Gauss' problem for that trajectory, as the positions of Mars and Saturn are known from [96]. Being $\boldsymbol{v_{1,MP}}$ the desired velocity when leaving Mars and having a particular direction, then the heliocentric velocity $\boldsymbol{v_b}$ after fly-by should have the same direction, in order to reduce the required delta-V to achieve $\boldsymbol{v_{1,MP}}$. Equation (38) shows how to compute the cosine of angle α between $\boldsymbol{v_{1,MP}}$ and $\boldsymbol{v_{Mars}}$, from which v_b can be calculated using the theorem of the cosine (39). The required delta-V $\boldsymbol{v_{\Delta M}}$ for leaving Mars at the proper velocity $\boldsymbol{v_{1,MP}}$ may be calculated from (40). Whenever it is not possible to give form to a triangle from vectors $\boldsymbol{v_{\infty,b}}$, $\boldsymbol{v_{Mars}}$ and $\boldsymbol{v_b}$ (due to a high value of α respect to a low value of v_∞), the GA algorithm sets the delta-V to a very high number and the failed interplanetary mission is discarded by the GA when generating the next offspring (see criteria function, equation (17)).

$$\cos\alpha = \frac{\overline{v}_{Mars} \cdot \overline{v}_{1,MS}}{v_{Mars} v_{1,MS}} \tag{38}$$

$$v_b^2 + v_{Mars}^2 - 2 v_b v_{Mars} \cos\alpha = v_\infty^2 \tag{39}$$

$$\overline{v}_{1,MJ} = \overline{v}_b + \overline{v}_{\Delta M} \tag{40}$$

Once the probe arrives in the sphere of influence of Saturn, a last delta-V is required in order to achieve an orbit around the planet. The speed of the spacecraft should be

reduced down to nearly the speed of Saturn around the Sun. Indeed, the relative velocity of the probe in orbit around Saturn respect to Saturn itself, would be negligible in comparison to the orders of magnitude of Saturn orbital speed and the speed of the probe when approaching Saturn. Therefore, the required delta-V $\boldsymbol{v_{\Delta x}}$ may be calculated as the difference between the velocity $\boldsymbol{v_{2,MP}}$ of the spacecraft at the end of the second arch of the interplanetary trajectory, and the velocity $\boldsymbol{v_{Jupiter}}$ or $\boldsymbol{v_{Saturn}}$ of planet Jupiter or Saturn (both called $\boldsymbol{v_p}$) around the Sun (41).

$$\bar{v}_{\Delta S} = \bar{v}_{2,Mx} - \bar{v}_P \tag{41}$$

5.1.4 GA Optimisation Results for the Trajectory to Jupiter

The GA has been run in order to optimise the trajectory for Earth to Jupiter, with a fly-by in Mars, and orbiting around Jupiter after the end of the mission. The tuning parameters and the allowed values are described in Table 6. The evolution of the best solution found by the GA at each generation is shown in Figure 41. The final optimum trajectory is shown in Figure 42, where the orbits of the planets Earth, Mars and Jupiter have also been represented for the given range of times. The positions of Earth, Mars and Jupiter are shown at departure, fly-by and arrival, respectively. The departing date for this optimum solution is 93,255 days after J2000 [14], the total delta-V is 9.32 km/s (2.82 km/s at departure from the Earth, 2.15 km/s after the fly-by in Mars, and 4.34 km/s at arrival in Jupiter), and the mission duration have been 3,836 days. When arriving in Jupiter, an alternative to orbiting around this planet is to continue the journey without applying the last delta-V, in which case the fuel budget shall be lower and the total delta-V shall be 4.97 km/s.

	Minimum value	**Maximum value**
Date of departure, t_E	J2000	100,000 days after J2000
Earth-to-Mars time of flight, t_{EM}	100 days	1000 days
Mars-to-Jupiter time of flight, t_{MJ}	500 days	20,000 days

Table 6: Range of values for the tuning parameters (GA applied to Earth-Jupiter trajectory optimisation)

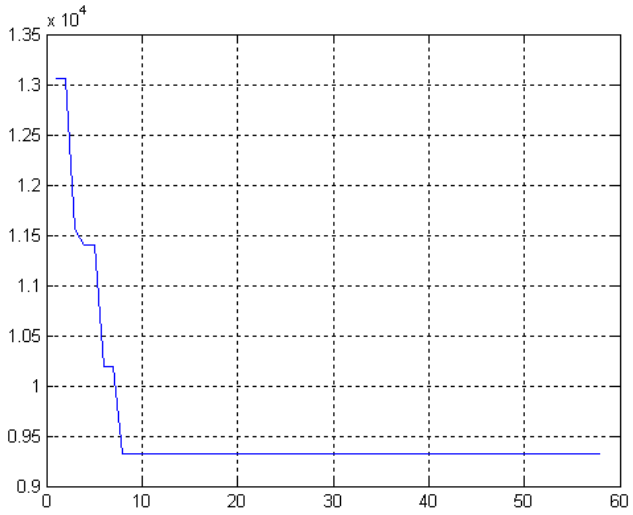

Figure 41: Evolution of the best delta-V during the execution (GA applied to Earth-Jupiter trajectory optimisation)

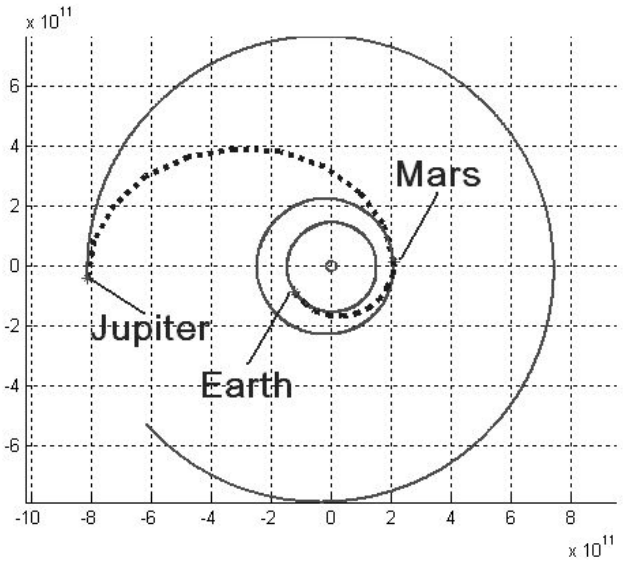

Figure 42: Trajectory from Earth to Jupiter with fly-by in Mars (GA applied to Earth-Jupiter trajectory optimisation)

The obtained results may be compared to a Hohmann manoeuvre. The Hohmann transfer [40] is a two-impulses transfer between two different coplanar circular orbits, and it is an optimum impulsive transfer manoeuvre if the radius of the final orbit is less than 15.6 times the radius of the initial orbit (Jupiter fulfils this condition). The two impulses are performed at opposites ends of the transfer orbit, and their values may be

calculated from (45) and (46), where the speeds of the planets of departure and arrival are already taken into account.

$$\Delta v_1 = \sqrt{\frac{\mu}{r_1}}\left[\sqrt{\frac{2r_2}{r_1+r_2}}-1\right] \tag{42}$$

$$\Delta v_2 = \sqrt{\frac{\mu}{r_2}}\left[1-\sqrt{\frac{2r_1}{r_1+r_2}}\right] \tag{43}$$

Although the orbits of the Earth and Jupiter have different inclinations and are not perfectly circular, the calculation of the total delta-V of a Hohmann transfer provides a quite realistic idea of the delta-V required for such an interplanetary mission, in case an optimum direct transfer is performed without fly-by in Mars. Considering the best possible case (alignment between Sun position, Earth position at departure and Jupiter position at arrival; and also, Earth orbital radius as if in perihelion, and Jupiter orbital radius as if in aphelion), the total delta-V for a Hohmann transfer would be 14.08 km/s, and the duration of this direct transfer would be 941 days. If the last delta-V to achieve an orbit around Jupiter is skipped, then the only thrust of the trajectory shall be at the Earth, and the delta-V shall be 8.51 km/s. Thus, in both cases the fly-by in Mars is useful to significantly reduce the fuel budget, however the mission duration (which was not a parameter to optimise by the GA) has significantly increased. It is also important to note that each of the delta-Vs at departure and arrival are lower in case of a fly-by in Mars, than with the Hohmann transfer, while the delta-V after the fly-by is also relatively small; this means that smaller engines will be required for the probe, which is an additional and important advantage respect to applying a Hohmann transfer. The results are summarized in Table 7.

	Δv at the Earth	Δv after fly-by in Mars	Δv after arrival Jupiter	**Total Δv**	**Mission time**
Hohmann transfer	8.51 km/s	-	5.57 km/s	**14.08 km/s**	**941 days**
GA	2.82 km/s	2.15 km/s	4.34 km/s	**9.32 km/s**	**3,836 days**

Table 7: Range of values for the tuning parameters (GA applied to Earth-Jupiter trajectory optimisation)

5.1.5 GA Optimisation Results for the Trajectory to Saturn

The GA has been applied in order to optimise the trajectory from Earth to Saturn, with a fly-by in Mars, and orbiting around Saturn after the end of the mission. The tuning parameters and the range of possible values are described in Table 8.

	Minimum value	**Maximum value**
Date of departure, t_E	10,000 days before J2000	8,000 days after J2000
Earth-to-Mars time of flight, t_{EM}	100 days	500 days
Mars-to-Saturn time of flight, t_{MS}	500 days	20,000 days

Table 8: Range of values for the tuning parameters (GA applied to Earth-Saturn trajectory optimisation)

The evolution of the best solution found by the GA at each generation is shown in Figure 43.

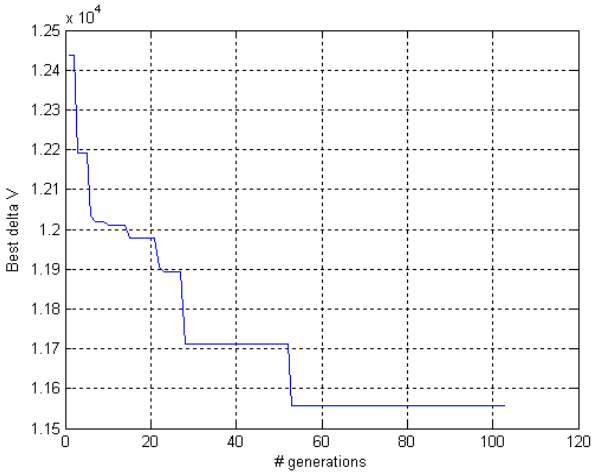

Figure 43: Evolution of the best delta-V during the execution (GA applied to Earth-Saturn trajectory optimisation)

The final optimum trajectory is described in Figure 44, where the orbits of the planets Earth, Mars, Jupiter and Saturn have also been represented for the given range of times.

The positions of Earth, Mars and Saturn are shown at departure, fly-by and arrival, respectively. The departing date for this optimum solution is 8,622 days before J2000 [20], the total delta-V is 11.57 km/s (4.61 km/s at departure from the Earth, 2.02 km/s after the fly-by in Mars, and 4.93 km/s at arrival to Saturn). The mission duration was 972 days. When arriving to Saturn, an alternative to orbiting around this planet is to continue the journey without applying the last delta-V, in which case the fuel budget will be lower and the total delta-V will be 6.63 km/s.

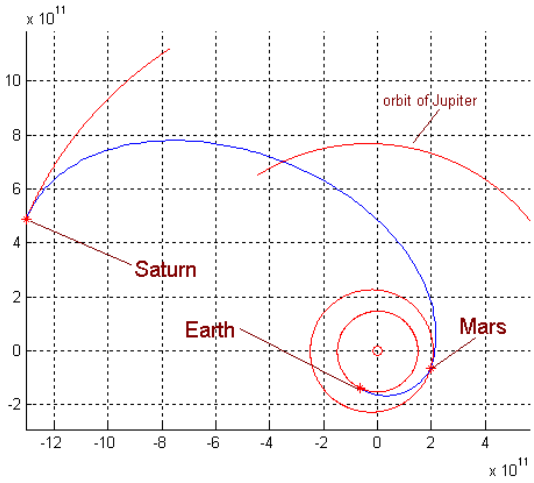

Figure 44: Trajectory from Earth to Saturn with fly-by in Mars (GA applied to Earth-Saturn trajectory optimisation)

The obtained results may be compared to a Hohmann manoeuvre. The Hohmann transfer [40] is a two-impulses transfer between two different coplanar circular orbits, and it is an optimum impulsive transfer manoeuvre if the radius of the final orbit is less than 15.6 times the radius of the initial orbit (Saturn fulfils this condition). Although the orbits of the Earth and Saturn have different inclinations and are not perfectly circular, the calculation of the total delta-V of a Hohmann transfer provides a quite realistic idea of the delta-V required for such an interplanetary mission, in case an optimum direct transfer is performed without fly-by in Mars. Considering the best possible case (alignment between Sun position, Earth position at departure and Saturn position at arrival; and also, Earth orbital radius as if in perihelion, and Saturn orbital radius as if in aphelion), the total delta-V for a Hohmann transfer would be 15.52 km/s, and the duration of this direct transfer would be 2,059 days. If the last delta-V to achieve an orbit around Saturn is skipped, then the only thrust of the trajectory will be at the Earth, and the delta-V will be 10.07 km/s. Thus, in both cases the fly-by in Mars is useful to

significantly reduce the fuel budget and the mission duration. It is also important to note that each of the delta-Vs at departure and arrival are lower in case of a fly-by in Mars than with the Hohmann transfer, while the delta-V after the fly-by is also relatively small. This means that smaller engines will be required for the probe, which is an additional and important advantage respect to applying a Hohmann transfer. The results are summarized in Table 9.

	Δv at the Earth	Δv after fly-by in Mars	Δv after arrival to Saturn	**Total Δv**	**Mission time**
Hohmann transfer	10.07 km/s	-	5.45 km/s	**15.94 km/s**	**2,059 days**
GA	4.61 km/s	2.02 km/s	4.93 km/s	**11.57 km/s**	**2,390 days**

Table 9: Range of values for the tuning parameters (GA applied to Earth-Saturn trajectory optimisation)

5.2 Swarm Algorithms applied to Optimisation of a Trajectory from Earth to Jupiter (EMJ)

5.2.1 Introduction

A very interesting aerospace optimisation problem nowadays is the design of interplanetary trajectories in the Solar System, using gravitational fly-bys in order to reduce fuel consumption and/or mission duration [11]. This is a complex optimisation problem due to the complexity of the models and the lack of analytical solutions, as well as to the presence of strong discontinuities. An exhaustive search of the space of input variables is unaffordable even with modern, state-of-the-art computing technology. Thus, a feasible approach requires artificial intelligence and modern optimisation techniques based on the intelligent selection of some potential solutions. These individuals evolve in order to generate better solutions until some optimisation criteria are met. In this section, two evolutive algorithms are considered, both based on particle swarm optimisation. In both cases, the trajectory to optimise departs from Earth and, after a fly-by in Mars, arrives in Jupiter. In the single-criteria case, only the fuel

consumption is considered as a variable to minimize, while in the multi-criteria case the fuel consumption and the total time of the mission are simultaneously are taken into account, building a Pareto front of non-dominated solutions that provides an interesting view of the possible options for the space mission. In both single- and multi-criteria cases, the swarm algorithms optimize some tuning parameters of the trajectory: launch epoch, and times of flight between planets. Results are compared to other works on the same problem. They demonstrate the benefit of applying these evolutive techniques to decrease both mission duration and propellant cost when using intermediate gravity assist bodies.

Preliminary works [3][5][10][89] have shown the efficiency of intelligent computational techniques when dealing with complex systems such as astrodynamic applications. The results proved the value of genetic algorithms (GA) for the optimisation of ascent trajectories, given a set of constrains. These evolutive strategies have been also applied to interplanetary trajectories [6][7][8]. This section is based on [11] and is an extension of [8] and section 5.1 of this thesis, where GAs were used. In this section, two particle swarm algorithms are applied to the optimisation of a complex interplanetary trajectory of a spacecraft, aimed to reach Jupiter from a LEO (Low Earth Orbit) around the Earth, and performing a gravitational assistance in Mars, while minimizing the fuel budget and duration of the mission. The complexity of this dynamic system and the lack of analytical solutions of the model suggest the utilization of evolutive strategies to deal with it. In one case PSO (Particle Swarm Optimization) is applied to optimise the fuel consumption, whereas in the other case a Pareto-front is generated with a MOPSO (Multi Objective Particle Swarm Optimisation) that optimises both propellant and total mission duration simultaneously.

5.2.2 Implementation of PSO algorithm for Interplanetary Trajectory Optimisation

A major difficulty when optimising parameters in an interplanetary mission is to provide algorithms that converge towards an optimum. Exhaustive search is unaffordable except for very simple cases. In general, using an algorithm based on an intelligent exploration of solutions such as GA [52] or Particle Swarm Optimisation is

necessary. PSO is a population-based, swarm intelligent, search algorithm based on the simulation of the social behaviour of birds within a flock or any other swarm organism [69], using physical movements of the individuals within the swarm. Many different techniques can be applied to implement PSO, so the evolutionary mechanism of the swarm algorithm adapts to the global and local exploration of the search space area. Whereas other evolutionary algorithms are based on parent representation and selection of individuals, PSO and Multi-Objective PSO (MOPSO) rely on leaders to guide the whole population. PSO refers to single-output analysis, while MOPSO refers to PSO when more than one output is available [13]. One of the first works on swarm algorithms is reported in [69]. A survey of the state-of-the-art of particle swarm optimization is summarized in [92]. A comparison of multiobjective evolutionary algorithms is presented in [108], while in [71] some suggestions are explained for improving both spread and convergence in swarm algorithms and Pareto-fronts. Data structure and turbulence concepts are applied to swarm algorithms in [46].

In the PSO described in this section, we consider a population of 400 individuals and up to 1000 updates of the population. As in previous works [5][8], a condition has been introduced in the algorithm in order to stop the execution automatically after 50 iterations without further improvement. This assumption, taken after running many simulations with different number of individuals and generations, is reasonable as it is very unlikely that the best solution remains constant for so long. In this case, only 75 updates of the population were necessary, meaning that convergence was achieved after only 25 iterations. A fitness function that maximizes a performance index z has been considered (44), in order to minimize the total delta-V, Δv_{Total},

$$z = 1/\Delta v_{Total} \tag{44}$$

Each individual is defined by its position in the space of input variables (date of departure, t_E, the Earth-to-Mars time of flight, t_{EM}, and the Mars-to-Jupiter time of flight, t_{MJ}). The updated position of every *i-th* individual after the *k-th* step can be described with the general formulae (45) and (46), based on hh)(1) and hh)(2):

$$\bar{v}_{i,k} = \bar{v}_{i,k-1} + \sum_{j=leaders} c_{i,j}\left(\bar{x}_{i,j} - \bar{x}_{i,k-1}\right) + c_0 \tag{45}$$

$$\bar{x}_{i,k} = \bar{x}_{i,k-1} + \bar{v}_{i,k} \tag{46}$$

where:

$\bar{v}_{i,k}$ is the velocity vector of the *i-th* individual in transition from step *k-1* to step *k*.

$c_{i,j}$ are random numbers

c_0 is a random number (sometimes called "turbulence factor") that introduces a random component into the updated final position of each individual, similar to mutation in GAs

$\bar{x}_{i,j}$ is the *j-th* leader for the *i-th* individual

$\bar{x}_{i,k}$ is the position (decision vector) of the *i-th* individual in *k-th* step.

As shown in (45) and (46), a leader is defined for each individual in the population, and the position of each individual is changed by weighting the current distance to each leader with some random numbers, and combining these weighted distances into a single vector that represents the change in the position of the individual in the space of input variables.

Whereas formulae (45) and (46) are quite intuitive, it is not obvious to define the leaders for each individual, and there are many different criteria in scientific articles, using local leaders, past history of each particle, etc. Nearly all PSO approaches use as one of the leaders the individual with the best global results in the population. There are other interesting considerations when designing a PSO algorithm, such as distribution of random weights, dynamic assignment of leaders, turbulence factors, etc. A very simple approached has been used in this section for PSO. No turbulence has been considered, as the complexity of the models and the discontinuities of the space of output variables jeopardize the success of any individual based in purely random decisions. Turbulence, however, is not fully discarded for future works, as random decision components such as variable mutation rates [8] may lead to interesting and satisfactory results.

Two leaders have been considered in equation (45): the best global solution; and the best historical solution for each individual (sometimes known as "personal" solution). These leaders are weighted according to random numbers, taken from a uniform distribution between 0 and 1. This approach yields to computing the change in position for each individual according to (47):

$$\bar{v}_{i,k} = \bar{v}_{i,k-1} + c_1\left(\bar{x}_{i,leader} - \bar{x}_{i,k-1}\right) + c_2\left(\bar{x}_{i,pbest} - \bar{x}_{i,k-1}\right) \quad (47)$$

where:

c_1 and c_2 are random numbers, selected from an uniform distribution between 0 and 1.

$\bar{x}_{i,leader}$ is the nearest Pareto-front solution respect to the *i-th* individual.

$\bar{x}_{i,pbest}$ is the best historical "personal" solution of the *i-th* individual

Whereas this simple approach leads to good results, there has been already some research in order to further improve the PSO algorithm, in order to tailor the design of the swarm algorithm to the characteristics of the analysed problem and eventually leading to some general approach for interplanetary trajectory optimisation that up to date is missing in literature to the best of our knowledge.

5.2.3 Implementation of MOPSO algorithm for Interplanetary Trajectory Optimisation

Multiobjective optimisation problems are quite common in every area of knowledge [47][108]. In such problems, the objectives to be optimised are normally in conflict respect to each other, hence there is not a single solution for these problems. Traditionally, multiobjective optimisation problems are solved with multi-input, single-output target functions, combining all the objectives into just one formula to be optimised. This approach has been proved to be useful in many problems, but it is not always easy to find the best fitness function, being a more interesting approach to obtain a global view of all the different dominant solutions, presented as Pareto-fronts. Such global view may be particularly necessary when analysing the influence of the variations of some inputs in all the outputs. Whereas this analysis can be made for single-output functions (applying gradient methods, for instance), some difficulties may arise when there are many variables, local solutions or discontinuities. Swarm algorithms avoid these drawbacks and support the multi-output, Pareto-front concept better than other AI algorithms [28][29]. The function to optimise in a general multiobjective minimization problem may be written as in equation (48):

$$\bar{y} = f(\bar{x}) = [f_1(\bar{x}), f_2(\bar{x}), ..., f_k(\bar{x})] \quad (48)$$

where $\bar{x} = [x_1, x_2, ..., x_n]$ is the vector of decision variables and $\bar{y} = [y_1, y_2, ..., y_n]$ is the vector of outputs.

If (48) cannot be analytically evaluated (like in this article), it is mandatory to use some numerical calculations and iterative processes in order to find the optimum solution. Given two different potential solutions, represented as vectors $\bar{y}_A$ and $\bar{y}_B$, the first one would be dominant respect to the second one if (49) is verified. If the inequality is strictly fulfilled, then $\bar{y}_B$ is called a dominated solution.

$$y_{A,i} \leq y_{B,i} \qquad \forall i \tag{49}$$

Note that minimization problems are considered here, without loss of generalization, as (48) and (49) can be easily adapted to maximization problems. Dominated solutions are not interesting in optimisation problems, as the interpretation of (49) is that solution $\bar{y}_A$ is better than $\bar{y}_B$. Therefore, $\bar{y}_B$ can be discarded. However, it may happen that $\bar{y}_A$ surpassed $\bar{y}_B$ in only some (but not all) outputs, and vice versa; in that case, $\bar{y}_A$ and $\bar{y}_B$are both non-dominated solutions to each other, and both may offer interesting properties and results.

The aim of the MOPSO is to find a set of non-dominated solutions (Pareto-front), which have to be saved in some data structure during the execution of the algorithm. The size of this structure may grow in size very quickly, as more non-dominated solutions are found in every step. After each step, the new position of the individuals in the population must be compared to every solution in the Pareto-front, to perform some substitution if necessary. This increases the complexity and the computational time in every step, as the Pareto-front becomes bigger. In order to ensure that solutions are varied and heterogeneous, some mechanism must be used so the distribution of the decision vectors in the Pareto-front is smooth and uniform. Dealing with these three issues (size of the Pareto front, comparison of Pareto front to individuals in the population, and mechanism to ensure a smooth distribution) lead to establish a limit of the number of solutions in the Pareto-front, which in our case matches the size of the population. Individuals with new positions in the space of input variables become candidates to join the Pareto-front, if they are non-dominated respect to any other Pareto-front solution. Before they finally join the Pareto-front (discarding some other

solution already in the Pareto-front), a comparison is performed between the candidate to join and the candidate to be dismissed, in terms of distances to other solutions in the Pareto-front. The substitution is performed by considering only the highest distance, so the density in the Pareto-front is kept as low as possible. In spite of this technique, that ensures some variety of the solutions, sometimes the Pareto-front is not as broad as expected, due to isolated solutions and strong discontinuities in global, single-output optima.

In this work, the aim is the intelligent optimisation of interplanetary trajectories from the Earth to Jupiter, using a swarm-based Pareto-front technique, and applying fly-bys in Mars. The tuning parameters of the swarm algorithm are the date of departure, t_E, the Earth-to-Mars time of flight, t_{EM}, and the Mars-to-Jupiter time of flight, t_{MJ}. The MOPSO algorithm provides a useful Pareto-front of non-dominated solutions, from which the user may decide which solution or set of solutions are the most appropriate to design the Earth-to-Jupiter mission. That decision is based on the obtained delta-V and total time of flight of the mission. In fact, the consideration of non-dominated solutions is useful for analyzing the trade-off between these two output variables. For example, some optimum solution may require the lowest delta-V but at the expense of a higher total time of flight, while some other solution may reduce the time of flight but increasing the delta-V. Indeed, a Pareto-front is an excellent asset for this trade-off analysis.

Two leading vectors are considered for each individual: the best global solution and the best historical solution achieved by each individual (sometimes known as the best "personal" solution). The best personal solution is particular for each individual and matches the last achieved solution that is non-dominated by any other historical decision vector of that individual. The best global leader is not clearly defined in multiobjective problems as more than one output is computed. In this algorithm, given any individual, the nearest non-dominated solution (i.e., the nearest solution of the Pareto-front) will be considered in calculations. For each individual in the population, both leaders (which differ from one individual to another) are combined to define a certain "velocity" in the space of input variables (i.e. a change in the decision vector), according to (50). In the following iteration, the position of the individual in the space of input variables is updated as in (51).

$$\bar{v}_{i,k} = \bar{v}_{i,k-1} + c_1\left(\bar{x}_{i,leader} - \bar{x}_{i,k-1}\right) + c_2\left(\bar{x}_{i,pbest} - \bar{x}_{i,k-1}\right) \quad (50)$$

$$\bar{x}_{i,k} = \bar{x}_{i,k-1} + \bar{v}_{i,k} \quad (51)$$

where:

$\bar{v}_{i,k}$ is the velocity vector of the *i-th* individual in transition from step *k-1* to step *k*.

c_1 and c_2 are random numbers, selected from an uniform distribution between 0 and 1.

$\bar{x}_{i,leader}$ is the nearest Pareto-front solution respect to the *i-th* individual.

$\bar{x}_{i,pbest}$ is the latest, historically non-dominated solution of the *i-th* individual.

$\bar{x}_{i,k}$ is the position (decision vector) of the *i-th* individual in *k-th* step.

The MOPSO algorithm in this work considers 1000 individuals (also known as particles) and 1000 steps, leading to quite interesting results. Whereas this simple approach has been successful, there is some on-going research in order to tailor the parameters of the MOPSO algorithm to the optimisation problem of interplanetary trajectories, and to improve the results provided in the Pareto-front.

5.2.4 Description of the Optimisation Problem

The model of the whole system has been implemented in Matlab. It includes the Solar System and the spacecraft. The main parameters of the model, such as the ephemeris of the planets, the Keplerian elements and their rates, etc., are calculated from tables and formulae provided by [96]. With the purpose of modelling the departure, arrival and fly-by manoeuvres of the probe at each planet, these planets are modelled as dimensionless points of mass. In fact, the longest planetary radius in the Solar System corresponds to Jupiter (66,850 to 71,500 km), and this is much shorter than any of the major semi-axes of the patched conic orbits involved in the analysed interplanetary mission (for example, the perihelion of the Earth's orbit is around 147×10^6 km). As a result of the comparison of these dimensions, it is possible to conclude that, for a mission analysis optimisation, the dimensions of the planets are irrelevant.

The departure from the Earth at epoch t_E, to reach Mars after a certain time t_{EM}, requires an arc of an orbit around the Sun, that is defined by the heliocentric velocities at

departure from Earth $\boldsymbol{v_{1,EM}}$ and arrival to Mars $\boldsymbol{v_{2,EM}}$. The required thrust at the Earth $\boldsymbol{\Delta v_E}$ must achieve $\boldsymbol{v_{1,EM}}$ taking into account the velocity $\boldsymbol{v_{Earth}}$ of planet Earth. In order to calculate $\boldsymbol{v_{1,EM}}$ and $\boldsymbol{v_{2,EM}}$, the Gauss' problem has to be solved [20] (see section 2.2.6). In this case we have applied the *p*-iteration algorithm. Considering that the probe departs from a given LEO around the Earth, the initial speed of the probe respect to the Earth is negligible, in comparison to the velocity $\boldsymbol{v_{Earth}}$ of the Earth around the Sun. We may assume that the probe moves respect to the Sun at the same speed than the Earth. Thus, equation bb) has been applied and solved for Mars, yielding to equation (52), which can be used to calculate the required delta-V $\boldsymbol{\Delta v_E}$ to achieve $\boldsymbol{v_{1,EM}}$ and to leave the Earth.

$$\Delta\overline{v}_E = \overline{v}_{1,EM} - \overline{v}_{Earth} \tag{52}$$

The gravitational assistance manoeuvre consists on using the gravity field of a planet – Mars, in this case - to accelerate the probe (see section 2.2.7). The final value and direction of the velocity depends on the approaching and leaving directions respect to the planet. Considering the size of the trajectories from Earth to Mars and from Mars to Jupiter, planet Mars may be considered in our models as a dimensionless point of mass at which the spacecraft changes its velocity instantly due to this gravitational assistance. Knowing the arrival velocity $\boldsymbol{v_{2,EM}}$ to planet Mars, it is possible to calculate the absolute value of the departing speed respect to Mars, after performing a gravitational assistance. Therefore the required thrust $\boldsymbol{\Delta v_M}$ for leaving Mars at the proper velocity $\boldsymbol{v_{1,MJ}}$ can be obtained from (53), based on equation cc):

$$\overline{v}_{1,MJ} = \overline{v}_b + \Delta\overline{v}_M \tag{53}$$

where $\boldsymbol{v_b}$ is the velocity of the probe respect to the heliocentric reference system.

Solving the Gauss' problem for the trajectory from Mars to Jupiter (given a time of flight t_{MJ}) leads to the calculation of velocities at the beginning and at the end of this trajectory, $\boldsymbol{v_{1,MJ}}$ and $\boldsymbol{v_{2,MJ}}$ respectively. A final delta-V $\boldsymbol{\Delta v_J}$ (54) is necessary if an orbit around Jupiter is required. This is not necessary if the probe visits Jupiter and then goes away.

$$\Delta\overline{v}_J = \overline{v}_{2,MJ} - \overline{v}_{Jupiter} \tag{54}$$

The tuning parameters for the swarm algorithms are the date of departure t_E, the Earth-to-Mars time of flight t_{EM}, and the Mars-to-Jupiter time of flight t_{MJ}. The purpose of the PSO is to minimize the total delta-V Δv_{Total} (55) spent in the whole interplanetary mission.

$$\Delta v_{Total} = \Delta v_E + \Delta v_M + \Delta v_J \tag{55}$$

In the case of the MOPSO, both the total fuel budget Δv_{Total} (55) and the total time of flight of the mission, t_{Total} (56), are considered when building a Pareto-front of possible solutions.

$$t_{Total} = t_{EM} + t_{MJ} \tag{56}$$

Also note that, due to the extremely slow variation of all the planetary orbital elements (except obviously the mean longitude), the effect of the variations in the velocity of the planet can be ignored [40][96]. For a complete description of the problem, see [8] and section 5.1 of this PhD thesis.

5.2.5 PSO and MOPSO Results for Interplanetary Trajectory

The final optimum trajectory obtained by the GA in section 5.1.4 [8] is shown in Figure 42 (dashed line), where the orbits of the planets Earth, Mars and Jupiter have also been represented for the possible range of times (solid line). The positions of Earth, Mars and Jupiter are shown at departure, gravitational assistance and arrival, respectively. The total delta-V is 9.32 km/s. When arriving to Jupiter, an alternative to orbiting around this planet is to continue the journey without applying the last thrust, in which case the total delta-V would be 4.97 km/s.

The final optimum trajectory obtained by the PSO is shown in Figure 45 (dashed line), where the orbits of the planets Earth, Mars and Jupiter have also been represented for the possible range of times (solid line). Again, the positions of Earth, Mars and Jupiter are shown at departure, gravitational assistance and arrival, respectively. The total delta-V is in this case 9.43 km/s. When arriving to Jupiter, if the probe goes away, the last thrust is not necessary and then the total delta-V is reduced to 5.01 km/s.

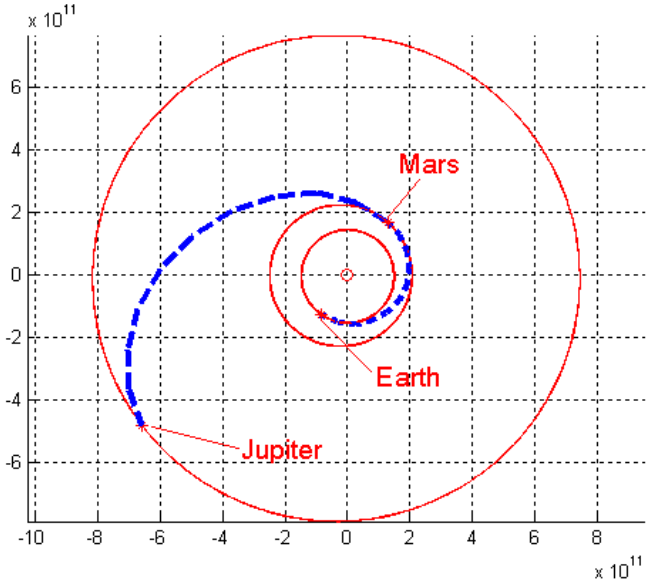

Figure 45: Optimum interplanetary trajectory from Earth to Jupiter, with gravitational assistance in Mars, obtained by PSO

The set of non-dominated solutions (Pareto-front) provided by the application of the MOPSO algorithm to the interplanetary trajectory defines the evolution and organization of the individuals (swarm particles), as clearly appreciated in Figure 46, where obtaining the actual Pareto-front is trivial. Many individuals are grouped around the coordinates of delta-V 10 km/s and mission duration 5000 days, quite in line with the results obtained by PSO and GA. However, in spite of the high number of individuals, no one has reached the optimum delta-V and mission duration solution provided by the single optimisation PSO and GA algorithms, due to the much larger area of the space of output variables covered by the multiobjective analysis (two output variables, rather than one), hence the less intensive exploration of possible solutions. Therefore, whereas the Pareto-front provides a useful view of possible non-dominated solutions, some further optimisation may be required once the user has chosen one of those solutions.

Figure 46 is a quite useful plot for mission design and can be used as an initial step to further optimise the interplanetary trajectories with single or multi objective algorithms. For instance, the mission design team may choose some adequate combination of delta-V and mission duration that fits some requirements, and then continue and develop their work from that area of the space of output variables using other optimisation

techniques. Ideally, the Pareto-front should be a set of all the non-dominated solutions of the optimisation problem, so the MOPSO algorithm provides a Pareto-front further down and left in Figure 46 and with a better and smoother distribution. While this is not always possible due to the strong discontinuities of the analysed interplanetary problem, some future work is planned in order to improve the behaviour of the MOPSO and to tailor its design and adapt its parameters to interplanetary mission design.

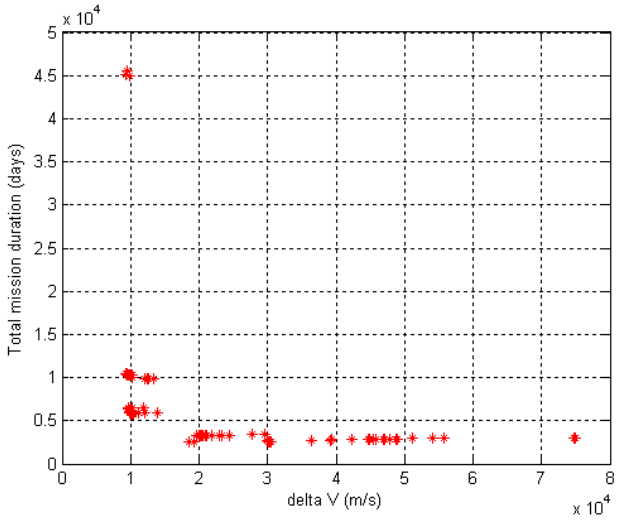

Figure 46: Whole population of MOPSO algorithm, representing possible trajectories from Earth to Jupiter with gravitational assistance in Mars

Figure 47 shows the trajectory for one of the MOPSO solutions in the range of delta-V around 10 km/s and mission duration 5000 days. This particular solution gives a total delta-V of 9.61 km/s. When arriving to Jupiter, an alternative to orbiting around this planet is to continue the journey without applying the last thrust, in which case the total delta-V is 5.22 km/s. This interplanetary trajectory is only one of the many calculated by the MOPSO.

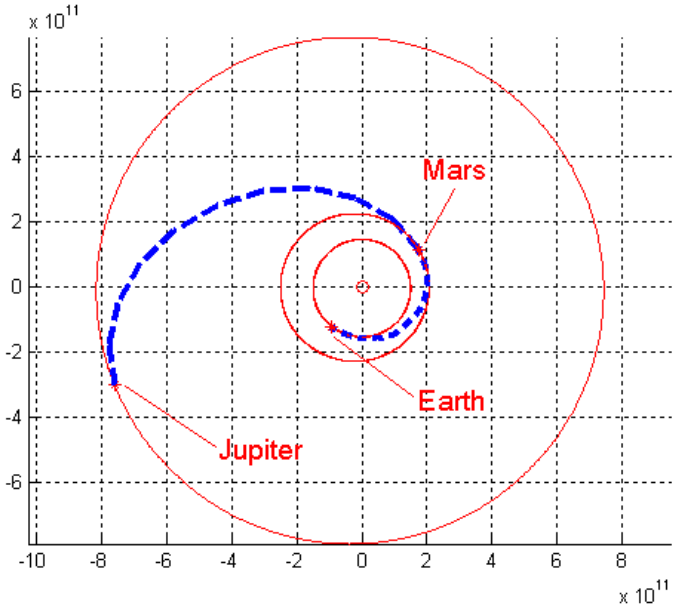

Figure 47: Interplanetary trajectory from Earth to Jupiter, with gravitational assistance in Mars, associated to one of the many swarm particles in the population generated by the MOPSO algorithm

Table 10 summarizes the results obtained by the application of the GA, PSO and the trajectory from MOPSO, and includes the values computed with the Hohmann transfer [40]. The Hohmann transfer is a two-impulses transfer between two different coplanar circular orbits, and it is an optimum impulsive transfer manoeuvre if the radius of the final orbit is less than 15.6 times the radius of the initial orbit (Jupiter fulfils this condition). Although the orbits of the Earth and Jupiter have different inclinations and are not perfectly circular, the calculation of the total delta-V of a Hohmann transfer provides a quite realistic approximation of the delta-V required for such an interplanetary mission, in case an optimum direct transfer is performed without fly-by in Mars. In Table 10, the values of delta-V at departure from the Earth, after the gravitational assistance in Mars, and at arrival to Jupiter are also presented.

In all cases the gravitational assistance in Mars is useful to significantly reduce the fuel budget. Each of the delta-Vs at departure and arrival are lower in case of a gravitational assistance in Mars, while the delta-V after the gravitational assistance is also relatively small. This means that smaller engines will be required for the probe, which is an additional and important advantage respect to applying a Hohmann transfer, as pointed out in [8]. Moreover, the multiobjective analysis and Pareto-front approach is very

useful to provide a wider view of all the possible solutions, so the optimum can be chosen by the mission design team without any constrain such as past experience or single objective decision.

	Δv at the Earth (km/s)	Δv after fly-by in Mars (km/s)	Δv after arrival to Jupiter (km/s)	**Total Δv (km/s)**
Hohmann transfer	8.51	-	5.57	**14.08**
GA	2.82	2.15	4.34	**9.32**
PSO	2.72	2.29	4.42	**9.43**
MOPSO	2.64	2.58	4.38	**9.61**

Table 10: Comparison of Delta-Vs for an optimum Hohmann transfer and the fly-by trajectory obtained by the GA, PSO and MOPSO algorithms (EMJ mission)

The total propellant shown in Table 10 refers to the case when the spacecraft remains orbiting around Jupiter. In case it continues the journey without applying the last thrust, the total delta-V would be smaller in all the cases, but the conclusions are the same.

Regarding the mission duration, Table 11 presents the results obtained by each optimization technique for the interplanetary trajectory. The departing date is calculated as number of days after J2000 [14].

	Departing date (after J2000)	**Mission time (days)**
Hohmann transfer		941
GA	93,225	3,836
PSO	19,131	3,687
MOPSO	1,958	6,351

Table 11: Comparison of results (departing date and mission duration) for an optimum Hohmann transfer and the fly-by trajectory obtained by the GA, PSO and MOPSO algorithms (Earth-Jupiter trajectory optimisation)

The first conclusion that can be drawn from Table 11 is that the mission duration and launch epoch may strongly vary. Quite a different launch epoch is reasonable due to the periodical nature of this input parameter. This departing date is expected to be periodical as the relative positions of Earth, Mars and Jupiter are repeated over time. Therefore, even very different launch epochs are reasonable and expected.

The time of flight (TOF) for GA, PSO and MOPSO is quite big in comparison to the Hohmann transfer. This is reasonable and expected as the interplanetary trajectory involves a gravity assistance in Mars, thus the date is strongly constrained and the launch window is less flexible than in the case of a Hohmann trajectory. Nevertheless, a lower TOF could be achieved at a higher consumption of fuel (higher delta-V) as appreciated in the Pareto-front in Figure 46.

The fact that similar results were obtained by GA and PSO for such different initial epochs suggest that, whereas single optimisation problems can be useful under some circumstances, Pareto-front analysis provides a much wider view. Indeed, GA and PSO provide one possible solution, but the mission designer has no opportunity to analyse other possible solutions, i.e., at different launch epochs. This analysis could be performed by running GA, PSO or any other single optimisation algorithm several times with different constrains for launch epochs, based on the experience of the mission design team. However, a more structured and highly automatic approach is provided by MOPSO, as the Pareto-front is useful to support decisions for such open problems with many different possible solutions.

The high potential of a powerful tool based on Pareto-fronts is quite interesting so the experience of the mission design team is not the limiting factor in interplanetary analysis and planning. The trajectory obtained by the MOPSO algorithm must be compared bearing in mind that this is just one of the many trajectories provided by the Pareto-front, chosen for comparison as it is in the same range of output values than the other solutions from GA and PSO.

5.3 Swarm Algorithms applied to Optimisation of a Trajectory from Earth to Saturn (EMVVJS)

5.3.1 Introduction

A very interesting state-of-the-art problem in aeronautical engineering nowadays is the optimisation of interplanetary trajectories implying gravitational fly-bys, which reduce total delta-V and duration of the mission. The lack of analytical solutions for such a complex problem, where exhaustive search of solutions is unaffordable, justifies the usage of artificial intelligence and modern optimisation techniques. Some examples of PSO and MOPSO applied to such problems can be found in sections 5.2 and 5.4 of this thesis, as well as in [6] and [11].

In addition to the EMJ mission [6] already presented in section 5.2, in [11] the results of a EMVVJS mission were also discussed. A EMVVJS mission consists on departure from Earth and fly-bys in Venus, Venus again, and Jupiter; the mission ends once in orbit around Saturn. Mission Cassini from NASA followed this route [42]. This mission is more ambitious and realistic than an Earth to Jupiter mission with a fly-by in Mars. The solution of a EMVVJS mission is complex due to the higher number of input variables. As in section 5.2, the variables to optimise are the date of departure from the Earth and the times of flight between planets, thus the dimension of this new optimisation problem is higher than when only one fly-by is performed. In spite of such a complexity, this optimisation problem is useful in real missions, due to the fact that better results in terms of fuel consumption can be achieved when sending a probe to the outer Solar System, if one or several fly-bys are made around Venus and Earth (also the case of mission Galileo from NASA). The usefulness of this kind of routes to Venus before reaching the outer Solar System has been confirmed in [11]. For details on the implemented PSO and MOPSO algorithm, please refer to sections 5.2.2 and 5.2.3 of this thesis. The results have been compared to [42].

5.3.2 PSO and MOPSO Results for Interplanetary Trajectory to Saturn

A complex mission has been analysed, based in Cassini [42]. This mission leaves the Earth, performs two fly-bys in Venus, one in the Earth, one in Jupiter and finally reaches Saturn. The delta-V has been computed as if a final LEO were achieved in Saturn. The PSO has been run 250 times in order to achieve statistically significant results.

First, the PSO included 1,000 individuals. Figure 48 shows the optimised delta-V against the number of iterations. The worse optimised value of delta-V is 6.85 km/s, and the best one 6.42 km/s. The mean value of these maximum and minimum optimised values is 6.66 km/s, while the mean value of all the optimised values is 6.58 km/s, and the standard deviation is merely 0.13 km/s. The best solution trajectory is represented in Figure 49, which duration is 5501 days.

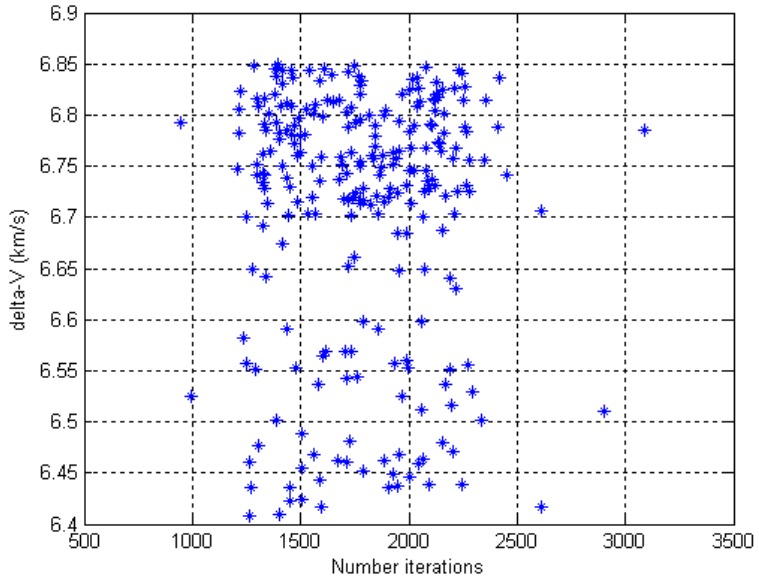

Figure 48: Optimised delta-V and number of iterations, for all the runs of the PSO algorithm applied to the EVVEJS mission (1,000 individuals)

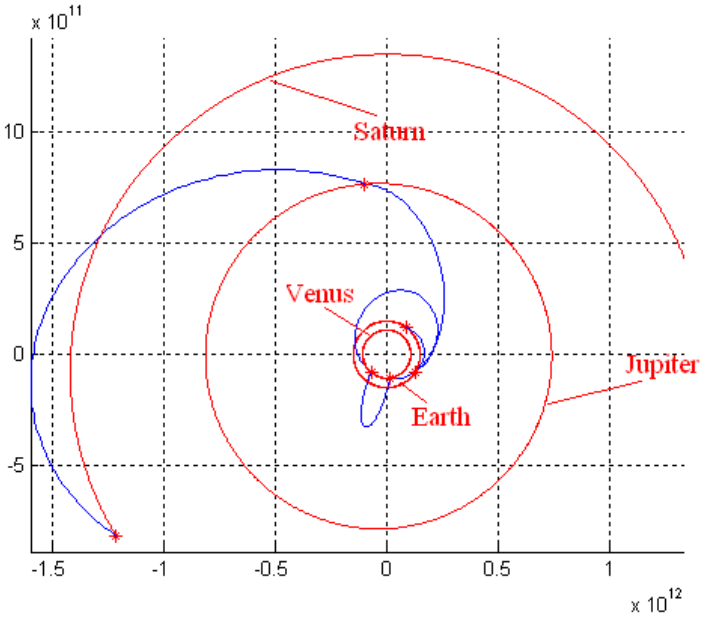

Figure 49: Interplanetary trajectory from Earth to Jupiter, with gravitational assistance in Mars, associated to one of the many swarm particles in the population generated by the MOPSO algorithm

A second batch of runs was performed with 10,000 individuals. Figure 50 shows the optimised delta-V against the number of iterations. The worse optimised value of delta-V is 7.70 km/s, and the best one 5.43 km/s. The mean value of these maximum and minimum optimised values is 6.57 km/s, while the mean value of all the optimised values is 6.67 km/s, and the standard deviation is 0.43 km/s. Figure 51 shows the best trajectory obtained with PSO, which duration is 6446 days.

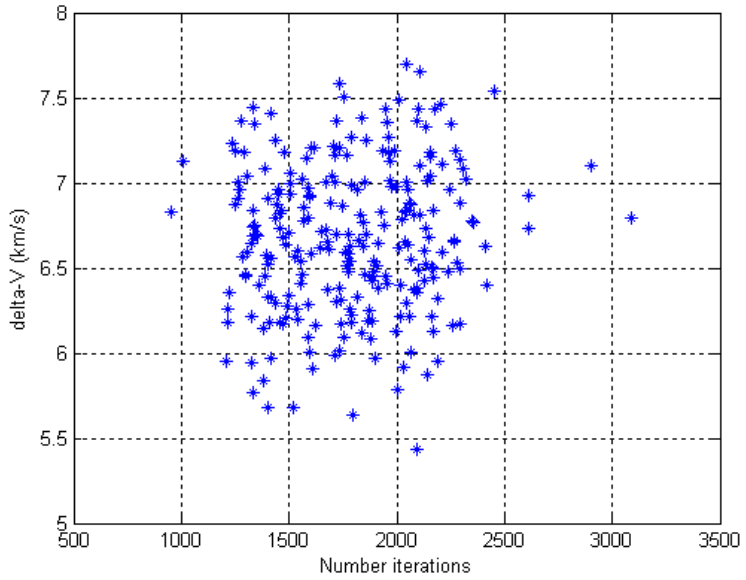

Figure 50: Optimised delta-V and number of iterations, for all the runs of the PSO algorithm applied to the EVVEJS mission (10,000 individuals)

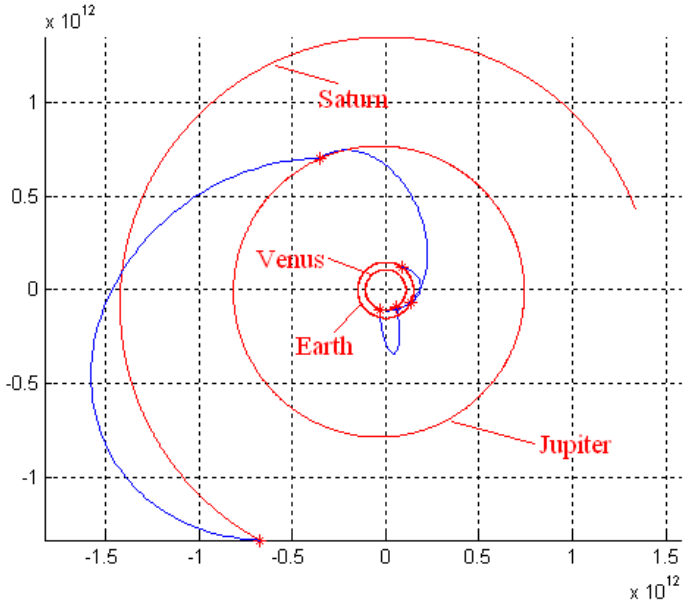

Figure 51: Trajectory obtained with a PSO algorithm (10,000 individuals)

An exhaustive analysis has been performed of the MOPSO algorithm applied to a EVVEJS mission, varying number of individuals in the whole population (20, 50, 75, 100, 200, 400) and size of the Pareto-front archive (20, 50, 75). The results are shown in Figure 52. Colours represent total number of individuals in the population: light

blue=20, green=50, magenta=75, dark blue=100, red=200, black=400. Symbols represent total number of individuals in the Pareto-front archive: * for 20, + for 50, x for 75. Please note that in some cases the Pareto-front archive became stable (a defined number of 100 iterations without changes), however not all of the solutions in that archive were part of a real Pareto-front, i.e., not all of them were non-dominated. Populations of 50, 75, 200 and 400 have obtained good results, specially if the Pareto-front archive had a size of 50 or 75 individuals. Poorer results were obtained with only 50 individuals in the Pareto-front.

Single-objective solutions are normally not possible in multi-objective analysis, thus the single-objective solutions obtained in section 5.3.2 (or similar) do not appear in Figure 52. Indeed, in spite of the high number of individuals, no one has reached the optimum delta-V and mission duration solution provided by the single optimisation PSO, due to the much larger area of the space of output variables covered by the multiobjective analysis (two output variables, rather than one), hence the less intensive exploration of possible solutions. Therefore, whereas the Pareto-front provides a useful view of possible non-dominated solutions, some further optimisation may be required once the user has chosen one of those solutions. In particular, populating the edges of the Pareto-front for single-objective analysis is not trivial and lead normally to poor results [92]. If the Pareto-front is extended left and up, this single-objective solutions would appear in the edge of such a front. If a single-objective solution is desired, then a single-objective technique should be used. The multi-objective techniques are appropriate to design a mission when there are concerns about the total time mission, and not only the total delta-V (for instance, human flight missions). While the criteria to define convergence may result arbitrary and be a study in its own, to the best of our knowledge there are not such a study and the situation has not changed respect to that reported in [92].

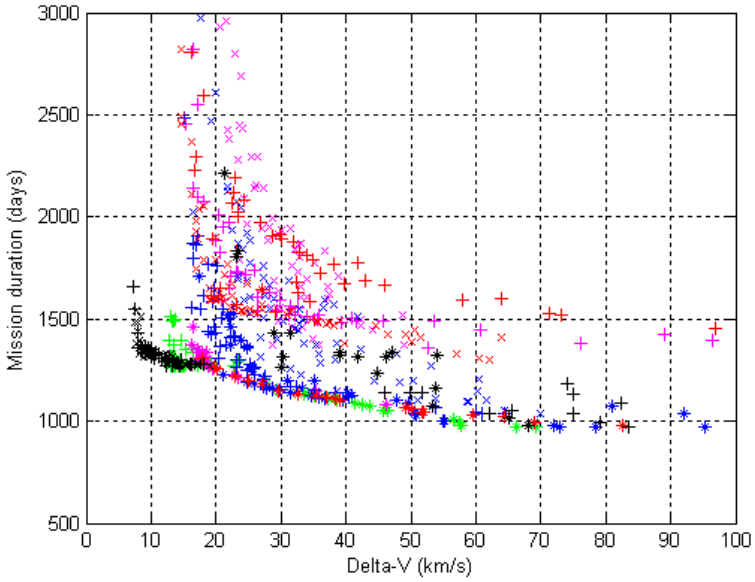

Figure 52: Multiple Pareto fronts obtained with MOPSO for EVVEJS mission.

For completeness, an arbitrary solution has been chosen from the Pareto front, near the lower-left "corner", obtained with 400 individuals, and a Pareto-front archive of size 50. For this solution, the delta-V is 7.88 km/s and the mission duration is 1361 days. The trajectory of that solution is shown in Figure 53.

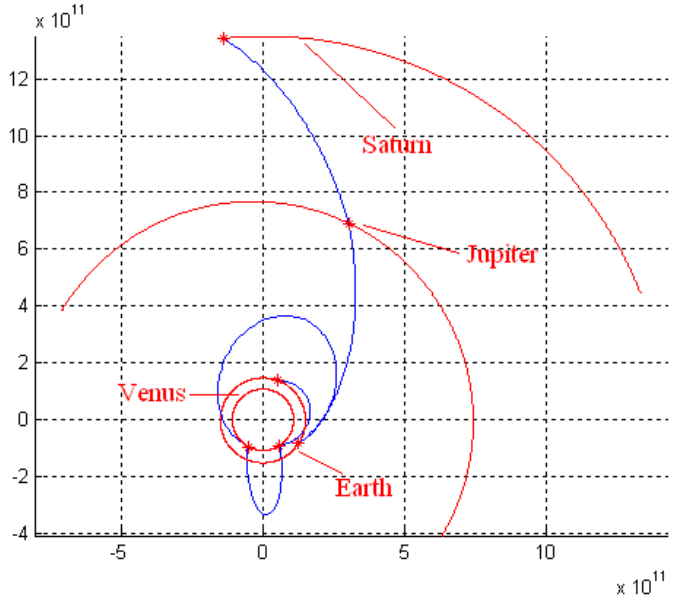

Figure 53: Interplanetary trajectory EVVEJS, associated to one of the many swarm particles in the population generated by the MOPSO algorithm

Table 12 summarizes the best results obtained by the application of the PSO algorithm to the EVVEJS mission, with 1,000 and 10,000 individuals. Table 12 also include the values computed with a Hohmann transfer. For completeness, information on the arbitrary trajectory chosen from the Pareto-front generated by MOPSO is also included in the table.

	First Δv Earth (km/s)	First Δv Venus (km/s)	Second Δv Venus (km/s)	Second Δv Earth (km/s)	Δv Jupiter (km/s)	Δv Saturn (km/s)	**Total Δv (Km/s)**
Hohmann transfer	10.29	-	-	-	-	5.44	**14.08**
PSO (1,000 indiv.)	3.09	1.85	0.90	0.11	0.013	0.44	**6.41**
PSO (10,000 indiv.)	3.62	0.96	0.19	0.13	0.078	0.46	**5.44**
MOPSO	3.61	1.88	0.71	0.52	0.24	0.91	**7.88**

Table 12: Comparison of Delta-Vs for an optimal Earth-Saturn Hohmann transfer and the fly-by trajectory obtained by the GA, PSO and MOPSO (EVVEJS mission)

In all cases the gravitational assistances in several planets are useful to significantly reduce the fuel budget. Analogously to the EMJ mission, each of the delta-Vs at departure and arrival in Earth and Saturn are lower in case of gravitational assistances in Venus, Earth and Jupiter, while the delta-V after each gravitational assistances are also relatively small, allowing smaller engines for the probe as an added value respect to a Hohmann transfer [8]. Once again, the multiobjective analysis and Pareto-front approach is very useful to provide a wider view of all the possible solutions, but comparisons between single-objective and multi-objective optimisation results are not strictly relevant beyond mere curiosity.

The total delta-V shown in Table 12 refers to the case when the spacecraft remains orbiting around Saturn. In case it continues the journey without applying the last thrust, the total delta-V would be smaller in all the cases, but the conclusions are the same.

Table 13 presents the results obtained by each optimization technique for the interplanetary trajectory. The departing date is calculated as number of days after J2000 [14]. Results from the arbitrary trajectory chosen from the Pareto-front obtained with MOPSO are also included in Table 13 for completeness, however, as explained above, comparisons between single-objective and multi-objective optimisations are misleading.

	Departing date (after J2000)	**Mission time (days)**
Hohmann transfer	-	2,223
PSO (1,000 ind.)	-777.1	5,501
PSO (10,000 ind.)	-776.1	6,446
MOPSO	-761.7	5,929

Table 13: Comparison of results (departing date and mission duration) for an optimum Hohmann transfer and the fly-by trajectory obtained by the PSO and MOPSO algorithms (EVVEJS mission)

Unlike the case of an EMJ mission, the higher number of involved planets strongly decreases the periodicity of the system, so in this case all the launch dates are similar. This similarity would disappear, if the launch date range scanned by the PSO and MOPSO algorithms were broader enough. Like in the EMJ mission, the Hohmann transfer yields to shorter mission durations. Once again, the structured and highly automatic approach provided by MOPSO is an asset when analyzing multiple constraints in mission design, providing a tool that eases the vision of several possible solutions at one glance.

5.4 MOPSO applied to Optimisation of Interplanetary Trajectories with Low Thrust

5.4.1 Introduction

One of the most fascinating aerospace problems today is the automatic, self-evolving optimisation of interplanetary trajectories. The computation of optimal interplanetary trajectories has raised a lot of interest in space agencies around the world, particularly in last decade as more interplanetary missions have been planned and analysed. In our article [6], we presented results of the optimisation, with a Multi-Objective Particle Swarm Optimisation (MOPSO) algorithm, of a mission from the Earth to the Kuiper belt, using gravitational assistances as necessary, minimizing mission duration and fuel consumption.

When optimising complex multi-output problems, the lack of analytical solutions yields to models requiring numerical simulations, whereas fully mathematical formulations are not possible and exhaustive exploration of the space of variables is unaffordable. The existence of numerous local minima, the weak "attraction" of the global solution in its surroundings (small attraction basin) or the presence of strong discontinuities are only some of the many difficulties that increase the complexity of optimisation problems. Indeed, the usage of some algorithm based on intelligent exploration of the space of variables is mandatory, such as genetic algorithms or swarm algorithms (see [52][38][39][61] and section 2.3.2 of this thesis).

Traditionally, a multi-input, single-output target function is optimised, combining all the outputs (fuel and duration in this case) into just one formula. However, it is not always easy to choose the right single-output formula, i.e., how important is the fuel respect to the duration of the mission. Whereas this conservative approach is useful in many contexts, sometimes the mission design engineer would rather prefer a global view of all the different (and sometimes spread) dominant solutions instead. A whole set of Pareto-front solutions is particularly necessary when analyzing if slightly worsening some outputs leads to much better improvements in other outputs, or if small changes in inputs may lead to catastrophic results. Whereas it is possible to carry out these kinds of

analysis from single-output solutions (i.e., applying gradient methods), some difficulties may arise whenever there are many variables, local solutions and discontinuities. Swarm algorithms avoid this trouble and support an innovative paradigm based on a muti-input, multi-output analysis, providing Pareto-front of dominant solutions of the problem, rather than just one single-output solution.

As in genetic algorithms, swarm optimization begins at a randomly generated population, however individuals are not mixed: for each particle, a velocity vector is defined, which represents the change in the position of the particle between two consecutive steps. Each velocity vector depends on the best historical position of that particle, the best global position in the whole population, and the best local position of a nearby group of individuals. The objective of the optimisation is to obtain the set of dominant solutions (Pareto-front) for a mission from the Earth to the Kuiper belt (which is assumed as located at 40 AU from the Sun). The launcher leaves the spacecraft on a escape trajectory from the Earth at 3.0 km/s in any desired direction (adjustable by the optimisation algorithm). The propulsion system of the spacecraft achieves an specific impulse of 2500 s and a maximum thrust of 0.05 N. The payload weights 500 kg. Gravitational assistances are considered in the algorithm. The outputs (Pareto-front variables) are the mission duration and the fuel consumption: the first is important to reduce the number of years of such a long mission, whereas the last defines the initial total weight of the spacecraft. Obviously, the higher the fuel consumption, the shorter the mission, and vice-versa; thus, the simultaneous optimisation of both variables is complex and compromising.

Rather than combining these outputs into a single formula, the objective of the swarm algorithm is to obtain a set of dominant solutions, i.e., solutions that are not surpassed in both outputs simultaneously by any other solution. Hence, rather than combining these output variables into a single criteria function to be optimised, a set of Pareto-fronts is reported, where each solution is non-dominated. The swarm algorithm designs the dominant trajectories, adjusting the sequence of impulses, thrusts directions and chosen planets for gravitational assistances during the mission. Hence, the swarm algorithm tunes the input parameters to generate such Pareto-fronts, providing a global view of the best possible solutions and selecting the most appropriate combinations of delta-V and

mission duration. Based on the obtained Pareto-front of dominant solutions, the user is able to choose the best optimal trajectory.

Multiobjective optimisation problems are quite common in every area of knowledge [47]. Some works are presented in section 5.1 of this thesis. In such problems, the objectives to be optimised are normally in conflict with respect to each other, which means that there is not a single solution for these problems. Following some previous work on Genetic Algorithms (GA) [3][5] in aerospace applications, two articles were subsequently published [7][8] and a third one was presented in a congress [9], the three of them with a direct relation to interplanetary missions. In these papers, genetic optimisation of single-criteria functions was applied for delta-V minimization, applying impulsive manoeuvres and planetary gravitational assistances. Whereas the approach in those two papers has some continuity in [6], two important differences should be noticed. First, a swarm algorithm instead of a GA has been implemented in [6] in order to achieve a Pareto-front of dominant solutions (rather than a single solution. Besides, two output variables total mission duration and delta-V, are simultaneously optimised. Second, the computation of the interplanetary trajectories is based on low thrust arcs, rather than on impulsive manoeuvres. With this purpose, first the desired trajectory is designed as if impulsive manoeuvres were possible. Then the impulsive manoeuvres are converted into low thrust ones, using a specific algorithm based on some hypothesis.

5.4.2 Implementation of MOPSO for Interplanetary Trajectory Optimisation with Low Thrust

Many different techniques can be applied when implementing MOPSO (see section 2.3.2.3). Regardless the implemented technique, PSO is a population-based, swarm intelligent, search algorithm based on the simulation of the social behaviour of birds within a flock [38], or any other swarm organism, using physical movements of the individuals within the swarm. This mechanism adapts to the global and local exploration of the search space area. Whereas other evolutionary algorithms are based on parent representation and selection of individuals, PSO and Multi-Objective PSO (MOPSO) relies on leaders to guide the whole population. PSO refers to single-output analysis, while MOPSO refers to PSO when more than one output is available [90].

One of the first works on swarm algorithms is reported in [69]. A survey of the state-of-the-art of particle swarm optimization is summarized in [92]. A comparison of multiobjective evolutionary algorithms is presented in [108], while in [71] some suggestions are explained for improving both spread and convergence in swarm algorithms and Pareto-fronts.

The aim of the MOPSO is to find a set of non-dominated solutions (Pareto-front), which have to be saved in some data structure during the execution of the algorithm. The size of this structure may grow in size very quickly, as more non-dominated solutions are found in every step. After each step, the new position of the individuals in the population must be compared to every solution in the Pareto-front, to perform some substitution if necessary. This increases the complexity and the computational time every step, as the Pareto-front becomes bigger. In order to ensure the variety of solutions, some mechanism must be used so the distribution of the decision vectors in the Pareto-front is smooth and uniform.

Dealing with these three issues leads to establish a limit of the number of solutions in the Pareto-front, up to 50 in the work presented in this section [6]. New positions of the individuals become candidates if they are non-dominated respect to any other Pareto-front solution. Before they finally join the Pareto-front (discarding some other solution already in the Pareto-front), a comparison is performed between the candidate to join and the candidate to be dismissed, in terms of distances to other solutions in the Pareto-front. The substitution is performed by considering only the highest distance, so the density in the Pareto-front is kept as low as possible. This ensures that variety of the solutions. Sometimes the Pareto-front is not as broad as expected, due to isolated solutions and strong discontinuities in global, single-output optima.

The tuning input parameters of the swarm algorithm are:

- the departure date from the Earth;
- the times of flight between planets;
- the final anomaly of the spacecraft when arriving at the Kuiper belt;
- the final time of flight from the visited planet (last gravitational assistance).

As a result, the swarm algorithm provides an useful Pareto-front of dominant solutions, from which the user may decide which solution or set of solutions are the most appropriate to design the Earth-to-Kuiper mission. That decision is based on the obtained delta-V and total time of flight of the mission.

Therefore, in this application the decision variables are,

x1 = times of flight

x2 = final anomaly

The output vector is,

y1 = delta-V

y2 = total time of flight.

The consideration of non-dominated solutions is used to take into account the trade-off between the two output variables. For example, some optimum solutions may provide the lowest delta-V but at the expense of higher total time of flight, while some other solution may reduce the time of flight but increasing delta-V. The set of non-dominant solutions provides a Pareto-front.

Two leading vectors are considered for each individual: the best personal solution, and the best global solution. The best personal solution is particular for each individual and matches the last achieved solution that is non-dominated by any other historical decision vector of that individual. The best global leader is clearly defined in PSOs but not in MOPSOs. In our algorithm, the global leader role has been substituted by a Pareto-front leader, tailored to each individual in the population as the nearest element in the Pareto-front. Both leaders (best personal leader, and best Pareto-front leader) are combined to define a certain "velocity" (i.e. a change in the decision vector) of each individual, according to (*). In the next step, the new individual "position", i.e. decision vector, will be updated as in (*). Up to 1000 steps have been considered in the simulation. The algorithm is interrupted if there are not changes in the Pareto-front of non-dominated solutions after 50 consecutive steps.

We may assume that the MOPSO performs a reasonable wide analysis of all possible combinations of times of flight, times of departure, etc, based on the intelligent modification of the decision input vector $\bar{x}$. If a given time of flight is feasible for

impulsive manoeuvres, but it is not for low thrust and therefore a different time of flight is necessary, this should be calculated by the MOPSO under a different decision vector $\bar{x}$. Then, when approximating an impulsive manoeuvre by a set of low thrust manoeuvres, the time of flight must remain fixed for that given $\bar{x}$. Subsequently, the total delta-V for that decision vector will increase as impulsive manoeuvres are substituted by low thrust manoeuvres. That is, shorter time after the whole delta-V is applied for nearly the same distance implies accelerating the probe further to achieve the specified time of flight. In some cases the low thrust requires a higher time of flight than that specified on $\bar{x}$. Then the mission becomes unfeasible and the MOPSO penalizes it with an arbitrary high delta-V and total time of flight, so this solution is not taken into account in next steps of the optimization algorithm.

5.4.3 Description of the Optimisation Problem

A model of the solar system has been implemented, based on the ephemeris of the planets as obtained from tables and formulae provided by [96]. As a first approach, we may assume impulsive manoeuvres at departure from the Earth and during fly-bys in each planet. Under this assumption, the trajectory from Earth to any other planet, after a given time, may be calculated by solving the Lambert´s problem [20]. This trajectory consists on an arc of an orbit around the Sun that is defined by heliocentric velocities at departure from Earth and at arrival in Mars. This planet has been chosen as the first planet for a fly-by after departure from Earth; however, any other combination of planets could be chosen to perform fly-bys (of particular interest are the fly-bys in Venus and Earth to reach outer planets). From the velocities at departure and arrival, the required delta-V for the impulsive manoeuvre at departure (provided by the launcher) may be calculated. The process is repeated for the second to the third planet trajectory and so on, calculating the appropriate delta-V during the fly-by to achieve the specified trajectory in the given time of flight. In this way, given a set of variables (decision vector $\bar{x}$, containing time of departure, times of flight between planets and final anomaly in the Kuiper belt), it is possible to calculate the total delta-V and total time of flight of the mission. The tuning of $\bar{x}$ is performed by the MOPSO, in order to obtain an interesting Pareto-front of non-dominated solutions. The sequence of planets is at the moment specified by the user. However, it would be interesting to make this list

automatic, for instance, using evolutionary programming on top of the MOPSO, which would analyse the convenience of certain combinations of planets in comparison to others.

In order to simplify this complex numerical problem, we may consider that the probe departs from a given LEO around the Earth, being the initial speed of the spacecraft respect to the Earth almost negligible in comparison to the velocity of the Earth around the Sun (designing an optimal launch is also possible [3][5][1], as explained in Chapter 4 of this thesis). Therefore, we may assume that the probe moves respect to the Sun at the same speed than the Earth. At arrival in a certain planet, the gravitational assistance trajectory consists on accelerating the probe, taking advantage of the gravity field of the planet and its heliocentric speed, respect to the heliocentric speed of the launcher. Each planet is considered a dimensionless point of mass at which the spacecraft changes magnitude and direction of its velocity instantaneously.

The instantaneous delta-V (impulsive manoeuvres) approach is interesting and has been depicted in papers such as [7][8][9]. But the implementation of low-thrust manoeuvres in this particular application would lead to some different results that require numerical simulations and not only the solution of the Lambert´s problem. In particular, since the low-thrust engines of the probe cannot provide the high demanding levels of thrust, an algorithm has been implemented in order to convert the impulsive manoeuvres into continuous. Although the following hypothesis may not be always true [6], they may be considered reasonable in most cases, justified by experience, and valid to find a first guess when implementing the MOPSO with low-thrust manoeuvres. Given an impulsive manoeuvre, its effects are maximized if it is performed at departure from Earth, or at the periapsis of a fly-by trajectory in any planet. Therefore, Deep Space Manoeuvres (DSMs) are in principle discarded except for trajectory corrections if necessary (out of the scope of this section).

Let it be a certain time of flight between planets, for which we may calculate a trajectory that fulfils such a time of flight. An impulsive manoeuvre shall be required at departure (after launching or after a fly-by), in order to achieve such a trajectory for the given time of flight. Let it be that, for any reason, the manoeuvre is not applied at departure, but after a certain time, and still the total time of flight must be fulfilled. If

so, the probe shall cover the same required distance, but now the time available after the manoeuvre shall not be the whole specified time of flight as before, but shorter, as the manoeuvre was applied not at departure, but some time later. This leads to the conclusion that the impulsive manoeuvre is greater if delayed than if applied at departure, as the same distance must be covered in less time. This reason seems to support the early application of delta-Vs, rather than DSMs. Said that, it is important to remark once again that celestial mechanics are too complex to assume that this hypothesis is always valid, however this is valid as a first approach for our intelligent algorithm.

Given a certain big impulsive manoeuvre, which value is too high for the actual engines, a less powerful manoeuvre can be performed in the same direction as the closest and most reliable approximation of the desired manoeuvre. After the feasible manoeuvre is performed, a new impulsive manoeuvre may be calculated, which (if necessary) could be once again substituted by a less demanding manoeuvre. This process can be repeated until the last computed impulsive manoeuvre is feasible for the low thrust engines. If the process does not end for the given time of flight, the conclusion is that the desired effects of the original high-impulsive manoeuvre are not possible with a set of lower impulsive manoeuvres, and such high-impulsive manoeuvre is therefore discarded. A set of many low impulsive manoeuvres is the approximation of a sustained low thrust, continuous manoeuvre. In the process described above, this continuous thrust would saturate the engines until the appropriate trajectory is achieved (i.e., arrival at destination in a given time).

When an impulsive manoeuvre is required to correct the trajectory (if necessary) during a fly-by, this is usually applied at periapsis, with the purpose of maximizing its effects (Orbeth principle [20][21][40]). This is not possible when applying low thrust, as the continuous manoeuvre may last days or even months, whereas the fly-by lasts just a few hours. Therefore, when using low-thrust, the desired impulsive manoeuvre is calculated as if it were applied after the fly-by at the planet, before it is converted into a low-thrust manoeuvre.

According to the hypothesis presented before, the following algorithm has been implemented in order to convert an impulse manoeuvre into a low thrust manoeuvre (Figure 54):

1) Given a certain planet (originally, this could be planet Earth at departure), compute the trajectory to the next planet, using some Lambert's solver. Such trajectory demands some initial impulsive manoeuvre, which is the required delta-V at departure from Earth, or when leaving any other planet after a fly-by.

2) The required delta-V for the impulsive manoeuvre is calculated as the difference between the actual speed of the probe (due to the speed of the Earth, or due to the fly-by at any planet) and the speed that is required by the Lambert solver in the step above. If the delta-V calculated is feasible for the low thrust engines, then this delta-V is applied accordingly during the given delta time (1000 seconds in this case).

Otherwise, if this delta-V is too high, the thrusters will saturate during a given delta time, according to (4). The trajectory of the probe is then simulated using a Runge-Kutta integration method during that time. After that, a new position is achieved and the two steps above are repeated.

$$\Delta\bar{V}_{actual} = \frac{\Delta V_{\max}}{\Delta V}\Delta\bar{V} \qquad (57)$$

where:

$\Delta\bar{V}_{actual}$ is the applied delta V in the considered time step

$\Delta\bar{V}$ is the desired delta V

$\Delta\bar{V}_{\max}$ is the maximum feasible delta V provided by the continuous low thrust engines in each time step, as defined by (58):

$$\Delta V_{\max} = I_{sp} g \ln\left(\frac{m}{m + \dot{m}\Delta t}\right) \qquad (58)$$

where:

$\Delta V_{\max}$ is the maximum low-thrust delta-V provided by the engines during the given delta time Δt

I_{sp} is the specific impulse of the propulsion system

g is the acceleration of the Earth´s gravity field in the surface of the planet

m is the mass of the probe, and $\dot{m}$ is the flux of mass due to the propulsion system

3) Once the desired trajectory is achieved (free drift in some heliocentric arc of Keplerian orbit), the whole process is repeated after the fly-by at destination.

4) The final position in the Kuiper belt is calculated as for any planet. Once this position is achieved, the algorithm stops.

This algorithm is repeated for every trajectory between any two planets, or between the last planet and the Kuiper belt.

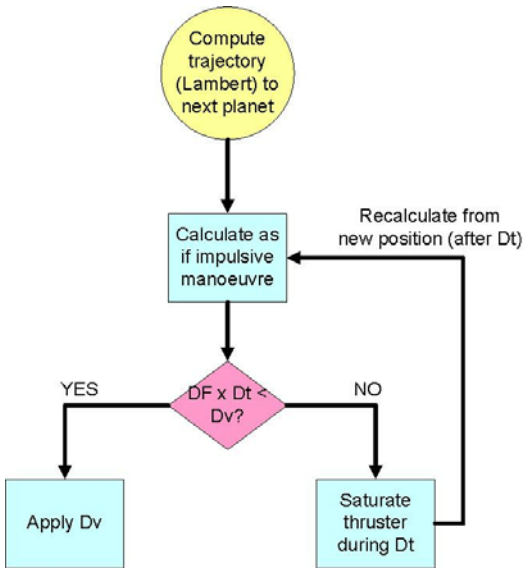

Figure 54: Scheme of the algorithm that calculates the low thrust trajectory between two planets.

5.4.4 MOPSO Results for Interplanetary Trajectory with Low Thrust

Attending to the possible fly-bys in the solar system, the following cases have been considered:

- fly-by in Mars
- fly-bys in Mars and Jupiter
- fly-bys in Mars and Saturn
- fly-bys in Mars, Jupiter and Saturn.

Figure 55 shows several Pareto-fronts for the Earth-Kuiper mission for the different gravitational assistances in different sets of planets. The Pareto-front in black represents the non-dominated solutions for the Earth-Kuiper mission when a fly-by in Mars is performed. The obtained delta-Vs are quite high, while the times of flights (TOFs) are quite low, but not lower than with other gravitational assistances in Jupiter and Saturn.

Indeed, the black front is dominated by the blue (Mars and Jupiter) and green (Mars and Saturn) fronts. The red Pareto-front represents the solutions for fly-bys in Mars, Jupiter and Saturn, which seems to provide a worse solution than with a fly in Mars and in only one more planet. Red and black fronts could be part of the same Pareto-front, as none of them dominate each other.

Some conclusions may be drawn in this case. The fly-by only in Mars is not a good solution, as it provides higher delta-Vs than other options, even if the total time of flight is quite reasonable. Fly-bys in Mars, Jupiter in Saturn lead to inappropriate spatial configurations that are not useful. Perhaps in different periods of time the positions of these three planets were more adequate to provide a good trajectory, but in the analysed period the imposition of a third fly-by prevents the MOPSO from finding good optimum solutions. However, it is important to note that the best delta-V is achieved by this Mars-Jupiter-Saturn trajectory (upper-left end of the red Pareto-front). The trajectory for this minimum delta-V solution is shown in Figure 55, where most of the planets and also the Kuiper belt are reached after a nearly 180 degree orbit from the previous planets (Hohmann-alike transfers). Especially notable is the extremely short low-thrust period after Saturn, as the fly-by seems to be enough to reach the proper trajectory to the Kuiper belt.

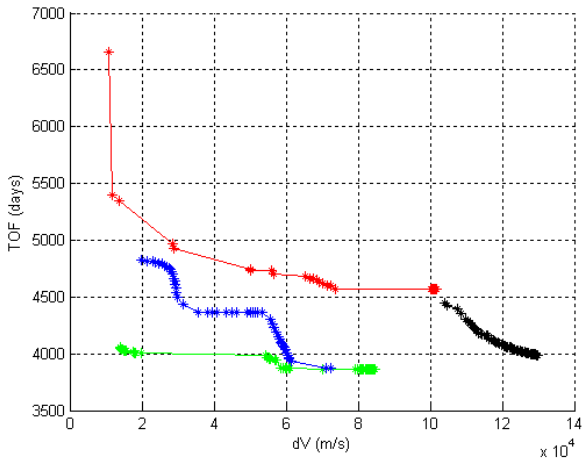

Figure 55: Pareto-fronts of solutions (TOF vs. delta-V) for a mission from the Earth to any position in the Kuiper belt, with fly-bys in Mars (black); Mars and Jupiter (blue); Mars and Saturn (green); and Mars, Jupiter and Saturn (red).

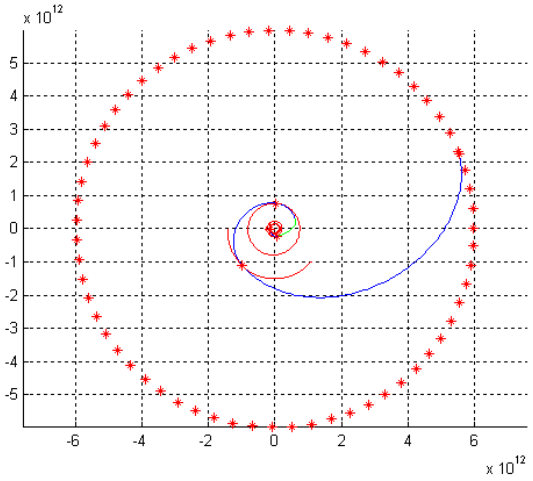

Figure 56: Trajectory for an Earth-Kuiper belt mission with fly-bys in Mars, Jupiter and Saturn. Low-thrust segments are represented by green non-keplerian arcs.

In spite of the few solutions that minimize the delta-V for the triple gravitational assistance, most of the Pareto-front solutions come from two fly-bys, in Mars and then Jupiter or Saturn. In general, Saturn provides the best results, as the green Pareto-front dominates nearly all the other Pareto-front solutions. However, it is important to remark that, if the launch epoch (i.e. departure from the Earth) is appropriate, Jupiter may be better for fly-bys (due to its greater gravitational field [64]), and that there may exist

some other combinations of planets (involving fly-bys Venus and also the Earth [89], quite sophisticated and out of our scope, to reach Jupiter or any other outer planet.

Pareto-front diagrams like the one shown in Figure 55 are necessary to have a global view of the different solutions obtained for the mission. For instance, it is possible to see how a fly-by in Mars, Jupiter and Saturn provides the best delta-V solution but the TOF is higher. With just a bit higher delta-V, TOFs could be drastically reduced if the fly-by in Jupiter is skipped. At a quite greater consumption of delta-V, an even slightly lower TOF can be obtained, and at some point a fly-by in Jupiter instead of Saturn may also be convenient. Depending on the period considered, a Mars-Jupiter-Kuiper belt mission may require a longer total TOF at the same delta-V that a Mars-Saturn-Kuiper belt mission.

5.5 Conclusions

A genetic algorithm has been used in section 5.1 to optimise the manoeuvres involved in an interplanetary trajectory, from a LEO around the Earth to planets Jupiter and Saturn, with a fly-by in Mars [7]. Remarkable results have been achieved in comparison with the optimum Hohmann transfer, saving simultaneously delta-V in all the different stages of the mission, whereas the duration of the mission has also been shortened. The relatively low delta-V at each stage (departure, fly-by and arrival) is useful to reduce the size of the spacecraft's engines respect to those required if a Hohmann transfer were applied. Since some assumptions were made in the models (valid from the point of view of mission analysis), the obtained solution could be refined using more accurate numerical simulations (considering, for example, precise orbital perturbations and corrections). Since such a simulation would be extremely time-consuming, the solution provided by the genetic algorithm in this article would not only be useful, but also a must, when designing such an Earth-to-Jupiter or Earth-to-Saturn interplanetary mission. As an added value to traditional GAs, a variable mutation rate has been implemented in section 5.1, with satisfactory results. Raising the dynamic mutation rate when the population becomes homogeneous broadens the exploration area in the space of potential solutions, avoiding local maximums, and improving the successful and efficient search for an optimum solution. Other approaches to the optimisation problem

described in section 5.1 are also possible, such as Pareto-based methods [52], which have also been proved as efficient multi-objective optimizer methods [47][65][108]. The application of this and other techniques [6], and the subsequent comparison with the results obtained by the GA in this article, was the interesting target of works to come, and already presented in this thesis (see sections 5.2, 5.3 and 5.4), where both mission time and delta-V could be optimised simultaneously, obtaining sets of dominant solutions rather than a single optimum solution.

Two swarm particle optimisation algorithms, one single objective (PSO) and one multiobjective (MOPSO) were analysed and applied to optimise the manoeuvres involved in interplanetary trajectories in sections 5.2 and 5.3. Two missions have been studied: EMJ, and EVVEJS. Satisfactory results have been achieved in comparison to the optimum Hohmann transfer. The relatively low thrust at each stage (departure, gravitational assistance and arrival) is useful to reduce the size of the spacecraft's engines. The two swarm algorithms have provided interesting results in the case of a EMJ mission. While the PSO provides similar results than GA, the MOPSO algorithm increases the searched area in the space of output variables and provides a better view that may be used for further optimisation with single or multi objective algorithms. The two swarm algorithm also provided interesting results in the case of a EVVEJS mission. Variable limits were implemented in the single-objective case for each of the decision variables, approaching the best solution while scanning its surroundings more accurately as the algorithms goes on. In the case of multi-objective, several Pareto-fronts has been provided, based on different sizes of population and Pareto archive. The results for all the cases has been presented and analyzed with plots and numerical data.

A MOPSO algorithm has been implemented in section 5.4 to solve the problem of designing a mission from the Earth to the Kuiper belt, using low thrust. A first approach was made using impulsive manoeuvres, which are subsequently converted into low-thrust manoeuvres. Different decision vectors are tried by the MOPSO, using some techniques to obtain better, non-dominated solutions after each step. Fly-bys in Mars, Jupiter and Saturn have been considered and proved to be quite useful to reduce the total delta-V of the mission (otherwise very expensive in terms of delta-V and TOF). Fly-by only in Mars has been discarded, as it implies a higher delta-V for the same TOF that can be achieved with a second fly-by in Jupiter or in Saturn with less effort. Fly-bys

in Mars, Jupiter and Saturn have also been discarded, as the result is dominated by other Pareto-fronts; however, the optimum delta-V solution belongs to this triple fly-by mission. Most of the solutions of the Mars-Jupiter missions are dominated by the Mars-Saturn missions, as Saturn seems the best option to optimise the problem in most of the cases. The presented low-thrust algorithm always gives good results if the hypothesis of the starting point are fulfilled, which could be not necessarily the case. Some other low-thrust approaches may be interesting, in combination with Pareto-fronts and MOPSOs. In particular, exponential sinusoids, or non-tangential low-thrust. Whereas the solutions provided by MOPSO are highly interesting, a second objective is fulfilled: to show the powerful combination of Pareto-fronts and the optimization MOPSO algorithm in order to obtain non-dominated solutions, in an approach quite different from the traditional single-output optimisation. Indeed, the Pareto-front approach provides a global view that the user would not achieve with a single-criteria optimum. The results are both satisfactory and promising, providing a guiding path of new innovative techniques to solve the proposed low-thrust optimisation problem. The shown technique could be easily applied to other kinds of missions, like visiting the outer planets for instance. The Pareto-front MOPSO approach provides solutions that could be extended for more than two-outputs criteria functions, as it has been presented.

Future works are recommendable in order to improve the behaviour of the PSO and the MOPSO. Whereas the results from PSO were satisfactory, some tailoring of the algorithm to interplanetary missions may be an interesting improvement both in computing time and results. Similarly, whereas the results from MOPSO were also interesting when obtaining the Pareto front of non-dominated solutions, some modifications and tailoring to interplanetary missions is worth in order to improve the computation of the Pareto-front towards further optima non-dominated solutions and to smooth the density of the Pareto-front. Indeed, the results presented in this article were quite interesting and open branches to several lines of investigations for optimisation of interplanetary trajectories based in PSO and MOPSO algorithms. Future works also require some improvement on the extremely time-consuming algorithms. This is a necessary step to obtain more Pareto-fronts optimization for other combinations of fly-bys. Particularly interesting are the fly-bys in the inner planets (specially Venus, as in section 5.3), as well as fly-bys in Earth after departure, that altogether prove useful to reach Jupiter and Saturn with lower delta-V, but probably with a higher total TOF. A

robust, quick Lambert solver is required for MOPSO, as otherwise the task to populate a Pareto-front diagram with trajectories visiting many planets may become unaffordable for more intensive analysis. Eventually, evolutive programming on top of the MOPSO could be interesting, to automatically define the best sequence of planets to visit during the mission.

Chapter 6

Conclusions and Future Work

The goal of this PhD thesis was to describe the research drawn on the application of artificial intelligence to aeronautical problems. Following subjects were covered by this thesis: modelling, simulation and fuzzy control of satellites in Lagrange points; optimisation of launchers with genetic algorithms; and optimisation of interplanetary trajectories with particle swarm algorithms. The conclusions of this research is presented and discussed in this chapter. Also, some guidelines for future works, based on the research of this thesis, are analyzed.

6.1 Conclusions

6.1.1 Control of satellites in Lagrange points

In Chapter 3 the modelling, simulation and control of satellites in Lagrange points were analyzed.

The research focused on the implementation of PD controllers, both conventional and fuzzy, for five satellites, each one placed in the surroundings of the five Earth-Moon Lagrange points. The models include the simulation of effects such as: solar wind, non-spherical gravitational fields and the drift of the orbital parameters of the Moon.

The models were developed in Modelica and Dymola, containing some intrinsic advantages:

- Object oriented models: This feature is useful to create basic elements that may be combined to implement more complex entities or whole scenarios. The objects may be modified or increased in a simple way, without the need of changing the implementation of the simulation.

- Simulation of dynamic systems: Nowadays, many engineering fields require the implementation of complex models whose parameters vary in time. Dymola provides a good environment to develop models under Modelica, supporting the implementation of such complexity.

- Equation-oriented language: Modelica is a modelling language that supports the implementation of equations directly, this is, the equations are written and coded in Modelica and later solved by Dymola. This approach is desirable, rather than implementing the solution of the equations of the models, as with any conventional programming language. In particular, when coding the positions of the Lagrange points, the programmer does not need to solve the equations, but

just to write the equations in Modelica so later Dymola finds the solution during the simulation.

Results of the fuzzy controllers were presented and compared to those from classic controllers. In total, four different controllers were implemented: P and PD, classic and fuzzy.

Under free drift conditions, only two of the five Lagrange points are stable, in case of perfectly circular orbits of the two central bodies around each other, and in absence of other perturbations. In the real Earth-Moon system, none of the Lagrange points are stable, however the drift around two of the five Lagrange points is quite low. Nevertheless, it is possible to keep the position of the satellites in any of the five Lagrange points at quite low thrust if, in general, any of the studied controllers (P and PD, classic and fuzzy) is applied. The low consumption ensure position keeping for long periods of time, aiming longer missions.

The best results were obtained when the derivative control was included and thus the initial transitory smoothed. Fuzzy logic is also an added value to conventional P and PD controllers, due to the presence of a threshold in thrust activation, reducing the thrust demand and permitting fuel saving and even longer missions.

6.1.2 Genetic Algorithms and Launchers Trajectory Optimisation

Three different launch problems were analyzed in Chapter 4.

First, a vertical launch was studied (section 4.1), in which the target function was the final momentum of the spacecraft. Second (section 4.2), a launch trajectory was studied, combining several manoeuvres into a whole launch, and performing manoeuvres in each segment such as the pitch-push over manoeuvre, gravity turn and the tangential bilinear manoeuvre. The purpose of this launch trajectory was to place a satellite into a circular orbit around the Earth. Finally (section 4.3), based on similar models and manoeuvres than in the second case, a third case was considered with the purpose of placing a satellite into an hyperbolical trajectory, maximizing the final speed.

A variable mutation rate was introduced in the execution of the GAs, in order to ensure variety in new individuals whenever the population became homogeneous and tend to gather in the same area of the space of input variables; and to encourage crossing, whenever the population is spread and become heterogeneous. Indeed, raising the dynamic mutation rate when the population becomes homogeneous broadens the exploration area in the space of potential solutions, avoiding local maximums, and improving the successful and efficient search for an optimum solution.

For the first case (section 4.1), a quite simple model was implemented, based on a two-stages launcher and including atmospheric models so the air drag was taken into account. The tuning parameters of the model were the fuel masses available in each of those stages. Whereas the GA found the optimum solution, this was not enough to achieve the escape velocity, given a purely vertical launch trajectory. Indeed, a purely vertical launch is normally avoided in real scenarios, as confirmed by this analysis.

For the second case (section 4.2), a more complex model was in place, also simulating manoeuvres such as gravity turn, pitch-push over and bilinear tangential manoeuvre. The purpose of the optimisation was to achieve a circular orbit maximizing final height. The model of the rockets of the launcher was based on a real launcher. The payload of the launcher was fixed, whereas the tuning parameters were the duration of the vertical ascent, the duration of the pitch-push over and the final pitch-push over angle. The evolution of some relevant figures was shown for the optimal solution: altitude in time, altitude profile, launcher speed and launcher acceleration, as well as the launcher flight path angle and the evolution of the mass of the launcher. The shape and values shown on those plots were commented.

The third case (section 4.3) is an extension of the second case in section 4.2. Realistic manoeuvres were once again simulated: gravity turn, pitch-push over and bilinear tangential manoeuvre. Whereas the problem and models were similar to those in the second case, this time a different criteria function was implemented, as the goal of the GA was to maximize final speed in a hyperbolical escape trajectory. The tuning parameters were the duration of the vertical ascent, the duration of the pitch-push over and the final pitch-push over angle. Like in the second case, the evolution of some

relevant figures was shown and commented for the optimal solution: altitude, speed, acceleration, flight path angle and mass of the launcher.

As shown for all of the different cases analyzed and studied in this PhD thesis, GAs are a useful methodology to optimise the trajectory of launchers, leading to successful results. The solutions obtained by GA can be used directly, or taken as an initial guess for further refining and finding a more accurate solution based in complex tables and models from industry.

6.1.3 Optimisation of Interplanetary Trajectories

Evolutive algorithms have been proposed in Chapter 5 to solve the problem of optimising interplanetary orbits, based on GA, PSO and MOPSO. The probe leaves the Earth and arrives in some other planet in the Solar System, after performing gravitational fly-bys whose aim is to accelerate the probe at low or no fuel cost. Lowering the fuel cost is useful to increase mission durations, to increase the available weight for payload, and to design lighter launchers when leaving the Earth.

Whereas the evolutive algorithms implemented in Chapter 5 were used independently and their solutions could be used directly, those algorithms may be combined between them or with other optimisation techniques in order to further improve the obtained solutions, which therefore could be used as an initial step for more sophisticated optimisations based on detailed aerospace models for particular missions and spacecrafts from industry.

GA algorithms (section 5.1) were used to optimise two trajectories: Earth-to-Jupiter and Earth-to-Saturn. In both cases, a fly-by in Mars was performed. The goal of the GA is to minimize the total delta-V spent during the interplanetary mission. The propulsion system is able to perform impulsive manoeuvres. The tuning parameters were the launch epoch and the time of flight of each of the two arcs of the interplanetary trajectory (from Earth to Mars, and from Mars to the destination). Each arc is solved using the methodology of the Lambert problem: based on the position of the planets and the tuning time of flight, the required speeds at the extreme of the arc are computed.

A PSO algorithm (section 5.2) was used to optimise a trajectory from the Earth to Jupiter, using impulsive manoeuvres and a fly-by in Mars. The goal was to minimize the total delta-V during the interplanetary mission. The tuning parameters were the launch epoch and the time of flight of each of the two arcs of the interplanetary trajectory (from Earth to Mars, and from Mars to Jupiter). Each arc was solved with the Lambert methodology. The results were compared to those obtained with a GA applied to the same problem.

In section 5.2 as well, a MOPSO algorithm was implemented to solve again the Earth-to-Jupiter trajectory, obtaining an interesting Pareto-front that minimizes two conflicting outputs such as total mission duration and total mission delta-V. One of the individual MOPSO solutions was compared to the ones obtained with PSO and GA.

Similar approach than in section 5.2 was applied to a more complex mission in section 5.3, where the trajectory consisted on a departure from Earth, two fly-bys in Venus, one in Earth and one in Jupiter, before finally arriving in Saturn. This optimisation problem was analysised with PSO and MOPSO, and the results were compared. These results proved useful and confirm that, in order to reach the outer planets, fly-bys around Venus and Earth may yield to lower fuel consumption, thus increasing mission duration and payload.

A different approach was followed in section 5.4, where a MOPSO was implemented to optimise interplanetary trajectories. In that case, both the mission duration and the total delta-V were minimized, obtaining interesting Pareto-fronts for several missions. The common goal of all the studied missions was to reach the Kuiper belt, performing fly-bys in one or several planets. The available fly-bys for each of the four missions were: only Mars; Mars and Jupiter; Mars and Saturn; Mars, Jupiter and Saturn. The propulsion system provided low, continuous thrust, rather than impulsive manoeuvres. The design of the low thrust segments of the trajectory was based on impulsive manoeuvres (computed with the Gauss algorithm) that then were replaced by continuous low-thrust manoeuvres with approximately the same delta-V. The Pareto-front results were presented and discussed.

6.2 Future Work

The successful results presented in [4] and [10] regarding the modelling and simulation of satellites in Lagrange points can be further researched, considering different tunings of the P and PD controllers, both classic and fuzzy. Other types of controllers should also be interesting, such as non-linear controllers. An approach combining impulsive manoeuvres and continuous thrust would be remarkable, if the advantages of both techniques are successfully mixed. Regarding the development of models in Modelica, some of the objects developed in [4] and [10] could become part of some aerospace library, perhaps covering the current gap in the Modelica Standard Library, where not too many aerospace and astrophysics objects could be found.

In [3][5][1] some interesting results were presented in the field of launch design and trajectories optimisation. Whereas the possibilities are endless, guidelines for future work are the research of other launch manoeuvres and strategies, including models based on actual launchers and different payloads. More complex modelling is also possible, regarding the launch direction and the orientation of the final orbit or trajectory.

Concerning the optimisation of interplanetary trajectories with gravitational assistances, some research should be done for more complex trajectories involving inner planets or fly-bys in the same planet during a single mission. Particularly well-known is the option to perform a fly-by in Venus and Earth in order to reach outer planets, as shown in [11]. Whereas the works presented in [7][8][9][11] focus on impulsive manoeuvres, a quite interesting approach based in low-thrust was shown in [6]. Continuous low-thrust optimisation is a highly complex subject currently researched around the world in universities and space agencies, and some advances in that subject would be definitely an asset in future missions. Due to the complexity of the time-consuming models, and considering the high number of calls to these models when running evolutive algorithms, some improvements would lead to reducing computing time and increasing the number of analyzed solutions, as well as raising the density of the Pareto-fronts provided by multi-output algorithms.

References

[1] Alonso Zotes, F., Santos., M.. Optimización de lanzamiento vertical de dos fases utilizando un algoritmo genético multicriterio. Actas XXVIII JJAA, Spain, 2007.

[2] Alonso Zotes, F., Santos, M. GA Optimization of the height of a Low Earth Orbit. FLINS 2008, 21-24 September, 2008. Madrid, Spain. 2008a.

[3] Alonso Zotes, F., Santos, M. GA Optimization of the height of a Low Earth Orbit, in: Ruan, D., Montero, J., Lu, J. Martínez, L., D'hondt, P., Kerre E. (Eds.), *Computational Intelligence in Decision and Control*. World Scientific, pp. 719-724. 2008b.

[4] Alonso Zotes, F., Santos M. Modelado y simulación con Modelica y Dymola de cinco satélites en los puntos de Lagrange del sistema Tierra-Luna. Actas XXVII JJAA, Spain 2008c.

[5] Alonso Zotes, F., Santos, M.. Multi-criteria Genetic Optimisation of the Manoeuvres of a Two-Stage Launcher. *Information Sciences*. 180(6), 896-910. 2010a.

[6] Alonso Zotes, F., Santos, M., 2010b. From Earth to Kuiper belt: swarm optimisation algorithm applied to interplanetary missions. Proc. WPP-308, 4th International Conference on Astrodynamics Tools and Techniques ICATT, Madrid, Spain, 3-6/05/2010. 2010b.

[7] Alonso Zotes, F., Santos, M. Delta-V genetic optimization of a trajectory from Earth to Saturn with fly-by in Mars. Conf. Proc. 2010 IEEE WCCI (CEC 2010, 18-23/07/2010, Barcelona, Spain), pp. 1836-1841. 2010c.

[8] Alonso Zotes, F., Santos, M. Genetic Optimization of an Interplanetary Trajectory from Earth to Jupiter, in: Ruan, D., Li, T., Xu, Y.,Chen, G., Kerre G. (Eds.), *Computational Intelligence. Foundations and Applications*. World Scientific, pp. 1048-1053. 2010d.

[9] Alonso Zotes, F., Santos, M.. Genetic Optimization of an Interplanetary Trajectory from Earth to Jupiter. FLINS 2010, 2-4 August, 2010, Chengdu, China. 2010e.

[10] Alonso Zotes, F., Santos, M. Modelling, Simulation and Control of Satellites in the lagrangian points of the Earth-Moon System. *Revista Iberoamericana de Automática e Informática Industria (RIAI)*, vol. 8, 204-215, 2011a.

[11] Alonso Zotes, F., Santos, M. Particle Swarm Optimisation of an Interplanetary Trajectory from Earth to Jupiter. *Engineering Applications of Artificial Intelligence (EAAI). Elsevier*. In press, 2011b.

[12] Alonso, J., Serrano, N., De Pedro, T., González, C., García., R. Optimization of an Autonomous Car Fuzzy Control System via Genetic Algoritms. Proceedings I Workshop on Genetic Fuzzy Systems. Universidad de Granada. March 2005.

[13] Alvarez-Benitez, J.E., Everson, R.M., Fieldsend, J.E.. A MOPSO algorithm based exclusively on Pareto Dominance Concepts, *LNCS*, 3410, 459-473, 2005.

[14] Aoki, S., Soma M., Kinoshita, H., Inoue, K. Conversion matrix of epoch B 1950.0 FK 4-based positions of stars to epoch J 2000.0 positions in accordance with the new IAU resolutions. *Astronomy and Astrophysics* 128(3) 263–267. 1983.

[15] Aranda, J., Crespo, J. Modal Analysis Applied to Spacecraft Attitude Control. AIAA Guidance, Navigation and Control Conference and Exhibit, 2007.

[16] Aranda, J., De la Cruz, J.M., Díaz, J.M. Design of a multivariable robust controller to decrease the motion sickness incidence in fast ferries. Control Engineering Practice, 2005.

[17] Aranda, J., De la Cruz, J.M., Parrilla, M., Ruipérez, P. Evolutionary algorithms for the design of a multivariable control for an aircraft control flight. AIAA Paper 2000-4556. AIAA Guidance, Navigation and Control Conference Proceedings, 2000.

[18] Aranda, J., Muñoz, R., Dormido-Canto, S., Díaz, J.M., Dormido-Bencomo, S. An Analysis of Models Identification Methods For High Speed Crafts. *Journal of Maritime Research*. 2005

[19] Arianespace. Aestus engine specifications at Arianespace's Internet site: http://www.arianespace.com/index/index.asp. Last retrieved: 2011.

[20] Bate, Roger R., Mueller, Donald D., White, Jerrry E. Fundamentals of Astrodynamics. Dover Publications, 1971.

[21] Battin, R. H. An Introduction to the Mathematics and Methods of Astrodynamics. Revised Edition. *AIAA Educational Series*, 1999.

[22] Becerra, V.M., Myatt, D.R., Nasuto, S.J., Bishop, J.M., Izzo, D.. An efficient pruning technique for the global optimisation of multiple gravity assist trajectories. In Proceedings of Global Optimisation. 2005.

[23] Bengoa, G.; Alonso, F.; García, D.; Graziano, M.; Beech, T.; Ortega, G.. FAMOS-V2: Formation Flying and Rendezvous and Docking Tool for Exploration Mission in Circular and Elliptical Orbits. Proceedings from the 2nd International Symposium on Formation Flying Missions and Technologies; September. 2004.

[24] Bodley, C.S.; Flanders, H.A. Tethered satellite program control strategy. Proceedings of the International Conference, Arlington, VA, Sept. 17-19. 1986.

[25] Bryson, A.E., Ho, Y.C.. Applied Optimal Control. 2nd ed. Hemisphere Publishing, Washington D.C. 1975.

[26] Calise A. J., Brandt, N., 2004. Generation of launch vehicle abort trajectories using a hybrid optimization method. *Journal of Guidance, Control, Dynamics*. 27(6) 930-937. 2004.

[27] Callaway, D.W. Coplanar Air Launch with Gravity-turn. Air Force Institute of Technology. 2004.

[28] Coello Coello C. A., Lechuga M. S., 2002. MOPSO: A proposal for multiple objective particle swarm optimization. IEEE Congress on Evolutionary Computation, Honolulu, HI, USA. 2002.

[29] Coello Coello, C.A., Lamont, G., Van Veldhuizen, D. Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd edn. Springer, Heidelberg. 2007.

[30] Coverstone-Carroll,V., Hartmann, J.W., Mason, W.J. Optimal multi-objective low-thrust spacecraft trajectories. *Computer Methods in Applied Mechanics and Engineering*. 186, 387-402. 2000.

[31] Cox, E. Adaptive fuzzy systems. Spectrum, IEEE, 30:2. pp. 7-31. Feb. 1993.

[32] Cox, E. Fuzzy fundamentals. Spectrum, IEEE, 29:10. pp. 58-61. Oct. 1992.

[33] Dachwald, B.. Optimization of very-low-thrust Trajectories using Evolutionary Neurocontrol. IAC-04-A.6.03, 55th International Astronautical Congress. Vancouver, Canada. 2004.

[34] De la Cruz, J.M., Besada-Portas, E., Torre-Cubillo, L., Andrés-Toro, B., López-Orozco, J.A. "Evolutionary Path Planner for UAVs in Realistic Environments" Proceedings Genetic and Evolutionary Computation Conference (GECCO) July 2008, Atlanta, Georgia, USA, pp. 1477-1484. 2008.

[35] Doruk R.O., Kocaoglan E. Satellite attitude control by integrator back-stepping with internal stabilization. *Aircraft Engineering and Aerospace Technology*, vol. 80(1), pp. 3-10. 2008.

[36] Drinkwater, M. R., Floberghagen, R., Haagmans, R., Muzi D., Popescu, A. GOCE: ESA's First Earth Explorer Core Mission. Space. Sciences Review, 108, 1–2, 419–432. 2002.

[37] Dymola. http://www.dymola.com. Last retrieved: 2011.

[38] Eberhart R.C., Shi Y, Particle swarm optimization: developments, applications and resources. Proc. Congress on Evolutionary Computation 2001, Seoul, Korea. Piscataway, NJ, IEEE Press. 2001.

[39] Eberhart, R. C., Shi. Y. Comparison between Genetic Algorithms and Particle Swarm Optimization. Proceedings of 7th Conference on Evolutionary Programming, 611-619, March 1998.

[40] Elices, T. Introducción a la Dinámica Espacial, 1989.

[41] Engelbrecht A.P., Fundamentals of Computational Swarm Intelligence. John Wiley & Sons, 2005.

[42] ESA-ACT-Informatics. GTOP Database: Global Optimisation Trajectory Problems and Solutions. http://www.esa.int/gsp/ACT/inf/op/globopt/evvejs.htm. Last retrieved: 2011

[43] ESA-ESOC. Focal Point for ESA space debris activities.

[44] Fahr, H. J. Global Energy Transfer from Pick-up Ions to Solar. Wind Protons. *Solar Physics*, 208, 335–344. 2002.

[45] Fahr, H. J., Chahshei I. V. On the thermodynamics of MHD waveheated solar wind protons, *Astron. Astrophys.*, 395, 991–1000. 2002.

[46] Fieldsend, J. E., Singh, S. A Multi-Objective Algorithm based upon Particle Swarm Optimisation, and Efficient Data Structure and Turbulence. Proc. 2002 U.K. Workshop on Computational Intelligence, Birmingham, UK, 2-4/09/2002, 37-44. 2002.

[47] Fonseca, C. M., Fleming, P.J. Multiobjective optimization and multiple constraints handling with evolutionary algorithms-part I: a unified formulation. IEEE Transactions on Systems, Man and Cybernetics, Part A, 28 (1), 1998.

[48] Gang C., W. Zi-Ming, X. Min, C. Si-Lu, 13th International Space Planes and Hypersonic Systems and Technologies AIAA, 3269. 2005.

[49] Garzón Florez, C.M. and A. Florez. Túneles gravitacionales. *Revista Colombiana de Física*, vol. 38, 3, 1186-1189. 2006.

[50] Gaylor, D.E.. Analysis of Low Thrust Orbit Transfers Using the Lagrange Planetary Equations, Lightsey Research Group at the University of Texas Center for Space Research, Master Thesis, http://gps.csr.utexas.edu/gaylor_pub/cm1project.pdf. 2000. Last retrieved: 2011.

[51] Glenn Research Center. Earth Atmosphere Model by NASA, http://exploration.grc.nasa.gov/education/rocket/atmosmet.html. Last retrieved: 2011.

[52] Goldberg, D.E. Genetic algorithms in search, optimization and machine learning. Adisson-Wesley, MA. 1989.

[53] Gooding, R.H. A procedure for the solution of Lambert's orbital boundary-value problem. *Celestial Mechanics and Dynamical Astronomy*, 48:145–165, 1990.

[54] Gurfil, P. and Kasdin, N.J. Dynamics and Control of Spacecraft Formation Flying in Three-Body Trajectories, AIAA Guidance, Navigation, and Control Conference and Exhibit, Montreal, Canada, Aug. 6-9. AIAA Paper 2001-4026. 2001.

[55] Hamilton, N. Formation Flying Satellite Control Around the L2 Sun-Earth Libration Point. Master's thesis, George Washington University, Washington D.C. (available at Storming Media, www.stormingmedia.us). 2001. Last retrieved: 2011.

[56] Hartmann, J. W., Coverstone-Carroll V. L., Williams, S. N. Optimal Interplanetary Spacecraft Trajectories via a Pareto Genetic Algorithm. *Journal of the Astronautical Sciences*, 46(3), 267-282, AAS 98-202. 1998.

[57] Hechler, M. Launch Windows for libration point missions. *Acta Astronautica*, 64, 139–151. 2009.

[58] Herschel-Planck. http://www.esa.int/esaMI/Herschel/SEMZETEH1TF_0.html. 2010. Last retrieved: 2011.

[59] Howell, K.C. and Marchand, B. G. Design and Control of Formations near the Libration Points of the Sun-Earth/Moon Ephemeris System, Purdue University. 2003b.

[60] Howell, K.C. and Marchand, B.G. Control Strategies for Formation Flight in the Vicinity of the Libration Points, AIAA/AAS Space Flight Mechanics Conference, Ponce, Puerto Rico, Feb. 9-13, AAS Paper 03-113. 2003a.
http://www.esa.int/SPECIALS/ESOC/SEMU2CW4QWD_0.html. Last retrieved: 2011.

[61] Hu, X., Eberhart, R. C., 2002. Multiobjective Optimization using Dynamic Neighborhood Particle Swarm Optimization. Congress on Evolutionary Computation CEC 2002, 1677-1681, May 2002.

[62] Izzo D. Lambert's Problem for Exponential Sinusoids. *Journal of Guidance Control and Dynamics*, 29(5), 1242-1245. 2006.

[63] Izzo, D. Lambert's Problem for Exponential Sinusoids. ACT7 ESTEC Internal Report ACT-RPT-4100-DI-LMSP01. April 2005.

[64] Izzo, D., 2009. ESA Advanced Concepts team. http://www.esa.int/gsp/ACT/inf/op/globopt.htm. Last retrieved: 2011.

[65] Izzo, D., Becerra, V.M., Myatt, D.R., Nasuto, S.J., Bishop, J.M. Search space pruning and global optimisation of multiple gravity assist spacecraft trajectories. *Journal of Global Optimisation*, 38(2), 283-296. 2006.

[66] Jang, J.-S. R. and Sun, C.T. Functional Equivalence Between Radial Basis. Function Networks And Fuzzy Inference Systems, IEEE Trans. on Neural Networks, 4(1), 156-159, 1993.

[67] Jantzen J. Tuning Of Fuzzy PID Controllers, Technical University of Denmark. H 871, http://www.iau.dtu.dk/~jj/pubs/fpid.pdf. 1998. Last retrieved: 2011.

[68] Kaula W. M. Theory of Satellite Geodesy, Blaisdell Publishing Company. Waltham, Massachusetts. 1996.

[69] Kennedy J., Eberhart R.C. Particle Swarm Optimization. Proceedings of the 1995 IEEE International Conference on Neural Networks, 1942-1948, New Jersey 1995.

[70] Lancaster, E.R. and Blanchard, R.C. A unified form of Lambert's theorem. NASA technical note TN D-5368, 1969.

[71] Li, X. Better Spread and Convergence: Particle Swarm Multiobjective Optimization Using the Maximin Fitness Function, GECCO 2004, *LNCS* 3102, 117-128. 2004.

[72] Marín, F.J., García, F., Sandoval, F. Algoritmos Genéticos: una estrategia para la búsqueda y la optimización. *Informática y Automática*, vol. 25, n. 3,4. 1992.

[73] Marscher, W. Universal conic time of flight computations, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1964.

[74] Martín, J.A., González, C., García, R., De Pedro, T. Fuzzy expressions processor with modifiers and edges, conceptual design and testing. 4 th EUSFLAT 2005 Barcelona, Septiembre 2005.

[75] Martín, J.A., González, C., García, R., De Pedro, T. Fuzzy Modeling for Coal Seams. A case study for a Hard-Coal Mine. Lecture Notes on Computer Science. Elsevier Vol. 3643. 2005.

[76] Mattsson, S. E., Elmqvist H., Otter M. Physical system modelling with Modelica. *Control Engineering Practice*, 6(4), 501–510. 1998.

[77] McConaghy, T.T., Debban, T.J., Petropoulos A.E., Longuski, J.M. Design and Optimization of Low-Thrust Trajectories with Gravity Assists, *Journal of Spacecraft and Rockets*, 40(3), 380-390. 2003.

[78] Ming, X.; Dancer, M.W.; Balaskrishnan S.N.; Pernicka, H.J. Stationkeeping of an L2 libration point satellite with θ-D technique. Proceedings of the 2004 American Control Conference ACC (June 30-July 2, Boston Massachusetts). 2004.

[79] Modelica. http://www.modelica.org. Last retrieved: 2011.

[80] Moreno V., Aeyels D. Minimización del Tiempo de Vuelo de Satélites Utilizando Programación Dinámica. *RIAI*, 3(3), 108–115. 2006.

[81] Myatt D.R., Becerra, V.M., Nasuto, S.J., Bishop, J.M., 2003. Advanced Global Optimisation for Mission Analysis and Design. ESA Ariadna final report 03/4101, www.esa.int/gsp/ACT. 2003. Last retrieved: 2011.

[82] Nagarajan, R. Paulraj, M.P. Yaacob, S. Zain, Z.M. Soh Kay Hoh, W. Arshad, A.S. PID and Adaptive Predictive Fuzzy Logic Controller for a micro-satellite. International Conference on Electronic Design. 2008.

[83] NASA Orbital Debris Office http://orbitaldebris.jsc.nasa.gov. Last retrieved: 2011.

[84] Pamadi, B. J. Simple Guidance Method for Single Stage to Low Earth Orbit. *Journal of Guidance*, Control and Dynamics. 18(6), 1420-1426. 1995.

[85] Parsopoulos K. E., Vrahatis M. N.. Particle swarm optimization method in multiobjective problems, Proceedings of the 2002 ACM Symposium on Applied Computing (SAC 2002), pp. 603–607. 2002.

[86] Passino, K. M., Yurkovich, S. Fuzzy Control, Addison Wesley, http://www.ece.osu.edu/~passino/FCbook.pdf, 1998. Last retrieved: 2011.

[87] Perez, E. (Arianespace). Ariane 5 User's Manual. Issue 4, rev. 0, November (2004).

[88] Pérez, J., Seco, F., Milanés, V., Jiménez, A., Díaz, J., De Pedro, T. An RFID-Based Intelligent Vehicle Speed Controller Using Active Traffic Signals. Sensors Vol. 10, 5872-5887, 2010.

[89] Petropoulos, A.E., Longuski, J.M., Bonglio, E.P. Trajectories to Jupiter via gravity assists from Venus, Earth, and Mars. *Journal of Spacecraft and Rockets*, 37(6). 2000.

[90] Petropoulos,A.E., Longuski J. M.. Shape-Based Algorithm for Automated Design of Low-Thrust, Gravity-Assist Trajectories. *Journal of Spacecraft and Rockets*, Vol. 41, No. 5, September-October 2004.

[91] Rauwolf, G., Coverstone-C. V. Near-optimal low-thrust orbit transfers generated by a genetic algorithm. *Journal of Spacecraft and Rockets*, 33(6) 859-862. 1996.

[92] Reyes-Sierra, M., Coello, C. A. Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art. International *Journal of Computational Intelligence Research*, 2(3), 287-308. 2006.

[93] Sánchez Peña, R. S. Alonso R. J. Control de vehículos espaciales. *RIAI*, 2(3), 6–24. 2005.

[94] Santos, M., López, R., De la Cruz, J.M. Fuzzy control of the vertical acceleration of fast ferries. *Control Engineering Practice*, vol 13, nº3, pp 305-313, Elsevier 2005.

[95] Scheeres, D.J. and Vinh, N.X. Dynamics and control of relative motion in an unstable orbit, AIAA/AAS. Astrodynamics Specialist Conference, Denver, Colorado, Aug. 14-17. AIAA Paper 2000-4135. 2000.

[96] Standish, E. M.. Keplerian Elements for Approximate Positions of the Major Planets. Solar System Dynamics Group. JPL/Caltech. Available at http://ssd.jpl.nasa.gov/planet_pos. 2010. Last retrieved: 2011.

[97] Stuchi, T.J.; Yokoyama, T.; Correa, A.A.; Solórzano, R.H.; Sánchez, D.M.; Winter, S.M.G.; Winter, O.C. Dynamics of a spacecraft and normalization around Lagrangian points in the Neptune–Triton system. *Science Direct, Advances in Space Research* 42, 1715-1722. 2008.

[98] Sutton, P., Biblarz, O., Rocket Propulsion Elements, 7th ed. Wiley-Interscience, USA, 2001.

[99] Tiller, M. M. Introduction to physical modeling with Modelica. Kluwer Academic Publishers, Springer. 2001.

[100] U.S. Government. U.S. Standard Atmosphere 1976. U.S. Government Printing Office, Washington, D.C. 1976.

[101] Vancouver Film School. http://www.youtube.com/watch?v=vrlIB1rzlZs. Last retrieved: 2011.

[102] Vasile M., DePascale, P.. Preliminary design of low thrust multiple gravity assist trajectories. *AIAA, Journal of Spacecraft and Rockets*, 43(5):1065- 1076. 2006.

[103] Vasile, M., Schütze, O., Junge, O., Radice, G., Dellnitz, M. Spiral trajectories in global optimisation of interplanetary and orbital transfers. Ariadna Study Report AO4919 05/4106, ESA. 2006.

[104] Vasile, M., Summerer, L., DePascale, P. Design of Earth-Mars transfer trajectories using evolutionary branching technique. *Acta Astronautica*, 56(8), 705-720. 2005.

[105] Vulcain engine specifications at ESA's Internet site (last retrieved: December 2009): http://www.esa.int/SPECIALS/Launchers_Access_to_Space/ASELVQI4HNC_0.html. Last retrieved: 2011.

[106] Walter, A. An Initial Guess Generator for Launch and Reentry Vehicle Trajectory Optimization. Ph.D, Institut für Flugmechanik und Flugregelung der Universität Stuttgart, Germany (2001) http://elib.uni-stuttgart.de/opus/volltexte/2001/919/pdf/Markl_Al.pdf. Last retrieved: 2011.

[107] Ward, Thomas A. Aerospace Propulsion Systems. John Wiley & Sons (Asia), 2010.

[108] Zitzler, E., Laumanns, M., Thiele, L., Fonseca, C.M., Grunert da Fonseca, V. Performance assessment of multiobjective optimizers: an analysis and review, IEEE Transactions on Evolutionary Computation, 7(2), 117- 132. 2003.

[109] Zitzler, E., Thiele, L., Deb. K., Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Evolutionary Computation 8(2): 173-195. 2000.