Hybrid Multi-Objective Trajectory Optimization of Low-Thrust Space Mission Design

by

David Morante González

A dissertation submitted by in partial fulfillment of the requirements for

the degree of Doctor of Philosophy in

Programa Interuniversitario en Mecánica de Fluidos

Universidad Carlos III de Madrid

Advisors:
Manuel Sanjurjo-Rivo
Manuel Soler-Arnedo

Tutor:
Manuel Sanjurjo-Rivo

October 2019

Dedicated to Míriam Fernández Mata and to my mother

Acknowledgements

This work has been financially supported by Universidad Carlos III de Madrid through a PIF scholarship and has been made possible by the advocacy of several individuals. Firstly, I would like to express my most sincere gratitude to my advisors, Prof. Manuel Sanjurjo and Prof. Manuel Soler. Their guidance, sage advice, commitment, encouragement and understanding have been essential for the completion of the thesis. Thank you Sanjurjo for transmitting your passion and knowledge for Astrodynamics to me, and for always counting on me for the most exotic tasks. Your contribution has been indispensable to achieve the most challenging goals of this thesis. Thank you Soler for transmitting your passion and knowledge for Optimal Control to me, and for teaching me to write scientific documents. Your contribution has been fundamental to find the conductive thread of this thesis. Thank you both for your infinite patience when awaiting me to deliver the first draft. You must have learned that when I say "*queda poco*" or "*ya casi está*" may imply a several months. In the end, the wait was worth it.

Next, I want to thank the remaining members of the ASAP research group. Thank you Dr. Daniel González-Arribas for being a unique office mate and for sharing good moments in and out the office. I will always admire how you always find a solution for every single coding problem and at the same time go to the gym and play tennis. Thank you Prof. Javier García-Heras for showing our group your amazing organizational skills and tools. Although you were once my teacher, now we have proven to make a great team, specially when it has to do with flying quadcopters. Thank you Daniele, Aniel, Eduardo and Nacho for sharing your amazing investigations with me during the research group meetings.

I also offer my special thanks to José Manuel Sánchex Pérez and the Mission Analysis professionals at ESOC for their warm welcome to Darmstadt, where I spent four months. There, I was fortunate to learn how the European Space Agency works and to discuss about the current needs of mission designers at the agency in terms of computational tools. The lessons I learnt there allowed me to make significant improvements related to the application of operational constraints in the algorithms presented herein. José Manuel Sánchex Pérez always offered excellent input and advice. Besides, you and the people at ESOC, specially the Young Graduate Trainees, made the German winter more enjoyable than expected.

My deepest gratitude to Prof. Bruce A. Conway, who supervised my three months stay at the University of Illinois. Thank you Bruce for your continuous help, dedication and support during my internship, which was inspirational and motivating, since you wrote the first book I read to start working with my thesis. Your invaluable feedback has been paramount for the completeness of this work. The lessons I learnt there allowed me to get a deeper insight into the problem of optimizing low-thrust interplanetary trajectories with maneuver scheduling constraints. Thank you Bruce because you made me feel like home in America, and because you let me fly in your plane over Urbana.

A special mention is deserved by Jacob A. Englander, who hosted me at NASA Goddard Spaceflight Center. Thank you Jacob for your continuous fights with the administration to get me there. Although I only spent one month within your team, it was one of the most amazing experiences I have ever lived, personally and professionally. I was fortunate to use the algorithms I had implemented for my thesis to prepare a proposal for a real mission within a team of renowned professionals. The lessons I learnt there allowed me to significantly enhance the capabilities of the implemented tool. Thank you Jacob because your guiding, advice, knowledge and encouragement is part of this thesis.

Many thanks to Javier Roa and to Jesús Peláez for their invaluable assistance and help during the implementation of the generalized logarithmic spirals code. Thank you also to Hodei, Gonzalo and Davide for guiding my first steps of this research venture, while I was a member of the Space Dynamics Group at the Universidad Politécnica de Madrid. Many thanks to my colleagues at the aerospace department at the Universidad Carlos III de Madrid: Marco, Rocco, Luca, Gonzalo, Xin, Víctor, Manu, Carlos, Álex, Filippo, Álvaro, Adri, and the new incorporations. Thank you for your company during lunch and breaks, and because only you can understand what doing PhD thesis implies.

Este documento no sólo es fruto del trabajo realizado durante cuatro años de investigación pre-doctoral y 9 meses escribiendo (sí, un embarazo), sino también de 28 años donde todos y cada uno de mis pasos me han hecho escribir esta tesis doctoral tal y como es. Por ello, nada de esto habría sido posible sin el apoyo incondicional y constante de mi familia. Graciás, Mamá, por tu educación y cariño, por enseñarme que puedo conseguir todo lo que me proponga a base de esfuerzo, por animarme en los momentos más difíciles y por enseñarme a ser mejor persona. Gracias, Papá, por ser mi modelo a seguir, por levantarte todos los días para trabajar 12 horas en un bar con el objetivo de darnos lo mejor. Gracias a mi hermanos, Alberto, Jesús y Paloma (mi niña) por ser fieles compañeros de vida, aventuras y confesiones. En mí hay un poco de todos vosotros y por ende en esta tesis.

Gracias a mis amigos de siempre, por eso, por seguir ahí. Gracias Tania por crear la leyenda del pato-pez, por no permitir que nos robasen los órganos en Mallorca, por lo bailes de pasillo, por Reina Victoria y por Albacete y por tantas otras cosas que me darían para escribir otra tesis. Gracias Rosa por mantenernos a todos siempre informados de cualquier cotilleo que pueda haber. Gracias Ferro por amenizar cualquier ocasión con una bonita (?) canción. Gracias Edu, Fran y Yuste porque sin vosotros nuestras reuniones estarían vacías. Mil gracias a todos y cada uno de mis compañeros de carrera de cañitas: Enrique, Sara, Andrei, Miguel, Paula, Koldo, Marta, Roci, Sergio y Osorio. Gracias porque juntos a vosotros logré terminar la carrera que me abrió las puertas al doctorado, y junto a vosotros conseguí escribir esta tesis. Gracias Bea por enseñarme el mundo amarillo. Gracias a mis amigos de BP: Samu, J. Luis, Álvaro y demás por enseñarme que siempre hay vivir apostando el all-in.

Gracias a ti, Míriam, por ser el pilar fundamental de mi vida. Gracias porque mientras escribía todas y cada una de las palabras de esta tesis tú estabas en mi cabeza. Gracias por aguantar todos los contratiempos, por la paciencia, por la compresión y por cuidarme. Mil gracias por hacer que escribir está tesis haya sido mucho más fácil. Gracias por hacerme ver que hay mundo más allá del doctorado y por ofrecerme el punto de vista de una periodista profesional. Gracias por animarme todos los días a seguir escribiendo y a empujarme siempre más alto. Gracias por ser y por estar.

No podría cerrar esta sección sin poner una mirada en el futuro, el cual tiene nombre de start-up: Ienai SPACE. Gracias Dani, Sara y Mick por permitirme formar parte de esta “astroaventura” empresarial. Gracias por mostrarme que todas la habilidades y conocimientos adquiridos durante el doctorado pueden tener una aplicación en el sector comercial. Gracias por adentrarme en el fascinante mundo del diseño de motores de propulsión eléctrica. Gracias por ofrecerme la oportunidad de aprender sobre finanzas, ventas o negociación con inversores. Gracias por creer, por luchar, por ilusionar y por trabajar incansablemente por una idea. Gracias por hacer que lo mejor esté aún por venir...

Published and Submitted Content

The following publications are partially included in the thesis.

Journal Papers

1. D. Morante, M. Sanjurjo-Rivo, and M. Soler, "Multi-Objective Low-Thrust Interplanetary Trajectory Optimization Based on Generalized Logarithmic Spirals", *Journal of Guidance, Control, and Dynamics*, vol. 42, pp. 476-490, 2019 (https://doi.org/10.2514/1.G003702). Material from this publication is included mainly in Chapter 3 and Chapter 5 and to a smaller extent in Chapter 1, Chapter 2. Author roles: Daniel Morante (DM) carried out the development and implementation of the method and the analysis of the results; the paper was written by DM with feedback from Sanjurjo-Rivo (MSR) and Manuel Soler (MS).

Conference Proceedings

1. D. Morante, M. Sanjurjo-Rivo, and M. Soler, "MOLTO-OR: A Multi-Objective Low-Thrust Optimization Tool for Orbit Raising", in *Proceedings of* 7^{th} *International Conference on Astrodynamics Tools and Techniques (ICATT)*, nov 2018 (https://indico.esa.int/event/224/contributions/3859/contribution.pdf). Material from this publication is included mainly in Chapter 4. Author roles: DM carried out the development and implementation of the algorithm and the analysis of the results; the paper was written by DM with feedback from MSR and MS.
2. D. Morante, M. Sanjurjo-Rivo, and M. Soler, "MOLTO-IT: A Multi-Objective Low-Thrust Optimization Tool for Interplanetary Trajectories", in *Proceedings of* 7^{th} *International Conference on Astrodynamics Tools and Techniques (ICATT)*, nov 2018 (https://indico.esa.int/event/224/contributions/3860/contribution.pdf). Material from this publication is included mainly in Chapter 5. Author roles: DM carried out the development and implementation of the algorithm and the analysis of the results; the paper was written by DM with feedback from MSR and MS.
3. D. Morante, J. C. Macías, M. Sanjurjo-Rivo, and M. Soler, "Multiobjective Trajectory Optimizations for Orbit Raising with Combined Chemical-Electric Propulsion", in *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, aug 2017 (http://www.univelt.com/book=6564). Material from this publication is included mainly in Chapter 4. Author roles: DM carried out the development and implementation of the algorithm and the analysis of the results; the paper was written by DM with feedback from MSR and MS.
4. D. Morante, M. Sanjurjo-Rivo, and M. Soler, "Heuristic method based on Generalized Logarithmic Spirals for Low-Thrust trajectory design", in *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, sep 2016 (https://doi.org/10.2514/6.2016-5638).

5. D. Morante, M. Sanjurjo-Rivo, and M. Soler, "Sequential Method for Computing Multiobjective Optimal Low-Thrust Earth Orbit Transfers", in *Proceedings of* 6th *International Conference on Astrodynamics Tools and Techniques (ICATT))*, apr 2016. Material from this publication is included mainly in Chapter 4. Author roles: DM carried out the development and implementation of the algorithm and the analysis of the results; the paper was written by DM with feedback from MSR and MS.

6. D. Morante, M. Sanjurjo-Rivo, and M. Soler, "Low-thrust earth-orbit transfer optimization using analytical averaging within a sequential method", in *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, aug 2015 (http://www.univelt.com/book=5479). Material from this publication is included mainly in Chapter 4. Author roles: DM carried out the development and implementation of the algorithm and the analysis of the results; the paper was written by DM with feedback from MSR and MS.

The material from these sources included in this thesis is not singled out with typographic means and references

Other Research Merits

The following publications describe additional research performed during the thesis that is related to various degrees. Nevertheless, their content is not included in this thesis.

Journal Papers

1. H. Urrutxua, D. Morante, M. Sanjurjo-Rivo, J. Peláez, "DROMO formulation for planar motions: solution to the Tsien problem.", *Celestial Mechanics and Dynamical Astronomy*, vol. 122, pp. 143-168, 2015 (https://doi.org/10.1007/s10569-015-9613-8).

Conference Proceedings

1. R. Vega, D. Morante, M. Sanjurjo-Rivo, and M. Soler, "Analysis of a Rendezvous Mission in Non-Keplerian Orbit using Electric Propulsion", in *Proceedings of* 7^{th} *International Conference on Astrodynamics Tools and Techniques (ICATT)*, nov 2018.
2. P. Machuca, D. González-Arribas, D. Morante, M. Sanjurjo-Rivo, and M. Soler,"Robust optimization of descent trajectories on irregular-shaped bodies in the presence of uncertainty," in *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, aug 2017

“Per aspera ad astra.”

Universidad Carlos III de Madrid

Abstract

Department of Bioengineering and Aerospace Engineering

Hybrid Multi-Objective Trajectory Optimization of Low-Thrust Space Mission Design

by David Morante González

The overall goal of this dissertation is to develop multi-objective optimization algorithms for computing low-thrust trajectories. The thesis is motivated by the increasing number of space projects that will benefit from low-thrust propulsion technologies to gain unprecedented scientific, economic and social return. The low-cost design of such missions and the inclusion of concurrent engineering practices during the preliminary design phase demand advanced tools to rapidly explore different solutions and to benchmark them with respect to multiple conflicting criteria. However, the determination of optimal low-thrust transfers is a challenging task and remains an active research field that seeks performance improvements. This work contributes to increase the efficiency of searching wide design spaces, reduce the amount of necessary human involvement, and enhance the capabilities to include complex operational constraints. To that end, the general low-thrust trajectory optimization problem is stated as a multi-objective Hybrid Optimal Control Problem. This formulation allows to simultaneously optimize discrete decision-making processes, discrete dynamics, and the continuous low-thrust steering law. Within this framework, a sequential two-step solution approach is devised for two different scenarios.

The first problem considers the optimization of low-thrust multi-gravity assist trajectories. The proposed solution procedure starts by assuming a planar shape-based model for the interplanetary trajectory. A multi-objective heuristic algorithm combined with a gradient-based solver optimize the parameters defining the shape of the trajectory, the number and sequence of the gravity assists, the departure and arrival dates, and the launch excess velocity. In the second step, candidate solutions are deemed as initial guesses to solve the Nonlinear Programming Problem resulting from applying a direct collocation transcription scheme. In this step, the sequence of planetary gravity assists is known and provided by the heuristic search, dynamics is three-dimensional, and the steering law is not predefined. Operational constraints to comply with launch asymptote declination limits and fixed reorientation times during the transfer apply. The presented approach is tested on a rendezvous mission to Ceres, on a flyby mission to Jupiter, and on a rendezvous mission to Pluto. Pareto-optimal solutions in terms of time of flight and propellant mass consumed (or alternatively delivered mass) are obtained. Results outperform those found in the literature in terms of optimality while showing the effectiveness of the proposed methodology to generate quick performance estimates.

The second problem considers the simultaneous optimization of fully electric, fully chemical and combined chemical-electric orbit raising transfers between Earth's orbits is considered. In the first step of the solution approach, the control law of the electric engine is parameterized by a Lyapunov function. A multi-objective heuristic algorithm selects the optimal propulsion system, the transfer type, the low-thrust control history, as well as the number, orientation, and magnitude of the chemical firings. Earth's shadow, oblateness and Van-Allen radiation effects are included. In the second step, candidate solutions are deemed as initial guesses to solve the Nonlinear Programming Problem resulting from applying a direct collocation scheme. Operational constraints to avoid the GEO ring in combination to slew rate limits and slot phasing constraints are included. The proposed approach is applied to two transfer scenarios to GEO orbit. Pareto-optimal solutions trading off propellant mass, time of flight and solar-cell degradation are obtained. It is identified that the application of operational restrictions causes minor penalties in the objective function. Additionally, the analysis highlights the benefits that combined chemical-electric platforms may provide for future GEO satellites.

Universidad Carlos III de Madrid

Resumen

Department of Bioengineering and Aerospace Engineering

Hybrid Multi-Objective Trajectory Optimization of Low-Thrust Space Mission Design

by David Morante González

El objetivo principal de esta trabajo es desarrollar algoritmos de optimización multi-objetivo para la obtención de trayectorias espaciales con motores de bajo empuje. La tesis está motivada por el creciente número de misiones que se van a beneficiar del uso de estas tecnologías para conseguir beneficios científicos, económicos y sociales sin precedentes. El diseño de bajo coste de dichas misiones ligado a los principios de ingeniería concurrente requieren herramientas computacionales avanzadas que exploren rápidamente distintas soluciones y las comparen entre sí respecto a varios criterios. Sin embargo, esta tarea permanece como un campo de investigación activo que busca continuamente mejoras de rendimiento durante el proceso. Este trabajo contribuye a aumentar la eficiencia cuando espacio de diseño es amplio, a reducir la participación humana requerida y a mejorar las capacidades para incluir restricciones operacionales complejas. Para este fin, el problema general de optimización de trayectorias de bajo empuje se presenta como un problem híbrido de control óptimo. Esta formulación permite optimizar al mismo tiempo procesos de toma de decisiones, dinámica discreta y la ley de control del motor. Dentro de este marco, se idea un algoritmo secuencial de dos pasos para dos escenarios diferentes.

El primer problema considera la optimización de trayectorias de bajo empuje con múltiples fly-bys. El proceso de solución propuesto comienza asumiendo un modelo plano y *shape-based* para la trayectoria interplanetaria. Un algoritmo de optimización heurístico y multi-objetivo combinado con un resolvedor basado en gradiente optimizan los parámetros de la espiral que definen la forma de la trayectoria, el número y la secuencia de las maniobras gravitacionales, las fechas de salida y llegada, y la velocidad de lanzamiento. En el segundo paso, las soluciones candidatas se usan como estimación inicial para resolver el problema de optimización no lineal que resulta de aplicar un método de transcripción directa. En este paso, las secuencia de fly-bys es conocida y determinada por el paso anterior, la dinámica es tridimensional, y la ley de control no está predefinida. Además, se pueden aplicar restricciones operacionales relacionadas con las declinación de la asíntota de salida e imponer tiempos de reorientación fijos. Este enfoque es probado en misiones a Ceres, a Júpiter y a Plutón. Se obtienen soluciones óptimas de Pareto en función del tiempo de vuelo y la masa de combustible consumida (o la masa entregada). Los resultados obtenidos mejoran los disponibles en la literatura en términos de optimalidad, a la vez que reflejan la efectividad de la metodología propuesta para generar estimaciones rápidas.

El segundo problema considera la optimización simultanea de transferencias entre órbitas terrestres que usan propulsión eléctrica, química o una combinación de ambas. En el primer paso del método de solución, la ley de control del motor elétrico se parametriza por una función de Lyapunov. Un algoritmo de optimización heurístico y multi-objetivo selecciona el sistema propulsivo óptimo, el tipo de transferencia, la ley de control del motor de bajo empuje, asícomo el número, la orientación y la magnitud de los impulsos químicos. Se incluyen los efectos de la sombra y de la no esfericidad de la Tierra, además de la radiación de Van-Allen. En el segundo paso, las soluciones candidatas se usan como estimación inicial para resolver el problema de optimización no lineal que resulta de aplicar un método de transcripción directa. El método de solución propuesto se aplica a dos transferencias a GEO diferentes. Se obtienen soluciones ótimas de Pareto con respecto a la masa de combustible, el tiempo de vuelo y la degradación de las células solares. Se indentifica que la aplicación de las restricciones operacionales penaliza mínimamente la función objetivo . Además, los análisis presentados destacan los beneficios que la propulsión química y eléctrica combinada proporcionarían a los satélites en GEO.

Contents

List of Figures

List of Tables

Abbreviations

CCEP	**C**ombined **C**hemical-**E**lectric **P**ropulsion
CCET	**C**ombined **C**hemical-**E**lectric **T**ransfer
COCP	**C**ontinuous **O**ptimal **C**ontrol **P**roblem
COE	**C**lassical **O**rbital **E**lements
COV	**C**alculus **O**f **V**ariations
CP	**C**hemical-**P**ropulsion
CSV	**C**artesian **S**tate **V**ector
DDP	**D**ifferential **D**ynamic **P**rogramming
EMTG	**E**volutionary **M**ission **T**rajectory **G**enerator
EOM	**E**quations **O**f **M**otion
EP	**E**lectric-**P**ropulsion
ESA	**E**uropean **S**pace **A**gency
FET	**F**ully **E**lectric **T**ransfer
FCT	**F**ully **C**hemical **T**ransfer
GA	**G**enetic **A**lgorithm
GALLOP	**G**ravity-**A**ssist **L**ow-thrust **L**ocal **O**ptimization **P**rogram
GB	**G**radient-**B**ased
GEO	**G**eosynchronous **E**arth **O**rbit
GTO	**G**eosynchronous **T**ransfer **O**rbit
HJB	**H**amilton-**J**acobi-**B**ellman
HOCP	**H**ybrid **O**ptimal **C**ontrol **P**roblem
HS	**H**euri**S**tic
HY	**HY**brid
IT	**I**nterplane**T**ary

KKT	**K**arush-**K**uhn-**T**ucker
KM	**S**tark-**M**odel
LEO	**L**ow **E**arth **O**rbit
LT	**L**ow **T**hrust
MEE	**M**odified **E**quinoctial **E**lements
MGA	**M**ulti-**G**ravity **A**ssist
MINLP	**M**ixed-**I**nteger **N**on**L**inear **P**rogramming
MO	**M**ulti-**O**bjective
MOLTO	**M**ulti-**O**bjective **L**ow-**T**hrust **O**ptimizer
MPBVP	**M**ulti-**P**oint **B**oundary **V**alue **P**roblem
NASA	**N**ational **A**eronautics and **S**pace **A**dministration
NBP	**N**-**B**ody **P**roblem
NLP	**N**on-**L**inear **P**rogramming
NSGA	**N**on-dominated **S**orting **G**enetic **A**lgorithm
OCP	**O**ptimal **C**ontrol **P**roblem
OR	**O**rbit-**R**aising
PDE	**P**artial **D**ifferential **E**quation
PMP	**P**ontryagin **M**inimum **P**rinciple
PR	**P**erturbed-**R**estricted
PSO	**P**article **S**warm **O**ptimization
PSV	**P**olar **S**tate **V**ector
SDC	**S**tatic-**D**ynamic **C**ontrol
SM	**K**epler-**M**odel
SO	**S**ingle-**O**bjective
STOUR	**S**atellite **TOUR** Design Program
TBP	**T**wo **B**ody **P**roblem
TOF	**T**ime **O**f **F**light

Symbols

α	In-plane thrust steering angle	rad
β	Yaw thrust steering angle	rad
γ	Solar array polynomial parameters	-
δ	Flyby in-plane rotational angle	deg
ϵ	Auxiliary parameter	-
ζ	B-plane flyby angle	deg
η	Electric engine efficiency coefficient	$-$
η_a	Q-law effectivity coefficient	$-$
η_r	Q-law effectivity coefficient	$-$
ξ	Spiral control parameter	$-$
θ	Polar angle	rad
κ	Lagrange multiplier of the path constraints	-
λ	Costate vector	-
μ	Gravitational parameter	$m^3 s^{-2}$
ν	Lagrange multiplier of the boundary constraints	-
ξ	Spiral control parameter	$-$
ρ	Fitness function penalty parameter	$-$
σ	Sequence of discrete events	$-$
ϕ	Transition map functions	$-$
ϕ_e	Earth's latitude	deg
ψ	Flight path angle	rad
ω	Argument of pericenter	rad

Θ	Set of transfer type options	-
Ξ	Binary shadow function	-
ϖ	Spiral anomaly	deg
Π	Set of propulsive system options	-
Λ	Mac-Ilwain Coordinate	m
Φ_p	Proton flux	$cm^{-2}s^{-1}$
Ψ_p	Proton fluence	cm^{-2}
Ω	Right Ascension of ascending node	rad
$\mathbb{R}$	Set of real numbers	-
$\mathbb{Z}$	Set of integer numbers	-
$\mathcal{C}$	Set of chemical transfer parameters	-
$\mathcal{E}$	Set of electric transfer parameters	-
$\mathcal{H}$	Set of hybrid transfer parameters	-
$\mathcal{I}$	Inertial reference frame	-
$\mathcal{L}$	Running cost (Lagrange term)	-
$\mathcal{M}$	Terminal cost (Mayer term)	-
$\mathcal{O}$	Set of orbit raising problem parameters	-
$\mathcal{Q}$	Set of admissible discrete states	-
$\mathcal{U}$	Set of admissible continuous controls	-
$\mathcal{V}$	Set of admissible discrete controls	-
$\mathcal{X}$	Set of admissible continuous states	-
$\mathcal{Z}$	Set of unknown NLP decision variables	-
$\mathbf{a}_P$	Perturbing acceleration vector	ms^{-2}
$\mathbf{a}_T$	Thrust acceleration vector	ms^{-2}
$\mathbf{b}$	Second basis vector of intrinsic frame	-
$\mathbf{d}$	Thrust unitary direction vector	-
$\mathbf{i}$	First basis vector of cartesian frame	-
$\mathbf{j}$	Second basis vector of cartesian frame	-
$\mathbf{k}$	Third basis vector of cartesian frame	-

$\mathbf{n}$	Third basis vector of intrinsic frame	-
$\boldsymbol{q}$	Discrete state vector	-
$\mathbf{r}$	Position vector	m
$\mathbf{t}$	First basis vector of intrinsic frame	-
$\boldsymbol{u}$	Continuous control vector	-
$\boldsymbol{v}$	Discrete control vector	-
$\mathbf{v}$	Velocity vector	ms^{-1}
$\boldsymbol{x}$	Continuous state vector	-
$\boldsymbol{z}$	Unknown NLP variables	-
a	Semimajor axis	m
b	Auxiliary parameter	-
c	NLP constraints	-
c_T	Thrust polynomial coefficients	-
c_m	Fuel rate polynomial coefficients	-
D	Displacement Damage Dose	-
e	Eccentricity	-
E	Energy	MeV
f	Dynamical function	
g	Path constraints	-
g_0	Earth's gravitational acceleration at sea level	ms^{-2}
h	Boundary constraints	-
H	Hamiltonian	
i	Orbital inclination	rad
J	Cost functional	$-$
K_1	Spirals constants of motion	m^2s^{-1}
K_2	Spirals constants of motion	m^2s^{-2}
l	Auxiliary parameter	-
I_{sp}	Specific impulse	ms^{-1}
L	True anomaly	rad
m	Spacecraft mass	kg
m_p	Propellant mass	kg
m_Q	Q-law scaling parameter	-

M	Mean anomaly	rad
n_Q	Q-law scaling parameter	-
n_c	Number of NLP constraints	$-$
n_g	Number of path constraints	$-$
n_h	Number of boundary constraints	$-$
n_k	Number of objective functions	$-$
n_q	Dimension of the discrete state vector	$-$
n_x	Dimension of the continuous state vector	$-$
n_u	Dimension of the continuous control vector	$-$
n_v	Dimension of the discrete control vector	$-$
n_s	Number of discrete event functions	$-$
p	Semi-latus rectum	m
P	Power	W (Js^{-1})
Q	Lyapunov coefficient	
r	Distance	m
r_Q	Q-law scaling parameter	-
R_E	Radius of the Earth	m
s	Discrete event functions	$-$
S_P	Non-ionizing energy loss	$-$
t	Time variable	s
t_0	Initial time	s
t_f	Final time	s
T	Orbital Period	s
T	Electric engine thrust	N
v	Velocity	m/s
V	Value function	$-$
w	Weighting parameters	-
w_E	Earth's angular velocity	rads^{-1}
W_p	Q-law weighting factor	$-$
W_f	Q-law weighting factor	$-$
W_g	Q-law weighting factor	$-$
W_h	Q-law weighting factor	$-$
W_k	Q-law weighting factor	$-$

Subscripts

a	Available	–
fb	Flyby	–
e	Electrical	–
f	Final	–
g	Generated	–
ch	Chemical	–
la	Launch	–
N	North	–
S	South	–
th	Threshold	–
0	Initial	–
on	On-switching	–
off	Off-switching	–

1

Motivation and Thesis Outline

Contents

1.1 Motivation

The exploration and exploitation of outer space play an essential role in the efficient functioning of modern societies. It contributes to advance scientific knowledge and technology innovation, to meet global challenges on Earth, as well as to generate substantial commercial revenues. Historically, space activities have been dominated by space-faring countries with large economies, a few big commercial enterprises, and little competition. However, over the past decade the number of private and public players involved in space activities has increased. As a consequence, the space sector is undergoing fundamental transformations towards a more global and diverse ecosystem with a mix of government and commercial initiatives, a variety of contractors, and stiff competition. Meanwhile, missions of growing levels of sophistication, complexity, and scientific return are being proposed for the forthcoming years. Indeed, envisioned projects include mega-constellations of small satellites orbiting Earth, probes landing on the moons of outer planets, and even human settlements being established on Mars.

In such scenario, reducing the cost and schedule of accessing and using space without compromising quality and safety becomes a major goal. The potential benefits translate not only into economic gains for commercial space actors, yet into enhancing or enabling future scientific missions that cannot currently be accomplished due to budget or technological limitations. For such purpose, novel mission architectures and breakthrough technologies have become primary tools. Among them, the development of new commercial launch systems, the thriving generation of small satellites prompted by miniaturized but fully functional electronics, the recent advances in computers and material sciences, and the implementation of distributed mission concepts will be shaping the global space sector during the next decades. On top of that, ambitious future projects will continue to benefit from the high fuel efficiency inherent to the well-stablished electric propulsion systems. Similarly, the use of gravity assisted maneuvers[1] will remain as the chief means to lower the cost of reaching distant targets in the Solar System.

Space mission analysis and design activities are also experiencing a paradigm shift to more rapid and cost-effective processes based on concurrent engineering principles. Contrary to traditional methods, in concurrent engineering the elements of the mission architecture along with the spacecraft subsystems are designed simultaneously. This strategy demands multi-disciplinary software tools able to provide real-time performance tradeoffs between the available options. However, these requirements are difficult to be achieved in missions where the spacecraft has to travel from the injection orbit into its final destination using multiple gravity assists and/or electric propulsion. Mission designers have to obtain the transfer trajectory, the steering law of the electric engine, and/or the optimal sequence of swing-bys that best accomplish the mission goals, while satisfying subsystems' constraints and operational restrictions. Consequently, this task becomes a very expensive process in terms of human and computer hours, where any increase in automation or reduction in execution times are highly desirable.

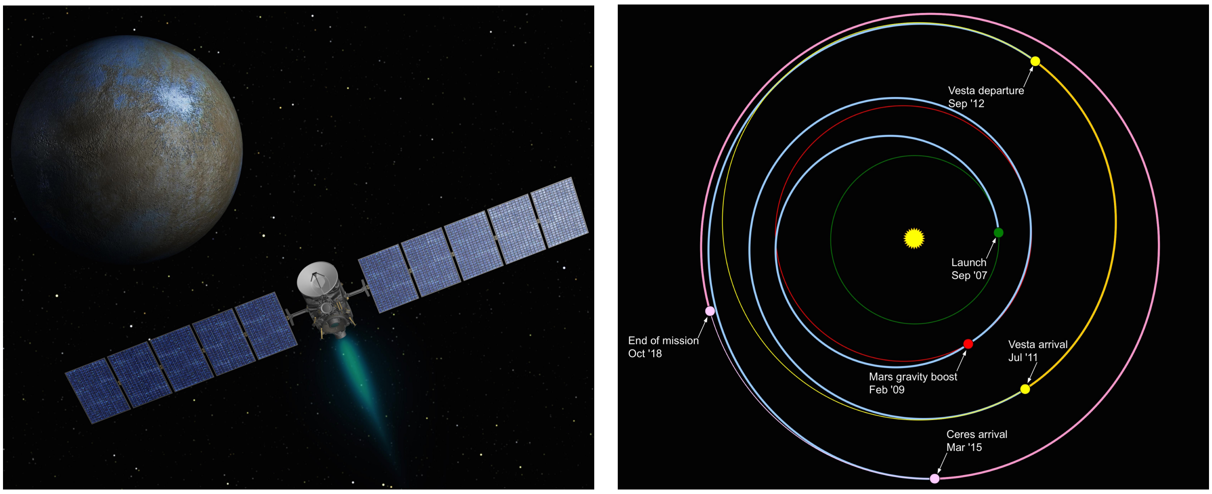
[1] Gravity assists maneuvers are also know as gravitational slingshots or swing-bys.

1.1.1 The Role of Electric Propulsion

Today's mature space propulsion technologies include electric propulsion (EP), and chemical propulsion (CP). The former features a higher propellant efficiency to produce the same overall effect, offering substantial leverage for reducing launch mass, increasing payload capacity or prolonging the service life. Additionally, EP can operate over long time periods, allowing to continuously modify the trajectory. Thus, more frequent launch opportunities are enabled, and the sensitivity to launcher injection accuracy is reduced. The current disadvantage of EP is its very low-thrust levels, unlike their chemical counterparts, because they are limited by the available electrical power on-board. To date, only solar photovoltaic cells have been implemented in real missions as power source. Because attainable solar power diminishes as the spacecraft moves farther from the Sun, the spacecraft becomes less capable of maneuvering and eventually useless. Subsequently, EP have their greatest benefits in missions within the inner solar system requiring small thrust [1] (e.g., station-keeping maneuvers) or long journeys (e.g., interplanetary transfers). An excellent overview of EP principles can be found in [2].

The first major breakthrough was NASA's Deep Space 1 (1998) [3], which flew by asteroid 9969 Braille. Thereafter, a series of successful deep-space missions using EP were carried out. In 2003, JAXA's Hayabusa [4] performed a rendezvous with asteroid 1998 SF36 Itokawa, after a flyby on Earth, and returned samples to Earth in 2010. In the same year ESA launched SMART-1 [5], a technology demonstrator, which used an EP thruster and multiple Lunar gravity assists, to target the Moon. Later, in 2017, NASA launched the Dawn mission [6], which was developed to rendezvous two of the largest objects in the main asteroid belt, Asteroid Vesta and dwarf planet Ceres, after a Mars flyby (See Fig. 1.8). It was followed in 2014 by Hayabusa-2 [7] to rendezvous with the near-Earth asteroid 162173 Ryugu in 2018 after an Earth flyby. It is expected to leave the asteroid at the end of 2019 and return to Earth around the end of 2020. Bepicolombo [8], the joint ESA/JAXA mission. It was launched in 2018 and will arrive at Mercury in late 2025 after a flyby of Earth, two flybys of Venus and six flybys of Mercury.

Most of these missions have combined the benefits steaming from EP with gravity assisted maneuvers at intermediate planets to further reduce propellant needs. In fact, journeys to distant targets (e.g., Mercury, Jupiter, most comets and asteroids) or sample-return missions may be infeasible by direct trajectories, even when using EP. Thus, gravity assists may be required to enable interplanetary endeavours. During the close approach there is an energy exchange between the spacecraft and the planet that is flown by. Because of the difference in their masses the change in the orbit of the planet is negligible, but the spacecraft experiences and important change in its velocity (i.e., increased or decreased velocity). The swing-by technique has been widely used since the 1970s. It was first applied during the Mariner-10 mission to Mercury [9], but most notably for the Voyager program to study the outer Solar System [10]. Notwithstanding their benefits, this mission architecture typically increase the flight time, its operational and design complexity, and narrow the launch window options.



(A) Dawn Space probe. Credit: NASA

(B) Baseline Trajectory. Credit: NASA

FIGURE 1.1: Artistic view of NASA's Dawn Mission

TABLE 1.1: Representative Electric Thrusters for Deep Space Missions

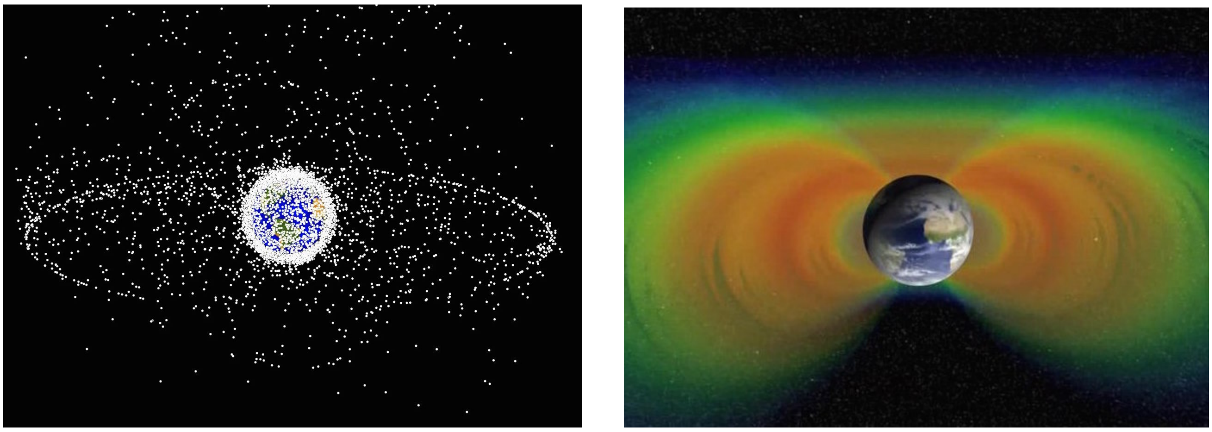
Thruster	Manufacturer	Power (W)*	Thrust (mN)*	$I_{sp}(s)$*	Application	Launch	Ref.
NSTAR	Boeing, USA	2567	93	3127	Deep Space 1	1998	[3]
-	–	-	-	-	Dawn	2007	[6]
PPS-1350	Snecma, FR	1200	73	1640	SMART-1	2003	[5]
μ10	JAXA/ISAS, JP	1400	32	3200	Hayabusa	2003	[4]
					Hayabusa-2	2014	[7]
T6	QineticQ, UK	4500	210	4700	Bepicolombo	2018	[8]
NEXT-C	NASA, USA	6900	236	4190	DART	2021	[11]
SPT-140	Fakel, RS	4500	250	1770	Psyche	2022	[12]
HERMeS	NASA, USA	12500	589	2600	PEE	2022	[13]

* Values at 1 Astronomical Unit (AU) distance from the Sun.

EP technologies used in previous deep-space missions have been summarized in Table 1.1. Therefrom, it is revealed how the steadily increasing performances in terms of power, thrust and specific impulse (I_{sp})[2] have been allowing missions of growing complexity and scientific return over time. In the future, even more powerful EP systems combined with gravitational maneuvers are expected to continue enabling new mission concepts. These missions include the NASA's DART (Double Asteroid Redirection Test) mission [11] forecasted for early 2021. It will demonstrate kinetic asteroid deflection by crashing into the smaller body of the binary system Didymos. Also, the NASA mission Psyche [12], slated for launch in 2022, will investigate for the first time a metal asteroid after a Mars flyby. Additionally, the Power and Propulsion Element (PEE) mission was announced for 2022 as the first part of the construction of the Deep Space Gateway.[3] It will be the first possible application of the advanced propulsion system HERMeS (Hall Effect Rocket with Magnetic Shielding) [13], which will demonstrate three times the power of state-of-the-art electric propulsion thrusters.

[2] Specific impulse directly relates to the propellant consumption efficiency.

[3] Retrieved from: `https://spacenews.com/nasa-selects-maxar-to-build-first-gateway-element/`. Accessed: June 2019.



(A) The GEO belt. Credit: Public

(B) Van-Allen Belts. Credit: Public

FIGURE 1.2: Artistic view of Earth's Environment

Within the commercial space industry EP has found widespread acceptance in the geostationary telecommunication market. Although their application had been limited to station-keeping purposes, they are increasingly being used to deliver satellites from geostationary transfer orbit (GTO), where most of them are dropped off by their launcher, to their final position in the Geostationary orbit (GEO). However, the low-thrust levels provided by EP yields to long transfer times and to multiple revolutions around the Earth during orbit raising. Additionally, as long as the injection orbit perigeum is below 10.000 km, each revolution takes the satellite through the lower Van Allen Belts (see Fig.1.2b). This region, populated by high energetic protons, has a degrading effect on spacecraft electronics and solar arrays. On top of that, transfers from GTO may imply a high number of crossings with the GEO belt (see Fig.1.2a). This volume is characterized by a high number of large satellites, 883 according to ESA[4]. Before each crossing, the collision risk must be assessed, which may increase the operational cost of the mission.

Subsequently, new all-electric communication satellites offer a real paradigm shift: at the cost of a longer orbit transfer (i.e., longer time-to-market) and risk, propellant mass is significantly reduced (i.e., lower investment). As an illustration, propellant for orbit transfer with CP and EP can represent up to 40% and 15% of a GEO satellite mass respectively (see Figure 1.3). However, the transfer time is increased from days or weeks to months. Notably, it wasn't until 2015 that the first ever all-electric satellites (Eutelsat 115WB and ABS-3A) were launched. They incorporated the 702SP architecture manufactured by Boeing. Thanks to the great mass savings, they could be launched together for the first time.[5] Two years later, Eutelsat 115WB, based on the Airbus Eurostar E3000EOR platform, reached an operative status. To date, eight all-electric satellite have been launched. Among them, Eutelsat 172 West B used the saved mass to lower the launch costs and reached orbit after four months, whereas SES-14 used it to carry an exceptionally large payload performing a seven months orbit raising.[6]

[4] Retrieved from: https://discosweb.esoc.esa.int/web/guest/statistics. Accessed: June 2019.

[5] Retrieved from: https://spacenews.com/spacex-lofts-pair-of-all-electric-satellites-for-abs-and-eutelsat/. Accessed: June 2019.

[6] Retrieved from: https://spacenews.com/airbus-charges-ahead-with-electric-propulsion/. Accessed June 2019.

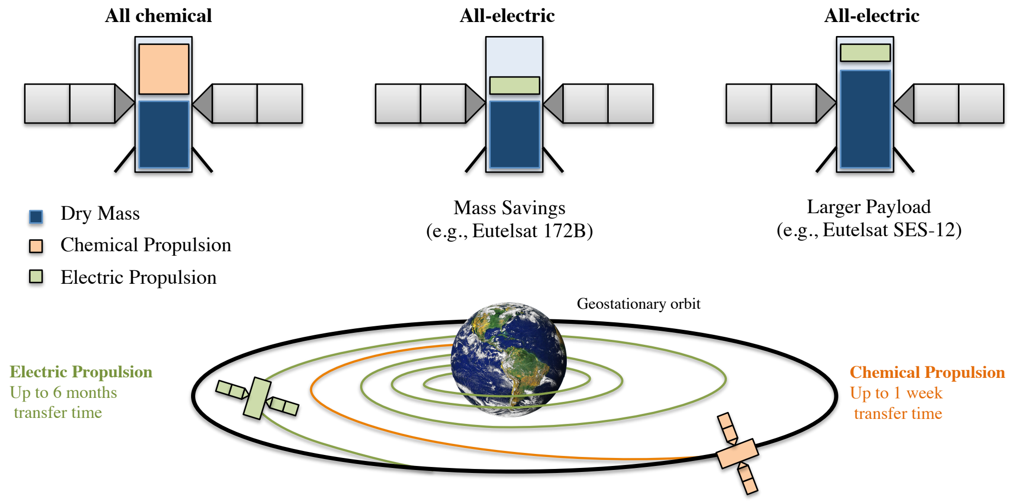


Figure 1.3: Artistic view of all-electric and all-chemical GEO satellite platforms

Table 1.2: Representative Electric Thrusters for all-electric Geostationary satellite platforms

Thruster	Manufacturer	Power (W)*	Thrust (mN)*	$I_{sp}(s)$*	Application	Launch	Ref.
XIPS-25	L-3 ETI, USA	4400	165	3500	Eutelsat 115WB	2015	[14]
-	–	-	-	-	ABS-3A	2015	-
-	-	-	-	-	Eutelsat 117WB	2016	-
-	-	-	-	-	ABS-2A	2016	-
-	-	-	-	-	SES-15	2017	-
SPT-140D	Fakel, RS	4500	250	1770	Eutelsat 172B	2017	[15]
-	-	-	-	-	SES-14	2018	-
-	-	-	-	-	SES-12	2018	-
PPS-5000	Snecma, FR	5000	320	1700	SES-17	2020	[16]
-	-	-	-	-	Hotbird 13F	2021	-
-	-	-	-	-	Hotbird 13F	2021	-
LIPS-300	LIP, CH	5000	170	4100	DFH-4SP	2020	[17]

* Values at 1 AU distance from the Sun.

At this point, all major GEO satellites manufacturers are fabricating all-electric platforms. This sudden success is being boosted by the emergence of new launch services (e.g., SpaceX's Falcon 9) which have become an economically attractive option for light telecommunication satellites. In fact, it is expected that all-electric satellites will represent half of the market by 2020.[7] For instance, Airbus is producing its new platform, Eurostar Neo, under the ESA's NEOSAT project. It is planned to be incorporated in 2021 for Hotbird 13F and Hotbird 13G. Within the same NEOSAT project, Thales Alenia is marketing its all-electric Spacebus Neo platform, which will be implemented in the SES-17 satellite by 2020. In parallel, ESA, SES and OHB have established the electra program [18], a public-private partnership aimed at developing the SmallGeo full-electric satellite platform by 2021. Last but not least, China is constructing its DFH-4SP bus and Lockheed Martin the LM2100 to enter the all-electric satellite market in 2020. A summary of the all-electric satellites already launched and planned for the coming years is shown in Table 1.2. Current available thrusters receive input power less than 5 kW, but increased performance are expected to reduce the transfer time.

[7] Retrieved from: https://spacenews.com/all-electric-satellites-halfway-to-becoming-half-of-all-satellites/. Accessed: June 2019.

In addition to station-keeping and complete orbit raising, EP is being used for orbit topping or partial orbit raising. The operation consists on a hybrid transfer that uses a Combined Chemical-Electric propulsion system (CCEP): a CP subsystem is responsible for initially raising the orbit and a EP subsystem is to conclude the transfer and to inject the satellite into GEO. Customers may be provided with significantly more payload mass than all-chemical satellites, yet maintaining an acceptable on-orbit delivery time and reducing the radiation exposure of all-electric satellites. In 1998, the GSTAR-III telecommunication satellite [19] become the first orbit topping use of EP. Although their electric thrusters were devoted for station-keeping, a failure during the chemical transfer forced operators to use them for finalizing the orbit raising. It was followed in 2001 by ESA's ARTEMIS [20], which used EP for partial orbit raising to compensate for a malfunction in the launch vehicle's upper stage. Thereafter, Lockheed Martin's AEHF SV1 (2010), SV2 (2012) and SV3 (2013) satellites incorporated EP to perform a three months orbit raising. They used the on-board CP to initially boost the perigee over the Van Allen belts [21]. Boeing have also introduced partial electric orbit raising capabilities on their heavy 702HP platform, whose first application was Viasat-2 (2017).[8] Meanwhile, Airbus is studying the hybrid transfer concept on its large E3000SXL and E3000LX platforms.[9]

Satellite-constellations operators at medium and low orbits are also considering EP. Its potential applications include, but are not limited to, the following: orbit raising, constellation deployment, drag compensation for maintaining very low altitudes, controlled re-entry and decommissioning. For instance, ESA may use electric propulsion for the second generation of Galileo satellites for orbit raising and station-keeping maneuvers in order to increase the payload mass. Airbus-OneWeb satellites will be launched at an altitude of 500 km before raising to their operational orbit of 1200 km by their onboard electric propulsion.[10] SpaceX claims that the Starlink constellation will use electric thrusters to adjust position on orbit, to maintain intended altitude, and to de-orbit.[11] Regarding CubeSats, given the small size and volume of these satellites, EP systems may be a suitable option.[12] Therefore, development of a new generation of miniaturized EP platforms is expected, and it is reasonable to assume that this market will rapidly increase. EP is also becoming more attractive for spacecraft precision pointing and positioning, opening up new mission concepts. A dramatic example of a different type of mission enabled by EP is GOCE [22], which was able to make high-accuracy and high-resolution measurements of Earth's gravitational field by flying at 260 km. By using the EP thruster, GOCE was able to counteract the atmospheric drag.

[8] Retrieved from: https://www.viasat.com/news/what-happens-after-viasat-2-launches-step-1-orbit-raising. Accessed: June 2019.

[9] Retrieved from: https://artes.esa.int/projects/e3000-mechanical-platform-electric-orbit-raising. Accessed: June 2019.

[10] Retrieved from: https://www.airbus.com/newsroom/press-releases/en/2016/01/airbus-defence-and-space-and-oneweb-create-oneweb-satellites-company-the-next-stage-of-the-oneweb-adventure.html. Accessed: June 2019.

[11] Retrieved from: https://spacenews.com/spacex-launches-60-starlink-satellites-begins-constellation-buildout/. Accessed: June 2019.

[12] Retrieved from: https://spacenews.com/more-startups-are-pursuing-cubesats-with-electric-thrusters/. Accessed: June 2019.

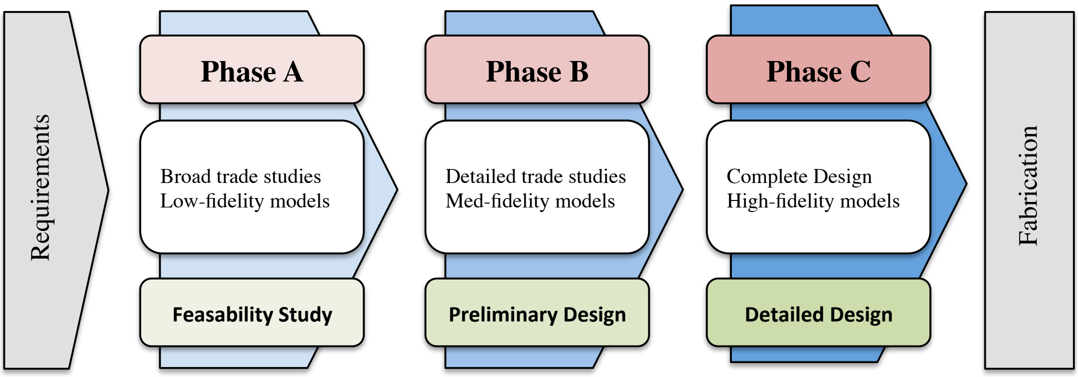


Figure 1.4: Space Mission Design Phases

Last but not least, it is worth mentioning that other highly efficient but less mature propulsion technologies are arising and aiming at competing with or complementing current EP systems. Among them, solar sailing and tethers have been developed. The major advantage of these systems relies on their propellantless nature, i.e., they do no consume fuel at all. Solar sails are large ultra-lightweight reflecting surfaces that utilize solely the freely available solar radiation pressure for propulsion. In 2010, JAXA launched the world's first interplanetary solar sail spacecraft IKAROS (Interplanetary Kite-craft Accelerated by Radiation Of the Sun) to Venus [23]. However, large sail areas are required to propel a conventional spacecraft and a deployment mechanism have to be included. Tethers, which uses the Earth's or other planet's magnetic field and ionosphere to generate a propulsive force, have been considered for de-orbiting or orbit-raising, among other applications [24]. For instance, in 2007, YES2 (Young Engineers' Satellite 2) satellite employed a 32 km tether to de-orbit a small re-entry capsule, "Fotino" [25]. Both solar sails and tethered systems, like EP, produce very low-thrust levels. Therefore they are known under the same name, low-thrust propulsion.

1.1.2 The Role of Space Mission Analysis and Design

The space mission analysis and design activity aims at defining a space system that meets the mission requirements at the minimum over-all cost and risk [26]. The result may include spacecraft's systems and subsystems (e.g., payload, power, structure, attitude, communication, thermal), the mission planning and scheduling (from launch phase to operations, and ultimately spacecraft disposal), the ground segment (e.g., ground stations, control centers) and launch segment (e.g., launch facility, launch vehicle). Previous sections have acknowledged that missions of increasing levels of sophistication and complexity are being proposed for the coming years in order to deliver greater scientific, commercial or social return. Consequently, the analysis and design of space missions under such exacting requirements become a challenging and expensive task in terms of both human and computer-hours. Thus, finding ways to lower these costs while maintaining the quality of the mission is a major goal in the space community. Nowadays, mission analysts and designers necessitate innovative methodologies, alternative approaches, as well as advanced and rapid tools to efficiently assist them during the whole design process.

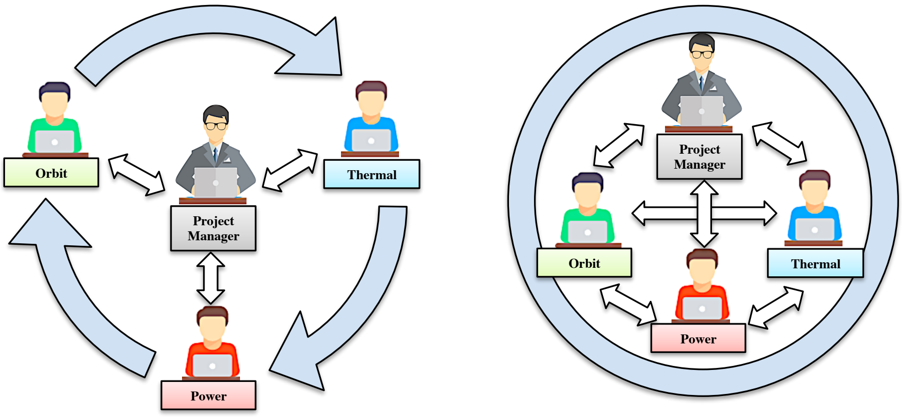


FIGURE 1.5: Traditional Engineering (left) vs Concurrent Engineering (right).

Approach to space mission analysis and design steps through different phases (see Fig. 1.4). The process typically starts with a feasibility or concept study (Phase A). Sometimes, there is also a Pre-Phase A (NASA's nomenclature) or Phase 0 (ESA's terminology) stage. During this early period, scientists and decision makers are interested in high-level trade-off analysis, i.e., exploring as many options as possible and assessing them against multiple, and often conflicting criteria. They are typically conducted on a short duration schedule with limited resources and input information. However, the success of this early phase has been demonstrated to drastically reduce resultant system life-cycle cost (up to 80%) and to increase the chances of a successful final design [27]. Thereafter, the preliminary design (Phase B) is intended to calculate a set of compromise solutions capable of meeting mission needs, which have already been further specified. At this stage, detailed trade studies are performed to validate the design against project goals. During the detailed design phase (Phase C) one or more baseline and back-up solutions are further studied, already including high-fidelity models of the corresponding systems and components. At the end of this phase, the mission project moves into the construction phase.

Conventionally, elements of the space mission architecture are designed consecutively. However, this approach is being complemented and progressively replaced by concurrent engineering practices, especially during Phase A [28]. It involves the multi-disciplinary design of the components collectively and in parallel, and pursues the goal of increasing competitiveness by decreasing lead-time while improving quality and cost (see Fig. 1.5). Nowadays, it is key to the low-cost design of space missions. Therefore, Team-X, formally called the Advanced Products Development Team, was created by the JPL in 1995. It was followed by others such as the Integrated Design Center (IDC) at Goddard Space Flight Center and COMPASS at Glenn Research Center. Similarly, the Concurrent Design Facility (CFD) from ESA, was created in 1999 to rapidly perform feasibility studies for future missions. At CFD, it is claimed that the duration of the Phase A has been shortened from months to weeks.[13] This concept has also been stablished at the German Aerospace Center (DLR) Concurrent Engineering Facility (CEF), at the Satellite Design Office (SDO) of Airbus, and at the PASO office of CNES.

[13] Retrieved from: https://www.esa.int/Our_Activities/Space_Engineering_Technology/20_years_of_ESA_s_Concurrent_Design_Facility_an_oral_history/(print). Accessed: July 2019.

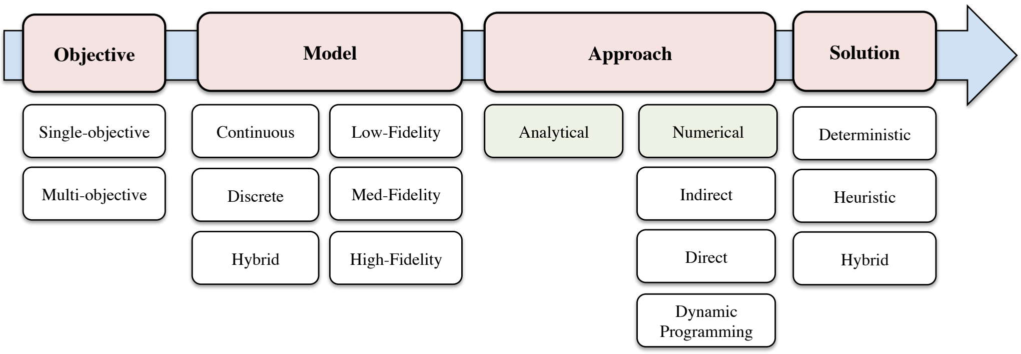


Figure 1.6: Spacecraft Trajectory Optimization Problem Elements

Notably, one of the most demanding design scenarios occurs when the spacecraft is not directly injected by the launcher into the operational orbit. In such case, an on-board propulsion module has to thrust the spacecraft thereto. Mission designers seek to determine the transfer trajectory and the associated maneuver plan that comply with the mission requirements. The selected path dictates the propellant expenditures and the time at which the spacecraft will be operational, thus utterly impacting mission feasibility, cost and return. Therefore, the minimum-cost transfer has to be selected among the feasible set of trajectories. This goal is typically achieved by formulating the trajectory design as an optimization problem, such that a certain objective function or performance index is minimized and mission constraints are met. The general spacecraft trajectory optimization process requires the definition of the appropriate objectives, the mathematical modeling of the dynamics and constraints of the system, the development of a solution approach and the selection of a numerical solving technique when needed (see Fig. 1.6). Although a formal revision of these elements will be provided in Chapter 2, the most relevant findings for this thesis are nonetheless summarized herein.

Firstly, it has been found that spacecraft trajectory optimization problems can be classified regarding the number of optimization criteria as either single or multi-objective. The former produces an optimal point design solution, whereas the latter searches for a whole set of equally optimal solutions with respect to various competing criteria [29]. Secondly, it has been noted that problem modeling accuracy ranges from low-fidelity and constraint-relaxed formulations, to high-fidelity and operationally restricted representations. A different taxonomy can be derived according to the continuous, discrete or hybrid nature of the models. Thirdly, solution approaches essentially fall into two categories: analytical and numerical [30]. Analytical approaches are mainly based on optimal control theory, whereas numerical approaches are broadly classified as direct, indirect or dynamic programming [29]. Basically, each numerical approach converts the trajectory optimization problem into a different mathematical problem: direct approaches into a Nonlinear Programming Problem (NLP), indirect approaches into a Multi-Point Boundary Value Problem (MPBVP) and Dynamic Programming into a partial differential equation (PDE). Finally, solutions techniques for the aforementioned mathematical problems can be regarded as deterministic, heuristic or as a hybrid combination of both.

Table 1.3: Performance Criteria for Trajectory Optimization Methods

Criteria	Description
Speed	Computational time required to achieve a solution
Accuracy	Degree of fidelity to reality
Flexibility	Degree of versatility to cope with a wide range of scenarios
Robustness	Degree of numerical stability or sensitivity to the input parameters
Optimality	Degree of closeness to the optimal solution

Each possible combination of objective functions, models, solution approaches and numerical techniques is characterized by both positive and negative performances, intrinsically limiting their application to either conceptual, preliminary or detailed mission design. In particular, when selecting the most suitable ones there exits an important trade between flexibility, robustness, accuracy, speed and optimality (see Table 1.3). On the one hand, conceptual or preliminary studies prioritize speed and flexibility at the cost of accuracy and optimality. Speed allows to rapidly obtain solutions and make design decisions, while flexibility permits to explore a wide variety of possible scenarios and to contrast them against multiple criteria. During this early design process, where possibilities are broad, any smart automation is highly desirable, such that minimal user input and oversight is required. Moreover, the concept of concurrent engineering demands trajectory designers to include multidisciplinary approaches, aiming at optimizing the trajectory and the spacecraft high-level architecture simultaneously. These features may be accomplished with multi-objective heuristic techniques and low-to-medium fidelity models. However, the lack of accuracy may lead to an erroneous exploration of the design space and to obtain infeasible or unrealistic trajectories. Thus, a careful selection of models is mandatory.

On the other hand, detailed design focuses on accuracy, robustness and optimality at the expense of speed. Accuracy will guarantee that the spacecraft will follow an orbit that is close enough to the computed nominal trajectory. A poor model will lead to unexpected maneuvers to correct the course, reducing the amount of fuel available for nominal operations and, ultimately, prohibiting the success of the mission. Robustness will ensure that small modifications in the final design will not change significantly the nominal solution, whereas optimality warrants that there is no better solution among the feasible options. Additionally, during this phase, complex operational constraints that limits the thrust maneuverability or impose pointing restrictions need to be included in the optimization problem. Constraints will increase the cost of the mission, but they are required to obtain a flyable trajectory. Previous requirements may be accomplished with single-objective deterministic techniques and high-fidelity models. Commonly, obtaining high-fidelity solutions demands mission designers to provide an initial estimate of the trajectory that is taken from the preliminary low-fidelity solutions. The selection of the most adequate initial guess is a crucial task because it directly relates to the success of the method and its computational time.

1.1.3 Research Gaps on Low-Thrust Trajectory Optimization

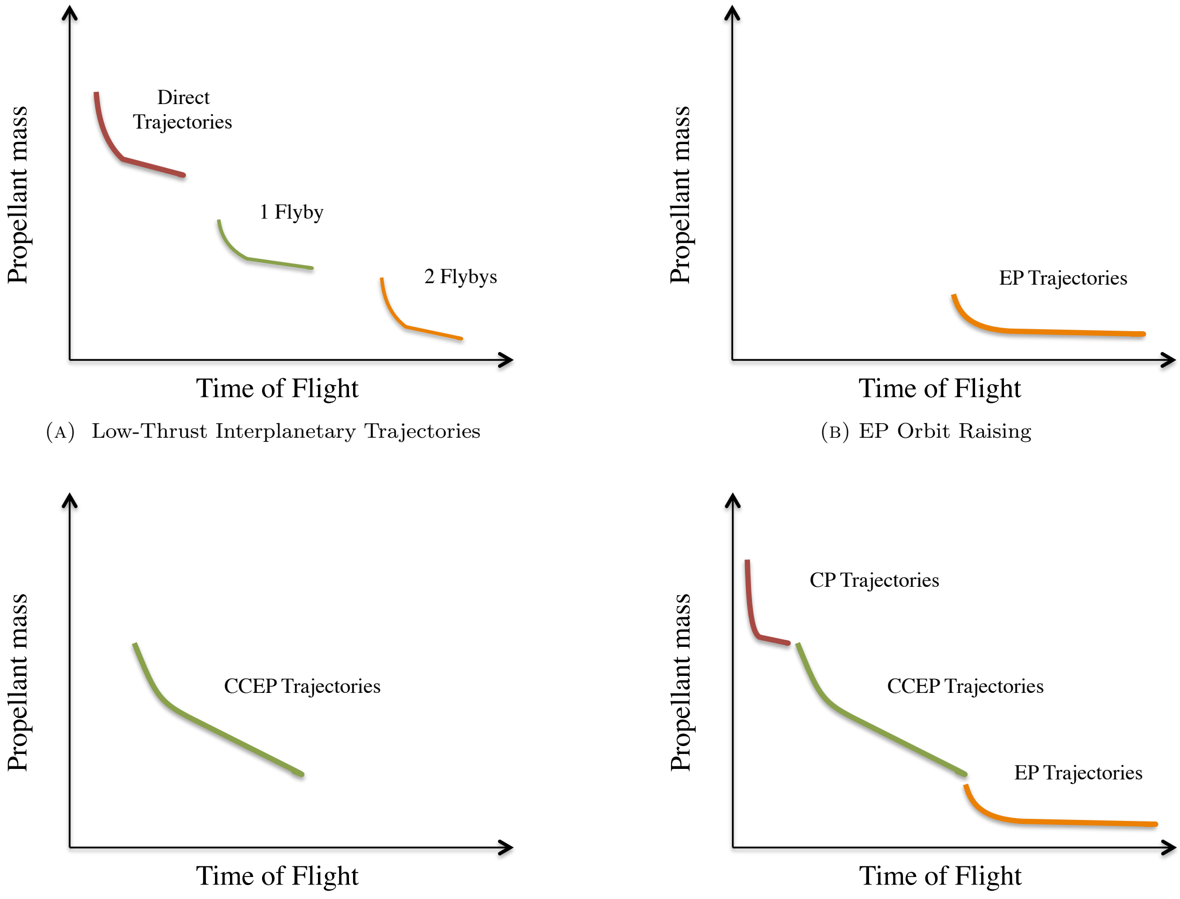
The optimization of trajectories involving CP is a well-known problem and has been profusely studied in the literature; [31–35] provide a partial, but representative list of such prior works. Conversely, the optimization of trajectories involving low-thrust maneuvers are significantly more challenging. Note that the expression“low-thrust” encompasses a broad variety of quite different propulsion concepts, from solar sail to tether techniques. In this thesis, low-thrust propulsion refers to EP only, unless noted otherwise. During the optimization of CP trajectories only a finite and small number of variables have to be considered. Meanwhile, low-thrust optimization requires the determination of a continuous steering law throughout the entire transfer. Besides, the highly non-linear and non-convex dynamics, the space environment perturbations, and the existence of many local minima further complicates the optimization process [36]. Although numerous approaches have been reported to solve low-thrust problems, there is still on-going research. It mainly consists of performance improvements and difficult corner cases. In fact, the 2015 NASA Technology Roadmaps[14] [37] stated the following capability goals:

- Enable design of more ambitious low-thrust missions.
- Increase efficiency of searching broad design space.
- Reduce computational time from days to hours per simulation.
- Enforce new mission critical constraints.
- Optimize alternative objectives.

Previous objectives are specified for low-thrust trajectories that exploit multi-body dynamics or those involving numerous revolutions around a central body. The former group contains the so-called Low-Thrust Multi-Gravity Assisted trajectories (LT-MGA)[15], whereas the latter includes Low-Thrust Orbit Raising transfers (LT-OR). Other difficult scenarios such as low-thrust scape and capture trajectories, and low-energy transfers are attracting research interest, yet they are out of the scope of this thesis. In this work, only the optimization of LT-MGA and LT-OR trajectories will be investigated. These problems can be formulated, in the most general form, as Multi-Objective Hybrid Optimal Control Problems (MO-HOCP). Frameworks for these problems have been proposed by Chilan and Conway [38], and by Ross et al. [39]. In fact, EP systems have a hybrid nature with two distinct discrete working modes (i.e., thrusting and coasting). The hybrid formulation allows to formally include discrete events such as flybys, discrete decision-making processes and mission-planning in the optimization. Although LT-MGA and LT-OR scenarios share the complexity associated to the optimization of low-thrust maneuvers, they exhibit particular challenges deriving from the interplanetary or the planetocentric environment respectively, as well as from problem-specific constraints.

[14] They are a set of documents that consider a wide range of needed technology candidates and development pathways for the 2015-2035 period. In particular, it is mentioned the goals 5.4.2.1 and 5.4.2.7.

[15] In this thesis only high-energy gravity assists will be considered.



(A) Low-Thrust Interplanetary Trajectories

(B) EP Orbit Raising

(C) CCEP Orbit Raising

(D) CP, EP and CCEP Orbit Raising

FIGURE 1.7: Illustration of typical Pareto front solutions

On the one hand, LT-MGA solutions comprise the optimal launch date, the sequence of planetary gravity assists, the flight times between encounters, and the low-thrust steering law. Mission designers are particularly interested in rapidly and automatically obtaining the trade-offs between time of flight and propellant mass as a function of the number gravity assists (see Fig. 1.7a). Traditionally, they have resorted to intuition and experience to develop the optimal flyby sequence. Indeed, much of the work has been done for a user-specified sequence. Examples include the direct methods implemented in the software DITAN [40] and jTOP [41]. They were used for designing Bepicolombo and PROCYON trajectories respectively. However, several efforts have been made to develop rapid tools to automatically determine this sequence without a-priori information. This goal has been accomplished by the tool Mystic [42], which implements a dynamical programming algorithm and was used to design the trajectory for Dawn, and the indirect method developed by J. Olympio [43]. They have proven to be able to automatically determine the optimal sequence of gravity assists exploiting the multi-body dynamics in high-fidelity models. However, these methods demands an initial guess and have difficulties to satisfy operational constraints, which are included as penalties in the objective function [44]. Additionally, computational time is prohibitive for preliminary design [45]. Additionally, computational time is prohibitive for preliminary design [45].

Among the medium-fidelity methods, only GA-EMTG developed by Englander et al. [46] have proven to automatically find gravity assists sequences and to perform a multi-objective search. This method implements a hybrid heuristic-deterministic technique and does not require initial guess. However, computational times range from several hours to days [46]. Therefore, faster assessments at the cost of fidelity and optimality are desirable. For such purpose, shape based-methods have been widely used. Several examples exist in the literature which have proven to be useful for the preliminary design os interplanetary missions (e.g., [47–53]). However, they have been applied to problems without intermediate flybys (e.g., [51–53]), to predefined gravity-assisted maneuvers (e.g., [47, 49]) or to problems where the objective is to minimize the fuel consumption as single-objective (e.g., [48–50]). Thus, the multi-objective exploration of flybys sequence with shape-based methods have not been done yet, as far as the author knowledge. Additionally, previous methods only consider predefined thrust-coast sequences, which may not properly approximate trajectories with multiple revolutions around the Sun.

Important operational constraints such as minimum flyby-distances or launch asymptote bounds have been included in some prior works [40–43, 46]. Establishing a minimum flyby distance will prevent the spacecraft from crashing into the surface, entering the planet's atmosphere or other dangerous regions such as Saturn's rings or the jovian radiation belts. Limiting the declination of the launch asymptote will ensure that the mission can be launched from the selected launch site. Notably, none of the previous approaches have resorted in a major operational constraint. The solutions, during the thrust arcs, are always for the optimal time-histories of the thrust pointing angles. However, missions are not flown with the vehicle continuously changing its attitude. Commonly, the vehicle is re-oriented periodically. For instance, during the interplanetary cruise, Dawn's thrust direction was inertially fixed for a week [54]. Therefore, developing an algorithm to automatically determine the optimal re-orientation times for an interplanetary mission is demanded.

On the other hand, LT-OR are very demanding because they involve a great number of orbital revolutions (e.g. hundreds) due to the strong gravity field in the proximity of the central body. In case the transfer occurs in the Earth's environment, accurate gravitational models including the oblateness effect have to be used. Besides, atmospheric drag have to be considered for low orbits, whereas luni-solar and solar pressure perturbations impacts the dynamics of high orbits. Additionally, when considering solar EP, the period on the shadow of the Earth has a significant effect since no thrust is generated when the spacecraft passes through it. If the spacecraft transverses the Van-Allen radiation belts during the transfer, the radiation damage on the solar arrays should be included as it may reduce the power available, and thus the maximum thrust for maneuvering. Furthermore, the resulting trajectory has to avoid the crossings with the GEO ring due to the likelihood of impact with operational satellites. Other operational constraints such as slew rate limitations based on the Attitude and Control System or phasing to a certain orbital slot may have to be imposed in the solution.

Most authors have focused on solving the complete low-thrust orbit raising problem. In this case, the solution space comprises a set of increasingly fuel efficient solutions at the cost of longer flight times (see Fig. 1.7b). For instance, high-fidelity trajectories have been obatined in the works published by Betts [55] and Schäff [56], and by the tool OPTELEC [57]. However, they are time-consuming for the preliminary design. To alleviate the computational effort and complexity, the control law and/or the trajectory can be predefined. This strategy allowed some authors to obtain analytical solutions (e.g., [58, 59]). Other researchers combined it with averaging techniques (e.g., [60–66]) or asymptotic analysis (e.g., [67, 68]) to estimate the dynamics. However, previous approaches are restricted for certain special cases. A different approach utilizes closed form feedback control laws derived from predetermined Lyapunov functions [69–72]. Several works [73, 74] employed an heuristic algorithm to optimize the parameters of the Q-law Lyapunov function proposed by Petropoulos [70]. This technique was proven to permit a rapid evaluation of the trade-off space and to provide reasonable performance estimates.

Regarding the application of the aforementioned operational constraints, Betts [55] has reported successful results on imposing slew rate restrictions. The software tools OPTELEC [57] and LOTTO [75] claim to be able to include slew rate restrictions, slot phasing, GEO ring avoidance and radiation constraints, although they only furnished a brief description of the models employed. Schäff [56] shortly presented two different approaches for avoiding the GEO ring. In the first approach, he formulated the condition as a cost function forbidding the spacecraft to travel through the GEO region. However, this formulation could only be applied to target an orbit 500 km below GEO and a small number of crossings. In the second approach, he considered the introduction of a constraint on the evolution of the apogee altitude in order to avoid the crossings. However, this approach was found to introduce a significant penalization in terms of propellant mass. Therefore, an algorithm able to avoid a high number of crossings while targeting GEO with a small penalty in terms of propellant mass is desirable.

Fewer authors have studied the partial low-thrust orbit-raising followed by a set of chemical firings, i.e., using CCEP systems. Therefore, the whole transfer sequence, including the CP maneuvers, has to be optimized. In this scenario, Pareto front solutions are moved towards more rapid and fuel-demanding trajectories (see Fig. 1.7c). An excellent insight into CCEP transfer problem has been provided by [76–84]. Nevertheless, they have neglected various important characteristics of the problem. For instance, many of them did not account for major space environmental effects such as [76–78, 83], who ignored the Earth-shadow eclipse effect during the EP transfer. References [76, 78–81, 83] did not quantify the solar-cell degradation caused by transit through the Van Allen radiation belts. Other studies were not flexible enough as they assumed a fixed or constrained starting orbit for the beginning of the EP phase [76, 77, 80–82] or imposed a constant spacecraft attitude [76, 78]. Last but not least, [82–84] performed an offline optimization procedure (i.e., curve fits of stored optimal solutions) to estimate the performances. Consequently, a methodology to optimize general CCEP orbit-raising transfers, including space environmental effects and radiation damaged is required.

Notably, none of the previous approaches for LT-OR trajectories have considered the simultaneous optimization of fully electric transfer and combined chemical-electric transfers. For completeness, the inclusion of fully chemical orbit raising transfers would be desirable. The problem of selecting the optimal EP and/or CP technologies among a user provided list of available options have been also unaddressed. The rapid solution to those problems within a unique algorithm would provide mission designers with a highly flexible software tool that will automatically and in just one run search over a large design space of orbit raising trajectories with respect to time of flight and propellant mass (see Fig. 1.7d). Additionally, incorporating solar-array radiation damage as an alternative optimization objective, would enable a deeper insight of the trades between the optimal solutions. Such results would make the design-making process easier and faster, and would be very suitable for concurrent engineering teams.

In summary, improved and more efficient low-thrust algorithms and software are required to significantly enhance the capabilities to design more ambitious and cost-effective missions. As a rule, it can stated that better tools lead to better mission. However, despite decades of heritage, the topic of low-thrust spacecraft trajectory optimization remains an active field of research and development, with countless approaches available yet much room for improvement. Notably, more efficient algorithms to rapidly perform multi-objective optimization of transfer trajectories while including design decision-making and complex operational constraints are demanded. Hence, the motivation of the research described in this thesis is to develop a unified optimization framework for LT-MGA and LT-OR trajectories where a variety of existing and refined optimization methods will provide new and improved capabilities in terms of speed and flexibility to be used depending on the specific requirements and difficulties of each mission.

1.2 Objectives of the Thesis

In light of the gaps of traditional and state-of-the-art optimization methods, this dissertation aims at developing solution approaches that improve the efficiency of searching wide design spaces, reduce the amount of necessary human involvement, and enhance the capabilities to include complex operational constraints for LT-MGA and LT-OR trajectories. To reach this main goal the following intermediate objectives must be achieved:

O.1 Formal statement of the problem within a rigorous mathematical framework.

O.2 Definition of the dynamical and constraints models.

O.3 Design of a solution approach to perform the multi-objective search.

O.4 Implementation of the proposed approach into an optimization software tool.

O.5 Validation of the implemented algorithm against benchmark scenarios.

O.6 Performance comparisons against state-of-the-art approaches.

1.3 Contributions

The following summary lists the original contributions of this dissertation:

C.1 A general mathematical framework for modeling the multi-objective optimization of constrained spacecraft transfer trajectories is introduced. The framework is based on hybrid optimal control theory and is capable of reflecting continuous and discrete dynamics, along with discrete events and decision-making processes.

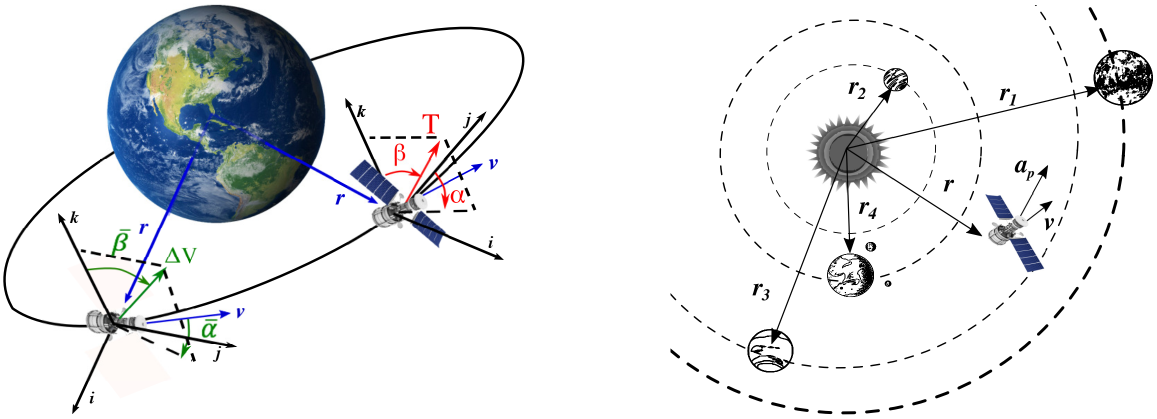
C.2 A numerical solution approach able to concurrently optimize fully electric, fully chemical and combined chemical-electric orbit raising transfers within the Earth environment is provided. The selection of the optimal propulsion technology is also a decision variable. It is based on a two-step sequential algorithm. The first step is able to rapidly perform multi-objective trade studies and to provide sub-optimal propulsive maneuver plans using low-fidelity models and a heuristic technique. The algorithm is flexible, since it does not impose any restriction on the initial or final orbits, neither on the chemical firings. The second step is able to obtain medium-fidelity single-point design solutions, where new models for avoiding the GEO ring in combination to slew rate limits and slot phasing constraints are included within a direct transcription scheme. Results show that the operational constraints can be enforced with minimal penalties in terms of propellant mass. Computational time ranges from minutes to hours for typical LEO to GEO or GTO to GEO transfers. This contribution lead to the development of a journal article that is under review process.

C.3 A numerical solution approach able to automatically optimize the sequence of gravity assists and the steering law of the electric engine for interplanetary transfers is provided. It is based on a two-step sequential algorithm. The first step combines an outer loop that provides a multi-objective optimization via a heuristic algorithm with an inner loop that supplies deterministic optimization of a planar shape-based parametrization of the trajectory. The algorithm is flexible, since it is not constrained to a fixed number or sequence of gravity assists. However, its application is limited to targets with a moderate inclination. The second step is able to obtain medium-fidelity single-point design solutions, where a new model for automatically optimizing the reorientation schedule is included within a direct transcription scheme. Moreover, the analysis of the maximum allowable orientation error with respect to the optimal maneuver plan that fulfills the mission requirements is provided. Results show that the operational constraints can be enforced with minimal penalties in terms of propellant mass. Computational time ranges from minutes to hours for typical transfers involving three or less flybys. This contribution lead to the publication of the journal article of Ref. [85]

1.4 Problem Statement

For the sake of clarity, the problem under consideration in this work is summarized herein. In particular, two different scenarios are considered:

- **Interplanetary Transfers**: A spacecraft is to travel from a given departure planet to rendezvous or flyby a target body within the solar system. The spacecraft may benefit from gravity assisted maneuvers of other planets, as well as from the continuous thrust provided by an electric engine (see Fig, 1.8b). The problem is to find the set of optimal Pareto solutions in terms of flight time and propellant mass consumption. The solution has to comprise the optimal launch and arrival date, the sequence and configuration of the planetary encounters, as well as the time-history of the thrust magnitude and direction. The electric engine can be switched off due to propellant saving reasons or due to available power constraints. Additionally the initial mass of the spacecraft may be free and subject to optimization. Operational constraints to comply with launch asymptote declination constraints and fixed reorientation times during the transfer may apply.
- **Orbit-Raising Transfers**. A spacecraft is to travel from a given departure orbit to a target orbit within the Earth's environment. The spacecraft may be equipped with on-board chemical propulsion, electric propulsion or combined chemical-electric propulsion (see Fig, 1.8a). The problem is to find the set of optimal Pareto solutions in terms of flight time, propellant mass consumption and radiation damage during the transit through the Van-Allen radiation Belts. The solution has to comprise the optimal propulsion system, the number, magnitude, direction and location of the chemical firings, as well the time-history of the thrust magnitude and direction of the electric engine. The electric engine can be switched off due to propellant saving reasons or due eclipse conditions. Operational constraints to arrive at a certain slot in the target orbit, the avoidance of the geostationary ring during the transfer or slew rate limitations may apply.



(A) Orbit-Raising Problem

(B) Interplanetary Transfer Problem

FIGURE 1.8: Illustration of the problem statement

1.5 Proposed Solution Approach

In order to accomplish the goals, the aforementioned problem is formulated as a Multi-Objective Hybrid Optimal Control Problem (MO-HOCP) and a sequential two-step algorithm is proposed to solve it. The algorithm is termed MOLTO (Multi-Objective Low-thrust trajectory optimizer) and its architecture is schematically depicted in Fig.1.9.

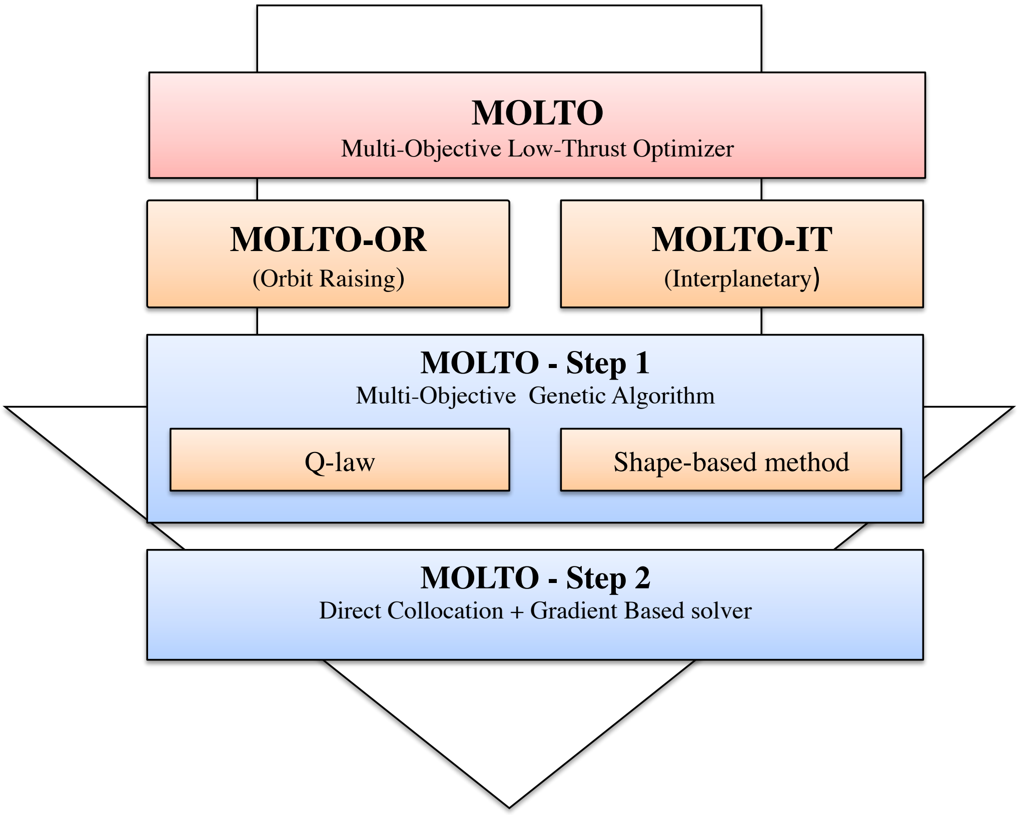

MOLTO
Multi-Objective Low-Thrust Optimizer
MOLTO-OR
(Orbit Raising)
MOLTO-IT
(Interplanetary)
MOLTO - Step 1
Multi-Objective Genetic Algorithm
Q-law
Shape-based method
MOLTO - Step 2
Direct Collocation + Gradient Based solver

Figure 1.9: MOLTO: General Overview

In the first step, (i.e., MOLTO Step 1) low-fidelity models and/or heuristic control laws are applied to greatly simplify the dimension and complexity of the problem. In particular, the MO-HOCP is converted into a Multi-Objective Mixed-Integer Nonlinear Programming Problem (MO-MINLP) that is then solved with a Multi-objective Genetic algorithm (MO-GA) [86]. This step is intended to quickly find near-optimal estimations of the main design parameters and mission performances. The solution obtained therefrom can be used as an initial guess for the second step (i.e., MOLTO Step 2). In the second step, the fidelity of the models is improved in order to improve the accuracy and optimality of the solutions. In this case, the MO-HOCP is converted into a Non-linear Optimization Problem (NLP) via Hermite-Simpson direct transcription scheme that is solved with a Gradient Based Solver. In this stage, the operational constraints can be easily included. Further details of the algorithm for each scenario (i.e., Interplanetary and Orbit-Raising transfers) is provided in the following lines.

For orbit raising transfers:

- **MOLTO-OR Step 1**: The thrust magnitude, direction and on/off switchings of the electric engine are parametrized using a feedback control law, the Q-law developed by Petropoulos [70] and the chemical impulses are modeled as instantaneous velocity changes. The Van allen radiation belt are approximated via an analytical model [87]. The Q-law parameters, and the location, direction and magnitude of the chemical firings represents the real variables to be determined, whereas the number of chemical maneuvers and the propulsion system decision-making process is modeled as integer variables. No operational constraints are applied at this stage. Dynamics are modeled as a two-body problem and including Earth-shadow eclipses and Earth-oblateness perturbations. They are explicitly simulated by a time-marching method

- **MOLTO-OR Step 2**: The propulsive system and the number of CP firings are known and provided by the Step 1. The state of the spacecraft is discretized over a selected grid. The eclipse regions during the transfer are approximated by a smoothed binary-function and the on/off switchings of the electric engine during sunlight are modeled as a relaxed-binary control. The GEO ring avoidance condition is modeled as a set of interior point constraints, slew rate limits as path constraints, and slot phasing as a terminal constraint. This strategy allow to include them in the NLP as a set of non-linear constraints.

For interplanetary transfers:

- **MOLTO-IT Step 1**: A two-dimensional patched-conic model of the trajectory is assumed, in particular the shape based method proposed by Roa et al. [88] to represent a trajectory leg (i.e. from one planet to the next one). The engine on/off switchings are determined by a predefined user-defined sequence. The planetary flyby sequence and the number of revolutions around the central body during each leg are modeled as set of integer variables. In this case, the MO-GA is combined with a sequence of inner-loop NLPs to solve for the shape-based parameters, the gravity assisted altitude and orientation, and to impose the planetary encounters constraints. The MO-GA is in charge of providing the initial guess for the inner-loop NLP solver. No operational constraints are applied at this stage. Dynamics are implicitly simulated by the analytical solution.

- **MOLTO-IT Step 2**: It is assumed that the sequence of planetary gravity assisted maneuvers is known and provided by the Step 1. The state of the spacecraft is discretized over a selected grid and the dynamics are three-dimensional. The on/off switchings of the electric engine are modeled as a relaxed-binary control. Flybys are enforced as interior point constraints, launch asymptote limits are defined as an initial constraint and the rendezvous condition as a terminal constraint. The fixed reorientation time constraints are included in the NLP problem via a embedding technique.

1.6 Outline of the Document

The remainder of this dissertation is structured as follows:

- Chapter 2 presents the general formulation of a trajectory optimization problem and discusses the different elements involved in the formulation and solution of such problem. Thereafter, a critical and exhaustive review of the existing methodologies and tools for optimizing low-thrust transfers is provided.
- Chapter 3 is dedicated to present the mathematical framework of this thesis. After introducing the mathematical formalism to describe Hybrid Dynamical Systems, the general Multi-Objective Hybrid Optimal Control problem is formulated.
- Chapter 4 covers the modeling of the dynamics and constraints employed in MOLTO-OR Step 1 and MOLTO-OR Step 2. Thereafter, the two steps of the solution approach implemented in MOLTO-OR are described. Finally, numerical examples and analysis are included for typical LEO to GEO and GTO to GEO transfers.
- Chapter 5 details the modeling of the dynamics and constraints employed in MOLTO-IT Step 1 and MOLTO-IT Step 2. Thereafter, the two steps of the solution approach implemented in MOLTO-IT is described. Finally, numerical examples and analysis are included for a rendezvous mission to Ceres, a Jupiter flyby mission, and a rendezvous mission to Pluto.
- Chapter 6 summarizes the main conclusions drawn in this dissertation, high-lights the capabilities of the implemented algorithm, indicates its main limitations, and outlines recommendations for future lines of research.

2

State of the Art: Low-Thrust Trajectory Optimization

Contents

2.1 Introduction

This chapter is intended to introduce the general problem of low-thrust trajectory optimization, to categorize the solution approaches, and to survey the state-of-the art methodologies and tools, so that the work performed in this thesis will be in context. For such purpose, the entire process of solving a spacecraft trajectory optimization problem is first divided into four steps. They include mathematical modeling of the spacecraft dynamics, defining the number and type of mission objectives, developing an appropriate approach, and lastly, achieving the solution. Additionally, the main differences between Continuous Optimal Control Problems and Hybrid Optimal Control problems will be highlighted. Traces of these key elements can be found in the survey works presented by Betts [89], Conway [90], Rao [91], and Shirazi et al. [29]. Thereafter, the available solution methodologies will be reviewed. However, the subject is a vast one with a large literature, and the research herein presented will be unapologetically incomplete.

2.2 Optimal Control Problem Statement

Let the discussion begin by mathematically formulating the trajectory optimization problem as a classical Optimal Control Problem (OCP). Formally, it can be posed as the problem of determining the state/trajectory, $\boldsymbol{x}(t) \in \mathcal{X} \subset \mathbb{R}^{n_x}$, which belongs to the set $\mathcal{X}$ of permissible states, the control $\boldsymbol{u}(t) \in \mathcal{U} \subset \mathbb{R}^{n_u}$, which belong to the set $\mathcal{U}$ of feasible controls, the initial time $t_0 \in \mathbb{R}$, and the terminal time $t_f \in \mathbb{R}$, where $t \in [t_0, t_f]$ is the independent variable, that optimizes (i.e., minimizes or maximizes) the following performance index:

$$J(\boldsymbol{x}, \boldsymbol{u}) = \mathcal{M}[\boldsymbol{x}(t_0), t_0, \boldsymbol{x}(t_f), t_f] + \int\limits_{t_0}^{t_f} \mathcal{L}[\boldsymbol{x}(t), \boldsymbol{u}(t), t] dt \tag{2.1}$$

subject to:[1]

$$\dot{\boldsymbol{x}}(t) = \mathrm{f}[\boldsymbol{x}(t), \boldsymbol{u}(t), t], \quad \forall t \in [t_0, t_f] \tag{2.2}$$

$$\mathrm{h}[\boldsymbol{x}(t_0), t_0, \boldsymbol{x}(t_f), t_f] \leq 0 \tag{2.3}$$

$$\mathrm{g}[\boldsymbol{x}(t), \boldsymbol{u}(t), t] \leq 0, \quad \forall t \in [t_0, t_f] \tag{2.4}$$

$$\boldsymbol{x}(t) \in \mathcal{X} \subset \mathbb{R}^{n_x}, \quad \boldsymbol{u}(t) \in \mathcal{U} \subset \mathbb{R}^{n_u}, \quad \forall t \in [t_0, t_f] \tag{2.5}$$

In the above, $\mathrm{f} : \mathcal{X} \times \mathcal{U} \times \mathbb{R} \longrightarrow \mathbb{R}^{n_x}$ represents the set of differential equations of motion in a explicit form. The minimization of the objective function in Eq.(2.1) is subject to initial and terminal conditions on the state vector (Eq.(2.3)), given by the function $\mathrm{h} : \mathcal{X} \times \mathbb{R} \longrightarrow \mathbb{R}^{n_\mathrm{h}}$, admissible values for the continuous and discrete control and state variables (Eq.(2.5)), and further inequality path constraints (Eq.(2.4)), given by the function $\mathrm{g} : \mathcal{X} \times \mathcal{U} \times \mathbb{R} \longrightarrow \mathbb{R}^{n_\mathrm{c}}$. Note that, maximizing or minimizing a cost criterion is equivalent since both approaches can be converted into each other by inverting the sign of the cost criterion [91].

[1] Explicit time-dependency of state and control variables will be usually omitted in this work for purpose of better readability of the text. Hence it holds that $\boldsymbol{x} = \boldsymbol{x}(t)$, $\boldsymbol{u} = \boldsymbol{u}(t)$.

2.2.1 Dynamical Modeling

The first step to solve any spacecraft trajectory optimization problem involves a firm understanding of the dynamics inherent to the system. It refers to the mathematical modeling of the problem, which involves choosing a set of variables to represent the dynamical state of the system, deriving the set of dynamical differential equations to describe the evolution or time history of the state, and selecting the control variables, which represent the degrees of freedom of the system. In spacecraft trajectory optimization problems, the state of the vehicle is also referred as the trajectory (i.e., its position in space with respect to time), while the set of differential equations are also known as Equations of Motion (EOM). Finally, the set of control variables is regarded as the steering strategy or maneuver plan of the propulsive system.

2.2.1.1 State Representation

The spacecraft is typically considered to be a point-mass. Thus, six independent parameters or generalized coordinates are necessary to describe its three-dimensional motion. In practice, there are several forms of representing the spacecraft state, each of them having positive and negative aspects [92]. They can be classified as sets based on position and velocity (e.g., cartesian or polar coordinates), and based on orbital elements (e.g., classical or equinoctial). An overview of the most prominent ones is presented hereafter:

- **Cartesian State Vector (CSV)**: The most common model for describing a spacecraft trajectory refers to its position and velocity vectors. They are typically projected on an inertial cartesian frame, such that $\boldsymbol{x}_{CSV} = [\mathrm{r_x}, \mathrm{r_y}, \mathrm{r_z}, \mathrm{v_x}, \mathrm{v_y}, \mathrm{v_z}]$. Here, $(\mathrm{r_x}, \mathrm{r_y}, \mathrm{r_z})$ are the projections of the position vector, and $(\mathrm{v_x}, \mathrm{v_y}, \mathrm{v_z})$ are the projections of the velocity vector.
- **Polar State Vector (PSV)**: They are mainly used for two-dimensional or planar representations of the problem dynamics. They consists on the following set: $\boldsymbol{x}_{PSV} = (\mathrm{r}, \theta, \mathrm{v}, \psi)$, where r is the distance to the central body, θ is the polar angle, v is the modulus of the velocity with respect to an inertial frame, and ψ is the flight path angle.
- **Classical Orbital Elements (COE)**: Another form of mathematical model to represent the spacecraft dynamics is in terms of classical orbit elements $\boldsymbol{x}_{COE} = (a, e, i, \Omega, \omega, M)$. They are named as the semi-major axis, eccentricity, inclination, right-ascension of the ascending node, argument of perigee and mean anomaly respectively. Instead of the true anomaly, the mean motion, the true anomaly or the eccentric anomaly can be used [93].
- **Modified Equinoctial Elements (MEE)**: The other model for completely defining the state of the spacecraft is by the use of the set of modified equinoctial orbital elements $\boldsymbol{x}_{MEE} = (p, f, g, h, k, L)$. Here, p is the semi-latus rectum of the orbit, and L is named the true longitude. The elements (f, g) are related to the projection of the eccentricity vector on the inertial frame, while (h, k) are associated to the inclination of the orbit.

Table 2.1: Comparison of state vector models for low-thrust thrust transfers

Criteria	CSV	PSV	COE	MEE
Element set	$\mathrm{r_x, r_y, r_z, v_x, v_y, v_z}$	$\mathrm{r}, \theta, \mathrm{v}, \psi$	$a, e, i, \Omega, \omega, M$	p, f, g, h, k, L
Having physical meaning	High	High	High	Low
Suffering form singularities	No	No	Yes	No
Practical for numerical averaging	No	No	Yes	Yes
Practical for analytical solutions	No	Yes	Yes	Yes
Numerical stability	Low	Low	High	High

The CSV representation is widely used for low-thrust interplanetary trajectories. They allow to naturally impose the restrictions associated to flyby or rendezvous a planet, as well as to easily formulate the problem including multi-body gravitational attractions. Additionally, the resulting cartesian EOM are singularity free. However, in planetocentric environments, where multi-revolution occurs, strong oscillations of the cartesian state variables occurs, which increases to numerical instability. Thus, more efficient state representations are required to reduce the computational cost for this transfers. The PSV formulation is simple but limited to planar transfers. This fact may not be a problem during the preliminary design on interplanetary transfers, since most planets almost lie in the same orbital plane. The use of PSV allowed many authors to obtain analytical solutions of the EOM for certain cases. The extension to the three-dimensional case by including $\mathrm{r_z}$, is known as cylindrical coordinates (PSV3).

Conversely, the COE representation is typically applied in planetocentric environments. They are very intuitive as they are related to the physical geometry of the trajectory. The solution of the two-body problem can be stated in terms of the constant COE. For low-thrust trajectories this formulation is appealing because the solution can be described in terms of "almost constant" orbital elements. This fact have allowed many authors to obtain analytical or semi-analytical representations of the trajectory. Unfortunately, they have a number of singularities that may complicate the numerical integration of the EOM. For instance, at zero inclination ($i = 0$) the right ascension of ascending node (Ω) loses meaning. Similarly, for zero eccentricity ($e = 0$) the argument of perigee (ω) becomes undetermined. This is the case of many of the orbits of interest such as GEO. These singularities cause rapid oscillations when the spacecraft is near a singular point and, therefore, lead to difficulties in numerically integrating the EOM [94].

Similarly, the MEE is used for multi-revolution transfer. They are non-singular for all values of eccentricity and inclination. Therefore, they are most used in low-thrust orbit raising transfers to GEO. However, unlike COE, the physical interpretation of the MEE set is not intuitive. Both, COE and MEE allow to easily imposed the constraint of reaching a certain orbit, where the specific location in the orbit is not important. They also permit to fasten the integration of the EOM by applying orbital averaging techniques. When using COE or MEE for optimization of multi-revolution transfer, the angular variable, i.e., true anomaly or true longitude, is often used as independent variable instead of the physical time [55], as this transfers are characterized by very long transfer times. However, neither COE or MEE are well suited when perturbations of the two-body problem are significant, such as transfers to the moon or to libration points.

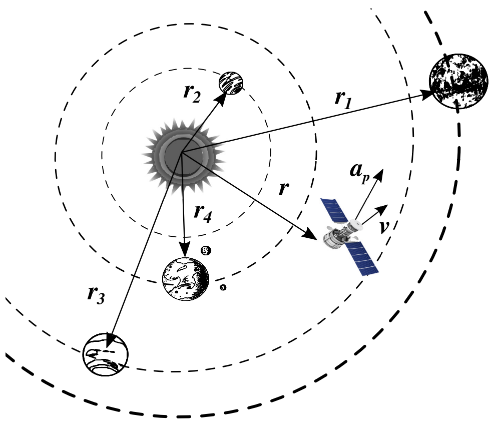


FIGURE 2.1: Perturbed Restricted N-body Problem Illustration.

The presented sets of elements for modeling the spacecraft state are compared in Table 2.1. There is significant freedom in the choice of a suitable set of state variables or orbital elements. Therefore, depending on the specific mission or on the mission designer's experience, one set may be used in favor of others to provide better results. Notably, other forms of state representations than the ones explained herein may be used for spacecraft trajectory optimization. To be more specific, there are twenty two identified candidate orbit element sets plus variations. These other forms of orbital elements are well explained in a survey presented by Hintz [92]. Additionally, the evolution of the spacecraft mass m is typically required to fully describe the dynamics of the system. It is used to compute the acceleration $\mathrm{a_T}$ produced by the spacecraft given the thrust force T produced by the low-thrust propulsion subsystem, and it varies with respect to time as propellant mass is consumed and ejected.

2.2.1.2 Equations of Motion

The EOM are the mathematical description of Eq.(2.2) in the formulation of the OCP. Consider a spacecraft traveling in space under the gravitational attraction of n-bodies in the solar system and subject to the acceleration produced by a low-thrust engine and other space environmental effects (see Fig. 2.1). Its EOM can be generally described as a Perturbed Restricted N-Body Problem (PR-NBP). In case the gravitational bodies are perfectly spherical, the PR-NBP is mathematically expressed in CSV coordinates with respect to an inertial cartesian reference frame $\mathcal{I}$ as follows:

$$\dot{\mathbf{v}} = -\sum_{i=1}^{n} \frac{\mu_i(\mathbf{r}(t) - \mathbf{r}_i(t))}{|\mathbf{r}(t) - \mathbf{r}_i(t)|^3} + \mathbf{a}_T + \mathbf{a}_P, \quad \dot{\mathbf{r}} = \mathbf{v}, \quad \dot{m} = \dot{m}(\boldsymbol{x}, \boldsymbol{u}, t) \tag{2.6}$$

Here, μ_i and $\mathbf{r}_i$ are the gravitational constant and position vector of the i^{th} attracting central mass respectively, whereas $\dot{m}$ is the propellant consumption rate of the propulsion system. Note that, if $n = 2$ or $n = 1$ the formulation is known as the Perturbed-Restricted Three-Body-Problem (PR-3BP) or as the Perturbed-Restricted Two-Body-Problem (PR-TBP) respectively. The perturbing acceleration $\mathbf{a}_P$ represents the summation of any accelerations due to the space environment other than the gravitational attraction (e.g., solar radiation, atmospheric drag).

The engine acceleration $\mathbf{a}_T$ can be expressed as a function of the thrust generated, which generally depends on the spacecraft relative position with respect to the Sun, the total mass, and the control variables as follows:

$$\mathbf{a}_T = \frac{T}{m}\boldsymbol{u} \tag{2.7}$$

In the above, $\boldsymbol{u} = [u_1, u_2, u_3]$ represent the direction cosines of the thrust pointing vector with respect to $\mathcal{I}$. The following path constraint have to be fulfilled $\sqrt{u_1^2 + u_2^2 + u_3^2} = 1$. Alternatively, the thrust azimuth α, and declination β steering angles with respect to $\mathcal{I}$ can be considered as control variables. In such case, the dimension of the control space is reduced from three to two, and the path constraint do not need to be applied. The thrust acceleration vector can be computed as:

$$\mathbf{a}_T = \frac{T}{m}[\cos\alpha\cos\beta, \sin\alpha\cos\beta, \sin\beta] \tag{2.8}$$

A different selection of control variables are possible. For example, the thrust pointing angles with respect to other frame can be chosen. Similarly, the EOM (Eq.(2.6)) can be formulated using other state vector such as PSV, COE or MEE. However, regardless of the selection of the state vector, and assuming a control representation as in Eq.(2.7), a general mathematical expression for the dynamical equations can be derived as a function of $M_F(\boldsymbol{x}, t)$ and $D_F(\boldsymbol{x}, t)$, which are state and time dependent matrices:

$$\dot{\boldsymbol{x}} = M_F(\boldsymbol{x}, t)\boldsymbol{u} + D_F(\boldsymbol{x}, t) \tag{2.9}$$

Notably, computing trajectories under the PR-NBP formulation, yet highly accurate and required for the detailed design, is computationally expensive. Thus, simplified or surrogate models are demanded for the preliminary design. The first step is to reduce the number of attracting bodies up to an acceptable value for the specific scenario. For instance, a low-thrust mission to the Moon requires a PR-3BP formulation. However, PR-TBP dynamics provides suitable results for transfers between Earth-orbits. Notably, for interplanetary transfers, a patched-conic approach is often assumed. This simplification splits the trajectory into a sequence PR-TBP, i.e., the trajectory changes from being heliocentric to planetocentric when the spacecraft enters the sphere of influence of a particular planetary body. An additional approximation assumes that the radius of this sphere is infinitesimal and the flyby occurs instantaneously [43]. As a second step, analytical solutions, averaging techniques or asymptotic analysis can further speed-up the simulation of the dynamics at the cost of fidelity.

- **Analytical solutions**: Analytical techniques were at the very origin of spacecraft trajectory optimization. They seek to obtain closed-forms solutions for the dynamical systems, such that the EOM do not need to be integrated.

  $$\dot{\boldsymbol{x}} = \mathrm{f}(\boldsymbol{x}, \boldsymbol{u}, t) \longrightarrow \boldsymbol{x} = \boldsymbol{x}(\boldsymbol{x}, \boldsymbol{u}, t) \tag{2.10}$$

  These techniques are available in special cases only. Two well-known and widely used analytical solutions correspond to the Kepler and Stark models.

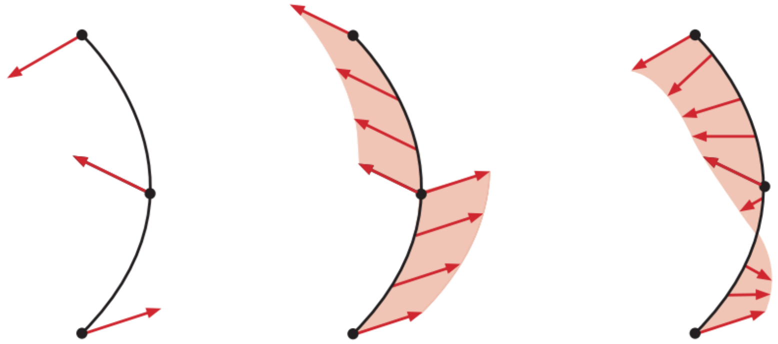


FIGURE 2.2: From left to right: The Kepler model, the Stark model, and the Continuous model

- **Kepler Model (KM)**: It is a reduced model that uses pure Keplerian arcs connected at nodes with impulsive velocity vector discontinuities that approximate the effect of performing a low-thrust maneuver during the Keplerian arc.
- **Stark Model (SM)**: The Stark model yields exact closed-form solutions for a spacecraft in a two-body gravitational field subject to a thrust acceleration that is inertially constant in both magnitude and direction.

Additionally, analytical solutions can be derived under other conditions than the KM or the SM. For instance under constant radial or tangential thrust without space perturbations. Moreover, there are analytical solutions that include some space environmental effects. They will be reviewed through the next section.

- **Asymptotic solutions**: The propulsive acceleration is considered as a perturbation effect acting on a well-known or unperturbed trajectory (e.g., a Keplerian orbit). Thus, the perturbed trajectory can be approximated as a series expansion:

$$\dot{\boldsymbol{x}} = \mathrm{f}(\boldsymbol{x},\boldsymbol{u},t) \longrightarrow \boldsymbol{x}(\epsilon,t) \approx \boldsymbol{x}_0(t) + \epsilon\boldsymbol{x}_1(t) + \mathcal{O}(\epsilon^2) \tag{2.11}$$

where ϵ is an non-dimensional thrust acceleration, and has to fulfill that $\epsilon \ll 1$, $\boldsymbol{x}_0$ is the unperturbed trajectory, and $\boldsymbol{x}_1$ is the first-order perturbation term, which can be obtained analytically under certain circumstances (e.g., constant tangential or radial acceleration). Commonly, second-order terms are not included in the expansion.

- **Averaging techniques**: The method of averaging consists in the elimination of high-frequency components from the EOM by averaging over a short time scale (typically the orbital period). The averaged equations contains only secular and long-periodic terms.

$$\dot{\boldsymbol{x}} = \mathrm{f}(\boldsymbol{x},\boldsymbol{u},t) \longrightarrow \dot{\bar{\boldsymbol{x}}} = \frac{1}{\mathrm{T}}\int_t^{t+\mathrm{T}} \mathrm{f}(\boldsymbol{x}(t),\boldsymbol{u}(t),t)dt \tag{2.12}$$

where $\bar{\boldsymbol{x}}$ is the mean state vector, and T is the orbital period. This is particularly useful in planetocentric scenarios with multiple-revolutions due to the quasi-periodic nature of the orbits. However, averaging results in a loss of exact position information which is desired to assess the power availability to the spacecraft or to rendezvous.

2.2.2 Objective Functions

The objective function, also called value function or performance index, represents the cost of the mission in minimization problems or the benefit in maximization ones. The form defined in Eq.(2.1) is known as the Bolza objective function [91]. The function $\mathcal{M} : \mathcal{X} \times \mathbb{R} \longrightarrow \mathbb{R}^{n_j}$ represents the Mayer term and denotes the cost related to the initial $\boldsymbol{x}(t_0)$ and final $\boldsymbol{x}(t_f)$ states, and the initial t_0 and final t_f times. The integrand function $\mathcal{L} : \mathcal{X} \times \mathcal{U} \times \mathbb{R} \longrightarrow \mathbb{R}^{n_j}$ is called the Lagrange form and denotes the running cost, i.e., it depends on the value of the state $\boldsymbol{x}(t)$ and control $\boldsymbol{u}(t)$ at every time instant t along the trajectory. The objective function may contain just the Lagrange form, just the Mayer term, or both. Notably, as stated by Rao [91], any objective function in Lagrange form can be converted to a Mayer form by adding an extra state, such that $\dot{\boldsymbol{x}}_{n_x+1} = \mathcal{L}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)$. Various forms of objective functions can be categorized with respect to two different aspects: the type and number of objectives.

In most trajectory optimization problems, according to Conway [90], there are two common types of objectives: either some function related to the control effort or to the time required to accomplish the mission. The former typically relates to the spacecraft thrust acceleration level, $J = \int_{t_0}^{t_f} |\mathbf{a}_T| dt$, or to the propellant mass consumed, $J = m(t_f) - m(t_0)$. The latter simply takes the Mayer form $J = t_f$. Alternative objectives, such as launch mass or absorbed radiation, as well as mission-specific criteria may be considered. Besides, in some formulations, the objective function is used to include the infeasibility of the constraints as a penalty term, e.g., $J = J_0 + w \int_{t_0}^{t_f} |\mathrm{g}| dt$. Here, J_0 is a cost function in the form of Eq.(2.1), w is a given penalty weighting parameter, and g is the path constraints function of Eq.(2.4) in the form of equality constraints. However, different solutions would be obtained for different values of the penalty parameter. Regarding the number of objective functions n_k, the OCPs can be classified as either single-objective or multiple-objective.

- **Single-objective**: The goal is to search for a solution in the feasible set that provides the minimum value of a scalar-valued function, i.e., $n_j = 1$. In this case, a single-point solution, under mild regularity assumptions, is obtained. From a mathematical point of view, a feasible solution $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ is optimal if it satisfies the following condition:

$$J(\boldsymbol{x}^*, \boldsymbol{u}^*) \leq J(\boldsymbol{x}, \boldsymbol{u}), \quad \forall \boldsymbol{u} \in \mathcal{U} \quad \text{and} \quad \forall \boldsymbol{x} \in \mathcal{X} \tag{2.13}$$

- **Multi-objective**: The aim is to minimize a vector-valued function formed by $n_k > 1$ conflicting criteria, i.e., $J = [J_1, J_2, \ldots, J_{n_j}]$. The solution in the objective space typically consists of a $(n_j - 1)$-dimensional hypersurface [95] known as the Pareto-optimal set[2] [96]. A feasible solution $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ is weak Pareto-optimal if there does not exit another feasible solution $(\boldsymbol{x}, \boldsymbol{u})$ that could improve all the objectives simultaneously such that:

$$J_i(\boldsymbol{x}, \boldsymbol{u}) \leq J_i(\boldsymbol{x}^*, \boldsymbol{u}^*), \quad \forall i \in \{1, \ldots, n_j\} \quad \forall \boldsymbol{u} \in \mathcal{U} \quad \text{and} \quad \forall \boldsymbol{x} \in \mathcal{X} \tag{2.14}$$

  Otherwise, the point $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ is said to be dominated.

[2] Pareto-optimal set is also known as Pareto front, Pareto frontier, Pareto-efficient set or nondominated front.

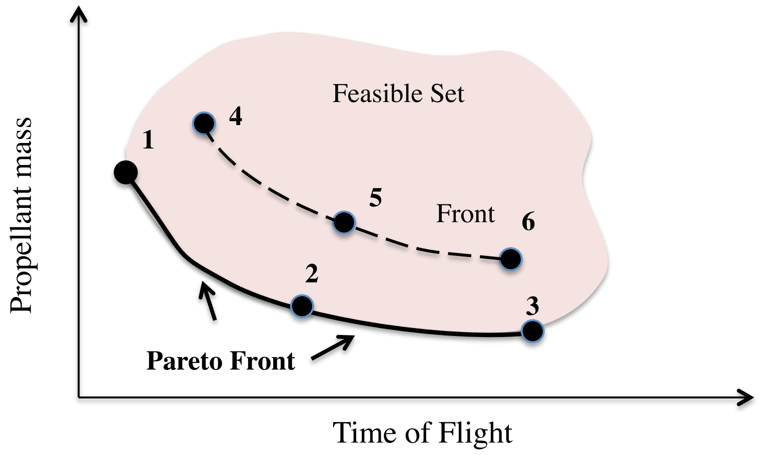


FIGURE 2.3: Illustration of a Pareto front

Solving multi-objective optimization problems, also known as vector optimization or multipurpose optimization, is far more difficult and computationally expensive than solving single-objective problems. However, decision making during the conceptual or preliminary design greatly benefits from the trade-offs provided by multi-objective optimization. Therefore, multi-objective optimization have attracted lots of researches over the last few decades. Solving a multi-objective optimization problem is sometime understood as approximating or computing a representative set of Pareto optimal solutions. One of the most common solution methods for searching for the Pareto front consists on solving a series of single-objective optimization problems. This process is called scalarization. In this approach, a modified version of the vector valued objective function is formulated as a weighted sum of each component $\sum_{i=1}^{n_j} w_i J_i$, which results in a single objective function. Here, w_i are the weighting parameters. They designate the relative importance of each individual cost function J_i. However, this approach does not allow for obtaining non-convex regions of the Pareto front [97].

Most optimization problems appearing in low-thrust trajectory design have multiple objectives that are often equally important and conflicting. As an example, consider the optimization of propellant mass consumed and transfer time-of-flight. The feasible objective space along with six designs is illustrated in Figure 2.3. Because both propellant mass and flight time are minimized, the Pareto front is located in the lower left region of the feasible objective space. Design-1, design-2 and design-3, are along the Pareto front and compose the Pareto-optimal; all other designs are non-optimal. Although design-3 has the lowest propellant mass, design-1 has shorter time of flight; thus they are equally optimal in terms of Pareto. Note that, solution design-1 would have been obtained by a single-objective problem minimizing time-of flight. Similarly, solution design-3 would be the solution of uniquely minimizing propellant mass. Solution design-2 could be obtained by minimizing a scalar combination of time of-light and propellant mass. For further background in the associated multi-objective optimization in engineering applications, the reader should refer to Marler et al. [97].

2.2.3 Continuous and Hybrid Optimal Control Problems

Up to this point, defined state and control variables can be classified as continuous-valued, i.e., they can assume infinite values in a given continuum. However, for certain problems, it is interesting to include discrete-valued or discontinuous states, i.e., they can take values in a finite or countable set. For example, discrete state variables can describe the different working modes of an electric engine (*on* or *off*). When switched-on, the engine operates at maximum thrust, whereas when switched-off the thruster is coasting. Notably, changing the mode of operation implies changing the set of differential equations. Therefore, the time-history of the discrete state is required to determine the trajectory. Similarly, discrete controls may be included in the system to model controller switchings, changing operating modes, or spacecraft design decision-making processes. For instance, selecting a propulsive system among a list of available options can be represented by a discrete control variable. Each alternative provides distinct performances, and consequently a different resulting trajectory.

Similarly, the spacecraft dynamics presented in Eq.(2.6) are continuous, since they are governed by differential equations. However, spacecraft dynamics may include discrete-event dynamics. Discrete events produces instantaneous changes in the spacecraft continuous or discrete state. Performing a gravity assist maneuver or a chemical engine firing before the low-thrust phase are examples of discrete events. Notably, the sequence and number of discrete-events, i.e., the sequence and number of flybys or chemical maneuvers are not know a-priori. The coupling of the discrete events and continuous dynamics, along with the continuous and discrete controls, make hybrid systems the most appropriate theoretical framework to address this issues. The trajectory optimization problem of a hybrid dynamical system is formally tackled as a Hybrid Optimal Control Problem (HOCP).[3] It is a extension of Continuous Optimal Control Problem (COCP), where all the variables involved in the dynamics are continuous. An illustration to compare continuous and hybrid dynamics is depicted in Figure 2.4

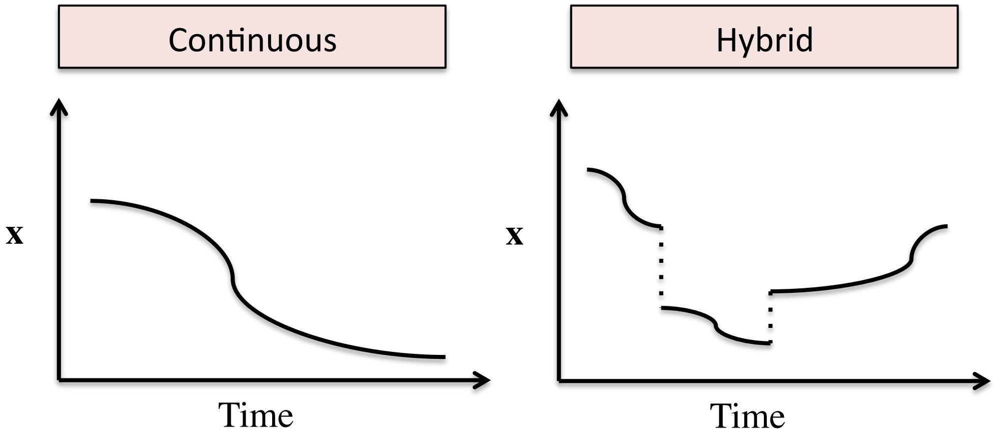


FIGURE 2.4: Continuous versus Hybrid Dynamics

[3] A formal statement of Hybrid Optimal Control Problems will be given in Chapter 3

2.3 Numerical Approaches and Solutions for COCPs

Hitherto, the elements required to properly formulate a spacecraft trajectory optimization problem have been presented, namely objective functions, spacecraft state representation, equations of motion, and control variables. The next step is then to develop a proper approach for finding the optimal solution. Historically, low-thrust trajectory optimization problems have been formulated as purely continuous optimal control problems (COCP). However, well-developed techniques for solving COCP are totally or partially transferred to solve more complex trajectory optimization problems such as Hybrid Optimal Control Problems as it will be detailed in the next section. Therefore, in this section, the different solution methods presented in the literature for solving COCP, will be first characterized. Since this step is a vast subject, only an overview of approaches with a brief discussion is provided herein. For a fundamental background in the associated methodologies, the reader should refer to [89–91].

As a rule, two types of approaches exists: analytical approaches an numerical approaches. Analytical approaches results in analytical solutions for the optimal trajectory. They can only be obtained in special cases. Therefore, they are seldom feasible for the majority of the spacecraft trajectory optimization problems. Most of researchers have been dedicated to numerical methods in order to find the solution to most meaningful trajectory optimization problems. Numerical approaches can be divided in three well-known methods: indirect methods, direct methods and dynamic programming. Indirect methods rely on the Pontryagin minimum principle, dynamic programming on the Hamilton-Jacobi-Bellman theory, and direct methods on the Karush-Kuhn-Tucker optimality conditions. Furthermore, each method result in a different mathematical problem that can be solved with the aid of gradient-based, heuristic or hybrid techniques. Each combination exhibits differentiating positive and negative aspects. Hereafter, an overview of these approaches along with their related techniques will be briefly discussed. The overall schema of numerical approaches is depicted in Fig. 2.5.

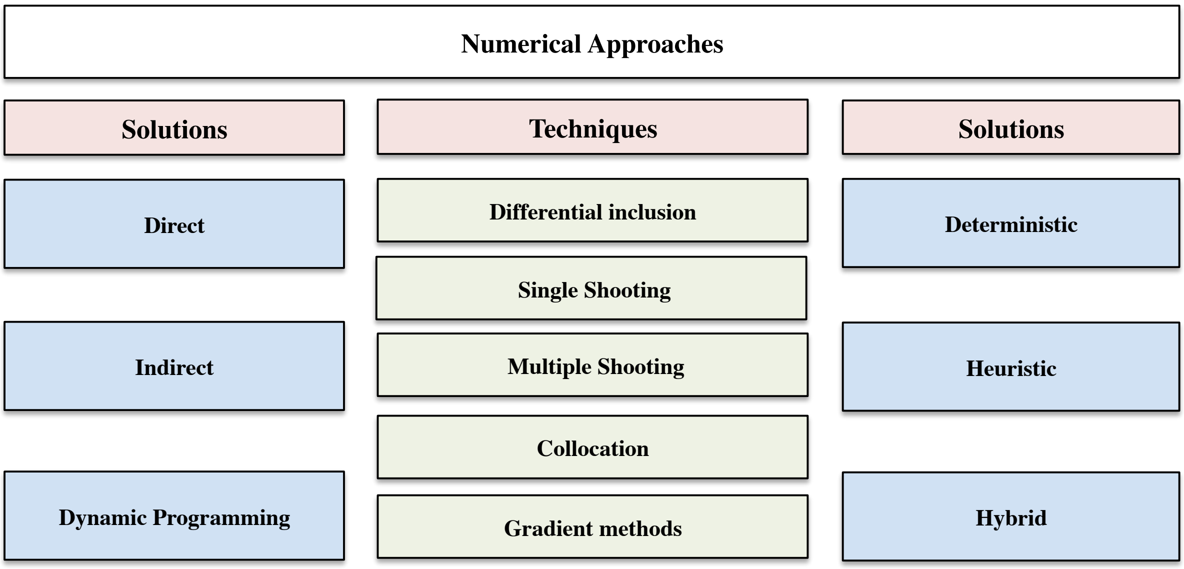


FIGURE 2.5: Numerical approaches, techniques and solutions for COCPs

2.3.1 Indirect Method

In the indirect approach, the goal is to solve the multi-point boundary value problem (MPBVP) that results from applying the Pontryagin Minimum Principle (PMP) and the calculus of variations to the OCP given by Eqs.(2.1)-(2.5). The PMP, which was developed by Pontryagin and his coworkers in 1962 [98], characterizes the first-order necessary conditions that an optimal solution $(\boldsymbol{x}^*, \boldsymbol{u}^*)$, also called extremal, must satisfy. These conditions are typically expressed by defining the Hamiltonian $H : \mathcal{X} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_g} \times \mathcal{U} \times \mathbb{R} \longrightarrow \mathbb{R}$, such that:

$$H(\boldsymbol{x}, \lambda, \mu, \boldsymbol{u}, t) = \mathcal{L} + \lambda^T \mathrm{f} - \kappa^T \mathrm{g} \tag{2.15}$$

where $\lambda(t) \in \mathbb{R}^{n_x}$ denotes the costate or adjoint and $\kappa \in \mathbb{R}^{n_g}$ is the Lagrange multiplier associated with the path constraints (Eq. (2.4)). The PMP is then formally written as:

$$\boldsymbol{u}^* = \arg\min_{u} H, \qquad \boldsymbol{u} \in \mathcal{U} \tag{2.16}$$

It implies that at any instant in time t on the optimal trajectory $\boldsymbol{x}^*(t)$, the optimal control variables $\boldsymbol{u}^*(t)$ are chosen such that the Hamiltonian H is minimized. Additionally, the states and costates must obey the Euler-Lagrange equations:

$$\dot{\boldsymbol{x}} = \frac{\partial H}{\partial \lambda}, \qquad \dot{\lambda} = -\frac{\partial H}{\partial \boldsymbol{x}} \tag{2.17}$$

Furthermore, the initial and final costates must comply with the transversality conditions:

$$\lambda(t_0) = -\frac{\partial \Phi}{\partial \boldsymbol{x}(t_0)} + \nu^T \frac{\partial \mathrm{g}}{\partial \boldsymbol{x}(t_0)}, \qquad \lambda(t_f) = \frac{\partial \Phi}{\partial \boldsymbol{x}(t_f)} - \nu^T \frac{\partial \mathrm{g}}{\partial \boldsymbol{x}(t_f)} \tag{2.18}$$

The Hamiltonian at the initial and final time must verify the complementary conditions:

$$\mathcal{H}(t_0) = \frac{\partial \Phi}{\partial t_0} - \nu^T \frac{\partial \mathrm{g}}{\partial t_0}, \qquad \mathcal{H}(t_f) = -\frac{\partial \Phi}{\partial t_f} - \nu^T \frac{\partial \mathrm{g}}{\partial t_f}. \tag{2.19}$$

In the above, $\nu \in \mathbb{R}^{n_\mathrm{h}}$ represents the Lagrangian multiplier associated with the boundary conditions (Eq.(2.3)). Finally, κ must satisfy the following switching structure:

$$\kappa_j(t) = 0, \quad \text{when} \quad \mathrm{g}_j(\boldsymbol{x}, \boldsymbol{u}, t) < 0, \quad j = 1 \ldots, n_c \tag{2.20}$$

$$\kappa_j(t) \leq 0, \quad \text{when} \quad \mathrm{g}_j(\boldsymbol{x}, \boldsymbol{u}, t) = 0, \quad j = 1 \ldots, n_c \tag{2.21}$$

The Euler-Lagrange equations (Eq.(2.17)) together with the equality boundary conditions (Eq.(2.5)), transversality (Eq.(2.18))and complementary conditions (Eq.(2.19)) define the MPBVP. Note that the control can be obtained as a function of the state and costate at each instant, explicitly or numerically (e.g., Newton's method), applying the PMP (Eq.(2.16)). While some analytical solutions exist for certain special cases [30, 99], an indirect method aims to numerically solve the MPVBP to determine candidate extremal trajectories. The optimal solution is then found by choosing the extremal trajectory with the lowest cost. Typically, one of the following methods are employed: indirect single/multiple shooting, indirect collocation, or gradient-based methods. A brief conceptual depiction of their implementation is provided hereby. For a more comprehensive review, the reader is referred to [91] and to [99].

- **Indirect Single Shooting**: An indirect single shooting method is an iterative technique in which an initial guess is made of the unknown initial states and costates. Using this guess, together with the known initial conditions, the state and costate dynamical system is integrated using a time-marching method. Upon reaching t_f, the terminal conditions obtained from the numerical integration are compared to the known terminal conditions on the states and costates. The unknown initial conditions are the adjusted and the process is repeated until the difference between the known and the computed terminal conditions is less than some specified threshold. The single-shooting scheme describes the problem in terms of a small number of unknowns, but suffers from numerical instabilities.

- **Indirect Multiple Shooting**: In a multiple-shooting method, the time interval $[t_0, t_f]$ is broken up into $N + 1$ subintervals. The shooting method is then applied over each subinterval $[t_i, t_{i+1}]$ with the initial values of the state and adjoint at each subinterval being unknowns that need to be determined. Additionally, continuity conditions have to be enforced at the interface of each subinterval, such that the difference between the final computed values in each subinterval and the initial values of the next subinterval is driven to zero. The parameters are updated until the boundary and intermediate matching conditions are satisfied. This method increases the number of the unknown variables but improves the robustness of the indirect single-shooting.

- **Indirect Collocation**: In an indirect collocation method, the states and costates are discretized over a predefined time-grid, such that the states and costates are known only at discrete time-samples, typically the beginning and end of each time segment into which the total time is subdivided. The state and costate differential equations are transformed into a set of discrete defect constraints, which relates the values at the beginning of the subinterval to the values at the end. The unknown discretized states and costates are modified to drive the defect constraints to zero and to meet the boundary conditions. Different methods can be found regarding the quadrature rule that approximates the differential equations: local and global collocation methods. This approach requires the determination of a high number of variables, but improves the convergence domain.

- **Gradient Methods**: they are based on the special structure of these necessary conditions. They are more intuitive than the previous ones because the optimization variable is the history of control values, thus can be guessed more intuitively [100]. Starting from the initial condition, in indirect gradient method the dynamical system is integrated forward in time until the final time t_f is reached. Then, the adjoint equations are integrated backwards in time until t_0. The backward integration is initialized with the relevant optimality conditions at the final time. To performed the forward integration and the initialization of the adjoint variables, a control function of the time has to be initially guessed. These unknowns are the decision variables, which are iteratively varied until the optimality conditions are satisfied. Depending on the update procedure for the control, gradient methods of first order [100] and second order [101] are distinguished.

2.3.2 Direct Method

The basic idea of direct methods is to transcribe the OCP given by Eqs.(2.1)-(2.5) into a parameter optimization problem or nonlinear programming problem (NLP), where the objective function (Eq.(2.1)) is “directly” optimized. The goal of a general NLP problem is to determine a vector of unknown decision variables $\boldsymbol{z} \in \mathcal{Z} \subset \mathbb{R}^{n_z}$, which belongs to the set $\mathcal{Z}$ of feasible decision variables, such that:

$$\begin{aligned} &\textit{Minimize} && J(\boldsymbol{z}) \\ &\textit{subject to} && \mathrm{c}(\boldsymbol{z}) = 0 \\ & && \mathrm{k}(\boldsymbol{z}) \leq 0 \end{aligned} \tag{2.22}$$

In the above, $J : \mathcal{Z} \longrightarrow \mathbb{R}^{n_j}$ is the cost function, $\mathrm{c} : \mathcal{Z} \longrightarrow \mathbb{R}^{n_g}$ represents the equality constraints, and $\mathrm{k} : \mathcal{Z} \longrightarrow \mathbb{R}^{n_k}$ represents the inequality ones. An optimal solution $\boldsymbol{z}^*$ to the NLP problem has to fulfill first-order necessary optimality conditions. These conditions, also known as Karush-Kuhn-Tucker conditions (KKT) [102, 103], can be formulated as follows:

$$\mathrm{c}_i(\boldsymbol{z}) = 0, \qquad (i = 1, ..., n_g) \tag{2.23}$$

$$\mathrm{k}_i(\boldsymbol{z}) \leq 0, \qquad (i = 1, ..., n_k) \tag{2.24}$$

$$\kappa_i \geq 0, \qquad (i = 1, ..., n_k) \tag{2.25}$$

$$\kappa_i \ \mathrm{k}_i(\boldsymbol{z}) = 0, \qquad (i = 1, ..., n_k) \tag{2.26}$$

$$\nabla_z J(\boldsymbol{z}) + \sum_{i=1}^{n_c} \nu_i \nabla_z \ \mathrm{c}_i(\boldsymbol{z}) + \sum_{i=1}^{n_k} \kappa_i \nabla_z \ \mathrm{k}_i(\boldsymbol{z}) = 0 \tag{2.27}$$

where $\nu \in \mathbb{R}^{\mathrm{g}}$ and $\kappa \in \mathbb{R}^{\mathrm{k}}$ are known as the Lagrange multipliers due to the equality and inequality constraints respectively. Conceptually, transcribing the original OCP into a NLP starts by defining a time-grid, such that $t_0 < t_i < t_N$ and $t_N = t_f$. Thereafter, the control and/or state variables are discretized over the mesh, such that the continuous OCP converts to a discrete OCP:

$$\begin{aligned} &J(\boldsymbol{x}(t), \boldsymbol{u}(t)) && \longrightarrow J(\boldsymbol{x}(t_i), \boldsymbol{u}(t_i), t_i) \\ &\boldsymbol{x}(t)) = \mathrm{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) && \longrightarrow \boldsymbol{x}(t_{i+1})) - \mathrm{f}(\boldsymbol{x}(t_i), \boldsymbol{u}(t_i), t_i) = 0 \\ &\mathrm{g}[\boldsymbol{x}(t_0), t_0, \boldsymbol{x}(t_f), t_f] \leq 0 && \longrightarrow \mathrm{g}[\boldsymbol{x}(t_0), t_0, \boldsymbol{x}(t_N), t_N] \leq 0 \\ &\mathrm{c}[\boldsymbol{x}(t), \boldsymbol{u}(t), t] \leq 0 && \longrightarrow \mathrm{c}[\boldsymbol{x}(t_i), \boldsymbol{u}(t_i), t_i] \leq 0 \end{aligned} \tag{2.28}$$

Finally, the discretized states and controls are treated as NLP variables, i.e., $\boldsymbol{z} = [\boldsymbol{x}(t_i), \boldsymbol{u}(t_i)]$, such that Eq.(2.28) transforms to a NLP problem in the form of Eq.(2.22). The dynamical, path and boundary constraints are converted into a set of NLP equality and inequality constraints. The NLP is then numerically solved using well-known optimization techniques [91]. Note that the dynamical equations are converted to a set of non-linear equality constraints. Notably, there exist several techniques for selecting the discretization grid and for transcribing the differential equations. Among these techniques, it is possible to distinguish differential inclusion, direct single/multiple shooting, and direct collocation. A brief conceptual depiction of their working principles is provided hereby. For a more in depth explanation, the reader is referred to [89].

- **Differential Inclusion**: in differential inclusion only the state variables are discretized over a predefined time-grid. This method enforces the equations of motion at each discrete time by applying inequality constraints on the rates of change of the states. These inequality constraints are obtained by substituting the upper and lower bounds on the control vector into the equations of motion. An implicit integration rule is then used to write the time rates of change as functions of the state variables alone. The discretized states are updated until the inequality constraints are satisfied and the cost function is minimized. The advantage given by differential inclusions is that it effectively eliminates the explicit dependence on control values at each node. However, this technique can become numerically unstable and the formulation can be problem dependent.

- **Direct Single Shooting**: direct shooting methods discretize the control on a specified time-grid, and solve for the discretized controls that optimize the cost function. Like indirect single shooting methods, dynamics are satisfied by integrating the differential equations using a time-marching algorithm from t_0 to t_f. Additionally, the cost is determined using a quadrature approximation that is consistent with the integrator used to solve the differential equations. Upon reaching t_f, the terminal conditions obtained from the numerical integration are compared to the known terminal conditions on the states. The control parameters are then updated to drive the cost and the infeasibility of the terminal constraints to a lower value. This scheme has the ability to describe the problem in terms of a small number of unknowns, but suffers from numerical instabilities.

- **Direct Multiple Shooting**: in a manner similar to that for indirect methods, in a direct multiple-shooting method, the temporal domain $[t_0, t_f]$ is divided into $N + 1$ subintervals. The aforementioned direct shooting method is then used over each segment $[t_i, t_{i+1}]$ with the values of the state at the beginning of each subinterval and the discretized controls being unknowns in the optimization. In multiple shooting, the end of one segment will not necessarily match up with the start of the next. This difference is known as a defect, and it is added to the constraint vector. The optimization parameters are updated until the cost in minimized and the constraints satisfied. This method reduces the high sensitivity of the single shooting algorithm at the cost of increasing the number of unknowns.

- **Direct Collocation**: direct collocation-based methods discretize both the control and state variable time histories, i.e., the states and controls are known only at discrete points, typically the beginning and end of each time segment into which the total time is subdivided. The system-governing equations are transformed into discrete defect constraints, which relate the values at the beginning of the subinterval to the values at the end. Different methods are characterized by the choice of quadrature rule to approximate the differential equations between each two subintervals: local and global collocation methods. The unknown discretized states and controls are updated to reduce the cost criterion and to drive the defect constraints to zero. This approach leads to a large-scale NLP problem, but exhibits superior rates of convergence.

2.3.3 Predefined Control Laws

In previous lines, it has been assumed that the control law $\boldsymbol{u}(t)$ is discretized over a time-grid t_i, such that $\boldsymbol{u}(t_i)$ become optimization parameters of an NLP problem. Due to the fact that low-thrust transfers are typically very long or involve many revolutions around the central body, a high number of discrete samples are required to accurately represent the thrust steering. This may result in a time-consuming numerical approach. In order to speed-up the computational process, it is common to use a predefined or heuristic control laws, such that the thrust direction and magnitude is prescribed as a function of a small set of parameters. Heuristic control laws generally yield suboptimal trajectories, but follow a policy that a mission designer deems acceptable for the preliminary design. In fact, some predefined control laws may allow to obtain an analytical representation of the trajectory. They can be categorized into six main groups, depending on the heuristic function that is used to parametrize the control:

- **Blended Control (BC)**: The optimal steering that maximize the variation (i.e, increase or decrease) of each element of the state vector independently $\boldsymbol{u}_x(\boldsymbol{x}) \in \mathbb{R}^{n_x}$ are computed as a function of the position in the orbit. They are commonly obtained analytically. Then, the complete control law to simultaneously modify all the elements of the state vector results from the following weighted sum:

$$\boldsymbol{u}^*(G_x, t) = \sum G_x(t)\boldsymbol{u}_x(\boldsymbol{x}) \tag{2.29}$$

  where $G_x \in \mathbb{R}^{n_x}$ are time-varying weighting functions that fulfills $\sum G_x(t) = 1$. Their time-discretized values $\boldsymbol{z} = G_x(t_i)$ are the unknowns to be determined. Commonly, BC are derived for MEE or COE formulations. Further, they can incorporate a coasting mechanisms based on the effectivity of the maneuver, i.e., as a function of the instantaneous rate of change of each element and the maximum obtainable. If this efficiency factor is below a threshold, the spacecraft turns to coasting mode, until the efficiency improves.

- **Calculus of Variations based (COV)**: The PMP (Eq.(2.16)) is used to obtain the optimal control history. For simplicity, a minimum-time OCP with the EOM provided by Eq.(2.9 is assumed along with the equality path constraint $\boldsymbol{u}^T\boldsymbol{u} = 1$. Thus, the control law can be explicitly obtained as:

$$\boldsymbol{u}^*(\lambda, t) = -\frac{M_F^T(\boldsymbol{x})\lambda(t)}{||M_F^T(\boldsymbol{x})\lambda(t)||} \tag{2.30}$$

  The costates $\lambda(t)$ are interpolated through an appropriate time-grid, such that the values $\boldsymbol{z} = \lambda(t_i)$ become the optimization parameters. Thus, they are directly optimized by the solver rather than governed by the Euler-Lagrange equations. Additionally, analytical formulations for the transversality conditions do not need to be derived. This is why this method is included in the direct optimization techniques. However, many authors have classified it as a hybrid, i.e., a combination of direct and indirect principles.

- **Lyapunov Control (LC)**: It defines an energy-like (i.e., a positive-definite) scalar Lyapunov function of the state $V(\Delta x(t), \boldsymbol{z}) \in \mathbb{R}$. Here, $\Delta x(t) = \boldsymbol{x}(t) - \boldsymbol{x}_f$, and $\boldsymbol{x}_f$ is the target state. The set of free parameters $\boldsymbol{z} \in \mathbb{R}^{n_z}$ are to be determined as part of the solution of the NLP problem. The Lyapunov function has to fulfill the following condition:

$$\dot{V}(\boldsymbol{z}) = \nabla_x V(\Delta x, \boldsymbol{z}) \cdot \ \mathrm{f}(\boldsymbol{x}, \boldsymbol{u}) \leq 0 \tag{2.31}$$

The thrust steering law is then obtained by minimizing the variation of $\dot{V}$ with respect to the control law (i.e, making it as negative as possible) as follows:

$$\boldsymbol{u}^*(\boldsymbol{z}, t) = \arg\min_u \nabla_x V(\Delta x(t), \boldsymbol{z}) \cdot \ \mathrm{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{2.32}$$

Notably, this control law naturally drives the spacecraft to the desired final state, avoiding the need to include the final boundary conditions in the NLP. Similarly to BC, a coasting mechanism can be incorporated in terms of an efficiency parameter.

- **Shape-based Approaches (SB)**: In this approach, the state vector $\boldsymbol{x}(t)$, usually the trajectory, is assumed to have a predefined form, e.g., $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{z}, t)$, where $\boldsymbol{z} \in \mathbb{R}^{n_z}$ are the unknowns to be determined by the NLP solver. The control law is obtained by forcing the EOM to be satisfied:

$$\boldsymbol{u}^*(\boldsymbol{z}, t) : \dot{\boldsymbol{x}}(\boldsymbol{z}, t) - \ \mathrm{f}(\boldsymbol{x}(\boldsymbol{z}, t), \boldsymbol{u}^*, t) = 0 \tag{2.33}$$

An analytical solution for the control is derived therefrom. Note that the obtained control may not satisfy the constrained related to the maximum thrust available. Thus it may lead to unfeasible trajectories. The solution may not fulfill the boundary constraints, thus they must be included in the formulation of the NLP.

- **Neurocontroller (NC)**: The problem of finding an optimal strategy that leads to an optimal trajectory is thus transformed into the determination of the optimal network transfer function $N : \mathcal{X} \times \mathbb{R}^{n_z} \times \mathbb{R} \longrightarrow \mathcal{U}$. This function acts as a map from the current spacecraft state, the desired final state $\boldsymbol{x}_f$, and the network's internal parameters $\boldsymbol{z} \in \mathbb{R}^{n_z}$ to the instantaneous steering. Thus, it holds that:

$$\boldsymbol{u}^*(\boldsymbol{z}, t) = N(\boldsymbol{z}, \boldsymbol{x}_f, \boldsymbol{x}, t) \tag{2.34}$$

The controller parameters $\boldsymbol{z}$ are to be determined as part of the NLP solution.

- **Finite Fourier Series (FFS)**. The low-thrust steering history is assumed to be represented by a Finite Fourier series expansion, such that:

$$\boldsymbol{u}^*(a_k, b_k, t) = \sum_{k=0} a_k(t) \cos\left(\frac{2\pi k\theta}{\Delta\theta}\right) + b_k(t) \cos\left(\frac{2\pi k\theta}{\Delta\theta}\right) \tag{2.35}$$

where the time-varying coefficients a_k and b_k are time-discretized, such that $\boldsymbol{z} = [a_k(t_i), b_k(t_i)]$ are optimized by the NLP solver. The angle θ represent any orbit anomaly, and $\Delta\theta$ represents the with of the interval in which the Fourier expansion applies. Note that, increasing the number of coefficients will improve the accuracy of the representation at the cost of increasing the number of unknowns and the complexity.

2.3.4 Dynamic Programming

The method of Dynamic Programming is based on the Bellman's principle of optimality [104]: "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision." Even though Dynamic Programming was originally developed for discrete-time systems, it was extended to continuous-time problems. The continuous-time equivalent of the Bellman's principle resulted in the Hamilton-Jacobi-Bellman (HJB) theorem [105]. The HJB theorem describes the sufficient conditions for the trajectory $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ to be an optimum. These conditions are derived for an unconstrained problem version of the OCP given by Eqs.(2.1)-(2.5), where the objective function depends only on the state at the final time. Let us defined the value function $V : \mathbb{R}^{n_x} \times \mathbb{R} \rightarrow \mathbb{R}$ as follows:

$$V(\boldsymbol{x}, t) = \min \left\{ \Phi[\boldsymbol{x}(t), t, \boldsymbol{x}(t_f), t_f] + \int_{t}^{t_f} \mathcal{L}[\boldsymbol{x}(t), \boldsymbol{u}(t), t] dt \right\}, \quad \boldsymbol{u} \in \mathcal{U} \tag{2.36}$$

The HJB theorem states that the optimal control $\boldsymbol{u}^*$ is given by the relationship:

$$\boldsymbol{u}^*(\boldsymbol{x}, t) = \arg \min_{u} \left(\mathcal{L}[\boldsymbol{x}(t), \boldsymbol{u}(t), t] + \nabla_x V(\boldsymbol{x}, t)^T f(\boldsymbol{x}, \boldsymbol{u}, t) \right), \quad \boldsymbol{u} \in \mathcal{U} \tag{2.37}$$

where the value function must satisfy the Hamilton-Jacobi-Bellman Partial Differential Equation (HJB-PDE):

$$-\frac{\partial V}{\partial t}(\boldsymbol{x}, t) = \min_{u} \left\{ \mathcal{L}[\boldsymbol{x}(t), \boldsymbol{u}(t), t] + \nabla_x V(\boldsymbol{x}, t)^T f(\boldsymbol{x}, \boldsymbol{u}, t) \right\}, \quad \boldsymbol{u} \in \mathcal{U} \tag{2.38}$$

Additionally, the value function must comply with the boundary condition:

$$V(\boldsymbol{x}(t_f), t_f) = \Phi(\boldsymbol{x}(t_f), t_f) \tag{2.39}$$

Analytical solutions of the HJB-PDE only exist for special system classes, e.g., linear systems with quadratic costs. In most cases, the solution has to be found numerically. There the problem of the curse of dimensionality arises, since for solutions the time, state, control, and possibly output spaces have to be sampled, which means that the computational complexity increases exponentially with the dimensions of the state, control, and output spaces [106]. Efforts to resolve the curse of dimensionality have resulted in approximate dynamic programming [106]. There, for example, function approximations are used to approximate the value function (Eq.(2.36)) or control policies (Eq.(2.37)). However, the global optimality of the solution cannot be guaranteed anymore in the latter approach. The most successful solution of these approximate methodologies for spacecraft trajectory optimization relies in Differential Dynamic Programming [107]. It is a gradient-based second-order technique that relies on HJB theorem and successive minimization of quadratic approximations of Eq.(2.36). DDP proceeds by iteratively performing a backward pass on the nominal trajectory to generate a new control sequence given by Eq.(2.37), and then a forward pass to compute and evaluate the cost of the trajectory.

2.3.5 Gradient-Based, Heuristic and Hybrid Solutions

Most previous approaches (e.g., indirect/direct single/multiple shooting and collocation) have converted the COCP to the problem of determining an unknown vector of decision variables $\boldsymbol{z} \in \mathbb{R}^{n_z}$. For direct methods, the unknown decision vector has to fulfill a set of non-linear constraints, while minimizing an objective function (i.e., solving an NLP problem). On the other hand, in indirect methods the unknown parameters only have to meet a set of nonlinear constraints (i.e. solving a system of algebraic equations). Notably, a root-finding problem can be formulated as an NLP with a fictitious or constant cost [91]. Methods for solving NLPs and systems of algebraic equations can be classified as gradient-based (also known as deterministic methods) heuristic or hybrid algorithms. They all are iterative methods, that use a different set of rules for evolving. Hereafter, the main lines for each of them will be drawn.

- **Gradient-based**: In a gradient-based method, an initial guess is made of the unknown decision vector $\boldsymbol{z}$. At the k^{th} iteration, a search direction $\boldsymbol{p}_k$, and a step length α_k, are determined. The search direction provides a direction in $\mathbb{R}^{n_z}$ along which to change the current value $\boldsymbol{z}_k$, while the step length provides the magnitude of the change. The update from $\boldsymbol{z}_k$ to $\boldsymbol{z}_{k+1}$ has the form: $\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + \alpha_k \boldsymbol{p}_k$. The iterations proceed until the KKT conditions are met. To compute the search direction, these methods require the user provide information for the gradient of the constraint and the objective function (if necessary). The most widely used methods are classified as sequential quadratic problems (e.g., SNOPT, NPSOL) or interior point methods (e.g., IPOPT, KNITRO). Extensive information about their implementations can be found in Refs. [108] and [109] respectively.

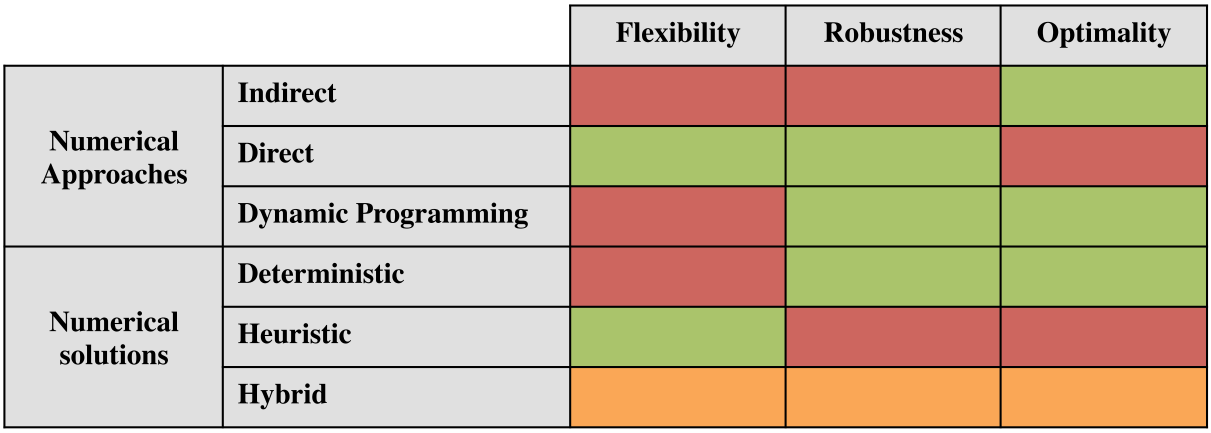
- **Heuristic:** The search is performed in a stochastic/metaheuristic manner without requiring gradient information. The most known class of heuristics are evolutionary algorithms. They start by generating a set of candidate solutions or individuals $\boldsymbol{z}_{i,0}$ for $i = 1, \ldots, n$, termed population. Thereafter, the population is iteratively modified by applying a set of stochastic rules $\Pi : \mathcal{Z} \longrightarrow \mathcal{Z}$, which may incorporate random processes, such that the population at $(k+1)^{th}$ iteration is computed as $\boldsymbol{z}_{i,k+1} = \Pi(\boldsymbol{z}_{i,k})$, and the iterations proceed until a stopping criteria is met (e.g., max number of iterations). The candidate with the lowest cost is deemed as the solution to the problem. Well known stochastic rules are genetic algorithms (GA) [110], which emulate evolutionary processes in genetics, and particle swarm optimization (PSO) [111], which is based on the idea of swarms of animals.

- **Hybrid:** Hybrid approaches combine a set of rules exploiting gradient-information and a set of rules based on heuristics searches to iteratively operate over a solution or a set of candidate solutions. Gradient-information is exploited to drive the constraints to zero, while heuristic rules are applied to efficiently explore large design domains or to manage integer variables. They are typically combined on a two-loop approach. The heuristic solver operates over a subset of decision variables in the outer loop. In the inner loop, the remaining subset of design parameters are optimized with the gradient-based method.

2.3.6 Discussion

The main benefit of using the indirect approach is that it provides assurances that the first-order optimality conditions are satisfied. Additionally, they may offer an interesting theoretical insight into the problem physical and mathematical characteristics. However, difficulties arise in that explicit derivations of the costate and control equations are required, which can be very difficult depending on the OCP being considered, and prior knowledge of the activeness of inequality constraints is necessary. Numerical techniques applied to the resulting MPBVP normally require an appropriate initial guess of the costates, which is often nonintuitive since they generally do not have physical interpretations. Moreover, the resulting trajectory is very sensitive to their values and the region of convergence tends to be quite narrow. The indirect approach is further complicated by the need to re-form the Hamiltonian and re-derive the adjoint equations and boundary conditions as different state variables, constraints and dynamics are considered. Because of these practical difficulties, indirect methods are not suitable to solve highly constrained spacecraft trajectory optimization problems.

On the other side, direct methods have the advantage that the user does not have to be concerned with deriving the first-order necessary conditions. Furthermore, direct methods are easier to initialize due to a larger domain of convergence and the physically intuitive meaning of the optimization variable. Although they still rely on a tentative guess and may not converge to the optimal solution, direct methods find at least a suboptimal solution unlike indirect approaches. Another point of success of direct methods is that even complex control or state constraints can be handled easily and that, in case of path inequality constraints, the sequence of free and constrained arcs does not need to be known a-priori. As a major drawback, with a direct method is always uncertain whether the trajectory found by solving the NLP is truly an optimal solution to the original OCP or a suboptimal one. Notwithstanding, direct methods have been extensively used for solving highly constrained spacecraft trajectory optimization problems in spite of the fact that they present lower optimality than indirect methods.



		Flexibility	Robustness	Optimality
Numerical Approaches	**Indirect**	🟥	🟥	🟩
	Direct	🟩	🟩	🟥
	Dynamic Programming	🟥	🟩	🟩
Numerical solutions	**Deterministic**	🟥	🟩	🟩
	Heuristic	🟩	🟥	🟥
	Hybrid	🟧	🟧	🟧

Figure 2.6: Methods and techniques in numerical approaches.

Dynamic programming has two main advantages when compared to all other methods presented. First, the whole state space is searched, thus an optimal solution is also the global optimum. Second, all controls are precomputed once a solution is found. This implies that closed-loop control policies instead of an open-loop control trajectory can be obtained, as well as it can be naturally extended to tackle uncertain and stochastic problems. The main drawback of dynamic programming relies on the curse of dimensionality. Therefore, memory and computational times of standard dynamic programming grow very quickly with the number of state variables and become impractical for high-dimensional state space. The direct application of dynamic programming is therefore limited in practice to problems with low state-space dimensionality. Notably, the curse of dimensionality is resolved when using approximated techniques, based on local approximations of the value function, such as Differential Dynamic Programming. However, the obtained solution is no longer guaranteed to be globally optimal and the closed-loop control is only locally valid.

Regarding the solution approaches, gradient-based approaches provide deterministic conditions for convergence. They are able to handle a large number of problem variables and constraints. However, they require the constraint and objective function to be twice differentiable. Consequently, they are not well suited for problems that use tabular data, or suffer from discontinuities. These methods require the user to provide an initial guess and the obtained solution will be in the neighborhood of the initial guess. Heuristic methods are well suited for problems with a reduced number of variables but with a high-dimensional space. While a gradient method is a local method a heuristic method is a global technique. These methods are more flexible, since they do not require the involved functions to be differentiable. However, when using heuristic algorithms, it is always uncertain if the obtained solution is optimal, since no optimality conditions are applied. In fact, in every different run a different solution can be obtained. Moreover, constraints are difficult to be met, since no gradient information is exploited. Hybrid approaches exhibit intermediate performances in terms of flexibility, robustness and optimality with respect to deterministic and heuristics methods.

Qualitative performance comparison of dynamic programming, direct methods, and indirect approaches, along with gradient-based and heuristic solutions for solving continuous optimal control problems is shown in Fig 2.6 in terms of three criteria: flexibility, robustness and optimality. The definition of each criteria can be found in Table 1.3. The green color means that the method affects positively on the selected criteria, the red color means that it influences negatively, whereas orange implies intermediate performances. Notably, direct methods exhibit good performances in terms of flexibility and robustness, whereas dynamic programming is more suitable when seeking for optimality and robustness. Regarding numerical solution approaches, hybrid methods provide a good compromise between optimality, robustness, and flexibility, when compared to purely heuristic or gradient-based solutions. Finally, note that the presented assignment of methods to the three categories is not necessarily unique since some methods combine characteristics from several categories to generate improved performances.

2.4 Numerical Solution Approaches for HOCPs

Numerical approaches to solve HOCPs are also categorized as dynamic programming, direct methods, or indirect methods. They inherit all of the positive and negative aspects from their application to COCP [39]. However, optimal control for hybrid systems is challenging due to the close interconnection of continuous and discrete dynamics. Methods for COCPs problems are not able to handle HOCPs since discrete decisions influence the continuous optimization. Similarly, methods for purely discrete optimization problems are unsuitable since the discrete optimization strongly depends on the continuous optimal control. Combining methods from continuous optimal control and discrete optimization is not straightforward. Continuous optimal control relies on infinitesimal variations of control and state variables and derivatives of functions. Such concepts are difficult to translate to discrete decision problems. In contrast, discrete optimization often relies on graph based search methods, which are not applicable for continuous optimal control problems as these are infinite dimensional.

First-order necessary optimality conditions for HOCPs are provided by the so-called hybrid minimum principle in (HMP) [112], which is generalization of the PMP for control systems with both continuous and discrete states and dynamics. It includes state and adjoint differential equations, a minimization of the Hamiltonian with respect to the continuous control, initial and terminal conditions for the state and/or adjoint variables, jump conditions for the adjoint variables, and Hamiltonian value conditions specifying the optimal discrete event times. However, no condition with respect to the sequence of discrete events can be given. This fact would imply that the sequence of gravity assists (when considered instantaneous) have to be provided by the user. For this case, the HMP converts the HOCP into a MPBVP, which can be solved applying indirect shooting, collocation or gradient-methods [112]. Dynamic programming theory has been extended in [113] to tackle general classes of HOCPs, which in fact can be solved with DDP techniques. Though several algorithms have been developed, the convergence of the approximated value function to the true value function is in general still to be shown [112].

HOCPs that are solved by direct methods are usually formulated as Mixed-integer Nonlinear Programming (MINLP), i.e. NLPs where the optimization variables may be real or discrete. If the discrete state is identified with a finite sequence of phases and the discrete control can be described by an integer variable, then the HOCP can be converted to a MINLP by applying direct single/multiple shooting or collocation, where the continuous/discrete controls are discretized/parametrized. The solution to MINLPs has been shown to be NP-hard to solve [114], i.e., it is “at least as hard as any NP-problem”. Therefore, various methods have been developed to reduce the computational time. The most prominent method in hybrid spacecraft trajectory optimization consists on a two-nested optimization loop. The inner loop solves for the continuous variable with a gradient-based solver, and the outer loop handles the discrete variables with a heuristic algorithm. Other methods include: branch and bound, branch and cut, outer approximation, generalized Benders decomposition [115].

TABLE 2.2: Representative Tools Implementing Direct Methods for Low-Thrust Trajectory Optimization

Name	Ref	Company/Organization/Author	Approach	Solution	Obj.	Dynamics	States	Transfers
ASTOP	[116]	Space Flight Solutions	Single Shooting	GB	SO	PR-NBP	CSV	IT
COPERNICUS	[117]	Texas Univ., JSC	Multiple Shooting	GB	SO	PR-NBP	CSV	G
jTOP	[41]	Tokio Univ., JAXA	Multiple Shooting	GB	SO	PR-NBP	CSV	G
DITAN	[40]	ESA, Milano Univ.	Collocation	GB	SO	PR-NBP	CSV	G
MODHOC	[118]	Strathclyde Univ.	Collocation	HY	MO	PR-NBP	CSV	G
MANTRA	[119]	ESA	Collocation	GB	SO	PR-NBP	CSV	G
DIRETTO	[120]	Milano Univ.	Collocation	GB	SO	PR-NBP	CSV	G
MAVERICK	[121]	Colorado Boulder Univ.	Collocation	GB	SO	PR-NBP	CSV	G
MColl	[122]	NASA.	Collocation	GB	SO	PR-NBP	CSV	G
COLT	[123]	Purdue Univ.	Collocation	GB	SO	PR-NBP	CSV	G
GMAT	[124]	NASA	Collocation	GB	SO	-	-	G
STK	[125]	AGI	Collocation	GB	SO	-	-	G
OTIS	[126]	GCR, Boeing	Collocation	GB	SO	-	-	G
POST	[127]	NASA	Single Shooting	GB	SO	-	-	G
SOCS	[55]	Boeing	Collocation	GB	SO	-	-	G
DIDO	[128]	TOMLAB	Collocation	GB	SO	-	-	G
GPOPS	[129]	Univ. of Florida	Collocation	GB	SO	-	-	G
OPTELEC	[57]	Airbus	Multiple Shooting	GB	SO	PR-TBP	MEE	PC
LOTOS	[56]	ASTOS Solutions	Collocation	GB	SO	PR-TBP	MEE	PC
XIPSTOP	[130]	Boeing	Collocation	GB	SO	PR-TBP	MEE	PC
GALLOP	[131]	JPL,Purdue Univ.	Multiple-Shooting	GB	SO	KM	CSV	IT
COLTT	[132]	Colorado Boulder	Multiple-Shooting	GB	SO	KM	CSV	IT
LInX	[133]	J.H. Univ., Nabla Zero	Multiple-Shooting	GB	SO	KM	CSV	IT
BOLTT	[134]	Colorado Boulder	Multiple-Shooting	GB	SO	KM	CSV	IT
MALTO	[135]	JPL	Multiple-Shooting	GB	SO	KM	CSV	IT
EMTG	[46]	GSFC, Illinois Univ.	Multiple-Shooting	HY	MO	KM	CSV	IT
PaGMO	[136]	ESA	Multiple-Shooting	HY	SO	KM	CSV	IT
GA-GALLOP	[137]	Purdue Univ.	Multiple-Shooting	HY	MO	KM	CSV	IT
-	[138]	Zuiani et al.	Multiple-Shooting	GB	SO	SM	CSV	IT
DIFINC	[139]	Coverstone et al.	Differential Inclusion	GB	SO	PR-TBP	CSV	IT
-	[140]	Gerald et al.	Single Shooting	HS	SO	PR-TBP	PSV	IT
-	[141]	Pontani et al.	Single Shooting	HS	SO	PR-TBP	PSV	IT

* GB=Gradient-Based, HS=Heuristic, HY=Hybrid, SO=Single-Objective, MO=Multi-Objective, IT=Interplanetary, PC=Planetocentric, G=General, SM=Stark-Model, KM=Kepler-Model, AVG=Averaging, AN=Analytical, CSV=Cartesian-State-Vector, MEE=Modified-Equinoctial-Elements, COE=Classical-Orbital-Elements, PSV3=Cylindrical-Coordinates, PR=Perturbed-Restricted, TBP=Two-Body-Problem, NBP=N-Body Problem

Table 2.3: Representative Tools Implementing Predefined Control laws for Low-Thrust Trajectory Optimization

Name	Ref	Company/Organization/Author	Approach	Solution	Obj.	Dynamics	States	Transfers
HYTOP	[142]	Aerospace Corp.	Blended Control	GB	SO	PR-TBP	MEE	PC
-	[64]	Yang Gao	Blended Control	GB	SO	PR-TBP+AN+AVG	COE	PC
-	[143]	Yang Gao	COV-Based	GB	SO	PR-TBP+AVG	MEE	PC
-	[144]	Strathclyde Univ	Blended Control	HY	MO	SM+AVG	COE	PC
SEPDOC	[63]	Kluever et al.	Blended Control	GB	SO	PR-TBP+AVG	COE	PC
-	[65]	Hudson et. al	Finite-Fourier-Expansion	GB	SO	PR-TBP+AN+AVG	COE	PC
-	[145]	Chang et. al	Lyapunov Control	GB	SO	PR-TBP	CSV	PC
GA-Q-Law	[70]	JPL	Lyapunov Control	HS	MO	PR-TBP	MEE	PC
STOUR-LTGA	[47]	JPL, Purdue Univ.	Shape-based	HS	SO	PR-TBP+AN	PSV	IT
IMAGO	[50]	Pascale et al.	Shape-based	HS	SO	PR-TBP+AN	MEE	IT
-	[48]	Wall et al.	Shape-based	HS	SO	PR-TBP+AN	PSV	IT
-	[52]	Taheri et al.	Shape-based	HS	SO	PR-TBP+AN	PSV3	IT
-	[53]	Gondelach et al.	Shape-based	HS	SO	PR-TBP+AN	PSV3	IT
-	[88]	Roa et al.	Shape-based	HS	SO	PR-TBP+AN	PSV	IT
InTrance-GA	[146]	DLR	Neural control	HY	SO	PR-TBP	CSV	IT

* GB=Gradient-Based, HS=Heuristic, HY=Hybrid, SO=Single-Objective, MO=Multi-Objective, IT=Interplanetary, PC=Planetocentric, G=General, SM=Stark-Model, KM=Kepler-Model, AVG=Averaging, AN=Analytical, CSV=Cartesian-State-Vector, MEE=Modified-Equinoctial-Elements, COE=Classical-Orbital-Elements, PSV3=Cylindrical-Coordinates, PR=Perturbed-Restricted, TBP=Two-Body-Problem, NBP=N-Body Problem

Table 2.4: Representative Tools Implementing Dynamic Programming for Low-Thrust Trajectory Optimization

Name	Ref	Company/Organization/Author	Approach	Solution	Obj.	Dynamics	States	Transfers
MYSTIC	[42]	NASA	DDP	-	SO	PR-NBP	CSV	G
-	[147]	Colorado Boulder Univ.	DDP	-	SO	PR-TBP	MEE	PC
HDDP	[148]	Lantoine et al.	HDDP	-	SO	SM/KM	CSV	G

* GB=Gradient-Based, HS=Heuristic, HY=Hybrid, SO=Single-Objective, MO=Multi-Objective, IT=Interplanetary, PC=Planetocentric, G=General, SM=Stark-Model, KM=Kepler-Model, AVG=Averaging, AN=Analytical, CSV=Cartesian-State-Vector, MEE=Modified-Equinoctial-Elements, COE=Classical-Orbital-Elements, PSV3=Cylindrical-Coordinates, PR=Perturbed-Restricted, TBP=Two-Body-Problem, NBP=N-Body Problem

Table 2.5: Representative Tools Implementing Indirect Methods for Low-Thrust Trajectory Optimization

Name	Ref	Company/Organization/Author	Approach	Solution	Obj.	Dynamics	States	Transfers
VARITOP	[149]	JPL	Single Shooting	GB	SO	PR-TBP	CSV	IT
SEPTOP	[150]	JPL	Single Shooting	GB	SO	PR-TBP	CSV	IT
NEWSEP	[151]	JPL	Single Shooting	GB	SO	PR-TBP	CSV	IT
SAIL	[152]	JPL	Single Shooting	GB	SO	PR-TBP	CSV	IT
HILTOP	[153]	SpaceFlight Sol.	Single Shooting	GB	SO	PR-TBP	CSV	IT
ETOPH	[154]	CNES	Single Shooting	GB	SO	PR-TBP	CSV	IT
ITOP	[155]	Aerospace Corp.	Single Shooting	GB	SO	PR-TBP	MEE	PC
LT20	[156]	Milano Univ.	Single Shooting	GB	SO	PR-TBP	MEE	PC
Tfmin	[157]	CNES	Single Shooting	GB	SO	PR-TBP	COE	PC
-	[158]	Kéchichian	Single Shooting	GB	SO	PR-TBP	MEE	PC
T-3D	[159]	Thales	Single Shooting	GB	SO	PR-TBP+AVG	MEE	G
ELECTRO	[160]	OHB	Single Shooting	GB	SO	PR-TBP+AVG	MEE	PC
MIPELEC	[161]	CNES	Single Shooting	GB	SO	PR-TBP+AVG	MEE	PC
SEPSPOT	[162]	NASA	Single Shooting	GB	SO	PR-TBP+AVG	MEE	PC
GA-SEPTOP	[163]	JPL	Single Shooting	HY	MO	PR-TBP	CSV	IT
LOTTO	[75]	SES Engineering	Single Shooting	GB	SO	PR-TBP	MEE	PC
-	[164]	Torino Univ.	Single Shooting	HS	SO	PR-TBP	CSV	IT
-	[165]	Pontani et al.	Single Shooting	HS	SO	PR-TBP	PSV	IT
-	[166]	Lee et al.	Single Shooting	HS	MO	PR-TBP	CSV	IT
BNDSCO	[167]	Hamburg. Univ	Multiple Shooting	HS	SO	-	-	G
-	[168]	Meng et al.	Multiple-Shooting	GB	SO	PR-TBP	MEE	PC
-	[169]	Olympio	Gradient method	-	SO	PR-NBP	PSV	G

[*] GB=Gradient-Based, HS=Heuristic, HY=Hybrid, SO=Single-Objective, MO=Multi-Objective, IT=Interplanetary, PC=Planetocentric, G=General, SM=Stark-Model, KM=Kepler-Model, AVG=Averaging, AN=Analytical, CSV=Cartesian-State-Vector, MEE=Modified-Equinoctial-Elements, COE=Classical-Orbital-Elements, PSV3=Cylindrical-Coordinates, PR=Perturbed-Restricted, TBP=Two-Body-Problem, NBP=N-Body Problem

2.5 Review of Existing Low-Thrust Optimization Tools

The preliminaries required for formulating and solving low-thrust trajectory optimization problems have been briefly explained through previous sections. Hereafter, an overview of existing and representative low-thrust trajectory optimization tools and research works will be presented. Their main characteristics, capabilities and limitations will be outlined and compared to each other. Firstly, analytical solution approaches will be presented and followed by indirect, direct and dynamic programing methods. A special section is dedicated to analyze the predefined control laws applied within direct methods schemes. A total of 90 references have been investigated, among which 18 correspond to analytical solutions methodologies, while the remaining 72 are numerical approaches. Numerical approaches corresponding to indirect, direct, predefined control laws and dynamic programming have been summarized in Tables 2.2-2.5 respectively. They include information about the name of the tool, the developing company, organization or author, the type of numerical approach, objective, dynamics, state vector, and application.

The yearly distribution for the publication dates of the examined references is shown in Fig. 2.7a. It can be seen that half of the references has been published in the last decade. Notably, among the analyzed numerical methods, direct methods represent a 65%, while indirect and dynamic programming are the 30% and 5% respectively. The most widely implemented direct method has been the single-shooting algorithm (38%), followed by collocation (32%), multiple-shooting (18%), and differential inclusion (2%). Similarly, the most common indirect method is single shooting (86%), followed by multiple-shooting (9%) and gradient methods (5%). Remarkably, a 75% of the numerical solution approaches use a gradient-based solver to tackle the resulting mathematical problem, while a 20% use purely heuristic algorithms and the remaining 5% apply hybrid algorithms. Finally, most approaches have been dedicated to solve single-objective problems (83%), while the remaining 17 % exhibit the capability of solving multi-objective optimization problems. These statistics are illustrated in Figure 2.8.

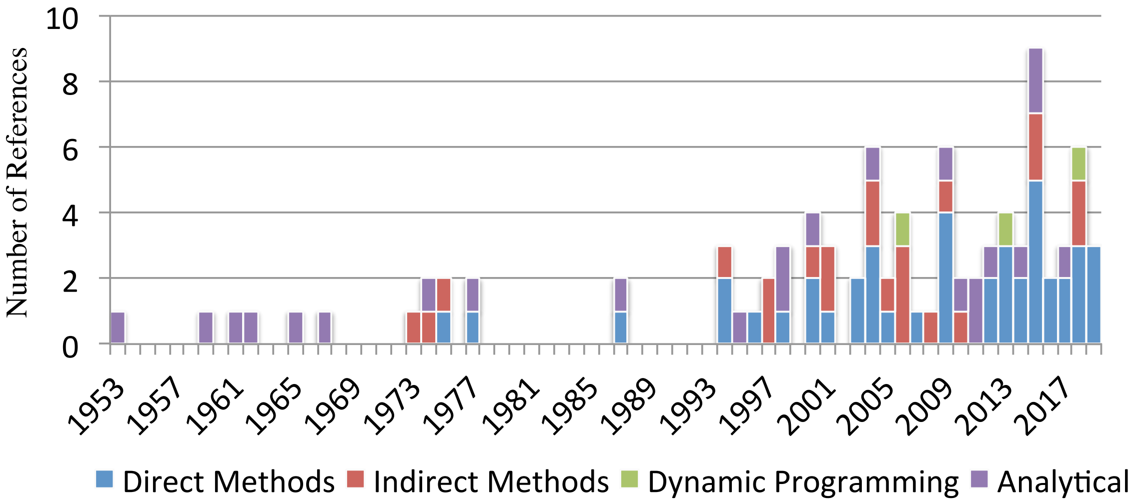


Figure 2.7: Illustration of the problem statement

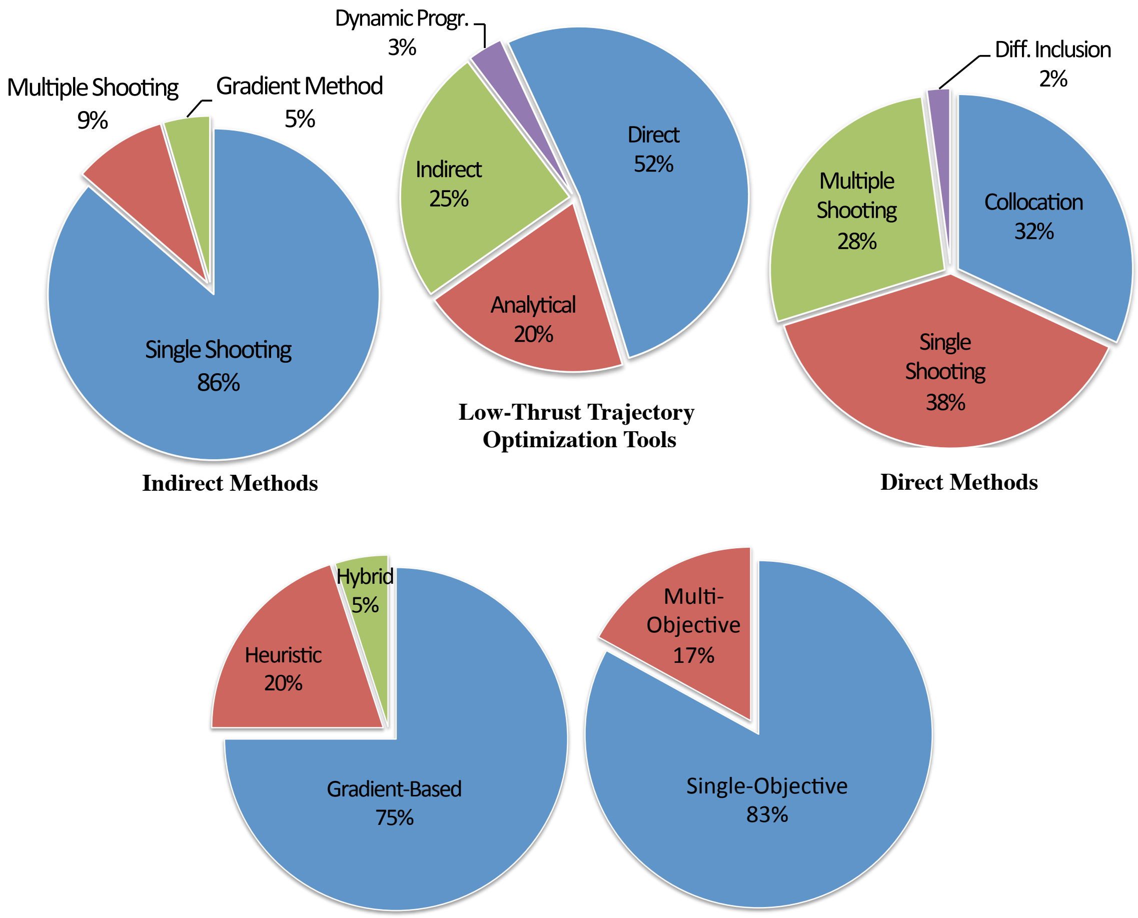


FIGURE 2.8: Overview of investigated Low-Thrust Optimization tools

2.5.1 Analytical solutions

There have been valuable efforts to solve simple low-thrust trajectory cases analytically. For instance, by either fixing the direction of the thrust, e.g., constant tangential or radial thrust, or by simplifying the boundary conditions, e.g., solving coplanar circle-to-circle transfers. They are convenient for rapidly evaluating low-thrust trajectories, or to be combined with a numerical optimization technique, either as an initial guess or as a dynamical model. One of the first pioneers in the history of analytical solutions was Tsien [170]. In his work of 1953, which has been exquisitely reproduced by Battin [171], analytical approximated planar solutions are derived in case of radial and circumferential thrust for initially circular orbits. An alternative closed-form solution in terms of an orbital anomaly and elliptic functions was derived by Izzo et al. [172]. Bombardelli et al. [173] and Gonzalo et al. [68] proposed a first-order asymptotic solution for the trajectory in the case of constant tangential and radial acceleration respectively. Exact solutions to the tangential thrust problem have eluded researchers, but explicit solutions for certain variables can be found. For instance, the expressions defining the escape conditions or the amplitude of the bounded motion have been provided by different authors (e.g., Prussing et al. [174] and Mengali et al. [175]).

In 1961 Edelbaum's [176] original analysis involved a low-thrust transfer between two circular orbits with a constant out-of-plane angle. He derived analytical expressions for the total velocity change and time of flight, and served as a starting point for many subsequent analysis. Later, Kéchichian [177] reformulated Edelbaum's problem [176] by applying optimal control theory to the minimum-time transfer problem to obtain the optimal time varying semi-major axis, inclination and yaw angles. Edelbaum [178] provided a complete first-order asymptotic solution for the Hamiltonian system resulting from power-limited transfer between coplanar elliptic orbits of arbitrary size and orientation. Fernandes et al. [179] obtained a first-order analytical solution, which includes short periodic terms, of the resulting average Hamiltonian system resulting from the optimal low-thrust transfers between coplanar orbits with small eccentricities. Zuiani et al. [138] presented a first-order analytical solution for transfers between general orbits. He exploits the benefits of using a set of non-singular orbital elements.

Ruggiero et al. [180] developed analytical solutions for the optimal steering angles that maximize the instantaneous change of each COE independently. In [181], Kéchichian derived analytical solutions for transferring between circular orbits for two different scenarios: for the simultaneous change of semimajor axis and inclination, and for changing the argument of the ascending node and the semimajor-axis. Burt [182] presented closed-form analytical formulas to compute the velocity increment and trip time for adjusting the eccentricity at a constant semi-major axis. This is accomplished with a constant in-plane acceleration perpendicular to the semi-major axis of the ellipse. Pollard [183] extended Burt's approach to the case of discontinuous acceleration by analyzing the perigee-and apogee-centered burn arcs, and extended the analysis to simultaneously change the eccentricity and inclination. Many of the aforementioned analytical approaches are implemented in the preliminary design software tool CAMELOT (Computational-Analytical Multi-fidelity Low-thrust Optimization Toolbox) [184].

Furthermore, there exists some trajectory analytical results for transfers incorporating Earth environmental effects. For instance, Kéchichian [61] obtained analytical solutions under the assumption of constant tangential thrust. He included the effect of J_2 and engine shut down during eclipses along small-to-moderate eccentricity orbits in terms of non-singular elements. Kluever [185] included periods of zero thrusting due to the Earth shadow eclipses and develop a semi-analytical algorithm to solve the Edelbaum's problem. Kechichian [186] and Colasurdo et al. [187] also developed a purely analytical method for obtaining low-thrust and multi-revolution transfers between coplanar circular orbits in the presence of Earth shadow, constraining the eccentricity to remain zero during the transfer. A two-variable asymptotic expansion method applicable to transfers from elliptic orbits was considered by Flandro [188], who included shadow penalty terms due to eclipses. Gao [64] obtained analytical solutions of the averaged equations when a predefined control law is applied, including shadow and J_2.

2.5.2 Indirect Methods

The most common indirect method is the indirect single shooting. It has been implemented in the tools SEPTOP (Solar Electric Propulsion Trajectory Optimization Program) [150], VARITOP (VARIational calculus Trajectory Optimization Program) [149], NEWSEP (NEW Solar Electric Propulsion trajectory optimization program) [151], and SAIL [152]. These tools have been developed at the Jet Propulsion Laboratory (JPL) and they are part of the Low-Thrust Trajectory Tool Suite (LTTT). The most general of the suite is VARITOP, which handles nuclear electric propulsion as well as solar electric propulsion and sail trajectories. However, solar electric engines and solar sails are more accurately modeled in the SEPTOP and SAIL programs respectively. NEWSEP is a variation of SEPTOP that can accept discrete values of a thruster's throttle table rather than approximating the polynomial as its predecessor. They have been extensively used to design a variety of missions. For instance, NEWSEP provided trajectory support for the Deep Space 1 mission [152]. Runtimes for these tools range from hours to days [152], especially for those trajectories with numerous intermediate flybys.

Indirect single shooting algorithms were also implemented in the tools HILTOP (Heliocentric Interplanetary Low Thrust Optimization Program) [153] and ETOPH (Electric Transfer Optimization with Planetocentric and Heliocentric phases) [154]. HILTOP was employed in numerous NASA and industry studies of missions to most planets, comets and asteroids. This tool lead to the development of MAnE-EP (Mission Analysis Environment for Electric Propulsion), which is an updated version of HILTOP. The tool ETOPH incorporates a smoothing technique for overcoming the difficulty of predefining the sequence of active constraints, and to reduce the numerical instabilities associated with the bang-bang structure of the control. Aforementioned tools implement a patched two-body dynamics with CSV. Therefore, they are well suited for solving interplanetary trajectories, requiring the user to provide the flyby sequence, yet not for orbit-raising trajectories. They use a gradient-based solvers and they require an initial guess that is typically difficult to obtain and are limited to obtain a single optimized solution.

Previous limitations are surmounted by using heuristic or hybrid techniques. Pontani and Conway [165] employed a PSO algorithm to solve an Earth-Mars rendezvous problem. They ignored the transversality conditions, as the objective function was optimized by the PSO and the constraints on the final state were included as penalties. A similar technique was presented by Lee et al. [166]. They combined a GA with simulated annealing to obtain trade-offs between delivered mass and required flight time for two-body and a three-body orbit transfers. Coverstone et al. [163] used a multi-objective GA to choose initial guesses for SEPTOP and optimized with respect to delivered mass, flight time and number of revolutions for an Earth-Mars rendezvous mission. Rosa and Casalino [164] employed a GA to search for the combination of unknown parameters that minimizes the error on the boundary conditions; the minimum-error combination was provided as a guess to a gradient-based solver to obtain a converged solution. The procedure was tested in direct and multiple-gravity-assist missions to Mars.

Previous single shooting methods are not able to analyze planet-centered trajectories beyond a simple escape or capture maneuver, mainly because the EOM are expressed in CSV. Therefore, single shooting methods with MEE or COE have been developed. In [158], Kéchichian analytically derived the Hamiltonian system in terms of non-singular elements without additional perturbations than a constant thrust acceleration. He solved for the unknown initial costates for a LEO-GEO transfer using a deterministic solver. The initial guess was obtained by setting to zero the values of the initial costates. A similar approach was implemented in the software tool Tfmin[4] [157]. However, the technique from Kéchichian allows to rendezvous in the target orbit, while Tfmin was developed for free final longitude. Later, Kéchichian [189] extended his approach to account for the effect of J_2 perturbation, derived the set of dynamical and adjoint equations, and solves it for a LEO-GEO case. The initial guess was obtained by solving the problem without the oblateness effect. Kéchichian [190] further developed the low-thrust rendezvous in equinoctial elements by considering Earth zonal harmonics up to J_4.

However, previous approaches neither account for the effect of switching off the engine during eclipse, nor include coasting periods to obtain minimum-fuel consumption trajectories. For such purpose, software tools such as ITOP (Indirect Trajectory Optimization Program) [155], LT20 (Low-Thrust Trajectory Optimizer) [156], and LOTTO [75] were developed. They all are high-fidelity tools capable of solving min-time or min-fuel orbit transfers by implementing a switching function. They include eclipse shadowing, non-spherical Earth potential, solar radiation pressure, third-body perturbations, drag force, and altitude constraints via penalty functions. LOTTO further include slew rate restrictions and longitude targeting. Notably, ITOP was used for designing the electric orbit-raising maneuvers for the Al Yah 3 satellite [155]. ITOP and LT2O use gradient information to solve for the unknown initial costates. On the contrary, LOTTO uses a robust heuristic search method without relying on an initial guess. It selects the initial values for the costates that minimizes the error on the final constraints.

Accurately integrating the trajectory for the indirect shooting method is very time-consuming, due to the non-linearities in the dynamics, the long flight-times and the high number of orbital revolutions. Thus, many authors have taken advantage of orbital-averaging techniques to greatly increase the speed of computation at the expense of fidelity. One of the most known softwares is SEPSPOT[5] (Solar Electric Propulsion Steering Program for Optimal Trajectories) [162]. It was developed in the mid-1970's by Edelbaum et al. [191] to solve minimum-time transfers with a set of non-singular elements. The program includes options for oblateness, shadowing with or without delay in thruster startup, an analytic radiation and power degradation model, and altitude constraints as penalties. For several decades, SEPSPOT has been NASA's primary tool for the optimization of planetocentric low-thrust trajectories. However, the convergence probability is greatly diminished when solar cell degradation effects are included. The program has the option to solve hybrid transfers. For the initial high-thrust stage, one or two impulses of fixed magnitude can be included, and the initial orbit is assumed to be circular.

[4] Tfmin is freely available to download at `http://apo.enseeiht.fr/tfmin/`

[5] It was previously named SECKSPOT (Solar Electric Control Knob Setting Program by Optimal Trajectories)

Other examples with averaged EOM include ELECTRO (ELECtric propulsion TRajectory Optimisation) [160], MIPELEC (Satellite Positioning with Electric Propulsion) [161] and T3D [159]. MIPELEC[6] is based on the theory developed by Geffroy and Epenoy [161] to solve min-time orbit-raising transfers with MEE, without shadow or oblateness effects. It is initialized by a user-provided guess or by an planar analytical approximation. ELECTRO implements EOM based on MEE to solve min-time transfers, including shadow and oblateness effects. An arbitrary user-provided guess is transformed into a feasible guess by an initial restoration phase. The tool T3D solves min-time and min-fuel transfers including coast arcs by a smoothing mechanism, third-body perturbations, solar radiation pressure, oblateness, atmospheric drag and eclipse effects. A continuation method is implemented to run from an arbitrary guess. The main difference between the averaging method implemented in MIPELEC and those in T3D and ELECTRO, is that the true longitude is the independent variable instead of time.

The remaining indirect methods, namely multiple-shooting, collocation and gradient-based, have been less popular, yet also have provided successful results. For instance, the general-software tool BNDSCO [167] implements indirect multiple-shooting. Oberle and Grimm [167] applied it intensively to study Earth-Mars low-thrust transfers. Meng et al. [168] implemented an indirect multiple-shooting algorithm where the transversality conditions were ignored, and the EOM were expressed in MEE. The unknown costates and the objective function were optimized by a gradient-based solver. He successfully solved a transfer from GTO to GEO. Olympio [43] developed an indirect gradient-based method using second-order derivative information. He was able to automatically find gravity assists naturally exploiting the multi-body dynamics including space and capture phases. He also applied it to design an orbit raising transfer from LEO to MEO. Finally, although indirect collocation methods have been used in other fields, the author has not found any example of its application to low-thrust trajectory optimization.

2.5.3 Direct Methods

A variety of methods for computing multi-gravity assisted interplanetary and Earth-orbit transfers in accurate dynamical models implement direct methods combined with gradient-based solvers: POST (Program to Optimize Simulated Trajectories) [127] and ASTOP (Arbitrary Space Trajectory Optimization Program) [116] implements single shooting, Copernicus [117] and jTOP [41] use multiple shooting, while others such as DITAN (Direct Interplanetary Trajectory Analysis), MODHOC (Multi-Objective Direct Hybrid Optimal Control) [118], OTIS (Optimal Trajectories by Implicit Simulation) [126], MANTRA [119], GMAT (General Mission Analysis Tool) [124], DIRETTO (DIREct collocation tool for Trajectory Optimization) [120], MAVERICK [121], Mcoll [122], COLT (Collocation with Optimization for Low-Thrust) [132], SOCS (Sparse Optimal Control Software) [55], GPOPS (Gauss Pseudospectral Optimization Software) [129], DIDO (Direct and Indirect Dynamic Optimization) [128], and STK/Astrogator (Systems Tool Kit) [125] implement collocation methods.

[6] MIPELEC is freely available to download at https://logiciels.cnes.fr/en/content/mipelec

Some previous approaches correspond to software tools specifically designed for optimizing low-thrust trajectories (e.g., DITAN, MANTRA, DIRETTO, MAVERICK, Mcoll, COLT), while others are general-purpose products for solving OCPs that have been used for solving low-thrust transfer problems (e.g., MODHOC, OTIS, SOCS, GPOPS, DIDO). Notably, MODHOC is able to automatically search over a multi-objective design space and to handle discrete variables. Others are general space mission analysis tools that have specific modules for low-thrust trajectory optimization (e.g., GMAT, STK). They all have proven to be effective for the design of low-thrust transfers. For instance, MANTRA and DITAN were used to design the multiple-flyby trajectory for Bepicolombo [8], while jTOP was used for the the trajectory for the micro-spacecraft PROCYON [41]. They implement multi-body dynamics, but require the user to provide the sequence of flybys as well as an appropriate initial guess to converge. Additionally, the computational load make them unsuitable for the preliminary design.

Consequently, faster tools were developed at the cost of fidelity. One of the most widely-used algorithms for interplanetary transfers is the Sims-Flanagan Transcription (SFT) scheme. It implements a multiple-shooting scheme, the analytical Kepler model for the control, and instantaneous flybys. Most known tools include: GALLOP (Gravity Assisted Low-Thrust Local Optimization Program) [131], COLTT (CCAR Optimal low-Thrust Tool) [132], LInX (Low-thrust Interplanetary eXplorer) [133], MALTO (Mission Analysis Low-Thrust Optimizer) [135], EMTG[7] (Evolutionary Mission Trajectory Generator) [192], BOLTT (Boulder Optimal Low-Thrust Tool) [134] and PaGMO[8] (Parallel Global Multi-Objective Optimizer) [136]. Solutions from these tools are usually used as initial guesses for higher-fidelity tools. For instance, MALTO and GALLOP provide initial guesses for Copernicus and OTIS, while EMTG's solutions were used to feed GMAT [193]. A similar approach to the SFT was developed by Zuiani et al. [138], yet implementing the analytical Stark model between the multiple-shooting nodes.

Some of the previous methods used hybrid solutions approaches to avoid the need for the user to provide a suitable initial guess. For instance, Vavrina and Howell [137] presented GA-GALLOP, a program that use a GA to automatically provide initial guesses for GALLOP and to explore the multi-objective design space in terms of flight time and final mass. It was applied to Mars and Jupiter missions including one flyby. Yam et al. [136] used monotonic basin hopping (MBH) to automatically feed PaGMO. The approach was applied to maximize the final mass on a mission to Mercury involving up to six flybys. However, the tool require the user to provide the flyby sequence. An automated solution for the number and sequence of gravity assists has been addressed by Englander and Conway [192] in EMTG. In their approach they combine two nested optimization algorithms. The outer loop uses a GA to select the flyby number and sequence while the inner loop solves the corresponding sequence of interplanetary legs using MBH along with the SFT scheme. The method was proven to automatically determine the flyby sequences that maximize the delivered mass for missions to Mercury, the asteroid belt, and Pluto. This methodology was also tested on multi-objective problems [46].

[7] EMTG is freely available to download at https://opensource.gsfc.nasa.gov/projects/emtg/

[8] PaGMO is freely available to download at https://github.com/esa/pagmo

A different approach has been considered by Gerald and Converstone-Carrol [140], and by Pontani et al. [141], who only relied on population-based heuristic methods to find a solution of the direct shooting transcription resulting from planar low-thrust interplanetary transfers without flybys. The former implemented a GA to solve for the time-discretized thrust directional angles that minimize the transfer time for an Earth-Mars transfers, and that minimizes the fuel consumption for an Earth-Mercury trajectory. They included a binary optimization variable to determine wether the engine is in thrusting or coasting mode. Constraints on the final state have been applied as penalties in the objective function. The latter modeled the thrust steering law as a linear combination of B-Spline functions and used a particle swarm algorithm to optimize the parameters defining them. They claimed that despite its simplicity and intuitiveness, the particle swarm methodology proved to be quite effective in finding the optimal solution to orbital rendezvous optimization problems with considerable numerical accuracy.

Other available software tools are especially dedicated to solve minimum-time and minimum-fuel electric orbit-raising problem including operational constraints, such as LOTOS (Low-thrust Orbit Transfer Optimization Software) [56], XIPSTOP (Xenon Ion Propulsion System Trajectory Optimization Program) [130], and OPTELEC [57]. The tools LOTOS and XIPSTOP implement a direct collocation scheme combined with a gradient-based solver, while OPTELEC uses multiple-shooting with a gradient-based solver. All of them include the possibility of imposing eclipse or radiation constraints, slew rate and power consumption restrictions, slot targeting, avoidance of the GEO ring, Sun-angle or sensor pointing constraints. They implement a perturbed two-body dynamics along with accurate models for Earth Oblateness. They have proven to successfully solve numerous transfers to GEO. For example, XIPSTOP and OPTELEC are used to calculate the maneuvers for Boeing's and Airbus all-electric platforms respectively. Notably, LOTOS and OPTELEC are able to compute hybrid transfers, where the chemical orbit-raising is followed by an electric orbit-raising phase.

The remaining class of direct approaches refers to differential inclusion. Only one algorithm was found by the author. The tool DIFINC (DIFferential INClusion) was presented in [139] by Coverstone and William to compute low-thrust trajectories in the two-body problem with cartesian coordinates. This formulation removes explicit control dependence from the problem statement thereby reducing the dimension of the parameter space of the resulting nonlinear programming problem. They presented three interplanetary trajectory examples: an Earth-Mars constant specific impulse transfer, an Earth-Jupiter constant specific impulse transfer, and an Earth-Venus-Mars variable specific impulse gravity assist. The work was later extended by Hargens and Coverstone [194]. They implemented DIFINC in terms of the modified equinoctial orbital elements and applied it to solve several missions including both nuclear electric and solar electric propulsion systems. The results obtained showed good agreement between this method and solutions obtained with industry-standard software, such as VARITOP.

2.5.4 Predefined Control Laws

In this section, the direct methods that have approximated the control law by predefined guidance schemes will be detailed. They yield to sub-optimal solutions but are faster. The first class of the investigated control laws implement the COV-based guidance, and includes HYTOP [142] (HYbrid Trajectory Optimization Program) and the work done by Gao [143]. The former was developed in 1994 by Ilgen, uses orbital averaging and can calculate time-optimal and minimum propellant orbit raising transfers, constrained by Earth shadowing and oblateness. The software has been also applied to obtain a wide range of maximum-payload transfers to GEO using combined-chemical-electric propulsion. It has been also used to provide initial guesses to the indirect optimization software ITOP. In the work presented by Gao [143], a multiple-shooting scheme combined with orbital averaging was used to solve a series of minimum-time LEO-GEO and GTO-GEO transfers were solved using MEE, oblateness and a cylindrical shadow model. Results showed good agreement with the unaveraged dynamics.

The second class of methods include BC. In 1998, Kluever and Oleson proposed SEPDOC [63] (Solar Electric Propulsion Direct Optimal Control), which includes three extremal laws for changing semi-major axis, eccentricity and inclination. It includes averaging, power degradation models, oblateness and shadow. It exhibits better convergence than SEPSPOT in typical minimum-time LEO-GEO and GTO-GEO transfers. A COE correction scheme was developed by Ruggiero et al. [180], including coasting arcs but neglecting environmental perturbations. Gao's [64] employed three types of steering laws: perigee-centered tangential, apogee-centered inertial, and piecewise constant yaw. He derived analytic expressions for the averaged EOM in COE including shadow, coasting, and J_2. The weighting parameters were optimized using a deterministic algorithm for min-time and min-fuel transfers. In [144] Zuiani et al. proposed two-tangential control laws for planar transfers: perigee and apogee centered respectively. The parameters where optimized with a multi-objective GA with respect to the time of flight, total engine operation time, time within the radiation belt, and longest eclipse duration.

Hudson and Sheeres [65] represent each component of the thrust acceleration as a Fourier series (FS) in eccentric anomaly, and then average EOM in COE over one orbit to define a set of secular equations. The equations are a function of only 14 of the thrust FS coefficients, regardless of the order of the original Fourier series. Thus the continuous control is reduced to a set of 14 parameters. She solved a targeting problem using a least square method to solve for the unknown coefficients. Then, Ko and Sheeres [195] identified minimal sets of six FS parameters to represent the perturbing acceleration effectively, instead of 14. Given the initial and desired final orbital state, a set of six FS coefficients can be computed analytically, and the required control accelerations can be constructed to achieve any orbital maneuver. The method was demonstrated in [196] on two types of low-thrust spiral maneuvers: a repositioning maneuver in GEO and a maneuver to simultaneously change orbit radius and inclination. Results were successfully used as an initial guess for the STK optimization engine.

A different approach utilizes closed form feedback control laws derived from Lyapunov functions. For instance, Ilgen [69] developed a Lyapunov guidance law based on MEE. Gao [143] used it as an initial guess for his COV-based method. Petropoulos [70] presented the Proximity Quotient guidance law (Q-Law), which is expressed in terms of MEE, implements shadow and oblateness effects, and a coasting mechanism without averaging. A multi-objective GA was used to optimize the free parameters and was implemented in the tool GA-Qlaw. It proved to permit a rapid trade-off evaluation and to provide reasonable performance estimates for the preliminary design of planetocentric transfers [73, 74]. Additionally, it was integrated into the high-fidelity tool Mystic [45] to assist in generating starting guesses. Another well-known Lyapunov function was introduced and rigorously proved by Chang et al. [197]. The controller is expressed in CSV and was used by Betts [55] to generate initial guesses for a direct collocation scheme implemented in SOCS for transfers to GEO and Molniya orbits. Gurfil [198] developed a Lyapunov controller in terms of COE and used it to determine orbital transfer between elliptical orbits.

Some of the analyses may be described as shape-based, that is, the trajectory shape is directly assumed, with the requisite thrust computed a posteriori. Notably, the first shape-based method was the logarithmic spiral presented as early as 1950 by Forbes [199] and 1959 by Tsu [200] and Bacon [58]. A remarkable variant on the logarithmic spiral was given by Pinkham [201] and Lawden [202]. Pinkham's spiral can be used, for example, to escape from an initially circular orbit, or from any point on an elliptic orbit. Although Lawden's spiral was developed with transfer between two arbitrary states in mind, the spiral does not offer enough degrees of freedom to accomplish this. Therefore, despite the various analytic results available for the logarithmic spiral, the solution essentially has a constant flight path angle. In an attempt to correct these shortcomings, the exponential sinusoid was developed Petropoulos and Longuski [47], which has two parameters, apart from the scaling and phase parameters. Izzo [203] explored the potential of exponential sinusoids for solving the accelerated multi-revolution Lambert's problem. These early works are extensively reviewed by Petropoulos and Sims [36].

In Ref. [47], Petropoulos and Longuski apply a broad search algorithm with pruning criteria along with exponential sinusoids to generate candidate trajectories for GALLOP. The technique was implemented in the software STOUR-LTGA (Satellite Tour Design Program for Low-Thrust Gravity-Assist trajectories), which automatically searches for low-thrust, gravity-assis trajectories using a heuristic broad search algorithm. The user has to specify a sequence of gravity assist bodies, a range of launch dates, and a range of launch velocities for trajectories, subject to various constraints, such as time of flight and propellant consumption limits. They solved a rendezvous mission to Ceres via a Mars flyby, and a flyby mission to Jupiter via Venus-Earth-Mars flybys. However, the cost estimated by exponential sinusoid methods does not properly estimate the optimal value. It is due to the fact that neither coasting nor rendezvous phases have been included in the model. Vasile et al. [204] study the optimality of the exponential sinusoid and concludes that this model is far from satisfying the necessary condition of optimality.

Later works include Wall and Conway [48], who modeled the trajectory as an inverse polynomial with unbounded tangential thrust. The advantage of this approach compared to Petropoulos and Longuski's is the possibility to satisfy all boundary conditions. A GA was used in both works to select the unknown launch date, the time of flight, and the number of heliocentric revolutions to optimize a multi-rendezvous asteroid problem. Wall [49] extended their approach to three dimensional case by using cylindrical coordinates. De Pascale and Vasile [50], Novak and Vasile [51], Taheri and Abdelkhalik [52], and Gondelach and Noomen [53] created ingenious three-dimensional shape-based models incorporating pseudo-equinoctial elements, spherical coordinates, finite Fourier series, and hodographic shaping respectively. These approaches can handle boundary, time of flight and thrust constraints and were used to solve various rendezvous problems without intermediate flybys via grid search over the free parameters. In fact, the pseudo-equinoctial approach was implemented in the tool IMAGO [50] (Interplanetary Mission Analysis Global Optimization), an successfully used as initial guess for DITAN.

Previous methods, except for the hodographic method, assumed a tangential thrust. In order to improve the versatility of the solution, Roa et al. [205] found an entire new family of Generalized Logarithmic Spirals based on the thrust profile of the logarithmic spirals. Therefore, it is a planar shape with unbounder thrust levels. The flexibility of this approach was later improved by adding an additional degree of freedom in the solution [206] and modeling the transversal motion with a polynomial shaping approach [207]. By using a thrust-coast-thrust sequence for rendezvous legs, and thrust-coast sequence for flybys legs, he was able to solve a rendezvous problem to Ceres via Mars flyby. Recently, Roa et al. opted in [88] to use his shaped-based method together with a branch and prune algorithm for the direct exploration of the search space to generate as many candidate trajectories as possible for a multiple-flyby mission to Jupiter. However, in his approach he predefined the sequence of flybys and did not include coast arcs. Candidate trajectories were used as initial guesses for GALLOP.

The last class of predefined control laws explores artificial neurocontrollers. The tool InTrance (INtelligent spacecraft TRAjectory optimization using NeuroController Evolution) was designed by Dachwald [208] only for heliocentric single-phase trajectory optimization problems. InTrance was later extended by Carnelli et al. [146] to include intermediate gravity assisted maneuvers in InTrance-GA. Dynamics is expressed in terms of patched two-body problems, where the flybys are unpowered but not instantaneous. It implements an artificial neural network to act as neucontroller and combine it with evolutionary algorithms (a GA) to train the NC and to determine the optimal spacecraft steering strategy that minimizes the total transfer time. The targeting constraints are handled by penalizing the objective function. This combination is known as evolutionary neurocontrol. Results are presented for a Mercury rendezvous with a Venus gravity assist and for a Pluto flyby with a Jupiter gravity assist. Computing times were 11 hours for the former case and 6 hours for the latter scenario. They found a very good agreement with other software standards as IMAGO, GALLOP and DITAN.

2.5.5 Dynamic Programming Methods

Whiffen [42] presented the Static/Dynamic Control (SDC) algorithm, a class of Differential Dynamic Programming (DDP) method. The algorithm was implemented in the generic tools for high-fidelity trajectories Mystic. It implements multi-body dynamics and is able to naturally obtain the optimal sequence of flybys, including escape, capture phases. The tool itself can be seen as the state-of the art for the design of low-thrust trajectories and it has been successfully used to design NASA's cancelled Jupiter Icy Moon Orbiter (JIMO) and also to design and navigate the NASA's DAWN discovery mission to asteroid Vesta and Ceres. Results from this algorithm has been published in numerous papers, such as [209, 210]. However, Mystic uses a pure penalty method to account for the constrained violation, which may lead to unfeasible trajectories, slow convergence, or no convergence at all. Additionally, its application to solve multi-revolution planetocentric transfers is limited by its computation time to about 250 revolutions [42]. Last but not least, it requires a good initial guess to run.

A faster yet less accurate algorithm was presented by Lantoine and Russell [148] and implemented in the tool HDDP (Hybrid Differential Dynamic Programming). It is an extension of the classic DDP algorithm that combines DDP with some well-proven nonlinear mathematical programming techniques. It exploits second-order derivative information, and includes two options for the Dynamical modeling: the Stark model and the Kepler model. In [211], Lantoine and Russell presented a maximum final mass Earth-Mars rendezvous transfer and a 17 revolution minimum-fuel Earth-orbit transfer. Computational times were 60 sec for the former, and 20 min for the latter. A more appropriate method for handling high revolutions was developed by Aziz [147]. The proposed method discretize the trajectory in terms of MEE and the control schedule with respect to an orbit anomaly and perform the optimization with DDP. He included spherical gravity and third- body perturbations. He solved geocentric transfers up to 2,000 revolutions. He was able to generate a Pareto front trading time-off flight and propellant mass, by independent runs of his single-objective algorithm within a matter of hours.

3

Mathematical Framework

Contents

3.1 Introduction

Let a spacecraft be equipped with either a low-thrust engine, chemical propulsion or a combination of both, and/or subject to the possibility of performing gravity assisted maneuvers. The dynamics, and consequently the trajectory, can be modeled as a hybrid dynamical system, i.e., a system with interacting continuous and discrete dynamics. The continuous dynamics determines the trajectory during the thrusting and coasting phases of the electric engine. Each phase represents a different working condition and consequently a different continuous dynamical description of the system. The discrete dynamics characterizes the discontinuous behavior of the system such as the on/off switchings of the low-thrust engine, the instantaneous firings of the chemical engine or the effect of performing a gravitational slingshot (when considered instantaneous). This interconnection between discrete and continuous dynamics allows to formally pose the optimization problem as a Hybrid Optimal Control Problem (HOCP).

Unlike the classical Continuous Optimal Control Problem formulation introduced in chapter 2, the HOCP framework can be generally extended to any spacecraft trajectory optimization problems that include, not only the determination of the low-thrust control history, yet also decision-making or mission planning processes as part of the optimal solution. General frameworks for the description of HOCPs and its corresponding mathematical formalism have been presented, e.g., by Branicky et al. [212] and Buss et al. [213]. Particular frameworks for space mission planning have been proposed by Chilan and Conway [38] and Ross and D' Souza [39]. In this chapter, the general mathematical framework for hybrid dynamical systems is presented and the Multi-Objective HOCP based on the one proposed by Buss et al. [213] is formulated. Thereafter, two of the major elements of our proposed solution approach, i.e., genetic algorithms and the direct Hermite-Simpson transcription scheme will be described.

3.2 Hybrid Dynamical Systems

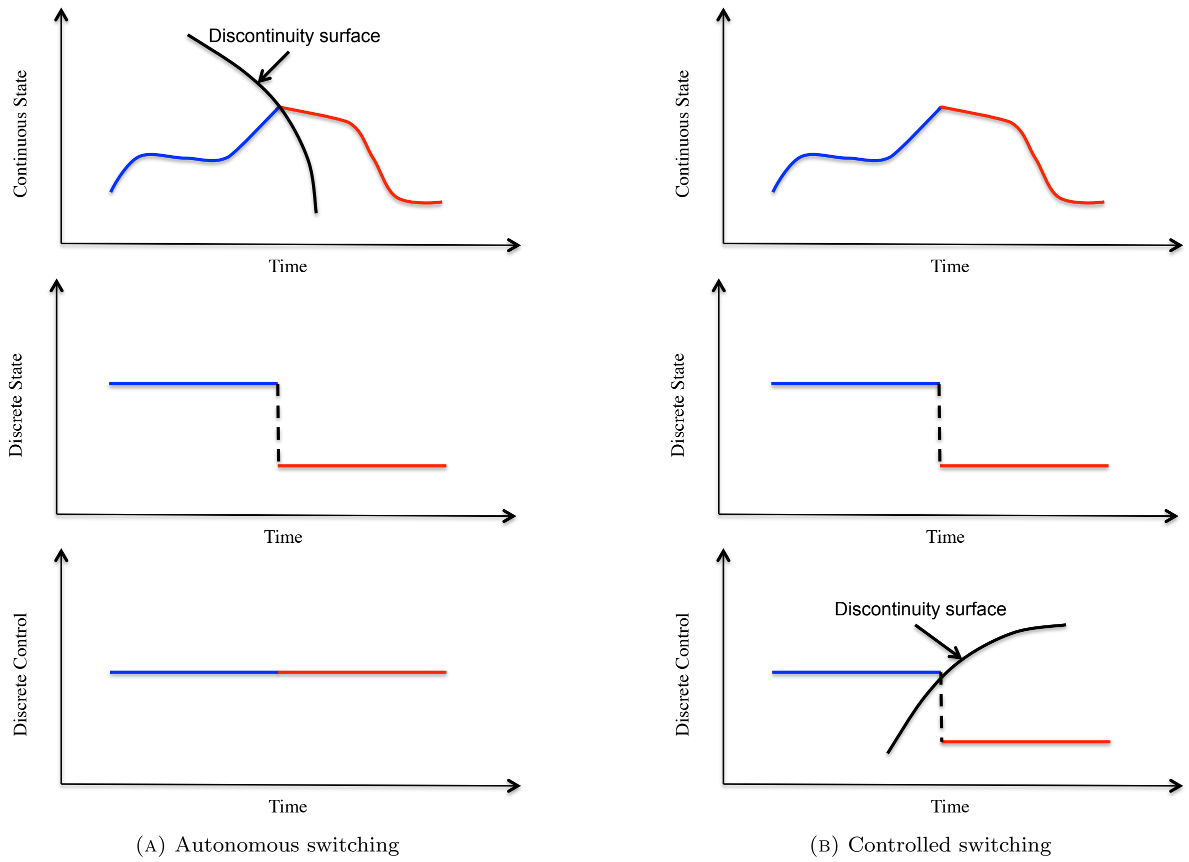
The state of a hybrid dynamical system is determined by the continuous state vector $\boldsymbol{x}(t) \in \mathcal{X} \subset \mathbb{R}^{n_x}$, which is constrained to be in the set $\mathcal{X}$ of permissible continuous states and the discrete state vector $\boldsymbol{q}(t) \in \mathcal{Q} \subset \mathbb{Z}^{n_q}$, which is constrained to be in the set $\mathcal{Q}$ of permissible discrete states. The system can be controlled by a continuous input vector $\boldsymbol{u}(t) \in \mathcal{U} \subset \mathbb{R}^{n_u}$, which belongs to the set $\mathcal{U}$ of permissible continuous controls, and by a discrete input vector $\boldsymbol{v}(t) \in \mathcal{V} \subset \mathbb{Z}^{n_v}$, which belongs to the set $\mathcal{V}$ of permissible discrete controls. Both input vectors[1] can be dynamical variables or static parameters depending on whether they are time-varying or time-independent respectively. Therefore, the evolution of the state vector with respect to the independent time variable $t \in \mathbb{R}$ is given by its hybrid dynamics as follows:

[1] Input vectors can be also termed as control vectors, control inputs, control variables, controls or decision variables

$$\dot{\boldsymbol{x}} = \mathrm{f}(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, \mathrm{t}) \qquad \text{if} \quad s_j(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, t) \neq 0, \qquad j = 1, \ldots, n_s. \tag{3.1}$$

$$[\boldsymbol{x}(t_i^+), \boldsymbol{q}(t_i^+)] = \phi_j(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, t_i^-) \qquad \text{if} \quad s_j(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, t_i^-) = 0, \qquad j \in \{1, \ldots, n_s\}\,. \tag{3.2}$$

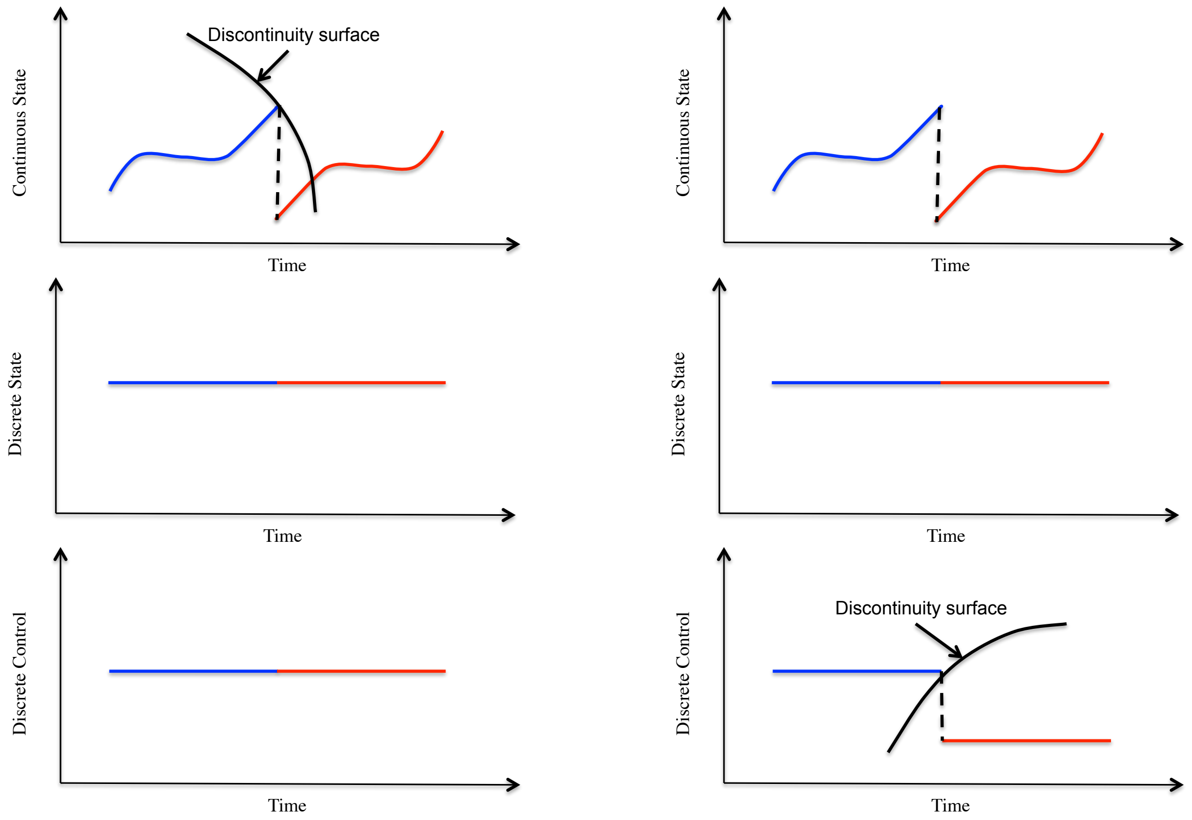
The continuous behavior of the hybrid dynamical system is described by the set of differentiable equation $\mathrm{f} : \mathcal{X} \times \mathcal{Q} \times \mathcal{U} \times \mathcal{V} \times \mathbb{R} \longrightarrow \mathbb{R}^{\mathrm{n_x}}$, whereas the discontinuous behavior is characterized by the set of discrete event functions, which includes the n_s discontinuity surfaces $s_j : \mathcal{X} \times \mathcal{Q} \times \mathcal{U} \times \mathcal{V} \times \mathbb{R} \longrightarrow \mathbb{R}$ and transition map functions $\phi_j : \mathcal{X} \times \mathcal{Q} \times \mathcal{U} \times \mathcal{V} \times \mathbb{R} \longrightarrow \mathcal{X} \times \mathcal{Q}$ for $j = 1, \ldots, n_s$. Discontinuity surfaces pose the condition that both state and controls must satisfy for a discrete event to be triggered. In case the discontinuous surface depends only on the state vector, it represents an autonomous event, whereas if it depends uniquely on the controls, it defines a controlled event.The times t_i at which these events occur, are called event transition times. The successor states $\boldsymbol{x}(t_i^+)$ and $\boldsymbol{q}(t_i^+)$ just after a discrete event is given by the transition map functions. In case only the discrete state is changed after a discrete event, it is called a switching event, whereas if it is the continuous state experience a discrete jump, it is known as impulsive event. As long as all discontinuity surfaces $s_j(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, t) \neq 0$ for $j = 1, \ldots, n_s$, the system trajectory evolves continuously according to Eq.(3.1).



(A) Autonomous switching

(B) Controlled switching

FIGURE 3.1: Illustration of switching discrete events



(A) Autonomous impulse

(B) Controlled impulse

Figure 3.2: Illustration of impulsive discrete events

Therefore, in a hybrid dynamical system, four basic types of discrete events can be found: autonomous switching, controlled switching, autonomous impulses, and controlled impulses [212]. Note that a general discrete event, as expressed in Eq.(3.5), would comprise a combination of all of them. As an example, let us consider a hybrid system defined by a continuous state $\boldsymbol{x}$, a discrete state $\boldsymbol{q}$, and a discrete control $\boldsymbol{v}$, and subject to Eqs.(3.1)-(3.5). Each type of discrete events have a different effect in the hybrid dynamics as it is illustrated in Figs. 3.1-3.2. Further discussion is provided in the following lines.

- **Autonomous switching**: An autonomous switching occurs when the continuous state trajectory crosses the discontinuity surface in the continuous state-time space (see Fig. 3.1a). In this case, the discontinuity surface depends only on the continuous state and on time, i.e., $s = s(\boldsymbol{x}, t)$. The switching causes the discrete state to change, whereas the continuous states before and after the switching are equal, i.e., $\boldsymbol{x}(t_i^+) = \boldsymbol{x}(t_i^-)$ and $\boldsymbol{q}(t_i^+) = \phi(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{v}, t_i^-)$. In the new discrete state, the continuous state trajectory follows different equation of motions than in the previous discrete state. In spacecraft systems, autonomous switching occurs, for example, when the electric engine is switched-off due to power availability constraints (e.g., the spacecraft crosses through the Earth-shadow or it is very far from the Sun).

- **Controlled switchings**: Controlled switching differs from autonomous switching in that the discontinuity surface is not a function of the continuous state but it depends on the controls, i.e., $s = s(\boldsymbol{v}, t)$. Therefore, the discrete event occurs in the control-time space (see Fig. 3.1b). Controlled switching models logical decisions that can be made at a desired point of time to change the system dynamics, e.g., switching-off the electric engine for propellant savings reasons.

- **Autonomous impulses**: An autonomous impulse resets the value of the continuous state, when the continuous state trajectory hits the discontinuity surface (see Fig. 3.2a). In a similar fashion than autonomous switching, the discontinuity surface depends only on the continuous state and on time, i.e., $s = s(\boldsymbol{x}, t)$. However, after an autonomous impulse, the discrete state and thus the differential equations remains unchanged, whereas the continuous state jumps according to the transition maps function, i.e. $\boldsymbol{x}(t_i^+) = \phi(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{v}, t_i^-)$ and $\boldsymbol{q}(t_i^+) = \boldsymbol{q}(t_i^-)$. Examples for autonomous impulses in spacecraft dynamics are gravity assisted-maneuvers, since a discrete change is the heliocentric velocity is experienced when it encounters a planet in space and time.

- **Controlled impulses**: The difference of controlled impulses to autonomous ones is that the impulse is triggered by a discontinuity surface that depends on the controls, i.e., $s = s(\boldsymbol{v}, t)$. Similarly to controlled switchings, the event occurs in the control-time space (see Fig. 3.2b). Incrementing the velocity of a spacecraft by an instantaneous firing of a chemical engine is an example of a controlled impulse.

3.3 Multi-Objective Hybrid Optimal Control Problem

The Multi-Objective HOCP is to find the set of feasible continuous $\boldsymbol{u}(t)$ and discrete $\boldsymbol{v}(t)$ control inputs belonging to the Optimal Pareto front that minimizes the multi-objective function $J(\boldsymbol{u}, \boldsymbol{v}, t)$, typically a vector-valued function of the hybrid system state, control and time:

$$\min J(\boldsymbol{u}, \boldsymbol{v}) = \mathcal{M} + \int_{t_0}^{t_f} \mathcal{L}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{q}, \boldsymbol{v}, t) dt, \tag{3.3}$$

subject to

$$\dot{\boldsymbol{x}} = \mathrm{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{q}, \boldsymbol{v}, t) \quad \text{if} \quad s_j(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{q}, \boldsymbol{v}, t) \neq 0, \quad j = 1, \dots, n_s, \tag{3.4}$$

$$\left[\boldsymbol{x}(t_i^+), \boldsymbol{q}(t_i^+)\right] = \phi_j(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{q}, \boldsymbol{v}, t_i^-) \quad \text{if} \quad s_j(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{q}, \boldsymbol{v}, t_i^-) = 0, \quad j \in \{1, \dots, n_s\}, \tag{3.5}$$

$$\boldsymbol{u}(t) \in \mathcal{U} \subset \mathbb{R}^{n_u}, \quad \boldsymbol{v}(t) \in \mathcal{V} \subset \mathbb{Z}^{n_v}, \quad \forall t \in [t_0, t_f], \tag{3.6}$$

$$\boldsymbol{x}(t) \in \mathcal{X} \subset \mathbb{R}^{n_x}, \quad \boldsymbol{q}(t) \in \mathcal{Q} \subset \mathbb{Z}^{n_q}, \quad \forall t \in [t_0, t_f], \tag{3.7}$$

$$0 \leq \mathrm{g}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{q}, \boldsymbol{v}, t), \quad t \in [t_0, t_f], \tag{3.8}$$

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, t_0), \qquad \boldsymbol{q}(t_0) = \boldsymbol{q}_0(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, t_0) \tag{3.9}$$

$$\boldsymbol{x}(t_f) = \boldsymbol{x}_f(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, t_f), \qquad \boldsymbol{q}(t_f) = \boldsymbol{q}_f(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, t_f) \tag{3.10}$$

In the above, the Lagrange integrand term $\mathcal{L} : \mathcal{X} \times \mathcal{Q} \times \mathcal{U} \times \mathcal{V} \times \mathbb{R} \longrightarrow \mathbb{R}^{n_j}$ is a vector real-valued function of the state and control variables and of time, and n_j is the number of objective functions. The Mayer type part $\mathcal{M} : \mathcal{X} \times \mathcal{Q} \times \mathbb{R} \longrightarrow \mathbb{R}^{n_k}$ is a general vector function of the event transition times t_i for $i = 0, \ldots, N$ and of the continuous $\boldsymbol{x}(t_i^-)$ and the discrete $\boldsymbol{q}(t_i^-)$ states just before the discrete events and the continuous $\boldsymbol{x}(t_i^+)$ and the discrete $\boldsymbol{q}(t_i^+)$ states just after the discrete events. Thus, it is expressed as:

$$\mathcal{M} := \mathcal{M}\left(\boldsymbol{x}(t_0^+), \ldots, \boldsymbol{x}(t_N^-); \boldsymbol{q}(t_0^+), \ldots, \boldsymbol{q}(t_N^-); t_0, \ldots, t_N\right) \tag{3.11}$$

Here, t_0 and $t_N = t_f$ are the beginning and final times, which are associated to an initial and final event function respectively, whereas the remaining $N - 1$ transition times are related to interior event functions. The minimization of the multi-objective function in Eq.(3.3) is subject to initial and terminal conditions on the state vector (3.9)-(3.10), admissible values for the continuous and discrete control and state variables (3.6)-(3.7) and further inequality constraints (3.8) given by the function $\mathrm{g} : \mathcal{X} \times \mathcal{Q} \times \mathcal{U} \times \mathcal{V} \times \mathbb{R} \longrightarrow \mathbb{R}^{n_g}$. Obviously, valid hybrid optimal trajectories must obey both the continuous and discrete dynamics (3.4)-(3.5). Let us define the optimal sequence of discrete events as:

$$\sigma = [(t_1, s_k), \ldots, (t_i, s_j), \ldots, (t_N, s_m)], \quad \text{for} \quad k, j, m \in 1, \ldots, n_s \tag{3.12}$$

The key challenge when solving HOCPs is that the optimal sequence of discrete events σ is not known a-priori. Therefore, it has to be determined as part of the solution. Note that, in Eq.(3.12) the sequence of discontinuity functions may have an arbitrary order, and even a discontinuity function can be activated more than once during the trajectory, unless otherwise specified, thus increasing the combinatorial complexity of the problem. Additionally, when facing multi-objective problems, instead of searching for a unique optimal law for the continuos and discrete control inputs as in single objective optimization, the aim is to obtain a whole set of different solutions that are equally optimal in terms of Pareto efficiency.

As an illustration, let us define the HOCP where a spacecraft is to travel from Earth to Saturn benefiting from as many gravity assisted maneuvers as desired and limited to a maximum time-of-flight. The patched conics approach is used and flybys are considered instantaneous, i.e., as discrete events. In such case, there are nine discontinuity functions, i.e., $(s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9)$ representing a planetary encounter with Mercury, Venus, Earth, Mars, Jupiter, Saturn, Neptune, Uranus and Pluto respectively. Multi-objective solutions with respect to propellant mass and flight of time are to be obtained. In this case an optimal compromise sequence of gravity assists σ_1 is obtained, such that:

$$\sigma_1 = [(t_1, s_3), (t_2, s_2), (t_3, s_3), (t_4, s_5)] \tag{3.13}$$

where t_1, t_2, t_3, t_4 represent the optimal flyby maneuver times of the sequence Earth-Venus-Earth-Jupiter. A different compromise solution would result in a different optimal sequence.

3.4 Hermite-Simpson Collocation Scheme

Hermite-Simpson method is a basic form of collocation [126] that may be used to transcribe multiphase OCPs and continuous OCPs to a standard NLP problem. Multiphase OCP are simplified cases of Hybrid Optimal Control Problems (HOCP), where the number and sequence of active discrete events σ is known a-priori. Each discrete event occurs at the discrete event time t_k, which may be known or free. The time interval between two consecutive discrete events defines a phase, i.e. $[t_k, t_{k+1}]$. If all the controls involved are continuous, the resulting NLP is continuous. Conversely, if any control input is integer, the Hermite-Simpson transcription results in a Mixed Integer NLP. For the sake of clarity, only continuous state and control variables will be considered, as it is the case of this thesis. The detailed procedure can be derived as follows. The time interval for each phase $[t_k, t_{k+1}]$ is subdivided into n subintervals $[t_i, t_{i+1}]$ for $i = 1, \ldots, n$. It holds that $t_1 = t_k$, and $t_{n+1} = t_{k+1}$. Let the continuos state $\boldsymbol{x}(t)$ of the system be approximated on each segment $[t_i, t_{i+1}]$ by a cubic polynomial of the form:

$$\boldsymbol{x}(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{3.14}$$

whose time derivative can be represented by the second-order polynomial:

$$\dot{\boldsymbol{x}}(t) = a_1 + 2a_2 t + 3a_3 t^2 \tag{3.15}$$

where (a_0, a_1, a_2, a_3) are the coefficients of the polynomial. Let the continuos control $\boldsymbol{u}(t)$ of the system be approximated on each segment $[t_i, t_{i+1}]$ by a linear segment of the form:

$$\dot{\boldsymbol{u}}(t) = b_0 + b_1 t \tag{3.16}$$

where (b_0, b_1) are the coefficients of the segment. The parameters representing the state and controls at the endpoints $\boldsymbol{x}_i = \boldsymbol{x}(t_i)$, $\boldsymbol{u}_i = \boldsymbol{u}(t_i)$, $\boldsymbol{x}_{i+1} = \boldsymbol{x}(t_{i+1})$, and $\boldsymbol{u}_{i+1} = \boldsymbol{u}(t_{i+1})$ are assumed to be the unknowns of the NLP problem. They are denoted as nodes. Knowing them implies that $\mathrm{f}_i = f(\boldsymbol{x}_i, \boldsymbol{u}_i, t_i)$ and $\mathrm{f}_{i+1} = f(\boldsymbol{x}_{i+1}, \boldsymbol{u}_{i+1}, t_{i+1})$ are also known.

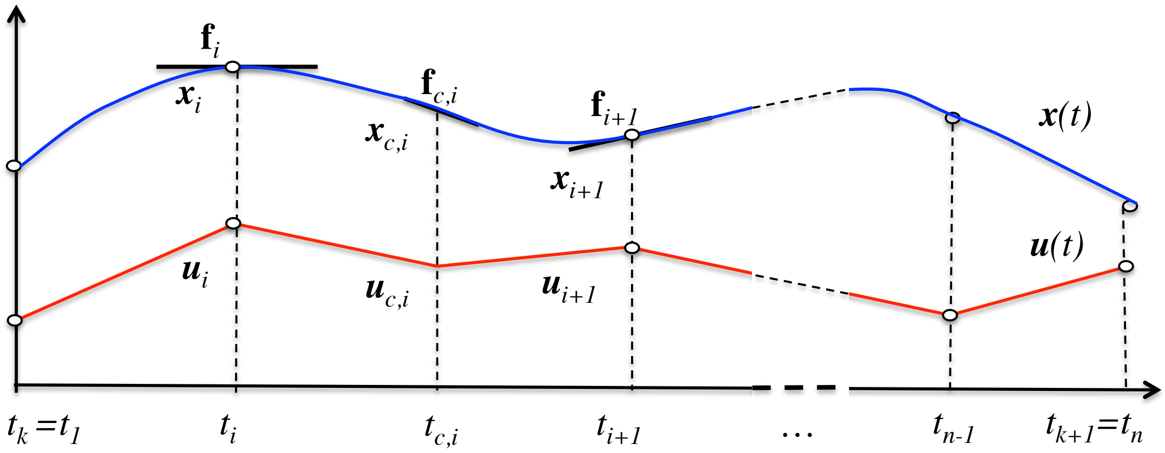


FIGURE 3.3: Hermite-Simpson collocation scheme illustration

Let $[\boldsymbol{x}_{i,C}, \boldsymbol{u}_{i,C}]$ be the state and control at t_c, the middle point of $[t_i, t_{i+1}]$. They are called "collocation points" and their numerical expression are:

$$\boldsymbol{x}_{i,C} = \frac{1}{2}(\boldsymbol{x}_i + \boldsymbol{x}_{i+1}) + \frac{\Delta t_i}{8}(\ \mathrm{f}_i -\ \mathrm{f}_{i+1}), \tag{3.17}$$

for the state collocation point and

$$u_c = \frac{\boldsymbol{u}_i + \boldsymbol{u}_{i+1}}{2} \tag{3.18}$$

for the control collocation point. Here, $\Delta t_i = t_{i+1} - t_i$. The time derivative of the state collocation point is:

$$\dot{\boldsymbol{x}}_{i,C} = -\frac{3}{2\Delta t_i}(\boldsymbol{x}_i - \boldsymbol{x}_{i+1}) - \frac{1}{4}(\ \mathrm{f}_i +\ \mathrm{f}_{i+1}), \tag{3.19}$$

The difference between the interpolated and calculated derivatives at the collocation point, i.e. $\dot{\boldsymbol{x}}_{i,C} - \mathrm{f}(\boldsymbol{x}_{i,C}, \boldsymbol{u}_{i,C})$, defines the Simpson's system defect constraint

$$\mathrm{d_i}(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}_\mathrm{i+1}, \boldsymbol{u}_\mathrm{i}, \boldsymbol{u}_\mathrm{i+1}) = \boldsymbol{x}_\mathrm{i} - \boldsymbol{x}_\mathrm{i+1} + \frac{\Delta \mathrm{t_i}}{6}(\ \mathrm{f_i} + 4\ \mathrm{f}(\boldsymbol{x}_\mathrm{i,C}, \boldsymbol{u}_\mathrm{i,C}) +\ \mathrm{f_{i+1}}) \tag{3.20}$$

These constraints are known as Hermite-Simpson defect constraints. The NLP solver will select $(\boldsymbol{x}_i, \boldsymbol{u}_i, \boldsymbol{u}_{i+1}, \boldsymbol{x}_{i+1})$ to drive the defect constraint to zero. In this way the interpolating polynomial will approximate the true dynamics within the accuracy of the numerical integration. Additionally, the states $\boldsymbol{x}(t_k)$ and controls $\boldsymbol{u}(t_k)$ at the endpoints of each phase t_k must satisfy the event transition equation of Eq.(3.5), which can be expressed as a set of non-linear constraint. For instance, this method can be applied to transcribe and HOCP when the sequence of flybys is known.The reader is referred to [126] for more details.

3.5 Genetic Algorithm

A genetic algorithm (GA) is a stochastic optimization technique for solving NLPs based on the biological principles of Darwinian evolution [214]. GAs incorporates probabilistic transition rules on a population and are capable of optimizing problems without gradient information, such as combinatorial or mixed-integer nonlinear problems (those combining continuous and discrete variables). Unlike gradient-based methods, GAs were developed as a framework for global searches of the design space. That is, the candidate solutions are not confined to the locally optimal solution in the vicinity of the initial guess. Moreover, GAs does not require a user-defined initial guess to start the optimization procedure and can be initiated without any prior knowledge of the design space. For many problems, especially low-thrust trajectory optimization problems, developing a suitable initial guess is an overwhelming difficult task. In GAs, the initial population can be generated randomly by using a uniform random number generator, which ensures a broad initial sampling of the solution space. Last but not least, GAs can operate both over single-objective or multi-objective design spaces, providing a set of optimal Pareto optimal solutions in just one run of the algorithm.

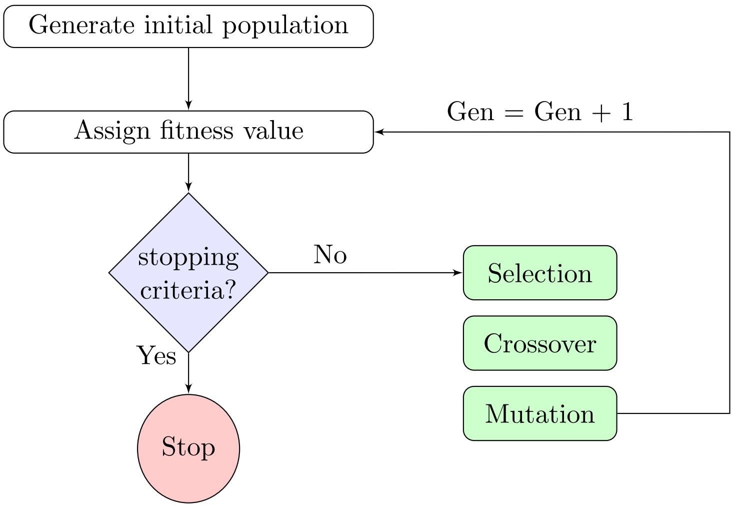


FIGURE 3.4: Genetic algorithm flowchart

Nevertheless, the GA is computationally expensive. It often requires many function evaluations to achieve a good solution. For purely continuous problems, or problems in which a local search is acceptable, classical gradient-based techniques can be significantly more efficient. The latter typically require fewer function evaluations because of their use of gradient information to perform the iterative search. Furthermore, because GAs do not use gradient information to guide the search, there is no proof of convergence such as the Karush-Kuhn-Tucker conditions for gradient-based methods and it may be difficult to meet complex constraints. In fact, GAs were originally designed for unconstrained problems, and thus may not be effective on tightly constrained problems with small feasible regions [137]. Therefore, GAs are a powerful technique for difficult yet unconstrained optimization problems where the design space is broad, multi-objective, mixed-integer, discontinuous or non-intuitive. Notably, they have found wide acceptance for the conceptual or preliminary design of space missions.

GAs incorporate operators that mimic natural selection and reproduction using a probabilistic search. More specifically, They operate with an entire population of designs and incorporate probabilistic transition rules executed by three key genetic operators: selection, crossover, and mutation. The algorithm starts with a random or user-provided initial population and iteratively evolve it, construct a new, and hopefully better population with each successive generation, until a stopping criteria is met (typically a maximum number of generations). The location and sequence of the core operators in the algorithm are illustrated in the flowchart of a simple GA depicted in Figure 3.4. These operators have to be defined in such a way that solutions whose objective values (or fitness value in GA terminology) are close to the real Pareto frontier should be selected to generate the next population and to ensure diversity, i.e., the obtained subset of Pareto solutions should distribute uniformly over the real Pareto frontier. In addition to the three standard operators, there are auxiliary operators that may be incorporate to model other genetic phenomena, or to improve convergence properties [137].

One of the most known and effective multi-objective evolutionary algorithms is the Non-dominated Sorting Genetic Algorithm II (NSGA-II). The NSGA-II attempts to achieve broad coverage of the Pareto front by emphasizing the designs that are closest to the Pareto front [86]. The fitness value of an individual in the population is based on the number of solutions that dominates it, which is known as non-dominated sorting. Because the designs that are closer to the Pareto front are associated with a lower fitness value and, the evolution of the population is biased towards the Pareto front. Furthermore, the NSGA-II includes a strategy to develop a wide set of solutions along the Pareto front by affording preference to less crowded designs in two different locations within the algorithm. Initially, the parent population is created randomly, consistent with the standard GA, and then the GA operators apply. Hereafter, a conceptual description of the selection, crossover and mutation operators is provided:

- **Selection**: This operator determines which individuals (also known as genes) from the population will survive to form the "parents" of the next generation. Individuals with a lower fitness value are more likely to survive. Several methods of selection exist but one of the most effective is tournament selection [215]. In tournament selection, two individuals are randomly selected from the population to compete against each other. The fitness values of the competing designs determine which individual wins the tournament and is placed in a parent pool. Those two individuals are then set aside and the tournament is continued until all designs in the population have competed, and the winners placed in the parent pool. Additionally, the NSGA-II algorithm also distinguishes between designs with the same fitness by sorting each front in the population in terms of crowding distance.

- **Crossover**: Whereas selection determines which individuals should reproduce, crossover creates new designs to explore the design space. To form new combinations, the crossover operator mates individuals from the set of parents to produce offspring. This process allows the individuals from the parent pool to be passed on to a new generation while generating new patterns that may be advantageous. The NSGA-II algorithm implements uniform crossover, which is very effective for the discovery of new patterns by producing diverse offspring [216]. Like most other crossover methods, uniform crossover begins with a random selection of two parents from the parent pool created by selection. The two parents are then mated to produce two children (two new individuals) that will comprise part of the next generation. Once parents have mated, they are discarded and two new parents are selected at random from the parent pool to generate additional offspring.

- **Mutation**: After crossover, the mutation operator encourages diversity by randomly altering some of the variables of each individual of the newly created population [86]. Such an operator introduces new variations into the gene, which highly increases the capacity of exploring non-intuitive parts of the design space. Mutation typically occurs at a low probability rate, randomly switching a small percentage of variables.

It is worth-mentioning that equality and inequality constraints can add significant complexity to the optimization problem, and several different techniques for the accommodation of constraints in GAs have been devised. The basic genetic operators are not formulated to explicitly manage constraints and thus, most popular constraint handling methods involve penalizing the fitness value corresponding to infeasible designs. By application of the penalty method, the problem is transformed into one that is unconstrained. Hence, the fitness function becomes a combination of the objective function and a penalty function. The genetic operators previously mentioned apply over the new fitness value.

4

MOLTO-OR: Orbit Raising

Contents

4.1 Introduction

In this chapter, we present a solution approach for a spacecraft that has to travel from a given departure orbit to a a selected target orbit within the Earth's gravitational field by means of its onboard propulsion system. This scenario well fits the case of telecommunication satellites that are injected into GTO and has to transfer to GEO. The satellite may perform a fully-chemical transfer (FCT), fully-electric transfer (FET) or a combined chemical-electric transfer (CCET). It may be equipped with any propulsive system defined on a list of available options. Operational constraints to arrive at a certain slot in the target orbit, the avoidance of the geostationary ring during the transfer or slew rate limitations may apply. The goal is to efficiently and simultaneously explore such design space. As described in chapter 1, efficient means that a good compromise between computational time and accuracy is met. Simultaneous means that the optimization of each transfer case with propulsive options is performed with the same algorithm and at the same time, avoiding the need to run them independently.

The aforementioned problem is formulated as a Multi-Objective Hybrid Optimal Control Problem (MO-HOCP). Chemical engine firings and the on-off switchings of the electric engine, if required, are modeled as discontinuous events. Conversely the continuous dynamics is represented by the geocentric coasting and thrusting arcs. The interconnection between continuous and discrete dynamics is represented in Figure 4.1 as a finite state machine diagram. A solution algorithm termed MOLTO-OR (Multi-Objective Low Thrust Trajectory Optimizer for Orbit Raising) is proposed. It consists on two sequential steps: MOLTO-OR Step 1 and MOLTO-IT Step 2. Each of them solves the problem at different degrees of fidelity and with different computational performances: the former provides a low-fidelity solution with low-computational effort, whereas the latter produces medium-fidelity trajectories that require more computational time. Both steps are interconnected since the solution from MOLTO-OR Step 1 is used as an initial guess for MOLTO-OR Step 2. Hereafter, the modeling of the hybrid dynamics, along with the solution approach applied at each stage will be introduced separately.

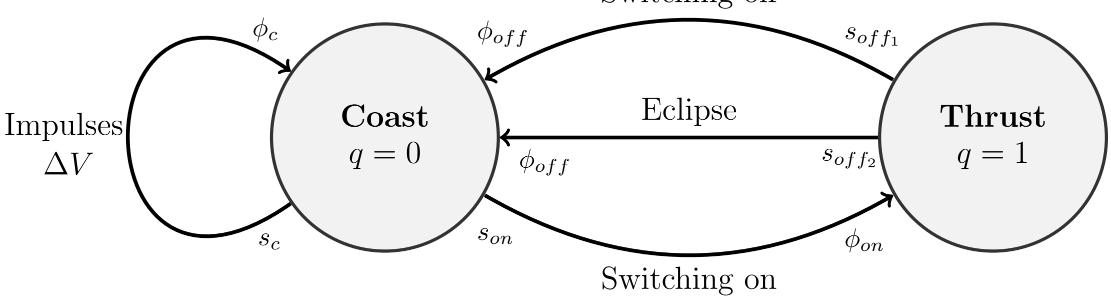


Figure 4.1: MOLTO-OR: Spacecraft Hybrid Dynamical System Diagram

4.2 Modeling

4.2.1 MOLTO-OR Step1

The continuous state vector of the spacecraft is determined by the set of modified equinoctial elements (p, f, g, h, k, L). Here, the true longitude L is treated as the independent variable, instead of the physical time t. Additionally, the evolution of the spacecraft mass m is needed to fully describe its dynamics. Consequently, the set of continuous state variables results in $\boldsymbol{x} = [p, f, g, h, k, t, m]$. The working condition of the electric propulsion system is determined by the discrete state variable $q \in \{0, 1\}$, where '0' designates the coasting mode and '1' indicates the thrusting mode.

The spacecraft is controlled both by acting with the electric or chemical engine. The former produces a continuous thrust force when switched on. Its steering law and the on/off switchings are determined by $\mathbf{u}(t) = [W_p, W_f, W_g, W_h, W_k, \eta_{a,th}, \eta_{r,th}, m_Q, n_Q, r_Q]$. They are a set of continuous static controls and will be described through this section. In particular, they determine the orientation angles α and β. Here, α is the azimuth angle measured in the orbit plane from the circumferential direction and positive away from the gravitational centre, whereas β is the declination angle measured out of the orbital plane and positive along the angular momentum. The chemical engine provides n-instantaneous velocity changes on the spacecraft. They are determined by the set of continuous static controls $\boldsymbol{u}_{c,j} = [\vartheta_j, \Delta V_j, \bar{\alpha}_j, \bar{\beta}_j]$ for $j = 1, \ldots, n$, where ϑ_j represents the true longitude at which the j^{th} impulse of magnitude ΔV_j and direction $\bar{\alpha}_j$ and $\bar{\beta}_j$ is performed. Here $\bar{\alpha}_j$ is the in-plane angle measured from the tangential direction and positive away from the gravitational centre and $\bar{\beta}_j$ is the declination angle. An illustration of the geometry of the problem is displayed in Figure 4.3.

The transfer type: FCT, FET or CCET is selected by the discrete static control $\Theta \in \mathbb{Z}$, whereas the propulsive system is selected by the static control $\Pi \in \mathbb{Z}$. The simultaneous coexistence of a chemical and a electric module creates some constraints to the system. Throughout this work, and much of the literature, it is assumed that the chemical firings are executed fully before the electric phase begins. This constraint arises because the high amount of power required by the electric propulsion may require deployed solar array panels [217]. The large appendages of the solar panels, when deployed, may suffer instabilities due to the high forces applied by the chemical propulsion.

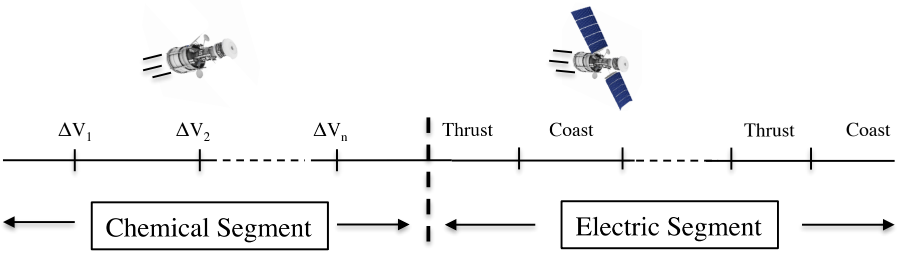


Figure 4.2: MOLTO-OR: Transfer Sequence

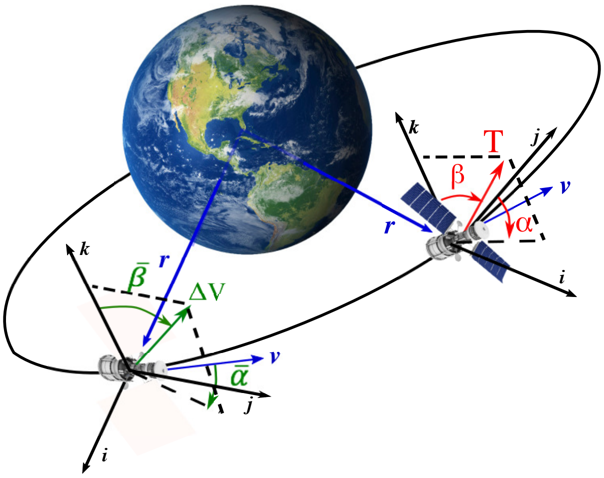


FIGURE 4.3: Chemical Burns (left) and Low-Thrust (right) steering angles

Continuous Dynamics

The continuous evolution of the continuous state vector of the system with respect to the true longitude is given as:

$$\mathrm{f}: \begin{cases} \dfrac{dp}{dL} = \dfrac{1}{\Gamma}\sqrt{\dfrac{p}{\mu}}\dfrac{2p}{w}\mathrm{a}_\theta \\ \dfrac{df}{dL} = \dfrac{1}{\Gamma}\sqrt{\dfrac{p}{\mu}}\left(\sin L a_r + \dfrac{1}{w}\left\{(w+1)\cos L + f\right\} a_\theta - \dfrac{g}{w}\left\{h\sin L - k\cos L\right\} \mathrm{a_h}\right) \\ \dfrac{dg}{dL} = \dfrac{1}{\Gamma}\sqrt{\dfrac{p}{\mu}}\left(-\cos L a_r + \dfrac{1}{w}\left\{(w+1)\sin L + g\right\} a_\theta + \dfrac{f}{w}\left\{h\sin L - k\cos L\right\} \mathrm{a_z}\right) \\ \dfrac{dh}{dL} = \dfrac{1}{\Gamma}\sqrt{\dfrac{p}{\mu}}\dfrac{l^2\cos L}{2w}\mathrm{a_h} \\ \dfrac{dk}{dL} = \dfrac{1}{\Gamma}\sqrt{\dfrac{p}{\mu}}\dfrac{l^2\sin L}{2w}\mathrm{a_h} \\ \dfrac{dt}{dL} = \dfrac{1}{\Gamma} \\ \dfrac{dm}{dL} = -\dfrac{q}{\Gamma}\dot{m} \end{cases} \tag{4.1}$$

where the auxiliary parameters are defined as:

$$\Gamma = \sqrt{\frac{p}{\mu}}\frac{1}{w}\left\{h\sin L - k\cos L\right\}\mathrm{a}_h + \sqrt{\mu p}\left(\frac{w}{p}\right)^2 \tag{4.2}$$

$$w = 1 + f\cos L + g\sin L \tag{4.3}$$

$$l^2 = 1 + h^2 + k^2 \tag{4.4}$$

$$\chi^2 = h^2 - k^2 \tag{4.5}$$

In the above, μ represents the gravitational constant of the Earth , $\dot{m}$ is the propellant consumption rate of the electric engine and $(\mathrm{a_r}, \mathrm{a}_\theta, \mathrm{a_h})$ are the components of the perturbing acceleration $\mathbf{a}_p$ projected onto the rotating radial frame, whose unitary vectors are:

$$\mathbf{i} = \frac{\mathbf{r}}{\mathrm{r}}, \qquad \mathbf{k} = \frac{\mathbf{r}\times\mathbf{v}}{\mathrm{rv}}, \qquad \mathbf{j} = \mathbf{k}\times\mathbf{i} \tag{4.6}$$

Here, $\mathbf{r} \in \mathbb{R}^3$ is the cartesian position vector, $\mathbf{v} \in \mathbb{R}^3$ is the cartesian velocity vector, $\mathrm{r} = ||\mathbf{r}||$ is the radial distance and $\mathrm{v} = ||\mathbf{v}||$ is the magnitude of the velocity. Expressions to convert from the modified equinoctial elements to the cartesian state vector are as follows:

$$\mathbf{r} = \begin{bmatrix} \mathrm{r}/l^2(\cos L + \chi^2 \cos L + 2hk \sin L) \\ \mathrm{r}/l^2(\sin L - \chi^2 \sin L + 2hk \cos L) \\ 2\mathrm{r}/l^2(h \sin L + k \cos L) \end{bmatrix} \tag{4.7}$$

$$\mathbf{v} = \begin{bmatrix} -1/l^2\sqrt{\mu/p}(\sin L + \chi^2 \sin L - 2hk \cos L + g - 2fhk + \chi^2 g) \\ -1/l^2\sqrt{\mu/p}(-\cos L + \chi^2 \cos L + 2hk \cos L - f + 2ghk + \chi^2 f) \\ 2/l^2\sqrt{\mu/p}(h \cos L + k \sin L + fh + gk) \end{bmatrix} \tag{4.8}$$

For the purpose of this thesis, the only non-two-body accelerations are due to Earth J_2 and thrust. Adding different perturbations is as simple as converting them to the rotating radial frame and including them in $\mathbf{a}_p$. The Earth oblateness is accounted in a similar manner to [55]:

$$\mathbf{a_{J_2}} = \begin{bmatrix} -3J_2/(2\mathrm{r}^4)(1 - 12(h \sin L - k \cos L)^2/l^4) \\ -12J_2/\mathrm{r}^4(h \sin L - k \cos L)(h \cos L + k \sin L)/l^4 \\ -6J_2/\mathrm{r}^4(h \sin L - k \cos L)(1 - h^2 - k^2)/l^4 \end{bmatrix} \tag{4.9}$$

where $J_2 = 0.0010826$.

The perturbing acceleration due to thrust $\mathbf{a}_T$ and the mass flow rate $\dot{m}$ used in Eqs.(4.1) have different expressions depending on the discrete state of the spacecraft. Assuming that the thrust produced by the electric engine is T, it holds that:

$$\mathbf{a}_T, \dot{m} : \begin{cases} \mathbf{a}_T = \dfrac{T}{m}\mathbf{d}, \quad \dot{m} = -\dfrac{T}{g_0 I_{sp}} & \text{if} \quad q = \mathit{1} \\ \mathbf{a}_T = 0, \quad \dot{m} = 0 & \text{if} \quad q = \mathit{0} \end{cases} \tag{4.10}$$

Here, I_{sp} is the specific impulse and g_0 is the Earth's gravitational acceleration at sea level. The unitary vector $\mathbf{d}$ points toward the direction of the thrust vector and can be projected onto the rotating orbital frame and is expressed as a function of the steering angles as follows:

$$\mathbf{d} = \cos\beta \sin\alpha \, \mathbf{i} + \cos\beta \cos\alpha \, \mathbf{j} + \sin\beta \, \mathbf{k} \tag{4.11}$$

The thrust magnitude T of the electric engine and its fuel consumption rate $\dot{m}$ depends on the specific propulsion system. Here, it is expressed as function of the power available to the engine P_a and its efficiency η as follows:

$$T = \frac{2\eta P_a}{g_0 I_{sp}} \tag{4.12}$$

It is assumed that P_a is equal to power generated by the solar arrays and constant during sunlight. When the spacecraft transverses the Earth shadow, the solar arrays do not generate any power. It is also considered that the solar arrays are degraded by the absorption of particles within the Van Allen radiation belts. Their models will be presented later in this section.

The thrust vector azimuth and declination angles, α and β respectively, are obtained using a Lyapunov feedback control method called the Q-law. It was originally proposed by Petropoulos [70] using orbital elements, yet in this work we use the formulation presented by Gávor and José M. in [74] with modified equinoctial elements. It solves the orbit transfer in an inverse square field where there is no constraints on the final true anomaly. The Q-law is based on a proximity quotient, Q, which captures the interdependencies between the orbital elements by means of scaling functions that quantify the proximity of the osculating orbit to the target orbit. During the transfer at each instant the Q-law method chooses the thrust angles that reduce the Q value the most quickly. A coasting mechanism that is based on variable effectivity of the thrust in reducing Q at different true anomalies is incorporated. The modified Lyapunov function, or Q function, is defined as:

$$Q = \sum_{œ} S_{œ} W_{œ} \left(\frac{œ - œ_t}{\dot{œ}_{xx}} \right)^2 \tag{4.13}$$

Using the semi-major axis as the first variable instead of the semilatus rectum proved to yield a better control when using the equinoctial orbital elements to formulate the Q-law. However, for the propagation of the orbit, the latter is used (p), as the right hand side of the differential equation is less expensive to evaluate. In Eq.(4.13) œ are the current, while $œ_t$ are the desired orbital elements, whereas $\dot{œ}_{xx}$ are the maximum rate of change of the corresponding variable over the thrust direction and true anomaly on the osculating orbit. The remaining terms are $W_{œ}$ the scalar weighting factors for each of the equinoctial orbital elements, and $S_{œ}$ scaling factor. Here, $S_{œ} = 1$ for $œ = f, g, h, k$ and

$$S_a = \left[1 + \left(\frac{|a - a_t|}{m_Q a_t} \right)^{n_Q} \right]^{1/r_Q} \tag{4.14}$$

The latter has been introduced to prevent convergence to $a = \infty$, since for $a = \infty$ all the $\dot{œ}_{xx}$ tend as well to an infinity value. Here, the set (m_Q, n_Q, r_Q) are named the scaling factors. The classical Q-law supplies these values analytically, however they are expensive to evaluate due to their complexity and trigonometric expressions. In this formulation analytical expressions are found for a, h, and k, yet not for f and g in closed form. However a moderately good approximation is used for these values to avoid computing them numerically. The maximum rate of change of the MEE are computed as follows:

$$\dot{a}_{xx} = 2a\frac{T}{m}\sqrt{\frac{a}{\mu}}\sqrt{\frac{1+\sqrt{f^2+g^2}}{1-\sqrt{f^2+g^2}}} \tag{4.15}$$

$$\dot{f}_{xx} \approx 2\frac{T}{m}\sqrt{\frac{p}{\mu}} \tag{4.16}$$

$$\dot{g}_{xx} \approx 2\frac{T}{m}\sqrt{p\mu} \tag{4.17}$$

$$\dot{h}_{xx} = \frac{1}{2}\frac{T}{m}\sqrt{p\mu}\frac{s^2}{\sqrt{1-g^2}+f} \tag{4.18}$$

$$\dot{k}_{xx} = \frac{1}{2}\frac{T}{m}\sqrt{p\mu}\frac{s^2}{\sqrt{1-f^2}+g} \tag{4.19}$$

The time derivative of Q can be determined as:

$$\frac{dQ}{dt} = \sum_{\oe} \frac{\partial Q}{\partial \oe} \dot{\oe}, \quad \oe = a, f, g, h, k \tag{4.20}$$

The Q-law method is based on choosing the thrusting angles α and β at each instant during the transfer such that $\dot{Q}$ is the most negative, therefore ensuring the most rapid decrease of the“distance” Q to the target orbit. These angles can be computed analytically, as $\dot{Q}$ can be rewritten as:

$$\frac{dQ}{dt} = D_1 \cos\beta \cos\alpha + D_2 \cos\beta \sin\alpha + D_3 \sin\beta \tag{4.21}$$

where D_1, D_2 and D_3 are parameters obtained from the derivatives present in Eq.(4.20) and can be calculated as:

$$D_1 = \sum_{\oe} \frac{\partial Q}{\partial \oe} \frac{\partial \dot{\oe}}{\partial a_t} \tag{4.22}$$

$$D_2 = \sum_{\oe} \frac{\partial Q}{\partial \oe} \frac{\partial \dot{\oe}}{\partial a_r} \tag{4.23}$$

$$D_3 = \sum_{\oe} \frac{\partial Q}{\partial \oe} \frac{\partial \dot{\oe}}{\partial a_n} \tag{4.24}$$

To obtain the optimal thrusting angles, Eq.(4.21) has to be differentiated with respect to α and β, which results in:

$$\frac{\partial \dot{Q}}{\partial \alpha} = -D_1 \cos\beta \sin\alpha + D_2 \cos\beta \cos\alpha \tag{4.25}$$

$$\frac{\partial \dot{Q}}{\partial \beta} = -D_1 \sin\beta \cos\alpha - D_2 \sin\beta \sin\alpha + D_3 \cos\beta \tag{4.26}$$

The solution of this problem is:

$$\alpha = \arctan(-D_2, -D_1) \tag{4.27}$$

$$\beta = \arctan(\frac{-D_3}{\sqrt{\sqrt{D_1^2 + D_2^2}}}) \tag{4.28}$$

Therefore, the continuous thrusting angles are determined as a function of the continuous static controls: $(W_a, W_f, W_g, W_h, W_k)$, which are named the weighting factors , and (m_Q, n_Q, r_Q), which are called the scaling factors.

Discrete Dynamics

The discrete dynamics of the system are governed by controlled events, i.e. the discrete velocity impulses of the CP and the discrete on/off switchings of the EP system, and by autonomous events, i.e. the electric engine shut down during eclipse. The discrete event functions associated to them, including discontinuity surfaces and transition map functions, will be formally defined in the following lines.

Chemical Propulsion Burns: The effect of a chemical engine, modeled as instantaneous, translates into discrete changes of the continuous state vector that are given by the transition map functions ϕ_c. The discontinuity surfaces s_c constrain the firings to occur when the spacecraft reaches the true longitude selected by the control parameter ϑ as long as the electric engine is switched off. Note that, the discrete state of the spacecraft is not affected by this maneuver, i.e. $q(L_j^+) = q(L_j^-)$. Therefore, the discrete dynamics is expressed as follows:

$$\text{Burn:} \begin{cases} s_{c,j} : L_j^- = \vartheta_j, \; j = 1, \dots, n \\ \phi_{c,j} : q(L_j^+) = q(L_j^-), \; \boldsymbol{x}(L_j^+) = \mathcal{C}(\boldsymbol{x}(L_j^-), \boldsymbol{u}_{c,j}), \; j = 1, \dots, n \end{cases} \tag{4.29}$$

where n is the number of allowed impulses and $\mathcal{C} : \mathcal{X} \times \mathcal{U} \times \mathbb{R} \longrightarrow \mathcal{X}$ is the procedure to compute the successor continuous state $\boldsymbol{x}(L_j^+)$ of the spacecraft after the j^{th} chemical maneuver as a function of the predecessor continuous state $\boldsymbol{x}(L_j^-)$ and the continuous static controls $\boldsymbol{u}_{c,j} = [\theta_j, \Delta\text{V}_j, \bar{\alpha}_j, \bar{\beta}_j]$. This procedure consists on the following steps:

- Obtaining the pre-maneuver cartesian position and velocity vectors of the spacecraft from the equinoctial elements using Eq.(4.7).
- The demanded impulse is added to the velocity vector projecting them onto the same reference frame as $\mathbf{v}(L_i^+)) = \mathbf{v}(L_i^+)) + \boldsymbol{\Delta}\text{V}_j$.
- The post-maneuver equinoctial elements are computed by converting the post-maneuver cartesian and velocity vectors to equinoctial elements.
- Compute the post-maneuver mass as $m(L_i^+) = m(L_i^-)e^{\Delta\text{V}_j/(g_0 I_{sp,c})}$.

Here, $I_{sp,c}$ is the specific impulse of the chemical engine. Let us defined σ_c as the time-ordered sequence of chemical firings:

$$\sigma_c = [\dots, (L_j, s_{c,j}), \dots] \tag{4.30}$$

which is not known a-priori and have to be determined as part of the solution.

Electric Engine on/off switching: In order to account for the on/off switchings during sun light, two quantities are introduced to measure the effectivity of thrust at a given point on the transfer. This allows differencing between thrusting and coasting arcs. Critical values of these coefficients can be predetermined to cut thrust at certain areas of the orbit, increasing travel time and reducing the used propellant mass. In order to calculate the effectivity coefficients, first the maximum and minimum $\dot{Q}$ has to be calculated with respect to the thrusting angles and the orbital position.

$$\dot{Q}_{min} = \min_{L,\alpha,\beta} \dot{Q} \tag{4.31}$$

$$\dot{Q}_{max} = \max_{L,\alpha,\beta} \dot{Q} \tag{4.32}$$

The extrema of $\dot{Q}$ over α and β can be easily determined:

$$\min_{\alpha,\beta} \dot{Q} = -\sqrt{D_1^2 + D_2^2 + D_3^2} \tag{4.33}$$

$$\max_{\alpha,\beta} \dot{Q} = +\sqrt{D_1^2 + D_2^2 + D_3^2} \tag{4.34}$$

The numerical Brent's method is used to find the extrema on $L = (0, 2\pi)$. The absolute and relative effectivity coefficient are defined as:

$$\eta_a = \frac{\dot{Q}}{\dot{Q}_{min}} \tag{4.35}$$

$$\eta_r = \frac{\dot{Q} - \dot{Q}_{max}}{\dot{Q}_{min} - \dot{Q}_{max}} \tag{4.36}$$

These values are computed at each integration step and compared to the cut-off values $\eta_{a,th}$ and $\eta_{r,th}$. Thrusting takes place in the optimal direction if the calculated efficiencies are above the threshold. They are modeled by the discontinuity surfaces s_{on} and $s_{off,1}$ and by the transition maps ϕ_{on} and ϕ_{off}. Additionally, the spacecraft shutdown during eclipse is represented by the discrete event function $s_{off,2}$ and transition map ϕ_{off}, as follows:

$$\text{Switching on:} \begin{cases} s_{on} : q(L_i^-) = 0, \quad \eta_a(L_i^-) > \eta_{a,th}, \quad \eta_r(L_i^-) > \eta_{r,th}, \quad \Xi(\boldsymbol{x}, L_i^-) = 1 \\ \phi_{on} : q(L_i^+) = 1, \quad \boldsymbol{x}(L_i^+) = \boldsymbol{x}(L_i^-) \end{cases} \tag{4.37}$$

$$\text{Switching off:} \begin{cases} s_{off_1} : q(L_i^-) = 1, \quad \eta_a(L_i^-) < \eta_{a,th}, \quad \eta_r(L_i^-) < \eta_{r,th} \\ s_{off_2} : q(L_i^-) = 1, \quad \Xi(\boldsymbol{x}, L_i^-) = 0 \\ \phi_{off} : q(L_i^+) = 0, \quad \boldsymbol{x}(L_i^+) = \boldsymbol{x}(L_i^-) \end{cases} \tag{4.38}$$

In previous Eqs.(4.54)-(4.55), the effect of the eclipse is included via the shadow function $\Xi(\boldsymbol{x}, L_i^-) : \mathcal{X} \times \mathbb{R} \longrightarrow \mathbb{Z}$. It represents a binary-valued function that takes the value '1' when the spacecraft is in sunlight and '0' when it is in umbra conditions. Let us defined σ_{sw} as the time-ordered sequence of the on-off switchings of the electric engine:

$$\sigma_{sw} = [\dots, (L_i, s_{on/off}), \dots] \tag{4.39}$$

which is not known a-priori and have to be determined as part of the solution.

Transfer Type and Propulsive System selection: The transfer type is selected by the integer static control $\Theta \in \mathbb{Z}$. Here the CP, EP and CCEP are represented by $\Theta = 1, 2, 3$ respectively. The selection of the propulsive system and solar arrays are modeled by the static discrete control $\Pi \in \mathbb{Z}$. Such parameter contains the list of available propulsive options. Thus, parameters such as the specific impulse $I_{sp,e}$ and mass m_e of the electric engine, the specific impulse $I_{sp,ch}$ and mass m_{ch} of the chemical engine , and the solar array parameters are functions of Π. As a consequence, the initial mass of the system is composed of the following:

$$m_0 = m_e(\Pi) + m_{ch}(\Pi) + m_p(\Pi) + m_{payload} \tag{4.40}$$

where m_p is the propellant mass required for the transfer, including the electric and chemical engine, and $m_{payload}$ is the payload mass.

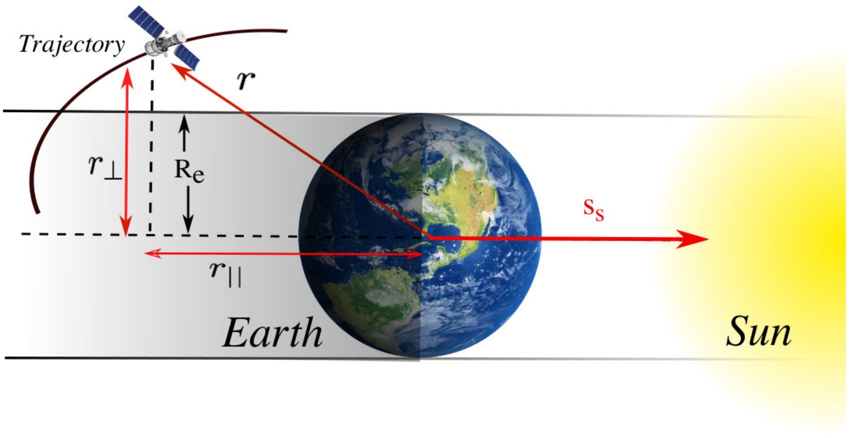


FIGURE 4.4: MOLTO-OR: Eclipse Geometry

Earth Shadow: It is reasonable to assume that both the Sun and Earth are spherical bodies as suggested in Ref. [55]. It is further assumed that the Sun is infinitely far away from the Earth resulting in a cylindrical shadow. The geometry of the eclipse, under the cylindrical shadow approximation, is depicted in Figure 4.4 including the spacecraft orbit. The radius of the cylinder is therefore equal to the radius of the earth R_e and is axis is determined by the unitary vector from the Earth to the Sun $\boldsymbol{\hat{s}}_s$. Let us define $\mathrm{r}_\perp$ as the distance of the spacecraft to the axis of the cylinder and it can be obtained as:

$$\mathrm{r}_\perp = \sqrt{\mathrm{r}^2 - \mathrm{r}^\mathrm{T}\boldsymbol{\hat{s}}} \tag{4.41}$$

where $\mathrm{r} = ||\mathbf{r}||$. The condition for the spacecraft to be inside the cylinder can be simply expressed as $r_\perp \leq R_e$. In addition to this, in order to be inside the shadow region of the cylinder, the constraint $\mathbf{r}^T\boldsymbol{s} \leq 0$ must be fulfilled. Combing both conditions, a necessary and sufficient condition for an eclipse to happen is derived as:

$$\mathbf{r}^T\boldsymbol{\hat{s}} + \sqrt{\mathrm{r}^2 - R_e^2} \leq 0 \tag{4.42}$$

Thus, the binary shadow function Ξ has the following form:

$$\Xi = \begin{cases} 0 & \text{if} \quad \mathbf{r}^T\boldsymbol{\hat{s}} + \sqrt{\mathrm{r}^2 - \mathrm{R_e^2}} \leq 0 \\ 1 & \text{otherwise} \end{cases} \tag{4.43}$$

Radiation Environment: Following the same approach as in Ref. [87], the geomagnetic field is modeled by considering a magnetic dipole with an axis parallel to the Earth's magnetic axis passing through its center of mass. The omnidirectional radiation flux is defined as the flux of all particles averaged over all directions at any location owing to charged particles, namely protons and electrons, and it can be computed as a function of the McIlwain's coordinate Λ and the latitude ϕ_e. The parameter Λ represents the distance to the magnetic field line at the equator and it has the following expression:

$$\Lambda = \frac{\mathrm{r}}{R_e \sin^2\left(\frac{\pi}{2} - \phi_e\right)} \tag{4.44}$$

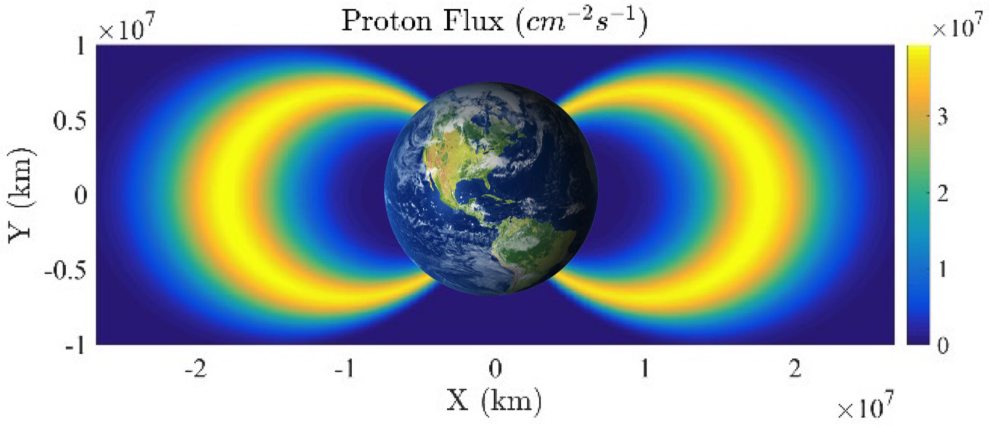


FIGURE 4.5: Analytic Integral Proton Flux for Energy levels greater than 1 MeV

There exits suitable radiation models such as the NASA AP-8 and AE-8 model [218] or the more recent NASA AP-9 and AE-9, which provide tabulated data for the flux values for various energies. However, usage of tabulated values does not lend itself well for optimization purposes. To this end, approximate analytical expressions are derived in Ref. [87]. Furthermore, it is known that the inner Van Allen belt, which is mainly composed of protons, is significantly more hazardous compared to the outer belt, which is mainly composed of electrons. In fact, it has been shown that the radiation damage from electrons during orbit-raising represents a small fraction of what the satellite undergoes in GEO over a period of 10-15 years [87].

Hence, in this work, only the radiation damage caused by protons is considered. The omni-directional flux of protons of energy equal or greater than E can be expressed by the following analytical form derived in Ref. [87]:

$$\Phi_p(\Lambda, \phi_e, E) = a(\Lambda, E)\mathrm{e}^{-b(\Lambda,E)\phi_e^2} \tag{4.45}$$

where $a(\Lambda, E)$ and $b(\Lambda, E)$ are given by:

$$a(\Lambda, E) = a_0\mathrm{e}^{(a_1E+a_2(a_3+\Lambda)^2)} \tag{4.46}$$

$$b(\Lambda, E) = b_0 + b_1E + b_2\Lambda + b_3E\Lambda + b_4\Lambda^2 + b_5\Lambda^3 \tag{4.47}$$

Radiation models like AP-8 or AP-9 can be used to find the values of the fitting parameters a_j and b_j that best approximate the radiation data. In our approach we will use the AP-8 data [87]. The fitting parameters obtained are summarized in Table 4.1 and the proton flux for energy levels equal or greater than 1 MeV is plotted in Figure 4.5. Given that the coordinates (Λ, ϕ_e) depends uniquely on the position of the spacecraft, the radiation flux can be expressed in terms of the continuous state vector $\boldsymbol{x}$ and the true longitude L, so that $\Phi_p(\boldsymbol{x}, L, E) = \Phi_p(\Lambda, \phi_e, E)$. Therefore, the total fluence of protons of all energy levels greater than or equal to E encountered along the orbit-raising trajectory until a certain time t is given by:

$$\Psi_p(\boldsymbol{x}, L, E) = \int_{t_0}^{t} \Phi_p(\boldsymbol{x}, L, E)dt \tag{4.48}$$

a($\mathcal{L}$,E)	Units	b($\mathcal{L}$,E)	Units
$a_0 = 2.094 \times 10^8$	cm^{-2} s$^-1$	$b_0 = -0.00971$	-
$a_1 = -1.673$	MeV^{-1}	$b_1 = 0.0000982$	MeV^{-1}
$a_2 = -2.07$	-	$b_2 = 0.01484$	-
$a_3 = -2.825$	-	$b_3 = 0.0001561$	MeV^{-1}
		$b_4 = -0.004581$	-
		$b_5 = 0.0004356$	-

Table 4.1: Van Allen Radiation Belts: AP-8 Fitting Parameters

Radiation Damage: The long term radiation effects important to be considered for instrument and spacecraft design fall into two main categories: ionizing and non-ionizing effects. The latter is also known as Displacement Damage Dose. Both causes degradations of microelectronics, optical components and solar cells. In terms of solar-cell degradation, ionization effect has little effect, whereas Displacement Damage Dose is the major source of degradation. The radiation damage that may occur to the payload or other on-board electronics is not assessed here, but it should be less than that encountered by the array as better shielding can be used to protect them.

The amount of degradation is a function of the type of solar cells, amount and material of the shielding provided, the energies of radiation encountered along the path an the number of particles for each energy level. We utilize the parametric relationships from Ref. [219] in terms of the displacement damage dose and non-ionizing energy loss in order to compute the power P_a available to the spacecraft:

$$\frac{P_a}{P_0} = 1 - C \log\left(1 + \frac{D_d}{D_x}\right) \tag{4.49}$$

where D_d is the displacement dose of protons for given solar cells, D_x and C are constants specific to the type of solar cells. The displacement damage dose, according to Ref. [87], is defined as:

$$D_d = \int_{E_l}^{E_u} \frac{\partial \Psi_p}{\partial E} S_P(E) dE \tag{4.50}$$

In the above, $S_P(E)$ is the non-ionizing energy loss. Then, it holds that:

$$\frac{\partial \Psi_p}{\partial E} = \frac{\partial}{\partial E} \int_{t_0}^{t} \Phi_p(\boldsymbol{x}, L, E) dt = \int_{t_0}^{t} \frac{\partial \Phi_p(\boldsymbol{x}, L, E)}{\partial E} dt \tag{4.51}$$

Hence, we have

$$D_d = \int_{t_0}^{t} \int_{E_l}^{E_u} \frac{\partial \Phi_p(\boldsymbol{x}, L, E)}{\partial E} S_P(E) dE dt \tag{4.52}$$

The rate of change of the radiation flux with respect to the energy levels can be obtained analytically from Eq.(4.45) as:

$$\frac{\partial \Phi_p(\boldsymbol{x}, L, E)}{\partial E} = [a_1 - \phi_e^2(b_1 + b_3\Lambda)]\Phi_p(\boldsymbol{x}, L, E) \tag{4.53}$$

Thus, in order to obtained the accumulated radiation flux, Eq.(4.52) can be integrated with respect to the true longitude, instead of the physical time t, at the same time that the dynamics of the spacecraft (Eq.(4.1)). Note that the maximum thrust T during the EP phase diminishes with input power P accordingly with Eq.(4.49), unless the system is designed so that there is always enough power to the EP at full-power. The minimization of the power loss is critical because it determines the power at the beginning of the operational life of the satellite.

4.2.2 MOLTO-OR Step2

In a similar fashion as MOLTO-OR Step 1, the continuous state vector of the spacecraft is determined by the set $\boldsymbol{x} = [p, f, g, h, k, t, m]$, where the true longitude L is the independent variable. Also, the discrete state of the spacecraft is described by the variable $q \in \{0, 1\}$, where '0' designates the coasting mode and '1' indicates the thrusting mode. The chemical engine provides n-instantaneous velocity changes on the spacecraft, which are determined by the set of continuous static controls $\boldsymbol{u}_{c,j} = [\vartheta_j, \Delta V_j, \bar{\alpha}_j, \bar{\beta}_j]$. However, in this step, the electric engine is controlled by the set of dynamic continuous controls: α and β, being the azimuth and declination angles respectively as described in the previous section. Additionally, the controlled on/off switchings of the electric engine during sunlight are managed by the binary control input $v_e \in \{0, 1\}$. The coasting state is required when '0', while a burning phase is demanded for '1'.

Continuous Dynamics

The evolution of the continuous state vector of the system with respect to the true longitude is given as in Eqs.(4.1), where the thrust acceleration is directly controlled by the azimuth α and declination β angles.

Discrete Dynamics

The discrete event functions associated to the discrete dynamics, including discontinuity surfaces and transition map functions for the chemical propulsion firings and the on/off switchings of the electric engine, will be formally defined in the following lines.

Chemical Propulsion Burns The chemical engine firings discontinuity surface $s_{c,j}$ and transition map functions $\phi_{c,j}$ are modeled as in the previous step in Eq.(4.29).

Electric Engine on/off switching In this case, the electric engine switchings are controlled by the dynamic discrete input $\boldsymbol{v}_e$ as long as the spacecraft is in sunlight conditions. They are modeled by the discontinuity surfaces s_{on} and $s_{off,1}$ and by the transition maps ϕ_{on} and ϕ_{off}. Additionally, the spacecraft shutdown during eclipse is represented by the discrete event function $s_{off,2}$ and transition map ϕ_{off}.

Therefore, the following equations are included for a complete descriptions of the system:

$$\text{Switching on:} \begin{cases} s_{on} : q(L_i^-) = 0, \quad \boldsymbol{v}_e(L_i^-) = 1, \quad \Xi(\boldsymbol{x}, L_i^-) = 1 \\ \phi_{on} : q(L_i^+) = 1, \quad \boldsymbol{x}(L_i^+) = \boldsymbol{x}(L_i^-) \end{cases} \tag{4.54}$$

$$\text{Switching off:} \begin{cases} s_{off_1} : q(L_i^-) = 1, \quad v_e(L_i^-) = 0 \\ s_{off_2} : q(L_i^-) = 1, \quad \Xi(\boldsymbol{x}, L_i^-) = 0 \\ \phi_{off} : q(L_i^+) = 0, \quad \boldsymbol{x}(L_i^+) = \boldsymbol{x}(L_i^-) \end{cases} \tag{4.55}$$

As in MOLTO-OR Step 1, the effect of the eclipse is included via the binary shadow function $\Xi(\boldsymbol{x}, L_i^-) : \mathcal{X} \times \mathbb{R} \longrightarrow \mathbb{Z}$. Note that, an explicit relation between the discrete state of the spacecraft, the discrete input vector and the shadow function can be deduced:

$$q(v_e, \boldsymbol{x}, L) = v_e \Xi(\boldsymbol{x}, L) \tag{4.56}$$

Transfer Type and Propulsive System selection: The transfer type and propulsive system is provided by MOLTO-OR Step 1. Therefore, in this step, no optimization of the propulsive system or the transfer type is performed.

Earth Shadow and Radiation environment: The same models as in MOLTO-OR Step 1 are considered for this step.

Constraints

During a typical transfer to the Geostationary orbit, several operational constraints can apply. Hereafter, the models used for imposing slew-rates limits, avoiding the geostationary ring for imposing slot-phasing constraints will be presented.

Slew Rate Limits: As a practical matter, limits are often imposed on the angular rate of change of the thrust direction vector as a requirement from the attitude subsystem. An unconstrained steering law may not be realizable by the space system. For a small true longitude interval ΔL, the angle θ_T between two unit direction vectors is defined by:

$$\boldsymbol{d}^T(L)\boldsymbol{d}(L + \Delta L) = \cos\theta_T \tag{4.57}$$

It then follows that an approximate rotation rate, assuming a linear variation, can be computed as:

$$\dot{\theta}_T \approx \frac{1}{t(L + \Delta L) - t(L)} \cos^{-1}\left[\boldsymbol{d}^T(L)\boldsymbol{d}(t + \Delta L)\right] \leq \dot{\theta}_{T,lim} \tag{4.58}$$

In the above, $\dot{\theta}_{T,lim}$ is the maximum allowed rotational velocity. The value of ΔL is typically chosen to coincide the integration step of the dynamical equations.

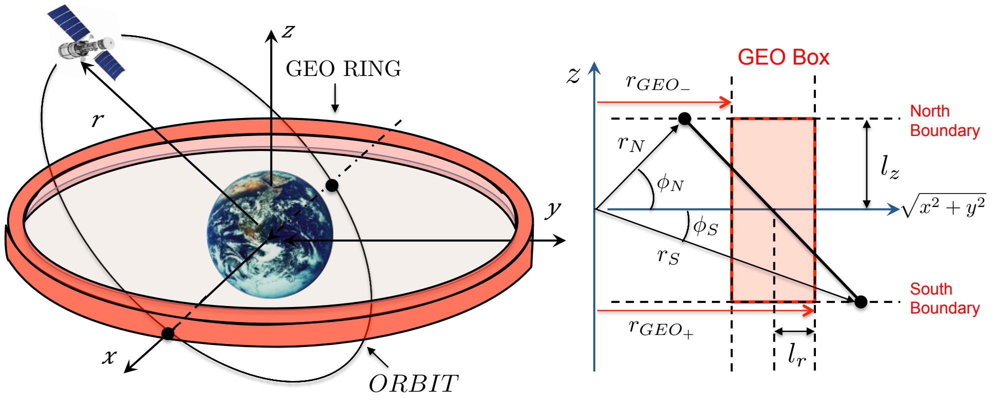


FIGURE 4.6: MOLTO-OR: GEO Ring Geometry

GEO Ring Avoidance: The GEO belt or GEO ring is understood as the volume in space where most of the operational satellites in GEO are located. The region is also populated by uncontrolled objects that are subject to variations in altitude due to orbit perturbations. During the transfer, any crossing of the GEO ring poses a certain collision risk with high value assets. In general, a spacecraft might cross the GEO ring at the beginning, mid of the transfer, and at the end. Especially at the end of a low-thrust orbit transfer to the GEO the spacecraft may cross the GEO belt several times since the spacecraft targets zero eccentricity and inclination at GEO altitude.

The GEO Ring considered in our study is defined as a toroid around Earth with a rectangular cross-section centered at GEO (see Figure 4.6). The rectangular cross-section is termed GEO box and its width and height are $2l_r$ and $2l_z$ respectively. The inner r_{GEO-} and outer r_{GEO+} radius of the toroid are defined as:

$$\mathrm{r}_{GEO-} = \mathrm{r}_{GEO} - l_r \tag{4.59}$$

$$\mathrm{r}_{GEO+} = \mathrm{r}_{GEO} + l_r \tag{4.60}$$

Here, r_{GEO} is the radius of the GEO orbit. Additionally, we define the North and South boundaries as the planes parallel to Equatorial plane but located at a distance l_z above and below respectively. The intersection of the spacecraft's trajectory with such planes determines r_N and r_S, where the subscript N and S refers to the North and South boundary respectively and the symbol r_{NS} is used to refer to both indistinctly.

The conditions to avoid crossing the GEO ring can be evaluated as a function of the projections onto the Equatorial plane of the radii at the north/south boundary of the GEO arc as a set of complementary non-linear constraints:

$$\mathrm{r}_{NS}\cos\phi_{SN} \leq r_{GEO-} \quad \text{or} \quad \mathrm{r}_{NS}\cos\phi_{SN} \geq \mathrm{r}_{GEO+} \tag{4.61}$$

where ϕ_{NS} is the elevation angle, measured out of the Equatorial plane, for r_{NS} and it is computed as:

$$\phi_{NS} = \pm\sin^{-1} l_z/\mathrm{r}_{NS} \tag{4.62}$$

In the above, it is select the + sign for r_N and the − sign for r_S. The previous complementary Eqs.(4.61) can be reformulated as a set of nonlinear inequality constraints as follows:

$$(\mathrm{r}_{GEO-} - \mathrm{r}_N \cos\phi_N)(\mathrm{r}_{GEO-} - \mathrm{r}_S \cos\phi_S) \geq 0 \tag{4.63}$$

$$(\mathrm{r}_{GEO+} - \mathrm{r}_N \cos\phi_N)(\mathrm{r}_{GEO+} - \mathrm{r}_S \cos\phi_S) \geq 0 \tag{4.64}$$

$$(\mathrm{r}_{GEO} - \mathrm{r}_{NS} \cos\phi_{SN})^2 - (2l_{\mathrm{r}})^2 \geq 0 \tag{4.65}$$

Note that the previous expression are not valid for small inclinations, where no crossings of the trajectory with the north/sound boundaries occurs. In such case, the value of l_z can be reduced and the constraints could be applied until the inclination reaches zero. Also, at least the last orbit must be unconstrained in order to allow targeting of the final orbit.

Boundary and Phasing constraints: Typically, the design of an orbit raising trajectory constrains the spacecraft to depart from a specific orbit $(a_0, e_0, i_0, w_0, \Omega_0, v_0)$ and at a certain initial epoch t_0. These orbits are typically defined by their orbitals elements but the constraints can be expressed equivalently in terms of the equinoctial elements as:

$$\begin{aligned}
p(L_0) &= a_0(1-e_0^2), & h(L_0) &= \tan(i_0/2)\sin(\Omega_0) \\
f(L_0) &= e_0\cos(w_0+\Omega_0), & k(L_0) &= \tan(i_0/2)\cos\Omega_0, \\
g(L_0) &= e_0\sin(w_0+\Omega_0), & L_0 &= \Omega_0 + w_0 + v_0, \\
m(L_0) &= m_0, & t(L_0) &= t_0.
\end{aligned} \tag{4.66}$$

Similarly, the spacecraft has to arrive at the target orbit $(a_f, e_f, i_f, w_f, \Omega_f)$. The associated constraint in equinoctial elements can be expressed as Eq.(4.66) but changing the suffix '0' to 'f'. The final transfer epoch $t_f = t(L_f)$, final mass $m_f = m(L_f)$ and final true longitude L_f can be constrained to a specific value or may be free and subject to optimization. In case satellite is to be transferred to GEO, it will have to occupy a single slot above the equator. Phasing is the operation to target the certain longitude of the assigned slot. This constraint is applied to the final longitude as:

$$\cos L_f = \cos L_T(t_f), \quad \sin L_f = \sin L_T(t_f) \tag{4.67}$$

where L_T is the target final true longitude, which is a function of the arrival time, as the Earth rotates with respect to the inertial reference frame. Note that, the previous constraint is expressed as a trigonometric relation in order to make it independent of the number of revolutions. Typically, the slot-phasing condition is given as a function of a target longitude δ_T with respect to the Greenwich meridian. The position of the Greenwich meridian is computed by assuming a constant rotational velocity for the Earth. Thus, the target true longitude can be computed as follows:

$$L_T(t_f) = \delta_T + \delta_G(t_0) + w_E(t_f - t_0) \tag{4.68}$$

where $w_E = 7.2722 \cdot 10^{-5}$ rad/sec is the mean Earth's rotational velocity and $\delta_G(t_0)$ is the longitude of the Greenwich meridian with respect to the first point of Aries at the initial epoch.

4.3 Solution Approach

We present MOLTO-OR, which is a two-step solution approach for the multi-objective HOCP problem under consideration. The algorithm is schematically depicted in Figure 4.7. In MOLTO-OR Step 1 we incorporate a parametric model of the low-thrust control law based on the Lyapunov function Q-law [70], pursuing the goal of developing a flexible and robust algorithm able to rapidly find solutions, which would approximate the optimal performances as well as the mission design variables, needing minimum information from the user. In MOLTO-OR Step 2 we include the complete model of the trajectory and control seeking for accuracy and robustness in addition to the possibility of including complex constraints with ease.

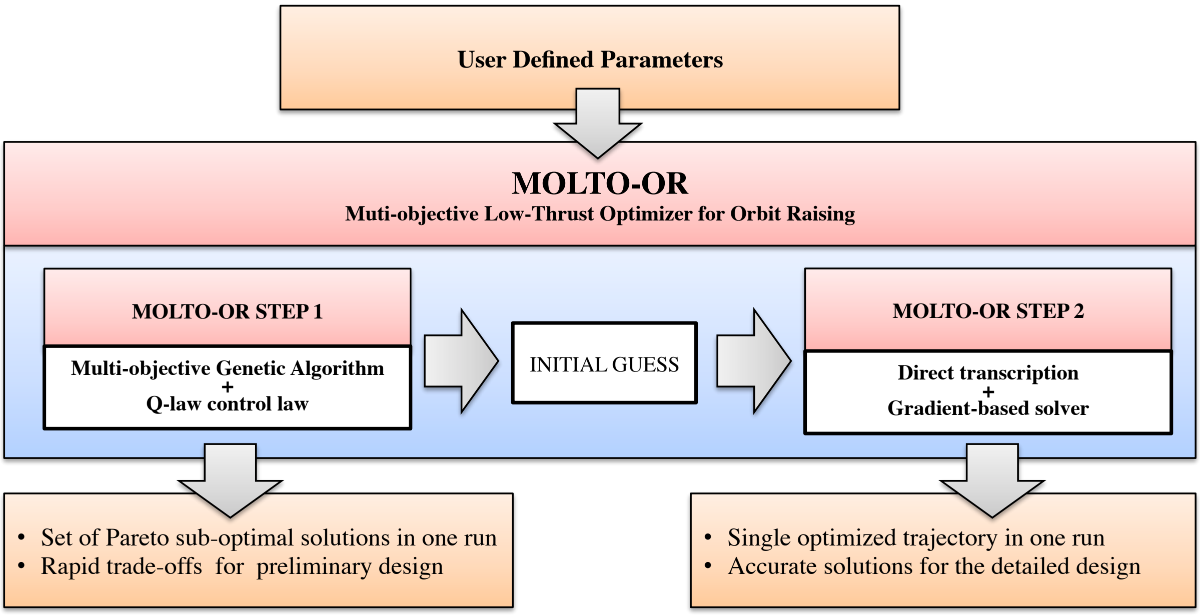

User Defined Parameters
MOLTO-OR
Muti-objective Low-Thrust Optimizer for Orbit Raising
MOLTO-OR STEP 1
Multi-objective Genetic Algorithm + Q-law control law
INITIAL GUESS
MOLTO-OR STEP 2
Direct transcription + Gradient-based solver
• Set of Pareto sub-optimal solutions in one run
• Rapid trade-offs for preliminary design
• Single optimized trajectory in one run
• Accurate solutions for the detailed design

Figure 4.7: MOLTO-OR: Algorithm Scheme

Table 4.2: MOLTO-OR Step 1 VS MOLTO-OR Step 2 main features

	MOLTO-OR Step 1	MOLTO-OR Step 2
Problem type	Multi-objective MINLP	Multiphase OCP
Solution Approach	Genetic Algorithm	Hermite-Simpson + Gradient-based solver
Dynamical Model	3D + J_2 + shadow + radiation	3D + J_2 + shadow + radiation
Control Model	Lyapunov Q-law	Optimal control
Propulsive system	Free	Fixed
Transfer Type	Free	Fixed
Thrust Model	Constant	Constant
I_{sp} Model	Constant	Constant
Ephemerides	Analytic approximation	Analytic approximation
Programming Language	C++	AMPL
Heuristic solver	NSGA-II	N/A
Gradient-based solver	N/A	Ipopt

4.3.1 MOLTO-OR Step1

In this step, we convert the multi-objective HOCP into an unconstrained multi-objective mixed-integer parameter optimization problem with a small set of design variables. Due to that fact and to the requirement of evaluating many different scenarios simultaneously, a population-based heuristic algorithms has been chosen as the most adequate technique to solve it. In particular, a Genetic Algorithm based on the well-known NSGA-II (Non-dominated Sorting Genetic Algorithm) has been selected. MOLTO-OR Step1 has been fully implemented in C++, given that this step requires the rapid evaluation of FCT, FET and CCET. In the following lines, we describe the particular design variables and evaluation procedure separately for each transfer case. Thereafter, the complete optimization algorithm will be explained.

Fully Chemical Transfer (FCT): During a FCT, a series of n-chemical maneuvers are performed sequentially. Solar-cell degradation is not considered for this transfer as the crossing of the radiation belts is very fast. The location, magnitude and direction of the j^{th} impulse is defined by $\boldsymbol{u}_{c,j} = [\vartheta_j, \Delta V_j, \bar{\alpha}_j, \bar{\beta}_j]$ for $j = 1, \ldots, n-2$. The trajectory between the $(j^{th} - 1)$ and the j^{th} impulses is assume to follow a Keplerian arc and is computed analytically between ϑ_{j-1} and ϑ_j. A Lambert's arc is assumed for the last two impulses in order to guarantee that the satellite is inserted into the final orbit exactly. Three additional design variables are included to define the last two-impulses: the parameter ϑ_{n-1} that represents the true longitude on the orbit where the first Lambert's impulse occurs; the parameter t_c that represents the time of flight on the transfer arc; and ϑ_n that is the true longitude on the final orbit at the time of the satellite arrival. Therefore, the set of parameters $\mathcal{C}$ defining the FCT is:

$$\mathcal{C} = (\boldsymbol{u}_{c,1}, \ldots, \boldsymbol{u}_{c,n-2}, \vartheta_{n-1}, t_c, \vartheta_n) \tag{4.69}$$

Note that the number of impulses n is not a design variable. Instead, it is assumed to be equal to the maximum number of allowable impulses n_{max} defined by the user. The number of impulses is then optimized by driving to zero the magnitude of the unnecessary ΔV's.

Fully Electric Transfer (FET): In the case of a FET, the parametric feedback control naturally drives the spacecraft to the final orbit. The dynamics are satisfied by integrating the differential equations using a time-marching algorithm. Let us define the set of parameters $\mathcal{E}$ that defines the low-thrust guidance as:

$$\mathcal{E} = (W_p, W_f, W_g, W_h, W_k, \eta_{a,th}, \eta_{r,th}, m_Q, n_Q, r_Q) \tag{4.70}$$

Here, $(W_p, W_f, W_g, W_h, W_k)$ are the weighting factors associated to each equinoctial element, $(\eta_{a,th}, \eta_{r,th})$ are the absolute and relative effectivity coefficients, which allow introducing coasting arcs and finally (m_Q, n_Q, r_Q) are the scaling factors.

Let the $\mathcal{Q}_{law}$ function be defined as the procedure that computes the spacecraft thruster orientation as a function of $\mathcal{E}$ the actual state $\boldsymbol{x}$ and the desired final state $\boldsymbol{x}_f$ using the Q-law as guidance. Hence, it holds:

$$[\boldsymbol{u}_e, v_e, Q] = \mathcal{Q}_{law}(\mathcal{E}, \boldsymbol{x}, \boldsymbol{x}_f, L) \tag{4.71}$$

The dynamical system in Eq.(4.1) is integrated using a Runge-Kutta Fehlberg 7(8) scheme until the proximity quotient Q reaches a threshold Q_{th}, which means that the spacecraft has already targeted the orbit. Note that transfer time and final true longitude are not design variables and are obtained after the integration of the trajectory. During the integration of the equations of motion, the displacement damage dose at every instant has to be computed, updating the power available to the spacecraft and the magnitude of the thrust when necessary. To compute the radiation damage caused to the solar arrays, a discrete set of energy levels $E = (E_1, E_2, ..., E_m)$ is considered, such that $E_1 = E_l$ and $E_m = E_u$ with E_l and E_u being respectively the lower and the upper bounds of the set of m energy levels under consideration. This is reasonable as the non-ionizing energy loss $S_P(E)$ is typically provided as tabular data. Based on this discretization the energy integral in Eq.(4.52) can be approximately obtained as the following sum:

$$\sum_{i=1}^{m-1} \frac{1}{2} \left[\frac{\partial \Phi_p(\boldsymbol{x}, L, E_i)}{\partial E} S_P(E_i) + \frac{\partial \Phi_p(\boldsymbol{x}, L, E_{i+1})}{\partial E} S_P(E_{i+1}) \right] (E_{i+1} - Ei) \tag{4.72}$$

Combined-Chemical-Electric Transfer (CCET): The CCET consists on a chemical segment followed by an electric phase. The chemical phase is obtained following the same procedure as for the FCT, except for the last lambert arc. In this case, the last two impulses are regarded as optimization variables and applied in a similar manner as the $n^{th} - 2$ previous firings. The final state after the n^{th} impulse is used to define the initial orbit for the electric phase. Thereafter, the Q-law is applied to target the desired orbit with EP thruster. Thus, the set $\mathcal{H}$ of parameters that determines a CCET is defined by:

$$\mathcal{H} = (\boldsymbol{u}_{c,1}, \ldots, \boldsymbol{u}_{c,n}, W_p, W_f, W_g, W_h, W_k, \eta_{a,th}, \eta_{r,th}, m_Q, n_Q, r_Q) \tag{4.73}$$

Optimization Algoritm: The complete algorithm has to select the optimal propulsive system and transfer type as well as to optimize the corresponding steering law. The type transfer is selected by the integer variable Θ and the propulsive system is determined by the integer variable Π. Let us define $\mathcal{Z}$ and $\mathcal{P}$ as the set of transfer options and propulsive systems respectively. Thus, the set $\mathcal{O}$ of design variables for the complete algorithm is represented by:

$$\mathcal{O} = \mathcal{C} \cup \mathcal{E} \cup \mathcal{H} \cup \mathcal{Z} \cup \mathcal{P} \tag{4.74}$$

where the sets $\mathcal{C}, \mathcal{E}$ and $\mathcal{H}$ of control parameters are the above-mentioned for each case. The complete optimization problem is thus defined as finding the set of parameters $\mathcal{O}$ such that the following multi-objective fitness function is minimized:

$$J = (-m_f, t_f, -P_f) \tag{4.75}$$

Table 4.3: MOLTO-OR Step 1 Variables

Variable	Meaning	Lb	Ub
θ_i	True Longitude for i^{th} CP maneuver	0	2π
ΔV_i	Magnitude of for i^{th} CP maneuver	0	ΔV_{max}
$\bar{\alpha}_i$	In-Plane angle of the i^{th} CP maneuver	$-\pi$	π
$\bar{\beta}_i$	Out-of-Plane angle of the i^{th} CP maneuver	$-\pi/2$	$\pi/2$
t_c	Transfer time to last CP maneuver	$t_{c,max}$	$t_{c,min}$
$W_{p,f,g,h,k}$	Q-law weighting factors	1	100
$\eta_{a,th}, \eta_{r,th}$	Q-law effectivity coefficients	0	0.98
m_Q, n_Q, r_Q	Q-law scaling factors	0.01	10
Θ	Transfer options	1	3
Π	Propulsion subsystem option	-	-

Table 4.4: MOLTO-OR: User-Defined parameters

Variable	Meaning
t_0	Initial Launch date
$a_0, e_0, i_0, \omega_0, \Omega_0$	Initial Orbital elements
$a_f, e_f, i_f, \omega_f, \Omega_f$	Final Orbital elements
n_{max}	Max number of chemical impulses
$t_{c,max}, t_{c,min}$	Max/Min transfer time to last CP maneuver
m_0	Initial Spacecraft mass
$I_{sp,e,i}$, $m_{e.i}$, η_i	Parameters for the i^{th} available electric engine
$I_{sp,ch,i}$, $m_{ch,i}$, $\Delta V_{max,i}$	Parameters for the i^{th} available chemical engine
$P_{0,i}, C_i, D_{x,i}, SP(E)_i$	Parameters for the i^{th} available solar array

Here, m_f, t_f, P_f are respectively the mass, time and power of the solar arrays at the end of the transfer. A summary of the optimization variables can be found in Table 4.3, along with their lower bounds (Lb) and upper bounds (Ub). The objective function for the Genetic Algorithm is computed following the flow chart depicted in Fig. 4.8.

The user only needs to provide information about the initial and final orbit along with the mass, propulsive and power generation system characteristics. The user-defined parameters and summarized in Table 4.4. Note that, the algorithm used to compute the lambert arc at the end of a FCT may not converge. In this case, a penalty vector is associated to lead the heuristic solver to discard them for the next generations. Similarly, the convergence of the proximity quotient Q to $Q < Q_{th}$ is not always assured. Thus, a maximum number of integration steps is imposed to avoid the integration scheme from running indefinitely. If this value is exceeded, a penalty is added to the fitness function.

$$J = (-m_f - \rho, t_f + \rho, -P_f - \rho) \tag{4.76}$$

In the above, ρ is a sufficiently large value with respect to the value of fitness function for the feasible trajectories, such that infeasible trajectories will be pruned out by the genetic algorithm during the selection process. Alternatively, the propellant mass consumed m_p can be considered as objective instead of the final mass m_f.

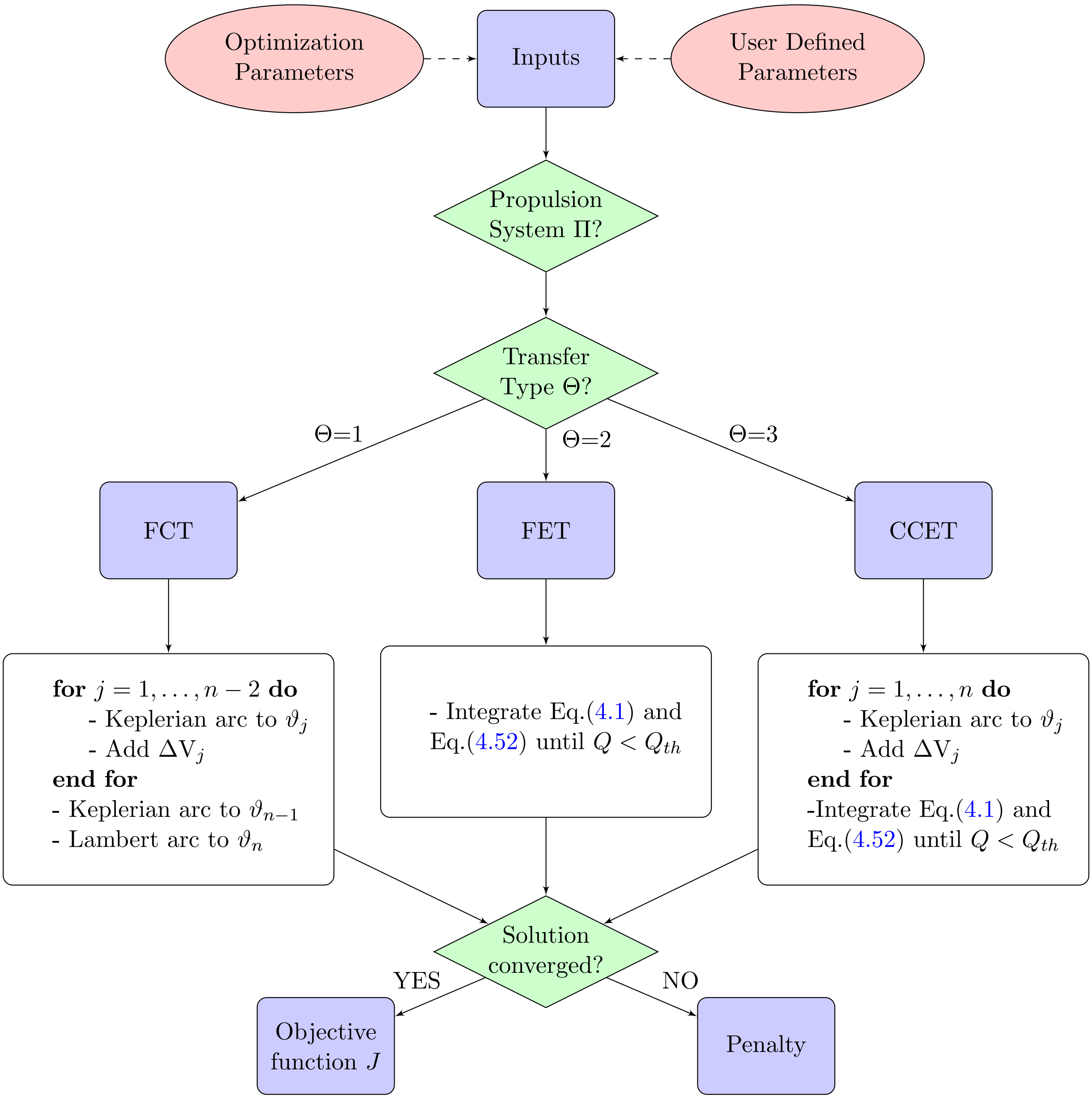

Optimization Parameters
Inputs
User Defined Parameters
Propulsion System Π?
Transfer Type Θ?
Θ=1
Θ=2
Θ=3
FCT
FET
CCET
for $j = 1, \ldots, n-2$ **do**
- Keplerian arc to ϑ_j
- Add ΔV_j
end for
- Keplerian arc to ϑ_{n-1}
- Lambert arc to ϑ_n
- Integrate Eq.(4.1) and Eq.(4.52) until $Q < Q_{th}$
for $j = 1, \ldots, n$ **do**
- Keplerian arc to ϑ_j
- Add ΔV_j
end for
-Integrate Eq.(4.1) and Eq.(4.52) until $Q < Q_{th}$
Solution converged?
YES
NO
Objective function J
Penalty

FIGURE 4.8: MOLTO-OR Step1: Flow chart of the fitness function

4.3.2 MOLTO-OR Step2

In MOLTO-OR Step 2, the goal is to reduce the multi-objective HOCP to a single-objective Large-scale Nonlinear parameter optimization problem and solve it with robust classical gradient-based solvers. For such purpose, we assume that the propulsive system and the number of chemical firings is known and provided by MOLTO-OR Step 1. However, the thrust/coast sequence of the EP is not imposed. It will be optimized by the algorithm instead. As shown in Eq.(4.56) the sequence is determined by the binary control $\boldsymbol{v}_e$ and the binary shadow function δ. However, gradient-based solvers only accept continuous variables and continuous and differentiable functions within the search domain. The binary control $\boldsymbol{v}_e$ is relaxed so that it can continuously vary within the $[0, 1]$ interval. This is not problematic, as the optimal solution for minimum-fuel fixed-time problems is known to be bang-bang.

Table 4.5: MOLTO-OR Step 2 Variables

Variable	Meaning	Lb	Ub
θ_i	True Longitude for i^{th} CP maneuver	0	2π
ΔV_i	Magnitude of for i^{th} CP maneuver	0	ΔV_{max}
$\bar{\alpha}_i$	In-Plane angle of the i^{th} CP maneuver	$-\pi$	π
$\bar{\beta}_i$	Out-of-Plane angle of the i^{th} CP maneuver	$-\pi/2$	$\pi/2$
p_j, f_j, g_j, h_j, k_j	Modified equinoctial elements at the j^{th} node	-	-
t_j	Elapsed Time at the j^{th} node	0	ToF
m_j	Spacecraft Mass at the j^{th} node	0	m_0
α_j	In-Plane angle of the EP at the j^{th} node	$-\pi$	π
β_j	Out-of-Plane angle of the EP at the j^{th} node	$-\pi/2$	$\pi/2$
v_j	Throttle parameter at the j^{th} node	0	1
L_k	True Longitude of the k^{th} north/south bound	-	-
L_0, L_f	Initial/Final true Longitude	-	-

Additionally, the binary shadow function Ξ is approximated by a smoothing function, where ϵ is the smoothing parameter, as:

$$\Xi = \frac{1}{1 + \mathrm{e}^{-\epsilon(r^T \hat{\boldsymbol{s}} + \sqrt{r^2 - R_e^2})}} \tag{4.77}$$

As a result, the discrete state q of the spacecraft is no longer a binary function. We discretize the continuous states $\mathbf{x}_j = \boldsymbol{x}(L_j)$ and control inputs $\boldsymbol{u}_j = \boldsymbol{u}(L_j)$ and $\boldsymbol{v}_j = \boldsymbol{v}(L_j)$ on a selected uniform grid L_j, for $j = 1, \ldots, n$, where n is number of grid points. The dynamical equations are imposed as defect constraints based on the Hermite-Simpson collocation scheme [126] that has been explained in Section 3.4. The energy integral in Eq.(4.52) is evaluated with the same scheme shown in Eq.(4.72) , whereas the time integral uses the Hermite-Simpson discretization.

Slew rates limits are imposed as path constraints at each node as formulated in Eq.(4.58). The value $\delta L = L_j - L_{j-1}$ is chosen as the spacing between grid-points. The phasing constraint (see Eq.(4.67)) is imposed as a terminal constraint at the last node L_n. The application of the GEO ring avoidance constraints in Eqs.(4.63)-(4.65) requires the determination of the spacecraft radii at the North/south boundary of the GEO box. Thus, interior point constraints have to applied at L_k, for $k = 1, \ldots, m$, as follows:

$$r_z(L_k) = \pm l_z \tag{4.78}$$

where r_z is the vertical projection of the position vector $\boldsymbol{r}$ onto the cartesian reference frame. Then, it holds that $\mathrm{r}_{NS} = \mathrm{r}(L_k)$. Design variables at this stage are summarized in Table 5.6. Note that gradient-based solvers only accept scalar functions to be minimized or maximized. Therefore, the user has to select to optimize time of flight, propellant mass, final power available or a weighted sum of them. MOLTO-OR Step 2 uses the Interior Point solver Ipopt [109] to solve the resulting Large-scale NLP and the AMPL [220] programming language as interface, which employs automatic differentiation to compute gradients of the objective and constraint functions of the pertinent optimization problem.

4.4 Results

In this section, MOLTO-OR will solve two different orbit raising problems to deploy a satellite into GEO. The two most common injection orbits, GTO and LEO, will be used as departure orbits. The classical orbital elements for the orbits involved are presented in Table 4.6. The spacecraft mass, and the solar array characteristics for each case are summarized in Table 4.7. The solar arrays are comprised by Gallium Arsenide (GaAs) cells [221]. The proton non-ionizing energy loss $S_P(E)$ values for different energy levels are determined from Ref. [222]. The solar array shielding is capable of stopping protons of all energies less than or equal to 5 MeV. The contribution of energies greater than 1000 MeV is neglected. Two different electric thrusters and one chemical engine are considered. They are summarized in Table 4.8. The magnitude of the chemical firings are limited up to 3 km/s. The initial date is set for 1 January 2000 for all the simulations, which are performed using a Intel Core i7 (2,5GHz) computing system.

The proposed scenarios, i.e., GTO-GEO and LEO-GEO will be used to carry out four main analysis. Firstly, the unconstrained fully electric transfer for the GTO-GEO case is studied in detail. The use of the Q-law Lyapunov function will be evaluated and compared to other promising near-optimal low-thrust control law in terms of runtime and optimality. In particular, the approaches developed by Gao [64] and by Hudson and Sheeres [65] will be considered. Secondly, the penalties in the objective function associated to the imposition of operational constraints will be evaluated for the GTO-GEO case using MOLTO-OR Step 2. Thirdly, the capabilities of MOLTO-OR to concurrently optimize CP, EP and CCEP with respect to time of flight, propellant mass and solar-cell degradation will be studied for the LEO-GEO case. Finally, MOLTO-OR Step 1 will be allowed to optimally select the propulsion system and the transfer type for both the LEO-GEO and the GTO-GEO transfer cases.

Table 4.6: Orbit Parameters

Orbits	a/R_e	e	i (deg)	Ω (deg)	ω (deg)
LEO	1.086	0	28.5	0	0
GTO	3.820	0.731	27	99	0
GEO	6.6107	0	0	–	–

Table 4.7: Spacecraft parameters and transfer cases

Transfer	m_0 (kg)	P (kW)	C	D_x
GTO-GEO	450	5	0.2904	1.10E+9
LEO-GEO	1200	10	0.2904	1.10E+9

Table 4.8: Propulsive system options

Name	Type	m (kg)	I_{sp} (s)	η
E1	Electric	50	3300	0.65
E2	Electric	50	1500	0.65
C1	Chemical	200	330	-

4.4.1 Case 1: GTO-GEO Unconstrained Fully Electric Transfer

In this first case, the GTO-GEO transfer case is solved with MOLTO-OR Step 1 considering only a fully electric transfer with engine E1. Trajectories are optimized with respect to propellant mass and time of flight. Thereafter, the set of solutions corresponding to a mission duration of approximately 75, 100, 120, 150, 200 days was selected for feeding MOLTO-OR Step 2 . Fixed-time minimum-fuel solutions were obtained without applying any operational constraint. Results are compared with those obtained from implementing other predefined control laws. In particular, the blended control law proposed by Gao [64] and the Fourier-based approximation presented by Hudson and Sheeres [65] were implemented in MATLAB and IPOPT was used for obtaining minimum-fuel fixed-time transfers. They are compared with MOLTO-OR Step 1 in terms of performances for the same mission flight times. Finally, solutions from Gao's and Hudson's approach are used as initial guesses for MOLTO-OR Step 2 and results are compared with those obtained by using the initial guess provided by MOLTO-OR Step 1.

MOLTO-OR Step 1 solutions were obtained using the Genetic Algorithm parameters summarized in Table 4.9. Mutation and Crossover fractions were selected after a manual tuning process. Several runs were performed and the average total computational time, as well as per generation and per population are shown in Table 4.10. Note that the computational speed could be improved if more cores were used during the simulation. After all simulations reached the maximum number of generations, equally Pareto-optimal solutions were obtained. On average, the entire population was feasible by completing the 3^{th} generation. At the 10^{th} generation, 75% of the population lied along a distinct non-dominated front, whereas from generation 10 to 20 the front was progressively shifted towards lower times of flight and propellant masses. From generation 20 to 30, solutions spread along the Pareto front to generate a more uniform set. After 40 generations were completed, members of the population were uniformly distributed along the front and no later improvement, in terms of non-domination, was observed.

An example of one of the obtained Pareto fronts is shown in Figure 4.9. Note that increasing the time of flight implies decreasing the propellant mass required. Trajectories performances range from 60 days and 35.15 kg of propellant mass to 200 days and 28.1 kg of propellant. From the minimum time solution up to 90 days, the propellant required decreases at a rate of 228 grams per day. From 90 days to 120 days, it decreases at a rate of 56 grams per day. From 120 days to 200 days, it diminishes at a rate of 1.25 grams per day. In order to feed MOLTO-OR Step 2, five different solutions were selected: the closest ones to the the flight times 75, 100, 120, 150 and 200. Note that MOLTO-OR Step 1 does not allow to impose constrains to match a specific time of flight. However, due to the well spread Pareto front computed, solutions were found in the neighborhood of the selected times within a margin of the order of hours. The selected trajectories and their performances in terms of flight time and propellant mass are displayed in Table 4.11 along with the number of revolutions.

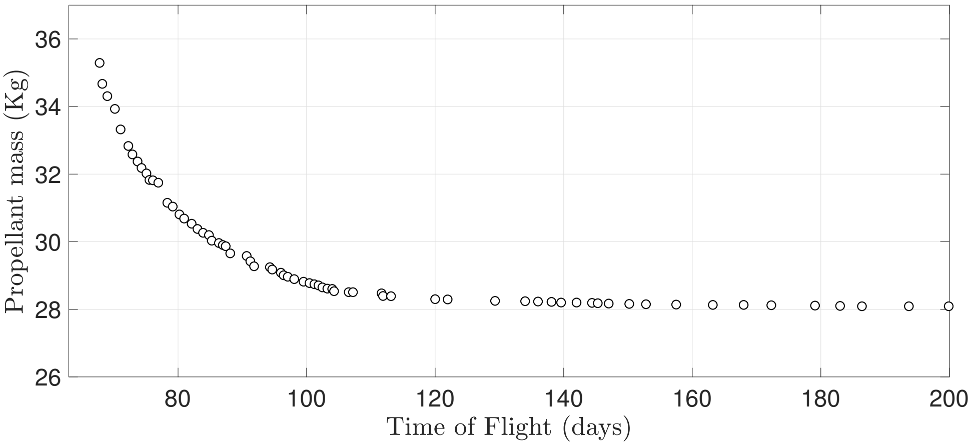


FIGURE 4.9: MOLTO-OR Step 1: GTO-GEO Pareto front

TABLE 4.9: MOLTO-OR Step 1: GTO-GEO Genetic Algorithm Parameters

Population size	100
Max. Generations	50
Mutation Fraction	0.3
Crossover Fraction	0.8

TABLE 4.10: MOLTO-OR Step 1: GTO-GEO Computational Time

Variable	Avg. CPU Time
Population	0.41 s
Generation	10.12 s
Total	8.54 min

TABLE 4.11: MOLTO-OR Step 1 selected solutions

Time (days)	67.621	74.912	100.072	120.010	149.931	199.572
Mass (kg)	35.158	33.321	28.821	28.475	28.264	28.134
Revolutions	95.26	101.68	156.91	184.76	229.27	279.84

MOLTO-OR Step 2 was run to obtain the minimum-propellant trajectory for each of the mission times under consideration. Two different grids were selected: one with a node spacing of 20 degrees in true longitude and the other one with 10 degrees separation in true longitude. Results are shown in Table 4.12 and Table 4.13 respectively. It can be noted that the finer the grid the more optimal the solution and the more computationally expensive the solutions. On average, MOLTO-OR Step 1 with 20 degrees node spacing consumes 1 kg less propellant. Solutions with the 10 degrees grid improves on average 37.8 grams with respect to the 20 degrees grid at the cost of doubling the computational time. Obtained trajectories for the 10 degrees grid are shown in Figures 4.13. The red color represents the shadow region, the clear blue the thrusting arcs, and the dark blue the coasting arcs. Note that when increasing the time of flight, the coasting arcs last longer, while the thrusting arcs are grouped around the apogee to both decrease the inclination and increase the perigee altitude more efficiently.

Table 4.12: MOLTO-OR Step 2 with Q-Law Initial Guess $\Delta L = 20$

Time (days) Mass (kg)	67.64 35.080	75 31.357	100 28.098	120 27.364	150 27.173	200 26.941
Revolutions	95.40	101.48	156.25	184.84	229.26	279.81
Iter. Runtime (min)	63 1.96	83 1.98	144 5.62	162 6.92	110 6.18	170 10.74
Variables Constraints	19,363 16,125	22,439 17,237	34,558 26,536	40,531 31,057	50,067 38,259	60,896 46,301

Table 4.13: MOLTO-OR Step 2 with Q-Law Initial Guess $\Delta L = 10$

Time (days) Mass (kg)	67.27 34.883	75 31.386	100 28.056	120 27.319	150 27.122	200 26.919
Revolutions	95.49	101.47	156.29	185.08	229.30	279.81
Iter. Runtime (min)	62 3.73	67 2.58	157 11.17	278 20.42	227 22.03	191 22.58
Variables Constraints	37,055 30,879	42,958 33,034	66,338 51,008	78,33 60,187	97,285 74,446	119,343 90,810

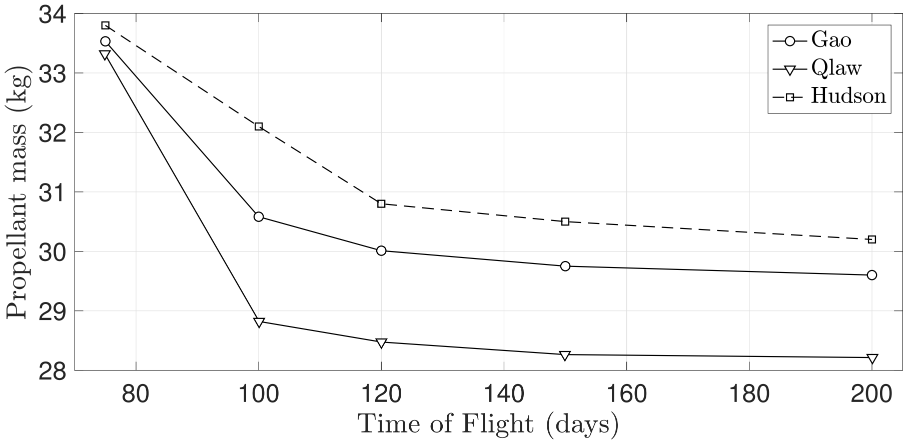


Figure 4.10: GTO-GEO: Propellant Mass vs Time of Flight

Now, the parametric control law proposed by Gao [64], and by Hudson and Sheeres [65] were implemented and optimized for the selected mission times, i.e. 75, 100, 120, 150 and 200 days, by means of propellant mass reduction. The former uses analytical orbital averaging and a blended control law including J_2 and Earth shadow effects. The latter models the thrust as a Fourier series to obtain analytical expressions without Earth oblateness of shadow effects. Both methods resulted in a nonlinear optimization problem that was solved with IPOPT. Each minimum-fuel fixed-time problem was solved independently, i.e. without using a genetic algorithm. Obtained performances in terms of flight time and propellant mass are compared with those obtained with MOLTO-OR Step 1 in Figure 4.10. It can be seen that solutions obtained by the Q-law outperforms those obtained with the other approximate methods. It is generally able to obtain a lower propellant mass for the same transfer time than Gao's or Hudson's approach.

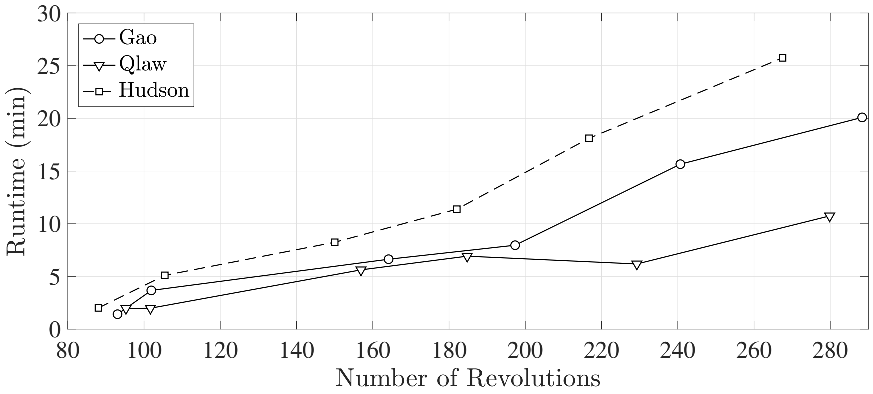


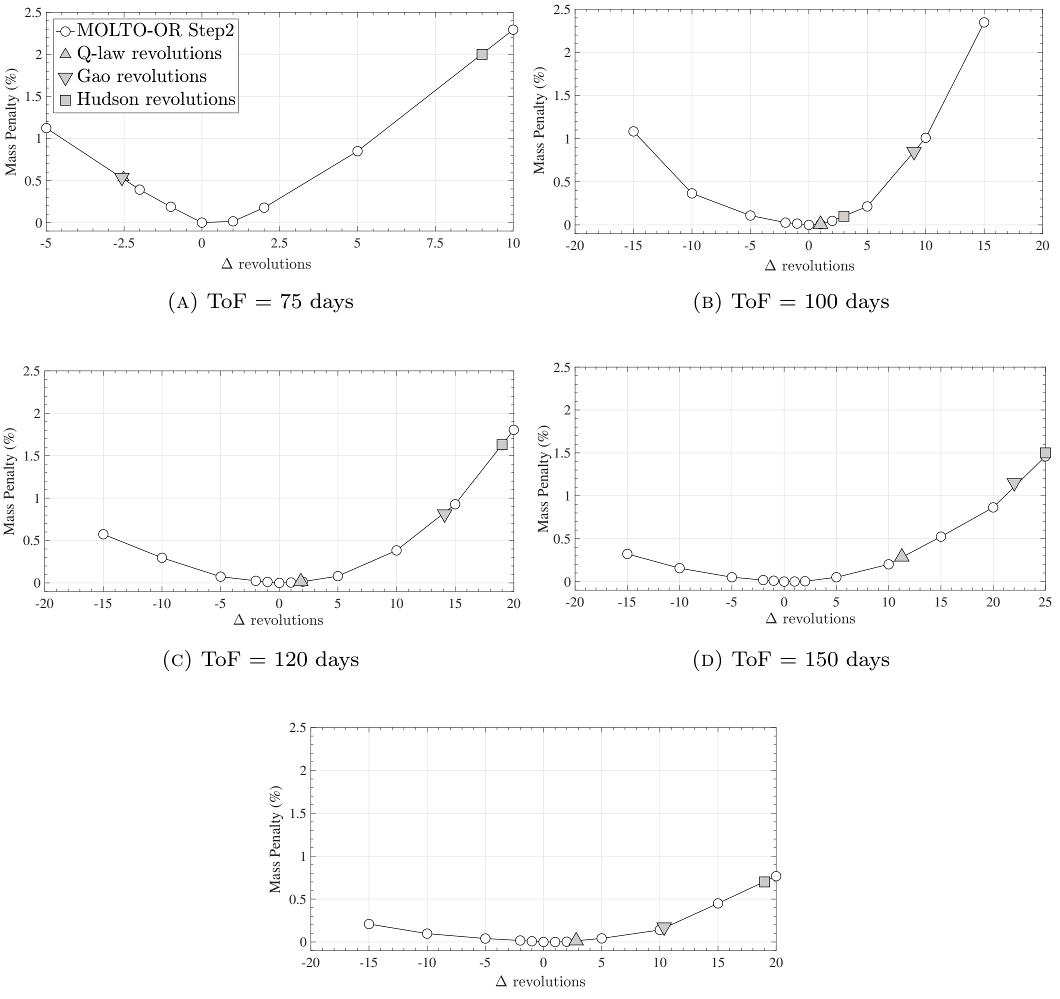
Figure 4.11: GTO-GEO: Number of Iterations vs Number of Revolutions

Table 4.14: MOLTO-OR: Number of revolutions

Case	75 days	100 days	120 days	150 days	200 days
MOLTO-OR Step 1	101	156	184	229	279
Gao [64]	101	164	197	240	288
Hudson [65]	114	159	202	243	296
Global Optimum	105	156	183	218	277

MOLTO-OR Step 2 algorithm was fed by the previous solutions obtained from the implementation of the Gao's and Hudson's method. The grid was chosen to be uniformly spaced in true longitude with a separation of 10 degrees. The running performances in terms of computational time are shown in Figure 4.11. It can be seen that MOLTO-OR Step 2 converges faster when using the initial guess provided by MOLTO-OR Step 1, especially for long transfers. Additionally, different performances in terms of flight time and propellant masses were obtained even though for the same mission time. It was found that the NLP solver did not vary the number of revolutions provided by the initial guess. In order to further investigate the influence of the number of revolutions for each case, a parametric study was performed. For such purpose, an additional constraint was imposed on the final true longitude, forcing the solver to deviate to a solution with a different number of revolutions. Results obtained are plotted in Figure 4.12.

Performances obtained and the number of revolutions predicted by the three methods are highlighted. When the number of revolutions was fixed, MOLTO-OR Step 2 converged to the same solution regardless of the initial guess. Therefrom, it can be deduced that for fixed-time minimum propellant mass problems, there exits a local optimal solution per each feasible number of revolutions. Furthermore, there is an optimal number of revolutions for which the propellant mass is globally minimal. The number of revolutions of each approach are compared in Table 4.14 along with the globally optimal value found in the parametric search. Trajectories obtained with the Q-law are closer to that optimum, differing less than a 0.5% in propellant mass. The existence of multiple local minima was previously addressed by Betts [55], and by Graham and Rao [223]. It can be attributed to the periodic nature of the trajectories and to the interaction of the nonlinear and non-convex constraints.



(A) ToF = 75 days

(B) ToF = 100 days

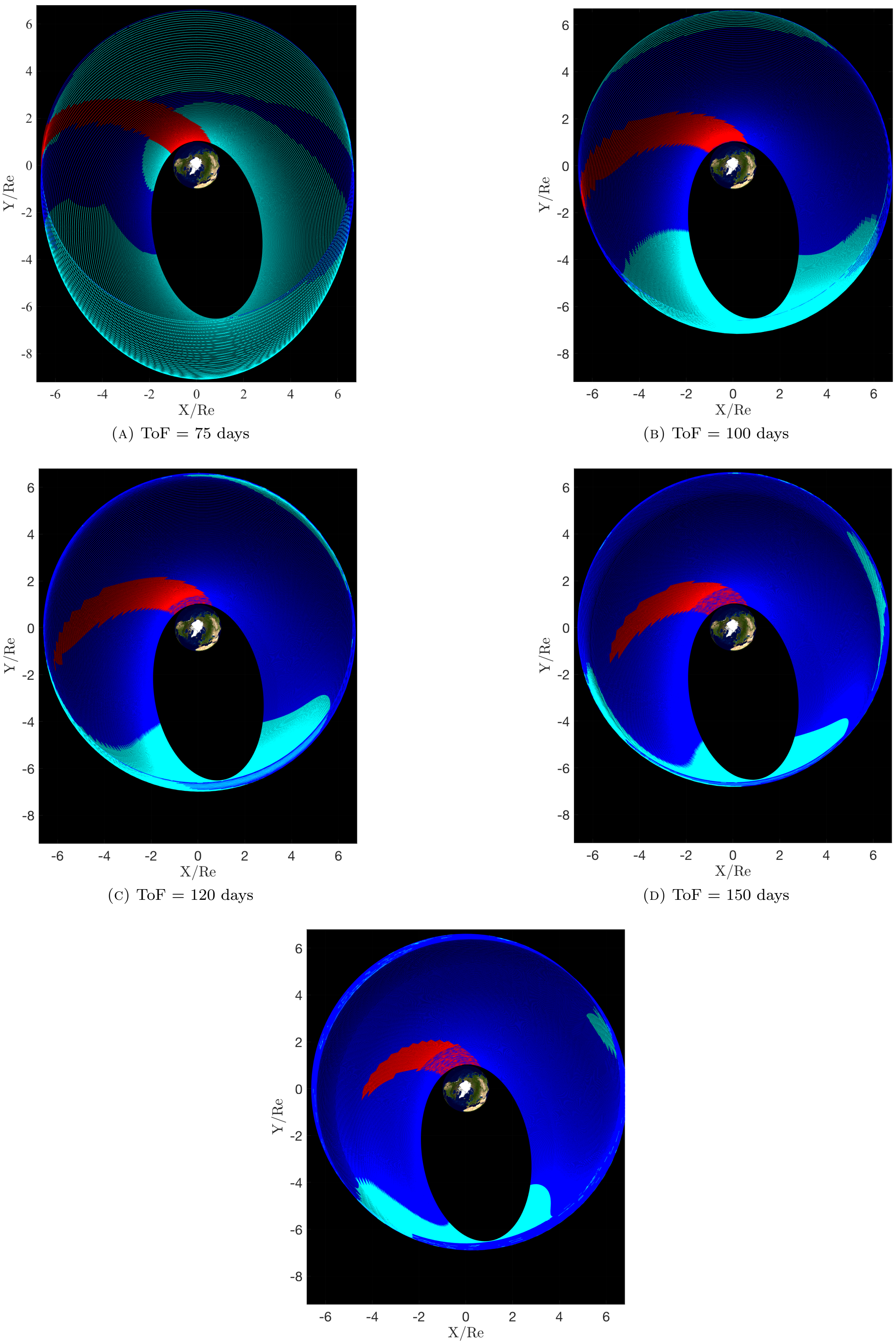
(C) ToF = 120 days

(D) ToF = 150 days

(E) ToF = 200 days

FIGURE 4.12: GTO-GEO: Locally optimal solutions for a fixed number of revolutions

Finally, the evolution of the semi-major axis, eccentricity and inclination for the trajectories obtained with the Q-law, Hudson's and Gao's approach are displayed in Figures **??**-4.20 respectively for 75, 100, 120, 150 and 200 days. They are compared with the best trajectory found during the parametric search with MOLTO-OR Step 2. Notably, the evolution of the orbital elements predicted by Hudson is the most different with respect to the optimal one presenting a completely different evolution of the semi-major axis. On the other hand, although the number of revolutions provided by the Q-law is the closest to the optimal number of revolutions, it cannot be stated that the evolution of the orbital elements provided by the Q-law is closer to the optimal one than the Gao's approach. The time history of the in-plane and out-of-plane thrusting angles from MOLTO-OR Step 1 and MOLTO-OR Step 2 are compared in Figs. 4.14 and Figs. 4.15 for the 75 days and 200 days transfers respectively. It can be noted that solutions from MOLTO-OR Step 2 exhibit a different thrust/coast sequence during sunlight than MOLTO-OR Step 1. Additionally, a maximum difference of 1 deg is observed. This fact may explain why the trajectories from both steps are significantly different.



(A) ToF = 75 days

(B) ToF = 100 days

(C) ToF = 120 days

(D) ToF = 150 days

(E) ToF = 200 days

FIGURE 4.13: MOLTO-OR Step 2: GTO-GEO Fully Electric Transfer

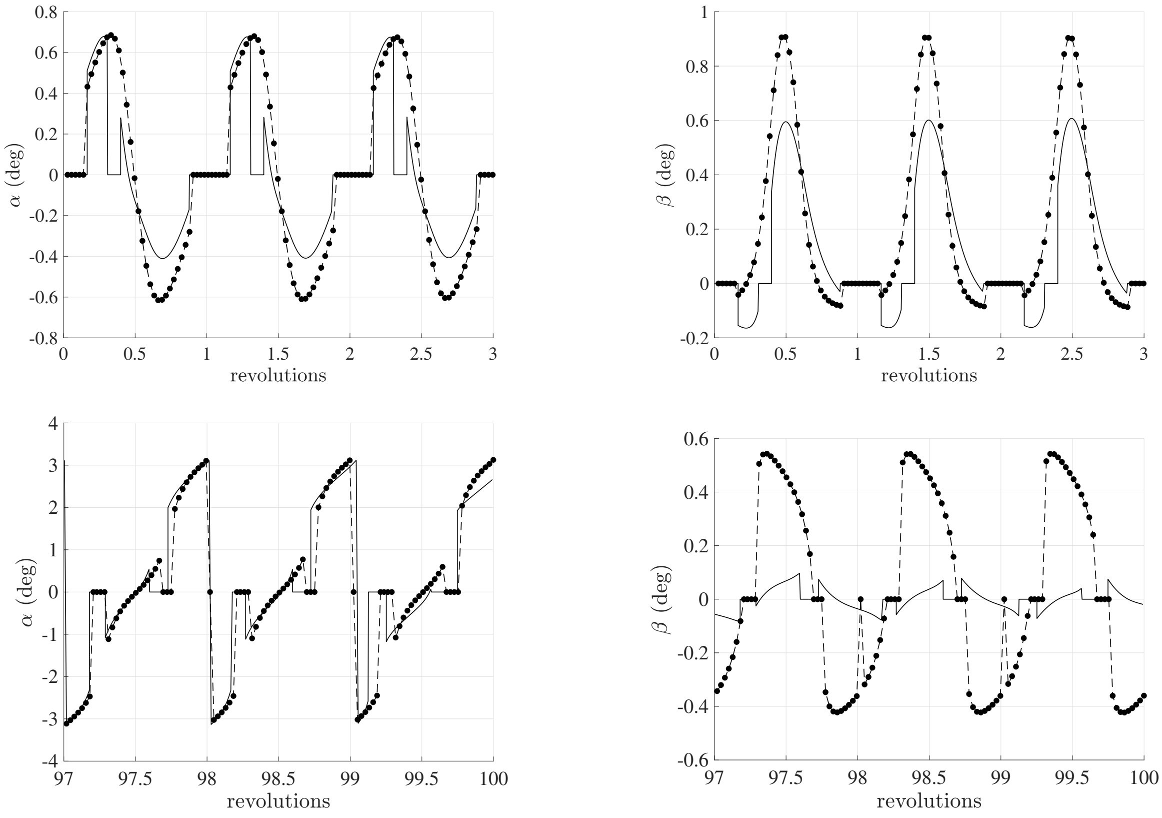


Figure 4.14: MOLTO-OR Step 1 (lines) and MOLTO-OR Step 2 (dots) steering laws for GTO-GEO: ToF = 75 days

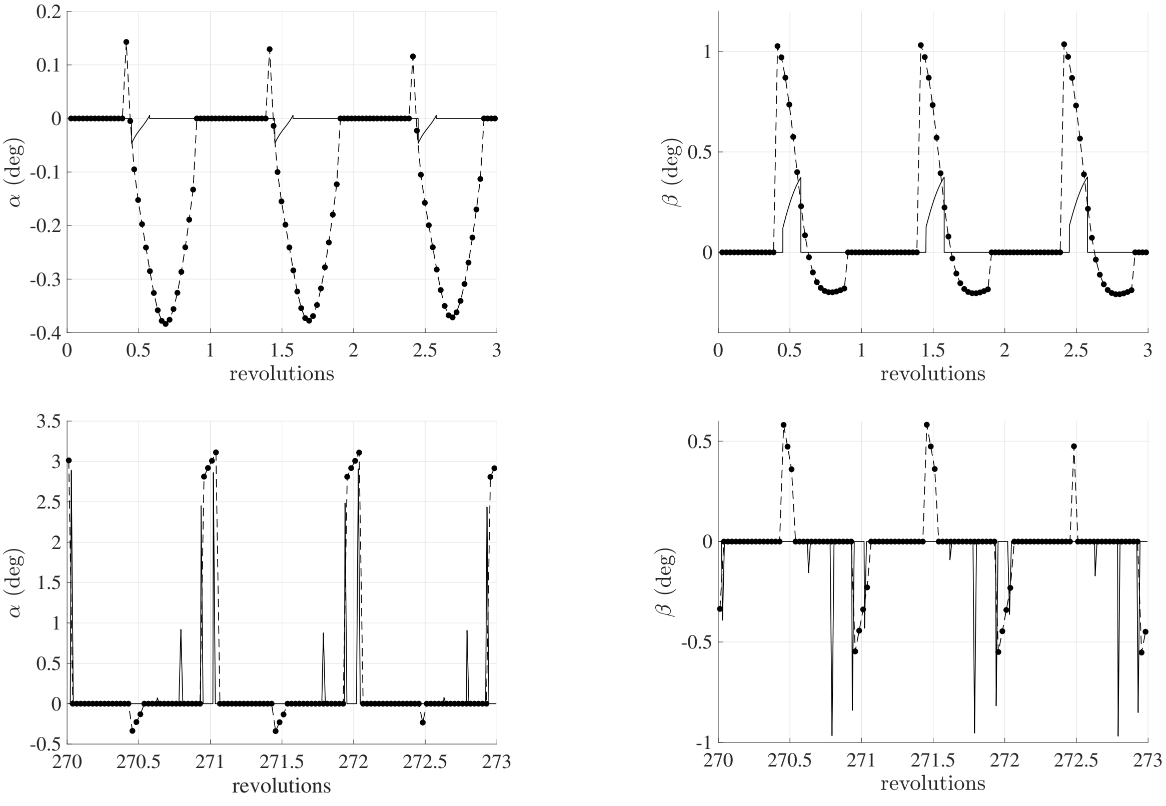
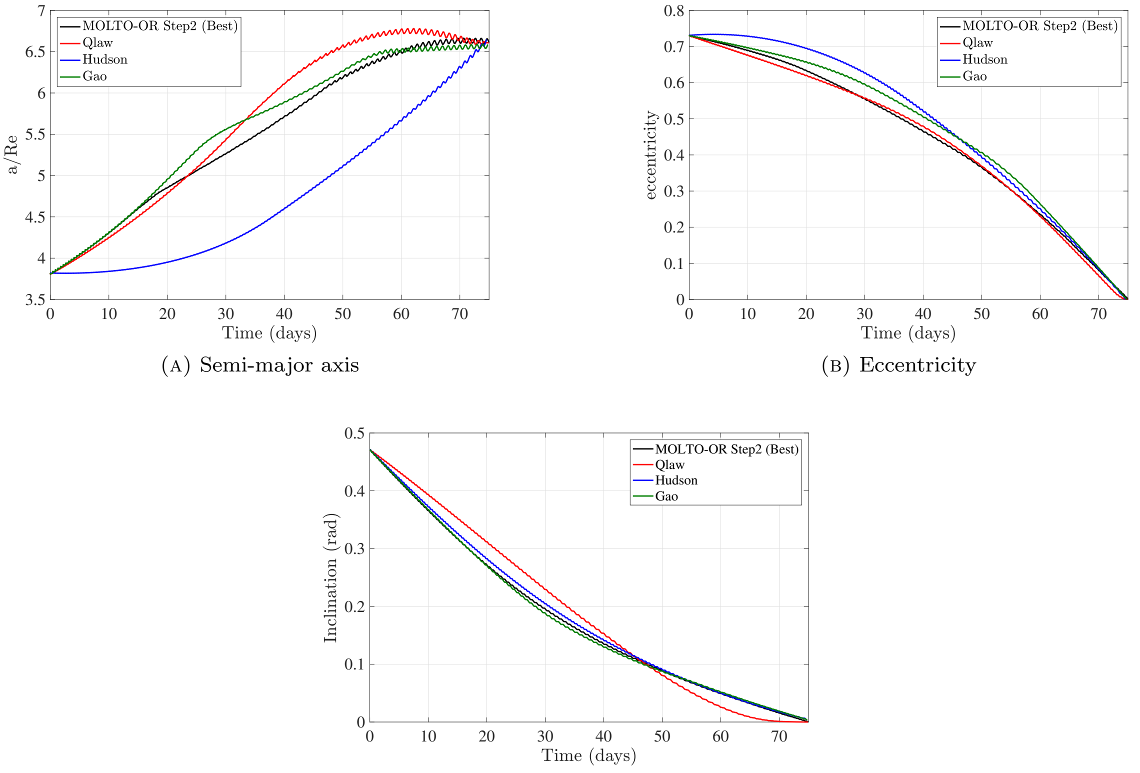


Figure 4.15: MOLTO-OR Step 1 (lines) and MOLTO-OR Step 2 (dots) steering laws for GTO-GEO: ToF = 200 days

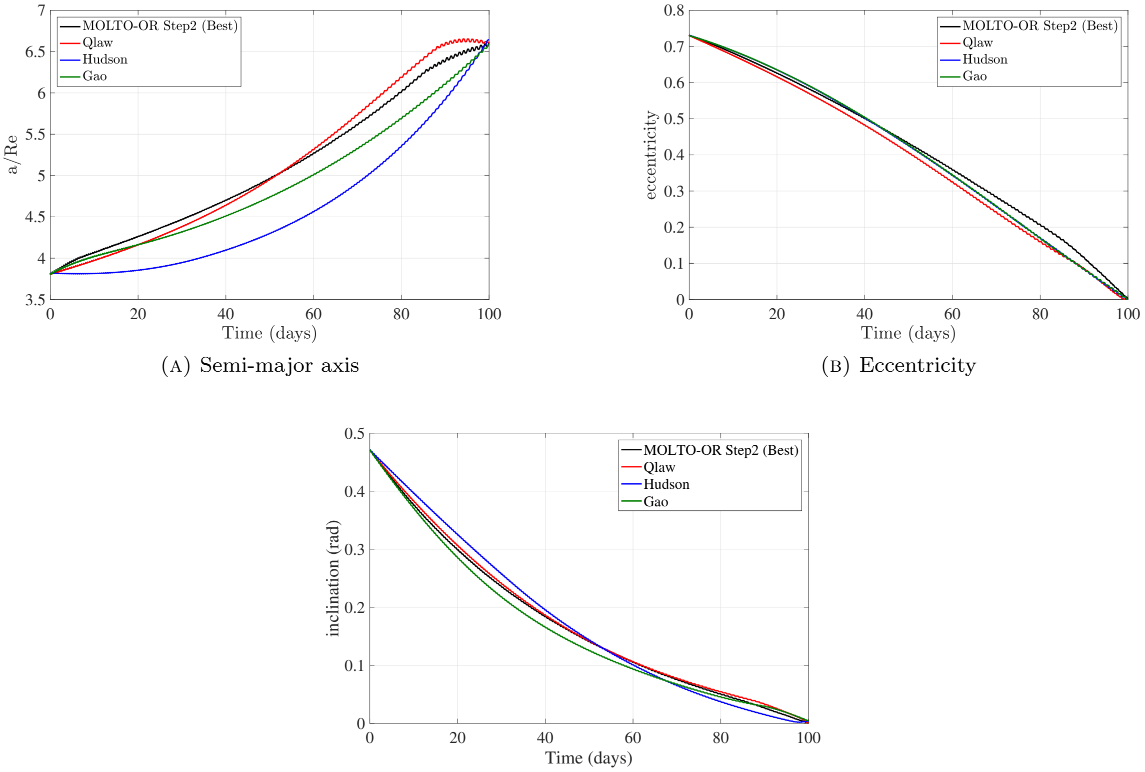


(A) Semi-major axis

(B) Eccentricity

(C) Inclination

FIGURE 4.16: GTO-GEO: ToF = 75 days

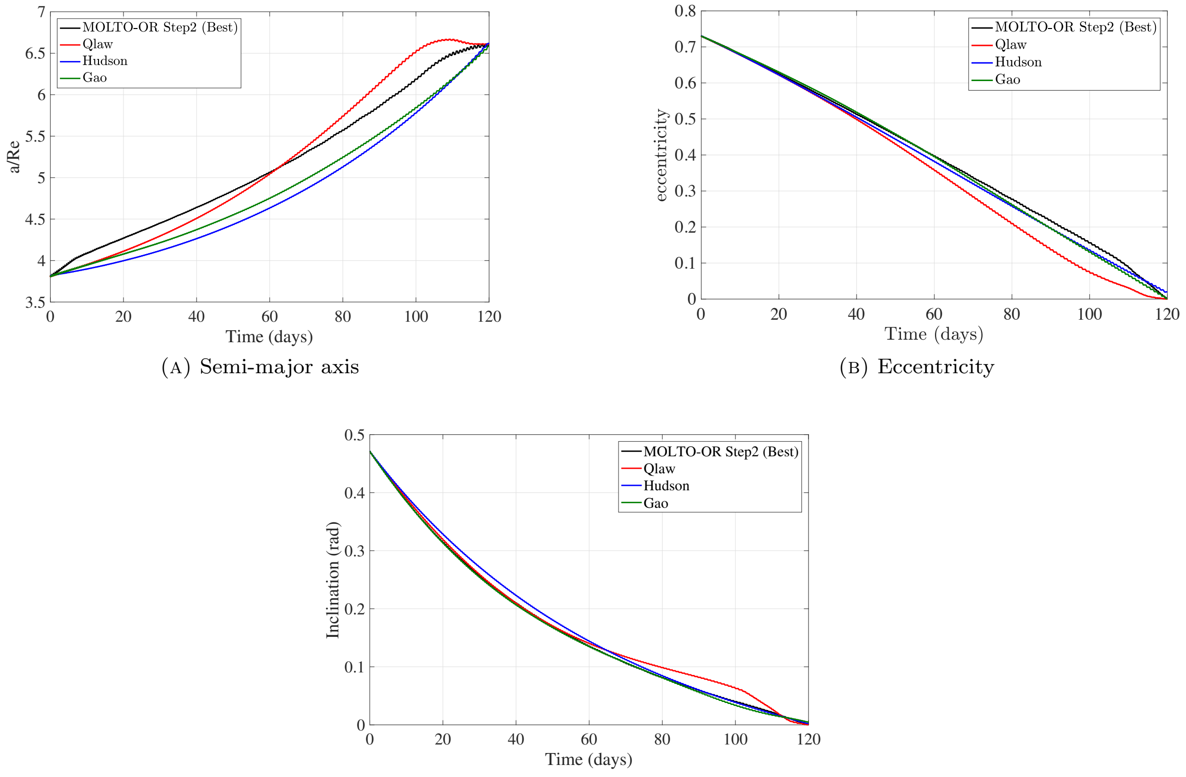


(A) Semi-major axis

(B) Eccentricity

(C) Inclination

FIGURE 4.17: GTO-GEO: ToF = 100 days

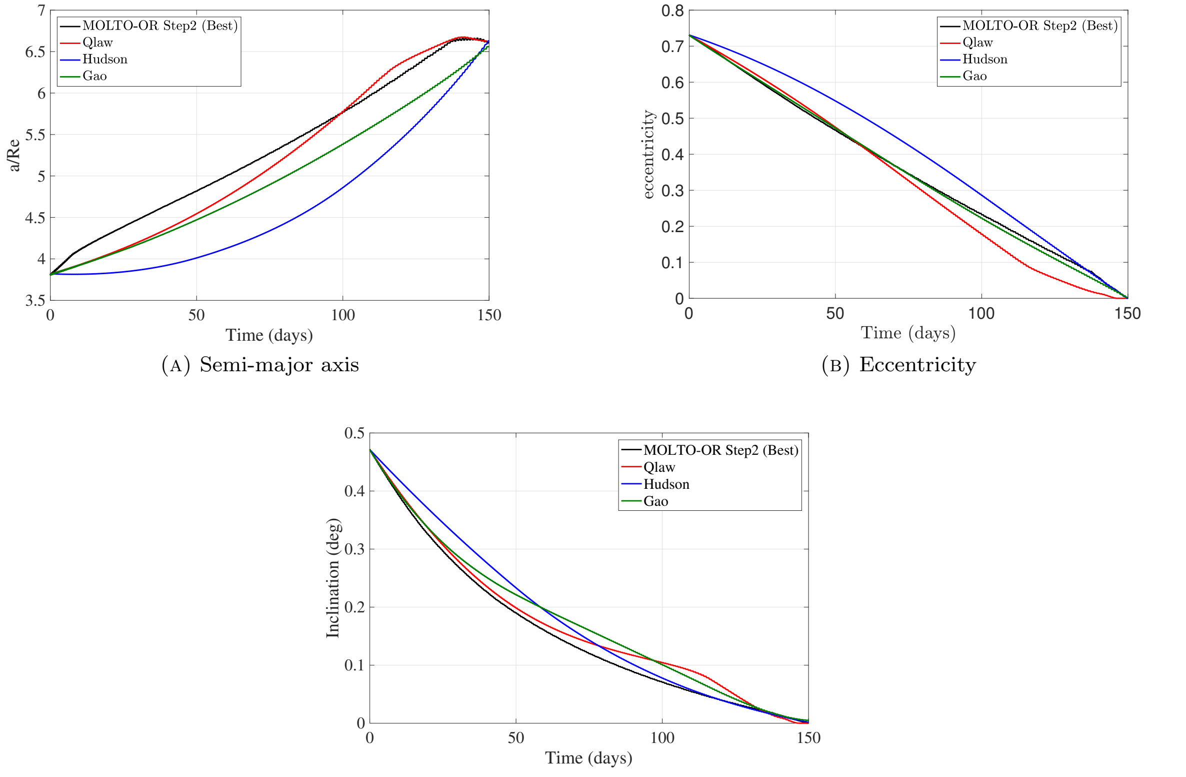


(A) Semi-major axis

(B) Eccentricity

(C) Inclination

Figure 4.18: GTO-GEO: ToF = 120 days

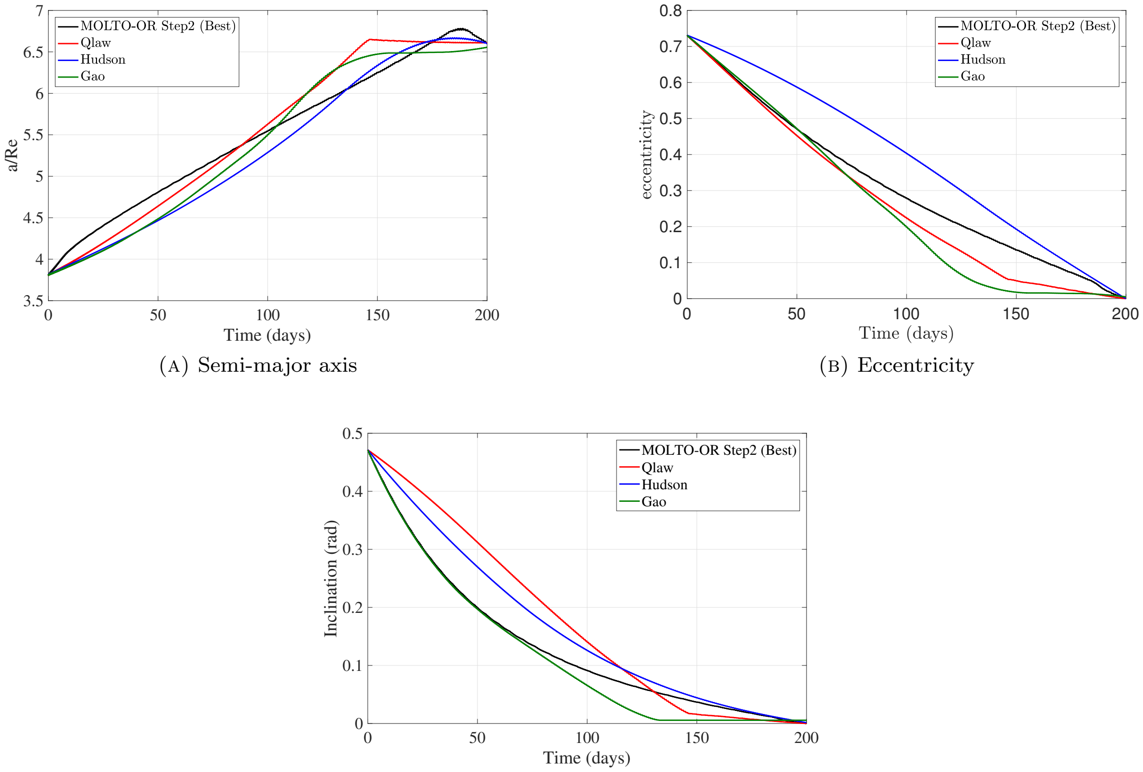


(A) Semi-major axis

(B) Eccentricity

(C) Inclination

Figure 4.19: GTO-GEO: ToF = 150 days



(A) Semi-major axis

(B) Eccentricity

(C) Inclination

FIGURE 4.20: GTO-GEO: ToF = 200 days

4.4.2 Case 2: GTO-GEO Constrained Fully Electric Transfer

In this second case, the previously solved unconstrained GTO-GEO transfer is now tackle imposing operational constraints. An analysis on how the application of constraints in MOLTO-OR Step 2 affects the objective function will be performed. The operational restrictions under consideration are: avoidance of the geostationary ring, a slew rate limitation of 25 deg/h and phasing at a longitude of 90 degrees. Each of them will be evaluated independently for the GTO-GEO transfer scenario for a series of fixed-time minimum fuel cases. We have selected the same trajectories as in the previous examples from MOLTO-OR Step 1 as initial guesses for the selected flight times: 75, 100, 120, 150 and 200 days. All simulations converged to a local optimal solution and the performances obtained are summarized in Table 4.15. The number of GEO ring crossings, the maximum angular velocity during the transfer, the final longitude in the GEO orbit, the computational time and the propellant mass penalty with respect to the unconstrained or nominal case are shown.

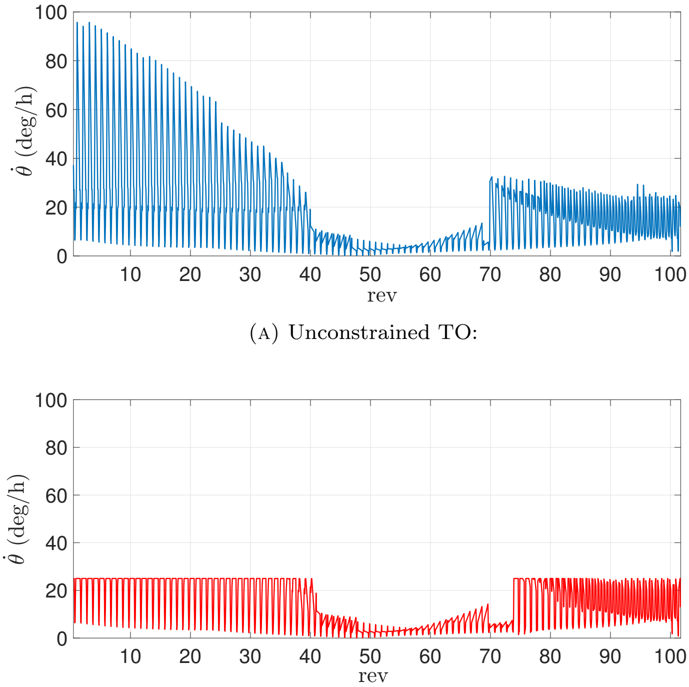
Regarding the GEO ring avoidance, it can be seen that the nominal trajectory with 150 days has the maximum number of crossings and therefore suffers from the largest penalty in the fuel mass. The avoidance of the 104 crossing can be accomplished by carrying less than 1% more fuel. It can be highlighted that avoiding crossings also increments the maximum angular velocity and it changes the arrival longitude. Illustrations of the trajectories nearby the GEO ring for

Table 4.15: GTO-GEO with Operational Constraints

		Transfer Time (days)				
	Variable	**75**	**100**	**120**	**150**	**200**
Case 1	GEO ring crossings	0	24	11	104	4
No Active	$\dot{\theta}_{max}(\text{deg}\,/h)$	100.16	52.62	52.31	43.19	45.41
Constraints	$lon(\text{deg})$	255.60	81.85	282.77	120.72	324.76
	$m_p(kg)$	31.28	28.05	27.31	27.04	26.91
	Runtime (min)	1.98	5.62	6.92	6.18	10.74
Case 2	GEO ring crossings	0	0	0	0	0
Active GEO	$\dot{\theta}_{max}(\text{deg}\,/h)$	100.16	55.78	55.86	53.47	46.57
Avoidance	$lon(\text{deg})$	255.60	81.26	345.93	217.14	320.10
	$\Delta m_p(\%)$	0	0.45	0.67	0.94	0.57
	Runtime (min)	3.23	9.22	10.45	12.23	13.47
Case 3	GEO ring crossings	1	24	11	104	4
Active Slew	$\dot{\theta}_{max}(\text{deg}\,/h)$	25	25	25	25	25
Rate Limits	$lon(\text{deg})$	255.65	81.94	264.31	89.03	285.63
	$\Delta m_p(\%)$	1.3164	0.0057	0.0044	0.0007	0.0037
	Runtime (min)	4.32	6.21	8.72	7.38	13.72
Case 4	GEO ring crossings	1	25	11	105	4
Active Slot	$\dot{\theta}_{max}(\text{deg}\,/h)$	105.07	63.51	47.71	54.95	61.19
Phasing	$lon(\text{deg})$	90	90	90	90	90
	$\Delta m_p(\%)$	0.21	0.035	0.73	0.11	0.038
	Runtime (min)	4.12	5.94	9.72	7.67	12.95
Case 5	GEO ring crossings	0	0	0	0	0
Active All	$\dot{\theta}_{max}(\text{deg}\,/h)$	25	25	25	25	25
Constraints	$lon(\text{deg})$	90	90	90	90	90
	$\Delta m_p(\%)$	4.07	3.02	5.37	6.37	2.01
	Runtime (min)	4.67	11.82	13.34	15.61	18.95

the nominal cases and without crossings are represented in Figures 4.23a-4.23h. The trajectory is projected onto a rotating radial frame. It can be seen how the algorithm is able to eliminate all the crossings, except for the last one that is required to reach GEO. Trajectories crosses the GEO Ring at the beginning, mid and end of the transfer. The combination of increasing the argument of perigee while reducing inclination makes the radius of the ascending node increase and decrease continuously. This region tends to be above GEO as the time of flight increases due to the change of inclination starting at a higher apogee.

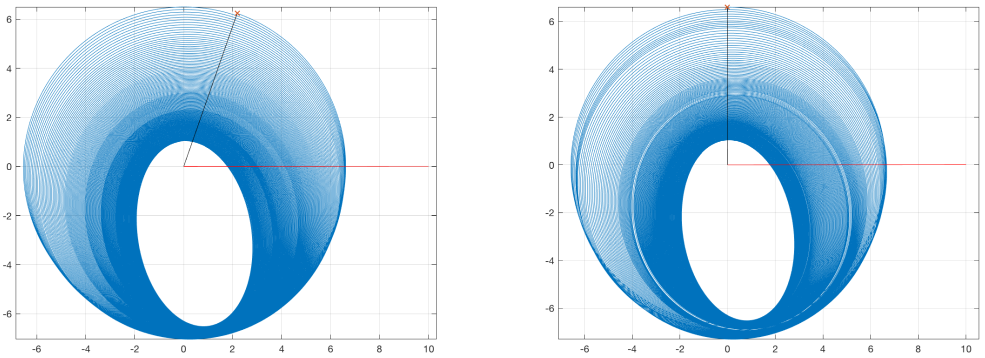
Imposing slew limits constraints has a more clear effect in the case of higher angular velocity, i.e., the minimum time case. For that case, the penalty is 1.31%, whereas for the others is three orders of magnitude less. As an illustration, in Figure 4.21 it can be compared the angular velocity of the thrust vector for the nominal case with the constrained one. As one may expect, the angular velocity is higher at the the beginning the transfer, as the orbital periods are smaller and maneuvers have to be performed faster. In the constrained case, it can be seen how the optimal solution consists on a saturated profile. Note that, the limitation in the angular velocity for the cases under consideration did not increase the number of GEO crossings.



(A) Unconstrained TO:

(B) Constrained

FIGURE 4.21: MOLTO-OR Step 2: Slot phasing constraint for ToF = 75 days

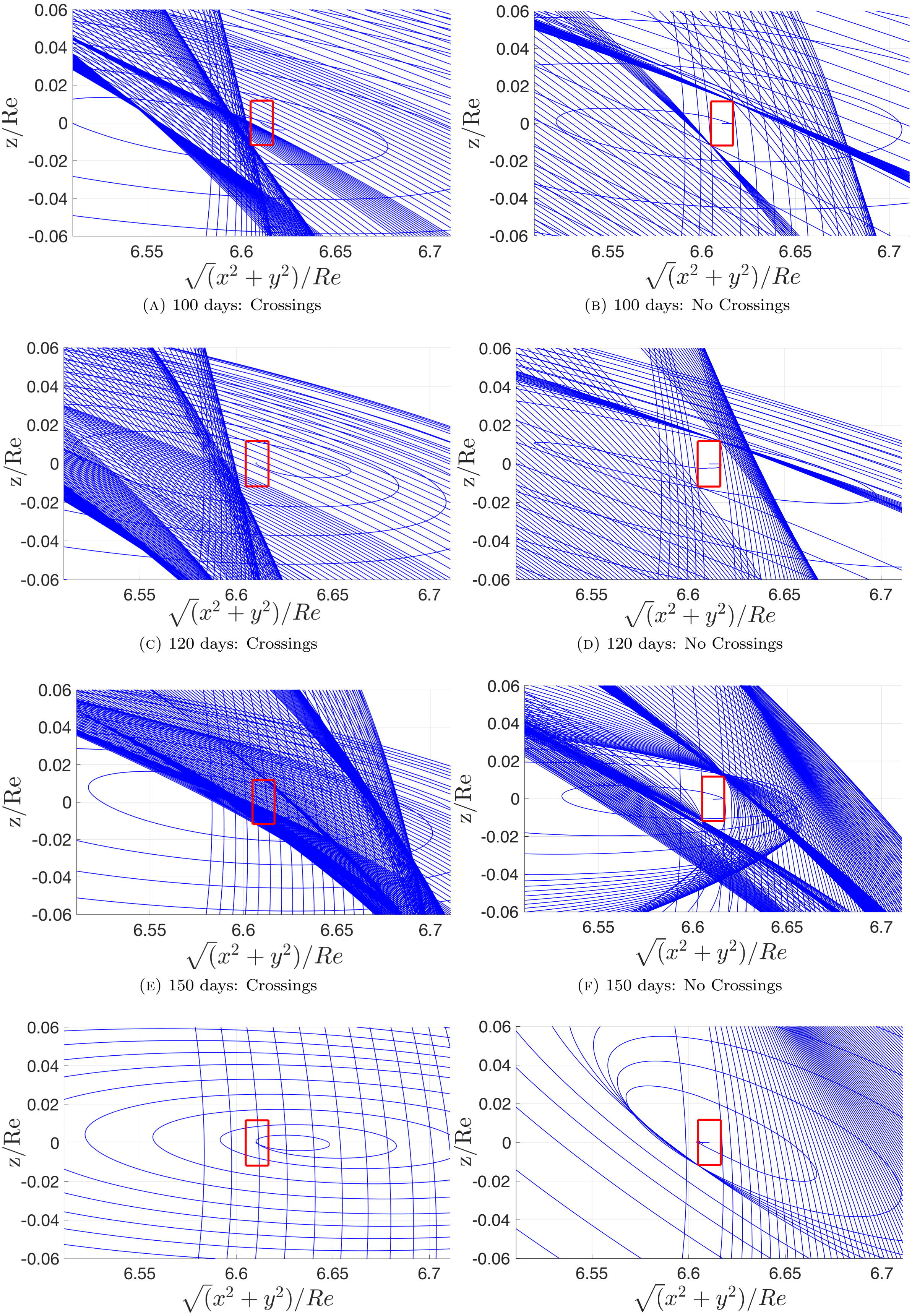


(A) Unconstrained

(B) Constrained

FIGURE 4.22: MOLTO-OR Step 2: slot phasing constraint for ToF = 100 days

An illustration of the application of the slot phasing constraint is depicted in Figure 4.22, where the cross-mark represent the arrival location. The trajectory is shifted from an optimal arrival longitude of 81 degrees to 90 degrees. The phasing constraint was found to have a larger penalty for the 120 days case. This can be explained as we are forcing the spacecraft to arrive almost at a complimentary slot, i.e. differing 180 degrees, to the locally optimal one, which in fact, is the worst case. In the last scenario all afore-mentioned constraints are imposed simultaneously. The 150 days transfer was found to be the most penalized. Note that the combined effect of all the constraints, i.e. Case 5, has a greater impact in propellant mass than just adding the penalties of each constraint separately, i.e. Case 2, 3 and 4. Furthermore, computational times for the constrained case double the ones for the unconstrained scenario.



(A) 100 days: Crossings

(B) 100 days: No Crossings

(C) 120 days: Crossings

(D) 120 days: No Crossings

(E) 150 days: Crossings

(F) 150 days: No Crossings

(G) 200 days: Crossings

(H) 200 days: No Crossings

Figure 4.23: MOLTO-OR Step 2: GEO Ring Detail

4.4.3 Case 3: LEO-GEO Free Transfer Type

In this third example, MOLTO-OR Step 1 is used to obtain the full 3D Pareto Front for the LEO-GEO transfer case using the E1 and C1 engines. Its capability to concurrently optimize FC, FET and CCET will be exploited. The objectives under consideration are the total time of flight, propellant mass consumed and radiation damage. The radiation damage is measured as the power loss of the solar array due to the displacement damage dose. The electric engine is assumed to use all the power available to the spacecraft. Thus, the thrust decreases with the radiation displacement dose. The chemical engine is constrained to provide a maximum of two impulses. The parameters selected for the genetic algorithm are summarized in Table 4.16, whereas the mean execution times of the simulation are shown in Table 4.17. Note that the computational time per population is lower than in the fully electric GTO-GEO case. This is due to the fact that hybrid transfer with reduced flight times are faster to compute. However, a fully electric LEO-GEO trajectory is more computationally expensive to evaluate than a purely fully-electric GTO-GEO, because the former typically implies a higher number of revolutions.

Table 4.16: MOLTO-OR Step 1: LEO-GEO Genetic Algorithm Parameters

Population size	200
Max. Generations	100
Mutation Fraction	0.3
Crossover Fraction	0.8

Table 4.17: MOLTO-OR Step 1: LEO-GEO Computational Time

Variable	Avg. CPU Time
Population	0.1920 s
Generation	9.8 s
Total	16.8 min

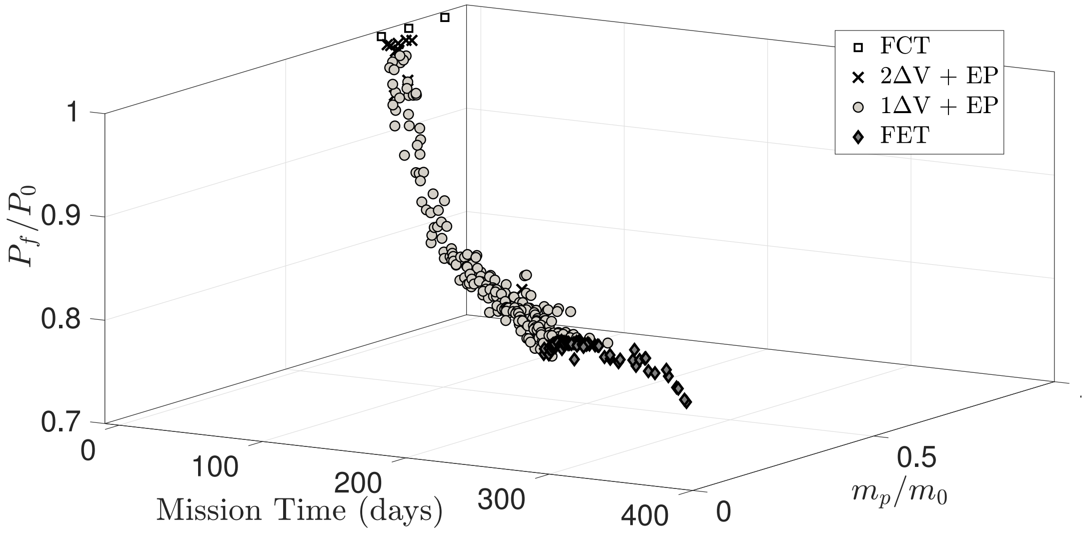


Figure 4.24: MOLTO-OR Step 1: 3D Pareto Front View for LEO-GEO

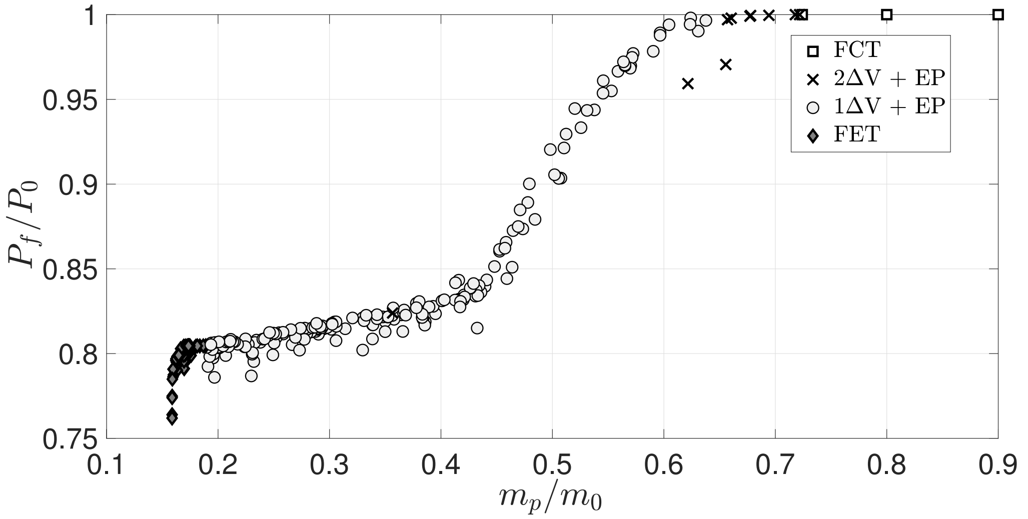


FIGURE 4.25: MOLTO-OR Step 1: Final power vs propellant mass Pareto for LEO-GEO

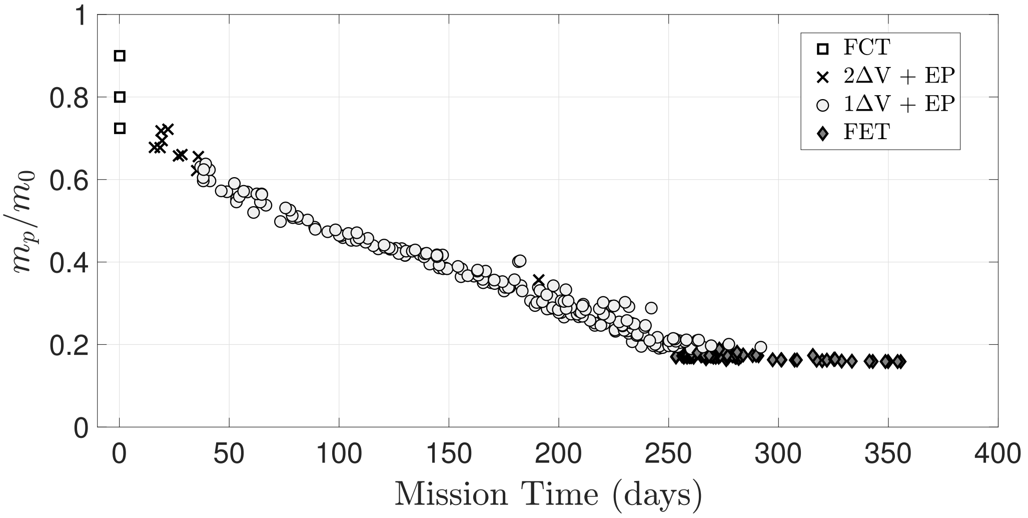


FIGURE 4.26: MOLTO-OR Step 1: Propellant mass vs mission time Pareto for LEO-GEO

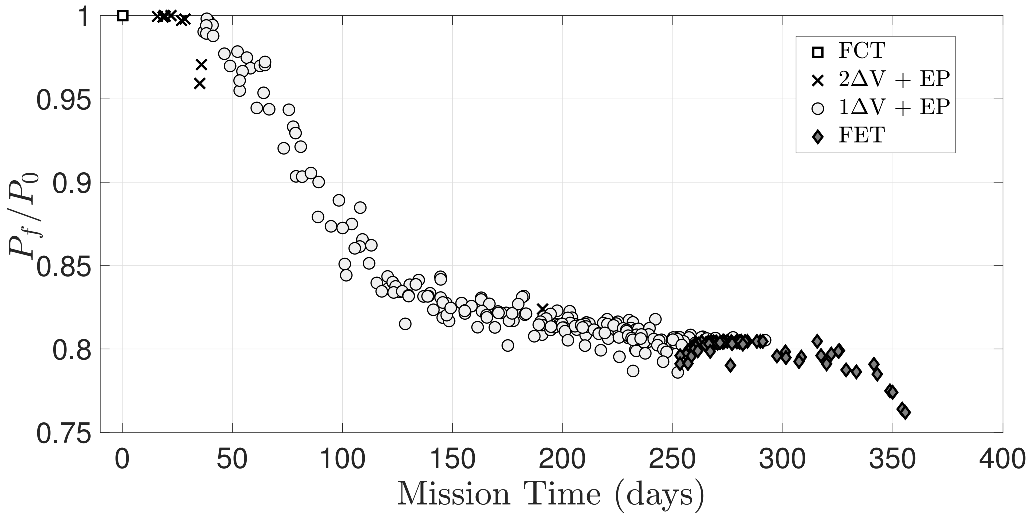


FIGURE 4.27: MOLTO-OR Step 1: Final power vs mission time Pareto for LEO-GEO

Results of the 3D Pareto Front and its corresponding projections are shown in Figures 4.24-4.27. It can be seen that FCT are optimal for short transfer times (less than 1 day), avoiding the Van-Allen radiation Belts at the cost of consuming propellant mass fractions higher that 73%. Therefore, the power of the solar arrays is not degraded. FET are optimal for transfer times longer than 250 days for minimum fuel trajectories whereas for minimum radiation damage they are not optimal until 300 days. Propellant mass fractions are within the range of 18-20% and they decrease at a rate of 0.006% per day. The solar array power losses are between 18-24% and increase at a rate of 0.12% per day. Intermediate performances are obtained with CCET consisting on 2 chemical impulses followed by a low-thrust arc for trajectories shorter than one month, and on 1 chemical impulse before the electric phase for longer transfers. It can be seen that the propellant mass consumed for CCET decreases at a constant rate of 0.18% per day of transfer time increased. However, regarding the radiation damage two different regimes are identified: until 125 days the power loss increases at a rate of 0.17% per day, whereas for longer transfers it occurs at a rate of 0.03% per day.

The different total $\Delta V's$ for each propulsion system, i.e. chemical and electric, are represented in Figure 4.28 as a function of time. Two different cases are considered: minimum radiation damage trajectories and minimum propellant mass solutions. Note that in this scenario, the maximum ΔV for the CP is 5500 m/s whereas for the electric engine is 6700 m/s. Additionally, it can be seen that, for the same time of flight, dedicating more ΔV, i.e. more fuel for the chemical engine, and for the EP is desirable to reduce the radiation damage instead of the propellant mass. The difference between the ΔV of the electric engine for reducing the radiation damage and for minimizing the propellant mass is maximum for 200 days with a value of 1000 m/s. Finally, at 50 days transfer time the ΔV dedicated for the electric transfer exceeds the quantity dedicated for the chemical phase.

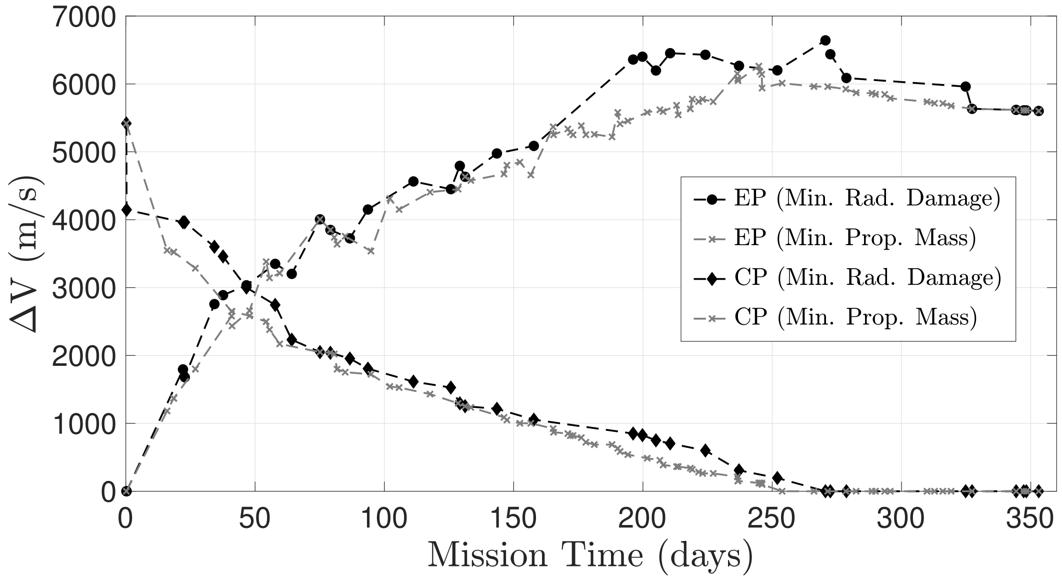


FIGURE 4.28: ΔV of the CP and EP vs Time of Flight for the minimum propellant mass and minimum radiation damage solutions

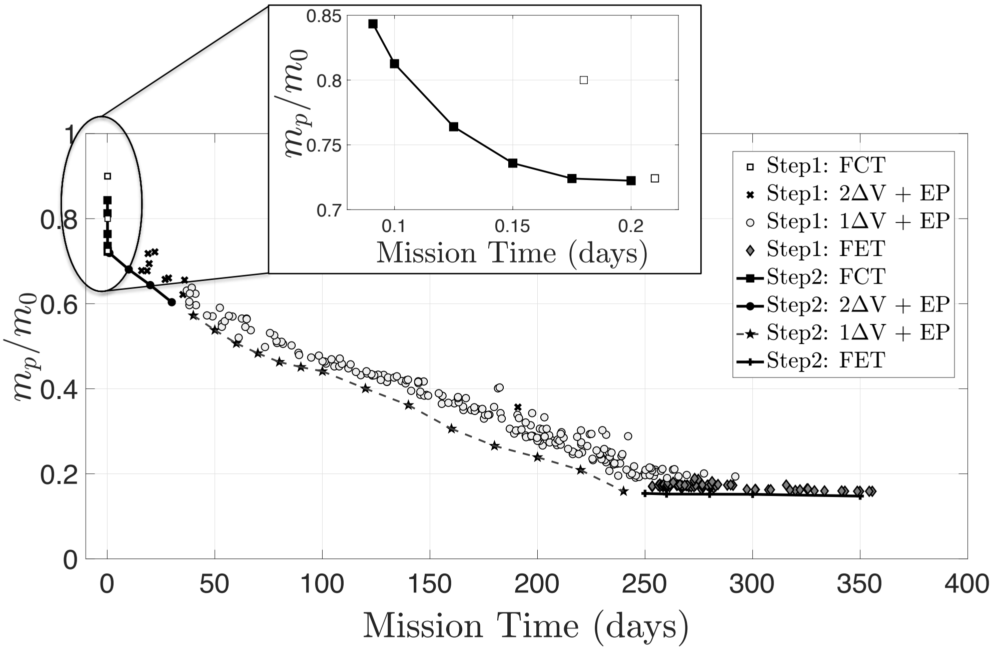
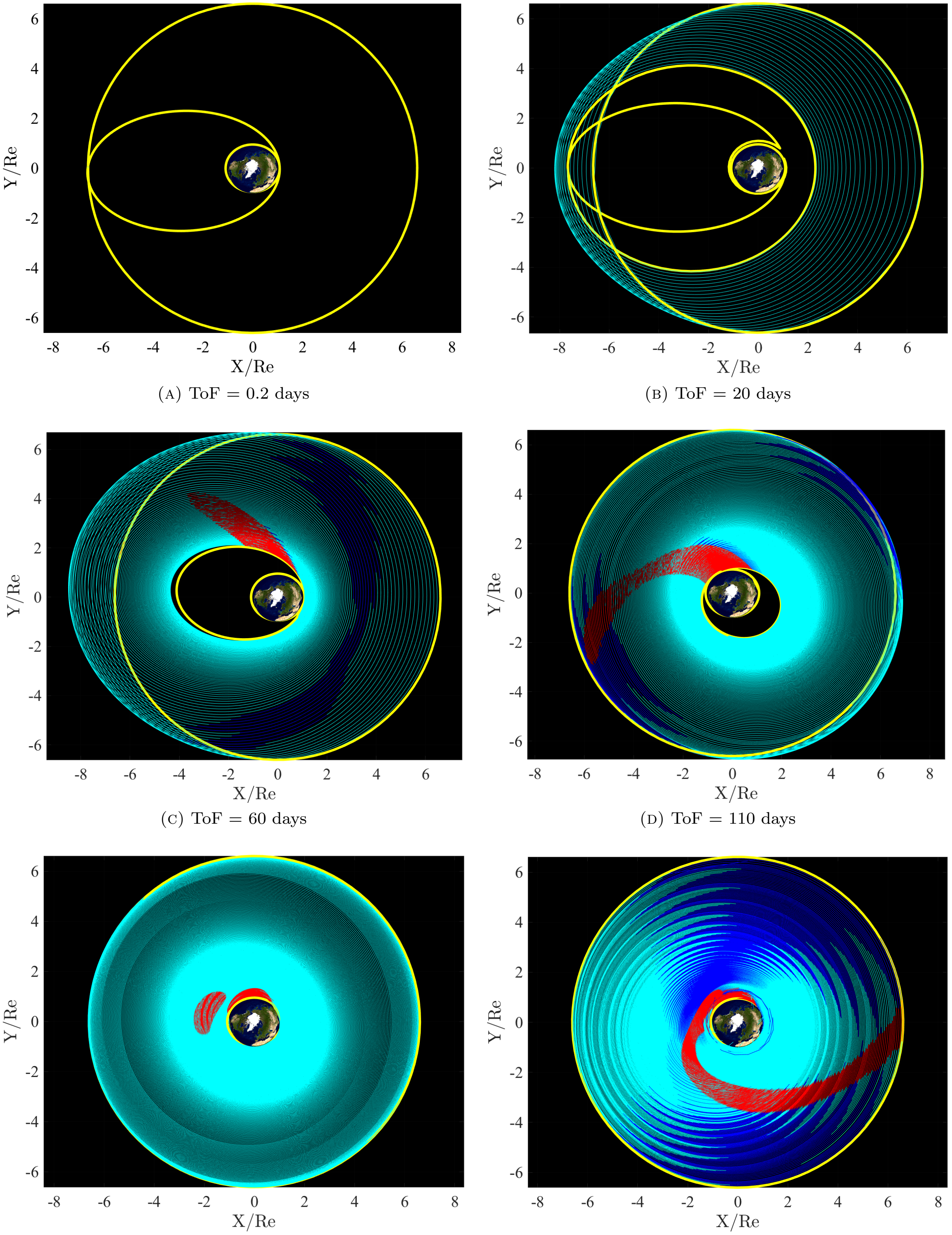


FIGURE 4.29: MOLTO-OR Step 1 and MOLTO-OR Step 2 Pareto fronts

In Figure 4.29, the minimum fuel-time performances obtained from MOLTO-OR Step 1 are shown and compared with the ones outputted from MOLTO-OR Step 2. In MOLTO-OR Step 2 different fixed-time minimum fuel problems were solved, using the trajectories from MOLTO-OR Step1. No operational constraints were considered. A detailed view of the fully chemical transfer is shown, where the point at 0.2 days corresponds to the Hohmann transfer and the solution at 0.05 days correspond to the minimum time solutions with a maximum allowable ΔV of 3 km/s. The average gain in terms of fuel by reoptimization with MOLTO-OR Step2 is 3%. As it can be noted that results provided by MOLTO-OR Step 1 are close to the optimal ones and makes the NLP solver converge within less than 1000 iterations and a tolerance of 10^{-6}. Regarding the computational time, it increases as the number of revolutions of the trajectory increases, ranging from a couple of minutes for FCT up to one hour for the FET with 350 days transfer, due to the number of variables that results from the discretization scheme.

In Figures 4.30a-4.30f, a set of representative trajectories are displayed, including FCT, CCET and FET. The red color represent the shadow region, the clear blue the thrusting arcs, and the dark blue the coasting arcs. Note that, a minimum-fuel hybrid transfer with a fixed flight time incorporates coasting arcs during the electric orbit raising. Therefore, as many authors previously did, it is not optimal to assume a constant thrust after the chemical phase. Trajectory 4.30b includes two-chemical impulses. The former raises the apogee higher than GEO, and the second one decreases the eccentricity of such orbit. In trajectory 4.30c a chemical firing raises the apogee to a lower value than GEO, and the electric engine then raises it above GEO. Trajectory 4.30e correspond to the minimum time fully electric transfer, and it consists on thrusting arcs during sunlight. In this case, the apogee always remain below GEO. Finally, trajectory 4.30f extends the transfer time up to 300 days, including coasting arcs in a region centered at the ascending and descending nodes to change the altitude more efficiently.



(A) ToF = 0.2 days

(B) ToF = 20 days

(C) ToF = 60 days

(D) ToF = 110 days

(E) ToF = 250 days

(F) ToF = 300 days

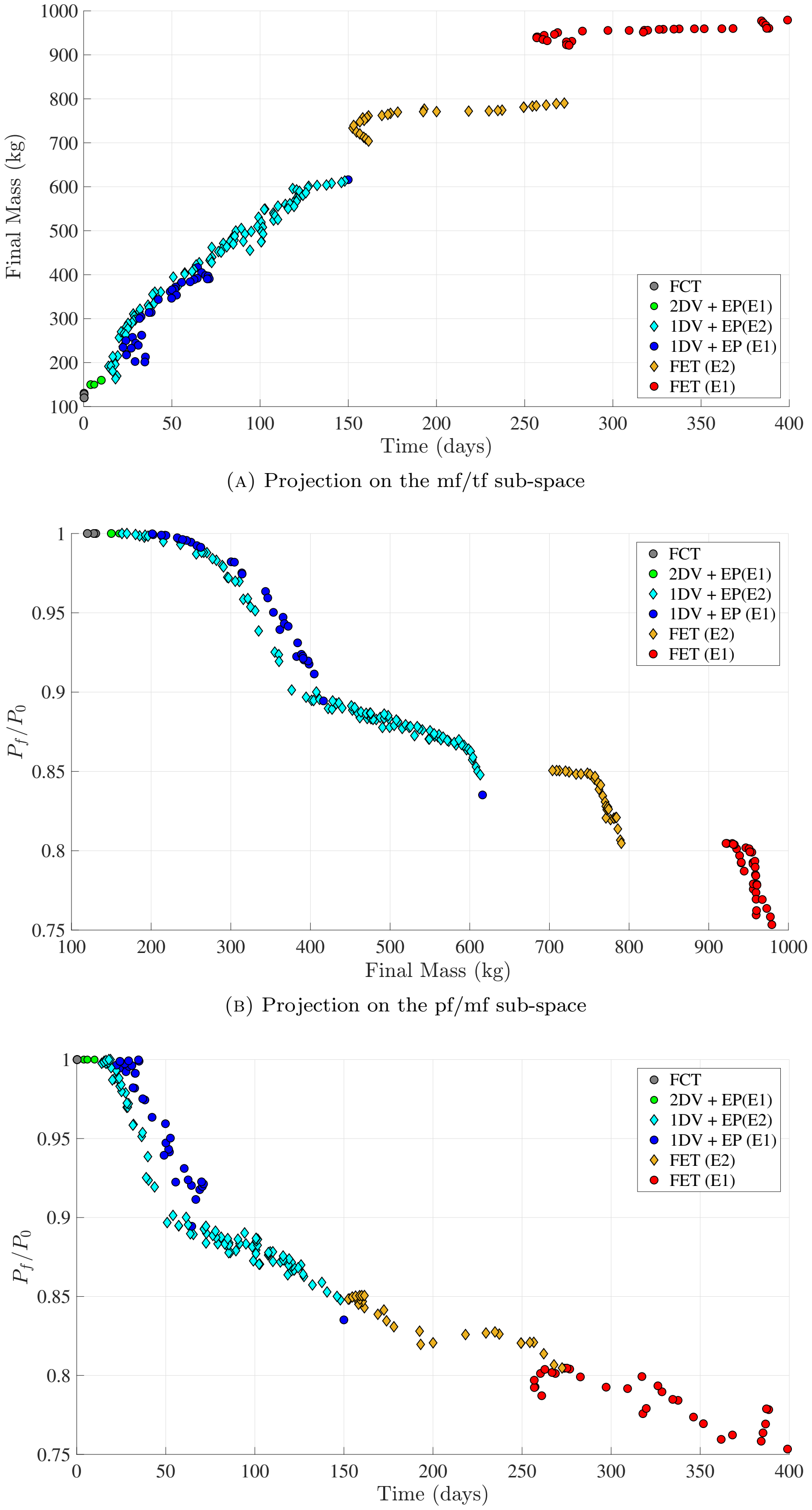
FIGURE 4.30: MOLTO-OR Step 2: Equatorial projection of the LEO-GEO Hybrid Transfer

4.4.4 Case 4: LEO-GEO and GTO-GEO Free Transfer Type and Propulsion

In this fourth case, the previous LEO-GEO and GTO-GEO transfers are solved with MOLTO-OR Step 1 with respect to final dry mass, time of flight and radiation damage. The algorithm is allowed not only to select the optimal transfer type but also is in charge of choosing among the electric engines E1 and E2 summarized in Table 4.8. Note that the latter exhibit a higher thrust yet a lower specific impulse, and thus a lower fuel efficiency than the former. It is also assumed that the chemical engine is jettisoned after the chemical phase and is constrained to produce two chemical firings at maximum. The Genetic Slgorithm parameters summarized in Table 4.16 were used for both cases. Similar computational times as the ones presented Table 4.17 were observed. The obtained Pareto fronts projections are displayed in Figures 4.31a-4.31c for the LEO-GEO case and in Figures 4.32a-4.32c for the GTO-GEO case.

For the LEO-GEO case, the hybrid transfer presents two different areas. From 25 to 50 days, the E1 propulsive system is more optimal to reduce the radiation damage, whereas E2 is preferred for maximizing the final mass. For the remaining part of the hybrid transfer, i.e. up to 150 days, E2 is optimal for both cases. Considering the fully electric transfer, E2 provides shorter transfer times, ranging from 150 to 275 days and final masses ranging from 700 to 800 kg. On the other hand, E1 provides longer transfer times, ranging from 250 to 400 days and final masses ranging from 900 to 1000 kg. These two propulsive systems compete within the time of flight interval ranging from 250 to 275 days. In such region of the Pareto, E2 provides better performances in terms of radiation damage than E1 for the same flight time, whereas E1 is more efficient in terms of maximizing final mass.

For the GTO-GEO case, the hybrid transfer is optimal with E2 for maximizing the final mass between 10 and 35 days, while E1 is optimal for minimizing the radiation dose between 10 and 60 days. Regarding the fully electric transfer, E2 provides shorter transfer times, ranging from 35 to 65 days and final masses ranging from 300 to 350 kg. In such region, a trajectory lasting 47 days, can deliver 345 kg and it will arrived with 0.98% power ratio, whereas a different steering law can deliver 295 kg, while arriving with 0.96% propellant ratio. On the other hand, E1 provides longer transfer times, ranging from 65 to 170 days and final masses ranging from 350 to 375 kg. In such region, a trajectory lasting 100 days, can deliver 370 kg and will arrive with 0.945% power ratio, whereas a different steering law can deliver 350 kg, while arriving with 0.978% propellant ratio.

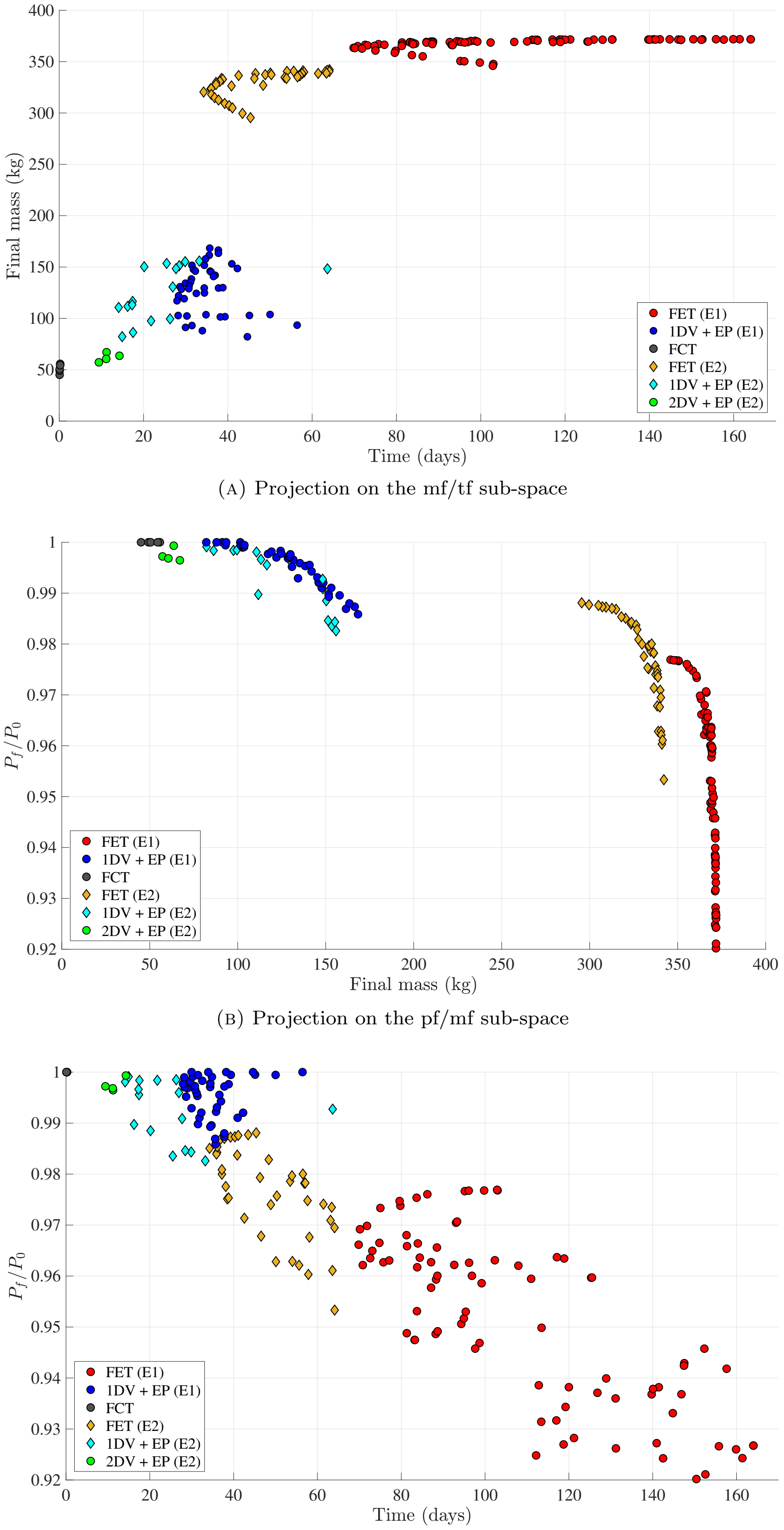


(A) Projection on the mf/tf sub-space

(B) Projection on the pf/mf sub-space

(C) Projection on the pf/tf sub-space

FIGURE 4.31: MOLTO-OR Step 1: GTO-GEO Pareto



(A) Projection on the mf/tf sub-space

(B) Projection on the pf/mf sub-space

(C) Projection on the pf/tf sub-space

FIGURE 4.32: MOLTO-OR Step 1: GTO-GEO Pareto

5

MOLTO-IT: Interplanetary Trajectories

Contents

5.1 Introduction

In this chapter, we present a solution approach to solve the second problem that is considered in this thesis. In this case, a spacecraft is to travel from a given departure planet to rendezvous or flyby a target body within the solar system. The spacecraft may benefit from gravity assisted maneuvers of other planets, as well as from the continuous thrust provided by an electric engine. These scenario well fits the case of deep-space probes that are injected into an Earth's hyperbolic space orbit to transfer to travel to far destinations. The goal is efficiently find the set of optimal Pareto solutions in terms of flight time and propellant mass consumption as described in chapter 1. Efficient means that a good compromise between computational time and accuracy of the solution is met. The solution has to comprise the optimal launch and arrival date, the sequence and configuration of the planetary encounters, as well as the time-history of the thrust magnitude and direction. Additionally, the initial mass of the spacecraft may be free and subject to optimization. Operational constraints to comply with launch asymptote declination constraints and fixed reorientation times during the transfer may apply.

In a similar fashion to chapter 4, the problem under consideration is formulated as a Multi-Objective Hybrid Optimal Control Problem (MO-HOCP). Gravity assists and the on-off switchings of the electric engine are modeled as discrete events. Conversely the continuous dynamics is represented by heliocentric coasting and thrusting arcs. The interconnection between continuous and discrete dynamics is represented in Figure 5.1 as a finite state machine diagram. A solution algorithm termed MOLTO-IT (Multi-Objective Low Thrust Trajectory Optimizer for interplanetary transfers) is proposed. It consists on two sequential steps: MOLTO-IT Step1 and MOLTO-IT Step2. Each of them solves the problem at different degrees of fidelity and with different computational performances: the former provides a low-fidelity solution with low-computational effort, whereas the latter produces a medium-fidelity solution with more computational time. Both steps are interconnected since the solution from MOLTO-IT Step 1 is used as an initial guess for MOLTO-IT Step 2. Hereafter, the modeling of the hybrid dynamics, along with the solution approach applied at each stage will be introduced separately.

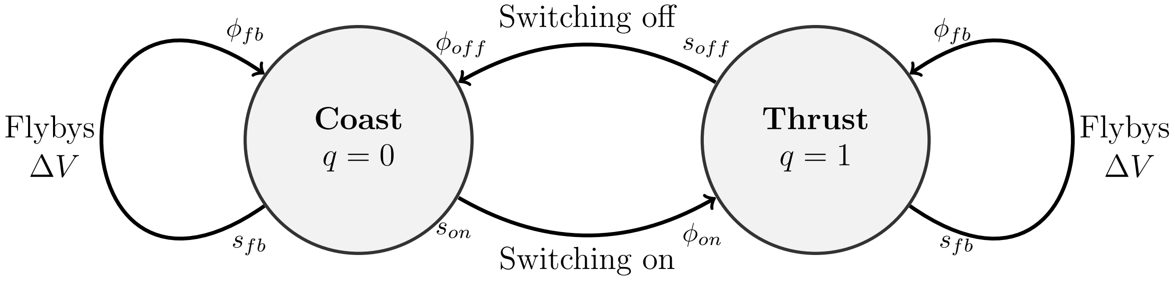


FIGURE 5.1: MOLTO-IT: Spacecraft Hybrid Dynamical System Diagram

5.2 Modeling

5.2.1 MOLTO-IT Step 1

The spacecraft is assumed to always lie in the same orbital plane. Its continuous state vector is represented by the set of intrinsic coordinates $\boldsymbol{x}(t) = [\mathrm{r}(t), \mathrm{v}(t), \psi(t), \theta(t)]$. Here $\mathrm{r} \in \mathbb{R}$ is the distance to the Sun, $\mathrm{v} \in \mathbb{R}$ is the velocity with respect to an inertial reference frame, which, without loss of generality, is centered in the Sun, $\psi \in \mathbb{R}$ is the flight path angle and θ is the polar angle with respect to principal direction of the inertial frame. Besides, the categorical state variable $q(t) \in \{0, 1\}$ determines the working condition of the electric engine, when '0' implies coasting and '1' demands thrusting. Additionally, when thrusting, the magnitude and direction of the acceleration vector can be adjusted as required by the continuos control parameter $\xi \in \mathbb{R}$.

Continuous Dynamics

The evolution of the continuous state of the spacecraft, $\boldsymbol{x}(t) = [\mathrm{r}(t), \mathrm{v}(t), \psi(t), \theta(t)]$, is modeled as a particle moving in a central gravity field under the action of a perturbing acceleration $\boldsymbol{a}_p$, i.e., as a Perturbed Two-Body-Problem. The equation of motions are described by the following set of differential equations:

$$\mathrm{f}: \begin{cases} \dfrac{d\mathrm{v}}{dt} = -\dfrac{\mu_s}{r^2}\cos\psi + \mathbf{a}_p \cdot \mathbf{t} \\ \mathrm{v}\dfrac{d}{dt}(\psi + \theta) = \dfrac{\mu_s}{r^2}\sin\psi + \mathbf{a}_p \cdot \mathbf{n} \\ \dfrac{d\mathrm{r}}{dt} = \mathrm{v}\cos\psi \\ \dfrac{d\theta}{dt} = \dfrac{v}{r}\sin\psi \end{cases} \tag{5.1}$$

In the above, μ_s represents the gravitational constant of the Sun and $\mathbf{a}_p$ is the perturbing acceleration. The perturbing acceleration $\mathbf{a}_p$ has different expressions depending on the discrete state of the spacecraft:

$$\mathbf{a}_p : \begin{cases} \mathbf{a}_p = \dfrac{\mu_s}{\mathrm{r}^2}(\xi\cos\psi\,\mathbf{t} + (1 - 2\xi)\sin\psi\,\mathbf{n}), & \text{if} \quad q = \mathit{1} \\ \mathbf{a}_p = 0, & \text{if} \quad q = \mathit{0} \end{cases} \tag{5.2}$$

Note that, when thrusting, the spacecraft is subject to a predefined acceleration profile. The one presented here was introduce by Roa et a. [205, 206]. The acceleration is projected onto an intrinsic reference frame, which is defined by the following unitary vectors:

$$\mathbf{t} = \frac{\mathbf{v}}{||\mathbf{v}||}, \qquad \mathbf{b} = \frac{\mathbf{r} \times \mathbf{v}}{||\mathbf{r} \times \mathbf{v}||}, \qquad \mathbf{n} = \mathbf{b} \times \mathbf{t} \tag{5.3}$$

where $\mathbf{v} \in \mathbb{R}^3$ is the velocity vector of the spacecraft. The vectors $\mathbf{t} \in \mathbb{R}^3$ and $\mathbf{n} \in \mathbb{R}^3$ are directed along the tangential and normal direction respectively and are contained in the orbital plane. Figure (5.2) depicts the geometry of the problem referred to the inertial reference frame.

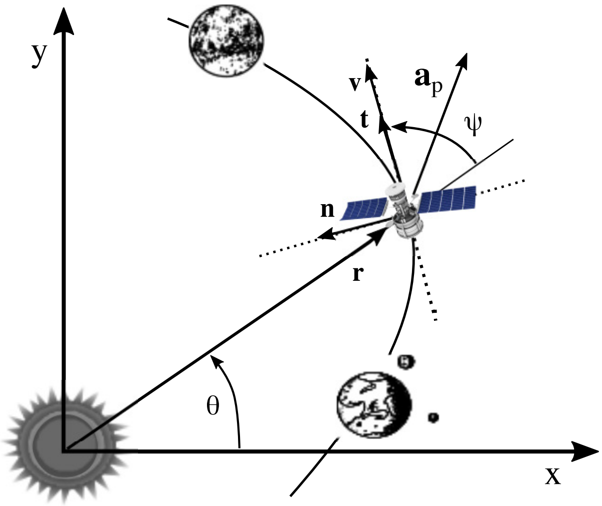


FIGURE 5.2: MOLTO-IT Step 1: Geometry of the problem

Notably, Roa et al. [205] derived two constants of motion $K_1 \in \mathbb{R}$ and $K_2 \in \mathbb{R}$ using the continuous dynamical equations of Eqs.(5.1) and the predefined control law of Eq.(5.2). They are extensions of the laws of conservation of energy and angular momentum respectively. In particular, they can be solved from the initial conditions $(r_0, \mathrm{v}_0, \psi_0)$:

$$K_1 = \mathrm{v}_0^2 - \frac{2\mu}{r_0}(1-\xi) \tag{5.4}$$

$$K_2 = \mathrm{r}_0\mathrm{v}_0^2 \sin\psi_0 \tag{5.5}$$

Making use of K_1, K_2 and ξ, they obtained closed-form analytical solutions for the trajectory and time, as a function of the polar angle θ, avoiding the need to explicitly or implicitly impose the constraints related to the continuous equation of motion. Depending on the sign value of the constant K_1, three families of solutions are obtained: elliptic ($K_1 < 0$), parabolic ($K_1 = 0$), and hyperbolic ($K_1 > 0$). There are two types of hyperbolic spirals: spirals of type I correspond to $K_2 < (2(1-\xi))$, whereas spirals of type II satisfy $K_2 > (2(1-\xi))$. Each spiral type exhibits different properties:

- **Elliptic spirals** ($K_1 < 0$),: the trajectory is bounded and never escapes to infinity.
- **Parabolic spirals** ($K_1 = 0$): the particle reaches infinity with zero velocity along a spiral branch.
- **Hyperbolic spirals** ($K_1 > 0$): the trajectory exhibits an asymptote when approaching infinity, where the velocity is finite and nonzero

An example of each spiral type is illustrated in Figure 5.3. The analytical solutions for the trajectory are summarized in Table 5.1 as a function of the constants $c_1, c_2, c_3, c_4, c_5, c_6, c_7$ and c_8, which can be derived from the initial conditions and the control parameter ξ, and of the $\varpi(\theta)$ that represents the spiral anomaly. For a complete analytical representation and detailed derivations the reader is referred to [206].

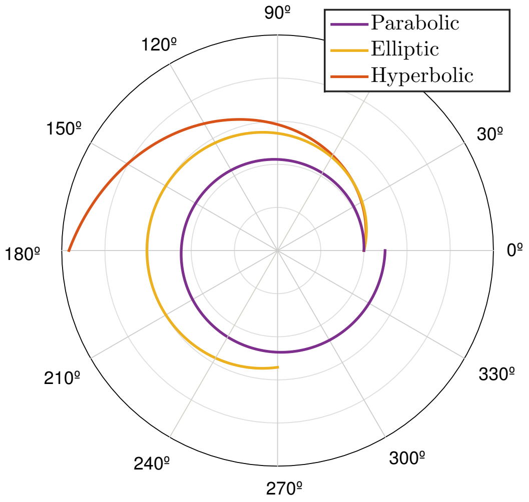


FIGURE 5.3: Families of controlled generalized spirals

TABLE 5.1: MOLTO-IT Step 1: Parametric shape-based model

Family	Trajectory
Elliptic	$\mathrm{r}(\theta) = \dfrac{1}{c_1 + c_2 \cosh \varpi(\theta)}$
Parabolic	$\mathrm{r}(\theta) = c_3 e^{(c_4\theta - 1)}$
Hyperbolic I	$\mathrm{r}(\theta) = \dfrac{1}{\sinh \dfrac{\varpi(\theta)}{2}(c_5 \sinh \dfrac{\varpi(\theta)}{2} + c_6 \cosh \dfrac{\varpi(\theta)}{2})}$
Hyperbolic II	$\mathrm{r}(\theta) = \dfrac{1}{c_7 + c_8 \cos \varpi(\theta)}$

Therefore, the continuous state of the spacecraft is completely determined as a function of the polar angle θ by providing K_1, K_2 and ξ, or, in other words, by the initial states $\boldsymbol{x}_0 = [\mathrm{r}_0, \mathrm{v}_0, \psi_0, t_0]$ at θ_0 and ξ. Hence, it holds:

$$\boldsymbol{x}(\theta) = \boldsymbol{X}_T(\boldsymbol{x}_0(\theta_0), \theta_0, \xi; \theta) \tag{5.6}$$

where X_T is a closed-form analytical representation of the trajectory when thrusting. Additionally, when the spacecraft is coasting mode, another analytical solution can be derived from Eqs.(5.1): a Keplerian arc. Therefore, its state can be easily determined as a function of the polar angle given the state $\boldsymbol{x}_0$ and θ_0 when the coasting begins as follows:

$$\boldsymbol{x}(\theta) = \boldsymbol{X}_C(\boldsymbol{x}_0(\theta_0), \theta_0; \theta) \tag{5.7}$$

where X_c is a closed-form analytical representation of the trajectory when coasting.

Additionally, an estimate of the propellant consumption can be made by assuming a constant specific impulse I_{sp} for the low-thrust engines. This simplification permits the required propellant mass to be expressed as a fraction of the initial spacecraft mass, based on the time integral of the thrust acceleration and the rocket equation:

$$\frac{m_p}{m_0} = 1 - e^{-\Delta V/(g_0 I_{sp})}, \qquad \Delta V = \int |\mathbf{a}_p| dt \tag{5.8}$$

where g_0 is the gravity of Earth at sea level. However, due to the computational cost of evaluating the elliptic integrals included in the definition of t, the integrand in Eq.(5.8), can be expressed in terms of the angular variable θ, resulting in a faster procedure:

$$\Delta V = \int |\mathbf{a}_p| \frac{\mathrm{r}}{\mathrm{v} \sin\psi \cos\psi} d\theta \tag{5.9}$$

Discrete Dynamics

The discrete dynamics of the spacecraft allows to characterize the effect of both performing flyby maneuvers and turning the engine on and off. In the following lines we define the set of discrete event functions, including discontinuity surfaces and transition map functions, necessary to include such effects in the model.

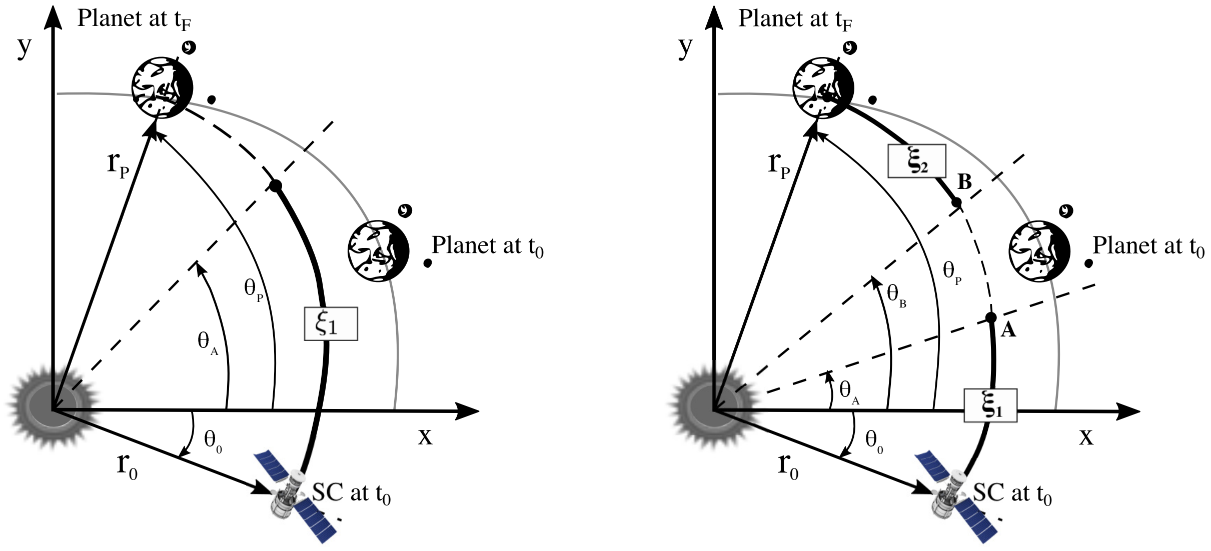
Engine on-off switching: In this work we predefine a thrust-coast sequence for an interplanetary leg where the target is to flyby the next planet. We assume that the spacecraft first traverses a generalized logarithmic spiral from θ_0 to an intermediate point θ_A, and then describes a Keplerian orbit from θ_A to θ_F (see Fig. 5.4a). Nevertheless, if the target is to rendezvous a planet the transfer leg will be decomposed in three arcs instead of just two (see Fig. 5.4b): a generalized logarithmic spiral from θ_0 to θ_A, a coast arc from θ_A to θ_B, and a second spiral segment from θ_B to θ_F. Therefore, the discrete on/off switchings events can be defined as:

$$\text{Switching-on} : \begin{cases} \phi_{on} : & q(t_i^+) = 1, \quad \mathrm{r}(t_i^+) = r(t_i^-), \quad \theta(t_i^+) = \theta(t_i^-), \\ & \mathrm{v}(t_i^+) = \mathrm{v}(t_i^-), \quad \psi(t_i^+) = \psi(t_i^-) \\ s_{on} : & \theta - \theta_0 = 0 \quad \text{or} \quad \theta - \theta_B = 0 \end{cases} \tag{5.10}$$

$$\text{Switching-off} : \begin{cases} \phi_{off} : & q(t_i^+) = 0, \quad \mathrm{r}(t_i^+) = \mathrm{r}(t_i^-), \quad \theta(t_i^+) = \theta(t_i^-), \\ & \mathrm{v}(t_i^+) = \mathrm{v}(t_i^-), \quad \psi(t_i^+) = \psi(t_i^-) \\ s_{off} : & \theta - \theta_A = 0 \end{cases} \tag{5.11}$$

They are modeled by the discontinuity surfaces s_{on} and s_{off}, and by the transition maps ϕ_{on} and ϕ_{off}. They express the condition of switching on the engine ($q = 1$) at the beginning of the leg (θ_0), then switch off ($q = 0$) when the spacecraft reaches θ_A, and finally switch on again when the polar angles equals θ_B. Consequently, the interplanetary leg for a trajectory to flyby on a planet can be analytically expressed as:

$$\boldsymbol{x}(\theta) = \boldsymbol{X}_F(\boldsymbol{x}_0(\theta_0), \theta_0, \xi_1, \theta_A; \theta) \tag{5.12}$$



(A) Flyby leg configuration

(B) Rendezvous leg configuration

FIGURE 5.4: MOLTO-IT Step 1: Transfer strategy

Similarly, the interplanetary leg for a trajectory to rendezvous with a planet can be expressed as a closed-form analytical function, such that:

$$\boldsymbol{x}(\theta) = \boldsymbol{X}_R(\boldsymbol{x}_0(\theta_0), \theta_0, \xi_1, \xi_2, \theta_A, \theta_B; \theta) \tag{5.13}$$

. where X_R is a closed-form analytical representation of the trajectory to flyby a planet.

Flybys: Let us define $\boldsymbol{x}_{b,j}(t) = [\mathrm{r}_{b,j}(t), \mathrm{v}_{b,j}(t), \psi_{b,j}(t), \theta_{b,j}(t)]$ as the continuous state vector of a planet b_j, where $r_{b,j}(t) \in \mathbb{R}$, $\mathrm{v}_{b,j}(t) \in \mathbb{R}$, $\mathrm{v}_{b,j}(t) \in \mathbb{R}$ and $\theta_{b,j}(t) \in \mathbb{R}$ represent their intrinsic coordinates respectively, as defined by the spacecraft. Flybys are assumed to produce an instantaneous change in the heliocentric velocity of the spacecraft, given by the transition map $\phi_{fb,j}$ and occurring when the state vector of the spacecraft intersects the discontinuity surface s_{fb}:

$$\text{Flybys}: \begin{cases} \phi_{fb,j}: & [\mathrm{v}(t_i^+), \psi(t_i^+)] = \boldsymbol{\Delta}_j(\mathrm{v}(t_i^-), \psi(t_i^-), \boldsymbol{p}_j, \mathrm{v}_{b,j}(t_i^-), \psi_{b,j}(t_i^-), \mu_{b,j}, t_i^-), \\ & q(t_i^+) = q(t_i^-), \quad \mathrm{r}(t_i^+) = \mathrm{r}(t_i^-), \quad \theta(t_i^+) = \theta(t_i^-). \\ s_{fb,j}: & \mathrm{r} - \mathrm{r}_{b,j}(t) = 0, \quad \text{and} \quad \theta - \theta_{b,j}(t) = 0, \quad j \in \{1, \dots, n_s\} \end{cases} \tag{5.14}$$

where $\mu_{b,j}$ is the gravitational constant of the corresponding planet b_j. As modeled by the discrete event function $s_{fb,j}$, a flyby is only possible if the spacecraft heliocentric distance and polar angle matches the heliocentric distance of polar angle of a planet. Note that, the there are as many discontinuity surfaces as n_{fb} available planets to flyby. Following a planar model for the flyby, if a planet b_j is encountered at t_i, the heliocentric post-flyby velocity $\mathrm{v}(t_i^+)$ and flight-path angle $\psi(t_i^+)$ can be obtained assuming a hyperbolic trajectory around the planet. The flyby trajectory is a function of the pre-flyby velocity $\mathrm{v}(t_i^-)$ and flight-path angle $\psi(t_i^+)$, the planet heliocentric velocity $\mathrm{v}_{b,j}(t_i^-)$ and flight path angle $\psi_{b,j}(t_i^-)$, and additional static control variables or parameters $\boldsymbol{p}_j = [\delta]$, which are subject to optimization. Hence, it holds:

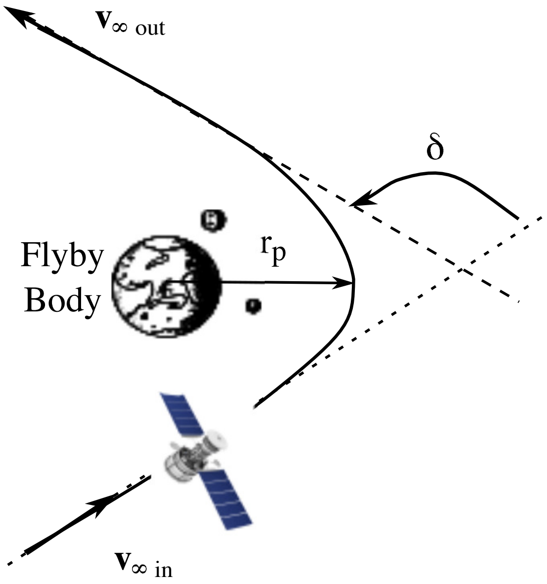


Figure 5.5: MOLTO-IT Step 1: 2D Flyby geometry

$$\boldsymbol{\Delta}_j : \begin{cases} \hat{\mathrm{v}}(t_i^-) = \sqrt{(\mathrm{v}(t_i^-)\cos\psi(t_i^-) - \mathrm{v}_b(t_i^-)\cos\psi_b(t_i^-)^2 + (\mathrm{v}(t_i^-)\sin\psi(t_i^-) - \mathrm{v}_b(t_i^-)\sin\psi_b(t_i^-))^2} \\ \\ \alpha = atan\dfrac{\mathrm{v}(t_i^-)\sin\psi(t_i^-) - \mathrm{v}_b(t_i^-)\sin\psi_b(t_i^-)}{\mathrm{v}(t_i^-)\cos\psi(t_i^-) - \mathrm{v}_b(t_i^-)\cos\psi_b(t_i^-)} + \delta \\ \\ \mathrm{v}(t_i^+) = \sqrt{(\mathrm{v}_b(t_i^-)\sin\psi_b(t_i^-) + \hat{\mathrm{v}}(t_i^-)\sin\alpha)^2 + (\mathrm{v}_b(t_i^-)\cos\psi_b(t_i^-) + \hat{\mathrm{v}}(t_i^-)\cos\alpha)^2} \\ \\ \psi(t_i^+) = atan\dfrac{\mathrm{v}_b(t_i^-)\sin\psi_b(t_i^-) + \hat{\mathrm{v}}(t_i^-)\sin\alpha}{\mathrm{v}_b(t_i^-)\cos\psi_b(t_i^-) + \hat{\mathrm{v}}(t_i^-)\cos\alpha} \end{cases} \tag{5.15}$$

Here, $\hat{\mathrm{v}}(t_i^-)$ is the spacecraft incoming relative velocity to the planet, which rotates an angle δ on the orbital plane of the hyperbola to scape. An illustration of the planar flyby model is provided in Figure 5.5. The angle δ is typically constrained by the minimum allowable flyby radius $r_{p,min}$. They are related by the following equation:

$$\delta_{max/min} = \pm 2\,\mathrm{arcsin}(1/(1 + \mathrm{r}_{p,min}\hat{\mathrm{v}}^2(\mathrm{t}_\mathrm{i}^-)/\mu_{\mathrm{b,j}})) \tag{5.16}$$

Finally, let us define $\sigma_{fb,i}$ as the time-ordered sequence of flyby events:

$$\sigma_{fb} = [(t_1, s_{fb,j})\ldots,(t_i, s_{fb,j}),\ldots,(t_n, s_{fb,j})] \tag{5.17}$$

The previous sequence is not known a-priori and has to be determined as part of the solution, including the number and the planetary bodies.

Launch Event: The launch event is characterized by an instantaneous change in the initial heliocentric velocity and flight path angle, given by the transition function ϕ_{la} that is performed by the launcher at the departure epoch, as defined by the event surface s_{la}. Then, it holds:

$$\text{Launch}: \begin{cases} \phi_{la}: & [\mathrm{v}(t_0^+), \psi(t_0^+)] = \Delta_{la}(\boldsymbol{x}_{b,1}, t_0^-), \quad \mathrm{r}(t_0^+) = \mathrm{r}(t_0^-), \\ & \theta(t_0^+) = \theta(t_0^-),, \quad q(t_0^+) = q(t_0^-). \\ s_{la}: & t_0 - t_0^* = 0. \end{cases} \tag{5.18}$$

The state of the spacecraft after the launch event is given by:

$$\boldsymbol{\Delta}_{la}: \begin{cases} \mathrm{v}(t_i^+) = \sqrt{(\mathrm{v}(t_i^-)\sin\psi(t_i^-) + \mathrm{v}_{\infty,0}\sin\psi_{\infty,0})^2 + (\mathrm{v}(t_i^-)\cos\psi(t_i^-) + \mathrm{v}_{\infty,0}\cos\psi_{\infty,0})^2} \\ \\ \psi(t_i^+) = atan\dfrac{\mathrm{v}(t_i^-)\sin\psi(t_i^-) + \mathrm{v}_{\infty,0}\sin\psi_{\infty,0}}{\mathrm{v}(t_i^-)\cos\psi(t_i^-) + \mathrm{v}_{\infty,0}\cos\psi_{\infty,0}} \end{cases} \tag{5.19}$$

In the above, $\mathrm{v}_{\infty,0}$ and $\psi_{\infty,0}$ are the departure hyperbolic excess velocity and flight path angle respectively. The launch hyperbolic excess velocity magnitude must be constrained to the maximum available by the launcher vehicle, hence it is problem specific.

Constraints

Hereto, the continuous and discrete variables, the continuous control parameter, as well as the modeling of the continuous and discrete dynamics have been presented. In order to fully describe the HOCP, the constraints applicable in our approach have to be defined. Fundamentally, as this stage uses low-fidelity models only boundary conditions and a simple approach for limiting the maximum control effort of the shape-based method are considered.

Boundary conditions: A spacecraft has the state $\boldsymbol{x}_0$ at θ_0 and at t_0. It has to travel in a certain time of flight (ToF) to a target planet that has an angular position θ_P, radial distance r_P, velocity v_P and flight path angle ψ_P at $t_0 + ToF$. The elements of the state vector of the planets are obtained by assuming constant orbital elements for each leg. Therefore, a flyby trajectory to the planet must satisfy two non-linear constraints:

$$\mathrm{r}_F(\boldsymbol{x}_0(\theta_0), \theta_0, \xi_1, \theta_A; \theta_P) - \mathrm{r}_P = 0 \tag{5.20}$$

$$t_F((\boldsymbol{x}_0(\theta_0), \theta_0, \xi_1, \theta_A; \theta_P) - ToF = 0 \tag{5.21}$$

Note that, in the previous expressions there still remain one free parameter, either the control parameter ξ_1 or the switching-off polar angle θ_A that can be subjected to optimization.

On the other hand, if the trajectory is to rendezvous a planet, four constraints, involving position, velocity, and time have to be imposed:

$$\mathrm{r}_R(\boldsymbol{x}_0(\theta_0), \theta_0, \xi_1, \xi_2, \theta_A, \theta_B; \theta_P) - \mathrm{r}_P = 0 \tag{5.22}$$

$$t_R(\boldsymbol{x}_0(\theta_0), \theta_0, \xi_1, \xi_2, \theta_A, \theta_B; \theta_P) - ToF = 0 \tag{5.23}$$

$$\mathrm{v}_R(\boldsymbol{x}_0(\theta_0), \theta_0, \xi_1, \xi_2, \theta_A, \theta_B; \theta_P) - \mathrm{v}_P = 0 \tag{5.24}$$

$$\psi_R(\boldsymbol{x}_0(\theta_0), \theta_0, \xi_1, \xi_2, \theta_A, \theta_B; \theta_P) - \psi_P = 0 \tag{5.25}$$

In this case, there exits no additional free parameter, although no corrective maneuver is needed at the encounter.

Propulsive constraints: The presented shape-based method provides a predefined acceleration for the trajectory, without considering any limits on the thrust or the power available to the spacecraft. Consequently, a trajectory computed by this method may not be realizable. Typically, because the acceleration profile of the spirals exceeds the capabilities of the real engine. However, an estimation can be made. Let us consider an interplanetary leg between two celestial bodies. The shaped-base method connects both bodies with a trajectory given by $\mathrm{r}_s = \mathrm{r}_s(\theta)$, and the acceleration profile $\mathbf{a}_T(t)$. Let us consider a spacecraft with initial mass m_0, and equipped with low-thrust engine, whose maximum thrust produced is modeled as a general function of the distance to the Sun and and time, i.e., $T = T(\mathrm{r}, t)$. We assume that a trajectory is feasible if the following constraints is satisfied:

$$\Delta V_s = \int |\mathbf{a}_T| \frac{\mathrm{r}}{\mathrm{v}\sin\psi\cos\psi} d\theta \leq \Delta V_a = \int \frac{T(\mathrm{r}_s(\theta), t)}{m_0} dt \tag{5.26}$$

The previous conditions states that the trajectory would be realizable if the total ΔV_s provided by the spirals is lower that total ΔV_a that the real engine could provide continuously thrusting along the same trajectory.

5.2.2 MOLTO-IT Step 2

In this case, more accurate dynamical and hardware models are used. The continuous state $\boldsymbol{x} = [\mathbf{r}, \mathbf{v}, m]$ of the spacecraft is determined by its position $\mathbf{r}(t) \in \mathbb{R}^3$ and velocity $\mathbf{v}(t) \in \mathbb{R}^3$ vectors with respect to an inertial reference frame, which, without loss of generality, is centered in the Sun. Besides, the evolution of the spacecraft mass $m(t)$ is needed to fully determine its dynamics. The discrete state variable $q(t) \in \{0, 1\}$ determines the working condition of the electric engine. When thrusting, the engine can only operate at its maximum available thrust level T and its orientation can be controlled by the dynamical continuous input vector $\boldsymbol{u} = [\alpha, \beta]$, where α and β represent the azimuth and declination angles respectively. Additionally, the switch between modes of operation can be managed by the binary control input $v(t) \in \{0, 1\}$. The coasting state is required when '0', while the thrusting state is required with '1' as long as the power system requirements are satisfied.

Continuous Dynamics

The evolution of the continuous state of the spacecraft $\boldsymbol{x} = [\mathbf{r}, \mathbf{v}, m]$ is modeled as a particle moving in a central gravity field under the action of a perturbing acceleration $\mathbf{a}_p$, i.e., as a Perturbed Two-Body-Problem. The equation of motions are described by the following set of differential equations:

$$\mathrm{f}: \begin{cases} \dot{\mathbf{v}} = -\dfrac{\mu_s}{\mathrm{r}^3}\mathbf{r} + \mathbf{a}_p(\boldsymbol{x}, q, \boldsymbol{u}, t) \\ \dot{\mathbf{r}} = \mathbf{v} \\ \dot{m} = \dot{m}(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{u}, t) \end{cases} \tag{5.27}$$

where μ_s represents the gravitational constant of the Sun. The perturbing acceleration $\mathbf{a}_p$ and the mass flow rate $\dot{m}$ have different expressions depending on the discrete state of the spacecraft. Assuming that the thrust produced by the electric engine T is the only external force, it holds:

$$\mathbf{a}_p, \dot{m}: \begin{cases} \mathbf{a}_p = \dfrac{T}{m}\mathbf{d}, \quad \dot{m} = -\dfrac{T}{g_0 I_{sp}} & \text{if} \quad q = \mathit{1} \\ \mathbf{a}_p = 0, \quad \dot{m} = 0 & \text{if} \quad q = \mathit{0} \end{cases} \tag{5.28}$$

Here, I_{sp} is the specific impulse and g_0 is the Earth's gravitational acceleration at sea level. The unitary vector $\mathbf{d}$ points toward the direction of the thrust vector and can be projected onto the orbital frame and expressed as a function of the control inputs as follows:

$$\mathbf{d} = \cos\beta \sin\alpha\, \mathbf{i} + \cos\beta \cos\alpha\, \mathbf{j} + \sin\beta\, \mathbf{k} \tag{5.29}$$

where the unitary vectors are defined as:

$$\mathbf{i} = \frac{\mathbf{r}}{\mathrm{r}}, \qquad \mathbf{k} = \frac{\mathbf{r} \times \mathbf{v}}{\mathrm{rv}}, \qquad \mathbf{j} = \mathbf{k} \times \mathbf{i} \tag{5.30}$$

In the above, $\mathrm{r} = ||\mathbf{r}||$ is the radial distance and $\mathrm{v} = ||\mathbf{v}||$ is the magnitude of the velocity. An illustration of the geometry of the problem is depicted in Figure 5.6.

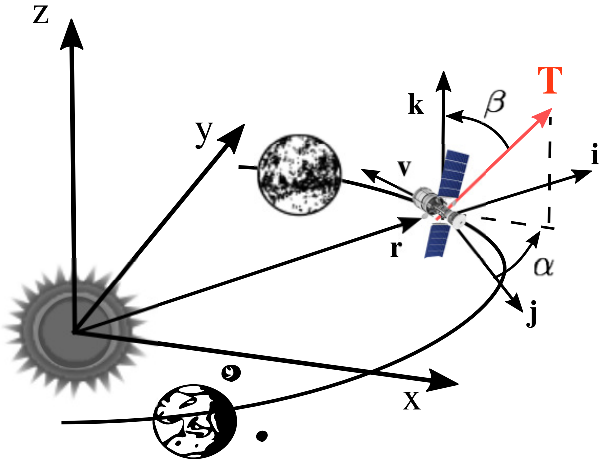


FIGURE 5.6: MOLTO-IT Step 2: Geometry of the problem

The thrust T of the specific impulse I_{sp} can be modeled as constants or as polynomials functions of the power available P_a to the spacecraft of the form:

$$T = c_{T0} + c_{T1}P_a + c_{T2}P_a^2 + c_{T3}P_a^3 + c_{T4}P_a^4 \tag{5.31}$$

$$\dot{m} = c_{m0} + c_{m1}P_a + c_{m2}P_a^2 + c_{m3}P_a^3 + c_{m4}P_a^4 \tag{5.32}$$

where the coefficients $c_{T,i}$ and $c_{m,i}$ are obtained from curve fits from laboratory tests data. Each thruster has an associated minimum power P_{min} and maximum power P_{max}. If $P_a < P_{min}$, the thruster cannot operate. If $P_a > P_{max}$, the performance polynomials are evaluated at P_{max}. In this tool, P_a is given as a function of the power generated by the solar panels P_g :

$$P_g = \frac{P_0}{\mathrm{r}^2}\left(\frac{\gamma_0 + \gamma_1/\mathrm{r} + \gamma_2/\mathrm{r}^2}{1 + \gamma_3\mathrm{r}}\right) \tag{5.33}$$

where γ_i are user-defined parameters for the solar panel and P_0 is the nominal power, which in turn is a function of the time since launch:

$$P_0 = P_{0-BOL}(1-\tau)^t \tag{5.34}$$

In the above, P_{0-BOL} is the base power delivered on the day of launch, τ is the decay rate of the solar arrays measured as a percentage per year, and t is the time since launch in years. Equation (5.34) may also be used to model the decay of a radioisotope thermal generator. Finally, he available power P_a is then obtained as the difference between the power generated by the spacecraft P_g and the power required to operate the spacecraft bus $P_{s/c}$, such that:

$$P_a = (1 - \delta_{power})(P_g - P_{s/c}) \tag{5.35}$$

where δ_{power} is a user-defined power margin.

Discrete Dynamics

As for MOLTO-IT Step 1, the discrete dynamics of the spacecraft allows to characterize the effect of both performing flyby maneuvers and turning the engine on and off. Hereafter, we define the set of discrete event functions, including discontinuity surfaces and transition map functions, necessary to include such effects in the model of MOLTO-IT Step 2.

Engine on-off switching: The switch between the thrust/coast modes of operation can be described by a controlled discrete event or by an autonomous event. The former occurs as a consequence of a controlled decision, for propellant savings reasons, whereas the latter happens as a consequence of the power subsystem requirements (when there is not enough power available for the engine to operate, e.g., $P_a < P_g$). Both are summarized in the following functions:

$$\text{Switching-on}: \begin{cases} \phi_{on}: & q(t_i^+) = 1, \quad \mathbf{v}(t_i^+) = \mathbf{v}(t_i^-), \\ & \mathbf{r}(t_i^+) = \mathbf{r}(t_i^-), \quad m(t_i^+) = m(t_i^-) \\ s_{on}: & q(t_i^-) = 0, \quad v(t_i^-) = 1, \quad 0 \leq P_a(\boldsymbol{x}, t_i^-) - P_g(\boldsymbol{x}, t_i^-) \end{cases} \tag{5.36}$$

$$\text{Switching-off}: \begin{cases} \phi_{off}: & q(t_i^+) = 0, \quad \mathbf{v}(t_i^+) = \mathbf{v}(t_i^-), \\ & \mathbf{r}(t_i^+) = \mathbf{r}(t_i^-), \quad m(t_i^+) = m(t_i^-) \\ s_{off,1}: & q(t_i^-) = 1, \quad v(t_i^-) = 0 \\ s_{off,2}: & q(t_i^-) = 1, \quad 0 \geq P_a(\boldsymbol{x}, t_i^-) - P_g(\boldsymbol{x}, t_i^-) \end{cases} \tag{5.37}$$

Here, the event surface s_{on} refers to the controlled switching-on whereas $s_{off,1}$ and $s_{off,2}$ represents the event surface for the controlled and autonomous switching-off respectively. Let us defined σ_{sw} as the time-ordered sequence of the on-off switchings of the engine:

$$\sigma_{sw} = [\ldots, (t_i, s_{on/off}), \ldots] \tag{5.38}$$

which is not known a-priori and have to be determined as part of the solution.

Flybys: Let the continuous state vector of a planet b_j be defined as $\boldsymbol{x}_{b,j}(t) = [\mathbf{r}_{b,j}, \mathbf{v}_{b,j}]$, where $\mathbf{r}_{b,j}(t) \in \mathbb{R}^3$ and $\mathbf{v}_{b,j}(t) \in \mathbb{R}^3$ represent its position and velocity heliocentric vectors respectively. Flybys are assumed to produce an instantaneous change in the heliocentric velocity vector of the spacecraft, given by the transition map $\phi_{fb,j}$ and occurring when the position vector of the spacecraft intersects the discontinuity surface s_{fb}:

$$\text{Flybys}: \begin{cases} \phi_{fb,j}: & \mathbf{v}(t_i^+) = \boldsymbol{\Delta}_j(\mathbf{v}(t_i^-), \boldsymbol{p}_j, \mathbf{v}_{b,j}(t_i^-), \mu_{b,j}, t_i^-), \\ & q(t_i^+) = q(t_i^-), \quad \mathbf{r}(t_i^+) = \mathbf{r}(t_i^-), \quad m(t_i^+) = m(t_i^-) \\ s_{fb,j}: & ||\mathbf{r} - \mathbf{r}_{b,j}(t)|| = 0, \quad j \in \{1, \ldots, n_s\} \end{cases} \tag{5.39}$$

In the above, $\mu_{b,j}$ is the gravitational constant of the corresponding planet b_j.

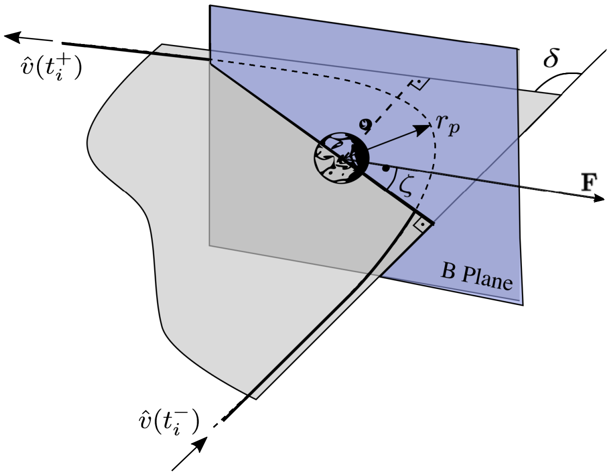


FIGURE 5.7: MOLTO-IT Step 2: 3D Flyby geometry

The spacecraft's mass is are assumed not to change because powered flybys are not considered. Hence, if a planet b_j is encountered at t_i, the heliocentric post-flyby velocity $\mathbf{v}(t_i^+)$ is obtained assuming a hyperbolic trajectory around the planet, which is a function of the pre-flyby velocity $\mathbf{v}(t_i^-)$, the planet heliocentric velocity $\mathbf{v}_{b,j}(t_i^-)$ and additional parameters $\boldsymbol{p}_j = [r_{p,j}, \zeta_j]$, that includes the minimum distance of approach $r_{p,j}$ and the B-Plane angle ζ. Hence, it holds:

$$\mathbf{\Delta}_j : \begin{cases} \hat{\mathbf{v}}(t_i^-) = \mathbf{v}(t_i^-) - \mathbf{v}_{b,j}(t_i^-) \\ \delta = 2\arcsin(1/(1 + r_{p,j}\hat{\mathrm{v}}^2(\mathrm{t_i^-})/\mu_{\mathrm{b,j}})) \\ \hat{\mathbf{v}}(t_i^+) = \cos\delta\hat{\mathbf{i}} + \cos\zeta\sin\delta\hat{\mathbf{j}} + \sin\zeta\sin\delta\hat{\mathbf{k}} \\ \mathbf{v}(t_i^+) = \mathbf{v}_{b,j}(t_i) + \hat{\mathbf{v}}(t_i^+) \end{cases} \tag{5.40}$$

where the unit vectors are

$$\hat{\mathbf{i}} = \frac{\hat{\mathbf{v}}(t_i^-)}{\hat{\mathrm{v}}(\mathrm{t_i^-})}, \quad \hat{\mathbf{j}} = \frac{\hat{\mathbf{i}} \times \mathbf{v}_{b,j}(t_i^-)}{||\hat{\mathbf{i}} \times \mathbf{v}_{b,j}(t_i^-)||}, \quad \hat{\mathbf{k}} = \hat{\mathbf{i}} \times \hat{\mathbf{j}} \tag{5.41}$$

Here $\hat{\mathbf{v}}(t_i^-)$ is the spacecraft incoming relative velocity to the planet, which rotates an angle δ on the orbital plane of the hyperbola to scape as $\hat{\mathbf{v}}(t_i^+)$. The orbital plane is determined by the angle ζ measured on the B-plane with respect to the vector $\mathbf{F}$, which is contained on the B-plane and is parallel to ecliptic plane (see Fig. 5.7). The B-plane is defined as the plane passing through the center of the planet and normal to the arrival asymptote $\hat{\mathbf{v}}(t_i^-)$. In this case, the time-ordered sequence of flyby events, $\sigma_{fb,i}$, is known-priori:

Launch event: The launch event is characterized by an instantaneous change in the initial heliocentric velocity vector, given by the transition function ϕ_{la} that is performed by the launcher at the departure epoch, as defined by the event surface s_{la}. Then, it holds:

$$\text{Launch} : \begin{cases} \phi_{la} : & \mathbf{v}(t_0^+) = \Delta_{la}(\boldsymbol{x}_{b,1}, t_0^-), \quad \mathbf{r}(t_0^+) = \mathbf{r}(t_0^-), \\ & m(t_0^+) = m(t_0^-), \quad q(t_0^+) = q(t_0^-). \\ s_{la} : & t_0 - t_0^* = 0. \end{cases} \tag{5.42}$$

The velocity of the spacecraft after launch is given by:

$$\Delta_{la} = \mathbf{v}_{b,1}(t_0^-) + \mathbf{v}_{\infty,0} \begin{bmatrix} \cos RLA \cos DLA \\ \sin RLA \cos DLA \\ \sin DLA \end{bmatrix} \tag{5.43}$$

where $v_{\infty,0}$ is the departure hyperbolic excess velocity and RLA and DLA are two angles that describe the right ascension and declination of the launch asymptote respectively. Typically, limits on the DLA are imposed, due to limitations of the launch site. Moreover, the modulus of the impulse $\boldsymbol{v}_{\infty,0}$ is typically defined as polynomial functions of the delivered mass of the spacecraft and $C3 = v_{\infty,0}^2$ as:

$$m_0 = a_0 + a_1 C3 + a_2 C3^2 + a_3 C3^3 + a_4 C3^4 + a_5 C3^5 \tag{5.44}$$

where the coefficients a_i are chosen by a curve fit to launcher performance data available.

Constraints

Similar to MOLTO-IT Step 1, boundary conditions for satisfying the flyby conditions, i.e., $\mathbf{r}(t_f) = \mathbf{r}_{b,n}(t_f)$ or to meet the rendezvous constraints, i.e., $\mathbf{v}(t_f) = \mathbf{v}_{b,n}(t_f)$. An additional constraint to account for fixed reorientations is included.

Fixed re-orientation times constraint: Let us consider a thrusting arc that occurs within the time interval $[t_0, t_f]$. During this arc, an unknown number of reorientations is desired to happen subject to a minimum period of time t_{min}. Let us break up the time internal into N subintervals $[t_i, t_{i+1}]$. For each of them, a fixed inertial orientation is imposed such that $\alpha_i = cte$ and $\beta_i = cte$. However, these sub-arcs may not fulfill with the reorientation time constraint. Thus, some of these intervals may have to be enlarged, while others may have to be deleted because of being unnecessary. Therefore, a new decision parameter, $\delta \in [-1, 1]$, is introduced do each subinterval to model this decision process. Note that this parameter is continuous, and therefore useful for being used in continuos optimization models. However, it is not suitable for modeling decision processes. Thus, the following relation is defined:

$$\sigma_i = \frac{2}{\pi} atan(\epsilon \delta_i) + 1 \tag{5.45}$$

where ϵ is tuning parameter that has to be set by the user. Here, σ is a continuous approximation of a binary function that can only take the discrete values 0 or 1. Let us define the following set of re-orientation times non-linear equations:

$$(1 - \sigma_i)^2 (t_{i+1} - t_i - t_{min}) > 0, \quad \text{for} \quad i = 1, \dots, N \tag{5.46}$$

$$\sigma_i^2 (t_i - t_{i+1}) = 0, \quad \text{for} \quad i = 1, \dots, N \tag{5.47}$$

Note that, when $\sigma = 1$, the interval is forced to collapse, whereas when $\sigma = 0$, the minimum reorientation time has to be meet. An illustration of the process is depicted in Fig, 5.8.

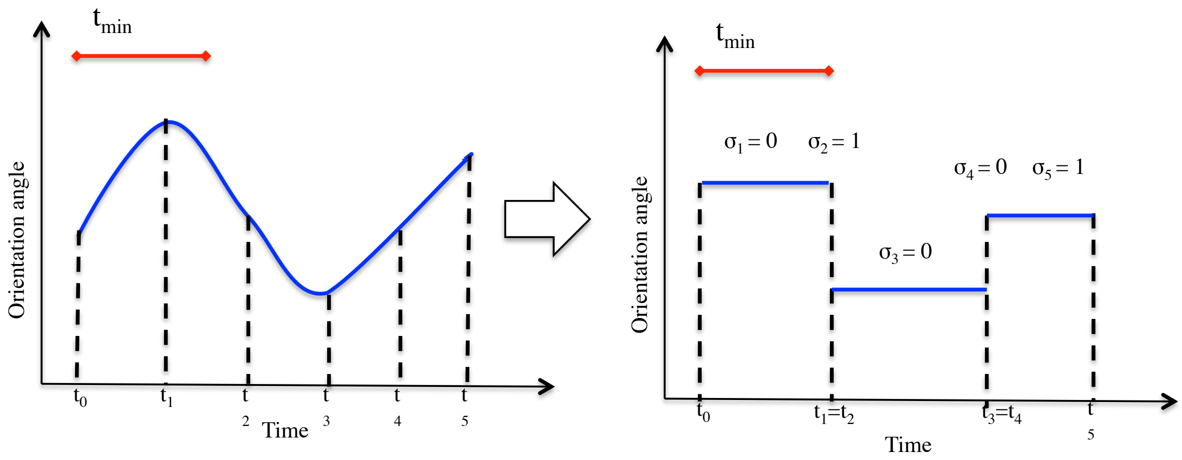


FIGURE 5.8: MOLTO-IT Step 2: fixed reorientation time constraint model

5.3 Solution Approach

In this section, MOLTO-IT (Multi-Objective Low-Thrust Trajectory Optimizer for Interplanetary Transfers) is presented. It consists on a two-step solution approach to tackle the problem under consideration. The algorithm is schematically depicted in Figure 5.9 and an overview of its main elements is presented in Table 5.2. In the first step, namely MOLTO-IT Step 1, the shaped-based parametric model of the trajectory is incorporated. It pursues the goal of being a flexible and robust algorithm able to rapidly find solutions for LT-MGA trajectories defined by a user input. The sequence of gravity assists is automatically determined by the tool. The outputted results would approximate the optimal performances as well as the mission design variables. They also served as an initial guess for the second step, called MOLTO-IT Step. It includes more accurate dynamical and propulsion models to improve fidelity. In addition, operational constraints can be included with ease. Hereafter, both steps are described in detail.

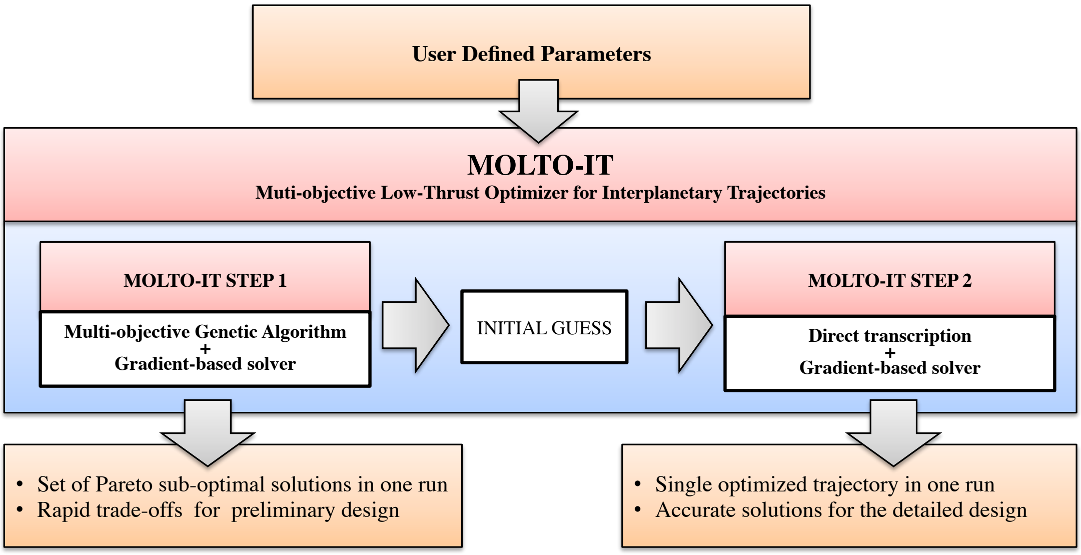

User Defined Parameters
MOLTO-IT
Muti-objective Low-Thrust Optimizer for Interplanetary Trajectories
MOLTO-IT STEP 1
Multi-objective Genetic Algorithm
+
Gradient-based solver
INITIAL GUESS
MOLTO-IT STEP 2
Direct transcription
+
Gradient-based solver
• Set of Pareto sub-optimal solutions in one run
• Rapid trade-offs for preliminary design
• Single optimized trajectory in one run
• Accurate solutions for the detailed design

FIGURE 5.9: MOLTO-IT: Algorithm Scheme

TABLE 5.2: MOLTO-IT Step 1 VS MOLTO-IT Step 2 main features

	MOLTO-IT Step 1	MOLTO-IT Step 2
Problem type	Multi-objective MINLP	Multiphase OCP
Solution Approach	Genetic Algorithm + Gradient-Based solver	Hermite-Simpson + Gradient-based solver
Dynamical Model	Planar + patched conic	3D + patched conic
Control Model	Generalized Logarithmic Spirals	Optimal control
Flyby Model	Planar + Instantaneous	3D + Instantaneous
Flyby sequence	Free	Fixed
Thrust Model	Unconstrained	Polynomial approximation
I_{sp} Model	Constant	Polynomial approximation
Ephemerides	Constant orbital elements	NAIF-SPICE toolkit
Programming Language	MATLAB	MATLAB
Heuristic solver	NSGA-II	N/A
Gradient-based solver	fmincon	Ipopt

5.3.1 MOLTO-IT Step1

MOLTO-IT Step 1 makes use of the Controlled Generalized Logarithmic Spirals and of the predefined thrusting sequence described in Section 5.2.1. Therefore, the infinite dimensional multi-objective Hybrid Optimal Control Problem (MO-HOCP) is reduced to a multi-objective Mixed-Integer Nonlinear Programming problem (MO-MINLP). The real and integer nature of the variables and the requisite of evaluating many different scenarios simultaneously, make population-based heuristic algorithms, such as genetic algorithms or particle swarm optimization, the most adequate techniques to solve it. However, they are not well suited for handling the nonlinear constraints arising from both the flyby and rendezvous conditions. This kind of constraints can be tackled much more efficiently with off-the-self gradient-based solvers. Consequently, MOLTO-IT Step 1 consists on two nested optimization loops.

A genetic algorithm and a gradient-based solver optimize the outer loop and the inner loop respectively. The genetic algorithm is based on the well-known NSGA-II (Non-dominated Sorting Genetic Algorithm) and has been implemented in MATLAB, while the inner loop solver uses the sequential quadratic solver implemented in fmincon, a built-in function of MATLAB. The process begins when the user provides the information summarized in the Table 5.3. It mainly includes information about the departure and arrival planets, the mission type (i.e., flyby or rendezvous), along with the list of available planets to flyby, the minimum and maximum number of allowed flybys, the launch window opening and closing dates, maximum and minimum flight time limits, the modulus of the launch hyperbolic excess velocity at departure, and optionally, the launcher performance model. The algorithm schematically depicted in Figure 5.10.

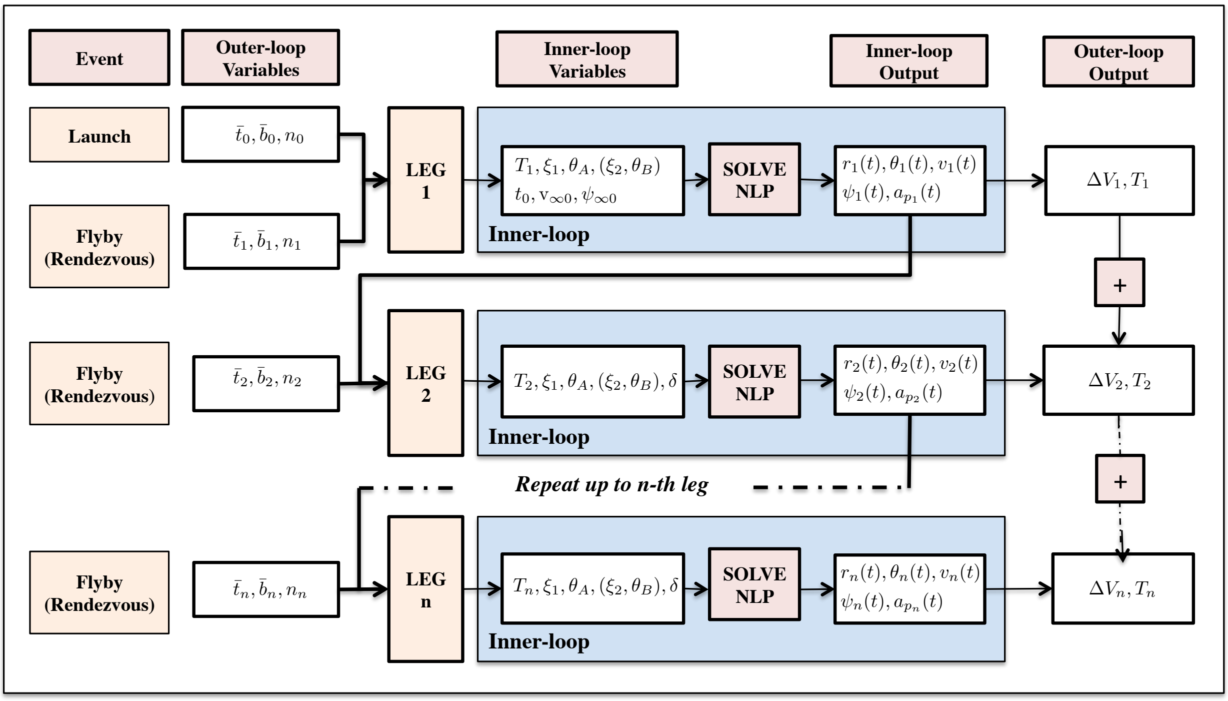

Event
Outer-loop Variables
Inner-loop Variables
Inner-loop Output
Outer-loop Output
Launch
$\bar{t}_0, \bar{b}_0, n_0$
Flyby (Rendezvous)
$\bar{t}_1, \bar{b}_1, n_1$
LEG 1
$T_1, \xi_1, \theta_A, (\xi_2, \theta_B)$
$t_0, \mathrm{v}_{\infty 0}, \psi_{\infty 0}$
SOLVE NLP
$r_1(t), \theta_1(t), v_1(t)$
$\psi_1(t), a_{p_1}(t)$
$\Delta V_1, T_1$
Inner-loop
+
Flyby (Rendezvous)
$\bar{t}_2, \bar{b}_2, n_2$
LEG 2
$T_2, \xi_1, \theta_A, (\xi_2, \theta_B), \delta$
SOLVE NLP
$r_2(t), \theta_2(t), v_2(t)$
$\psi_2(t), a_{p_2}(t)$
$\Delta V_2, T_2$
Inner-loop
Repeat up to n-th leg
+
Flyby (Rendezvous)
$\bar{t}_n, \bar{b}_n, n_n$
LEG n
$T_n, \xi_1, \theta_A, (\xi_2, \theta_B), \delta$
SOLVE NLP
$r_n(t), \theta_n(t), v_n(t)$
$\psi_n(t), a_{p_n}(t)$
$\Delta V_n, T_n$
Inner-loop

FIGURE 5.10: MOLTO-IT Step 1: Algorithm scheme

Table 5.3: MOLTO-IT Step 1: User Defined Parameters

Description	Variable
Departure/Arrival Body	b_o, b_f
Mission type	Flyby/Rendezvous
Launch Window opening/closing Date	$t_{0,min}$, $t_{0,max}$
Minimum/Maximum Flyby Number	$n_{fb,min}$, $n_{fb,max}$
Flyby bodies	-
Minimum/Maximum Revolution Number	n_{min},n_{max}
Minimum Flyby radius	$r_{fb,min}$
Minimum/Maximum Time of Flight	T_{min},T_{max}
Minimum/Maximum Launch Hyperbolic velocity	$\mathrm{v}_{\infty 0,\mathrm{min}}$, $\mathrm{v}_{\infty 0,\mathrm{max}}$

The outer loop

The outer loop algorithm has two main assignments: on the one hand, it is in charge of optimizing the discrete design variables associated to the flyby sequence with respect to a multi-objective fitness function; on the other hand, it runs as an automatic initial guess generator for the gradient-based solver in the inner loop, providing starting values for the launch date and the transfer times for each interplanetary leg. The solution of the inner-loop, if exists, will be in the vicinity of the initial guess provided by the outer-loop. The decision vector defining the population of the GA, consists on the variables summarized in Table 5.4. The departure epoch and the transfer time per leg are real decision variables, whereas the number of revolutions and the flyby bodies are represented by integer variables. The user has to provide a list of available planets to flyby, e.g., (Jupiter, Mars, Earth). Then, each planet is identified by a positive integer number, e.g., Jupiter (1), Mars (2), and Earth (3). Therefore, a sequence of planetary flybys can be defined by a sequence of integer variables, $b_1, b_2, \ldots, b_{n_{fb,max}}$, where b_i is defined as:

$$b_i \in \{1, \ldots, n_b\} \subset \mathbb{Z}, \quad \text{for} \quad i = 1, \ldots, n_{fb,min} \tag{5.48}$$

$$b_i \in \{0, \ldots, n_b\} \subset \mathbb{Z}, \quad \text{for} \quad i = n_{fb,min}, \ldots, n_{fb,max} \tag{5.49}$$

where n_b is the number of available planets. Note that in the above, the number 0 represents a “null-flyby”, which means that no flyby is desired. Thus, the total number of flybys for each sequence b is determined by the number of non-zero entries, and it may be different for each member of the population. This technique is called the ’null-gene’ method as introduced in [224] and [225]. When the algorithm parses a decision vector, it skips over the null values and construct a trajectory only from the values that represent planets. Then, each population in the outer loop defines a different sequence of inner loop problems that are solved by the gradient-based solver. Finally, if the sequence of NLP problems are successfully solved, the resulting cost for each interplanetary leg, typically in terms of time of flight and propellant mass, is recovered by the outer loop and provided as the fitness value for the genetic algorithm:

$$J = [\sum_{i=1}^{n+1} \Delta V_i, \sum_{i=1}^{n+1} T_i] \tag{5.50}$$

where n is the number of flybys. In case the NLP solver for the inner-loop does not converge, a flag is sent to the outer loop, which assigns a penalty to the objective function depending on the interplanetary leg that failed. Assuming that the i^{th} leg did not converged, the cost function takes the following form:

$$J = [(n+2-i)\rho, (n+2-i)\rho] \tag{5.51}$$

where ρ is a sufficiently large value in comparisons with the value of fitness function for the feasible trajectories. Thus, trajectories that fails at the fist leg re more likely to be pruned than those that fail at the last leg. Another type of penalization is included when the propulsive constraint of Equation **??** is applied. In this case it is included as an equality constraint for the genetic algorithm:

$$c = \sum_{i=1}^{n+1} (\Delta V_i - \Delta V_{a,i}), \quad \text{if} \quad \Delta V_{a,i} \leq \Delta V_i \tag{5.52}$$

Therefore, only trajectories that comply with the constraint for all legs will remain in the Pareto front. Additionally, the number n of flybys can be included as a third objective. This pursues the goal of preserving diversity during the generations and to force the algorithm to keep exploring all possible number of flybys.

The inner loop

For each interplanetary leg defined by the outer-loop and inner-loop problem is defined. The number of variables for the NLP and the number of constraints depends on the type of transfer leg under consideration. The inner-loop variables and its bounds are summarized in Table 5.5 as a function of the type of event at each end point: Launch (L), Flyby (F) or Rendezvous (R). The launch epoch and the transfer time per leg are subjected to lower and upper bounds centered in the value provided by the outer loop. The widths of such intervals are $2T_{lim}$ for the transfer time and $2t_{lim}$ for the launch date. They have a twofold purpose: they improve the convergence of the solver and they maintain the diversity in the solutions, helping to produce a well-spread Pareto front. The larger the widths, the less spread the Pareto front, although each population has more chances to converge to a feasible solution. A good value for $2T_{lim}$ can be determined as half of the synodic period of the departure and arrival planet for this leg.

The angular variables θ_A and θ_B are defined as a fraction of the total travelled angle for each leg, in order to avoid variable limits. The initial guess for $\psi_{\infty 0}$ is selected depending on the semi-major axis of the next body., i.e. $\pi/2$ for outer planets, $-\pi/2$ for inner planets and 0 for resonant flybys. Regarding the computation of the state of the different bodies at different epochs, constant and planar orbital elements are assumed for each leg, given that accurate ephemerides are provided in tabular data which may not be differentiable and lead to non-convergence. The osculating orbital elements of each body are computed at the starting epoch using JPL NAIF-SPICE ephemerides. As the trajectories considered in this model are planar, the position and velocity of the target bodies are projected onto the ecliptic plane. The general form of the fitness and constraint function for the NLP solver is described in Algorithm 1.

Table 5.4: MOLTO-IT Step 1: Outer-loop Variables

Variable	Meaning	Lower Bound	Upper Bound
$\tilde{t}_0$	Departure Epoch Guess	t_{0min}	t_{0max}
$\tilde{T}_i$	Transfer Time per leg Guess	T_{imin}	T_{imax}
b_i	Flyby Body	-	-
n_i	Leg number of revolutions	n_{min}	n_{max}

Table 5.5: MOLTO-IT Step 1: Inner-loop Variables

Variable	Meaning	Lower Bound	Upper Bound	Initial Guess	Event1	Event2
t_0	Departure Epoch	$\bar{t}_0 - t_{lim}$	$\bar{t}_0 + t_{lim}$	$\bar{t}_0$	L	R/F
$\mathrm{v}_{\infty 0}$	Initial Excess Velocity Module	$\mathrm{v}_{\infty 0,\mathrm{min}}$	$\mathrm{v}_{\infty 0,\mathrm{max}}$	$\mathrm{v}_{\infty 0,\mathrm{max}}$	L	R/F
$\psi_{\infty 0}$	Initial Excess Velocity Angle	$-\pi$	π	$\frac{\pi}{2}, -\frac{\pi}{2}, 0$	L	R/F
δ	Flyby Deflection Angle	-1	1	0	F	R/F
T	Leg Transfer Time	$\bar{T}_i - T_{lim}$	$\bar{T}_i + T_{lim}$	$\bar{T}_i$	L/R/F	R/F
ξ_1	Spiral Control Parameter	0	1	0.4	L/R/F	R/F
θ_A	Thrust to Coast switching angle	0	1	0.01	L/R/F	R/F
ξ_2	Spiral Control Parameter	0	1	0.4	L/R/F	R
θ_B	Coast to Thrust switching angle	0	1	0.99	L/R/F	R

Table 5.6: MOLTO-IT Step 2: Variables

Variable	Meaning	Lower Bound	Upper Bound
t_0	Initial Date	0	2π
RLA	Right ascension of the launch asymtote	-180 deg	180 deg
DLA	Right ascension of the launch asymtote	-90 deg	90 deg
v_∞	Launch excess velocity	$v_{\infty 0,min}$	$v_{\infty 0,max}$
$r_{x,j}, r_{y,j}, r_{z,j}$	Spacecraft position at the j^{th} node	-	-
$v_{x,j}, v_{y,j}, v_{z,j}$	Spacecraft velocity at the j^{th} node	-	-
m_j	Spacecraft Mass at the j^{th} node	0	m_0
α_j	In-Plane angle at the j^{th} node	-180 deg	180 deg
β_j	Out-of-Plane angle at the j^{th} node	-90 deg	90 deg
v_j	Throttle parameter at the j^{th} node	0	1
t_i	Date of the i^{th} flyby	0	ToF
$r_{fb,i}$	Flyby radius of the i^{th} flyby	$r_{fb,min}$	$r_{fb,max}$
ζ_i	B-angle of the i^{th} flyby	-180 deg	180 deg

Algorithm 1 Step 1 inner-loop Fitness and Constraint function for leg i^{th}

```
Input : $(t_0, v_{\infty 0}, \psi_{\infty 0}), \{\delta\}, T_i, \xi_1, \theta_A(\xi_2, \theta_B)$
Ouput : $\Delta V$, c
                                                        ▷ Obtain Initial Transfer State
if $Event\text{-}1$ = Launch then

    Compute state of planet $b_1$: $\boldsymbol{x}_{b,1}(t_0)$
    Add launcher Impulse: $\boldsymbol{x}(\theta_0) = \Delta_{la}(\boldsymbol{x}_{b,1}, \mathrm{v}_{\infty 0}, \psi_{\infty 0})$ [Eq.(5.43)]

else

    Obtain initial state $\boldsymbol{x}(\theta_0)$ from leg $(i-1)^{th}$
    Compute state of planet $b_1$: $\boldsymbol{x}_{b,1}(t_0)$
    Compute state after Flyby: $\boldsymbol{x}(\theta_0) = \Delta_{fb}(\boldsymbol{x}(\theta_0), \boldsymbol{x}_{b,1}, \delta)$ [Eq.(5.40)]

end if

Compute final time: $t_F = t_0 + T_i$
Compute state of planet $b_2$: $\boldsymbol{x}_{b,2}(t_F)$
                                                        ▷ Obtain Final Transfer State
if $Event\text{-}2$ = Flyby then

    Compute thrust arc from $\theta_0$ to $\theta_A$: $\boldsymbol{x}(\theta) = \boldsymbol{X}_T(\boldsymbol{x}(\theta_0), \xi_1; \theta)$
    Compute $\Delta V_1$ for the spiral arc [Eq.(5.9)]
    Compute coast arc from $\theta_A$ to $\theta_F$: $\boldsymbol{x}(\theta) = \boldsymbol{X}_C(\boldsymbol{x}(\theta_A); \theta)$

else if $Event\text{-}2$ = Rendezvous then

    Compute thrust arc from $\theta_0$ to $\theta_A$: $\boldsymbol{x}(\theta) = \boldsymbol{X}_T((\boldsymbol{x}(\theta_0), \xi_1; \theta)$
    Compute $\Delta V_1$ for the spiral arc [Eq.(5.9)]
    Compute coast arc from $\theta_A$ to $\theta_B$: $\boldsymbol{x}(\theta) = \boldsymbol{X}_C((\boldsymbol{x}(\theta_A); \theta)$
    Compute thrust arc from $\theta_B$ to $\theta_F$: $\boldsymbol{x}(\theta) = \boldsymbol{X}_T((\boldsymbol{x}(\theta_B), \xi_2; \theta)$
    Compute $\Delta V_2$ for spiral arcs [Eq.(5.9)]

end if
                                                        ▷ Compute Constraints
if $Event\text{-}2$ = Flyby then

    $c_1 : r_{b,2}(t_F) - r(\theta_F) = 0$
    $c_2 : T_i - t(\theta_F) = 0$

else if $Event\text{-}2$ = Rendezvous then

    $c_1 : r_{b,2}(t_F) - r(\theta_F) = 0$
    $c_2 : T_i - t(\theta_F) = 0$
    $c_3 : \mathrm{v_{b,2}(t_F) - v(\theta_F) = 0}$
    $c_4 : \psi_{b,2}(t_F) - \psi(\theta_F) = 0$

end if
                                                        ▷ Compute Cost
Compute summation of all $\Delta V_i$ from spiral arc
```

5.3.2 MOLTO-IT Step 2

MOLTO-IT Step 2 pursues the goal of re-optimizing selected trajectories from MOLTO-IT Step 1 incorporating more accurate models. In fact, it should be noted that the global search algorithm is biased towards the solutions that the shape-based parameterization represents and they may not be realizable with the real propulsion subsystem. In this step, the multi-objective HOCP is reduced to a single-objective Large-scale NLP. For such purpose, it is assumed that the sequence of gravity assisted maneuvers is fixed and provided by MOLTO-IT Step 1. The thrust/coast sequence is not imposed, yet it is optimized by the algorithm using the initial guess of the previous step. As shown in Eq.(5.37) the sequence is determined by the binary control $\boldsymbol{v}$. However, gradient-based solvers only accept continuous variables. Therefore, the binary control $\boldsymbol{v}$ is relaxed so that it can continuously vary within the [0,1] interval. This is not problematic, as the optimal solution is known to be bang-bang for fixed-time minimum fuel problems.

The state vector $\mathbf{r}_j = \mathbf{r}(t_j)$, $\mathbf{v}_j = \mathbf{v}(t_j)$, $m_j = m(t_j)$ and the control inputs $\boldsymbol{u}_j = \boldsymbol{u}(t_j)$ and $\boldsymbol{v}_j = \boldsymbol{v}(L_j)$ are discretized on a selected grid t_j, for $j = 1, \ldots, n$, where n is number of nodes on the grid. Additional parameters related to the launch and flyby events are included. All variables are listed in Table 5.6. The dynamical equations are imposed as defect constraints based on the Hermite-Simpson collocation scheme [126] described in Section 3.4. The dynamical, power and propulsive models described in Section 5.2.2 are included. In this step, only a single-objective function is allowed per run. Typically, either the time of flight, propellant consumed, dry mass or a weighted combination of them are considered. Further constraints can be added to the problem, such as limiting the time that the spacecraft needs for reorienting as given by Eq.(5.46). MOLTO-IT is fully coded in MATLAB and make use of the interior point solver IPOPT [109]. Derivatives are provided by finite differences and the state of the planets are directly obtained by the JPL NAIF-SPICE ephemerides.

5.4 Results

In this section, the capabilities and features of MOLTO-IT to assist interplanetary mission designers in terms of flexibility, accuracy, optimality and computational cost will be evaluated. As representative missions, we present a rendezvous mission from Earth (E) to the asteroid Ceres (C). This is a compelling example due to the challenge of matching the state of Ceres (i.e., position, velocity, and time) with the state of the spacecraft when compared to the less restricted flyby case (i.e., position and time). Thereafter a flyby mission from Earth to Jupiter (J) will be solved. This is a challenging scenario due to the large amount of flybys available. Finally, an hypothetical rendezvous mission from Earth to Pluto (P) is considered. In this case, the initial mass of the spacecraft becomes an optimization variable, which is related to the launch hyperbolic excess velocity via the launcher's performance model. All examples were run in a Intel Core i7 (2,5GHz) computing system. The GA implemented in MOLTO-IT Step 1 uses the MATLAB parallel toolbox and four cores.

Missions to Jupiter and Ceres have been previously addressed in the literature. For instance, Petropoulos and Longuski [47] presented solutions with a two-step approach. Firstly, they apply the tool STOUR-LGTA, which models the trajectory as an exponential sinusoid, to conduct a broad search of the design space. Then, they provide the best trajectories as initial guesses for the Sims-Flanagan based [226] program GALLOP [131]. Thus, we compare MOLTO-IT Step 1 with STOUR-LGTA, and MOLTO-IT Step 2 with GALLOP. Even though Petropoulos and Longuski [47] optimize over a predefined sequence of gravity assists, we allow the algorithm to automatically search for the optimal sequence. To facilitate proper comparisons, we select the same propulsion model in our MOLTO-IT Step 2 than Petropoulos and Longuski [47] for GALLOP. The parameters used are described in Table 5.7. Additionally, the reference power P_0 is 10kW and the thruster is assumed to need at least 0.649kW to operate, while not being able to use power in excess of 2.6kW. These were introduced by Williams and Coverstone-Carroll [227] for recreating the NSTAR engine used on Deep Space 1. A constant I_{sp} value of 3000s is chosen for MOLTO-IT Step 1, similarly to Petropoulos and Longuski [47] for STOUR-LGTA.

The mission to Pluto is inspired by the recent success of New Horizons. Unlike New Horizons, which was a fast flyby, the notional mission presented here would rendezvous with the Pluto system, enabling in depth science. The same scenario has been solved by Englander and Conway [192] using EMTG. This software is based on a two-nested loops approach. The outer-loop is based on the“null-gene” transcription presented by Englander, Conway, and Williams [228] and a discrete GA. The inner-loop is based on the Sims-Flanagan transcription [226] combined with the monotonic basin hopping global search algorithm. In a similar manner that in our approach, their algorithm is able to automatically find optimal sequences of gravity assists, while optimizing the low-thrust control law and other design parameters. To allow proper comparisons, the same propulsive system than in [192] is used. The spacecraft for these mission has been given a VSI electric thruster with a nuclear power source. Their characteristics are summarized in Table 5.8. A constant I_{sp} value of 2000s is chosen for the simulations with MOLTO-IT Step 1, since it corresponds to the average value the engine can provide.

Table 5.7: Propulsion and power system 1

Variable	Value
Propulsion System	$c_{T0} = -1.9137$ N, $c_{T1} = 36.242$ N/kW
	$c_{m0} = 0.47556$ kg s^{-1}, $c_{m1} = 0.90209$ kg s^{-1}/kW
Power System	$P_0 = 10$ kW, $P_{min} = 10$ kW, $P_{max} = 10$ kW
	$\gamma_0 = 1.1063$, $\gamma_1 = 0.1495$ AU,
	$\gamma_2 = -0.299$ AU^2, $\gamma_3 = -0.0432$ AU^{-1}

Table 5.8: Propulsion and power system 2

Variable	Value
Propulsion System	VSI with 60% efficiency and I_{sp} in [1000,3000]s
Power System	1kW radioisotope with 2% decay per year
	$P_{min} = 200$ W

5.4.1 Case 1: Earth to Ceres Rendezvous

The first case is a low-thrust rendezvous mission to the asteroid Ceres. Both, direct and via Mars (M) flyby trajectories are allowed. Firstly, MOLTO-IT Step 1 searches for optimal trajectories over the year 2003 with a mission duration between 200 and 1400 days with respect to transfer time and propellant mass. The launch hyperbolic excess velocity is fixed to 1.6 km/s and the initial spacecraft mass is 568 kg. In order to estimate the propellant consumed, a constant specific impulse of 3000 s is chosen and Eq.(5.9) is applied. Thereafter, the minimum propellant mass solution of the Pareto front is refined in MOLTO-IT Step 2. The transfer time, launch velocity and initial mass are maintained, while the required propellant mass is optimized. The engine thrust, mass consumption and power available laws presented for the NSTAR engine are used. Finally, results from both steps are compared with each other, and with the solutions that Petropoulos and Longuski [47] reported for the same mission scenario. A summary of the problem characteristics and involved parameters is outlined in Table 5.9.

MOLTO-IT Step 1 was run several times with the GA parameters set to the values presented in Table 5.10. The mutation and crossover fractions were selected during a tuning process. After all simulations reached the maximum number of generations, equally Pareto-optimal solutions were obtained. On average, the entire population was feasible by completing the 10^{th} generation. At the 20^{th} generation, 80% of the population lied along a distinct non-dominated front, whereas from generation 20 to 30 the front was progressively shifted towards lower times of flight and propellant masses. After 30 generations were completed, members of the population were uniformly distributed along the front and no later improvement, in terms of non-domination, was observed. The average evaluation time per individual and per generation along with the total time to complete the 50 generations are shown in Table 5.11. Note that the computational speed could be improved if more cores were used during the simulation.

TABLE 5.9: E-C Problem definition

Description	Value
Optimization Objective	Transfer Time vs Propellant mass
Launch Window open date	Jan 01, 2003
Launch Window close date	Dec 31, 2003
Mission type	Rendezvous
Launch v_∞	1.6 km/s
Launch asymptote declination bounds	Free
Minimum Flyby number	0
Maximum Flyby number	1
Available Flyby Bodys	Mars
Minimum Flyby Radius	200 km
Minimum Flight Time	200 days
Maximum Flight Time	1400 days
Launch mass	568 kg
I_{sp} (MOLTO-IT Step1)	3000 s
Propulsion and Power (MOLTO-IT Step 2)	NSTAR

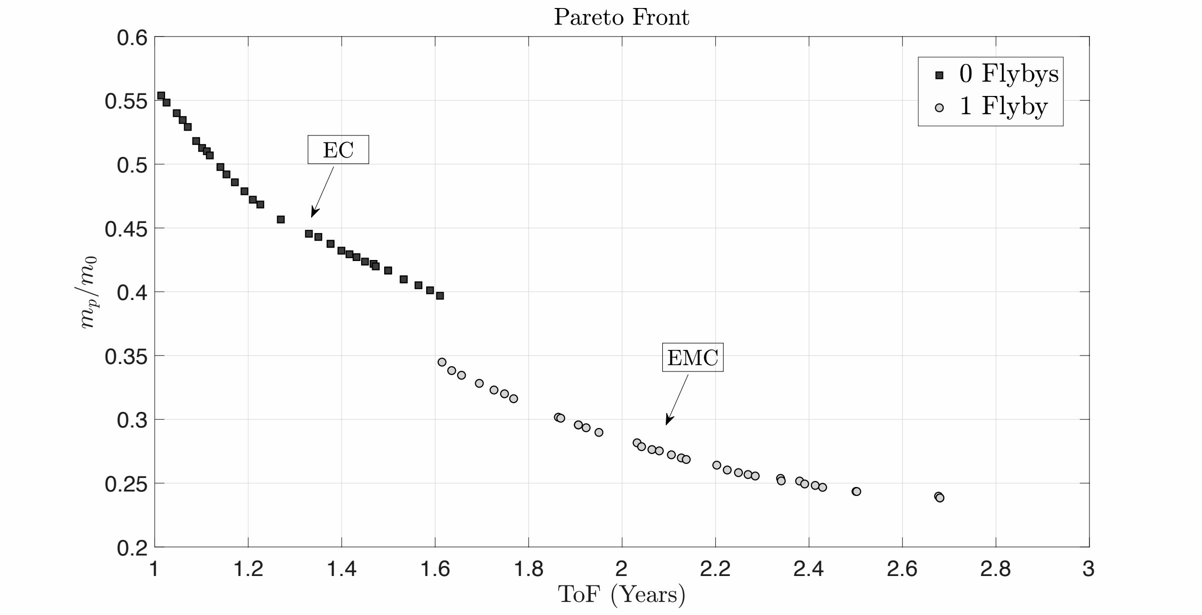


Figure 5.11: MOLTO-IT Step1: E -C Pareto front.

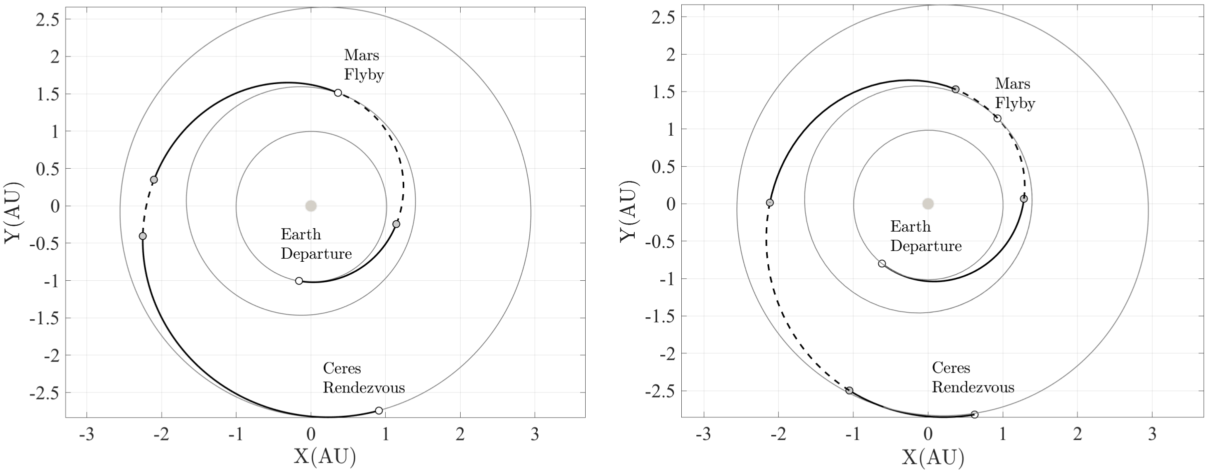
Table 5.10: MOLTO-IT Step 1: E-C Genetic algorithm parameters

Population size	100
Max. Generations	50
Mutation Fraction	0.3
Crossover Fraction	0.3

Table 5.11: MOLTO-IT Step 1: E-C Computational time

Variable	Avg. CPU Time
Population	0.46 s
Generation	6.5 s
Total	5.4 min

As an illustration, the last Pareto front solutions obtained from one of the simulations are presented in Figure 5.11. Trajectories with propellant fractions higher than 0.6 were pruned out during the optimization. It can be noted that a well spread set of solutions is obtained in just one run. They cover flight times ranging from 1 year to 2.8 years, and propellant mass fractions from 0.55 to 0.22. Additionally, results comprise both direct and via Mars flybys trajectories. Direct trajectories are optimal for flight times lower than 1.6 years, whereas trajectories with the intermediate flyby are preferable for longer times and lower propellant masses. Promising launch dates are revealed in early June 2003 for both direct and flyby transfer. The optimal Mars flyby date remains constant for all solutions in early February 2004. However, the day of encounter with Ceres is delayed as the transfer time increases. The earliest rendezvous option is in July 2004, whereas the latest chance can be accomplished in February 2006. Therefore, MOLTO-IT Step 1 provides the user with a wide variety of trajectories and the possibility of exploring various criteria at once, while requiring minimal information and interaction.



(A) MOLTO-IT Step 1

(B) MOLTO-IT Step 2

FIGURE 5.12: E-M-C Minimum fuel trajectory for 2.8 years

TABLE 5.12: MOLTO-IT Step 2: E-M-C parameters and performances

Nodes	100
Tolerance	10^{-6}
Iterations	50
Computational Time	3.2 min

TABLE 5.13: E-M-C Minimum propellant mass detailed solution

		MOLTO-IT		Petropoulos and Longuski [47]	
Parameter	Units	Step 1	Step 2	STOUR-LGTA	GALLOP
Launch Date	-	Jun 12, 2003	May 13, 2003	May 6, 2003	May 6, 2003
Launch v_∞	km/s	1.6	1.6	1.6	1.6
Initial Mass	kg	N/A	568	N/A	568
M Flyby Date	-	Feb 10, 2004	Dec 31, 2003	Feb 01, 2004	Feb 01, 2004
M Flyby v_∞	km/s	2.65	2.16	1.43	1.92
M B-Plane angle	deg	0	54.68	2.3	82.3
M Flyby Altitude	km	200	200	5432	200
Mass	kg	N/A	531	N/A	N/A
C Arrival Date	-	Feb 20, 2006	Jan 20, 2006	Jun 12, 2006	Feb 09, 2006
C Arrival v_∞	km/s	0	0	0.237	0
Mass	kg	N/A	437	N/A	N/A
Prop. fraction	-	0.224	0.229	0.256	0.233

The minimum propellant solution of the Pareto front of Fig. 5.11 was chosen as an initial guess for MOLTO-IT Step 2. A grid of 50 nodes per leg uniformly distributed along time were used to discretize the trajectory and the tolerance for the NLP-solver was set to 10^{-6}, both for optimality and constraints infeasibility. Initially, the NLP infeasibility was 10^{-2}, mainly due to the mismatch between the out-of-plane position and velocity of Ceres and the one provided by the planar model of the guessed trajectory. In spite of this, MOLTO-IT Step 2 converged to an optimal and feasible solution after 50 NLP-iterations and after 3.2 min. In Table 5.12, the aforementioned parameters and performances of MOLTO-IT Step 2 are summarized.

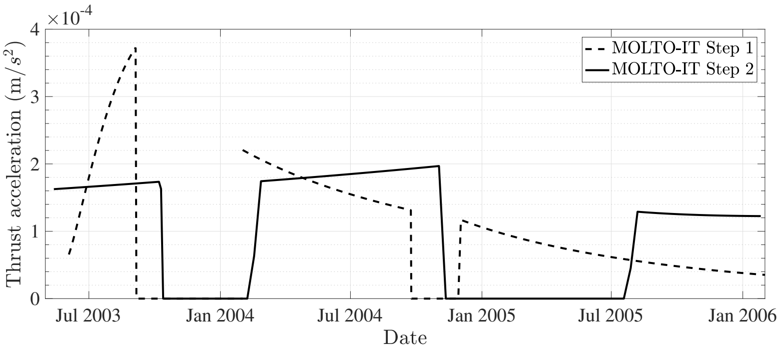


FIGURE 5.13: Time history of thrust acceleration for E-M-C trajectory

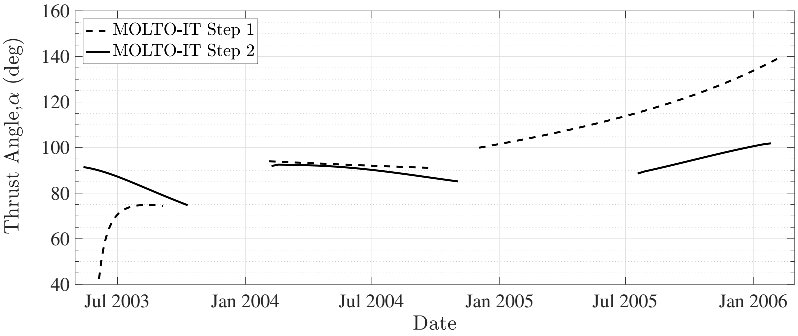


FIGURE 5.14: Time history of in-plane thrust angles for E-M-C trajectory

The geometry of the trajectories from MOLTO-IT Step 1 and MOLTO-IT Step 2 are shown in Figs. 5.12a and 5.12b respectively. The dashed lines represent the coast arcs, whereas the continuos lines denotes the thrusting arcs. Note the ability of MOLTO-IT Step 2 to automatically change the thrusting schedule provided by the initial guess. This feature si also observed in Figs. 5.13 and 5.14, where the thrust acceleration and in-plane steering angle are presented for both solutions. As an example, a coast arc is automatically introduced after the Mars flyby. Additionally, the last thrust arc is significantly shorten after the second step. This is because the acceleration profile of the shape-based method during this arc is lower than the maximum acceleration, thus it requires a longer thrusting period to rendezvous with Ceres. Notably, the peak acceleration obtained in MOLTO-IT Step 1 doubles the maximum acceleration available in MOLTO-IT Step 2. Nevertheless, this fact has not prevented the algorithm to converge.

A detailed description of the aforementioned trajectories are presented in Table 5.13. Note that the optimal launch occurs 1 month before predicted by MOLTO-IT Step 1. Additionally, MOLTO-IT Step 2 is able to adjust the B-plane flyby angle to perform a plane change at Mars. These solutions are compared with the results presented by Petropoulos [47] using STOUR-LGTA and GALLOP. The solution from STOUR-LGTA is dominated by the trajectory from MOLTO-IT Step 1, exhibiting a 14% higher propellant mass fraction for a 143 days longer transfer. Remarkably, STOUR-LGTA solution does not fulfill the rendezvous condition, i.e. an additional maneuver is required at arrival, whereas MOLTO-IT Step 1 solution does. Similarly, the solution from MOLTO-IT Step 2 outperforms the trajectory from GALLOP defining a 1-month earlier flyby on Mars, arriving 20 days earlier at Ceres and consuming 1.72% less fuel.

5.4.2 Case 2: Earth to Jupiter Flyby

In this second case, a low-thrust mission to flyby the planet Jupiter (J) is addressed. Direct trajectories, as well as those involving up to 3 intermediate gravity assists on Mars (M), Venus (V) and Earth (E) are considered. The goal is to search for optimal trajectories with respect to transfer time and propellant mass and within the launch window starting in January 2029 and closing in December 2030. The minimum duration for the mission is 100 days, while the transfer time must not exceed 1500 days. The launch hyperbolic excess velocity is fixed to 2 km/s and the initial spacecraft mass is 568 kg. In order to estimate the propellant consumed during the simulations for MOLTO-IT Step 1, a constant specific impulse of 3000 s is chosen and Eq.(5.9) is applied. During the optimizations with MOLTO-IT Step 2, the engine thrust, mass consumption and power available laws presented for the NSTAR engine are used. A summary of the problem characteristics and involved parameters is outlined in Table 5.14.

Firstly, four different simulations are performed with MOLTO-IT Step 1, considering a fixed number of 0, 1, 2 and 3 flybys respectively, yet letting the optimizer select the optimal bodies at which the gravity assisted maneuvers are performed. From each simulation a different set of Pareto fronts are obtained. Secondly, from each of the previous four Pareto fronts, a set of solutions is chosen. They are used as initial guess for running MOLTO-IT Step 2 to generate four new refined Pareto fronts. Solutions from both steps, including important event dates, flyby velocities and maximum accelerations, are compared. Thirdly, a new run of MOLTO-IT Step 1 is carried out, where the optimal number and sequence of flybys is now determined by the optimizer. The resulting Pareto front is juxtaposed with those obtained in the cases with fixed flybys numbers. Finally, the minimum propellant mass solutions obtained with MOLTO-IT Step 1 and with MOLTO-IT Step 2 are compared with each other, and with the solutions that Petropoulos and Longuski [47] reported for the same mission scenario.

Table 5.14: E-J Problem definition

Variable	Value
Optimization objective	Transfer time vs Propellant mass
Launch Window open date	Jan $01, 2029$
Launch Window close date	Dec $31, 2030$
Mission type	flyby
Launch v_∞	2 km/s
Launch asymptote declination bounds	Free
Minimum Flyby number	0
Maximum Flyby number	3
Available Flyby Bodys	Mars, Venus, Earth
Minimum Flyby Radius	200 km
Minimum Flight Time	100 days
Maximum Flight Time	1500 days
Launch mass m_0	360 kg
I_{sp} (MOLTO-IT Step1)	3000 s
Propulsion and Power (MOLTO-IT Step 2)	NSTAR

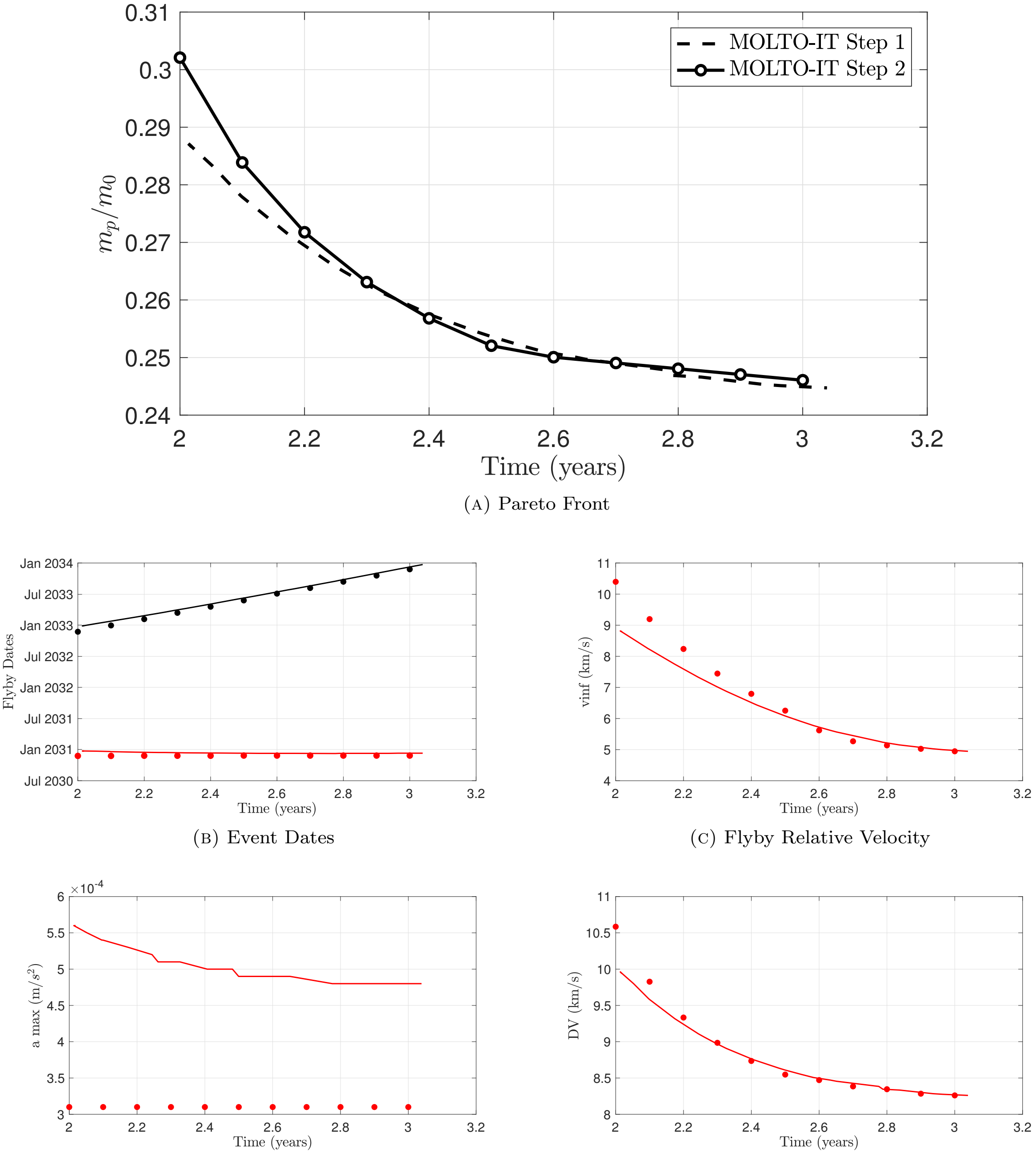
Table 5.15: MOLTO-IT Step 1: E-J GA Parameters for fixed flyby cases

Population size	50
Max. Generations	50
Mutation Fraction	0.3
Crossover Fraction	0.3

In the following pages, the results obtained from running MOLTO-IT with a fixed number of flybys are shown: In Fig. 5.15 for direct trajectories, in Fig. 5.16 for trajectories with 1 flyby, in Fig. 5.17 for trajectories with 2 flybys and in Fig. 5.18 for trajectories with 3 Flybys. Firstly, MOLTO-IT Step 1 was run using the GA parameters summarized in Tab. 5.15. During these simulations the following sequences have been reported to be optimal: EJ, EVJ, EMJ, EVEJ and EVEMJ, which agrees with the results available in the literature [47]. Computational times for each scenario are summarized in Tabs. 5.16-5.22. It can be seen how the average time required to evaluate one member of the population increases as the number of flybys increases from 0.14 s for direct trajectories to 0.45 s for trajectories with three flybys. This result agrees with the fact that in the former case, the inner loop only solves 1 NLP per population, whereas in the latter case 4 NLPs need to be tackled, one per each interplanetary leg. Consequently, the total time also increases, from 1.53 min to 4.85 min.

Thereafter, Pareto fronts were obtained with MOLTO-IT Step 2. For such purpose, a time grid of 0.1 years was defined between the minimum and maximum mission time. For each discrete time in the grid and for each flyby configuration, a minimum-fuel fixed-time problem was solved. As initial guess, the closest candidate from MOLTO-IT Step 1 in terms of time of flight was chosen. Results have been displayed in Figs. 5.15-5.18 for comparing the re-optimized performances with the solutions from the previous step. The average number of NLP-iterations and computational times for obtaining a single-point solution of the Pareto with MOLTO-IT Step 2 are outlined: in Tabs. 5.17- 5.23 for trajectories from 0 to 3 flybys respectively. It can be noticed that the number of required NLP-iterations rises from 23 to 108 as the number of interplanetary legs increases, as well as the execution times from 0.74 min to 12.5 min.

Pareto fronts for EJ trajectories approximately match. For the remaining cases MOLTO-IT Step 2 reduces on average 30% the propellant mass obtained with MOLTO-IT Step 1. Additionally, other plots comparing solutions from both steps are included: event dates (i.e., launch, flybys and arrival), flybys relative velocities, and peak acceleration and total ΔV for each interplanetary leg. Optimal launch dates in December 2030 are found for EJ transfers, in April 2030 for EVJ, in April 2029 for EMJ, and in October 2029 for EVEJ and EVEMJ. Notably, the predicted flyby dates by using the shape-based strategy approximately coincide with the optimal ones obtained with the complete dynamical model, except for the EVJ sequence. In this case, the trajectory obtained by MOLTO-IT Step 1 arrives faster at Venus, in particular, one period of Venus earlier. The remaining parameters do not typically agree, due to the different dynamical formulations between both steps. Nevertheless, solutions from MOLTO-IT Step 1 represent a good initial estimation for MOLTO-IT Step 2.



(A) Pareto Front

(B) Event Dates

(C) Flyby Relative Velocity

(D) Maximum Acceleration

(E) ΔV

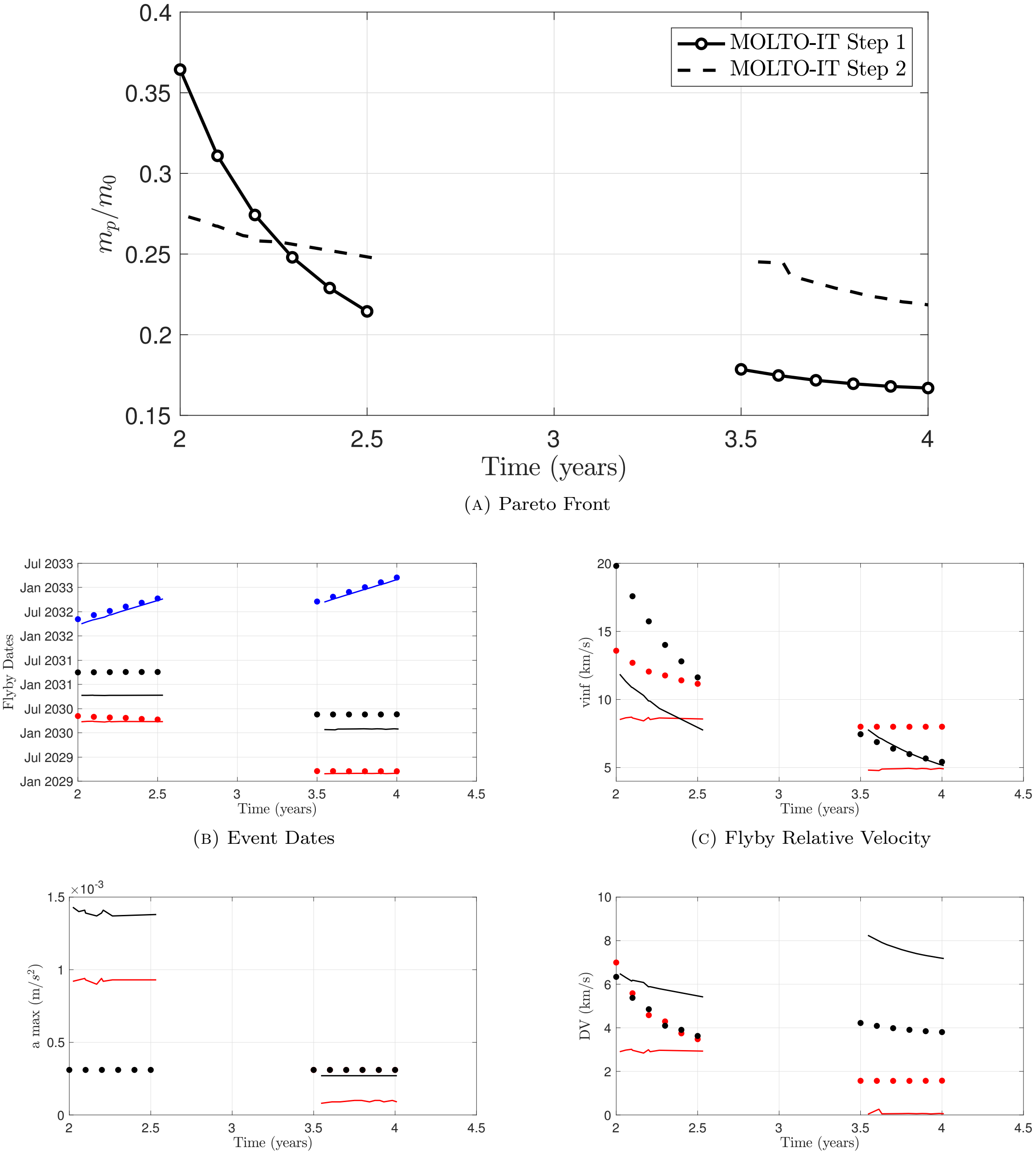
FIGURE 5.15: MOLTO-IT Step 1 (lines) and MOLTO-IT Step 2 (dots) for direct transfers.

TABLE 5.16: MOLTO-IT Step 1: E-J 0 flyby case computational time

Variable	Avg. CPU Time
Population	0.14 s
Generation	1.9 s
Total	1.53 min

TABLE 5.17: MOLTO-IT Step 2: E-J 0 flyby case parameters

Nodes	50
Tolerance	10^{-6}
Avg. Iterations	23
Avg. Computational Time	0.74 min



(A) Pareto Front

(B) Event Dates

(C) Flyby Relative Velocity

(D) Maximum Acceleration

(E) ΔV

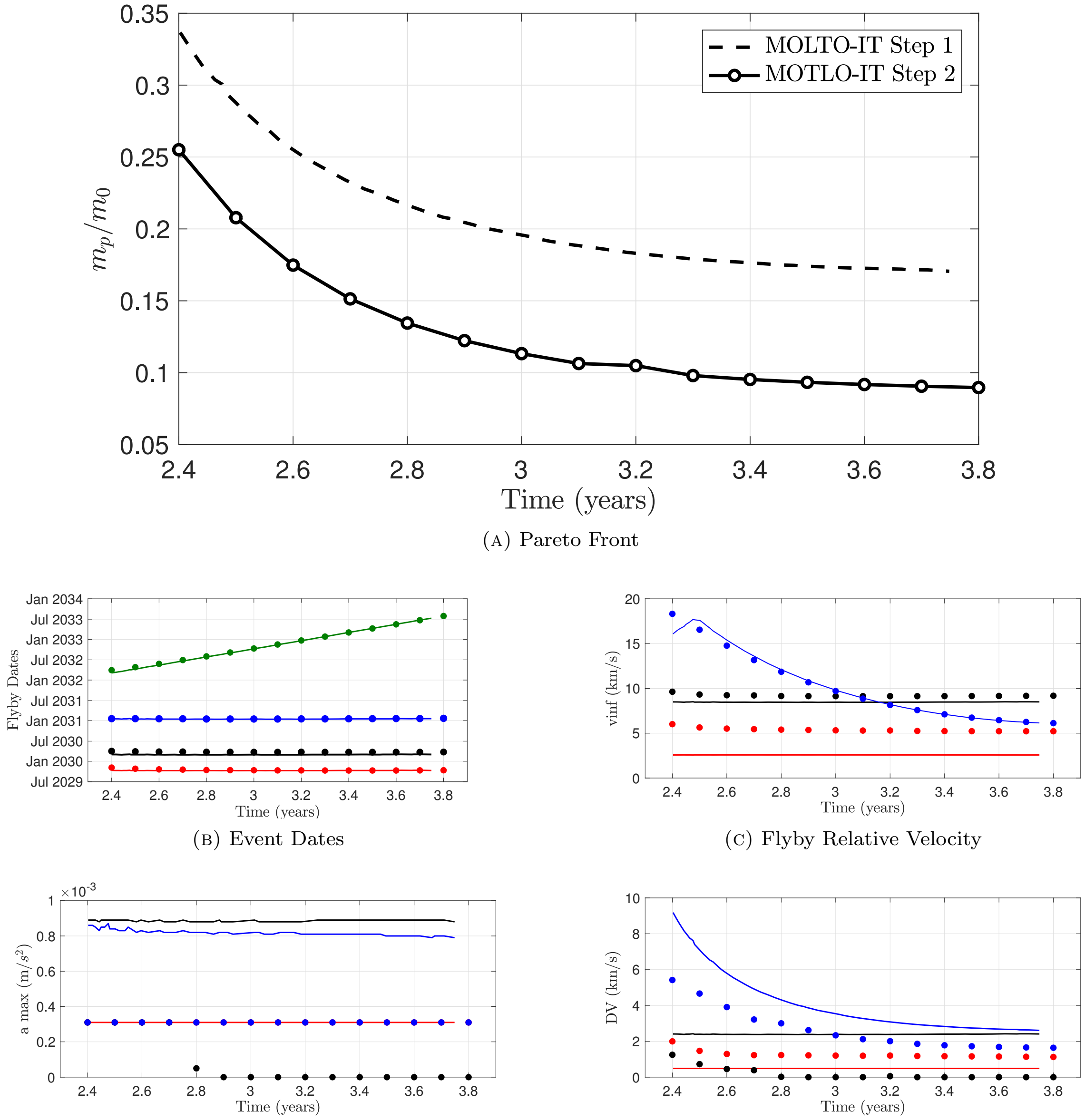
FIGURE 5.16: MOLTO-IT Step 1 (lines) and MOLTO-IT Step 2 (dots) for 1 flyby.

TABLE 5.18: MOLTO-IT Step 1: E-J 1 flyby case computational time.

Variable	Avg. CPU Time
Population	0.23 s
Generation	3.1 s
Total	2.6 min

TABLE 5.19: MOLTO-IT Step 2: E-J 1 flyby case parameters.

Nodes	100
Tolerance	10^{-6}
Avg. Iterations	46
Avg. Computational Time	2.9 min



(A) Pareto Front

(B) Event Dates

(C) Flyby Relative Velocity

(D) Maximum Acceleration

(E) ΔV

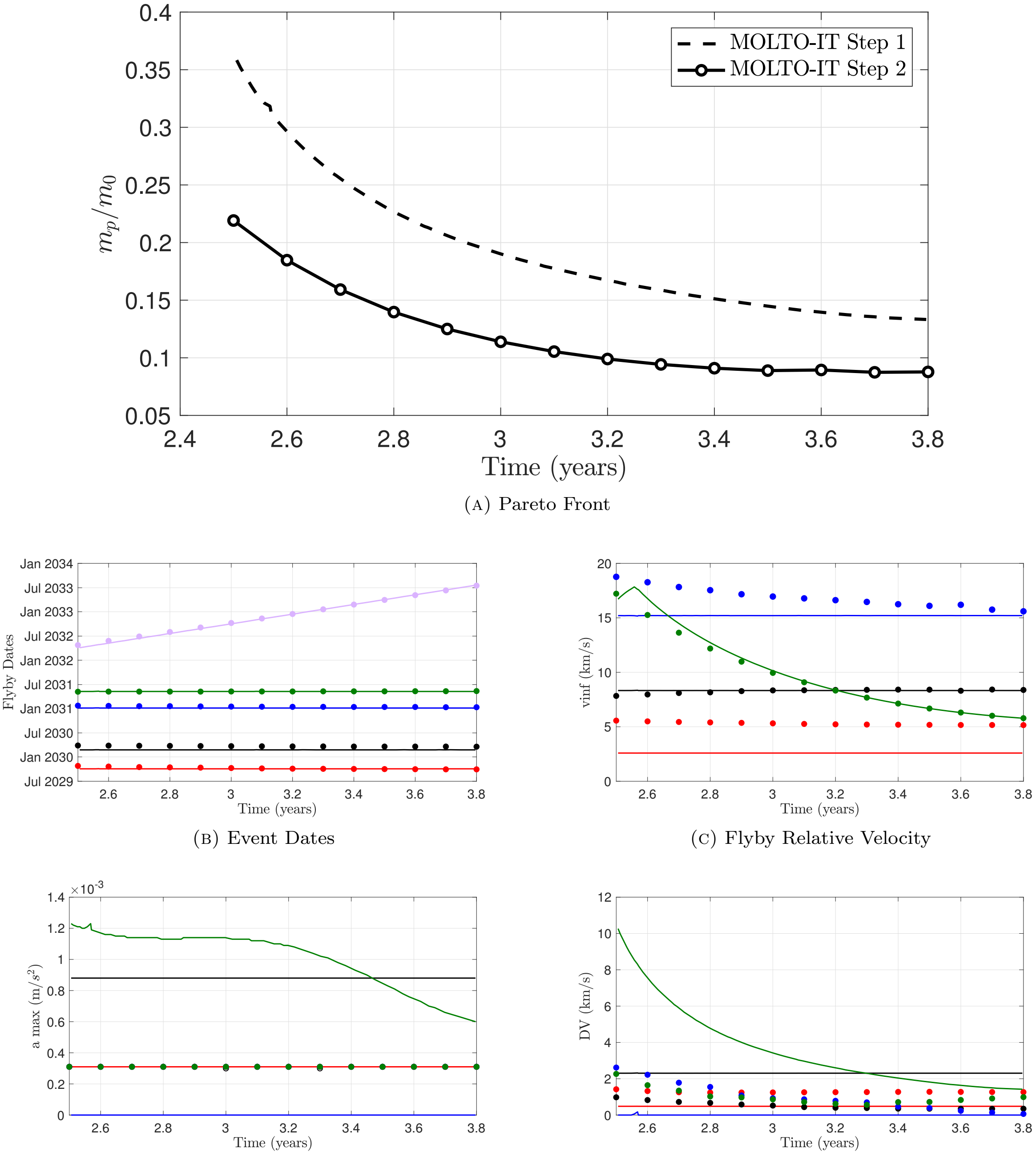
FIGURE 5.17: MOLTO-IT Step 1 (lines) and MOLTO-IT Step 2 (dots) for 2 flybys.

TABLE 5.20: MOLTO-IT Step 1: E-J 2 flybys case computational time.

Variable	Avg. CPU Time
Population	0.33 s
Generation	4.3 s
Total	3.63 min

TABLE 5.21: MOLTO-IT Step 2: E-J 2 flybys case parameters.

Nodes	150
Tolerance	10^{-6}
Avg. Iterations	81
Avg. Computational Time	5.8 min



(A) Pareto Front

(B) Event Dates

(C) Flyby Relative Velocity

(D) Maximum Acceleration

(E) ΔV

Figure 5.18: MOLTO-IT Step 1 (lines) and MOLTO-IT Step 2 (dots) for 3 flybys.

Table 5.22: MOLTO-IT Step 1: E-J 3 flybys case computational time

Variable	Avg. CPU Time
Population	0.45 s
Generation	5.7 s
Total	4.85 min

Table 5.23: MOLTO-IT Step 2: E-J 3 flybys case parameters

Nodes	200
Tolerance	10^{-6}
Avg. Iterations	108
Avg. Computational Time	12.5 min

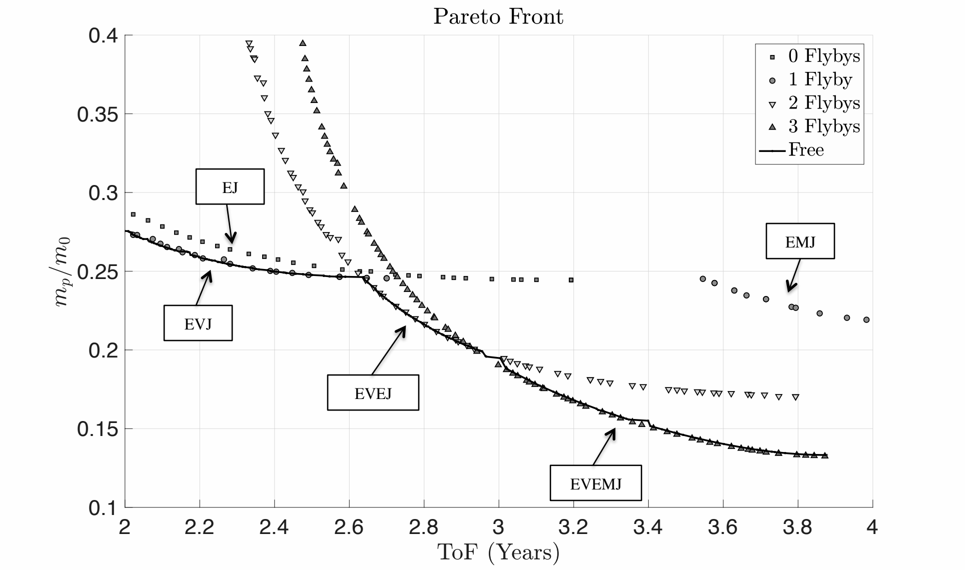


FIGURE 5.19: MOLTO-IT Step 1: E -J Pareto front for fixed and free number of flybys.

TABLE 5.24: MOLTO-IT Step 1: E-J Free flyby number case computational time

Variable	Avg. CPU Time
Population	0.33 s
Generation	4.12 s
Total	4 min

Thereafter, the problem was solved again applying MOLTO-IT Step 1 without setting a fixed flyby number, i.e., the algorithm was in charge of automatically obtaining the optimal amount of gravity assists for each region in the Pareto front. The same parameters for the GA indicated in Tab. 5.15 were selected. Besides, the number of flybys was included as the third objective in the fitness function. This technique has been incorporated to maintain the diversity in the solution, forcing the algorithm to keep at least one member of the population for each possible configuration. Without this approach, trajectories with long flyby sequences may be pruned out early in the process, because feasible and low-cost trajectories are more difficult to find as the number of flybys increases. A different method would imply increasing the number of population. However, this strategy would result in unnecessarily large execution times.

Figure 5.19 illustrates the population of Pareto optimal trajectories obtained therefrom. It can be seen that the algorithm converged to three different flyby sequences: EVJ, EVEJ and EVEMJ for an increasing time of flight respectively. Note that, in this case EJ and EMJ configurations are not part of the Pareto solution. In order to corroborate that the algorithm outputted the globally optimal sequences of flybys, solutions from the previous fixed flybys cases are also displayed. It can be observed that the free flyby solution corresponds to the envelope of the fixed flyby cases. Notably, the total time needed to compute all the fixed flyby solutions was 12.6 min, whereas the free flyby case terminated successfully after 4 min (see Table 5.24). Hence, the automatic algorithm is able to find the same solutions in terms of Pareto optimality three times faster than running the four fixed flyby cases independently.

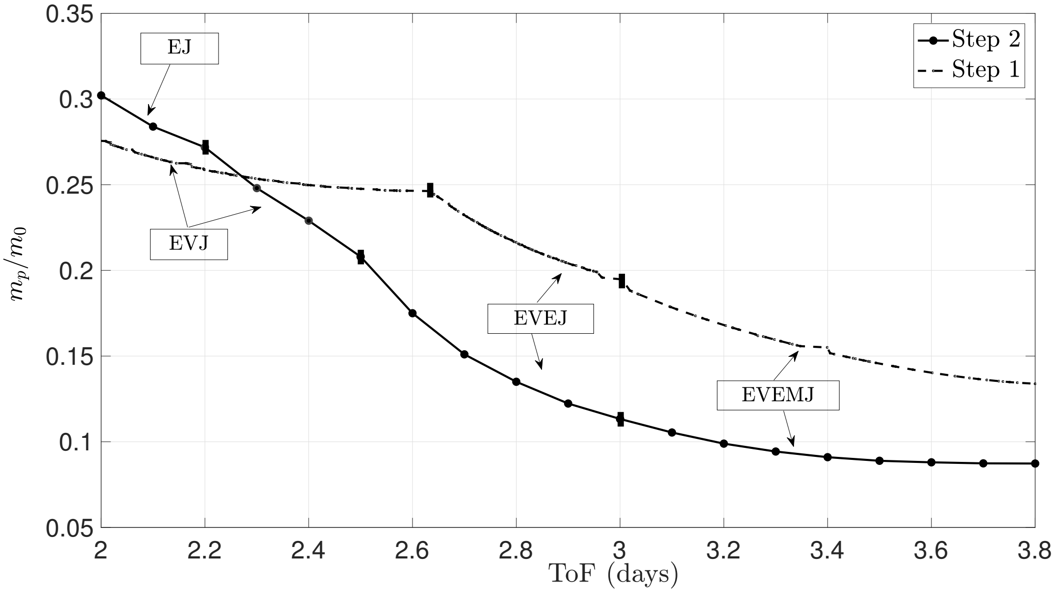
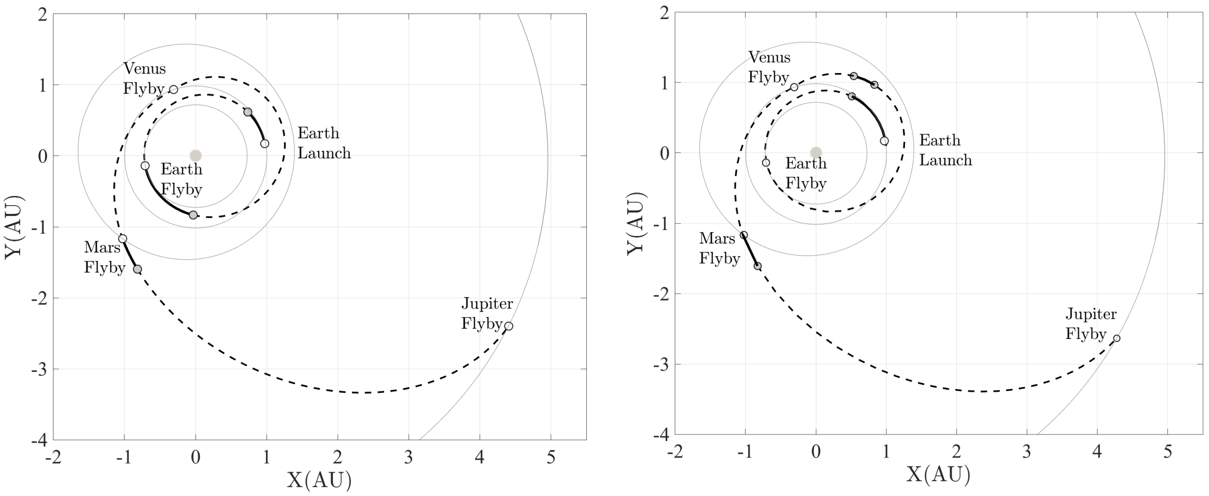


Figure 5.20: MOLTO-IT Step 1 Pareto vs MOLTO-IT Step 2 Pareto

In Figure 5.20, the non-dominated solutions obtained from MOLTO-IT Step 1 and MOLTO-IT Step 2 from the previous simulations are compared. It is clear that MOLTO-IT Step 2 Pareto is mainly below the one obtained in MOLTO-IT Step 1, because in the former the control profile is not constrained to follow the one of the shape-based method. The relative reduction in propellant mass achieved after MOLTO-IT Step 2 ranges from 20% to 40%. Additionally, it is worth noting that the switching point from EVEJ to EVEMJ is well predicted in MOLTO-IT Step 1 in terms of flight time. However, the region corresponding to one flyby is not so well predicted as it has already been discussed. Previous comparisons suggest that the shape-based solution provides a good starting point for the direct optimizer without any a-priori knowledge of the optimal solution and that performances obtained are close to the true optimal trajectories, even for long flyby sequences. This approximation works better for transfers to the outer planet than for those to the inner solar system.

The geometry of the trajectories with minimum fuel expenditures from MOLTO-IT Step 1 and MOLTO-IT Step 2 are displayed in Fig. 5.21a and Fig. 5.21b respectively. They correspond to a 3.2 years EVEMJ configuration. Both results are listed and compared in Table 5.25 with those obtained by Petropoulos and Longuski [47] using STOUR-LGTA and GALLOP for the same flyby sequence. It can be seen that the solution from MOLTO-IT Step 1 exhibits a 55% lower propellant mass than STOUR-LGTA for a 552 days shorter transfer mission. Remarkably, the average thrust acceleration for each leg predicted by the spirals is much lower than the one predicted by the exponential sinusoids. Besides, a purely coasting leg from Venus to Mars is found in our solution. Similarly, MOLTO-IT Step 2 solution presents a 66% lower propellant mass than the one obtained by GALLOP. Probably, the solution found by STOUR-LGTA corresponds to a local minimum, thus GALLOP could only converge in the neighborhood of this trajectory. This fact highlights the importance of selecting a good initial guess trajectory.



(a) MOLTO-IT Step 1

(b) MOLTO-IT Step 2

Figure 5.21: E-V-E-M-J Minimum fuel trajectory for 3.2 years

Table 5.25: E-V-E-M-J Minimum propellant mass detailed solution.

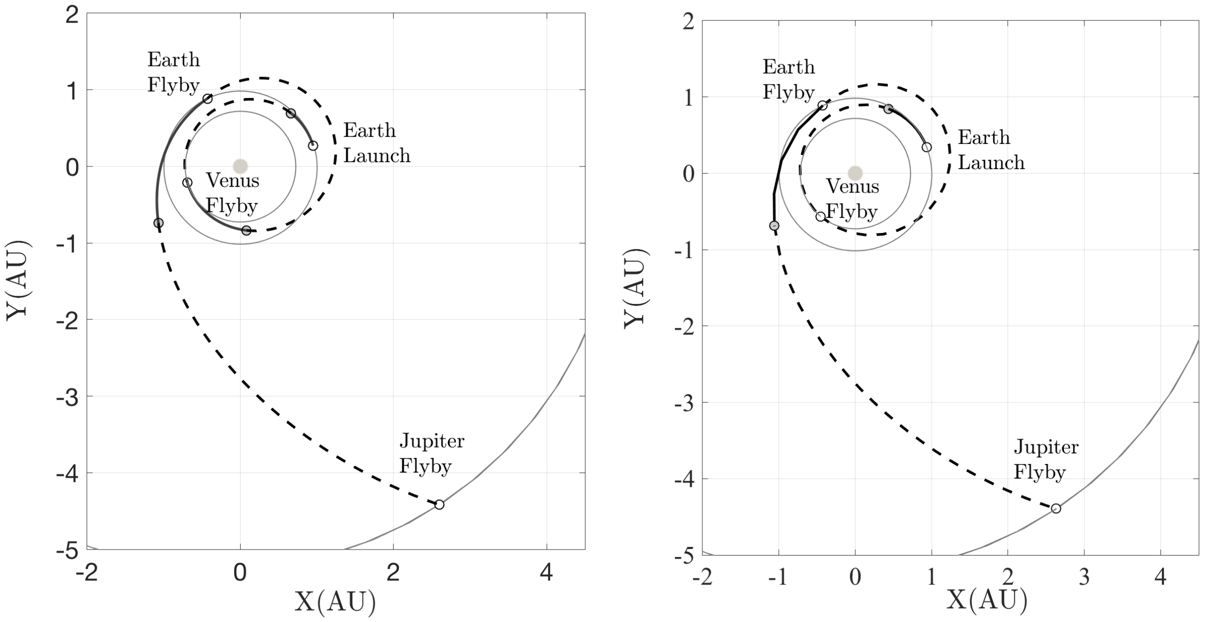
		MOLTO-IT		Petropoulos and Longuski [47]	
Parameter	Units	Step 1	Step 2	STOUR-LGTA	GALLOP
Launch Date	-	Oct 1, 2029	Sept 28, 2029	Sept 3, 2029	Sept 3, 2029
Launch v_∞	km/s	2	2	2	2
Initial mass	kg	N/A	360	360	360
V Flyby Date	-	Feb 22, 2030	Mar 19, 2030	Feb 15, 2030	Feb 15, 2030
V Flyby v_∞	km/s	2.59	5.15	3.64	3.77
V Flyby Altitude	km	119,985	15,720	6,533	30,000
V B-Plane angle	deg	180.0	69.6	178.3	60.7
Avg. thrust accel.	mm/s^2	0.03	0.10	0.12	N/A
Mass	kg	N/A	344.47	N/A	N/A
E Flyby Date	-	Jan 04, 2031	Jan 11, 2031	Jan 15, 2031	Dec 30, 2030
E Flyby v_∞	km/s	8.33	8.32	6.50	5.18
E Flyby Altitude	km	201	200	655	1,035
E B-Plane angle	deg	180.0	-175.5	-180.0	-176.5
Avg. thrust accel.	mm/s^2	0.15	0.01	0.16	N/A
Mass	kg	N/A	339.66	N/A	N/A
M Flyby Date	-	May 09, 2031	May 13, 2031	May 26, 2031	May 26, 2031
M Flyby v_∞	km/s	15.21	15.43	13.70	11.26
M Flyby Alttitude	km	200	200	200	200
M B-Plane angle	deg	0.0	-9.0	-1.8	-5.5
Avg. thrust accel.	mm/s^2	0.00	0 .00	0.10	N/A
Mass	kg	N/A	339.66	N/A	N/A
Arrival Date	-	Aug 14, 2033	Aug 21, 2033	Jan 20, 2035	Jan 20, 2035
Arrival v_∞	km/s	5.65	5.62	5.85	6.25
Avg. thrust accel.	mm/s^2	0.09	0.02	0.14	N/A
Mass	kg	N/A	328.92	N/A	N/A
Prop. fraction	-	0.132	0.088	0.294	0.256

5.4.3 Case 3: Earth to Jupiter Flyby with Fixed Reorientation Times

This example is based on the results obtained for the Jupiter flyby mission in the previous section. In those tests, the spacecraft was allowed to continuously reorient for performing the low-thrust maneuvers. However, at this time the spacecraft is constrained to maintain the inertial attitude during a given minimum period of time to comply with the operational constraint. For running this tests the EVEJ configuration for a mission duration of 2.9 years is selected. The geometry of the trajectories obtained from MOLTO-IT Step 1 and MOLTO-IT Step 2 are displayed in Fig. 5.22a and Fig. 5.22b respectively. Detailed results for both mission profiles are presented in Table 5.26. The trajectory from the second step consumes 43.9 kg of propellant. The profile consists on a thrust-coast sequence from Earth to Venus, a coast phase From Venus to Earth and a thrust/coast arc to encounter Jupiter. The duration of the first thrust arc is 50 days, while the duration of the second arc is 100 days.

The unconstrained solution from MOLTO-IT Step 3 was used to feed MOLTO-IT Step 2 again but with the operational constraint activated. Four different simulations were carried out. Each of them imposed a minimum re-orientation time of 10, 15, 20, 40 days respectively. The number of re-orientations is not imposed but optimized by the software. The initial date is fixed and the mission duration is constrained to be 2.9 years to agree with the initial guess and properly evaluate the penalty in the propellant mass consumed. The thrust arcs provided by the initial guess are divided into sub-arcs of the same duration. For these simulations, the first and second thrust arcs were divided in 5 and 14 phases respectively. Then, at each sub-arc a constant orientation angle is assumed and the nonlinear condition defined in Eq. **??** is applied. The algorithm is in charge of selecting which sub-arcs has to be enlarged to fulfill with the conditions and which ones have to be deleted because of being unnecessary. Solutions were obtained on an average of 50 NLP-iterations and 6 minutes.

The obtained in-plane and out-of-plane thrusting angles, measured with respect to the inertial reference frame, for each case are plotted in Figs 5.23a-5.23d and compared with the unconstrained case. In all cases, the total duration of the thrust arcs do not vary with respect to the free case, i.e. the optimal on/off switching dates do not vary even though the minimum re-orientations times are fixed. More detailed information is presented in Tables 5.28-5.31, where the notation M_i refers to the i^{th} maneuver. Information regarding the duration of the maneuver, and the fixed inertial in-plane and out-of-plane angles is incorporated. The optimal inertial angles correspond to the average value of the free case for the same time-period. Note that the longest maneuvers are found when the angular velocity of the free solution is lower. In this example, it happens at the beginning of the first arc and at the end of the second one. In Table 5.27 the propellant mass fractions for each case are shown. It can be be seen that the required propellant mass increases as the reorientation time increases, because the less freedom to maneuver the less optimal the trajectory. Notably, the mission can be accomplished with only three re-orientations carrying 1.31% of additional propellant mass, i.e. 576 grams more.



(A) MOLTO-IT Step 1

(B) MOLTO-IT Step 2

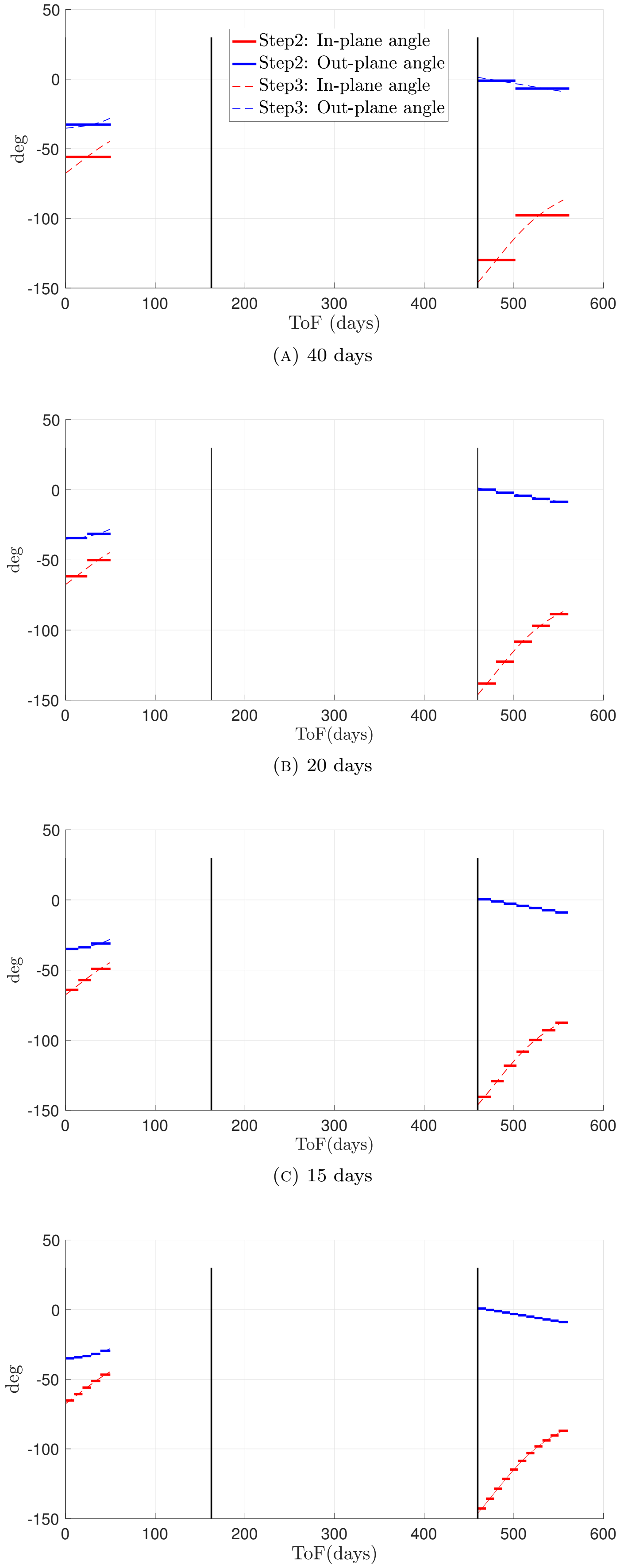
FIGURE 5.22: E-V-E-M-J Minimum fuel trajectory for 3.2 years

TABLE 5.26: E-V-E-J Minimum propellant detailed solution

		MOLTO-IT	
Parameter	Units	Step 1	Step 2
Launch Date	-	Oct 1, 2029	Oct 12, 2029
Launch v_∞	km/s	2	2
Initial mass	kg	N/A	360
V Flyby Date	-	Feb 20, 2030	Mar 24, 2030
V Flyby v_∞	km/s	2.49	5.36
V Flyby Altitude	km	11,285	16,285
V B-Plane angle	deg	180	-62
Mass	kg	N/A	287
Earth Flyby Date	-	Jan 06, 2031	Jan 15, 2031
E Flyby v_∞	km/s	9.05	9.13
E Flyby Altitude	km	200	200
E B-Plane angle	deg	180	180
Mass	kg	N/A	287
Arrival Date	-	Oct 29, 2032	Sep 05, 2032
Arrival v_∞	km/s	10.62	10.68
Mass	kg	N/A	263
Propellant mass fraction	-	0.20	0.12

TABLE 5.27: E-V-E-J Propellant mass fractions for different reorientation times

Case	Propellant mass fraction
Free	0.1220
10 days	0.1220
15 days	0.1223
20 days	0.1224
40 days	0.1236



(A) 40 days

(B) 20 days

(C) 15 days

(D) 10 days

FIGURE 5.23: MOLTO-IT Step 2: Steering angles for free fixed reorientations times

Table 5.28: E-V-E-J Maneuver plan for minimum 40 days reorientation times

Variable	M1	M2	M3
Duration (days)	50.27	41.13	58.97
In-plane ang. (deg)	-55.56	-131.02	-99.16
Out-of-plane ang. (deg)	-32.37	-1.12	-7.20

Table 5.29: E-V-E-J Maneuver plan for minimum 20 days reorientation times

Variable	M1	M2	M3	M4	M5	M6
Duration (days)	25.36	24.65	20.81	21.18	23.75	33.22
In-plane ang. (deg)	-61.34	-49.69	-138.67	-122.19	-106.77	-92.62
Out-of-plane ang. (deg)	-34.41	- 30.87	0.10	-2.38	-5.07	-8.47

Table 5.30: E-V-E-J Maneuver plan for minimum 15 days reorientation times

Variable	M1	M2	M3	M4	M5	M6	M7	M8	M9
Duration (days)	17.19	16.48	16.29	10.00	15.00	15.00	15.11	17.02	21.62
In-plane ang. (deg)	-63.18	-55.14	-47.64	-141.08	-129.23	-117.69	-107.36	-98.35	-90.19
Out-of-plane ang. (deg)	-34.84	-33.03	-30.03	0.49	-1.28	-3.07	-4.89	-6.84	-9.10

Table 5.31: E-V-E-J Maneuver plan for minimum 10 days reorientation times

Variable	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14
Duration (days)	10.00	10.00	10.00	10.00	10.00	10.43	10.46	10.47	10.58	10.78	11.14	11.32	11.33	11.79
In-plane ang. (deg)	-64.98	-60.14	-55.28	-50.56	-46.23	-142.69	-134.37	-126.05	-117.98	-110.44	-103.59	-97.50	-92.38	-88.24
Out-of-plane ang. (deg)	-35.16	-34.31	-33.17	-31.59	-29.29	0.74	-0.50	-1.74	-3.00	-4.29	-5.63	-7.01	-8.37	-9.67

5.4.4 Case 4: Earth to Pluto Rendezvous

The last example considers a low-thrust rendezvous mission to Pluto. The goal is to search for trajectories that maximizes the dry mass at arrival within the launch window opening in January 2025 and closing in January 2035 for a flight time up to 25 years. The algorithm is allowed to choose up to four flybys of Venus (V), Earth (E), Mars (M), Jupiter (J), Saturn (S), or Uranus (U). The spacecraft is equipped with the nuclear propulsion system of Table 5.8. Unlike in the previous examples, the initial mass and launch velocity are not fixed. Instead, the spacecraft launches on an Atlas V 551 with a Star 48 upper stage. Thus, the initial mass becomes an optimization variable, and the launch velocity is obtained from the launcher performance model shown in Eq.(5.44), where the fitting coefficients are taken from Ref. [229]. A Newton's method solves the resulting non-linear equation. Standard preliminary design margins are applied as Englander and Conway [192]: 15% power margin, 10% propellant margin and 90% duty cycle.

Firstly, two different tests are run with MOLTO-IT Step 1: one applied the shaped-based method as default, whereas the other imposes a propulsion constraint of Eq.(5.26) to help to prune out infeasible trajectories. The pruning criteria also account for the power and propellant margins as well as for the duty cycle. To estimate the propellant consumed, a constant specific impulse of 2000 s is chosen and Eq.(5.9) is applied. Secondly, the maximum dry mass solution of each Pareto front is refined in MOLTO-IT Step 2 including launch asymptote declination bounds. The initial mass remains an optimization variable which relates with the launch velocity via the launcher performance and only the flight time is limited up to 25 years. The time history for the I_{sp} is also optimized by the algorithm. It has been found that trajectories without the propulsion constraint are infeasible, whereas the one with the constraint fairly approximate the optimal solution. Results are compared with those reported by Englander and Conway [192].

Table 5.32: E-P Problem definition

Variable	Value
Optimization objective	Transfer Time vs Dry Mass
Launch Window open date	Jan 01, 2025
Launch Window close date	Jan 01, 2035
Mission type	Rendezvous
Maximum Flight Time	25 years
Launcher vehicle	Atlas V 551 with Star 48 upper stage
Launch asymptote declination bounds	[-28.5,28.5] (Kennedy Space Center)
Minimum Flyby number	0
Maximum Flyby number	4
Flyby Bodys	Venus, Earth, Mars, Jupiter, Saturn, Uranus
Minimum flyby Radius	200 km
Launch mass bounds	[1400, 3500] kg
Propulsion and Power system	VSI with 60% efficiency and I_{sp} in [1000,3000]s
Power margin	15%
Propellant margin	10%
Duty cycle	90%

Table 5.33: MOLTO-IT Step 1: E-P GA parameters for fixed flyby cases

Population size	50
Max. Generations	100
Mutation Fraction	0.3
Crossover Fraction	0.3

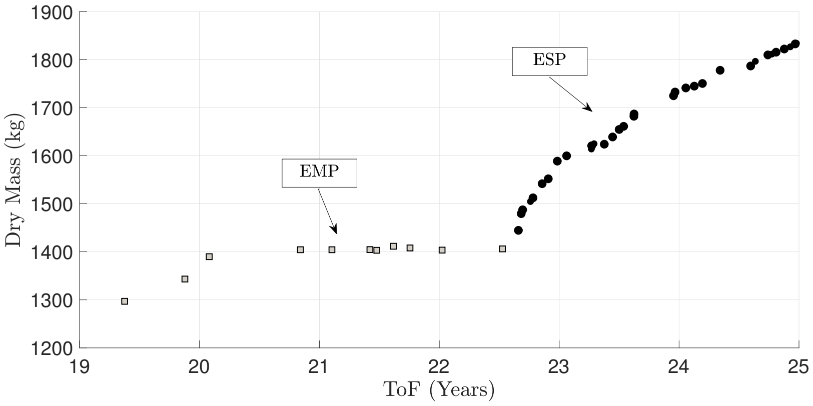


Figure 5.24: MOLTO-IT Step 1: E-P Pareto front solution without propulsion constraint

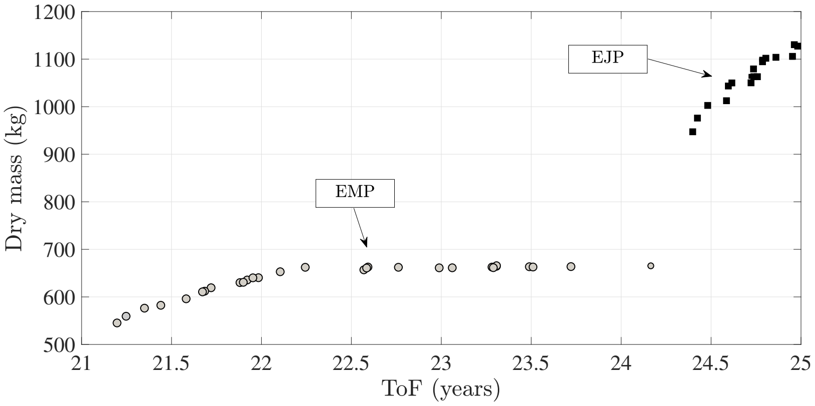


Figure 5.25: MOLTO-IT Step 1: E-P Pareto front solution with propulsion constraint

MOLTO-IT Step 1 was run with the GA parameters presented in Table 5.33 and without applying the propulsion constraint. The Pareto front after 1000 generations is displayed in Figure 5.24. Two mission profiles have been found to be optimal: trajectories with one flyby on Mars, and trajectories with one flyby on Saturn. Dry masses range from 1300 kg to 1850 kg, while all solutions reached the maximum allowed value of 3500 kg for the launch mass. However, when this trajectories were used as initial guess for feeding MOLTO-IT Step 2, the algorithm did not converge for any case. It can be concluded that the shaped based solutions were infeasible for the real mission scenario. Therefore, a new simulation was run with MOLTO-IT Step 1 but incorporating the propulsion constraint for each leg. In Figure 5.25 the solutions obtained after 10 min are shown. Now, the optimal sequences comprise trajectories to Pluto via Mars, and via Jupiter. Dry masses range from 550 kg to 1135 kg, while launch masses range from 1700 kg to 2100 kg. Notably, the use the propulsion constraint has oriented the GA to different optimal sequences, as well as the obtained performances have significantly changed.

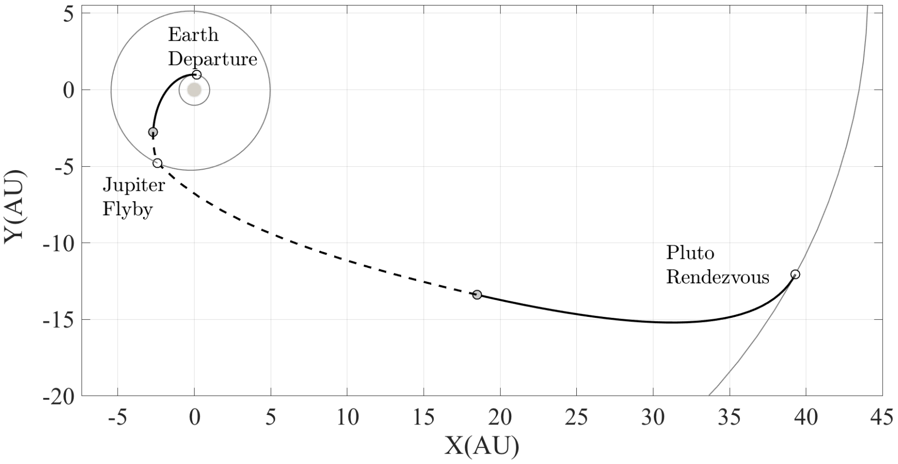


FIGURE 5.26: MOLTO-IT Step 2: E-J-P Minimum fuel trajectory planar view

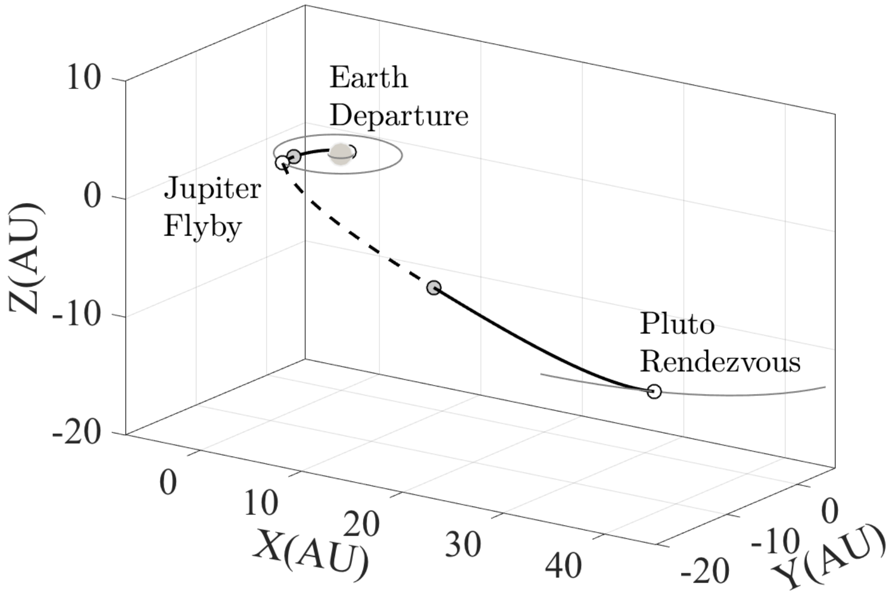


FIGURE 5.27: MOLTO-IT Step 2: E-J-P Minimum fuel trajectory 3D view

TABLE 5.34: E-J-P Maximum delivered mass detailed solution

		MOLTO-IT		Englander and Conway [47]
Parameter	Units	Step 1	Step 2	EMTG
Launch Date	-	Dec 13, 2028	Dec 12, 2028	Dec 15, 2028
Launch v_∞	km/s	8.4	9.15	8.77
DLA	deg	-	1.20	4.5
Initial mass	kg	1923	1734	1870
Jupiter Flyby Date	-	Aug 28, 2030	Sep 30,2030	Oct 6, 2030
Jupiter Flyby v_∞	km/s	957022	857022	788039
Jupiter Flyby Altitude	km	8.567	9.285	6.533
Jupiter B-Plane angle	deg	0	-22.78	4.3
Mass	kg	1798	1674	1703
Arrival Date	-	Dec 03,2053	Dec 06,2053	Dec 13, 2053
Arrival v_∞	km/s	0	0	0
Delivered Mass	kg	1187	1061	1064
Dry Mass	kg	1135	994	984

The maximum delivered mass solution was used as initial guess for MOLTO-IT Step 2. In this case 100 uniformly distributed nodes along time were used for the discretization of each interplanetary leg. This value is higher than in the previous examples because the uniform discretization with 50 nodes did not provide enough resolution close to Pluto, since most nodes are gathered close to Jupiter. A non-uniform grid with respect to time would have produced more efficient results. The trajectory converged after 13 min, 120 NLP-iterations with a NLP-tolerance of 10^{-6}. The geometry is displayed in Fig. 5.26 and in Fig. 5.27. It is noted that two optimal thrusting arcs are found: one after leaving Earth and lasting for 1.5 years to adjust the trajectory for the flyby with Jupiter, and the other one starting 15 years before arriving to Pluto. Fig. 5.28 shows the time history of power and propulsion parameters over the course of the mission. Notably, the optimizer chooses the I_{sp} to be close to the lower bound of 1000 s at the beginning and end of the mission, but prefers a higher I_{sp} and therefore a lower thrust for the end of the first thrust arc and at the beginning of the second one.

In Tab. 5.34 a detailed description of the solutions from MOLTO-IT are shown along with the EJP trajectory found in [192]. Note the ability of MOLTO-IT Step 1 to find the optimal launch date within the ten years launch window, as well as to estimate the launch mass with a 10% error. The dry mass at Pluto is predicted with less than 15% error. Notably, without the propulsion constraints the launch mass predicted by MOLTO-IT Step 1 was 3500 kg, which implies a 252% error. Then, looking at the solutions from MOLTO-IT Step 2 and from EMTG it is found that our trajectory delivers 1% more dry mass. This difference may be due to the fact that EMTG used a different set of parameters for the launcher's performances. The author did not have access to those data and searched for a different, yet reliable source. Remarkably, the EMTG solution was found after 67 hours. They claimed that the optimal solution was identified in less than half of that time. Instead, our solution was found after 10 min of running MOLTO-IT Step 1 and 13 min of running MOLTO-IT Step 2. Therefore, assuming 33 hours for EMTG, our methodology found the same solution 86 times faster.

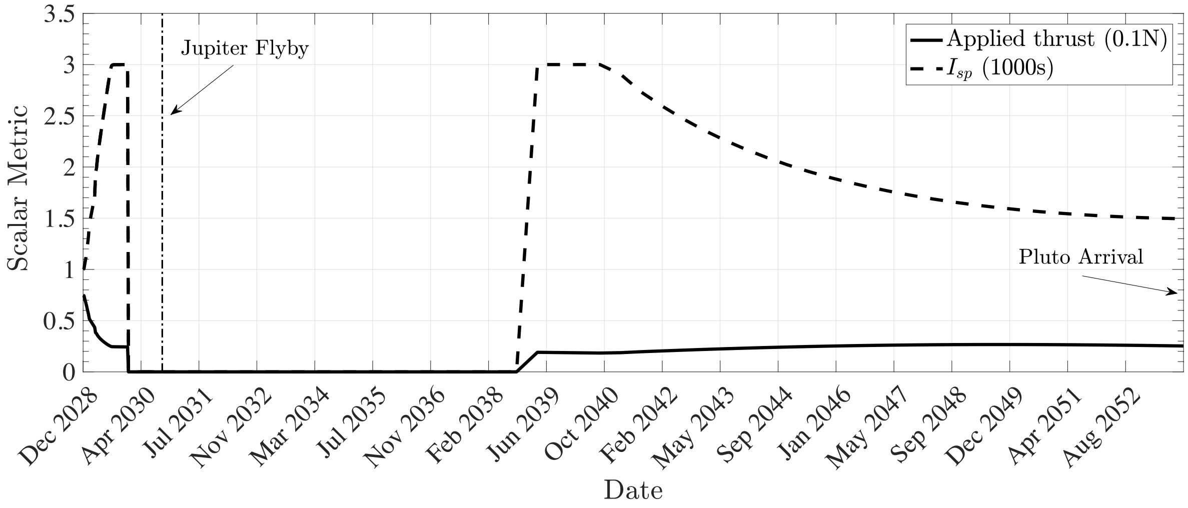


FIGURE 5.28: MOLTO-IT Step 2: Time history of propulsion for E-J-P Trajectory

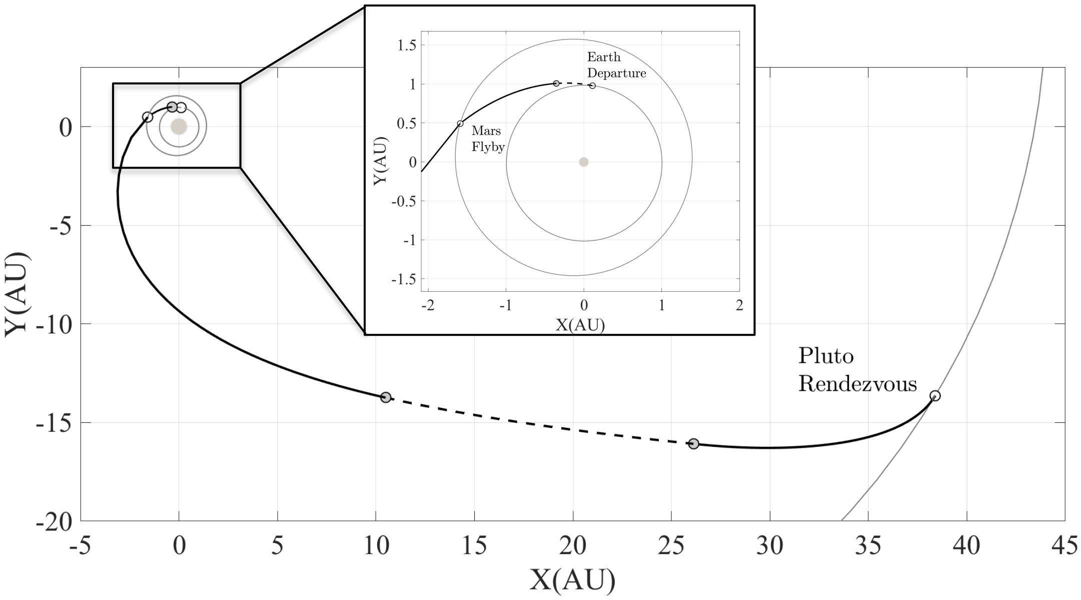


FIGURE 5.29: E-M-P Minimum fuel trajectory ecliptic projection

TABLE 5.35: E-M-P Maximum delivered mass detailed solution

		MOLTO-IT	
Parameter	Units	Step 1	Step 2
Launch Date	-	Dec 5, 2026	Dec 15, 2026
Launch v_∞	km/s	8.78	9.23
DLA	deg	-	27.56
Initial mass	kg	1807	1700
Mars Flyby Date	-	Feb 27, 2027	Mar 19, 2027
Mars Flyby v_∞	km/s	13.23	15.92
Mars Flyby Altitude	km	200	200
Mars B-Plane angle	deg	0	0
Mass	kg	1750	1680
Arrival Date	-	Sep 01, 2051	Dec 09, 2051
Arrival v_∞	km/s	0	0
Delivered Mass	kg	765	514
Dry Mass	kg	650	395

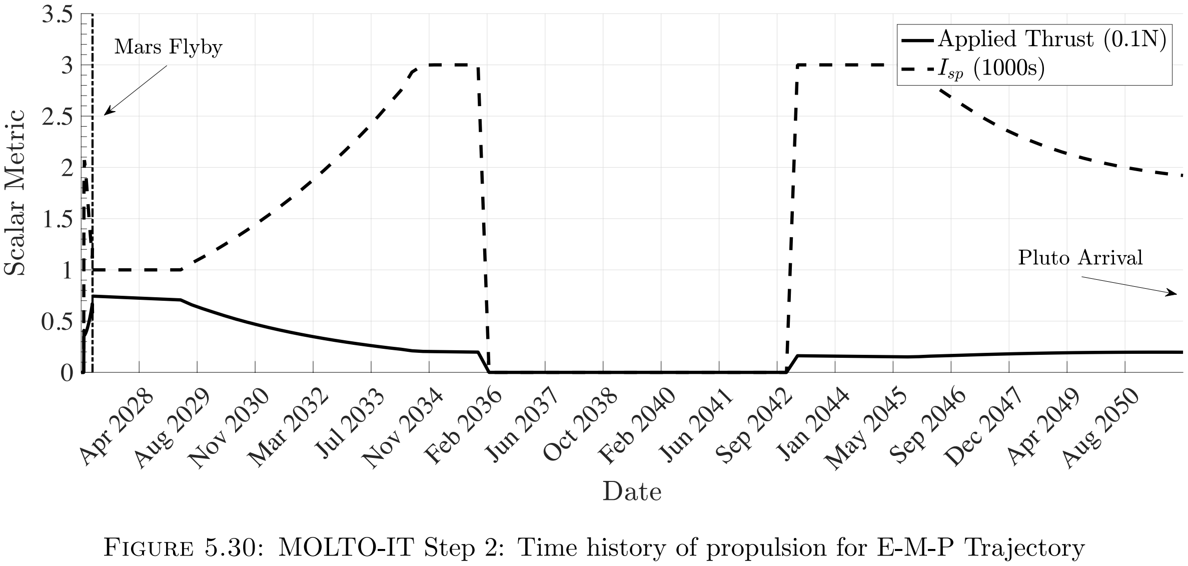


FIGURE 5.30: MOLTO-IT Step 2: Time history of propulsion for E-M-P Trajectory

Finally, the maximum dry mass trajectory obtained for the EMP sequence is re-optimized in MOLTO-IT Step 2. The resulting trajectory is shown in Figure 5.29. In this case, 50 nodes were used for the discretization of the EM leg, and 200 nodes for the MP leg. The trajectory converged after 23 min and 125 NLP-iterations with a NLP-tolerance of 10^{-6}. It is noted that two optimal thrusting arcs are found. The first one starts two months after leaving Earth and lasts 8 years after the Mar Flyby. The second thrust arc starts 10 years before arriving to Pluto. Fig. 5.30 shows the time history of power and propulsion parameters over the course of the mission. In a similar fashion than in the previous example, the optimizer chooses lower values for I_{sp} at the beginning and end of the mission, but prefers a higher I_{sp} and therefore a lower thrust for the end of the first thrust arc and for the beginning of the second one. In Tab. 5.35 a detailed description of the solutions from MOLTO-IT are shown. Notably, the estimate launch mass by the shape-based method differs 6% from the optimal one, whereas the dry mass differs 40%. However, the main events days are better predicted. In this case, Jacob and Conway [192] did not provided detailed results for this sequence, but reported to have obtained a EMP trajectory with a dry mass of 329 kg. Therefore, our solution delivers 16% more dry mass.

6

Conclusions and Future Work

Contents

6.1 Conclusions

In this dissertation, two novel multi-objective optimization algorithms for designing low-thrust space missions have been presented and tested. The main goals of this thesis were threefold: 1) increase the efficiency of searching wide design spaces, 2) reduce the amount of necessary human involvement in the process, and 3) enhance the capabilities to automatically include complex operational constraints in the optimization solution. For such purpose, an in-depth literature review of available methodologies and software tools for tackling the problem of optimizing low-thrust trajectories has been presented in chapter 2. It has been noted that current techniques could be improved towards more flexible and rapid tools, able to not only optimize the control-history of the low-thrust engine under complex operational restrictions, yet also to optimize decision-making processes and mission planning in an automatic manner and with respect to multiple conflicting criteria. Such enhanced capabilities would allow to design more ambitious and cost-effective space missions.

In light of the previous gaps, three main contributions have been identified in this thesis. Firstly, a general mathematical framework based on multi-objective Hybrid Optimal Control have been introduced in chapter 3. Secondly, a numerical solution approach called MOLTO-OR (Multi-Objective Low-Thrust Trajectory Optimizer for Orbit Raising) able to concurrently optimize fully electric, fully chemical and combined chemical-electric transfers between Earth-centered orbits has been presented in chapter 4. It is based on a sequential two-step procedure, which consists on MOLTO-OR Step 1 and MOLTO-OR Step 2. Thirdly, a numerical solution approach termed MOLTO-IT (Multi-Objective Low-Thrust Trajectory Optimizer for Interplanetary Trajectories) able to automatically optimize the number and sequence of gravity assists for low-thrust interplanetary transfers has been introduced in chapter 5. It is based on a sequential two-step procedure, which consists on MOLTO-IT Step 1 and MOLTO-IT Step 2. From each contribution a set of conclusions have been derived:

- The multi-objective Hybrid Optimal Control Problem is a suitable framework for modeling low-thrust trajectory optimization problems including design-making processes or mission planning. In this thesis, the selection of the optimal number and sequence of gravity assists maneuvers, chemical engine firings, and electric engine on-off switchings, as well as the selection of the optimal transfer type (fully electric, fully chemical or combined chemical-electric) and propulsion system have been included as discrete events in the formulation of the problem. Additionally, the determination of the trajectory and the optimal steering law of the electric engine have been incorporated in the model as continuous dynamics. Therefore, the proposed model increases the flexibility of mathematically formulating low-thrust trajectory optimization problems and widens the design space to address different missions profiles within a single framework.

- Regarding MOLTO-OR algorithm:

  * The predefined Q-law steering law implemented in MOLTO-OR Step 1 was proven to provide more efficient GTO-GEO fully electric transfers when compared to other approximate methods available in the literature in terms of propellant mass consumed and time-of-flght. The Q-law approximated the optimal number of revolutions obtained with MOLTO-OR Step 2 with a maximum error of 11 orbits and a maximum penalty of 0.3% in propellant mass. The benchmark methods lead to a maximum error of 22 revolutions with respect to the optimal solution, and a maximum penalty of 2% in propellant mass.
  * MOLTO-OR Step 2 successfully imposed operational restrictions to avoid the GEO ring, to arrive at a certain slot, and to limit the spacecraft slew rate in fully electric transfers. Results were obtained for a GTO-GEO transfer. Operationally constrained trajectories were obtained with computational times ranging from 4.67 min to 18.95 min, while unconstrained solutions were found from 1.98 min to 10.74 min. Notably, constrained solutions were able to avoid 104 crossings of the GEO ring, while matching a final Earth's longitude of 90 deg, and reducing a 75% the maximum slew rate velocity. Obtained penalties with respect to the unconstrained case were below 6.37% in terms of propellant mass.
  * MOLTO-OR Step 1 was tested on a LEO-GEO transfer to simultaneously optimize fully electric, fully chemical and combined chemical-electric transfers with respect to propellant mass, time of flight and final power ratio. The algorithm was run during 16.8 min to obtain a well-spread Pareto front. Results show that minimizing the radiation damage requires more propellant mass dedicated for the chemical phase than for minimizing the propellant mass. Obtained trajectories were used to feed MOLTO-OR Step 2 for solving a series of fixed-time min-fuel problems. The average gain in terms of fuel mass by re-optimization with MOLTO-OR Step 2 was 3%. The computational times for this step range from a couple of minutes for chemical transfers to one hour for the electric transfer.
  * MOLTO-OR Step 1 solved the previous GTO-GEO and LEO-GEO scenarios. In this case, the algorithm was also able to select the optimal propulsive system among a user-provided list of available options, consisting on two electric engines and one chemical thruster. Well spread Pareto front solutions were obtained in terms of flight time, delivered mass and final power ratio after 17 min for the LEO-GEO case and after 9 min for the GTO-GEO case. Different strategies were found for the hybrid transfers under considerations in this thesis. Low-thrust-high-efficient electric engines are preferred to minimize the radiation absorbed when the transfer times are short (i.e., the magnitude of the chemical firings is high). On the contrary, high-thrust-low-efficient electric engines are better for minimizing propellant usage.

- Regarding MOLTO-IT algorithm:

  * MOLTO-IT Step 1 was able to automatically find direct and via Mars flyby transfers to rendezvous with Ceres for a one year launch window. A well-spread Pareto front in terms of flight time and propellant mass consumed were obtained after 3.2 min. The most propellant optimal solution found with MOLTO-IT Step 1 exhibits a 12% lower propellant mass fraction than STOUR-LGTA (Satellite Tour Design Program Low-Thrust Gravity-Assist) for a 143 days shorter transfer mission. For the same scenario, MOLTO-IT Step 2 reduces 1.7% the propellant mass compared to GALLOP (Gravity Assisted Low-Thrust Local Optimization Program) for a 20 days shorter mission.

  * MOLTO-IT Step 1 successfully solved a flyby mission to Jupiter the algorithm, automatically finding trajectories with up to three flybys, including Venus, Earth and Mars gravity assists for a two year launch window. A well-spread Pareto fronts in terms of flight time and propellant mass consumed were obtained after 4 min. The most propellant optimal solution from MOLTO-IT Step 1 exhibits a 55% lower propellant mass than STOUR-LGTA for a 552 days shorter transfer mission. For the same scenario, MOLTO-IT Step 2 reduces 65% the propellant mass of GALLOP.

  * MOLTO-IT Step 2 capability to impose fixed reorientation times constraint was tested on an Earth-Venus-Earth-Jupiter trajectory. It has been noted that the longest maneuvers are found when the angular velocity of the free solution was lower. Additionally, it was found that the required propellant mass increased as the reorientation time increased. Notably, the mission can be accomplished with only three reorientations carrying 1.31% of additional propellant mass with respect to the free case.

  * MOLTO-IT Step 1 feature to account for thrust constraints was demonstrated on a rendezvous mission to Pluto where the initial spacecraft mass is free. A well spread Pareto front with respect to time of flight and delivered mass was obtained after 10 min. The optimal launch date for an Earth-Jupiter-Pluto mission was found within the ten years launch window, and the launch mass was estimated with a 10% error. The dry mass at Pluto is predicted with less than 15% error. Notably, without the propulsion constraints the launch mass predicted by MOLTO-IT Step 1 exhibited a 252% error. The solution from MOLTO-IT Step 2 were compared with those obtained with the tool EMTG (Evolutionary Mission Trajectory Generator). It delivers 1% more dry mass and was obtained 86 times faster in terms of computing time.

Overall, we can conclude that the solution approaches studied in this dissertation have a strong potentiality in assisting mission designers toward concurrent engineering designs, due to their proven features to provide fast results with minimal user interaction, to allow for deep insights of the trades between different mission concepts, and to include complex operational constraints in the optimization process.

6.2 Future Work

The work carried out within this thesis opens up numerous avenues of research. In this section, we will proceed to explore the most prominent ones:

- The capability of optimizing the propulsion subsystem in MOLTO-OR Step 1 could be extended to include other subsystems such as power (e.g., solar array size, material), attitude (e.g. maximum slew rate velocity) or the selection of the optimal launcher (e.g., initial mass, initial orbit). In case the parameters are to be optimized within a given continuous interval, they can be treated as a set of continuous variables by the genetic algorithm. In case they are to be selected among a user-provided list available options, they would be model as a set of integer variables to be optimized by the heuristic algorithm. Such methodology would provide mission designers in one run with a set of complete spacecraft designs and transfer trajectories that are optimal with respect to various criteria.

- Increase the number of operational constraints and the accuracy of dynamical model implemented in MOLTO-OR Step 2. For instance, including thrust pointing constraints would allow the spacecraft to comply with constraints deriving from the need of having the solar panels facing the Sun, or avoiding the blinding of the star-trackers if necessary. Additionally, limiting the time spent in eclipse may allow the spacecraft to comply with power subsystem constraints. These restrictions can be incorporated into MOLTO-OR Step 2 without changing the previous formulation. Such results would allow to avoid the need to perform non-optimal maneuvers during the real mission, thus optimizing the propellant usage during the mission and enhancing its success.

- Improve the features of MOLTO-IT Step 1 to compute interplanetary transfers by including additional thrust-coast arcs. This configuration would be suitable for optimizing multi-revolution transfers, such as those typically required to visit the planets of the inner solar system. Additionally, an heuristic estimation of the required out-of-plane thrust could be included in the model to improve the quality of the performance calculations. Furthermore, allowing the optimizer to select the optimal arrival body would permit to tackle the problem of exploring and mining asteroids where the optimal target is free, yet subject to mission requirements and constraints. The target body may be modeled as an integer variable and optimized by the genetic algorithm.

- Finally, the algorithm MOLTO could be extended to solve other important problems is astrodynamics, such as low-thrust transfers within the three-body dynamics following a similar approach consisting on two sequential step. In the first step, and heuristic algorithm along with simplified dynamical models would allow to obtain quick performance estimates. In the second step, candidate trajectories would be re-optimized using more accurate models. A gradient-based solver would be in charge of optimizing the NLP resulting from the direct collocation of the problem.

Bibliography

[1] S Mazouffre. Electric propulsion for satellites and spacecraft: Established technologies and novel approaches. *Plasma Sources Science and Technology*, 25:033002, 06 2016. doi: 10.1088/0963-0252/25/3/033002.

[2] M. Martinez-Sanchez and J. E. Pollard. Spacecraft electric propulsion-an overview. *Journal of Propulsion and Power*, 14(5):688–699, 2019/02/18 1998. doi: 10.2514/2.5331.

[3] M. D. Rayman, P. Varghese, D. H. Lehman, and L. L. Livesay. Acta astronautica. *Results from the Deep Space 1 Technology Validation Mission*, 47:475–487, 2000.

[4] J. Kawaguchi, A. Fujiwara, and T. K. Uesugi. The ion engines cruise operation and the earth swingby of 'hayabusa' (muses-c). In *55th International Astronautical Congress of the International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law*. American Institute of Aeronautics and Astronautics, 2019/01/09 2004. doi: 10.2514/6.IAC-04-Q.5.02.

[5] G. Racca, A. Marini, L. Stagnaro, and et al. Smart-1 mission description and development status. *Planetary and Space Science*, 50:1323–1337, 2002.

[6] M. D. Rayman, T. C. Frashetti, C. A. Raymond, and C. T. Russel. Dawn: A mission in development for exploration of main belt asteroids vesta and ceres. *Acta Astronautica*, 58:605–616, 2006.

[7] Sei-ichiro Watanabe, Yuichi Tsuda, Makoto Yoshikawa, Satoshi Tanaka, Takanao Saiki, and Satoru Nakazawa. Hayabusa2 mission overview. *Space Science Reviews*, 208(1):3–16, Jul 2017. ISSN 1572-9672. doi: 10.1007/s11214-017-0377-1.

[8] Rüdiger Jehn, Daniel García Yárnoz, Johannes Schoenmaekers, and Vicente Companys. Trajectory design for bepicolombo based on navigation requirements. *Journal of Aerospace Engineering, Sciences and Applications*, 4:1–9, 01 2012. doi: 10.7446/jaesa.0401.01.

[9] Donna L. Shirley. The mariner 10 mission to venus and mercury. *Acta Astronautica*, 53: 375–385, 08 2003. doi: 10.1016/S0094-5765(03)00155-3.

[10] C E. Kohlhase and P A. Penzo. Voyager mission description. *Space Science Reviews*, 21, 12 1977. doi: 10.1007/BF00200846.

[11] Brian Kantsiper. The double asteroid redirection test (dart) mission electric propulsion trade. In *IEEE Aerospace Conference*, pages 1–7, 03 2017. doi: 10.1109/AERO.2017.7943736.

[12] Linda Elkins-Tanton. Asteroid 16 psyche: Nasa's 14th discovery mission. *Elements*, 14: 68–68, 02 2018. doi: 10.2138/gselements.14.1.68.

[13] Martina Mammarella, Nicole Viola, Christopher Paissoni, Pietro Maria Vernicari, Giorgio Saccoccia, and Jose Gonzalez del Amo. Electric propulsion for high-power deep space transportation system: investigation on mutual influences and preliminary sizing. 09 2017.

[14] Dan Goebel, Manuel Martinez-Lavin, Thomas Bond, and Andrew King. Performance of xips electric propulsion in on-orbit station keeping of the boeing 702 spacecraft. 07 2002. ISBN 978-1-62410-115-1. doi: 10.2514/6.2002-4348.

[15] J. Fife, Jr. W. Hargus, L. Mason, R. Jankovsky, T. Haag, L. Pinero, and J. Snyder. Preliminary performance results of the high performance hall system spt-140. In *36th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, July 2000. doi: 10.2514/6.2000-3250.

[16] Olivier Duchemin, Pierre Dumazert, Sébastien Carichon, Claude Boniface, L Guarrigues, and jean-pierre Boeuf. Performance and lifetime predictions by testing and modeling for the pps-5000 hall thruster. In *39th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, 07 2003. ISBN 978-1-62410-098-7. doi: 10.2514/6.2003-4555.

[17] J.-J Chen, T.-P Zhang, Y.-H Jia, and H Geng. Numerical study on lifetime of lips-300 ion thruster ion optics. *Guti Huojian Jishu/Journal of Solid Rocket Technology*, 39:44–49, 02 2016. doi: 10.7673/j.issn.1006-2793.2016.01.008.

[18] P Rathsman, A Demairé, E Rezugina, H Lübberstedt, and M De Tata. Electra - the implementation of all-electric propulsion on a geostationary satellite. *Proceedings of the International Astronautical Congress, IAC*, 7:5333–5343, 01 2013.

[19] R. Bennett, P. Misra, and W. Upson. Gstar iii mission recovery progress following partial orbit injection event of september 11, 1988. In *Guidance, Navigation and Control Conference*, 1989. doi: 10.2514/6.1989-3638.

[20] Rainer Killinger, Ralf Kukies, Michael Surauer, Angeo Tomasetto, and Leo van Holtz. Artemis orbit raising inflight experience with ion propulsion. *Acta Astronautica*, 53(4): 607 – 621, 2003. ISSN 0094-5765. doi: 10.1016/S0094-5765(03)80022-X. The New Face of Space Selected Proceedings of the 53rd International Astronautical Federation Congress.

[21] Dan Lev, Roger Myers, Kristina Lemmer, Jonathan Kolbeck, Michael Keidar, Hiroyuki Koizumi, Han Liang, Daren Yu, Tony Schönherr, Jose Gonzalez, Wonho Choe, Riccardo

Albertoni, Andrew Hoskins, Shen Yan, William Hart, Richard R Hofer, Ikkoh Funaki, Alexander Lovtsov, Kurt Polzin, and Olivier Duchemin. The technological and commercial expansion of electric propulsion in the past 24 years. In *International Electric Propulsion Conference*, Atlanta, USA, 10 2017.

[22] R Haagmans, D Muzi, A Popescu, Rune Floberghagen, Michael Kern, and M Fehringer. The goce gravity mission: Esa's first core explorer. *Proceedings of 3rd International GOCE User Workshop*, 01 2007.

[23] Osamu Mori, Yoji Shirasawa, Yuya Mimasu, Yuichi Tsuda, Hirotaka Sawada, Takanao Saiki, Takayuki Yamamoto, Katsuhide Yonekura, Hirokazu Hoshino, Junichiro Kawaguchi, and Ryu Funase. *Overview of IKAROS Mission*, pages 25–43. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. doi: 10.1007/978-3-642-34907-2_3.

[24] Vladimir S. Aslanov and Alexander S. Ledkov. 1 - space tether systems: review of the problem. In Vladimir S. Aslanov and Alexander S. Ledkov, editors, *Dynamics of Tethered Satellite Systems*, pages 3 – 104. Woodhead Publishing, 2012. ISBN 978-0-85709-156-7. doi: 10.1533/9780857096005.3.

[25] Michiel Kruijff, Patrick Hambloch, Erik Heide, and Marco Stelzer. The second young engineers' satellite (yes2). In *58th International Astronautical Congres*, volume 10, Hyderabad, India, 09 2007.

[26] Massimiliano Vasile, Stephen Kemble, A Santovincenzo, and M Taylor. *Mission and system design*, pages 685–713. 01 2014. doi: 10.1007/978-3-642-41101-4_25.

[27] National Aeronautics and Space Administration. *NASA Systems Engineering Handbook*. U.S. Government Printing Office, 2008. ISBN 9780160797477.

[28] Stephen D Wall. Use of concurrent engineering in space mission design. *Proceedings of EuSEC*, 2000.

[29] Abolfazl Shirazi, Josu Ceberio, and Jose Lozano. Spacecraft trajectory optimization: A review of models, objectives, approaches and solutions. *Progress in Aerospace Sciences*, 08 2018. doi: 10.1016/j.paerosci.2018.07.007.

[30] Bruce A Conway. *Spacecraft trajectory optimization*, volume 29. Cambridge University Press, New York, 2010.

[31] Derek F. Lawden. Impulsive transfer between elliptical orbits. In George Leitmann, editor, *Optimization Techniques*, volume 5 of *Mathematics in Science and Engineering*, pages 323 – 351. Elsevier, 1962. doi: 10.1016/S0076-5392(08)62099-8.

[32] Nguyen X. Vinh, Shau Hern Kuo, and Christian Marchal. Optimal time-free nodal transfers between elliptical orbits. *Acta Astronautica*, 17(8):875 – 880, 1988. ISSN 0094-5765. doi: 10.1016/0094-5765(88)90170-1.

[33] J. E. Prussing and J. H. Chiu. Optimal multiple-impulse time-fixed rendezvous between circular orbits. *Journal of Guidance, Control, and Dynamics*, 9(1):17–22, 2018/10/10 1986. doi: 10.2514/3.20060.

[34] Donald J Jezewski and Harvey L Rozendaal. An efficient method for calculating optimal free-space n-impulse trajectories. *AIAA Journal*, 6(11):2160–2165, 2018/10/10 1968. doi: 10.2514/3.4949.

[35] Ossama Abdelkhalik and Daniele Mortari. N-impulse orbit transfer using genetic algorithms. *Journal of Spacecraft and Rockets*, 44(2):456–460, 2018/09/25 2007. doi: 10.2514/1.24701.

[36] Anastassios E. Petropoulos and Jon A. Sims. A review of some exact solutions to the planar equations of motion of a thrusting spacecraft. In *2nd International Symposium Low-Thrust Trajectories*, Toulouse, France, 01 2002.

[37] Nasa technology roadmaps. Technical report, NASA, July 2015.

[38] Christian M. Chilan and Bruce A. Conway. Automated design of multiphase space missions using hybrid optimal control. *Journal of Guidance, Control, and Dynamics*, 36(5):1410–1424, 2018/08/30 2013. doi: 10.2514/1.58766.

[39] I. Michael Ross and Christopher N. D'Souza. Hybrid optimal control framework for mission planning. *Journal of Guidance, Control, and Dynamics*, 28(4):686–697, 2018/08/30 2005. doi: 10.2514/1.8285.

[40] M. Vasile, F. Bernelli-Zazzera, N. Fornasari, and P. Masarati. Design of interplanetary and lunar missions combining low thrust and gravity assists. *Final report of esa/esoc study contract no. 14126/00/d/cs, ESA/ESOC*, 2001.

[41] S. Campagnola, N. Ozaki, Y. Sugimoto, C. H. Yam, C. Hongru, Y. Kawabata, S. Ogura, B. Sarli, Y. Kawakatsu, R. Funase, and S Nakasuka. Low-thrust trajectory design and operations of procyon, the first deep-space micro-spacecraf. In *International Symposium on Space Flight Dynamics*, Munich, Germany, 2015.

[42] Gregory Whiffen. Mystic: Implementation of the static dynamic optimal control algorithm for high-fidelity, low-thrust trajectory design. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*. American Institute of Aeronautics and Astronautics, 2017/09/26 2006. doi: 10.2514/6.2006-6741.

[43] J. T. Olympio. *Optimisation and Optimal Control Methods for Planet Sequence Design of Low-Thrust Interplanetary Transfer Problems with Gravity Assists*. PhD thesis, Ecole des Mines de Paris, October 2008.

[44] Elijah Polak. *Computational methods in optimization: a unified approach*, volume 77. Academic press, 1971.

[45] Seungwon Lee, Paul Von Allmen, Wolfgang Fink, Anastassios E. Petropoulos, and Richard J. Terrile. Design and optimization of low-thrust orbit transfers. *IEEE Aerospace Conference Proceedings*, 01 2005.

[46] Jacob A. Englander, Matthew Vavrina, and A R Ghosh. Multi-objective hybrid optimal control for multiple-flyby low-thrust mission design. In *AAS/AIAA Space Flight Mechanics Meeting*, Williamsburg, VA, 01 2015.

[47] A. E. Petropoulos and J. M. Longuski. Shape-based algorithm for automated design of low-thrust, gravity-assist trajectories. *Journal of Spacecraft and Rockets*, 41(5):787–796, 2004. doi: 10.2514/1.13095.

[48] Bradley J. Wall and Bruce A. Conway. Shape-based approach to low-thrust rendezvous trajectory design. *Journal of Guidance, Control, and Dynamics*, 32(1):95–101, 2018/04/09 2009. doi: 10.2514/1.36848.

[49] Bradley Wall. Shape-based approximation method for low-thrust trajectory optimization. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 08 2008. ISBN 978-1-62410-001-7. doi: 10.2514/6.2008-6616.

[50] P. De Pascale and M. Vasile. Preliminary design of low-thrust multiple gravity-assist trajectories. *Journal of Spacecraft and Rockets*, 43(5):1065–1076, 2017/10/02 2006. doi: 10.2514/1.19646.

[51] D. M. Novak and M. Vasile. Improved shaping approach to the preliminary design of low-thrust trajectories. *Journal of Guidance, Control, and Dynamics*, 34(1):128–147, 2018/04/09 2011. doi: 10.2514/1.50434.

[52] Ehsan Taheri and Ossama Abdelkhalik. Initial three-dimensional low-thrust trajectory design. *Advances in Space Research*, 57(3):889 – 903, 2016. ISSN 0273-1177. doi: 10.1016/j.asr.2015.11.034.

[53] David Gondelach and Ron Noomen. Analytical low-thrust transfer design based on velocity hodograph. In *AAS/AIAA Astrodynamics Specialist Conference*, Vail, Colorado, 08 2015.

[54] Dongsuk Han. Orbit transfers for dawn's vesta operations: Navigation and mission design experience. In National Aeronautics Jet Propulsion Laboratory and Space Administration, editors, *23rd International Symposium on Space Flight Dynamics*, Pasadena, California, October 2012.

[55] John T. Betts. Very low-thrust trajectory optimization using a direct sqp method. *Journal of Computational and Applied Mathematics*, 120:27–40, AUgust 2000. doi: 10.1016/S0377-0427(00)00301-0.

[56] Sven Schäff. *Low-Thrust Multi-Revolution Orbit Transfers*, pages 337–367. Springer Optimization and Its Applications, 01 2016. ISBN 978-3-319-41506-2. doi: 10.1007/978-3-319-41508-6_13.

[57] Slim Locoche. Optelec: an optimisation software for low-thrust orbit transfer including satellite and operation constraints. In *7th International Conference on Astrodynamics Tools and Techniques*, Munich, Germany, 10 2018.

[58] R. H. Bacon. Logarithmic spiral: an ideal trajectory for the interplanetary vehicle with engines of low sustained thrust. *American Journal of Physics*, 27(3):164–165, 1959. doi: 10.1119/1.1934788.

[59] A. E. Petropoulos and James M. Longuski. Shape-based algorithm for the automated design of low-thrust, gravity assist trajectories. *Journal of Spacecraft and Rockets*, 41(5): 787–796, 2018/10/10 2004. doi: 10.2514/1.13095.

[60] Theodore N Edelbaum. Optimum low-thrust rendezvous and station keeping. *AIAA Journal*, 2(7):1196–1201, 2018/10/10 1964. doi: 10.2514/3.2521.

[61] Jean A. Kéchichian. Orbit raising with low-thrust tangential acceleration in presence of Earth shadow. *Journal of Spacecraft and Rockets*, 35(4):516–525, 2018/10/10 1998. doi: 10.2514/2.3361.

[62] A. E. Petropoulos. Some analytic integrals of the averaged variational equations for a thrusting spacecraft. *The Interplanetary Network Progress Report*, 150:1–29, April 2002.

[63] Craig A. Kluever and Steven R. Oleson. Direct approach for computing near-optimal low-thrust earth-orbit transfers. *Journal of Spacecraft and Rockets*, 35(4):509–515, 2018/10/10 1998. doi: 10.2514/2.3360.

[64] Yang Gao. Near-optimal very low-thrust earth-orbit transfers and guidance schemes. *Journal of Guidance, Control, and Dynamics*, 30(2):529–539, 2018/07/17 2007. doi: 10.2514/1.24836.

[65] Jennifer S. Hudson and Daniel J. Scheeres. Reduction of low-thrust continuous controls for trajectory dynamics. *Journal of guidance, control, and dynamics*, 32(3), May-June 2009.

[66] Zahi Bassem Tarzi. *Optimum Low Thrust Elliptic Orbit Transfer Using Numerical Averaging.* PhD thesis, University of California, Los Angeles, 01 2012.

[67] J Kevorkian and JD Cole. *Perturbation methods in applied mathematics*, volume 7. Springer-Verlag, New York, 1982. doi: 10.1007/978-1-4757-4213-8.

[68] Juan Luis Gonzalo and Claudio Bombardelli. Asymptotic solution for the two body problem with radial perturbing acceleration. *Advances in the Astronautical Sciences*, 152: 359–377, 01 2014.

[69] Marc R Ilgen. Low thrust otv guidance using lyapunov optimal feedback control techniques. In *Advances in the Astronautical Sciences*, volume 85, pages 1527–1545, Victoria, Canada, 1993. The Society.

[70] Anastassios E. Petropoulos. *Low-Thrust Orbit Transfers Using Candidate Lyapunov Functions with a Mechanism for Coasting*. American Institute of Aeronautics and Astronautics, Providence, Rhode Island, 2018/10/10 2004. doi: 10.2514/6.2004-5089.

[71] Sonia Hernandez and Maruthi R. Akella. Lyapunov-based guidance for orbit transfers and rendezvous in levi-civita coordinates. *Journal of Guidance, Control, and Dynamics*, 37(4):1170–1181, 2014. doi: 10.2514/1.62305.

[72] Yang Dalin, Xu Bo, and Gao Youtao. Optimal strategy for low-thrust spiral trajectories using lyapunov-based guidance. *Advances in Space Research*, 56, 05 2015. doi: 10.1016/j.asr.2015.05.030.

[73] Seungwon Lee, Anastassios E. Petropoulos, and Paul von Allmen. Low-thrust orbit transfer optimization with refined q-law and multi-objective genetic algorithm. In *Advances in the Astronautical Sciences*, volume 123, 2006.

[74] Gábor I. Varga and José M. Sánchez Pérez. Many-revolution low-thrust orbit transfer computation using equinoctial q-law including j2 and eclipse effects. In *6th International Conference on Astrodynamics Tools and Techniques*, Darmstatd,Germany, April 2016.

[75] Georges Krier and David Mostaza. Fast and robust optimization of high fidelity continuous thrust transfer orbits with constraints. In *The 25th International Symposium on Space Flight Dynamics*, Munich, 10 2015.

[76] Arnon Spitzer. Near optimal transfer orbit trajectory using electric propulsion. In *Advances in the Astronautical Sciences*, volume 89, pages 1031–1044, 1995.

[77] Steven R. Oleson, Roger M. Myers, Craig A. Kluever, John P. Riehl, and Francis M. Curran. Advanced propulsion for geostationary orbit insertion and north-south station keeping. *Journal of Spacecraft and Rockets*, 34(1):22–28, 2018/10/10 1997. doi: 10.2514/2.3187.

[78] Laurie M. Mailhe and Stephen D. Heister. Design of a hybrid chemical/electric propulsion orbital transfer vehicle. *Journal of Spacecraft and Rockets*, 39(1):131–139, 2018/10/10 2002. doi: 10.2514/2.3791.

[79] David Y. Oh, Thomas Randolph, Scott Kimbrel, and Manuel Martinez-Sanchez. End-to-end optimization of chemical-electric orbit-raising missions. *Journal of Spacecraft and Rockets*, 41(5):831–839, 2018/10/10 2004. doi: 10.2514/1.13096.

[80] Alan Jenkin. Representative mission trade studies for low-thrust transfers to geosynchronous orbit. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Providence, Rhode Island, 2018/10/10 2004. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2004-5086.

[81] Alan Jenkin. Maximum deliverable mass for low-thrust geosynchronous transfers under realistic mission constraints. In *Advances in the Astronautical Sciences*, volume 123, pages 2265–2284, 2006.

[82] Craig A. Kluever. Optimal geostationary orbit transfers using onboard chemical-electric propulsion. *Journal of Spacecraft and Rockets*, 49(6):1174–1182, 2018/10/10 2012. doi: 10.2514/1.A32213.

[83] Craig A. Kluever. Designing transfers to geostationary orbit using combined chemical–electric propulsion. *Journal of Spacecraft and Rockets*, 52(4):1144–1151, 2018/10/10 2015. doi: 10.2514/1.A33259.

[84] S. Ceccherini, L. Ferella, and F. Topputo. Assesment of hybrid propulsion for geostationary transfer orbits: A mission design approach. *67 International Astronautical Congress*, Guadalajara, Mexico, 2016.

[85] David Morante, Manuel Sanjurjo Rivo, and Manuel Soler. Multi-objective low-thrust interplanetary trajectory optimization based on generalized logarithmic spirals. *Journal of Guidance, Control, and Dynamics*, 42(3):476–490, 2019. doi: 10.2514/1.G003702.

[86] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, Apr 2002. ISSN 1089-778X. doi: 10.1109/4235.996017.

[87] Atri Dutta, N. Jeremy Kasdin, Edgar Choueiri, and Philippe Francken. Minimizing proton displacement damage dose during electric orbit raising of satellites. *Journal of Guidance, Control, and Dynamics*, 39(4):963–969, 2017/08/07 2016. doi: 10.2514/1.G000503.

[88] J. Roa, A. E. Petropoulos, and R. S. Park. Semi-analytic preliminary design of low-thrust missions. In *Astrodynamics Specialist Conference*, Stevenson, WA, August 2017. American Astronautical Society Paper 17-623.

[89] John T. Betts. Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193–207, 2018/04/06 1998. doi: 10.2514/2.4231.

[90] Bruce A. Conway. A survey of methods available for the numerical optimization of continuous dynamic systems. *Journal of Optimization Theory and Applications*, 152(2):271–306, Feb 2012. ISSN 1573-2878. doi: 10.1007/s10957-011-9918-z.

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

[92] Gerald R. Hintz. Survey of orbit element sets. *Journal of Guidance, Control, and Dynamics*, 31(3):785–790, 2008. doi: 10.2514/1.32237.

[93] W.D. McClain and D.A. Vallado. *Fundamentals of Astrodynamics and Applications*. Space Technology Library. Springer Netherlands, 2001. ISBN 9780792369035.

[94] Paul Cefola. Equinoctial orbit elements - application to artificial satellite orbits. In *Astrodynamics Conference*, number AIAA Paper 72-937, 10 1972. doi: 10.2514/6.1972-937.

[95] C. Hillermeier. Generalized homotopy approach to multiobjective optimization. *Journal of Optimization Theory and Applications*, 110(3):557–583, Sep 2001. ISSN 1573-2878. doi: 10.1023/A:1017536311488.

[96] V Pareto. Manuale di economica politica, societa editrice libraria. Technical report, MacMillan Press, New York, 1971.

[97] R Marler and Jasbir Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26:369–395, 04 2004. doi: 10.1007/s00158-003-0368-6.

[98] L.S. Pontryagin. Mathematical theory of optimization processes. 1962.

[99] Arthur E. Bryson, Y.-C Ho, and George M. Siouris. Applied optimal control: Optimization, estimation, and control. *Systems, Man and Cybernetics, IEEE Transactions on*, 9: 366 – 367, 07 1979. doi: 10.1109/TSMC.1979.4310229.

[100] R. G Gottlieb. Rapid convergence to optimum solutions using a min-h strategy. *AIAA Journal*, 5(2):322–329, 1967. doi: 10.2514/3.3960.

[101] A. G. Longmuir and E. V. Bohn. Second-variation methods in dynamic optimization. *J. Optim. Theory Appl.*, 3(3):164–173, 1969. ISSN 0022-3239. doi: 10.1007/BF00929441.

[102] William Karush. Minima of functions of several variables with inequalities as side conditions. Master's thesis, Department of Mathematics, University of Chicago, Chicago, IL, USA, 1939.

[103] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, Berkeley, Calif., 1951. University of California Press.

[104] Richard Bellman and Robert E Kalaba. *Dynamic programming and modern control theory*, volume 81. Citeseer, 1965.

[105] Michael R. Caputo. *Dynamic Programming and the Hamilton-Jacobi-Bellman Equation*, pages 511–536. Cambridge University Press, 2005. doi: 10.1017/CBO9780511806827.021.

[106] Warren B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality (Wiley Series in Probability and Statistics)*. Wiley-Interscience, New York, NY, USA, 2007. ISBN 0470171553.

[107] David Q. Mayne. Differential dynamic programming-a unified approach to the optimization of dynamic systems. In *Control and Dynamic Systems*, volume 10, pages 179 – 254. Academic Press, 1973. doi: 10.1016/B978-0-12-012710-8.50010-8.

[108] Philip R. Gill. *User's Guide for SNOPT Version 7 : Software for Large-Scale Nonlinear Programming*. Technical report, Department of Mathematics, University of California, San Diedo, La Jolaa CA 92093-0112, June 16,2008.

[109] A. Wächter and L. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(25): 25–57, 2006. doi: 10.1007/s10107-004-0559-y.

[110] Melanie Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, USA, 1998. ISBN 0262631857.

[111] Riccardo Poli, James Kennedy, and Tim Blackwell. Particle swarm optimization: An overview. *Swarm Intelligence*, 1, 10 2007. doi: 10.1007/s11721-007-0002-0.

[112] Benjamin Passenberg. *Theory and Algorithms for Indirect Methods in Optimal Control of Hybrid Systems*. PhD thesis, Technische Universitat München, 2012.

[113] Matthias Rungger and Olaf Stursberg. Continuity of the value function for exit time optimal control problems of hybrid systems. pages 4210–4215, 12 2010. doi: 10.1109/CDC.2010.5718054.

[114] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman &amp; Co., New York, NY, USA, 1990. ISBN 0716710455.

[115] Ioannis Androulakis. *MINLP: Branch and Bound Global Optimization Algorithm*. Springer, Boston, MA, 01 2008. doi: 10.1007/978-0-387-74759-0_377.

[116] Jerry Horsewood. Program manual for astop, an arbitrary space trajectory optimization program. Technical Report NASA-CR-120660, NASA, Analytical Mechanics Associates, Inc., 01 1975.

[117] G. Johnson, S. Munoz, and J. Lehman. Copernicus: A generalized trajectory design and optimization system. Technical report, University of Texas, TX, USA, 2003.

[118] Lorenzo Ricciardi and Massimiliano Vasile. Modhoc-multi objective direct hybrid optimal control. 11 2018.

[119] J. Schoenmaekers. Mantra – flight dynamics interplanetary manoeuvre optimisation software specification document. Technical report, ESA/ESOC, Darmstadt, Germany, 2005.

[120] Chen Zhang, Francesco Topputo, Franco Bernelli-Zazzera, and Y.-S Zhao. An exploration of numerical methods for low-thrust trajectory optimization in n-body models. *Proceedings of the International Astronautical Congress, IAC*, 7:4971–4977, 01 2013.

[121] B. A. Jones J. F. C. Herman, J. S. Parker and G. H. Born. High-speed, high-fidelity low-thrust trajectory optimization through parallel computing and collocation method. In *Advances in the Astronautical Sciences Spaceflight Mechanics 2015*, number AAS 15–298, 2015.

[122] Daniel J. Grebow and Thomas A. Pavlak. Mcoll: Monte collocation trajectory design too. In *Advances in the Astronautical Sciences Astrodynamics*, volume 162, 2017.

[123] Robert Pritchett, Kathleen Howell, and Daniel Grebow. Low-thrust transfer design based on collocation techniques: Applications in the restricted three-body problem. In *Advances in the Astronautical Sciences Astrodynamics*, volume 162, 08 2017.

[124] Steven P. Hughes, Rizwan H. Qureshi, Steven D. Cooley, and Joel Parker. Verification and validation of the general mission analysis tool (gmat). 08 2014. ISBN 978-1-62410-308-7. doi: 10.2514/6.2014-4151.

[125] Erick Gallesio and Sophia Antipolis. *STK Reference Manual*, 11 2000.

[126] C. R. Hargraves and S. W. Paris. Direct trajectory optimization using nonlinear programming and collocation. *Journal of Guidance, Control, and Dynamics*, 10(4):338–342, 1987. doi: 10.2514/3.20223.

[127] G. L. Brauer, D. E. Cornick, and R. Stevenson. Capabilities and applications of the program to optimize simulated trajectories (post). program summary document. Technical Report NASA-CR-2770, NASA, Washington, United States, 1977.

[128] I. Ross. *User's Manual for DIDO: A MATLAB Application Package for Solving Optimal control Problems*. TOMLAB Optimization, Sweden, 2004.

[129] M. A. Patterson and A. V. Rao. Gpops–ii: A matlab software for solving multiple-phase optimal control problems using hpadaptive gaussian quadrature collocation methods and sparse nonlinear programming. *ACM Transactions on Mathematical Software*, 41(1):1:1–1:37., October 2014. doi: doi:10.1145/2684421.

[130] Cesar Ocampo. Method of simultaneously reducing inclination and eccentricity for geostationary orbit transfer. Technical Report Patent 6,341,749, Hughes Electronics Corporation, 2002.

[131] T. Troy McConaghy, Theresa J. Debban, Anastassios E. Petropoulos, and James M. Longuski. Design and optimization of low-thrust trajectories with gravity assists. *Journal of Spacecraft and Rockets*, 40(3):380–387, 2018/04/06 2003. doi: 10.2514/2.3973.

[132] J. Herman, A. Zimmer, J. Reijneveld, K. Dunlop, Y. Takahashi, S. Tardivel, and D. Scheeres. Human exploration of near earth asteroids: Mission analysis for a chemical and electric propulsion mission. *Acta Astronautica*.

[133] Martin Ozimek, Jack Riley, and Juan Arrieta. The low-thrust interplanetary explorer: A medium-fidelity algorithm for multi-gravity assist low-thrust trajectory optimization. In *AAS/AIAA Space Flight Mechanics Meeting*, Hawaii, 01 2019.

[134] J. Herman S. De Smet, J. Parker and R. Noomen. Mission design for a crewed earth-venus-mars-flyby mission using solar electric propulsion. In *32nd annual AAS Guidance and Control Conference*, number AAS 15-091 AAS 15-091, Breckenridge, CO,, 01 2015.

[135] E. Rinderle M. Vavrina J. Sims, P. Finlayson and T. Kowalkowski. Implementation of a low-thrust trajectory optimization algorithm for preliminary design. *AIAA/AAS Astrodynamics Specialist Conference*, August 2006.

[136] C. Yam, D. Di Lorenzo, and D. Izzo. Low-thrust trajectory design as a constrained global optimization problem. *Part G: Journal of Aerospace Engineering*, 225.

[137] M Vavrina and K Howell. Multiobjective optimization of low-thrust trajectories using a genetic algorithm hybrid. In *AAS/AIAA Space Flight Mechanics Meeting*, Savannah, GA, February 2009. American Astronautical Society Paper 09- 151.

[138] Federico Zuiani, Massimiliano Vasile, Alessandro Palmas, and Giulio Avanzini. Direct transcription of low-thrust trajectories with finite trajectory elements. *Acta Astronautica*, 72:108 – 120, 2012. ISSN 0094-5765. doi: 10.1016/j.actaastro.2011.09.011.

[139] Victoria Coverstone and Steven Williams. Optimal low thrust trajectories using differential inclusion concepts. *Journal of the Astronautical Sciences*, 42:379–393, 09 1994.

[140] Gerald A. Rauwolf and Victoria L. Coverstone-Carroll. Near-optimal low-thrust orbit transfers generated by a genetic algorithm. *Journal of Spacecraft and Rockets*, 33(6): 859–862, 2019/09/03 1996. doi: 10.2514/3.26850.

[141] Mauro Pontani, Pradipto Ghosh, and Bruce A. Conway. Particle swarm optimization of multiple-burn rendezvous trajectories. *Journal of Guidance, Control, and Dynamics*, 35 (4):1192–1207, 2012. doi: 10.2514/1.55592.

[142] Mark R. Ilgen. Hybrid method for computing optimal low thrust otv trajectories. *Advances in the Astronautical Sciences*, 87:941–958, 01 1994.

[143] Gao Yang. Direct optimization of low-thrust many-revolution earth-orbit transfers. *Chinese Journal of Aeronautics - CHIN J AERONAUT*, 22:426–433, 08 2009. doi: 10.1016/S1000-9361(08)60121-1.

[144] Federico Zuiani, Y Kawakatsu, and Massimiliano Vasile. Multi-objective optimisation of many-revolution, low-thrust orbit raising for destiny mission. *Advances in the Astronautical Sciences*, 148:783–802, 01 2013.

[145] Dong Eui Chang, David F. Chichka, and Jerrold E. Marsden. Lyapunov-based transfer between elliptic keplerian orbits. *Discrete and Continuous Dynamical Systems Series B*, 2(1):57–67, 02 2002.

[146] Ian Carnelli, Bernd Dachwald, and Massimiliano Vasile. Evolutionary neurocontrol: A novel method for low-thrust gravity-assist trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 32(2):616–625, 2009. doi: 10.2514/1.32633.

[147] Jonathan David Aziz. *Low-Thrust Many-Revolution Trajectory Optimization*. PhD thesis, University of Colorado, 2018.

[148] Gregory Lantoine and Ryan Russell. A hybrid differential dynamic programming algorithm for constrained optimal control problems. part 1: Theory. *Journal of Optimization Theory and Applications*, 154, 08 2013. doi: 10.1007/s10957-012-0039-0.

[149] S. N. Williams. An introduction to the use of varitop: A general purpose low-thrust trajectory optimization program. Technical Report jpl d-11475, Jet Propulsion Laboratory, California Institute of Technology, CA, 1994.

[150] Jr. Sauer, C. G. Optimization of multiple target electric propulsion trajectories. In *11th Aerospace Sciences Meeting*, page 11, Washington, D.C., Jan 01 1973. American Institute of Aeronautics and Astronautics.

[151] Tara Polsgrove, Randall Hopkins, Dan Thomas, Tracie M Crane, and Larry D Kos. Comparison of performance predictions for new low-thrust trajectory tools. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 08 2006. doi: 10.2514/6.2006-6742.

[152] Larry Kos, Tara Polsgrove, Randall Hopkins, Dan Thomas, and Jon Sims. Overview of the development for a suite of low-thrust trajectory analysis tools. 08 2006. ISBN 978-1-62410-048-2. doi: 10.2514/6.2006-6743.

[153] F. I. Mann and J. L. Horsewood. Program manuel for hiltop, a heliocentric interplanetary low thrust trajectory optimization program. Technical Report NASA-CR-143894, NASA, Analytical Mechanics Associates Inc., 1974.

[154] Régis Bertrand and Jacques Foliard. Low-thrust optimal trajectories for rendezvous with near earth asteroids. In *Proceedings of the 18th International Symposium on Space Flight Dynamics*, number ESA SP-548, Munich, Germany, 10 2004.

[155] A.S. Feistel and C.L. Ranieri. Modeling perturbations and operational considerations when using indirect optimization with equinotical elements. *Advances in the Astronautical Sciences*, 134:1737–1756, 01 2009.

[156] Chen Zhang, Francesco Topputo, Franco Bernelli-Zazzera, and Yu-Shan Zhao. Low-thrust minimum-fuel optimization in the circular restricted three-body problem. *Journal of Guidance, Control, and Dynamics*, 38(8):1501–1510, 2015. doi: 10.2514/1.G001080.

[157] J. B. Caillau, J. Gergaud, and J. Noailles. Tfmin: Short reference manual. *Optimization Online Digest*, 2001.

[158] Jean A. Kéchichian. Optimal low-Earth-orbit-geostationary-earth-orbit intermediate acceleration orbit transfer. *Journal of Guidance, Control, and Dynamics*, 20(4):803–811, 1997. doi: 10.2514/2.4116.

[159] T. Dargent and Vincent Martinot. An integrated tool for low thrust optimal control orbit transfers in interplanetary trajectories. In *International Symposium on Space Flight Dynamics*, page 143, Munich, Germany, 01 2004.

[160] Pelayo Penarroya Juan C. Bastante. Electro: A sw tool for the electric propulsion trajectory optimisation. In *7th International Conference on Astrodynamics Tools and Techniques*, Munich, Germany, 10 2018.

[161] S. Geffroy and R. Epenoy. Optimal low-thrust transfers with constraints—generalization of averaging techniques. *Acta Astronautica*, 41:133–149, 1997. doi: 10.1016/S0094-5765(97)00208-7.

[162] Lester L. Sackett, Harvey L. Malchow, and Theodore N. Edelbaum. Solar electric geocentric transfer with attitude constraints: Analysis. Technical Report NASA CR–134927, NASA, 1975.

[163] V. Coverstone-Carroll, J.W. Hartmann, and W.J. Mason. Optimal multi-objective low-thrust spacecraft trajectories. *Computer Methods in Applied Mechanics and Engineering*, 186(2):387 – 402, 2000. ISSN 0045-7825. doi: 10.1016/S0045-7825(99)00393-X.

[164] Matteo Rosa Sentinella and Lorenzo Casalino. Genetic algorithm and indirect method coupling for low-thrust trajectory optimization. In *42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference &amp; Exhibit*, 2006. doi: 10.2514/6.2006-4468.

[165] Mauro Pontani and Bruce A. Conway. Particle swarm optimization applied to space trajectories. *Journal of Guidance, Control, and Dynamics*, 33(5):1429–1441, 2010. doi: 10.2514/1.48475.

[166] Seungwon Lee, Wolfgang Finkt, Paul von Allmed, Paul Von Allmen, Anastassios E. Petropoulos, and Richard J Terrile. Evolutionary computing for low-thrust navigation. 08 2005. doi: 10.2514/6.2005-6835.

[167] H. Oberle and W. Grimm. *BNDSCO: A program for the numerical solution of optimal control problems.* PhD thesis, Inst. für Angewandte Math. der Univ. Hamburg, Germany, 2001.

[168] Yazhe Meng, Hao Zhang, and Yang Gao. Low-thrust minimum-fuel trajectory optimization using multiple shooting augmented by analytical derivatives. *Journal of Guidance, Control, and Dynamics*, 42:1–16, 12 2018. doi: 10.2514/1.G003473.

[169] Joris Olympio. Algorithm for low-thrust optimal interplanetary transfers with escape and capture phases. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit.* American Institute of Aeronautics and Astronautics, 2017/09/26 2008. doi: 10.2514/6.2008-7363.

[170] H.S. Tsien. Take-off from satellite orbit. *Journal of the American Rocket Society*, 23, 07 1953. doi: 10.2514/8.4599.

[171] Richard H. Battin. An introduction to the mathematics and methods of astrodynamics. *New York, American Institute of Aeronautics and Astronautics, Inc., 1987, 824 p.*, 01 1987. doi: 10.2514/4.861543.

[172] Dario Izzo and Francesco Biscani. Explicit solution to the constant radial acceleration problem. *Journal of Guidance, Control, and Dynamics*, 38(4):733–739, 2015. doi: 10.2514/1.G000116.

[173] Claudio Bombardelli, Giulio Baù, and Jesus Pelaez. Asymptotic solution for the two-body problem with constant tangential thrust acceleration. *Celestial Mechanics and Dynamical Astronomy*, 110:239–256, 07 2011. doi: 10.1007/s10569-011-9353-3.

[174] John E. Prussing. Constant radial thrust acceleration redux. *Journal of Guidance, Control, and Dynamics*, 21(3):516–518, 1998. doi: 10.2514/2.7609.

[175] Giovanni Mengali and Alessandro A. Quarta. Escape from elliptic orbit using constant radial thrust. *Journal of Guidance, Control, and Dynamics*, 32(3):1018–1022, 2009. doi: 10.2514/1.43382.

[176] Theodore N. Edelbaum. Propulsion requirements for controllable satellites. *ARS Journal*, 31(8):1079–1089, 1961. doi: 10.2514/8.5723.

[177] Jean A. Kéchichian. Reformulation of Edelbaum's low-thrust transfer problem using optimal control theory. *Journal of Guidance, Control, and Dynamics*, 20(5):988–994, 1997. doi: 10.2514/2.4145.

[178] Theodore N. Edelbaum. An asymptotic solution for optimum power limited orbit transfer. *AIAA Journal*, 4:15, 10 1965. doi: 10.2514/3.3725.

[179] Sandro Fernandes, Francisco Carvalho, and Rodolpho Vilhena de Moraes. Optimal low-thrust transfers between coplanar orbits with small eccentricities. *Computational and Applied Mathematics*, 641, 07 2015. doi: 10.1007/s40314-015-0249-9.

[180] Armando Ruggiero, Pierpaolo Pergola, Salvo Marcuccio, and Mariano Andrenucci. Low-thrust maneuvers for the efficient correction of orbital elements. *Proceedings of the 32nd International Electric Propulsion Conference, IEPC Paper*, pages 1–13, 01 2011.

[181] Jean A. Kéchichian. *Analytic Representations of Optimal Low-Thrust Transfer in Circular Orbit*, pages 139–177. Cambridge Aerospace Series. Cambridge University Press, 2010. doi: 10.1017/CBO9780511778025.007.

[182] E.G.C. Burt. On space manoeuvres with continuous thrust. *Planetary and Space Science*, 15(1):103 – 122, 1967. ISSN 0032-0633. doi: 10.1016/0032-0633(67)90070-0.

[183] J. Pollard. Simplified analysis of low-thrust orbital maneuvers. page 42, 08 2000.

[184] Marilena Di Carlo, Juan Manuel Romero Martin, and Massimiliano Vasile. Camelot: Computational-analytical multi-fidelity low-thrust optimisation toolbox. *CEAS Space Journal*, 10, 09 2017. doi: 10.1007/s12567-017-0172-6.

[185] Craig A. Kluever. Using Edelbaum's method to compute low-thrust transfers with Earth-shadow eclipses. *Journal of Guidance, Control, and Dynamics*, 34(1):300–303, 2011. doi: 10.2514/1.51024.

[186] Jean A. Kechichian. Low-thrust eccentricity-constrained orbit raising. *Journal of Spacecraft and Rockets*, 35(3):327–335, 1998. doi: 10.2514/2.3330.

[187] Guido Colasurdo and Lorenzo Casalino. Optimal low-thrust maneuvers in presence of earth shadow. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2004. doi: 10.2514/6.2004-5087.

[188] G. Flandro. Asymptotic solution for solar electric low thrust orbit raising witheclipse penalty. In *Mechanics and Control of Flight Conference*, number AIAA Paper 1974-0802, 1974. doi: 10.2514/6.1974-802.

[189] Jean A. Kéchichian. The streamlined and complete set of the nonsingular J2-perturbed dynamic and adjoint equations for trajectory optimization in terms of eccentric longitude. *The Journal of the Astronautical Sciences*, 55, 09 2007. doi: 10.1007/BF03256528.

[190] Jean A. Kéchichian. Inclusion of higher order harmonics in the modeling of optimal low-thrust orbit transfer. *The Journal of the Astronautical Sciences*, 56, 03 2008. doi: 10.1007/BF03256541.

[191] Theodore N Edelbaum, L. Sackett, and H. Malchow. Optimal low thrust geocentric transfer. In *10th Electric Propulsion Conference*. doi: 10.2514/6.1973-1074.

[192] Jacob A. Englander and Bruce A. Conway. Automated solution of the low-thrust interplanetary trajectory problem. *Journal of Guidance, Control, and Dynamics*, 40(1):15–27, 2017/09/21 2016. doi: 10.2514/1.G002124.

[193] Ryne Beeson, Jacob A. Englander, Steven P Hughes, and Maximilian Schadegg. An automatic medium to high fidelity low-thrust global trajectory tool-chain; emtg-gmat. In *25th AAS/AIAA Space Flight Mechanics Meeting*, Williamsburg, VA, 01 2015.

[194] Jennifer Hargens and Victoria Coverstone. Low-thrust interplanetary mission design using differential inclusion. 08 2002. ISBN 978-1-62410-124-3. doi: 10.2514/6.2002-4730.

[195] Hyun Chul Ko and D. Scheeres. Essential thrust-fourier-coefficient set of averaged gauss equations for orbital mechanics. *Journal of Guidance, Control, and Dynamics*, 37:1236–1249, 07 2014. doi: 10.2514/1.62407.

[196] Daniel Kolosa and Jennifer Hudson. A tfc approach to low-thrust trajectory optimization in stk. In *2018 Space Flight Mechanics Meeting*, 01 2018. doi: 10.2514/6.2018-0960.

[197] D. Chang, David Chichka, and J. Marsden. Lyapunov functions for elliptic orbit transfer. 01 2002.

[198] Pini Gurfil. Nonlinear feedback control of low-thrust orbital transfer in a central gravitational field. *Acta Astronautica*, 60:631–648, 04 2007. doi: 10.1016/j.actaastro.2006.10.001.

[199] George F Forbes. The trajectory of a powered rocket in space. *Journal of the British Interplanetary Society*, 9(2):75–79, 1950.

[200] T. C. Tsu. Interplanetary travel by solar sail. *ARS Journal*, 29(6):422–427, 2018/04/06 1959. doi: 10.2514/8.4791.

[201] G. Pinkham. Reference solution for low thrust trajectories. *Journal of American Rocket Society*, 32(5):775–776, May 1962.

[202] D. F. Lawden. Optimal programming of rocket thrust direction. *Acta Astronautica*, 1: 41–56., 1995.

[203] Dario Izzo. Lambert's problem for exponential sinusoids. *Journal of Guidance, Control, and Dynamics*, 29(5):1242–1245, 2006. doi: 10.2514/1.21796.

[204] M. Vasile, O. Schütze, and O. Junge. Spiral trajectories in global optimization of interplanetary and orbital transfers. Technical Report ARIADNA AO05/4106, ESA, 2005.

[205] Javier Roa, Jesús Peláez, and Juan Senent. New analytic solution with continuous thrust: Generalized logarithmic spirals. *Journal of Guidance, Control, and Dynamics*, 39(10): 2336–2351, 2018/04/06 2016. doi: 10.2514/1.G000341.

[206] Javier Roa and Jesús Peláez. Introducing a degree of freedom in the family of generalized logarithmic spirals. In *26th Spaceflight Mechanics Meeting*, number 16-317, Springfield, VA, 2016/02/16 2015. American Astronautical Society Paper 16-317.

[207] Javier Roa and Jesus Pelaez. Three-dimensional generalized logarithmic spirals. In *26th AAS/AIAA Space Flight Mechanics Meeting*, Napa, CA, 02 2016.

[208] B. Dachwald. *Low-Thrust Trajectory Optimization and Interplanetary Mission Analysis Using Evolutionary Neurocontrol.* PhD thesis, Univ. der Bundeswehr München; Fakultät für Luft- und Raumfahrttechnik; Inst. für Raumfahrttechnik, Munich, Germany, 2004.

[209] Gregory J. Whiffen and John A. Sims. Application of a novel optimal control algorithm to low-thrust trajectory optimization. 2001.

[210] Gregory J. Whiffen and John Sims. Application of the SDC optimal control algorithm to low-thrust escape and capture trajectory optimization. Technical report, 2002.

[211] Gregory Lantoine and Ryan Russell. A hybrid differential dynamic programming algorithm for constrained optimal control problems. part 2: Application. *Journal of Optimization Theory and Applications*, 154, 08 2013. doi: 10.1007/s10957-012-0038-1.

[212] Michael S Branicky, Vivek S Borkar, and Sanjoy K Mitter. A unified framework for hybrid control: Model and optimal control theory. *IEEE transactions on automatic control*, 43 (1):31–45, 1998. doi: 10.1109/9.654885.

[213] Martin Buss, Markus Glocker, Michael Hardt, Oskar von Stryk, Roland Bulirsch, and Günther Schmidt. *Nonlinear Hybrid Dynamical Systems: Modeling, Optimal Control, and Applications*, pages 311–335. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002. ISBN 978-3-540-45426-7. doi: 10.1007/3-540-45426-8_18.

[214] David E. Goldberg and Richardson J.J. Genetic algorithms with sharing for multi-modal function optimization. In *2nd International Conference on Genetic Algorithms*, pages 41–49, 01 1987.

[215] David E. Goldberg and Kalyanmoy Deb. A comparative analysis of selection schemes used in genetic algorithms. volume 1 of *Foundations of Genetic Algorithms*, pages 69 – 93. Elsevier, 1991. doi: 10.1016/B978-0-08-050684-5.50008-2.

[216] Gilbert Sywerda. Uniform crossover in genetic algorithms. In *Proceedings of the Third International Conference on Genetic Algorithms*, pages 2–9, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc. ISBN 1-55860-006-3.

[217] Dan M. Goebel and Ira Katz. *Fundamentals of electric propulsion: Ion and Hall thrusters*, volume 1. John Wiley & Sons, 2008. doi: 10.1002/9780470436448.

[218] C. E. Jordan. Nasa radiation belt models ap-8 and ae-8. Technical Report GL-TR-89-0267, Radex Inc, Bedford, MA, September 1989.

[219] S. R. Messenger, M. A. Xapsos, E. A. Burke, R. J. Walters, and G. P. Summers. Proton displacement damage and ionizing dose for shielded devices in space. *IEEE Transactions on Nuclear Science*, 44(6):2169–2173, December 1997. doi: 10.1109/23.659032.

[220] Robert Fourer, David M. Gay, and Brian W. Kernighan. Ampl: A modeling language for mathematical programming. Technical report, Duxbury Press, November 2002.

[221] S. R. Messenger, E. M. Jackson, J. H. Warner, and R. J. Walters. Scream: A new code for solar cell degradation prediction using the displacement damage dose approach. In *IEEE Photovoltaic Specialists Conference*, pages 1106–1111. IEEE, June 2010. doi: 10.1109/PVSC.2010.5614713.

[222] Insoo Jun, Michael A. Xapsos, Scott R. Messenger, Edward A. Burke, Robert J. Walters, Geoff P. Summers, and Thomas Jordan. Proton nonionizing energy loss (niel) for device applications. *IEEE Transactions on Nuclear Science*, 50(6):1924–1928, 2003. doi: 10.1109/TNS.2003.820760.

[223] Kathryn F. Graham and Anil V. Rao. Minimum-time trajectory optimization of multiple revolution low-thrust earth-orbit transfers. *Journal of Spacecraft and Rockets*, 52(3):711–727, 2018/10/17 2015. doi: 10.2514/1.A33187.

[224] C H Yam, D D Lorenzo, and D Izzo. Low-thrust trajectory design as a constrained global optimization problem. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 225(11):1243–1251, 2011. doi: 10.1177/0954410011401686.

[225] Ahmed Gad and Ossama Abdelkhalik. Hidden genes genetic algorithm for multi-gravity-assist trajectories optimization. *Journal of Spacecraft and Rockets*, 48(4):629–641, 2017/09/21 2011. doi: 10.2514/1.52642.

[226] J. A. Sims and S. N. Flanagan. Preliminary design of low-thrust interplanetary missions. In *Astrodynamics Specialist Conference*, Girdwood, Alaska, 1999/08/16 1999. American Astronautical Society Paper 99-338.

[227] Steven N Williams and Victoria Coverstone-Carroll. Benefits of solar electric propulsion for the next generation of planetary exploration missions. *Journal of the Astronautical Sciences*, 45(2):143–159, 1997.

[228] Jacob A. Englander, Bruce A. Conway, and Trevor Williams. Automated mission planning via evolutionary algorithms. *Journal of Guidance, Control, and Dynamics*, 35(6):1878–1887, 2017/09/21 2012. doi: 10.2514/1.54101.

[229] Marcie A. Wise, Jarret M. Lafleur, and Joseph H. Saleh. Regression analysis of launch vehicle payload capability for interplanetary missions. In *61st International Astronautical Congress*, Prague, CZ, 2010.