Politecnico di Milano
Facoltà di Ingegneria Industriale

Corso di Laurea Specialistica in

Ingegneria Spaziale

Global optimization of multiple gravity assist trajectories: development of STA Interplanetary Module v3.0

Relatrice:
Prof. Michèle Lavagna

Autore:
Michele Scotti
matr. N. 734144

Anno accademico 2012/2013

Contents

List of Figures

List of Tables

Nomenclature

C_3	Residual specific energy at infinite distance from the attractor.
DSM	Deep Space Maneuver
ESA	European Space Agency
GA	Gravity assist
GAM	Gravity Assist Maneuver
IM	Interplanetary Module
MGA	Multiple Gravity Assist
SOI	Sphere of influence
STA	Space Trajectory Analysis
TOF	Time of flight

Chapter 1

Introduction

This chapter gives an historical perspective to the design of interplanetary trajectories, presenting the state of art regarding this problem.

The necessity of interplanetary trajectories is of course strictly related to the desire to explore the Solar System. The ever-present human curiosity for the sky led to the development of sophisticated observation devices, and since Galileo in 17th century, our knowledge of the space around Earth increased thanks to telescopes.

In 1903, Konstantin Tsiolkovsky was the first one to propose a way to travel in space to watch from a close distance our neighbors, by the means of rocket motors. Little more than fifty years had to pass before the first Earth orbiter, the Sputnik 1, was launched in 1959. During the second half of the 20th century, the development of the space industry has been essentially propelled by political reasons: soon after the Sputnik, the Moon and Venus were the target of a number of spacecrafts, notably Luna 1 (1959) which performed the first flyby of the Moon, and the first successful Venus encounter from Mariner 2 (1962). Three years later, Mariner 4 achieved the same result on Mars, approaching the planet during July 1965.

During the late sixties, while the United States managed to make a success of the program Apollo, culminating in the landing on the Moon in 1969, the interest in Venus and Mars did not fade: Venera 5, 6 and 7 have been launched in 1969-70 by the Soviet Union, whereas USA sent Mariner 6 and 7 towards Mars. The other American exploration program, Pioneer, was aimed at Jupiter and Saturn, which were visited in 1972 and 1973 respectively. Further progress was obtained during the 70s, with Mariner 10 meeting Venus and Mercury in 1974 (three flybys of Mercury), Viking 1 and 2 landing on Mars (1975), and the outstanding missions Voyager 1 and 2 returning invaluable information about

Jupiter, Saturn, Uranus and Neptune.

While the USSR kept focusing on Venus (Venera 11-16 and Vega 1-2), the 80s saw the Japanese and European spacecrafts addressed to the Halley comet (actually encountered also by Vega 1 and 2); in 1989 the United States finally launched Galileo, which has been delayed because of the Space Shuttle Challenger disaster. The mission, featuring a Jupiter orbiter and an atmospheric probe equipped with a heavy heat shield, had to perform several gravity assists to reach its destination.

Even if the political drive was reduced after the end of the Cold War, a number of exploration missions were launched during the next decade: among them, NEAR Shoemaker (rendezvous and landing on the near-Earth asteroid Eros), Mars Surveyor and Pathfinder (with a planetary rover on the latter) and Cassini-Huygens (Saturn orbiter with lander on Titan) certainly deserve a mention for their impressive achievements.

In the new millennium Japan managed to bring back samples from small Itokawa asteroid; the Mars Exploration Rovers operated on the red planet; the Rosetta spacecraft was launched in 2004 to eventually meet the comet Churyumov–Gerasimenko. Many scientific spacecrafts are now operational and still providing useful data, and more of them are constantly designed for future implementation.

One of the few things all of those endeavors have in common is the huge amount of energy necessary to achieve the desired trajectory. If traditional high thrust propulsion is used, the required propellant mass fraction reduces substantially the room for the payload. A very common technique to obtain significant energy variations, using little to none propellant, is to exchange momentum with massive bodies, so that great variations of the spacecraft inertial velocity can be achieved almost for free. This technique is known as *Gravity Assist Maneuver*, or *Gravitational Slingshot*, and its efficiency (regarding the mission goals) can be improved either with help from the autonomous propulsion system (*Powered Gravity Assist*, PGA) or exploiting atmospheric interaction (*Aerogravity Assist*, AGA). Furthermore, this approach allows to visit several celestial bodies in a single mission, increasing considerably the expected scientific return.

This is the rationale for this work: in order to find the cheapest solutions for the interplanetary transfer problem, a tool for the design of Multiple Gravity Assist (MGA) trajectories has been developed.

1.1 Multiple Gravity Assist trajectory design: state of the art

The opportunity offered by planetary gravity assists has been acknowledged even before the beginning of the actual exploration (from Tsiolkovsky); however the design of such trajectories presents great challenges, because the relation between the parameters of the trajectory (such as launch date, planetary encounter dates, closest approach distances, correction maneuvers and many others) and the objectives of the mission is often very complicated, so that it is quite difficult to identify the optimal trajectory. Furthermore, the compliance to strict constraints (most notably regarding the required autonomous propulsion and the total duration of the transfer) is usually instrumental to the success of the mission.

During the early days of the Solar System exploration, many interplanetary missions have been designed and implemented; the trajectory design process often relied on crude approximations, empiric reasoning and verification tools. One of the most famous simplified models is commonly associated with the Tisserand's plane: by neglecting the eccentricity and the inclination of the planets heliocentric orbits, and ignoring the actual phasing of the planetary encounter, the problem is greatly simplified, reducing to two the number of parameters necessary to identify an interplanetary transfer leg (pericenter radius and period) [1]. This allows to examine a large interval of feasible solutions, in order to identify a reduced set of suitable solutions to be used as a starting point for subsequent local refinement. Since the cost function multimodality for such problems increase rapidly with the number of required GAM, it is clear that the possibility to find the global minimum is bound to the ability to start with at least an initial guess located inside its basin of attraction.

The analysis of Tisserand's plane can be very effective, but it is not well suited for automated search: thus the necessity for a global search analysis tool. During the last decades, many global algorithms were implemented; they can be classified as:

- ♢ deterministic methods: they are based on the exhaustive research of the solution space; one of the most famous in this category is STOUR [2], originally developed in 1983 and subsequently extended to investigate the gravity assist possibilities to reach Jupiter and Pluto [3]; other examples of this class of methods are DIRECT ([4]), MCS ([5]) and more recently GASP ([6]).

♢ Monte-Carlo methods: they are characterized by random generation of parameters, according to some probability distribution function. For the Cross Entropy algorithm [7], Gaussian distributions are used: their mean values and their variances are updated each iteration to follow the best individuals. The method PGSL, presented in [8], represents instead the probability distributions as an histograms.

♢ Evolutionary algorithms: an initial population of solutions is generated by sampling (often with uniform probability distribution) the search space. This population is then modified by recombination of the best individuals (cross-over) and/or the local variation of a sub-set of the solutions (mutation). Some examples for this class are GAOT [9], GATBX [10] and FEP [11]. Swarm intelligence methods are usually labeled as evolutionary algorithms as well, even if the improvement of the population happens by "movement" instead of by evolution: the velocity of each particle is evaluated every iteration, combining the attraction toward the present best individual (swarm attractor) and the best position retained in the past by each particle (particle attractor). The classic formulation from Kennedy [12] is inherently suited for scalar optimization, whereas multiple swarms are more appropriate for multi-objective problems. The first (and probably most famous) example of swarm intelligence algorithms is PSO [12].

An extensive analysis of the effectiveness of such methods for space trajectory optimization problems has been carried out in two remarkable works, developed as a project of ESA's Ariadna initiative ([13], [14]). Several algorithms have been tested on a number of significant test cases (planet to planet, multiple gravity assist, low thrust and weak stability boundary), and their results have been compared.

In order to better understand which tools are more suitable for the MGA problem, it is always useful to test different optimization techniques. For this reason, a global search tool for MGA trajectories has been developed as part of STA, an astrodynamics suite (see §2.1). The performances of two methods have been analyzed: DG-MOPSO and NSGA-II, both falling under the category of evolutionary algorithms. The main reason for this choice is the availability of these methods under said software suite (STA), thanks to the work of Francesco Castellini [15]; besides, in [14] it is concluded that a fairly basic MOPSO algorithm (as described in [16]) performs very well for MGA optimization. Hence, it is appealing to assess the performance of the optimizer developed by Francesco Castellini (which is a modified version of the Coello formulation [17]) on a MGA problem. For what concerns NSGA-II, it has been integrated into the global op-

timization tool GOSpel (see [18]), and its effectiveness makes it appealing for this study.

1.2 Document structure

The subsequent part of this document is organized as follows:

- ♢ chapter 2 describes how the new tool has been characterized, deriving its main features from the analysis of the requirements;
- ♢ in chapter 3, the formulation of the objective function is detailed, along with the testing of the most critical parts;
- ♢ the last chapter contains the performance analysis of new tool for a selected number of study cases, and the overall conclusions.

Chapter 2

STA Interplanetary Module

The new design tool has been conceived inside the framework of **Space Trajectory Analysis** (**STA**), a composite software suite which has been initially developed as an open source program under the aegis of the European Space Agency (ESA), and is currently reserved for internal use. The multi-gravity-assist design tool has been labeled **Interplanetary Module** (**IM**), and this work is aimed to describe the structure, the functioning and the results of the said module.

2.1 STA: brief description of the software suite

STA began at the European Space Agency in 2005 as an internal activity to develop an open source software suite, involving the universities and the research institutions of many countries. Its purpose is to support the analysis phase of a space mission, providing the tools necessary to estimate the relevant information about a wide range of trajectories (orbits around the planets of the Solar System, atmospheric entry and descent paths, rendezvous and others).

The functionalities of STA are organized into modules, which perform different tasks; however, they share the graphic environment (**VESTA**, an open-source graphical 3D engine for animation of space scenarios and visualization of mission analysis results) and a common core of astrodynamic functions (mostly for dates and coordinates management/conversions, orbit propagation and celestial bodies ephemerides).

The software language is **C++**, and the development framework is **Qt**: this allows a multi-platform deployment of the software. STA computation capabilities are based on **Eigen**, an high-level C++ library of template headers for linear algebra, matrix and vector operations, numerical solvers and related algorithms.

2.2 The Interplanetary Module

The Interplanetary Module is a tool for preliminary design of an interplanetary trajectory. It is aimed to high thrust propulsion missions, namely those where the maneuvering capabilities of the spacecraft are ensured by a propulsion system able to provide thrust levels above 100 N (but usually higher) for a short amount of time (impulsive maneuvers). The IM allows the end user to provide a quick solution for an interplanetary transfer problem, alongside with the assessment of the costs, in terms of ΔV and time of flight (TOF).

The work here described refers to the third version of the Interplanetary Module: the first version (developed by G. Schouten in 2006 [19]) was actually just a cost evaluation tool, where the user had to provide the planetary sequence, the launch/encounter dates and the pericenter radius for each passage. The second version (developed by B. Naessens in 2007 [20]) was much more complex, comprising features such as pork chop analysis, numerical simulation and flyby optimization; however it was never integrated into the software suite (being basically a stand-alone tool), and does not make use of the Eigen library features.

2.3 Requirements

The requirements for the IM have been defined in accordance with the STA Project Plan [21] and the IM Task Description Form [22]. Following the directives from the appropriate ECSS standard (Space engineering - Software, [23]), a detailed System Requirements Document has been issued [24]. The following paragraphs summarize the requirements breakdown, separating the top level drivers from the corresponding derived requirements.

2.3.1 Top level requirements

The top level requirements define the main drivers for the software development, describing at high level the problem to be addressed.

Tag	**STA-IM-F-0000: high thrust interplanetary trajectory optimization**
Description	The IM shall be able to assist the user in the process of design and optimization of an interplanetary trajectory.
Derived from	Project Plan of STA 4.0 “Ordovician” [21]

Tag	**STA-IM-F-1000: high thrust interplanetary trajectory visualization**
Description	The IM must provide a way to visualize the optimal trajectory.
Derived from	Task Description Form Interplanetary "Ordovician" [22]

Tag	**STA-IM-I-000: input**
Description	The module shall be able to run an optimization on the basis of input data derived from the dedicated GUI.
Derived from	Task Description Form Interplanetary "Ordovician" [22]

2.3.2 Functional requirements

Tag	**STA-IM-F-0100: Quick Optimizer**
Description	The IM shall allow the user to run a quick optimization of an interplanetary transfer problem.
Derived from	High thrust interplanetary trajectory optimization (STA-IM-F-0000)

Tag	**STA-IM-F-0110: Objective Function for Quick Optimizer**
Description	The Quick Optimizer shall be able to evaluate the cost of an interplanetary transfer (in term of ΔV and TOF) given a sequence of planets (including departing and arrival planets) and the TOF between each planet.
Derived from	Quick Optimizer (STA-IM-F-0100)

Tag	**STA-IM-F-0111: planets ephemerides for Quick Optimizer**
Description	The Quick Optimizer must able to retrieve the state vector of a given planet at any specified time between January 1st 2000 and December 31th 2200.
Derived from	Objective Function for Quick Optimizer (STA-IM-F-0110)

Tag	**STA-IM-F-0112: Lambert Solver for Quick Optimizer**
Description	The Quick Optimizer must implement a Lambert solver, i.e. a routine which takes as input the TOF, initial and final position (expressed as 3D position vectors); the output shall be the shortest trajectory that meet the requirements.
Derived from	Objective Function for Quick Optimizer (STA-IM-F-0110)

Tag	**STA-IM-F-0113: ΔV computation for Quick Optimizer**
Description	Given a sequence of Lambert arcs, the Quick Optimizer shall be able to compute the total ΔV to be provided by propulsion systems (including the launcher ΔV).
Derived from	Objective Function for Quick Optimizer (STA-IM-F-0110)

Tag	**STA-IM-F-0114: TOF computation for Quick Optimizer**
Description	The Quick Optimizer shall be able to compute the total TOF.
Derived from	Objective Function for Quick Optimizer (STA-IM-F-0110)

Tag	**STA-IM-F-0120: Optimization for Quick Optimizer**
Description	The Quick Optimizer shall be able to find the Pareto front of the cost function (as defined in STA-IM-F-0110).
Derived from	Quick Optimizer (STA-IM-F-0100)

Tag	**STA-IM-F-0121: Global Optimizer for Quick Optimizer**
Description	The Global Optimizer shall be able to explore a broad search space, in order to quickly find non-dominated solutions to the problem.
Derived from	Optimization for Quick Optimizer (STA-IM-F-0120)

Tag	**STA-IM-F-0122: Local Optimizer for Quick Optimizer**
Description	The Local Optimizer shall be able to refine the solutions which have been computed by the Global Optimizer, with a numerical targeting procedure.
Derived from	Optimization for Quick Optimizer (STA-IM-F-0120)

Tag	**STA-IM-F-0200: Custom Optimizer**
Description	The IM shall allow the user to customize the features of the optimization, such as choosing whether the planet sequence is free or constrained, the possibility of including multi-revolution Lambert arcs and DSM.
Derived from	High thrust interplanetary trajectory optimization (STA-IM-F-0000)

Tag	**STA-IM-F-1100: trajectory visualization**
Description	The IM must be able to show the trajectory into STA graphic environment.
Derived from	High thrust interplanetary trajectory visualization (STA-IM-F-1000)

Tag	**STA-IM-F-1200: trajectory exporting**
Description	The IM must be able to produce a detailed report of the optimal trajectory, using a standard message format if possible.
Derived from	High thrust interplanetary trajectory visualization (STA-IM-F-1000)

2.3.3 Interface requirements

Tag	**STA-IM-I-010: departure date boundaries selection**
Description	The user must be able to select the boundaries for the departure date.
Derived from	Input (STA-IM-I-000)

Tag	**STA-IM-I-020: planetary sequence selection**
Description	The user must be able to select the planetary sequence, i.e. the departure planet, the gravity assist planets and the destination planet.
Derived from	Input (STA-IM-I-000)

Tag	**STA-IM-I-030: orbit closure selection**
Description	The user must be able to choose whether or not the optimization process must take in account the closure ΔV around the destination planet.
Derived from	Input (STA-IM-I-000)

Tag	**STA-IM-I-040: scalar figure of merit**
Description	The user must be able to choose a scalar figure of merit in order to single out one "optimal" trajectory.
Derived from	Input (STA-IM-I-000)

2.4 Program structure

From the analysis of the requirements, the structure of the module is derived. Two complexity levels should be implemented (Quick Optimizer and Custom Optimizer): it has been decided to focus this work on the simpler formulation, meaning that the only three classes of supported thrusting maneuvers are:

- ♢ launch phase from the departing planet, usually provided by an expendable launch vehicle;
- ♢ connection impulses at the pericenter of each gravity assist hyperbola (see §3.3.3);
- ♢ orbital insertion at the arrival planet (if required).

As a consequence, no correction burns are allowed during the coasting phases: they are usually referred as Deep Space Maneuvers (DSM), and their inclusion into the problem formulation would increase the number of parameters for a fixed number of gravity assists (thus boosting the multimodality of the objective function). The development of the Custom Optimizer is postponed to future implementations. As clarified in chapter 3, by excluding the possibility of DSM, the solution parametrization reduces to

$$\boldsymbol{x} = [t_{\text{dep}}, \text{TOF}_{\text{arc}\,1}, \text{TOF}_{\text{arc}\,2}, ..., \text{TOF}_{\text{arc}\,N+1}]$$

i.e. the departing date (t_{dep}) and the time duration of each interplanetary arc ($\text{TOF}_{\text{arc}\,i}$); N is the number of required Gravity Assist Maneuvers.

The planetary sequence is inserted into the optimization problem as an input from the user. There are two main reasons for this choice: first of all, the difficulty in handling a mixed set of discrete and continuous variables is not easily overcame by most optimization algorithms; in addition, the fly-by of specified celestial bodies could be a mission requirement, and this program structure allows the compliance to this kind of condition.

It is inferred from STA-IM-F-0110 (Objective Function for Quick Optimizer) that multi-objective optimization is required; the minimization goals are the total ΔV (defined as the total velocity change to be provided by the autonomous propulsion system) and the overall duration of the interplanetary transfer $\mathrm{TOF}_{\mathrm{tot}}$. The evaluation of "the best" solution is subsequently addressed, according to a scalar figure of merit.

The functioning diagram of a program adopting said formulation is illustrated in figure 2.1, and it reflects the structure of the Interplanetary Module.

The input block functions are carried out thanks to the Interplanetary Dialog, which is represented in figure 2.2. The output is delivered as a group of files containing detailed data about the evolution of the population, the final Pareto front, and the .OEM ephemeris of the optimal trajectory (according to the scalar figure of merit selected from the GUI). The latter complies to the standard CCSDS for the Orbit Ephemeris Message (OEM), and the other files are easily loaded into any work environment for post-processing purposes. Every other block will be described in the following chapters.

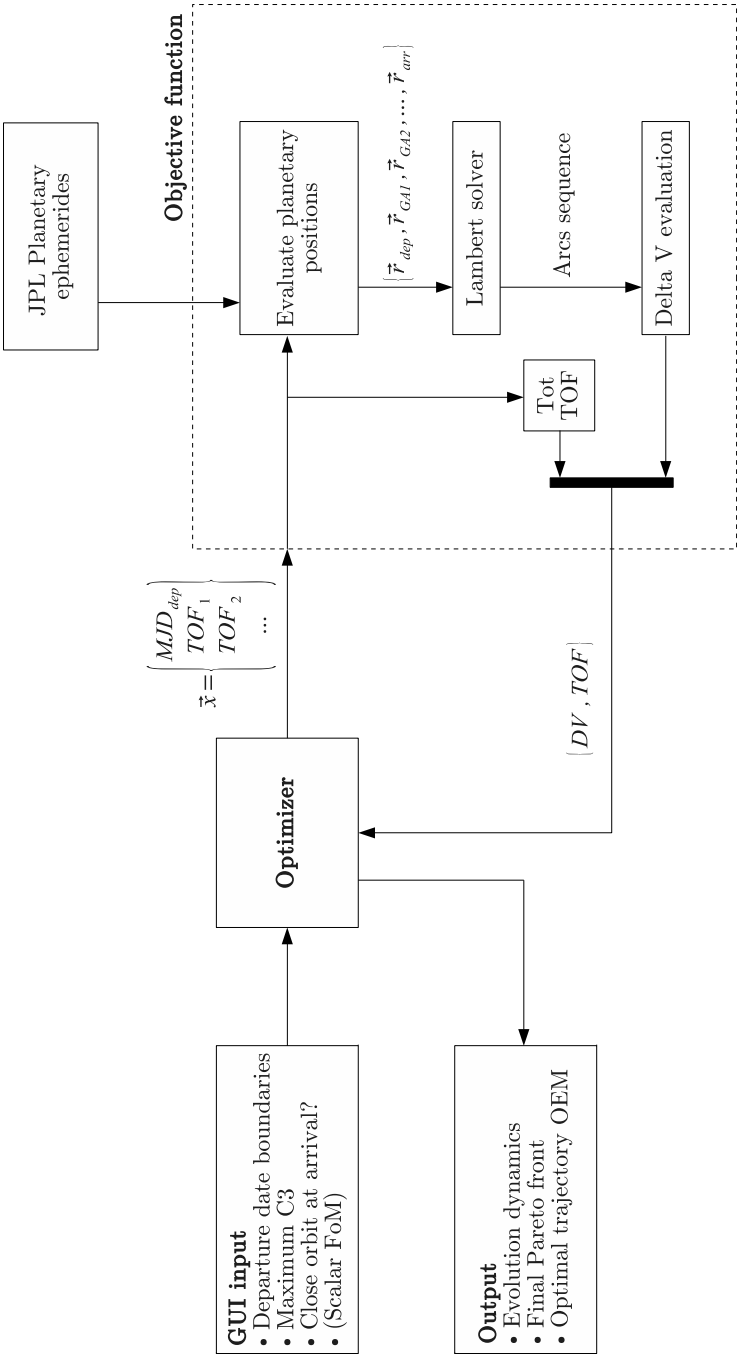

Figure 2.1: Workflow diagram of the Interplanetary Module

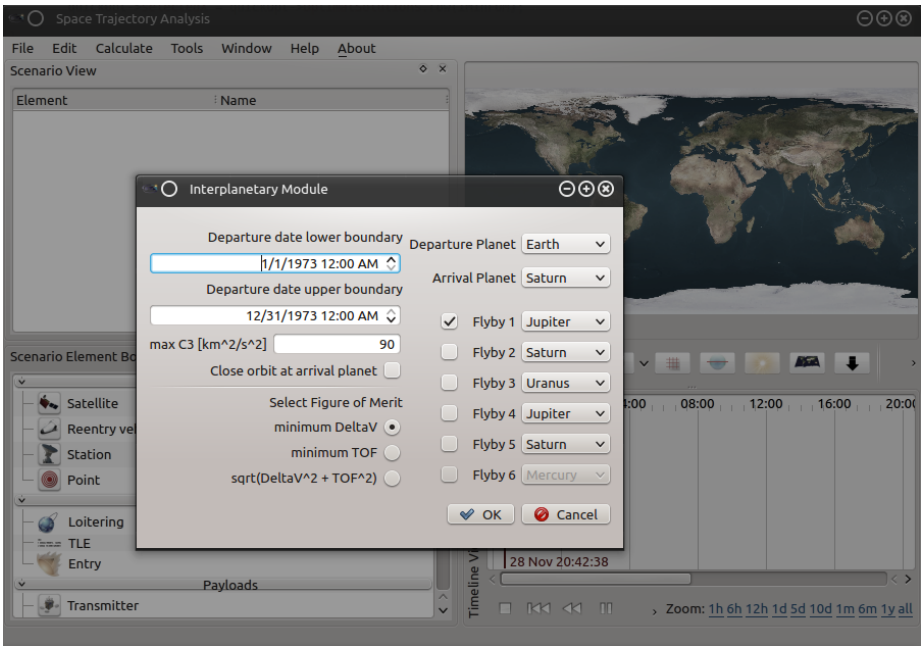

Figure 2.2: Interplanetary module input dialog

Chapter 3

Problem formulation

This chapter deals with the cost evaluation of an interplanetary trajectory: this feature importance is paramount, since it enables the selection of one optimal solution among the virtually infinite set of feasible trajectories. However, the cost computations procedure is not unique, and it is strongly dependent on the mathematical model adopted to describe the trajectory.

3.1 Coordinate systems

The determination of a gravity-assisted interplanetary trajectory requires different coordinate systems. Most of the trajectory is made by heliocentric arcs, thus the primary concern is to define a frame centered on the Sun; on the other hand, during any planetary fly-by it is convenient to express the state vector of the spacecraft into a planetocentric reference frame. No planetary-fixed frames will be used (meaning any frame fixed with a reference on the planet), since any celestial body is always assumed spherical, and no superficial feature is deemed as important during early design phases.

In order to keep as simple as possible the mathematical modeling, all the coordinate systems are rectangular and right handed, all sharing the same reference plane and preferred direction (Earth mean equator and vernal equinox at the standard epoch J2000); the only difference is the center of the reference frame, which corresponds with the current main attractor. From now on, when a coordinate system is referred to, only the center will be specified.

The above-mentioned coordinate systems all represent a reasonable approximation for an inertial reference frame, and will be treated as such.

3.1.1 Nomenclature

Heliocentric vectors will be represented with a capital letter, whereas planetocentric vectors will be denoted by lower case letters. Referring to planetocentric vectors, no ambiguity is present, since the obvious subject is always the spacecraft (for instance, $\boldsymbol{v}$ is the velocity of the spacecraft with respect to the planet). If the reference frame is heliocentric, it is necessary to distinguish when a vector refers to a planet or the spacecraft: this will be accomplished by placing a superscript, so that $\boldsymbol{R}^s$ is the heliocentric position of the spacecraft and $\boldsymbol{V}^p$ is the heliocentric velocity of a generic planet. If a specific planet must be labeled, the corresponding astronomical symbol will be adopted: for instance, the heliocentric position of Venus is $\boldsymbol{R}^{\varphi}$.

3.2 Trajectory models

The trajectory of a spacecraft moving into the Solar System can be estimated with different degrees of accuracy. Neglecting both the spacecraft spatial extension (point mass approximation) and its effects over the celestial bodies, the most precise solution would arise from the integration of Newton's second law of motion:

$$m\frac{d^2\boldsymbol{x}}{dt^2} = \boldsymbol{F}_{\text{tot}} \tag{3.1}$$

where $\boldsymbol{x}$ is the position of the point mass, and $\boldsymbol{F}_{\text{tot}}$ is the vector sum of all forces acting on the point mass, both expressed in an inertial reference frame. The components of $\boldsymbol{F}_{\text{tot}}$ originate from gravitational sources and electromagnetic sources (i.e. solar radiation pressure); they are difficult to express as functions of time, preventing the possibility to obtain a closed-form solution for equation 3.1. If all forces must be taken in account (high precision applications) some numeric integration technique must be used.

Of course the numeric approach is not suitable for orbital design purposes, since the evaluation of the ΔV and the time of flight associated with a single trajectory requires a substantial computing effort; this is unacceptable because the optimization process requires the evaluation of several thousands (if not millions) interplanetary trajectory. It is clear that further simplification is needed.

3.2.1 Patched conics model

If all forces save for the main gravitational pull are neglected, the model becomes what is widely known as the *restricted two body problem* ("restricted" refers to the assumption of negligible attraction of the spacecraft on the celestial bodies),

and therefore can be solved in closed-form. When the journey takes place into the Solar System, it can be assumed that the main attractor is always the Sun, save for the regions in close proximity of massive bodies (only planets will be considered in this work).

The surface over which the solar and the planetary attraction are equal is approximately a spheric surface, and the region within is often referred as the planetary *Sphere of Influence* (SOI) . Its center is the same as the planet, and according for instance to Curtis ([25]), its radius can be computed as

$$r_{\mathrm{SOI}} = R\left(\frac{m_p}{m_s}\right)^{2/5} \tag{3.2}$$

where R is the distance between the Sun and the planet; the numeric value for each Solar System planet is listed in table 3.1, along with other relevant physical and astronomical data.

Body	**r** [km]	$\boldsymbol{\mu}$ [km^{3}s^{-2}]	**a** [km]	$\boldsymbol{r_{\mathrm{SOI}}}$ [km]
Sun	6.960×10^5	1.327×10^{11}		
Mercury	2.240×10^3	2.203×10^4	5.791×10^7	1.12×10^5
Venus	6.052×10^3	3.249×10^5	1.082×10^8	6.16×10^5
Earth	6.378×10^3	3.986×10^5	1.496×10^8	9.25×10^5
Mars	3.396×10^3	4.283×10^5	2.279×10^8	5.77×10^5
Jupiter	7.149×10^4	1.267×10^8	7.786×10^8	4.82×10^7
Saturn	6.027×10^4	3.793×10^7	1.433×10^9	5.48×10^7
Uranus	2.556×10^4	5.794×10^6	2.872×10^9	5.18×10^7
Neptune	2.476×10^4	6.835×10^6	4.495×10^9	8.66×10^7

Table 3.1: Relevant physical and astronomical data for the Sun and the planets of the Solar System (source: [25]).

The patched conics model of an interplanetary ballistic flight leads therefore to a composite trajectory; the simplest example could be an Earth-Mars transfer, where the first leg is an hyperbola about the Earth, followed by an heliocentric orbit and finally an hyperbola about Mars. Each SOI crossing corresponds to a smooth connection between two keplerian conics, thus the name "patched conics".

3.2.2 Linked conics model

Further reduction of the model is possible: since the Sun mass is three orders of magnitude greater than the mass of the largest planet (Jupiter), the size of each SOI is negligible with respect to the length of the heliocentric ballistic arcs. If each planetary SOI is assumed to be zero-dimensional, the whole fly-by can

be represented as an impulsive maneuver; as a consequence, the gravity assist becomes a sharp point in the trajectory.

The convenience of this model resides in the possibility to decouple the design of the heliocentric arcs from the analysis of the gravity assist. The velocity variation cost is associated with the impulsive events (departure, gravity assists and arrival), whereas the cost in terms of time of flight depends on the heliocentric arcs.

3.3 Cost evaluation algorithm

The fundamental task of any interplanetary trajectory design tool is the estimation of some cost associated with a specific trajectory, according to the mission requirements. Cost computation can involve multiple factors, for instance the heat flux towards the spacecraft, the distance from the Sun, the necessity to avoid dangerous zones, and many others; however, most of these factors are very specific and their relative importance can vary significantly among different missions.

In order to keep the tool as general as possible, only two figures of merit will be taken in account: the total velocity variation provided by the propulsion systems (including the launch vehicle) and the total time of flight, since virtually any mission will benefit from the minimization of both ΔV and TOF.

Using the linked conics model, the evaluation of the trajectory cost is straightforward; the number of parameters involved is conveniently low, being reduced basically to a sequence of planets (departure, gravity assist(s) and arrival) and the time of flight of each heliocentric arc.

The cost evaluation algorithm can be summarized as follows:

- ♢ collect trajectory parameters (sequence of planets and arcs TOF). From the TOF sequence, each event date (departure, gravity assist(s) and arrival) can be evaluated;
- ♢ evaluate the appropriate planetary state vector at each event (ephemerides interpolation in t_{dep}, t_{ga1}, t_{ga2}, ..., t_{arr});
- ♢ compute $N_{\mathrm{ga}} + 1$ ballistic heliocentric arcs, where N_{ga} is the number of gravity assists to be performed. This action is fulfilled by a Lambert solver, since the boundary positions and the TOF are known for each arc;
- ♢ evaluate the ΔV necessary for each impulsive event (can be null for non-powered gravity assists).

The minimization objectives are thus $\Delta V_{\text{tot}} = \Delta V_{\text{dep}} + \sum_i \Delta V_{\text{ga}\,i} + \Delta V_{\text{arr}}$ and $\text{TOF}_{\text{tot}} = \sum_i \text{TOF}_{\text{arc}\,i}$.

The components of the algorithm are illustrated in the following paragraphs.

3.3.1 Planetary ephemerides

The first task of the validation process concerns the planetary ephemerides. The state vector of each major celestial body at any given time (within the range of applicability) can be computed inside the simulation suite from the file of Chebyshev polynomials coefficients named DE406, made available by Jet Propulsion Laboratory (JPL). These ephemerides result from high precision numeric integration of the N-body problem, taking in account every significant physical effect on all the celestial bodies of the Solar System; the initial condition is a least-squares fit of a large set of observations. The detailed description of the observations and other relevant information can be found in [26].

It is necessary to note that DE406 is a reduced version of the full precision version DE405, which in turn has been outdated by more recent versions (DE421); however, the accuracy level achieved by DE406 is more than enough for the purpose of this work. The implementation of the interpolation has been checked against the results of JPL Horizon on-line service over a one year period (from January 1st to December 31st 2012); the comparison is shown in figure 3.1, and it clearly shows that precision level is completely satisfactory.

3.3.2 Lambert solver

The Lambert solver addresses what is widely known as the Lambert's problem. One possible formulation of the problem states: *given two sorted position vectors inside a central gravitational field and the corresponding flight duration, compute a keplerian trajectory which satisfies said constraints.* In other words, it is the boundary value problem

$$\begin{cases} \dfrac{d^2\boldsymbol{r}}{dt^2} = -\dfrac{\mu}{r^3}\boldsymbol{r} \\ \boldsymbol{r}(t_1) = \boldsymbol{r}_1 \\ \boldsymbol{r}(t_2) = \boldsymbol{r}_2 \end{cases}$$

with $t_1 < t_2$. If the solution is a closed conic, two solution would be feasible, and they are sometimes labeled as *short way* and *long way*. In this work, this ambiguity is resolved by specifying in advance the direction of the angular momentum vector: it is assumed that the z component of $\boldsymbol{h}$ is always positive. This choice seems pretty obvious since otherwise a great amount of energy would be

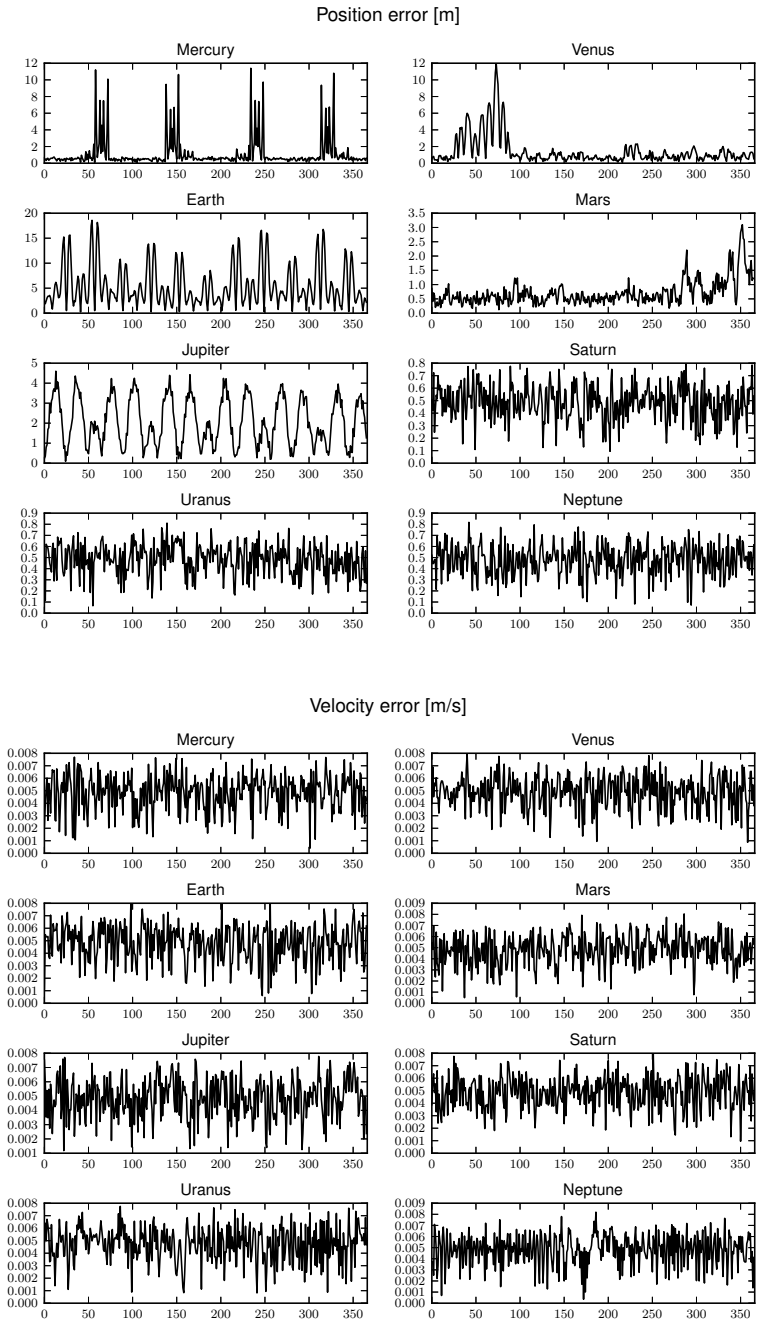

Figure 3.1: Comparison between DE406 planetary ephemerides and Horizon on-line service.

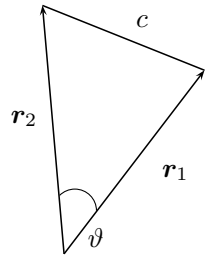

Figure 3.2: Lambert's problem nomenclature.

necessary during the departure phase in order to reverse the angular momentum already associated with the vehicle because of Earth rotation.

The number of solutions can be determined as a function of the non-dimensional time:

$$T = \sqrt{\frac{8\mu}{s^3}}(t_2 - t_1)$$

where the semi-perimeter is $s = \frac{1}{2}(r_1 + r_2 + c)$. If $T \leqslant 2\pi$ there is only one solution (parabolic if $T = 2\pi$), otherwise two new solutions will arise for each 2π increment of T, as summarized in table 3.2. If more than one solution is

T	N
$0 < T \leqslant 2\pi$	1
$2\pi < T < 4\pi$	3
$T = 4\pi$	4
$4\pi < T < 6\pi$	5
$T = 6\pi$	6
...	...

Table 3.2: Number of solutions to the Lambert's problem as a function of the non-dimensional TOF.

available, at least one of them is a multi-revolution elliptic path (meaning that $\vartheta > 2\pi$).

The Lambert problem must be solved iteratively, and a lot of formulations have been proposed over the centuries. At the time of writing, the most popular solvers are from Battin [27] and from Lancaster and Blanchard [28] (improved by Gooding, [29]). Both are based on the transformation of known data into auxiliary variables, so that convergence is assured in a few iterations; the auxiliary

variables are:

$$x = \sqrt{1 - \frac{s}{2a}}$$
$$q = \frac{\sqrt{r_1 r_2}}{s} \cos\frac{\vartheta}{2}$$

Since $q^2 = \frac{s-c}{s}$, q is always in the range $(-1, 1)$; it is also apparent that it does not depend on the TOF, and therefore, once the geometry is defined, it can be considered a fixed parameter (it is not involved into the iteration process). The actual iteration variable is x, and its most useful feature is the universality (meaning it is univocally defined for any conic). The value of x identifies the type of conic:

$$-1 < x < 1 \rightarrow \text{elliptic orbit}$$
$$x = 1 \rightarrow \text{parabolic orbit}$$
$$1 < x < \infty \rightarrow \text{hyperbolic orbit}$$

$x < -1$ is not possible because it would imply $\Delta t < 0$, whereas it is greater than zero by definition. The relation between x and T for some values of q is depicted in figure 3.3; m identifies the number of full revolutions for the associated multi-revolutions solution. Figure 3.3 has been obtained from data generated by the T evaluation routine (i.e. the objective function for the Lambert solver); it matches the corresponding figures found in [28] and [29], so that said routine is considered validated.

Given T, for each value of m (always integer by definition), no more than two solutions exist. In this work only "open paths" will be considered, meaning that m is always set to zero, resulting in a bijective relation between T and x. However, it would be easy to modify the input for the solver, so that m could become an optimization parameter, allowing multi-revolutions paths to be considered during the optimization process.

Essentially, Battin solver and Lancaster/Blanchad/Gooding solver differ for the iteration process: Newton's method with hyper-geometric series expansion in Battin's work, Halley's method for Lancaster and Blanchard (with great convergence improvements thanks to Gooding's starter formulas). Both have been implemented for the interplanetary module, in order to compare their performances and because they have different strengths. The comparison results are shown in the next paragraph.

Once x is found, the terminal velocities can be computed with small computational expense.

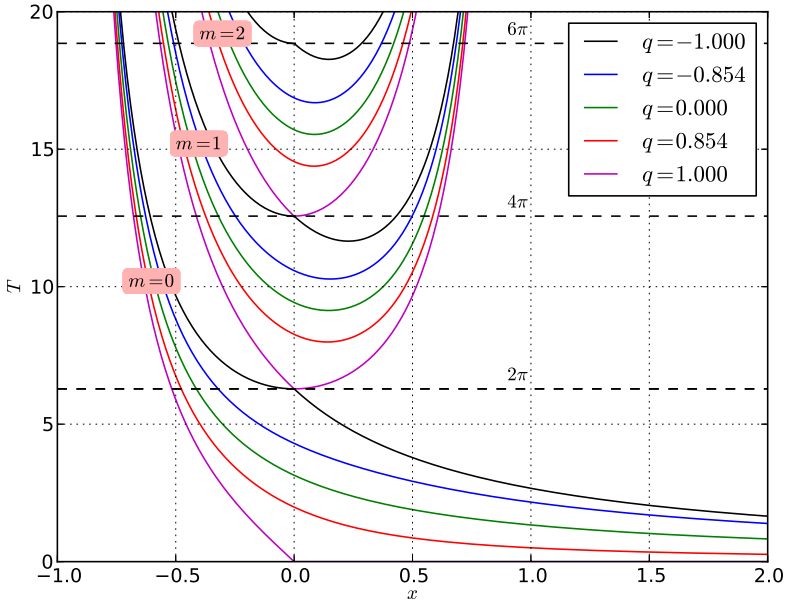

Figure 3.3: Non-dimensional transfer time (T) as a function of the iteration variable x.

Validation

The Lambert solver has been validated with a simple coherence test. Several thousands of input sets ($\boldsymbol{r}_1$, $\boldsymbol{r}_2$ and TOF) were generated starting from the following random parameters (all of them are uniformly distributed over the respective range): The actual solver input requires two position vector and one

Parameter	Range
Starting date	Jan 1st 1950 to Dec 31th 2100
TOF [days]	10 to 4000
Departing planet	Mercury to Neptune
Destination planet	Mercury to Neptune

Table 3.3: Generating parameters set for Lambert testing input.

time interval: $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are evaluated thanks to the planetary ephemerides (the planetary positions are evaluated at the starting date and at starting date + TOF respectively), whereas the TOF do not need any modification.

These input vectors are then fed to both solvers (Battin's and Gooding's), which in turn compute the required arcs, in particular the terminal velocities. At this point, the full state vector of both starting point and terminal points are

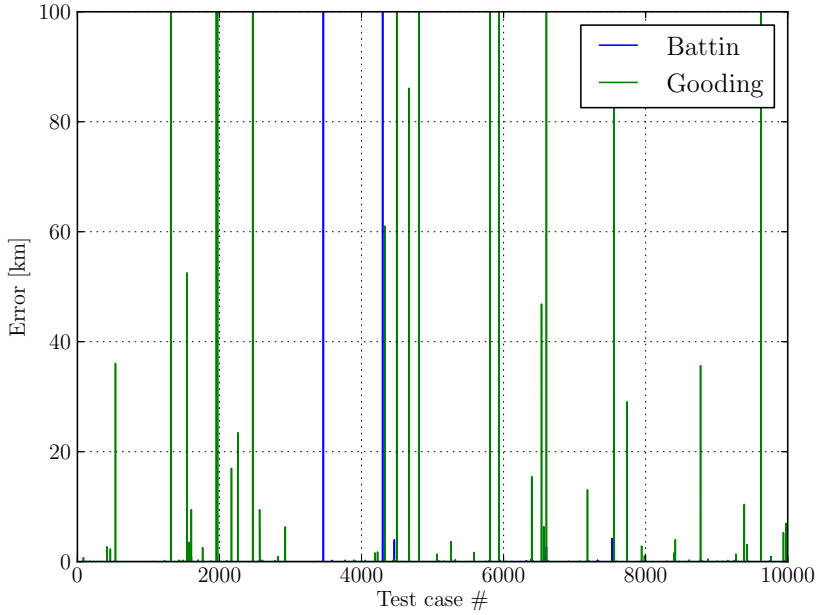

Figure 3.4: Lambert solver error.

known, so it is possible to propagate (the "propagation" is actually analytical, since the reference dynamic model is the Two Bodies model) the initial state vector for the prescribed TOF: the resulting position is eventually compared to $\boldsymbol{r}_2$ (the terminal position for the input set). Of course they should be identical, but the two algorithms actually fail in a number of occasions. The difference between the input $\boldsymbol{r}_2$ and the propagated one is depicted in figure 3.4

Although the error is never large (little more than 1000 km), it is clear that Battin's formulation is much more robust, since it fails in just a few cases, whereas Gooding algorithm is less precise. For this reason, Battin's solver has been constantly used for the subsequent development of this work.

3.3.3 ΔV computation

Once the sequence of heliocentric arcs is computed, it is necessary to evaluate the total ΔV associated with the mission.

Departure

The initial phase of the mission requires the spacecraft to escape the SOI of the departing planet with a defined hyperbolic excess velocity $\boldsymbol{v}_\infty$. The heliocentric

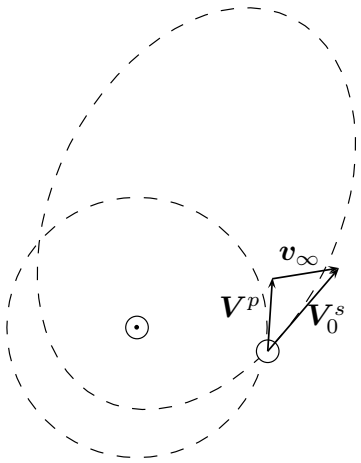

Figure 3.5: Velocity composition at the departure planet.

velocity at the SOI crossing must match the starting velocity for the first heliocentric arc ($\boldsymbol{V}_0^s$), which has already been computed. Since $\boldsymbol{v}_\infty$ is a planetocentric velocity, it must be summed to the velocity of the planet $\boldsymbol{V}^p$ (evaluated at the time of departure) in order to obtain $\boldsymbol{V}_0^s$, as depicted in figure 3.5. It is easy then to compute the first contribution to the total ΔV:

$$\Delta V_{\text{dep}} = v_\infty = \|\boldsymbol{V}_0^s - \boldsymbol{V}^p\| \tag{3.3}$$

Since the only relevant information regarding the launch phase is $\boldsymbol{v}_\infty$, it does not matter how that velocity is obtained: it can be achieved with single launch phase (direct injection into heliocentric orbit), multiple phases (launch into parking orbit followed by a subsequent maneuver), or any suitable strategy. For this reason, most of the parameters about the launch phase are effectively decoupled from the interplanetary trajectory, increasing the design flexibility.

In case of direct injection, the maximum v_∞ (or the equivalent parameter $C_3 = v_\infty^2$) is usually available in the launcher manual as a function of the spacecraft mass and the escape orbit declination.

Gravity assist

The impulsive velocity change needed to connect two arcs is obtained through a gravity assist; it is stated as "impulsive" because under the linked conics model, the SOI is assumed to be zero-dimensional. The heliocentric velocities $\boldsymbol{V}^s_{\text{arc}_i}$ and $\boldsymbol{V}^s_{\text{arc}_{i+1}}$ are known (they are the final velocity of the i-th arc and the initial velocity of the i+1-th arc respectively), so it is easy, referring to figure 3.6, to

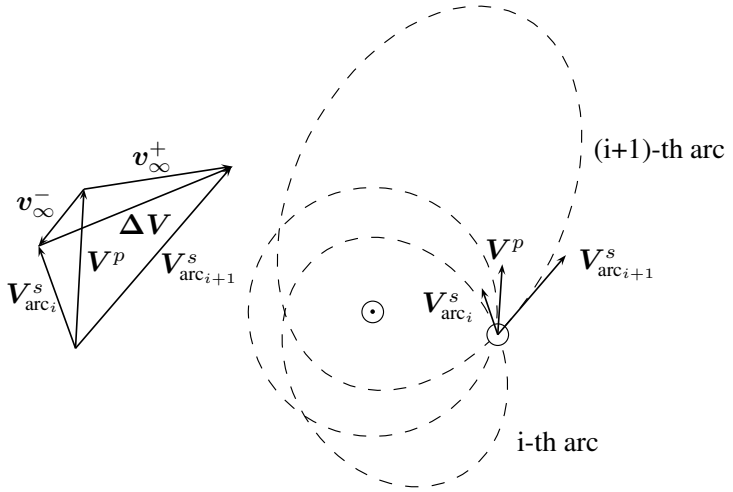

Figure 3.6: Relation between heliocentric and planetocentric velocities.

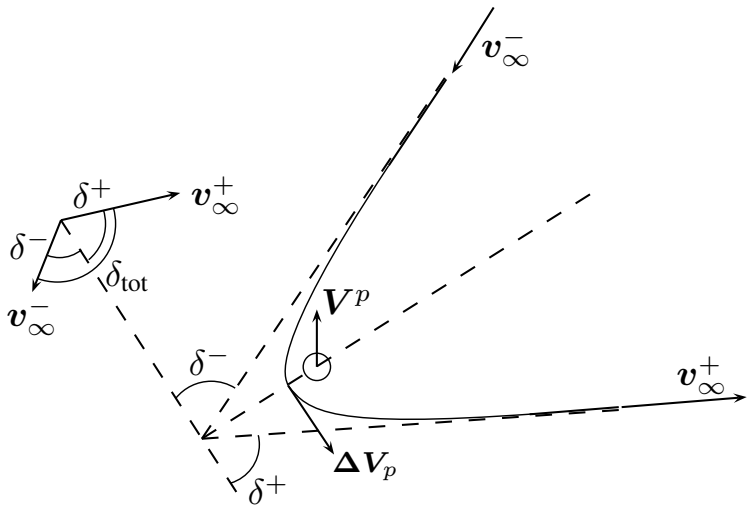

Figure 3.7: Geometry of the powered gravity assist.

evaluate the velocity variation required to connect said arcs:

$$\boldsymbol{\Delta V} = \boldsymbol{V}^s_{\mathrm{arc}_{i+1}}(t_{i+1} = 0) - \boldsymbol{V}^s_{\mathrm{arc}_i}(t_i = \mathrm{TOF}_i)$$

since the time interval associated with the i-th arc is defined as $t_i = [0, \mathrm{TOF}_i]$.

It is desirable that most of the energy associated with this velocity change is gathered through a momentum exchange with the planet (because the remaining part must be supplied by the spacecraft autonomous propulsion system). In order to assess how much firing will be needed, the heliocentric point of view is not sufficient: it is necessary to expand the SOI and to analyze the trajectory around the planet. Observing figure 3.6, it is apparent that the planetocentric velocities at the SOI crossing can be calculated as:

$$\begin{aligned}\boldsymbol{v}^-_\infty &= \boldsymbol{V}^s_{\mathrm{arc}_i}(t_i = \mathrm{TOF}_i) - \boldsymbol{V}^p \\ \boldsymbol{v}^-_\infty &= \boldsymbol{V}^s_{\mathrm{arc}_{i+1}}(t_{i+1} = 0) - \boldsymbol{V}^p\end{aligned}$$

where $\boldsymbol{V}^p$ is the same in both cases, and it is evaluated ad the instant of the gravity assist. The direction of these vectors identify the asymptotes of the hyperbola, as shown in figure 3.7. Since $\boldsymbol{v}^-_\infty$ and $\boldsymbol{v}^+_\infty$ depend on the terminal velocities of two different Lambert arcs, in the general case their magnitudes are different: for this reason, every gravity assist is assumed to be "powered", meaning that a pericentric maneuver $\boldsymbol{\Delta V}_p$ is always evaluated (if the non-powered gravity assist is possible the magnitude of said maneuver will be null).

The total bending angle δ_{tot} can be easily evaluated as

$$\delta_{\mathrm{tot}} = \arccos \frac{\boldsymbol{v}^-_\infty \cdot \boldsymbol{v}^+_\infty}{v^-_\infty v^+_\infty} \tag{3.4}$$

As depicted in figure 3.7, δ_{tot} is the sum of two contributions, one associated with $\boldsymbol{v}^-_\infty$ (and thus labeled as δ^-) and the other associated with $\boldsymbol{v}^+_\infty$ (δ^+).

The plane of the flyby trajectory is known: it contains the center of the attractor and its normal direction is simply

$$\hat{\boldsymbol{n}} = \frac{\boldsymbol{v}^-_\infty \times \boldsymbol{v}^+_\infty}{v^-_\infty v^+_\infty}$$

hence, in order to completely define both the ingoing and the outgoing hyperbola leg, two additional parameters each are needed. The magnitudes of $\boldsymbol{v}^-_\infty$ and $\boldsymbol{v}^+_\infty$ provide one each, whereas the pericenter radius (clearly the same for both legs) serve as the last parameter. Unfortunately, the relation among r_p, v^-_∞ and v^+_∞ cannot be solved explicitly; instead, some iterative method must be

implemented. Following a slightly modified procedure from Labunsky [1], the constraint to be enforced is $\delta_{\text{tot}} = \delta^- + \delta^+$; the relation between each deflection angle (δ^- and δ^+) and the pericenter radius can be computed with the following procedure:

$$\begin{aligned} v_\infty^2 &= \lim_{r\to\infty}\left[\mu\left(\frac{2}{r}-\frac{1}{a}\right)\right] = -\frac{\mu}{a} \Rightarrow a = -\frac{\mu}{v_\infty^2} \\ r_p &= a(1-e) = \frac{\mu}{v_\infty^2}(e-1) \\ e &= \frac{1}{\sin\delta} = 1 + r_p\frac{v_\infty^2}{\mu} \Rightarrow \sin\delta = \frac{\mu}{\mu + r_p v_\infty^2} \end{aligned}$$

and therefore the expression to be evaluated by the iterative method is

$$\begin{aligned} F(r_p) &= \delta_{\text{tot}} - \delta^- - \delta^+ = \\ &= \delta_{\text{tot}} - \arcsin\frac{\mu}{\mu + r_p v_\infty^{-2}} - \arcsin\frac{\mu}{\mu + r_p v_\infty^{+2}} = 0 \\ &= \delta_{\text{tot}} - \arcsin\frac{-a^-}{-a^- + r_p} - \arcsin\frac{-a^+}{-a^+ + r_p} = 0 \end{aligned}$$

Its first derivative with respect to the pericenter radius is:

$$\frac{dF}{dr_p} = \frac{-a^-}{(-a^- + r_p)^2\sqrt{1-\left(\frac{-a^-}{-a^-+r_p}\right)^2}} + \frac{-a^+}{(-a^+ + r_p)^2\sqrt{1-\left(\frac{-a^+}{-a^+ + r_p}\right)^2}}$$

Remembering that the semi-major axis is negative, it is clear that $\frac{dF}{dr_p} > 0 \quad \forall r_p$. Since $F(r_p = 0) = \delta_{\text{tot}} - \pi \leqslant 0$ (from equation 3.4 it is apparent that δ_{tot} belongs to the interval $[0, \pi]$), $F(r_p)$ is a monotonically growing, single-rooted function in the range $r_p = [0, +\infty)$. Moreover, its second derivative is always negative, hence $F(r_p)$ is convex upwards. This suggests to use the minimum admissible value for r_p as a starting point, so that the convergence is quick and certain; moreover, if $F(r_{p\,\min}) > 0$ the process can be stopped at once, since it would converge on a value smaller than the allowed minimum pericenter radius (maneuver not feasible).

In figure 3.8, $F(r_p)$ is diagrammed as a function of r_p, for a gravity assist around the Earth with $v_\infty^- = v_\infty^+ = 5\,\text{km/s}$. $F(r_p)$ is computed for different values of total bending angle, and the minimum approach distance allowed ($\min(r_p) = 1.1\ R^\oplus$) is highlighted. It is possible to observe how $\delta_{\text{tot}} = \pi$ would be feasible only if $r_p = 0$ were acceptable (it would imply that the attractor is a point mass).

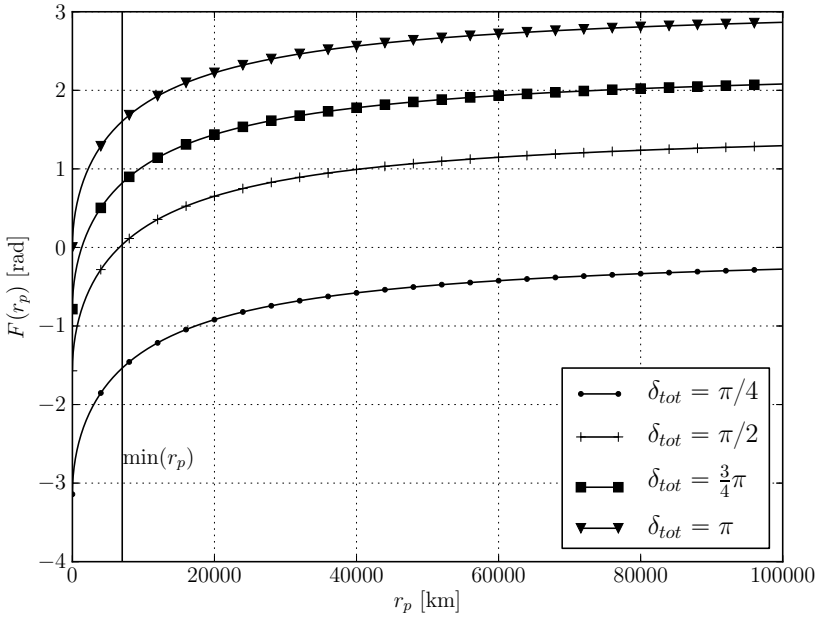

Figure 3.8: Objective function for r_p computation: gravity assist of Earth, $v_\infty^- = v_\infty^+ = 5\,\mathrm{km/s}$.

Orbital insertion burn

The user is allowed to choose whether or not the orbit must be closed around the arrival planet. If the closure is not required, the last planetary passage closest approach remains undefined (according to the linked conics model the insertion point is not specified); otherwise it is assumed that the spacecraft is inserted in a highly elliptical planetocentric orbit (a quite common for scientific exploration missions). This orbit has pericentric radius $r_p = 1.1r^p$ (i.e. 110% of the planetary radius) and its apocenter is $r_p = 30r^p$; since this setting is the same for every cost evaluations it is not expected to weight much on the overall performance analysis.

The planetocentric orbit is assumed to have the same inclination of the incoming hyperbola (which is however undefined according to the linked conics model), so that the only required maneuver is a pericentric tangent burn. Its magnitude can be calculated as the difference of the pericentric velocity of the

incoming hyperbola and the corresponding velocity of the desired ellipse:

$$v_p^{\text{hyp}} = \sqrt{\frac{2\mu}{r_p} + v_\infty^2}$$
$$v_p^{\text{ell}} = \sqrt{2\mu\left(\frac{1}{r_p} - \frac{1}{r_p + r_a}\right)}$$
$$\Delta V_{\text{clo}} = \left|v_p^{\text{hyp}} - v_p^{\text{ell}}\right|$$

Chapter 4

Results

In this chapter the Interplanetary Module performances have been evaluated over a small set of real missions. The actual trajectory plan will be regarded as a good estimate of the global optimum, even if it is designed according to precise force models, and it can be driven by requisites unrelated to the minimization of ΔV and TOF.

The optimization algorithms must be initialized with the population size and the number of generations/iterations (both optimizers stopping criterion is based on the number of the iterations). It has been decided, following [14], to bind the population size and the number of iterations to the complexity of the function. The relation between the number of variables and the optimization parameters are summarized in table 4.1.

Variables	$n = \mathrm{N_{GA}} + 2$
Population	$n \times 100$
Iterations	$n \times 100$

Table 4.1: Population size and number of iterations as a function of the problem complexity.

The results will be presented with two main plots:

♢ the **time history plot** shows the objectives for each individual on the Pareto front, at selected instants of the optimization process;

♢ in the **distance plot** the abscissa of each individual is the departure date, whereas the ordinate is a *distance* from the actual departure date. This is computed simply as $d = \sum_i \left| x_i^{\mathrm{act}} - x_i^{\mathrm{opt}} \right|$, where x_i^{act} and x_i^{opt} represent the i-th element of the actual and the optimized solutions respectively.

The distance d can be regarded as an indication about the goodness of a solution, but it must be kept in mind that the actual solution can be influenced by a variety of factors (which have been neglected in this analysis).

All the results have been obtained running the program on a laptop computer; its features are listed in table 4.2.

CPU	Intel i5M430
Cores	2
Clock frequency	2.27 GHz
Memory	4 Gb
Operating system	Linux (Kubuntu v12.10)
Word size	64 bit

Table 4.2: Characteristics of the machine where the tests have been run.

4.1 Direct transfer: Venus Express

The first benchmark mission is Venus Express because its interplanetary trajectory represents an effective example of an high thrust direct transfer between two planets.

Venus Express is the first ESA mission aimed to Venus exploration: it has been built reusing the design of Mars Express with some important modifications (mostly related to the increased solar and radiation fluxes). Its main goal is the thorough analysis of the Venusian atmosphere and superficial temperature. The relevant data about the spacecraft interplanetary trajectory are listed in 4.3.

Departure	
Launch date	November 9th, 2005
Launch site	Baikonur
m_{launch}	1270 kg
C_3 (Soyuz + Fregat)	8.5 km^2/s^2
Arrival	
Arrival date	April 11th, 2006
Capture orbit type	Highly elliptical
Capture orbit h_p	400 km
Capture orbit h_a	33 000 km
Main closure burn	1.251 km/s

Table 4.3: Venus Express interplanetary trajectory data (sources: [30], [31], [32]).

The ΔV associated with the transfer has been computed using the IM objective function (see §3.3.3), with an exhaustive search spanning one year for

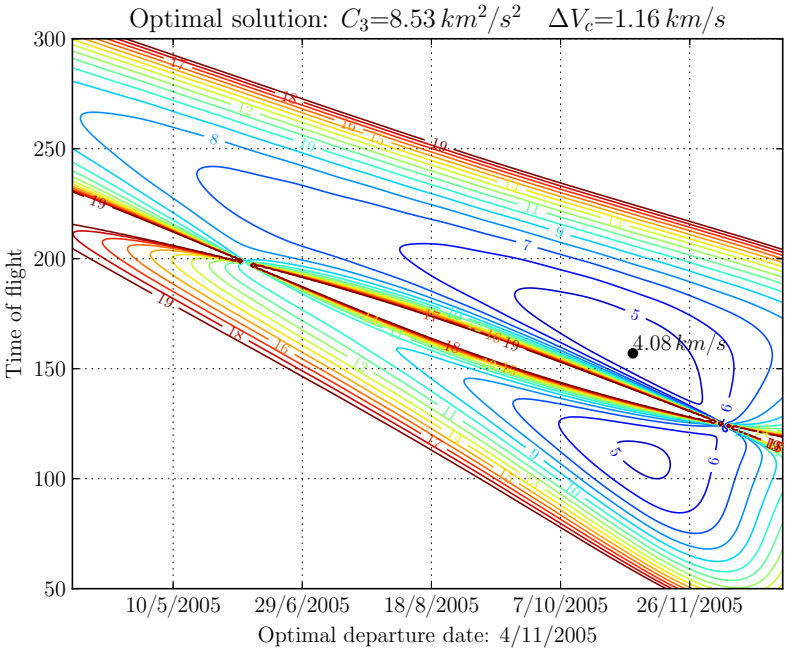

Figure 4.1: Venus Express pork chop, as evaluated from the IM objective function.

the departure date and from 50 to 300 transfer days; the step sizes is 1 day for both departure date and time of flight. The results are shown in figure 4.1.

As illustrated, the optimal departure date, the launch C_3 and the closure ΔV are close to the actual values. The actual departure date does not coincide with the optimal solution: as a matter of fact, the launch was postponed due to particulate contamination (insulating material detached from the Fregat upper stage) found in the launcher fairing [33]. This result proves only that the Lambert solver and the terminal ΔV evaluation work correctly (no gravity assist is performed).

In order to test the optimizers capabilities, the problem has been fed to the IM module, using the parameters listed in table 4.4. The departure date boundaries have been chosen to cover a three year period around the actual departure date; it has been verified that further enlargement of the departing interval would lead to better solutions (for instance, departing on May 25th 2007 the total ΔV would have been about 3 km/s), which are confirmed to be local minima via exhaustive search. The maximum C_3 has been derived from the launcher manual [32], given the spacecraft launch mass. Of course no gravity assists have been requested.

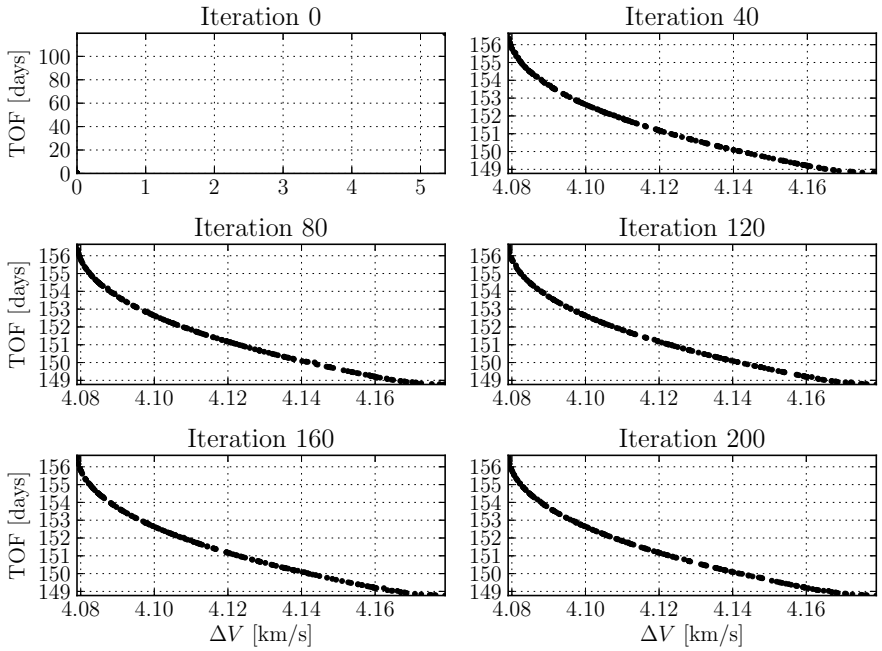

Figure 4.2: Venus Express, DG-MOPSO: population time history.

The optimization has been run with both optimizers, using the input parameters listed in table 4.4 and the results are shown in figures 4.2, 4.3, 4.4 and 4.5. It is apparent that in this case DG-MOPSO has found a larger set of non-dominated solutions, even if both have found a solution very close to the global optima. From the history plot of DG-MOPSO (figure 4.2) it can be observed that the starting iteration is empty: this is because the DG-MOPSO algorithm populates the Pareto front during the process, as soon as non-dominated solutions are found among the population. On the other hand, NSGA-II keeps track of the full population into the output file, henceforth the plot is "full" from the starting iteration.

Departure date lower boundary	May 1st, 2004
Departure date upper boundary	April 1st, 2007
Maximum C_3	$8.5\,\mathrm{km}^2/\mathrm{s}^2$
Gravity assist sequence	None
Orbit insertion required	Yes
Population	200
Generations	200

Table 4.4: Input parameters for Venus Express trajectory optimization.

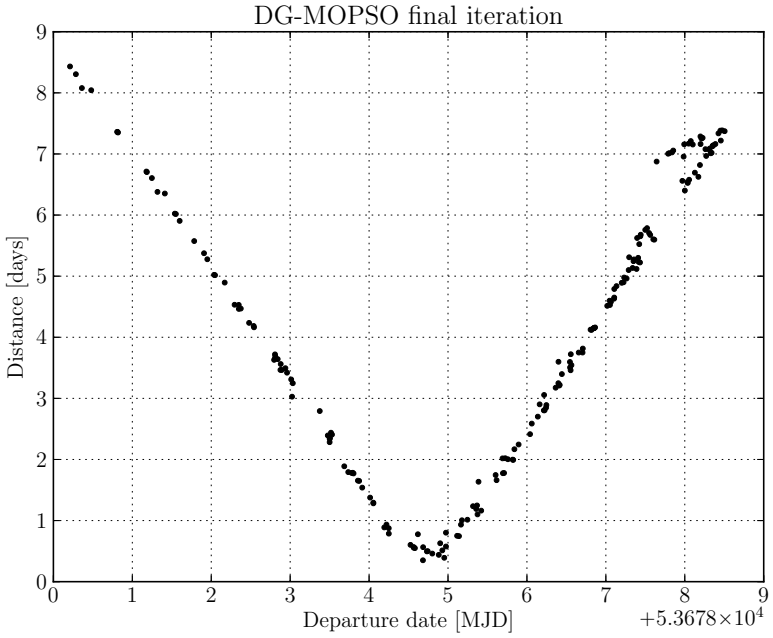

Figure 4.3: Venus Express, DG-MOPSO: non-dominated solutions at the last iteration (departure date vs distance from actual trajectory).

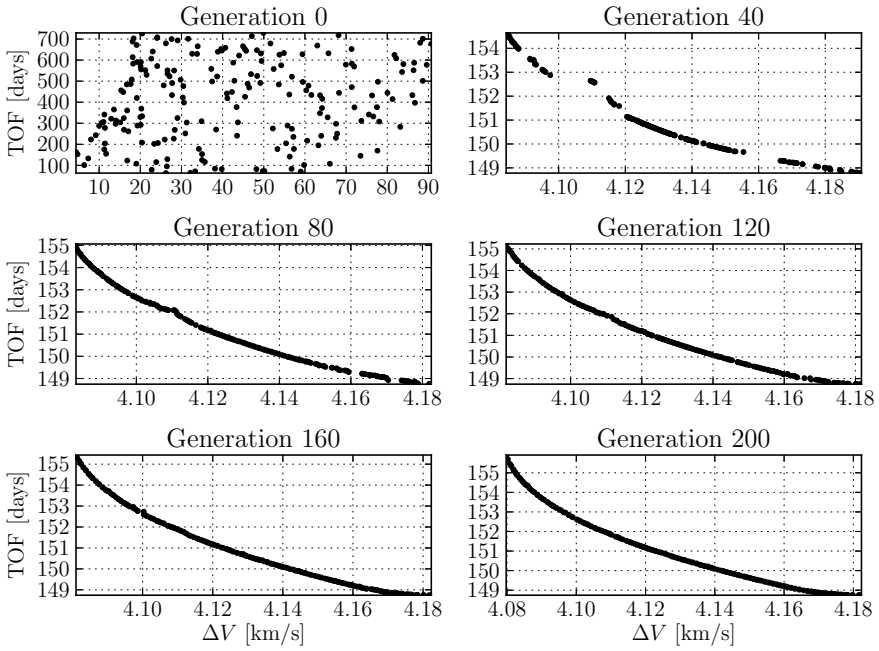

Figure 4.4: Venus Express, NSGA-II: population time history.

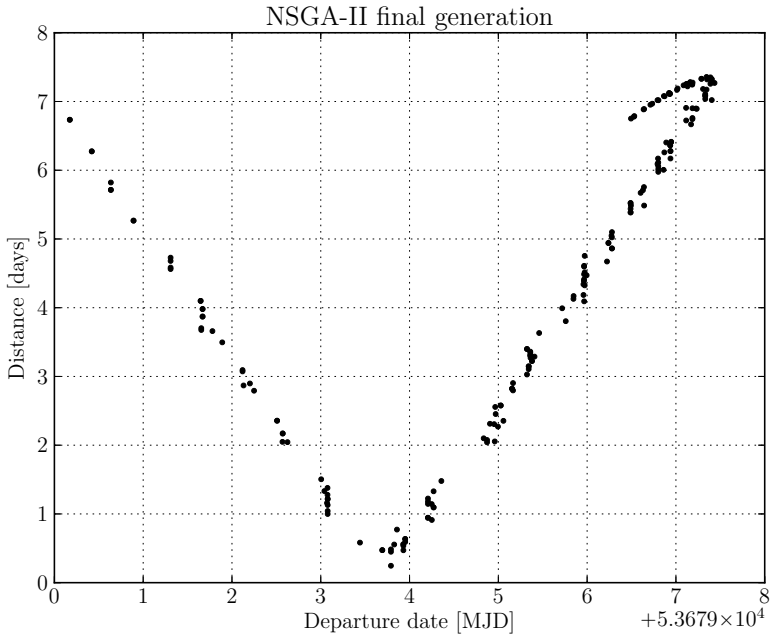

Figure 4.5: Venus Express, NSGA-II: non-dominated solutions at the last iteration (departure date vs distance from actual trajectory).

In order to find a single "best solution" a scalar figure of merit must be chosen: since the resulting TOF magnitude does not pose any technical issue, the total ΔV has been selected. According to this criterion, the optimal solutions found by the two test algorithms are listed in table 4.5. It is useful to remark that they have been singled out in the post-processing phase, hence this operation does not affect the optimization process in any way (the optimizers output is always a Pareto front).

It can be observed that the in both cases the optimization effectively stagnates very close to the aforementioned solution: the difference can be ascribed to the trajectory numeric refinement and to the presence of trajectory correction maneuvers during the coasting phase. The main features of the selected solution (cheapest in terms of ΔV) are summarized in table 4.5). It must be noted that the optimal solution, or a solution close enough to it, has been found during the early generations (for both cases); the subsequent offspring bring marginal improvement, and therefore could be avoided. However, since the whole process takes less than a minute (on the machine described in the beginning of this chapter), this is not regarded as an issue.

	DG-MOPSO	**NSGA-II**
Departure date	November 4th, 2005	November 3rd, 2005
Total time of flight	157 days	157 days
Arrival date	April 10th, 2006	April 9th, 2006
C_3	$8.41\,\mathrm{km}^2/\mathrm{s}^2$	$8.41\,\mathrm{km}^2/\mathrm{s}^2$
Total ΔV	$4.1\,\mathrm{km/s}$	$4.1\,\mathrm{km/s}$
Computation time	26.8 s	43.4 s

Table 4.5: Optimal solutions for Venus Express interplanetary trajectory.

4.2 One gravity assist: Pioneer 11

The next test case comprises one single gravity assist: the mission Pioneer 11 represents an excellent example to this regard, and therefore has been used as a comparison term for the optimization results.

Pioneer 11 is part of the NASA Pioneer Program, and along with its twin Pioneer 10, was designed to explore the asteroid belt and the outer Solar System. Its interplanetary trajectory is composed by two legs: from Earth to Jupiter and from Jupiter to Saturn. The subsequent arc has not been taken in account, since it escapes the Solar System and no reliable information is available about it.

This mission is interesting from the point of view of this work, since it offers the opportunity to test the reliability of the gravity assist evaluation.

Departure	
Launch date	April 6th, 1973
Launch site	Kennedy Space Center
m_{launch}	259 kg
C_3 (Atlas Centaur)	$89.0\,\mathrm{km}^2/\mathrm{s}^2$
Jupiter passage	
Closest approach date	December 3rd, 1974
Closest approach distance	$1.60r_{♃} = 114\,000\,\mathrm{km}$
Saturn passage	
Closest approach date	September 1st, 1979
Closest approach distance	$1.36r_{♄} = 81\,670\,\mathrm{km}$

Table 4.6: Pioneer 11 interplanetary trajectory data (sources: [34], [35], [36]).

Even if the global optimization tool is fairly handy, a preliminary analysis of the problem with some approximated method is strongly advised: in this case, the periodicity of the launch windows is clearly the same as the synodic period of Jupiter with respect to the Earth (almost 400 days). For this reason, the

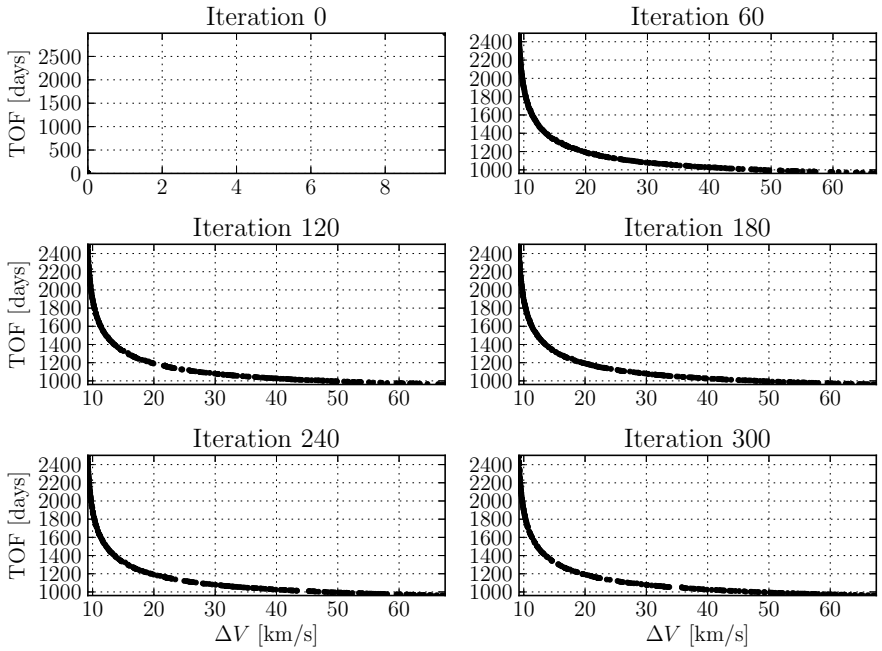

Figure 4.6: Pioneer 11, DG-MOPSO: population time history.

boundaries for the departing date are quite narrow, as it can be observed in table 4.7, otherwise the optimization would converge on better solutions.

Departure date lower boundary	January 1st 1973
Departure date upper boundary	December 31st 1973
Maximum C_3	$90\,\mathrm{km}^2/\mathrm{s}^2$
Gravity assist sequence	J
Orbit insertion required	No
Population	300
Generations	300

Table 4.7: Input parameters for Pioneer 11 trajectory optimization.

The results of the optimization are shown in figures 4.6, 4.7, 4.8 and 4.8: both optimizers are able to find many solutions very close to the actual mission data. The optimal solutions (associated with minimum ΔV) are summarized in table 4.8. The optimal solutions are close to the real mission data, apart from the closest approach distance. Also for this mission, the actual pericenter radius has been adjusted via DSM, prior to Jupiter orbital insertion (see [34]).

Under the current dynamic model, the pericenter radius for the Saturn passage is undefined: in fact, according to the linked conics model, the SOI of

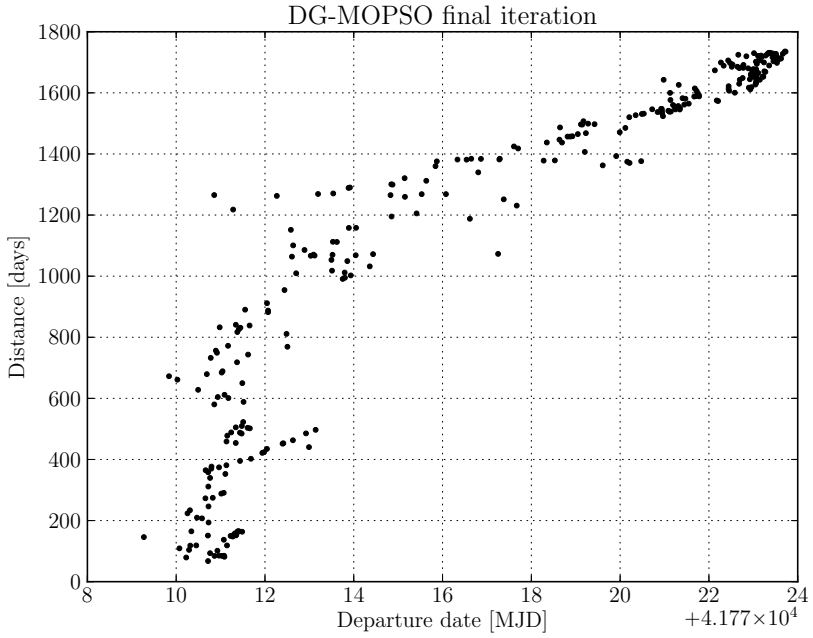

Figure 4.7: Pioneer 11, DG-MOPSO: non-dominated solutions at the last iteration (departure date vs distance from actual trajectory).

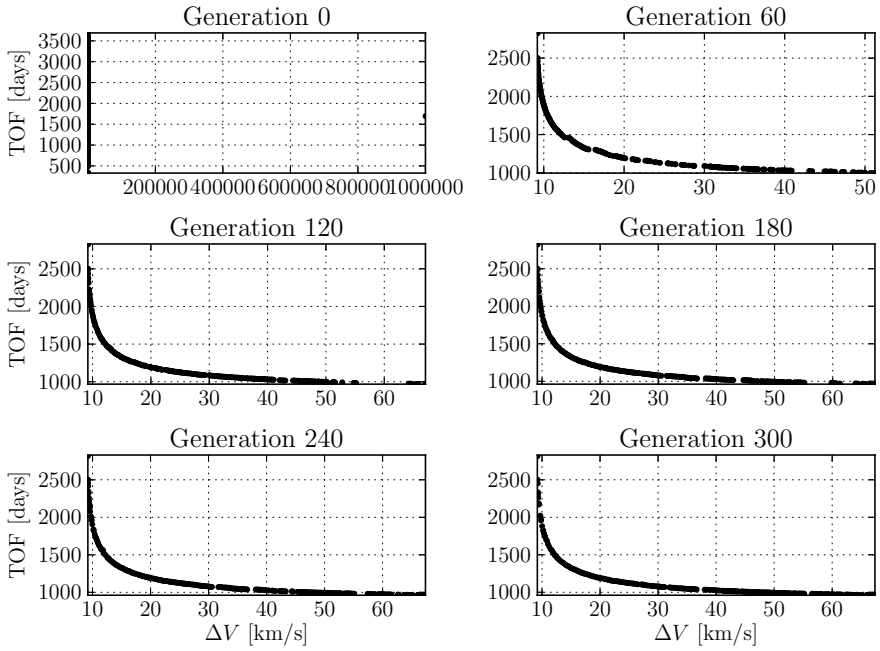

Figure 4.8: Pioneer 11, NSGA-II: population time history.

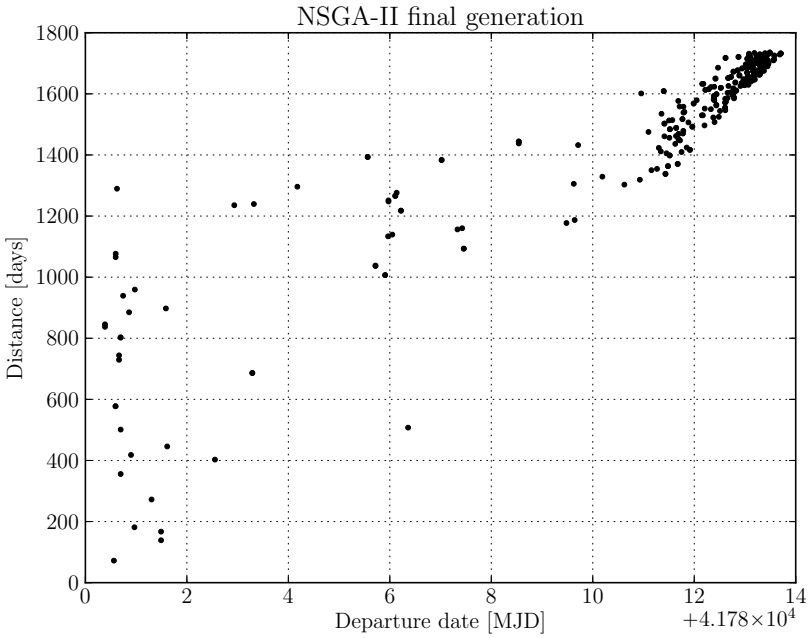

Figure 4.9: Pioneer 11, NSGA-II: non-dominated solutions at the last iteration (departure date vs distance from actual trajectory).

Saturn is non-dimensional. To calculate the closest approach distance it would be necessary to define a subsequent heliocentric arc or the characteristics of a closure orbit; this is not very important during early design stages, as the B-plane targeting can be accomplished with a few inexpensive trajectory correction maneuvers.

	DG-MOPSO	**NSGA-II**
Departure date	April 11th, 1973	April 9th, 1973
Total time of flight	2831 days	2781 days
C_3	$84.64\,\mathrm{km}^2/\mathrm{s}^2$	$84.64\,\mathrm{km}^2/\mathrm{s}^2$
Total ΔV	$9.2\,\mathrm{km/s}$	$9.2\,\mathrm{km/s}$
Jupiter passage		
Date	March 15th 1975	February 27th, 1975
Closest approach	$3.86r_{\mathcal{Y}}$	$3.47r_{\mathcal{Y}}$
ΔV	$0.357\,\mathrm{m/s}$	$0.015\,\mathrm{m/s}$
Arrival date	January 10th, 1981	November 19th, 1980
Computation time	9.7 s	26.8 s

Table 4.8: Optimal solutions for Pioneer 11 interplanetary trajectory.

4.3 Multiple gravity assists: Voyager 2

In order to assess the performances of the optimizers on a true multi-gravity-assist trajectory, the mission Voyager 2 has been analyzed. This probe has been launched before its twin sister Voyager 1, but it had the chance to visited more planets. The relevant data about the trajectory are listed in table 4.9.

Departure	
Launch date	August 20th 1977
Launch site	Kennedy Space Center
m_{launch}	722 kg
C_3 (Titan IIIE)	102.4 km^2/s^2
Jupiter GA	
Closest approach date	July 9th, 1979
Closest approach distance	$8.97r_{♃} = 641\,500$ km
Saturn GA	
Closest approach date	August 25th, 1981
Closest approach distance	$2.68r_{♄} = 161\,270$ km
Uranus GA	
Closest approach date	January 24, 1986
Closest approach distance	$4.19r_{♅} = 107\,060$ km
Neptune GA	
Closest approach date	August 25, 1989
Closest approach distance	$1.20r_{♆} = 29\,710$ km

Table 4.9: Voyager 2 interplanetary trajectory data (source: [37]).

Departure date lower boundary	January 1st 1974
Departure date upper boundary	December 31st 1980
Maximum C_3	105 km^2/s^2
Gravity assist sequence	JSUN
Orbit insertion required	No
Population	500
Generations	500

Table 4.10: Input parameters for Voyager 2 trajectory optimization.

The optimization has been run with the parameters illustrated in table 4.10: the search window for the departure time is fairly large, and nevertheless very good results have been achieved; they are depicted in in figures 4.10, 4.11, 4.12 and 4.12. It must be noted that DG-MOPSO has converged on a single launch window (around July 1976), whereas NSGA-II preserved a larger diversity, as

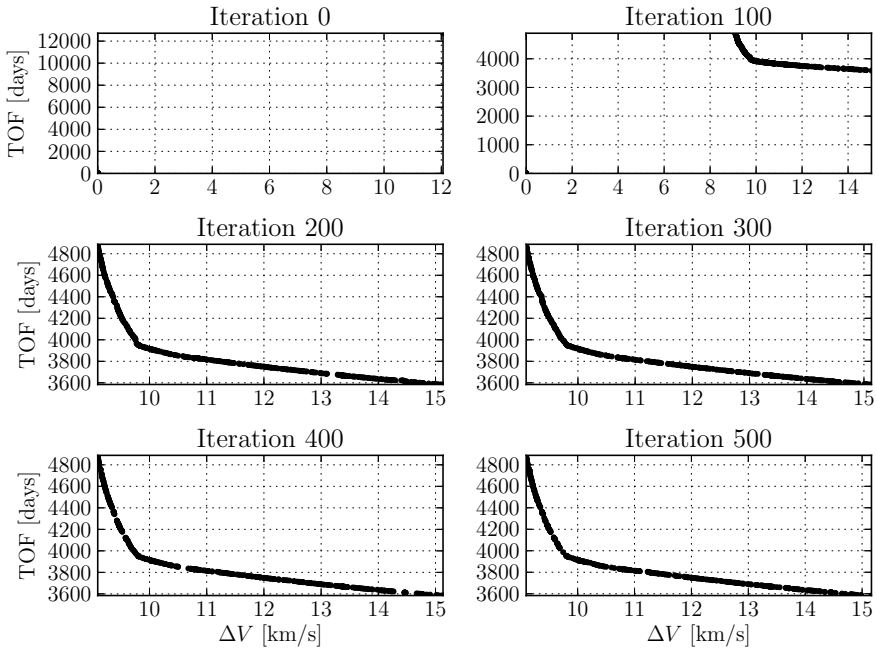

Figure 4.10: Voyager 2, DG-MOPSO: population time history.

can be observed in particular in figure 4.13; again, these groups of solutions are separated by a time interval almost equal to Jupiter synodic period. The global solution is almost the same for both optimizers: it is better than the actual trajectory, in fact the total ΔV would have been reduced by launching in 1976.

4.4 Conclusions and recommendations

The objective of this work was the development the 3rd version of STA Interplanetary Module: the software has been implemented according to the requirements listed in §2.3, and its capabilities were tested with respect to three past successfully interplanetary missions. The results are very good, but this is connected to the kind of missions selected. In fact the adopted formulation is well suited for the design of missions directly targeted to the outer Solar System, which typically have large C_3 values (and in turn a smaller launch mass allowed for the probe). Many interesting missions, like Galileo and Cassini-Huygens, are much heavier than the Pioneer or the Voyager probes. The trajectory implemented for these spacecrafts foresee a few gravity assists around Venus and Earth, and to obtain the maximum energy increase it is crucial to correct some of the coasting arcs with a few not-so-negligible DSMs (not feasible under the

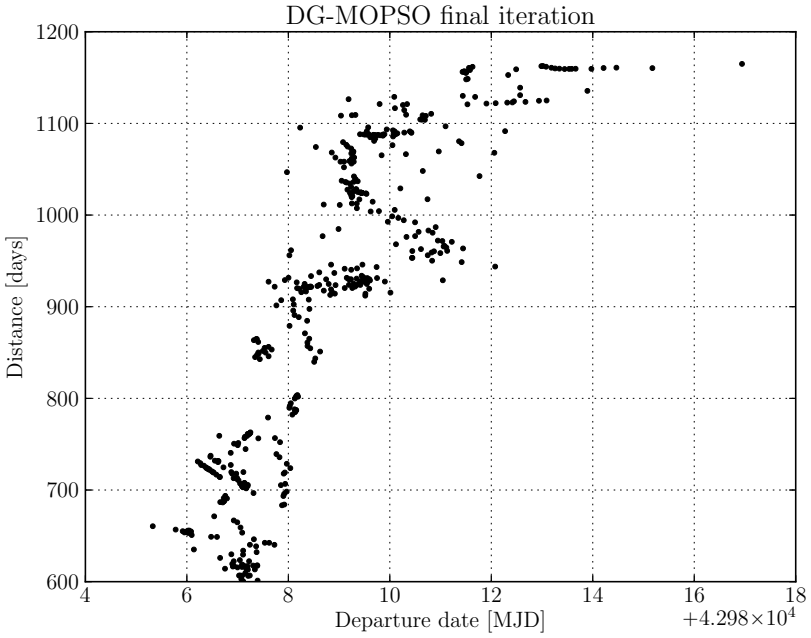

Figure 4.11: Voyager 2, DG-MOPSO: non-dominated solutions at the last iteration (departure date vs distance from actual trajectory).

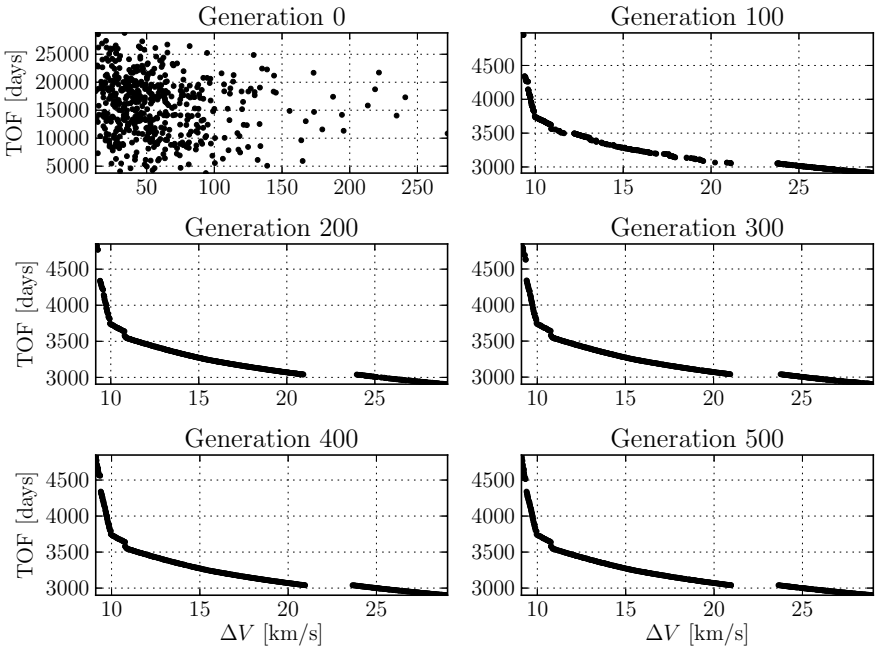

Figure 4.12: Voyager 2, NSGA-II: population time history.

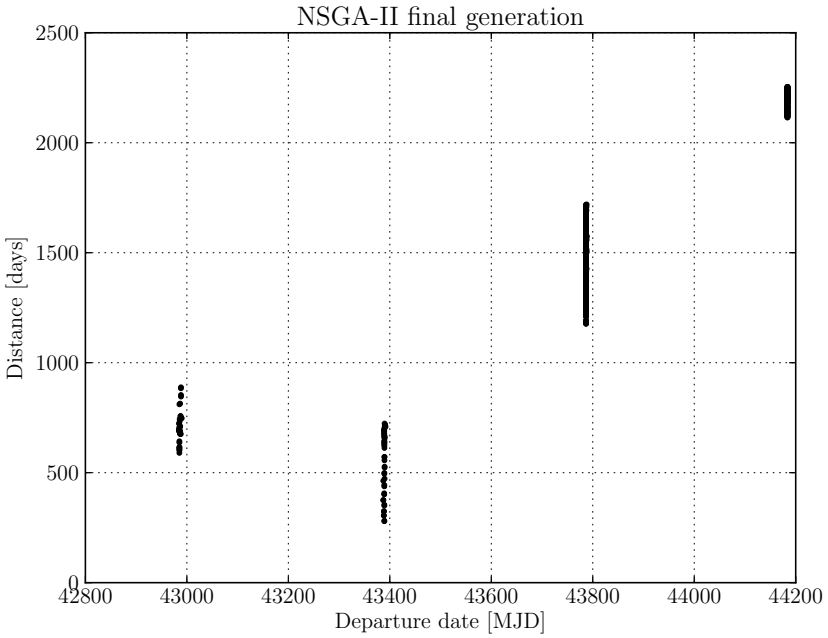

Figure 4.13: Voyager 2, NSGA-II: non-dominated solutions at the last iteration (departure date vs distance from actual trajectory).

current trajectory parametrization).

It would be desirable to implement a different formulation for the dynamic model, in order to augment the design flexibility, even if the complexity of the model will increase.

Most of the requirements have been implemented and tested; those which have not been satisfied are listed here, along with the reasons for the missing feature:

♢ **STA-IM-F-0122: Local Optimizer for Quick Optimizer**: the local optimizer is not believed necessary for a preliminary design tool. The numerical targeting becomes useful during the advanced phases of the design, in order to refine the trajectory and define small correction maneuvers;

♢ **STA-IM-F-0200: Custom Optimizer**: the custom optimizer is associated with advanced features which have not been integrated yet due to lack of time. Some of them (like the multi-revolution capability) are already implemented, but are not used by any part of the module.

	DG-MOPSO	**NSGA-II**
Departure date	July 28th 1976	July 29th 1976
Total time of flight	4888 days	4848 days
C_3	$82.8\,\mathrm{km}^2/\mathrm{s}^2$	$82.8\,\mathrm{km}^2/\mathrm{s}^2$
Total ΔV	$9.1\,\mathrm{km/s}$	$9.1\,\mathrm{km/s}$
Jupiter passage		
Date	September 12th 1978	September 2nd 1978
Closest approach	$1.82r_{♃}$	$1.80r_{♃}$
ΔV	$3.03\,\mathrm{m/s}$	$6.4\,\mathrm{m/s}$
Saturn passage		
Date	May 3rd 1981	May 12th 1981
Closest approach	$2.81r_{♄}$	$2.70r_{♄}$
ΔV	$10.1\,\mathrm{m/s}$	$1.02\,\mathrm{m/s}$
Uranus passage		
Date	March 6th 1986	May 2nd, 1986
Closest approach	$4.72r_{♅}$	$4.53r_{♅}$
ΔV	$0.1\,\mathrm{m/s}$	$0.765\,\mathrm{m/s}$
Arrival date	December 16th, 1989	November 6th, 1989
Computation time	$34.2\,\mathrm{s}$	$141.7\,\mathrm{s}$

Table 4.11: Optimal solutions for Pioneer 11 interplanetary trajectory.

Bibliography

[1] Alexei V. Labunsky, Oleg V. Papkov, and Konstantin G. Sukhanov. *Multiple gravity assist interplanetary trajectories*. Gordon and Breach Science Publishers, 1998.

[2] R. E. Diehl, D. I. Kaplan, and P. A. Penzo. "Satellite tour design for the Galileo mission". In: *AIAA 21st Aerospace Science Meeting*. Reno, NV, Jan. 1983.

[3] Jon Andrew Sims, Andrew James Staugler, and James Michael Longuski. "Trajectory options to Pluto via gravity assists from Venus, Mars, and Jupiter". In: *AIAA/AAS Astrodynamics Conference*. San Diego, CA, July 1996, pp. 400–410.

[4] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. "Lipschitzian Optimization Without the Lipschitz Constant". In: *J. Optim. Theory Appl.* 79.1 (Oct. 1993), pp. 157–181. ISSN: 0022-3239.

[5] Waltraud Huyer and Arnold Neumaier. "Global optimization by multilevel coordinate search". In: *J. Global Optimization* (1999), pp. 331–355.

[6] D. Izzo et al. "Search space pruning and global optimisation of multiple gravity assist spacecraft trajectories." In: *J. Global Optimization* 38.2 (2007), pp. 283–296.

[7] R.Y. Rubinstein and D.P. Kroese. *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning*. Information Science and Statistics. Springer, 2004.

[8] B. Raphael and I.F.C. Smith. "A direct stochastic algorithm for global search". In: *J of Applied Mathematics and Computation* 146.2-3 (2003), pp. 729–758.

[9] Christopher Houck North et al. *A Genetic Algorithm for Function Optimization: A Matlab Implementation*. Tech. rep. 1996.

[10] Andrew Chipperfield et al. *Genetic Algorithm TOOLBOX For Use with MATLAB*. Tech. rep. 1994.

[11] Xin Yao and Yong Liu. "Fast Evolutionary Programming". In: *Proceedings of the Fifth Annual Conference on Evolutionary Programming*. MIT Press, 1996, pp. 451–460.

[12] J. Kennedy and R. Eberhart. "Particle swarm optimization". In: *Neural Networks, 1995. Proceedings., IEEE International Conference on*. Vol. 4. Perth, WA, Australia: IEEE, Nov. 1995, 1942–1948 vol.4.

[13] Pierluigi Di Lizia and Gianmarco Radice. *Advanced Global Optimisation Tools for Mission Analysis and Design*. Ariadna study report, contract number: 18139/04/NL/MV. University of Glasgow, department of Aerospace Engineering, 2004.

[14] D.R. Myatt et al. *Advanced Global Optimisation for Mission Analysis and Design*. Ariadna study report, contract number: 18138/04/NL/MV. University of Reading, department of Cybernetics, 2004.

[15] Francesco Castellini. "Global optimization techniques in space mission design". MA thesis. Politecnico di Milano, 2007/2008.

[16] Tim Blackwell and Jürgen Branke. "Multi-swarm Optimization in Dynamic Environments." In: *EvoWorkshops*. Ed. by Günther R. Raidl et al. Vol. 3005. Lecture Notes in Computer Science. Springer, 2004, pp. 489–500.

[17] C. A. Coello Coello and M. S. Lechuga. "MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization". In: *Proceedings of the Evolutionary Computation on 2002. CEC '02. Proceedings of the 2002 Congress - Volume 02*. CEC '02. Washington, DC, USA: IEEE Computer Society, 2002, pp. 1051–1056.

[18] Nikhil Padhye. "Interplanetary Trajectory Optimization with Swing-bys Using Evolutionary Multi-objective Optimization". In: *Proceedings of the 2008 GECCO Conference Companion on Genetic and Evolutionary Computation*. GECCO '08. Atlanta, GA, USA: ACM, 2008, pp. 1835–1838.

[19] Guusje Schouten. "The Space Trajectory Analysis tool and the development of its interplanetary module". MA thesis. Delft University of Technology, Oct. 2006.

[20] Bert Naessens. "Development of the interplanetary module of STA and its application to the Deimos Sample Return mission". MA thesis. Delft University of Technology, Oct. 2007.

[21] Cesar Bernal. *Project Plan of STA 4.0 "Ordovician"*. Tech. rep. STA-ECM-PP-CB-1101. Noordwijk, The Netherlands: ESA-ESTEC, Jan. 2011.

[22] Guillermo Ortega. *Task Description Form - Interplanetary Module v3.0*. Tech. rep. STA-ECM-TDF-1108. Noordwijk, NL: ESA-ESTEC, Mar. 2011.

[23] Requirements & Standards Division ECSS Secretariat. *ECSS. Space engineering - Software*. Tech. rep. ECSS-E-ST-40C. Noordwijk, The Netherlands: ESA-ESTEC, Mar. 2009.

[24] Michele Scotti. *STA Interplanetary Module - System Requirements Document*. Tech. rep. STA-PDM-SRS-1101. Noordwijk, The Netherlands: ESA-ESTEC, May 2011.

[25] H. Curtis. *Orbital Mechanics: For Engineering Students*. Aerospace Engineering. Elsevier Science, 2004.

[26] E. M. Standish. *JPL Planetary and Lunar Ephemerides, DE405/LE405*. Inter-office memorandum. Jet Propulsion Laboratory, Aug. 1998.

[27] Richard H. Battin. *An introduction to the mathematics and methods of astrodynamics*. Revised. AIAA, 1999.

[28] E. R. Lancaster and R. C. Blanchard. *A unified form of Lambert's theorem*. Tech. rep. NASA, 1969.

[29] R. H. Gooding. "A procedure for the solution of Lambert's orbital boundary-value problem". In: *Celestial mechanics and dynamical astronomy* 48 (1990), pp. 145–165.

[30] *Venus Express mission facts*. URL: `http://www.esa.int/Our_Activities/Space_Science/Venus_Express/Venus_Express_mission_facts` (visited on 09/30/2013).

[31] *Venus orbit insertion*. URL: `http://sci.esa.int/venus-express/38947-orbit-insertion` (visited on 09/30/2013).

[32] *Soyuz user's manual ST-GTD-SUM-01*. 3rd ed. STARSEM. Apr. 2001.

[33] *Venus Express preliminary investigations bring encouraging news*. URL: `http://www.esa.int/Our_Activities/Operations/Venus_Express_preliminary_investigations_bring_encouraging_news` (visited on 09/30/2013).

[34] R. B. Miller et al. *Tracking and data system support for the Pioneer Project*. Technical Memorandum 33-584. Pasadena, CA, United States: NASA JPL, Dec. 1975.

[35] Richard O. Fimmel, William Swindell, and Eric Burgess. *Pioneer Odissey.* Revised. NASA Scientific and Technical Information Office, 1977.

[36] Thomas A. Mutch. *Pioneer Saturn encounter.* Technical Report 80807. Moffett Field, CA, United States: NASA Ames Research Center, Sept. 1979.

[37] Raymond L. Heacock. "The Voyager spacecraft". In: *Proceedings of the Evolutionary Computation on 2002. CEC '02. Proceedings of the 2002 Congress - Volume 02.* The Institution of Mechanical Engineers, 1980.