Escola Tècnica Superior d'Enginyeries
Industrial i Aeronàutica de Terrassa

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Study of Earth-to-Mars Transfers with Low-Thrust Propulsion

Project Report

Author: Xavi López Hellín

Director: Dr. Elena Fantino

Collaborators: Pierpaolo Pergola

Place: ETSEIAT – UPC – Terrassa

Date: September 12, 2011

Table of contents

List of Figures iv
List of Tables v
List of Equations vi
Acknowledgements viii
Abstract ix

1. INTRODUCTION 1

2. PROJECT SCOPE 2
2.1 Project Description 2
2.2 Work Breakdown Structure 3

3. BACKGROUND 4
3.1. Low-Thrust propulsion 4
3.1.1. Low-Thrust propulsion systems 4
3.1.2. Low-Thrust propulsion applications 7
3.1.3. Future applications of low-thrust propulsion 12
3.1.4. Comparison between Inner and Outer Solar System applications 14
3.2. Mars Exploration 15
3.2.1. Past and current Mars Missions 15
3.2.2. Future Missions 17

4. BASIC CODE 19
4.1 Main function "EarthMarsLT" 20
4.2 Equations of motion "EQmotion" 23
4.3 Integration Function "IntFcn" 27
4.4. Other auxiliary functions 29
4.5. Corrections and improvements 29
4.6. Results and plots 31

5. STUDY OF TRANSFERS 41
5.1. Chemical Hohmann transfer 41
5.2. Changing the initial conditions of the spacecraft 42
5.3. Changing the departing orbit 52
5.4. Characteristics of existing low-thrust propulsion thrusters 56
5.5. Changing the thruster characteristics 57
5.6. Transfers plots 63

6. CONCLUSIONS 71
6.1. Next Steps 72

7. PROJECT BUDGET 73

8. REFERENCES 74

9. ANNEX 1: MATLAB ® CODES 78

List of Figures

Figure 1. Project Work Breakdown Structure 3
Figure 2. Operational scheme of a gridded-ion engine 5
Figure 3. Operational schematic of a Hall thruster 6
Figure 4. Deep Space 1 trajectory 8
Figure 5. Trajectory of the Dawn mission to Ceres and Vesta 9
Figure 6. Artist's impression of SMART-1 engine in operation 10
Figure 7. Hayabusa transfers to and from the Itokawa asteroid 11
Figure 8. Artist's view of the two BepiColombo spacecrafts 13
Figure 9. Mars 2001 Odyssey Earth-Mars trajectory 17
Figure 10. Earth, Mars and spacecraft predetermined initial states 19
Figure 11. Heliocentric transfer 31
Figure 12. Spacecraft's position in Earth and Mars reference frame 32
Figure 13. Spacecraft's position in Earth reference frame (Earth escape) 32
Figure 14. Spacecraft's position in Mars reference frame (Mars arrival) 33
Figure 15. Heliocentric Distance 33
Figure 16. Instantaneous Mass 34
Figure 17. Parameters in Earth frame 35
Figure 18. Geocentric Distance in Earth frame 35
Figure 19. Geocentric Velocity in Earth frame 36
Figure 20. Energy with respect to Earth in Earth frame 36
Figure 21. Eccentricity at Earth 37
Figure 22. Parameters in Mars frame 37
Figure 23. Marscentric distance in Mars frame 38
Figure 24. Marscentric velocity in Mars frame 38
Figure 25. Energy with respect to Mars in Mars frame 39
Figure 26. Eccentricity at Mars 39
Figure 27. Spacecraft's initial position in the Earth frame of reference 43
Figure 28. Transfer time (a_0 = 22500 km) 45
Figure 29. Transfer time (a_0 = 23500 km) 45
Figure 30. Transfer time (a_0 = 24500 km) 46
Figure 31. Transfer time (a_0 = 25500 km) 46
Figure 32. Fuel mass consumed (a_0 = 22500 km) 46
Figure 33. Fuel mass consumed (a_0 = 23500 km) 47
Figure 34. Fuel mass consumed (a_0 = 24500 km) 47
Figure 35. Fuel mass consumed (a_0 = 25500 km) 47
Figure 36. Transfer time vs $V_{SC\text{-}init}$ ($\theta_{SC\text{-}init}$=0º) 50
Figure 37. Transfer time vs $V_{SC\text{-}init}$ ($\theta_{SC\text{-}init}$=180º) 50
Figure 38. Transfer time vs $V_{SC\text{-}init}$ for all values of $\theta_{SC\text{-}init}$ 51
Figure 39. Fuel mass consumed vs $V_{SC\text{-}init}$ for all values of $\theta_{SC\text{-}init}$ 51

Figure 40. Transfer time vs $\theta_{SC\text{-}init}$ for different departing orbits 53
Figure 41. Transfer time vs $V_{SC\text{-}init}$ for different departing orbits 54
Figure 42. Fuel mass consumed vs $\theta_{SC\text{-}init}$ for different departing orbits 55
Figure 43. Fuel mass consumed vs $V_{SC\text{-}init}$ for different departing orbits 55
Figure 44. Transfer time for different ion engines 59
Figure 45. Fuel mass consumed for different ion engines 59
Figure 46. Transfer time for different hall thrusters 60
Figure 47. Fuel mass consumed for different hall thrusters 60
Figure 48. Transfer time for T6, NEXT and HiPEP ion thrusters 61
Figure 49. Fuel mass consumed for T6, NEXT and HiPEP ion thrusters 62
Figure 50. Heliocentric transfer (HiPEP) 64
Figure 51. Heliocentric distance and instantaneous mass (HiPEP) 64
Figure 52. Heliocentric transfer (NEXT) 65
Figure 53. Heliocentric distance and instantaneous mass (NEXT) 65
Figure 54. Heliocentric transfer (T6) 66
Figure 55. Heliocentric distance and instantaneous mass (T6) 67
Figure 56. Spacecraft's position in Earth and Mars reference frame (T6) 67
Figure 57. Spiralling around Earth in Earth reference frame (T6) 68
Figure 58. Spiralling around Mars in Mars reference frame (T6) 68
Figure 59. Parameters in Earth reference frame (T6) 69
Figure 60. Parameters in Mars reference frame (T6) 69

List of Tables

Table 1. Mars Missions since the early 1990s 16
Table 2. Results of Mars Missions since the early 1990s 16
Table 3. Future Mars Missions 17
Table 4. Input Data 20
Table 5. Plots Summary 23
Table 6. Input data to change initial position and velocity of the spacecraft 44
Table 7. Case of minimum transfer time 48
Table 8. Case of minimum fuel mass consumed 48
Table 9. Initial conditions and output values of the chosen transfer 49
Table 10. Input and output values of the minimum transfer time transfer 52
Table 11. Departing orbits 53
Table 12. Input and output values of the minimum transfer time transfer (changing the departing orbit study) 56
Table 13. Thrusters' characteristics 57
Table 14. Data of best performance transfers 62
Table 15. Data of selected transfers 63
Table 16. Project Budget 73

List of Equations

Equation 1. Sphere of Influence of Earth SOI_E and Mars SOI_M 21
Equation 2. Distance DU, time TU and mass MU adimensional units 21
Equation 3. Adimensional Thrust Th 21
Equation 4. Initial states of Earth, Mars and spacecraft 22
Equation 5. Initial state variables $\mathbf{x}_0$ 22
Equation 6. Vector **y** of the differential equation 24
Equation 7. Vector **y'** of the differential equation 24
Equation 8. Gravitational acceleration of the spacecraft $\mathbf{A}_{SC\text{-}grav}$ 24
Equation 9. Thrust acceleration of the spacecraft $\mathbf{A}_{SC\text{-}F}$ 24
Equation 10. Mass variation of the spacecraft $\dot{M}$ 24
Equation 11. Specific mechanical energy of the heliocentric transfer ξ and relative to Mars ξ_M, ... 24
Equation 12. Eccentricity e 25
Equation 13. Apogee radius R_a of the transfer 25
Equation 14. Periareion radius $R_{p\text{-}M}$ 25
Equation 15. Apoareion radius $R_{a\text{-}M}$ 25
Equation 16. True anomaly v_M around Mars 25
Equation 17. Eccentric anomaly E_M around Mars 25
Equation 18. Mean anomaly M_M around Mars 25
Equation 19. Tangential thrust vector 25
Equation 20. Circumpherential thrust vector 26
Equation 21. Antitangential thrust vector relative to Mars 26
Equation 22. Tangential thrust vector relative to Mars 27
Equation 23. Total acceleration of the spacecraft 27
Equation 24. Acceleration of Earth 27
Equation 25. Acceleration of Mars 27
Equation 26. Velocity at Mars V_{atM} 28
Equation 27. True anomaly of the heliocentric transfer v 28
Equation 28. Initial Mars angle 29
Equation 29. Eccentric anomaly at Mars E_M when $e_M<=1$ 30
Equation 30. Eccentric anomaly at Mars E_M when $e_M>1$ 30
Equation 31. Mean anomaly at Mars M_M when $e_M>1$ 30
Equation 32. First Δv of the Hohmann transfer (leaving Earth's SOI) 41
Equation 33. Second Δv of the Hohmann transfer (arriving at Mars' SOI) 41
Equation 34. Transfer time of the Hohmann transfer 42
Equation 35. Initial position of the spacecraft in heliocentric frame of reference. 42
Equation 36. Initial velocity of the spacecraft in heliocentric frame of reference. 42
Equation 37. Thrust Th 57

Equation 38. Thrust acceleration of the spacecraft $\mathbf{A}_{SC\text{-}F}$ (function of Th) 58
Equation 39. Mass variation of the spacecraft $\dot{M}$ (function of Th) 58

Acknowledgements

This work would not have been possible without the collaboration, guidance and patience of many people who has accompanied me through this project that has been the final challenge of my degree. This is why I want to address a few words to thank them their support.

I wish to thank Elena Fantino for the patience, guidance and valuable advice through this entire project. It would have been impossible to finish it without her essential pieces of advice and suggestions. I really appreciate her support; she was always when I need her advice, even though she was really busy with other matters.

I would also like to thank Pierpaolo Pergola (Università di Pisa, Dipartimento di Ingegneria Aerospaziale Via G. Caruso 8, 56122 Pisa, Italy / Alta SpA, Via A. Gherardesca 5, Ospedaletto, 56121 Pisa, Italy) who has kindly offer his previous work in studying Earth-to-Mars transfers with low-thrust propulsion, as an starting point of this project, offering me the opportunity to further developing the study in this field. I wish also to thank him for his guidance and advice that helped me to understand his work and the best next steps of the project.

Finally, I would also like to thank my family and my friends for their endless patience and confidence with me.

Abstract

Knowledge and investigation about low-thrust propulsion systems have been increasing during the last years and some spacecrafts employing this technology have succeeded in their objectives, which contribute to the possible selection of low-thrust technology for future space missions. Missions to Mars, which are currently being propelled by chemical engines, are subject of study in this field for future exploration missions to the planet.

This project presents a code that computes a transfer from Earth to Mars using low-thrust electric propulsion. This code is analysed and later employed to perform a study of different transfers by varying the parameters of the simulation. The study includes the analysis of the performance of electrical engines (both existing or under development) that have been used in the simulations of the transfer trajectory. The results of the best transfers in terms of minimum flight time and fuel mass consumed are presented and discussed.

The study shows that the main advantage of selecting low-thrust propulsions systems are the savings in propellant mass, with the corresponding increase in the mass of payload that can be delivered to Mars. This makes the mission even more interesting from the point of view of the scientific objectives, especially those implying the delivery of equipment to the surface. On the other hand, the drawback is represented by the higher transfer times, even of some years, than those implied by the adoption of chemical engines. However, this study also shows that as the thruster characteristics of the electrical engines increase in performance, such shortcoming becomes less significant.

1. INTRODUCTION

The present project aims to analyse, modify, develop and run an existing code to compute a transfer from Earth to Mars based on the numerical integration of a four-body (Sun, Earth, Mars and spacecraft) heliocentric model considering the gravitational perturbations from Earth and Mars and the spacecraft's continuous thrust. Once the existing code is defined and described, the objective is to analyse and characterize several transfer scenarios in terms of time of flight, propellant budget and thruster characteristics by varying the input parameters of the simulation code.

The development of low-thrust propulsion technology has increased and there are new missions that are currently using or are going to use this technology in the near future. This is why it is important to study the application of low-thrust in missions where chemical rockets are broadly and commonly used. Mars missions are a constant through the history of space exploration; therefore, the study of the application of low-thrust propulsion to future missions to Mars is especially appealing.

This project is divided into five parts. First of all, a background information chapter provides general information about low-thrust propulsion systems and their current and future applications in existing missions. Moreover, a summary of current and future missions to Mars are also briefly commented. In the second chapter, the basic code and the starting point of the project is described and analysed. Some corrections and improvements are added and the results corresponding to the predetermined values of the problem are shown. The third part of the project is where several transfer scenarios are analysed and characterized. Before beginning the analysis, a chemical Hohmann transfer is presented for the sake of comparison. The study consists in varying several parameters of the simulations, such as the initial position and velocity of the spacecraft, the departing orbit or the thruster characteristics of the low-thrust propulsion system employed; then, different transfers are compared and commented in terms of transfer time and fuel mass consumed. At the end of the analysis, the plots corresponding to the best transfers are presented. The fourth part is dedicated to the conclusions and the planning of the future development of the present project. Finally, a brief budget of the study is provided.

2. PROJECT SCOPE

2.1 Project Description

The project has its cornerstone in a numerical code that integrates a non-optimized Earth-to-Mars trajectory using low-thrust propulsion. The calculations are made in a heliocentric frame of reference, where a dynamical model including the gravitational perturbations of Earth and Mars and forces due to the thrust, is considered.

The code was provided by Pierpaolo Pergola. The code together with all the calculations made in this project are performed using the MATLAB ® environment [1]. The final code and other specific scripts that have been developed for this project are included as Annex 1: Matlab ® Codes.

The background information, the assumptions and the key elements of the project are the following:

- Background information: low-thrust propulsion systems, their applications and current and future missions to Mars.
- Basic code.
- Dynamical model: four-body (Sun, Earth, Mars and spacecraft) gravitational model with the addition of force vectors due to thrust in a heliocentric frame of reference: both, Mars and the Earth, are in circular, coplanar orbits (therefore without considering the planetary ephemerides). The initial orbit of the spacecraft is a geostationary transfer orbit, subsequently changed for further study.
- Simulation parameters: a predetermined set of parameters for the code are available that will be changed in order to study different transfer scenarios.
- Comparisons: a chemical Hohmann transfer will be adopted as reference chemical propulsion transfer.
- Engine characteristics: a list of existing or under development low-thrust engines will be presented with useful characteristics for the study.
- Output and results: transfer time and fuel mass consumed resulting from varying the initial position and velocity of the spacecraft; transfer time and fuel mass consumed resulting from varying the thruster characteristics of the spacecraft. The best transfers in terms of transfer time and fuel mass consumed will be presented after each study.

2.2 Work Breakdown Structure

The structure of the project is made of five main blocks: background, description and analysis of the basic code, study of different transfer scenarios, conclusions and project budget. Each block has a series of work packages, as shown in the Work Breakdown Structure in figure 1:

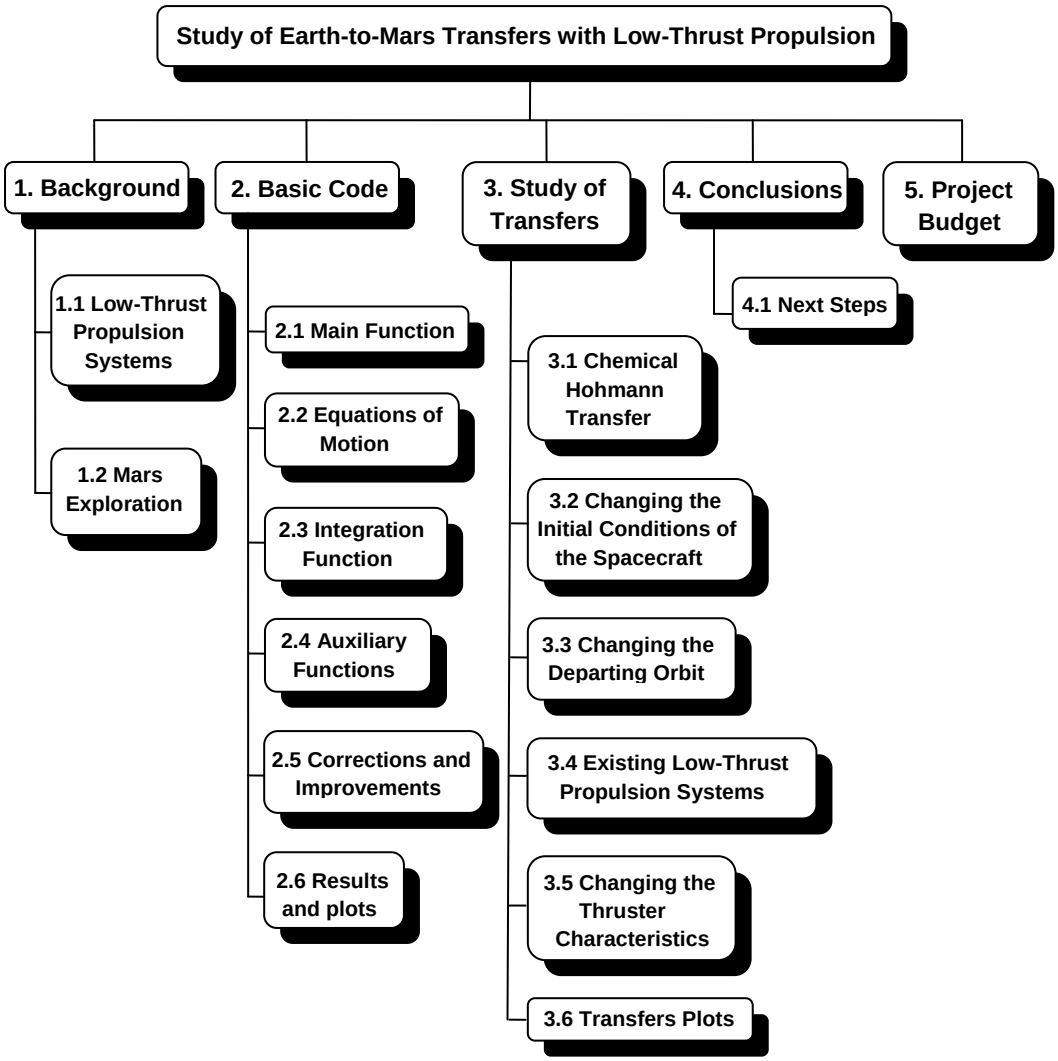

Figure 1. Project Work Breakdown Structure

3. BACKGROUND

Nowadays, much effort, time and resources are being invested in order to continue the development of low-thrust propulsion systems, and new missions that use this technology have been designed or are currently in scope for future planning.
At the same time, Mars exploration is still one of the prior objectives of the space agencies' current budget.

3.1. Low-Thrust propulsion

3.1.1. Low-Thrust propulsion systems

The stimulus for development of new propulsion systems has always been to increase the specific impulse in order to improve the propulsion system performance and reduce the fuel mass needed. Until now, after several developments, increasing the specific impulse has implied a relatively low thrust. This is because achieving high specific impulse and high thrust at the same time is a particularly demanding task in terms of the energy that needs to be imparted to the propellant.

The two main existing low-thrust propulsion systems are electric propulsion and solar sails, the former being that with more years of research.
Electric propulsion makes use of energy from an external source, normally solar power, to electrically accelerate the propellant to higher energies, that is to say, it is not limited to the energy available through the decomposition or combustion of molecular compounds like chemical propulsion.
The different types of electric propulsion are categorized depending on the method of accelerating the propellant:

- Electrothermal (resistojets, arcjets)
- Electrostatic (gridded ion thrusters, Hall-effect thrusters)
- Electromagnetic (PPT, PIT, MPD)

Although electrostatic systems are the industry's state-of-the-art because of their higher specific impulses, the most widely-used electric propulsion systems to date are the electrothermal thrusters.
In electrothermal propulsion thrusters, electrical energy is applied to heat a working fluid to increase the exhaust velocity. Resistojets are a form of electrothermal propulsion that operate by passing a gaseous propellant through an electric heater and then expanding it through a conventional converging diverging nozzle to create thrust. The typical flight operation is superheating catalytically decomposed hydrazine to leverage the propellant commonality of

standard monopropellant chemical propulsion systems. The specific impulse of resistojets is limited by the high molecular mass of hydrazine and the maximum sustainable temperature. Specific impulse values near 350s is achievable; 40 percent higher than the conventional chemical equivalent.

Arcjets are another form of electrothermal propulsion that passes propellants through an electric arc that heats the gas before it expands through a nozzle. Specific impulse is typically in the 500–600s range. Higher specific impulses are achieved because the maximum temperatures are not in contact with engine component walls, though efficiencies are less than those of resistojets.

Electrostatic thrusters such as gridded-ion and Hall thrusters are the leading concepts for primary electric propulsion.

Ion engines are the best developed type of electric propulsion device, dating in conception to the 50's and having been demonstrated in space in 1964 on a suborbital flight of the SERT I spacecraft [2]. The early history and concepts evolved through progressive refinements of various types of ion beam sources used in physics laboratories, the improvements being essentially dictated by the needs for high efficiency, low mass and long life for these sources to be used in space [3],[4],[5].

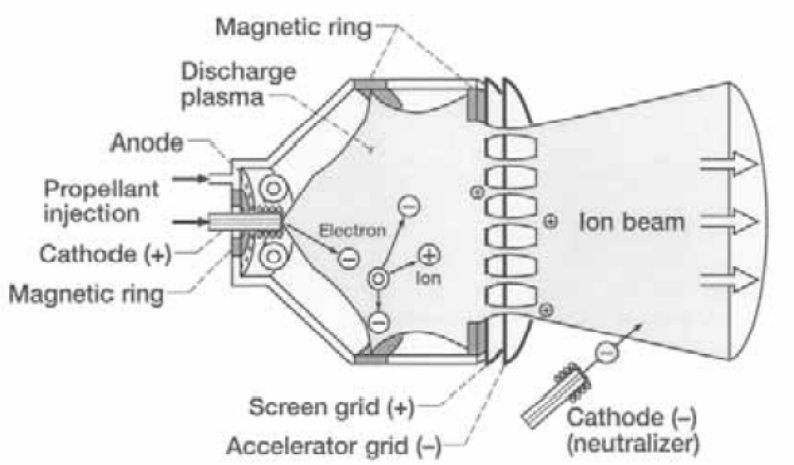

Figure 2. Operational scheme of a gridded-ion engine [4]

Ion thrusters can achieve very high exit velocities, and have typical specific impulses in the 3000–4500s range. An operational scheme is shown in figure 2. These thrusters work on the principle of the application of an electric field to accelerate ions between two grids. Xenon is often used as the propellant, which is ionised by bombardment by electrons produced by a central cathode. An auxiliary magnetic field is used to confine the electrons that then collide with the propellant, causing ionisation. The resulting positive ions are filtered from the electrons using an extraction grid that is slightly negatively biased. These ions are then accelerated between the two grids and ejected at high velocity in order

to produce the thrust. Electrons are also ejected from a second cathode towards the ejected positive ions to neutralise the ejected ions and avoid the charging of the spacecraft.

Hall-effect thrusters typically provide thrust in the region of 100mN and a specific impulse of 1500-1700s. Their principle is to use a rotating plasma of electrons to ionise a propellant injected though an anode. The configuration of the thrusters is such that a radial magnetic field is generated, via inner and outer magnetic coils. An axial electric field is also generated, and the combined effect of these fields generates the Hall effect, which confines the electrons to move in a direction perpendicular to E and B, therefore setting up the azimuthal rotation. The ions are too heavy to be significantly effected by the magnetic fields. They accelerate axially under the influence of the electric field and exit the engine at high velocity, producing thrust. An operational scheme is shown in figure 3.

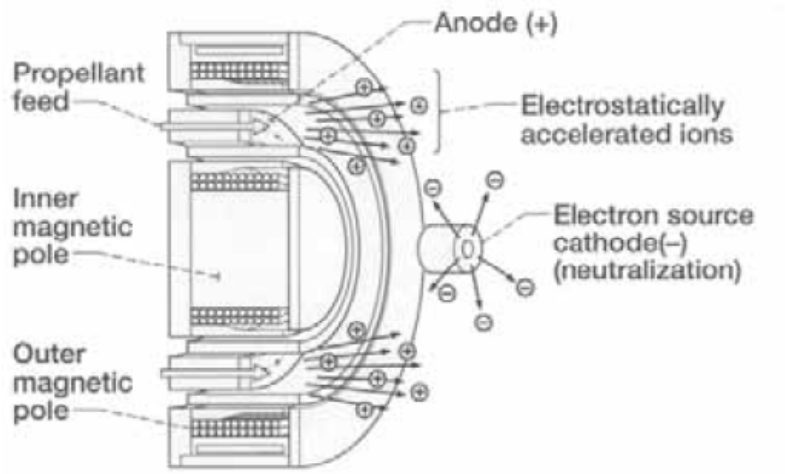

Figure 3. Operational schematic of a Hall thruster [4]

Electromagnetic propulsion devices leverage magnetic fields, self-field or applied, to accelerate plasma, typically with a Lorentz force (J × B) where the accelerating force is proportional to the cross product of the electric current density and the magnetic field. The pulsed plasma thruster (PPT) is a form of electromagnetic propulsion that uses a capacitor to store electrical energy and, when triggered, creates a pulsed arc discharge across the face of a block of propellant, typically Teflon. This arc ablates and ionizes a small amount of propellant and the self-induced magnetic field acts on the ions to create a Lorentz force accelerating the plasma.

Pulsed inductive thrusters (PIT) and magnetoplasmadynamic (MPD) thrusters are additional forms of electromagnetic propulsion. The majority of these concepts are proposed for high power levels, >100 kWe, and have not gained any flight experience.

The other type of low-thrust propulsion system under development is the solar sail, which utilises solar radiation pressure directly to provide thrust. Photons are reflected by the sail and the net momentum transfer applies a force on the sail. Solar sails have been successfully demonstrated in space, and have been used for attitude control for many years [3].

3.1.2. Low-Thrust propulsion applications

After several conceptual studies and a great deal of experimental flight tests, electric propulsion began its commercial applications in the early 1980s, and resistojets became common options for station keeping and attitude control (INTELSAT V series). In the early 1990s electrothermal arcjets were used for north-south station keeping (NSSK) of many communication satellites in geosynchronous Earth orbit (GEO). In 1994, ion thrusters were used for the first time in space for the NSSK of commercial satellites and in 1998 NASA's Deep Space 1 became the first interplanetary mission to benefit from ion propulsion.
Hall thrusters have been used aboard Soviet and Russian spacecraft since the 1970s and, in view of their high specific impulse, relatively high efficiency, and high thrust density, they continue to be developed by industry and government for purposes of orbit insertion, attitude control, and drag compensation. There have also been a few applications of pulsed plasma thrusters, but it was just at the beginning of the century when they were taken more seriously for application on western commercial spacecraft.
In total, the number of electrically propelled spacecraft has gone from single digits in the 1960s to double digits in the 1970s and 1980s and has reached the triple-digit mark in the late 1990s.
Although the primary motivation for development of space-worthy EP systems is the conservation of propellant mass for missions of large characteristic velocity increments, electric thrusters offer a number of attractive secondary operational benefits, including precision and variability of thrust levels and impulse increments, generous shutdown and restart capabilities, and the use of chemically passive propellants. Their major limitations are the need for sophisticated external power sources, very low to modest thrust density capabilities, and little empirical experience with unattended operation in the space environment. All these characteristics serve to circumscribe the classes of missions for which EP may reasonably be considered [6].

Nowadays, there are currently over 200 satellites using electric propulsion and the vast majority is for application in the region between low-Earth orbit (LEO) and geosynchronous Earth orbit (GEO), specifically commercial satellites with GEO operational orbits. Only four spacecraft flew beyond geosynchronous orbit

altitude, two from NASA, one from ESA to the Moon and one from JAXA to a near-Earth object:

- Deep Space 1 (DS1)
- Dawn
- SMART-1
- Hayabusa (MUSES-C)

As mentioned before, Deep Space 1 (DS1) was the first NASA interplanetary mission that used ion propulsion, using the NSTAR electrostatic ion thruster. It was launched on 24 October 1998 and retired on 18 December 2001. As part of the NASA New Millenium Program its main objective was to test its payload of advanced, high risk technologies. It carried out a flyby of asteroid 9969 Braille, which was the mission's science target, and an encounter with Comet Borrelly for further engineering testing.
The trajectory from launch through the end of the mission is illustrated in figure 4:

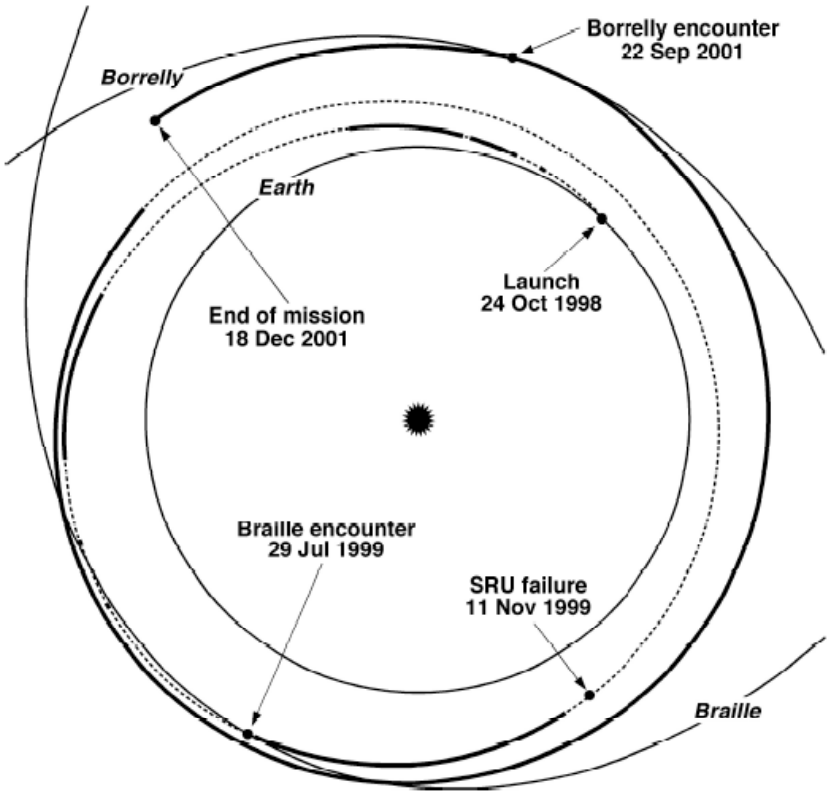

Figure 4. Deep Space 1 trajectory [7]

The dotted portions indicate periods of coasting, and the solid portions show when the ion propulsion system (IPS) was thrusting. The majority of thrusting prior to the failure of the stellar reference unit (SRU) was to test the IPS and to reach the encounter targets. The thrusting after the SRU failure was to reach comet Borrelly and to conserve hydrazine. In the hyperextended mission,

following the comet encounter, the thrusting was for new IPS tests and further hydrazine conservation.
During the course of the flight, the ion propulsion system accumulated 16265 hours of operation and expended 73,4 kg of Xenon for a Δv = 4,3 km/s (more than 2000 hours of this thrusting was at impulse power, consuming less than 4 g/hour at a relatively low I_{sp} = 2200 s). Following the initial unsuccessful attempts to commence and sustain thrusting, the ion propulsion system promptly initiated thrusting on all 199 attempts.

The next NASA interplanetary mission that used low-thrust propulsion was the Dawn mission, that was launched on 27 September 2007 and it is currently operational. Its trajectory is illustrated in figure 5:

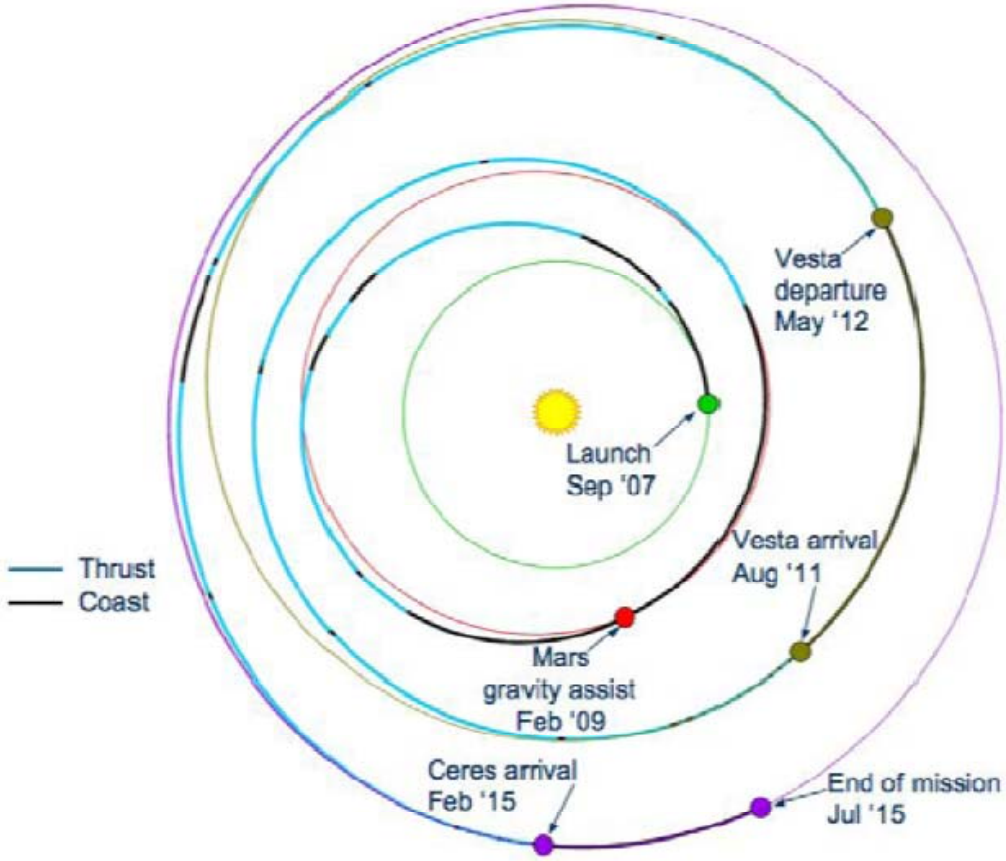

Figure 5. Trajectory of the Dawn mission to Ceres and Vesta [4]

Its main objective is to enter into orbit around the two most massive members of the asteroid belt, Vesta in August 2011 (after a previous Mars gravity assist) and the dwarf planet Ceres in February 2015, in order to characterize the conditions and processes of the Solar System's earliest epoch. It would be the first spacecraft to orbit a celestial body, study it, and then re-embark under powered flight to proceed to a second target.

This mission uses the same xenon ion thrusters as the Deep Space 1 spacecraft, with three redundant units, firing one at a time. Dawn's engines have a specific impulse of 3100s and a thrust of 90mN. It is thanks to the results of DS1's intensive testing of technologies that missions such as Dawn are possible.

Another application of low-thrust propulsion was the ESA mission SMART-1, the first European spacecraft to the Moon, first in the programme "Small Missions for Advanced Research in Technology" whose purpose is to test new technologies that will eventually be used in bigger projects. SMART-1 is the homologue of the NASA mission DS1 as its primary objective is to flight test Solar Electric Primary Propulsion as the key technology for future missions.
The thruster selected for primary propulsion on SMART-1 was the PPS-1350 (figure 6) developed by SNECMA. This is a stationary plasma thruster (SPT) with a chamber diameter of 100 mm, a maximum discharge power of 1.5 kW and a minimum demonstrated discharge power of 480 W. At the beginning of the SMART-1 mission, 1190 W were available for powering the thruster, giving a nominal thrust of 68 mN at a specific impulse of 1640 seconds. The qualified lifetime of the thruster is 7000 hours at maximum power, corresponding to a total impulse of 2 x 106 Ns.

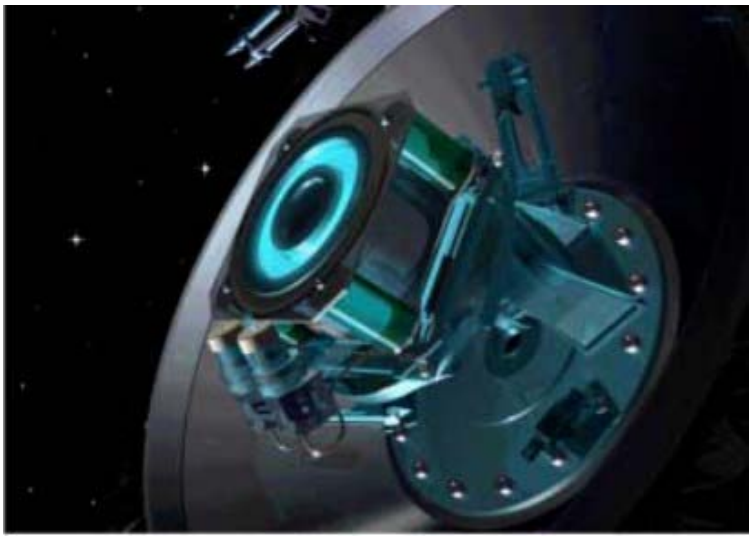

Figure 6. Artist's impression of SMART-1 engine in operation [8]

It was initially successfully launched into an elliptical GTO Earth orbit as an Ariane-5 auxiliary payload on 27 September 2003 and, after a low-thrust transfer orbit, entered a polar elliptical orbit around the Moon on 15 November 2004 for science operations.
The thrusters were used to provide a series of long thrust arcs to increase the semi-major axis of the orbit until the spacecraft could eventually approach the Moon. A thrust of approximately 70mN was developed with a spacecraft initial mass of approximately 350kg. The spiralling transfer took 14 months, and the

spacecraft was finally captured into lunar orbit after a gravitational capture sequence using the combined effects of the Earth's and Moon's gravity fields and a passage close to the Earth-Moon L_1 Lagrange point. The initially high-apocentre elliptical orbit about the Moon was lowered with assistance from the thrusters [1].
The planetary objective was to orbit the Moon for a nominal period of six months, with a possible extension of one year, that was finally implemented by pushing back the mission end date to August 2006.

The Japan Aerospace Exploration Agency (JAXA) has also carried out a mission with low-thrust propulsion technology. The name of the mission was Hayabusa (MUSES-C) and it was designed to verify the practicality of acquired technology developed to achieve future sample return missions. Its main objective was to return a sample of material from a small near-Earth asteroid named Itokawa to Earth for further analysis. The probe was launched on 9 May 2003, rendezvoused with Itokawa in September 2005 and two months later landed on the asteroid and collected samples in the form of tiny grains of asteroidal material. After overcoming many problems, the probe finally returned to Earth on 13 June 2010 having successfully carried out its mission objectives and completing the world's first ever return space mission to a celestial body other than the Moon. Its transfer to and from the asteroid are shown in figure 7.

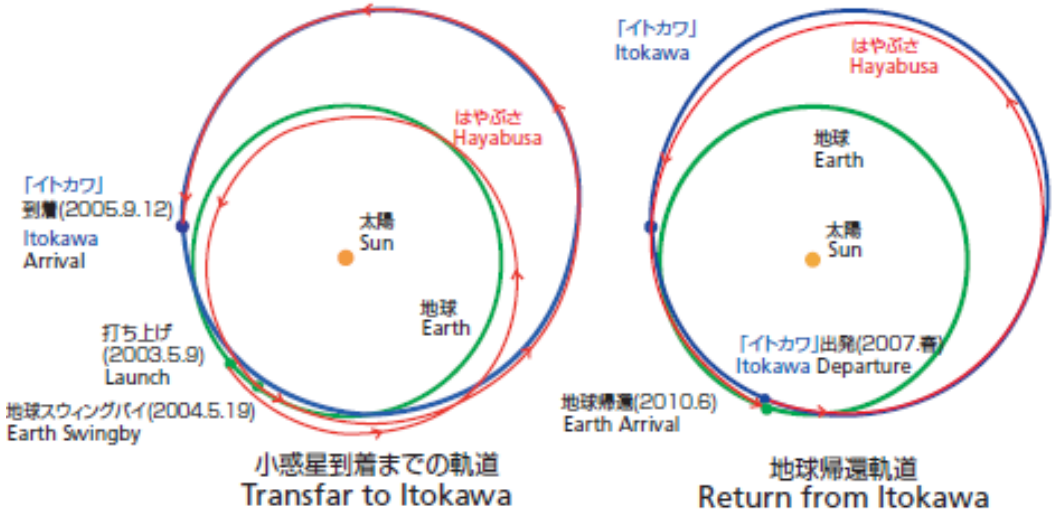

Figure 7. Hayabusa transfers to and from the Itokawa asteroid [9]

The spacecraft performed an Earth swingby and an interstellar flight with newly developed ion engines and it achieved the world's first simultaneous operation of three ion engines. It used xenon ions generated by microwave electron cyclotron resonance and a carbon composite material, resistant to erosion, for its acceleration grid. Since these engines do not use the discharge electrodes for

plasma generation, there is no concern of deterioration, and long life and high reliability is expected, which was successfully tested thorough the Hayabusa mission [9].

3.1.3. Future applications of low-thrust propulsion

After successfully achieving missions that test new technologies like the low-thrust propulsion for interplanetary missions, space agencies have been developing new missions that will use this type of technology:

- Space Technology 7 (ST7)
- BepiColombo
- Hayabusa 2

Space Technology 7 (ST7) is a project of the NASA's New Millenium Program and, when launched (2011-2013), it will do the first use of colloid thrusters in space. ST7 will flight test the Disturbance Reduction System (DRS), a system-level technology designed to demonstrate trajectory control and formation-flying spacecraft position control, which will feed into future missions such as the joint ESA and NASA Laser Interferometer Space Antenna (LISA) mission.

BepiColombo is a joint mission of the European Space Agency (ESA) and the Japan Aerospace Exploration Agency (JAXA) to the planet Mercury, expected to be launched in 2014.
This project consists of two orbiters, the Mercury Planetary Orbiter (MPO) will observe the surface and interior, whereas the Mercury Magnetospheric Orbiter (MMO) will observe the magnetic field and the magnetosphere. An artist's impression of the two spacecraft is shown in figure 8.
The spacecraft will have a six year interplanetary cruise to Mercury using solar-electric propulsion and gravity assists from the Moon, Earth, Venus and a final gravity capture at Mercury.
When approaching Mercury in 2020, the transfer module will be separated and the composite spacecraft will use rocket engines and a technique called 'weak stability boundary capture' to bring it into polar orbit around the planet. When the MMO orbit is reached, the MPO will separate and lower its altitude to its own operational orbit. Observations from orbit will be taken for at least one Earth year with the possibility of extension.

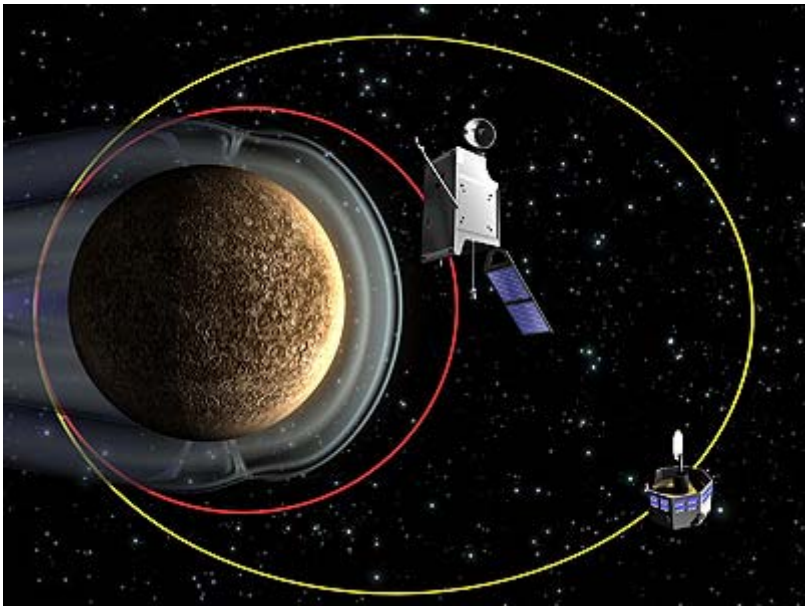

Figure 8. Artist's view of the two BepiColombo spacecrafts [10]

After having demonstrated various exploration technologies in the mission Hayabusa, JAXA has started developing a follow-up mission named at the moment Hayabusa 2 that will strengthen the weak points of its predecessor in order to dodge the problems that Hayabusa encountered during its journey.
Like its Hayabusa, it will visit a near Earth object, but in this case the target is planned to be a C-type asteroid, a type of objects which are mostly distributed around the outer side of the asteroid belt and are thought to contain more organic or hydrated minerals, samples that could be an important clue to investigate the origin of life on Earth. The candidate target is an asteroid with the provisional designation of 1999JU3.
The spacecraft is intended to be launched in the first half of 2010's. It will cruise toward the target using a combination of ion engines and an Earth swingby like Hayabusa. Once it arrived at the target it will conduct scientific investigation and collect the surface samples. After a stay of several months, it will leave the target to start the ion-engine-aided journey again, and bring back the capsule containing samples to the Earth after several years.

Apart from using low-thrust propulsion thrusters as primary propulsion systems in interplanetary missions, other different type of missions will continue using them as secondary propulsion system for other purposes such as low orbit transfers, attitude adjustments or drag compensation for low earth orbits. For example, the LISA Pathfinder spacecraft, designed by ESA and to be launched in 2013, will use colloid thrusters and FEEP for very precise altitude control.

In the domain of interplanetary flight, electric propulsion offers much more substantial advantages over chemical systems, which extend in several important

cases to enabling missions that simply could not be performed by means of any other reasonably projected propulsion technology and this has to be taken into account when planning future feasible interplanetary missions. Some examples include heavy cargo and/or piloted missions to Mars and the outer planets, many unpiloted probes beyond the solar system and out of the ecliptic plane or travelling to multiple targets with a single spacecraft in order to save in spacecraft and instruments development and launch costs.

The technology drivers for new electric propulsion thrusters include: ability to use available power (high maximum power with large throttle range), increased total throughput capability, and lower cost systems and integration.

As the available power and performance demand continue increasing, electric propulsion technologies will continue to supplant chemical alternatives for a wide range of missions. The technology will continue to focus on developing lower cost propulsion systems with higher power and longer lifetime capabilities.

3.1.4. Comparison between Inner and Outer Solar System applications

Nearly all of the above-mentioned missions take place in the Inner Solar System, (near-Earth objects or inner planets). However, this does not mean that the low-thrust applications are limited to these mission scenarios. There are plenty of studies that consider low-thrust propulsion as a feasible option for missions to the Outer Solar System [3]. Given that solar electric propulsion is not an efficient propulsion system at large distances from the Sun where the solar flux becomes weak, nuclear-powered propulsion is the alternative.

Multi-gravity assist transfers to reduce fuel and transfer time are common for both outer and inner Solar System missions, the former typically using gravity assists at Venus and Earth before proceeding to Jupiter or beyond. In this case, solar electric propulsion could be used to raise the aphelion of a heliocentric transfer to an outer planet and then using a nuclear electric propulsion system for capture.

As an example mission profile, a low-thrust mission to Pluto with Earth and Jupiter gravity assists and a 14-year mission constraint requires around 17km/sec of low-thrust ΔV [3]. Such amount can be reduced by increasing the number of Earth gravity assists, but in either scenario the ΔV is quite large even for low-thrust propulsion, and high specific impulse would be required. If the transfer time constraint is relaxed by a few years, the ΔV can be relatively smaller, which emphasizes the characteristic of high transfer times for missions using low-thrust propulsion systems.

3.2. Mars Exploration

Mars was the first planet of the Solar System targeted as a space mission. After several failures, the first close-up picture was taken in 1965 and since then there have been plenty of missions. The exploration of Mars has always played an important role in the space exploration programs of the United States, the Soviet Union, Europe and Japan. A great number of robots, orbiters, landers and rovers have been launched toward Mars since the 1960s in order to:

- gather data about the current conditions of the planet
- try to answer questions about the history of the planet
- yield further insight into the past and the future of the Earth
- prepare a possible human mission to Mars

3.2.1. Past and current Mars Missions

Table 1 shows the missions carried out to Mars since the early 1990s with the following details:

- Space Agency (SA)
- Name of the mission
- Date of launch
- Date of Mars arrival
- Date of end of mission or if it is currently operational (OP)
- Elements of the mission: orbiter (O), lander (L), hard lander (HL), rover (R), or if the mission partly implies a Mars gravity assist (GA)
- If it was a success (S) or a failure (F)

SA	Mission	Launch	Arrival	End	El.	S/F
NASA	**Mars Observer**	25-09-92	24-08-93	21-08-93	O	F
NASA	**Mars Global Surveyor**	07-11-96	11-09-97	05-11-96	O	S
Russia	**Mars 96**	16-11-96	-	17-11-96	O-L	F
NASA	**Mars Pathfinder**	04-12-96	04-07-97	27-09-97	L-R	S
Japan	**Nozomi (Planet-B)**	03-07-98	-	09-12-03	O	F
NASA	**Mars Climate Orbiter**	11-12-98	23-09-99	23-09-99	O	F
NASA	**Mars Polar Lander**	03-01-99	03-12-99	03-12-99	L	F
	Deep Space 2 (DS2)				HL	
NASA	**2001 Mars Odissey**	07-04-01	24-10-01	OP	O	S
ESA	**Mars Express**	02-06-03	25-12-03	OP	O	S
	Beagle 2			06-02-04	L	F
NASA	**MER-A Spirit**	10-06-03	04-01-04	03-10	R	S
NASA	**MER-B Opportunity**	07-07-03	25-01-04	OP	R	S
ESA	**Rosetta**	02-03-04	25-02-07	OP	GA	S

NASA	**Mars Reconnaissance Orbiter**	12-08-05	10-03-06	OP	O	S
NASA	**Phoenix**	04-08-07	25-05-08	10-11-08	L	S
NASA	**Dawn**	27-09-07	17-02-09	OP	GA	S

Table 1. Mars Missions since the early 1990s [11]

The causes of the failure or the results of the success are described in table 2.

Mission	Results of the mission
Mars Observer	F - Contact was lost few days before arrival
Mars Global Surveyor	S - Took more images than any previous mission
Mars 96	F - Failure at launch
Mars Pathfinder	S - Technology demonstrations carried out successfully
Nozomi (Planet-B)	F - Never entered orbit due to on route difficulties
Mars Climate Orbiter	F - Crashed on surface due to measure problems
Mars Polar Lander	F- Crashed on surface due to inadequate hardware testing
Deep Space 2 (DS2)	
2001 Mars Odissey	S - High resolution images of Mars
Mars Express	S - Imaging Mars in detail
Beagle 2	F - Contact was lost in December 2003
MER-A Spirit	S - Operating lifetime 15 times more high than expected
MER-B Opportunity	S - Operating lifetime 15 times more high than expected
Rosetta	S - Mars gravity assist as part of its trajectory to an asteroid
Mars Reconnaissance Orbiter	S - Returned more than 26 terabits of data
Phoenix	S - Returned more than 25 gigabits of data
Dawn	S - Mars gravity assist as part of its trajectory to its target

Table 2. Results of Mars Missions since the early 1990s [11]

By and large, the vast majority of past and current Mars Missions uses chemical thrusters as primary and secondary propulsion systems for transportation to Mars and initial Mars orbit insertion. Although they allow short transfer time, these chemical rockets are relatively inefficient, what puts an upper limit on the mass of payload that can be delivered to Mars.

The transfer between Earth and Mars is normally a type 1 trajectory (less than 180 degrees around the Sun; faster but requiring more energy) or type 2 trajectory (more than 180 degrees around the Sun) with several Trajectory Correction Manoeuvres (TCMs) previously scheduled and designed, with a transfer time between six and ten months approximately depending on the mission.

As an example, the trajectory of Mars 2001 Odissey is shown in figure 9.

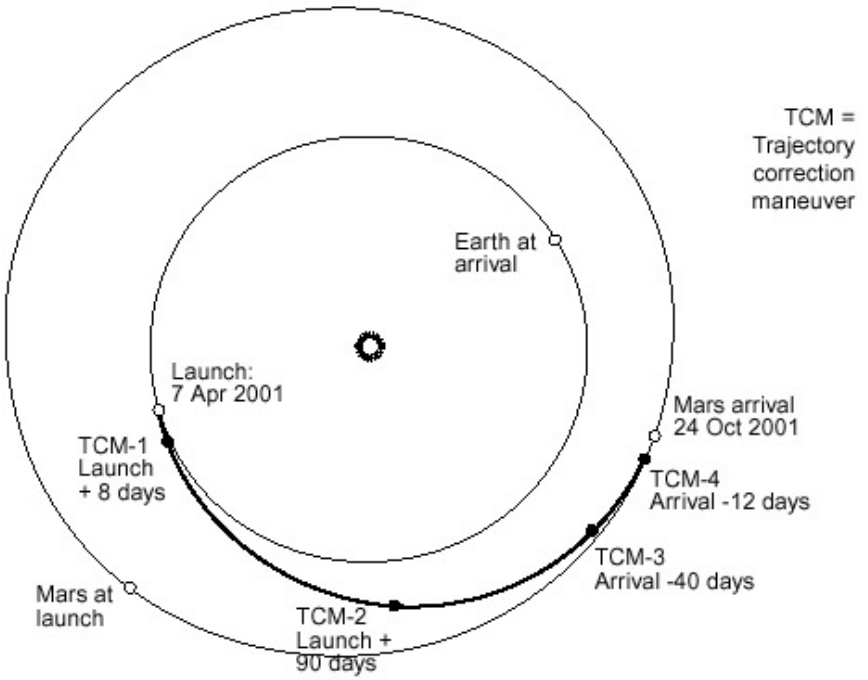

Figure 9. Mars 2001 Odyssey Earth-Mars trajectory [12]

3.2.2. Future Missions

In table 3 the future mission to Mars that are currently under planning or development are listed detailing the owner country, the estimated launch date and the elements of the mission.

Country	Mission	Estimated Launch	Elements
USA	**MSL Curiosity**	End 2011	Rover
Russia	**Fobos-Grunt**	End 2011	Orbiter, lander, Fobos sample return
China	**Yinghuo-1**	End 2011 (onboard Fobos-Grunt)	Orbiter
Finland	**MetNet**	2011-2019	Multi-lander network
Canada	**Northern Light**	2012	Lander and rover
USA	**MAVEN**	2013	Orbiter
ESA-USA	**Mars Trace Gas Orbiter**	2016	Orbiter
ESA-USA	**ExoMars**	2016 (onboard TGO)	Static lander and rover
India	**Mars Mission**	2015-2019	Orbiter
ESA-USA	**Mars sample return**	2020s	Orbiter, lander, rover and sample return

Table 3. Future Mars Missions [11]

In the case of the Fobos-Grunt mission, during its conceptual design before 2004, there were two alternatives for the interplanetary station, with and without electric jet propulsion. Finally the typically chemical propulsion system used in other similar missions was chosen.

Nowadays, missions that are still in their conceptual design could choose low-thrust propulsion systems as primary propulsion system as these technologies continue to increase their efficiency and reliability and allow an increase in payload mass to Martian orbit while keeping current cost levels.

4. BASIC CODE

As a prior work in terms of Earth-to-Mars transfers with low thrust propulsion and as starting point of this project, the basic code is presented in this chapter.
Part of the code is based on the article "Utilization of Ion Propulsion for Mars Orbiters" [13].

The code integrates a non-optimized Earth-to-Mars trajectory using low-thrust propulsion with the spacecraft initially in a geostationary transfer orbit (GTO). A heliocentric model with the Earth and Mars gravitational perturbations and additional force vectors due to thrust is assumed and represented in a heliocentric frame of reference. The model is also simplified by considering that both Mars and Earth are in circular, coplanar orbits, as shown in figure 10. Other simplifications will be mentioned later in the description of the code.

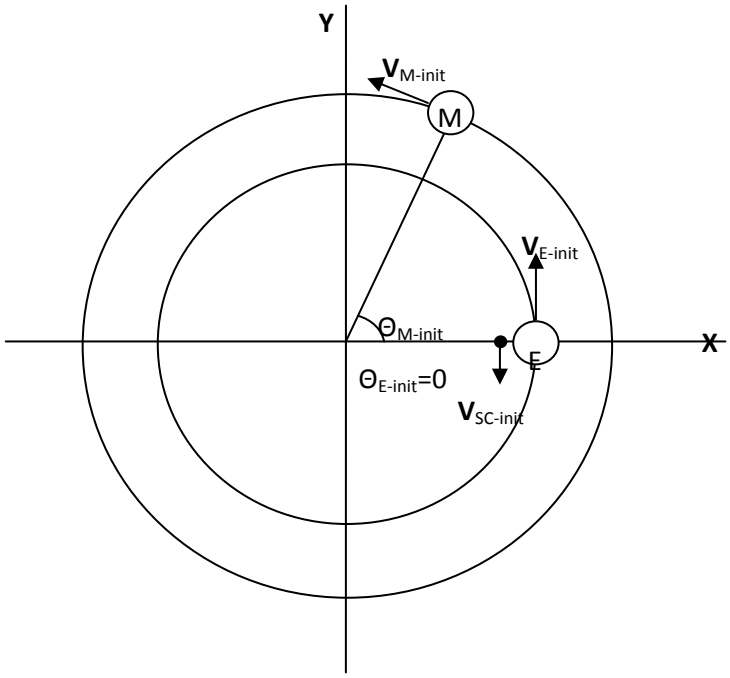

Figure 10. Earth, Mars and spacecraft predetermined initial states

The angle θ_M shown in figure 10 is the heliocentric anomaly of Mars, the angle between the radius vector of Mars and the positive unit vector on the **X** axis of the heliocentric frame of reference, whereas θ_E, zero in figure 10, is the heliocentric anomaly of the Earth, the angle between the radius vector of Earth and the positive unit vector on the **X** axis.

The basic code first finds an optimal initial Mars angle $\theta_{M\text{-}init}$ by iteration. Starting from an initial guess of $\theta_{M\text{-}init}$, the code iterates the number of times imposed by

the user until finding the final value. How the code works in order to find this angle will be later explained in the integration function chapter (4.3).
After finding the optimal initial Mars angle, the final trajectory is computed and the results are calculated and shown in a sequence of figures.

The code is programmed in MATLAB ® and is divided into five functions:

- "EartMarsLT" (main)
- "EQmotion" (Equations of motion)
- "IntFcn" (integrate function)
- "deg2rad"
- "rad2deg"

4.1 Main function "EarthMarsLT"

Firstly, all the variables are defined as global so that a single copy of each variable is shared in all functions, that is to say, any assignment to each variable is available in all functions.
Next, the input data for the program is defined; when running the program, a table appears at the beginning and the user can modify the values of all the parameters. Table 4 shows all the input data along with their predetermined values.

Parameter	Predetermined Value
Electric Power [W]	5886
Specific Impulse [s]	3000
Thrust Efficiency [-]	0,5
Initial Mass [kg]	1500
Perigee Altitude [km]	300
Semi-major Axis [km]	24582
Initial True Anomaly [deg]	180
Final Altitude [km]	400
Transfer Time [yrs]	4
Initial Earth Angle [deg]	0
Starting Thrust at n S.I.	8
Tolerance on Energy [ ]	-0,1
Final Thrust anomaly [deg]	45
Number of Iterations	2

Table 4. Input Data

With these predetermined values, the characteristics of the engines and initial states of the spacecraft are assumed. Moreover, the Earth is considered to lie

initially on the **X** axis of the heliocentric coordinate system, as shown in figure 10, and the deceleration begins when the spacecraft is inside eight times the Martian sphere of influence so that the spacecraft can slow down enough to enter orbit instead of reaching a hyperbolic trajectory around Mars.
After defining the input data, the Mars angle is assumed to be of 90 degrees, as an initial guess for the later iteration.

Basic constants are also defined. These are the standard gravitational parameter μ, mass M and radius R for Sun, Earth and Mars, the radii of the orbits around the Sun of Earth a_E and Mars a_M, and the radii of the spheres of influence [3] of Earth SOI_E,

$$SOI_E = a_E \left(\frac{M_E}{M_{sun}} \right)^{\frac{2}{5}}, \tag{1a}$$

and Mars SOI_M,

$$SOI_M = a_M \left(\frac{M_M}{M_{sun}} \right)^{\frac{2}{5}}. \tag{1b}$$

Then, all the constants and parameters needed are expressed in adimensional units for the calculation part of the code. This is done by previously defining the following units corresponding to distance, time and mass variables, respectively:

$$DU = 1 \text{ au} = 149597870 \text{ km},$$

$$TU = \sqrt{\frac{DU^3}{\mu_{sun}}} \ [\text{s}], \tag{2}$$

$$MU = M_{initial} \ [\text{kg}].$$

The thrust Th is calculated as

$$Th = \frac{2\eta P}{I_{sp} g_0} \cdot \frac{TU^2}{MU \cdot DU} \cdot 10^{-3}, \tag{3}$$

and all the angles are given in radians.

Once all the input data and constants are correctly defined, the iteration part of the code begins in order to find the optimal initial Mars angle. First, a flag parameter is set to 1, which means that the simulation will stop at the first intersection with the orbit of Mars and the Mars initial angle will be identified.

After this, the initial states of Earth, Mars and spacecraft are defined:

$$
\begin{aligned}
&\mathbf{R}_{E-init} = \left[a_E \cos\theta_{E-init} \, , \, a_E \sin\theta_{E-init} \right], \\
&\mathbf{V}_{E-init} = \left[-\sqrt{\frac{\mu_{sun}}{a_E}} \sin\theta_{E-init} \, , \, \sqrt{\frac{\mu_{sun}}{a_E}} \cos\theta_{E-init} \right], \\
&\mathbf{R}_{M-init} = \left[a_M \cos\theta_{M-init} \, , \, a_M \sin\theta_{M-init} \right], \\
&\mathbf{V}_{M-init} = \left[-\sqrt{\frac{\mu_{sun}}{a_M}} \sin\theta_{M-init} \, , \, \sqrt{\frac{\mu_{sun}}{a_M}} \cos\theta_{M-init} \right], \\
&\mathbf{R}_{SC-init} = \mathbf{R}_{E-init} + \left[\left(R_E + h_0\right) \cos\theta_{SC-init} \, , \, \left(R_E + h_0\right) \sin\theta_{SC-init} \right], \\
&\mathbf{V}_{SC-init} = \mathbf{V}_{E-init} + \left[-\sqrt{\frac{2\mu_E}{\left(R_E + h_0\right)} - \frac{\mu_E}{a_0}} \sin\theta_{SC-init} \, , \, \sqrt{\frac{2\mu_E}{\left(R_E + h_0\right)} - \frac{\mu_E}{a_0}} \cos\theta_{SC-init} \right].
\end{aligned}
\tag{4}
$$

Taking into account the predetermined values shown in table 4 and the assumption of circular and coplanar orbits for Earth and Mars, the initial states are shown in figure 10.

As for the spacecraft, it is assumed that its transfer begins at perigee on a GTO at an altitude of 300km. This and the fact that the predetermined true anomaly is 180 degrees mean the spacecraft is initially on the **X** axis between the Earth and the Sun. This allows the spacecraft to leave the Earth in the direction of the Earth's orbit.

There are 13 variables representing initial conditions: mass, coordinates and velocities of the spacecraft and coordinates and velocities of the planets:

$$
\mathbf{x}_0 = \left[\mathbf{R}_{SC-init}, \mathbf{V}_{SC-init}, M_{init}, \mathbf{R}_{E-init}, \mathbf{V}_{E-init}, \mathbf{R}_{M-init}, \mathbf{V}_{M-init} \right]. \tag{5}
$$

After defining all the initial states, the integrating function used is the MATLAB ® function ODE23, which is an initial value ODE problem solver. It uses the Runge-Kutta method to solve non-stiff differential equations. The relative and absolute error tolerances are set both to 10^{-4}. This part of the code is later explained as it contains auxiliary functions of integration properties and equations of motion that will be detailed in the next chapters.
Following the iteration process, the initial states are calculated again, but with the newly found initial Mars angle. Then, the final trajectory is computed using the

ODE23 function in order to find the mass of the spacecraft and the position and velocity of the planets and the spacecraft during the whole transfer.

Finally, once all the calculations of the transfer are complete, a group of plots are shown illustrating the characteristics of the computed Earth-to-Mars transfer. The plots are summarized in table 5.

Plots Summary
Heliocentric transfer
Position of the spacecraft with respect to the Sun
Mars and Earth reference frame
Position of the spacecraft with respect to the Earth Position of the spacecraft with respect to Mars
Heliocentric frame
Heliocentric distance [km] vs time [days] Instantaneous mass [kg] vs time [days]
Earth frame
Geocentric distance [km] vs time [days] Geocentric velocity [km/s] vs time [days] Energy of the spacecraft with respect to Earth [km^2/s^2] vs time [days] Perigee radius around Earth [km] vs time [days] Apogee radius around Earth [km] vs time [days] Eccentricity at Earth vs time [days]
Mars frame
Marscentric distance [km] vs time [days] Marscentric velocity [km/s] vs time [days] Energy of the spacecraft with respect to Mars [km^2/s^2] vs time [days] Perigee radius around Mars [km] vs time [days] Apogee radius around Mars [km] vs time [days] Eccentricity at Mars vs time [days]

Table 5. Plots Summary

4.2 Equations of motion "EQmotion"

This function is an input for the ODE23 MATLAB ® function, along with the time span, the initial conditions and the integration options. It calculates accelerations, velocities and flow mass that will be integrated in the main function to know the position, velocity and mass of the spacecraft as well as position and velocity of Earth and Mars all along the transfer. The vectors **y** and **y**' are formed with such variables and their time derivatives, respectively:

$$\mathbf{y} = \left[\mathbf{R}_{SC}, \mathbf{V}_{SC}, M, \mathbf{R}_E, \mathbf{V}_E, \mathbf{R}_M, \mathbf{V}_M \right], \tag{6}$$

$$\mathbf{y'} = \frac{d\mathbf{y}}{dt} = \left[\mathbf{V}_{SC}, \mathbf{A}_{SC}, \dot{M}, \mathbf{V}_E, \mathbf{A}_E, \mathbf{V}_M, \mathbf{A}_M \right]. \tag{7}$$

The acceleration of the spacecraft has two components: the gravitational acceleration $\mathbf{A}_{SC-grav}$ and the thrust acceleration $\mathbf{A}_{SC-F}$.

The first comes from the gravitational attractions of the Sun, the Earth and Mars:

$$\mathbf{A}_{SC-grav} = -\mu_{sun} \frac{\mathbf{R}_{SC}}{\left|\mathbf{R}_{SC}\right|^3} + \mu_E \left(\frac{\mathbf{R}_E - \mathbf{R}_{SC}}{\left|\mathbf{R}_E - \mathbf{R}_{SC}\right|^3} - \frac{\mathbf{R}_E}{\left|\mathbf{R}_E\right|^3} \right) + \mu_M \left(\frac{\mathbf{R}_M - \mathbf{R}_{SC}}{\left|\mathbf{R}_M - \mathbf{R}_{SC}\right|^3} - \frac{\mathbf{R}_M}{\left|\mathbf{R}_M\right|^3} \right). \tag{8}$$

From left to right, the several terms of the equation are due to the force of the Sun on the spacecraft, the force of the Earth on the spacecraft, the force of the Earth on the Sun, the force of Mars on the spacecraft and the force of Mars on the Sun, respectively.

The acceleration due to thrust changes depending on the specific phase of the transfer, where the position and other parameters of the spacecraft change the orientation of the thrust vector $\hat{\mathbf{e}}_f$ as the spacecraft orbits a given body. For each of these situations, the thrust acceleration $\mathbf{A}_{SC-F}$ and mass variation $\dot{M}$ of the spacecraft are calculated:

$$\mathbf{A}_{SC-F} = \frac{Th}{M} \hat{\mathbf{e}}_{\mathbf{f}}, \tag{9}$$

$$\dot{M} = -\frac{Th}{I_{sp} g_0}. \tag{10}$$

Before calculating these parameters for the different transfer phases, some preliminary orbital parameters are needed:

- Specific mechanical energy of the heliocentric transfer ξ and relative to Mars ξ_M,

$$\xi = \frac{\left|\mathbf{V}_{SC}\right|^2}{2} - \frac{\mu_{sun}}{\left|\mathbf{R}_{SC}\right|}, \quad \xi_M = \frac{\left|\mathbf{V}_{SC} - \mathbf{V}_M\right|^2}{2} - \frac{\mu_M}{\left|\mathbf{R}_{SC} - \mathbf{R}_M\right|}. \tag{11}$$

- Eccentricity e:

$$e = |\mathbf{e}| = \left| \frac{1}{\mu_{sun}} \left[\left(|V|^2 - \frac{\mu_{sun}}{|R|} \right) R - (R \cdot V) V \right] \right|. \quad (12)$$

- Apogee radius R_a of the transfer:

$$R_a = -\frac{\mu_{sun}(1+e)}{2\xi}. \quad (13)$$

- Periareion $R_{p\text{-}M}$ and apoareion $R_{a\text{-}M}$ radius, where the periareion is the point at which the spacecraft orbit makes its closest approach to Mars and the apoareion the point of the farthest approach:

$$R_{p-M} = -\frac{\mu_M(1-e_M)}{2\xi_M}, \quad (14)$$

$$R_{a-M} = -\frac{\mu_M(1+e_M)}{2\xi_M}. \quad (15)$$

- True ν_M, eccentric E_M and mean M_M anomalies around Mars:

$$\nu_M = \cos^{-1}\left(\frac{e_M \cdot (R_{SC} - R_M)}{e_M \cdot |(R_{SC} - R_M)|} \right), \quad (16)$$

$$\tan E_M = \frac{(1-e_M^2)\sin\nu_M}{e_M + \cos\nu_M}, \quad (17)$$

$$M_M = E_M - e_M \sin E_M. \quad (18)$$

There are three different transfer phases to consider:

- Earth escape
- Heliocentric transfer
- Mars capture

When orbiting the Earth, the thrust of the spacecraft is tangential, which means the thrust vector is directed parallel to the spacecraft's velocity vector relative to Earth in order to allow the spacecraft to accelerate away from Earth after launch,

$$\hat{\mathbf{e}}_f = \frac{V_{SC} - V_E}{|V_{SC} - V_E|}. \quad (19)$$

This tangential thrust continues until the spacecraft leaves the Earth's sphere of influence or the apogee radius of the heliocentric transfer is greater than the orbital radius of Mars.

When orbiting the Sun (aphelion smaller than the orbital radius of Mars), the thrust is circumpherential, that is to say, the thrust vector is directed perpendicular to the radius vector:

$$\hat{\mathbf{e}}_{\mathbf{f}} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{SC-X} \\ R_{SC-Y} \\ 0 \end{bmatrix} \frac{1}{|\mathbf{R}_{SC}|}. \tag{20}$$

This has two consequences: the tangential velocity increases and the radial velocity slightly decreases.

Finally, the Mars capture has more than one situation to be considered, and it begins when the spacecraft is inside n times the Martian sphere of influence.
First, if the periareion radius is smaller than the Mars equatorial radius plus the desired final altitude, the spacecraft needs to speed up in order to avoid entering the Martian atmosphere and burning up. Therefore, the thrust is circumpherential (Eq. 20), but in this case perpendicular to the radius vector of the spacecraft relative to Mars ($\mathbf{R}_{SC} - \mathbf{R}_{M}$).

Second, if the specific mechanical energy relative to Mars (ξ_M in Eq. 11) is greater than zero, it means that the trajectory of the spacecraft around Mars is hyperbolic and an anti-tangential thrust is needed so that the spacecraft can slow down and be captured in an elliptic orbit around Mars. The anti-tangential thrust is achieved by directing the thrust opposite the velocity of the spacecraft relative to Mars:

$$\hat{\mathbf{e}}_{\mathbf{f}} = -\frac{\mathbf{V}_{SC} - \mathbf{V}_{M}}{|\mathbf{V}_{SC} - \mathbf{V}_{M}|}. \tag{21}$$

This thrust vector is also needed in two other situations that require the spacecraft to be slowed down in order to be captured. One is when the apoareion radius is greater than the radius of Mars plus the final altitude and the mean anomaly M_M is near periareion ($2\pi - dM < M_M < 2\pi + dM$ where dM is the final thrust anomaly, an input parameter); and the other when the periareion radius is greater than the Mars radius plus the final altitude.
The latter situation is defined when the specific mechanical energy relative to Mars is greater than a certain negative tolerance (given in input) and the mean

anomaly M_M is near apoareion (π-dM<M_M<π+dM). In this case the spacecraft requires a tangential thrust in order to be captured:

$$\hat{e}_f = \frac{V_{SC} - V_M}{|V_{SC} - V_M|}. \tag{22}$$

The case of no thrust is also considered when none of the previous situations applies.

Once the thrust acceleration has been determined on the basis of the above analysis, the acceleration of the spacecraft is calculated by adding the gravitational and thrust accelerations (Eqs. 8-9).

$$A_{SC} = A_{SC-grav} + A_{SC-F}. \tag{23}$$

The acceleration of Earth A_E and Mars A_M are simply modelled by the two-body problem.

$$A_E = \mu_{sun} \frac{R_E}{|R_E|^3}, \tag{24}$$

$$A_M = \mu_{sun} \frac{R_M}{|R_M|^3}. \tag{25}$$

4.3 Integration Function "IntFcn"

This function is called in the main function "EarthMarsLT" as input to the MATLAB® function ODESET that creates an Ordinary Differential Equations options structure for the ODE23 function that will solve the non-stiff differential equations. The function is called by the solver after each time step.

First, a wait bar is created in order to be displayed every time the solver is in process, i.e., when finding the initial Mars angle during the iteration process and when computing the final trajectory.

The inputs of the function are the time and the orbital parameters (position, velocity and mass) shown in Eq. 4. With these inputs, other orbital parameters are calculated after each time step and they are all saved in a MATLAB ® structure array called *Parameters*. The orbital parameters calculated after each time step are:

- Specific mechanical energy of the transfer (ξ in Eq. 11)
- Heliocentric eccentricity (Eq. 12)
- Apogee radius of the transfer (Eq. 13)
- Velocity at Mars $\mathbf{V}_{atM}$:

$$\mathbf{V}_{atM} = \sqrt{2\left(\frac{\mu_{sun}}{a_M} + \xi\right)}. \tag{26}$$

- True anomaly of the heliocentric transfer ν:

$$\nu = \cos^{-1}\left(\frac{\mathbf{e}\cdot\mathbf{R}_{SC}}{\mathbf{e}\cdot\left|\mathbf{R}_{SC}\right|}\right). \tag{27}$$

- Specific mechanical energy relative to Earth (from Eq. 11 substituting the subindex M with E)
- Eccentricity at Earth (from Eq. 12)
- Perigee and apogee radius (from Eqs. 14-15 substituting the subindex M with E)
- Specific mechanical energy relative to Mars (Eq. 11)
- Eccentricity at Mars (from Eq. 12)
- Periareion and apoareion radius (Eqs. 14-15)
- True, eccentric and mean anomalies around Mars (Eqs. 16-17-18)

Then, the following part of the code in this function is to identify the initial Mars angle, $\theta_{M\text{-}init}$ in figure 10. It is an if-loop that stops the trajectory propagation at the first intersection with the orbit of Mars, computes the time required by the spacecraft to reach such position and moves the position of Mars backward in time.

It starts the calculation when the heliocentric distance of the spacecraft is greater than the Mars orbital radius minus the radius of the sphere of influence of Mars. This is not the spacecraft arriving at Mars, but the spacecraft intersecting an orbit of radius a_M-SOI_M. Moreover, a flag parameter has to be equal to one, which means that the angle has not been found yet and the iteration continues at each time step. If the above-mentioned is affirmative then a new structure array, called *Opt*, is created with the following parameters at this point of the trajectory:

- time t required to reach that position,
- position of the spacecraft $\mathbf{R}_{SC}$,
- position of Mars $\mathbf{R}_{M}$,

- heliocentric anomaly of the spacecraft θ_{SC},
- heliocentric anomaly of Mars θ_M,
- initial Mars angle θ_0.

The heliocentric anomalies are calculated taking into account they are the angles between the radius vector and the positive unit vector on the **X** axis.

The initial Mars angle is calculated by subtracting the angle that Mars has covered during the time t from the heliocentric anomaly of the spacecraft:

$$\theta_0 = \theta_{SC} - \varpi_M \cdot t = \theta_{SC} - \frac{\sqrt{\frac{\mu_{sun}}{a_M}}}{a_M} \cdot t \,. \tag{28}$$

By this calculation, it is assumed that the heliocentric anomaly of the spacecraft θ_{SC} is the heliocentric anomaly of Mars θ_M and this position is moved backward in time in order to find the initial position of Mars θ_0 that will enable the spacecraft to reach the sphere of influence of Mars after the time t.
Once the angle is calculated, its value is shown in the screen after each iteration until the final value is found.
There are other screen messages set in the code that are checked at each time step. If the module of the radius vector of the spacecraft is smaller or equal to the Mars radius plus the final altitude the message "Target Altitude Achieved" is written. However, if the same parameter is smaller than the Mars radius the message will be "Impact with the surface" and if the mass of the spacecraft is negative or equal to zero the message is "S/C mass finished".

4.4. Other auxiliary functions

The last two functions of the code are "deg2rad" and "rad2deg", which are auxiliary functions to turn the input angle from degrees to radians by multiplying it by 180/π and to turn it from radians to degrees by multiplying it by π/180.

4.5. Corrections and improvements

This chapter details some modifications of the basic code done in order to correct errors detected or improve the functionality of the program.

First, there is a problem with the calculation of the eccentric anomaly around Mars in the basic code. Eq. 17 is wrong because there should be a root in the

numerator, which means the value of the mean anomaly is also wrong and it could affect other parts of the program. The correct expression is:

$$\tan E_M = \frac{\sqrt{\left(1-e_M^2\right)}\sin \nu_M}{e_M + \cos \nu_M}. \tag{29}$$

However, this equation is only valid when the eccentricity is less than 1. In order to avoid mathematical problems, before calculating the eccentric and mean anomalies, the eccentricity value must be checked.
If eccentricity at Mars is less than 1 the previous equation for the eccentric anomaly and Eq. 18 for mean anomaly applies. But, if the eccentricity is greater than 1, the orbit is hyperbolic and the eccentric and mean anomalies must be calculated with the following equations:

$$\tanh E_M = \frac{\sqrt{\left(e_M^2-1\right)}\sin \nu_M}{e_M + \cos \nu_M}, \tag{30}$$

$$M_M = E_M - e_M \sinh E_M. \tag{31}$$

After these changes, comparing the results with the other version of the code there is no change in the initial Mars angle, but there are differences during the last phase of the transfer, arriving at Mars. Moreover, the elapsed running time of the program is significantly reduced, the results are shown approximately five times faster. Although this time could vary depending on the input data and the characteristics of the transfer.

Another modification of the code is to reduce the number of iterations when they are not necessary. The code stops iterating depending on the number of iterations the user has imposed. The predetermined value for number of iterations is set to n=2 and it could be thought that more iterations are necessary, but it is not true because after one step the initial Mars location is determined and the remaining steps are required only to refine the angle. Therefore, a tolerance has been added at the end of the for-loop of the main function of the code "EarthMarsLT", that stops the loop when the difference between the current initial Mars angle and the previous one calculated is smaller than 0,01 rad.
With this modification, even though the input for the number of iterations was a large number, the code will not iterate more than necessary and the elapsed time will be reduced.

4.6. Results and plots

In this chapter, the results provided by the basic code, with the corrections mentioned in the previous chapters, are presented.
Taking into account all the analysis previously described and the predetermined input data shown in table 4, the initial Mars angle after one iteration is -2,87 rad. And after the second and final iteration its value is:

$$\theta_0 = \theta_{M-init} = -2,8912\text{rad} = -165,65^{o}\,.$$

Something important to mention before presenting the plots is that they correspond to the simulation with the parameters of table 4, but with one difference, the time is set to six years. This change is made in order to see the spiralling around Mars. Before the code corrections the spiralling in this predetermined case occurred in less than four years, but after the corrections the transfer time has increased, which does not mean that this parameter will increase in all the scenarios, depending on the input data this fact can change.

Next, the plots summarized in table 5, which characterise the Earth-to-Mars transfer through its different transfer phases, are presented in a set of figures, from figure 11 to figure 26.

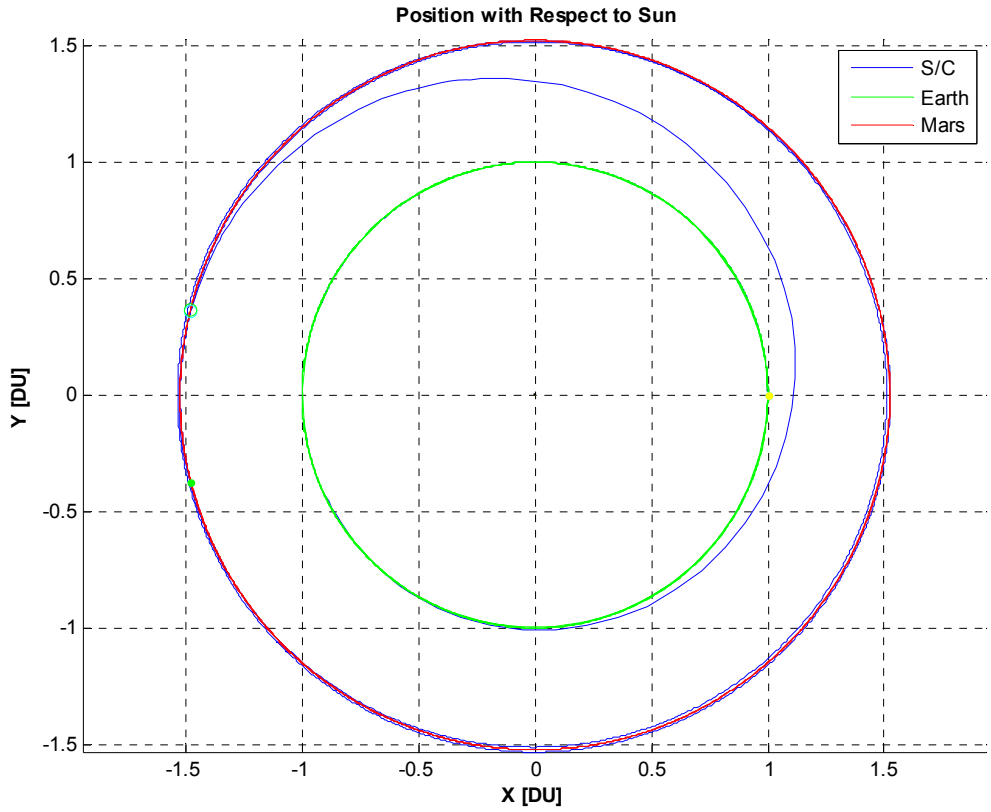

Figure 11. Heliocentric transfer

Figure 11 shows the heliocentric transfer of the spacecraft. The Earth orbit is coloured in green, Mars orbit in red and the transfer itself in blue. The yellow point situated in coordinates [1 0] is the initial position of the Earth, whereas the green point is the initial position of Mars, according to the calculated initial Mars angle of -165.65º. Other important positions shown in the figure are the position of Mars and the spacecraft when it enters the sphere of influence of Mars. This position of the spacecraft is represented by a cyan circle and Mars position at the same time with a green circle.

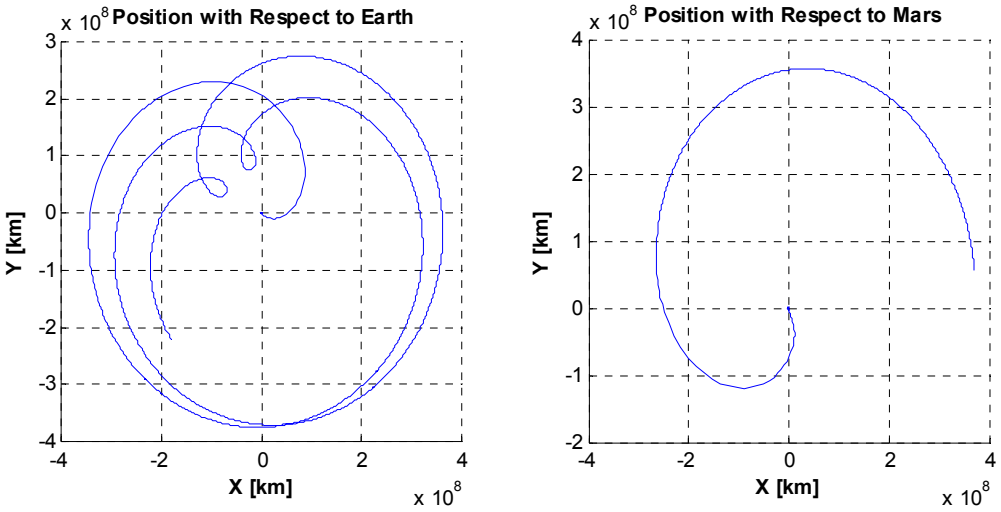

Figure 12. Spacecraft's position in Earth and Mars reference frame

If the left plot of figure 12 is zoomed close to the Earth's position the spiralling around Earth before escape can be seen more clearly, from its initial departing orbit to leaving Earth's sphere of influence, as shown in figure 13.

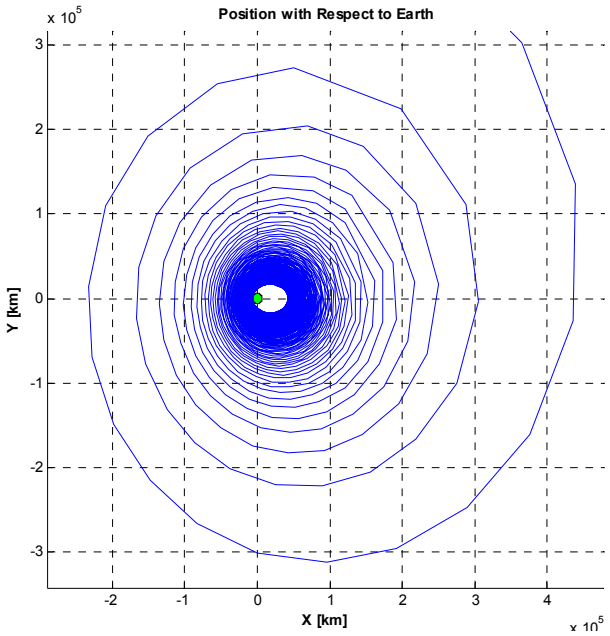

Figure 13. Spacecraft's position in Earth reference frame (Earth escape)

The same spiralling occurs around Mars at arrival, if the right plot of figure 12 is zoomed close to Mars position figure 14 is obtained where the plot displays the orbits around Mars until it reach its final position.

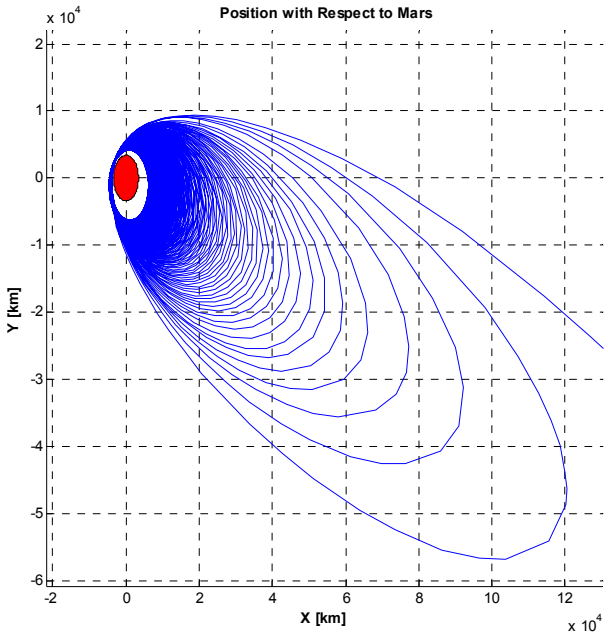

Figure 14. Spacecraft's position in Mars reference frame (Mars arrival)

Figure 15 displays the evolution of the heliocentric distance during all the transfer, where it can be seen that the transfer time between the sphere of influence of the Earth and the sphere of influence of Mars is not the whole six years of the simulation.

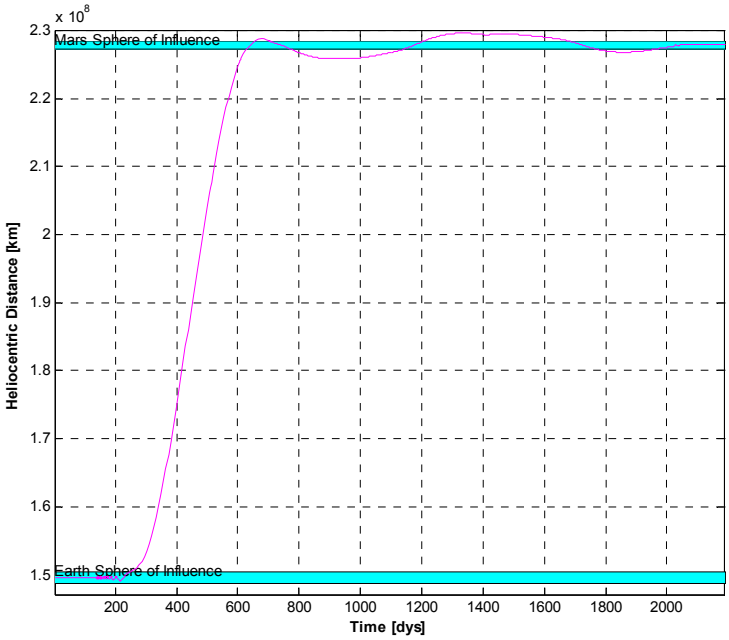

Figure 15. Heliocentric Distance

From what can be seen in figure 11 and figure 15, the spacecraft leaves Earth after practically one year, since the Earth initial position is on the positive **X** axis. Then, after arriving at Mars, the spacecraft continues following Mars orbit a significant period of time, which does not mean the Earth-to-Mars transfer time is so long, it is smaller, but the trajectory of the spacecraft has been computed throughout six years in this case in order to display in the plots the spiraling around Mars. This is also why some of the plots where time is represented have a maximum time of six years. This fact it is also commented after figure 26.

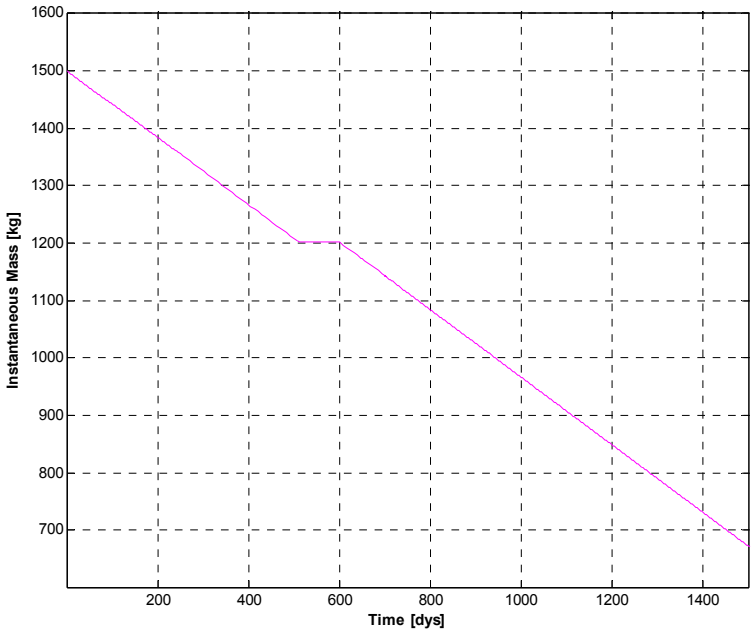

Figure 16. Instantaneous Mass

From figure 16 it can be said that the thruster is practically never switched off, except some days during the transfer where any of the conditions applied in the function "EQmotion" to calculate thrust acceleration are met.

Next, geocentric distance, geocentric velocity, energy with respect to Earth, perigee and apogee radius around Earth and eccentricity at Earth are represented in Earth reference frame in figure 17. The oscillations that appear at the end of the plots are consequence of the spiralling around Mars, and the two waves demonstrate that the spacecraft is following Mars orbit during that period of time.
Applying a zoom to some of the plots in figure 17 it is possible to check the variations of the parameters before Earth's escape (figures from 18 to 21).

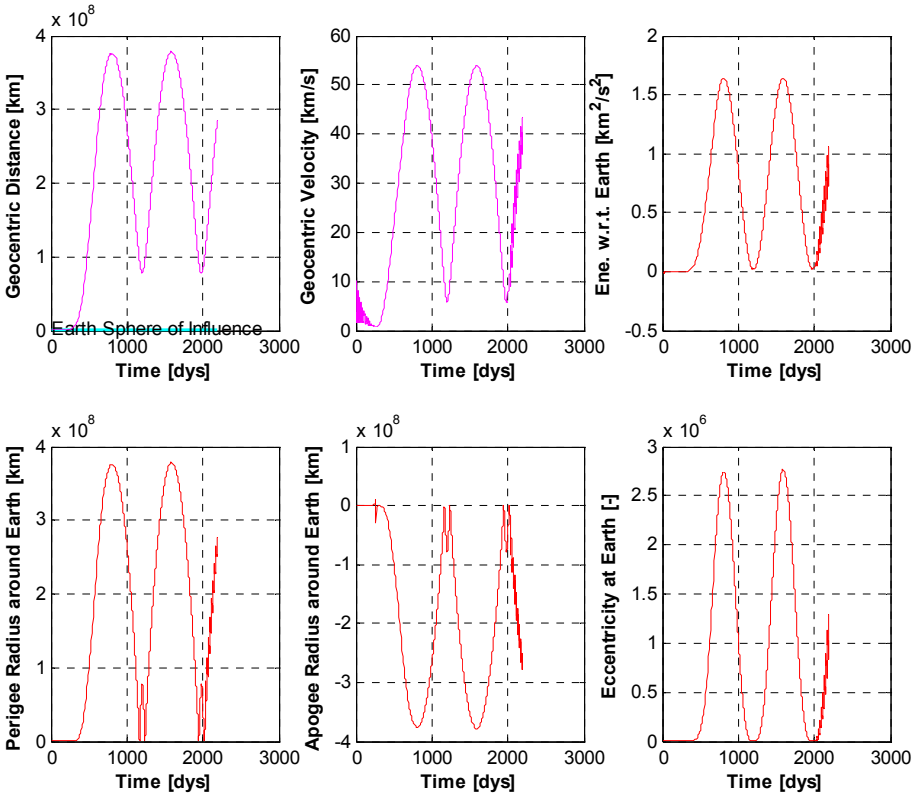

Figure 17. Parameters in Earth frame

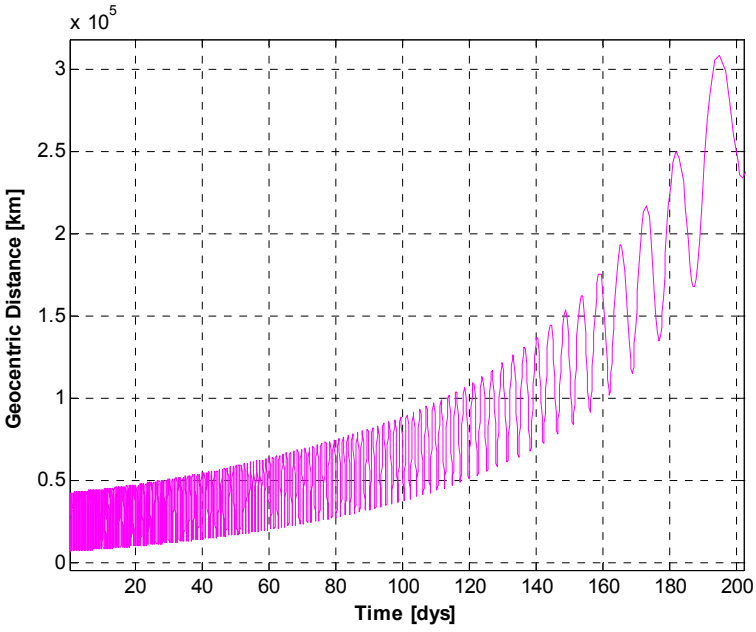

Figure 18. Geocentric Distance in Earth frame

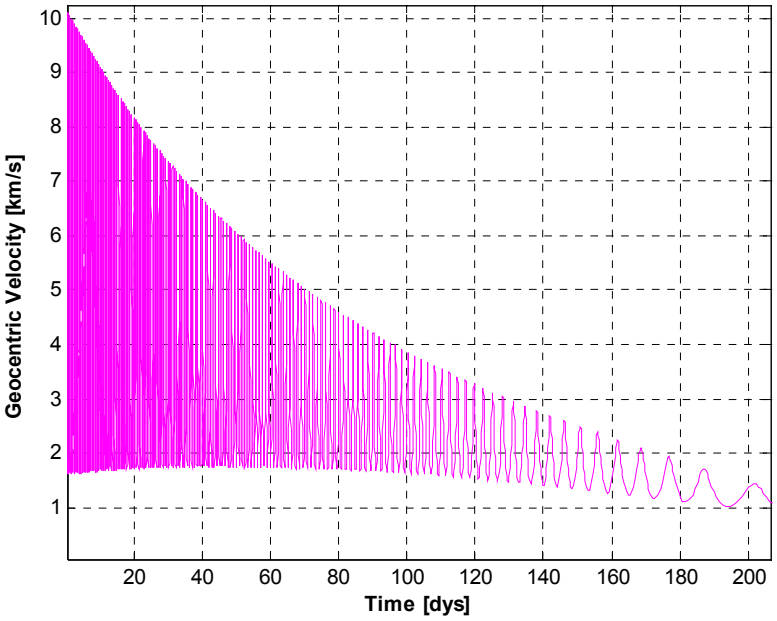

Figure 19. Geocentric Velocity in Earth frame

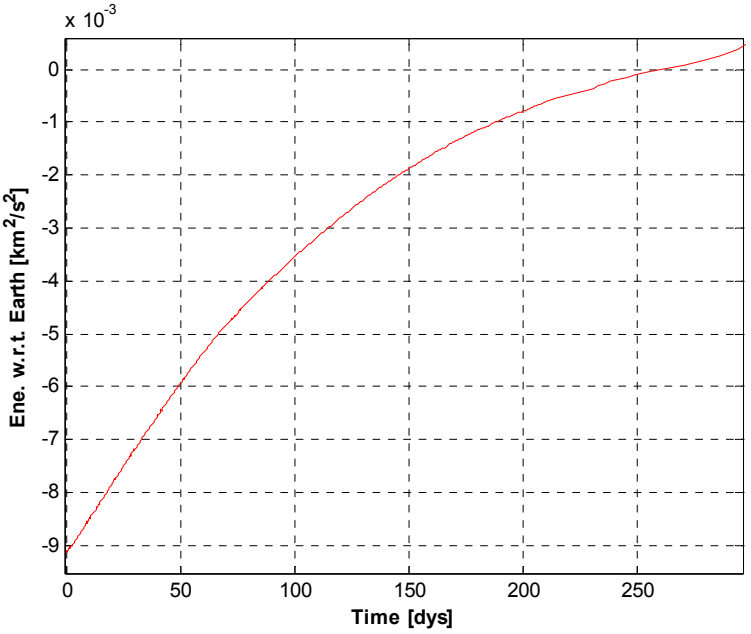

Figure 20. Energy with respect to Earth in Earth frame

Figure 18 and 19 show that there are oscillations in geocentric distance and velocity at the beginning of the transfer. These oscillations are reasonable because they are a reflection of the spiralling around the Earth before escape. They combine correctly into the energy relative to the Earth which increases steadily and without oscillations from negative to positive values as it can be seen in figure 20.

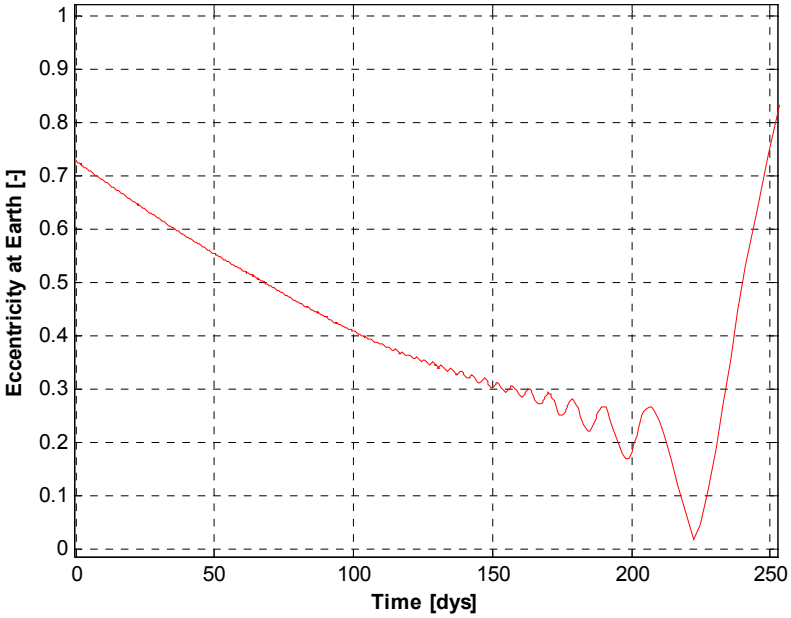

Figure 21. Eccentricity at Earth

The same parameters that are shown in figure 17, but with respect to Mars are shown in figure 22 for all the transfer and zoomed close to Mars from figure 23 to 26.

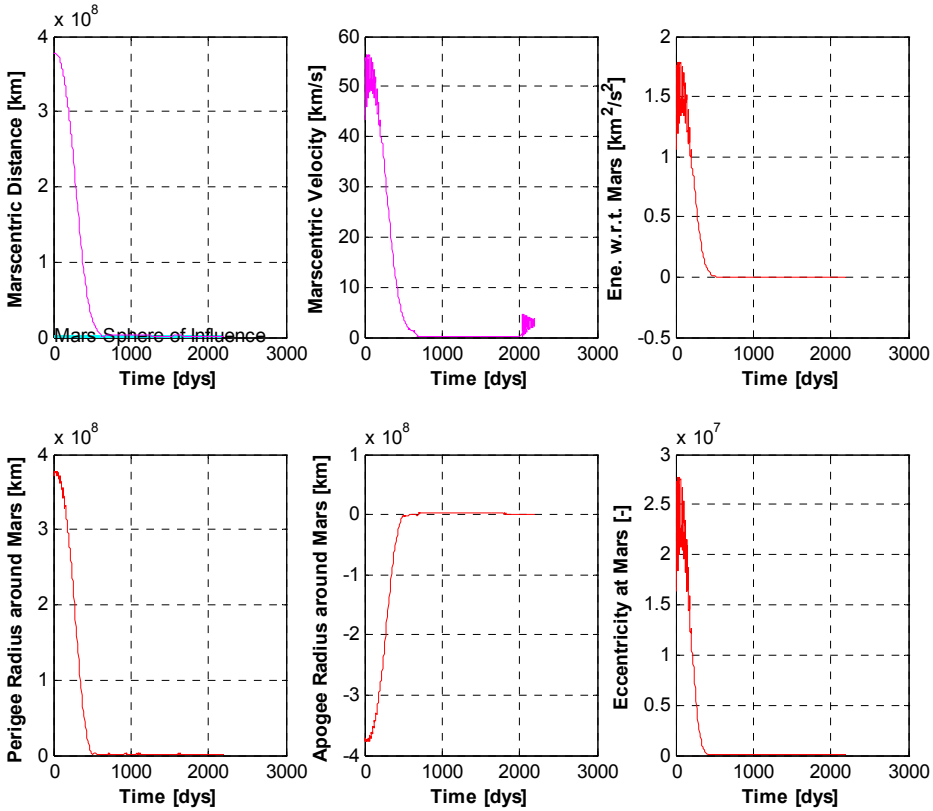

Figure 22. Parameters in Mars frame

The oscillations that appear at the beginning of the plots in figure 22 are consequence of the spiralling around Earth being represented in a Mars reference frame because it is a four body problem that it has been simplified to a two-body perturbation model.

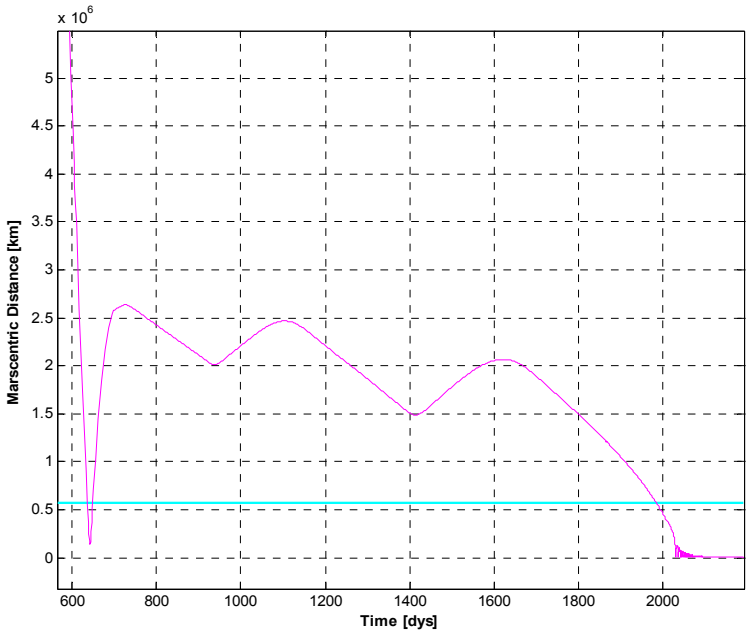

Figure 23. Marscentric distance in Mars frame

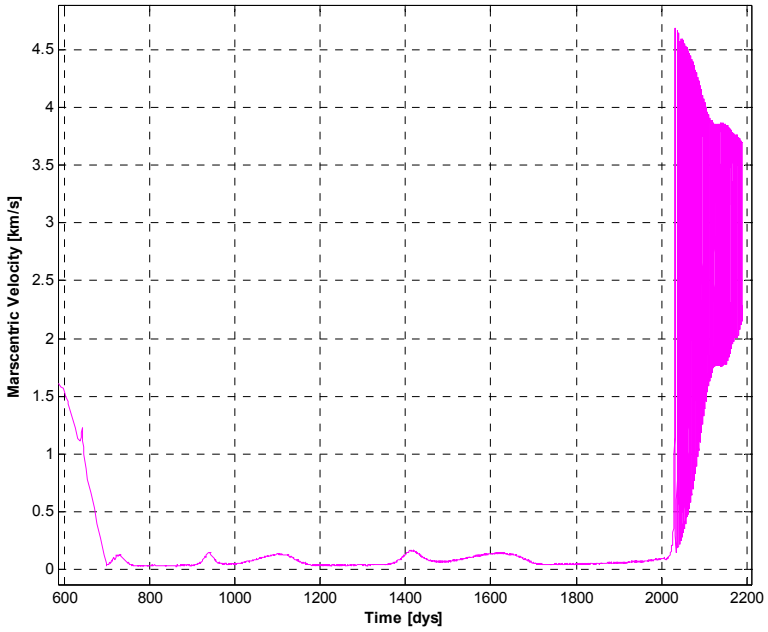

Figure 24. Marscentric velocity in Mars frame

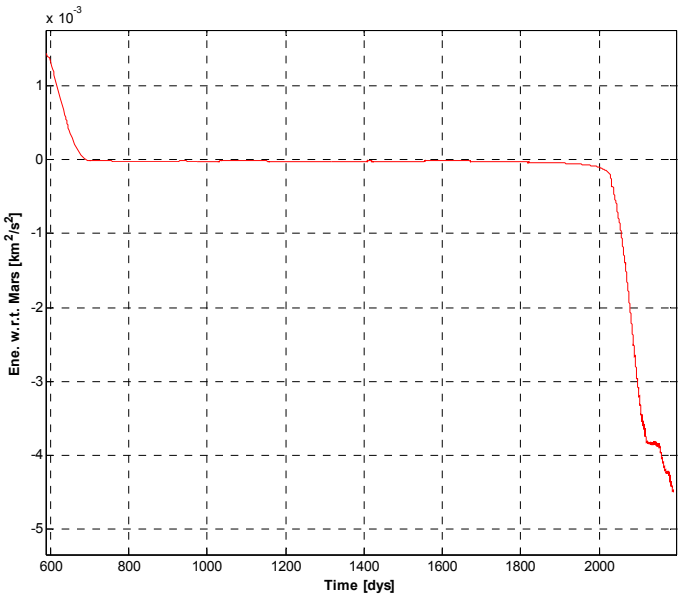

Figure 25. Energy with respect to Mars in Mars frame

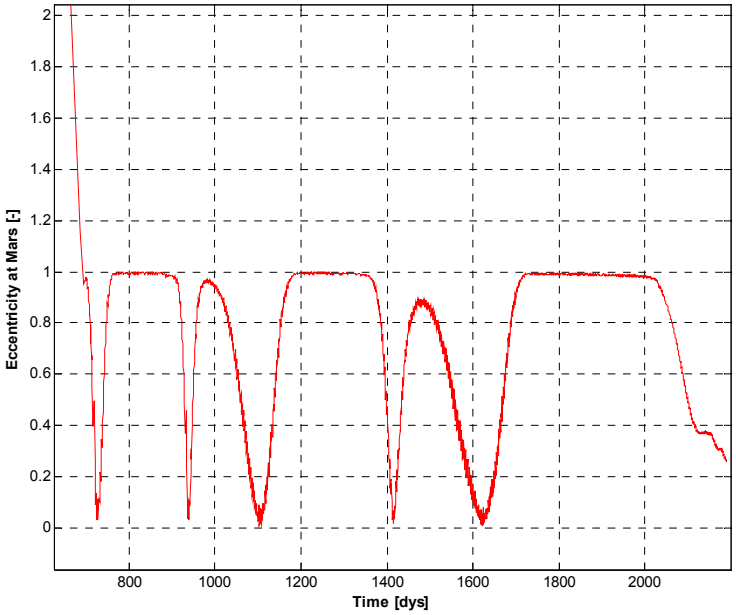

Figure 26. Eccentricity at Mars

From figures 23 to 26 it can be said that the time it takes the spacecraft from Mars arrival until it does the spiraling around the planet it is too much high, but there are two important things to take into account. First, from a mission point of view what is important in this study is the fact that the spacecraft enters the Mars sphere of influence. Although it does not reach the final altitude in a reasonable time later studies can apply better maneuvers that will get better times. It could be considered as a transfer time the time it takes to reach eccentricity at Mars smaller than one, rather than expect a constant in this parameter after the

spiralling. And second, this is not an optimum or best scenario and there can be other cases with best results concerning transfer time. That is why in the next chapter different scenarios will be studied by changing different parameters of the input data.

5. STUDY OF TRANSFERS

In previous chapters the initial conditions, set as the 13 variables forming the initial state vector of the problem (Eq. 5), were fixed and the input data had the predetermined values shown in table 4. By changing these initial conditions, it is possible to change the characteristics of the Earth-to-Mars transfer.
The purpose of the studies described in this chapter is to vary the parameters of the simulations, such as the initial conditions of the spacecraft or the thruster characteristics, in order to find the transfers that minimize transfer time and mass consumed by the spacecraft. This is an important issue of the study of low-thrust propulsion because, as mentioned earlier in the report, chemical rockets are relatively inefficient and, although the transfer time is smaller, they put an upper limit on the mass of payload that can be delivered to Mars. Therefore time and mass are two parameters to be taken into account.

5.1. Chemical Hohmann transfer

Before studying a range of transfers with low-thrust propulsion to reach Mars by applying different values of the input data, it is important to know the relevant parameters of simple chemical transfers from Earth to Mars, such as the Hohmann transfer, in order to compare the values and comment the main differences.

Considering a Hohmann transfer between Earth and Mars, or equally the heliocentric cruise phase in a patched conics method using Hohmann equations, the Δv of the first impulse or the velocity required to leave the Earth's sphere of influence is,

$$\Delta V_1 = V_\infty^E = \sqrt{\frac{\mu_{sun}}{a_E}}\left(\sqrt{\frac{2a_M}{a_E + a_M}} - 1\right) = 2{,}9447 \text{km/s}, \tag{32}$$

where a_E and a_M are the radius of the circular orbits of Earth and Mars around the Sun, respectively.
The Δv of the second impulse or the velocity relative to Mars as it arrives at Mars sphere of influence is,

$$\Delta V_2 = V_\infty^M = \sqrt{\frac{\mu_{sun}}{a_M}}\left(1 - \sqrt{\frac{2a_E}{a_E + a_M}}\right) = 2{,}6489 \text{km/s}. \tag{33}$$

The transfer time is calculated as half the period of the transfer ellipse:

$$t = \frac{\pi}{\sqrt{\mu_{sun}}}\left(\frac{1}{2}\left(a_E + a_M\right)\right)^{\frac{3}{2}} \cdot \frac{1}{3600 \cdot 24} = 258{,}86 \text{ days} = 0{,}71 \text{ years}. \quad (34)$$

It has to be taken into account that this transfer time begins when the spacecraft leaves the Earth's sphere of influence and finishes when it arrives at Mars sphere of influence. The time when the spacecraft is inside the spheres of influence can vary depending on the model or trajectory used. That is to say, the transfer time that will be an output of the following studies has to be reduced before making any comparison with the transfer time of the Hohmann transfer. For example, in figure 15, although the spacecraft takes more than half a year to leave the Earth's sphere of influence, the transfer time between leaving Earth's SOI and arriving at Mars SOI is a bit larger than a year. This means that in comparison with the Hohmann transfer the low-thrust transfer takes longer to arrive at Mars. However, comparing only time is not conclusive and it is necessary to comment also the efficiency of the transfer in terms of mass of payload delivered to Mars, or equally, the fuel mass consumed. These two parameters, transfer time and fuel mass consumed, will be relevant in the following studies.

5.2. Changing the initial conditions of the spacecraft

The first study about modifying the input data consists in changing the initial position $\mathbf{R_{SC\text{-}init}}$ and velocity $\mathbf{V_{SC\text{-}init}}$ of the spacecraft in order to find the initial conditions that minimize transfer time and mass consumed by the spacecraft.

Apart from the initial position $\mathbf{R_{SC\text{-}init}}$ and velocity $\mathbf{V_{SC\text{-}init}}$ of the spacecraft, the other initial conditions, initial mass of the spacecraft and initial position and velocity of the Earth and Mars, are maintained constant in this study.
By looking at the equation of $\mathbf{R_{SC\text{-}init}}$,

$$R_{SC-init} = R_{E-init} + R_{SC/E-init},$$

$$R_{SC-init} = R_{E-init} + \left[\left(R_E + h_0\right)\cos\theta_{SC-init}, \left(R_E + h_0\right)\sin\theta_{SC-init}\right], \quad (35)$$

and the equation of $\mathbf{V_{SC\text{-}init}}$,

$$V_{SC-init} = V_{E-init} + V_{SC/E-init},$$

$$V_{SC-init} = V_{E-init} + \left[-\sqrt{\frac{2\mu_E}{\left(R_E + h_0\right)} - \frac{\mu_E}{a_0}}\sin\theta_{SC-init}, \sqrt{\frac{2\mu_E}{\left(R_E + h_0\right)} - \frac{\mu_E}{a_0}}\cos\theta_{SC-init}\right], \quad (36)$$

it can be seen that the parameters that can be modified in order to change the initial position and velocity of the spacecraft are:

- perigee altitude of the initial GTO orbit h_0,
- semi-major axis of the initial GTO orbit a_0 and
- initial angle of the spacecraft with respect to the positive **X** axis of the Earth frame of reference $\theta_{SC\text{-}init}$.

It has to be pointed out here that the spacecraft always begins the transfer at perigee, but the angle $\theta_{SC\text{-}init}$ can be modified because what is being changed is the orientation of the orbit and therefore the initial position and velocity of the spacecraft. This can be clearly seen in figure 27.

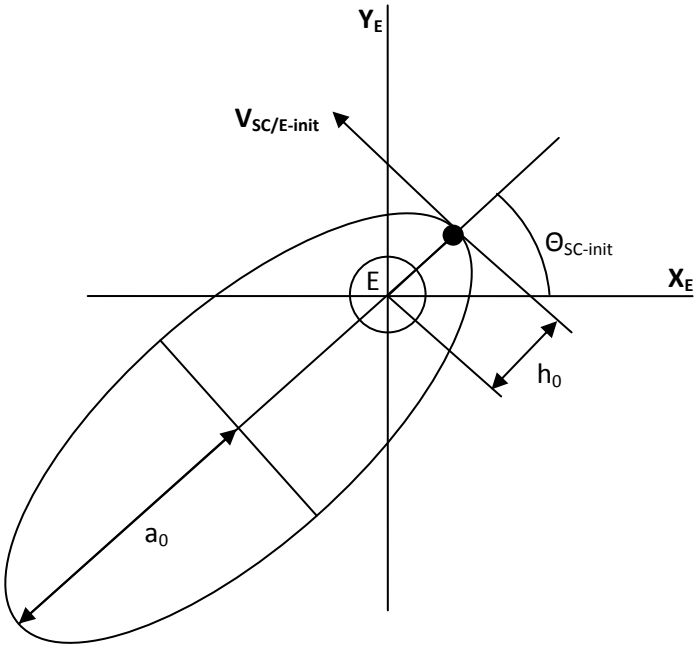

Figure 27. Spacecraft's initial position in the Earth frame of reference (not to scale)

Once the parameters that need to be changed, in order to define different initial positions and velocities of the spacecraft, are known, the output parameters of the study, transfer time and fuel mass consumed, are computed.
As it was said at the end of chapter 4.6, independently of the input transfer time applied as input data, the current transfer time from Earth escape to Mars arrival is smaller although the code propagates the trajectory of the transfer along the whole number of years set as transfer time.

In order to obtain the transfer time and mass consumed, a modification is introduced in the code so that the propagation of the final trajectory stops when the eccentricity at Mars e_M is less than one for the first time. Then, the time and mass of the spacecraft are recorded and therefore the current transfer time is

known and the fuel mass can be calculated by subtracting the current mass to the initial mass of the spacecraft:

$$\text{Transfer time} = t \text{ (when } e_M<1 \text{ by the first time)}$$
$$\text{Fuel mass consumed} = M_{init} - M \text{ (when } e_M<1 \text{ by the first time)}$$

In case the eccentricity at Mars e_M is not less than one at any time, it will mean that the spacecraft may arrive at Mars but does not stay, then the case is not considered and the transfer time is set to zero.

Another option would be to stop the propagation of the trajectory when the spacecraft enters the Mars sphere of influence and set the time at that moment as the transfer time, but the other case when the eccentricity at Mars is less than one for the first time, can guarantee with more probability that afterwards the spacecraft will stay captured.

In order to obtain the transfer time and fuel mass consumed for different initial positions, the values of h_0, a_0 and $\theta_{SC\text{-}init}$ are changed in a triple for-loop according to the values shown in table 6. For each combination of parameters, the main function "EarthMarsLT" is called taking into account that the input parameter that does not change is automatically defined and it is not introduced manually by the user. Every time the main function is called, h_0, a_0, $\theta_{SC\text{-}init}$ and the output parameters, i.e., transfer time and fuel mass consumed, are saved in a structure called *Data*.

Different values for the input data	
h_0=	100 km
	300 km
	500 km
	700 km
a_0=	22500 km
	23500 km
	24500 km
	25500 km
$\theta_{SC\text{-}init}$ =	0º
	45º
	90º
	135º
	180º
	225º
	270º
	315º

Table 6. Input data to change initial position and velocity of the spacecraft

Next, from figure 28 to 35 the transfer time and fuel mass consumed for the different situations are shown. Each figure corresponds to a value for the semi-major axis of the initial GTO orbit a_0, where transfer time or fuel mass consumed are represented in the vertical axis and each point corresponds to a value of the angle $\theta_{SC\text{-}init}$, represented in the horizontal axis, and a value of the perigee altitude of the initial GTO orbit h_0, represented by different colours. If a combination of parameters does not appear, it is because the transfer time has been set to zero meaning e_M is not one at any time along the trajectory and the case is not considered.

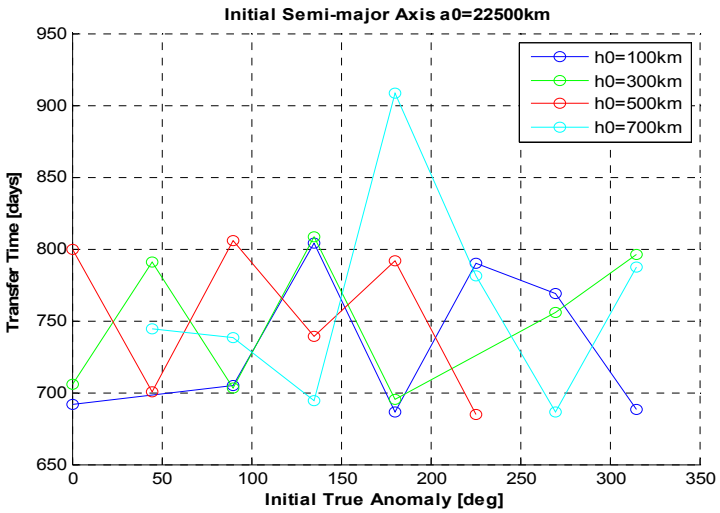

Figure 28. Transfer time (a_0 = 22500 km)

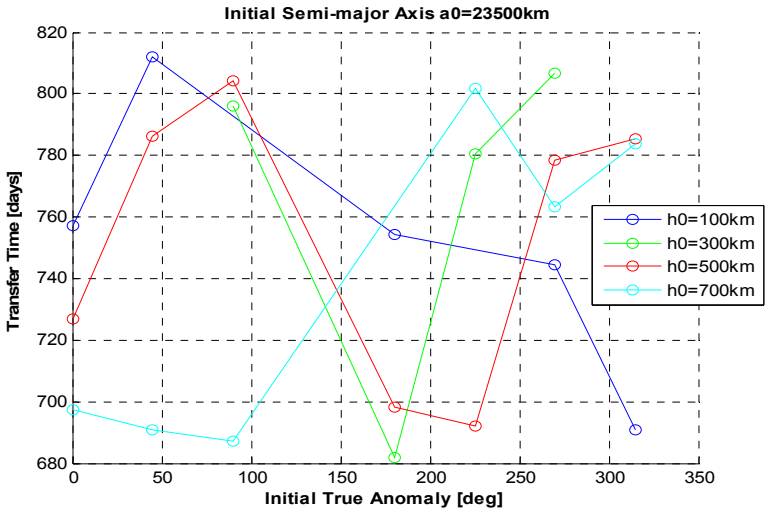

Figure 29. Transfer time (a_0 = 23500 km)

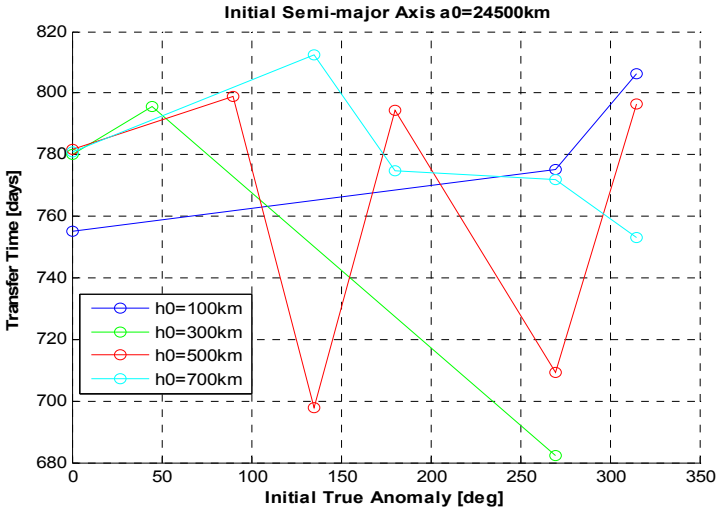

Figure 30. Transfer time (a_0 = 24500 km)

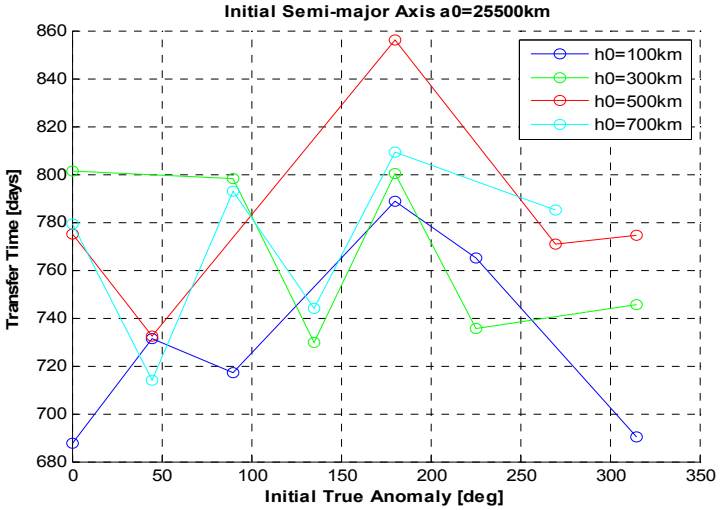

Figure 31. Transfer time (a_0 = 25500 km)

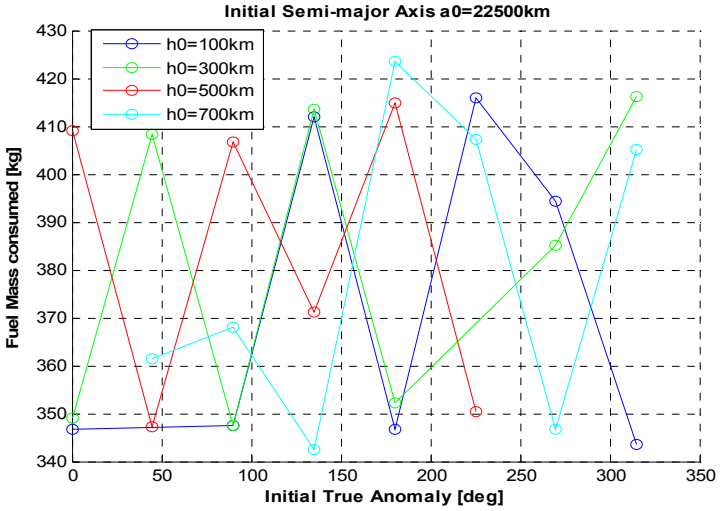

Figure 32. Fuel mass consumed (a_0 = 22500 km)

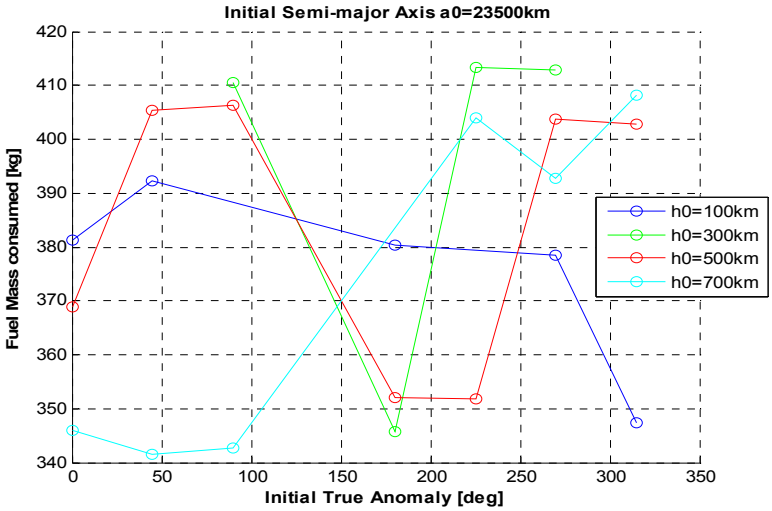

Figure 33. Fuel mass consumed (a_0 = 23500 km)

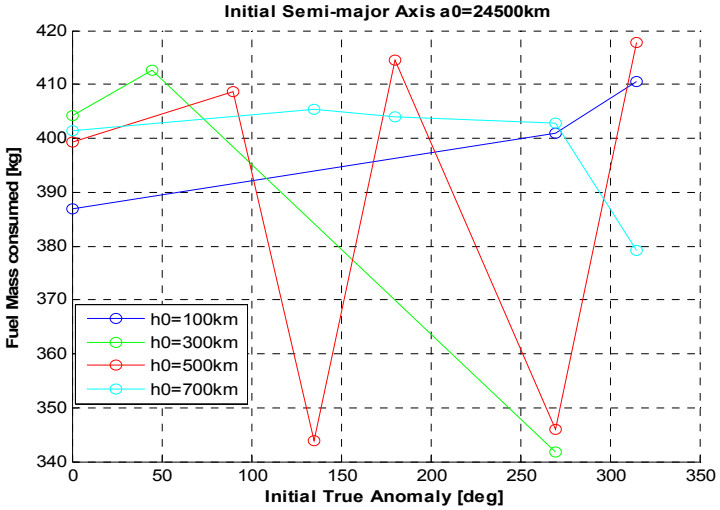

Figure 34. Fuel mass consumed (a_0 = 24500 km)

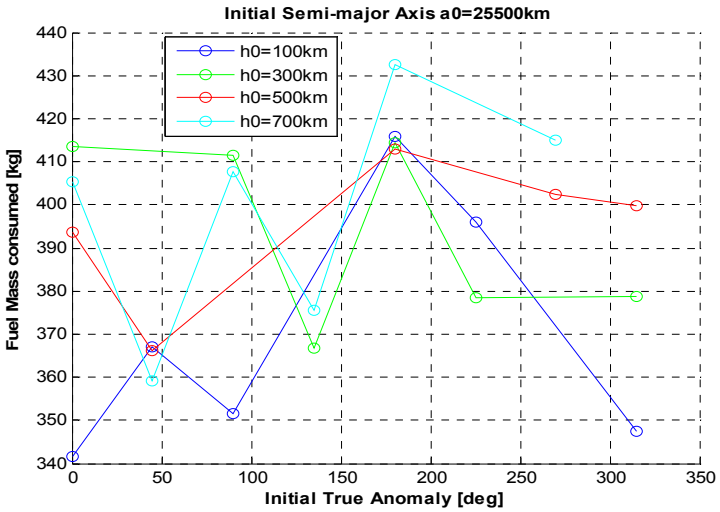

Figure 35. Fuel mass consumed (a_0 = 25500 km)

By looking at the figures, it can be said that the transfer time varies regularly from 680 to 800 days and the fuel mass consumed from 340 to 420 kg. If the thruster is never switched off along the trajectory, time and mass are proportional to each other and the propellant consumption plots can be obtained from the time plots. For example, when a_0=22500km, comparing figures 28 and 32 it can be seen that the distribution of points representing the different transfers are the same for transfer time and fuel mass consumed plots. However, this is not a general rule, because the case of zero thrust is considered in the function "EQmotion" of the code, when certain conditions are not met (chapter 4.2). Prove of that, when a_0=25500km and $\theta_{SC\text{-}init}$=180º, there is no proportionality since the minimum transfer time corresponds to h_0=100km, while the minimum fuel mass consumed corresponds to h_0=500km.

Although there are not great differences, it is useful to indicate the parameters that correspond to the minimum transfer time and the minimum fuel mass consumed:

Minimum transfer time at:	
h_0	300 km
a_0	23500 km
$\theta_{SC\text{-}init}$	180º
Minimum transfer time	681,9459 days
Fuel mass consumed	345,6536 kg

Table 7. Case of minimum transfer time

Minimum fuel mass consumed at:	
h_0	700 km
a_0	23500 km
$\theta_{SC\text{-}init}$	45º
Minimum transfer time	691,0371 days
Fuel mass consumed	341,4798 kg

Table 8. Case of minimum fuel mass consumed

There are other cases where neither the transfer time nor the fuel mass consumed have a minimum but both are close to the minimum values. Although these cases could be perfectly an option to be taken into account if only one set of parameters are chosen the case of minimum transfer is the chosen one because not only the transfer time has a minimum, but also the fuel mass consumed is really close to the minimum value.

Parameters of the chosen transfer	
h_0	300 km
a_0	23500 km
$\theta_{SC\text{-}init}$	180º
$\mathbf{R}_{SC\text{-}init}$ (helio frame)	[1 0] au
$\mathbf{V}_{SC\text{-}init}$ (helio frame)	[0 19,66] m/s
$\mathbf{R}_{SC/E\text{-}init}$ (Earth frame)	[-6678,13 0] km
$\mathbf{V}_{SC/E\text{-}init}$ (Earth frame)	[0 -10,12] m/s
Output Data	
Minimum transfer time	681,95 days
Fuel mass consumed	345,65 kg

Table 9. Initial conditions and output values of the chosen transfer

These initial conditions correspond to the spacecraft beginning its transfer at a perigee altitude of 300 km on a GTO with semi-major axis of 23500 km. Moreover, the spacecraft leaves the Earth in the direction of Earth's orbit, since $\theta_{SC\text{-}init}$=180º, meaning the spacecraft is on the **X** axis between the Earth and the Sun.

Apart from identifying the minimum transfer time case, the conclusion after analysing the plots is that there is no clear trend behind the oscillations. That is why an alternative approach study has been followed.

It has been considered the transfer time and fuel mass consumed for different transfers as functions of the initial heliocentric velocity of the spacecraft. In order to change the modulus and direction of such velocity a double for-loop is applied.
One of the loops is not changed and consists in varying the initial angle of the spacecraft with respect to the positive **X** axis of the Earth frame of reference $\theta_{SC\text{-}init}$. This means changing the orientation of the geocentric initial velocity $\mathbf{V}_{SC/E\text{-}init}$ since the difference between these angles is always 90º as it can be seen in figure 27. Therefore, the orientation of the initial heliocentric velocity $\mathbf{V}_{SC\text{-}init}$ will be also modified because the Earth's initial position is the same for all transfers studied.
The second loop changes the modulus of $\mathbf{V}_{SC\text{-}init}$; in particular, what is being changed is the initial geocentric velocity of the spacecraft $\mathbf{V}_{SC/E\text{-}init}$ since the modulus of the Earth's initial heliocentric velocity $\mathbf{V}_{E\text{-}init}$ is constant. In this study the perigee altitude of the departing orbit h_0 has a constant value of 300 km and the variation of the modulus of $\mathbf{V}_{SC/E\text{-}init}$ is applied by changing the semi-major axis of the initial departing orbit a_0 from a value of 300 km to a value of 30000 km. These means starting with a circular departing orbit and then passing to

elliptic departing orbits. The purpose of these values is to have a reasonable range of velocities to display.

As in the previous study, the transfer time and fuel mass consumed displayed are recorded when the eccentricity at Mars e_M is less than one by the first time. Figure 36 and 37 display transfer time in function of the initial heliocentric velocity for the cases of $\theta_{SC\text{-}init}$=0º and $\theta_{SC\text{-}init}$=180º.

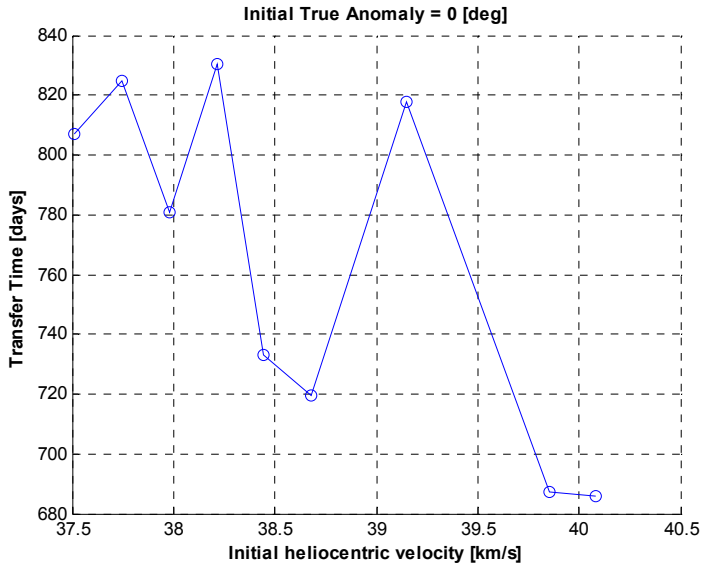

Figure 36. Transfer time vs $V_{SC\text{-}init}$ ($\theta_{SC\text{-}init}$=0º)

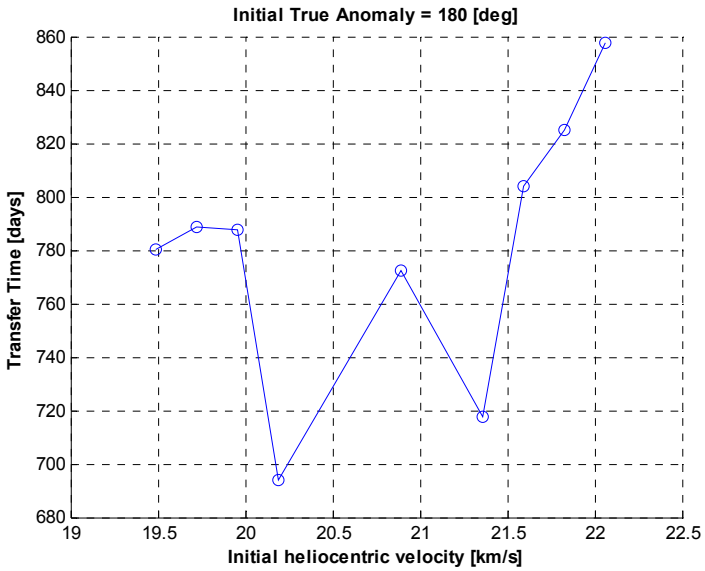

Figure 37. Transfer time vs $V_{SC\text{-}init}$ ($\theta_{SC\text{-}init}$=180º)

In spite of the oscillations that appear in both plots, it can be said that apparently the transfer time decreases when the modulus of the initial heliocentric velocity $\mathbf{V_{SC\text{-}init}}$ increases for $\theta_{SC\text{-}init}$=0º in figure 36 and transfer time increases when the modulus of $\mathbf{V_{SC\text{-}init}}$ increases for $\theta_{SC\text{-}init}$=180º in figure 37.

However, these observations don't reveal any general tendencies as it can be seen in figure 38 where transfer time is displayed with respect to the modulus of the initial heliocentric velocity $\mathbf{V_{SC\text{-}init}}$ for all values of $\theta_{SC\text{-}init}$.

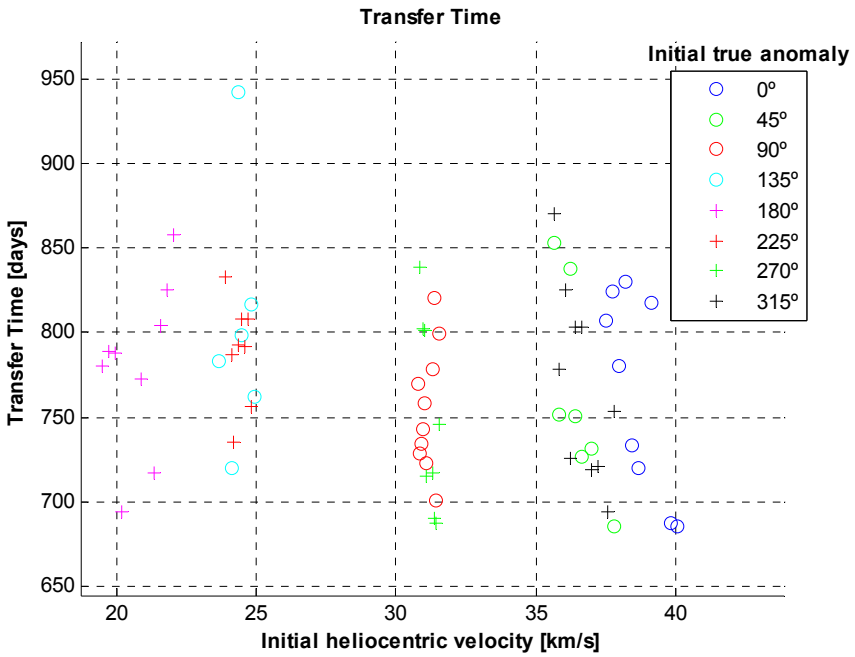

Figure 38. Transfer time vs $V_{SC\text{-}init}$ for all values of $\theta_{SC\text{-}init}$

And the fuel mass consumed is displayed in figure 39.

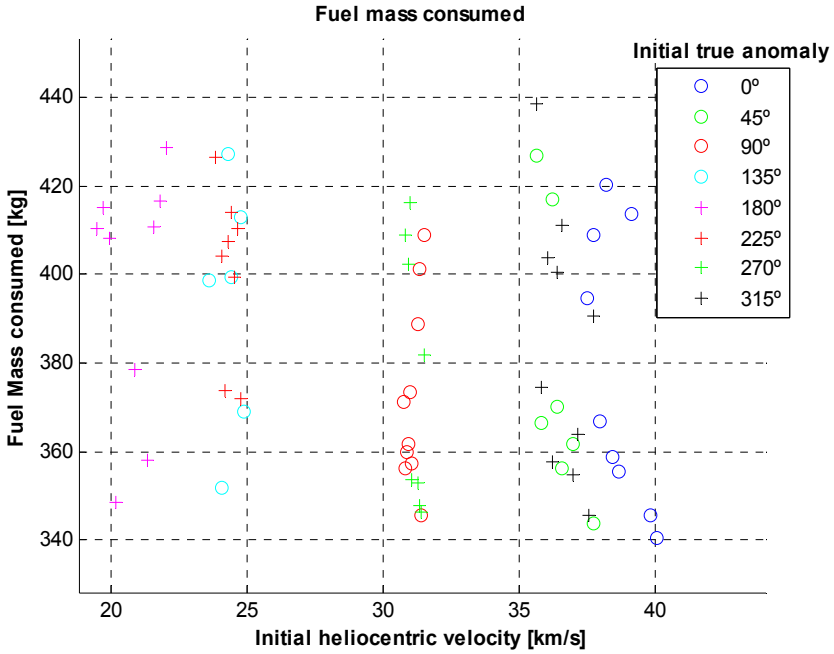

Figure 39. Fuel mass consumed vs $V_{SC\text{-}init}$ for all values of $\theta_{SC\text{-}init}$

In this study, the minimum transfer time and the minimum fuel mass consumed correspond to the same transfer, meaning that time and mass are proportional in this case, because the thruster is never or practically never switched off along the trajectory. The input and output parameters of the transfer are presented in table 10. The corresponding point in figure 38 and 39 is the point of maximum initial heliocentric velocity.

Parameters of the minimum transfer time transfer	
Input Data	
h_0	300 km
a_0	30000 km
$\theta_{SC\text{-}init}$	0º
$V_{SC/E\text{-}init}$	10,3 km/s
$V_{SC\text{-}init}$	40,08 km/s
Output Data	
Minimum transfer time	685,75 days
Minimum Fuel mass consumed	340,53 kg

Table 10. Input and output values of the minimum transfer time transfer

The next step could be making a range of velocities or angles with a great number of points, but since there are oscillations in the plots presented, there would not be many advantages from performing such an analysis and the best option is to change the approach of the study. In the following chapter another alternative approach is followed by making a range of departing orbits in order to change the initial conditions of the spacecraft.

5.3. Changing the departing orbit

In this following study instead of performing a range of initial heliocentric velocities to change the initial conditions of the spacecraft, what is being changed is the type of departing orbit. Proving three specific departing orbits, each one with different departure velocities, would contribute to clear conclusions.
The three departing orbits selected are a LEO (low-Earth orbit), a GTO (geostationary transfer orbit) and a GEO (geostationary Earth orbit) and their parameters as perigee altitude and semi-major axis are shown in table 11. LEO and GEO are circular orbits whereas the GTO is an elliptic orbit.

Departing orbits of the study		
Orbit	**h_0 [km]**	**a_0 [km]**
LEO	300	6678
GTO	300	24582
GEO	35786	42164

Table 11. Departing orbits

Apart from changing the departing orbit a for-loop is applied, as in the other studies, to change the initial angle of the spacecraft with respect to the positive **X** axis of the Earth frame of reference $\theta_{SC\text{-}init}$, which means changing the initial position of the spacecraft and the orientation of the initial heliocentric velocity $\mathbf{V}_{SC\text{-}init}$.

After performing the calculations the transfer time and fuel mass consumed are recorded when the eccentricity at Mars e_M is less than one for the first time.

Next the results of this study are displayed. First, the transfer time for the three different departing orbits and departure angles are presented with respect to $\theta_{SC\text{-}init}$ in figure 40 and with respect to the modulus of $\mathbf{V}_{SC\text{-}init}$ in figure 41.

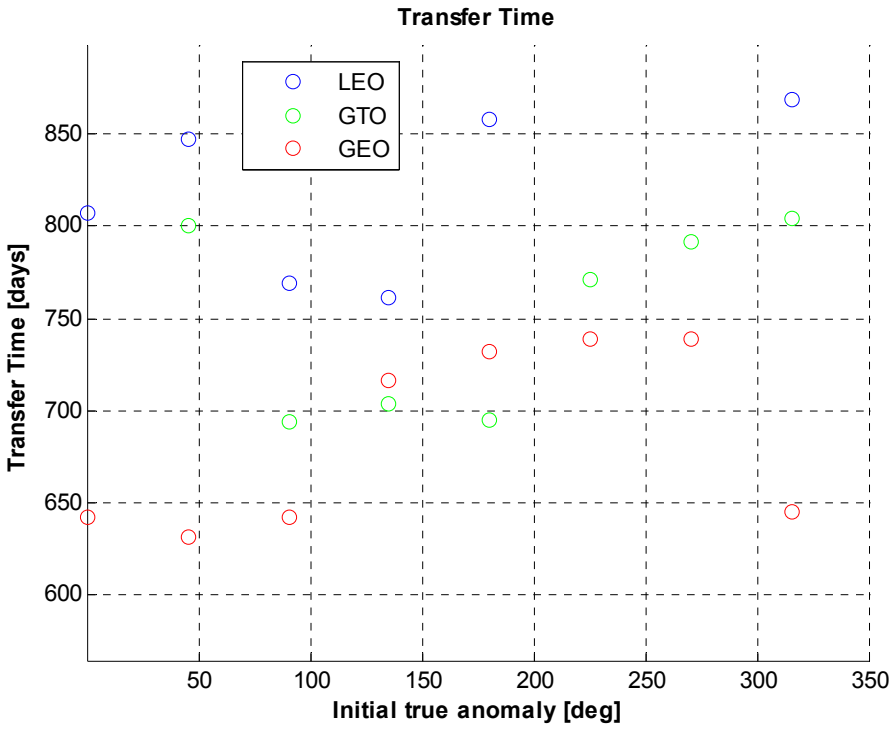

Figure 40. Transfer time vs $\theta_{SC\text{-}init}$ for different departing orbits

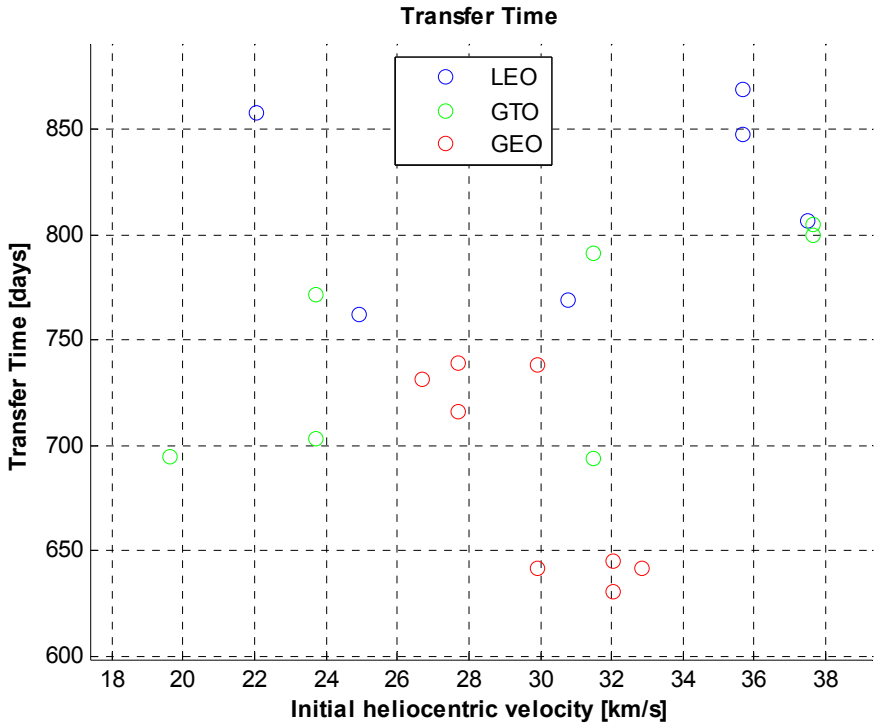

Figure 41. Transfer time vs $V_{SC\text{-}init}$ for different departing orbits

From what can be seen in figures 40 and 41, in general the transfers with greater transfer time correspond to cases with LEO as departing orbit. The reason is that, since the departing orbit is close to Earth, the spiralling around Earth takes more time than in the other transfers and the total transfer time increases. On the other hand, transfers with the shortest transfer time have a GEO as departing orbit because the initial position of the spacecraft is further from Earth and the spiralling takes less times until the spacecraft leaves the Earth's sphere of influence. The transfers starting from the GTO have intermediate transfer times relative to LEO and GEO.

There are some transfers than don't fulfil the above-mentioned conclusions. For example when $\theta_{SC\text{-}init}$=145º or $\theta_{SC\text{-}init}$=180º, the minimum time transfer corresponds to the GTO as departing orbit and not the GEO. This means that when $\theta_{SC\text{-}init}$ is closet o 180º, meaning the spacecraft leaves Earth in the direction of Earth's orbit, the best option is a GTO rather than a GEO as departing orbit.

The range of initial heliocentric velocities is more or less the same as in the previous studies for LEO and GTO as departing orbit, since they are orbits that where before considered by changing perigee altitude and semi-major axis. The difference in this study is mainly the GEO orbit that as it is shown in figure 41 has a narrow range of initial velocities, and it can be said that for GEO as departing orbit greater initial velocities have minimum transfer times.

The same distribution of points and the same conclusions, but referred to fuel mass consumed instead of transfer time, are shown in figure 42 and 43.

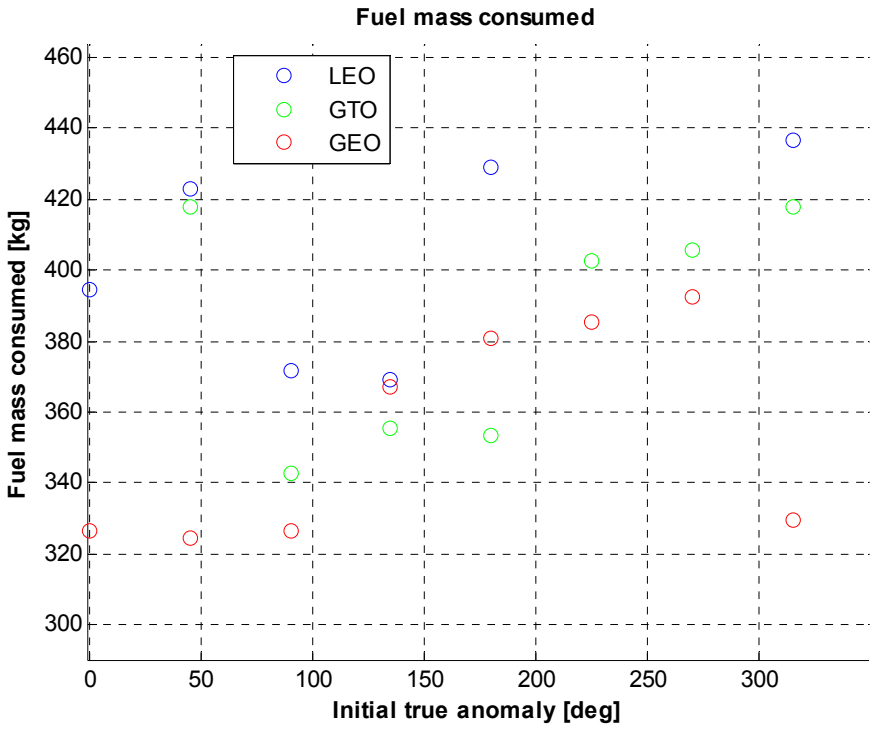

Figure 42. Fuel mass consumed vs $\theta_{SC\text{-}init}$ for different departing orbits

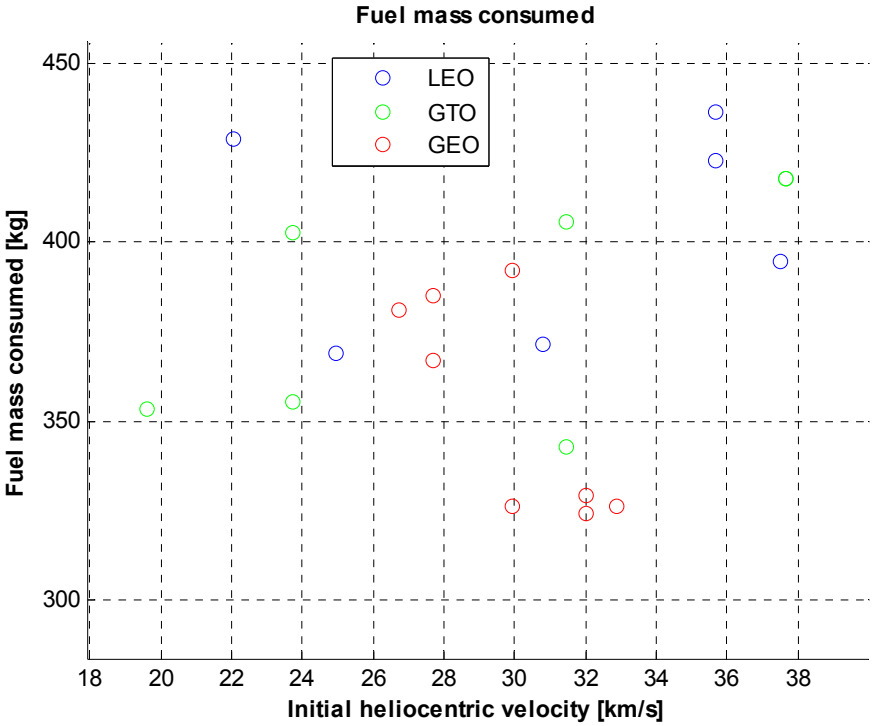

Figure 43. Fuel mass consumed vs $V_{SC\text{-}init}$ for different departing orbits

The relevant inputs and outputs of the study corresponding to the minimum transfer time and the minimum fuel mass consumed are presented in table 12.

Parameters of the minimum transfer time transfer	
Input Data	
Departing Orbit	GEO
h_0	35786 km
a_0	42164 km
$\theta_{SC\text{-}init}$	45º
$V_{SC\text{-}init}$	32,03 km/s
Output Data	
Minimum transfer time	630,95 days
Minimum Fuel mass consumed	324,29 kg

Table 12. Input and output values of the minimum transfer time transfer (changing the departing orbit study)

Until now, only the initial position and velocity of the spacecraft has been changed in order to study different transfers, but the thruster characteristics has been always the same and equal to their predetermined values. In the following chapter, these will be also modified.

5.4. Characteristics of existing low-thrust propulsion thrusters

Before performing the study of changing the thruster characteristics, it is necessary to know the performance of existing low-thrust propulsion thrusters that have been used in space missions, are ready to flight or are under development.

The thruster characteristics considered in the code as input data, and therefore the data of existing thrusters that would be needed in order to perform the study, are:

- Electric Power P
- Specific Impulse I_{sp}
- Thrust efficiency η

Table 13 presents a list of electrostatic ion engines and hall thrusters and their characteristics (nominal values). The main difference between the different electrostatic xenon ion thrusters presented is the method of ionizing the propellant atoms: electron bombardment (NSTAR, T5, T6, NEXT), radiofrequency excitation (RIT-10, RIT-22) or microwave excitation (μ10, μ20).

Thruster	Power [W]	I_{sp} [s]	η	Thrust [mN]
Ion engines				
NSTAR	2300	3100	0,61	92
T5	476	3200	0,55	18
T6	6800	4700	0,68	200
RIT-10	459	3400	0,52	15
RIT-22	5000	4500	0,66	150
μ10	340	3090	0,36	8,1
μ20	1015	3100	0,55	30,4
ETS-8	611	2665	0,5	23,2
NEXT	6900	4190	0,7	236
HiPEP	30000	8900	0,8	540
Hall thrusters				
PPS-1350	1500	1650	0,55	88
SPT-100	1350	1600	0,5	80
SPT-140	5000	1750	0,55	300

Table 13. Thrusters' characteristics [14][15][16][17][18][19][20]

There are thrusters that have already been employed in space such as the NSTAR used in the Deep Space 1 mission or the SPT-100 used in the ESA mission SMART-1. Others will be used in the near future, such as the T6 thruster in the BepiColombo mission to Mercury or the NEXT thruster developed by NASA. And there is one thruster that is currently under development characterized by greater power compared to the other thrusters, the High Power Electric Propulsion thruster (HiPEP).

5.5. Changing the thruster characteristics

Changing the thruster characteristics in the code is equivalent to changing the electric power P, the specific impulse I_{sp} and the thrust efficiency η. This means the parameter that will change is the thrust of the spacecraft, since power, specific impulse and thrust efficiency are needed to calculate it.

$$\mathrm{Th} = \frac{2\eta \mathrm{P}}{\mathrm{I_{sp}} g_0}. \tag{37}$$

The same expression is expressed in Eq. 3 considering adimensional units.
By changing the thrust parameter Th, the parameters that will be affected are the thrust acceleration $\mathrm{A_{SC-F}}$ and the mass variation $\dot{\mathrm{M}}$ of the spacecraft.

$$\mathbf{A}_{SC-F} = \frac{Th}{M}\hat{\mathbf{e}}_{\mathbf{f}} = \frac{2\eta P}{I_{sp}g_0 M}\hat{\mathbf{e}}_{\mathbf{f}}, \tag{38}$$

$$\dot{M} = -\frac{Th}{I_{sp}g_0} = -\frac{2\eta P}{\left(I_{sp}g_0\right)^2}. \tag{39}$$

The thrust acceleration $\mathbf{A}_{SC-F}$ will change its module but not its direction; the thrust vector $\hat{\mathbf{e}}_{\mathbf{f}}$, that determines if the thrust is tangential, anti-tangential or circumpherential, does not depend on the value of the thrust Th but on the transfer phase of the spacecraft and other parameters like previously analysed in chapter 4.2.

The objective of this study is to obtain a general performance of different low-thrust thrusters in terms of transfer time and fuel mass consumed. It will be performed by doing the same analysis as in chapter 5.3, changing the departing orbit of the spacecraft and the value of $\theta_{SC\text{-}init}$, but this time applying the thruster characteristics of the thrusters presented in table 13 instead of applying the thruster predetermined values (P=5886W, I_{SP}=3000 and η=0,5). The departing orbits studied are the same LEO, GTO and GEO presented in table 11.

The thrusters shown in table 13 with a power value smaller than 1000 W are not presented in the plots because the transfer time is too high, greater than 10 years, and they are not appropriate for this study. The same happens with the thruster μ20, although its performance is better than the previously mentioned thrusters it is worse than the remaining ones. Therefore, the thrusters displayed in the study are:

- NSTAR (NASA Solar Electric Propulsion Technology Application Readiness)
- T6 ion thruster
- RIT-22 (Radio-frequency Ion Thruster)
- NEXT (NASA's Evolutionary Xenon *Thruster)*
- HiPEP (High Power Electric Propulsion thruster)
- PPS-1350 Hall thruster
- SPT-100 (Stationary Plasma Thruster - 100mm diameter)
- SPT-140 (Stationary Plasma Thruster – 140mm diameter)

In order to show clearly the results, the thrusters are divided into ion engines and hall thrusters and represented in different plots.
Figure 44 displays the transfer time as a function of the angle $\theta_{SC\text{-}init}$ for different ion engines, whereas figure 45 displays the fuel mass consumed by the same engines. The legend in the plots shows the ion engine and the departing orbit

applied in parenthesis. The points that are missing correspond to the cases in which the eccentricity at Mars e_M is never smaller than one and are therefore rejected.

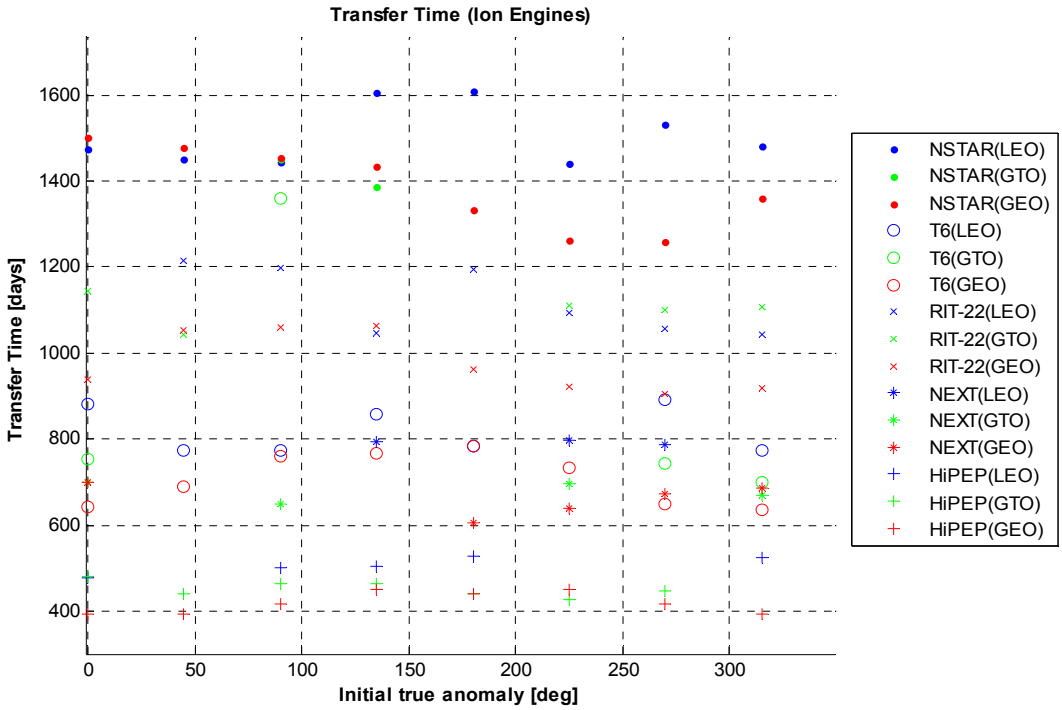

Figure 44. Transfer time for different ion engines

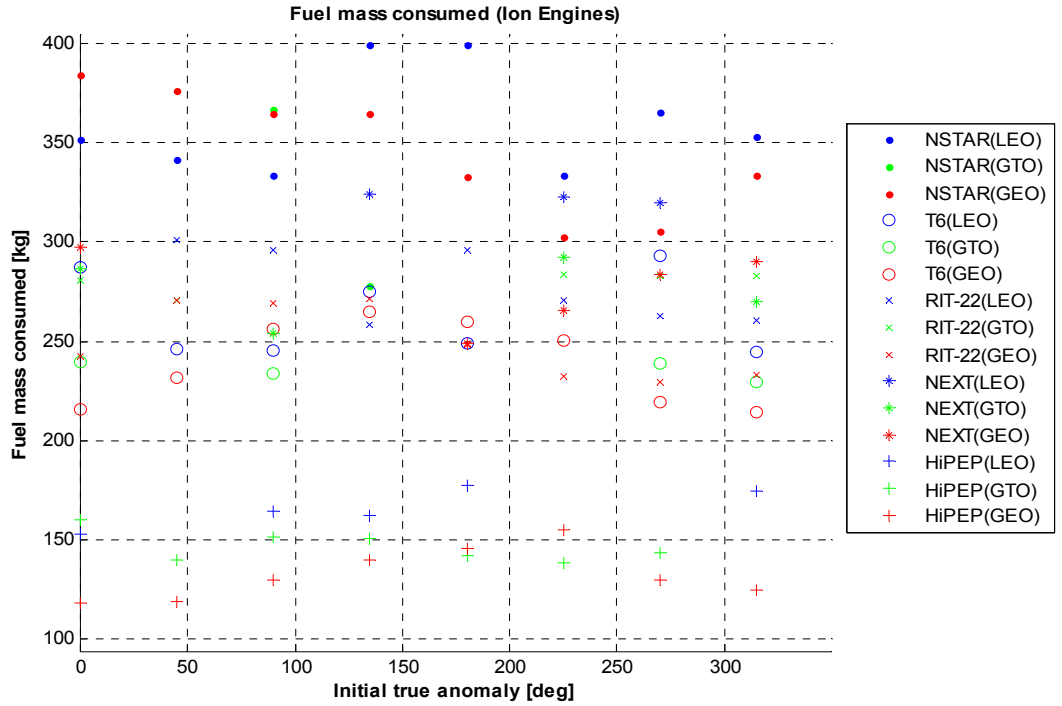

Figure 45. Fuel mass consumed for different ion engines

Next, in figure 46 and 47 the same parameters are displayed for the hall thrusters.

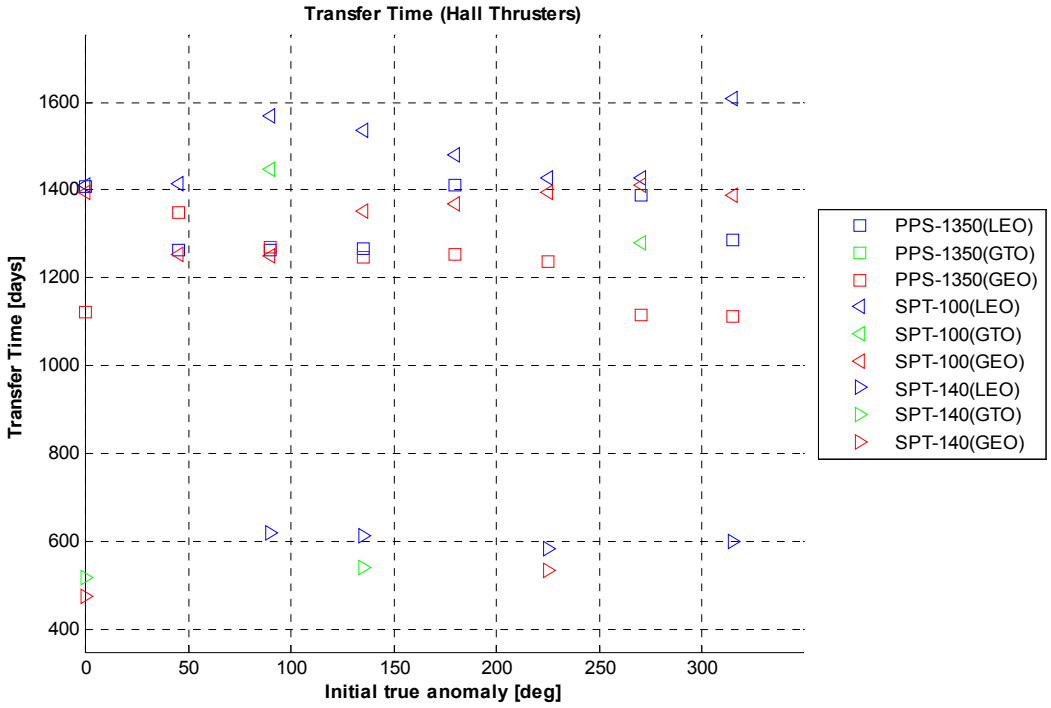

Figure 46. Transfer time for different hall thrusters

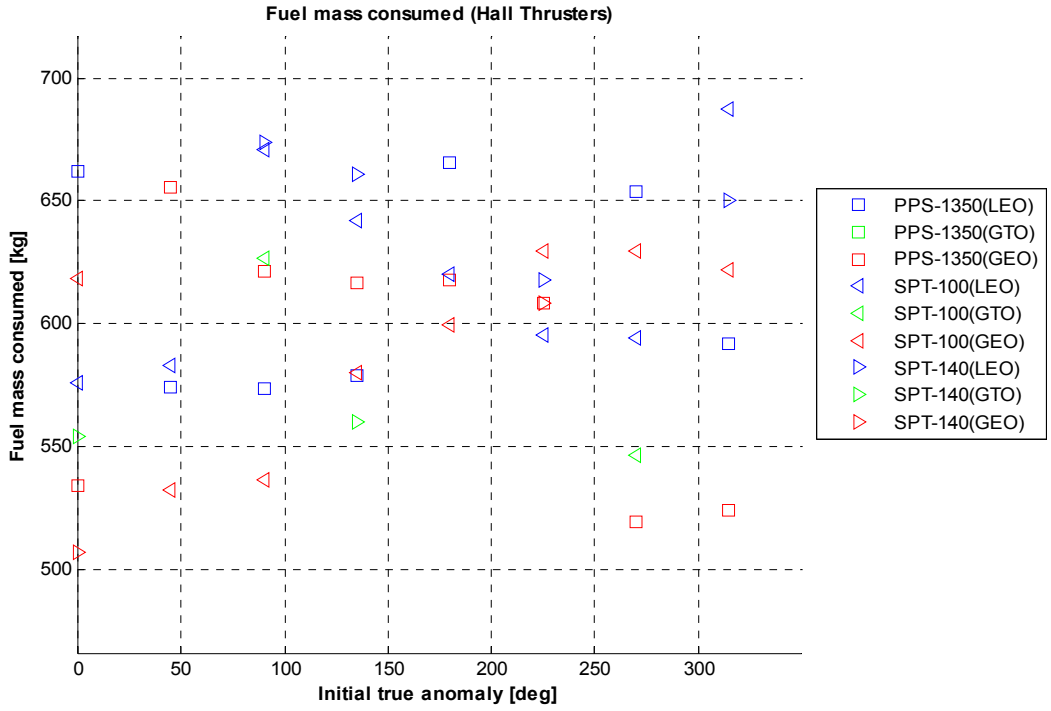

Figure 47. Fuel mass consumed for different hall thrusters

The first comparison to be made is the performance between ion engines and hall thrusters. Although the transfer time for the hall thruster SPT-140 is considerably low, in the same range as the best ion engine, figure 47 shows that the fuel mass consumed for all the hall thrusters is greater than any of the ion engines due to their smaller specific impulse. Therefore, the first conclusion is that an electrostatic ion engine is a better option in comparison with hall thrusters in terms of fuel mass consumed.

Focusing on the ion engines, in particular in the transfer time in figure 44, it can be said that, as it happens in the previous study, independently of the thruster the best departing orbit to minimize transfer time is GEO, since the spiralling around the Earth takes less time than in LEO or GTO.

It is clear from figures 44 and 45 that the best ion engine is the High Power Electric Propulsion thruster (HiPEP) as its transfers offer the minimum transfer time and fuel mass consumed. Since it is a thruster currently under development and with characteristics far from the other thrusters, it is also reasonable to study the other ones that already exist and some already have flown in real missions. What can be said from the other ion engines is that, the NSTAR, the older thruster, has the worst results, and the best ones are the T6 and the NEXT thrusters. In general, the T6 thruster offers less consumption, although the NEXT present smaller transfer times. In order to show clearly the performance of the thrusters that present better results, the same outputs as in figures 44 and 45 are shown in figures 48 and 49 for the T6, NEXT and HiPEP thrusters, this time as a function of the initial heliocentric velocity.

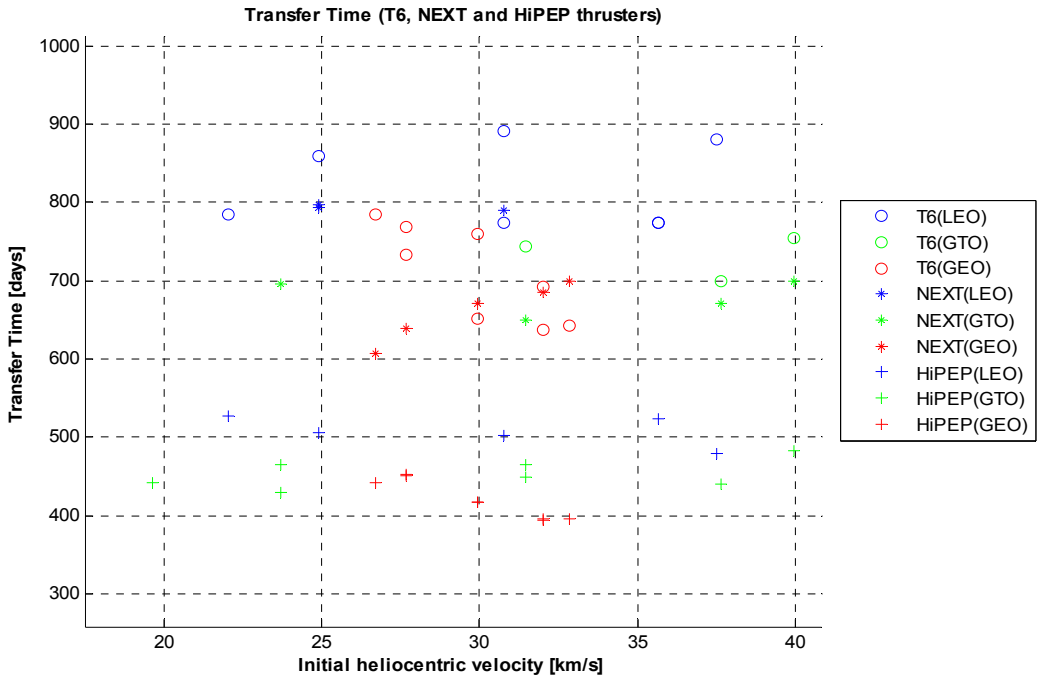

Figure 48. Transfer time for T6, NEXT and HiPEP ion thrusters

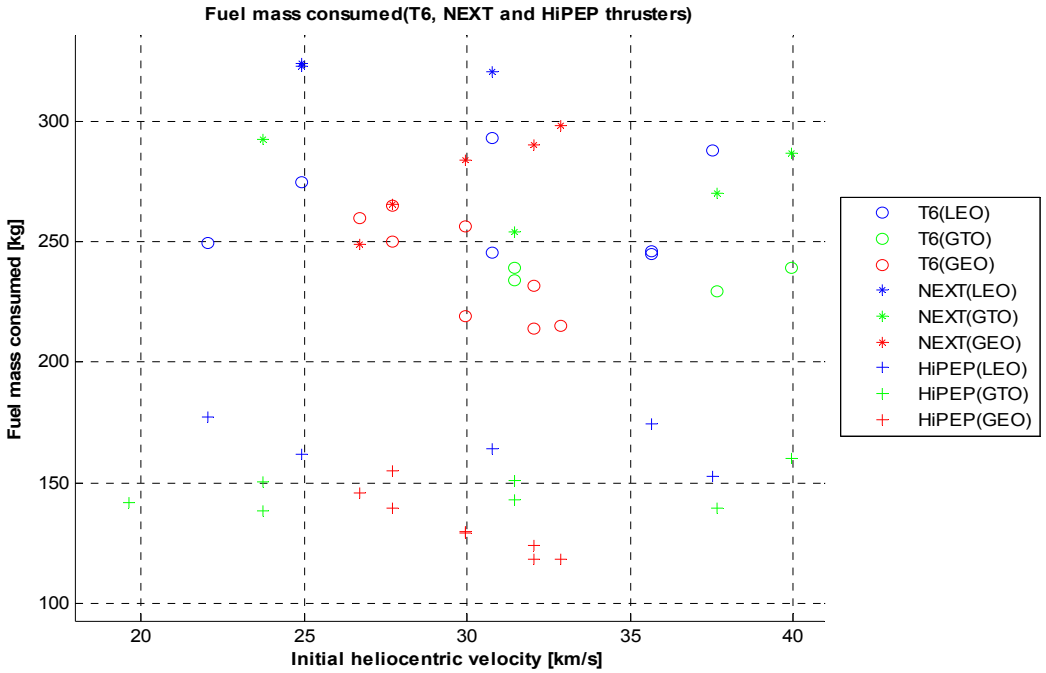

Figure 49. Fuel mass consumed for T6, NEXT and HiPEP ion thrusters

Next, in table 14, there is a summary of the transfers that offer best performance for each of the thrusters displayed in figures 48 and 49, in terms of minimum transfer time and fuel mass consumed. There are two transfers selected for each of the thrusters.

Transfers Data						
Thruster	**T6**		**NEXT**		**HiPEP**	
Dep. Orbit	GEO	GEO	GEO	GEO	GEO	GEO
$\theta_{SC\text{-}init}$ [deg]	0	315	180	225	0	45
$V_{SC\text{-}init}$ [km/s]	32,86	32,03	26,71	27,7	32,86	32,03
Transfer time [days]	642,62	637,28	607,1	639,75	395,44	394,85
Fuel mass consumed [kg]	215,3	214,17	248,78	265,63	118,18	118,55

Table 14. Data of best performance transfers

From the data of these selected transfers, it is easy to see that HiPEP is by far the best option, as it has been previously commented. This result is expected because it is a new generation ion engine under development that offers characteristics considerably better than the rest of existing thrusters (table 13).
About the departing orbit, the GEO is the best option for all the selected transfers, the difference is that NEXT best transfers occur when $\theta_{SC\text{-}init}$ is around 180º whereas T6 and HiPEP best transfers have a value of $\theta_{SC\text{-}init}$ around 0º.

For T6 and NEXT thrusters, the transfer corresponding to minimum transfer time coincides with the minimum fuel mass consumed transfer. However this is not true in the case of HiPEP, although the two transfers have quite similar outputs.
Comparing the outputs of minimum transfer time using T6 or NEXT ion engines in table 14, on one hand, it can be revealed that the advantage of using NEXT ion engine instead of T6 is reducing the transfer time by 30,18 days with the drawback of increasing 34,61 kg the consumption. On the other hand, using T6 instead of NEXT has the opposite results, that is to say, reducing the consumption and increasing the transfer time.
In the following chapter, different plots are presented characterizing some of the transfers previously commented.

5.6. Transfers plots

The plots displayed in this chapter belong to three different transfers, each one corresponding to T6, NEXT or HiPEP thrusters. The transfer for each ion engine are selected from table 14 and represented in table 15.

Selected Transfers			
Thruster	**T6**	**NEXT**	**HiPEP**
Dep. Orbit	GEO	GEO	GEO
$\theta_{SC\text{-}init}$ [deg]	0	180	0
$V_{SC\text{-}init}$ [km/s]	32,86	26,71	32,86
Transfer time [days]	642,62	607,1	395,44
Fuel mass consumed [kg]	215,3	248,78	118,18

Table 15. Data of selected transfers

The transfer using NEXT ion engine corresponds to the minimum transfer time and fuel mass consumed transfer while the transfer using HiPEP corresponds to the minimum fuel mass consumed transfer. The case of T6 thruster is not the minimum, although it is close enough and it has been selected to show the spiralling around Mars in the plots.

First, figures 50 and 51 show the main plots corresponding to the overall minimum transfer time and fuel mass consumed, when using the HiPEP thruster.

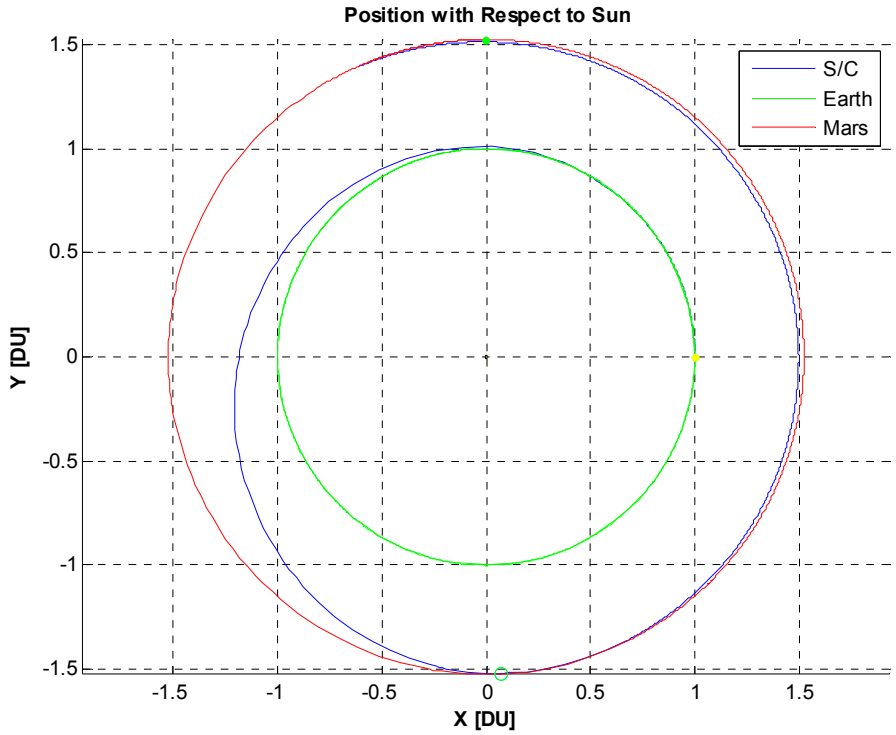

Figure 50. Heliocentric transfer (HiPEP)

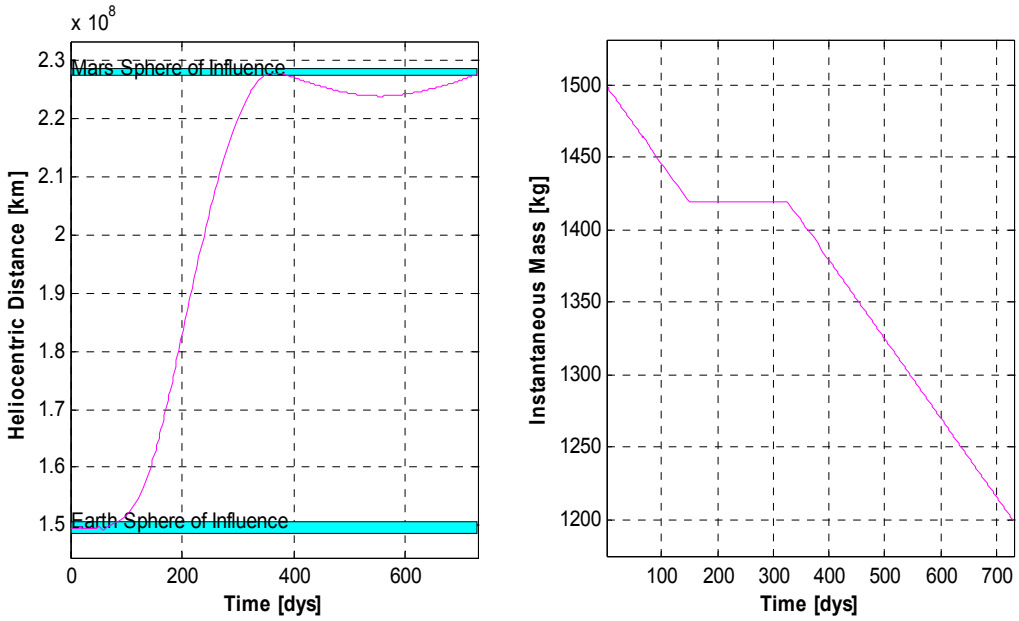

Figure 51. Heliocentric distance and instantaneous mass (HiPEP)

The direct trajectory shown in figure 50 and 51 shows how the spacecraft leaves the Earth's sphere of influence approximately after a quarter of the Earth's orbit and then arrives to Mars sphere of influence after a bit less than 400 days. During a long period of the transfer the thrust is set to zero, as shown in the right plot of

figure 51 when the instantaneous mass is constant. This fact demonstrates the great power of the thruster HiPEP and its low consumption.

Figures 52 and 53 show the main plots corresponding to the transfer selected using NEXT ion engine.

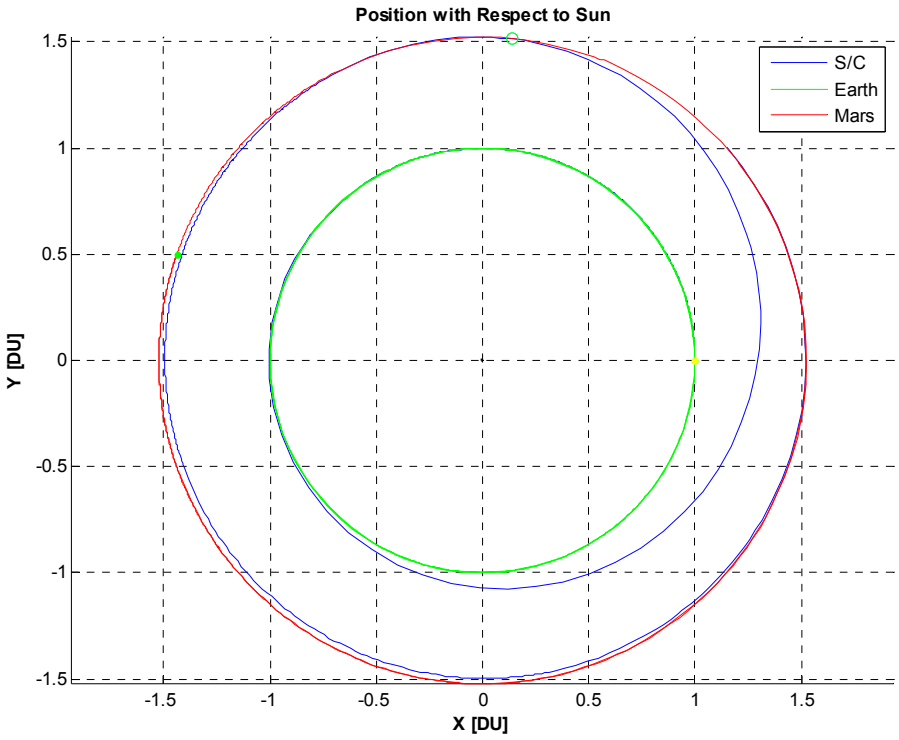

Figure 52. Heliocentric transfer (NEXT)

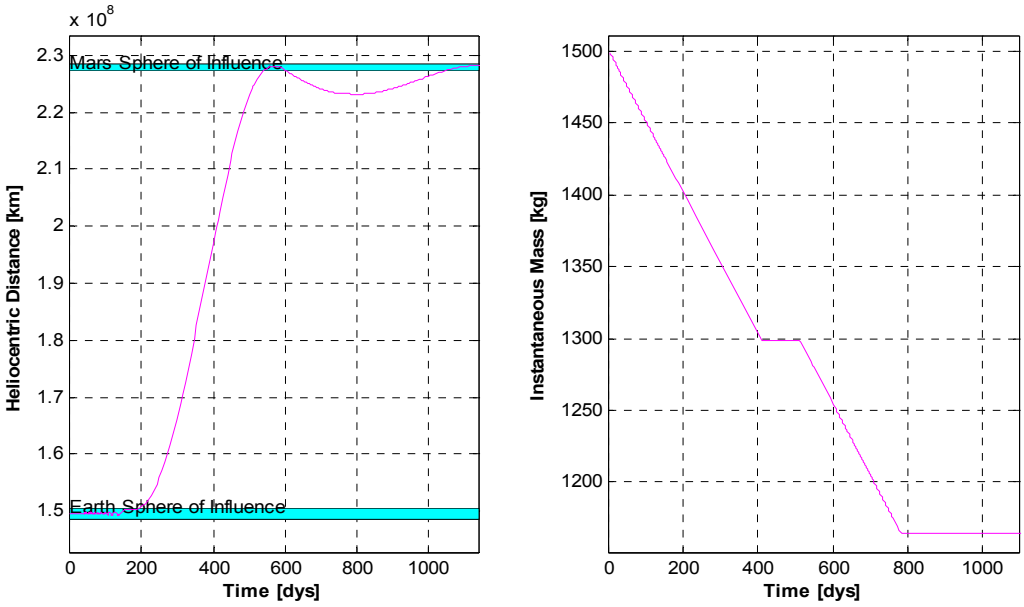

Figure 53. Heliocentric distance and instantaneous mass (NEXT)

This case is also a direct trajectory and compared with the previous transfer, since the thruster has worse performance, the spiralling around the Earth lasts longer and the spacecraft follows the Earth's orbit during more time, as shown in figure 52. Moreover, the thrust is set to zero during a small period of time.

Finally, the last transfer, corresponding to the case using a T6 thruster, is shown in figures 54 and 55. In this case, the trajectory also considers the spiralling around Mars at arrival.
Comparing this transfer with the previous one, using the NEXT ion engine, the first part of the trajectory is rather similar and the difference is at the second part of the transfer because the trajectory until encountering Mars lasts a bit more.

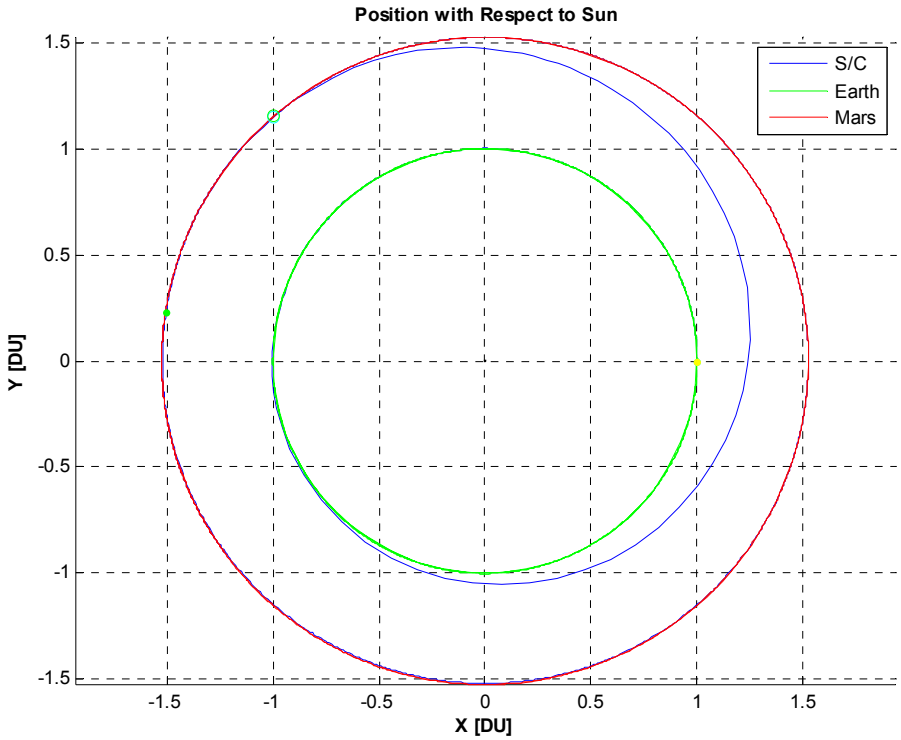

Figure 54. Heliocentric transfer (T6)

From what can be seen in the plots of figure 55, the spacecraft doesn't leave the Mars sphere of influence after arrival and the thrust is practically never off.

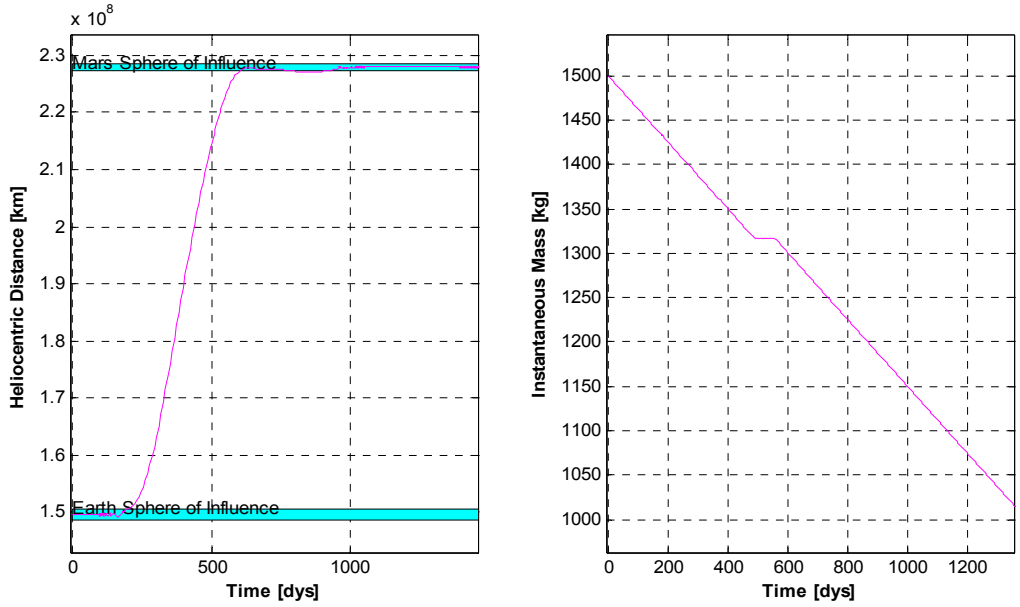

Figure 55. Heliocentric distance and instantaneous mass (T6)

Figure 56 displays the position of the spacecraft with respect to Earth and Mars respectively.

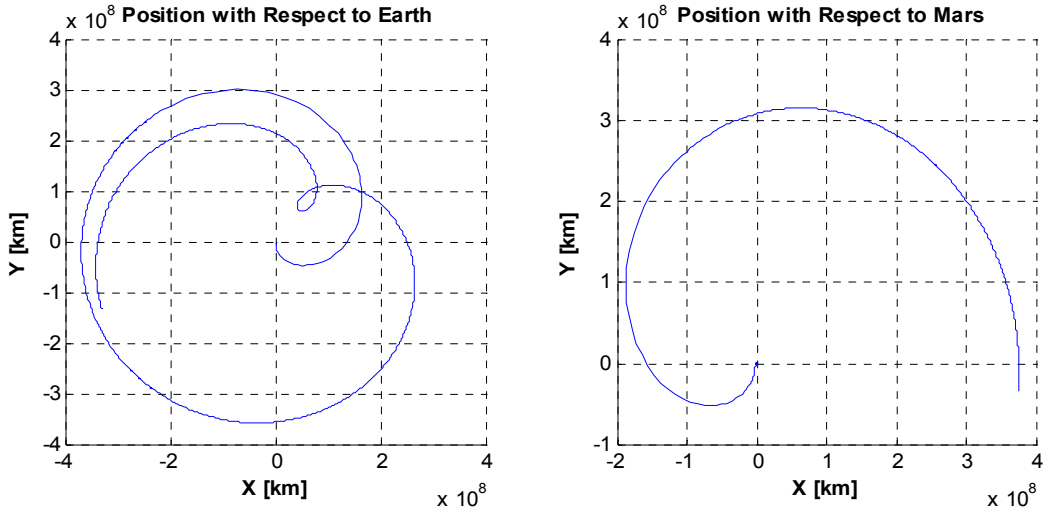

Figure 56. Spacecraft's position in Earth and Mars reference frame (T6)

When applying zoom to the left plot of figure 56, the spiralling of the spacecraft around the Earth appears, where it can be seen the circular GEO as departing orbit (figure 57). This spiralling in the other transfers would be similar, with the only difference of more or less spins around Earth depending on the thruster characteristics and the initial conditions of the spacecraft.

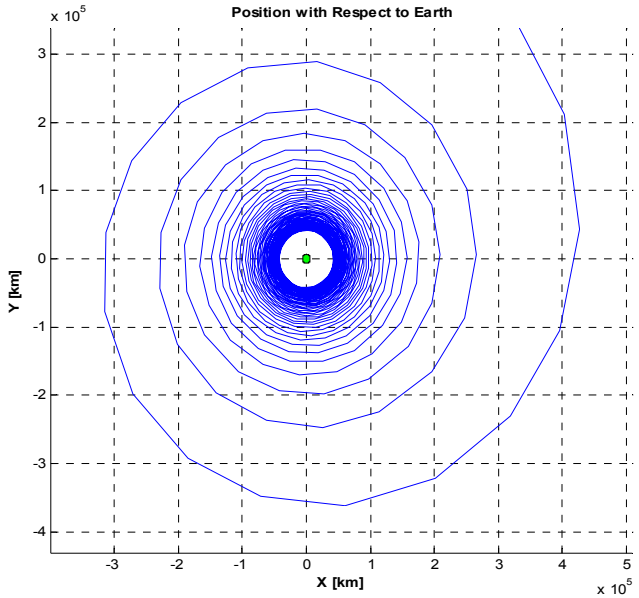

Figure 57. Spiralling around Earth in Earth reference frame (T6)

Figure 58 displays the spiralling around Mars corresponding to the different elliptical orbits.

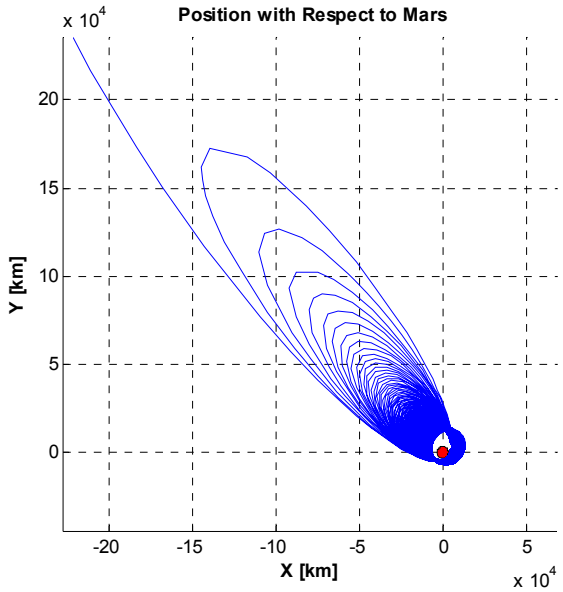

Figure 58. Spiralling around Mars in Mars reference frame (T6)

For this specific case, some relevant parameters of the transfer are displayed in figures 59 and 60, representing the Earth escape in the geocentric reference frame (figure 59) and Mars arrival in the Mars-centric reference frame (figure 60).

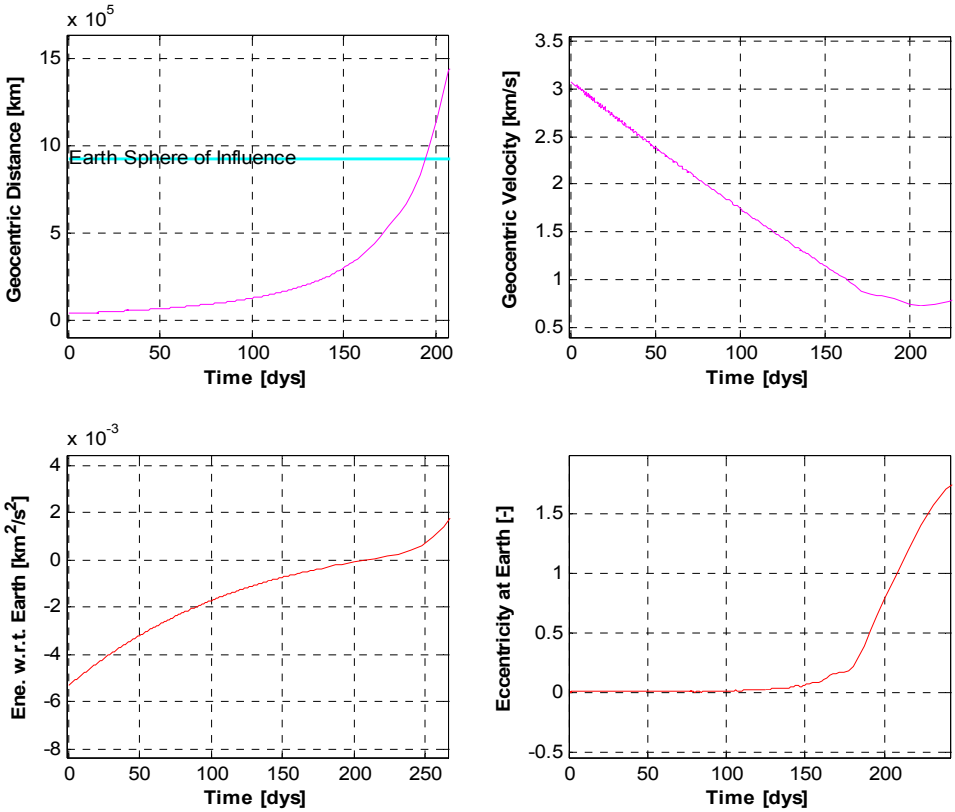

Figure 59. Parameters in Earth reference frame (T6)

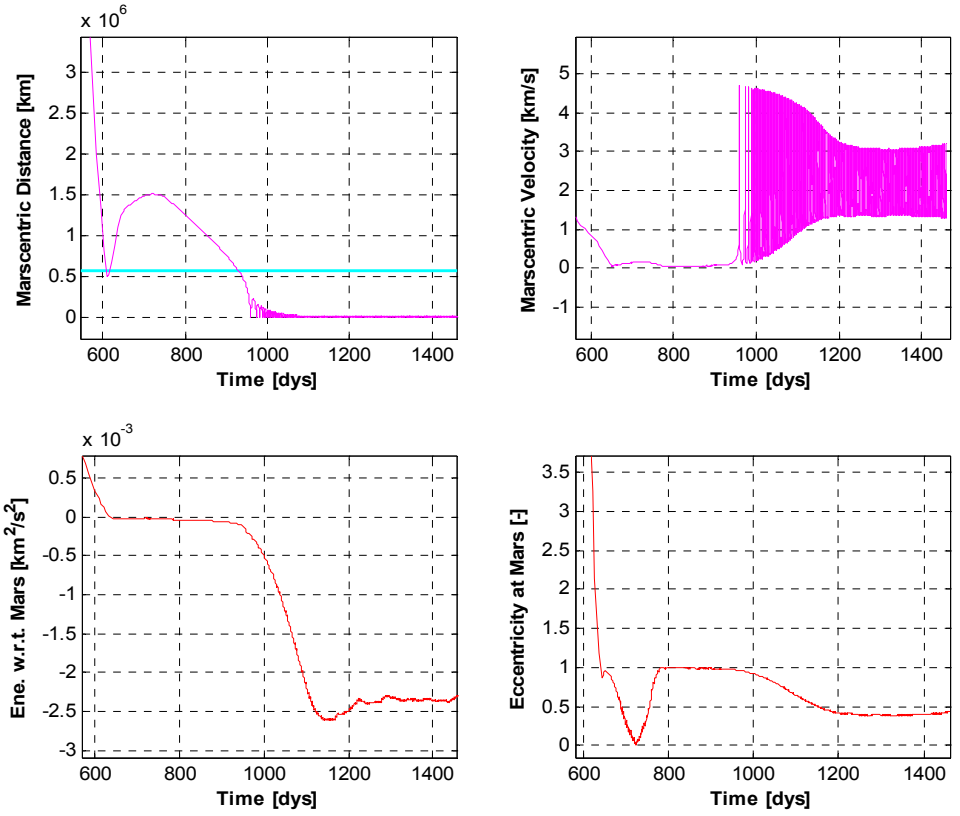

Figure 60. Parameters in Mars reference frame (T6)

Figure 59 shows how the geocentric velocity increases, the geocentric velocity decreases, the energy with respect to Earth steadily increases from negative to positive values and the eccentricity at the Earth goes from zero to more than one along the part of the transfer between the initial position and the spacecraft leaving the Earth's sphere of influence. Compared with the transfer of chapter 4.6 (figures 18 and 19) the oscillations in geocentric distance and velocity are less intense, mainly because of the shortest duration of the spiralling around the Earth.

Figure 60 shows how after entering the Mars sphere of influence, the energy with respect to Mars changes from positive to negative values. And it also shows the oscillations in marscentric velocity due to the spiralling around Mars or how the eccentricity at Mars takes values from zero to one until reaching a constant value around 0.5. Compared with the transfer of chapter 4.6 (from figures 23 to 26), it takes less time between reaching an eccentricity at Mars smaller than one and taking a constant value after the spiralling, which contributes to a better transfer time and demonstrates that it is possible to reach a determined orbit around Mars after a reasonable time.

6. CONCLUSIONS

This project has presented a code that computes a transfer from Earth to Mars within a gravitational four-body problem (Sun, Earth, Mars and spacecraft) including also the force exerted by the spacecraft's continuous thrust. After describing and further developing this code, a study of different transfers has been performed by varying several parameters of the simulations. These parameters are the initial position and velocity of the spacecraft, the departing orbit or the thruster characteristics of the engine used to perform the transfer. Several trajectories have been displayed in terms of transfer time and fuel mass consumed, and the cases when these variables have minimum values have been selected and further commented.

Although practically all missions to Mars have had or have chemical engines as main propulsion system, low-thrust engines constitute an alternative that deserves to be considered. Studies like this project indicate that transferring from the Earth to Mars using low-thrust propulsion technology is possible and has advantages.
Comparison of the two propulsion systems tells that the main advantage of low-thrust is the savings in propellant mass, which allows increasing the mass of payload that can be delivered to Mars. This fact can be of great interest when it translates into carrying more scientific instruments or to deliver larger rovers or equipment to the Mars surface.
The main drawback is the larger transfer time. Compared to the approximately 260 days that the chemical engines take to travel from the sphere of influence of the Earth to the sphere of influence of Mars, the existing low-thrust engines need more than a year in the best direct transfer case. Considering what happens inside the spheres of influence, the spiralling around Earth and Mars, the transfer time can increase above the 1000 days. There exists a compromise between the reduction of fuel mass and the increase of transfer time, and the selection of the best propulsion system could vary depending on the type of mission. Nevertheless, as further development and research are devoted to the technology of low-thrust engines, their thruster characteristics can increase considerably, making the system even more advantageous and reducing the problem of the high transfer time. Prove of that, the transfers simulated with the High Power Electric Propulsion thruster (HiPEP) offer considerably better results than any of the ion engines studied in this project.

It has to be also pointed out that the analysis performed in this project is only a preliminary study, where most of the assumptions were rather conservative. For example, the thrusters offer their maximum performance along the entire

trajectory. Moreover, the model consider only three types of thrust directions (tangential, anti-tangential and circumpherential).
Employing the work and main results of this study as a first step of a greater project, and improving the model in a more realistic way, would increase the significance of the results and would allow to extrapolate them to real missions and offer the possibility to take more advantage of the low-thrust propulsion systems technology.

6.1. Next Steps

Following the study performed in this project, there are a set of future developments that have been identified aiming at further improving the model considered and the preliminary analysis:

- The model should consider that the orbits of the Earth and Mars are not circular and coplanar, but elliptical and with different inclination respect to the ecliptic.
- Planetary ephemerides should be taken into account in order to perform an analysis regarding launch windows.
- Additional external perturbations should be added to the model, especially at the Earth escape and Mars capture.
- A more detailed model should be applied regarding the thruster performance.
- The model should consider a wider range of thrust directions for Earth escape, Mars capture and interplanetary phase, apart from tangential, anti-tangential or circumpherential.
- A further analysis regarding the variation of the initial mass of the spacecraft should be performed and studied.
- The same approach could be applied to other planets or near-Earth objects.

7. PROJECT BUDGET

The project budget can be addressed in two different approaches:

- The current budget of a mission to Mars
- The budget for developing this project in the frame of a final year's thesis.

The budget considered here will be the second approach, because the first budget would consider a lot more studies apart from the trajectory analysis like that presented in this project. Moreover, the budget corresponding to a Mars mission can considerably vary depending on the aims and objectives of the type of mission.

Focusing on the second approach, the final year's thesis, two main sources of cost have to be considered:

- The cost of computers and fungible material used to create the documents and collect the necessary information to develop the study.
- The man-hour cost of performing the project.

The total working time has been estimated as 420 hours and the man-hour cost considered has been the cost of a student working in internship regime.

The costs that are not included in this project budget are the cost of bibliography, which is assumed to be covered by the University, and the cost of software licenses used to perform simulations and calculations.
Table 16 presents the approximated project budget.

Project Budget			
Concept	Stationery	Man-hour	Total
Unit cost [€]	-	8	-
Total cost [€]	600	3360	**3960**

Table 16. Project Budget (costs are approximated and expressed in 2011 Euros)

The total cost of performing this study is 3960 €.

8. REFERENCES

[1]. MATLAB ® (Matrix Laboratory), Numerical Computing Environment and Programming Language, developed by MathWorks, http://www.mathworks.com/.

[2]. Sovey, J. S., Rawlin, V. K., and Patterson, M. J.: *Ion Propulsion Development Projects in U. S.: Space Electric Rocket Test 1 to Deep Space 1,* Journal of Propulsion and Power, Vol. 17, No. 3, May-June 2001.

[3]. Stephen Kemble: *Interplanetary Mission Analysis and Design*, Springer-Praxis, 2006.

[4]. Dr.Thawar T. Ariff: *Aerospace Technologies Advancements*, Chapter 12: Low-Thrust Propulsion Technologies, Mission Design, and Application, INTECH, 2010.

[5]. Prof. Manuel Martínez-Sánchez: Space Propulsion Lectures 13-14: Electrostatic Thrusters, Course 2009-10.

[6]. *Encyclopedia of Physical Science and Technology*, Third Edition, Volume 5, Academic Press, 2002.

[7]. Marc D. Rayman: *The successful conclusion of the Deep Space 1 Mission: Important results without a flashy title*, JPL, California Institute of Technology, 2003.

[8]. http://sci.esa.int/science-e/www/object/index.cfm?fobjectid=34216 (as of June 2011).

[9]. http://www.jaxa.jp/projects/sat/muses_c/index_e.html (as of June 2011).

[10]. http://www.esa.int/esaSC/ (as of June 2011).

[11]. http://mars.jpl.nasa.gov (as of June 2011).

[12]. http://www.spaceref.com/focuson/2001mars/overview (as of June 2011).

[13]. Richard R. Rieber: *Utilization of Ion Propulsion for Mars Orbiters*, Aerospace Engineering Sciences, University of Colorado, Boulder, Colorado 80309.

[14]. Sovey, J. S., Rawlin, V. K., and Patterson, M. J.: "Ion Propulsion Development Projects in U. S.: Space Electric Rocket Test 1 to Deep Space 1." Journal of Propulsion and Power, Vol. 17, No. 3, May-June 2001.

[15]. Dan M. Goebel and Ira Katz, "Fundamentals of Electric Propulsion: Ion and Hall Thrusters", Chapter 9: Flight Ion and Hall Thrusters, Jet Propulsion Laboratory California Institute of Technology, March 2008.

[16]. http://www2.qinetiq.com (as of August 2011).

[17]. http://cs.astrium.eads.net (as of August 2011).

[18]. Kazutaya Nishiyama, "30mN-Class Microwave Discharge Ion Thruster", IEPC 2003-62.

[19]. George R. Schmidt, Michael J. Patterson and Scott W. Benson: "The NASA Evolutionary Xenon Thruster (NEXT): The Next Step for U.S. Deep Space Propulsion", IAC-08-C4.4.2.

[20]. J. E. Foster et al. "The high power electric propulsion (hipep) ion thruster", In 40th Joint Propulsion Conference, 2004.

[PAGE INTENTIONALLY LEFT IN BLANK]

Escola Tècnica Superior d'Enginyeries
Industrial i Aeronàutica de Terrassa

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Study of Earth-to-Mars Transfers with Low-Thrust Propulsion

Annex 1: Matlab ® Codes

Author: Xavi López Hellín

Director: Dr. Elena Fantino

Collaborators: Pierpaolo Pergola

Place: ETSEIAT – UPC – Terrassa

Date: September 12, 2011

9. ANNEX 1: MATLAB ® CODES

The required simulations in this project have been performed using the Matlab ® environment. For completeness of the project and further development, the code (functions and scripts) are presented in this annex.

9.1. Main function "EarthMarsLT"

This is the main function of the global code, the script is called EarthMarsLT.

```
close all
clear all
clc
tic

global Th Isp g0
global mu_earth mu_sun mu_mars
global SOI_E SOI_M a_mars R_mars a_earth R_earth
global DU TU MU
global h0 hf
global I Parameters
global Wbar tspan StopW
global n tollE dM
global Opt flagOpt flagOptStop flagtime transfer_time M_end

I = 0;
flagOpt = 1;
flagtime = 1;

%% Input Data ------------------------------------------------------------------
prompt=['Electric Power [W]        ';'Specific Impulse [s]      ';...
    'Thrust Efficiency [-]     ';'Initial Mass [kg]         ';...
    'Perigee Altitude [km]     ';'Semi-major Axis [km]      ';...
    'Initial True Anomaly [deg]';'Final Altitude [km]       ';...
    'Transfer Time [yrs]       ';'Initial Earth Angle [deg] ';...
    'Starting Thrust at n S.I. '; 'Tollerance on Energy [ ]  ';
    'Final Thrust anomaly [deg]';'Number of Iterations      '];
Def=['5886 ';'3000 ';'0.5  ';'1500 ';'300  ';...
    '24582';'180  ';'400  ';'4    ';'0    ';...
    '8    ';'-0.1 ';'45   ';'2    '];

prompt=cellstr(prompt);
Def=cellstr(Def);
answer = inputdlg(prompt,'Input Data',1,Def,'on');

temp1=char(answer(1));
temp2=char(answer(2));
temp3=char(answer(3));
temp4=char(answer(4));
temp5=char(answer(5));
temp6=char(answer(6));
temp7=char(answer(7));
temp8=char(answer(8));
temp9=char(answer(9));
temp10=char(answer(10));
temp11=char(answer(11));
temp12=char(answer(12));
temp13=char(answer(13));
temp14=char(answer(14));

Pow = str2num(temp1); %Electric Power [W]
Isp = str2num(temp2); %Specific Impulse [s]
eta = str2num(temp3); %Thrust Efficiency [-]
M_Init = str2num(temp4); %Initial Mass [kg]
h0 = str2num(temp5); %Initial orbit - Perigee Altitude [km]
a0 = str2num(temp6); %Initial orbit - Semi-major Axis [km]
```

```
ScAng = str2num(temp7); %Initial True Anomaly [deg]
hf = str2num(temp8); %Final Altitude [km]
Tend = str2num(temp9); %End of the simulation [yrs]
EarthAng = str2num(temp10); %Initial Earth Angle [deg]
n = str2num(temp11); %Starting deceleration thrust at nSOI
tollE = str2num(temp12); %Tollerance on Energy [ ]
dM = str2num(temp13); %Final Thrust anomaly [deg]
iterations = str2num(temp14); %Number of Iterations
MarsAng = 90; % Initial Guess of Mars angle

%% Gravitational Constants-------------------------------------------------------
% Sun
mu_sun = 1.32712428e11;  % km^3/s^2
Msun = 1.988435e30;      % kg
R_sun = 696000;          % km
au = 149597870;          % km
% Earth
R_earth = 6378.1363;     % km
mu_earth = 398600.4415; % km^3/s^2
a_earth = 1*au;          % km
Mearth = 5.9742e24;      % kg
SOI_E = (Mearth/Msun)^(2/5) * a_earth; %km
g0 = 9.81;  % m/s^2
% Mars
R_mars = 3397;           % km
mu_mars = 4.305e4;       % km^3/s^2
a_mars = 1.52367934*au;  % km
Mmars = 6.4191e23;       % kg
SOI_M = (Mmars/Msun)^(2/5) * a_mars; %km

%% Adimensional Units-----------------------------------------------------------
DU = au;
TU = sqrt(DU^3/mu_sun);
MU = M_Init;
mu_sun = mu_sun * (TU^2/DU^3);   % =1
mu_earth = mu_earth * (TU^2/DU^3); %mu_earth/mu_sun
mu_mars = mu_mars * (TU^2/DU^3); %mu_mars/mu_sun

a_earth = a_earth / DU;
a_mars = a_mars / DU;
R_earth = R_earth / DU;
R_mars = R_mars / DU;
R_sun = R_sun / DU;
SOI_E = SOI_E/DU;
SOI_M = SOI_M/DU;
h0 = h0/DU;
a0 = a0/DU;
hf = hf/DU;

Th = 2*eta*Pow/(Isp*g0); % N
g0 = g0 * (TU^2/DU) / 1000;
Isp = Isp / TU;
M_Init = M_Init / MU;
Th = Th * (TU^2/(MU*DU)) / 1000;

EarthAng = deg2rad(EarthAng);
ScAng = deg2rad(ScAng);
MarsAng = deg2rad(MarsAng);
dM = deg2rad(dM);

Tend = Tend * (86400*365/TU);
tspan = [0 Tend];

for i = 1:iterations
    disp(['Iterantion Number: ', num2str(i)])

%% Initial Mars Angle Identification-------------------------------------------
    flagOptStop = 1;   % 1 to stop the simulationat at ...
    % the first Mars encounter and identify the Mars initial angle.

% Initial States
    R_E_init = [a_earth*cos(EarthAng) a_earth*sin(EarthAng)];
    V_E_init = [-sqrt(mu_sun/a_earth)*sin(EarthAng) ...
        sqrt(mu_sun/a_earth)*cos(EarthAng)];
```

```
    R_M_init = [a_mars*cos(MarsAng) a_mars*sin(MarsAng)];
    V_M_init = [-sqrt(mu_sun/a_mars)*sin(MarsAng) ...
        sqrt(mu_sun/a_mars)*cos(MarsAng)];
    R_sc_temp = [(R_earth + h0)*cos(ScAng) (R_earth + h0)*sin(ScAng)];
    R_sc_init = R_E_init + R_sc_temp;
    V_sc_temp = [-sqrt(2*mu_earth/(R_earth + h0)-mu_earth/a0)*sin(ScAng)...
        sqrt(2*mu_earth/(R_earth + h0) - mu_earth/a0)*cos(ScAng)];
    V_sc_init = V_E_init + V_sc_temp;
    x0 = [R_sc_init V_sc_init M_Init R_E_init V_E_init R_M_init V_M_init];

% Integration
    refine = 1;
    opts = odeset('RelTol',1e-4,'AbsTol',1e-4, ...
        'stats','off','Refine',refine,'OutputFcn',@IntFcn);
    [t,x] = ode23(@EQmotion, tspan, x0, opts);

% Trajectory Computation
    Previous_MarsAng=MarsAng;
    clear MarsAng
    MarsAng = Opt.Theta_0;

% Clear Variables
    clear t x x0 R_M_init V_M_init transfer_time M_end
    clear global I Parameters Opt
    clear global flagOpt flagOptStop Wbar StopW flagtime
    global I Parameters Opt flagOpt flagOptStop Wbar StopW
    global flagtime transfer_time M_end

    I = 0;
    flagOpt = 1;
    flagtime =1;
    flagOptStop = 0; % Computation continuation
    i = i + 1;

%Reduce the number of iterations
    if abs(MarsAng-Previous_MarsAng)<0.01
        break
    end
end

%% Final Trajectory Computation-------------------------------------------------------
% New Mars State
R_M_init = [a_mars*cos(MarsAng) a_mars*sin(MarsAng)];
V_M_init = [-sqrt(mu_sun/a_mars)*sin(MarsAng) sqrt(mu_sun/a_mars)*cos(MarsAng)];
x0 = [R_sc_init V_sc_init M_Init R_E_init V_E_init R_M_init V_M_init];

% Integration
[t,x] = ode23(@EQmotion, tspan, x0, opts);

% Results
st = sprintf('Transfer Time: %.2f [days]',transfer_time*TU/86400);
disp(st);
disp('-----------------------------------')
mfuel=(M_Init-M_end)*MU;
st = sprintf('Fuel Mass consumed: %.2f [kg]',mfuel);
disp(st);
disp('-----------------------------------')

%% Variable Assignment---------------------------------------------------------------
R_sc = x(:,1:2);
V_sc = x(:,3:4);
M = x(:,5);
R_E = x(:,6:7);
V_E = x(:,8:9);
R_M = x(:,10:11);
V_M = x(:,12:13);
R_sc_M = R_sc - R_M;
V_sc_M = V_sc - V_M;
R_sc_E = R_sc - R_E;
V_sc_E = V_sc - V_E;
POS_H = sqrt(x(:,1).^2 + x(:,2).^2);
VEL_H = sqrt(x(:,3).^2 + x(:,4).^2);
POS_E = sqrt(R_sc_E(:,1).^2 + R_sc_E(:,2).^2);
VEL_E = sqrt(V_sc_E(:,1).^2 + V_sc_E(:,2).^2);
```

```
POS_M = sqrt(R_sc_M(:,1).^2 + R_sc_M(:,2).^2);
VEL_M = sqrt(V_sc_M(:,1).^2 + V_sc_M(:,2).^2);
MAS = x(:,5);

%% Plots-------------------------------------------------------------------------
alpha_temp = 0:2*pi/99:2*pi;
x_temp = cos(alpha_temp);
y_temp = sin(alpha_temp);

% Heliocentric transfer
figure()
grid on, hold on, axis equal
title('\bf Position with Respect to Sun')
xlabel('\bfX [DU]')
ylabel('\bfY [DU]')
plot(R_sc(:,1),R_sc(:,2),'b-')
plot(R_E(:,1),R_E(:,2),'g-','linewidth',1.4)
plot(R_M(:,1),R_M(:,2),'r-','linewidth',1.4)
plot(R_sc(1,1),R_sc(1,2),'.c')
plot(R_E(1,1),R_E(1,2),'.y')
plot(R_M(1,1),R_M(1,2),'.g')
plot(Opt.R_sc(1),Opt.R_sc(2),'oc')
plot(Opt.R_M(1),Opt.R_M(2),'og')
fill(R_sun*x_temp,R_sun*y_temp,'y')
legend('S/C','Earth','Mars')

% Position in Mars & Earth reference frame
figure()
grid on, hold on, axis square
title('\bf Position with Respect to Earth')
xlabel('\bfX [km]')
ylabel('\bfY [km]')
fill(R_earth*x_temp*DU,R_earth*y_temp*DU,'g')
plot(R_sc_E(:,1)*DU,R_sc_E(:,2)*DU,'linewidth',1.4)

figure()
grid on, hold on, axis square
title('\bf Position with Respect to Mars')
xlabel('\bfX [km]')
ylabel('\bfY [km]')
fill(R_mars*x_temp*DU,R_mars*y_temp*DU,'r')
plot(R_sc_M(:,1)*DU,R_sc_M(:,2)*DU,'linewidth',1.4)

% Heliocentric Distance and Instantaneous Mass
figure ()
subplot(1,2,1)
grid on, hold on, box on
xlabel('\bfTime [dys]')
ylabel('\bfHeliocentric Distance [km]')
fill([t(1)*TU/86400,t(1)*TU/86400,t(end)*TU/86400,t(end)*TU/86400],...
    [(SOI_E + a_earth)*DU,(-SOI_E + a_earth)*DU,...
    (-SOI_E + a_earth)*DU,(SOI_E + a_earth)*DU],'c')
fill([t(1)*TU/86400,t(1)*TU/86400,t(end)*TU/86400,t(end)*TU/86400],...
    [(SOI_M + a_mars)*DU,(-SOI_M + a_mars)*DU,...
    (-SOI_M + a_mars)*DU,(SOI_M + a_mars)*DU],'c')
text(t(1)*TU/86400,(SOI_E + a_earth)*DU,'Earth Sphere of Influence')
text(t(1)*TU/86400,(SOI_M + a_mars)*DU,'Mars Sphere of Influence')
plot(t*TU/86400,POS_H*DU,'m','linewidth',1.4)

subplot(1,2,2)
grid on, hold on, box on
xlabel('\bfTime [dys]')
ylabel('\bfInstantaneous Mass [kg]')
plot(t*TU/86400,MAS*MU,'m','linewidth',1.4)

% Parameters in Earth reference frame
figure ()
subplot(2,3,1)
grid on, hold on, box on
xlabel('\bfTime [dys]')
ylabel('\bfGeocentric Distance [km]')
plot([t(1)*TU/86400 t(end)*TU/86400],[SOI_E*DU SOI_E*DU],'c','linewidth',1.5)
text(t(1)*TU/86400,(SOI_E)*DU,'Earth Sphere of Influence')
plot(t*TU/86400,POS_E*DU,'m','linewidth',1.4)
```

```
subplot(2,3,2)
grid on, hold on, box on
xlabel('\bfTime [dys]')
ylabel('\bfGeocentric Velocity [km/s]')
plot(t*TU/86400,VEL_E*DU/TU,'m','linewidth',1.4)

subplot(2,3,3)
grid on, hold on, box on
xlabel('\bfTime [dys]')
ylabel('\bfEne. w.r.t. Earth [km^2/s^2]')
plot(t(2:end)*TU/86400,Parameters.eneE,'r','linewidth',1.4)

subplot(2,3,4)
grid on, hold on, box on
xlabel('\bfTime [dys]')
ylabel('\bfPerigee Radius around Earth [km]')
plot(t(2:end)*TU/86400,Parameters.RpE*DU,'r','linewidth',1.4)

subplot(2,3,5)
grid on, hold on, box on
xlabel('\bfTime [dys]')
ylabel('\bfApogee Radius around Earth [km]')
plot(t(2:end)*TU/86400,Parameters.RaE*DU,'r','linewidth',1.4)

subplot(2,3,6)
grid on, hold on, box on
xlabel('\bfTime [dys]')
ylabel('\bfEccentricity at Earth [-]')
plot(t(2:end)*TU/86400,Parameters.eccE,'r','linewidth',1.4)

% Parameters in Mars frame
figure ()
subplot(2,3,1)
grid on, hold on, box on
xlabel('\bfTime [dys]')
ylabel('\bfMarscentric Distance [km]')
plot([t(1)*TU/86400 t(end)*TU/86400],[SOI_M*DU SOI_M*DU],'c','linewidth',1.5)
text(t(1)*TU/86400,(SOI_M)*DU,'Mars Sphere of Influence')
plot(t*TU/86400,POS_M*DU,'m','linewidth',1.4)

subplot(2,3,2)
grid on, hold on, box on
xlabel('\bfTime [dys]')
ylabel('\bfMarscentric Velocity [km/s]')
plot(t*TU/86400,VEL_M*DU/TU,'m','linewidth',1.4)

subplot(2,3,3)
grid on, hold on, box on
xlabel('\bfTime [dys]')
ylabel('\bfEne. w.r.t. Mars [km^2/s^2]')
plot(t(2:end)*TU/86400,Parameters.eneM,'r','linewidth',1.4)

subplot(2,3,4)
grid on, hold on, box on
xlabel('\bfTime [dys]')
ylabel('\bfPerigee Radius around Mars [km]')
plot(t(2:end)*TU/86400,Parameters.RpM*DU,'r','linewidth',1.4)

subplot(2,3,5)
grid on, hold on, box on
xlabel('\bfTime [dys]')
ylabel('\bfApogee Radius around Mars [km]')
plot(t(2:end)*TU/86400,Parameters.RaM*DU,'r','linewidth',1.4)

subplot(2,3,6)
grid on, hold on, box on
xlabel('\bfTime [dys]')
ylabel('\bfEccentricity at Mars [-]')
plot(t(2:end)*TU/86400,Parameters.eccM,'r','linewidth',1.4)

toc
%% ----------------------------------------------------------------------
```

9.2. Equations of motion "EQmotion"

Function containing the equations of motion of the problem, it is called EQmotion.

```
function ydot = EQmotion(t, y)
ydot = zeros(size(y));

global Th Isp g0
global mu_earth mu_sun mu_mars
global SOI_E SOI_M a_mars R_mars a_earth R_earth
global DU TU MU
global h0 hf
global I Parameters
global Wbar tspan StopW
global n tollE dM
global Opt flagOpt flagOptStop

%% Variables assignment
R_sc = y(1:2);
V_sc = y(3:4);
M = y(5);
R_E = y(6:7);
V_E = y(8:9);
R_M = y(10:11);
V_M = y(12:13);

%% 2-body perturbed equations
A_sc_grav = -mu_sun*R_sc/norm(R_sc)^3 + ... %Acceleration due to sun
    mu_earth*((R_E-R_sc)/norm(R_E-R_sc)^3-R_E/norm(R_E)^3)...%Perturbation (Earth)
    + mu_mars*((R_M-R_sc)/norm(R_M-R_sc)^3-R_M/norm(R_M)^3); %Perturbation (Mars)

%% Adding a Thrust
% Earth Escape
zeta = norm(V_sc)^2/2 - mu_sun/norm(R_sc);  % Helio specific energy
e_vec = ((norm(V_sc)^2  - mu_sun/norm(R_sc))*R_sc - ...
    dot(R_sc,V_sc)*V_sc)/mu_sun; % eccentricity vector
ecc = norm(e_vec);   % Helio eccentricity
Ra = -mu_sun/(2*zeta)*(1+ecc);   % Helio apogee radius
if norm(R_sc-R_E) < SOI_E && Ra < a_mars
    A_sc_F = Th/M*(V_sc-V_E)/norm(V_sc-V_E);  % Tangential Thrust
    mdot = -Th/(Isp*g0);

    % Mars Capture
elseif norm(R_sc-R_M) < (n*SOI_M)
    zeta_M = norm(V_sc-V_M)^2/2 - mu_mars/norm(R_sc-R_M);
    V_sc_M = V_sc - V_M;
    R_sc_M = R_sc - R_M;
    h_M = cross([R_sc_M;0],[V_sc_M;0]);
    e_vec_M = ((norm(V_sc_M)^2  - mu_mars/norm(R_sc_M))*R_sc_M -
dot(R_sc_M,V_sc_M)*V_sc_M)/mu_mars;
    ecc_M = norm(e_vec_M);
    Rp_M = -mu_mars/(2*zeta_M)*(1-ecc_M);
    Ra_M = -mu_mars/(2*zeta_M)*(1+ecc_M);
    nu_M = acos(dot(e_vec_M,R_sc_M)/(ecc_M*norm(R_sc_M))); %True anomaly [rad]
    if dot(R_sc_M,V_sc_M) < 0  %Checking for quadrant
        nu_M = 2*pi - nu_M;
    end
    if ecc_M <= 1
        E_M = atan2(sin(nu_M)*(1-ecc_M^2)^.5,ecc_M+cos(nu_M));%rad (ecc. anomaly)
        M_M = mod(E_M-ecc_M*sin(E_M),2*pi); %rad (mean anomally)
    elseif ecc_M > 1
        E_M = atanh((sin(nu_M)*((ecc_M^2)-1)^0.5)/(ecc_M+cos(nu_M))); %rad
        M_M = mod(E_M-ecc_M*sinh(E_M),2*pi); %rad
    end

    if Rp_M < (R_mars + hf)
        r = norm(R_sc_M);
        D1 = [0 -1 0; 1 0 0; 0 0 1]*[R_sc_M;0]/r; % Circumpherential Thrust
        A_sc_F = Th/M*D1(1:2);  %Accel due to thrust
        mdot = -Th/(Isp*g0);
    elseif zeta_M >= 0
```

```
        A_sc_F = -Th/M*(V_sc_M)/norm(V_sc_M);  % Anti-tangential Thrust
        mdot = -Th/(Isp*g0);
    elseif zeta_M > (tollE * (TU^2/DU^2)) && M_M < (pi+dM) && M_M > (pi-dM)
        A_sc_F = Th/M*(V_sc_M)/norm(V_sc_M);  % Tangential Thrust
        mdot = -Th/(Isp*g0);
    elseif Ra_M > (R_mars + hf) && M_M > (2*pi-dM) && M_M < (2*pi+dM)
        A_sc_F = -Th/M*(V_sc_M)/norm(V_sc_M);  % Anti-tangential Thrust
        mdot = -Th/(Isp*g0);
    elseif Rp_M > (R_mars + hf)
        A_sc_F = -Th/M*(V_sc_M)/norm(V_sc_M);  % Anti-tangential Thrust
        mdot = -Th/(Isp*g0);
    else
        A_sc_F = [0;0];
        mdot = 0;
    end

    % Heliocentric transfer
elseif Ra < a_mars
    r = norm(R_sc);
    D1 = [0 -1 0; 1 0 0; 0 0 1]*[R_sc;0]/r; % Circumpherential Th
    A_sc_F = Th/M*D1(1:2);  %Accel due to thrust
    mdot = -Th/(Isp*g0);

    % No thrust
else
    A_sc_F = [0;0];
    mdot = 0;
end

%% complete acceleration vector
A_sc = A_sc_grav + A_sc_F;

A_E = -mu_sun*R_E/norm(R_E)^3;
A_M = -mu_sun*R_M/norm(R_M)^3;

ydot(1:2) = V_sc;
ydot(3:4) = A_sc;
ydot(5) = mdot;
ydot(6:7) = V_E;
ydot(8:9) = A_E;
ydot(10:11) = V_M;
ydot(12:13) = A_M;
%% --------------------------------------------------------------------------
```

9.3. Integration Function "IntFcn"

Function called at each step during the integration period, it is called IntFcn.

```
function status = IntFcn(t,y,flag)

global Th Isp g0
global mu_earth mu_sun mu_mars
global SOI_E SOI_M a_mars R_mars a_earth R_earth
global DU TU MU
global h0 hf
global I Parameters
global Wbar tspan StopW
global n tollE dM
global Opt flagOpt flagOptStop flagtime transfer_time M_end

%% Waitbar
if (strcmp(flag,'init')) %if it's the first step
    Wbar = waitbar(0,'Please wait...','CreateCancelBtn',@Stopbutton); %creates a
waiting bar with possibility to cancel
    status = 0;
elseif (strcmp(flag,'done')) %if it's the last step
    if ishandle(Wbar)
        close(Wbar);
    end
else
```

```
    status = 0;
    WbarPer = t / tspan(end);
    waitbar(WbarPer) %advamce of the waiting bar according to the time fraction
end
if length(t)==1
    I = I + 1;

%% Orbital Parameters
    % Variables assignment
    R_sc = y(1:2);
    V_sc = y(3:4);
    M = y(5);
    R_E = y(6:7);
    V_E = y(8:9);
    R_M = y(10:11);
    V_M = y(12:13);

%% Values Computation
    Parameters.eneH(I) = norm(V_sc)^2/2 - mu_sun/norm(R_sc);%Helio specific energy
    e_vec =((norm(V_sc)^2 - mu_sun/norm(R_sc))*R_sc - dot(R_sc,V_sc)*V_sc)/mu_sun;
    Parameters.eccH(I) = norm(e_vec); %Helio eccentricity
    Parameters.RaH(I) = -mu_sun/(2*Parameters.eneH(I))*...
        (1+Parameters.eccH(I)); %Helio apogee radius
    Parameters.VatMars(I) = (2*(mu_sun/a_mars + Parameters.eneH(I)))^.5;
    Parameters.nu_mars(I) = acos(dot(e_vec,R_sc)/...
        (Parameters.eccH(I)*norm(R_sc))); %True anomaly [rad]
    if dot(R_sc,V_sc) < 0  %Checking for quadrant
        Parameters.nu_mars(I) = 2*pi - Parameters.nu_mars(I);
    end
    Parameters.eneE(I) = norm(V_sc-V_E)^2/2 - ...
        mu_earth/norm(R_sc-R_E); %Geo specific energy
    e_vec_E = ((norm(V_sc-V_E)^2 - mu_earth/norm(R_sc-R_E))*(R_sc-R_E) - ...
        dot((R_sc-R_E),(V_sc-V_E))*(V_sc-V_E))/mu_earth;
    Parameters.eccE(I) = norm(e_vec_E);
    Parameters.RpE(I) = -mu_earth/(2*Parameters.eneE(I))*(1-Parameters.eccE(I));
    Parameters.RaE(I) = -mu_earth/(2*Parameters.eneE(I))*(1+Parameters.eccE(I));

    Parameters.eneM(I) = norm(V_sc-V_M)^2/2 - ...
        mu_mars/norm(R_sc-R_M); %Mars specific energy
    V_sc_M = V_sc - V_M;
    R_sc_M = R_sc - R_M;
    e_vec_M = ((norm(V_sc_M)^2 - mu_mars/norm(R_sc_M))*R_sc_M - ...
        dot(R_sc_M,V_sc_M)*V_sc_M)/mu_mars;
    Parameters.eccM(I) = norm(e_vec_M);
    Parameters.RpM(I) = -mu_mars/(2*Parameters.eneM(I))*(1-Parameters.eccM(I));
    Parameters.RaM(I) = -mu_mars/(2*Parameters.eneM(I))*(1+Parameters.eccM(I));
    Parameters.nu_M(I) = acos(dot(e_vec_M,R_sc_M)/...
        (Parameters.eccM(I)*norm(R_sc_M))); %True anomaly [rad]

    if dot(R_sc_M,V_sc_M) < 0  %Checking for quadrant
        Parameters.nu_M(I) = 2*pi - Parameters.nu_M(I);
    end
    if Parameters.eccM(I) <= 1
        Parameters.E_M(I) = atan2(sin(Parameters.nu_M(I))*...
            (1-Parameters.eccM(I)^2)^.5,...
            Parameters.eccM(I)+cos(Parameters.nu_M(I))); %rad (eccentric anomally)
        Parameters.M_M(I) = mod(Parameters.E_M(I)-...
            Parameters.eccM(I)*sin(Parameters.E_M(I)),2*pi); %rad (mean anomally)
    elseif Parameters.eccM(I) > 1
        Parameters.E_M(I) = atanh((sin(Parameters.nu_M(I))*...
            ((Parameters.eccM(I)^2)-1)^.5)/(Parameters.eccM(I)+...
            cos(Parameters.nu_M(I)))); %rad (eccentric anomally)
        Parameters.M_M(I) = mod(Parameters.E_M(I)-...
            Parameters.eccM(I)*sinh(Parameters.E_M(I)),2*pi); %rad (mean anomally)
    end

%% Identification of initial Mars Angle
    if norm(R_sc)*DU >= (a_mars - SOI_M)*DU && flagOpt == 1
        flagOpt = 0;
        Opt.Dt = t;
        Opt.R_sc = R_sc;
        Opt.R_M = R_M;
        Opt.Theta_sc = acos(dot(R_sc,[1 0]')/norm(R_sc));
        Opt.Theta_M = acos(dot(R_M,[1 0]')/norm(R_M));
```

```
        % Check for Quadrants
        if R_sc(2) <= 0   % 3° & 4°
            Opt.Theta_sc = 2*pi - Opt.Theta_sc;
        end
        if R_M(2) <= 0   % 3° & 4°
            Opt.Theta_M = 2*pi - Opt.Theta_M;
        end
        omega_M = sqrt(mu_sun/a_mars)/a_mars;
        Opt.Theta_0 = Opt.Theta_sc - mod(omega_M*t,2*pi);
        if flagOptStop == 1
            disp('-----------------------------------')
            st = sprintf('Optimal Initial Mars Angle: %.14f [rad]',Opt.Theta_0);
            disp(st);
            disp('-----------------------------------')
            status = 1;
        end
    end

%% Transfer Time
    if Parameters.eccM(I)<1 && flagtime == 1
        flagtime = 0;
        transfer_time = t;
        M_end=M;
    end
    if length(transfer_time)~=1
        transfer_time = 0;
        M_end=1;
    end
    if norm (R_sc_M) <= (R_mars + hf)
        disp('-----------------------------------')
        disp('Target Altitude Achieved')
        disp('-----------------------------------')
        status = 1;
    end

%% Check for impact
    if norm(R_sc_M) <= R_mars
        disp('-----------------------------------')
        disp('Impact with the Surface')
        disp('-----------------------------------')
        status = 1;
    end

%% Check for ecceding initial mass
    if M <= 0
        disp('-----------------------------------')
        disp('S/C mass finished')
        disp('-----------------------------------')
        status = 1;
    end
end

%% Stop Button
if StopW ==1
    status = 1;
end

% --------------------------------------------------------------------------
function Stopbutton(varargin)
global StopW Wbar
StopW = 1;
delete (Wbar);
% --------------------------------------------------------------------------
```

9.4. Auxiliary functions

"deg2rad" and "rad2deg" are functions that change input angles from degrees to radians and from radians to degrees, respectively.

```
function [ans] = deg2rad(x)
ans = x * pi/180;

function [ans] = rad2deg(x)
ans = x * 180/pi;
```

9.5. Scripts

The scripts presented in this section make used of the above-mentioned functions and have been used to perform the studies described in chapter 5.

- First, the script corresponding to chapter 5.2, when varying h_0, a_0, and $\theta_{SC\text{-}init}$, is shown:

```
tic
global J Data
J = 1;
for ii = 100:200:700
    for jj = 22500:1000:25500
        for kk = 0:45:315
```

Following this piece of script there is the main function EarthMarsLT with the difference that now the input data is not manually introduced, but it is set with the predetermined values and the triple for-loop inputs:

```
Pow = 5886;
Isp = 3000;
eta = 0.5;
M_Init = 1500;
h0 = ii;
a0 = jj;
ScAng = kk;
hf = 400;
Tend = 4;
EarthAng = 0;
n = 8;
tollE = -0.1;
dM = 45;
iterations = 2;
MarsAng = 90;
```

Then, the main function EarthMarsLT continues with the same script previously presented, but without the plots part. The script ends with the following code:

```
%Gathering data of each transfer of the loop
        Data.Transfer_Time(J) = transfer_time*TU/86400;
        Data.mfuel(J) = mfuel;
        Data.h0(J)=ii;
        Data.a0(J)=jj;
        Data.ScAng(J)=kk;
        Data.R_sc_init_X(J) = R_E_init(1) + R_sc_temp(1);
        Data.R_sc_init_Y(J) = R_E_init(2) + R_sc_temp(2);
        Data.R_sc_init(J)=sqrt((Data.R_sc_init_X(J)^2)+...
            (Data.R_sc_init_Y(J)^2));
        Data.V_sc_init_X(J) = V_E_init(1) + V_sc_temp(1);
        Data.V_sc_init_Y(J) = V_E_init(2) + V_sc_temp(2);
        Data.V_sc_init(J)=sqrt((Data.V_sc_init_X(J)^2)+...
            (Data.V_sc_init_Y(J)^2));
        clear t x x0 R_M_init V_M_init i
        clear global I Parameters Opt flagOpt flagOptStop
        clear global Wbar StopW flagtime mfuel transfer_time
        J=J+1;
        end
```

```
    end
end

J=1;
for J=1:128
    if Data.ScAng(J)==0
        Data.ScAng(J)=0.00001;
    end
    if Data.Transfer_Time(J)==0 && Data.mfuel(J)==0
        Data.ScAng(J)=0;
    end
end
%a0=22500km
figure()
grid on, hold on
title('\bf Initial Semi-major Axis a0=22500km')
xlabel('\bfInitial True Anomaly [deg]')
ylabel('\bfTransfer Time [days]')
plot(nonzeros(Data.ScAng(1:8)),nonzeros(Data.Transfer_Time(1:8)),'bo-')
plot(nonzeros(Data.ScAng(33:40)),nonzeros(Data.Transfer_Time(33:40)),'go-')
plot(nonzeros(Data.ScAng(65:72)),nonzeros(Data.Transfer_Time(65:72)),'ro-')
plot(nonzeros(Data.ScAng(97:104)),nonzeros(Data.Transfer_Time(97:104)),'co-')
legend('h0=100km','h0=300km','h0=500km','h0=700km')
figure()
grid on, hold on
title('\bf Initial Semi-major Axis a0=22500km')
xlabel('\bfInitial True Anomaly [deg]')
ylabel('\bfFuel Mass consumed [kg]')
plot(nonzeros(Data.ScAng(1:8)),nonzeros(Data.mfuel(1:8)),'bo-')
plot(nonzeros(Data.ScAng(33:40)),nonzeros(Data.mfuel(33:40)),'go-')
plot(nonzeros(Data.ScAng(65:72)),nonzeros(Data.mfuel(65:72)),'ro-')
plot(nonzeros(Data.ScAng(97:104)),nonzeros(Data.mfuel(97:104)),'co-')
legend('h0=100km','h0=300km','h0=500km','h0=700km')

%a0=23500km
figure()
grid on, hold on
title('\bf Initial Semi-major Axis a0=23500km')
xlabel('\bfInitial True Anomaly [deg]')
ylabel('\bfTransfer Time [days]')
plot(nonzeros(Data.ScAng(9:16)),nonzeros(Data.Transfer_Time(9:16)),'bo-')
plot(nonzeros(Data.ScAng(41:48)),nonzeros(Data.Transfer_Time(41:48)),'go-')
plot(nonzeros(Data.ScAng(73:80)),nonzeros(Data.Transfer_Time(73:80)),'ro-')
plot(nonzeros(Data.ScAng(105:112)),nonzeros(Data.Transfer_Time(105:112)),'co-')
legend('h0=100km','h0=300km','h0=500km','h0=700km')
figure()
grid on, hold on
title('\bf Initial Semi-major Axis a0=23500km')
xlabel('\bfInitial True Anomaly [deg]')
ylabel('\bfFuel Mass consumed [kg]')
plot(nonzeros(Data.ScAng(9:16)),nonzeros(Data.mfuel(9:16)),'bo-')
plot(nonzeros(Data.ScAng(41:48)),nonzeros(Data.mfuel(41:48)),'go-')
plot(nonzeros(Data.ScAng(73:80)),nonzeros(Data.mfuel(73:80)),'ro-')
plot(nonzeros(Data.ScAng(105:112)),nonzeros(Data.mfuel(105:112)),'co-')
legend('h0=100km','h0=300km','h0=500km','h0=700km')

%a0=24500km
figure()
grid on, hold on
title('\bf Initial Semi-major Axis a0=24500km')
xlabel('\bfInitial True Anomaly [deg]')
ylabel('\bfTransfer Time [days]')
plot(nonzeros(Data.ScAng(17:24)),nonzeros(Data.Transfer_Time(17:24)),'bo-')
plot(nonzeros(Data.ScAng(49:56)),nonzeros(Data.Transfer_Time(49:56)),'go-')
plot(nonzeros(Data.ScAng(81:88)),nonzeros(Data.Transfer_Time(81:88)),'ro-')
plot(nonzeros(Data.ScAng(113:120)),nonzeros(Data.Transfer_Time(113:120)),'co-')
legend('h0=100km','h0=300km','h0=500km','h0=700km')
figure()
grid on, hold on
title('\bf Initial Semi-major Axis a0=24500km')
xlabel('\bfInitial True Anomaly [deg]')
ylabel('\bfFuel Mass consumed [kg]')
plot(nonzeros(Data.ScAng(17:24)),nonzeros(Data.mfuel(17:24)),'bo-')
plot(nonzeros(Data.ScAng(49:56)),nonzeros(Data.mfuel(49:56)),'go-')
```

```
plot(nonzeros(Data.ScAng(81:88)),nonzeros(Data.mfuel(81:88)),'ro-')
plot(nonzeros(Data.ScAng(113:120)),nonzeros(Data.mfuel(113:120)),'co-')
legend('h0=100km','h0=300km','h0=500km','h0=700km')

%a0=25500km
figure()
grid on, hold on
title('\bf Initial Semi-major Axis a0=25500km')
xlabel('\bfInitial True Anomaly [deg]')
ylabel('\bfTransfer Time [days]')
plot(nonzeros(Data.ScAng(25:32)),nonzeros(Data.Transfer_Time(25:32)),'bo-')
plot(nonzeros(Data.ScAng(57:64)),nonzeros(Data.Transfer_Time(57:64)),'go-')
plot(nonzeros(Data.ScAng(89:96)),nonzeros(Data.Transfer_Time(89:96)),'ro-')
plot(nonzeros(Data.ScAng(121:128)),nonzeros(Data.Transfer_Time(121:128)),'co-')
legend('h0=100km','h0=300km','h0=500km','h0=700km')
figure()
grid on, hold on
title('\bf Initial Semi-major Axis a0=25500km')
xlabel('\bfInitial True Anomaly [deg]')
ylabel('\bfFuel Mass consumed [kg]')
plot(nonzeros(Data.ScAng(25:32)),nonzeros(Data.mfuel(25:32)),'bo-')
plot(nonzeros(Data.ScAng(57:64)),nonzeros(Data.mfuel(57:64)),'go-')
plot(nonzeros(Data.ScAng(89:96)),nonzeros(Data.mfuel(89:96)),'ro-')
plot(nonzeros(Data.ScAng(121:128)),nonzeros(Data.mfuel(121:128)),'co-')
legend('h0=100km','h0=300km','h0=500km','h0=700km')

toc
% --------------------------------------------------------------------------------
```

- Secondly, the script corresponding also to chapter 5.2, when varying the initial heliocentric velocity and $\theta_{SC\text{-}init}$, is shown:

The first part of the script calculates the limits employed in the for-loop for the initial heliocentric velocity.

```
tic
global R_earth mu_earth h0 J K Data
J = 0;
%limits for the for-loop
R_earth = 6378.1363;    % km
mu_earth = 398600.4415; % km^3/s^2
h0=300; % km
a0min=(R_earth + h0);
min=sqrt(2*mu_earth/(R_earth + h0) - mu_earth/a0min);
a0max=30000;
max=sqrt(2*mu_earth/(R_earth + h0) - mu_earth/a0max);

for ii = min:(max-min)/11:max
    J=J+1
    K=0;
    for jj = 0:45:315
        K=K+1
```

Then, the rest of the main function EarthMarsLT is identical except two changes, the difference introducing the input data and the definition of the initial heliocentric velocity inside the for-loop to identify the initial Mars angle:

```
Pow = 5886;
Isp = 3000;
eta = 0.5;
M_Init = 1500;
h0 = 300;
ScAng = jj;
hf = 400;
Tend = 4;
EarthAng = 0;
n = 8;
```

```
tollE = -0.1;
dM = 45;
iterations = 2;
MarsAng = 90;

    V_E_init = [-sqrt(mu_sun/a_earth)*sin(EarthAng) ...
        sqrt(mu_sun/a_earth)*cos(EarthAng)];
    V_sc_temp = [-ii*(TU/DU)*sin(ScAng) ii*(TU/DU)*cos(ScAng)];
    V_sc_init = V_E_init + V_sc_temp;
```

And after the rest of EarthMarsLT without the plots section, the script continues:

```
%Gathering data of each transfer of the loop
        Data.Transfer_Time(J,K) = transfer_time*TU/86400;
        Data.mfuel(J,K) = mfuel;
        Data.ii(J,K)=ii;
        Data.ScAng(J,K)=jj;
        Data.R_sc_init_X(J,K) = R_E_init(1) + R_sc_temp(1);
        Data.R_sc_init_Y(J,K) = R_E_init(2) + R_sc_temp(2);
        Data.R_sc_init(J,K)=sqrt((Data.R_sc_init_X(J,K)^2)+...
            (Data.R_sc_init_Y(J,K)^2));
        Data.V_sc_init_X(J,K) = V_E_init(1) + V_sc_temp(1);
        Data.V_sc_init_Y(J,K) = V_E_init(2) + V_sc_temp(2);
        Data.V_sc_init(J,K)=sqrt((Data.V_sc_init_X(J,K)^2)+...
            (Data.V_sc_init_Y(J,K)^2));
        clear t x x0 R_M_init V_M_init i
        clear global I Parameters Opt flagOpt flagOptStop
        clear global Wbar StopW flagtime mfuel transfer_time
    end
end

figure(), hold on, grid on
title('\bfTransfer Time')
xlabel('\bfInitial heliocentric velocity [km/s]'), ylabel('\bfTransfer Time [days]')
plot(Data.V_sc_init(:,1)*(DU/TU),Data.Transfer_Time(:,1),'bo')
plot(Data.V_sc_init(:,2)*(DU/TU),Data.Transfer_Time(:,2),'go')
plot(Data.V_sc_init(:,3)*(DU/TU),Data.Transfer_Time(:,3),'ro')
plot(Data.V_sc_init(:,4)*(DU/TU),Data.Transfer_Time(:,4),'co')
plot(Data.V_sc_init(:,5)*(DU/TU),Data.Transfer_Time(:,5),'m+')
plot(Data.V_sc_init(:,6)*(DU/TU),Data.Transfer_Time(:,6),'r+')
plot(Data.V_sc_init(:,7)*(DU/TU),Data.Transfer_Time(:,7),'g+')
plot(Data.V_sc_init(:,8)*(DU/TU),Data.Transfer_Time(:,8),'k+')
legend('0º','45º','90º','135º','180º','225º','270º','315º')

figure(), hold on, grid on
title('\bfFuel mass consumed')
xlabel('\bfInitial heliocentric velocity [km/s]'), ylabel('\bfFuel Mass consumed
[kg]')
plot(Data.V_sc_init(:,1)*(DU/TU),Data.mfuel(:,1),'bo')
plot(Data.V_sc_init(:,2)*(DU/TU),Data.mfuel(:,2),'go')
plot(Data.V_sc_init(:,3)*(DU/TU),Data.mfuel(:,3),'ro')
plot(Data.V_sc_init(:,4)*(DU/TU),Data.mfuel(:,4),'co')
plot(Data.V_sc_init(:,5)*(DU/TU),Data.mfuel(:,5),'m+')
plot(Data.V_sc_init(:,6)*(DU/TU),Data.mfuel(:,6),'r+')
plot(Data.V_sc_init(:,7)*(DU/TU),Data.mfuel(:,7),'g+')
plot(Data.V_sc_init(:,8)*(DU/TU),Data.mfuel(:,8),'k+')
legend('0º','45º','90º','135º','180º','225º','270º','315º')

J=1;K=1;
for J=1:12
    for K=1:8
    if Data.ScAng(J,K)==0
        Data.ScAng(J,K)=0.00001;
    end
    if Data.Transfer_Time(J,K)==0 && Data.mfuel(J,K)==0
        Data.ScAng(J,K)=0;
        Data.V_sc_init(J,K)=0;
    end
    end
end
```

```
%transfer_time
figure(), hold on, grid on
title('\bfInitial True Anomaly = 0 [deg]')
xlabel('\bfInitial heliocentric velocity [km/s]'), ylabel('\bfTransfer Time [days]')
plot(nonzeros(Data.V_sc_init(:,1))*(DU/TU),nonzeros(Data.Transfer_Time(:,1)),'bo-')
figure(), hold on, grid on
title('\bfInitial True Anomaly = 45 [deg]')
xlabel('\bfInitial heliocentric velocity [km/s]'), ylabel('\bfTransfer Time [days]')
plot(nonzeros(Data.V_sc_init(:,2))*(DU/TU),nonzeros(Data.Transfer_Time(:,2)),'bo-')
figure(), hold on, grid on
title('\bfInitial True Anomaly = 90 [deg]')
xlabel('\bfInitial heliocentric velocity [km/s]'), ylabel('\bfTransfer Time [days]')
plot(nonzeros(Data.V_sc_init(:,3))*(DU/TU),nonzeros(Data.Transfer_Time(:,3)),'bo-')
figure(), hold on, grid on
title('\bfInitial True Anomaly = 135 [deg]')
xlabel('\bfInitial heliocentric velocity [km/s]'), ylabel('\bfTransfer Time [days]')
plot(nonzeros(Data.V_sc_init(:,4))*(DU/TU),nonzeros(Data.Transfer_Time(:,4)),'bo-')
figure(), hold on, grid on
title('\bfInitial True Anomaly = 180 [deg]')
xlabel('\bfInitial heliocentric velocity [km/s]'), ylabel('\bfTransfer Time [days]')
plot(nonzeros(Data.V_sc_init(:,5))*(DU/TU),nonzeros(Data.Transfer_Time(:,5)),'bo-')
figure(), hold on, grid on
title('\bfInitial True Anomaly = 225 [deg]')
xlabel('\bfInitial heliocentric velocity [km/s]'), ylabel('\bfTransfer Time [days]')
plot(nonzeros(Data.V_sc_init(:,6))*(DU/TU),nonzeros(Data.Transfer_Time(:,6)),'bo-')
figure(), hold on, grid on
title('\bfInitial True Anomaly = 270 [deg]')
xlabel('\bfInitial heliocentric velocity [km/s]'), ylabel('\bfTransfer Time [days]')
plot(nonzeros(Data.V_sc_init(:,7))*(DU/TU),nonzeros(Data.Transfer_Time(:,7)),'bo-')
figure(), hold on, grid on
title('\bfInitial True Anomaly = 315 [deg]')
xlabel('\bfInitial heliocentric velocity [km/s]'), ylabel('\bfTransfer Time [days]')
plot(nonzeros(Data.V_sc_init(:,8))*(DU/TU),nonzeros(Data.Transfer_Time(:,8)),'bo-')

%fuel mass consumed
figure(), hold on, grid on
title('\bfInitial True Anomaly = 0 [deg]')
xlabel('\bfInitial heliocentric velocity [km/s]'), ylabel('\bfFuel Mass consumed
[kg]')
plot(nonzeros(Data.V_sc_init(:,1))*(DU/TU),nonzeros(Data.mfuel(:,1)),'bo-')
figure(), hold on, grid on
title('\bfInitial True Anomaly = 45 [deg]')
xlabel('\bfInitial heliocentric velocity [km/s]'), ylabel('\bfFuel Mass consumed
[kg]')
plot(nonzeros(Data.V_sc_init(:,2))*(DU/TU),nonzeros(Data.mfuel(:,2)),'bo-')
figure(), hold on, grid on
title('\bfInitial True Anomaly = 90 [deg]')
xlabel('\bfInitial heliocentric velocity [km/s]'), ylabel('\bfFuel Mass consumed
[kg]')
plot(nonzeros(Data.V_sc_init(:,3))*(DU/TU),nonzeros(Data.mfuel(:,3)),'bo-')
figure(), hold on, grid on
title('\bfInitial True Anomaly = 135 [deg]')
xlabel('\bfInitial heliocentric velocity [km/s]'), ylabel('\bfFuel Mass consumed
[kg]')
plot(nonzeros(Data.V_sc_init(:,4))*(DU/TU),nonzeros(Data.mfuel(:,4)),'bo-')
figure(), hold on, grid on
title('\bfInitial True Anomaly = 180 [deg]')
xlabel('\bfInitial heliocentric velocity [km/s]'), ylabel('\bfFuel Mass consumed
[kg]')
plot(nonzeros(Data.V_sc_init(:,5))*(DU/TU),nonzeros(Data.mfuel(:,5)),'bo-')
figure(), hold on, grid on
title('\bfInitial True Anomaly = 225 [deg]')
xlabel('\bfInitial heliocentric velocity [km/s]'), ylabel('\bfFuel Mass consumed
[kg]')
plot(nonzeros(Data.V_sc_init(:,6))*(DU/TU),nonzeros(Data.mfuel(:,6)),'bo-')
figure(), hold on, grid on
title('\bfInitial True Anomaly = 270 [deg]')
xlabel('\bfInitial heliocentric velocity [km/s]'), ylabel('\bfFuel Mass consumed
[kg]')
plot(nonzeros(Data.V_sc_init(:,7))*(DU/TU),nonzeros(Data.mfuel(:,7)),'bo-')
figure(), hold on, grid on
title('\bfInitial True Anomaly = 315 [deg]')
xlabel('\bfInitial heliocentric velocity [km/s]'), ylabel('\bfFuel Mass consumed
[kg]')
```

```
plot(nonzeros(Data.V_sc_init(:,8))*(DU/TU),nonzeros(Data.mfuel(:,8)),'bo-')

toc
% ------------------------------------------------------------------------------
```

- Next, the following script corresponds to chapter 5.3, when varying the departing orbit of the spacecraft; it is called “changedeporbit”.

```
tic
clear all
clc

global h0data a0data J K

J=1 %LEO
h0data=300;
a0data=6678.1363;
EarthMarsLT;

save Data;
clear all;
global h0data a0data J K

load Data;
J=2 %GTO
h0data=300;
a0data=24582;
EarthMarsLT;

save Data;
clear all;
global h0data a0data J K

load Data;
J=3 %GEO
h0data=35786;
a0data=42164;
EarthMarsLT;

save Data;
toc

% transfer time
figure(), hold on, grid on
title('\bfTransfer Time')
xlabel('\bfInitial true anomaly [deg]'), ylabel('\bfTransfer Time [days]')
plot(Data.ScAng(1,:),Data.Transfer_Time(1,:),'bo')
plot(Data.ScAng(2,:),Data.Transfer_Time(2,:),'go')
plot(Data.ScAng(3,:),Data.Transfer_Time(3,:),'ro')
legend('LEO','GTO','GEO')
figure(), hold on, grid on
title('\bfTransfer Time')
xlabel('\bfInitial heliocentric velocity [km/s]'), ylabel('\bfTransfer Time
[days]')
plot(Data.V_sc_init(1,:)*(DU/TU),Data.Transfer_Time(1,:),'bo')
plot(Data.V_sc_init(2,:)*(DU/TU),Data.Transfer_Time(2,:),'go')
plot(Data.V_sc_init(3,:)*(DU/TU),Data.Transfer_Time(3,:),'ro')
legend('LEO','GTO','GEO')

%fuel mass consumed
figure(), hold on, grid on
title('\bfFuel mass consumed')
xlabel('\bfInitial true anomaly [deg]'), ylabel('\bfFuel mass consumed [kg]')
plot(Data.ScAng(1,:),Data.mfuel(1,:),'bo')
plot(Data.ScAng(2,:),Data.mfuel(2,:),'go')
plot(Data.ScAng(3,:),Data.mfuel(3,:),'ro')
legend('LEO','GTO','GEO')
figure(), hold on, grid on
title('\bfFuel mass consumed')
xlabel('\bfInitial heliocentric velocity [km/s]'), ylabel('\bfFuel mass consumed
[kg]')
plot(Data.V_sc_init(1,:)*(DU/TU),Data.mfuel(1,:),'bo')
```

```
plot(Data.V_sc_init(2,:)*(DU/TU),Data.mfuel(2,:),'go')
plot(Data.V_sc_init(3,:)*(DU/TU),Data.mfuel(3,:),'ro')
legend('LEO','GTO','GEO')
% --------------------------------------------------------------------------
```

The function EarthMarsLT called in this script is the main function with the addition of the $\theta_{SC\text{-}init}$ for-loop at the beginning of the script and the input data not being defined manually:

```
K=0;
for ii = 0:45:315
        K=K+1
%function EarthMarsLT (without manual input data)
close all
global Th Isp g0
global mu_earth mu_sun mu_mars
global SOI_E SOI_M a_mars R_mars a_earth R_earth
global DU TU MU
global h0 hf
global I Parameters
global Wbar tspan StopW
global n tollE dM
global Opt flagOpt flagOptStop flagtime transfer_time M_end

I = 0;
flagOpt = 1;
flagtime = 1;

Pow = 5886;
Isp = 3000;
eta = 0.5;
M_Init = 1500;
h0 = h0data;
a0 = a0data;
ScAng = ii;
hf = 400;
Tend = 6;
EarthAng = 0;
n = 8;
tollE = -0.1;
dM = 45;
iterations = 2;
MarsAng = 90;
```

And after the rest of EarthMarsLT without the plots section, the script continues:

```
%Gathering data of each transfer of the loop
        Data.Transfer_Time(J,K) = transfer_time*TU/86400;
        Data.mfuel(J,K) = mfuel;
        Data.h0(J,K)=h0*DU;
        Data.a0(J,K)=a0*DU;
        Data.ScAng(J,K)=ii;
        Data.V_sc_init_X(J,K) = V_E_init(1) + V_sc_temp(1);
        Data.V_sc_init_Y(J,K) = V_E_init(2) + V_sc_temp(2);
        Data.V_sc_init(J,K)=sqrt((Data.V_sc_init_X(J,K)^2)+...
            (Data.V_sc_init_Y(J,K)^2));

        clear t x x0 R_M_init V_M_init i
        clear global I Parameters Opt flagOpt flagOptStop
        clear global Wbar StopW flagtime mfuel transfer_time
end
% --------------------------------------------------------------------------
```

Data from the transfer is gathered in the parameter structure *Data* that is save and load during the script "changedeporbit" to save the information of the different departing orbits.

- Finally, the last script is referred to chapter 5.5, when varying the thruster characteristics. First, the previous script (changing the departing orbit) is used several times, each time changing the thruster characteristics of table 13. Then, the results are gathered each time in different structure parameters corresponding to the different thrusters studied (*DataNSTAR, DataT6, DataRIT22, DataNEXT, DataHiPEP, DataPPS, DataSPT100, DataSPT*). After gathering all the data, this script called "thrusters" applies.

```
load DataNSTAR
load DataT6
load DataRIT22
load DataNEXT
load DataHiPEP
load DataPPS
load DataSPT100
load DataSPT

% transfer time
figure(), hold on, grid on
title('\bfTransfer Time (Ion Engines)')
xlabel('\bfInitial true anomaly [deg]'), ylabel('\bfTransfer Time [days]')
plot(DataNSTAR.ScAng(1,:),DataNSTAR.Transfer_Time(1,:),'b.')
plot(DataNSTAR.ScAng(2,:),DataNSTAR.Transfer_Time(2,:),'g.')
plot(DataNSTAR.ScAng(3,:),DataNSTAR.Transfer_Time(3,:),'r.')
plot(DataT6.ScAng(1,:),DataT6.Transfer_Time(1,:),'bo')
plot(DataT6.ScAng(2,:),DataT6.Transfer_Time(2,:),'go')
plot(DataT6.ScAng(3,:),DataT6.Transfer_Time(3,:),'ro')
plot(DataRIT22.ScAng(1,:),DataRIT22.Transfer_Time(1,:),'bx')
plot(DataRIT22.ScAng(2,:),DataRIT22.Transfer_Time(2,:),'gx')
plot(DataRIT22.ScAng(3,:),DataRIT22.Transfer_Time(3,:),'rx')
plot(DataNEXT.ScAng(1,:),DataNEXT.Transfer_Time(1,:),'b*')
plot(DataNEXT.ScAng(2,:),DataNEXT.Transfer_Time(2,:),'g*')
plot(DataNEXT.ScAng(3,:),DataNEXT.Transfer_Time(3,:),'r*')
plot(DataHiPEP.ScAng(1,:),DataHiPEP.Transfer_Time(1,:),'b+')
plot(DataHiPEP.ScAng(2,:),DataHiPEP.Transfer_Time(2,:),'g+')
plot(DataHiPEP.ScAng(3,:),DataHiPEP.Transfer_Time(3,:),'r+')
legend('NSTAR(LEO)','NSTAR(GTO)','NSTAR(GEO)','T6(LEO)','T6(GTO)','T6(GEO)',...
    'RIT-22(LEO)','RIT-22(GTO)','RIT-22(GEO)','NEXT(LEO)','NEXT(GTO)',...
    'NEXT(GEO)','HiPEP(LEO)','HiPEP(GTO)','HiPEP(GEO)',...
    'Location','NorthEastOutside')
figure(), hold on, grid on
title('\bfTransfer Time (Hall Thrusters)')
xlabel('\bfInitial true anomaly [deg]'), ylabel('\bfTransfer Time [days]')
plot(DataPPS.ScAng(1,:),DataPPS.Transfer_Time(1,:),'bs')
plot(DataPPS.ScAng(2,:),DataPPS.Transfer_Time(2,:),'gs')
plot(DataPPS.ScAng(3,:),DataPPS.Transfer_Time(3,:),'rs')
plot(DataSPT100.ScAng(1,:),DataSPT100.Transfer_Time(1,:),'b<')
plot(DataSPT100.ScAng(2,:),DataSPT100.Transfer_Time(2,:),'g<')
plot(DataSPT100.ScAng(3,:),DataSPT100.Transfer_Time(3,:),'r<')
plot(DataSPT.ScAng(1,:),DataSPT.Transfer_Time(1,:),'b>')
plot(DataSPT.ScAng(2,:),DataSPT.Transfer_Time(2,:),'g>')
plot(DataSPT.ScAng(3,:),DataSPT.Transfer_Time(3,:),'r>')
legend('PPS-1350(LEO)','PPS-1350(GTO)','PPS-1350(GEO)','SPT-100(LEO)',...
    'SPT-100(GTO)','SPT-100(GEO)','SPT-140(LEO)','SPT-140(GTO)',...
    'SPT-140(GEO)','Location','NorthEastOutside')

figure(), hold on, grid on
title('\bfTransfer Time (T6, NEXT and HiPEP thrusters)')
xlabel('\bfInitial heliocentric velocity [km/s]'), ylabel('\bfTransfer Time
[days]')
plot(DataT6.V_sc_init(1,:)*(DU/TU),DataT6.Transfer_Time(1,:),'bo')
plot(DataT6.V_sc_init(2,:)*(DU/TU),DataT6.Transfer_Time(2,:),'go')
plot(DataT6.V_sc_init(3,:)*(DU/TU),DataT6.Transfer_Time(3,:),'ro')
plot(DataNEXT.V_sc_init(1,:)*(DU/TU),DataNEXT.Transfer_Time(1,:),'b*')
plot(DataNEXT.V_sc_init(2,:)*(DU/TU),DataNEXT.Transfer_Time(2,:),'g*')
plot(DataNEXT.V_sc_init(3,:)*(DU/TU),DataNEXT.Transfer_Time(3,:),'r*')
```

```
plot(DataHiPEP.V_sc_init(1,:)*(DU/TU),DataHiPEP.Transfer_Time(1,:),'b+')
plot(DataHiPEP.V_sc_init(2,:)*(DU/TU),DataHiPEP.Transfer_Time(2,:),'g+')
plot(DataHiPEP.V_sc_init(3,:)*(DU/TU),DataHiPEP.Transfer_Time(3,:),'r+')
legend('T6(LEO)','T6(GTO)','T6(GEO)','NEXT(LEO)','NEXT(GTO)','NEXT(GEO)',...
    'HiPEP(LEO)','HiPEP(GTO)','HiPEP(GEO)','Location','NorthEastOutside')

% fuel mass consumed
figure(), hold on, grid on
title('\bfFuel mass consumed (Ion Engines)')
xlabel('\bfInitial true anomaly [deg]'), ylabel('\bfFuel mass consumed [kg]')
plot(DataNSTAR.ScAng(1,:),DataNSTAR.mfuel(1,:),'b.')
plot(DataNSTAR.ScAng(2,:),DataNSTAR.mfuel(2,:),'g.')
plot(DataNSTAR.ScAng(3,:),DataNSTAR.mfuel(3,:),'r.')
plot(DataT6.ScAng(1,:),DataT6.mfuel(1,:),'bo')
plot(DataT6.ScAng(2,:),DataT6.mfuel(2,:),'go')
plot(DataT6.ScAng(3,:),DataT6.mfuel(3,:),'ro')
plot(DataRIT22.ScAng(1,:),DataRIT22.mfuel(1,:),'bx')
plot(DataRIT22.ScAng(2,:),DataRIT22.mfuel(2,:),'gx')
plot(DataRIT22.ScAng(3,:),DataRIT22.mfuel(3,:),'rx')
plot(DataNEXT.ScAng(1,:),DataNEXT.mfuel(1,:),'b*')
plot(DataNEXT.ScAng(2,:),DataNEXT.mfuel(2,:),'g*')
plot(DataNEXT.ScAng(3,:),DataNEXT.mfuel(3,:),'r*')
plot(DataHiPEP.ScAng(1,:),DataHiPEP.mfuel(1,:),'b+')
plot(DataHiPEP.ScAng(2,:),DataHiPEP.mfuel(2,:),'g+')
plot(DataHiPEP.ScAng(3,:),DataHiPEP.mfuel(3,:),'r+')
legend('NSTAR(LEO)','NSTAR(GTO)','NSTAR(GEO)','T6(LEO)','T6(GTO)','T6(GEO)',...
    'RIT-22(LEO)','RIT-22(GTO)','RIT-22(GEO)','NEXT(LEO)','NEXT(GTO)',...
    'NEXT(GEO)','HiPEP(LEO)','HiPEP(GTO)','HiPEP(GEO)',...
    'Location','NorthEastOutside')
figure(), hold on, grid on
title('\bfFuel mass consumed (Hall Thrusters)')
xlabel('\bfInitial true anomaly [deg]'), ylabel('\bfFuel mass consumed [kg]')
plot(DataPPS.ScAng(1,:),DataPPS.mfuel(1,:),'bs')
plot(DataPPS.ScAng(2,:),DataPPS.mfuel(2,:),'gs')
plot(DataPPS.ScAng(3,:),DataPPS.mfuel(3,:),'rs')
plot(DataSPT100.ScAng(1,:),DataSPT100.mfuel(1,:),'b<')
plot(DataSPT100.ScAng(2,:),DataSPT100.mfuel(2,:),'g<')
plot(DataSPT100.ScAng(3,:),DataSPT100.mfuel(3,:),'r<')
plot(DataSPT.ScAng(1,:),DataSPT.mfuel(1,:),'b>')
plot(DataSPT.ScAng(2,:),DataSPT.mfuel(2,:),'g>')
plot(DataSPT.ScAng(3,:),DataSPT.mfuel(3,:),'r>')
legend('PPS-1350(LEO)','PPS-1350(GTO)','PPS-1350(GEO)','SPT-100(LEO)',...
    'SPT-100(GTO)','SPT-100(GEO)','SPT-140(LEO)','SPT-140(GTO)',...
    'SPT-140(GEO)','Location','NorthEastOutside')

figure(), hold on, grid on
title('\bfFuel mass consumed(T6, NEXT and HiPEP thrusters)')
xlabel('\bfInitial heliocentric velocity [km/s]'), ylabel('\bfFuel mass consumed
[kg]')
plot(DataT6.V_sc_init(1,:)*(DU/TU),DataT6.mfuel(1,:),'bo')
plot(DataT6.V_sc_init(2,:)*(DU/TU),DataT6.mfuel(2,:),'go')
plot(DataT6.V_sc_init(3,:)*(DU/TU),DataT6.mfuel(3,:),'ro')
plot(DataNEXT.V_sc_init(1,:)*(DU/TU),DataNEXT.mfuel(1,:),'b*')
plot(DataNEXT.V_sc_init(2,:)*(DU/TU),DataNEXT.mfuel(2,:),'g*')
plot(DataNEXT.V_sc_init(3,:)*(DU/TU),DataNEXT.mfuel(3,:),'r*')
plot(DataHiPEP.V_sc_init(1,:)*(DU/TU),DataHiPEP.mfuel(1,:),'b+')
plot(DataHiPEP.V_sc_init(2,:)*(DU/TU),DataHiPEP.mfuel(2,:),'g+')
plot(DataHiPEP.V_sc_init(3,:)*(DU/TU),DataHiPEP.mfuel(3,:),'r+')
legend('T6(LEO)','T6(GTO)','T6(GEO)','NEXT(LEO)','NEXT(GTO)','NEXT(GEO)',...
    'HiPEP(LEO)','HiPEP(GTO)','HiPEP(GEO)','Location','NorthEastOutside')
% ------------------------------------------------------------------------
```

[END OF THE REPORT]