POLITECNICO DI MILANO

Scuola di Ingegneria Industriale e dell'Informazione

Corso di Laurea Magistrale in
Ingegneria Spaziale

Space Trajectory Optimization with Solar Electric Propulsion Models

Relatore: Prof. Franco BERNELLI ZAZZERA

Correlatori: Ing. Pierluigi DI LIZIA

Ing. Francesco TOPPUTO

Tesi di Laurea di:

Andrea DE BERNARDI Matr. 765712

Anno Accademico 2013 – 2014

Ringraziamenti

Desidero innanzitutto ringraziare il professor Franco Bernelli per avermi dato la possibilità di lavorare a questo progetto innovativo. Inoltre ringrazio sentitamente l'ingegner Pierluigi Di Lizia e l'ingegner Francesco Topputo per le numerose ore dedicate alla mia tesi e per essere sempre stati disponibili a chiarire ogni mio dubbio durante la stesura di questo lavoro. Intendo poi ringraziare Zhang Chen per avermi fornito testi e dati indispensabili per la realizzazione della tesi. Inoltre, vorrei esprimere la mia sincera gratitudine a tutti i miei amici che in questi anni hanno condiviso con me gioie e dispiaceri dell'università. Infine, ho desiderio di ringraziare con affetto i miei genitori per il sostegno ed il grande aiuto che mi hanno dato, ed in particolare ringrazio Claudia per essermi stata vicina in ogni momento di questo lungo viaggio.

Table of Contents

List of Figures

List of Tables

Abstract

One of the most challenging and interesting problems in space mission design is the low-thrust space trajectory optimization. Nevertheless, it can often result very difficult as it involves the solution of a two-point boundary value problem at least, characterized by a set of non-linear coupled differential equations. This problem can be faced according to different techniques. The most simple approach relies on the procedure of *direct transcriptions* and the use of the resulting *direct methods*.

Until nowadays, the main drawback of the trajectory optimization has been the lack of coupling of trajectory, propulsion, and power system. The focus of this thesis is to bridge this gap, providing a robust algorithm capable of optimizing space trajectories in time-optimal problems and in fuel-optimal problems, taking into account mathematical models of solar electric propulsion systems in the trajectory design process.

Keywords: trajectory optimization, low-thrust, solar electric propulsion models, nonlinear programming, direct methods.

Sommario

Uno dei problemi più impegnativi ed interessanti nella progettazione di missioni spaziali è l'ottimizzazione di traiettorie a bassa spinta. Tuttavia, spesso può risultare molto difficile in quanto comporta almeno la soluzione di un problema al contorno, caratterizzato da un insieme di equazioni differenziali non lineari accoppiate. Questo problema può essere affrontato secondo tecniche diverse, ma le più semplici adottano le procedure di *trascrizione diretta* ed utilizzano i risultanti *metodi diretti*.

Fino al giorno d'oggi il principale svantaggio dell'ottimizzazione di traiettorie è stata la mancanza di considerare contemporaneamente la traiettoria, il sistema propulsivo ed il sistema di generazione di potenza nella fase di progettazione. L'obiettivo di questa tesi è di colmare questa lacuna, fornendo un algoritmo robusto in grado di ottimizzare le traiettorie spaziali in problemi tempo-ottimali e di minimizzazione di propellente e di integrare nello stesso tempo le informazioni derivanti dai modelli di propulsione elettrica solare.

Parole chiave: ottimizzazione di traiettorie, bassa spinta, modelli di propulsione elettrica solare, programmazione non lineare, metodi diretti.

Chapter 1

Introduction

1.1 State of the Art

A lot of researchers tried to perform space trajectory optimization with low-thrust propulsion, in particular in the last years when the technological advances in electric propulsion enabled the design of interplanetary missions with significant savings of propellant mass.

The first types of approaches used to deal with space trajectory design problem are the *indirect methods*, which are based on the calculus of variations. According to these methods, a cost function is differentiated through the calculus of variations, ending up with a set of Euler-Lagrange equations which form a boundary value problem. The Euler-Lagrange equations consist in a set of coupled ordinary differential equations where only states and costates appear and only initial conditions are required to solve the problem. These approaches have the advantage that very accurate results can be obtained; however, a rather good initial approximation of the optimal trajectory is needed and a rather large amount of work has to be done by the user to derive the adjoint differential equations.

In the recent years, *Politecnico di Milano* has focused the attention on space trajectory indirect optimization. Rasotto [1] formulated a code capable of facing fuel-optimal transfers in two-body and three-body dynamics, including the possibility of performing intermediate fly-by, rendez-vous and gravity assist. Catucci [2] developed an algorithm capable of optimizing space trajectories in low-thrust propulsion, in particular the optimization is related to the propellant consumption in a time-fixed space transfer with many revolutions around the primary attractor.

To avoid the issues of the direct methods, the approaches used in this thesis are the *direct methods*, where the optimal control problem is transformed into a nonlinear programming problem, as accurately presented in Chapter 2.

The formulation proposed by Betts [3] represents the master key for space trajectory optimization using direct methods. Starting from this theoretical

basis, Patel [4] provided methods which can be used to auto-generate feasible electric propulsion interplanetary trajectories, using Chebyshev polynomials to model the trajectory. Zhang and Topputo [5][6] gave a complete overview of direct transcription and collocation techniques, providing codes for a practical implementation into orbit transfer problems.

1.2 Motivation and Goal

The traditional shortcoming of the trajectory optimization is the absence of coupling of trajectory, propulsion, and power system models in design process. Indeed, the classical way to include all the three elements in the design is based on a step-by-step approach: the optimal trajectory is obtained first; then the best electric propulsion engine that permits it is searched for together with the power system to support the associated power request. However, this strategy is not integrated inside the optimization algorithm and, even though it can be iterated several times, it could not provide the choice of the best thruster and the correct respect of all the mission requirements.

Recently, the studies of Englander et al. [7][8] represent an important breakthrough in the solution of this issue. Indeed, they developed a fully automated tool (EMTG) that provides single-click automated optimization of a full-fidelity low-thrust trajectory, modeling several types of electric thrusters.

This thesis tries to bridge this gap, implementing in the codes given by Zhang and Topputo the engine models provided by Englander et al., thus obtaining at the end an algorithm capable of optimizing space trajectories in time-optimal problems and in fuel-optimal problems, taking into account models of solar electric propulsion in a completely automatic way. In this manner it is possible to evaluate vary fast if a particular type of engine is capable to perform a required mission by verifying that all mission requirements are met, including the actual performances of the adopted propulsion and power systems.

1.3 Structure of the Dissertation

Chapter 2 describes the trajectory optimization via direct transcription and collocation. In particular, it starts from an overview of the optimal control problem, then continues with the formulation of the space trajectory design

problem and the nonlinear programming problem. It ends with the definition of direct transcription procedure and the direct method approach.

Chapter 3 describes the integration between the SEP models and the trajectory optimization algorithm treated in the previous chapter. It also introduces the planar two-body dynamics used in the test cases and details the procedure to overcome the issues of setting the constraints of the problem.

Chapter 4 deals with the results obtained on practical test cases. More specifically, time-optimal problems and fuel-optimal problems are considered, with different type of thruster models integrated in each problem.

Chapter 5 ends the dissertation, discussing the validation of the method and suggesting future developments.

Chapter 2

Low-Thrust Trajectory Optimization via Direct Transcription and Collocation

2.1 The Optimal Control Problem

Optimal control theory is a mathematical optimization method for finding a control law to minimize certain objective function while simultaneously subject to a set of constraints.

Given a set of n first-order differential equations describing a generic dynamics

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t), \tag{2.1}$$

the m control functions $\boldsymbol{u}(t), t \in [t_i t_f]$ must be determined such that the following performance index

$$J = \varphi(\boldsymbol{x}(t_f), t_f) + \int_{t_i}^{t_f} L(\boldsymbol{x}, \boldsymbol{u}, t)\, dt, \tag{2.2}$$

is minimized and q final boundary conditions

$$\boldsymbol{\psi}(\boldsymbol{x}(t_f), \boldsymbol{u}(t_f), t_f) = 0, \tag{2.3}$$

are satisfied.

The solution to this problem is derived by the calculus of variations and its complete investigation is beyond the purposes of this thesis. Thus, only the derivation of the Euler-Lagrange equations will be briefly recalled. Introducing two kinds of Lagrange multipliers - the q-dimensional constant vector $\boldsymbol{\nu}$ for the final boundary constraints and the n-dimensional variable vector $\boldsymbol{\lambda}$ of adjoint or costate variables for the dynamics - the augmented performance index is defined as

$$\bar{J} = \varphi(\boldsymbol{x}(t_f), t_f) + \boldsymbol{\nu}^T \boldsymbol{\psi}(\boldsymbol{x}(t_f), \boldsymbol{u}(t_f), t_f) + \\ + \int_{t_i}^{t_f} [L(\boldsymbol{x}, \boldsymbol{u}, t) + \boldsymbol{\lambda}^T (\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t) - \dot{\boldsymbol{x}})] dt. \tag{2.4}$$

It is important to observe that the dynamics (2.1) is included in the performance index (2.2) in the same fashion of a *constraint*. In other words, the optimal solution must both minimize the objective function and satisfy the dynamics. This is an alternative way of thinking at the dynamics that is in contrast to the dynamical system theory.

The problem consists now in formulating the necessary condition for a stationary point of $\bar{J}$; this is achieved by imposing that its first variation is zero, namely $\delta \bar{J} = 0$. In order to write the necessary condition in a compact form, it is convenient to define the Hamiltonian of the problem as

$$H(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{u}, t) = L(\boldsymbol{x}, \boldsymbol{u}, t) + \boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t). \tag{2.5}$$

The necessary conditions for optimality, also referred as Euler-Lagrange equations, are

$$\dot{\boldsymbol{x}} = H_{\lambda}, \quad \dot{\boldsymbol{\lambda}} = -H_x, \quad H_u = 0, \tag{2.6}$$

where the first is equivalent to the dynamics (2.1), the second describes the dynamics of the costates, and the third is an algebraic equation for the control functions. The necessary equations are obtained by integrating by parts the last term in (2.4), and imposing the necessary conditions to the function $\delta \bar{J}$. Thus, a solution satisfying the equations (2.6) is not necessarily a minimum of $\bar{J}$; this is the reason why the conditions are only necessary. The differential-algebraic system (2.6) must be solved together with the final boundary conditions (2.3) and the following transversality conditions

$$\boldsymbol{\lambda}(t_i) = 0, \quad \boldsymbol{\lambda}(t_f) = [\varphi_x + \boldsymbol{\nu}^T \boldsymbol{\psi}_{\boldsymbol{x}}]_{t=t_f}. \tag{2.7}$$

The problem so defined represents a two-point boundary value problem.

The last of equations (2.6) is an application of the Pontryagin maximum principle. A more general expression is in fact

$$\boldsymbol{u} = \arg\min_{\boldsymbol{u} \in U} H(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{u}, t)$$

where U defines the domain of feasible controls. The maximum principle states that control variables must be chosen to optimize the Hamiltonian at every instant of time: the solution of the optimal control problem is an extremum for H. In essence the maximum principle is a constrained optimization problem in

the variables $\boldsymbol{u}(t)$ at all values of t.

A classical optimal trajectory design problem is stated in a slightly different fashion than the classical optimal control problem illustrated above. The formulation proposed by Betts [3] fits better the numerical methods used to solve the trajectory design problem.

Typically the dynamical system incorporates a number of constant parameters **p** through:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{p}, t), \tag{2.8}$$

and initial and final conditions are defined within some prescribed lower and upper bounds as

$$\begin{aligned} \boldsymbol{\psi}_{i,l} \le \boldsymbol{\psi}_i(\boldsymbol{x}(t_i), \boldsymbol{u}(t_i), \boldsymbol{p}, t_i) \le \boldsymbol{\psi}_{i,u}, \\ \boldsymbol{\psi}_{f,l} \le \boldsymbol{\psi}_f(\boldsymbol{x}(t_f), \boldsymbol{u}(t_f), \boldsymbol{p}, t_f) \le \boldsymbol{\psi}_{f,u}. \end{aligned} \tag{2.9}$$

In addition, the solution must satisfy algebraic path constraints of the form

$$\boldsymbol{g}_l \le \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{p}, t) \le \boldsymbol{g}_u, \tag{2.10}$$

as well as simple bounds on the state variables

$$\boldsymbol{x}_l \le \boldsymbol{x}(t) \le \boldsymbol{x}_u, \tag{2.11}$$

and on the control variables

$$\boldsymbol{u}_l \le \boldsymbol{u}(t) \le \boldsymbol{u}_u. \tag{2.12}$$

The basic problem is to determine the control vectors $\boldsymbol{u}(t)$ and the parameters **p** to minimize the performance index

$$J = \phi(\boldsymbol{x}(t_f), t_f), \tag{2.13}$$

here written in the Mayer form.

2.1.1 The Space Trajectory Design Problem

Some spacecraft equipments can create very small acceleration levels (if compared to those produced by conventional chemical rockets) while achieving very high specific impulses and long working hours. Examples of these equipments are the ion engines, hall-effect engines, solar sails, etc. "Spacecraft low-thrust trajectory design" is sought to extremize some performance measures

(e.g., maximize residual fuel or minimize transfer time), while simultaneously satisfying a series of constraints. These constraints can be as follows: states and controls have upper and lower bounds, the transfer orbit should be subject to the dynamic equations, and the spacecraft might be required to be in a destination orbit after a sequence of maneuvers. The study of such low-thrust optimal spacecraft trajectories is a specialization of the optimal control theory.

In space flight mechanics, the equations of motion (2.1) have the form

$$\dot{\boldsymbol{x}} = \begin{Bmatrix} \dot{\boldsymbol{r}} \\ \dot{\boldsymbol{v}} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{v} \\ \boldsymbol{g}(\boldsymbol{r}) + a_c \hat{\boldsymbol{u}} \end{Bmatrix} \tag{2.14}$$

where $\boldsymbol{g}(\boldsymbol{r})$ is the vector field (e.g., two- or n-body dynamics), a_c is the control thrust acceleration magnitude, and $\hat{\boldsymbol{u}}$ is the thrust unit vector. The control acceleration magnitude is upper bounded for technological reasons, meaning that $0 \le a_c \le a_c^{max}$. To minimize the total velocity change, and therefore the propellant mass, the objective function (2.2) is such that $L = a_c, \varphi = 0$. The Hamiltonian (2.5) is therefore

$$H = a_c + \boldsymbol{\lambda}_r \cdot \boldsymbol{v} + \boldsymbol{\lambda}_v \cdot [\boldsymbol{g}(\boldsymbol{r}) + a_c \hat{\boldsymbol{u}}] = a_c [1 + \boldsymbol{\lambda}_v \cdot \hat{\boldsymbol{u}}] + \boldsymbol{\lambda}_r \cdot \boldsymbol{v} + \boldsymbol{\lambda}_v \cdot \boldsymbol{g}(\boldsymbol{r}), \tag{2.15}$$

where $\boldsymbol{\lambda}_r, \boldsymbol{\lambda}_v$ are the costate vectors associated to the dynamics (2.14).

The control variables are a_c and $\hat{\boldsymbol{u}}$. As the Hamiltonian is linear in the control, the last of equations (2.6) cannot be applied. The optimal control law is chosen instead according to the Pontryagin's Minimum (or Maximum) Principle, which states that at any time the control variables are chosen to minimize the Hamiltonian. Because of this property, the following observations can be made:

1. the thrust unit vector $\hat{\boldsymbol{u}}$ is to be parallel and opposite to $\boldsymbol{\lambda}_v$; i.e. $\hat{\boldsymbol{u}} = -\alpha \boldsymbol{\lambda}_v$ with $\alpha > 0$; this explains why $\boldsymbol{\lambda}_v$ is also referred to as the *primer vector*;
2. it is important to choose a value of $a_c, 0 \le a_c \le a_c^{max}$, according to the sign of the *switching function*

$$S = 1 + \boldsymbol{\lambda}_v \cdot \hat{\boldsymbol{u}}. \tag{2.16}$$

In particular, $a_c = a_c^{max}$ when $S < 0$ and $a_c = 0$ when $S > 0$. This makes the function $a_c(t)$ to have a *bang-bang* structure; i.e., it is piece-wise discontinuous and is either zero or maximum. This property is of great importance to evaluate the optimal control profile a posteriori.

The necessary conditions only guarantee that the optimal trajectory is an extremum for the Hamiltonian. Thus, to assess the optimality of the solutions,

one is supposed to check the second-order conditions. However, due to the nature of the space trajectory problem, there is no upper bound to the propellant that can be consumed in one trajectory, so one may be confident that a solution that satisfies the necessary conditions is a local minimum and not a local maximum.

2.1.2 The Nonlinear Programming Problem

Essentially, any numerical method for solving the trajectory optimization problem incorporates some type of iteration with a finite set of unknowns. In Section 2.2 it is shown how an optimal control problem can be transformed into a nonlinear programming problem, NLP for brevity. A NLP problem is a decisional problem concerning a scalar algebraic function and an algebraic vector of constraints. As opposite to the optimal control problem, no dynamics is involved into a NLP problem.

Suppose that the n variables $\boldsymbol{x}$ must be chosen to solve

$$\min_{\boldsymbol{x}} F(\boldsymbol{x}), \tag{2.17}$$

subject to the m equality constraints

$$\boldsymbol{c}(\boldsymbol{x})=0, \tag{2.18}$$

where $m \leq n$. The Lagrangian of this problem is

$$L(\boldsymbol{x},\boldsymbol{\lambda})=F(\boldsymbol{x})-\boldsymbol{\lambda}^T\boldsymbol{c}(\boldsymbol{x}), \tag{2.19}$$

which is a scalar function of the n variables $\boldsymbol{x}$ and the m Lagrange multipliers $\boldsymbol{\lambda}$. The necessary conditions for a point $(\boldsymbol{x}^*,\boldsymbol{\lambda}^*)$ to be a constrained optimum require solving the following system

$$\begin{aligned}\nabla_x L(\boldsymbol{x},\boldsymbol{\lambda})&=\boldsymbol{g}(\boldsymbol{x})-\boldsymbol{G}^T(\boldsymbol{x})\boldsymbol{\lambda}=0,\\ \nabla_\lambda L(\boldsymbol{x},\boldsymbol{\lambda})&=-\boldsymbol{c}(\boldsymbol{x})=0,\end{aligned} \tag{2.20}$$

where $\boldsymbol{g}=\nabla_x F$ and $\boldsymbol{G}$ are the gradient of the objective function $F(\boldsymbol{x})$ and the Jacobian of the equality constraint vector $\boldsymbol{c}(\boldsymbol{x})$, respectively. The system (2.20) can be solved via a Newton's method to find the (n + m) variables $(\boldsymbol{x}^*,\boldsymbol{\lambda}^*)$. Given a generic initial guess $(\boldsymbol{x},\boldsymbol{\lambda})$, its corrections $(\Delta\boldsymbol{x},\Delta\boldsymbol{\lambda})$ to construct the new solution $(\boldsymbol{x}+\Delta\boldsymbol{x},\boldsymbol{\lambda}+\Delta\boldsymbol{\lambda})$ are given by solving the linear system

$$\begin{bmatrix} \boldsymbol{H}_L & -\boldsymbol{G}^T \\ \boldsymbol{G} & 0 \end{bmatrix} \begin{Bmatrix} \Delta \boldsymbol{x} \\ \Delta \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} -\boldsymbol{g} \\ -\boldsymbol{c} \end{Bmatrix}, \tag{2.21}$$

also referred as Karush-Kuhn-Tucker system; in (2.21), the term $\boldsymbol{H}_L$ is the Hessian of (2.19) in x, namely

$$\boldsymbol{H}_L = \nabla_x^2 F - \sum_{i=1}^{m} \lambda_i \nabla_x^2 c_i . \tag{2.22}$$

It is important to observe that an equivalent way to defining the search direction $\Delta \boldsymbol{x}$ is to minimize the quadratic form

$$\frac{1}{2} \Delta \boldsymbol{x}^T \boldsymbol{H}_L \Delta \boldsymbol{x} + \boldsymbol{g}^T \Delta \boldsymbol{x} \tag{2.23}$$

subject to the linear to the linear constraints

$$\boldsymbol{G} \Delta \boldsymbol{x} = -\boldsymbol{c} . \tag{2.24}$$

This is the reason why this problem is also referred to as a quadratic programming (QP) problem.

The NLP problem formulated above can be generalized to the case that occur when inequality constraints are imposed; the m constraints are of the form

$$\boldsymbol{c}(\boldsymbol{x}) \geq 0. \tag{2.25}$$

Constraints that are strictly satisfied, i.e. c_i ($\boldsymbol{x}$) > 0, are called inactive; the remaining active set of constraints are on their bounds, i.e. c_i ($\boldsymbol{x}$) = 0. If the active set of constraints is known, the inactive constraints are ignored and the problem is simply solved using the method for an equality constrained problem discussed above.

In summary, the general NLP problem requires finding the n vectors to solve

$$\min_{\boldsymbol{x}} F(\boldsymbol{x}), \tag{2.26}$$

subject to the m constraints

$$\boldsymbol{c}_L \leq \boldsymbol{c}(\boldsymbol{x}) \leq \boldsymbol{c}_U , \tag{2.27}$$

and bounds

$$\boldsymbol{x}_L \leq \boldsymbol{x} \leq \boldsymbol{x}_U . \tag{2.28}$$

In this formulation equality constraints can be imposed by setting $c_{j,L} = c_{j,U}$.

2.1.3 Direct Transcription

The optimal trajectory design is a continuous optimal control problem that can be solved with the Euler–Lagrange equations (2.6). This way of solving the optimal control problem is called *indirect method*. Another philosophy consists in translating the continuous optimal control problem into a NLP problem and solving for a finite set of variables: this procedure is called *direct transcription* and the approach is said *direct method*.

With a direct approach, the solution of the optimal control problem is strictly connected to the numerical integration of the differential equations. The core of this method consists in fact in the way of dealing with the dynamical system: the set of differential equations governing – in the case of this thesis – the motion of a spacecraft, can be transcribed into a *finite set* of equality constraints. If the solution to the NLP problem satisfies these constraints, then the original optimal control problem is solved within the degree of accuracy of the numerical scheme used.

Let's consider the optimal control problem formulated through the equations (2.8)-(2.13). The time domain can be discretized as

$$t_i = t_1 < t_2 < \ldots < t_N = t_f, \tag{2.29}$$

where the time labels are referred to as mesh points or nodes; t_i and t_f are the initial and final time, respectively, and $h = (t_f - t_i) / (N - 1)$ is the step size of the discretization. The states and the controls can be discretized over the mesh (2.29) by defining $\boldsymbol{x}_k = \boldsymbol{x}(t_k)$ and $\boldsymbol{u}_k = \boldsymbol{u}(t_k)$. The discretized states and the controls are now ready to be treated as a set of NLP variables. The whole variable vector of the problem is

$$\boldsymbol{y} = \left\{ \boldsymbol{x}_1, \boldsymbol{u}_1, \ldots, \boldsymbol{x}_N, \boldsymbol{u}_N \right\}^T. \tag{2.30}$$

The differential equations are replaced by a finite set of defects constraints derived by the numerical integration scheme. If a classical Runge-Kutta scheme is used, the defects are of the form

$$\boldsymbol{\zeta}_i \equiv \boldsymbol{x}_{i+1} - \boldsymbol{x}_i - h_i \sum_{j=1}^{k} \beta_j \boldsymbol{f}_{ij}, \tag{2.31}$$

with an appropriate definition of $\boldsymbol{f}_{ij}$; for a multistep method, the defects depend instead by only the mesh points through

$$\boldsymbol{\zeta}_{i+k-1} \equiv \boldsymbol{x}_{i+1} - \sum_{j=0}^{k-1} \alpha_j \boldsymbol{x}_{i+j} - h \sum_{j=0}^{k} \beta_j \boldsymbol{f}_{i+j}, \tag{2.32}$$

where $\boldsymbol{f}_k = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{p}, \boldsymbol{t}_k)$.

In any case, as a result of the transcription, the optimal control constraints (2.9)-(2.10) are replaced by the NLP constraints

$$\boldsymbol{c}_L \le \boldsymbol{c}(\boldsymbol{y}) \le \boldsymbol{c}_U, \tag{2.33}$$

where

$$\boldsymbol{c}(\boldsymbol{y}) \equiv \{\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, \ldots, \boldsymbol{\zeta}_{N-1}, \boldsymbol{\psi}_0, \boldsymbol{\psi}_f, \boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_N\}^T, \tag{2.34}$$

and

$$\boldsymbol{c}_L \equiv \{\boldsymbol{0}, \ldots, \boldsymbol{0}, \boldsymbol{g}_L, \ldots, \boldsymbol{g}_L\}^T, \tag{2.35}$$

with a corresponding definition of $\boldsymbol{c}_U$.

The first $n(N-1)$ equality constraints in (2.34) require that the defect vectors $\boldsymbol{\zeta}_k, k = 1, \ldots, N-1$, are zero, thereby they satisfy the differential equations (2.8) within the accuracy of the numerical integration scheme. The boundary conditions (2.9) are enforced directly by the equality constraints on $\boldsymbol{\psi}_0$ and $\boldsymbol{\psi}_f$, and the nonlinear path constraints (2.10) are imposed at the grid points. In a similar fashion the objective function, either in the form (2.2) or (2.13), can be written in terms of the NLP variable vector $\boldsymbol{y}$, namely $F = F(\boldsymbol{y})$. The optimal control problem has been so translated into the form (2.26)-(2.28) and it can be solved as a standard NLP problem through (2.17)-(2.22). The method so stated is said *direct transcription and collocation*.

2.2 Direct Transcription and Collocation

A variety of direct transcription methods emerged for solving optimal spacecraft trajectory problem, whose main difference is on how the state and control variables are discretized and how the dynamic constraints are satisfied.

In particular, in mathematics, a collocation method is a method for the numerical solution of ordinary differential equations, partial differential equations and integral equations. In optimal control problems, collocation method is used to transcribe differential dynamic constraints into a set of algebraic constraints. The basic idea is to choose a polynomials up to a certain degree with a number

of points in the domain (collocation points), and to evaluate the solution which satisfies the given equation at the collocation points.

2.2.1 Hermite-Simpson Method

A basic form of collocation method (Hermite–Simpson method) is illustrated in Figure 2.1. As for other direct methods, the whole trajectory in $[t_i, t_f]$ is decomposed into a number of subintervals. Within each of these segments $[t_k, t_{k+1}]$, the two end points, denote as 'nodes' (blue dots), set the corresponding state and control as NLP decision variables, i.e. $[x_k, u_k, x_{k+1}, u_{k+1}]$. The dynamic equation are performed to provide time derivative values at the two nodes, so four available information $[x_k, x_{k+1}, f(x_k, u_k), f(x_{k+1}, u_{k+1})]$ can be used to construct a 3rd-order hermite interpolate polynomial. So far, the interpolate polynomial can't meet dynamic constraints at any time within $[t_k, t_{k+1}]$, because it only naturally satisfies the dynamic constraints at the two boundary points. Then, denote the middle of $[t_k, t_{k+1}]$ as t_c and the corresponding state and control variables as $[x_c, u_c]$, this new point is called 'collocation point' (red diamond). If collocation constraint $\Delta = \dot{x}_c - f(x_c, u_c)$ is enforced equal to zero, the interpolate polynomial not only naturally satisfies the dynamics at the two boundary, but also meets the dynamics at the middle point t_c. If a very large scale of subintervals is used, the state motion will approach the real dynamics within the whole time domain $[t_i, t_f]$.

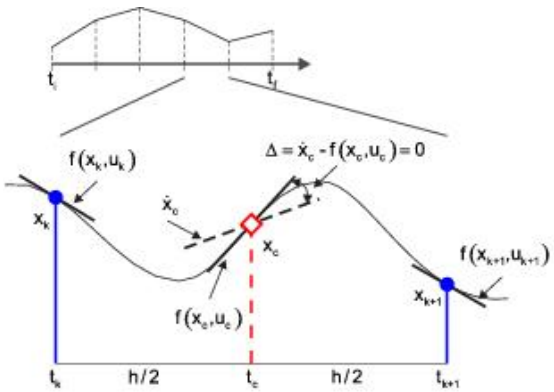

Figure 2.1. Formulate collocation constraint of Hermite-Simpson method.

The detailed procedure can be derived as follows. Let the states $x(t)$ be

represented on each segment through cubic-spline of the form

$$x(t)=a_0+a_1 t+a_2 t^2+a_3 t^3, \tag{2.36}$$

followed by the first order derivative of the polynomial

$$\dot{x}(t)=a_1+2\,a_2 t+3\,a_3 t^2, \tag{2.37}$$

where [a_0, a_1, a_2, a_3] are all coefficients of the polynomial. In order to simplify the argument, the time domain is transformed such that $t\in[0,h]$ (h is the time interval of the segment); let $x(0)=x_k$, $x(h)=x_{k+1}$, $\dot{x}(0)=\dot{x}_k$, $\dot{x}(h)=\dot{x}_{k+1}$, Eq. (2.36) and Eq. (2.37) are evaluated at $t = 0$ and $t = h$:

$$\begin{bmatrix} x(0) \\ \dot{x}(0) \\ x(h) \\ \dot{x}(h) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & h & h^2 & h^3 \\ 0 & 1 & 2h & 3h^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}.$$

The four coefficients of interpolate polynomial can be given by

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{3}{h^2} & -\frac{2}{h} & \frac{3}{h^2} & -\frac{1}{h} \\ \frac{2}{h^3} & \frac{1}{h^2} & -\frac{2}{h^3} & \frac{1}{h^2} \end{bmatrix} \begin{bmatrix} x(0) \\ \dot{x}(0) \\ x(h) \\ \dot{x}(h) \end{bmatrix}.$$

Take [a_0, a_1, a_2, a_3] into Eq. (2.36) and Eq. (2.37), then the interpolated value of $x(t)$ located at the center of the segment t_c is

$$x_c=x(\frac{h}{2})=\frac{1}{2}(x_k+x_{k+1})+\frac{h}{8}\left[f(x_k,u_k)-f(x_{k+1},u_{k+1})\right] \tag{2.38}$$

and the derivation of interpolation polynomial evaluated at the center of the segment is

$$\dot{x}_c=\dot{x}(\frac{h}{2})=-\frac{3}{2h}(x_k-x_{k+1})-\frac{1}{4}\left[f(x_k,u_k)+f(x_{k+1},u_{k+1})\right]. \tag{2.39}$$

The control variable located at the middle of the segment can be implemented by simple linear interpolation which is given by

$$u_c = \frac{u_k + u_{k+1}}{2}. \tag{2.40}$$

Finally, the difference between the interpolated Eq. (2.39) and calculated derivatives at the midpoints defines a constraint

$$\begin{aligned} \Delta &= \dot{x}_c - f(x_c, u_c) = \\ &= -\frac{3}{2h}(x_k - x_{k+1}) - \frac{1}{4}\left[f(x_k, u_k) + f(x_{k+1}, u_{k+1})\right] - f(x_c, u_c) = \\ &= \frac{3}{2h}(x_k - x_{k+1}) + \frac{1}{4}\left[f(x_k, u_k) + 4\mathrm{f}(x_c, u_c) + f(x_{k+1}, u_{k+1})\right] = \\ &= x_k - x_{k+1} + \frac{h}{6}\left[f(x_k, u_k) + 4\mathrm{f}(x_c, u_c) + f(x_{k+1}, u_{k+1})\right] = 0. \end{aligned} \tag{2.41}$$

When the NLP solver goes, selecting [x_k, u_k, x_{k+1}, u_{k+1}] to drive Δ to zero will enforce the interpolate polynomial to accurately approximate the true dynamics. One interesting thing is that the last row of Eq. (2.41) is actually an implicit hermite integration. So, if collocation constraints are satisfied, the system is said to be 'implicitly' integrated.

The structure of Jacobian with free final time t_f is shown in Figure 2.2, where the rows are constraints while the columns are NLP decision variables.

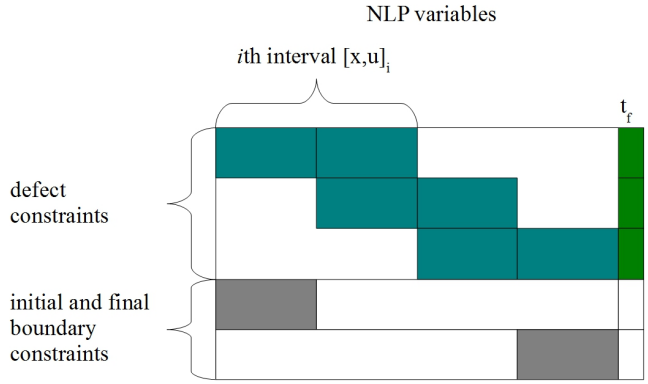

Figure 2.2. Jacobian structure for Hermite-Simpson method.

Chapter 3

Space Trajectory Optimization with Solar Electric Propulsion Models

3.1 Solar Electric Propulsion Models

In this thesis, several types of electric thrusters are modeled according to the effective characterization proposed by Englander et al. [7][8]. This characterization relies on linking the thrust supplied by the thrusters to the input power in terms of polynomials. The resulting performance polynomials assume the form:

$$T = a_T P^4 + b_T P^3 + c_T P^2 + d_T P + e_T \tag{3.1}$$

where T is the thrust measured in mN, P is the available power measured in kW, and the coefficients $\{a_T, b_T, c_T, d_T, e_T\}$ are drawn from public literature [7][8][9] [10] and are curve fits to laboratory test data. Each thruster also has an associated minimum power P_{min} and maximum power P_{max}. If the available power P is less than P_{min}, then T is zero. If P is greater than P_{max}, then the performance polynomials are evaluated at P_{max}.

The available power P is the difference between the power generated by the spacecraft $P_{generated}$ and the power required to operate the spacecraft bus $P_{s/c}$,

$$P = P_{generated} - P_{s/c}. \tag{3.2}$$

The power delivered by a solar array is given by

$$P_{generated} = \frac{P_0}{r^2}\left(\frac{\gamma_0 + \gamma_1 / r + \gamma_2 / r^2}{1 + \gamma_3 r + \gamma_4 r^2}\right) \tag{3.3}$$

where γ_i are user-defined solar panel coefficients, r is the distance between the Sun and the spacecraft in Astronomical Unit (AU) and P_0 is the "base power" delivered by the array at 1 AU.

The power required by the spacecraft bus $P_{s/c}$ is specific to each mission and is set to 300W to the test cases presented in this work.

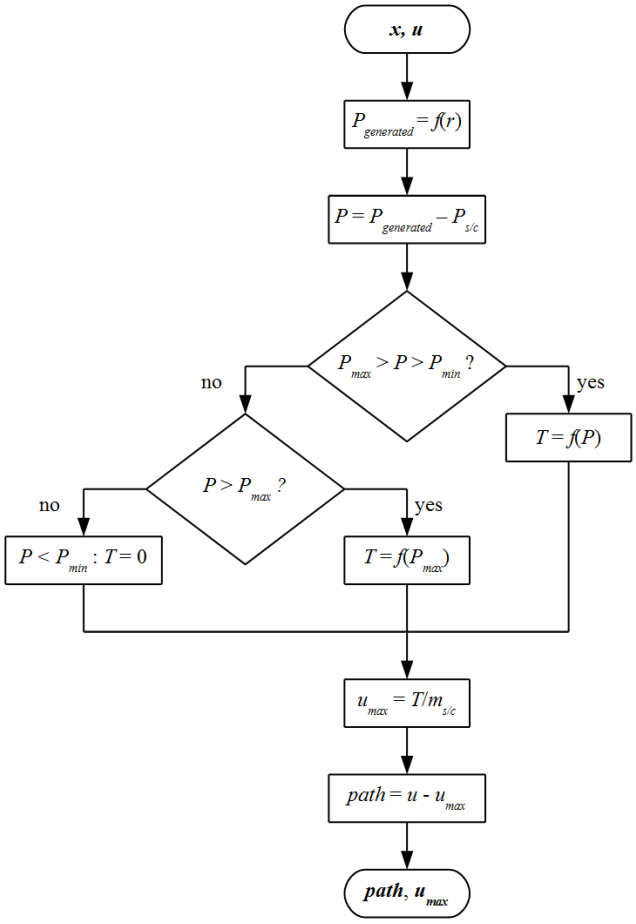

Figure 3.1. Schematic of the engine algorithm.

Now that all the elements of the models have been described, they are collected and resumed in the following algorithm (Figure 3.1):

- at each iteration, the variable vector, with states and controls for each mesh point, enters as input;
- the generated power is evaluated according to the positions contained in the vector, next the available power is found;
- thrust is then evaluated with (3.1) if the available power is inside the range of powers permitted by the thruster, otherwise it is evaluated with P_{max} if $P > P_{max}$ or $T = 0$ if $P < P_{min}$;
- next $u_{\max}$ is calculated dividing the thrust by the mass of the spacecraft $\left(u_{max} = T/m_{s/c}\right)$ and then it is used to find $u - u_{\max}$, where u is taken from

the input vector;

- finally, u_{max} and u - u_{max} are the output of the flow chart and afterward they are used in several parts of the optimization algorithm.

Focusing on the third and fourth passages of the algorithm, it can be seen the dependence of T on the state and, as a result, the dependance of u itself from the state, according to the relation $T = m_{s/c} \cdot u$.

3.2 Integration between Trajectory Optimization and SEP Models

Such type of engine models are integrated inside the optimization tool proposed by Zhang and Topputo [5][6], as illustrated in Figure 3.2 as schematic. As it is possible to note, it is composed by different blocks that together contribute to the optimization with IPOPT (Interior Point OPTimizer). In the following, a brief description of each piece of the algorithm is proposed.

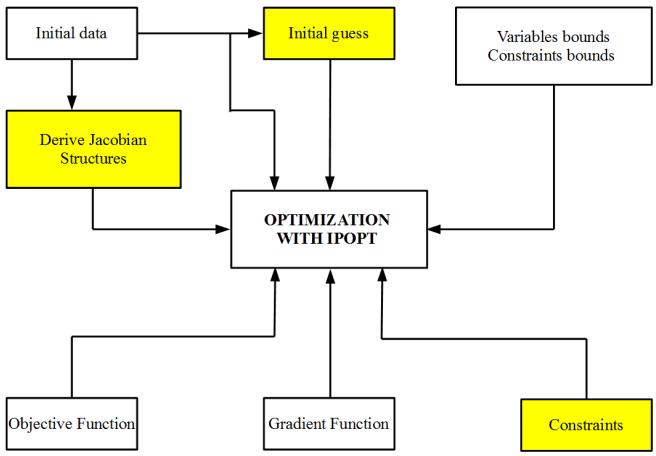

Figure 3.2. Block diagram of the optimization tool.

First of all, there is the statement of the global data of the problem, the initial and final boundary conditions and the data of the thrusters. Then the algorithm starts to evaluate and to store the different structures of the Jacobian matrix: defect constraints, boundary constraints and path constraints are differentiated with respect to the state and control variables, as well as to the time step h.

Secondly, there is the evaluation of the initial guess of the transfer orbit, through the numerical integration of the ordinary differential dynamics equations. The solver integrates the system of differential equations from the initial time to the final one with initial conditions, finding where a function, called event function, is zero. For this event function, it is specified that the integration is to terminate at a zero and that all the zeros are to be computed. In this work a zero is found when the spacecraft intersects the final orbit.

Next there is the declaration of the variables and constraints bounds, whose definition is described in Section 3.3.

Finally the optimizer IPOPT is run, by declaring the following functions:

- objective, in this thesis the minimization of the transfer time or the maximization of the final spacecraft mass;
- gradient of the objective function;
- constraints, where there is the construction of defect, boundary and path constraints;
- Jacobian, necessary if constraints are present, where there various part of the Jacobian matrix are assembled;
- Jacobian structure, necessary if constraints are present, is a sparse matrix with ones wherever the constraint Jacobian potentially has nonzero entries.

Essentially, the thruster algorithm shown in Figure 3.1 is integrated three times inside the optimization algorithm, as highlighted in Figure 3.2: the first time in the Jacobian box, when the derivatives of the path are evaluated; the second time in the evaluation of the initial guess, where the propulsive acceleration is needed by the ODE; finally in the constraints box, to evaluate the path constraints.

3.3 Planar Two-Body Dynamics

A simple planar low-thrust orbit transfer problem is considered to assess the performances of the proposed technique. For this type of problem, Cartesian coordinates are the simplest choice but the most disadvantageous too, because a lot of discrete points will be used to catch the rapidly changing position and velocity variables. On the contrary, slowly changing state variables make NLP solvers to improve both efficiency and robustness. Polar coordinates are then used for the planar two-body dynamics; i.e.

$$\begin{cases} \dot{r} = v_r \\ \dot{\theta} = \dfrac{v_t}{r} \\ \dot{v}_r = \dfrac{v_t^2}{r} - \dfrac{\mu}{r^2} + u \sin\phi \\ \dot{v}_t = -\dfrac{v_r v_t}{r} + u \cos\phi \,. \end{cases} \tag{3.4}$$

In Eq. (3.4), r is the spacecraft radius, θ is phase angle, v_r and v_t are the radial and transversal velocities respectively, μ is the gravitational constant, u is the propulsive acceleration and ϕ is the thrust angle (see Figure 3.4).

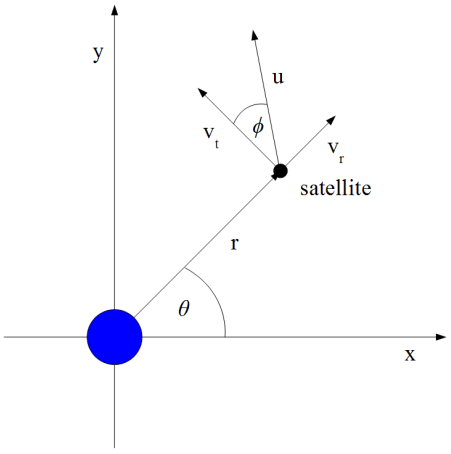

Figure 3.3. 2-body orbit transfer in polar coordinates.

Thereby, the variables discretized over the mesh are the states $\{r, \theta, v_r, v_t\}$ and the controls $\{u, \phi\}$. In a classical space trajectory design problem, each variable has an upper and lower bound which are constant limits along all the time domain. In particular, the thrust acceleration has limits of this type:

$$0 \le u(t) \le u_{max}\,, \tag{3.5}$$

with u_{max} maximum thrust acceleration.

With the introduction of the SEP models inside the dynamics equation, it is no longer possible to use this type of formulation, because u_{max} becomes state-dependent. It is then necessary to impose the satisfaction of a nonlinear path constraint, which is now dependent on u and r:

$$\underbrace{u - u_{max}(r)}_{g(u,r)} \leq 0 . \tag{3.6}$$

In terms of the algorithm, the Jacobian matrix undergoes a dimension change, with the addiction of N rows of algebraic path constraints, one for each mesh point, as shown in Figure 3.4.

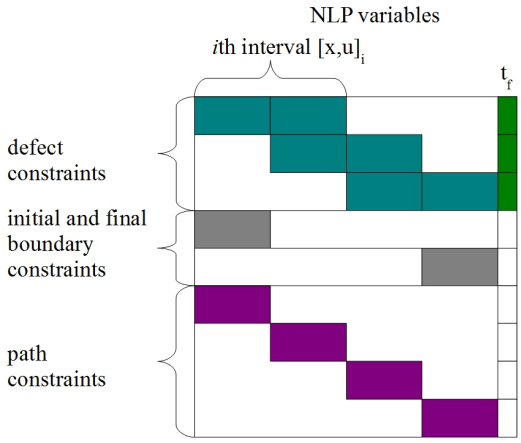

Figure 3.4. Jacobian structure with path constraints.

Chapter 4

Results

4.1 Assumptions

All transfers in this chapter are designed taking into account the assumptions proposed by Englander et al. [7], using reasonable initial mass, propulsion and power, as described in Table 4.1.

Table 4.1. Problem assumptions.

Parameter	**Value**
Initial spacecraft mass	3618 kg
Beginning of life (BOL) solar array output at 1 AU	20 kW
Spacecraft bus power	300 W
Solar panel γ_i (Ultraflex)	[1.1705, 0.0289, -0.2197, -0.0202, -0.0001]

Three electric thrusters are modeled, including NSTAR, XIPS-25 and BPT-4000. The first two are high-power ion thrusters and the last is a Hall thruster. The performance polynomial coefficients for all three thrusters are shown in Table 4.2. The power ranges over which the performance polynomials are valid are shown in Table 4.3. The performance curves are the red lines shown in Figure 4.1, Figure 4.2, Figure 4.3 and Figure 4.4, where the blue lines represent the minimum and maximum power admissible for each thruster.

Table 4.2. Thruster polynomials.

Thruster	a	b	c	d	e
NSTAR	5.145602	-36.720293	90.486509	-51.694393	26.337459
XIPS-25	0.0367	-0.4966	1.4111	35.3591	-0.3984
BPT-4000 (High-I_{sp})	-0.095437	1.637023	-9.517167	72.030104	-7.181341
BPT-4000 (High-Thrust)	0.173870	-1.150940	-2.118891	77.342132	-8.597025

Table 4.3. Thruster power ranges.

Thruster	min power (kW)	max power (kW)
NSTAR	0.525	2.600
XIPS-25	0.436	5.030
BPT-4000	0.302	4.839

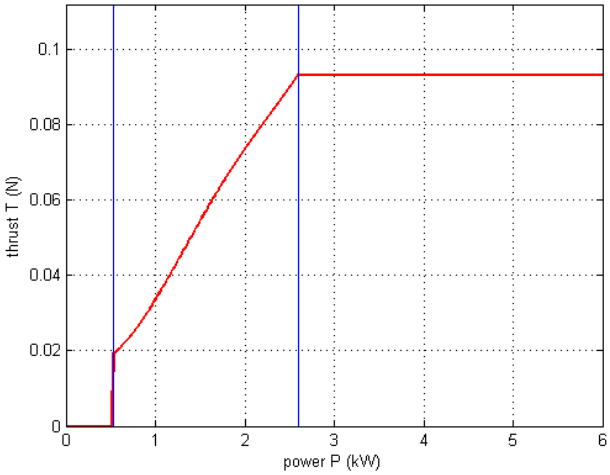

Figure 4.1. Thrust vs. power for the NSTAR.

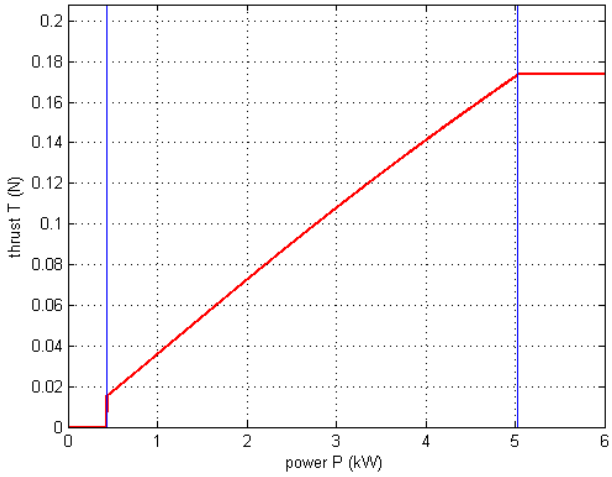

Figure 4.2. Thrust vs. power for the XIPS-25.

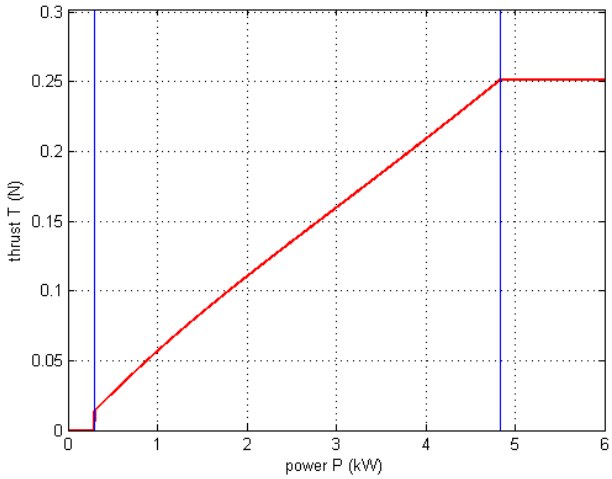

Figure 4.3. Thrust vs. power for BPT-4000 High I_{sp}.

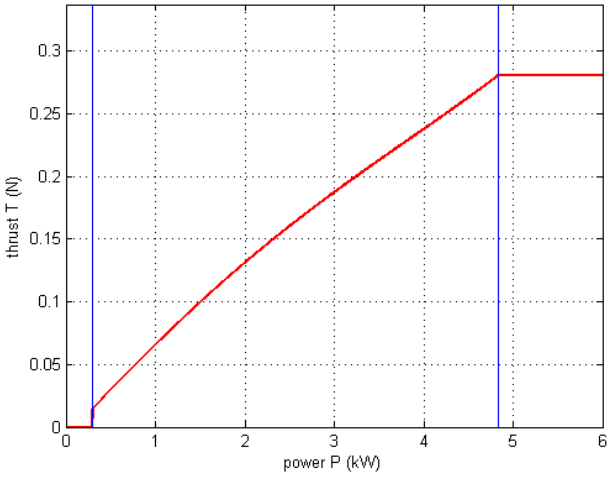

Figure 4.4. Thrust vs. power for the BPT-4000 High Thrust.

The planar two-body problem concerns a transfer orbit that insert the spacecraft from an initial circular orbit at 1 AU to a final circular orbit at 4 AU.

Initial and final boundary conditions are modelled on those of Zhang and Topputo [5][6]. The initial ones for the spacecraft in its initial circular orbit at t_i are

$$\begin{cases} r(t_i)=1 \\ \theta(t_i)=0 \\ v_r(t_i)=0 \\ v_t(t_i)=1 \end{cases} \tag{4.1}$$

and the final ones that places the spacecraft into a target circular orbit at t_f are

$$\begin{cases} r(t_f)=4 \\ v_r(t_f)=0 \\ v_t(t_f)=0.5. \end{cases} \tag{4.2}$$

The standard gravitational parameter is μ = 1, after a normalization of the distance and of the time such that the orbital period of the Earth is equal to 2π.

As introduced before in the text, the high efficient NLP solver IPOPT is used for large-scale nonlinear optimization (refer to the IPOPT documentation [11] for more details).

4.2 Time-Optimal Problem

The goal of the time-optimal problem is to find the functions $u(t)$ and $\phi(t)$ that minimize the performance index

$$J = t_f \tag{4.3}$$

under the dynamic constraint (3.4)-(3.6) and the boundary conditions (4.1)-(4.2) given above.

4.2.1 Constant Maximum Thrust Acceleration

In order to validate the modified code, the example proposed in Zhang and Topputo [5][6] is addressed first, with non-dimensional constant thrust acceleration $u_{max} = 0.01$. This problem is solved with $N = 200$ uniformly spaced points, i.e. 1201 NLP variables. Figure 4.5 shows the transfer trajectory, where the cyan line is the initial guess and the starred blue line denotes the final transfer orbit. Figure 4.6 shows the time history of the state variables. Figure 4.7 shows the profile of the optimal control variables u, ϕ, where it is possible to note that the thruster is providing its maximum thrust across the entire duration of the mission to shorten the transfer time.

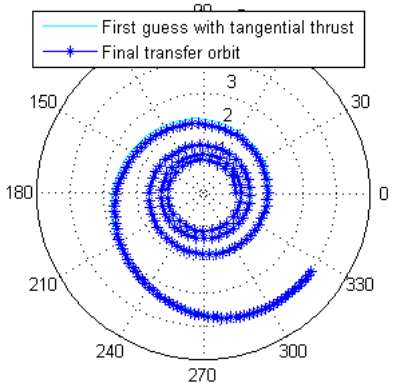

Figure 4.5. Optimal transfer orbit for time-optimal problem with constant u_{max}.

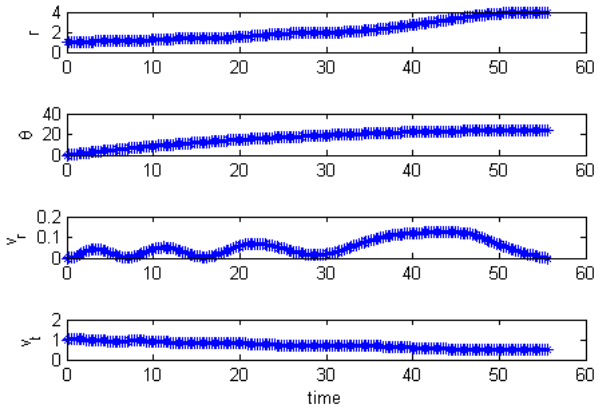

Figure 4.6. Time history of state variables for time-optimal problem with constant u_{max}.

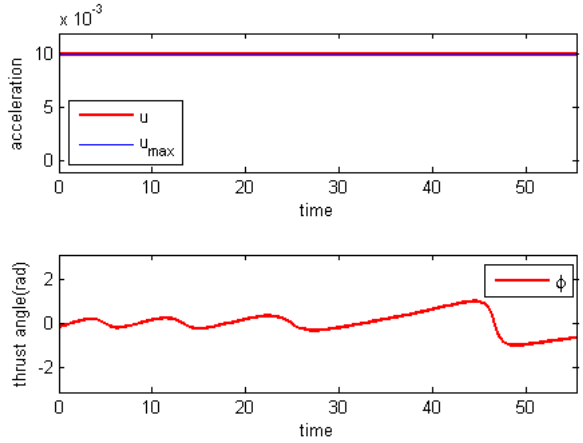

Figure 4.7. Time history of control variables for time-optimal problem with constant u_{max}.

The results are identical to those of the reference, including the non-dimensional final transfer time t_f, which is 55.5 (8.83 years). The only interesting difference is that the modified code, used in this thesis, slightly improves the computational efficiency, as shown in Table 4.4.

Table 4.4. Computational efficiency for time-optimal problem.

Method	No. of iterations	CPU secs in IPOPT	CPU secs in NLP function evaluations
Without path constraints	67	1.445	4.340
With path constraints	64	1.347	4.043

4.2.2 NSTAR Thruster

This section deals with the results of NSTAR thruster. This problem is solved with $N = 200$ uniformly spaced points. The non-dimensional final transfer time is 128.3 (20.4 years). Figure 4.8 shows the optimal orbit, Figure 4.9 shows the profile of the optimal state variables, and Figure 4.10 shows the time history of the control variables. Comparing these last two figures, it is interesting to note how the acceleration begins to decrease at a certain instant of time, following then a classical $1/r^2$ behavior. In particular, this happens when the radius is sufficiently high to reduce the available power below P_{max}, in this case about 3 AU.

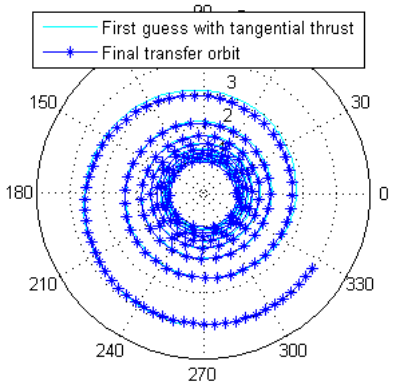

Figure 4.8. NSTAR optimal transfer for time-optimal problem.

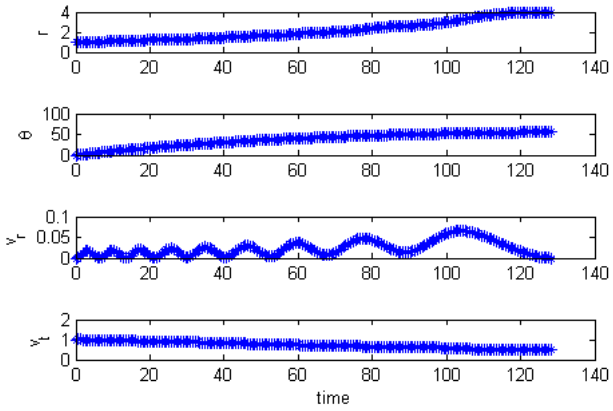

Figure 4.9. NSTAR time history of state variables for time-optimal problem.

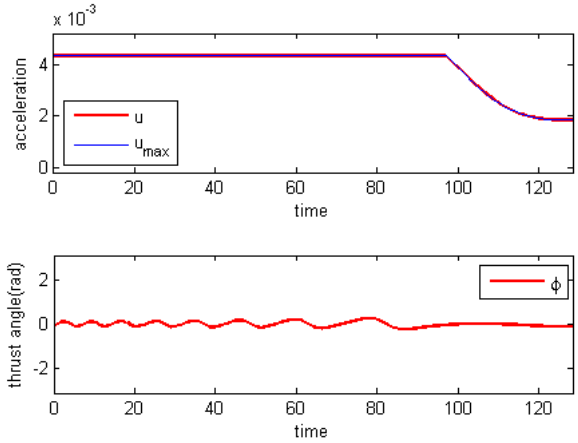

Figure 4.10. NSTAR time history of control variables for time-optimal problem.

4.2.3 XIPS-25 Thruster

The following section presents the results of XIPS-25 thruster. This problem is solved with $N = 200$ mesh points and the non-dimensional final transfer time is 88.4 (14.1 years). Figure 4.11 shows the transfer trajectory, Figure 4.12 shows the time history of state variables, and Figure 4.13 shows the optimal control variables. Also in this case the control acceleration decreases as the distance from the Sun increases. The behavior is however different from the previous

one, due to the different optimal trajectory found by the solver.

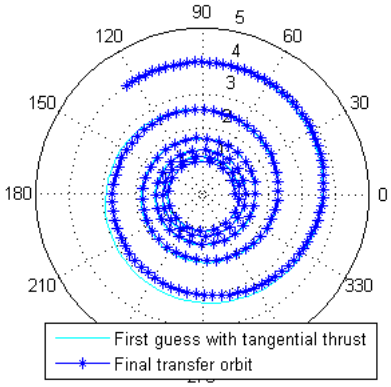

Figure 4.11. XIPS-25 optimal transfer for time-optimal problem.

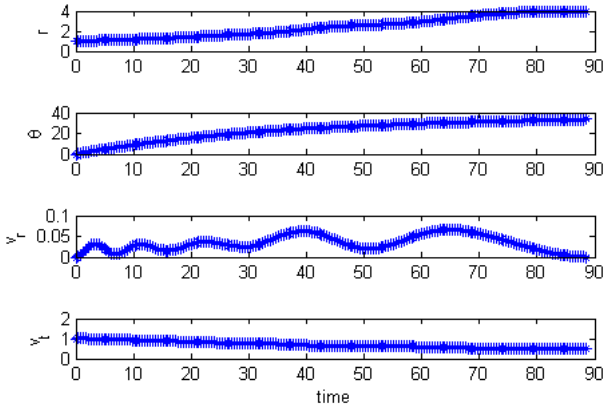

Figure 4.12. XIPS-25 time history of state variables for time-optimal problem.

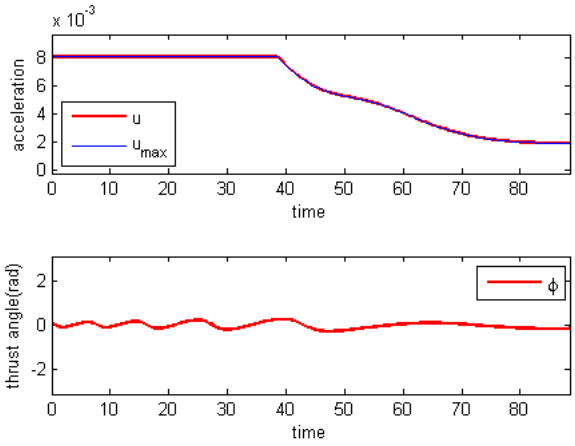

Figure 4.13. XIPS-25 time history of control variables for time-optimal problem.

4.2.4 BPT-4000 Thruster

This section deals with the performances of BPT-4000 thruster, both in the High I_{sp} mode and in High Thrust mode. This problem is solved with $N = 300$ uniformly spaced points. The non-dimensional final transfer time is 60.4 (9.6 years) for the first type, whereas it is 53.8 (8.6 years) for the second one. Figures 4.14-4.19 show the results in terms of orbit and time history of state and control variables. It can be seen that the comparison between the two graphs of the trajectory shows exactly the different transfer time between the two modes: the High I_{sp} one, indeed, takes about a half turn more to reach the final orbit than the High Thrust one.

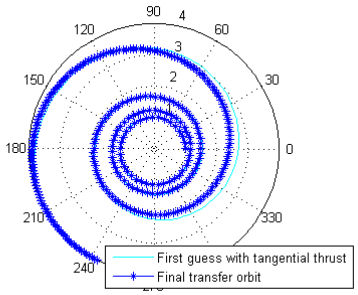

Figure 4.14. BPT-4000 High I_{sp} optimal transfer orbit for time-optimal problem.

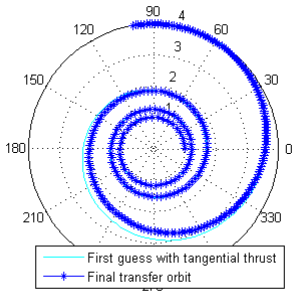

Figure 4.15. BPT-4000 High Thrust optimal transfer orbit for time-optimal problem.

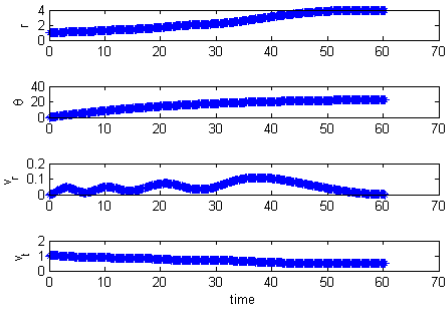

Figure 4.16. BPT-4000 High I_{sp} time history of state variables for time-optimal problem.

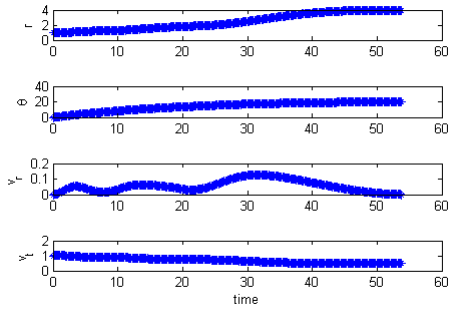

Figure 4.17. BPT-4000 High Thrust time history of state variables for time-optimal problem.

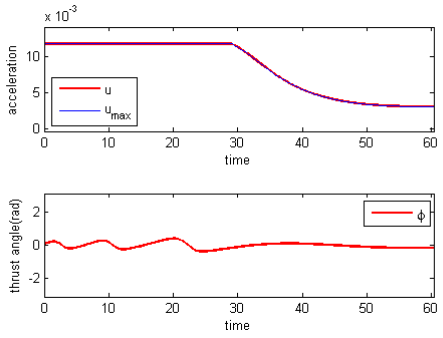

Figure 4.18. BPT-4000 High I_{sp} time history of control variables for time-optimal problem.

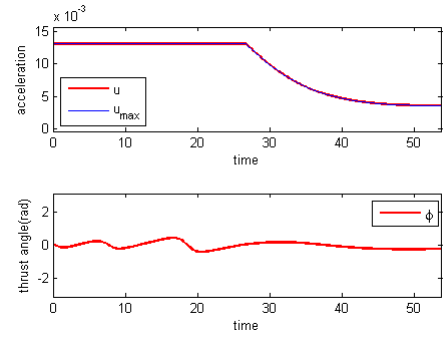

Figure 4.19. BPT-4000 High Thrust time history of control variables for time-optimal problem.

4.3 Fuel-Optimal Problem

In the fuel-optimal problem, the functions $u(t)$ and $\phi(t)$ are sought to minimize the performance index

$$J = \int_{t_i}^{t_f} u \, dt \,. \tag{4.4}$$

4.3.1 Constant Maximum Thrust Acceleration

As in the time-optimal section, the first example is the same one of Zhang and Topputo [5][6], with non-dimensional constant thrust acceleration $u_{\max} = 0.01$. This problem is solved with $N = 800$ mesh points, i.e. 4801 NLP variables, and a tangential thrust with magnitude of 0.5 $u_{\max}$ is used to produce the initial guess. Figure 4.20 shows the transfer trajectory, where the cyan line is the initial guess and the blue line denotes the final transfer orbit. Figure 4.21 shows the time history of state variables and Figure 4.22 shows the profile of the optimal control variables, u and ϕ. Making a comparison through the figures, it can be seen that the thruster is on duty only across the periapsis, whereas across the apoapsis it is performed only a maneuver to inject the spacecraft into a nearly circular orbit to acquire the final orbit. In particular, the acceleration vs time plot in Figure 4.22 shows a bang-bang structure, so indicating the optimality of the solution found.

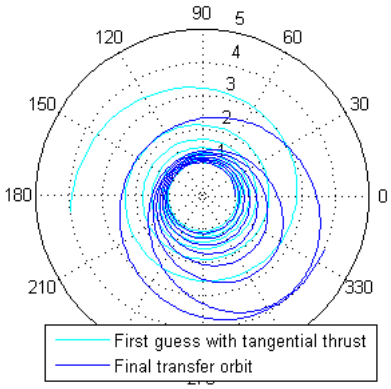

Figure 4.20. Optimal transfer orbit for fuel-optimal problem with constant u_{max}.

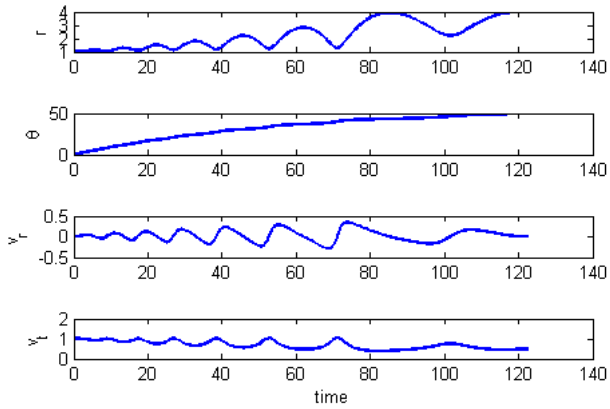

Figure 4.21. Time history of state variables of fuel-optimal problem with constant u_{max}.

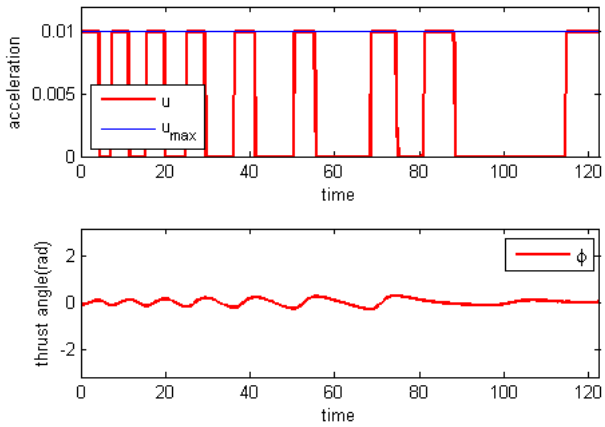

Figure 4.22. Time history of control variables of fuel-optimal problem with constant u_{max}.

The results are identical to those of the reference, including the non-dimensional final transfer time t_f, which is 122.6 (19.5 years) for both cases. Unlike the time-optimal case, the modified code does not improve the computational efficiency, as shown in Table 4.5.

Table 4.5. Computational efficiency for fuel-optimal problem.

Method	**No. of iterations**	**CPU secs in IPOPT**	**CPU secs in NLP function evaluations**
Without path constraints	304	21.264	142.360
With path constraints	368	34.408	321.345

4.3.2 NSTAR Thruster

This section deals with the results of NSTAR thruster. This problem is solved with $N = 500$ uniformly spaced points. The non-dimensional final transfer time is 283.4 (45.1 years). Figure 4.8 shows the optimal orbit, Figure 4.9 shows the

profile of the optimal state variables and Figure 4.10 shows the time history of the control variables. It can been seen that the bang-bang structure is bounded by the maximum acceleration allowed by the thruster, in particular when the spacecraft reaches the periapsis and the thruster is on duty.

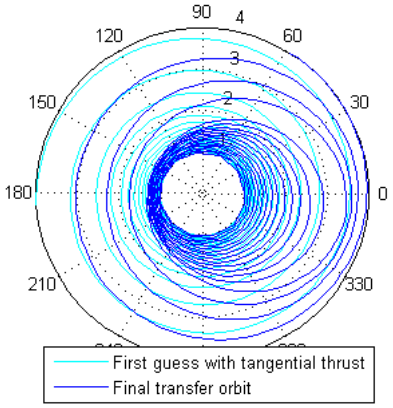

Figure 4.23. NSTAR optimal transfer orbit for fuel-optimal problem.

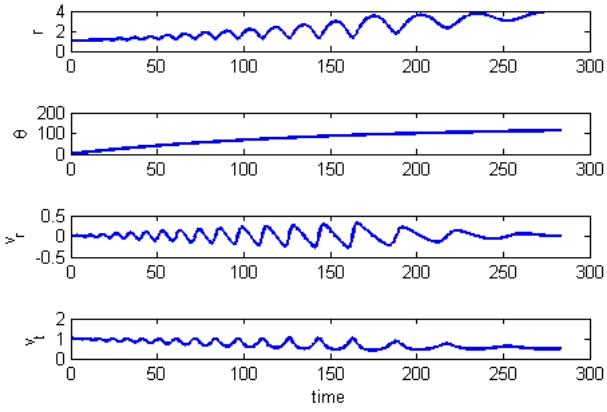

Figure 4.24. NSTAR time history of state variables for fuel-optimal problem.

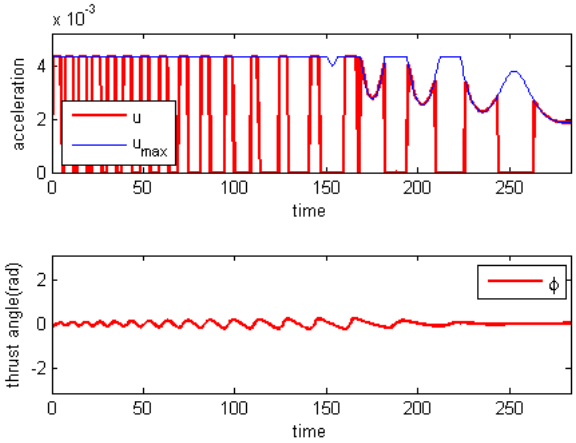

Figure 4.25. NSTAR time history of control variables for fuel-optimal problem.

4.3.3 XIPS-25 Thruster

The following section presents the results of XIPS-25 thruster. This problem is solved with $N = 700$ mesh points and the non-dimensional final transfer time is 178.6 (28.4 years). Figure 4.26 shows the transfer trajectory, Figure 4.27 the time history of state variables, and Figure 4.28 shows the optimal control variables. It is interesting to note the different behavior from the previous ion thruster, with less maneuver to reach the final orbit.

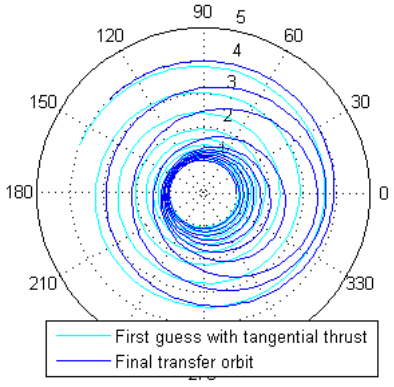

Figure 4.26. XIPS-25 optimal transfer orbit for fuel-optimal problem.

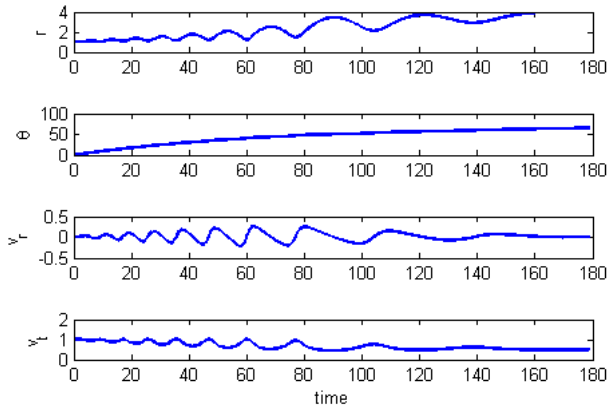

Figure 4.27. XIPS-25 time history of state variables for fuel-optimal problem.

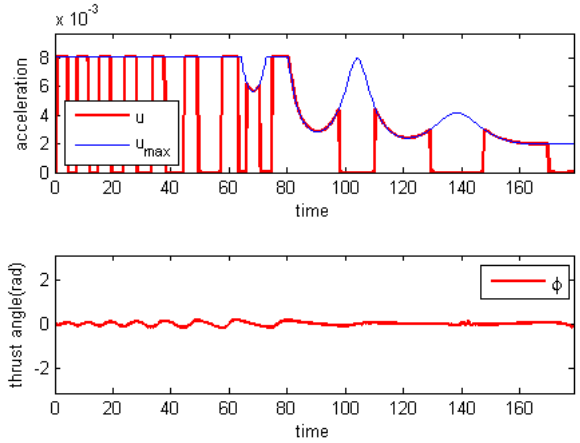

Figure 4.28: XIPS-25 time history of control variables for fuel-optimal problem.

4.3.4 BPT-4000 Thruster

This section deals with the performances of BPT-4000 thruster, both in the High I_{sp} mode and in High Thrust mode. These problems too are solved with $N = 700$ uniformly spaced points. The non-dimensional final transfer time is 120.1 (19.1 years) for the High I_{sp} mode, whereas it is 101.1 (16.1 years) for High Thrust mode. Figures 4.29-4.34 show the results in terms of orbit and time history of state and control variables. It can been seen the different optimal trajectory

performed by the thruster according to the used mode, and consequently the different bang-bang structure of the acceleration.

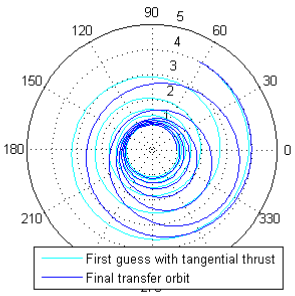

Figure 4.29. BPT-4000 High I_{sp} optimal transfer orbit for fuel-optimal problem.

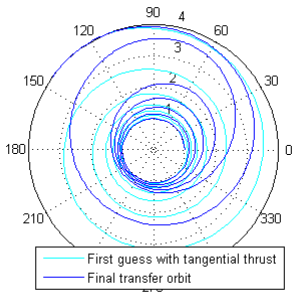

Figure 4.30. BPT-4000 High Thrust optimal transfer orbit for fuel-optimal problem.

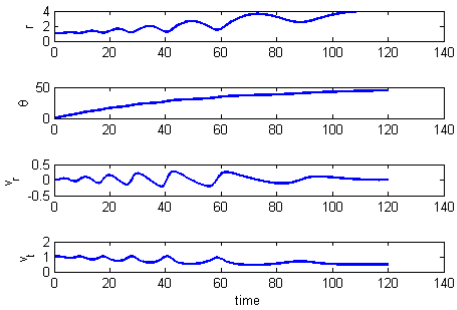

Figure 4.31. BPT-4000 High I_{sp} time history of state variables for fuel-optimal problem.

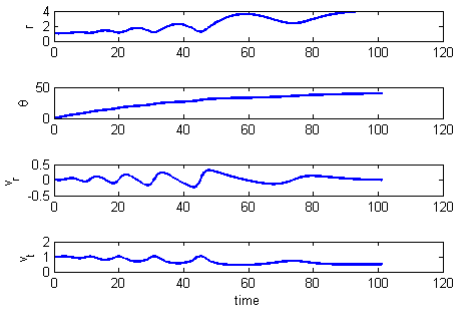

Figure 4.32. BPT-4000 High Thrust time history of state variables for fuel-optimal problem.

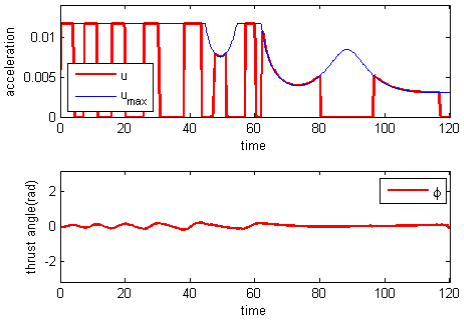

Figure 4.33. BPT-4000 High I_{sp} time history of control variables for fuel-optimal problem.

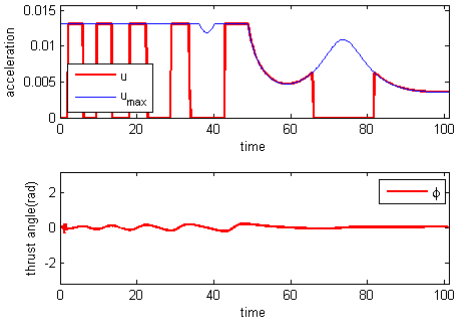

Figure 4.34. BPT-4000 High Thrust time history of control variables for fuel-optimal problem.

Chapter 5

Conclusions

This thesis consider the optimization of space trajectory with the innovative addition of solar electric propulsion models inside the algorithm of optimization. First of all, the optimal control problem and the direct method approach used to face it are discussed. Then, the models of SEP are introduced, with an explanation of the consequences in terms of the algorithm and of the mathematics of the method. Finally, both time-optimal and fuel-optimal problems are solved through the IPOPT NLP solver, with different thrusters as examples. The results confirm the initial expectations of the work, indeed the physical characteristics and limits of each thruster influence the final trajectory of the supposed mission, both in terms of orbit and time history of variables. The algorithm turns out to be relatively inexpensive, in terms of both computation time and human effort. The global search is not robust, as the optimizer may show convergence problems and is not always able to identify a global optimum solution. These problems can be partially alleviated by changing the number of mesh points, and consequently running the optimizer for a longer time.

The next steps to this project could be:

- consider the decreasing of the spacecraft mass inside the algorithm, due to the fuel consumption;
- modify the algorithm in such a way that it automatically provides the user with the best-performing thruster according to the required type of mission;
- consider more models of thrusters to enlarge the range of possibilities;
- modify the algorithm to provide the user with the wet weight of the spacecraft according to the required type of mission and required dry weight: it means considering models of all the subsystems of the spacecraft inside the optimization algorithm;
- finally, consider multiple-flyby inside the computation of the optimal trajectory.

Acronyms

EMTG	Evolutionary Mission Trajectory Generator
IPOPT	Interior Point Optimizer
NLP	Nonlinear Programming Problem
SEP	Solar Electric Propulsion

References

[1] M. Rasotto, Optimal low-thrust transfers in two-body and three-body dynamics, MSc Thesis, Politecnico di Milano, 2012.

[2] M. Catucci, Indirect optimization of time-fixed space transfers with variable specific impulse engine, MSc Thesis, Politecnico di Milano, 2013.

[3] J.T. Betts, Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics* 21.2 (1998), 193-207. doi:10.2514/2.4231

[4] P. Patel, Automating interplanetary trajectory generation for electric propulsion Trade Studies, PhD Thesis, University of Michigan, 2008.

[5] C. Zhang, F. Topputo, Survey of Direct Transcription for Low-Thrust Space Trajectory Optimization with Applications, *Abstract and Applied Analysis*, vol. 2014, Article ID 851720, 15 pages, 2014. doi:10.1155/2014/851720

[6] C. Zhang, F. Topputo, Space Trajectory Optimization Via Direct Transcription and Collocation: A Note for Practical Implementation, Scientific Report DSTA-SR 13-02, September 2013.

[7] J. A. Englander, D. H. Ellison, B. A. Conway, Global optimization of low thrust, multiple-flyby trajectories at a medium-high fidelity, *24th AAS/AIAA Space Flight Mechanics Meeting*, Sante Fe, January 2014, AAS 14-309.

[8] J. A. Englander, E. H. Cardiff, Asteroid via direct launch and solar electric propulsion, *24th AAS/AIAA Space Flight Mechanics Meeting*, Sante Fe, January 2014, AAS 14-277.

[9] D.Oh, D. Goebel, Performance evaluation of an expanded range XIPS ion thruster system for NASA science missions, American Institute of Aeronautics and Astronautics, 2006. doi:10.2514/6.2006-4466.

[10] R. Hofer, High-specific impulse operation of the BPT-4000 hall thruster for NASA science missions, American Institute of Aeronautics and Astronautics, 2010. doi:10.2514/6.2010-6623.

[11] https://projects.coin-or.org/Ipopt