



POLITECNICO DI MILANO

DEPARTMENT OF AEROSPACE SCIENCE AND TECHNOLOGY

DOCTORAL PROGRAMME IN AEROSPACE ENGINEERING

**ROBUST DESIGN OF LOW-THRUST TRAJECTORIES
THROUGH DIFFERENTIAL DYNAMIC PROGRAMMING
ENHANCING THE EFFECTS OF ORBITAL
PERTURBATIONS**

Marco Nugnes

Submitted in fulfilment of the requirements
for the Degree of Doctor of Philosophy

Supervisor:

Prof. Camilla Colombo

Tutor:

Prof. Filippo Maggi

PhD Programme Coordinator:

Prof. Pierangelo Masarati

Academic Year 2021/2022 - XXXIV Cycle

Copyright © 2023, Marco Nugnes. All rights reserved.

This content is original, written by the Author, Marco Nugnes. All the non-originals information, taken from previous works, are specified and recorded in the Bibliography.

The content must be not changed in any way or sold without the written permission of copyright owner.

When referring to this work, full bibliographic details must be given, i.e.

Marco Nugnes, “Robust Design of Low-thrust trajectories through Differential Dynamic Programming enhancing the Effects of Orbital Perturbations”, 2023, Politecnico di Milano, Faculty of Industrial Engineering, Department of Aerospace Science and Technologies, final dissertation for the degree of Doctor of Philosophy in Aerospace Engineering. Supervisor: Camilla Colombo.

*Dedicated to myself who struggled many years to achieve this result and to all the people
believing in me.*

Abstract

Several future space missions will be equipped with low-thrust engines as main propulsion subsystem. This is true not only for planetary missions such as orbit raising, de-orbiting, Geostationary Transfer Orbits (GTOs), but also for interplanetary missions and large satellite constellations. The advantage in using low-thrust systems grants a higher final operational mass at the expense of a longer time of flight. Indeed, the main task of a space mission is to host a dedicated payload, which is associated to a set of scientific instruments to accomplish multiple objectives. The higher is the final operational mass the higher is the number of payloads to be set onboard of a single satellite. In the last 10 years the advancement in technology allowed to develop different types of new electric engines: solar engines, Hall effect thrusters, and plasma thrusters. The use of low-thrust engines completely changes the design problem of a spacecraft trajectory. When chemical engines, characterised by high thrust level, are involved in the design, the spacecraft trajectory design is solved considering multiple ballistic arcs separated by impulsive manoeuvres.

The introduction of low-thrust engines turns the control policy from discontinuous manoeuvres to continuous laws. The design of a continuous control law requires the solution of a non-linear optimal control problem. One of the many techniques used to solve this problem is Differential Dynamic Programming (DDP) which couples the main advantages of classic direct and indirect methods. Despite its strong mathematical formulation the DDP technique did not find many applications over the past and the state representation was always chosen in terms of Cartesian coordinates. It is interesting to apply this technique to spacecraft trajectory design problems written in terms of orbital elements. Indeed, orbital elements represent a natural set to describe orbit mechanics because they provide a more physical insight into the problem. The variation of orbital elements can be rapidly associated to a geometrical representation of the spacecraft trajectory, whereas a variation of the position or the velocity vector in terms of Cartesian coordinates cannot be visualised until the full trajectory is plotted.

The aim of this research is to investigate the coupling between the optimisation technique of DDP with the use of orbital elements as dynamic formulation and to apply the algorithm to space mission trajectory design problems. The other topic investigated is the possibility to leverage the effect of orbital perturbations as accelerations that can be engineered in the space mission design and not see them as disturbing factors. This second objective is set by the COMPASS ERC project which funded this research. The main contributions of this work are the definition of a mathematical theory to couple DDP with orbital elements and the formulation of cost function to optimise the maximum shadow time experienced by a spacecraft. The algorithm is used to solve single or multi-revolution interplanetary and planetary transfers.

Sommario

Numerose missioni spaziali future verranno equipaggiate con motori a bassa spinta. Tale scelta non riguarderà solo missioni di carattere planetario come innalzamento dell'apocentro, rientro di un satellite o orbite di trasferimento Geostazionarie (GTOs), ma anche a missioni interplanetarie. Il vantaggio principale nell'utilizzo di motori a bassa spinta è quello di garantire una maggiore massa per la strumentazione scientifica a bordo (il cosiddetto *payload*) a discapito di un maggior tempo di trasferimento. Lo sviluppo di motori a bassa spinta è cominciato, negli ultimi anni del XX secolo, attraverso l'utilizzo di motori ionici che sfruttavano lo xenon come propellente. Negli ultimi 10 anni il progresso tecnologico ha permesso lo sviluppo di ulteriori tipi di motore a bassa spinta: motori solari, motori basati sull'effetto di Hall e motori al plasma.

L'uso di un motore a bassa spinta come sistema propulsivo cambia la progettazione della traiettoria del satellite. Infatti, per un satellite che sfrutta motori chimici caratterizzati da alte spinte, la traiettoria del satellite risulta essere l'unione di più archi balistici separati ognuno da una manovra impulsiva. Nel caso di motori a bassa spinta, l'accelerazione che viene fornita al satellite diventa continua trasformando la progettazione della traiettoria in un problema di controllo ottimo altamente non lineare. Una delle tecniche numeriche per poter risolvere tale problema consiste nella Programmazione Dinamica Differenziale (DDP) che unisce gli aspetti vantaggiosi dei classici metodi diretti e indiretti. Tuttavia, la DDP non ha vantato un grande utilizzo in particolare in ambito spaziale ed ha sempre usato un sistema di rappresentazione Cartesiana come caratterizzazione dello stato di un sistema dinamico.

Questa ricerca propone lo studio e l'applicazione della DDP in ambito della progettazione delle traiettorie di un satellite usando i parametri orbitali come rappresentazione della dinamica. I parametri orbitali sono una scelta naturale per risolvere i problemi di meccanica celeste perché offrono la possibilità di una visualizzazione più chiara della soluzione ottenuta. Ogni parametro orbitale può essere associato ad una caratteristica geometrica dell'orbita. Pertanto, la variazione di un parametro orbitale è sicuramente di più facile interpretazione rispetto alla variazione delle coordinate Cartesiane che risultano delle variabili che oscillano durante la rivoluzione di un satellite e che non forniscono alcun senso illustrativo fino a quando l'orbita tridimensionale non viene graficata. Un altro elemento analizzato in tale ricerca è la possibilità di usare le perturbazioni orbitali come accelerazioni a favore della progettazione della traiettoria di un satellite e non più come degli elementi di disturbo che devono essere contrastati con apposite manovre correttive. Questo secondo obiettivo è fissato dal progetto ERC COMPASS che ha finanziato la ricerca. I contributi principali di questo lavoro sono la definizione di una teoria matematica sull'accoppiamento della DDP con gli elementi orbitali e la formulazione di una funzione costo per ottimizzare il tempo massimo speso da un satellite all'interno dell'ombra generata in un'eclissi.

Acknowledgements

The doctoral programme in Aerospace Engineering at Politecnico di Milano represented a great adventure both from an academic point of view and especially as a life experience. I would like first to acknowledge my supervisor, Prof. Camilla Colombo for the opportunity to work on this research topic. Starting from November 2018 you were a lighthouse every time some troubles appeared in the development of this research. You always believed in my skills and this was the strongest support received in all this time.

Thank you to Prof. Francesco Topputo for your inspiring course about numerical optimisation. It was a pleasure being part of the Department of Aerospace Science and Technology at Politecnico di Milano. Thank you to all its members for being professional but also for the organisation of extra-curricula and social activities.

I would like to thank Dr. Gregory Lantoine for being a wonderful supervisor during the research stay at Jet Propulsion Laboratory. Your availability and kindness were fundamental to rapidly adapt in a foreign country right after the relax of Covid-19 pandemics restrictions.

Thank you to my thesis reviewers Prof. Lorenzo Casalino and Prof. Giulio Avanzini for spending your precious time in reading this thesis. Your valuable comments were fundamental in improving the thesis.

I would like to thank all the former and current members of the COMPASS group: Juan, Ioannis, Narcís, Mirko, Matteo, Stefan, Davide, Gabriella, Giacomo, Francesca, Alessandro, Andrea, Lorenzo, Martina, Edoardo. I much appreciated all the coffee breaks, aperitifs, pizzas, and dinners with you and represented a moral support for the largest part of the doctoral course. Juan, I will never forget your fake Italian. Giacomo, I will not forget the funny lunches based on your vegan burgers.

I thank also my family for being always present every time I encountered health issues and supporting me even if I am far from home.

Last but not least, I would like to dedicate some lines to Elena and Alberto. You literally represent a second family. Thank you Elena for your precious advices every time that I was lost. Thank you Alberto for your friendship because I know that it was difficult to tackle with my strong character. Thank you for all the nice times playing together, but also for the quarrels because they are the proof that your friendship is clean. Your support was fundamental to face the negative consequences of the Covid-19 pandemics and to let me come back in all the several sad moments to succeed in accomplishing this task.

Contents

Abstract	i
Sommario	iii
Acknowledgments	v
Nomenclature	xiv
List of Symbols	xiv
List of constants	xiv
List of acronyms	xiv
Nomenclature	xv
1 Introduction	1
1.1 Design of optimal spacecraft trajectories	1
1.2 Research motivations and objectives	2
1.3 Background	3
1.3.1 Numerical methods for trajectory optimisation	3
1.3.2 Orbital elements as state representation	7
1.3.3 Eclipse algorithms	8
1.4 Methodologies developed and implemented	9
1.5 Structure of the thesis	11
1.6 Contributions	12
2 Orbit dynamics	15
2.1 Gauss' equations	15
2.1.1 Keplerian elements	16

2.1.2	Nonsingular elements	19
2.2	Transition functions	21
2.2.1	Analytical computation of the partials	22
2.2.2	Scheme of the algorithm	24
2.3	Summary	26
3	Differential dynamic programming	28
3.1	Mathematical derivation continuous version	28
3.2	Mathematical derivation of the DDP discrete version	33
3.2.1	Step-size adjustment method	37
3.2.2	Hessian positive definiteness	40
3.2.3	Lagrange multipliers corrections	42
3.3	Constraints formulation	43
3.4	Summary of the algorithm	43
4	Heuristic techniques for the constrained problem in using orbital elements	46
4.1	Quasi-linear independence	46
4.2	Angular variables management	48
4.3	Continuation technique	49
4.4	Results	51
4.4.1	Earth to Mars interplanetary transfer	52
4.4.2	Earth to near-Earth asteroid transfer	58
5	Static/dynamic variables optimisation and multiple revolutions	63
5.1	Free final time and static parameters	63
5.2	Sundman transformation	70
5.3	Multiple revolutions	72
5.4	Results	75
6	Robustness and partials computation analysis	78
6.1	State transition matrix formulation	78
6.2	Results	81
6.2.1	Analytical vs STM partial evaluation	81
6.2.2	Robustness of the control law	83

6.3	Summary	87
7	Path constraints inclusion: eclipses	88
7.1	Eclipse algorithm for the analytical derivation of entry/exit umbra points	88
7.1.1	Rotation and translation	88
7.1.2	Intersection of the conical surface with the spacecraft orbit	92
7.1.3	Solutions decision making	94
7.2	Orbital scenario	96
7.2.1	Intersection eclipse with the oblate Earth	96
7.2.2	Characterisation of cone vertex and rotation matrix	97
7.2.3	Penumbra characterisation	100
7.3	Results	101
7.3.1	Low-Earth slightly inclined orbit	102
7.3.2	Low-Earth highly inclined orbit	103
7.3.3	Geosynchronous orbit	104
7.3.4	Highly elliptical orbit	105
7.3.5	Discussion	105
7.4	Definition of the constraining area	108
7.5	Definition of the function linking shadow time with the constraining area	110
7.6	Cost function formulation	111
7.7	Integration in the DDP algorithm	114
7.7.1	Application of the DDP with the shadow time requirement	116
8	Conclusions and future work	120
8.1	Conclusions	120
8.2	Future work	121
A	Gauss' planetary equations in different reference frames and orbital elements set	130
A.1	Gauss equations in terms of classic Keplerian orbital elements	130
A.2	Gauss equations in terms of modified equinoctial elements	132
B	Cardan's method to solve quartic equations	138

List of Figures

2.1	Partials computation flow diagram.	24
3.1	Nominal vs optimal trajectory representation during the step-size adjustment method	39
3.2	Nominal vs optimal control representation during the step-size adjustment method	39
3.3	Graphical representation of the DDP algorithm.	45
4.1	Angular index flow diagram.	49
4.2	Continuation scheme for the true anomaly convergence.	51
4.3	Jacobian matrix of the transition function with respect to the state for $m_{ref} = 150 \text{ kg}$	53
4.4	Jacobian matrix of the transition function with respect to the state for $m_{ref} = 3 \text{ kg}$.	53
4.5	Jacobian matrix of the transition function with respect to the control for $m_{ref} = 150 \text{ kg}$	53
4.6	Jacobian matrix of the transition function with respect to the control for $m_{ref} = 3 \text{ kg}$	53
4.7	Hessian matrix of the transition function with respect to the state for $m_{ref} = 150 \text{ kg}$	54
4.8	Hessian matrix of the transition function with respect to the state for $m_{ref} = 3 \text{ kg}$.	54
4.9	Hessian matrix of the transition function with respect to the control for $m_{ref} = 150 \text{ kg}$	54
4.10	Hessian matrix of the transition function with respect to the control for $m_{ref} = 3 \text{ kg}$	54
4.11	Hessian matrix of the transition function with respect to the state and control for $m_{ref} = 150 \text{ kg}$	54
4.12	Hessian matrix of the transition function with respect to the state and control for $m_{ref} = 3 \text{ kg}$	54
4.13	Jacobian matrix of the cost functional with respect to the state for $m_{ref} = 150 \text{ kg}$.	55
4.14	Jacobian matrix of the cost functional with respect to the state for $m_{ref} = 3 \text{ kg}$.	55

4.15	Hessian matrix of the cost functional with respect to the state for $m_{ref} = 150 \text{ kg}$.	55
4.16	Hessian matrix of the cost functional with respect to the state for $m_{ref} = 3 \text{ kg}$.	55
4.17	Nominal vs optimal control magnitude time history for Earth to Mars transfer: part 1.	56
4.18	First part vs final optimal control magnitude time history for Earth to Mars transfer: part 2.	56
4.19	Nominal vs optimal semi-major axis trajectory for Earth to Mars transfer.	57
4.20	Nominal vs optimal eccentricity trajectory for Earth to Mars transfer.	57
4.21	Nominal vs optimal inclination trajectory for Earth to Mars transfer.	57
4.22	Nominal vs optimal RAAN trajectory for Earth to Mars transfer.	57
4.23	Nominal vs optimal ω trajectory for Earth to Mars transfer.	57
4.24	Nominal vs optimal ν trajectory for Earth to Mars transfer.	57
4.25	Final interplanetary Earth to Mars interplanetary trajectory.	58
4.26	Nominal control time law for Earth to Mars transfer.	58
4.27	Optimal control time law for Earth to Mars transfer.	58
4.28	Nominal vs optimal control magnitude time history for Earth to Apophis transfer: part 1.	60
4.29	First layer vs final optimal control magnitude time history for Earth to Apophis transfer: part 2.	60
4.30	Nominal vs optimal semi-major axis trajectory for Earth to Apophis transfer.	60
4.31	Nominal vs optimal f trajectory for Earth to Apophis transfer.	60
4.32	Nominal vs optimal g trajectory for Earth to Apophis transfer.	60
4.33	Nominal vs optimal h trajectory for Earth to Apophis transfer.	60
4.34	Nominal vs optimal k trajectory for Earth to Apophis transfer.	61
4.35	Nominal vs optimal true longitude trajectory for Earth to Apophis transfer.	61
4.36	Optimal vs nominal 3D trajectory for Earth to Apophis transfer.	61
4.37	Nominal control law for Earth to Apophis transfer.	62
4.38	Optimal control law for Earth to Apophis transfer.	62
5.1	Nodes distribution using time as independent variable.	71
5.2	Nodes distribution using true anomaly as independent variable.	71
5.3	Nodes distribution using mean anomaly as independent variable.	71
5.4	Nodes distribution using eccentric anomaly as independent variable.	71
5.5	Nodes distribution using time as independent variable when multiple revolutions are considered.	73

5.6	Nodes distribution using eccentric anomaly as independent variable when multiple revolutions are considered.	74
5.7	Number of steps definition procedure in case Sundman transformation in terms of eccentric anomaly is applied.	74
5.8	Nominal vs optimal semi-major axis trajectory.	76
5.9	Nominal vs optimal eccentricity trajectory.	76
5.10	Nominal vs optimal orbital inclination trajectory.	77
5.11	Nominal vs optimal thrust magnitude trajectory.	77
5.12	Nominal vs optimal thrust components trajectory.	77
5.13	Final interplanetary GTO to GEO 3D trajectory.	77
6.1	Optimal control components associated to the unperturbed initial conditions. . .	84
6.2	Optimal control components associated to the perturbed initial conditions. . . .	84
6.3	Comparison between the control magnitudes associated to the unperturbed and perturbed optimal control problem.	85
6.4	Semi-major axis unperturbed vs perturbed initial condition.	85
6.5	Eccentricity unperturbed vs perturbed initial condition.	85
6.6	Inclination unperturbed vs perturbed initial condition.	86
6.7	RAAN unperturbed vs perturbed initial condition.	86
6.8	Pericenter anomaly unperturbed vs perturbed initial condition.	86
6.9	True anomaly unperturbed vs perturbed initial condition.	86
7.1	General rotation problem from one reference frame to another one.	89
7.2	General translation problem from one reference frame to another one.	91
7.3	Translation from ellipse centred to focal reference frame.	92
7.4	Intersection of the conical surface with the elliptical orbit.	95
7.5	Graphical representation of the variables used to define the shadow cone. . . .	98
7.6	Graphical representation of umbra and penumbra regions.	101
7.7	Cumulative eclipse time error for scenario 2.	107
7.8	Cumulative eclipse time error for scenario 4.	107
7.9	Eclipse time error as function of orbit inclination for scenario 2.	107
7.10	Eclipse time error as function of orbit eccentricity for scenario 2.	107
7.11	Eclipse time error as function of orbit inclination for scenario 4.	108
7.12	Eclipse time error as function of orbit eccentricity for scenario 4.	108
7.13	Intersection of the conical surface with the elliptical orbit.	109

7.14	General representation of the minimum distance of a point from a right-circular cone.	112
7.15	Geometrical representation of the segments to compute the minimum distance from a right-circular cone.	113
7.16	Minimum distance from a cone surface squared for a single revolution LEO spacecraft.	114
7.17	Optimal orbit raising considering the eclipse weighting factor equal to zero. . .	117
7.18	Optimal orbit raising considering the eclipse weighting factor equal to 1000. . .	118
7.19	Optimal orbit raising considering the eclipse weighting factor equal to 2000. . .	118

List of Tables

2.1	Butcher tableau for Runge-Kutta 8 th order integration scheme.	27
4.1	Satellite data and DDP tolerances.	51
4.2	Input for Earth to Mars interplanetary transfer.	52
4.3	Reference variables for Earth to Mars transfer.	52
4.4	Input for Earth to Apophis transfer.	59
4.5	Reference variables for Earth to Apophis transfer.	59
5.1	Initial orbital elements for the GTO to GEO transfer.	75
5.2	Input for the GTO to GEO reference transfer.	75
5.3	Comparison between the two different simulations.	75
6.1	Input for the Earth to Mars interplanetary transfer used for the comparison between STMs and analytical formulation.	82
6.2	Input for the Earth to Mars interplanetary transfer used for the comparison between STMs and analytical formulation.	82
6.3	Initial orbital elements and list of initial disturbances added.	83
6.4	Error between the prescribed target state and the actual state considering the different control laws.	86
7.1	Sun state vector used for all the orbital scenarios.	102
7.2	Spacecraft initial orbital elements for scenario 1.	102
7.3	Umbra entry and exit points for scenario number 1.	103
7.4	Penumbra entry and exit points for scenario 1.	103
7.5	Satellite initial orbital elements for scenario 2.	103
7.6	Umbra entry and exit points for scenario number 2.	103
7.7	Penumbra entry and exit points for scenario 2.	104
7.8	Satellite initial orbital elements for scenario 3.	104

7.9	Umbra entry and exit points for scenario number 3.	104
7.10	Penumbra entry and exit points for scenario 3.	104
7.11	Satellite initial orbital elements for scenario 4.	105
7.12	Umbra entry and exit points for scenario 4.	105
7.13	Penumbra entry and exit points for scenario 4.	105
7.14	Summary of the main characteristics of the eclipse algorithms. Legenda: O = Oblate, S = Sphere, CO = Conical, CY = Cylinder, A = Analytical, N = Numerical	106
7.15	Input for the LEO orbit raising.	116

Nomenclature

List of symbols

Latin Symbols

$\bar{\mathbf{b}}$	Nominal Lagrange multiplier vector
$\bar{\mathbf{u}}$	Nominal control
$\bar{\mathbf{x}}$	Nominal state vector
\bar{F}	Terminal cost
\bar{J}	Nominal value function
$\delta \mathbf{u}^*$	Optimal control variation
$\delta \mathbf{u}$	Control variation
$\delta \mathbf{w}$	Static parameter variation
$\delta \mathbf{x}$	State variation
δt_f	Free final time variation
\mathbf{f}	Continuou system dynamics
\mathbf{A}	Auxiliary matrix multiplying the control variations $\delta \mathbf{u}^T \dots \delta \mathbf{u}$ inside the expanded HJB equation
\mathbf{A}	Auxiliary matrix multiplying the state variations $\delta \mathbf{x}^T \dots \delta \mathbf{x}$ inside the expanded HJB equation
\mathbf{a}_p	Disturbing acceleration vector
\mathbf{B}	Auxiliary matrix multiplying the state and control variations $\delta \mathbf{x}^T \dots \delta \mathbf{u}$ inside the expanded HJB equation

b	Optimal Lagrange multipliers vector
D	Auxiliary matrix multiplying the control variation $\delta \mathbf{u}$ inside the expanded HJB equation
E	Auxiliary matrix multiplying the state variation $\delta \mathbf{x}$ inside the expanded HJB equation
F	Transition function
G	Auxiliary matrix associated to the optimal cost multiplying the static parameter variation $\delta \mathbf{w}$ inside the expanded HJB equation
K	Auxiliary matrix multiplying the control and Lagrange multipliers variations $\delta \mathbf{u}^T \dots \delta \mathbf{b}$ inside the expanded HJB equation
k	Intermediate function in the Runge-Kutta explicit scheme
N	Auxiliary matrix associated to the optimal cost multiplying the state and final time variations $\delta \mathbf{x}^T \dots \delta t_f$ inside the expanded HJB equation
O	Auxiliary matrix associated to the optimal cost multiplying the state and static parameters variations $\delta \mathbf{x}^T \dots \delta \mathbf{w}$ inside the expanded HJB equation
P	Auxiliary matrix associated to the optimal cost multiplying the state variations $\delta \mathbf{x}^T \dots \delta \mathbf{x}$ inside the expanded HJB equation
Q	Auxiliary matrix associated to the optimal cost multiplying the state variation $\delta \mathbf{x}$ inside the expanded HJB equation
R	Auxiliary matrix associated to the optimal cost multiplying the Lagrange multipliers variations $\delta \mathbf{b}^T \dots \delta \mathbf{b}$ inside the expanded HJB equation
r	General position vector
S	Auxiliary matrix associated to the optimal cost multiplying the state and Lagrange multipliers variations $\delta \mathbf{x}^T \dots \delta \mathbf{b}$ inside the expanded HJB equation
T	Auxiliary matrix associated to the optimal cost multiplying the Lagrange multipliers and static parameter variations $\delta \mathbf{b}^T \dots \delta \mathbf{w}$ inside the expanded HJB equation
U	Auxiliary matrix associated to the optimal cost multiplying the Lagrange multipliers and final time variations $\delta \mathbf{b}^T \dots \delta t_f$ inside the expanded HJB equation
u	Control thrust vector
u*	Pseudo-optimal control
V	Auxiliary matrix associated to the optimal cost multiplying the static parameter and final time variations $\delta \mathbf{w}^T \dots \delta t_f$ inside the expanded HJB equation
W	Auxiliary matrix associated to the optimal cost multiplying the static parameter variations $\delta \mathbf{w}^T \dots \delta \mathbf{w}$ inside the expanded HJB equation
w	Static parameter

X	Auxiliary matrix multiplying the state and Lagrange multipliers variations $\delta\mathbf{x}^T \dots \delta\mathbf{b}$ inside the expanded HJB equation
x	Optimal state vector
Z	Auxiliary matrix multiplying the Lagrange multipliers variations $\delta\mathbf{b}$ inside the expanded HJB equation
$\tilde{\mathbf{E}}$	Gershgorin matrix
$\tilde{\mathbf{G}}$	Auxiliary matrix multiplying the static parameter variation $\delta\mathbf{w}$ inside the expanded HJB equation
$\tilde{\mathbf{L}}$	Lower triangular matrix
$\tilde{\mathbf{N}}$	Auxiliary matrix multiplying the state and final time variations $\delta\mathbf{x}^T \dots \delta t_f$ inside the expanded HJB equation
$\tilde{\mathbf{O}}$	Auxiliary matrix multiplying the state and static parameter variations $\delta\mathbf{x}^T \dots \delta\mathbf{w}$ inside the expanded HJB equation
$\tilde{\mathbf{T}}$	Auxiliary matrix multiplying the Lagrange multipliers and static parameters variations $\delta\mathbf{b}^T \dots \delta\mathbf{w}$ inside the expanded HJB equation
$\tilde{\mathbf{U}}$	Auxiliary matrix multiplying the Lagrange multipliers and final time variations $\delta\mathbf{b}^T \dots \delta t_f$ inside the expanded HJB equation
$\tilde{\mathbf{u}}$	Adimensional control thrust vector
$\tilde{\mathbf{V}}$	Auxiliary matrix multiplying the static parameters and final time variations $\delta\mathbf{w}^T \dots \delta t_f$ inside the expanded HJB equation
$\tilde{\mathbf{W}}$	Auxiliary matrix multiplying the static parameters variations $\delta\mathbf{w}^T \dots \delta\mathbf{w}$ inside the expanded HJB equation
\tilde{a}	Adimensional semi-major axis
$Isp\tilde{g}_0$	Equivalent specific impulse
\tilde{M}	Auxiliary matrix multiplying the final time variation δt_f inside the expanded HJB equation
\tilde{m}	Adimensional spacecraft mass
\tilde{t}	Adimensional time
\tilde{u}_h	Adimensional control thrust component orthogonal to the orbital plane
\tilde{u}_n	Normal tangential control thrust component
\tilde{u}_t	Adimensional tangential control thrust component

\tilde{Y}	Auxiliary matrix multiplying the final time variations $\delta t_f^T \dots \delta t_f$ inside the expanded HJB equation
a	Semi-major axis
a^{RK}, b^{RK}, c^{RK}	Butcher tableau coefficient tot apply a generic Runge-Kutta scheme
a_h	Acceleration component normal to the orbital plane due to J_2 orbital perturbation
a_r	Radial acceleration due to J_2 orbital perturbation
a_ϑ	Transversal acceleration due to J_2 orbital perturbation
c	Constant tolerance for the step-size adjustment methods
e	Orbital eccentricity
f	Axial projection of the eccentricity vector onto the nodal frame
f^i	Scalar component of the vectorial function \mathbf{f}
g	Orthogonal projection of the eccentricity vector onto the nodal frame
H	Hamiltonian function
hf	Axial projection of the nodal vector onto the orbital frame
i	Orbital inclination
i_{nom}	Nominal anomlay index
i_{opt}	Optimal anomaly index
I_{sp}	Specific impulse
J	Value function or cost functional
J^*	Optimal value function
J_{aug}	Augmented Lagrangian cost function
k	Orthogonal projection of the nodal vector onto the orbital frame
L	Running cost
l	Mean anomaly
$L_{eclipse}$	Eclipse cost function
L_{ref}	Reference length used for the adimensionalisation process
m	Spacecraft mass
m_{ref}	Reference mass used for the adimensionalisation process

N	Number of time steps
n_u	Number of control components
n_x	Number of states
p	Semi-latus rectum
s	Constant time step
t_f	Free final time
t_{ref}	Reference time used for the adimensionalisation process
tol_a	Tolerance on semi-major axis constraint violation
tol_e	Tolerance on eccentricity constraint violation
tol_f	Tolerance on axial eccentricity vector projection constraint violation
tol_g	Tolerance on orthogonal eccentricity vector projection constraint violation
tol_h	Tolerance on axial nodal vector projection constraint violation
tol_i	Tolerance on orbit inclination constraint violation
tol_k	Tolerance on orthogonal vector projection constraint violation
tol_Γ	Tolerance on true longitude vector projection constraint violation
tol_ν	Tolerance on true anomaly constraint violation
tol_Ω	Tolerance on RAAN constraint violation
tol_ω	Tolerance on pericentre constraint violation
tol_ϑ	Tolerance on the error prediction of the optimal cost functional
tol_{con}	Tolerance on the constraints violation
u_h	Control thrust component orthogonal to the orbital plane
u_n	Normal control thrust component
u_t	Tangential control thrust component
w_u	Weighting coefficient for the running cost
M	Auxiliary variable associated to the optimal cost multiplying the final time variation δt_f inside the expanded HJB equation
Y	Auxiliary variable associated to the optimal cost multiplying the final time variations $\delta t_f \dots \delta t_f$ inside the expanded HJB equation

Greek Symbols

α	Optimal constant coefficient matrix of the feedback control law
β	Optimal coefficient matrix for the state variation of the feedback control law
η	Optimal coefficient matrix for the static parameters variation of the feedback control law
γ	Optimal coefficient matrix for the Lagrange multipliers variation of the feedback control law
Φ	Auxiliary matrix multiplying the control and static parameters variations $\delta \mathbf{u}^T \dots \delta \mathbf{w}$ inside the expanded HJB equation
Π	Auxiliary matrix multiplying the control and final time variations $\delta \mathbf{u}^T \dots \delta t_f$ inside the expanded HJB equation
ρ	Optimal coefficient matrix for the final time variation of the feedback control law
φ	Endpoint constraints function
ξ	Orbital elements vector
Γ	True longitude
λ_{min}	Minimum eigenvalue of the control Hessian matrix
μ	Gravitational parameter
ν	True anomaly
Ω	Right Ascension of the Ascending Node
ω	Pericentre anomaly
ω_{\oplus}	Earth mean angular velocity
Φ	Eccentric longitude
ϕ	System dynamics flow
σ	Penalty parameter
τ	Adimensional time to solve the free final time problem
ε_c	Constant shift
ε_1	Tuning coefficient of the constant optimal feedback control variation
ε_2	Tuning coefficient of the optimal Lagrange multiplier variation
ϑ	Error between the optimal cost functional and nominal cost functional
ζ	Generic angle

Subscripts

1	Initial time step
\odot	Astronomical symbol of the Sun
\oplus	Astronomical symbol of the Earth
<i>con</i>	Constraints
<i>cone</i>	Conical
<i>ell</i>	Ellipse
<i>f</i>	Final time step
<i>j</i>	Single component of a vector
<i>max</i>	Maximum
<i>min</i>	Minimum
<i>nom</i>	Nominal
<i>opt</i>	Optimal
<i>orb</i>	Orbit
<i>ref</i>	Reference
<i>RK</i>	Runge-Kutta

Superscripts

<i>j</i>	Step <i>i</i> of the DDP algorithm
T	Transpose operator

Mathematical Notation

\square	Total time derivation of the variable \square
$\frac{\partial}{\partial \square}$	Partial derivation with respect to the variable \square
$\frac{d}{d \square}$	Total derivation with respect to the variable \square
\min_{\square}	Minimisation problem
$\nabla \mathbf{f}_{\square}$	Jacobian matrix of the function \mathbf{f} with respect to the variables $\square \square$
∇f_{\square}	Gradient of the scalar function f with respect to the variables $\square \square$

- \otimes Krocker tensor product
- \mathbf{f}_{\square} Jacobian matrix of the function \mathbf{f} with respect to the variable \square
- $\text{Mat}(\square)$ Function turning a multi-dimensional arraya into a bi-dimensional matrix
- $\text{Vec}(\square)$ Function turning a multi-dimensional arraya into a column vector
- f_{\square} Partial derivative of the scalar function f with respect to the variable \square
- $\mathbf{f}_{\square\square}$ Hessian matrix of the function \mathbf{f} with respect to the variables $\square\square$
- $f_{\square\square}$ Double partial derivative of the scalar function f with respect to the variables $\square\square$

List of constants

- Astronomical Unit distance $\text{AU} = 1.495978707 \times 10^8 \text{ km}$
- Earth eccentricity $E_{\oplus} = 0.081820368414195$
- Earth equatorial radius $R_{\oplus} = 6378.1363 \text{ km}$
- Earth flattening $f = 1/298.257223563$
- Earth gravitational constant $\mu_{\oplus} = 398600.4415 \text{ km}^3/\text{s}^2$
- Earth mean angular velocity $\omega_{\oplus} = 7.292115 \times 10^{-5} \text{ deg/s}$
- Earth polar radius $R_{pol} = 6356.751 \text{ km}$
- Standard-free fall $g_0 = 9.80665 \text{ m/s}^2$
- Sun gravitational constant $\mu_{\odot} = 1.3271244 \times 10^{11} \text{ km}^3/\text{s}^2$
- Universal gravity constant $G = 6.67259 \times 10^{-20} \text{ km}^3/(\text{kg} \cdot \text{s}^2)$

List of acronyms

- COMPASS Control for Orbit Maneuvering through Pertubations for Applications to Space Systems
- DDP Differential Dynamic Programming
- ERC European Research Council
- FET Finite Elements in Time
- GEO Geostationary Earth Orbit
- GTO Geostationary Transfer Orbit
- HDDP Hybrid Differential Dynamic Programming

HJB	H amilton- J acobi- B ellman
IAC	I nternational A stronautical C ongress
JPL	J et P ropulsion L aboratory
KKT	K arush- K uhn- T ucker
LEO	L ow E arth O rbital
MEE	M odified E quinoctial E lements
MEO	M edium E arth O rbital
MJD	M odified J ulian D ate
MPBVP	M ulti P oint B oundary V alue P roblem
NASA	N ational A eronautics S pace A dministration
NEO	N ear- E arth O bject
NLP	N on- L inear P rogramming
ODE	O rdinary D ifferential E quation
PMP	P ontryagin M aximum P inciple
RAAN	R ight A scension of the A scending N ode
RK	R unge- K utta
RKF	R unge- K utta- F ehlberg
SQP	S equential D ynamic P rogramming
SRP	S olar R adiation P ressure
STM	S tate T ransition M atrix

CHAPTER 1

Introduction

Low-thrust propulsion subsystems represent the baseline for many of the future planetary and interplanetary space missions. This propulsion choice affects the design of spacecraft trajectories which are the result of a complex non-linear optimal control problem. This thesis will propose an effective strategy to solve in an innovative way non-linear optimal control problems in terms of orbital elements. In this chapter we will introduce the motivations and objectives of the study. Subsequently, a summary of the current state of the art on numerical techniques used to solve non-linear optimal control problems will be given. A brief discussion of direct and indirect approaches, in order to highlight the reasons which led us to the selection of the differential dynamic programming technique for the design of low-thrust optimal trajectories is also presented. An overview of low-thrust past and future space missions is also given to justify the possible applications of the work presented in this thesis. A careful literature review about the differential dynamic programming together with all its past applications is presented to enhance the novelty of the proposed methodologies with respect to the literature in this area. Finally, a summary of the novel contribution developed and implemented in this study is provided.

1.1 Design of optimal spacecraft trajectories

Every space mission requires to define in its design an operative trajectory for one or more spacecraft according to the main scientific requirements. After the assessment of the operative trajectory a very important task in the preliminary design of the space mission is to derive the transfer trajectory that will move the spacecraft from the launch base to the nominal orbit. This design is very difficult because it must satisfy multiple constraints that are defined in a concurrent design engineering approach. Some examples of relevant design constraints can be provided by the telecommunication subsystem which requires the spacecraft to communicate with the ground stations during specific phases of the space missions such as during flybys [1]. The consequence on the design of the trajectory is to be kept always in view of the relay spacecraft and/or not having the Sun directly behind the Earth, which is introducing noise in the communication [2]. Another relevant constraint is related to the power subsystem, which can require the spacecraft to be always in sunlight or satisfy a maximum time to be in the shadow of the eclipse due to the storage capability of the onboard batteries. In this case the trajectory of the spacecraft must be adjusted to avoid the shadow created by the Sun and a generic occulting body

during its transfer. The use of multiple flybys to change the size of the orbit moving towards the outer or inner planets is also a constraint in the design of the spacecraft trajectory to overcome some limitations imposed on the maximum thrust provided by the propulsion subsystem or by the payload subsystem when several scientific instruments are required to be onboard.

The main problem to be solved from a mathematical point of view is to design an optimal trajectory for a spacecraft which is minimising a specific or multiple cost functions subject to several additional internal or final constraints. This problem has been considered for many decades and it is still at the moment a very important topic to be discussed thanks to the advancement of computer computational power and development of numerical optimisation techniques. The first attempts in solving this problem involved only the use of multiple ballistic arcs separated by impulsive manoeuvres. In the last years the use of low-thrust technologies is imposing the definition of a continuous control law for the design of the optimal trajectory. The inclusion of all the constraints was not always possible in the design of the optimal trajectory and the result that the satisfaction of some conditions have to be checked a posteriori. This is why the literature offers many research papers related to optimal design of spacecraft trajectories dealing with both the optimisation technique itself or the way to improve the quality of the design considering the inclusion of constraints. One of the objectives of this research is to contribute in both these directions to the several methods present in the current literature highlighting the main advantages and the possible applications.

1.2 Research motivations and objectives

The aim of this research is to investigate the use of the differential dynamic programming technique to solve low-thrust optimal control problems where the trajectory is written in terms of orbital elements as state representation of the dynamics. The formulation of the differential dynamic programming has been discovered in the second half of the XX century. However, the applications of such technique were limited mainly to solve reservoir problems [3],[4],[5][6]. The first and only application of differential dynamic programming to a space mission has been carried out for the NASA Dawn mission [7] to design the interplanetary trajectory of the spacecraft [8]. Both the two previous examples employed Cartesian coordinates as state representation of the dynamics. In the last years the differential dynamic programming started to be used as numerical tool to solve a trajectory optimal control problem for space applications. However, the applications were still limited to Cartesian coordinates as state representation of the dynamics. For celestial mechanics problem orbital elements represent a valid alternative to Cartesian coordinates to represent the satellite state. This is true for many common astrodynamics problems such as orbit propagation, eclipses determination and constellation coverage because of the strict relation between orbital elements and geometry.

The first objective of this research is to study and extend the use of differential dynamic programming with orbital elements representation as alternative optimisation technique to solve the spacecraft trajectory design problem. In this framework the target is to solve multiple revolution ellipses transfer with low-thrust which has hardly been done with DDP so far. The second objective is related to the inclusion of eclipse as internal path constraint during the optimal design of the spacecraft trajectory. The eclipse represents always a hard constraint to be studied in the design of the spacecraft trajectory and the compliance with this requirements is usually verified

after the design of the trajectory. The real bottleneck leading to this approach which is not optimal is the lack of a systematic way to include eclipse for each time instant in the optimisation process. All the eclipse algorithm developed in literature consider a fixed spacecraft orbit for the determination of the entry and exit umbra points. However, spacecraft equipped with low thrust engines are subject to continuous accelerations which are changing the orbital elements at each time instant making the existing eclipse algorithm not reliable. This is why the second main topic of the research is to fill this hole creating a systematic way to include the eclipse path inside the optimisation process so that the spacecraft trajectory accounts for the presence of the shadow during its transfer. The last objective of the research is to define a wide variety of space mission opportunities that can be designed using the optimisation technique presented in this dissertation. A comparative assessment of the proposed optimisation technique is compared with other methods proposed in literature to evaluate efficiency, advantages and disadvantages according to the type of problem that is solved.

1.3 Background

1.3.1 Numerical methods for trajectory optimisation

In this section an overview of the classic methods to design a spacecraft trajectory is presented. In particular, most of the methods can be classified under direct and indirect approaches [9]. However, other different techniques such as the shape-based method are used as a valid alternative for the definition of a initial suboptimal solution.

Direct methods

A direct method is an optimisation technique that transcribes a general non-linear optimal control problem into a non-linear programming problem so that the objective function is "directly" optimised. The transcription process is the process that converts the state or both the state and control variables into parameters which are selected from finite dimensional spaces. Imaging to consider an univariate function $F(x)$, a direct method constructs a sequence of points x_1, x_2, \dots, x^* such that the objective function is minimised and typically $F(x_1) > F(x_2) > \dots > F(x^*)$ [20]. At least in principle, the direct method needs only to compare values for the objective function so that it attempts to find its minimum. A direct method does not need to write the adjoint equations, the control equations, or the transversality conditions. After the transcription, the direct method uses numerical algorithms (for instance, sequential quadratic programming, penalty methods, augmented lagrangian) to compute candidate points which minimise the non-linear programming problem. The candidate points are not generally the real optimal solution. The candidate solution is refined computing the corrections which satisfy the first-order necessary optimality conditions also known as Karush-Kuhn-Tucker (KKT) conditions [10].

Single shooting, multiple shooting and collocation methods are widely used to solve numerically the optimisation problem. In the single shooting only the control variables are discretised and the optimisation is carried out through all the time interval. In the multiple shooting technique the time domain is partitioned in many connected segments and in each segment the

single shooting approach is applied [11]. Finally, collocation methods discretise both state and control variables and transform differential equations into non-linear constraints [12]. Several Sequential Quadratic Programming (SQP) solvers exist based on the direct collocation method which automatically transforms the non-linear optimal control problem into a corresponding non-linear programming problem and apply state-of-the-art method to solve the quadratic sub-problems. Example of NLP solvers are SNOPT [13], IPOPT [14], SLSQP [15], KNITRO [16], GPOPS [17], PSOPT [18], LOQO [19]. The advantage in using direct methods is the possibility to handle easily complicated constrained non-linear problems and to provide a solution also when a poor initial guess is available [20]. However, these methods suffer multiple-revolution trajectories because a large number of variables is usually required to obtain an accurate solution and cannot guarantee an optimal solution.

Many applications of direct methods are present in literature to solve trajectory optimal control problems. A direct trajectory optimisation method using non-linear programming and collocation method using cubic polynomials to represent state variables and linear interpolation for the control variables is presented by Hargraves and Paris [21]. The same collocation is used by Tang and Conway [22] to solve interplanetary optimal trajectories. Gath and Well [23] present a software package called GESOP which is a combination of direct multiple shooting method and direct collocation method. The collocation phase adopts either a third order Hermite polynomials or a trapezoidal approximation. Fahroo and Ross [24] describe a direct trajectory optimisation based on Chebyshev pseudospectral method. In this case the Lagrange polynomial approximations for state and control variables are evaluated using the Chebyshev method. Other applications of direct collocation methods to solve low-thrust trajectory optimal control problems are analysed by Herman and Spencer [25], and Somaverrapu et al. [26]. Vasile and Bernelli [27] present an original direct optimization approach has been used to design an optimal interplanetary trajectory. The proposed approach is characterized by a transcription of both states and controls by Finite Elements in Time (DFET). A set of additional parameters, not included among states and controls, are allowed and can be used for a combined optimization of both the trajectory and other quantities peculiar to the original optimal control problem (parametric optimization). In particular, in this paper, the orbital elements of each swing-by hyperbola are treated as additional parameters and swing-by trajectories are not transcribed with collocation but using multiple shooting. Direct methods can be also combined with other optimisation techniques such as genetic algorithm to create an hybrid approach as discussed by Vavrina and Howell [28].

Indirect methods

An indirect method differs from its counterpart because it attempts to find a root of the first order necessary condition $F'(x) = 0$. Therefore, an indirect method must compute the slope $F'(x)$ and decide if it is sufficiently close to zero. The optimal control necessary conditions requires to explicitly derive the adjoint equations, the control equations, and all of the transversality conditions. In this case, the optimal control problem is transformed in a Multipoint Boundary Value Problem (MPBVP) as a consequence of the application of Pontryagin Maximum Principle (PMP) which represents the first-order necessary conditions that an optimal solution must satisfy [29]. At this point the necessary conditions are discretised so that a constrained parameter optimisation problem is obtained. The optimal control is the solution of

the MPBVP which requires the application of Newton-like algorithm. Also in this case single shooting, multiple shooting and collocation techniques can be applied to indirect methods.

The main advantages in using an indirect method is in the quality of the solution obtained that is accurate and it is guaranteed to be optimal because of the satisfaction of the first-order necessary conditions. However, the method is very sensitive to the initial choice of the costates and it is not obvious which could be a proper selection for their choice due to their non-physical interpretation. Indirect methods have been used for many applications starting from the optimisation of rocket ascent trajectories as presented by Colasurdo et al. [30]. Fahroo and Ross [31] analyse a hybrid approach combining a direct method with an indirect collocation technique. A different combination using genetic algorithm which provide a tentative guess for the subsequent indirect optimisation is investigated by Sentinella and Casalino [32].

Ranieri and Ocampo [33] introduce an indirect optimisation to compute the minimum propellant spiral escapes and captures in terms of spherical coordinates. Lagrange multipliers are estimated in a two steps. In the first step, an adjoint control transformation converts the thrust direction to the multipliers controlling the trajectory, while in the second phase curve fits are matched with the values of the initial multipliers. La Mantia and Casalino [34] proposes a different approach to optimise low-thrust capture trajectories using Cartesian coordinates and Edelbaum's approximation to model the suboptimal spiral phase. Bokelmann and Russell [35] solve an optimisation problem for Europa capture trajectory using primer vector theory and impulsive manoeuvres. Other applications of indirect methods are related to the optimisation of low-thrust trajectories with multiple gravity assists as presented by Casalino et al. [36], and deflection missions for the design of a kinetic impactor trajectory as discussed by Casalino and Simeoni [37].

Differential dynamic programming

This research uses a Differential Dynamic Programming (DDP) algorithm as optimisation technique to solve the low-thrust optimal control problem. The first derivation of the DDP algorithm was presented by Mayne [38] as a second-order gradient method to determine optimal trajectories for non-linear discrete time systems. The formulation has been expanded by Jacobson and Mayne [39] providing a detailed description of the DDP algorithm also for continuous time systems considering both global and weak variations of the control action. Gershwin and Jacobson [40] improved the constrained DDP algorithm introducing some checking conditions to verify if the update of the Lagrange multipliers can be consistent with the second-order Taylor expansion. The formulation of the DDP algorithm was investigated and further developed by many authors for many years [41], until it reached a first stop with the definition of the Static/Dynamic Control (SDC) approach introduced by Whiffen [8]. This technique was the first one used to design an interplanetary space mission (NASA Dawn) and represented the state-of-the art for several years. The advance in computer technology made DDP a research topic starting from the first decade of 2000. In particular, Colombo et al. [42] analysed in their work a stage-wise approach to incorporate an adaptive refinement of the discretisation mesh within the optimisation process based on interpolation techniques. The most important development related to DDP was provided by Lantoine and Russell [43] which introduced the Hybrid Differential Dynamic Programming (HDDP) where the partial derivatives are propagated using State Transition Matrices (STMs) and it is particularly useful for multi-phase problems defin-

ing the new state-of-the art for this type of optimisation technique. Maestrini [44] extended the HDDP applying differential algebra to compute the STMs as polynomial maps without solving numerically the ordinary differential equations. The DDP algorithm was extended to stochastic problems thanks to the work of Campagnola et al. [45] that generalised in their work the classic DDP algorithm assuming the existence of state and control multiplicative process noises.

The DDP deserves a separate class as numerical technique for the solution of non-linear optimal control problems because it represents a mixture between the benefits and disadvantage of both direct and indirect methods. The DDP shares with the direct methods the ability to handle also poor initial guesses and it does not require a first guess solution for the adjoint variables. As the indirect methods it can be proved that DDP is strongly related to PMP which represents the starting point for the derivation of the first-order necessary optimality conditions. The main feature of the DDP algorithm is the ability to derive a feedback control law as result of the optimisation that is granting robustness to the optimal control law, and the possibility to perform real-time corrections if unknown disturbances have to be included in the dynamics.

The DDP can be summarised as a practical solution to the so called "curse of dimensionality" problem [46] inherent with the basic formulation of the dynamic programming that consists in the rapid growth of memory and computational time as the dimension of the problem increases. This creates a limitation in the application of the dynamic programming to very complicated non-linear optimal control problem. The DDP is based on a successive backward quadratic Taylor expansion of the cost functional in the neighbourhood of a nominal and non-optimal trajectory. The successive backward expansion allows to transform the large scale non-linear optimal control problem into a series of problems of small dimension. This characteristic makes the DDP an attractive alternative to classic Non-Linear Programming (NLP) solvers when highly dimensional problems are considered. The optimisation process in the DDP algorithm can be summarised in a backward sweep where the coefficients of the feedback control law are determined solving Hamilton-Jacobi-Bellman (HJB) equation, and a forward integration where the new control law is used to propagate the system dynamics to check the reduction in the cost functional [39].

Shape-based optimisation

Another alternative for the design of spacecraft trajectories is the shape-based method. In this case an analytical solution for motion variables for a given class of trajectory shape functions is provided. Once the trajectory shape is selected a priori, the thrust profile is derived a posteriori from the inverse solution of the equations of motion. The parameters of the shape function can be computed thanks to an optimisation algorithm. The outcome of the method is not guaranteed to be an optimal solution but it can be used as initial guess for more complicated optimisation methods.

The shape-based method idea is to provide an approximate analytical solution to model the low-thrust spacecraft trajectories. The first important reference is described by Petropoulos and Sims [47] that presented a review of shape-based techniques. However, only basic examples involving constant radial acceleration and planar motion are solved. De Pascale and Vasile [48] have developed a three-dimensional shape-based method using pseudoequinoctial elements ca-

pable of handling also boundary constraints. This allows to apply the technique also to solve low-thrust rendezvous trajectories and gravity assist manoeuvres. Petropoulos and Longuski [48] extended the work presenting a shape-based algorithm for automated design of low-thrust trajectories including gravity assists. Some selected trajectories from the shape-based method are considered as initial estimates for an optimisation program employing direct methods. Wall and Conway [49] discusses a new shape-based technique for low-thrust rendezvous trajectories using a 6th degree inverse polynomial for the in-plane motion and a polynomial with four coefficients for small out-of-plane motion [50]. A further improvement of the shape-based technique is provided by Novak and Vasile [51] who developed a general three-dimensional shaping method in spherical coordinates. They show that the planar exponential sinusoid and the inverse polynomial shaping are just two-dimensional particular cases of the three-dimensional approach. The solution obtained from the shaping method is improved considering a feedback linear-quadratic controller. Finally, Vellutini and Avanzini [52] proposes a shape-based design of low-thrust trajectories for spacecraft transfers from low-Earth orbit (LEO) to Lagrangian point L_1 of the Earth-Moon system assuming the two primary masses on a circular relative orbit.

1.3.2 Orbital elements as state representation

This research investigates the coupling of DDP optimisation algorithm with orbital elements as state representation of the dynamics. Orbital elements represents one of the main keywords of this dissertation because of the poorness of this topic in literature and the choice of problem to be solved. The non-linear optimal control problem that must be solved is the design of a spacecraft trajectory. This means that orbit dynamics is characterising the equations of motions and they do not admit a single formulation in terms of Cartesian coordinates but also in orbital elements that is rarely used. However, orbital elements represent a valid alternative to represent the orbit dynamics because they offer more physical insight about the orbit geometry (e.g., orbit shape, orbital plane inclination) and its time evolution. Cartesian coordinates have not the same power and no information about the orbit geometry is available until the three-dimensional trajectory is plotted.

The current literature presents few examples of trajectory optimisation in terms of orbital elements as state representation of the dynamics. The first relevant example is provided by Betts [53] which uses a collocation method applied to modified equinoctial elements. This technique is slitted in three operations which are a direct transcription into a NLP problem, the solution of sparse NLP using sequential programming, and a mesh refinement. This technique was applied to a very low-thrust problem which required 416123 variables and 249674 constraints. The second reference can be found in Petropoulos [54] who introduces the so-called "Q-law" as simple approximation or initial guess for propellant-optimal non-linear optimal control problem. This method is also based on modified equinoctial elements and consists in identifying a candidate Lyapunov function whose properties are check according to the value of the proximity quotient. A different approach based on the analytic orbit averaging technique for simulating many-revolution trajectories using low tangential acceleration is discussed by Gao [55]. Liu and Tongue [56] provide a complete treatment of escape trajectory design making use of Modified Equinoctial Elements (MEE)-based adjoint equations incorporating the third-body and higher-order Earth gravitational perturbations. This work exploits a classic indirect method as optimisation technique to minimise the transfer time [56].

A first-order approximated analytical solution of Gauss planetary equations is analysed by Avanzini et al. [57] that is integrated in a direct transcription method based on the decomposition of the trajectory into a number of finite elements. Gauss' planetary equations are solved over each element by means of a perturbative approach, for constant thrust modulus and direction. Another application of indirect methods to solve multiple coast arcs optimal space trajectories in terms of modified equinoctial elements is investigated by Pontani [58] who derives the complete set of necessary conditions for optimality to minimise the transfer time in the presence of eclipse conditions. The only application of orbital elements to HDDP is formulated by Aziz et al. [59] who introduces a Sundman transformation to change the independent variable of the spacecraft equations of motion from time to an orbit angle. A Geostationary Transfer Orbit (GTO) where modified equinoctial elements are used as state representation is solved. In particular, the optimal control problem is solved targeting a final orbit but not a specific position on the final orbit, that is equivalent to impose a constraint on the final angular variable used as independent variable.

1.3.3 Eclipse algorithms

In this work a new analytical eclipse algorithm is investigated for the computation of entry/exit points of a spacecraft from the umbra shadow. There are many works dealing with the analysis of the eclipses in literature. The first reference can be found in Escobal [60], who defines an analytical procedure to determine the true anomalies corresponding to the entry point and exit point from an eclipse in the framework of classic Keplerian elements assuming the Earth's surface as a perfect sphere and a cylindrical shadow. These assumptions simplify the modelling from a three dimensional problem to a planar one due to the spherical symmetry and it results in a quartic equation where the unknown is the cosine of the entry/exit true anomalies. Escobal suggests also a procedure to handle the same problem modelling the Earth's surface as an oblate ellipsoid of rotation, but this time the algorithm is numerical, iterative and it gives the possibility to define the penumbra region starting from user-made assumptions related to the amplitude of the region itself.

Vallado [61] presents a numerical shadow analysis for both cylindrical and conical cases starting from the same assumptions and shadow function developed by Escobal and solving the quartic polynomial equation in the true anomaly with a Newton-Raphson numerical scheme, and so no further modelling is added to the solution of the problem. Fixler [62] introduces an analytical procedure to determine the umbra and penumbra regions assuming a conical shadow and a spherical Earth resulting in another transcendental equation to be solved numerically. However, this method is based on projection of the Sun position vector onto the satellite orbital plane which modifies slightly the real geometry of the problem. Kraft [63] solves the same problem arriving at the results given by Fixler by a different derivation. The angular displacement of the Sun direction with respect to the orbit pericenter is used by Colombo and McInnes [64] to include eclipse in the orbit dynamics expressed in terms of orbital elements assuming a planar motion in the ecliptic plane. Ortiz Longo et al. [65] extended the methods developed by Fixler and Kraft to a perturbed environment taking into account the effect of J_2 orbital perturbation. Montenbruck and Gill [66] used a spherical Earth conical shadow model based on the angular separation and diameters of the Sun and the Earth. A more complete analysis has been carried out by Dreher [67] considering also the effect of the atmospheric refraction on the light rays, but

the modelling is always based on an unperturbed environment and a spherical Earth. Vokrouhlicky et al. [68] proposed the concept of "osculating spherical Earth" to account for the errors introduced by assuming a spherical model with respect to an oblate Earth's surface. Adhya et al. [69] are the first ones introducing an analytical procedure for the eclipse phenomenon modelling the Earth's surface as an oblate ellipsoid of rotation. Their methodology can only be used to state if the satellite is in light, penumbra or umbra region and it is limited for low-Earth orbiting satellites. However, a numerical investigation by Vokrouhlicky et al. [70] has shown that the oblateness of the Earth did not bring significant differences compared to a spherical Earth for the LEO satellites. On the contrary, Woodburn [71] showed that the cylindrical assumption which neglects totally the penumbra region has important consequences for the precise numerical integration of orbit trajectories depending on the numerical scheme and the definition of the boundaries to be used for the occulted region.

1.4 Methodologies developed and implemented

The present research focuses on the design of optimal low-thrust spacecraft trajectories using the DDP numerical technique based on orbital elements as state representation of the dynamics including eclipse as internal path constraint.

The orbit dynamics is modelled through Gauss' planetary equations to account both for conservative and non-conservative accelerations. A discrete DDP algorithm based on small control variations is used as optimisation technique and a new analytical algorithm is used to model the eclipse through a conical shadow to minimise the total time spent by the spacecraft in the umbra during the transfer. The DDP technique used in this dissertation modifies the algorithm proposed in previous works [43],[42],[59], in which the main differences are related to the use of strong variations instead of weak variations, or other computational strategies for the cost functional partial derivatives and the choice of the state representation.

In this work, the transition function used for the discretisation of the dynamics and of the DDP algorithm is Runge-Kutta 8th order explicit scheme. The evaluation of the cost functional partial derivatives is carried out considering the analytical derivative of the numerical scheme. The model is parametrised to account any kind of system dynamics and disturbance acceleration if an analytical formulation is provided thanks to the use of MATLAB symbolic manipulator. Multiple endpoint constraints formulation are proposed to solve the optimal control problem according to the type of transfer. Heuristics techniques are defined to improve the convergence of the DDP algorithm and a new Hessian shifting technique is proposed to make the choice of the diagonal shifting user independent. Multiple-revolution transfers with more than 500 revolutions are solved for the first time using the DDP algorithm thank to the coupling with orbital elements.

A complete description of the theory coupling DDP algorithm with orbital elements denoted as *quasi-independence* theory is presented in Chapter 4. First, the orbital dynamics expressed in terms of Gauss' planetary equations is shown to be stiff because of the relative magnitude in the orbital elements rates. The stiff dynamics has consequences on the application of the DDP algorithm when a rendezvous optimal control problem is solved. Indeed, the algorithm tries to correct and reduce the violation related to the fast variable without leading the spacecraft to

the proper and correct final target orbit. The formulation of the dynamics in terms of orbital elements results also in a second bottleneck that is the difficulty to optimise both slow variables and fast variables at the same time. From a numerical point of view the consequence of trying to optimise all the orbital elements at the same time is the presence of ill-conditioned matrices that need to be inverted to define the optimal feedback control law. The two phenomena can be explained looking at the expressions of the pericenter anomaly and true anomaly rates. Imaging that the out-of-plane policy is very small the Jacobian matrices of the two orbital elements rates with respect to the control are opposite. Therefore, if a given optimal control policy minimises the pericenter anomaly, on the other hand it will maximise the true anomaly. It is not possible to minimise or maximise the pericenter anomaly and the true anomaly at the same time. This occurs because the change in true anomaly is not the direct effect of the application of the control law, but it is caused by the change of the eccentricity vector which is the reference axis used for the measurement of the true anomaly. For all the other orbital elements it is possible to apply a control action which produces a variation of a single orbital element that is independent from the others. Such phenomenon occurs only in orbital elements because they provide a better insight of the orbit dynamics and they are revealing an aspect that is actually hidden when Cartesian coordinates are used as state representation of the dynamics. Indeed, if the initial condition of the spacecraft and the nominal control law are showing that the pericenter anomaly and the fast variable must not follow the same trend to achieve the target it is better to choose a different combination of initial condition and control law. However, if a rendezvous problem must be accomplished and the initial conditions cannot be changed a continuation scheme is proposed in two main layers to solve the problem. The first layer solves the optimal control problem considering only the endpoint constraints associated to the slow orbital elements so that an optimal control law is defined to reach the target orbit. In the second layer the optimal control law derived in the first layer is used as initial guess to solve a new optimal control problem to target a new final anomaly. This novel approach allows to reduce the magnitude of the fast variable variation and make it comparable to the other elements avoiding the creation of ill-conditioned matrices. In case the design involves multiple revolutions, this dissertation suggests to change the independent variable from the absolute time to the eccentric anomaly to improve the refinement of the discretisation and the convergence properties of the algorithm. It is shown that the choice of time in multiple-revolution problems creates discontinuities in the evaluation of the cost functional partial derivatives in the backward sweep due to the large distance between two consecutive nodes. The change of variable is accomplished by using a Sundman transformation that is not making more difficult the dynamics formulation because Gauss' equations can be expressed as function of the eccentric anomaly. Differently from a Cartesian coordinates transformation, the Sundman transformation [59] allows to be consistent with the type of reference set of coordinates without the need to transform at each time step eccentric anomaly in time to solve the dynamics. This is another main advantage of the use of the orbital elements that is highlighted in this dissertation. When highly non-linear dynamics are considered as in the case of Gauss' planetary equations the use of STMs as done in the HDDP is not advised because of the complexity in the numerical solution of the partial differential equations. This is why a comparative assessment is presented to prove that the cost in deriving the analytical formulation of the partial derivatives of the explicit numerical scheme bring many benefits from a computational point of view and it can be applied to any kind of system dynamics.

The second main research objective is related to the inclusion of an internal path constraint prescribing a maximum time for the spacecraft to be in the shadow of a generic eclipse. A new

analytical cost function is derived as result of the research to optimise the maximum shadow time during the spacecraft transfer. First, a new analytical method to account for the eclipse umbra entry and exit point is derived considering a conical shadow and an oblate ellipsoid of rotation. Using the same approach to model analytically the conical surface of the shadow a forbidden region within the eclipse umbra is generated according to the maximum prescribed shadow time. The minimum distance of the spacecraft from this forbidden region is defined as figure of merit to be the eclipse cost function that must be added in the general Augmented Lagrangian formulation. The minimisation of this cost function is equivalent to the minimisation of the distance of the spacecraft from the forbidden region that if it is close to zero mean that the spacecraft respects the maximum shadow time. The formulation is first derived in terms of Cartesian coordinates and then it is adapted to use orbital elements. Moreover, the definition of the proposed eclipse cost function is not strictly related to the DDP algorithm but can be used in any kind of numerical optimisation technique.

1.5 Structure of the thesis

The dissertation is divided into nine chapters, which introduce different topics regarding low-thrust optimisation. At the beginning of each chapter there is a brief introduction explaining the theoretical content developed in this thesis and the applications used as results of the methodology introduced. Chapter 2 introduces the several formulations of the orbit dynamics used for all the results derived in this research. The chapter continues with the definition of the analytical formulations for the computation of the partial derivatives of a generic Runge-Kutta explicit scheme. The last part of the chapter is devoted to present the structure of the DDP algorithm to handle a parametrised dynamics and perturbation. In the Appendix A the complete derivation of all the Gauss' equations expressing the orbit dynamics in several reference frames and in terms of many sets of orbital elements is contained.

In Chapter 3 the main topic is the differential dynamic programming algorithm. The formulation of the optimal control problem is presented and the analytical formulation of all the backward equations is derived starting from Bellman's principle of optimality. The chapter describes also some heuristic techniques that come from the experience of the author to solve different types of optimal control law trajectories. In the final part of the chapter, the different ways to formulate the constraints are presented highlighting the main difference between the formulations and the summary of the DDP algorithm is proposed.

Chapter 4 deals with the most important theoretical contribution of this research work. A complete mathematical framework, denoted as *quasi-independence* theory, of what happens when orbital elements are used as state representation of the dynamics to apply DDP is derived. The chapter continues describing how to handle the angular variables which are not present in the classic state representation in terms of Cartesian coordinates. A continuation scheme is proposed to overcome the bottleneck when rendezvous problem must be solved in terms of orbital elements and a numerical example is presented to show the effectiveness of the methodology.

The dissertation expands the basic DDP algorithm presented in Chapter 3 introducing the optimisation of free final time and static parameters in Chapter 5. Starting from the same assumptions the complete backward equations are derived. From this complete formulation all

the simplified versions of the DDP algorithm can be defined according to the variables to be optimised. The second part of the chapter presents the Sundman transformation to change the independent variable from time to eccentric anomaly and explains its advantages in the application of the DDP algorithm. This tool is particularly relevant for the solution of multi-revolution optimal trajectory design.

Chapter 6 presents a comparative analysis between the state-of-the art method for the evaluation of the cost functional partial derivatives. This operation deserves a separate chapter because it represents the most expensive part in the execution of the DDP algorithm and a simplification of this part provides a significant advantage in the overall algorithm execution. The last part of the chapter is devoted to a robustness analysis of the optimal control law derived by the DDP algorithm. Starting from the feedback control law obtained by the DDP a perturbation is introduced in the initial conditions to check if the optimal control law is still reliable or a new optimal control problem must be solved.

The second important aspect of the dissertation is contained in Chapter 7 which describes an innovative way to include an internal path constraint associated to the eclipse in the DDP algorithm. The chapter starts with the analytical derivation of a new algorithm for the eclipse entry and exit point evaluations and end with a figure of merit to minimise the maximum prescribed shadow time. This novelty is very flexible because the formulation of the cost function can be used also with different optimisation methods. Some results are presented at the end of the chapter to show the effectiveness of the new figure of merit.

Finally, Chapter 8 summarises the contributions of this dissertation and proposes several insights to improve the DDP algorithm in terms of orbital elements both from the theoretical and applicative point of view that will be subject of future work.

1.6 Contributions

The contents of this dissertation are going to be published in two journal papers and have been presented in several international conferences. An introduction to the DDP based on orbital elements has been presented at the 70th International Astronautical Congress (IAC) in 2019 at Washington [72]. A first part of the theoretical framework coupling DDP algorithm and orbital elements was discussed in the 31st AAS/AIAA Space Mechanics Meeting [73] that was held remotely due to the Covid-19 pandemics restrictions. The full derivation of the *quasi-independence* theory has been described in the paper submitted to the Journal of Guidance, Control, and Dynamics [74]. The analytical derivation of the eclipse algorithm was presented first at the 33rd AAS/AIAA Space Mechanics Meeting [75] in 2022 at Charlotte and the full derivation of new approach together with the differences with respect to the state-of-the art methods and some applications is the main topic of a new journal publication submitted to the Journal of Guidance, Control, and Dynamics [76]. The integration of the eclipse algorithm with the DDP method was part of the research stay at Jet Propulsion Laboratory (JPL) and the outcome of the research have been discussed at the 73rd IAC in 2022 at Paris [77]. The full description of the cost function derivation together with some relevant applications involving eclipse in the design of low-thrust optimal trajectories will be part of a new journal publication co-authored with Dr. Gregory Lantoine who supervised the research stay at JPL.

Journal Publications

- Nugnes, M. and Colombo, C., “Low-thrust trajectory optimisation problem mathematical theory using differential dynamic programming based on orbital elements”, *Journal of Guidance, Control, and Dynamics*, 2022 (submitted for peer review).
- Nugnes, M. and Colombo, C., “Analytical determination of eclipse entry and exit points considering a conical shadow and oblate Earth”, *Acta Astronautica*, 2022 (submitted for peer review).
- Nugnes, M., Colombo, C. and Tipaldi, M., “A System-Level Engineering Approach for Preliminary Performance Analysis and Design of Global Navigation Satellite System Constellations”, *International Review of Aerospace Engineering (I.RE.AS.E)*, Vol. 13, No. 3, 2020, pp. 80-98. DOI: <https://doi.org/10.15866/irease.v13i3.18424>
- Nugnes, M., Colombo, C. and Tipaldi, M., “Coverage Area Determination for Conical Fields of View considering an oblate Earth”, *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 10, 2019, pp. 2333-2345. DOI:10.2514/1.g004156

Conferences Publications

- Nugnes, M. and Colombo C., “Multi-Revolution Low-Thrust Trajectory Optimisation using Differential Dynamic Programming in Orbital Element formulation”, 73rd International Astronautical Congress (IAC), Paris, 2022.
- Nugnes, M. and Colombo, C., “A New Analytical Method for Eclipse Entry/Exit Positions Determination considering a Conical Shadow and an Oblate Earth Surface”, 2022 AAS/AIAA Astrodynamics Specialist Conference, Charlotte, 2022.
- Zuliani, C., Santoro, V., Nugnes, M. and Colombo, C., “Social Benefits Assessment of Earth Observation Missions through the Sustainable Development Goals 2030”, 72nd International Astronautical Congress (IAC), Dubai, 2021.
- Navarro, V., Grieco, R., Soja, B., Nugnes, M., Klopotek G., Tagliaferro, G., See, L., Falzarano, R., Weinacker R. and Ventura-Travaset, J., “Data Fusion and Machine Learning for Innovative GNSS Science Use Cases”, 34th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2021), St. Louis, Missouri, September 2021, pp. 2656-2669. <https://doi.org/10.33012/2021.18115>
- Nugnes, M. and Colombo, C., “A new methodology for the solution to the stiffness problem applied to low-thrust trajectory optimisation in terms of orbital elements using the differential dynamic programming”, 31st AAS/AIAA, 2021.
- Scalera, M., A., Nugnes, M. and Colombo, C., “A system-level engineering approach to define the social value rating of earth remote sensing missions through sustainable development goals”, 71st International Astronautical Congress (IAC), The CyberSpace Edition, 2020.

- Huang, S., Colombo, C., Gonzalo, J., L., Masserini, A., Nugnes, M., Vallini, L., and Petit, M., “Preliminary mission analysis of active debris removal service for large constellations”, 71st International Astronautical Congress (IAC), The CyberSpace Edition, 2020.
- Nugnes, M. and Colombo, C., “Low-thrust Trajectory Optimisation through Differential Dynamic Programming Method Based on Keplerian Orbital Elements”, 70th International Astronautical Congress (IAC), Washington D.C., 2019.
- Nugnes, M., Colombo, C. and Tipaldi, M., “A System Engineering Tool for the Optimisation of a GNSS Constellation Design”, Proceedings of the 5th IEEE International Workshop on Metrology for Aerospace, Rome, Italy, 2017, pp. 300-304.
DOI:10.1109/MetroAeroSpace.2018.8453600

CHAPTER 2

Orbit dynamics

This chapter presents the equations of motions used to represent the system dynamics within the non-linear optimal control problems solved in the following chapters. The orbit dynamics will be presented using orbital elements as state representation of the dynamics. Indeed, employing orbital elements as a formulation of the dynamics is one of the main target of this research. Part of this chapter was published as modelling section for the journal articles presenting the theoretical derivation between the coupling of DDP with orbital elements as state representation of the dynamics [74]. This chapter is organised as follows: first, Gauss' equations in terms of Keplerian elements and modified equinoctial elements are presented together with the reference quantities used to derive the non-dimensional equations of motions, then the parametric analytical derivation of a generic Runge-Kutta explicit integration scheme is presented. The chapter ends with an overall structure of how the parametrised DDP algorithm works together with the analytical expressions of J_2 orbital acceleration in terms of classic Keplerian elements and modified equinoctial elements.

2.1 Gauss' equations

The novelty of this work is the definition of a mathematical theory which describes the coupling between orbital elements as state representation of the dynamics and the DDP algorithm. Differently from traditional direct and indirect methods where spherical coordinates and orbital elements are usually preferred with respect to Cartesian coordinates, most of the works related to low-thrust trajectory optimisation using DDP adopt Cartesian coordinates as state representation for the integration of the orbital dynamics. However, orbital elements represent a valid alternative with respect to Cartesian coordinates presenting many advantages. First, orbital elements are related to geometrical quantities or anomalies that can be easily visualised from the physical point of view and give insight into the orbit problem. Moreover, the rate of variation of the orbital elements is usually very slow with respect to Cartesian coordinates. Therefore, it should be reasonable to use orbital elements for the integration of the dynamics coupled with the DDP for the solution of the optimal control problem. On the other side, in the Cartesian state representation all the variables are oscillating and have similar rate of variation. However, it is almost impossible to understand the evolution and motion of the spacecraft starting from its initial orbit looking at the Cartesian elements variation.

2.1.1 Keplerian elements

Orbit dynamics is represented using Gauss' planetary equations which handle both conservative and non-conservative accelerations [78]. For low-thrust applications it is important to introduce the rate of the mass variation to consider the reduction in time of the propellant mass. Gauss' planetary equations can be represented in many reference frames and using different sets of orbital elements. A complete summary of the different set of Gauss' planetary equations is reported in the Appendix A.1. For brevity in this paragraph only the expression of Gauss' planetary equations in terms of classic Keplerian elements in the tangential, normal, orthogonal reference frame denoted as $[\hat{\mathbf{t}}, \hat{\mathbf{n}}, \hat{\mathbf{h}}]$ it is reported while in Appendix A the expressions of Gauss' planetary equations in terms of both true anomaly and eccentric anomaly in $[\hat{\mathbf{t}}, \hat{\mathbf{n}}, \hat{\mathbf{h}}]$ and radial, transversal, and orthogonal reference frame denoted as $[\hat{\mathbf{r}}, \hat{\boldsymbol{\theta}}, \hat{\mathbf{h}}]$ are presented [79].

$$\begin{aligned}
\frac{da}{dt} &= 2\sqrt{\frac{a^3}{\mu(1-e^2)}}(1+2e\cos\nu+e^2)\frac{u_t}{m} \\
\frac{de}{dt} &= \sqrt{\frac{a(1-e^2)}{\mu(1+2e\cos\nu+e^2)}}\left[2(e+\cos\nu)\frac{u_t}{m}-\frac{1-e^2}{1+e\cos\nu}\sin\nu\frac{u_n}{m}\right] \\
\frac{di}{dt} &= \sqrt{\frac{a(1-e^2)}{\mu}}\frac{\cos(\omega+\nu)}{1+e\cos\nu}\frac{u_h}{m} \\
\frac{d\Omega}{dt} &= \sqrt{\frac{a(1-e^2)}{\mu}}\frac{\sin(\omega+\nu)}{\sin i(1+e\cos\nu)}\frac{u_h}{m} \\
\frac{d\omega}{dt} &= \sqrt{\frac{a(1-e^2)}{\mu}}\frac{1}{e\sqrt{1+2e\cos\nu+e^2}}\left\{\left[2\sin\nu\frac{u_t}{m}+\frac{2e+e^2\cos\nu+\cos\nu}{1+e\cos\nu}\frac{u_n}{m}\right]+\right. \\
&\quad \left.-\frac{\cos i\sin(\omega+\nu)}{\sin i}\frac{u_h}{1+e\cos\nu}\frac{1}{m}\right\} \\
\frac{d\nu}{dt} &= \sqrt{\frac{\mu}{[a(1-e^2)]^3}}(1+e\cos\nu)^2-\sqrt{\frac{a(1-e^2)}{\mu}}\frac{1}{e\sqrt{(1+2e\cos\nu+e^2)}}\left[2\sin\nu\frac{u_t}{m}\right. \\
&\quad \left.+\frac{2e+e^2\cos\nu+\cos\nu}{1+e\cos\nu}\frac{u_n}{m}\right] \\
\frac{dm}{dt} &= -\frac{\sqrt{u_t^2+u_n^2+u_h^2}}{Isp\cdot g_0}
\end{aligned} \tag{2.1}$$

where $a, e, i, \Omega, \omega, \nu$ are the osculating semi-major axis, eccentricity, inclination, Right Ascension of the Ascending Node (RAAN), pericentre anomaly, and true anomaly, respectively. The vector $[u_t, u_n, u_h]$ represent the components of the control actions while m is the mass of the satellite, Isp is the specific impulse, μ is the gravitational parameter, and g_0 the Earth gravity acceleration. The former formulation of the dynamics in Eq. (2.1) is not efficient for numerical integration because of the difference in terms of order of magnitudes between the orbital parameters. Therefore, Gauss' equations are rewritten considering adimensional orbital elements and

control actions. The adimensionalisation requires the definition of a set of reference variables to be used. There are two different philosophies that can be adopted for the definition of the reference quantities: all the reference variables depends on the engine specification and planetary constant, or the reference variables depend both on the planetary constant and a reference length chosen such that the semi-major axis lies in the interval $[0,1]$. Before describing the two different sets of reference quantities it is useful to derive the dimensional Gauss' planetary equations in terms of the new adimensional variables so defined:

$$\tilde{a} = \frac{a}{L_{ref}}, \quad \tilde{m} = \frac{m}{m_{ref}}, \quad \tilde{t} = \frac{t}{t_{ref}}, \quad \tilde{\mathbf{u}} = \frac{\mathbf{u}}{u_{ref}} \quad (2.2)$$

where $L_{ref}, m_{ref}, t_{ref}$ and u_{ref} are the reference length, mass, time and thrust, respectively used for the adimensionalisation, and the subscript \sim indicated the non-dimensional variable. Replacing the non-dimensional variables defined in Eq. (2.2) in the previous set of Gauss' planetary equation the result is:

$$\begin{aligned} \frac{d\tilde{a}}{d\tilde{t}} &= \sqrt{\frac{L_{ref}}{\mu} \frac{t_{ref} u_{ref}}{m_{ref}}} 2 \sqrt{\frac{\tilde{a}^3}{1-e^2} (1+2e \cos \nu + e^2)} \frac{\tilde{u}_t}{\tilde{m}} \\ \frac{d\tilde{e}}{d\tilde{t}} &= \sqrt{\frac{L_{ref}}{\mu} \frac{t_{ref} u_{ref}}{m_{ref}}} \sqrt{\frac{\tilde{a}(1-e^2)}{1+2e \cos \nu + e^2}} \left[2(e + \cos \nu) \frac{\tilde{u}_t}{\tilde{m}} - \frac{1-e^2}{1+e \cos \nu} \sin \nu \frac{\tilde{u}_n}{\tilde{m}} \right] \\ \frac{d\tilde{i}}{d\tilde{t}} &= \sqrt{\frac{L_{ref}}{\mu} \frac{t_{ref} u_{ref}}{m_{ref}}} \sqrt{\tilde{a}(1-e^2)} \frac{\cos(\omega + \nu)}{1+e \cos \nu} \frac{\tilde{u}_h}{\tilde{m}} \\ \frac{d\tilde{\Omega}}{d\tilde{t}} &= \sqrt{\frac{L_{ref}}{\mu} \frac{t_{ref} u_{ref}}{m_{ref}}} \sqrt{\tilde{a}(1-e^2)} \frac{\sin(\omega + \nu)}{\sin i (1+e \cos \nu)} \frac{\tilde{u}_h}{\tilde{m}} \\ \frac{d\tilde{\omega}}{d\tilde{t}} &= \sqrt{\frac{L_{ref}}{\mu} \frac{t_{ref} u_{ref}}{m_{ref}}} \sqrt{\tilde{a}(1-e^2)} \left\{ \frac{1}{e \sqrt{1+2e \cos \nu + e^2}} \left[2 \sin \nu \frac{\tilde{u}_t}{\tilde{m}} + \right. \right. \\ &\quad \left. \left. \frac{2e + e^2 \cos \nu + \cos \nu \tilde{u}_n}{1+e \cos \nu} \frac{\tilde{u}_n}{\tilde{m}} \right] - \frac{\cos i \sin(\omega + \nu)}{\sin i (1+e \cos \nu)} \frac{\tilde{u}_h}{\tilde{m}} \right\} \\ \frac{d\tilde{\nu}}{d\tilde{t}} &= \sqrt{\frac{\mu}{L_{ref}^3} t_{ref}} \frac{(1+e \cos \nu)^2}{\sqrt{[\tilde{a}(1-e^2)]^3}} - \sqrt{\frac{L_{ref}}{\mu} \frac{t_{ref} u_{ref}}{m_{ref}}} \frac{\sqrt{\tilde{a}(1-e^2)}}{e \sqrt{(1+2e \cos \nu + e^2)}} \left[2 \sin \nu \frac{\tilde{u}_t}{\tilde{m}} \right. \\ &\quad \left. + \frac{2e + e^2 \cos \nu + \cos \nu \tilde{u}_n}{1+e \cos \nu} \frac{\tilde{u}_n}{\tilde{m}} \right] \\ \frac{d\tilde{m}}{d\tilde{t}} &= - \frac{t_{ref} u_{ref}}{m_{ref}} \frac{\sqrt{\tilde{u}_t^2 + \tilde{u}_n^2 + \tilde{u}_h^2}}{Isp \cdot g_0} \end{aligned} \quad (2.3)$$

Looking at the equations three groups of coefficients appear:

$$\sqrt{\frac{L_{ref}}{\mu} \frac{t_{ref} u_{ref}}{m_{ref}}}, \quad \sqrt{\frac{\mu}{L_{ref}^3} t_{ref}}, \quad \frac{1}{Isp \cdot g_0} \frac{t_{ref} u_{ref}}{m_{ref}} \quad (2.4)$$

It is useful to impose each group equal to 1 so that it is possible both to simplify the differential equations notation and derive a way to define the reference variables. Even if the number of reference variables to be defined is equal to four, the actual number of unknowns is equal to three because it is impossible to decouple the reference thrust from the reference mass. Therefore, from Eq. (2.4) it is possible to write a system of three equations in 3 unknowns (reference length, time and acceleration) which admits a unique solution.

$$\begin{cases} \sqrt{\frac{L_{ref}}{\mu}} \frac{t_{ref} u_{ref}}{m_{ref}} = 1 \\ \sqrt{\frac{\mu}{L_{ref}^3}} t_{ref} = 1 \\ \frac{1}{Isp \cdot g_0} \frac{t_{ref} u_{ref}}{m_{ref}} = 1 \end{cases} \longrightarrow \begin{cases} L_{ref} = \frac{\mu}{(Isp \cdot g_0)^2} \\ t_{ref} = \frac{\mu}{(Isp \cdot g_0)^3} \\ a_{ref} = \frac{u_{ref}}{m_{ref}} = \frac{(Isp \cdot g_0)^4}{\mu} \end{cases} \quad (2.5)$$

It is clear that using this approach to define the reference quantities there is no possibility to tune the magnitude of each reference quantity. The last step is to define a way to compute the reference mass and thrust. Being the method applied for low-thrust trajectory optimisation it is convenient to set the reference thrust equal to the maximum magnitude the thruster can provide and derive the reference mass from the reference acceleration.

The second approach to define the reference quantities defines a value for the reference length, L_{ref} , "a priori" so that the value of the non-dimensional semi-major axis is in the interval $[0,1]$. This choice is quite easy because according to the final target the satellite has to reach moving inwards/outwards it is possible to select the initial or the final semi-major axis respectively as the reference length. Therefore, the selection of the other reference quantities will be carried out considering only two of the group coefficients defined before in Eq. (2.4) and forcing them being equal to 1, while the remaining group will assume a generic value.

$$L_{ref} = a_{min/max}, \quad t_{ref} = \sqrt{\frac{L_{ref}^3}{\mu}}, \quad a_{ref} = \frac{u_{ref}}{m_{ref}} = \frac{\mu}{L_{ref}^2} \quad (2.6)$$

The reference variables have been derived using the first two group of coefficients. Therefore, the last group associated to the mass rate will change and it is possible to define an equivalent specific impulse so that formally the aspect of the mass rate is the same.

$$I\tilde{sp}g_0 = \sqrt{\frac{L_{ref}}{\mu}} Isp \cdot g_0 \quad (2.7)$$

The second set of reference quantities is particularly useful when the first set of adimensionalising quantities in Eq. (2.6) is not numerically efficient. Indeed, whenever the first group results in a reference time such that the time step used for the numerical integration is too coarse, it is possible to use the second set of reference variables where the reference length can be selected a priori so that also the reference time and time step can be tuned accordingly. Then, the final set of adimensional Gauss' planetary equations considering the first set of reference quantities

in Eq. (2.4) is the following:

$$\begin{aligned}
 \frac{d\tilde{a}}{d\tilde{t}} &= 2\sqrt{\frac{\tilde{a}^3}{1-e^2}(1+2e\cos\nu+e^2)}\frac{\tilde{u}_t}{\tilde{m}} \\
 \frac{de}{d\tilde{t}} &= \sqrt{\frac{\tilde{a}(1-e^2)}{1+2e\cos\nu+e^2}} \left[2(e+\cos\nu)\frac{\tilde{u}_t}{\tilde{m}} - \frac{1-e^2}{1+e\cos\nu}\sin\nu\frac{\tilde{u}_n}{\tilde{m}} \right] \\
 \frac{di}{d\tilde{t}} &= \sqrt{\tilde{a}(1-e^2)}\frac{\cos(\omega+\nu)}{1+e\cos\nu}\frac{\tilde{u}_h}{\tilde{m}} \\
 \frac{d\Omega}{d\tilde{t}} &= \sqrt{\tilde{a}(1-e^2)}\frac{\sin(\omega+\nu)}{\sin i(1+e\cos\nu)}\frac{\tilde{u}_h}{\tilde{m}} \\
 \frac{d\omega}{d\tilde{t}} &= \sqrt{\tilde{a}(1-e^2)} \left\{ \frac{1}{e\sqrt{1+2e\cos\nu+e^2}} \left[2\sin\nu\frac{\tilde{u}_t}{\tilde{m}} + \frac{2e+e^2\cos\nu+\cos\nu}{1+e\cos\nu}\frac{\tilde{u}_n}{\tilde{m}} \right] \right. \\
 &\quad \left. - \frac{\cos i \sin(\omega+\nu)}{\sin i} \frac{\tilde{u}_h}{1+e\cos\nu} \frac{\tilde{u}_h}{\tilde{m}} \right\} \\
 \frac{d\nu}{d\tilde{t}} &= \frac{(1+e\cos\nu)^2}{\sqrt{[\tilde{a}(1-e^2)]^3}} - \frac{\sqrt{\tilde{a}(1-e^2)}}{e\sqrt{(1+2e\cos\nu+e^2)}} \left[2\sin\nu\frac{\tilde{u}_t}{\tilde{m}} \right. \\
 &\quad \left. + \frac{2e+e^2\cos\nu+\cos\nu}{1+e\cos\nu}\frac{\tilde{u}_n}{\tilde{m}} \right] \\
 \frac{d\tilde{m}}{d\tilde{t}} &= -\frac{\sqrt{\tilde{u}_t^2 + \tilde{u}_n^2 + \tilde{u}_h^2}}{Isp \cdot g_0}
 \end{aligned} \tag{2.8}$$

The only difference in the previous set of equations in Eq. (2.8) considering the second set of reference of quantities is in the mass rate equation which becomes the following:

$$\frac{d\tilde{m}}{d\tilde{t}} = -\frac{1}{Ispg_0}\sqrt{\tilde{u}_t^2 + \tilde{u}_n^2 + \tilde{u}_h^2} \tag{2.9}$$

2.1.2 Nonsingular elements

The formulation in Eq. (2.8) considers classic Keplerian elements as representation of the state. This introduces limitations for two specific class of orbits that cannot be considered because of the singularities related to the equations of motion. Indeed, Gauss' variational equations in terms of classic Keplerian elements are singular for circular orbits ($e = 0$) and equatorial orbits ($i = 0$). Even if there are only two singularities, both circular orbits and equatorial orbits are of particular interest for space missions. It is then necessary to introduce a set of non-singular elements to handle the previous two cases. In this work the set of modified equinoctial elements proposed by Walker et al. [80] is considered to overcome the singularities. These elements are valid for circular, elliptic, and hyperbolic orbits. These *direct* modified equinoctial equations exhibit no singularity for zero eccentricity and orbital inclinations equal to 0 and 90 degrees. However, two of the components are singular for an orbital inclination of 180 degrees. The modified equinoctial elements are derived starting from the classic Keplerian elements using

the following expressions:

$$\begin{aligned}
 p &= a(1 - e^2) \\
 f &= e \cos(\omega + \Omega) \\
 g &= e \sin(\omega + \Omega) \\
 h &= \tan(i/2) \cos \Omega \\
 k &= \tan(i/2) \sin \Omega \\
 \Gamma &= \Omega + \omega + \nu
 \end{aligned} \tag{2.10}$$

where p is the semi-latus rectum, f and g represents the projections of the eccentricity vector onto the nodal orbital frame, h and k are the projections of the nodal vector onto the orbital frame, and Γ is the true longitude. The inverse formulations expressing the classic orbital elements as function of the modified equinoctial elements are required to move from one representation to the other one:

$$\begin{aligned}
 a &= \frac{p}{1 - f^2 - g^2} \\
 e &= \sqrt{f^2 + g^2} \\
 i &= 2 \arctan\left(\sqrt{h^2 + k^2}\right) = \text{atan2}\left(2\sqrt{h^2 + k^2}, 1 - h^2 - k^2\right) \\
 \Omega &= \text{atan2}(k, h) \\
 \omega &= \arctan(g/f) - \arctan(k/h) = \text{atan2}(gh - fk, fh + gk) \\
 \nu &= \Gamma - (\Omega + \omega)
 \end{aligned} \tag{2.11}$$

with atan2 representing the inverse arctangent trigonometric function using as inputs the sine and cosine of the angular variable, respectively as $\text{atan2}(\sin \zeta, \cos \zeta)$ with ζ a generic angle. Gauss' equations in terms of modified equinoctial elements are derived starting from their formulation in terms of classic Keplerian elements. The direct relations in Eq. (2.10) are differentiated with respect to time and the orbital elements rate given by Gauss' expressions are replaced. Finally, classic orbital elements are replaced in terms of modified equinoctial elements used the inverse formulations given by Eq. (2.11). For the angular variables it is better to derive a specific expression for the sine and cosine trigonometric function.

$$\sin \nu = \frac{f}{\sqrt{f^2 + g^2}} \sin \Gamma - \frac{g}{\sqrt{f^2 + g^2}} \cos \Gamma \tag{2.12}$$

$$\cos \nu = \frac{f}{\sqrt{f^2 + g^2}} \cos \Gamma + \frac{g}{\sqrt{f^2 + g^2}} \sin \Gamma \tag{2.13}$$

Also in this case many sets of equations exist according to the reference frame and type of anomaly used and they are reported in the Appendix A.2. Gauss' equation in reference frame $[\hat{\mathbf{t}}, \hat{\mathbf{n}}, \hat{\mathbf{h}}]$ in terms of the true longitude, Γ :

$$\begin{aligned}
 \frac{da}{dt} &= 2\sqrt{\frac{a^3}{\mu}} \sqrt{\frac{1 + 2f \cos \Gamma + 2g \sin \Gamma + f^2 + g^2}{1 - f^2 - g^2}} \frac{u_t}{m} \\
 \frac{df}{dt} &= \sqrt{\frac{a}{\mu}} \sqrt{\frac{1 - f^2 - g^2}{1 + 2f \cos \Gamma + 2g \sin \Gamma + f^2 + g^2}} \left\{ 2(f + \cos \Gamma) \frac{u_t}{m} - [2g \right. \\
 &\quad \left. + \frac{\sin \Gamma (1 - f^2 - g^2)}{1 + f \cos \Gamma + g \sin \Gamma}] \frac{u_n}{m} \right\} - \sqrt{\frac{a(1 - f^2 - g^2)}{\mu}} g \frac{h \sin \Gamma - k \cos \Gamma}{1 + f \cos \Gamma + g \sin \Gamma} \frac{u_h}{m} \\
 \frac{dg}{dt} &= \sqrt{\frac{a}{\mu}} \sqrt{\frac{1 - f^2 - g^2}{1 + 2f \cos \Gamma + 2g \sin \Gamma + f^2 + g^2}} \left\{ 2(g + \sin \Gamma) \frac{u_t}{m} + [2f \right. \\
 &\quad \left. + \frac{\cos \Gamma (1 - f^2 - g^2)}{1 + f \cos \Gamma + g \sin \Gamma}] \frac{u_n}{m} \right\} + \sqrt{\frac{a(1 - f^2 - g^2)}{\mu}} f \frac{h \sin \Gamma - k \cos \Gamma}{1 + f \cos \Gamma + g \sin \Gamma} \frac{u_h}{m} \quad (2.14) \\
 \frac{dh}{dt} &= \frac{1}{2} \sqrt{\frac{a(1 - f^2 - g^2)}{\mu}} \frac{1 + h^2 + k^2}{1 + f \cos \Gamma + g \sin \Gamma} \cos \Gamma \frac{u_h}{m} \\
 \frac{dk}{dt} &= \frac{1}{2} \sqrt{\frac{a(1 - f^2 - g^2)}{\mu}} \frac{1 + h^2 + k^2}{1 + f \cos \Gamma + g \sin \Gamma} \sin \Gamma \frac{u_h}{m} \\
 \frac{d\Gamma}{dt} &= \sqrt{\frac{\mu}{[a(1 - f^2 - g^2)]^3}} (1 + f \cos \Gamma + g \sin \Gamma)^2 + \\
 &\quad \sqrt{\frac{a(1 - f^2 - g^2)}{\mu}} \frac{h \sin \Gamma - k \cos \Gamma}{1 + f \cos \Gamma + g \sin \Gamma} \frac{u_h}{m} \\
 \frac{dm}{dt} &= -\frac{\sqrt{u_t^2 + u_n^2 + u_h^2}}{Isp \cdot g_0}
 \end{aligned}$$

A non-dimensional set can be obtained as well using the same reference quantities defined in the previous section.

2.2 Transition functions

The most important aspect in the definition of the DDP algorithm is the computation of the cost functional partials. This is true for two main reasons:

- The partials computation is the most expensive part from a computational point of view
- A misprediction of the partials can lead to the divergence of the algorithm

The partials computation is an operation that must be carried out at each iteration of the DDP algorithm and it is impossible to parallelise the different operations because each prediction depends on the value of the previous state. However, according to the different way the partials can be evaluated the convergence can be made faster or slower. The second aspect defined in the

list is crucial because it is affecting the convergence of the algorithm. Indeed, a misprediction of the cost functional partials or a discontinuity at a single time step during the backward regression causes the divergence of the algorithm. This phenomenon can occur even if the system dynamic is continuous but highly non-linear due to a wrong discretisation of the problem. The partials computation is different if the continuous or discrete version of the DDP algorithm is used. If the continuous version is adopted the partials computation requires only the Jacobian and Hessian matrix of the continuous dynamics. For the discrete version of the algorithm a transition function that is used to map the state at step j to step $j + 1$ is required. In the early 1970s when the DDP algorithm was developed a simple Euler scheme was used for the computation of the partials of the transition function [40]. This approach was not efficient to apply the DDP algorithm to large non-linear dynamical systems due to the low accuracy of the Euler scheme. The advancement in the numerical integration of differential equations allowed to use as transition function higher-order explicit schemes like Runge-Kutta-Fehlberg (RKF) [81]. In this case the partials computation is not equivalent to the Jacobian and Hessian matrices associated to the continuous dynamic, but the Jacobian and Hessian matrices of the numerical schemes are required. This issue can be solved in two ways: computing analytically the expressions of the partial derivatives of the numerical schemes which is tedious and complicated as the order of the numerical scheme increases, or exploiting tools as STMs [82] which require the solution to a system of differential equations for each iteration of the DDP algorithm.

2.2.1 Analytical computation of the partials

This research uses the analytical expression of the Jacobian and Hessian matrices of a generic Runge-Kutta explicit scheme to evaluate the cost functional partials required for the application of the DDP algorithm. The derivation of the analytical expression of such schemes is cumbersome but it has been decided to be used for two main reasons:

- The analytical evaluation of the partial derivatives makes the algorithm faster.
- A recursive formulation can be identified to compute the partial derivatives for all the stages of the numerical schemes.

A generic Runge-Kutta explicit scheme can be formulated as follows [81].

$$\mathbf{x}_{j+1} = \mathbf{x}_j + s \sum_{z=1}^{n_{RK}} \mathbf{b}_z^{RK} \mathbf{k}_z(\mathbf{x}_j; t_j) \quad (2.15)$$

with \mathbf{x}_j and \mathbf{x}_{j+1} representing the system state at time j and $j + 1$, h is the time step, n_{RK} the number of stages of the Runge-Kutta scheme, and \mathbf{k}_z are intermediate function evaluated in the interval $[t_j, t_{j+1}]$.

$$\mathbf{k}_j = \mathbf{f} \left(\mathbf{x}_j + h \sum_{z=1}^{j-1} a_{jz}^{RK} \mathbf{k}_z, t_j + s c_j^{RK} \right), \quad \mathbf{k}_1 = \mathbf{f}(\mathbf{x}_0, t_0) \quad (2.16)$$

The coefficients a_i^{RK} , b_i^{RK} , c_i^{RK} are specific of a given Runge-Kutta scheme and can be found in the Butcher Tableau, and \mathbf{x}_0 represents the initial state. The Butcher Tableau for the Runge-Kutta 8th order scheme used in this work is reported in Table 2.1 at the end of the chapter.

Starting from Eq. 2.16 the analytical differentiation can be carried out to define the generic expression of Jacobian and Hessian matrices. The recursivity of the expression allows to evaluate the partial derivatives of the intermediate function, k_i , at the next state of the numerical scheme from the current one.

$$\frac{d\mathbf{k}_j}{d\mathbf{x}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}|\bar{\mathbf{x}}} \left(\mathbf{I} + s \sum_{z=1}^{j-1} a_{jz} \frac{d\mathbf{k}_z}{d\mathbf{x}} \right) \quad (2.17)$$

$$\frac{d\mathbf{k}_j}{d\mathbf{u}} = \frac{\partial \mathbf{f}}{\partial \mathbf{u}|\bar{\mathbf{x}}} + \frac{\partial \mathbf{f}}{\partial \mathbf{x}|\bar{\mathbf{x}}} s \sum_{z=1}^{j-1} a_{jz} \frac{d\mathbf{k}_z}{d\mathbf{u}} \quad (2.18)$$

$$\frac{d^2 \mathbf{k}_j}{d\mathbf{x}^2} = \left(\mathbf{I} + s \sum_{z=1}^{j-1} a_{jz} \frac{d\mathbf{k}_z}{d\mathbf{x}} \right)^T \frac{\partial^2 \mathbf{f}}{\partial \mathbf{x}^2|\bar{\mathbf{x}}} \left(\mathbf{I} + s \sum_{z=1}^{j-1} a_{jz} \frac{d\mathbf{k}_z}{d\mathbf{x}} \right) + \frac{\partial \mathbf{f}}{\partial \mathbf{x}|\bar{\mathbf{x}}} s \sum_{z=1}^{j-1} a_{jz} \frac{d^2 \mathbf{k}_z}{d\mathbf{x}^2} \quad (2.19)$$

$$\begin{aligned} \frac{d^2 \mathbf{k}_j}{d\mathbf{x} d\mathbf{u}} &= \left(\mathbf{I} + s \sum_{z=1}^{j-1} a_{jz} \frac{d\mathbf{k}_z}{d\mathbf{x}} \right)^T \left(\frac{\partial^2 \mathbf{f}}{\partial \mathbf{x}^2|\bar{\mathbf{x}}} s \sum_{z=1}^{j-1} a_{jz} \frac{d\mathbf{k}_z}{d\mathbf{u}} + \frac{\partial^2 \mathbf{f}}{\partial \mathbf{x} \partial \mathbf{u}|\bar{\mathbf{x}}} \right) \\ &+ \frac{\partial \mathbf{f}}{\partial \mathbf{x}|\bar{\mathbf{x}}} s \sum_{z=1}^{j-1} a_{jz} \frac{d^2 \mathbf{k}_z}{d\mathbf{x} d\mathbf{u}} \end{aligned} \quad (2.20)$$

$$\begin{aligned} \frac{d^2 \mathbf{k}_j}{d\mathbf{u}^2} &= \frac{\partial^2 \mathbf{f}}{\partial \mathbf{u}^2|\bar{\mathbf{x}}} + \left(\sum_{z=1}^{j-1} a_{jz} \frac{d\mathbf{k}_z}{d\mathbf{u}} \frac{\partial^2 \mathbf{f}}{\partial \mathbf{x} \partial \mathbf{u}|\bar{\mathbf{x}}} \right)^T + \sum_{z=1}^{j-1} a_{jz} \frac{d\mathbf{k}_z}{d\mathbf{u}} \frac{\partial^2 \mathbf{f}}{\partial \mathbf{x} \partial \mathbf{u}|\bar{\mathbf{x}}} \\ &+ \left(s \sum_{z=1}^{j-1} a_{jz} \frac{d\mathbf{k}_z}{d\mathbf{u}} \right)^T \frac{\partial^2 \mathbf{f}}{\partial \mathbf{x}^2|\bar{\mathbf{x}}} s \sum_{z=1}^{j-1} a_{jz} \frac{d\mathbf{k}_z}{d\mathbf{u}} + \frac{\partial \mathbf{f}}{\partial \mathbf{x}|\bar{\mathbf{x}}} s \sum_{z=1}^{j-1} a_{jz} \frac{d^2 \mathbf{k}_z}{d\mathbf{u}^2} \end{aligned} \quad (2.21)$$

where $\bar{\mathbf{x}}$ denotes an intermediate function evaluation of the dynamics between state \mathbf{x}_j and state \mathbf{x}_{j+1} , the symbol $\partial/\partial(\cdot)$ and $d/d(\cdot)$ represent the partial derivative and total derivative of a function with respect to a specific variable, respectively. The following notation has been used for the derivation of the transition function partial derivatives.

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}|\bar{\mathbf{x}}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_{|\bar{\mathbf{x}}}, \quad \frac{\partial^2 f_j}{\partial \mathbf{x}^2|\bar{\mathbf{x}}} = \begin{bmatrix} \frac{\partial^2 f_j}{\partial x_1^2} & \frac{\partial^2 f_j}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f_j}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f_j}{\partial x_1 \partial x_2} & \frac{\partial^2 f_j}{\partial x_2^2} & \cdots & \frac{\partial^2 f_j}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f_j}{\partial x_n \partial x_2} & \frac{\partial^2 f_j}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f_j}{\partial x_n^2} \end{bmatrix}_{|\bar{\mathbf{x}}} \quad (2.22)$$

The expression of the Hessian matrix for the scalar function f_{xx}^j represent the classic Hessian function applied to each scalar component of the vectorial function, \mathbf{f} , denoting the continuous dynamics of system. Therefore, the Hessian matrix is multi-dimensional array where the single Hessian matrix of the scalar functions represent the different layers described by the third dimension.

Particular attention must be devoted to the operations involving the Hessian matrices because the products of vectors and third-order dimensional arrays or matrices and third-order

dimensional arrays exist. Keeping the philosophy to assign last index of a third-order dimensional array to the Hessian matrix of each scalar component of the continuous dynamics, the only operation that must be reshuffled is the last one occurring in Eqs. (2.19-2.21). In this case each layer of the multi-dimensional arrays is generated considering the product of each row of the Jacobian $\partial f/\partial x$ and the matrix which is created cutting vertically the multi-dimensional array. A simple example to show how this operation is carried out is proposed imaging a vectorial function with 3 components, 3 states and 2 control variables.

$$\begin{aligned}
 \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \frac{d^2 \mathbf{k}_j}{d\mathbf{x} d\mathbf{u}} &= \begin{bmatrix} f_{x_1}^1 & f_{x_2}^1 & f_{x_3}^1 \\ f_{x_1}^2 & f_{x_2}^2 & f_{x_3}^2 \\ f_{x_1}^3 & f_{x_2}^3 & f_{x_3}^3 \end{bmatrix} \left(\begin{bmatrix} f_{x_1 u_1}^1 & f_{x_1 u_2}^1 \\ f_{x_2 u_1}^1 & f_{x_2 u_2}^1 \\ f_{x_3 u_1}^1 & f_{x_3 u_2}^1 \end{bmatrix} \begin{bmatrix} f_{x_1 u_1}^2 & f_{x_1 u_2}^2 \\ f_{x_2 u_1}^2 & f_{x_2 u_2}^2 \\ f_{x_3 u_1}^2 & f_{x_3 u_2}^2 \end{bmatrix} \begin{bmatrix} f_{x_1 u_1}^3 & f_{x_1 u_2}^3 \\ f_{x_2 u_1}^3 & f_{x_2 u_2}^3 \\ f_{x_3 u_1}^3 & f_{x_3 u_2}^3 \end{bmatrix} \right) = \\
 &= \begin{bmatrix} f_{x_1}^1 & f_{x_2}^1 & f_{x_3}^1 \end{bmatrix} \begin{bmatrix} f_{x_1 u_1}^1 & f_{x_1 u_2}^1 \\ f_{x_2 u_1}^1 & f_{x_2 u_2}^1 \\ f_{x_3 u_1}^1 & f_{x_3 u_2}^1 \end{bmatrix}, \begin{bmatrix} f_{x_1}^2 & f_{x_2}^2 & f_{x_3}^2 \end{bmatrix} \begin{bmatrix} f_{x_2 u_1}^1 & f_{x_2 u_2}^1 \\ f_{x_2 u_1}^2 & f_{x_2 u_2}^2 \\ f_{x_2 u_1}^3 & f_{x_2 u_2}^3 \end{bmatrix}, \\
 &\begin{bmatrix} f_{x_1}^3 & f_{x_2}^3 & f_{x_3}^3 \end{bmatrix} \begin{bmatrix} f_{x_3 u_1}^1 & f_{x_3 u_2}^1 \\ f_{x_3 u_1}^2 & f_{x_3 u_2}^2 \\ f_{x_3 u_1}^3 & f_{x_3 u_2}^3 \end{bmatrix}
 \end{aligned} \tag{2.23}$$

2.2.2 Scheme of the algorithm

It is possible now to summarise the procedure to evaluate the cost functional partials in Figure 2.1. Starting from the definition of the system dynamics, MATLAB symbolic toolbox is used to evaluate the analytical expressions of the system dynamics for the continuous version. Starting from the analytical Jacobian and Hessian of the continuous dynamics Eqs. (2.17-2.21) are used to evaluate the analytical Jacobian and Hessian matrix of the transition scheme providing the coefficient of the Runge-Kutta numerical scheme.

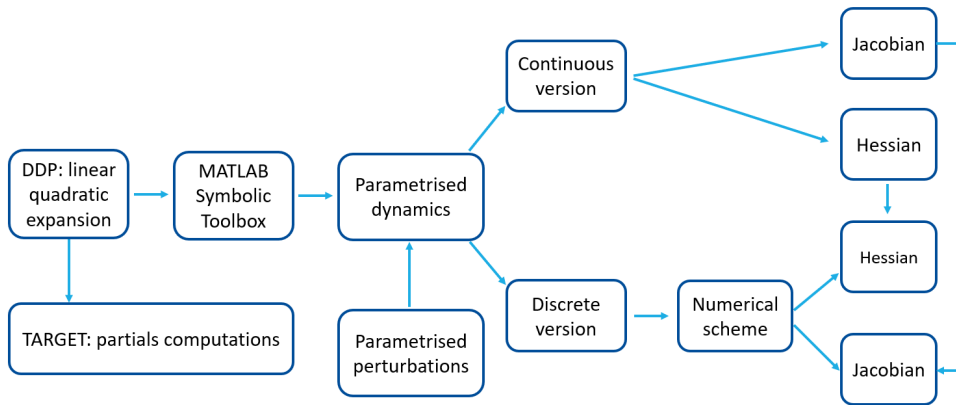


Figure 2.1: Partials computation flow diagram.

The algorithm can handle also the expression of orbital perturbations inside the system dynamics whose partials are computed analytically if a formulation in terms of orbital elements exist. An example of such disturbances is offered by the Earth geopotential model. This method express the Earth gravity field as an expansion of several harmonics which are function of the

orbital elements. The inclusion of the first harmonic, denoted as J_2 , contributes already to improve the accuracy of the orbit propagation. This happens because the first harmonic is 3 order of magnitudes higher than the following harmonics. The analytical expression of the disturbing accelerations in the radial, transversal and normal reference frame is reported in terms of classic orbital elements [78].

$$\begin{aligned} a_{r|J_2} &= -\frac{3}{2} \frac{J_2 \mu_{\oplus} R_{\oplus}^2}{r^4} [1 - 3 \sin^2 i \sin^2 (\omega + \nu)] \\ a_{\theta|J_2} &= -\frac{3}{2} \frac{J_2 \mu_{\oplus} R_{\oplus}^2}{r^4} \sin^2 i \sin 2 (\omega + \nu) \\ a_{h|J_2} &= -\frac{3}{2} \frac{J_2 \mu_{\oplus} R_{\oplus}^2}{r^4} \sin 2i \sin (\omega + \nu) \end{aligned} \quad (2.24)$$

The same expression in the same reference frame can be expressed also as function of the modified equinoctial elements [80].

$$\begin{aligned} a_{r|J_2} &= -\frac{3}{2} \frac{J_2 \mu_{\oplus} R_{\oplus}^2}{r^4} \left[1 - 12 \frac{(h \sin \Gamma - k \cos \Gamma)^2}{(1 + h^2 + k^2)^2} \right] \\ a_{\theta|J_2} &= -12 \frac{J_2 \mu_{\oplus} R_{\oplus}^2}{r^4} \frac{(h \sin \Gamma - k \cos \Gamma) (h \cos \Gamma + k \sin \Gamma)}{(1 + h^2 + k^2)^2} \\ a_{h|J_2} &= -6 \frac{J_2 \mu_{\oplus} R_{\oplus}^2}{r^4} \frac{(h \sin \Gamma - k \cos \Gamma) (1 - h^2 - k^2)}{(1 + h^2 + k^2)^2} \end{aligned} \quad (2.25)$$

where J_2 is main spherical harmonic coefficient used to model the Earth gravitational field, R_{\oplus} is the Earth mean equatorial radius, and r is the spacecraft position vector magnitude. The analytical expression of the orbit perturbations can be added inside Gauss' equations so that the symbolic toolbox can directly compute the new updated partial derivatives. If no analytical expression is present the same scheme can be kept using the derivation chain rule.

$$\frac{d}{d\xi} \left(\frac{d\mathbf{f}}{dt} \right) = \frac{\partial \mathbf{f}(\xi, \mathbf{a}_p(\xi))}{\partial \xi} + \frac{\partial \mathbf{f}}{\partial \xi} \mathbf{a}_p + \frac{\partial \mathbf{a}_p}{\partial \xi} \mathbf{f} \quad (2.26)$$

where $\partial \mathbf{f} / \partial \xi$ is the partial derivative with respect to an orbital elements of the unperturbed Gauss' equation, while \mathbf{a}_p and $\partial \mathbf{a}_p / \partial \xi$ are the expressions of the disturbing accelerations and their partial derivatives with respect to the generic orbital element. Using this generic scheme a complete parametrisation of the orbit perturbations is provided because it does not matter if the expression of the partial derivative of the disturbances are computed analytically or numerically. In Chapter 8 a multi-revolution planetary transfer including J_2 orbital perturbation is presented taking into account also the effect of eclipses which are the only orbit perturbations considered in this work.

2.3 Summary

In this chapter the orbital dynamics used as equations of motions is presented. Gauss' equations are derived in several reference frame and in terms of different sets of orbital elements. The two different types of adimensionalisation are analysed and guidelines are provided to select the best philosophy according to the scenario. Transition functions to discretise the equations of motions using a parametrised Runge-Kutta scheme are presented and the analytical derivation of their partial derivatives is carried out. Finally, the overall structure of the DDP algorithm is shown together with the expressions of J_2 orbital perturbation in terms of classic Keplerian and nonsingular elements.

n_{RK}	$a_{i,j}$	c_i
1	0	0
2	$\frac{1}{8}$	$\frac{1}{8}$
3	$\frac{1}{48}$	$\frac{1}{24}$
4	$\frac{3}{32}$	$\frac{5}{16}$
5	$\frac{75}{64}$	$\frac{3}{8}$
6	$\frac{3}{16}$	$\frac{59}{400}$
7	$\frac{80}{29443841}$	$\frac{39}{2400023248}$
8	$\frac{614563906}{16016141}$	$\frac{3419169821}{1290146811}$
9	$\frac{94692911}{54672703}$	$\frac{1299019798}{1}$
10	$\frac{5246721663}{130847787}$	$\frac{1}{1}$
11	$\frac{846189014}{185892177}$	$\frac{1}{1}$
12	$\frac{718116943}{40863854}$	$\frac{1}{1}$
13	$\frac{491063109}{14005451}$	$\frac{1}{1}$
b_i	$\frac{335480064}{1068277825}$	$\frac{1}{4}$

Table 2.1: Butcher tableau for Runge-Kutta 8th order integration scheme.

CHAPTER 3

Differential dynamic programming

This chapter presents the analytical derivation of the DDP algorithm either for continuous and discrete formulation. Starting from Bellman principle of optimality the cost functional is defined and the backward equations representing the necessary conditions of optimality are obtained.

The derivation of the backward equations for the unconstrained optimal control problem is carried out considering a continuous formulation of the dynamics and it follows Jacobson and Mayne [39], whereas for the constrained problem algorithm the discrete formulation is used following Colombo et al. [42].

The last part of the chapter explains some techniques to improve the converge of the algorithm and presents a list of the different step to apply the DDP implemented in this thesis.

3.1 Mathematical derivation continuous version

The first formulation of the DDP has been formulated for continuous-time dynamic systems by Jacobson and Mayne [39] that is used as reference for the current section. A generic continuous-time dynamic systems is described by a set of non-linear ordinary differential equations and a set of initial conditions:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}; t) \\ \mathbf{x}(t_1) = \mathbf{x}_1 \end{cases} \quad (3.1)$$

where $\mathbf{x}(t)$ is a n_x -dimensional vector function of time describing the state of the dynamic system, $\mathbf{u}(t)$ is a n_u -dimensional vector function of time that represents the control actions at each time instant. A cost functional, J , is introduced to assess the performance of the dynamic system:

$$J(\mathbf{x}_1; t_1) = \int_{t_1}^{t_f} L(\mathbf{x}, \mathbf{u}; t) + \bar{F}(\mathbf{x}(t_f); t_f) \quad (3.2)$$

with L being a scalar time function relating the cost functional at each time instant, and \bar{F} a scalar function depending only on the final time instant, t_f , which may be given explicitly or implicitly. The bar symbol, \bar{F} is used to distinguish the terminal cost function at the final state

from the transition function that will be denoted with \mathbf{F} . The general purpose of a non-linear optimal control problem is to find the optimal control policy which minimises or maximises the cost functional. Dynamic programming represents one of the possible methods to accomplish the optimisations problems. This methods is based on Bellman's principle of optimality [83]:

"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."

The main idea of dynamic programming is to decompose the overall problem in a sequence of many decision sub-problems where the cost functional (also known as "value function") represents the value of each decision sub-problem. The optimal control for each decision sub-problem is independent of the initial states and controls used for the previous sub-problems. Therefore, the optimal control of the overall problem can be represented as the sequence of the sub-problems optimisation at that decision step plus the optimal cost function until the final time. For continuous-time systems Bellman's principle of optimality is described by a Partial Differential Equation (PDE) known as Hamilton-Jacobi-Bellman (HJB) equation:

$$-\frac{\partial J^*}{\partial t} = \min_{\mathbf{u}(t)} \left[L(\mathbf{x}, \mathbf{u}; t) + \frac{\partial J^*}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}, \mathbf{u}; t) \right] \quad (3.3)$$

with $\partial J^*/\partial t$ is the partial derivative of the optimal cost functional with respect to time, and $\partial J^*/\partial \mathbf{x}$ is the Jacobian of the optimal cost functional. The classic way to solve HJB equation is to discretise both states and controls so that necessary and sufficient conditions for global optimality can be satisfied. However, this process is not straightforward when the high-dimensional spaces are considered because huge storage requirements are required to solve the overall problem leading to the unfeasibility to solve HJB equation. This concept is known in literature as "curse of dimensionality" [46]. The DDP overcomes the issue thanks to a linear-quadratic expansion of the HJB equations starting from a nominal non-optimal first guess solution. This way, the optimal solution will no longer be a global optimum but a local optimum solution because the search space has been restricted to the neighbourhood of the first guess solution used for the Taylor expansion.

The starting point for the derivation of the DDP equations is to express both the state and the control in Eq. (3.3) as the sum of the nominal trajectory and a variation.

$$\mathbf{x} = \bar{\mathbf{x}} + \delta \mathbf{x} \quad \mathbf{u} = \bar{\mathbf{u}} + \delta \mathbf{u} \quad (3.4)$$

where $\delta \mathbf{x}$ and $\delta \mathbf{u}$ are the state and control variations, respectively, measured with respect to the nominal state, $\bar{\mathbf{x}}$, and control, $\bar{\mathbf{u}}$. Replacing Eq. (3.4) inside the HJB equation (3.3) the following expression is obtained:

$$-\frac{\partial J^*}{\partial t}(\bar{\mathbf{x}} + \delta \mathbf{x}; t) = \min_{\delta \mathbf{u}} \left[L(\bar{\mathbf{x}} + \delta \mathbf{x}, \bar{\mathbf{u}} + \delta \mathbf{u}; t) + \frac{\partial J^*}{\partial \mathbf{x}}(\bar{\mathbf{x}} + \delta \mathbf{x}; t) \mathbf{f}(\bar{\mathbf{x}} + \delta \mathbf{x}, \bar{\mathbf{u}} + \delta \mathbf{u}; t) \right] \quad (3.5)$$

We assume that each term in Eq. (3.5) admits a Taylor series expansion [39].

$$J^*(\bar{\mathbf{x}} + \delta \mathbf{x}; t) = J^*(\bar{\mathbf{x}}; t) + \mathbf{J}_x^*(\bar{\mathbf{x}}; t) \delta \mathbf{x} + 0.5 \delta \mathbf{x}^T \mathbf{J}_{xx}^*(\bar{\mathbf{x}}; t) \delta \mathbf{x} \quad (3.6)$$

$$L(\bar{\mathbf{x}} + \delta\mathbf{x}, \bar{\mathbf{u}} + \delta\mathbf{u}; t) = L(\bar{\mathbf{x}}, \bar{\mathbf{u}}; t) + \mathbf{L}_x^T(\bar{\mathbf{x}}, \bar{\mathbf{u}}; t)\delta\mathbf{x} + \mathbf{L}_u^T(\bar{\mathbf{x}}, \bar{\mathbf{u}}; t)\delta\mathbf{u} \\ + 0.5\delta\mathbf{x}^T\mathbf{L}_{xx}(\bar{\mathbf{x}}, \bar{\mathbf{u}}; t)\delta\mathbf{x} + 0.5\delta\mathbf{u}^T\mathbf{L}_{uu}(\bar{\mathbf{x}}, \bar{\mathbf{u}}; t)\delta\mathbf{u} + \delta\mathbf{x}^T\mathbf{L}_{xu}(\bar{\mathbf{x}}, \bar{\mathbf{u}}; t)\delta\mathbf{u} \quad (3.7)$$

$$\mathbf{J}_x^*(\bar{\mathbf{x}} + \delta\mathbf{x}; t) = \mathbf{J}_x^*(\bar{\mathbf{x}}; t) + \mathbf{J}_{xx}^*(\bar{\mathbf{x}}; t)\delta\mathbf{x} + 0.5\delta\mathbf{x}^T\mathbf{J}_{xxx}^*(\bar{\mathbf{x}}; t)\delta\mathbf{x} \quad (3.8)$$

$$\mathbf{f}(\bar{\mathbf{x}} + \delta\mathbf{x}, \bar{\mathbf{u}} + \delta\mathbf{u}; t) = \mathbf{f}(\bar{\mathbf{x}}, \bar{\mathbf{u}}; t) + \mathbf{f}_x(\bar{\mathbf{x}}, \bar{\mathbf{u}}; t)\delta\mathbf{x} + \mathbf{f}_u(\bar{\mathbf{x}}, \bar{\mathbf{u}}; t)\delta\mathbf{u} \\ + 0.5\delta\mathbf{x}^T\mathbf{f}_{xx}(\bar{\mathbf{x}}, \bar{\mathbf{u}}; t)\delta\mathbf{x} + 0.5\delta\mathbf{u}^T\mathbf{f}_{uu}(\bar{\mathbf{x}}, \bar{\mathbf{u}}; t)\delta\mathbf{u} + \delta\mathbf{x}^T\mathbf{f}_{xu}(\bar{\mathbf{x}}, \bar{\mathbf{u}}; t)\delta\mathbf{u} \quad (3.9)$$

where \mathbf{J}_x^* , \mathbf{J}_{xx}^* are the gradient and Hessian of the optimal cost functional with respect to the state, \mathbf{L}_x , \mathbf{L}_{xx} are the gradient and Hessian of the cost function, L , with respect to the state, and \mathbf{f}_x , \mathbf{f}_{xx} are the Jacobian and Hessian matrices of the continuous dynamics with respect to the state. The variables with subscripts u represent the same variational functions computed with respect to the control in place of the state. In general, in this work the subscript f_\square will be used to denote the partial derivative of a function with respect to that generic variable. The general description of DDP divides the methodology in two possible approaches:

- a *local* version, where the nominal trajectory used as first guess solution is selected without any criterion [43].
- a *global* version, where the nominal trajectory used for the Taylor expansion is the result of an early optimisation where all the state and control variations are set to zero [42].

The adjective *global* is not related to the type of optimum solution obtained from the algorithm because the main assumptions of the DDP is that the solution represents a local optimum solution. The meaning of *local* and *global* is related to the size of the control variation. In the *global* approach the partial derivatives of the cost functional and of the transition functions are not evaluated considering the nominal control, but using a control policy which is obtained minimising the expanded HJB equation putting all the variations equal to zero. This preliminary minimisation allows the DDP algorithm to increase the search space to find the optimal control variation, $\delta\mathbf{u}$, keeping the Taylor linear-quadratic expansion still consistent. The advantage of using a larger control variation requires to solve a preliminary minimisation problem at each decision step. The personal experience of the author suggests that in the trade-off between *local* and *global* version of DDP algorithm it is better to adopt the *local* one. The main reason is that after the solution of the preliminary optimal control problem it is not guaranteed that the first guess solution will admit a large control variation. Moreover, it is not possible to quantify how much the control variation must be to be considered "large". This is the reason why only the *local* version of the DDP algorithm will be presented hereafter both for the continuous and the discrete version.

The expanded HJB equation can be obtained replacing Eqs. (3.6-3.9) in Eq. (3.5). For the sake of simplicity the dependencies of each function will be dropped from now on.

$$-\frac{\partial J^*}{\partial t} - \frac{\partial \mathbf{J}_x^{*T}}{\partial t} \delta\mathbf{x} - 0.5\delta\mathbf{x}^T \frac{\partial \mathbf{J}_{xx}^*}{\partial t} \delta\mathbf{x} = \min_{\delta\mathbf{u}} [L + \mathbf{L}_x^T \delta\mathbf{x} + \mathbf{L}_u^T \delta\mathbf{u} \\ + 0.5\delta\mathbf{x}^T \mathbf{L}_{xx} \delta\mathbf{x} + 0.5\delta\mathbf{u}^T \mathbf{L}_{uu} \delta\mathbf{u} + \delta\mathbf{x}^T \mathbf{L}_{xu} \delta\mathbf{u} + (\mathbf{J}_x^* + \mathbf{J}_{xx}^* \delta\mathbf{x})^T (\mathbf{f} + \mathbf{f}_x \delta\mathbf{x} + \\ + \mathbf{f}_u \delta\mathbf{u} + 0.5\delta\mathbf{x}^T \mathbf{f}_{xx} \delta\mathbf{x} + 0.5\delta\mathbf{u}^T \mathbf{f}_{uu} \delta\mathbf{u} + \delta\mathbf{x}^T \mathbf{f}_{xu} \delta\mathbf{u})] \quad (3.10)$$

The term \mathbf{J}_{xxx}^* has been dropped because it can be proven that it is a higher-order infinitesimal [39]. The terms multiplying more than two variations are dropped from the expansion because

represent higher-order infinitesimal. The optimal control variation, $\delta \mathbf{u}^*$, is determined differentiating the right-hand side of Eq. (3.10).

$$\mathbf{L}_u + \mathbf{L}_{uu}\delta \mathbf{u}^* + \mathbf{L}_{ux}\delta \mathbf{x} + \mathbf{J}_x^T \mathbf{f}_u + \mathbf{J}_x^{*T} \mathbf{f}_{uu}\delta \mathbf{u}^* + \mathbf{J}_x^{*T} \mathbf{f}_{ux}\delta \mathbf{x} + \mathbf{f}_u^T \mathbf{J}_{xx}^* \delta \mathbf{x} = 0 \quad (3.11)$$

$$\delta \mathbf{u}^* = - \left(\mathbf{L}_{uu} + \mathbf{J}_x^{*T} \mathbf{f}_{uu} \right)^{-1} \left[\left(\mathbf{L}_u + \mathbf{J}_x^{*T} \mathbf{f}_u \right) + \left(\mathbf{L}_{ux} + \mathbf{J}_x^{*T} \mathbf{f}_{ux} + \mathbf{f}_u^T \mathbf{J}_{xx}^* \right) \delta \mathbf{x} \right] \quad (3.12)$$

Once the optimal control variation is obtained, this can be replaced in the expanded HJB equation and the minimum operator can be removed because one of the main assumptions used for the derivation of the DDP algorithm, next to the continuity and twice differentiability of all the function, is the positive-definiteness of the matrix, $\mathbf{L}_{uu} + \mathbf{J}_x^{*T} \mathbf{f}_{uu}$ (which will be shown to be also the Hessian matrix of the Hamiltonian function with respect to the control variable at the end of the section). The positive definiteness is a second order necessary and sufficient conditions guaranteeing that the optimal control variation $\delta \mathbf{u}^*$ is minimising the argument of the minimum operator. In order to simplify the notation it is better to write the optimal control variation in Eq. (3.12) as the sum of a constant term, $\boldsymbol{\alpha}$, and state feedback term, $\boldsymbol{\beta}$.

$$\delta \mathbf{u}^* = \boldsymbol{\alpha} + \boldsymbol{\beta} \delta \mathbf{x} \quad (3.13)$$

$$\begin{aligned} \boldsymbol{\alpha} &= - \left(\mathbf{L}_{uu} + \mathbf{J}_x^{*T} \mathbf{f}_{uu} \right)^{-1} \left(\mathbf{L}_u + \mathbf{J}_x^{*T} \mathbf{f}_u \right) \\ \boldsymbol{\beta} &= - \left(\mathbf{L}_{uu} + \mathbf{J}_x^{*T} \mathbf{f}_{uu} \right)^{-1} \left(\mathbf{L}_{ux} + \mathbf{J}_x^{*T} \mathbf{f}_{ux} + \mathbf{f}_u^T \mathbf{J}_{xx}^* \right) \end{aligned}$$

Other two important steps are required for the final derivation of the backward equations. The optimal cost functional, J^* , is expressed as the sum of a nominal cost functional, \bar{J} , and a term which is denoting the expected error between the optimal cost and the nominal one, ϑ [39].

$$J^* = \bar{J} + \vartheta \quad (3.14)$$

Secondly, the partial derivatives with respect to time of all the terms appearing at the left-hand side of Eq. (3.10) must be expressed in terms of total time derivatives which are required to compute the time variation of the optimal cost functional partials. This can be done applying the chain rule to the total time derivative of the functional cost.

$$\frac{dJ^*}{dt} = \frac{\partial \bar{J}}{\partial t} + \mathbf{J}_x^{*T} \mathbf{f} \quad (3.15)$$

$$\frac{d\mathbf{J}_x^*}{dt} = \frac{\partial \bar{\mathbf{J}}_x}{\partial t} + \mathbf{J}_{xx}^* \mathbf{f} \quad (3.16)$$

$$\frac{d\mathbf{J}_{xx}^*}{dt} = \frac{\partial \bar{\mathbf{J}}_{xx}}{\partial t} + \mathbf{J}_{xxx}^* \mathbf{f} \quad (3.17)$$

Replacing Eq. (3.13) into Eq. (3.10) and using Eqs. (3.14-3.17) the following backward partial

differential equations are obtained:

$$\left\{ \begin{aligned} -\frac{d\vartheta}{dt} &= \left(\mathbf{L}_u^T + \mathbf{J}_x^{*T} \mathbf{f}_u \right) \boldsymbol{\alpha} + \frac{1}{2} \boldsymbol{\alpha}^T \left(\mathbf{L}_{uu} + \mathbf{J}_x^{*T} \mathbf{f}_{uu} \right) \boldsymbol{\alpha} \\ -\frac{d\mathbf{J}_x^*}{dt} &= \left[\left(\mathbf{L}_x + \mathbf{J}_x^{*T} \mathbf{f}_x \right) + \left(\mathbf{L}_u + \mathbf{J}_x^{*T} \mathbf{f}_u \right)^T \boldsymbol{\beta} + \boldsymbol{\alpha}^T \left(\mathbf{L}_{uu} + \mathbf{J}_x^{*T} \mathbf{f}_{uu} \right) \boldsymbol{\beta} \right. \\ &\quad \left. + \left(\mathbf{L}_{xu} + \mathbf{J}_x^{*T} \mathbf{f}_{xu} \right) \boldsymbol{\alpha} + \mathbf{J}_{xx}^* \left(\mathbf{f}_u \boldsymbol{\alpha} + \frac{1}{2} \boldsymbol{\alpha}^T \mathbf{f}_{uu} \boldsymbol{\alpha} \right) \right] \\ -\frac{d\mathbf{J}_{xx}^*}{dt} &= \left[\left(\mathbf{L}_{xx} + \mathbf{J}_x^{*T} \mathbf{f}_{xx} \right) + \boldsymbol{\beta}^T \left(\mathbf{L}_{uu} + \mathbf{J}_x^{*T} \mathbf{f}_{uu} \right) \boldsymbol{\beta} + \right. \\ &\quad \left. + 2 \left(\mathbf{L}_{xu} + \mathbf{J}_x^{*T} \mathbf{f}_{xu} \right) \boldsymbol{\beta} + 2 \mathbf{J}_{xx}^* \left(\mathbf{f}_x + \mathbf{f}_u \boldsymbol{\beta} \right) + \mathbf{J}_{xx}^* \left(\boldsymbol{\alpha}^T \mathbf{f}_{uu} \boldsymbol{\beta} + \mathbf{f}_{xu} \boldsymbol{\alpha} \right) \right] \end{aligned} \right. \quad (3.18)$$

The system in Eq. (3.18) can be rewritten in terms of the Hamiltonian operator, H , that is very used in optimal control theory.

$$H(\mathbf{x}, \mathbf{u}; t) = L(\mathbf{x}, \mathbf{u}; t) + \mathbf{J}_x^{*T}(\mathbf{x}; t) \mathbf{f}(\mathbf{x}, \mathbf{u}; t) \quad (3.19)$$

Replacing Eq. (3.19) in Eq. (3.18) the system of backward equations is rewritten in the following way:

$$\left\{ \begin{aligned} -\frac{d\vartheta}{dt} &= \boldsymbol{\alpha}^T \mathbf{H}_u + \frac{1}{2} \boldsymbol{\alpha}^T \mathbf{H}_{uu} \boldsymbol{\alpha} \\ -\frac{d\mathbf{J}_x^*}{dt} &= [\mathbf{H}_x + \mathbf{H}_u \boldsymbol{\beta} + \boldsymbol{\alpha}^T \mathbf{H}_{uu} \boldsymbol{\beta} + \boldsymbol{\alpha}^T \mathbf{H}_{ux} + \mathbf{J}_{xx}^* \mathbf{f}_u \boldsymbol{\alpha}] \\ -\frac{d\mathbf{J}_{xx}^*}{dt} &= [\mathbf{H}_{xx} + \boldsymbol{\beta}^T \mathbf{H}_{uu} + 2 \mathbf{H}_{xu} \boldsymbol{\beta} + 2 \mathbf{J}_{xx}^* (\mathbf{f}_x + \mathbf{f}_u \boldsymbol{\beta})] \end{aligned} \right. \quad (3.20)$$

The optimal control variation coefficients can be expressed in terms of the Hamiltonian function as well:

$$\boldsymbol{\alpha} = -\mathbf{H}_{uu}^{-1} \mathbf{H}_u, \quad \boldsymbol{\beta} = -\mathbf{H}_{uu}^{-1} (\mathbf{H}_{ux} + \mathbf{f}_u^T \mathbf{J}_{xx}^*) \quad (3.21)$$

The system of partial differential equations is solved backward starting from a set of boundary conditions known at the final time.

$$\left\{ \begin{aligned} \vartheta(t_f) &= 0 \\ \mathbf{J}_x^*(t_f) &= \bar{\mathbf{F}}_x(\bar{\mathbf{x}}(t_f); t_f) \\ \mathbf{J}_{xx}^*(t_f) &= \bar{\mathbf{F}}_{xx}(\bar{\mathbf{x}}(t_f); t_f) \end{aligned} \right. \quad (3.22)$$

where t_f represents the final time, and $\bar{\mathbf{x}}(t_f)$ is the spacecraft nominal trajectory evaluated at the final time. If the *global* version of the DDP algorithm is considered, the equations can be rearranged considering that the optimal control variation admits only the linear feedback term, $\boldsymbol{\alpha} = 0$, and the value of the Hamiltonian partial derivative with respect to the control is equal to 0. In the *global* version of the DDP algorithm the second-order Taylor expansion is not considered starting from the nominal control and trajectory, but starting from a pseudo-optimal control that is the value of the control minimising the expanded HJB expression in Eq. (3.10) where all the variations are set equal to 0. The problem to be solved to derive the pseudo-optimal control, \mathbf{u}^* , is the following:

$$\mathbf{u}^* = \min_{\mathbf{u}} \left[L(\mathbf{x}, \mathbf{u}; t) + \mathbf{J}_x^{*T} \mathbf{f}(\mathbf{x}, \mathbf{u}; t) \right] = \min_{\mathbf{u}} H(\mathbf{x}, \mathbf{u}; t) \quad (3.23)$$

The minimising control is obtained differentiating the right-hand side of Eq. (3.23) assuming that the Hessian matrix with respect to the control function, \mathbf{H}_{uu} , is positive-definite.

$$\mathbf{H}_u(\mathbf{x}, \mathbf{u}^*; t) = \mathbf{L}_u(\mathbf{x}, \mathbf{u}^*; t) + \mathbf{J}_x^* \mathbf{f}_u(\mathbf{x}, \mathbf{u}^*; t) = 0 \quad (3.24)$$

The first-order necessary conditions of optimality ensures that the expression defined in Eq. (3.24) is equal to 0 when it is evaluated in \mathbf{u}^* . In the *global* version of the algorithm all the variables must be evaluated in the pseudo-optimal control $\tilde{\mathbf{u}}$, and if it is replaced in the function α defined in Eq. (3.13) or in Eq. (3.21) the result is 0. The derivation of the continuous DDP algorithm in its *global* version can be found in [39].

3.2 Mathematical derivation of the DDP discrete version

The extension of the DDP algorithm to the satisfaction of endpoint constraints and static/dynamic variables will be defined in this section considering only the discrete formulation of the DDP algorithm. The notation used for the second-order Taylor expansions follows [43], while the introduction of the auxiliary matrices and the final expression of the backward equations follows [42]. The HJB equation defined in Eq. (3.3) also admits a discrete version [39]:

$$J_i^*(\mathbf{x}_i, \mathbf{b}; t_i) = \min_{\mathbf{u}_i} [L_i(\mathbf{x}_i, \mathbf{u}_i; t_i) + J_{i+1}^*(\mathbf{x}_{i+1}, \mathbf{b}; t_{i+1})] \quad (3.25)$$

where J_{i+1} represents the optimal value function obtained at the successive step t_{i+1} , and \mathbf{b}_i is the vector of Lagrange multipliers used to adjoin the endpoint constraints to the cost function using the same notation as in [39], L , for the definition of the augmented value function, J . In this work the discrete version of the DDP algorithm is used so that the continuous system dynamics, \mathbf{f} , is replaced by the transition function, \mathbf{F} . The transition function is a function which maps the state at step \mathbf{x}_i to the state at the following step \mathbf{x}_{i+1} . In this work a generic Runge-Kutta explicit integration numerical scheme is used as transition function because it transforms the continuous system dynamics into a system of discrete equations, and compute each state of the dynamics marching forward starting from the initial condition.

$$\mathbf{x}_{i+1} = \mathbf{F}_i(\mathbf{x}_i, \mathbf{u}_i; t_i) \quad (3.26)$$

The starting point for the derivation of DDP equations is to define the linear-quadratic expansion of the HJB equation expressing the optimal variables as the sum of a nominal value and a variation with respect to the nominal variable:

$$\delta \mathbf{x}_i = \mathbf{x}_i - \bar{\mathbf{x}}_i \quad \delta \mathbf{u}_i = \mathbf{u}_i - \bar{\mathbf{u}}_i \quad \delta \mathbf{b} = \mathbf{b} - \bar{\mathbf{b}} \quad \delta \mathbf{x}_{i+1} = \mathbf{x}_{i+1} - \bar{\mathbf{x}}_{i+1} \quad (3.27)$$

where $\{\bar{\mathbf{u}}_i\}$ is a nominal policy used as first guess solution, $\{\bar{\mathbf{x}}_i\}$ is the relative trajectory obtained using the nominal policy, and $\bar{\mathbf{b}}$ is the initial guess for the Lagrange multipliers. The use of the curly brackets $\{\bar{\mathbf{u}}_i\}$ as in [39] denotes the collection of control actions used for each time step, while \mathbf{u}_i describes the control action at the single time step i . Analogously, $\{\bar{\mathbf{x}}_i\}$ represents the whole trajectory while \mathbf{x}_i is the satellite state at the single time step i . Replacing the variations introduced in Eq. (3.27) in Eq. (3.25) the result is:

$$\begin{aligned} J_i^*(\bar{\mathbf{x}}_i + \delta \mathbf{x}_i, \bar{\mathbf{b}} + \delta \mathbf{b}; t_i) = \min_{\mathbf{u}_i} [L_i(\bar{\mathbf{x}}_i + \delta \mathbf{x}_i, \bar{\mathbf{u}}_i + \delta \mathbf{u}_i; t_i) \\ + J_{i+1}^*(\bar{\mathbf{x}}_{i+1} + \delta \mathbf{x}_{i+1}, \bar{\mathbf{b}} + \delta \mathbf{b}; t_{i+1})] \end{aligned} \quad (3.28)$$

At this point it is possible to expand each term in Eq. (3.28) in Taylor series stopping at the second order around the point $(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{b}})$. The dependence on time will be omitted to simplify the notation [39] and the optimal cost function J^* will be written as J :

$$\begin{aligned} J_i^*(\bar{\mathbf{x}}_i + \delta\mathbf{x}_i, \bar{\mathbf{b}} + \delta\mathbf{b}) &= J_i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}) + \mathbf{J}_x^T(\bar{\mathbf{x}}_i, \bar{\mathbf{b}})\delta\mathbf{x}_i + \mathbf{J}_b^T(\bar{\mathbf{x}}_i, \bar{\mathbf{b}})\delta\mathbf{b} \\ &+ \frac{1}{2}\delta\mathbf{x}_i^T \mathbf{J}_{xx}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}})\delta\mathbf{x}_i + \frac{1}{2}\delta\mathbf{b}^T \mathbf{J}_{bb}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}})\delta\mathbf{b} + \delta\mathbf{x}_i^T \mathbf{J}_{xb}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}})\delta\mathbf{b} \end{aligned} \quad (3.29)$$

$$\begin{aligned} J_{i+1}(\bar{\mathbf{x}}_{i+1} + \delta\mathbf{x}_{i+1}, \bar{\mathbf{b}} + \delta\mathbf{b}) &= J_{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}) + \mathbf{J}_x^{i+1T}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}})\delta\mathbf{x}_{i+1} \\ &+ \mathbf{J}_b^{i+1T}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}})\delta\mathbf{b} + \frac{1}{2}\delta\mathbf{x}_{i+1}^T \mathbf{J}_{xx}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}})\delta\mathbf{x}_{i+1} + \frac{1}{2}\delta\mathbf{b}^T \mathbf{J}_{bb}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}})\delta\mathbf{b} \\ &+ \delta\mathbf{x}_{i+1}^T \mathbf{J}_{xb}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}})\delta\mathbf{b} \end{aligned} \quad (3.30)$$

$$\begin{aligned} L_i(\bar{\mathbf{x}}_i + \delta\mathbf{x}_i, \bar{\mathbf{u}}_i + \delta\mathbf{u}_i) &= L_i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) + \mathbf{L}_x^T(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i)\delta\mathbf{x}_i + \mathbf{L}_u^T(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i)\delta\mathbf{u}_i \\ &+ \frac{1}{2}\delta\mathbf{x}_i^T \mathbf{L}_{xx}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i)\delta\mathbf{x}_i + \frac{1}{2}\delta\mathbf{u}_i^T \mathbf{L}_{uu}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i)\delta\mathbf{u}_i + \delta\mathbf{x}_i^T \mathbf{L}_{xu}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i)\delta\mathbf{u}_i \end{aligned} \quad (3.31)$$

where \mathbf{J}_x^T and \mathbf{J}_b^T are the gradient of the value function with respect to the state and the Lagrange multipliers respectively, \mathbf{J}_{xx}^i and \mathbf{J}_{bb}^i are the Hessian matrices of the value function with respect to the state and the Lagrangian multipliers respectively, and \mathbf{J}_{xb}^i is the mixed Hessian matrix computed first with respect to the state and the Lagrange multipliers. The same notation is applied for the partials of the cost function, L_i , replacing the Lagrange multipliers with the control policy. To complete the linear-quadratic expansion of Eq. (3.28), the state variation of the successive step, $\delta\mathbf{x}_{i+1}$, must be expressed in terms of the current state variation, $\delta\mathbf{x}_i$. This can be done using the expression of the dynamics of the system.

$$\begin{aligned} \delta\mathbf{x}_{i+1} &= \mathbf{x}_{i+1} - \bar{\mathbf{x}}_{i+1} = \mathbf{F}^i(\bar{\mathbf{x}}_i + \delta\mathbf{x}_i, \bar{\mathbf{u}}_i + \delta\mathbf{u}_i) - \mathbf{F}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) \\ &= \mathbf{F}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) + \mathbf{F}_x^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i)\delta\mathbf{x}_i + \mathbf{F}_u^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i)\delta\mathbf{u}_i + \frac{1}{2}\delta\mathbf{x}_i^T \mathbf{F}_{xx}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i)\delta\mathbf{x}_i \\ &+ \frac{1}{2}\delta\mathbf{u}_i^T \mathbf{F}_{uu}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i)\delta\mathbf{u}_i + \delta\mathbf{x}_i^T \mathbf{F}_{xu}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i)\delta\mathbf{u}_i \end{aligned} \quad (3.32)$$

where \mathbf{F}_x^i and \mathbf{F}_u^i are the Jacobian matrices of the system dynamics with respect to the state and the control policy respectively, \mathbf{F}_{xx}^i and \mathbf{F}_{uu}^i are the Hessian matrices of the system dynamics with respect to the state and the control policy respectively, and \mathbf{F}_{xu}^i is the mixed Hessian matrix computed first with respect to the state and the control policy. All the Hessian matrices associated to the system dynamics are third-order tensors which are built considering the Hessian matrix of each scalar component of the vectorial function, \mathbf{F} , and are computed analytically. At this point it is possible to replace Eqs. (3.29-3.31) in Eq. (3.28) to obtain the final expanded HJB equation:

$$\begin{aligned} &J_i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}) + \mathbf{J}_x^T(\bar{\mathbf{x}}_i, \bar{\mathbf{b}})\delta\mathbf{x}_i + \mathbf{J}_b^T(\bar{\mathbf{x}}_i, \bar{\mathbf{b}})\delta\mathbf{b} + \frac{1}{2}\delta\mathbf{x}_i^T \mathbf{J}_{xx}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}})\delta\mathbf{x}_i + \\ &+ \frac{1}{2}\delta\mathbf{b}^T \mathbf{J}_{bb}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}})\delta\mathbf{b} + \delta\mathbf{x}_i^T \mathbf{J}_{xb}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}})\delta\mathbf{b} = \min_{\delta\mathbf{u}_i} [J_i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) + J_{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}) + \mathbf{E}_i\delta\mathbf{x}_i \\ &+ \mathbf{D}_i\delta\mathbf{u}_i + \mathbf{J}_b^{i+1T}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}})\delta\mathbf{b} + \delta\mathbf{x}_i^T \mathbf{A}_i\delta\mathbf{x}_i + \delta\mathbf{x}_i^T \mathbf{B}_i\delta\mathbf{u}_i + \delta\mathbf{u}_i^T \mathbf{C}_i\delta\mathbf{u}_i \\ &+ \delta\mathbf{b}^T \mathbf{R}_i\delta\mathbf{b} + \delta\mathbf{u}_i^T \mathbf{K}_i\delta\mathbf{b} + \delta\mathbf{x}_i^T \mathbf{X}_i\delta\mathbf{b}] \end{aligned} \quad (3.33)$$

To simplify the notation some matrices and vectors are introduced following the notation in Colombo et al. [42].

$$\begin{aligned}
 \mathbf{A}_i &= \frac{1}{2} \left[\mathbf{L}_{xx}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) + \mathbf{J}_x^{i+1\top}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}) \mathbf{F}_{xx}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) + \mathbf{F}_x^{i\top}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) \mathbf{J}_{xx}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}) \mathbf{F}_x^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) \right] \\
 \mathbf{B}_i &= \left[\mathbf{L}_{xu}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) + \mathbf{J}_x^{i+1\top}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}) \mathbf{F}_{xu}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) + \mathbf{F}_x^{i\top}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) \mathbf{J}_{xu}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}) \mathbf{F}_u^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) \right] \\
 \mathbf{C}_i &= \frac{1}{2} \left[\mathbf{L}_{uu}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) + \mathbf{J}_x^{i+1\top}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}) \mathbf{F}_{uu}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) + \mathbf{F}_u^{i\top}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) \mathbf{J}_{uu}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}) \mathbf{F}_u^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) \right] \\
 \mathbf{D}_i &= \left[\mathbf{L}_u^{i\top}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) + \mathbf{J}_x^{i+1\top}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}) \mathbf{F}_u^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) \right] \\
 \mathbf{E}_i &= \left[\mathbf{L}_x^{i\top}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) + \mathbf{J}_x^{i+1\top}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}) \mathbf{F}_x^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) \right] \\
 \mathbf{K}_i &= \mathbf{F}_u^{i\top}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) \mathbf{J}_{xb}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}) \\
 \mathbf{X}_i &= \mathbf{F}_x^{i\top}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) \mathbf{J}_{xb}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}})
 \end{aligned} \tag{3.34}$$

At this point the minimisation problems has to be solved. The operation is carried out differentiating the right hand side of Eq. (3.33) with respect to the control variation so that the optimal control variation can be determined [42].

$$\frac{\partial}{\partial(\delta \mathbf{u}_i)} = \mathbf{D}_i^\top + 2\mathbf{C}_i \delta \mathbf{u}_i^* + \mathbf{B}_i^\top \delta \mathbf{x}_i + \mathbf{K}_i \delta \mathbf{b} = 0 \tag{3.35}$$

Solving for the optimal control variation it is possible to derive a feedback control law which depends on the variation of the state and on the variation of the Lagrange multipliers.

$$\delta \mathbf{u}_i^* = \boldsymbol{\alpha}_i + \boldsymbol{\beta}_i \delta \mathbf{x}_i + \boldsymbol{\gamma}_i \delta \mathbf{b} \tag{3.36}$$

$$\boldsymbol{\alpha}_i = -\frac{1}{2} \mathbf{C}_i^{-1} \mathbf{D}_i^\top \quad \boldsymbol{\beta}_i = -\frac{1}{2} \mathbf{C}_i^{-1} \mathbf{B}_i^\top \quad \boldsymbol{\gamma}_i = -\frac{1}{2} \mathbf{C}_i^{-1} \mathbf{K}_i \tag{3.37}$$

It is possible to replace the optimal control variation in the right hand side of Eq. (3.33) and remove the minimisation operator. After some mathematical manipulations the final result is the following [42]:

$$\begin{aligned}
 &J_i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}) + \mathbf{J}_x^{i\top}(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}) \delta \mathbf{x}_i + \mathbf{J}_b^{i\top}(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}) \delta \mathbf{b} + \frac{1}{2} \delta \mathbf{x}_i^\top \mathbf{J}_{xx}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}) \delta \mathbf{x}_i + \frac{1}{2} \delta \mathbf{b}^\top \mathbf{J}_{bb}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}) \delta \mathbf{b} \\
 &+ \delta \mathbf{x}_i^\top \mathbf{J}_{xb}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}) \delta \mathbf{b} = L_i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) + J_{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}) - \frac{1}{4} \mathbf{D}_i \mathbf{C}_i^{-1} \mathbf{D}_i^\top + \mathbf{E}_i \delta \mathbf{x}_i \\
 &- \frac{1}{2} \mathbf{D}_i \mathbf{C}_i^{-1} \mathbf{B}_i^\top \delta \mathbf{x}_i + \mathbf{J}_b^{i+1\top}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}) \delta \mathbf{b} - \frac{1}{2} \mathbf{D}_i \mathbf{C}_i^{-1} \mathbf{K}_i \delta \mathbf{b} + \delta \mathbf{x}_i^\top \mathbf{A}_i \delta \mathbf{x}_i + \delta \mathbf{b}^\top \mathbf{R}_i \delta \mathbf{b} \\
 &- \frac{1}{4} \delta \mathbf{x}_i^\top \mathbf{B}_i \mathbf{C}_i^{-1} \mathbf{B}_i^\top \delta \mathbf{x}_i - \frac{1}{4} \delta \mathbf{b}^\top \mathbf{K}_i^\top \mathbf{C}_i^{-1} \mathbf{K}_i \delta \mathbf{b} + \delta \mathbf{x}_i^\top \mathbf{X}_i \delta \mathbf{b}_i - \frac{1}{2} \delta \mathbf{x}_i^\top \mathbf{B}_i \mathbf{C}_i^{-1} \mathbf{K}_i \delta \mathbf{b}
 \end{aligned} \tag{3.38}$$

The last operation to be carried out before deriving the final equations is to replace the optimal value function $J_i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}})$ with a nominal value function because the optimal one is not known. Therefore, it is possible to introduce a new variable, ϑ_i , which represents the error between the nominal value function computed at step i and the optimal value function at the same step [42].

$$\vartheta_i = J_i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}) - \bar{J}_i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}) \tag{3.39}$$

Moreover, it is possible to rewrite the discrete version of the HJB without the minimisation operator if the relation is applied to the nominal value function instead of the optimal value function:

$$\bar{J}_i(\mathbf{x}_i, \mathbf{b}) = L_i(\mathbf{x}_i, \mathbf{u}_i) + \bar{J}_{i+1}(\mathbf{x}_{i+1}, \mathbf{b}) \quad (3.40)$$

Using Eqs. (3.39-3.40) it is possible to rewrite Eq. (3.38) in the following way [42]:

$$\begin{aligned} \vartheta_i + \mathbf{Q}_i \delta \mathbf{x}_i + \mathbf{Z}_i \delta \mathbf{b} + \delta \mathbf{x}_i^T \mathbf{P}_i \delta \mathbf{x}_i + \delta \mathbf{b}^T \mathbf{R}_i \delta \mathbf{b} + \delta \mathbf{x}_i^T \mathbf{S}_i \delta \mathbf{b} &= \vartheta_{i+1} - \frac{1}{4} \mathbf{D}_i \mathbf{C}_i^{-1} \mathbf{D}_i^T \\ &+ \mathbf{E}_i \delta \mathbf{x}_i - \frac{1}{2} \mathbf{D}_i \mathbf{C}_i^{-1} \mathbf{B}_i^T \delta \mathbf{x}_i + \mathbf{Z}_{i+1} \delta \mathbf{b} - \frac{1}{2} \mathbf{D}_i \mathbf{C}_i^{-1} \mathbf{K}_i \delta \mathbf{b} + \delta \mathbf{x}_i^T \mathbf{A}_i \delta \mathbf{x}_i + \delta \mathbf{b}^T \mathbf{R}_i \delta \mathbf{b} \\ &- \frac{1}{4} \delta \mathbf{x}_i^T \mathbf{B}_i \mathbf{C}_i^{-1} \mathbf{B}_i^T \delta \mathbf{x}_i - \frac{1}{4} \delta \mathbf{b}^T \mathbf{K}_i^T \mathbf{C}_i^{-1} \mathbf{K}_i \delta \mathbf{b} + \delta \mathbf{x}_i^T \mathbf{X}_i \delta \mathbf{b} - \frac{1}{2} \delta \mathbf{x}_i^T \mathbf{B}_i \mathbf{C}_i^{-1} \mathbf{K}_i \delta \mathbf{b} \end{aligned} \quad (3.41)$$

The following additional matrices and vectors have been introduced to represent the partials of the optimal value function which represent the real unknowns to be determined [42]:

$$\begin{aligned} \mathbf{P}_i &= \frac{1}{2} \mathbf{J}_{xx}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}) & \mathbf{P}_{i+1} &= \frac{1}{2} \mathbf{J}_{xx}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}) \\ \mathbf{Q}_i &= \mathbf{J}_x^{iT}(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}) & \mathbf{Q}_{i+1} &= \mathbf{J}_x^{i+1T}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}) \\ \mathbf{Z}_i &= \mathbf{J}_b^{iT}(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}) & \mathbf{Z}_{i+1} &= \mathbf{J}_b^{i+1T}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}) \\ \mathbf{R}_i &= \frac{1}{2} \mathbf{J}_{bb}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}) & \mathbf{R}_{i+1} &= \frac{1}{2} \mathbf{J}_{bb}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}) \\ \mathbf{S}_i &= \mathbf{J}_{xb}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}) & \mathbf{S}_{i+1} &= \mathbf{J}_{xb}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}) \end{aligned} \quad (3.42)$$

The previous matrices defined in Eq. (3.34) change their formulation as well:

$$\begin{aligned} \mathbf{A}_i &= \frac{1}{2} \left[\mathbf{L}_{xx}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) + \mathbf{Q}_{i+1} \mathbf{F}_{xx}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) + 2 \mathbf{F}_x^{iT}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) \mathbf{P}_{i+1} \mathbf{F}_x^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) \right] \\ \mathbf{B}_i &= \left[\mathbf{L}_{xu}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) + \mathbf{Q}_{i+1} \mathbf{F}_{xu}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) + 2 \mathbf{F}_x^{iT}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) \mathbf{P}_{i+1} \mathbf{F}_u^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) \right] \\ \mathbf{C}_i &= \frac{1}{2} \left[\mathbf{L}_{uu}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) + \mathbf{Q}_{i+1} \mathbf{F}_{uu}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) + 2 \mathbf{F}_u^{iT}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) \mathbf{P}_{i+1} \mathbf{F}_u^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) \right] \\ \mathbf{D}_i &= \left[\mathbf{L}_u^{iT}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) + \mathbf{Q}_{i+1} \mathbf{F}_u^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) \right]^T \\ \mathbf{E}_i &= \left[\mathbf{L}_x^{iT}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) + \mathbf{Q}_{i+1} \mathbf{F}_x^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) \right]^T \\ \mathbf{K}_i &= \mathbf{F}_u^{iT}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) \mathbf{S}_{i+1} \\ \mathbf{X}_i &= \mathbf{F}_x^{iT}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) \mathbf{S}_{i+1} \end{aligned} \quad (3.43)$$

The final equations of the "local" DDP algorithm to compute the partial derivatives of the value function are determined equaling the coefficients of the linear quadratic expansion of the same

differentials:

$$\begin{aligned}
 \vartheta_i &= \vartheta_{i+1} - \frac{1}{4} \mathbf{D}_i \mathbf{C}_i^{-1} \mathbf{D}_i^T \\
 \mathbf{Q}_i &= \mathbf{E}_i - \frac{1}{2} \mathbf{D}_i \mathbf{C}_i^{-1} \mathbf{B}_i^T \\
 \mathbf{Z}_i &= \mathbf{Z}_{i+1} - \frac{1}{2} \mathbf{D}_i \mathbf{C}_i^{-1} \mathbf{K}_i \\
 \mathbf{P}_i &= \mathbf{A}_i - \frac{1}{4} \mathbf{B}_i \mathbf{C}_i^{-1} \mathbf{B}_i^T \\
 \mathbf{R}_i &= \mathbf{R}_{i+1} - \frac{1}{4} \mathbf{K}_i^T \mathbf{C}_i^{-1} \mathbf{K}_i \\
 \mathbf{S}_i &= \mathbf{X}_i - \frac{1}{2} \mathbf{B}_i \mathbf{C}_i^{-1} \mathbf{K}_i
 \end{aligned} \tag{3.44}$$

The previous equations must be solved backward starting from the final time step to the first one considering the following set of final conditions for the partial derivative of the value functions:

$$\begin{aligned}
 \vartheta_{N+1} &= 0 \\
 \mathbf{Q}_{N+1} &= \mathbf{J}_x^{N+1T}(\bar{\mathbf{x}}_{N+1}, \bar{\mathbf{b}}) \\
 \mathbf{Z}_{N+1} &= \mathbf{J}_b^{N+1T}(\bar{\mathbf{x}}_{N+1}, \bar{\mathbf{b}}) \\
 \mathbf{P}_{N+1} &= \frac{1}{2} \mathbf{J}_{xx}^{N+1}(\bar{\mathbf{x}}_{N+1}, \bar{\mathbf{b}}) \\
 \mathbf{R}_{N+1} &= \frac{1}{2} \mathbf{J}_{bb}^{N+1}(\bar{\mathbf{x}}_{N+1}, \bar{\mathbf{b}}) \\
 \mathbf{S}_{N+1} &= \mathbf{J}_{xb}^{N+1}(\bar{\mathbf{x}}_{N+1}, \bar{\mathbf{b}})
 \end{aligned} \tag{3.45}$$

3.2.1 Step-size adjustment method

In this paragraph the step-size adjustment method adopted for the derivation of the following results is prepared. The procedure follows the original work in [39] with the addition of some modifications introduced by the author to improve the performance of the algorithm. The step-size adjustment method is a procedure to adjust the magnitude of the control variation such that the linear-quadratic Taylor expansion is valid. Starting from the initial condition, for each time step in the forward numerical integration it is required to evaluate the difference between the state computed with the nominal control policy and the one obtained using the optimal control.

$$\delta \mathbf{x}_{i+1} = \mathbf{x}_{i+1} - \bar{\mathbf{x}}_{i+1} = \mathbf{F}_i(\mathbf{x}_i, \mathbf{u}_i) - \mathbf{F}_i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) \tag{3.46}$$

The next step is to compute the new optimal policy that is obtained summing the optimal control variation to the nominal control:

$$\mathbf{u}_i = \bar{\mathbf{u}}_i + \delta \mathbf{u}_i = \bar{\mathbf{u}}_i + \boldsymbol{\alpha}_i + \boldsymbol{\beta}_i \delta \mathbf{x}_i \tag{3.47}$$

The addendum related to the variation of the Lagrange multipliers is omitted at the moment because as it will be explained in the next section the first loop of the DDP algorithm considers the Lagrange multipliers constantly equal to the nominal value. Once the optimal policy is

derived, there will be a *nominal trajectory* resulting from the forward integration of the dynamics using the nominal policy, and an *optimal trajectory* resulting from the forward integration of the dynamics using the optimal policy as shown in Figures 3.1 and 3.2. The last step is to evaluate the difference in the value function computed using the nominal trajectory and policy and the optimal ones. This difference must be compared with the value of ϑ_1 , which represents the estimated difference between the optimal cost and the nominal cost. If the two quantities are not comparable, this means that the linear-quadratic Taylor expansion is not valid causing the algorithm to diverge. It is convenient to define a criterion to decide if the difference between the optimal and nominal value functions is too large taking their ratio and imposing that is at least higher than a constant value that is put conventionally equal to 0.5 [39].

$$\frac{J(\mathbf{x}, \bar{\mathbf{b}}) - \bar{J}(\bar{\mathbf{x}}, \bar{\mathbf{b}})}{\vartheta_1} \leq c, \quad c = 0.5 \quad (3.48)$$

It is obvious that the difference between the optimal value function and the nominal one is supposed to be negative. If this is not occurring the new optimal policy is not addressing a reduction in the value function. It can happen that the new optimal policy leads to a variation in cost that is too large making the Taylor expansion no longer valid. In this case it is required to decrease the magnitude of the control variation such that that linear quadratic-expansion is consistent. This is carried out ensuring that the largest terms of the optimal control variation, α_i , should be small enough to be of the same order of magnitude of the truncation error of the Taylor expansion. The new control policy is rewritten introducing a coefficient, ε_1 , to tune the magnitude of the control variation:

$$\delta \mathbf{u}_i = \varepsilon_1 \boldsymbol{\alpha}_i + \boldsymbol{\beta}_i \delta \mathbf{x}_i, \quad 0 < \varepsilon_1 \leq 1 \quad (3.49)$$

The reduction of the control policy will decrease also the variation between the new optimal trajectory and the nominal trajectory. The new difference between the optimal value function and the nominal value function should not be compared with ϑ_1 which does not take into account the introduction of the tuning coefficient ε_1 . The new expected error between the optimal value function and the nominal one is obtained multiplying the original value by the factor $\varepsilon_1 (1 - \frac{\varepsilon_1}{2})$. Therefore, the new criterion to assess if the new control policy should be accepted or not becomes [39]:

$$\frac{J(\mathbf{x}, \bar{\mathbf{b}}; \varepsilon_1) - \bar{J}(\bar{\mathbf{x}}, \bar{\mathbf{b}})}{\varepsilon_1 (1 - \frac{\varepsilon_1}{2}) \vartheta_1} \leq c, \quad 0.5 < c < 1, \quad 0 < \varepsilon_1 \leq 1 \quad (3.50)$$

Differently from the standard selection criterion [39], the value of the constant c is allowed to vary within a given interval. The ratio described in Eq. (3.50) describes how the actual cost reduction is departing from the expected cost which is computed from the truncated Taylor expansion. The higher is the value of the threshold, c , the better is the accuracy of the actual cost function with its quadratic approximation. This way it is possible to be sure that the control variation is not relatively large such that the linear-quadratic expansion provided is no longer valid. However, the reduction in the control variation makes the algorithm slower despite of an increase in convergence stability. One way to choose the value of the tuning coefficient ε_1 is to halve the precedent value starting from 1.

$$\varepsilon_1^{z+1} = \frac{\varepsilon_i^z}{2}, \quad \varepsilon_1^0 = 1 \quad (3.51)$$

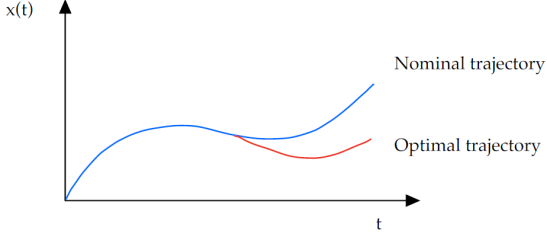


Figure 3.1: Nominal vs optimal trajectory representation during the step-size adjustment method

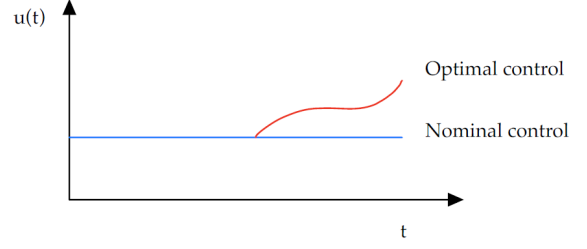


Figure 3.2: Nominal vs optimal control representation during the step-size adjustment method

The second part of the DDP algorithm is related to the constraints verification. This part is carried out after a first loop iteration where the optimal control policy is derived fixing the value of the Lagrange multipliers. Therefore, the value of the nominal control will be almost equal to the optimal control. The complete set of Eq. (3.44) is solved backward to compute the missing matrices related to the partial derivatives of the value function with respect to the Lagrange multipliers. It is required to define the optimal value of the Lagrange multipliers variation to be replaced in the optimal control policy. This operation is carried out starting from a Taylor expansion of the value function to check how a small multiplier variation, $\delta \mathbf{b}$, affects the value function.

$$J_1(\bar{\mathbf{x}}, \bar{\mathbf{b}} + \delta \mathbf{b}) = J_1(\bar{\mathbf{x}}, \bar{\mathbf{b}}) + \mathbf{J}_b^T(\bar{\mathbf{x}}, \bar{\mathbf{b}}) \delta \mathbf{b} + \frac{1}{2} \delta \mathbf{b}^T \mathbf{J}_{bb}^1(\bar{\mathbf{x}}, \bar{\mathbf{b}}) \delta \mathbf{b} \quad (3.52)$$

with J_1 denoting the value of the optimal cost function at the first time step, while \mathbf{J}_b^1 and \mathbf{J}_{bb}^1 are the Jacobian and Hessian matrices of the optimal cost functional with respect to the Lagrange multipliers vector at the initial time step, respectively. The optimal variation of the Lagrange multipliers can be computed differentiating the linear-quadratic expansion and putting the result equal to zero as shown in Gershwin and Jacobson [40].

$$\frac{\partial}{\partial(\delta \mathbf{b})} = \mathbf{J}_b^1(\bar{\mathbf{x}}, \bar{\mathbf{b}}) + \mathbf{J}_{bb}^1(\bar{\mathbf{x}}, \bar{\mathbf{b}}) \delta \mathbf{b} = 0, \quad \delta \mathbf{b} = -[\mathbf{J}_{bb}^1]^{-1}(\bar{\mathbf{x}}, \bar{\mathbf{b}}) \mathbf{J}_b^1(\bar{\mathbf{x}}, \bar{\mathbf{b}}) \quad (3.53)$$

where again Eq. (3.39) has been used to express the optimal cost function as sum of the nominal cost functional, \bar{J}_1 , and the expected error between the optimal cost and the nominal one, ϑ_1 , both evaluated at the initial time step. Once the optimal Lagrange multipliers variation is obtained the new control variation is computed such that a reduction in the constraints violation is obtained. A new forward integration of the dynamics is carried out considering the optimal policy and a new difference between the optimal value function and the nominal value function can be evaluated. This time, the difference should be compared with a measure of the error between the optimal cost using the new set of Lagrange multipliers and the nominal value of the cost function. Such figure of merit can be obtained starting from Eq. (3.52) [40].

$$J_1(\bar{\mathbf{x}}, \bar{\mathbf{b}} + \delta \mathbf{b}) = \bar{J}_1(\bar{\mathbf{x}}, \bar{\mathbf{b}}) + \vartheta_1 + \mathbf{J}_b^T(\bar{\mathbf{x}}, \bar{\mathbf{b}}) \delta \mathbf{b} + \frac{1}{2} \delta \mathbf{b}^T \mathbf{J}_{bb}^1(\bar{\mathbf{x}}, \bar{\mathbf{b}}) \delta \mathbf{b} \quad (3.54)$$

Taking the nominal cost at the left-hand side and replacing the value of the optimal Lagrange

multipliers variation, it is possible to compute the expected variation in the cost function associated to the optimal Lagrange multiplier variation, $\delta \mathbf{b}$:

$$\begin{aligned} J_1(\bar{\mathbf{x}}, \bar{\mathbf{b}} + \delta \mathbf{b}) - \bar{J}_1(\bar{\mathbf{x}}, \bar{\mathbf{b}}) &= \vartheta_1 - \mathbf{J}_b^{1T}(\bar{\mathbf{x}}, \bar{\mathbf{b}}) [\mathbf{J}_{bb}^1]^{-1}(\bar{\mathbf{x}}, \bar{\mathbf{b}}) \mathbf{J}_b^1(\bar{\mathbf{x}}, \bar{\mathbf{b}}) \\ &+ \frac{1}{2} \mathbf{J}_b^{1T}(\bar{\mathbf{x}}, \bar{\mathbf{b}}) [\mathbf{J}_{bb}^1]^{-1}(\bar{\mathbf{x}}, \bar{\mathbf{b}}) \mathbf{J}_b^1(\bar{\mathbf{x}}, \bar{\mathbf{b}}) \end{aligned} \quad (3.55)$$

Using the matricial notation introduced in Eq. (3.45) it is possible to rewrite the expression of the optimal Lagrange multipliers variation and the expected cost reduction as follows:

$$\delta \mathbf{b} = -\frac{1}{2} \mathbf{R}_1^{-1} \mathbf{Z}_1^T \quad (3.56)$$

$$J_1(\bar{\mathbf{x}}, \bar{\mathbf{b}} + \delta \mathbf{b}) - \bar{J}_1(\bar{\mathbf{x}}, \bar{\mathbf{b}}) = \vartheta_1 - \frac{1}{2} \mathbf{Z}_1 \mathbf{R}_1^{-1} \mathbf{Z}_1^T + \frac{1}{4} \mathbf{Z}_1 \mathbf{R}_1^{-1} \mathbf{Z}_1^T = \vartheta_1 - \frac{1}{4} \mathbf{Z}_1 \mathbf{R}_1^{-1} \mathbf{Z}_1^T \quad (3.57)$$

There is the possibility that the variation in the value function is not consistent with Eq. (3.54) within a certain tolerance. This implies that the variation of the Lagrange multipliers is so large to make the Taylor expansion no longer valid. Also in this case a tuning coefficient, ε_2 , can be introduced to control the magnitude of the Lagrange multipliers variation:

$$\delta \mathbf{b} = -\frac{1}{2} \varepsilon_2 \mathbf{R}_1^{-1} \mathbf{Z}_1^T, \quad 0 < \varepsilon_2 \leq 1 \quad (3.58)$$

The estimation of the value function reduction changes as well after the introduction of the tuning coefficient [42]:

$$J_1(\bar{\mathbf{x}}, \bar{\mathbf{b}} + \delta \mathbf{b}) - \bar{J}_1(\bar{\mathbf{x}}, \bar{\mathbf{b}}) = \vartheta_1 - \frac{1}{2} \left(\varepsilon_2 - \frac{\varepsilon_2^2}{2} \right) \mathbf{Z}_1 \mathbf{R}_1^{-1} \mathbf{Z}_1^T \quad (3.59)$$

3.2.2 Hessian positive definiteness

In this paragraph some heuristic techniques to ensure the positive definiteness of the Hessian matrix with respect to the control to achieve the convergence are reported. First, a summary of the existing techniques provided by the literature is provided. Finally, a personal heuristic technique devised by the author and implemented in this research work is presented. The DDP algorithm points to the minimisation of the cost functional just if the matrix \mathbf{C}_i is positive definite. Indeed, this is a necessary condition of optimality to achieve a local minimum using the optimal control variation. It can happen that the matrix \mathbf{C}_i is not positive definite or it is almost semi-positive definite due to the low order of magnitude of the numbers inside the matrix. The literature presents several techniques to guarantee the positive definiteness of the Hessian matrix:

- Constant shifting
- Adaptive shifting
- Trust-region

All the techniques share the same principle to find a positive scalar to ensure that the matrix \mathbf{C}_i is positive definite shifting its main diagonal. The constant shifting was first proposed by Yakowitz and Rutherford [46] sums the original Hessian matrix to the identity matrix multiplied by a positive constant factor decided by the user a priori. The original matrix \mathbf{C}_i is used for the computation of the backward equations stated in Eq. (3.34) while the corrected matrix $\tilde{\mathbf{C}}_i$ is adopted for the derivation of the coefficient multiplying the control variation.

$$\tilde{\mathbf{C}}_i = \mathbf{C}_i + \varepsilon_c \mathbf{I}_{3 \times 3} \quad (3.60)$$

The adaptive shifting was introduced by Liao and Shoemaker [84] as modification of the constant shifting. In this approach the positive scalar is decided slightly larger than the minimum eigenvalue of the Hessian matrix. A common choice as shifting term is the absolute value of the minimum eigenvalue of the matrix \mathbf{C}_i to ensure positive definiteness following Colombo et al. [42].

$$\tilde{\mathbf{C}}_i = \mathbf{C}_i + 2|\lambda_{\min}| \mathbf{I}_{3 \times 3} \quad (3.61)$$

The last technique is used by Lantoine and Russell [43] in the definition of its HDDP and it is the trust region sub-problem. This method solves a convex quadratic problem optimising a control variation to not go beyond a given trust-region radius which is delimiting a region of confidence. The outcome of the method is also a positive scalar which is used to shift the main diagonal of the Hessian matrix. However, this time the scalar value is function of the trust-region used for the method which is defined by the user. All the previous methods share one important problem that is the positive scalar used as shift for the main diagonal is provided by the user and it is not adapted according to the problem that must be solved. This is not an optimal way to apply the shifting technique because a small positive value may generate numerical difficulties, while if the positive scalar is too large the convergence can be very slowed. This research work is the definition of a new way to make the Hessian matrix with respect to the control positive definite. The main idea is to find a way to select the positive scalar to shift the main diagonal which is independent from the user selection and it is flexible with respect to the problem to be solved. This can be done exploiting a modified Cholesky factorisation proposed by Schnabel and Eskow [85]. The classic Cholesky factorisation decomposes a positive definite matrix in the product of two lower triangular or upper triangular matrices. The decomposition fails in case the matrix is not positive definite. The modified Cholesky factorisation can also be applied to negative definite matrices. The decomposition will not affect the original matrix but the sum of the matrix itself and a positive diagonal matrix, $\tilde{\mathbf{E}}$, which is shifting the main diagonal. The main aspect of this method is that the value of the diagonal matrix is chosen to satisfy Gorshgorin theorem [85] which is a mathematical condition that can be applied independently from the type of problem and it is not user dependent.

$$\mathbf{C}_i + \tilde{\mathbf{E}} = \tilde{\mathbf{L}}\tilde{\mathbf{L}}^T \quad (3.62)$$

The second important point is that the three matrices \mathbf{A}_i , \mathbf{C}_i and \mathbf{R}_i must be symmetric by construction. This can not happen due to numerical errors introduced by the different addenda in Eq. (3.34). Therefore, it is possible to restore the symmetry property for those matrices adding the transpose of the matrix itself and halving the sum. This heuristics is not reported in previous literature to the best of the author knowledge. Indeed, this heuristic is not introduced to guarantee the convergence of the algorithm but to satisfy the underlying assumptions of the

DDP backward equations derivation.

$$\begin{aligned} \mathbf{A}_i &= \frac{1}{2} (\mathbf{A}_i + \mathbf{A}_i^T) \\ \mathbf{C}_i &= \frac{1}{2} (\mathbf{C}_i + \mathbf{C}_i^T) \\ \mathbf{R}_i &= \frac{1}{2} (\mathbf{R}_i + \mathbf{R}_i^T) \end{aligned} \quad (3.63)$$

3.2.3 Lagrange multipliers corrections

The last important note regards the update of the Lagrange multipliers in the second loop of the DDP algorithm. Indeed, it can happen that one of the components of the Lagrange multipliers variation is negative and larger in magnitude with respect to the same component of the nominal vector of Lagrange multipliers. The result will be that one of the components of the new optimal Lagrange multipliers vector will be negative. This situation should be avoidable because it creates ambiguity in the evaluation of the cost reduction. Indeed, being the value function defined as a positive scalar, a cost reduction is necessarily related either to a functional cost reduction or to a constraints violation reduction. In this case the approach implemented in this thesis merges two different methods present in literature. The first case adjoins the endpoints constraints to the cost function by means of a vector of Lagrange multipliers, and the endpoint constraints are expressed as a quadratic form.

$$J^* = J + \mathbf{b}^T (\mathbf{x} - \mathbf{x}_f)^T (\mathbf{x} - \mathbf{x}_f) \quad (3.64)$$

All the terms appearing in Eq. (3.64) are always positive except for the vector of Lagrange multipliers. The presence of negative Lagrange multiplier could break the logic previously described because it can make the cost functional negative. A negative cost functional creates issues when Eq. (3.59) is evaluated because it can lead to accept iterations which are not consistent with the Taylor expansions. Therefore, it is safer to impose that all the negative components of the new Lagrange multipliers vector are set equal to zero as shown in Eq. (3.65) following Colombo et al. [42].

$$\mathbf{b} = \begin{cases} 0, & \text{if } b_i < 0 \\ \bar{\mathbf{b}} + \delta \mathbf{b}, & \text{otherwise} \end{cases} \quad (3.65)$$

Using Eq. (3.65) the cost functional is always positive even if the update in the Lagrange multipliers variation causes one or more components of the Lagrange multipliers to be negative. The second formulation is assuming that constraints are adjoined to the cost function through a vector of Lagrange multipliers multiplying linearly the endpoints constraints, and a positive penalty term which is multiplying the squared norm of the endpoint constraints as proposed by Lantoiné and Russell [43].

$$J^* = J + \mathbf{b}^T (\mathbf{x} - \mathbf{x}_f) + \sigma ||\mathbf{x} - \mathbf{x}_f||^2 \quad (3.66)$$

In this case even if a negative Lagrange multiplier appears, the presence of the penalty term is guaranteeing that the overall cost is always positive because the relative magnitude of the linear term, $\mathbf{b}^T (\mathbf{x} - \mathbf{x}_f)$, is smaller than the quadratic one creating an embedded safeguard process.

3.3 Constraints formulation

Another important aspect in the definition of the nonlinear control problem is given by the formulation used for the endpoint constraints. The constraints are adjoint with the cost function using Lagrange multipliers creating the final value function used in the derivation of the DDP algorithm. Therefore, the partial derivatives computation and the Lagrange multipliers' initial guess depend on the analytical formulation of the endpoint constraints. The main idea is to ensure that the cost functional is always positive so that the minimisation process translates in the reduction of the cost towards zero. Numerical simulations carried out by the author suggest to use two different formulations according to the orbital problem to be solved. For interplanetary problems the best way to define the endpoint constraints is to use the square of the variation between the actual final trajectory and the prescribed final state. The quadratic constraint formulation grants always that the cost function is positive.

$$\varphi = (\mathbf{x} - \mathbf{x}_f)^T (\mathbf{x} - \mathbf{x}_f) \quad (3.67)$$

where φ is the function expressing the endpoint constraints, \mathbf{x} is the satellite state vector in terms of classic orbital elements $[a, e, i, \Omega, \omega, \nu]$ or modified equinoctial elements $[a, f, g, h, k, \Gamma]$, and \mathbf{x}_f is the final prescribed satellite state vector. In this problem the final cost functional formulation will resemble the formulation in Eq. (3.64). For planetary problems the best way to introduce the endpoint constraints is to rely on the Augmented Lagrangian formulation. In this case the linear variation of the final state with respect to the prescribed state is adjoined thanks to the Lagrange multipliers and an additional term depending on the square of such variation multiplying a positive penalty factor.

$$\varphi = (\mathbf{x} - \mathbf{x}_f) + \sigma \|\mathbf{x} - \mathbf{x}_f\|^2 \quad (3.68)$$

The penalty term is introduced to guarantee that the final cost functional is always positive even if the vector of Lagrange multipliers is negative. The expression of the cost functional used for planetary problems is the one defined in Eq. (3.66). Other formulations for the endpoint constraints can be used, but particular attention should be devoted to the order of magnitude of each component of the vectorial function. This is more important when working with orbital elements.

3.4 Summary of the algorithm

This section summarises how the DDP algorithm proceeds in solving the optimal control problem. A graphical representation of the algorithm is presented in Figure 3.3.

1. Integration of the nominal trajectory

The first step in solving the DDP algorithm is to integrate the system dynamics guessing an initial value for the control law and the vector of Lagrange multipliers. All the tuning parameters such as tolerances and penalty coefficients are decided before the nominal integration of the dynamics. The cost functional is evaluated along the nominal trajectory using the augmented Lagrangian formulation and its value is stored.

2. Backward Sweep

The second step is to evaluate the cost functional partial derivatives. This operation is carried by solving the algebraic backward equations which represent the necessary conditions of optimality. For this purpose the auxiliary matrices introduced in Eq. (3.43) are evaluated at each step and replaced inside the backward equations. The coefficients of the optimal feedback control law (α, β) are evaluated and stored to be used in the next phase of the algorithm.

3. Check global convergence

After the backward sweep it is checked if convergence has been achieved. The value of the expected error at the first time instant, ϑ_1 , is compared with the tolerance provided by the user. If this check is true the second check on the violation of the endpoint constraints is carried out. If also all the endpoint constraints are satisfied according to a threshold given by the user, the algorithm stops. The optimal control law, its coefficients, the optimal cost and vector of Lagrange multipliers are provided as output of the optimisation process. Otherwise it proceeds to the next step.

4. First forward integration

The system dynamics is integrated forward again using the new optimal control law. The new cost functional is evaluated and it is compared to the nominal cost functional to check if the a minimisation is achieved or not. The step-size adjustment method is applied to during this process. The nominal control, cost and trajectory are replaced with the optimal ones coming out from the step-size adjustment procedure.

5. Check local convergence

After the integration of the dynamics with the new optimal control the value of the expected reduction, ϑ_1 , is controlled if it is below the given tolerance or not. If this is not true, a new iteration of the DDP algorithm begins computing a new set of optimal control law coefficients solving the backward equations. Therefore the process moves back to Step 2. If the value of the expected reduction is below the given threshold the first stage of the DDP ends and the algorithm enters the second stage where the correction of the Lagrange multipliers is carried out.

6. Complete backward sweep

At the beginning of the second stage of the DDP a complete backward sweep is carried out using the control law coming from the first stage. This time all the backward equations are used and the optimal control law coefficients (α, β, γ) are evaluated and stored.

7. Lagrange multipliers correction

The next step is to compute the optimal variation of the Lagrange multipliers from Eq. (3.58) and integrate again forward the system dynamics considering the complete optimal control law. The new functional cost is computed and the condition on the consistence of the Lagrange multiplier variation magnitude is checked thanks to Eq. (3.59). The new optimal control variables, state, cost and vector of Lagrange multipliers are replaced with the nominal ones and the first stage of DDP is repeated moving to step 2.

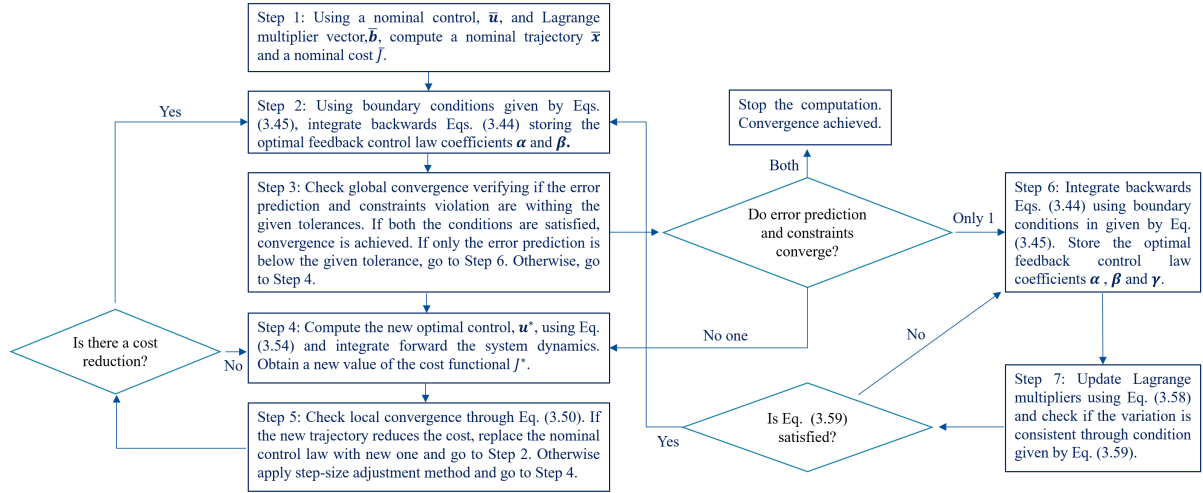


Figure 3.3: Graphical representation of the DDP algorithm.

CHAPTER 4

Heuristic techniques for the constrained problem in using orbital elements

In this chapter the core of this PhD is presented. A mathematical theory to explain what happens when DDP is coupled with orbital elements as state representation of the dynamics is derived. A technique to solve full rendezvous trajectory problems is proposed together with some applications to demonstrate its effectiveness. The work in this chapter was submitted as journal publication in [74].

4.1 Quasi-linear independence

The main scope of this chapter is to present a systematic theory that couples the use of the DDP algorithm with orbital elements as state representation of the orbit dynamics. This theory is based on two fundamental concepts: one related to the nature of orbital elements and the second to the DDP algorithm procedure. The first important point to highlight is the difference in the rate of variation of the orbital elements. This behaviour can be simply justified looking at the expression of Gauss' equations (2.1) presented in Chapter 2. All the orbital elements apart from the true anomaly contain coefficients that are multiplying control actions or perturbing accelerations, while in the true anomaly rate equation there is an addendum which is independent on the control action and it is related to the orbital motion. This means that if the control action at a single time instant is identically equal to zero or is very small, the rate of variation of all the orbital elements will be very small while the variation of the true anomaly will be relatively larger. This behaviour denotes that the system dynamics is stiff. The second aspect that must be stressed is also the reason why the name "quasi-independence" theory has been used to explain what happens when orbital elements are coupled with the DDP algorithm. Consider the last two Gauss' equations defined in Eq. (2.1) describing the rate of variations associated to the

pericentre anomaly and the true anomaly:

$$\begin{aligned}
 \frac{d\omega}{dt} &= \sqrt{\frac{a(1-e^2)}{\mu}} \left\{ \frac{1}{e\sqrt{1+2e\cos\nu+e^2}} \left[2\sin\nu \frac{u_t}{m} + \frac{2e+e^2\cos\nu+\cos\nu}{1+e\cos\nu} \frac{u_n}{m} \right] + \right. \\
 &\quad \left. - \frac{\cos i \sin(\omega+\nu)}{\sin i} \frac{u_h}{1+e\cos\nu} \frac{1}{m} \right\} \\
 \frac{d\nu}{dt} &= \sqrt{\frac{\mu}{[a(1-e^2)]^3}} (1+e\cos\nu)^2 - \sqrt{\frac{a(1-e^2)}{\mu}} \frac{1}{e\sqrt{(1+2e\cos\nu+e^2)}} \left[2\sin\nu \frac{u_t}{m} \right. \\
 &\quad \left. + \frac{2e+e^2\cos\nu+\cos\nu}{1+e\cos\nu} \frac{u_n}{m} \right]
 \end{aligned} \tag{4.1}$$

The part in the true anomaly rate multiplying the control actions is exactly the opposite of the pericentre anomaly rate. This is expected as the sum of the true anomaly and pericentre anomaly results in another angle denoted as true latitude, and both ω and ν are measured on the same plane. Imagining to consider the gradient of the two equations with respect to the control actions the first term in the true anomaly rate equation will disappear and the result will be equal and opposite.

$$\nabla_u \left(\frac{d\omega}{dt} \right) = - \left(\nabla_u \frac{d\nu}{dt} \right) \tag{4.2}$$

The consequence is that it is impossible to define a control policy which is minimising at the same time the pericentre anomaly and the true anomaly constraint. This result can be explained also from a geometrical point of view. Indeed, while each orbital parameter can be described like and independent geometric quantity (i.e., the semi-major axis represents the size of the orbit, the eccentricity represents the shape of the orbit, the inclination is one of the Euler angles with respect to a reference plane, etc.) the true anomaly is an angle defined with respect to a moving reference axis which is defined by the pericentre anomaly itself. This means that if for all the main 5 orbital parameters a single variation is related exclusively to a control action, the true anomaly can change also because it is changing the pericentre anomaly. The consequence is that it is impossible to define a direct cause-effect link between the control action and true anomaly like for all the other orbital elements and this is why even if from a formal point of view all the equations are linearly independent this is not true from a numerical optimisation point of view and from this consideration the name "quasi-independence" theory is proposed. A reduction of the pericentre anomaly will move apart the eccentricity vector and increase the angular distance between the eccentricity vector and the satellite position vector leading to an increase of the true anomaly.

Another important aspect to be underlined is the way the DDP proceeds in its second loop for the reduction of the constraints violation. The DDP computes an optimal Lagrange multipliers variation and defines a new control law to reduce the constraints violation. The higher is the constraint violation the higher will be the action of the control policy in reducing it. This philosophy becomes critical when orbital elements are considered. Indeed, due to the stiff dynamics the true anomaly variation and constraints violation will be always larger with respect to the other orbital elements and will make the linear-quadratic expansion no longer valid. Therefore, the new control law provided by the DDP algorithm will be such that the new trajectory will decrease the true anomaly constraint violation instead of moving the satellite from the original orbit to the final one. This reasoning is logically wrong for two reasons:

1. The true anomaly represent only a way to identify the position of the satellite on the final orbit and is not the main orbital parameter to be achieved.
2. The true anomaly cannot be changed independently using the control action.

4.2 Angular variables management

One of the most important differences in working with orbital elements instead of Cartesian coordinates is the presence of many angular variables in the state vector. Indeed, angular variables are limited in the interval $[0, 2\pi]$ and their rate can be very different. The DDP algorithm defines for each iteration a feedback control law that is based on the difference between the nominal trajectory obtained using a nominal policy and the optimal trajectory evaluated using the new control variation. Therefore, it is fundamental that the evaluation of the variation of each component of the state vector is correct. For Cartesian coordinates where the state is represented by position and velocity which are not limited in any kind of interval the evaluation is straightforward. In case of angular variables there is the possibility to miscalculate this difference because the same angle associated to the nominal trajectory is in the interval $[0, 2\pi]$ while the the same angle computed with the new optimal policy is out of the classic interval. Therefore, in this work a systematic control procedure based on the introduction of two indices has been developed to enforce the proper evaluation of the difference between the optimal and nominal state. The procedure introduces one index, i_{nom}^j , and a second index, i_{opt}^j , which represent integer numbers that are summed to the nominal trajectory and optimal trajectory respectively to recover the real value of each angle. Let $\delta \mathbf{x}_i$ be the state variation computed at the time step i during the forward integration of the dynamics.

$$\delta \mathbf{x}_i = (\mathbf{x}_i + 2\pi i_{\text{opt}}^j) - (\bar{\mathbf{x}}_i + 2\pi i_{\text{nom}}^j) \quad (4.3)$$

with \mathbf{x}_i the spacecraft state evaluated with the feedback control law, $\bar{\mathbf{x}}_i$ the state associated to the nominal control policy. The two indices are set identically equal to zero at the starting time and during the forward integration of the dynamics. According to the control policy adopted, their value is decreased or increased of one unit each time that one angular variable is either negative or larger than 2π respectively.

$$i_{\text{nom}}^j = \begin{cases} i_{\text{nom}}^j - 1, & \text{if } x_j(z) < 0 \\ i_{\text{nom}}^j + 1, & \text{if } x_j(z) \geq 2\pi \end{cases} \quad z = 3, 4, 5, 6 \quad (4.4)$$

where the index z is denoting the four angular variables composing the state vector in terms of orbital elements (inclination, RAAN, pericenter anomaly and true anomaly). The introduction of the index forces at each time step to put the angles in the same interval and to have the proper sign associated to the variation. For example, if the optimal angle at step i is -20 degrees and the nominal value is 40 degrees, the angular variation is equal to -60 degrees. If the index is not introduced and the angle are set automatically in the interval $[0, 2\pi]$ the optimal angle at step j will be 340 degrees resulting in a difference equal to 300 degrees. Even if the value of -60 degrees is equivalent to 300 degrees this is not correct for the application of the DDP algorithm. Indeed, the sign is wrong because the angular variable is decreasing and not increasing and the amplitude of the angular variation is very different. The consequence is that when the state

variation is multiplied by the coefficients of the DDP algorithm the control variation magnitude will be different. Therefore, even if the DDP algorithm is implemented in the correct way a wrong computation of the angular distances can lead to a divergence of the result for a logical error. A flow diagram presenting the control loop introduced for the angular variables thank to the two indices is shown in Figure 4.1.

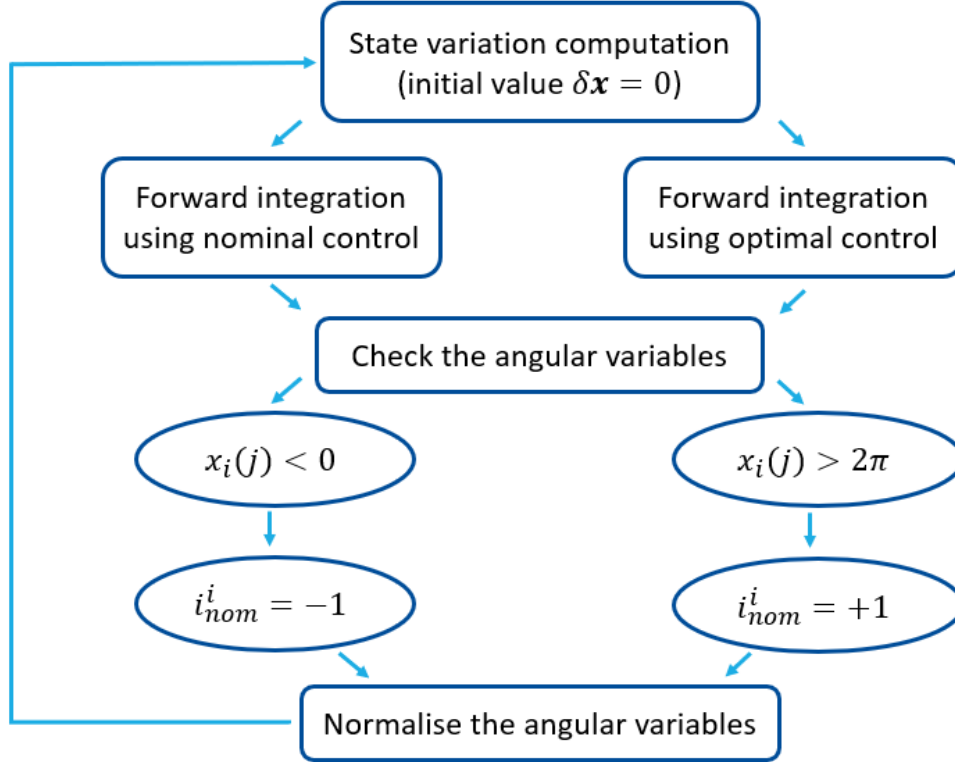


Figure 4.1: Angular index flow diagram.

4.3 Continuation technique

The previous considerations lead to define a systematic procedure based on a continuation scheme for the application of DDP to dynamics expressed in terms of orbital elements. The scheme is divided in 2 main layers:

- i In the first layer the optimal control problem considering all the endpoint constraints except for the one related to the true anomaly is solved.
- ii In the second part a continuation algorithm is applied starting from the optimal control law obtained in the first part used as nominal solution to solve the fully constrained optimal control problem.

The continuation technique has been extensively used in the context of indirect optimisation even if it is not strictly needed as direct convergence to the required value. In case the DDP algorithm is used as optimisation technique the continuation method is a safeguard tool that

must be used to guarantee direct convergence when it cannot be easily achieved. It is important to stress this aspect to avoid to consider the DDP algorithm a weaker optimisation technique with respect to classic indirect optimisation. The introduction of the continuation method in the DDP algorithm is useful also to divide the search set of the optimal solution in two classes. The first class contains candidate solutions for which the direct convergence is achieved and it can be identified considering the first guess solutions which are decreasing the pericenter anomaly and increasing the true anomaly or the other way around. Indeed, in this case the movement of the eccentricity vector is pushing positively the true anomaly toward the target value. The second class contains the candidate solutions for which both the true anomaly and the pericenter anomaly must increase or decrease. In this case the direct converge is more difficult and the continuation method is required to bypass the issue.

At the end of the first layer, the optimal control policy imposes the satellite to arrive at a prescribed final orbit configuration even if the position on the orbit may be wrong. In the second layer, the nominal trajectory of the satellite will be already on the final orbit and the only difference will be represented by the difference in true anomaly. The only detail to stress at this point is to ensure that the true anomaly variation is small enough to make the new computed optimal control law to accept the linear-quadratic expansion. This can be achieved considering a vector of final true anomalies starting from the initial and the final prescribed true anomalies $[\nu_1, \dots, \nu_f]$. The larger the vector of true anomalies, the smaller will be the true anomaly variation. In this way, the new control law will optimise both the true anomaly and the other orbital elements because the satellite is already starting from the final orbit configuration. The continuation scheme aims at solving many optimal control sub-problems considering each time one of the final true anomalies defined in this vector and using the optimal solution of each sub-problem as initial guess for the next one until the real final true anomaly is achieved. One important advice to improve the convergence of the algorithm is to select the initial and arrival dates such that the true anomaly has to decrease/increase with respect to the targeted value while the pericenter anomaly must increase/decrease respectively. This is not a limitation on the application of the algorithm but the exploitation of orbital mechanics described by the orbital elements and must be considered a positive value of orbital elements with respect to the Cartesian coordinates. The proposed approach based on the continuation scheme for the coupling between DDP and orbital elements is correct because no limitations are present for the choice of the initial guess to be used for the algorithm. The continuation scheme is summarised in Figure 4.2 and represents the second layer of the proposed solving procedure. The solution obtained from the first layer by solving the optimal control problem with the endpoints constraints to target the prescribed final orbit is characterised by perfect matching of all the orbital parameters apart from the true anomaly. A new optimal control problem is solved keeping the same endpoint constraints for the target orbit and introducing the last constraint on the final true anomaly which is not equal to the prescribed one but a different value. This intermediate final anomaly is computed starting from the one obtained from the first layer solution and increasing or decreasing its value of small quantity according to the direction to achieve the prescribed target true anomaly. The author suggests 0.5 degrees as reasonable value to be used as variation for the intermediate target true anomaly to ensure the convergence of the algorithm. The optimal control problem is solved, and the intermediate solution replaces the first-guess solution. A new optimal control problem is defined changing again the value of the final true anomaly until this value matches the prescribed one.

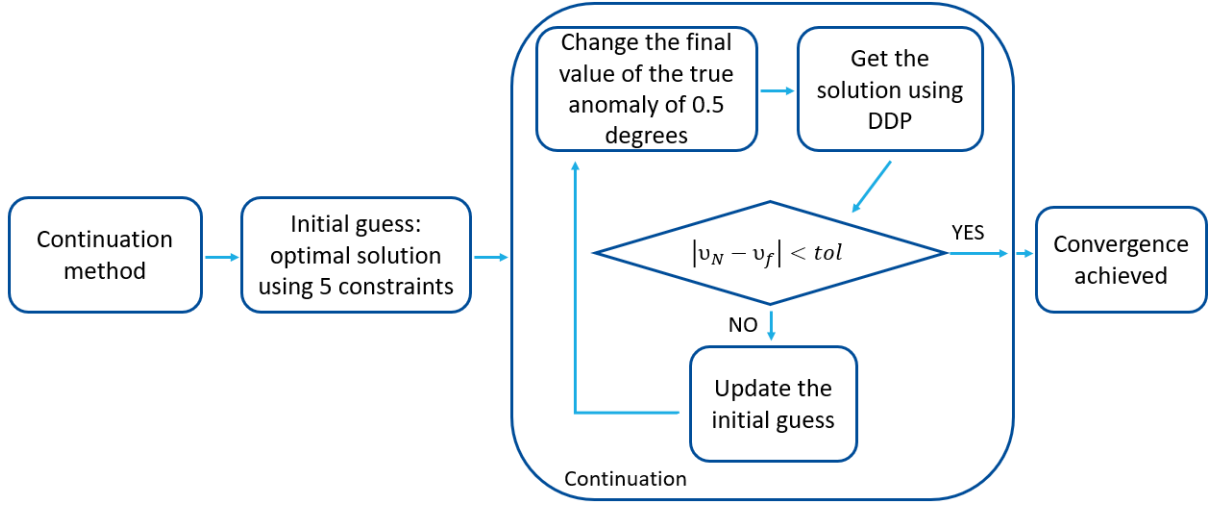


Figure 4.2: Continuation scheme for the true anomaly convergence.

4.4 Results

In this section the DDP algorithm is applied for solving two different orbit scenarios and demonstrate the effectiveness of the new proposed methodology:

- Mars interplanetary transfer
- Near-Earth asteroid transfer

This use case is not representative of real non-linear optimisation problems, where departure and arrival dates and time of flight are often subject to optimisation and not given. The following examples are simplified cases to present how the continuation method works. More interesting problems will be dealt in the following chapters. For each scenario, the same type of reference variables and set of tolerances is used. The numerical values are reported in Table 4.1.

Table 4.1: Satellite data and DDP tolerances.

Inputs	Numerical value
Specific Impulse	1500 s
Initial Mass	150 kg
Initial Thrust (tangential)	30 mN
Number of discrete time steps, N	50
tol_{ϑ}	1×10^{-6}
tol_{con}	0.1

Both the two examples solve the minimum thrust optimal control considering the square of the control magnitude as cost function to be minimised.

$$J = \frac{1}{2} w_u \sum_{z=1}^N \mathbf{u}_z^T \mathbf{u}_z s, \quad \mathbf{G} = 10 \cdot \mathbf{I}_{n_u} \quad (4.5)$$

with w_u representing a weighting coefficient which defines the relative importance during the minimisation process of the control magnitude with respect to the constraints violation, and s the constant time step used for the numerical integration that is introduced when moving from the continuous formulation of the cost function to the discrete one. The s is obtained dividing evenly the time of flight by the number of discrete time steps, N . The value function used for the problem is defined adjoining the constraints to the cost function using a set of Lagrange multipliers:

$$J_{aug} = J^* + \mathbf{b}^T (\mathbf{x} - \mathbf{x}_f)^T (\mathbf{x} - \mathbf{x}_f) \quad (4.6)$$

All the simulations are carried out using a processor Intel® Core (TM) i7-8700 CPU @ 3.20 GHz. A Runge-Kutta explicit 8th order integration scheme is used for the numerical integration of the system dynamics and its analytical partial derivatives are used for the solution of the backward equations and reported in Chapter 2.

4.4.1 Earth to Mars interplanetary transfer

The first reference scenario considers an interplanetary transfer to Mars starting from the Earth. The initial and final conditions for the problem are reported in Table 4.2. The state vectors in terms of orbital elements are computed from analytic ephemerides which approximate JPL ephemerides DE405. The adimensionalisation of the problem is carried out considering the first philosophy introduced in Section 2 and the reference variables adopted are reported in Table 4.3.

Table 4.2: Input for Earth to Mars interplanetary transfer.

Inputs	Numerical value
Departure date	Midnight 20/5/2003
Time of Flight	200 days
Arrival date	Midnight 6/12/2003
Relative velocity w.r.to Earth	1.66 km/s
μ	$1.3271244 \cdot 10^{11} \text{ km}^3/\text{s}^2$
u_{nom}	30 mN
tol_a	100 km
tol_e	$1 \cdot 10^{-4}$
tol_i	0.01°
tol_Ω	0.01°
tol_ω	0.01°
tol_ν	0.1°

Table 4.3: Reference variables for Earth to Mars transfer.

Reference variable	Numerical value
L_{ref}	$6.133209 \cdot 10^8 \text{ km}$
t_{ref}	$4.16942177 \cdot 10^7 \text{ s}$
m_{ref}	3 kg
u_{ref}	30 mN

The reference mass is not set to the initial mass to avoid the adimensional mass being a number in the interval $[0,1]$. Indeed, the mass is always present at the denominator of each control thrust component so that it is better to have the denominator far from 0 to avoid numerical issues. In particular, for this specific use case convergence is achieved for values of reference mass $m_{ref} < 3.75\text{ kg}$ keeping fixed all the other parameters. The effect of the wrong choice of reference mass affect the computation of the cost functional partial derivatives. The comparison between the transition function partial derivatives considering values of the reference mass equal to 150 kg and 3 kg is presented in Figures 4.3-4.12, respectively.

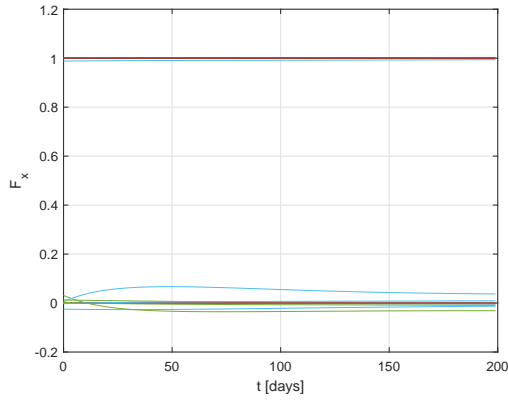


Figure 4.3: Jacobian matrix of the transition function with respect to the state for $m_{ref} = 150\text{ kg}$.

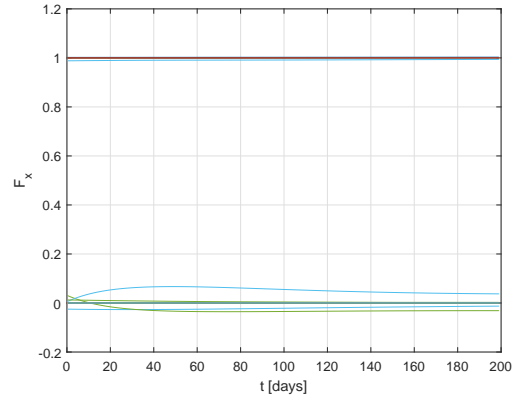


Figure 4.4: Jacobian matrix of the transition function with respect to the state for $m_{ref} = 3\text{ kg}$.

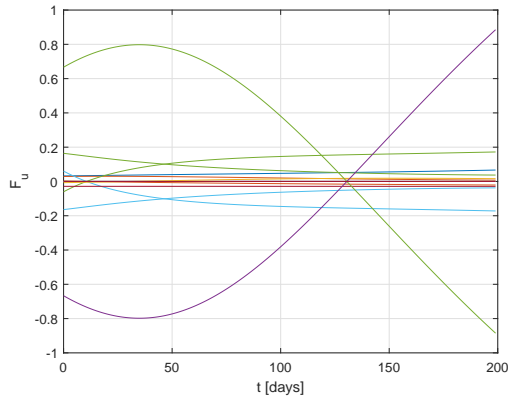


Figure 4.5: Jacobian matrix of the transition function with respect to the control for $m_{ref} = 150\text{ kg}$.

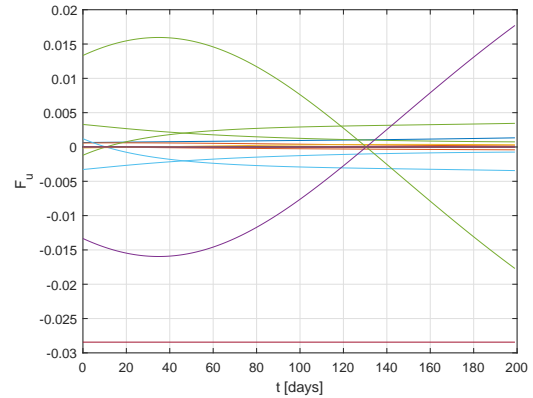


Figure 4.6: Jacobian matrix of the transition function with respect to the control for $m_{ref} = 3\text{ kg}$.

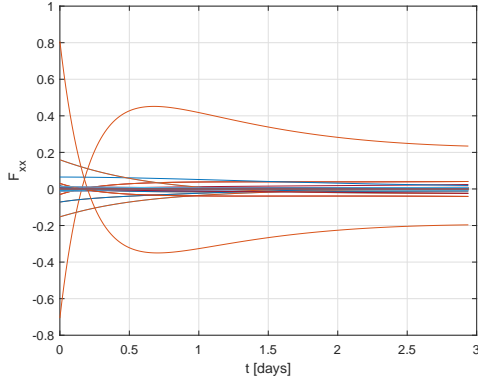


Figure 4.7: Hessian matrix of the transition function with respect to the state for $m_{ref} = 150 \text{ kg}$.

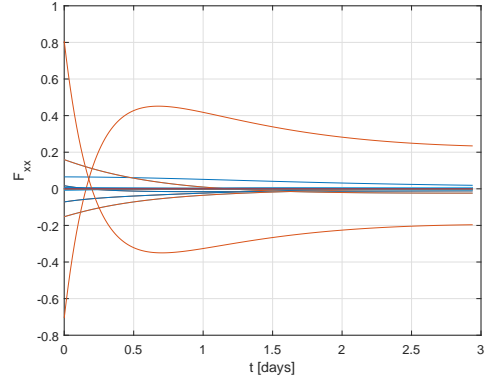


Figure 4.8: Hessian matrix of the transition function with respect to the state for $m_{ref} = 3 \text{ kg}$.

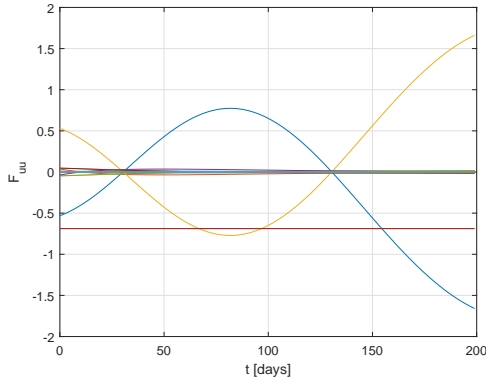


Figure 4.9: Hessian matrix of the transition function with respect to the control for $m_{ref} = 150 \text{ kg}$.

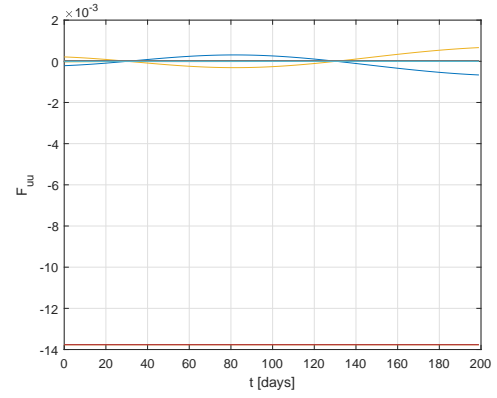


Figure 4.10: Hessian matrix of the transition function with respect to the control for $m_{ref} = 3 \text{ kg}$.

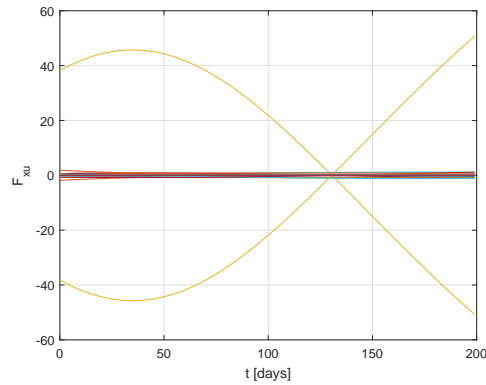


Figure 4.11: Hessian matrix of the transition function with respect to the state and control for $m_{ref} = 150 \text{ kg}$.

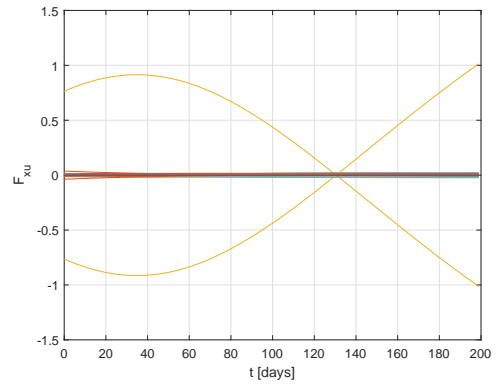


Figure 4.12: Hessian matrix of the transition function with respect to the state and control for $m_{ref} = 3 \text{ kg}$.

The trend of all the partial derivatives of the transition functions is the similar. However, the absolute values are different due to the difference of the reference mass that is at the denominator of the system dynamics. The discrepancy between the magnitude of the transition functions partial derivatives affects the cost functional partial derivative computed during the backward processing using Eqs. (3.44). The comparison between the cost functional partial derivatives at the end of the first state of the DDP algorithm using values of the reference mass equal to 150 kg and 3 kg is presented in Figures 4.13-4.16, respectively.

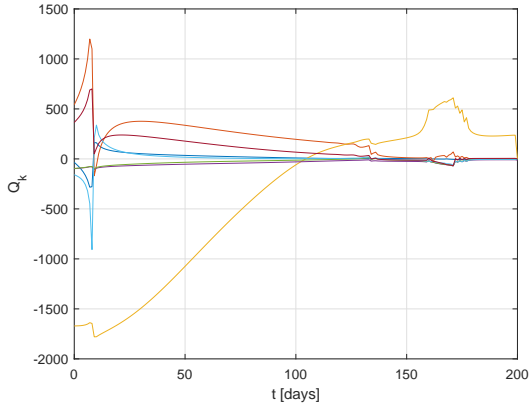


Figure 4.13: Jacobian matrix of the cost functional with respect to the state for $m_{ref} = 150 \text{ kg}$.

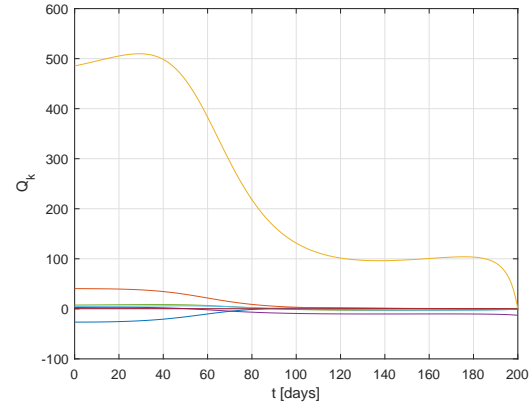


Figure 4.14: Jacobian matrix of the cost functional with respect to the state for $m_{ref} = 3 \text{ kg}$.

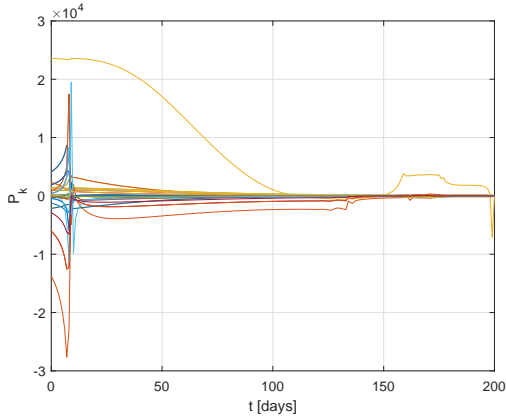


Figure 4.15: Hessian matrix of the cost functional with respect to the state for $m_{ref} = 150 \text{ kg}$.

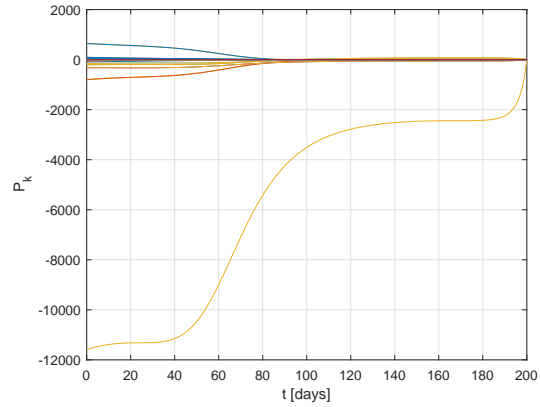


Figure 4.16: Hessian matrix of the cost functional with respect to the state for $m_{ref} = 3 \text{ kg}$.

It is clear that the cost functional partial derivatives computed using the reference mass equal to 150 kg are discontinuous. Therefore, the DDP algorithm is not able to converge because one of the basic assumptions of the algorithm is that all the cost functional partial derivatives must be smooth continuous functions. The initial guess for the control is a simple tangential thrust that is provided at each time step. The DDP algorithm is not constraining the initial guess used for the nominal control. It is obvious that the closer the nominal control to the optimal solution,

the lower is the computational time employed by the DDP algorithm to provide the optimal solution. The constraints are defined according to what presented in Section II as the quadratic difference between the actual state and the prescribed final state vector. The initial condition for the Lagrange multipliers is set to be proportion to the constraint violation.

$$\bar{\mathbf{b}} = 10 \cdot \varphi(\mathbf{x}_{N+1}) \quad (4.7)$$

The solution to the first part of the DDP algorithm is reported in Figure 4.17 showing both the magnitude of the initial tangential control law (blue) and the optimal control time history at the end of the first layer using 5 endpoint constraints (green), respectively. This represents

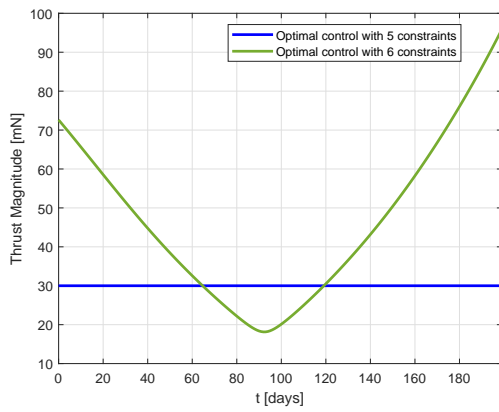


Figure 4.17: Nominal vs optimal control magnitude time history for Earth to Mars transfer: part 1.

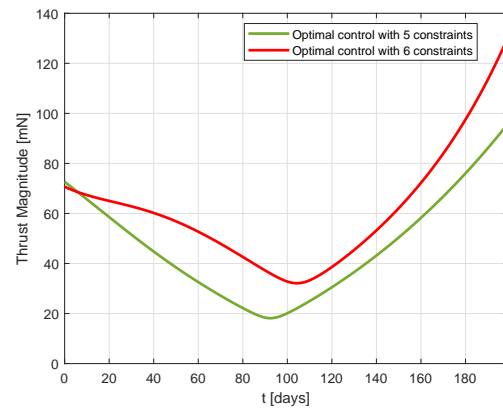


Figure 4.18: First part vs final optimal control magnitude time history for Earth to Mars transfer: part 2.

the solution to the optimal control problem considering five endpoint constraints. The optimal control obtained in this first part is used as initial guess to solve the fully constrained optimal control problem following the continuation scheme proposed in the Section III. In Figure 4.18 it is reported the difference between the optimal control law obtained solving the 5 endpoint constraints and the final optimal thrust obtained solving the fully constrained problem after the application of the continuation scheme. It is possible to appreciate that the DDP is just slightly adjusting the control law such that also the final true anomaly is matching the prescribed one without losing the final targeted orbit even if all 6 elements can now be changed. In this specific example the initial condition for the inclination has been set to a value slightly higher than zero to avoid the singularity embedded in the Gauss' dynamics in terms of classic Keplerian elements. However, the important aspect to be underlined is the possibility to couple orbital elements with the DDP algorithm. The final optimal trajectory given by the DDP algorithm for each orbital parameter is presented in Figures 4.19-4.24. The final interplanetary trajectory in the xy plane is reported in Figure 4.25. It is possible also to compare the two different time histories for the nominal and optimal control that are reported in Figures 4.26-4.27.

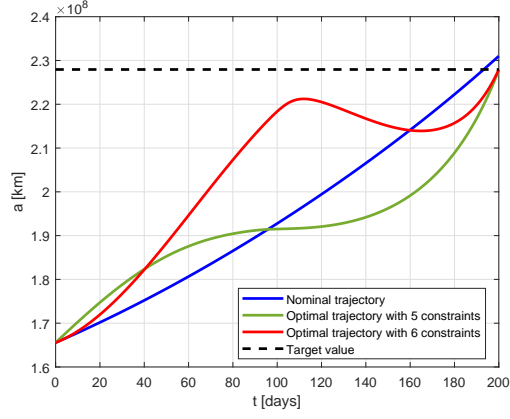


Figure 4.19: Nominal vs optimal semi-major axis trajectory for Earth to Mars transfer.

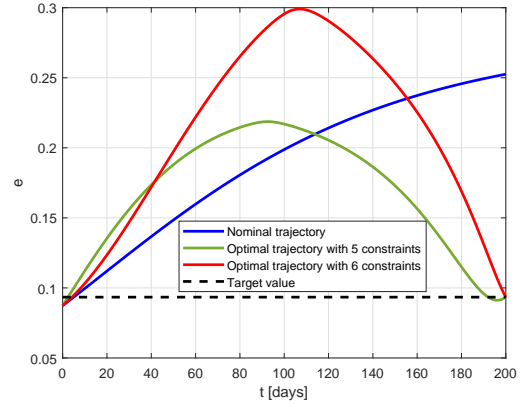


Figure 4.20: Nominal vs optimal eccentricity trajectory for Earth to Mars transfer.

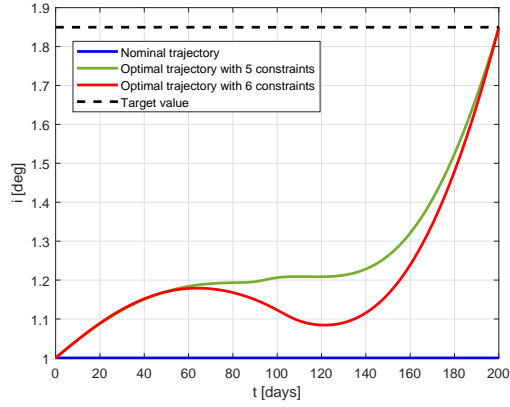


Figure 4.21: Nominal vs optimal inclination trajectory for Earth to Mars transfer.

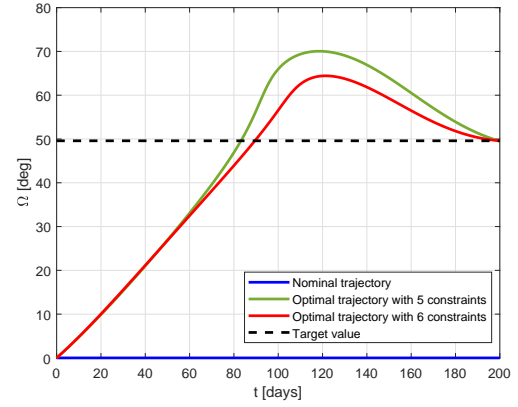


Figure 4.22: Nominal vs optimal RAAN trajectory for Earth to Mars transfer.

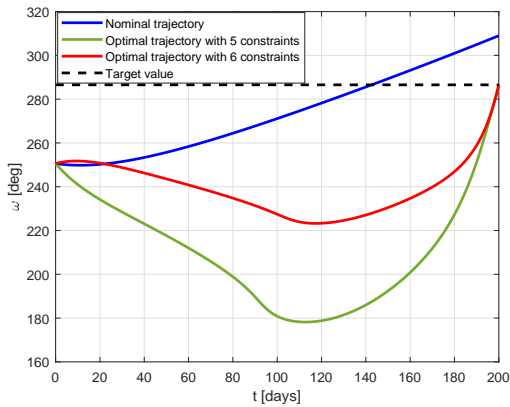


Figure 4.23: Nominal vs optimal ω trajectory for Earth to Mars transfer.

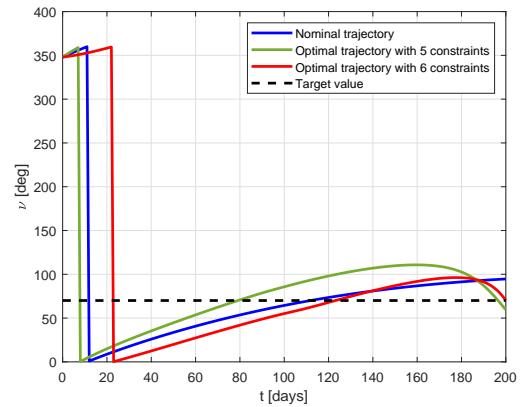


Figure 4.24: Nominal vs optimal ν trajectory for Earth to Mars transfer.

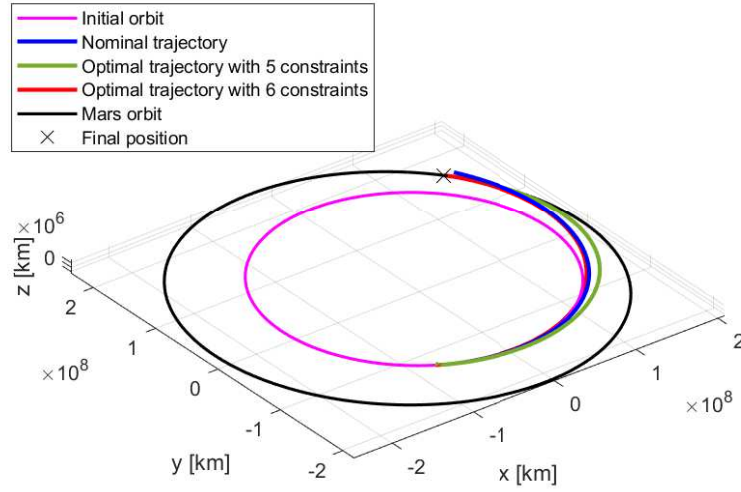


Figure 4.25: Final interplanetary Earth to Mars interplanetary trajectory.

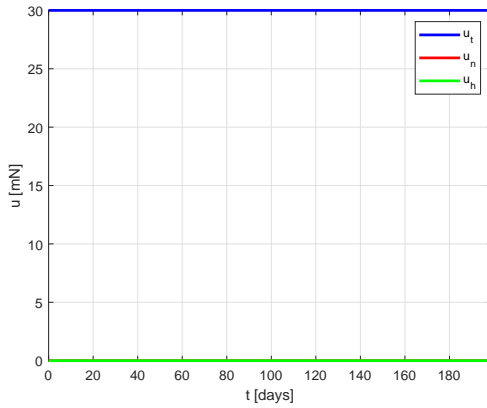


Figure 4.26: Nominal control time law for Earth to Mars transfer.

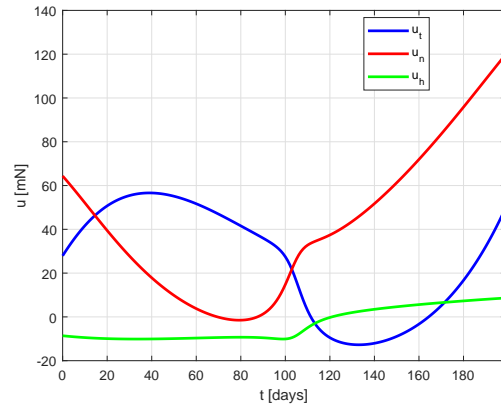


Figure 4.27: Optimal control time law for Earth to Mars transfer.

4.4.2 Earth to near-Earth asteroid transfer

The second example consists in an interplanetary transfer towards the near-Earth asteroid Apophis. The initial conditions and final conditions for the problem are reported in Table 4.4. Also in this case the state vectors in terms of orbital elements are computed from analytic ephemerides which approximate JPL ephemerides DE405. The adimensionalisation of the problem is carried out considering the first philosophy introduced in Section II and the reference variables adopted are reported in Table 4.5.

The orbit dynamics used to solve this exercise is the set of Gauss' equations in terms of modified equinoctial elements. This is necessary to avoid to change the initial condition related to Earth orbit like for the previous example for which there is the singularity if classic Keplerian

Table 4.4: Input for Earth to Apophis transfer.

Inputs	Numerical value
Departure date	Midnight 20/5/2022
Arrival date	Midnight 1/11/2022
Time of Flight	165 days
Relative velocity w.r.to Earth	1.66 km/s
μ	$1.3271244 \cdot 10^{11} \text{ km}^3/\text{s}^2$
u_{nom}	30 mN
tol_a	100 km
tol_f	$1 \cdot 10^{-5}$
tol_g	$1 \cdot 10^{-5}$
tol_h	$1 \cdot 10^{-5}$
tol_k	$1 \cdot 10^{-5}$
tol_Γ	0.1°

Table 4.5: Reference variables for Earth to Apophis transfer.

Reference variable	Numerical value
L_{ref}	$6.133209 \cdot 10^8 \text{ km}$
t_{ref}	$4.16942177 \cdot 10^7 \text{ s}$
m_{ref}	15 kg
u_{ref}	30 mN

elements are used. The aim of the example is to check how the DDP algorithm can fit different dynamics considering different set of orbital elements. Also, in this case the initial guess for the control is set equal to a tangential control law to show that this simple nominal control can be used for different problems. The initial condition for the Lagrange multipliers is set always proportional to the constraints violation.

$$\bar{\mathbf{b}} = 10^4 \cdot \boldsymbol{\varphi}(\mathbf{x}_{N+1}) \quad (4.8)$$

The solution to the first part of the problem provided by DDP algorithm is reported in Figure 4.28 showing both the magnitude of the initial tangential control law and the optimal control time history, respectively. Also in this case, the expectation is that starting from an optimal policy targeting the final orbit, the DDP algorithm is able to adjust the control thrust only to match the true longitude. The results of the final optimal control law is reported in Figure 4.29. The analysis of the final optimal control law shows that this time in the initial and final parts of the transfer the satellite has to thrust more to adjust the final true longitude due to the initial orbital configuration of Earth and Apophis at the initial time. However, the DDP algorithm succeeds to provide an optimal solution that is not possible to get solving directly the problem with all the six endpoint constraints. The comparison between the nominal and optimal trajectories of the modified equinoctial elements are reported in Figures 4.30-4.35. Looking at the time evolution of the parameters h and k it is possible to check this time that the initial orbital inclination is effectively equal to zero and the singularity of Gauss' equations occurring when using classic Keplerian elements is avoided.

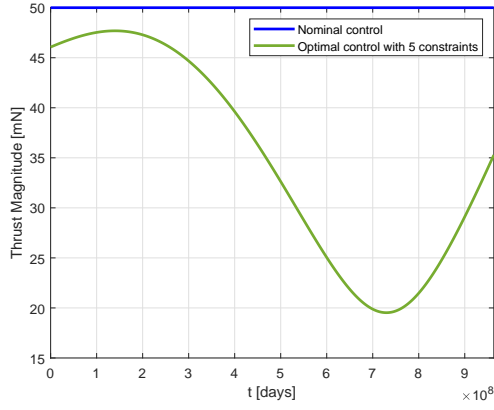


Figure 4.28: Nominal vs optimal control magnitude time history for Earth to Apophis transfer: part 1.

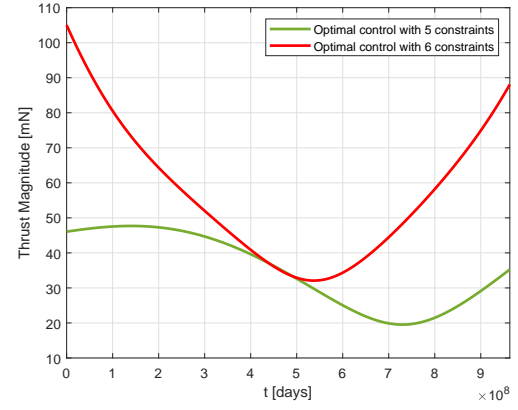


Figure 4.29: First layer vs final optimal control magnitude time history for Earth to Apophis transfer: part 2.

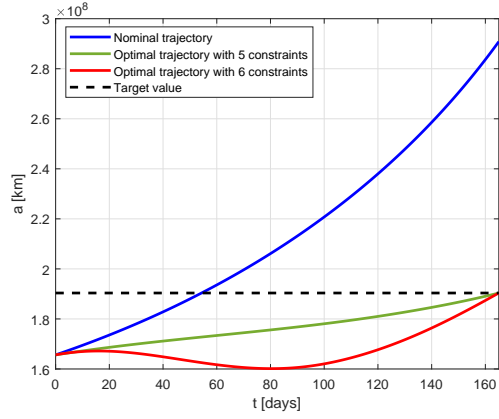


Figure 4.30: Nominal vs optimal semi-major axis trajectory for Earth to Apophis transfer.

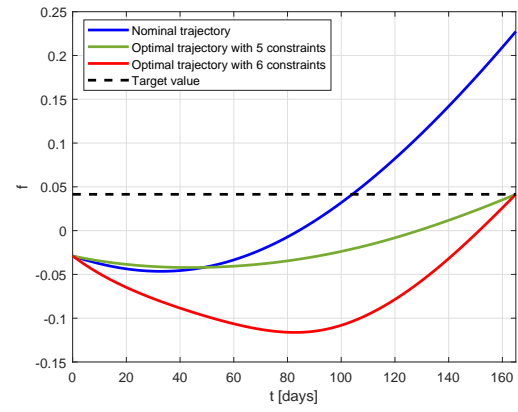


Figure 4.31: Nominal vs optimal f trajectory for Earth to Apophis transfer.

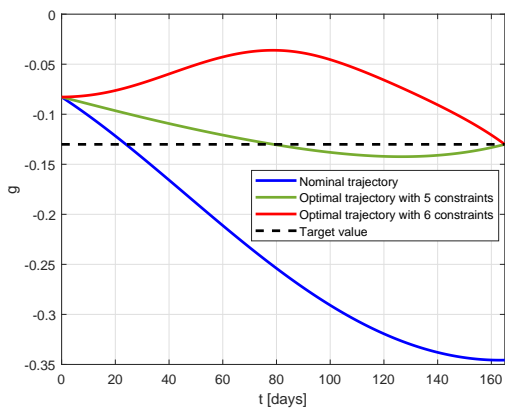


Figure 4.32: Nominal vs optimal g trajectory for Earth to Apophis transfer.

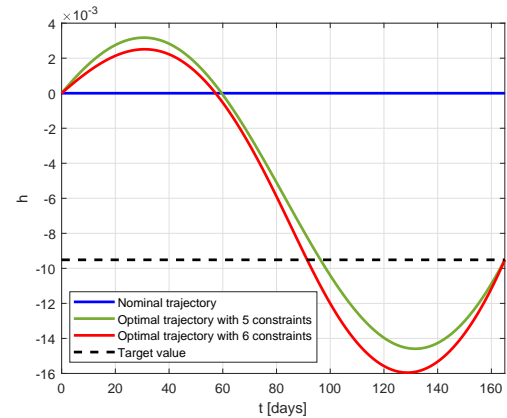


Figure 4.33: Nominal vs optimal h trajectory for Earth to Apophis transfer.

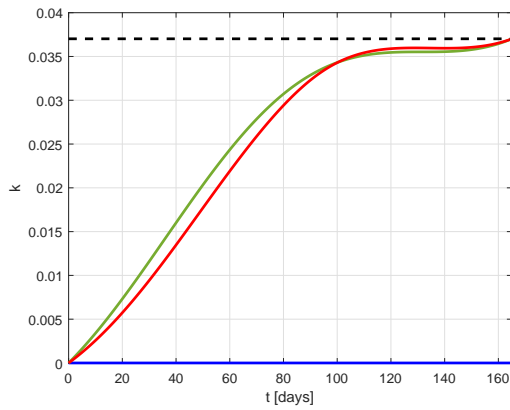


Figure 4.34: Nominal vs optimal k trajectory for Earth to Apophis transfer.

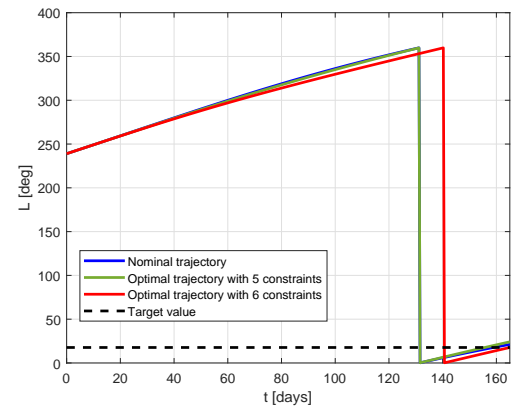


Figure 4.35: Nominal vs optimal true longitude trajectory for Earth to Apophis transfer.

The final 3D orbital trajectory related to Apophis transfer is reported in Figure 4.36. Finally, the three components of the nominal and optimal control are presented respectively in Figures 4.37-4.38.

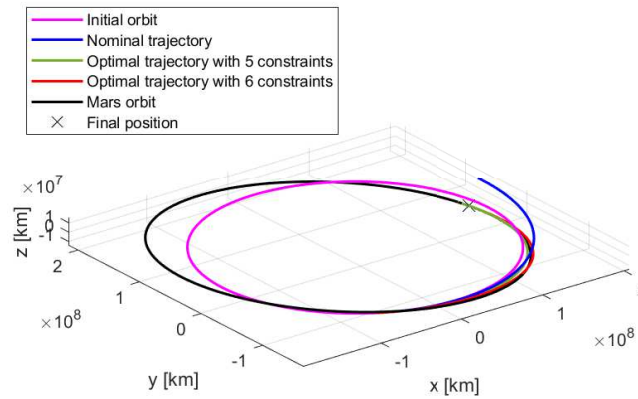


Figure 4.36: Optimal vs nominal 3D trajectory for Earth to Apophis transfer.

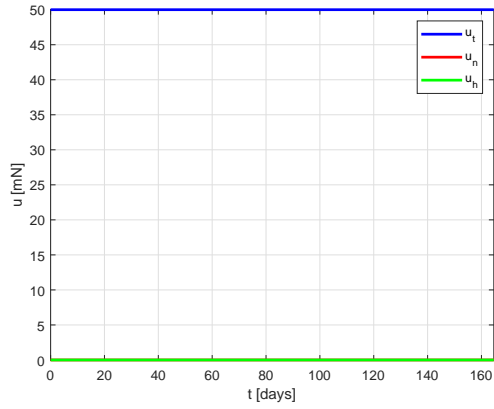


Figure 4.37: Nominal control law for Earth to Apophis transfer.

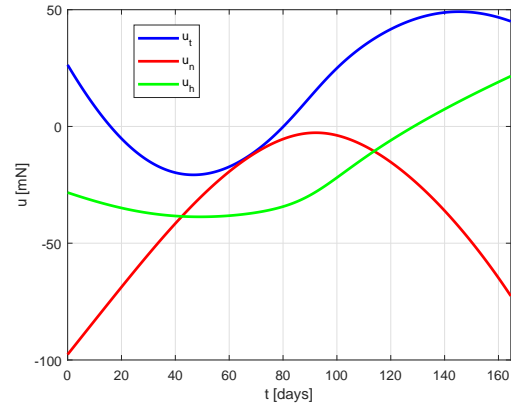


Figure 4.38: Optimal control law for Earth to Apophis transfer.

CHAPTER 5

Static/dynamic variables optimisation and multiple revolutions

This chapter presents another innovative contribution of this work: a modification of the DDP to account for static/dynamic variables optimisation. In the latter part of the chapter a systematic procedure to solve multiple revolutions optimal trajectory problem is proposed. This work was presented at the 73rd IAC [77] and will be the main subject of a future journal publication. This modification allows to solve multi-revolution orbit transfers with several spirals which are very difficult to be solved with classic direct and indirect optimisation methods.

5.1 Free final time and static parameters

One of the main advantages in using the DDP algorithm as optimisation technique is the adaptability to include many variables inside the optimisation process provided that these variables appear in the system dynamics and/or the cost functional so that partial derivatives can be evaluated with respect to such variables. The procedure for the derivation of the backward equations is always the same. Indeed, it is based on the linear-quadratic expansion of HJB equation. The difference is in the amount of terms appearing in each Taylor expansion that is related to the independent variables used. Two interesting variables which are worth to include in the optimisation process are the static parameters and the final time.

A static parameter is a generic variable which is not affecting the system dynamics during its propagation, but it is changing only the initial or final condition. An example of static variable is the departure time of a spacecraft, which is changing the initial condition for the propagation of the dynamics. The final time is an interesting variable to be optimised in all the cases where there is not a requirement on the transfer time for the spacecraft. Even if there is a constraint on the maximum transfer time it is useful to ask what is the optimal final time within this window. The discrete HJB equation is rewritten including also the dependencies on the static parameters and final time.

$$J_i^*(\mathbf{x}_i, \mathbf{b}, \mathbf{w}, t_f; t_i) = \min_{\mathbf{u}_i} [L_i(\mathbf{x}_i, \mathbf{u}_i, \mathbf{w}, t_f; t_i) + J_{i+1}^*(\mathbf{x}_{i+1}, \mathbf{b}, \mathbf{w}, t_f; t_{i+1})] \quad (5.1)$$

Similarly to what has been discussed in Chapter 3 the variations for the free final time, t_f , and

the generic static variables, \mathbf{w} , are introduced:

$$\delta t_f = t_f - \bar{t}_f, \quad \delta \mathbf{w} = \mathbf{w} - \bar{\mathbf{w}} \quad (5.2)$$

At this point, the Taylor expansion of each term up to the second order inside the HJB Eq. (5.1) is derived as follows:

$$\begin{aligned} J_i(\bar{\mathbf{x}}_i + \delta \mathbf{x}_i, \bar{\mathbf{b}} + \delta \mathbf{b}, \bar{\mathbf{w}} + \delta \mathbf{w}, \bar{t}_f + \delta t_f) &= J_i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) + \mathbf{J}_x^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{x}_i \\ &+ \mathbf{J}_b^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{b} + \mathbf{J}_w^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{w} + J_{t_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f \\ &+ \frac{1}{2} \delta \mathbf{x}_i^T \mathbf{J}_{xx}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{x}_i + \frac{1}{2} \delta \mathbf{b}^T \mathbf{J}_{bb}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{b} + \frac{1}{2} \delta \mathbf{w}^T \mathbf{J}_{ww}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{w} \\ &+ \frac{1}{2} \delta t_f J_{t_f t_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f + \delta \mathbf{x}_i^T \mathbf{J}_{xb}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{b} + \delta \mathbf{x}_i^T \mathbf{J}_{xw}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{w} \\ &+ \delta \mathbf{x}_i^T \mathbf{J}_{xt_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f + \delta \mathbf{b}^T \mathbf{J}_{bw}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{w} + \delta \mathbf{b}^T \mathbf{J}_{bt_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f \\ &+ \delta \mathbf{w}^T \mathbf{J}_{wt_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f \end{aligned} \quad (5.3)$$

$$\begin{aligned} J_{i+1}(\bar{\mathbf{x}}_{i+1} + \delta \mathbf{x}_{i+1}, \bar{\mathbf{b}} + \delta \mathbf{b}, \bar{\mathbf{w}} + \delta \mathbf{w}, \bar{t}_f + \delta t_f) &= J_{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \\ &+ \mathbf{J}_x^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{x}_{i+1} + \mathbf{J}_b^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{b} + \mathbf{J}_w^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{w} \\ &+ J_{t_f}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f + \frac{1}{2} \delta \mathbf{x}_{i+1}^T \mathbf{J}_{xx}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{x}_{i+1} \\ &+ \frac{1}{2} \delta \mathbf{b}^T \mathbf{J}_{bb}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{b} + \frac{1}{2} \delta \mathbf{w}^T \mathbf{J}_{ww}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{w} \\ &+ \frac{1}{2} \delta t_f J_{t_f t_f}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f + \delta \mathbf{x}_{i+1}^T \mathbf{J}_{xb}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{b} \\ &+ \delta \mathbf{x}_{i+1}^T \mathbf{J}_{xw}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{w} + \delta \mathbf{x}_{i+1}^T \mathbf{J}_{xt_f}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f \\ &+ \delta \mathbf{b}^T \mathbf{J}_{bw}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{w} + \delta \mathbf{b}^T \mathbf{J}_{bt_f}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f \\ &+ \delta \mathbf{w}^T \mathbf{J}_{wt_f}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f \end{aligned} \quad (5.4)$$

$$\begin{aligned} L_i(\bar{\mathbf{x}}_i + \delta \mathbf{x}_i, \bar{\mathbf{u}}_i + \delta \mathbf{u}_i, \bar{\mathbf{w}} + \delta \mathbf{w}, \bar{t}_f + \delta t_f) &= L_i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) + \mathbf{L}_x^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{x}_i \\ &+ \mathbf{L}_u^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{u}_i + \mathbf{L}_w^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{w} + L_{t_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f \\ &+ \frac{1}{2} \delta \mathbf{x}_i^T \mathbf{L}_{xx}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{x}_i + \frac{1}{2} \delta \mathbf{u}_i^T \mathbf{L}_{uu}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{u}_i \\ &+ \frac{1}{2} \delta \mathbf{w}^T \mathbf{L}_{ww}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{w} + \frac{1}{2} \delta t_f L_{t_f t_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f \\ &+ \delta \mathbf{x}_i^T \mathbf{L}_{xu}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{u}_i + \delta \mathbf{x}_i^T \mathbf{L}_{xw}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{w} + \delta \mathbf{x}_i^T \mathbf{L}_{xt_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f \\ &+ \delta \mathbf{u}_i^T \mathbf{L}_{uw}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{w} + \delta \mathbf{u}_i^T \mathbf{L}_{ut_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f + \delta \mathbf{w}^T \mathbf{L}_{wt_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f \end{aligned} \quad (5.5)$$

Also the Taylor expansion of the state variation at step $i + 1$ in terms of state variation at step i is modified accordingly to for account the dependencies on the static parameters and final time.

$$\begin{aligned}
 \delta \mathbf{x}_{i+1} = & \mathbf{F}^i(\bar{\mathbf{x}}_i + \delta \mathbf{x}_i, \bar{\mathbf{u}}_i + \delta \mathbf{u}_i, \bar{\mathbf{w}} + \delta \mathbf{w}, \bar{t}_f + \delta t_f) - \mathbf{F}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) = \\
 & \mathbf{F}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) - \mathbf{F}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) + \mathbf{F}_x^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{x}_i + \mathbf{F}_u^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{u}_i \\
 & + \mathbf{F}_w^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{w} + \mathbf{F}_{t_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f + \frac{1}{2} \delta \mathbf{x}_i^T \mathbf{F}_{xx}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{x}_i \\
 & + \frac{1}{2} \delta \mathbf{u}_i^T \mathbf{F}_{uu}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{u}_i + \frac{1}{2} \delta \mathbf{w}^T \mathbf{F}_{ww}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{w} \\
 & + \frac{1}{2} \delta t_f \mathbf{F}_{t_f t_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f + \delta \mathbf{x}_i^T \mathbf{F}_{xu}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{u}_i + \delta \mathbf{x}_i^T \mathbf{F}_{xw}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{w} \\
 & + \delta \mathbf{x}_i^T \mathbf{F}_{xt_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f + \delta \mathbf{u}_i^T \mathbf{F}_{uw}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{w} + \delta \mathbf{u}_i^T \mathbf{F}_{ut_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f \\
 & + \delta \mathbf{w}^T \mathbf{F}_{wt_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f
 \end{aligned} \tag{5.6}$$

It is possible to get the new expanded HJB equation replacing the Taylor expansion given by Eqs. (5.3-5.6) in Eq. (5.1).

$$\begin{aligned}
 & J_i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) + \mathbf{J}_x^T(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{x}_i + \mathbf{J}_b^T(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{b} + \mathbf{J}_w^T(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{w} \\
 & + J_{t_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f + \frac{1}{2} \delta \mathbf{x}_i^T \mathbf{J}_{xx}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{x}_i + \frac{1}{2} \delta \mathbf{b}^T \mathbf{J}_{bb}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{b} \\
 & + \frac{1}{2} \delta \mathbf{w}^T \mathbf{J}_{ww}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{w} + \frac{1}{2} \delta t_f J_{t_f t_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f + \delta \mathbf{x}_i^T \mathbf{J}_{xb}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{b} \\
 & + \delta \mathbf{x}_i^T \mathbf{J}_{xw}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{w} + \delta \mathbf{x}_i^T \mathbf{J}_{xt_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f + \delta \mathbf{b}^T \mathbf{J}_{bw}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{w} \\
 & + \delta \mathbf{b}^T \mathbf{J}_{bt_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f + \delta \mathbf{w}^T \mathbf{J}_{wt_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f = \min_{\delta \mathbf{u}_i} [L_i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \\
 & + \mathbf{E}_i \delta \mathbf{x}_i + J_{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) + \mathbf{D}_i \delta \mathbf{u}_i + \mathbf{J}_b^{i+1T}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{b} + \tilde{M}_i \delta t_f + \tilde{\mathbf{G}}_i \delta \mathbf{w} \\
 & + \delta \mathbf{x}_i^T \mathbf{A}_i \delta \mathbf{x}_i + \delta \mathbf{w}^T \tilde{\mathbf{W}}_i \delta \mathbf{w} + \delta t_f \tilde{Y}_i \delta t_f + \delta \mathbf{u}_i^T \mathbf{C}_i \delta \mathbf{u}_i + \frac{1}{2} \delta \mathbf{b}^T \mathbf{J}_{bb}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta \mathbf{b} \\
 & + \delta \mathbf{x}_i^T \mathbf{B}_i \delta \mathbf{u}_i + \delta \mathbf{x}_i^T \tilde{\mathbf{N}}_i \delta t_f + \delta \mathbf{x}_i^T \tilde{\mathbf{O}}_i \delta \mathbf{w} + \delta \mathbf{u}_i^T \mathbf{K}_i \delta \mathbf{b} + \delta \mathbf{u}_i^T \Pi_i \delta t_f + \delta \mathbf{u}_i^T \Phi_i \delta \mathbf{w} \\
 & + \delta \mathbf{w}^T \tilde{\mathbf{V}}_i \delta t_f + \delta \mathbf{x}_i^T \mathbf{X}_i \delta \mathbf{b} + \delta \mathbf{b}_i^T \tilde{\mathbf{T}}_i \delta \mathbf{w} + \delta \mathbf{b}_i^T \tilde{\mathbf{U}}_i \delta t_f + \delta \mathbf{w}_i^T \mathbf{J}_{wt_f}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \delta t_f]
 \end{aligned} \tag{5.7}$$

As for the algorithm derived in Chapter 3, some new additional matrices have been introduced next to the ones defined in Eq. (3.34) to simplify the notation of the expanded HJB equation. These matrices are obtained considering the addenda which are multiplying the same differential variation on the right-hand side of Eq. (5.7). The new additional matrices are defined as follows:

$$\begin{aligned}
 \tilde{\mathbf{G}}_i = & \left[\mathbf{L}_w^T(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) + \mathbf{J}_x^{i+1T}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{F}_w^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) + \mathbf{J}_w^{i+1T}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \right] \\
 \tilde{M}_i = & \left[L_{t_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) + \mathbf{J}_x^{i+1T}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{F}_{t_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) + J_{t_f}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \right] \\
 \tilde{\mathbf{U}}_i = & \left[\mathbf{J}_{xb}^{i+1T}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{F}_t^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) + \mathbf{J}_{bt_f}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \right] \\
 \tilde{\mathbf{T}}_i = & \left[\mathbf{J}_{xb}^{i+1T}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{F}_w^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) + \mathbf{J}_{bw}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \right] \\
 \tilde{\mathbf{N}}_i = & \left[\mathbf{L}_{xt_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) + \mathbf{J}_x^{i+1T}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{F}_{xt_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \right. \\
 & + \mathbf{F}_x^T(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{J}_{xx}^{i+1T}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{F}_{t_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \\
 & \left. + \mathbf{F}_x^T(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{J}_{xt_f}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \right]
 \end{aligned} \tag{5.8}$$

$$\begin{aligned}
 \tilde{\mathbf{O}}_i &= \left[\mathbf{L}_{xw}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) + \mathbf{J}_x^{i+1\top}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{F}_{xw}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \right. \\
 &\quad + \mathbf{F}_x^{i\top}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{J}_{xx}^{i+1\top}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{F}_w^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \\
 &\quad \left. + \mathbf{F}_x^{i\top}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{J}_{xw}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \right] \\
 \tilde{\mathbf{\Pi}}_i &= \left[\mathbf{L}_{ut_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) + \mathbf{J}_x^{i+1\top}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{F}_{ut_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \right. \\
 &\quad + \mathbf{F}_u^{i\top}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{J}_{xx}^{i+1\top}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{F}_{t_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \\
 &\quad \left. + \mathbf{F}_u^{i\top}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{J}_{xt_f}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \right] \\
 \tilde{\mathbf{\Phi}}_i &= \left[\mathbf{L}_{uw}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) + \mathbf{J}_x^{i+1\top}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{F}_{uw}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \right. \\
 &\quad + \mathbf{F}_u^{i\top}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{J}_{xx}^{i+1\top}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{F}_w^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \\
 &\quad \left. + \mathbf{F}_u^{i\top}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{J}_{xw}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \right] \\
 \tilde{\mathbf{V}}_i &= \left[\mathbf{L}_{wt_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) + \mathbf{J}_x^{i+1\top}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{F}_{wt_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \right. \\
 &\quad + \mathbf{F}_w^{i\top}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{J}_{xx}^{i+1\top}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{F}_{t_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) + \mathbf{J}_{wt_f}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \\
 &\quad \left. + \mathbf{F}_w^{i\top}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{J}_{xt_f}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) + \mathbf{J}_{xw}^{i+1\top}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{F}_{t_f}^{i\top}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \right] \\
 \tilde{\mathbf{W}}_i &= \frac{1}{2} \left[\mathbf{L}_{ww}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) + \mathbf{J}_x^{i+1\top}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{F}_{ww}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \right. \\
 &\quad + \mathbf{F}_w^{i\top}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{J}_{xx}^{i+1\top}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{F}_w^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) + \mathbf{J}_{ww}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \\
 &\quad \left. + \mathbf{F}_w^{i\top}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{J}_{xw}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) + \mathbf{J}_{xw}^{i+1\top}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{F}_w^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \right] \\
 \tilde{Y}_i &= \frac{1}{2} \left[\mathbf{L}_{t_ft_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) + \mathbf{J}_x^{i+1\top}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{F}_{t_ft_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \right. \\
 &\quad + \mathbf{F}_{t_f}^{i\top}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{J}_{xx}^{i+1\top}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{F}_{t_f}^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) + \mathbf{J}_{t_ft_f}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \\
 &\quad \left. + \mathbf{F}_{t_f}^{i\top}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{J}_{xt_f}^{i+1}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) + \mathbf{J}_{xt_f}^{i+1\top}(\bar{\mathbf{x}}_{i+1}, \bar{\mathbf{b}}, \bar{\mathbf{w}}, \bar{t}_f) \mathbf{F}_{t_f}^{i\top}(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) \right]
 \end{aligned}$$

The optimal control variation, $\delta \mathbf{u}_i^*$, is obtained differentiating the right-hand of Eq. (5.7) side with respect to the control variation. The difference with respect to the expression derived in Chapter 3 is that now the feedback law contains addenda related to the final time and static parameters variations, δt_f and $\delta \mathbf{w}$.

$$\frac{\partial}{\partial (\delta \mathbf{u}_i)} = \mathbf{D}_i^\top + 2\mathbf{C}_i \delta \mathbf{u}_i^* + \mathbf{B}_i^\top \delta \mathbf{x}_i + \mathbf{K}_i \delta \mathbf{b} + \mathbf{\Pi}_i \delta t_f + \mathbf{\Phi}_i \delta \mathbf{w} = 0 \quad (5.9)$$

The expression of the optimal feedback control law is:

$$\delta \mathbf{u}^* = -\frac{1}{2} \mathbf{C}^{-1} [\mathbf{D}_i^\top + \mathbf{B}_i^\top \delta \mathbf{x}_i + \mathbf{K}_i \delta \mathbf{b} + \mathbf{\Phi}_i \delta \mathbf{w} + \mathbf{\Pi}_i \delta t_f] \quad (5.10)$$

The optimal control law can be rewritten introducing the coefficients which are multiplying the different variations.

$$\delta \mathbf{u}^* = \boldsymbol{\alpha}_i + \boldsymbol{\beta}_i \delta \mathbf{x}_i + \boldsymbol{\gamma}_i \delta \mathbf{b} + \boldsymbol{\eta}_i \delta \mathbf{w} + \boldsymbol{\rho}_i \delta t_f \quad (5.11)$$

$$\begin{aligned}\alpha_i &= -\frac{1}{2}\mathbf{C}_i^{-1}\mathbf{D}_i^T & \beta_i &= -\frac{1}{2}\mathbf{C}_i^{-1}\mathbf{B}_i^T & \gamma_i &= -\frac{1}{2}\mathbf{C}_i^{-1}\mathbf{K}_i \\ \eta_i &= -\frac{1}{2}\mathbf{C}_i^{-1}\Phi_i & \rho_i &= -\frac{1}{2}\mathbf{C}_i^{-1}\Pi_i\end{aligned}\quad (5.12)$$

where η and ρ represent the feedback matrices relating the optimal control variation to the static parameters and final time variation, respectively. The optimal control variation in Eq. (5.10) must be replaced inside the expanded HJB Eq. (5.7) for the derivation of the backward equations expressing the necessary conditions of optimality.

$$\begin{aligned}& \vartheta_i + \mathbf{Q}_i\delta\mathbf{x}_i + \mathbf{Z}_i\delta\mathbf{b} + \mathbf{G}_i\delta\mathbf{w} + M_i\delta t_f + \delta\mathbf{x}_i^T\mathbf{P}_i\delta\mathbf{x}_i + \delta\mathbf{b}^T\mathbf{R}_i\delta\mathbf{b} + \delta\mathbf{w}_i^T\mathbf{W}_i\delta\mathbf{w}_i + \delta t_f Y_i\delta t_f \\& + \delta\mathbf{x}_i^T\mathbf{S}_i\delta\mathbf{b} + \delta\mathbf{x}_i^T\mathbf{O}_i\delta\mathbf{w} + \delta\mathbf{x}_i^T\mathbf{N}_i\delta t_f + \delta\mathbf{b}^T\mathbf{T}_i\delta\mathbf{w} + \delta\mathbf{b}^T\mathbf{U}_i\delta t_f + \delta\mathbf{w}^T\mathbf{V}_i\delta t_f = \\& \vartheta_{i+1} - \frac{1}{4}\mathbf{D}_i\mathbf{C}_i^{-1}\mathbf{D}_i^T + \left(\mathbf{E}_i - \frac{1}{2}\mathbf{D}_i\mathbf{C}_i^{-1}\mathbf{B}_i^T\right)\delta\mathbf{x}_i + \left(\mathbf{Z}_{i+1} - \frac{1}{2}\mathbf{D}_i\mathbf{C}_i^{-1}\mathbf{K}_i\right)\delta\mathbf{b} \\& + \left(\tilde{\mathbf{G}}_i - \frac{1}{2}\mathbf{D}_i\mathbf{C}_i^{-1}\Phi_i\right)\delta\mathbf{w} + \left(\tilde{M}_i - \frac{1}{2}\mathbf{D}_i\mathbf{C}_i^{-1}\Pi_i\right)\delta t_f + \delta\mathbf{x}_i^T\left(\mathbf{A}_i - \frac{1}{4}\mathbf{B}_i\mathbf{C}_i^{-1}\mathbf{B}_i^T\right)\delta\mathbf{x}_i \\& + \delta\mathbf{b}^T\left(\mathbf{R}_{i+1} - \frac{1}{4}\mathbf{K}_i^T\mathbf{C}_i^{-1}\mathbf{K}_i\right)\delta\mathbf{b} + \delta\mathbf{w}^T\left(\tilde{\mathbf{W}}_i - \frac{1}{4}\Phi_i^T\mathbf{C}_i^{-1}\Phi_i\right)\delta\mathbf{w} + \delta t_f\left(\tilde{Y}_i - \frac{1}{4}\Pi_i^T\mathbf{C}_i^{-1}\Pi_i\right)\delta t_f \\& + \delta\mathbf{x}_i^T\left(\mathbf{X}_i - \frac{1}{2}\mathbf{B}_i\mathbf{C}_i^{-1}\mathbf{K}_i\right)\delta\mathbf{b} + \delta\mathbf{z}_i^T\left(\tilde{\mathbf{O}}_i - \frac{1}{2}\mathbf{B}_i\mathbf{C}_i^{-1}\Phi_i\right)\delta\mathbf{w} + \delta\mathbf{x}_i^T\left(\tilde{\mathbf{N}}_i - \frac{1}{2}\mathbf{B}_i\mathbf{C}_i^{-1}\Pi_i\right)\delta t_f \\& + \delta\mathbf{b}^T\left(\tilde{\mathbf{T}}_i - \frac{1}{2}\mathbf{K}_i^T\mathbf{C}_i^{-1}\Phi_i\right)\delta\mathbf{w} + \delta\mathbf{b}^T\left(\tilde{\mathbf{U}}_i - \frac{1}{2}\mathbf{K}_i^T\mathbf{C}_i^{-1}\Pi_i\right)\delta t_f + \delta\mathbf{w}^T\left(\tilde{\mathbf{V}}_i - \frac{1}{2}\Phi_i^T\mathbf{C}_i^{-1}\Pi_i\right)\delta t_f\end{aligned}\quad (5.13)$$

As for the simpler case derived in Chapter 3 the partials of the optimal cost functional have been modified introducing new auxiliary variables. The final backward equations are obtained equating the coefficients multiplying the same differential variations.

$$\begin{aligned}\vartheta_i &= \vartheta_{i+1} - \frac{1}{4}\mathbf{D}_i\mathbf{C}_i^{-1}\mathbf{D}_i^T & \mathbf{Q}_i &= \mathbf{E}_i - \frac{1}{2}\mathbf{D}_i\mathbf{C}_i^{-1}\mathbf{B}_i^T \\ \mathbf{Z}_i &= \mathbf{Z}_{i+1} - \frac{1}{2}\mathbf{D}_i\mathbf{C}_i^{-1}\mathbf{K}_i & \mathbf{G}_i &= \tilde{\mathbf{G}}_i - \frac{1}{2}\mathbf{D}_i\mathbf{C}_i^{-1}\Phi_i \\ M_i &= \tilde{M}_i - \frac{1}{2}\mathbf{D}_i\mathbf{C}_i^{-1}\Pi_i & \mathbf{P}_i &= \mathbf{A}_i - \frac{1}{4}\mathbf{B}_i\mathbf{C}_i^{-1}\mathbf{B}_i^T \\ \mathbf{R}_i &= \mathbf{R}_{i+1} - \frac{1}{4}\mathbf{K}_i^T\mathbf{C}_i^{-1}\mathbf{K}_i & \mathbf{W}_i &= \tilde{\mathbf{W}}_i - \frac{1}{4}\Phi_i^T\mathbf{C}_i^{-1}\Phi_i \\ Y_i &= \tilde{Y}_i - \frac{1}{4}\Pi_i^T\mathbf{C}_i^{-1}\Pi_i & \mathbf{S}_i &= \mathbf{X}_i - \frac{1}{2}\mathbf{B}_i\mathbf{C}_i^{-1}\mathbf{K}_i \\ \mathbf{O}_i &= \tilde{\mathbf{O}}_i - \frac{1}{2}\mathbf{B}_i\mathbf{C}_i^{-1}\Phi_i & \mathbf{N}_i &= \tilde{\mathbf{N}}_i - \frac{1}{2}\mathbf{B}_i\mathbf{C}_i^{-1}\Pi_i \\ \mathbf{T}_i &= \tilde{\mathbf{T}}_i - \frac{1}{2}\mathbf{K}_i^T\mathbf{C}_i^{-1}\Phi_i & \mathbf{U}_i &= \tilde{\mathbf{U}}_i - \frac{1}{2}\mathbf{K}_i^T\mathbf{C}_i^{-1}\Pi_i \\ \mathbf{V}_i &= \tilde{\mathbf{V}}_i - \frac{1}{2}\Phi_i^T\mathbf{C}_i^{-1}\Pi_i\end{aligned}\quad (5.14)$$

The system of difference Eqs. (5.14) is integrated backwards starting from the following final

conditions:

$$\begin{aligned}
 \vartheta_{N+1} &= 0 & \mathbf{Q}_{N+1} &= \mathbf{J}_x^{N+1\top} & \mathbf{Z}_{N+1} &= \mathbf{J}_b^{N+1\top} \\
 \mathbf{G}_{N+1} &= \mathbf{J}_w^{N+1\top} & M_{N+1} &= \mathbf{J}_{t_f}^{N+1} & \mathbf{P}_{N+1} &= \frac{1}{2} \mathbf{J}_{xx}^{N+1} \\
 \mathbf{R}_{N+1} &= \frac{1}{2} \mathbf{J}_{bb}^{N+1} & \mathbf{W}_{N+1} &= \frac{1}{2} \mathbf{J}_{ww}^{N+1} & Y_{N+1} &= J_{t_f t_f}^{N+1} \\
 \mathbf{S}_{N+1} &= \mathbf{J}_{xb}^{N+1} & \mathbf{O}_{N+1} &= \mathbf{J}_{xw}^{N+1} & \mathbf{N}_{N+1} &= \mathbf{J}_{xt_f}^{N+1} \\
 \mathbf{T}_{N+1} &= \mathbf{J}_{bw}^{N+1} & \mathbf{U}_{N+1} &= \mathbf{J}_{bt_f}^{N+1} & \mathbf{V}_{N+1} &= \mathbf{J}_{wt_f}^{N+1}
 \end{aligned} \tag{5.15}$$

The auxiliary matrices defined before in Eqs. (5.8) can be rewritten considering the new expressions introduced to express the cost functional partials:

$$\begin{aligned}
 \tilde{\mathbf{G}}_i &= [\mathbf{L}_w^i + \mathbf{Q}_{i+1} \mathbf{F}_w^i + \mathbf{G}_{i+1}] & \tilde{M}_i &= [L_{t_f}^i + \mathbf{Q}_{i+1} \mathbf{F}_{t_f}^i + M_{i+1}] \\
 \tilde{\mathbf{U}}_i &= [\mathbf{S}_{i+1} \mathbf{F}_{t_f}^i + \mathbf{U}_{i+1}] & \tilde{\mathbf{T}}_i &= [\mathbf{S}_{i+1} \mathbf{F}_w^i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{w}}, \bar{t}_f) + \mathbf{T}_{i+1}] \\
 \tilde{\mathbf{N}}_i &= [\mathbf{L}_{xt_f}^i + \mathbf{Q}_{i+1} \mathbf{F}_{xt_f}^i + 2\mathbf{F}_x^{i\top} \mathbf{P}_{i+1} \mathbf{F}_{t_f}^i + \mathbf{F}_x^{i\top} \mathbf{N}_{i+1}] \\
 \tilde{\mathbf{O}}_i &= [\mathbf{L}_{xw}^i + \mathbf{Q}_{i+1} \mathbf{F}_{xw}^i + 2\mathbf{F}_x^{i\top} \mathbf{P}_{N+1} \mathbf{F}_w^i + \mathbf{F}_x^{i\top} \mathbf{O}_{i+1}] \\
 \tilde{\mathbf{P}}_i &= [\mathbf{L}_{ut_f}^i + \mathbf{Q}_{N+1} \mathbf{F}_{ut_f}^i + 2\mathbf{F}_u^{i\top} \mathbf{P}_{i+1} \mathbf{F}_{t_f}^i + \mathbf{F}_u^{i\top} \mathbf{N}_{i+1}] \\
 \tilde{\mathbf{\Phi}}_i &= [\mathbf{L}_{uw}^i + \mathbf{Q}_{i+1} \mathbf{F}_{uw}^i + 2\mathbf{F}_u^{i\top} \mathbf{P}_{i+1} \mathbf{F}_w^i + \mathbf{F}_u^{i\top} \mathbf{O}_{i+1}] \\
 \tilde{\mathbf{V}}_i &= [\mathbf{L}_{wt_f}^i + \mathbf{Q}_{i+1} \mathbf{F}_{wt_f}^i + 2\mathbf{F}_w^{i\top} \mathbf{P}_{i+1} \mathbf{F}_{t_f}^i + \mathbf{V}_{i+1} + \mathbf{F}_w^{i\top} \mathbf{N}_{i+1} + \mathbf{O}_{i+1} \mathbf{F}_{t_f}^i] \\
 \tilde{\mathbf{W}}_i &= \frac{1}{2} [\mathbf{L}_{ww}^i + \mathbf{Q}_{i+1} \mathbf{F}_{ww}^i + 2\mathbf{F}_w^{i\top} \mathbf{P}_{i+1} \mathbf{F}_w^i + 2\mathbf{W}_{i+1} + \mathbf{F}_w^{i\top} \mathbf{O}_{i+1} + \mathbf{O}_{i+1}^T \mathbf{F}_w^i] \\
 \tilde{Y}_i &= \frac{1}{2} [L_{t_f t_f}^i + \mathbf{Q}_{i+1} \mathbf{F}_{t_f t_f}^i + 2\mathbf{F}_{t_f}^{i\top} \mathbf{P}_{i+1} \mathbf{F}_{t_f}^i + 2Y_{i+1} + \mathbf{F}_{t_f}^{i\top} \mathbf{N}_{i+1} + \mathbf{N}_{i+1}^T \mathbf{F}_{t_f}^i]
 \end{aligned} \tag{5.16}$$

The new backward equations are obtained replacing the optimal control law in the expanded HJB equation and removing the minimisation operator. As usual the coefficients multiplying the same differential are equating to derive the optimal cost functional variation at the previous step. Some aspects must be discussed to apply the general DDP algorithm. First, how to include the final time inside the expression of the system dynamics. The other open points are how to compute the optimal variations to update the static variables and final time. Finally, which are the conditions to be respected to accept a DDP iteration. The first point is straightforward. Given a general system of ordinary differential equations representing the system dynamics a change of variable is introduced considering an adimensional time, τ . This adimensional time is the ratio of the independent time variable and a nominal final time, \bar{t}_f , that is used as first guess solution.

$$\tau = \frac{t}{\bar{t}_f} \tag{5.17}$$

Thanks to this change of variable the independent variable is no longer the dimensional time, t , but the adimensional time, τ , that will be always a variable limited in the interval $[0,1]$. The final time will appear explicitly inside the system dynamics.

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, t) \rightarrow \frac{d\mathbf{x}}{d\tau} = t_f \mathbf{f}(\mathbf{x}, \tau) \tag{5.18}$$

Thanks to this trick the partial derivatives of the system dynamics and of the cost functional with respect to the free final time can be computed.

The second aspect to be included to complete the updated version of the DDP algorithm is how to compute the optimal variations for the final time and static variables. The static variables are not affecting the dynamics during the propagation but only the initial conditions. Therefore, their role is similar from an algorithm point of view to the vector of Lagrange multipliers and its variation is computed at the second stage of the DDP algorithm. A different reasoning must be done for the final time variation. In this case this is a variable which is affecting the propagation of the dynamics. Its role is similar to an optimal policy and this is why the optimal variation of the final time is carried out at the first stage of the DDP algorithm. The starting point is always the Taylor expansion of the cost functional considering its variation when there is a single change of the static parameter or the free final time.

$$J^*(\mathbf{x}_1(\mathbf{w} + \delta\mathbf{w}); t) = J^*(\mathbf{x}_1(\mathbf{w}); t) + \frac{\partial J}{\partial \mathbf{x}_1} \frac{\partial \mathbf{x}_1}{\partial \mathbf{w}} \delta\mathbf{w} + \frac{\partial J}{\partial \mathbf{w}} \delta\mathbf{w} + \frac{1}{2} \delta\mathbf{w}^T \left[\frac{\partial \mathbf{x}_1}{\partial \mathbf{w}} \frac{\partial^2 J}{\partial \mathbf{x}_1^2} \frac{\partial \mathbf{x}_1}{\partial \mathbf{w}} + \frac{\partial^2 J}{\partial \mathbf{w} \partial \mathbf{x}_1} \frac{\partial \mathbf{x}}{\partial \mathbf{w}} + \frac{\partial J}{\partial \mathbf{x}_1} \frac{\partial^2 \mathbf{x}_1}{\partial \mathbf{w}^2} \right] \delta\mathbf{w} \quad (5.19)$$

with $J^*(\mathbf{x}_1(\mathbf{w} + \delta\mathbf{w}); t)$ representing the expected optimal cost functional due to a variation of the only static parameter vector, $\delta\mathbf{w}$. Generally, the static parameter affects only the variation of the initial condition. The cost functional does not depend explicitly on the initial condition. This means that all the partial derivatives with respect to the initial conditions are equal to 0. The previous expansion is simplified in the following way:

$$J^*(\mathbf{x}_1(\mathbf{w} + \delta\mathbf{w}); t) = J^*(\mathbf{x}_1(\mathbf{w}); t) + \mathbf{J}_w^T \delta\mathbf{w} + \frac{1}{2} \delta\mathbf{w}^T \mathbf{J}_{ww} \delta\mathbf{w} \quad (5.20)$$

The differentiation with respect to the static variable variation allows to determine the value of the updated static parameter.

$$\delta\mathbf{w} = -\mathbf{J}_{ww}^{-1} \mathbf{J}_w \quad (5.21)$$

A similar reasoning is done also for the free final time. The Taylor expansion of the cost functional depending only on the free final time is reported.

$$J^*(\mathbf{x}_1; t_f + \delta t_f) = J^*(\mathbf{x}_1; t_f) + J_{t_f} \delta t_f + \mathbf{J}_x^T \mathbf{F}_{t_f} \delta t_f + \frac{1}{2} \delta t_f (J_{t_f t_f} + \mathbf{J}_{t_f x} \mathbf{F}_{t_f} + \mathbf{J}_{x x} \mathbf{F}_{t_f t_f} + \mathbf{J}_{x t_f}) \delta t_f \quad (5.22)$$

with $J^*(\mathbf{x}_1; t_f + \delta t_f)$ representing the expected optimal cost functional due to a variation of the only final free time, δt_f . The expansion can be simplified because according to the system dynamics formulation the term $\mathbf{F}_{t_f t_f} = 0$. However, according to the formulation of the augmented cost functional with linear or quadratic endpoint constraint the partial derivatives of the cost functional with respect to the final time can be equal or not equal to 0. Therefore, no further simplification is done to derive the optimal variation of the free final time.

$$\delta t_f = - [J_{t_f t_f} + \mathbf{J}_{t_f x} \mathbf{F}_{t_f} + \mathbf{V}_{x t_f}]^{-1} [J_{t_f} + \mathbf{F}_{t_f}^T \mathbf{J}_x^T] \quad (5.23)$$

The last step for the complete update of the DDP algorithm is the evaluation of the acceptance conditions for the optimal variation defined for the free final time and static parameters. Considering the Taylor expansions developed before an estimate of the difference between the new

functional cost with the updated variation and the nominal one can be evaluated. These estimates are used to check if the new cost functional computed after the forward integration of the dynamics considering the new control law is consistent with the Taylor expansion.

$$\theta_w = J^*(\mathbf{x}_0(\mathbf{w} + \delta\mathbf{w}); t) - J^*(\mathbf{x}_0(\mathbf{w}); t) = \mathbf{J}_w^T \delta\mathbf{w} + \frac{1}{2} \delta\mathbf{w}^T \mathbf{J}_{ww} \delta\mathbf{w} \quad (5.24)$$

$$\begin{aligned} \theta_{t_f} = J^*(\mathbf{x}_1; t_f + \delta t_f) - J^*(\mathbf{x}_1; t_f) = & J_{t_f} \delta t_f + \mathbf{J}_x^T \mathbf{F}_{t_f} \delta t_f + \frac{1}{2} \delta t_f (J_{t_f t_f} + \mathbf{J}_{t_f x}^T \mathbf{F}_{t_f} \\ & + \mathbf{J}_{xx}^T \mathbf{F}_{t_f t_f} + \mathbf{J}_{xt_f}) \delta t_f \end{aligned} \quad (5.25)$$

The new estimates are summed with the estimates defined in Chapter 3 for the optimal control variation and Lagrange multiplier variations to verify if the overall variation of the cost functional is consistent or not.

From a schematic point of view, only 2 modifications to the DDP algorithm presented at the end of Chapter 3 must be introduced:

1. in step 4 the optimal control law is including also the optimal final time variation while the static variables are considered to be constant.
2. in step 7 the Lagrange multipliers correction is performed together with the static variables correction.

5.2 Sundman transformation

One of the largest differences between interplanetary and planetary transfers is the multi revolutionary aspect of the spacecraft trajectories. Most of the interplanetary missions involve ballistic arcs which are not involving a large number of revolutions. For planetary missions given the same transfer time the number of revolutions experienced by the spacecraft is higher. This difference affects the performance of the DDP algorithm regarding the time discretisation. The time discretisation of a single spacecraft interplanetary arc generates nodes which are not exactly distributed but the overall mesh is not coarse. When multiple revolutions are involved in the transfers, the difference in velocity of the spacecraft creates a large set of nodes close to the apocenter and the mesh is coarse close to the pericenter. The consequence is that during the backward sweep there is the possibility that a discontinuity in the partials computation is generated not due to the non linearity of the system dynamics, but due to the large distance between two consecutive nodes. This issue can be solved relying on a common instrument used in orbital mechanics which is the Sundman transformation [61]. A Sundman transformation is a method to change the independent variable from time to an orbital anomaly. The expression using the true anomaly, eccentric anomaly and mean anomaly are reported [59]:

$$dt = \frac{1}{\sqrt{\mu p}} \left(\frac{p}{1 + f \cos L + g \sin L} \right)^2 d\nu \quad (5.26)$$

$$dt = \frac{1}{\sqrt{\mu}} \left(\frac{p}{1 - f^2 - g^2} \right)^2 dl \quad (5.27)$$

$$dt = \frac{1}{\sqrt{\mu}} \frac{p}{1 + f \cos L + g \sin L} \sqrt{\frac{p}{1 - f^2 - g^2}} dE \quad (5.28)$$

The Sundman transformation introduces the choice between four different independent variables: time, true anomaly ν , mean anomaly l , and eccentric anomaly E . The fastest way to justify the choice of the best parameter as independent variable is to check how the nodes are distributed in a single revolution of an orbit when a uniform vector of the independent variable is considered. The result of the analysis is presented in Figures 5.1-5.4.

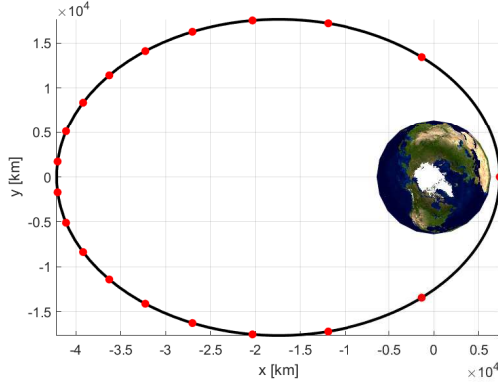


Figure 5.1: Nodes distribution using time as independent variable.

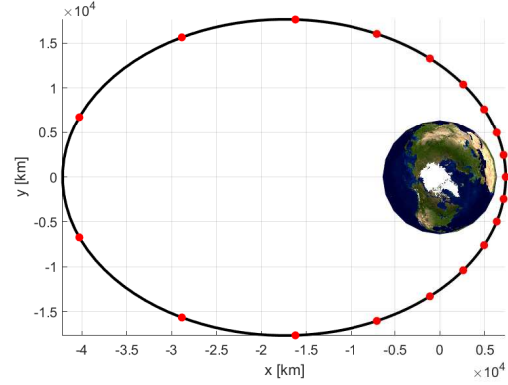


Figure 5.2: Nodes distribution using true anomaly as independent variable.

The nodes distribution using time or mean anomaly as independent variables is the same because the two variables are directly correlated through the orbital mean motion. The nodes are crowding the region close to the apocenter while the number of nodes nearby the pericenter is sparse as it is shown in Figures 5.1 and 5.3. The result is that the angular separation between two consecutive nodes close to the pericenter is relatively larger than the angular distance at the apocenter. The opposite reasoning can be done if true anomaly is considered as independent variable looking at Figure 5.2. The uniform distribution of the true anomaly generates a large number of nodes close to the pericenter while the node distribution at the apocenter is sparse.

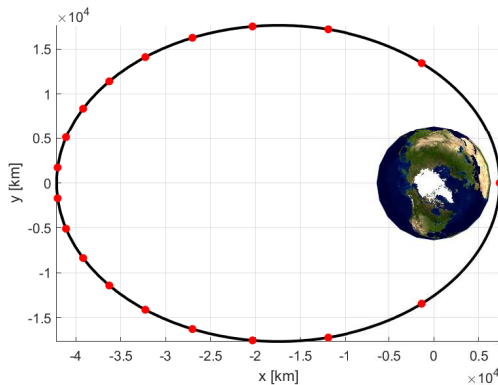


Figure 5.3: Nodes distribution using mean anomaly as independent variable.

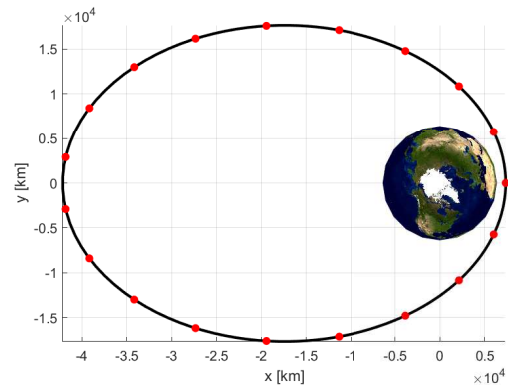


Figure 5.4: Nodes distribution using eccentric anomaly as independent variable.

The result is that the angular separation between two consecutive nodes close to the apocenter is relatively larger than the angular distance at the pericenter. Finally, Figure 5.4 present the nodes distribution when an equally spaced eccentric anomaly vector is used as independent variable. The mesh is uniform so that the angular separation between two consecutive nodes is always the same despite the position on the orbit. This property is very important because it allows to generate a finer mesh increasing the number of nodes used to discretise the eccentric anomaly vector.

Therefore, the most efficient way to use the Sundman transformation is to change the independent variable from time to eccentric anomaly. A discretisation in terms of eccentric anomaly has the property to allow a uniform discretisation of the spacecraft orbit despite the difference in velocity between pericenter and apocenter. This allows to overcome the issue to generate discontinuities in the partials computation due to a large distance between two consecutive nodes. The analytical formulation of the Sundman transformation allows to keep an analytical expression for the system dynamics also when eccentric anomaly is used as main independent variable. This means, that the overall scheme based on the symbolic formulation still works properly. The discussion about the node distribution is more relevant as the orbit considered is more eccentric as the ones used for GTO to GEO transfers. This is why this specific test case is used at the end of the chapter to prove the effectiveness of the DDP algorithm coupled with the eccentric anomaly Sundman transformation.

5.3 Multiple revolutions

For planetary transfers multiple revolutions represent one of the critical points during the resolution of optimal control problems. The larger is the number of revolutions the more difficult is the convergence to be achieved in classic indirect problems when the state is represented in terms of position and velocity. In particular, when classic direct collocation methods are used the higher the number of orbit revolutions is the higher is the dimension of the non-linear optimal control problem. One of the disadvantages in using direct collocation methods is the difficulty in handling large dimensional problems. The introduction of a Sundman transformation to change the independent variable from time to an anomaly improves the convergence properties but it is not sufficient to overcome the issue related to the dimension of the problem.

When orbital elements are used as state representation of the dynamics is a complete set of orbital elements as independent variables can be introduced. The orbital elements rates does not depend explicitly on time, but they depend on one of the main orbit anomalies introduced, so that the transformation is simplifying the formulation of the dynamics instead of making it more difficult as in the case of Cartesian coordinates formulation. It is obvious that the design of the problem changes when eccentric anomaly becomes the independent variable. Indeed, when time is used as independent variable, a given transfer time is prescribed by the mission requirements and the number of nodes for the mesh generation is completely decoupled from their distribution in the different revolutions. This means that it is not possible to refine the mesh in an even way modifying the number of discrete time steps. If eccentric anomaly is the independent variable, the new design variable is not the transfer time, but the number of total revolutions. The use of the eccentric anomaly as independent variable is particularly advantageous because thanks to its property to generate a uniform number of nodes in a single revolution as shown in the previous

section. The same analysis can be carried out to check what happens to the nodes distribution when a uniform distribution of time or eccentric anomaly is used as independent variable and multiple revolutions are involved.

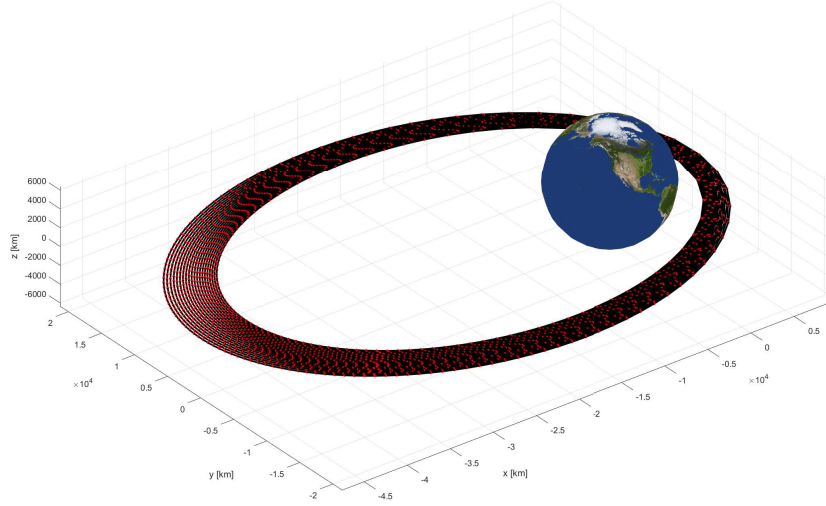


Figure 5.5: Nodes distribution using time as independent variable when multiple revolutions are considered.

When time or mean anomaly is considered as independent variable for multiple revolutions transfers the behaviour is similar to the single revolution case. The node distribution is crowding the apocenters of the different spirals but the angular separation between two consecutive nodes close to the pericenter increases as the size of the orbit as shown in Figure 5.5. The consequence is that it is even more difficult to keep the continuity and differentiability properties required to solve the backward propagation.

If the eccentric anomaly is used as independent variable the nodes distribution is the one presented in Figure 5.6. The angular separation between two consecutive nodes is constant for a single revolution. If correspondent nodes on different spirals are considered, these points create a straight line. This geometric feature has a very important consequence because it means that the angular separation between two consecutive nodes is not only constant for all the revolutions but its value is the same. This is very helpful to improve the convergence of the DDP algorithm and it can be used to identify a criterion to tune the mesh refinement. The resulting total number of discrete nodes is the product of the number of discrete nodes required per single revolution multiplied times the number of total revolution which is the design parameter. In this case the variation of the number of discrete steps per single revolution is a valid method to refine the mesh in an even way.

Differently from collocation methods, DDP algorithms do not suffer from the dimension of the non-linear optimal control problem, but smooth continuous and differentiation properties for the cost functional must be present. If time is used as independent variable the discretisation results in a coarse mesh where the difference between two consecutive nodes in the transfer trajectory is so large to make the DDP algorithm think that a discontinuity in the cost functional

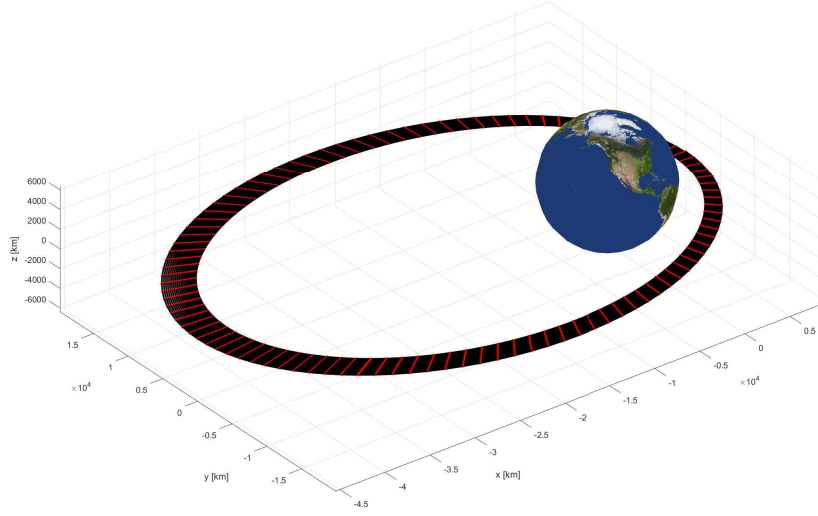


Figure 5.6: Nodes distribution using eccentric anomaly as independent variable when multiple revolutions are considered.

partials is present. The lost of continuity property does not make the DDP algorithm to converge. A refinement of the discrete time vector associated to a larger number of discrete steps is not a solution to this convergence problem because there is no way to know "a priori" what is the maximum distance between two consecutive nodes to make the convergence attainable. Moreover, the larger the number of discrete steps the larger will be the overall computation time of the DDP algorithm that can lead to an halt of the algorithm. This is exactly what happened in the only existence reference in literature [86] where Hybrid Differential Dynamic Programming (HDDP) is used to solve a GTO to GEO transfer: the use of numerical STMs for the computation of the cost functional partial derivatives, and the large number of discrete steps forced the computation to stop after 24 hours even if the design number of revolutions was equal to 200. The criterion for the selection of the number of discrete steps for multiple revolutions orbit transfers after the application of Sundman transformation in terms of eccentric anomaly is summarised in Figure 5.7.

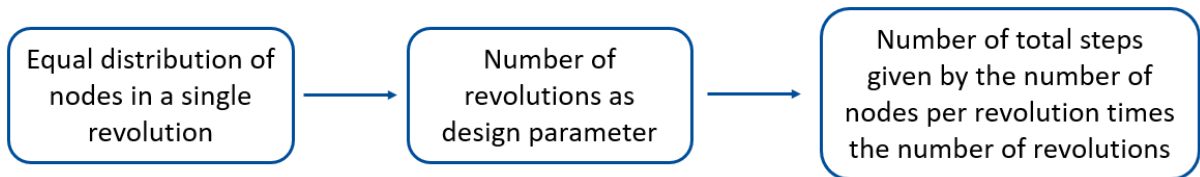


Figure 5.7: Number of steps definition procedure in case Sundman transformation in terms of eccentric anomaly is applied.

It is clear that after Sundman transformation time is a dependent variable and it must be recovered from the dynamics to know what is the actual transfer time associated to the designed number of revolutions. In the next section a GTO to GEO transfer problem is solved considering both time and eccentric anomaly as independent variables. The aim of the test case is to

check first the ability of the DDP algorithm to handle many-revolution transfers, and to compare the advantages of the Sundman transformation coupled with the orbital elements state representation. The two key parameters used to compare the two different representations will be the convergence properties, and the computational time to solve the same orbital transfer.

5.4 Results

A simple GTO to GEO transfer starting from a LEO orbit is used as test case to prove the effectiveness of Sundman transformation in improving the converge properties of DDP for multi-revolution transfers. The same transfer is solved first with time used as independent variable, and finally with eccentric anomaly after the application of the Sundman transformation. The input used for the GTO transfer are reported in Table 5.1. The nominal control provided to the DDP algorithm is always a simple tangential thrust.

Table 5.1: Initial orbital elements for the GTO to GEO transfer.

Inputs	Numerical value
Apogee height	35793 km
Perigee height	300 km
Orbit inclination	27.5°
RAAN w.r.to Earth	30°
Pericenter anomaly	0°
True anomaly	0°
Initial tangential thrust	50 mN

The target orbit to be achieved is a GEO orbit whose characteristics are reported in Table 5.2.

Table 5.2: Input for the GTO to GEO reference transfer.

Semi-major axis, km	Eccentricity	Inclination, deg	RAAN, deg	Pericenter anomaly, deg
42164	0	0	30	0

The results for the two simulations run with time as independent variable and eccentric anomaly as independent variable, respectively, are summarized in Table 5.3.

Table 5.3: Comparison between the two different simulations.

Inputs	Time Independent	Eccentric Anomaly
Time of flight	60 days	585 days
Number of revolutions	83	600
Number of steps	9000	2000
Computational time	6.2151 hours	30.2581 minutes
Propellant Mass	200.11622 kg	269.0989 kg
Maximum Thrust	1 N	416.178 mN

It is obvious that there is no correspondence between time of flight and number of revolutions due to incapacity to know a priori the relationship between the two variables. However, when time is used as independent variable the minimum number of time steps required to achieve convergence was equal to 9000. This number is very large and it is the main reason for an increase in the total computational time. Another difference that it is always related to the transfer is the magnitude of the control thrust. The larger is the transfer time the lower is the maximum thrust magnitude provided by the engine. In the formulation using eccentric anomaly as main independent variable a first guess for the number of steps equal to 2000 has been used. This is not the minimum number of steps guaranteeing convergence. Therefore, the same result can be achieved reducing further the total number of steps. Despite the larger number of revolutions the convergence does not require a relatively large computational time and this happens thanks to the new mesh discretisation. The total propellant mass results high because of the large time of flight but the maximum engine thrust is smaller and compatible with state-of-the-art electric thrusters that have been already used for space missions. The current literature presents some examples of HDDP coupled with Sundman transformation to solve a GTO transfer but the number of revolution is small [86]. In the HDDP the partials computation is carried out thanks to the use of STMs which are an optimal tool when an analytical formulation of the STM is provided. If such analytical formulation is not available the STMs must be computed integrating a large system of partial differential equations which is decreasing the performance of the algorithm. In this case the analytical expressions to evaluate the transition function partial derivatives are considerably improving the computational speed because they avoid to solve for each iteration of the algorithm a system of partial differential equations. The results of the GTO to GEO transfer discussed so far are shown in Figures 5.8-5.13. In particular, in Figures 5.8-5.10 the time history of orbital elements starting from the GTO orbit to GEO is presented. The magnitude of the control thrust together with the variation of each single component is plotted in Figures 5.11 and 5.12 and, finally, the 3D trajectory of the optimal transfer is presented in Figure 5.13.

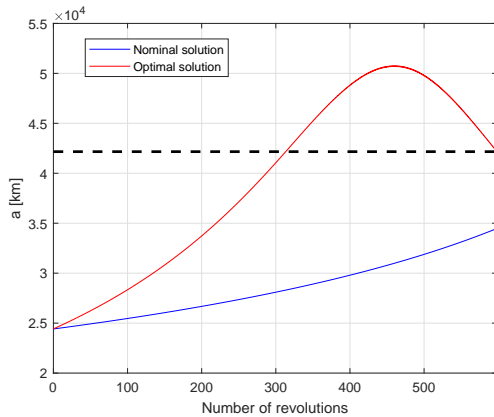


Figure 5.8: Nominal vs optimal semi-major axis trajectory.

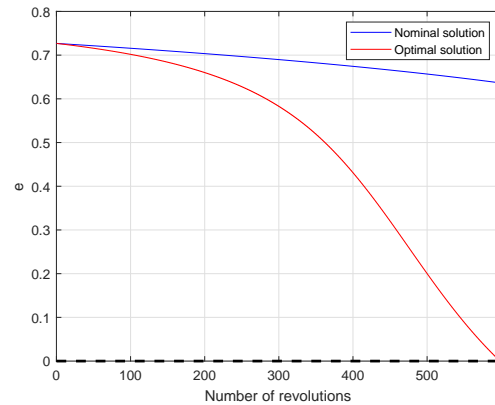


Figure 5.9: Nominal vs optimal eccentricity trajectory.

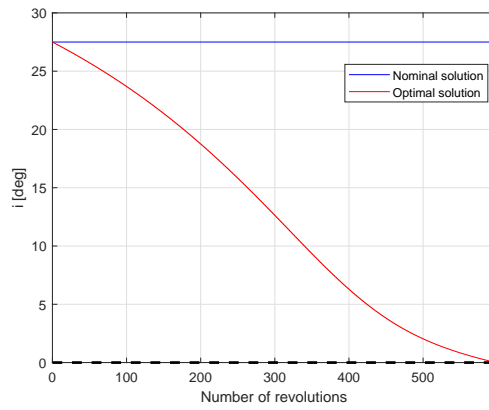


Figure 5.10: Nominal vs optimal orbital inclination trajectory.

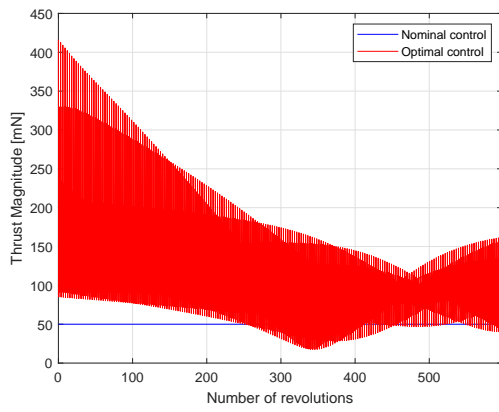


Figure 5.11: Nominal vs optimal thrust magnitude trajectory.

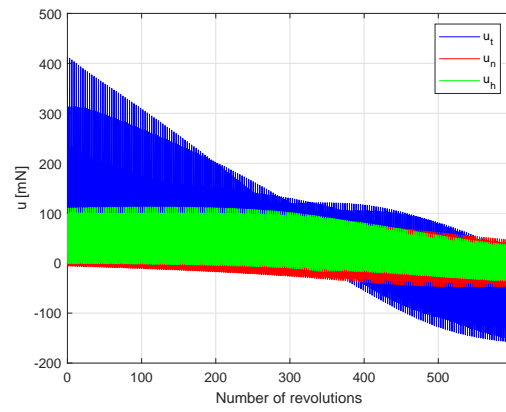


Figure 5.12: Nominal vs optimal thrust components trajectory.

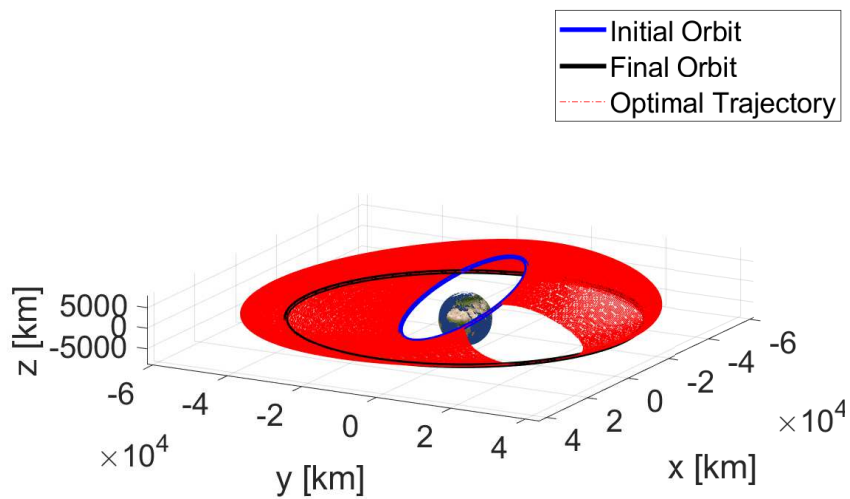


Figure 5.13: Final interplanetary GTO to GEO 3D trajectory.

CHAPTER 6

Robustness and partials computation analysis

This chapter will compare the methodology used to compute the cost functional partial derivatives in this research with the HDDP algorithm considered the state-of-the-art of DDP algorithms [43][86][59]. This method is a modification of the classic DDP algorithm based on the State Transition Matrices (STMs) for the evaluation of the cost functional partial derivatives. The HDDP extends the application of the algorithm to multiple phases non-linear problems and introduces a trust-region method to tune both the control variation magnitude and the positive definiteness of the Hessian matrix with respect to the control action. The chapter will briefly describe how STMs are defined and the system of differential equations to be used for their evaluation. The next part will be devoted to the comparison between the two approaches for the cost functional partials computation: the analytical formulation of the numerical scheme, and the use SMTs.

6.1 State transition matrix formulation

One of the possibilities to solve the system dynamics is to predict the variation of the state as function of the initial condition. This variation is then summed to the satellite initial state to get the state at the local time. This method is very useful whenever a robustness analysis is carried out for a given control law. Indeed, a system is defined robust if a small variation of the initial condition is not generating a large variation of the final state. This verification is immediate when STMs are involved in the problem.

A generic dynamical system such as the spacecraft motion can be modelled considering a system of first-order partial differential equations:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}(t); t) \quad (6.1)$$

The general solution to the system is called *flow*, ϕ , because it maps the initial state at the initial time, t_1 , to the time, t :

$$\mathbf{x}(t) = \phi(\mathbf{x}_1, t_1; t) \quad (6.2)$$

The system dynamics can be rewritten in terms of the solution flow:

$$\frac{d\phi}{dt} = \mathbf{f}(\phi(\mathbf{x}_1, t_1; t); t) \quad (6.3)$$

The local trajectory dynamics, $\delta\mathbf{x}$, can be defined by applying a Taylor series expansion about the reference trajectory given an initial deviation, $\delta\mathbf{x}_1$:

$$\delta\mathbf{x} = \phi(t; \mathbf{x}_0 + \delta\mathbf{x}_1, t_1) - \phi(t; \mathbf{x}_1, t_1) \quad (6.4)$$

The z -th order solution can be written using the Einstein summation to simplify the computations [82].

$$\delta x^j(t) = \sum_{v=1}^z \frac{1}{v!} \phi_{(t,t_1)}^{j,\gamma_1 \dots \gamma_v} \delta x_1^{\gamma_1} \dots \delta x_1^{\gamma_v} \quad (6.5)$$

where $\gamma_j \in 1, \dots, n_x$, superscripts γ_j denote the γ_j -th component of the state vector, and

$$\phi_{(t,t_1)}^{j,\gamma_1 \dots \gamma_v}(t; \mathbf{x}_1, t_1) = \frac{\partial^v \phi^j(t; \mathbf{x}_1, t_1)}{\partial x_1^{\gamma_1} \dots \partial x_1^{\gamma_v}} \quad (6.6)$$

We call the higher-order partials of the solution flow Eq. (6.7) the State Transition Tensors (STTs) which represent the generalisation of the STMs. These variables are mapping the initial deviations to the current time. The first-order case ($p = 1$) reduces to the usual state transition matrix. Park and Scheeres [82] describe carefully the procedure to obtain the differential equations for the evaluation of the STTs. In particular, for the HDDP we are interested in the evaluation of the STTs up to the second-order.

$$\dot{\phi}^{i,a} = f^{i,\alpha} \phi^{\alpha,a} \quad (6.7)$$

$$\dot{\phi}^{i,ab} = f^{i,\alpha} \phi^{\alpha,ab} + f^{i,\alpha\beta} \phi^{\alpha,a} \phi^{\beta,b} \quad (6.8)$$

with α and β which are generic indices not to be confused with the optimal control law coefficient matrix defined in Chapter 4. The initial conditions of the STTs are $\phi_{(t_1,t_1)}^{i,a} = 1$, if $i = 1$ and zero otherwise, which is equivalent to say that the first order STT is the identity matrix and all the higher order STTs are initialised to zero. Once these STTs are computed, they serve a role identical to the STMs except that higher-order effects are now included, and thus the solution is non-linear. Therefore, the main advantage in using STTs is that the local non-linear motion of a spacecraft trajectory can be mapped analytically and requires no integration apart from the determination of the STTs.

Majji et. al [87] offer a methodology to solve the system of Ordinal Differential Equations (ODEs) involving the STTs. This approach is created to fit MATLAB way to solve ODEs and it requires the introduction of some auxiliary functions to transform the STTs into column vectors and second-order tensors. The first auxiliary function turns a generic STT into a column vector. This operation is carried out easily plugging each column at the bottom of the previous one starting from the last column index to the first one. In case a multi-dimensional array is

considered the same operation is repeated for each slice of the multi-dimensional array. This simple operation is denoted as $\text{Vec}(\cdot)$ and it is defined according to the following [87]:

$$\text{Vec}(\phi_{i_1 i_2 i_3}) = \phi(i_1 + n_x(i_2 - 1) + n_x^2(i_3 - 1), 1) \quad (6.9)$$

with i_1, i_2 , and i_3 denoting the indices associated to the first, second and third dimension, respectively. The following example proposed considering both a 2x3 matrix and a 2x2x2 multi-dimensional array to understand how the auxiliary function works.

$$\text{Vec} \left(\begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix} \right) = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}, \quad \text{Vec} \left(\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 5 & 7 \\ 6 & 8 \end{bmatrix} \right) = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{bmatrix} \quad (6.10)$$

The second auxiliary operation transforms a higher-order STT into a matrix. This operation is denoted as $\text{Mat}(\cdot)$ and it reshuffles the different layers of the multi-dimensional array starting from the last one plugging them next to the previous layer until the first layer. This operation can be defined as follows:

$$\text{Mat}(\phi_{i_1 i_2 i_3}) = \phi(i_1, i_2 + n_x(i_3 - 1)) \quad (6.11)$$

Also in this case a simple example is used to show how the auxiliary operation works:

$$\text{Mat} \left(\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 5 & 7 \\ 6 & 8 \end{bmatrix} \right) = \begin{bmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \end{bmatrix} \quad (6.12)$$

Thanks to these two auxiliary operations it is possible to rewrite Eqs. (6.7-6.8) in the following ways:

$$\text{Vec}(\dot{\phi}^{(1)}) = [\mathbf{I}_{n_x} \otimes \text{Mat}(\nabla_x \mathbf{f})] \text{Vec}(\phi^{(1)}) \quad (6.13)$$

$$\text{Vec}(\dot{\phi}^{(2)}) = [\mathbf{I}_{n_x} \otimes \mathbf{I}_{n_x} \otimes \text{Mat}(\nabla_{xx} \mathbf{f})] \text{Vec}(\phi^{(2)}) + [\text{Mat}(\phi^{(1)})^T \otimes \text{Mat}(\phi^{(1)})^T \otimes \mathbf{I}_{n_x}] \text{Vec}(\nabla_{xx} \mathbf{f}) \quad (6.14)$$

where \otimes denotes the Kronecker product operation [87], \mathbf{I}_{n_x} is the identity matrix, $\phi^{(1)}$ and $\phi^{(2)}$ are the first-order and second-order STTs, respectively. The other two variables $\nabla_x \mathbf{f}$ and $\nabla_{xx} \mathbf{f}$ represent the Jacobian and Hessian matrices of the system continuous dynamics which are a second-order and third-order STTs, respectively. The expression of the second-order derivative for the application of the approach derived by Majii et al. [87] is different from the one presented in Chapter 2. Indeed, the third-order STT is not computed as the collection of the Hessian matrices of each scalar component of the vector dynamics put along the last dimension. The third-order STT is built considering each layer as the differentiation of each element of the rows of the Jacobian matrix.

$$\nabla_{xx} \mathbf{f} = \left(\begin{bmatrix} \frac{\partial^2 f_1}{\partial x_1^2} & \frac{\partial^2 f_1}{\partial x_2 \partial x_1} & \cdots & \frac{\partial^2 f_1}{\partial x_n \partial x_1} \\ \frac{\partial^2 f_2}{\partial x_1^2} & \frac{\partial^2 f_2}{\partial x_2 \partial x_1} & \cdots & \frac{\partial^2 f_2}{\partial x_n \partial x_1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f_n}{\partial x_1^2} & \frac{\partial^2 f_n}{\partial x_2 \partial x_1} & \cdots & \frac{\partial^2 f_n}{\partial x_n \partial x_1} \end{bmatrix}, \dots, \begin{bmatrix} \frac{\partial^2 f_1}{\partial x_1 \partial x_n} & \frac{\partial^2 f_1}{\partial x_2 \partial x_n} & \cdots & \frac{\partial^2 f_1}{\partial x_n^2} \\ \frac{\partial^2 f_2}{\partial x_1 \partial x_n} & \frac{\partial^2 f_2}{\partial x_2 \partial x_n} & \cdots & \frac{\partial^2 f_2}{\partial x_n^2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f_n}{\partial x_1 \partial x_n} & \frac{\partial^2 f_n}{\partial x_2 \partial x_n} & \cdots & \frac{\partial^2 f_n}{\partial x_n^2} \end{bmatrix} \right) \quad (6.15)$$

Thanks to this formulation both the first-order and second-order STMs variations are expressed as column vectors following a systematic procedure so that it is possible to apply *ode* built-in functions in MATLAB. This is the numerical procedure followed by Lantoine and Russell [43] in the definition of the HDDP algorithm to compute the STMs required to propagate the cost functional partial derivatives in the backward recursion.

6.2 Results

In this section some numerical examples are presented to compare the pros and cons of the analytical formulations of the numerical schemes and the use of STMs for the cost functional partials computation. If the HDDP uses the STMs instrument to check the robustness of the control law, in this research paper a different robustness analysis starting from the feedback control law which are derived by the DDP algorithm.

6.2.1 Analytical vs STM partial evaluation

The first analysis proposed in this section is the comparison between the two different ways to evaluate the cost functional partial derivatives: analytical formulation of the integration scheme and STMs. Even if the HDDP represents the state-of-the-art for the DDP algorithm application it is not efficient to adopt all its routines for all the type of problems. This analysis is very important because the evaluation of the partials is the most expensive part of the DDP algorithm and it is repeated several times at each iteration.

The main advantage in using the analytical evaluation of the transition function partials as in this research paper is the flexibility to adapt such method to any kind of dynamical system. Indeed, the analytical evaluation is based on the type of numerical integration scheme used to solve the dynamics and not the formulation of the system of ordinary differential equations. Moreover, according to the order of the numerical scheme it is possible to increase or decrease the number of function evaluations which are strictly related to the numerical scheme.

The use of the STMs can be very useful whenever an analytical formulation of the STM from the equations of motion is available. For example, a simple propagation of a spacecraft trajectory can be formulated using Laplace coefficients which are expressing the dynamics in terms of the initial position and velocity vectors. This means that an analytical STM is possible and the higher-order derivatives can be defined from the analytical differentiation of the STM. Another advantage is the possibility to compute the STM recursively starting from the values at previous steps. The main drawback is the availability of an analytical STM which depends on the

problem. If the system dynamics is highly non-linear and it does not admit an analytical STM, the only way to compute such instrument is by solving numerically the ordinary differential equations presented before. This task is very expensive from a numerical point of view because it requires the solution to a large number of ordinary differential equations which are reducing the performance of the DDP algorithm. Therefore, this type of partials evaluation is strictly attached to the formulation of the dynamics.

An Earth to Mars interplanetary transfer is used to check the difference between the two approaches. The inputs for the test case are presented in Table 6.1. The state vectors in terms of orbital elements are computed from analytic ephemerides which approximate JPL ephemerides DE405.

Table 6.1: Input for the Earth to Mars interplanetary transfer used for the comparison between STMs and analytical formulation.

Inputs	Numerical value
Departure date	8331.5 MJD
Arrival date	8612.5 MJD
Time of flight	281 days
Number of steps	200
Specific Impulse, I_{sp}	3800 s
Maximum Thrust, u_{max}	100 mN
Spacecraft Mass, m_0	500 kg

The analysis is carried out on a Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz. The results of the optimisation carried out with the two different approaches are summarised in Table 6.2.

Table 6.2: Input for the Earth to Mars interplanetary transfer used for the comparison between STMs and analytical formulation.

Inputs	Analytical formulation	STMs
Elapsed Time	16.06 minutes	136.44 minutes
Maximum Thrust, u_{max}	395.77 mN	396.12 mN
Propellant mass used, m_0	124.59 kg	125.10 kg
Numerical scheme order	RK8	RK45

The result of the optimisation is the same but the computation time is completely different. The analytical computation of the cost function partial derivatives improves the performance of the DDP algorithm with respect to the STM formulation because Gauss' equations cannot provide an analytical STM. This means that the STMs must be computed numerically increasing the number of function evaluations. The order of the numerical scheme used for the two approaches is different because if an *ode78* method is used also to compute the STMs the computational time will definitely increase.

A final discussion regarding the exact number of function evaluations underlines the importance in the correct choice of the tool to compute the cost functional partial derivatives. When a continuous formulation of the DDP is adopted only the partial derivative of the system dynamics are required because no transition function exists. This approach is the same used in the

Mystic software and it requires to solve $n_x + n_x^2$ traditional Riccati-like equations, with n_x being the number of states. The computation of STMs requires to solve $n_x^2 + n_x^3$ ordinary differential equations. This number can be reduced to $n_x^3/2 + n_x^2/2$ if symmetry properties of the second-order STM are considered. However, the number of equations to be solved is still higher than the classic continuous formulation. When the analytical formulation of the integration scheme is used there is not a fixed number of equations to be solved. First, there is not a system of ODEs to be integrated as in the STM approach but only a number of function evaluation that depends on the order of the integration scheme adopted to define the transition function. In any case the number of function evaluations is definitely smaller than the ones used in the STM approach.

6.2.2 Robustness of the control law

One of the most important properties of an optimal control law is the robustness. Robustness is the ability of a control law to provide an efficient result even if the initial condition is perturbed. This topic is fundamental because only when numerical simulations are carried out the initial conditions are fixed while in the real life the initial conditions can depart from the nominal one due to the existence of phenomena that cannot be modelled and create a disturbance in the initial condition.

If the optimal control law is not able to provide an efficient result when the initial condition is perturbed it represents a result that cannot be applied onboard of a satellite computer. One of the many advantages in using DDP is the way the optimal control law is determined. In Chapter 3 it has been discussed that both for the continuous version and the discrete version the optimal control variation is a feedback control law. This property is very important because it means that the optimal control changes accordingly to the variation of the spacecraft state.

It is possible to carry out an analysis to check the robustness property of the optimal control law derived by the DDP algorithm. The example considers the same Earth to Mars interplanetary transfer presented in the previous section and whose input variables are defined in Table 6.1. The initial condition is fully perturbed with a vector of small orbital elements that is presented in Table 6.3.

Table 6.3: Initial orbital elements and list of initial disturbances added.

Orbital element	Initial condition	Perturbation
Semi-major axis, a_1	1.41745575e8 km	1000 km
Eccentricity, e_1	0.0596897	0.05
Inclination, i_1	0°	1°
RAAN, Ω_1	0°	1°
Pericenter anomaly, ω_1	242°	1°
True anomaly, ν_1	148.56°	0.5°

The robustness analysis is performed considering the new initial condition and computing the new control law starting from the feedback coefficients of the optimal control law, \mathbf{u}_{opt} . This process requires to ask the coefficients of the control law, α_{opt} and β_{opt} , as output of the DDP optimisation where no disturbance is added to the initial conditions.

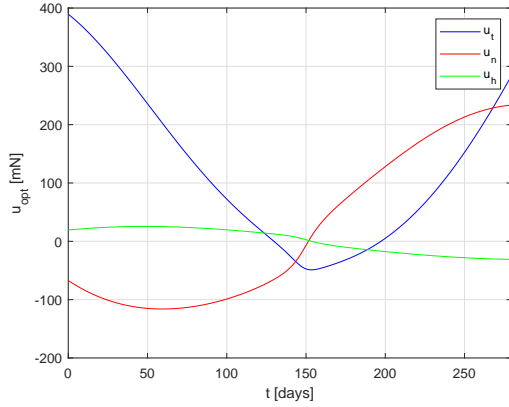


Figure 6.1: Optimal control components associated to the unperturbed initial conditions.

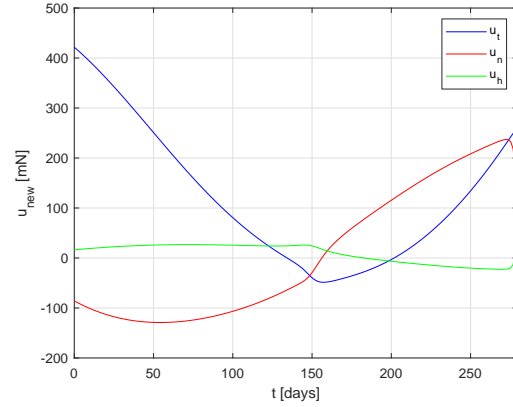


Figure 6.2: Optimal control components associated to the perturbed initial conditions.

$$\mathbf{u}_{new} = \mathbf{u}_{opt} + \boldsymbol{\alpha}_{opt} + \boldsymbol{\beta}_{opt}\delta\mathbf{x} \quad (6.16)$$

where $\delta\mathbf{x}$ is the difference between the satellite state computed using the optimal control law and the one evaluated using the new perturbed control action. The result of the analysis is presented in Figures 6.1 and 6.2. In Figure 6.1 (left), the control components time history associated to the original optimal control problem is presented, while Figure 6.2 (right) shows the same variables associated to the optimal control problem where the disturbances in the initial conditions have been introduced. In Figure 6.3 the difference between the magnitude of the two different optimal control law is plotted and it can be seen that the introduction of a disturbance in the initial condition does not change the trend significantly apart from the last part. This behaviour can be explained looking at the magnitude of the disturbances introduced in the initial orbital elements. The eccentricity disturbance is of the same order of magnitude of the original one, and a variation of orbital plane equal to 1 degree has been introduced. The optimal control law coming from the DDP algorithm is a feedback policy depending on the state variation. If the same optimal optimal control law obtained in the original problem without disturbances in the initial conditions is used, the final value of the orbit inclination will be larger than the prescribed final one. This means that a change-of-plane manoeuvre is needed to reduce the orbital inclination. The solver predicts that the most convenient way to apply the change of plane is at the end of the transfer when the satellite is at the larger distance from the main attractor so that the transversal component of the orbital velocity is smaller. At the same time, the algorithm adjusts the control law to increase the semi-major axis before with respect to the original problem with no disturbances in the initial conditions. This is why the out-of-plane control components increases at the end of the transfer while the in plane components decrease causing an overall reduction of the control thrust. On the contrary, the rapid increase of the semi-major axis is the reason why the in plane tangential component is larger at the beginning of the transfer causing also an increase of the total control thrust.

The previous analysis is confirmed looking also at the variation of the orbital elements when the disturbances are introduced in the initial conditions. The orbit inclination is reduced at the end of the transfer (Figure 6.6) where the satellite is close to the apocenter of the orbit (Figure 6.9). It is interesting to check how the final orbital elements depart from the prescribed target

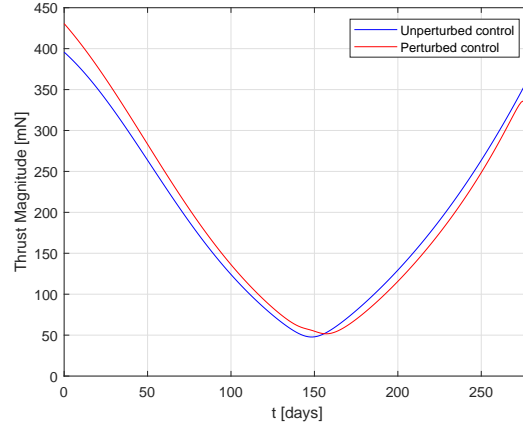


Figure 6.3: Comparison between the control magnitudes associated to the unperturbed and perturbed optimal control problem.

state whenever the initial condition is perturbed. The orbital elements variations are presented in Figures 6.4-6.9 where it is clear that the final state is still close to the prescribed target state but the time variation of the orbital elements is different.

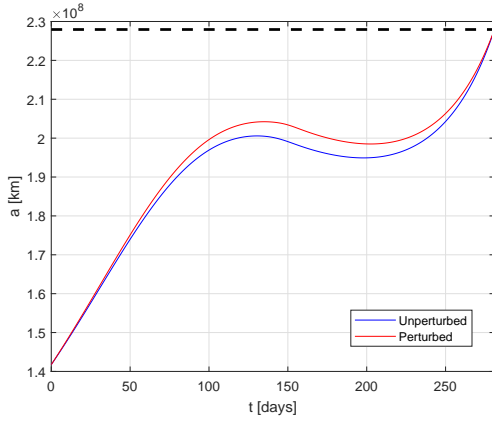


Figure 6.4: Semi-major axis unperturbed vs perturbed initial condition.

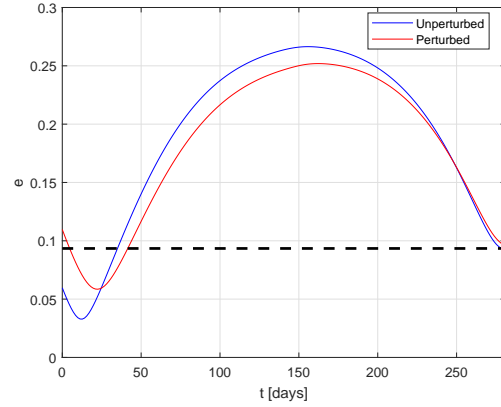


Figure 6.5: Eccentricity unperturbed vs perturbed initial condition.

A better insight of the robustness analysis can be provided listing the error between the final state and the target state if the initial condition is perturbed. Table 6.4 describes also what happens if the optimal control law derived in the unperturbed initial condition scenario is used also after the introduction of the disturbance.

It is clear that the optimal control law derived in the unperturbed scenario cannot be used without any modification because the errors generated at the end of the propagation are completely apart from the prescribed target state. On the other hand, the generation of the feedback control law provides a result that is consistent with the DDP assumptions defined in Chapter 2. Starting from a perturbation of the initial state the error obtained at the the final state is comparable with the disturbances defined in Table 6.3. Therefore, the outcome of the analysis is that the DDP algorithm can find an optimal solution that may or may not be robust according to

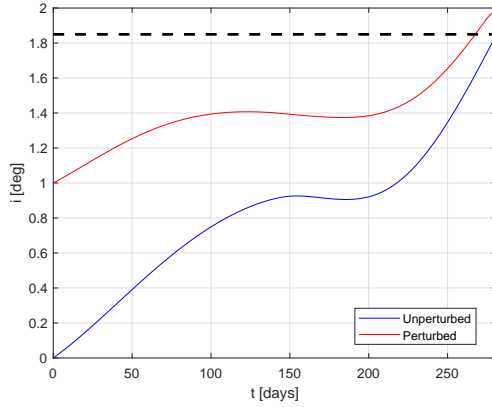


Figure 6.6: Inclination unperturbed vs perturbed initial condition.

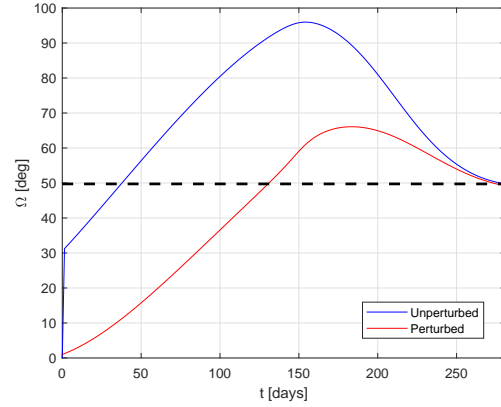


Figure 6.7: RAAN unperturbed vs perturbed initial condition.

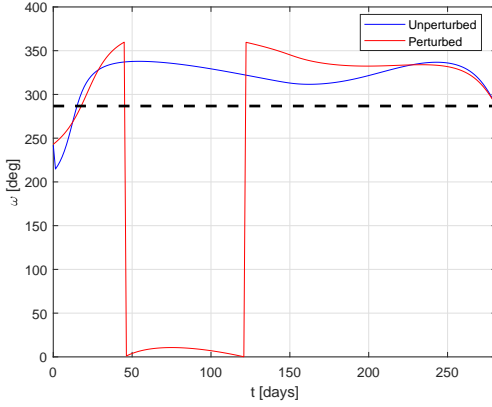


Figure 6.8: Pericenter anomaly unperturbed vs perturbed initial condition.

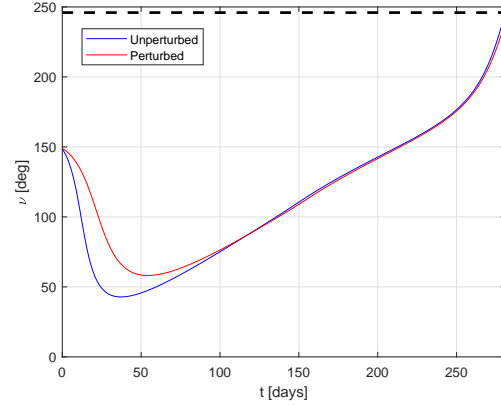


Figure 6.9: True anomaly unperturbed vs perturbed initial condition.

Table 6.4: Error between the prescribed target state and the actual state considering the different control laws.

Orbital element	Unperturbed	Perturbed	Perturbed with same control
Semi-major axis, δa_f	221.17675 km	18104.711 km	3.83194e8 km
Eccentricity, δe_f	6.08596e-7	7.76972e-5	0.01824
Inclination, δi_f	6.1448e-6°	8.9843e-4°	0.77°
RAAN, $\delta \Omega_f$	8.89168e-6°	0.73927°	17.0922°
Pericenter anomaly, $\delta \omega_f$	3.9327e-4°	0.41171°	7.0049°
True anomaly, $\delta \nu_f$	0.0015°	7.1921°	23.7599°

the magnitude of the perturbations introduced in the initial state. If the disturbances affecting the initial state are relatively large, the error with respect to the target orbit will behave as well. However, the analysis shows how the DDP optimisation technique is able to correct an optimal solution to account for changes in the initial state thanks to the feedback control law described in Eq. (6.16) provided as output.

6.3 Summary

In this chapter the comparison between the analytical formulations used in this research and the STMs as computational technique for the evaluation of the cost functional partial derivatives has been assessed. The main outcome of the analysis explains how the analytical formulations improves the computational speed of the algorithm because the number of function evaluations is always smaller with respect to the numerical integration of a system of partial differential equations.

The second part of the chapter presents a robustness analysis which is a peculiar characteristics of every kind of DDP algorithm, but it is improved when orbital elements are used as state formulation of the dynamics because the state variation of the orbital elements are smaller with respect to the Cartesian coordinates counterpart.

CHAPTER 7

Path constraints inclusion: eclipses

This chapter proposes a new method for including the eclipse as an internal path constraints inside the DDP algorithm. The eclipse requirement is formulated as maximum time for a spacecraft to be in the umbra shadow. Such requirement is very important during the preliminary design of a space mission because some subsystems such as the power and/or telecommunication subsystems can impose restrictions on the main trajectory of the spacecraft. For example, the batteries onboard of spacecraft represent an hardware with a limited power that is imposing the maximum energy that can be provided whenever solar power is not available such as in an eclipse. The work presented in this chapter was developed during a visiting research period at NASA Jet Propulsion Laboratory (JPL) under the supervision of Dr. Gregory Lantoine. The work related to the eclipse has been presented at the 2022 AAS/AIAA Astrodynamics Specialist Conference [75] and has been submitted to Acta Astronautica for peer-review [76]. The research related to the eclipse internal path constraint for the DDP algorithm has been presented at the 73rd IAC [77] and will be the main subject of a future journal publication.

7.1 Eclipse algorithm for the analytical derivation of entry/exit umbra points

In this section all the basic mathematical transformations that will be used for the derivation of the final equation are described. The full derivations consists in considering the generic Cartesian equation of a conical surface defined in its own reference frame that must be intersected with the Cartesian equation of the elliptical orbit defined in the perifocal reference frame. This operation requires the two geometrical figures to be defined in the same reference frame so that a rotation and a translation are required.

7.1.1 Rotation and translation

The first mathematical transformation that is used for the derivation of the model is the rotation and translation of the Cartesian equation of the cone from its reference frame x, y, z to another generic reference system $\tilde{x}, \tilde{y}, \tilde{z}$. The generic Cartesian of a cone in a generic reference

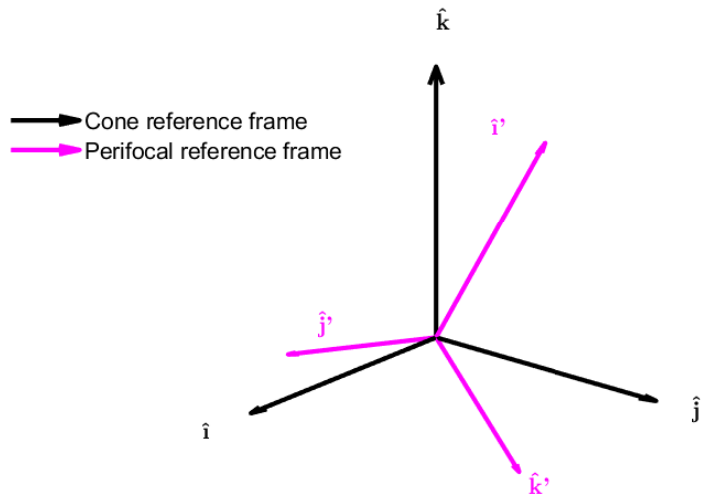
frame x, y, z is defined as follows:

$$\frac{x^2}{a_{cone}^2} + \frac{y^2}{b_{cone}^2} - \frac{z^2}{c_{cone}^2} = 0 \quad (7.1)$$

where a_{cone}, b_{cone} represent the semi-major and semi-minor axis of the ellipse obtained cutting the indefinite conical surface with the plane $z = c_{cone}$. This equation defines a right cone having the vertex in the origin of the reference frame and the circular base obtained cutting the cone with a plane normal to the z axis is an ellipse of semi-major axis and semi-minor axis equal to a_{cone} and b_{cone} , respectively. The first task is to express Eq. (7.1) in another reference system x', y', z' whose origin is coincident with x, y, z reference frame and parallel to the final $\tilde{x}, \tilde{y}, \tilde{z}$ reference frame. Such transformation is a pure rotation that can be described mathematically by using a the rotation matrix \mathbf{R} which aligns the reference frame x', y', z' to the reference system x, y, z :

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{Bmatrix} x' \\ y' \\ z' \end{Bmatrix} \quad (7.2)$$

The rotation matrix is obtained projecting the unit vectors $\hat{i}', \hat{j}', \hat{k}'$ aligned with the x', y', z' reference frame onto the unit vectors $\hat{i}, \hat{j}, \hat{k}$ aligned with the x, y, z reference system. An illustration of the rotation transformation is shown in Figure 7.1.

$$\mathbf{R}_{rot} = \begin{bmatrix} i_x & i_y & i_z \\ j_x & j_y & j_z \\ k_x & k_y & k_z \end{bmatrix} \quad (7.3)$$


The diagram shows a 3D coordinate system with two sets of unit vectors. The 'Cone reference frame' is shown with black arrows labeled \hat{i} , \hat{j} , and \hat{k} . The 'Perifocal reference frame' is shown with magenta arrows labeled \hat{i}' , \hat{j}' , and \hat{k}' . The \hat{k} axis is vertical, while \hat{i} and \hat{j} are in the horizontal plane. The \hat{i}' and \hat{j}' vectors are rotated relative to \hat{i} and \hat{j} , and \hat{k}' is rotated relative to \hat{k} . A legend indicates that black arrows represent the 'Cone reference frame' and magenta arrows represent the 'Perifocal reference frame'.

Figure 7.1: General rotation problem from one reference frame to another one.

Developing the matricial operations in Eq. (7.1), the transformation equations to obtain the rotation are obtained:

$$\begin{cases} x = R_{11}x' + R_{12}y' + R_{13}z' \\ y = R_{21}x' + R_{22}y' + R_{23}z' \\ z = R_{31}x' + R_{32}y' + R_{33}z' \end{cases} \quad (7.4)$$

It is possible to replace Eq. (7.4) in place of the variables x, y, z in Eq. (7.1) to obtain the conical surface in terms of the x', y', z' reference frame. After some mathematical manipulation the following expression is derived to represent the Cartesian equation of a conical frame defined in a different reference frame described by the rotation matrix in Eq. (7.3):

$$\begin{aligned} & (b_{cone}^2 c_{cone}^2 R_{11}^2 + a_{cone}^2 c_{cone}^2 R_{21}^2 - a_{cone}^2 b_{cone}^2 R_{31}^2)x'^2 + (b_{cone}^2 c_{cone}^2 R_{12}^2 + a_{cone}^2 c_{cone}^2 R_{22}^2 + \\ & - a_{cone}^2 b_{cone}^2 R_{32}^2)y'^2 + (b_{cone}^2 c_{cone}^2 R_{13}^2 + a_{cone}^2 c_{cone}^2 R_{23}^2 - a_{cone}^2 b_{cone}^2 R_{33}^2)z'^2 + \\ & + 2(b_{cone}^2 c_{cone}^2 R_{11}R_{12} + a_{cone}^2 c_{cone}^2 R_{21}R_{22} - a_{cone}^2 b_{cone}^2 R_{31}R_{32})x'y' + \\ & + 2(b_{cone}^2 c_{cone}^2 R_{11}R_{13} + a_{cone}^2 c_{cone}^2 R_{21}R_{23} - a_{cone}^2 b_{cone}^2 R_{31}R_{33})x'z' + \\ & + 2(b_{cone}^2 c_{cone}^2 R_{12}R_{13} + a_{cone}^2 c_{cone}^2 R_{22}R_{23} - a_{cone}^2 b_{cone}^2 R_{32}R_{33})y'z' = 0 \end{aligned} \quad (7.5)$$

Eq. (7.5) can be rewritten in an easier way introducing the following coefficients which are constant numbers once the rotation matrix and cone geometrical parameters are defined:

$$\begin{aligned} & \tilde{A}x'^2 + \tilde{B}y'^2 + \tilde{C}z'^2 + 2\tilde{D}x'y' + 2\tilde{E}x'z' + 2\tilde{F}y'z' = 0 \\ & \tilde{A} = b_{cone}^2 c_{cone}^2 R_{11}^2 + a_{cone}^2 c_{cone}^2 R_{21}^2 - a_{cone}^2 b_{cone}^2 R_{31}^2 \\ & \tilde{B} = b_{cone}^2 c_{cone}^2 R_{12}^2 + a_{cone}^2 c_{cone}^2 R_{22}^2 - a_{cone}^2 b_{cone}^2 R_{32}^2 \\ & \tilde{C} = b_{cone}^2 c_{cone}^2 R_{13}^2 + a_{cone}^2 c_{cone}^2 R_{23}^2 - a_{cone}^2 b_{cone}^2 R_{33}^2 \\ & \tilde{D} = b_{cone}^2 c_{cone}^2 R_{11}R_{12} + a_{cone}^2 c_{cone}^2 R_{21}R_{22} - a_{cone}^2 b_{cone}^2 R_{31}R_{32} \\ & \tilde{E} = b_{cone}^2 c_{cone}^2 R_{11}R_{13} + a_{cone}^2 c_{cone}^2 R_{21}R_{23} - a_{cone}^2 b_{cone}^2 R_{31}R_{33} \\ & \tilde{F} = b_{cone}^2 c_{cone}^2 R_{12}R_{13} + a_{cone}^2 c_{cone}^2 R_{22}R_{23} - a_{cone}^2 b_{cone}^2 R_{32}R_{33} \end{aligned} \quad (7.6)$$

After the rotation, Eq. 7.1 is expressed in the reference frame x', y', z' which is parallel to final one $\tilde{x}, \tilde{y}, \tilde{z}$, but with a different origin. It is necessary to perform a translation transformation to align the origin of the two reference systems. The translation equations can be derived considering a general position vector defined in the final shifted reference frame, $\tilde{\mathbf{r}}$, and the same position vector defined in original frame, \mathbf{r}' . The transformation requires the position vector of the shifted reference frame with respect to the original one, $\tilde{\mathbf{r}}_v$. In the general problem, the position vector of the vertex of the cone is defined with respect to the inertial reference system. Therefore, it is better to express the translation equations considering the position vector of the origin of the x', y', z' with respect to the shifted one.

$$\tilde{\mathbf{r}} = \tilde{\mathbf{r}}_v + \mathbf{r}' \quad (7.7)$$

A better representation of the translation transformation is shown in Figure 7.2. Extending the vectorial equation and taking into account that the final aim is to express everything in the $\tilde{x}, \tilde{y}, \tilde{z}$ reference frame, the following relations are obtained:

$$\begin{cases} x' = \tilde{x} - \tilde{x}_v \\ y' = \tilde{y} - \tilde{y}_v \\ z' = \tilde{z} - \tilde{z}_v \end{cases} \quad (7.8)$$

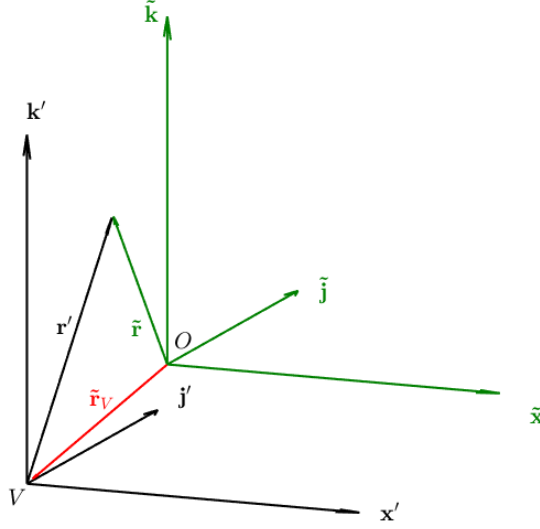


Figure 7.2: General translation problem from one reference frame to another one.

It is now possible to get the final expression of the generic conical surface defined in Eq. (7.1) in a new different reference frame replacing the formulas defined in Eq. (7.8) inside Eq. (7.6):

$$\begin{aligned} & \tilde{A}\tilde{x}^2 + \tilde{B}\tilde{y}^2 + \tilde{C}\tilde{z}^2 + 2\tilde{D}\tilde{x}\tilde{y} + 2\tilde{E}\tilde{x}\tilde{z} + 2\tilde{F}\tilde{y}\tilde{z} - 2(\tilde{A}\tilde{x}_v + \tilde{D}\tilde{y}_v + \tilde{E}\tilde{z}_v)\tilde{x} \\ & - 2(\tilde{D}\tilde{x}_v + \tilde{B}\tilde{y}_v + \tilde{F}\tilde{z}_v)\tilde{y} - 2(\tilde{E}\tilde{x}_v + \tilde{F}\tilde{y}_v + \tilde{C}\tilde{z}_v)\tilde{z} + (\tilde{A}\tilde{x}_v^2 + \tilde{B}\tilde{y}_v^2 + \tilde{C}\tilde{z}_v^2 \\ & + 2\tilde{D}\tilde{x}_v\tilde{y}_v + 2\tilde{E}\tilde{x}_v\tilde{z}_v + 2\tilde{F}\tilde{y}_v\tilde{z}_v) = 0 \end{aligned} \quad (7.9)$$

Again, it is possible to introduce a new variable set which are constant coefficients once the position vector of the conical surface vertex is known. This way Eq. (7.9) can be rewritten as follows:

$$\begin{aligned} & \tilde{A}\tilde{x}^2 + \tilde{B}\tilde{y}^2 + \tilde{C}\tilde{z}^2 + 2\tilde{D}\tilde{x}\tilde{y} + 2\tilde{E}\tilde{x}\tilde{z} + 2\tilde{F}\tilde{y}\tilde{z} + 2\tilde{H}\tilde{x} + 2\tilde{I}\tilde{y} + 2\tilde{J}\tilde{z} + \tilde{G} = 0 \\ & \tilde{G} = \tilde{A}\tilde{x}_v^2 + \tilde{B}\tilde{y}_v^2 + \tilde{C}\tilde{z}_v^2 + 2\tilde{D}\tilde{x}_v\tilde{y}_v + 2\tilde{E}\tilde{x}_v\tilde{z}_v + 2\tilde{F}\tilde{y}_v\tilde{z}_v \\ & \tilde{H} = -(\tilde{A}\tilde{x}_v + \tilde{D}\tilde{y}_v + \tilde{E}\tilde{z}_v) \\ & \tilde{I} = -(\tilde{D}\tilde{x}_v + \tilde{B}\tilde{y}_v + \tilde{F}\tilde{z}_v) \\ & \tilde{J} = -(\tilde{E}\tilde{x}_v + \tilde{F}\tilde{y}_v + \tilde{C}\tilde{z}_v) \end{aligned} \quad (7.10)$$

The expression obtained in Eq. (7.10) represents the analytical expression of a cylindrical conical surface characterised by a specific semi-major axis, semi-minor axis and height defined in a generic roto-translated reference system.

7.1.2 Intersection of the conical surface with the spacecraft orbit

The next topic to be discussed regards the intersection of the mathematical expression of the generic conical surface representing the eclipse shadow with an elliptical orbit. The intersection of the conical surfaces with the elliptical orbit requires that both the two mathematical expressions are defined with respect to the same reference frame. In the previous paragraph Eq. (??) represents the Cartesian equation of a cone that can be defined in whatever reference frame once the rotation matrix and position vector of the cone vertex are defined. The mathematical expression of an elliptical orbit can be retrieved starting from the canonic equation of an ellipse.

$$\frac{\bar{x}^2}{\tilde{a}_{orb}^2} + \frac{\bar{y}^2}{\tilde{b}_{orb}^2} = 1 \quad (7.11)$$

where \tilde{a}_{orb} and \tilde{b}_{orb} are the semi-major axis and semi-minor axis of the ellipse, respectively. The canonic equation of the ellipse is defined with respect a reference frame centered in the center of the ellipse. Therefore, Eq. (7.11) must to be modified using a translation transformation because the mathematical expression of an orbit requires the Earth or the celestial body located in one of the two foci to be the origin of the reference frame. The analytical expression of the elliptical orbit can be obtained applying a simple translation of the origin of the reference frame knowing that the distance between the the ellipse centre and the celestial body is equal to the focal distance, \tilde{c}_{orb} :

$$\tilde{\mathbf{c}}_{orb} = [-\sqrt{\tilde{a}_{orb}^2 - \tilde{b}_{orb}^2}, 0, 0] \quad (7.12)$$

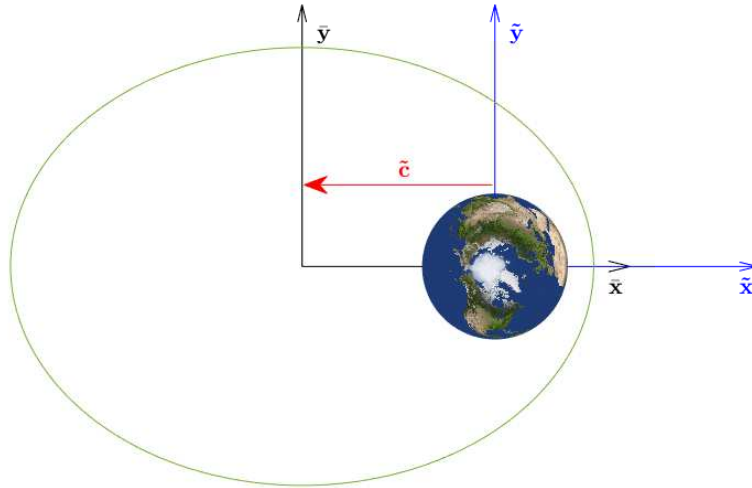


Figure 7.3: Translation from ellipse centred to focal reference frame.

Looking at Figure 7.3, the translation relations between the two reference systems are ob-

tained considering:

$$\begin{cases} \bar{x} = \tilde{x} - \tilde{c}_{orb} \\ \bar{y} = \tilde{y} \\ \bar{z} = \tilde{z} \end{cases} \quad (7.13)$$

The analytical equation of the orbit is derived replacing Eq. (7.13) in Eq. (7.11):

$$\tilde{b}_{orb}^2(\tilde{x}^2 + \tilde{c}_{orb}^2 - 2\tilde{x}\tilde{c}_{orb}) + \tilde{a}_{orb}^2\tilde{y}^2 = \tilde{a}_{orb}^2\tilde{b}_{orb}^2 \quad (7.14)$$

The new reference frame obtained is centered in the celestial body and has the x-axis aligned with the apse lines direction and the z-axis normal to the ellipse plane. Therefore, Eq. (7.14) represents the analytical expression of the elliptical orbit defined in the perifocal reference system usually denoted as \hat{e} , \hat{p} , \hat{h} . Therefore, it is convenient to choose to express also the conical surface in the perifocal reference frame so that the intersection between the two curves is possible. First of all, the ellipse is defined in the plane $\tilde{x}\tilde{y}$. This means that only the portion of the conical surface defined in that plane will eventually intersect the ellipse orbit. The expression of the curve associated to the conical surface in the orbital plane is obtained by imposing in Eq. (7.10) that $\tilde{z} = 0$.

$$\tilde{A}\tilde{x}^2 + \tilde{B}\tilde{y}^2 + 2\tilde{D}\tilde{x}\tilde{y} + 2\tilde{H}\tilde{x} + 2\tilde{I}\tilde{y} + \tilde{G} = 0 \quad (7.15)$$

At this point, the intersection with the elliptical orbit is carried out expressing in an explicit way the variable \tilde{y} in Eq. (7.14):

$$\tilde{y} = \pm \frac{\tilde{b}_{orb}}{\tilde{a}_{orb}} \sqrt{\tilde{b}_{orb}^2 - \tilde{x}^2 + 2\tilde{x}\tilde{c}_{orb}} \quad (7.16)$$

Replacing Eq. (7.16) in Eq. (7.15) leads to the following expression.

$$\begin{aligned} \tilde{A}\tilde{x}^2 + \tilde{B} \frac{\tilde{b}_{orb}}{\tilde{a}_{orb}} (\tilde{b}_{orb}^2 - \tilde{x}^2 + 2\tilde{x}\tilde{c}_{orb}) \pm 2\tilde{D}\tilde{x} \frac{\tilde{b}_{orb}}{\tilde{a}_{orb}} \sqrt{\tilde{b}_{orb}^2 - \tilde{x}^2 + 2\tilde{x}\tilde{c}_{orb}} + \\ + 2\tilde{H}\tilde{x} \pm 2\tilde{I} \frac{\tilde{b}_{orb}}{\tilde{a}_{orb}} \sqrt{\tilde{b}_{orb}^2 - \tilde{x}^2 + 2\tilde{x}\tilde{c}_{orb}} + \tilde{G} = 0 \end{aligned} \quad (7.17)$$

It is convenient to move all the terms containing the square root at the right-hand side so that it is possible to put the square root as common factor.

$$\begin{aligned} \bar{A}\tilde{x}^2 + \bar{B}\tilde{x} + \bar{C} &= \mp 2 \frac{\tilde{b}_{orb}}{\tilde{a}_{orb}} \sqrt{\tilde{b}_{orb}^2 - \tilde{x}^2 + 2\tilde{x}\tilde{c}_{orb}} (\tilde{D}\tilde{x} + \tilde{I}) \\ \bar{A} &= \left(\tilde{A} - \tilde{B} \frac{\tilde{b}_{orb}^2}{\tilde{a}_{orb}^2} \right) \\ \bar{B} &= 2 \left(\tilde{B} \frac{\tilde{b}_{orb}^2}{\tilde{a}_{orb}^2} \tilde{c}_{orb} + \tilde{H} \right) \\ \bar{C} &= \left(\tilde{G} + \tilde{B} \frac{\tilde{b}_{orb}^2}{\tilde{a}_{orb}^2} \right) \end{aligned} \quad (7.18)$$

In order to solve for the variable x which represents the unknown of the problem and corresponds to the abscissa of the entry/exit point in the perifocal reference frame, both sides of Eq. (7.18) are squared. In this way, the ambiguity given by the two halves of the ellipse is removed. The final equation will be a quartic equation in the unknown x .

$$\begin{aligned}
 C_1 \tilde{x}^4 + C_2 \tilde{x}^3 + C_3 \tilde{x}^2 + C_4 \tilde{x} + C_5 &= 0 \\
 C_1 &= \bar{A}^2 + 4\tilde{D}^2 \frac{\tilde{b}_{orb}^2}{\tilde{a}_{orb}^2} \\
 C_2 &= 2 \left(\bar{A}\bar{B} + 4\tilde{c}_{orb}\tilde{D}^2 \frac{\tilde{b}_{orb}^2}{\tilde{a}_{orb}^2} + 4\tilde{D}\tilde{I} \frac{\tilde{b}_{orb}^2}{\tilde{a}_{orb}^2} \right) \\
 C_3 &= \bar{B}^2 + 2\bar{A}\bar{C} - 4\tilde{D}^2 \frac{\tilde{b}_{orb}^4}{\tilde{a}_{orb}^2} + 4\tilde{I}^2 \frac{\tilde{b}_{orb}^2}{\tilde{a}_{orb}^2} + 16\tilde{c}_{orb}\tilde{D}\tilde{I} \frac{\tilde{b}_{orb}^2}{\tilde{a}_{orb}^2} \\
 C_4 &= 2\bar{B}\bar{C} - 8\tilde{D}\tilde{I} \frac{\tilde{b}_{orb}^4}{\tilde{a}_{orb}^2} + 8\tilde{c}_{ell}\tilde{I}^2 \frac{\tilde{b}_{ell}^2}{\tilde{a}_{ell}^2} \\
 C_5 &= \bar{C}^2 - 4\tilde{I}^2 \frac{\tilde{b}_{orb}^4}{\tilde{a}_{orb}^2}
 \end{aligned} \tag{7.19}$$

7.1.3 Solutions decision making

In the previous paragraph Eq. (7.19) has been derived to compute the abscissa of the entry/exit points generated by a conical shadow. This quartic equation can be solved analytically using Cardan's procedure [60]. Three possible scenarios can occur:

- 4 real solutions
- 2 real solutions and 2 complex solutions
- no real solutions and 4 complex solutions

In case of zero real solutions, the conical shadow is not intersecting the elliptical orbit, and so the satellite cannot be in eclipse. In case of two or four real solutions, there are four and eight possible points that are candidates for umbra and penumbra entry/exit points. Indeed, each real abscissa obtained from Eq. (7.19) can correspond only to a point on the upper-half or lower-half of the elliptical orbit. The ambiguity can be cancelled computing for each abscissa the point of the upper-half or lower-half of the ellipse using Eq. (7.16) and replacing the points in Eq. (7.15). If the results is not exactly zero, but a small value under a certain threshold the point belongs to the conical surface and it represents a real candidate for being an entry/exit point for the eclipse. If the condition is not satisfied the point will not lie on the conical surface and it will be discarded. This represents a first criterion that should be applied to filter for the wrong points and select the real candidates. A second ambiguity is generated because the conical surface can intersect the orbit both in the shadow and sunlight side. This time the ambiguity can be solved considering that the shadow must always be on the opposite side of the Sun. This

physical condition can be exploited to write a mathematical constraint for the selection of the real entry/exit eclipse points.

$$\mathbf{r}_{\oplus-\odot} \cdot \mathbf{r}_P < 0 \quad (7.20)$$

with $\mathbf{r}_{\oplus-\odot}$ representing the position vector of the Sun with respect to the Earth considered as origin of the Geocentric Equatorial Reference frame and \mathbf{r}_P the generic position vector connecting the Earth with the candidate entry/exit points on the elliptical orbit. The entry/exit points will lie in the opposite plane with respect to the Sun so that the dot product of the two vectors should be negative. The last step for the derivation of the final solution is to decide which of the two points represent the entry and exit. The simplest way is to convert the two points into anomalies which represent the true anomalies and verifying which is the entrance and exit anomaly.

$$\nu_P = \arccos \left(\frac{x_P}{r_P} \right) \quad (7.21)$$

A picture of the ambiguities generated by the intersection of the conical surface with the elliptical orbit is presented in Figure 7.4. The circular markers represent the real intersections of the conical surface with the elliptical orbit, while the triangular markers are the fake solutions coming out from the quadratic Eq. (7.16). The four solutions to the quartic Eq. (7.19) are the abscissas of the four real intersections that generate a real ordinate and a fake ordinate when the quadratic equation of the the ellipse is solved.

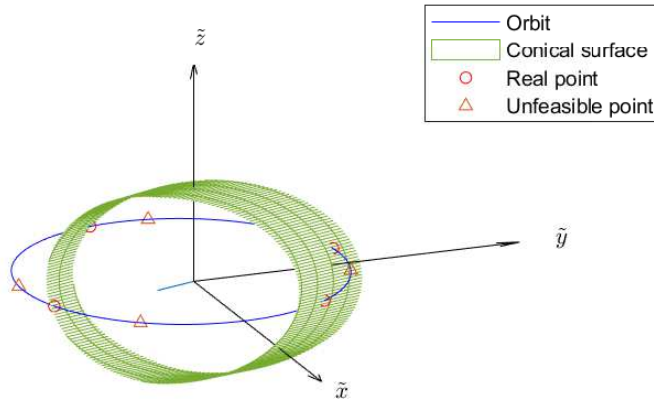


Figure 7.4: Intersection of the conical surface with the elliptical orbit.

7.2 Orbital scenario

In the previous section, the analytical procedure devised by the author to derive a quartic equation resulting in the intersections of a conical shadow with an elliptical orbit has been carried out. This section points out how to compute all the coefficients needed to solve for the intersection points.

7.2.1 Intersection eclipse with the oblate Earth

The input required for the algorithm are three:

- The position vector of the Sun, \mathbf{r}_\odot
- Spacecraft Keplerian elements
- The geometrical parameters of the occulting body.

The first step is to define the ellipse representing the base of the conical shadow. This cone is delimited by the Earth surface so that the elliptical base is given by the intersection of the plane normal to the cone axis and the oblate ellipsoid associated to the Earth surface. It is natural to assume that the axis of the conical shadow is parallel to the position vector of the Sun, \mathbf{r}_\odot . The Cartesian equation of the plane normal to this direction is:

$$n_x x + n_y y + n_z z = 0 \quad (7.22)$$

where n_x, n_y, n_z represent the components of the unit vector associated to \mathbf{r}_\odot . The second equation that is required is the one modelling the Earth surface as an oblate ellipsoid of rotation. This equation requires the geometrical parameters of the occulting body to be known.

$$\frac{x^2 + y^2}{R_{eq}^2} + \frac{z^2}{R_{pol}^2} = 1 \quad (7.23)$$

with R_{eq} and R_{pol} defining the equatorial and polar radii, respectively. The semi-major axis and semi-minor axis of the ellipse resulting from the intersection of the plane normal to the conical axis and the oblate ellipsoid can be computed as follows [88]:

$$\begin{aligned} a_{ell} &= R_{eq} \sqrt{1 - \frac{d^2}{R_{eq}^2(1 - E_\oplus^2 n_z^2)}} \\ b_{ell} &= R_{pol} \frac{\sqrt{1 - d^2/R_{eq}^2 - E_\oplus^2 n_z^2}}{1 - E_\oplus^2 n_z^2} \end{aligned} \quad (7.24)$$

where d is the distance between the plane and the origin of the reference system and it is equal to zero because this point belongs to the axis of the conical shadow, and E_\oplus is the eccentricity of the occulting body. Eq. (7.24) provides the values of a and b to be used for the definition of the Cartesian equation of the conical surface described by Eq. (7.1).

7.2.2 Characterisation of cone vertex and rotation matrix

The last geometrical parameter that is required to define the conical surface is the cone height that is assumed to be coincident with the position vector of the cone vertex connecting the Earth center with the cone vertex, \mathbf{r}_v . The following assumption is considered to determine the vertex of the cone:

The ellipse resulting from the intersection of the plane normal to the conical axis and passing through the Sun center can be obtained from the ellipse representing the conical shadow basis delimited by the occulting body through an homothetic transformation.

Even if this assumption is true just in specific cases, it can be numerically proved that this assumption is good and is not affecting the accuracy of the results provided by the algorithm. Using the previous assumption, it is possible to compute the points where the sun rays generating the conical surface depart from the intersection ellipse between the Sun surface and the plane normal to the conical axis.

$$\begin{aligned}\mathbf{r}_{up\odot} &= \mathbf{r}_{\odot} + R_{\odot}\hat{\mathbf{l}} \\ \mathbf{r}_{down\odot} &= \mathbf{r}_{\odot} - R_{\odot}\hat{\mathbf{l}}\end{aligned}\tag{7.25}$$

where $\mathbf{r}_{up\odot}$ and $\mathbf{r}_{down\odot}$ are the position vectors of the leading edge of the upward and downward sun rays starting from the Sun surface, respectively. Other two points on the Earth surface are required to define the equations of the sun rays generating the conical shadow. These points can be defined similarly using the following equations:

$$\begin{aligned}\mathbf{r}_{up\oplus} &= R_{\oplus}\hat{\mathbf{l}} \\ \mathbf{r}_{down\oplus} &= -R_{\oplus}\hat{\mathbf{l}}\end{aligned}\tag{7.26}$$

where $\hat{\mathbf{l}}$ identifies the apse line direction associated to the ellipse obtained from the intersection of the oblate ellipsoid and the plane normal to conical shadow and can be derived as follows:

$$\hat{\mathbf{l}} = \frac{1}{\sqrt{n_x^2 + n_y^2}}[-n_y, n_x, 0]\tag{7.27}$$

The vertex of the cone is computed considering the intersection of the two sun rays connecting the apsidal points of the two ellipses defined on the Sun surface and Earth surface.

$$\begin{aligned}\mathbf{m}_{up} &= \mathbf{r}_{up\odot} - \mathbf{r}_{up\oplus} \\ \mathbf{m}_{down} &= \mathbf{r}_{down\odot} - \mathbf{r}_{down\oplus} \\ t_{int} &= \frac{x_{down\oplus} - x_{up\oplus}}{m_{x_{up}} - m_{x_{down}}} \\ \mathbf{r}_v &= \mathbf{r}_{up\oplus} + t_{int}\mathbf{m}_{up}\end{aligned}\tag{7.28}$$

The magnitude of the vertex position vector, \mathbf{r}_v is the last geometrical quantity needed to define the Cartesian equation of the conical surface, c . A sketch of the different variables mentioned so far is shown in Figure 7.5.

$$c = \|\mathbf{r}_v\|\tag{7.29}$$

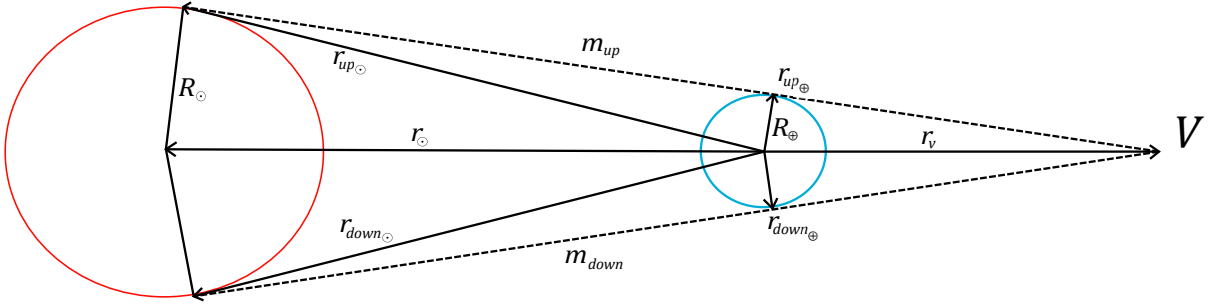


Figure 7.5: Graphical representation of the variables used to define the shadow cone.

There is not a unique way to define the vertex of the conical shadow depending on the way the couple of points delimiting the sun rays are defined. An alternative approach to compute the position vector of the cone vertex is to consider the sun rays as the tangent lines to the ellipse obtained from the intersection of the occulting body and the conical axis passing through the points on the Sun surface defined in Eq. (7.25). The equations to derive the geometrical parameters of the tangent lines are described hereafter. The first step is to intersect the general equation of a line with the analytical equation of the ellipse considered.

$$\begin{cases} y - y_p = m(x - x_p) \\ \frac{x^2}{a_{ell}^2} + \frac{y^2}{b_{ell}^2} = 1 \end{cases} \quad (7.30)$$

where x_p and y_p are the coordinates of the position vectors defined by Eq. (7.25) rotated in the intersection ellipse reference frame. Indeed, the canonical equation of the ellipse is valid only in a reference frame with the origin coincident with the center of the ellipse, the x -axis aligned with the apse line direction, the y -axis aligned with the ellipse semi-minor axis and the z -axis aligned with the axis of the cone. The rotation matrix moving from the geocentric equatorial inertial frame to the ellipse local reference frame can be obtained considering the expressions of the unit vectors defined in the inertial reference frame.

$$\begin{aligned} \hat{\mathbf{e}} &= \frac{1}{\sqrt{n_x^2 + n_y^2}} [-n_y, n_x, 0] \\ \hat{\mathbf{p}} &= \frac{1}{\sqrt{n_x^2 + n_y^2}} [n_x n_z, n_y n_z, -(n_x^2 + n_y^2)] \\ \hat{\mathbf{n}} &= \frac{\mathbf{r}_\odot}{|\mathbf{r}_\odot|} \end{aligned} \quad (7.31)$$

The rotation matrix moving from the geocentric inertial reference frame to the local ellipse reference frame is obtained as follows:

$$\mathbf{R}_{in \rightarrow loc} = \begin{bmatrix} e_x & e_y & e_z \\ p_x & p_y & p_z \\ n_x & n_y & n_z \end{bmatrix} \quad (7.32)$$

Replacing the general expression of a line in the canonical equation of the ellipse defines a

second-order equation in the unknown x .

$$\begin{cases} y = mx + (y_p - mx_p) \\ (b_{ell}^2 + a_{ell}^2 m^2)x^2 - 2a_{ell}^2 m(mx_p - y_p)x + a_{ell}^2 [(mx_p - y_p)^2 - b_{ell}^2] = 0 \end{cases} \quad (7.33)$$

The second-order equation can admit only two equal solutions if the condition of tangency is applied. This means that the discriminant associated to the second-order equation is equal to 0. This condition defines a new second-order equation where the unknown is angular coefficient of the tangent lines, m .

$$a_{ell}^2 b_{ell}^2 (a_{ell}^2 - x_p^2) m^2 + 2a_{ell}^2 b_{ell}^2 x_p y_p m + a_{ell}^2 b_{ell}^2 (b_{ell}^2 - y_p^2) = 0 \quad (7.34)$$

The solution to Eq. (7.34) is:

$$m = \frac{-x_p y_p \pm \sqrt{a_{ell}^2 y_p^2 + b_{ell}^2 x_p^2 - a_{ell}^2 b_{ell}^2}}{a_{ell}^2 - x_p^2} \quad (7.35)$$

At this point it is possible to replace Eq. (7.35) in Eq. (7.34) to derive the coordinates of the tangent points.

$$x_t = \frac{a_{ell}^2 m (mx_p - y_p)}{b_{ell}^2 + a_{ell}^2 m^2} \quad y_t = mx_t + (y_p - mx_p) \quad (7.36)$$

The last step is to rotate from the local ellipse reference frame to the inertial reference frame the tangent points so that Eq. (7.28) can be applied to get the position vector of the cone vertex.

$$\mathbf{r}_{up\oplus} = \mathbf{R}_{in \rightarrow loc}^T \begin{Bmatrix} x_{up}^t \\ y_{up}^t \\ 0 \end{Bmatrix} \quad \mathbf{r}_{down\oplus} = \mathbf{R}_{in \rightarrow loc}^T \begin{Bmatrix} x_{down}^t \\ y_{down}^t \\ 0 \end{Bmatrix} \quad (7.37)$$

The different ways to compute the cone vertex position vertex arise from the assumptions which are modifying the real path and geometry of the sun rays which are not creating a perfect conical surface. However, a numerical analysis where different couple of points are used to generate the cone vertex shows that the maximum error in the definition of the magnitude of the cone vertex position vector is in the order of 1000 km corresponding to a relative error less than 1%. This small error justifies the assumption used for the derivation of the algorithm and why the results are not considerably affected by the selection of a specific way to compute the vertex of the cone.

The last variable to be defined to get all the coefficients needed to solve the quartic equation is the rotation matrix \mathbf{R} to move from the conical surface reference frame to the perifocal reference frame. This matrix can be obtained combining two different rotation matrices. Indeed, the conical surface reference frame is parallel to the local ellipse reference frame but with a different origin of the reference system. Rotation matrices are not accounting for translation so that the rotation matrix defined in Eq. (7.32) allows to move from the inertial frame to the local ellipse or conical surface reference frame. It is well known in orbital mechanics how to compute the rotation matrix to move from the geocentric equatorial system to the perifocal reference frame using the orbital elements of a specific orbit [78].

$$\mathbf{R}_{in \rightarrow per} = \begin{bmatrix} \cos \omega \cos \Omega - \sin \omega \cos i \sin \Omega & -\sin \omega \cos \Omega - \cos \omega \cos i \sin \Omega & \sin i \sin \Omega \\ \cos \omega \sin \Omega + \sin \omega \cos i \cos \Omega & -\sin \omega \sin \Omega + \cos \omega \cos i \cos \Omega & -\sin i \cos \Omega \\ \sin \omega \sin i & \cos \omega \sin i & \cos i \end{bmatrix} \quad (7.38)$$

with i , Ω , ω representing the orbit inclination, right ascension of the ascending node, and the pericenter anomaly, respectively. Combining the two rotation matrices it is possible to get the final rotation matrix which rotates the conical surface reference frame into the perifocal frame.

$$\mathbf{R} = \mathbf{R}_{in \rightarrow per}^T \mathbf{R}_{in \rightarrow loc} \quad (7.39)$$

In this way all the geometric quantities needed to define the coefficients of Eq. (7.6) are known and the analytical procedure to solve the quartic equation can be applied to solve for the x -coordinate of the umbra entry/exit points defined in the perifocal reference frame. The analytical procedure to solve a quartic equation is summarised in Appendix B.

7.2.3 Penumbra characterisation

In this section the derivation of the entry/exit penumbra points is carried out. All the analytical equations presented in the previous section for the umbra cone are still valid also for the determination of the penumbra region. This is true because those equations are just modelling the intersection of a generic conical surface and an oblate ellipsoid of rotation.

The two methods depart in the definition of the geometrical characteristics of the conical surface and the criterion to be used for the selection of the correct solutions coming from the quartic equation. In particular, the difference in the two approaches occurs after the definition of the couple of points on the Sun surface and occulting body surface delimiting the sun rays. The axis of the penumbra conical surface is aligned with the umbra one and also the ellipse at the base of the conical surface is the same. However, the umbra conical surface is built connecting the upper points on the Sun surface with the upper points on the Earth surface and the lower points on the Sun surface with the lower points on the Earth surface. This type of connection can be summarized in "up \rightarrow up" and "down \rightarrow down". A graphical representation to understand how to define the umbra and penumbra regions is presented in Figure 7.6 where the upper points on the Sun and Earth surfaces are identified by s_{up} and E_{up} , while the downward points on the same surfaces are s_{down} and E_{down} , respectively.

The penumbra conical surface is generating connecting the upper points on the Sun surface to the lower points on the Earth surface, and the lower points on the Sun surface are connected to the upper points on the Earth surface. This type of connection is "up \rightarrow down" and "down \rightarrow up". The new equations for the computation of the penumbra cone vertex are as follows:

$$\begin{aligned} \mathbf{m}_{p1} &= \mathbf{r}_{up\odot} - \mathbf{r}_{down\oplus} \\ \mathbf{m}_{p2} &= \mathbf{r}_{down\odot} - \mathbf{r}_{up\oplus} \\ t_{int}^p &= \frac{x_{down\oplus} - x_{up\oplus}}{m_{x_{up}} - m_{x_{down}}} \\ \mathbf{r}_v^p &= \mathbf{r}_{up\oplus} + t_{int}^p \mathbf{m}_{up} \end{aligned} \quad (7.40)$$

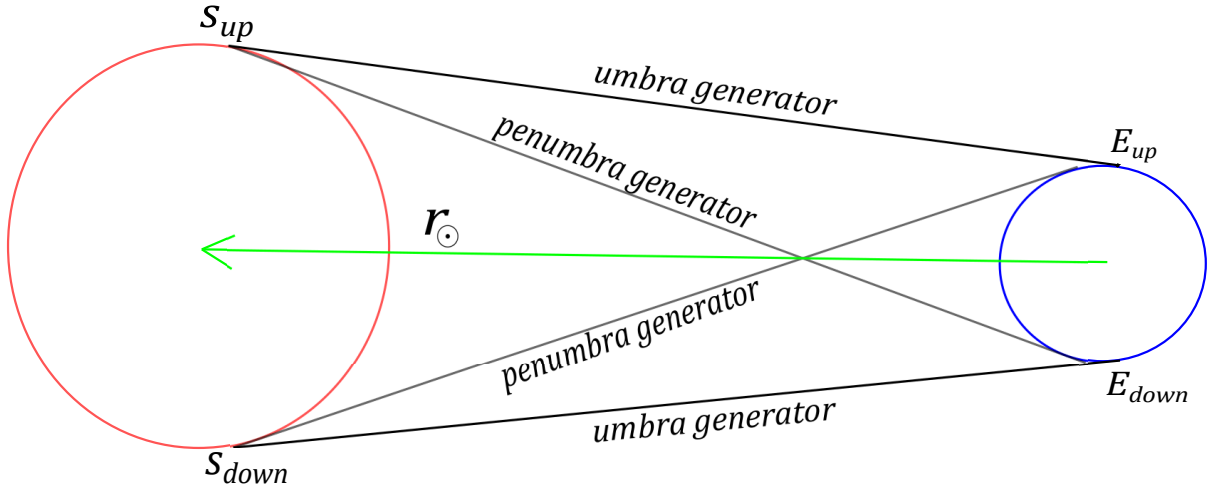


Figure 7.6: Graphical representation of umbra and penumbra regions.

where \mathbf{m}_{p_1} and \mathbf{m}_{p_2} are the angular coefficients of the two penumbra lines generator, t_{int}^p is the value of the parameter to identify the penumbra vertex considering the parametric representation of a Cartesian line in 3D space, and \mathbf{r}_v^p is the position vector of the penumbra cone vertex in the inertial frame.

In a similar way the second approach based on the computation of the tangent lines can be modified to get the penumbra cone vertex. For each point, Eq. (7.35) provides two different angular coefficients. One angular coefficient is related to the tangent line delimiting the umbra conical shadow and the second is associated to the penumbra conical surface. Therefore, after the identification of the correct angular coefficient it is possible to apply Eq. (7.40) to get the penumbra cone vertex.

It is obvious that also the criterion used for the selection of the exact solutions coming from the quartic equations should be modified if the penumbra conical surface is considered. The first criterion based on the selection of the points which really belong to the penumbra conical surface is unchanged. The second criterion that selects the exact points based on their positions with respect to the Sun position vector is the same if the anomalies are used but it is reversed if the distances with respect to the Sun centre are used.

7.3 Results

In this section the devised eclipse analytical algorithm is applied to different orbital scenarios for the determination of the entry/exit umbra and penumbra points together with the corresponding time spent in the umbra and penumbra regions. The same examples are solved also with state-of-art eclipse algorithms based on different assumptions to verify the correctness of the results.

Four different orbital scenarios are considered:

- Low-Earth slightly inclined orbit

- Low-Earth highly inclines orbit
- Geostationary orbit
- Highly elliptical orbit

All the simulations are carried out using a processor Intel® Core (TM) i7-9750H CPU @ 2.60 GHz. The orbital parameters assumed to identify the Sun position vector are summarized in Table 7.1:

Table 7.1: Sun state vector used for all the orbital scenarios.

r_{\odot}^x	148979647.6842771 km
r_{\odot}^y	5289205.7023694 km
r_{\odot}^z	-1142.3033745 km
v_{\odot}^x	-0.561916547046 km/s
v_{\odot}^y	29.87780687 km/s
v_{\odot}^z	-0.002623168305249 km/s

7.3.1 Low-Earth slightly inclined orbit

The first orbital scenario deals with a satellite in a slightly inclined low-Earth orbit. The orbital parameters used for the solution of the exercise are reported in Table 7.2.

Table 7.2: Spacecraft initial orbital elements for scenario 1.

Semi-major axis	8000 km
Eccentricity	0.2
Inclination	5°
RAAN	60°
Pericenter anomaly	30°
True Anomaly	140°

The entry and exit points for satellite umbra are computed using different methods. The first one is of course the new proposed analytical approach. The second method is the classic Escobal cylindrical shadow model for the umbra determination. The third method is using Fixler approach to define the same quantities. The fourth method is the iterative approach used to model the conical shadow taking into account for the Earth oblateness proposed always by Escobal. The results computed using the different methods are presented in Table 7.3.

The same considerations can be done for the entry and exit penumbra points. In this case the Escobal cylindrical model cannot be consider because the assumption of cylindrical shadow is not generating a penumbra region. The results presenting the different entry and exit anomalies for the penumbra region are shown in Table 7.4.

Table 7.3: Umbra entry and exit points for scenario number 1.

	ν_{entry}^u	ν_{exit}^u	Δt
New Approach	8.2289°	136.8815°	2254.77 s
Escobal Cylindrical	7.8676°	137.1116°	2265.32 s
Fixler	8.2122°	136.8832°	2255.02 s
Escobal Conical	8.2210°	136.8821°	2254.88 s

Table 7.4: Penumbra entry and exit points for scenario 1.

	ν_{entry}^p	ν_{exit}^p	Δt_{entry}	Δt_{exit}
New Approach	7.53°	137.34°	9.00 s	11.82 s
Fixler	7.51°	137.34°	8.97 s	11.78 s
Escobal Conical	7.52°	137.34°	9.01 s	11.83 s

7.3.2 Low-Earth highly inclined orbit

The second orbital scenario considers a satellite in a highly inclined low-Earth orbit. The orbital parameters used for the solution of the exercise are reported in Table 7.5.

Table 7.5: Satellite initial orbital elements for scenario 2.

Semi-major axis	8000 km
Eccentricity	0.2
Inclination	56°
RAAN	60°
Pericenter anomaly	30°
True Anomaly	140°

The entry and exit points for satellite umbra computed using different methods are presented in Table 7.6.

Table 7.6: Umbra entry and exit points for scenario number 2.

	ν_{entry}^u	ν_{exit}^u	Δt
New Approach	46.7973°	124.8715°	1438.04 s
Escobal Cylindrical	44.5680°	125.5206°	1485.38 s
Fixler	45.4431°	124.9756°	1459.93 s
Escobal Conical	46.7893°	124.8733°	1438.18 s

The same considerations can be done for the entry and exit penumbra points. The results presenting the different entry and exit anomalies for the penumbra region are shown in Table 7.7.

Table 7.7: Penumbra entry and exit points for scenario 2.

	ν_{entry}^p	ν_{exit}^p	Δt_{entry}	Δt_{exit}
New Approach	45.02°	125.98°	25.41 s	26.38 s
Fixler	43.86°	126.06°	25.08 s	26.03 s
Escobal Conical	45.01°	125.98°	25.38 s	26.38 s

7.3.3 Geosynchronous orbit

The third orbital scenario considers a satellite in a Geosynchronous orbit. The orbital parameters used for the solution of the exercise are reported in Table 7.8.

Table 7.8: Satellite initial orbital elements for scenario 3.

Semi-major axis	42164 km
Eccentricity	0
Inclination	0°
RAAN	60°
Pericenter anomaly	30°
True Anomaly	140°

The entry and exit points for satellite umbra computed using different methods are presented in Table 7.9.

Table 7.9: Umbra entry and exit points for scenario number 3.

	ν_{entry}^u	ν_{exit}^u	Δt
New Approach	83.5979°	100.4686°	4037.98 s
Escobal Cylindrical	83.3327°	100.7338°	4164.81 s
Fixler	83.5967°	100.4698°	4038.47 s
Escobal Conical	83.5978°	100.4687°	4037.94 s

The same considerations can be done for the entry and exit penumbra points. The results presenting the different entry and exit anomalies for the penumbra region are shown in Table 7.10.

Table 7.10: Penumbra entry and exit points for scenario 3.

	ν_{entry}^p	ν_{exit}^p	Δt_{entry}	Δt_{exit}
New Approach	83.06°	101°	127.95 s	127.95 s
Fixler	83.06°	101°	127.51 s	127.51 s
Escobal Conical	83.06°	101°	128.05 s	128.05 s

7.3.4 Highly elliptical orbit

The fourth orbital scenario considers a satellite in a highly elliptical orbit. The orbital parameters used for the solution of the exercise are reported in Table 7.11.

Table 7.11: Satellite initial orbital elements for scenario 4.

Semi-major axis	50000 km
Eccentricity	0.7
Inclination	10°
RAAN	20°
Pericenter anomaly	330°
True Anomaly	140°

The entry and exit points for satellite umbra computed using different methods are presented in Table 7.12.

Table 7.12: Umbra entry and exit points for scenario 4.

	ν_{entry}^u	ν_{exit}^u	Δt
New Approach	189.472°	195.437°	6696.34 s
Escobal Cylindrical	189.094°	195.829°	7556.60 s
Fixler	189.140°	195.668°	6723.60 s
Escobal Conical	189.471°	195.438°	6697.28 s

The same considerations can be done for the entry and exit penumbra points. The results presenting the different entry and exit anomalies for the penumbra region are shown in Table 7.13.

Table 7.13: Penumbra entry and exit points for scenario 4.

	ν_{entry}^p	ν_{exit}^p	Δt_{entry}	Δt_{exit}
New Approach	188.7°	196.1°	847.5 s	802.2 s
Fixler	188.4°	196.3°	797.2 s	842.3 s
Escobal Conical	188.7°	196.1°	846.2 s	802.0 s

7.3.5 Discussion

In this section the results obtained in Section V are discussed to compare the effectiveness of the proposed analytical method with respect to the state-of-the art algorithms. The first consideration that should be stressed looking at all the results is that the Escobal cylindrical shadow method represent the lowest accurate approach to determine the umbra entry and exit points. This result was expected due to the underlying assumption of cylindrical shadow that is very coarse. Therefore, the real comparison should be done with Fixler method and the Escobal conical shadow method. It is important to underline the assumption used by the two methods to

compute the umbra and penumbra regions exit/entry points. Fixler method assumes a conical shadow and a spherical Earth. It is based to solve a highly non-linear transcendental equation which represents a geometrical condition associated to the entry and exit points to/from the umbra and penumbra regions. Such non-linear equation requires an initial guess to identify the four possible roots of the equations which are not known a priori. Escobal conical model also is based on the numerical solution to a non-linear transcendental equation based on a specific shadow function and takes into account for the Earth oblateness using an iterative procedure. In this case the first guess solution needed for the solution to the non-linear equation is known because it comes from the cylindrical model. Thanks to the iterative procedure Escobal conical shadow model can be considered an accurate solution for each orbital scenario considered in the previous section. A summary of the characteristics of the different approaches is presented in Table 7.4.

Table 7.14: Summary of the main characteristics of the eclipse algorithms. Legenda: O = Oblate, S = Sphere, CO = Conical, CY = Cylinder, A = Analytical, N = Numerical

Method	Earth Model	Type of Solution	Shadow Model
New Algorithm	O	A	CO
Escobal Cylindrical	S	A	CY
Fixler	S	N	CO
Escobal Conical Oblate	O	N	CO

Looking at the different results in terms of entry and exit true anomalies and time spent in the umbra, the new proposed approach is very close to the outputs provided by Escobal conical model. However, the proposed approach is not based on the solution of a non-linear transcendental equation and on an iterative procedure to get the correct solutions. The proposed method simply solves analytically a quartic equation and defines the correct solution of such equations providing the same accurate solution. Even if all the different approaches are providing similar solutions it should be stressed that an error of $1-2^\circ$ has a different impact according to the orbital scenario. For a low-Earth orbit where the satellite moves quite fast such error can be considered negligible, but for a highly elliptical orbit the same error in terms of anomalies can result in a significant increase in the shadow time because satellites are moving slower. It is interesting to check what happens to the eclipse time error for many satellite revolutions. A simple test has been carried out propagating the trajectory of a satellite orbiting on the highly-inclined LEO and GEO orbit proposed in scenarios 2 and 4 for a number of revolutions equal to 500 and 300, respectively. The propagation includes the effect of J_2 orbital perturbation to improve the accuracy of the trajectory. All the different algorithms are used to compute the time spent in the umbra shadow for each revolution and the relative error with respect to the new algorithm is stored. The logarithm of the cumulative error at the end of propagation for the two scenarios are presented in Figures 7.7 and 7.8.

It can be noted looking at the different scenarios that the error between the different approaches increases moving to highly inclined orbits or highly elliptical orbits. This result was expected because in these two situations the cylindrical geometry for sure is not a reliable way to model the actual shadow. Therefore, it is useful to perform sensitivity analysis to check how the error changes according to a change in the orbit eccentricity or inclination. This reasoning

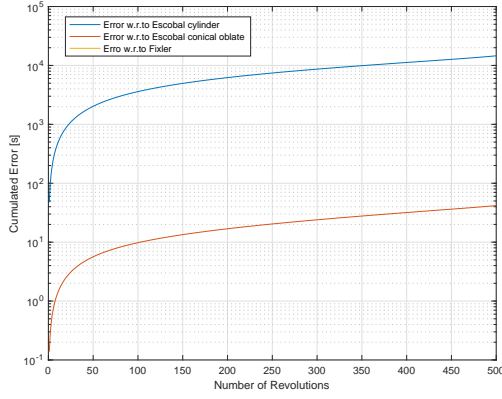


Figure 7.7: Cumulative eclipse time error for scenario 2.

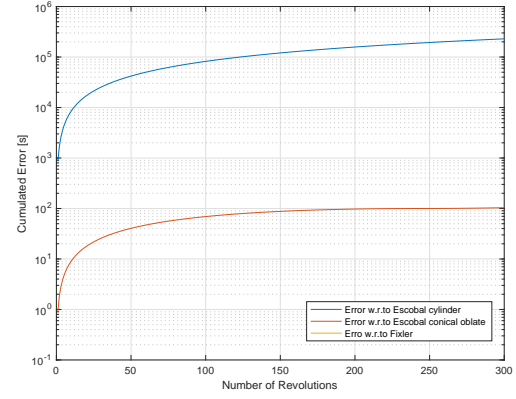


Figure 7.8: Cumulative eclipse time error for scenario 4.

has been applied starting again from scenarios number 2 and 4. Looking at Figures 7.9 and 7.10 the logarithm of the error between the time spent in the umbra shadow with respect to the one evaluated with the proposed algorithm is presented as function of the orbit inclination and eccentricity, respectively. The pictures show that the real parameter which is increasing the error in the prediction of the correct entry/exit points is the orbit inclination rather than the orbit eccentricity. This happens because Escobal cylindrical model is based on a planar approximation of the geometry which implies that the Sun, the Earth and the satellite are always in the same plane, while Fixler method derives its non-linear transcendental equation using the projection of the Sun position vector onto the satellite orbital plane.

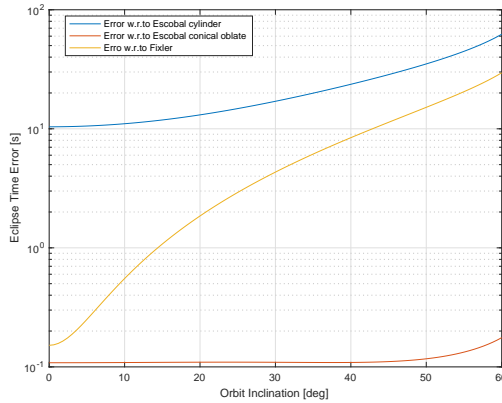


Figure 7.9: Eclipse time error as function of orbit inclination for scenario 2.

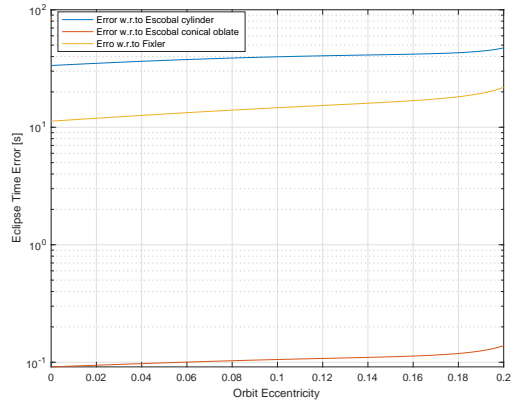


Figure 7.10: Eclipse time error as function of orbit eccentricity for scenario 2.

The same considerations can be applied for scenario number 4. Looking at Figures 7.11 and 7.12 also in this case the main parameter generating error between the different approaches is the orbit inclination. However, in this case the orbit inclination cannot increase as for the LEO case because the umbra region is definitely smaller due to the larger size of the orbit.

The last but not least important advantage of the devised approach is related to the number and type of inputs provided. Indeed, the Sun position vector, Earth position vector and spacecraft

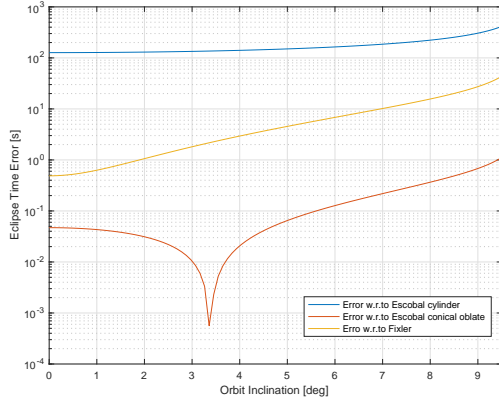


Figure 7.11: Eclipse time error as function of orbit inclination for scenario 4.

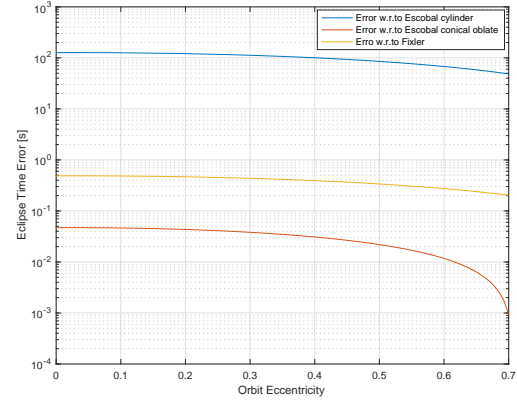


Figure 7.12: Eclipse time error as function of orbit eccentricity for scenario 4.

osculating orbital elements are variables which are always known during the high-fidelity propagation because SRP and Earth oblateness perturbing accelerations must be included. Therefore, no additional variables must be computed to derive the coefficients of the quartic equation.

7.4 Definition of the constraining area

The main task of the internal path constraint associated to eclipse is to minimise the total time spent in the shadow during the transfer obtained by the optimiser. As far as the knowledge of the author it is the first time that an optimal control problem is solved considering a constraint on the total shadow time during the evaluation of the trajectory itself.

This limitation is related to the existing algorithms associated to the eclipse computation. Both the algorithm based on Cartesian coordinates and orbital elements assume a fixed spacecraft orbit not changing with time. During the optimisation of the trajectory a thrust is provided by the low-thrust engines which is actually changing the shape and plane of the spacecraft orbit. This means that a variable orbit eclipse algorithm should be used for the evaluation of the entry/exit points.

This hard task can be solved thinking at the geometry of the 3D problem. During a generic transfer it is useful to identify the main characters during an eclipse:

- the Sun
- the occulting body
- the spacecraft

A further analysis must stress what happens to these 3 characters during the spacecraft transfers. The Sun and occulting body motions will be always the same despite the trajectory provided by the DDP algorithm. This means that the Sun and occulting body position vectors are just function of time. This information is very useful and it can be used to solve the problem of

optimising the spacecraft shadow time. Indeed, the umbra associated to the eclipse is generated by the relative position of the Sun and of the occulting body.

Therefore, also the shadow cone is independent of the trajectory generated by the DDP algorithm and it is just a function of time and it is completely unrelated with the spacecraft state. If a maximum shadow time is allowed according to some prescribed requirements that can be associated to the power budget of the batteries onboard of the spacecraft, the total shadow region can be decomposed in two sub-regions. A first region where the satellite is allowed to move satisfying the spacecraft requirements, and a second region denoted as "constraining area" where the satellite must not enter to violate the eclipse time requirements.

An example of constraining area is presented in Figure 7.13 where the red region is representing the constraining region while the black region is the total umbra generated by the occulting body where the satellite is allowed to move.

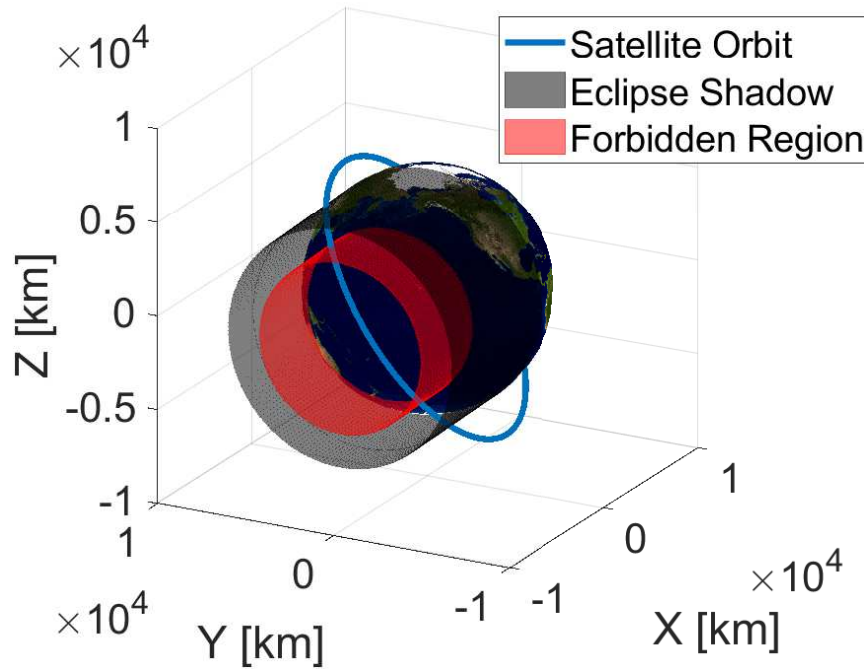


Figure 7.13: Intersection of the conical surface with the elliptical orbit.

If the shadow cone region is just function of time and independent from the satellite state, the constraining area will change with time because it will be always enclosed in the shadow but it will be also function of the spacecraft state. This dependence is fundamental because the DDP algorithm requires in the formulation of the cost functional the presence of the cost function to be optimised to admit partial derivatives with respect to the spacecraft state. If the partial derivatives are not existing no addendum appears in the backward equations which represent the satisfaction of the necessary conditions of optimality for the DDP algorithm.

7.5 Definition of the function linking shadow time with the constraining area

Once the idea of constraining region has been identified as a way to minimise the total spacecraft shadow time during the optimisation, some other issues must be solved to use this property inside the DDP algorithm:

- How to connect maximum shadow time to the geometric characteristics of the constraining area
- Identification of a figure of merit related to the maximum shadow time
- Inclusion inside the DDP algorithm

In this section the function to link the maximum eclipse time to the geometric characteristics of the constraining region is described. It is very complex to model the exact function describing how the variation of the spacecraft state is affecting the size of the constraining region. Several models can be identified according to the level of approximation which are also affecting the correlation between maximum shadow time and spacecraft state. This model is not the ending point of an objective but it is just an intermediate part in a very large process. This means that the properties of such function as smoothness, non-linearity and complexity must be taken into account. Partial derivatives of such function must be analytically evaluated several times for each iteration of the DDP algorithm. A trade-off between feasibility and simplicity of the model must be found. The following assumptions are adopted in this work:

- (a) The constraining region is assumed to be a conical region inside the conical shadow
- (b) The satellite is assumed to move with a constant velocity equal to the equivalent circular velocity at initial time throughout the eclipse region.
- (c) The satellite arc throughout the shadow is approximated with a rectilinear path.

Even if the assumptions seem to simplify considerably the problem, they are not describing a scenario that is different from the reality. Indeed, most of the cases the total time spent by the spacecraft in the umbra is in the order of few minutes. In this small amount of time it is reasonable to assume that the velocity is equal to a constant value because it is not changing so much from the reference value. Finally, the cone half-aperture angle is usually of small degrees and this means that is possible to approximate the circular arc with the cord connecting the extremes of the curvilinear orbit arc.

Thanks to these assumptions the constant tangential circular velocity can be evaluated as follows:

$$\bar{v} = \sqrt{\frac{\mu}{a_0}} \quad (7.41)$$

with \bar{v} representing the equivalent tangential circular velocity assumed to be constant and a_0 is the initial semi-major axis. The rectilinear cord which is approximated by the curvilinear arc will be the product of the constant velocity and the maximum prescribed shadow time, $\Delta T_{eclipse}$:

$$\bar{L} = \bar{v} \Delta T_{eclipse} \quad (7.42)$$

It is possible to compute the radius of the circular base of the constraining cone projecting each half of the rectilinear cord on the Earth surface. The half-aperture angle of the shadow cone can be used to compute the new spacecraft cone so that the constraining region is completely characterised.

$$\bar{c} = c_{shadow} - \frac{\bar{v}\Delta T_{eclipse}c_{shadow}}{2R_{\oplus}} \quad (7.43)$$

with \bar{c} being the height of the new constraining cone, c_{shadow} the height of the umbra cone and R_{\oplus} the Earth mean radius. The circular base radius of the constraining cone is computed as follows:

$$\bar{R} = R_{\oplus} - \frac{\bar{L}}{2} \quad (7.44)$$

It is clear that the approximated model will not minimise the exact required shadow time but it provides a way to include the internal path constraint for the first time the optimisation process during the spacecraft transfer.

7.6 Cost function formulation

The last aspect to be investigated is how to define a figure of merit that can be used as cost function to be inserted in the augmented Lagrangian formulation. The cost function must have some properties:

- i. It should be evaluated throughout the overall transfer
- ii. It must assume a value inside the forbidden region and a different value outside the region.
- iii. It should be a positive scalar variable

The first characteristic of the cost function suggests that it is a term that should be contained in the integral term in the continuous version or inside the summation in the discrete version. The second characteristic suggests that a possible candidate to be the cost function associated to the maximum shadow time is the minimum distance from a generic conical surface and in particular the constraining area defined in the previous section. There are several ways to compute the minimum distance of a generic point from the conical surface. One of the classic ways is to apply the method of Lagrange multipliers which results in a very complicated system of equations with a closed form solution that is not practical for the final objective. If the conical surface is assumed to have a right circular base the minimum distance can be evaluated using a geometric approach that allows to have both a very simple analytical expression and a figure of merit assuming a positive value outside the conical surface, zero on the surface and a negative value inside the constraining region.

Given the spacecraft position at point X , the minimum distance, d , is the shortest path from the conical surface. In case a right-circular cone is considered the shortest path is along the normal direction that intersects the conical axis in point W . The distance d can be written as:

$$d = ||X - W|| - (h - W_h) \sin \varphi \quad (7.45)$$

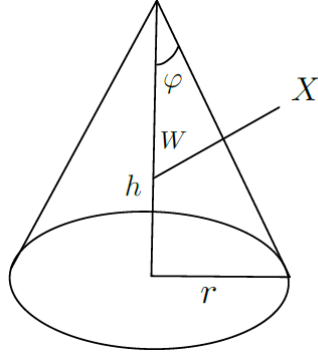


Figure 7.14: General representation of the minimum distance of a point from a right-circular cone.

It is known that the norm of a vector is invariant with respect to the reference frame. This means that the norm is the same despite the reference frame used and in this case a cylindrical reference frame is adopted. In this specific frame all the variables are decomposed in a radial and axial components.

$$d = \sqrt{(X_r - W_r)^2 + (X_h - W_h)^2} - (h - W_h \sin \phi) \quad (7.46)$$

It is obvious that no radial component for point W exists because it belongs to the conical axis. The sine of the half-aperture angle can be written as a function of the tangent that is related to the geometrical characteristics of the cone.

$$\tan \phi = \frac{r}{h} \quad (7.47)$$

The axial component of point W can be rewritten in terms of point X using classic trigonometry:

$$d = \sqrt{X_r^2 + (X_h - X_h + X_r \tan \phi)^2} - (h - X_h + X_r \tan \phi) \sin \phi \quad (7.48)$$

Using the properties of planar trigonometry a compact formulation for the minimum distance can be obtained.

$$d = \sqrt{X_r^2 (1 + \tan^2 \phi)} - (h - X_h + X_r \tan \phi) \sin \phi \quad (7.49)$$

$$d = X_r \sec \phi - (h - X_h) \sin \phi - X_r \frac{\sin^2 \phi}{\cos \phi} \quad (7.50)$$

$$d = X_r \left(\frac{1 - \sin^2 \phi}{\cos \phi} \right) - (h - X_h) \sin \phi \quad (7.51)$$

$$d = X_r \cos \phi - (h - X_h) \sin \phi \quad (7.52)$$

The formulation obtained for the minimum distance can be graphically seen from the Figure 7.15 where all the different segments are identified.

The final expression can be further simplified if the sine and cosine function are expressed in terms of the tangent and the geometric characteristics of the cone.

$$\cos \phi = \frac{1}{\sqrt{1 + \tan^2 \phi}}, \quad \sin \phi = \frac{\tan \phi}{\sqrt{1 + \tan^2 \phi}} \quad (7.53)$$

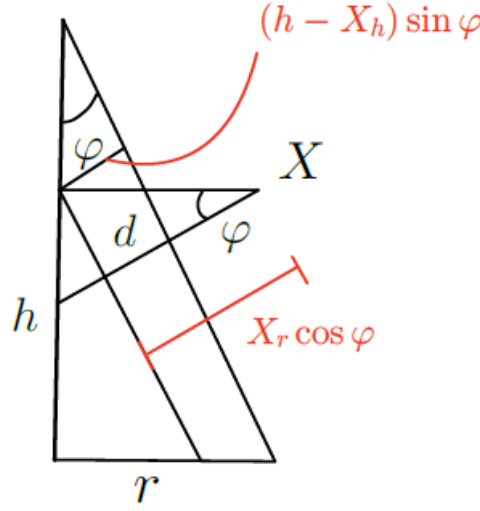


Figure 7.15: Geometrical representation of the segments to compute the minimum distance from a right-circular cone.

Given the Cartesian coordinates of the spacecraft position vector in the cone reference frame the radial and axial components in the cylindrical reference frame are evaluated in the following way:

$$X_r = \sqrt{x_s^2 + y_s^2}, \quad X_h = z_s \quad (7.54)$$

with x_s, y_s and z_s being the position vector coordinates of the spacecraft in the conical reference frame. Using Eqs. (7.47, 7.53, 7.54) inside Eq. (7.52) the minimum distance is expressed as function of the position vector coordinates and the geometrical parameters of the cone.

$$d = \sqrt{\frac{x_s^2 + y_s^2}{1 + \frac{r^2}{h^2}}} - z_s \frac{\frac{r}{h}}{\sqrt{1 + \frac{r^2}{h^2}}} \quad (7.55)$$

$$d = h \sqrt{\frac{x_s^2 + y_s^2}{r^2 + h^2}} - r \frac{z_s}{\sqrt{r^2 + h^2}} \quad (7.56)$$

$$d = \frac{h \sqrt{x_s^2 + y_s^2} - r z_s}{\sqrt{r^2 + h^2}} \quad (7.57)$$

This candidate function is perfect because it is possible impose a value equal to 0 anywhere apart from the value inside the constraining area. This variable can be squared to be sure that is assuming always positive and the minimisation of the cost function is equivalent to minimise to zero the distance from the constraining area. An example of the evolution of minimum distance squared for a single revolution spacecraft in LEO orbit is presented in Figure 7.16.

If the minimum distance from the constraining area is pushed to zero the spacecraft is moving outside the region itself and thanks to the way the constraining are is defined the spacecraft is respecting the shadow time requirements.

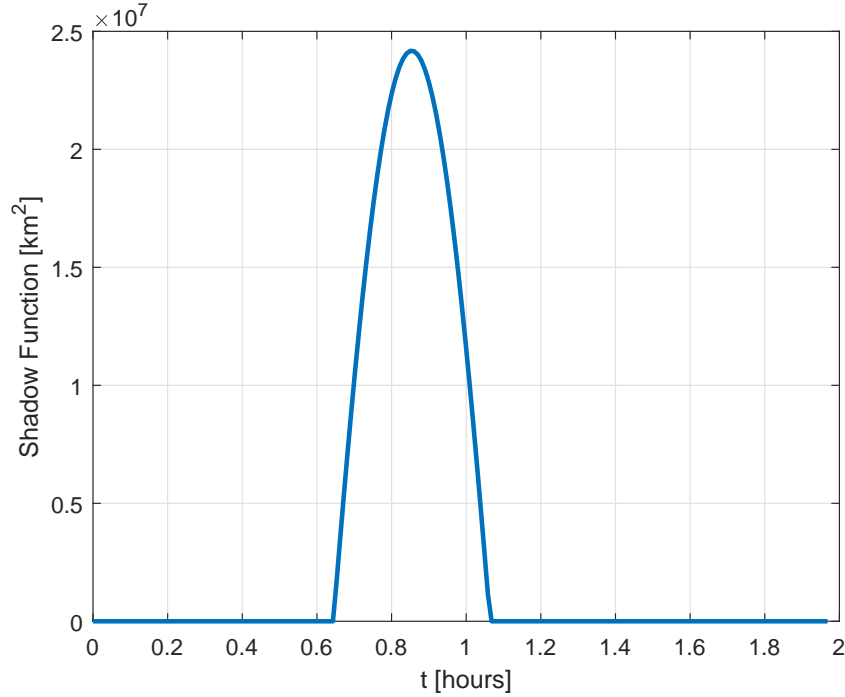


Figure 7.16: Minimum distance from a cone surface squared for a single revolution LEO spacecraft.

7.7 Integration in the DDP algorithm

After the definition of the figure of merit related to the maximum shadow time, it is possible to formulate the eclipse cost function that can be inserted inside the Augmented Lagrangian function to be optimised by the DDP algorithm.

$$L_{eclipse} = \sum_{j=0}^N \left(\frac{c\sqrt{\tilde{x}^2 + \tilde{y}^2} - r\tilde{z}}{r^2 + c^2} \right)^2 \quad (7.58)$$

The coordinates \tilde{x} , \tilde{y} and \tilde{z} are not directly the spacecraft position state in the inertial frame, but in the cone reference frame. Using the rotation matrix defined in the eclipse algorithm derivation it is possible to express the eclipse cost function in terms of the true satellite state vector.

$$\begin{cases} \tilde{x} = (x - x_v)R_{11} + (y - y_v)R_{12} + (z - z_v)R_{13} \\ \tilde{y} = (x - x_v)R_{21} + (y - y_v)R_{22} + (z - z_v)R_{23} \\ \tilde{z} = (x - x_v)R_{31} + (y - y_v)R_{32} + (z - z_v)R_{33} \end{cases} \quad (7.59)$$

This cost function is valid whenever Cartesian coordinates are used as state representation of the dynamics. What happens if orbital elements are used in place of position and velocity vectors? A different formulation must be found because the expression relating the position vector to the orbital elements are highly non linear and are complicating the formulation of the cost function.

In the original Escobal algorithm [60] assuming a cylindrical shadow and a spherical Earth

the concept of *shadow function* is introduced for the derivation of the quartic equation needed to be solved to compute the entry/exit eclipse points. The shadow function is just a geometrical condition that is imposed so that when the function is equal to 0, the entry/exit points are identified and when then function is not zero the spacecraft is in the shadow or in sunlight. This concept is very similar to the one developed for the definition of the eclipse function in terms Cartesian coordinates. The shadow function defined in Escobal [60] is the following:

$$L_{eclipse} = R_{\oplus}^2(1 + e \cos \nu)^2 + p^2(\beta \cos \nu + \epsilon \sin \nu)^2 - p^2 \quad (7.60)$$

where the coefficient β and ϵ are defined as

$$\beta = \frac{X_{\odot}P_x + Y_{\odot}P_y + Z_{\odot}P_z}{\sqrt{X_{\oplus}^2 + Y_{\oplus}^2 + Z_{\oplus}^2}} \quad (7.61)$$

$$\epsilon = \frac{X_{\odot}Q_x + Y_{\odot}Q_y + Z_{\odot}Q_z}{\sqrt{X_{\oplus}^2 + Y_{\oplus}^2 + Z_{\oplus}^2}} \quad (7.62)$$

and the unit vectors P and Q are defined as follows

$$\begin{aligned} P_x &= \cos \omega \cos \Omega - \sin \omega \sin \Omega \cos i \\ P_y &= \cos \omega \sin \Omega + \sin \omega \cos \Omega \cos i \\ P_z &= \sin \omega \sin i \end{aligned} \quad (7.63)$$

$$\begin{aligned} Q_x &= -\sin \omega \cos \Omega - \cos \omega \sin \Omega \cos i \\ Q_y &= -\sin \omega \sin \Omega + \cos \omega \cos \Omega \cos i \\ Q_z &= \cos \omega \sin i \end{aligned} \quad (7.64)$$

The augmented Lagrangian function representing the cost function to be optimised by the DDP can be formulated as follows:

$$J = \sum_{k=1}^N (\mathbf{u}_k^T \mathbf{R} \mathbf{u}_k + \nu_k w L_{eclipse}^k) h + \mathbf{b}^T \phi(\mathbf{x}_{N+1}) + \sigma \|\phi(\mathbf{x}_{N+1})\|^2 \quad (7.65)$$

where the coefficient ν_k has been introduced in front of the eclipse cost function because it should discriminate when the term must be evaluated or not, and w a weighting factor for the eclipse cost function. The eclipse cost function assumes a value equal to 0 whenever the satellite is outside the constraining area. This is true not only for the evaluation of the final cost functional but also for the evaluation of the partial derivatives. The consequence is that there will be a time instant when the partial derivatives of the cost functional will introduce a new contribution that can generate a first-order discontinuity. This is a criticality that must be carefully analysed because the creation of a discontinuity inside the backward sweep can lead to the divergence of the optimisation. The classic way to handle this issue is to introduce some smoothing functions which are transforming the hard discontinuity in a continuous function to improve the mathematical properties of the eclipse phenomenon. In this research the sigmoid function has been considered as smoothing function:

$$S(x) = \frac{1}{1 + e^{-x}} \quad (7.66)$$

The general formulation of the sigmoid function is modified to include some coefficients that are tuning the amplitude of the smoothing and interval where the correction must be applied.

$$S(x) = \frac{1}{1 + \gamma_1 e^{-x+\gamma_2}} \quad (7.67)$$

with γ_1 being the amplitude coefficient, and γ_2 the translation coefficient. Once the formulation of the final cost functional is provided the remaining part of the DDP algorithm and it can be applied for the optimisation of the maximum shadow time.

7.7.1 Application of the DDP with the shadow time requirement

In this section an orbit raising is presented to show the effectiveness of the eclipse cost function. First, the non-linear optimal control problem is solved without inserted the eclipse cost function inside the cost functional formulation. The same optimal control problem is then solved considering different values for the weighting coefficient multiplying the eclipse cost function to check how the new trajectory departs from the reference one, and to check if the spacecraft is departing from the constraining area defined in the previous section to respect the maximum shadow time. The results will be carefully discussed to enhance the advantages and bottlenecks of the method. The inputs used for the reference case are listed in Table 7.15.

Table 7.15: Input for the LEO orbit raising.

Inputs	Numerical value
Initial semi-major axis, a	8000 km
Initial eccentricity, e	0.1
Inclination, i	55°
RAAN, Ω	10°
Pericenter Anomaly, ω	20°
True anomaly, ν	0°
Final semi-major axis, a	8200 km
Final eccentricity, e	0.2
Specific Impulse, Isp	1500 s
Maximum Thrust, u_{max}	100 mN
Spacecraft Mass, m_0	450 kg

The first analysis is carried out considering the eclipse weighting factor equal to zero, $w = 0$. This analysis is carried out to check if a minimisation of the eclipse cost function is obtained independently from the addition of the new term inside the Augmented Lagrangian. The final 3D trajectory is presented in Figure 7.17 together with the shadow cone and the constraining region.

The eclipse cost function for the problem with the eclipse weighting factor equal to 0 at the end of the optimisation process increases. Indeed, the eclipse cost function corresponding to the nominal trajectory is equal to $\bar{L}_{eclipse} = 81.43$ while the eclipse cost function corresponding to the optimal trajectory is equal to $L_{eclipse}^* = 81.76$. The same example is solved increasing the eclipse weighting factor from zero to $w = 1000$. This is the first time the eclipse cost

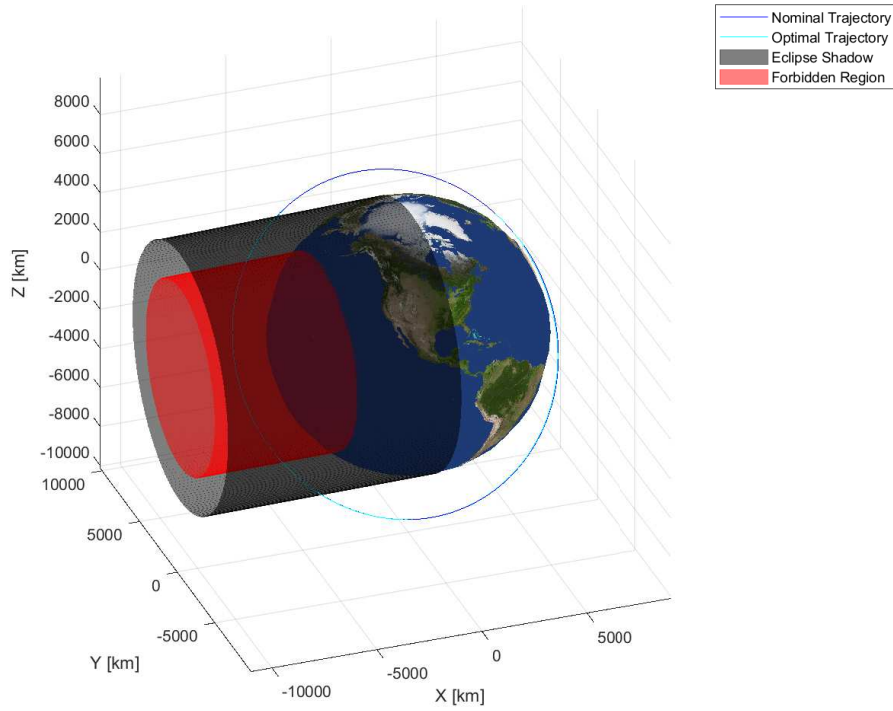


Figure 7.17: Optimal orbit raising considering the eclipse weighting factor equal to zero.

function is acting in the optimisation and a reduction of the time spend in the forbidden region is expected. The final 3D trajectory is presented in Figure 7.18 together with the shadow cone and the constraining region.

The eclipse cost function evaluated along the nominal trajectory is always the same while the one evaluated along the optimal trajectory is equal to $L_{eclipse}^* = 52.1864$. It should be remembered that this cost function is related to a squared distance and not to the eclipse time. if time is required it should be recovered considering an inverse formulation. Looking at the optimal trajectory, the optimiser chooses to minimise the time spend in the forbidden region decreasing the height so that the spacecraft will move faster in that region. The eclipse cost function can be further increase to check if a further reduction can be achieved. The final 3D trajectory considering an eclipse weighting factor $w = 2000$ is presented in Figure 7.19 together with the shadow cone and the constraining region.

The new eclipse cost function evaluated along the optimal trajectory is equal to $L_{eclipse}^* = 39.80$. The optimiser changed the optimal transfer decreasing again the height of the spacecraft orbit inside the forbidden region so that it will move faster. Therefore, the eclipse cost function formulation is effectively reducing the time inside the forbidden region. Another philosophy that can be adopted from the DDP algorithm to minimise the cost function is to change the spacecraft orbital plane instead of increasing its speed. However, this philosophy can be dangerous because the change of orbital plane is associated to a large control variation which can be not consistent with the linear-quadratic Taylor expansion.

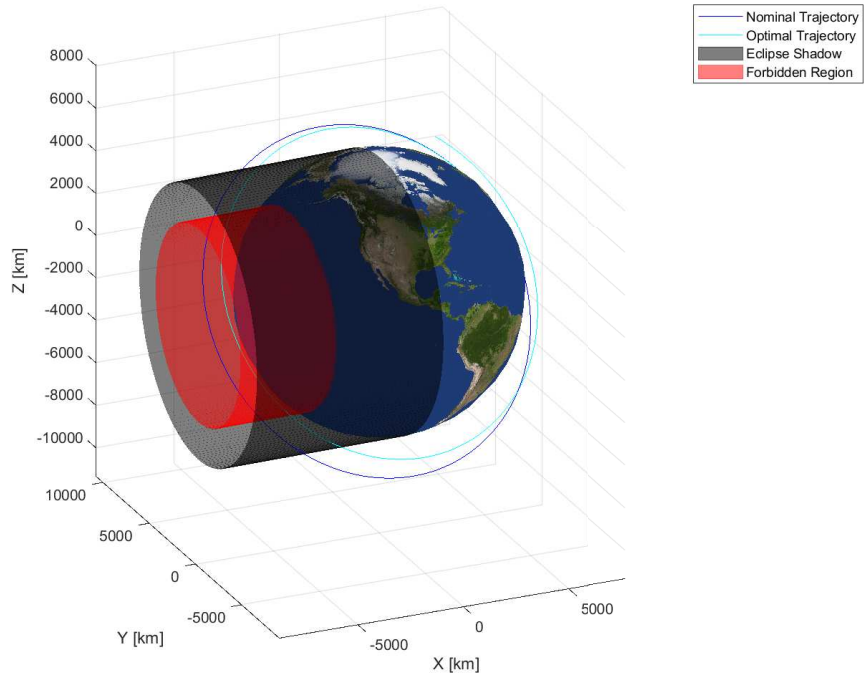


Figure 7.18: Optimal orbit raising considering the eclipse weighting factor equal to 1000.

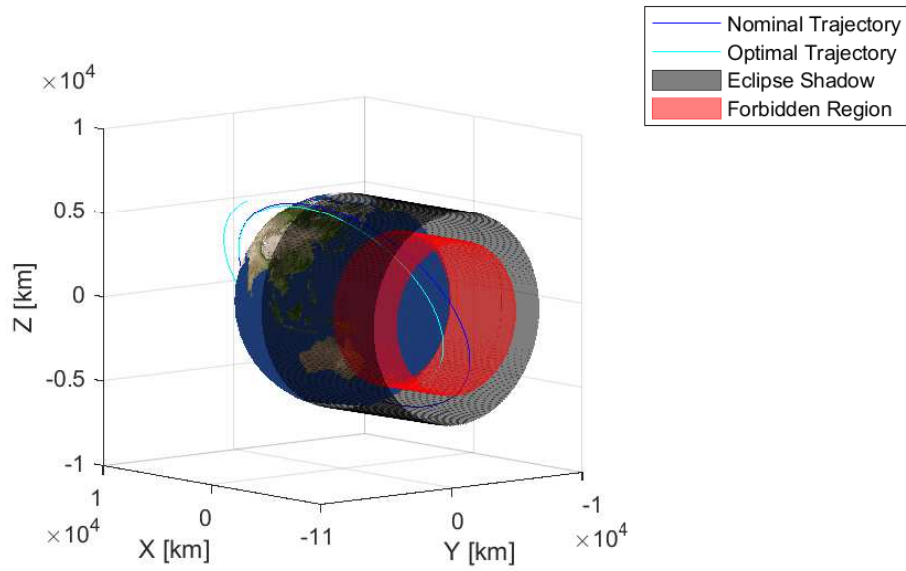


Figure 7.19: Optimal orbit raising considering the eclipse weighting factor equal to 2000.

From the several plots it is clear that the weighting factor is playing an important role in

driving the optimisation. Indeed, the weighting factor is establishing the relative importance of the different cost function inside the Augmented cost function. If the eclipse weighting factor increases the DDP algorithm will focus on the minimisation of the eclipse cost function rather than minimise the control magnitude. The introduction of the penalty term including the norm of the final constraints represents a new character to be balanced. In this case the optimiser must decide what is better to minimise between final constraints, eclipse cost function and thrust magnitude. Despite the difficulty in the selection of the weighting factors, the examples show how eclipse cost function included in the DDP allows the minimisation of the time spent inside the forbidden area. The advantage of such approach is that the formulation of the cost function can be applied to all the different indirect methods and not only to the DDP algorithm.

CHAPTER 8

Conclusions and future work

8.1 Conclusions

This research work investigated the coupling between orbital elements used as state representation of the dynamics and the DDP optimisation technique. Gauss' equations have been used to model the orbit dynamics to account both for conservative and non-conservative effects. Starting from the basic formulation of the original algorithm, a mathematical theory denoted as "quasi-independence" theory has been derived to describe the sources of the issues to solve rendezvous problems. A continuation technique is presented as methodology to overcome the issues that arise due to the presence of orbital elements. The "quasi-independence" theory highlights hidden properties associated to the choice of the initial guess that are not clear when using Cartesian coordinates. Indeed, if the initial guess is such that the pericenter anomaly and the true anomaly must be both increased or decreased, the optimisation will not be successful because the two variables cannot follow the same trend. This is a property that makes the user able to constraint the domain of functions that can be selected as first guess solution.

A recursive formulation to compute the cost-functional partial derivatives as analytical derivatives of the explicit RK integration schemes up to the 8th order is derived. The main differences between the analytical formulation of the partial derivatives and the use of STMs used in the state-of-the art HDDP is analysed. Heuristic techniques are defined to choose the proper formulation of the constraints according to the type of problem to be solved (planetary or interplanetary) together with a new approach to evaluate the constant value to shift the main diagonal of the Hessian with respect to the control variable which is independent from user selection. The formulation of Gauss' dynamics in several reference frames and in terms of both classic Keplerian elements and modified direct equinoctial elements is derived. A systematic way to use MATLAB symbolic toolbox to evaluate the cost functional partial derivatives for both continuous and discrete formulation is defined to apply the DDP algorithm. A numerical example to check also the robustness of the optimal feedback control law is presented to stress the ability of the control law to be reliable also when disturbances are included in the initial conditions.

A change of independent variable from time to eccentric anomaly through Sundman transformation is proposed to solve multi-revolutionary planetary problems. In this case the Sundman transformation is fundamental to create a fine mesh to avoid that the large distance between

two consecutive nodes can create discontinuities in the evaluation of the cost functional partial derivatives during the backward regression. The formulation based on Sundman transformation is changing the design variable of the problem that becomes the number of revolutions in place of the transfer time. The introduction of the eccentric anomaly as independent variable allows to create an equispaced mesh and to refine it increasing the number of discrete steps. The technique improves the convergence properties of the DDP algorithm when applied to solve many-revolutions orbital transfers even if hundreds of revolutions are considered. This represents one of the strongest advantages in using the DDP formulation adopted in this research with respect to other optimisation techniques. Orbital perturbations are included in the system dynamics whenever their analytical formulation is available. Earth geopotential is used as reference case to prove the effectiveness of the DDP algorithm when orbital perturbations are included. An internal path constraint related to eclipse orbit perturbation is included in the DDP optimisation algorithm. The method is based on a novel analytical formulation to define the entry/exit points of a conical shadow and the objective of the path constraint is to impose a maximum shadow time for the spacecraft when eclipse occurs.

Finally, some applications to several space missions are presented to prove the effectiveness of the DDP algorithm as numerical optimisation technique that can be coupled with orbital elements to solve low-thrust trajectory optimal control problems. The test cases explain how the continuation technique devised by the author can be used to solve any kind of orbital transfer and how the orbital elements improve the computational speed of the algorithm together with the accuracy of the results. The use of analytical formulations allows a full parametrisation of the algorithm apart from the selection of the tolerances to be used to satisfy the endpoint constraints and the check variables for the DDP technique.

8.2 Future work

The DDP algorithm is still not very used at the moment as optimisation technique for the design of low-thrust trajectories. If orbital elements are considered as state representation of the dynamics coupled with DDP the applications in the current literature is limited to less than five works. This small number of applications is first related to some limitations associated to the DDP algorithm. The DDP algorithm suffers from the lack of a systematic approach to decide how to fix the check tolerances to accept or reject of an iteration of the algorithm to be consistent with the Taylor expansions, the choice of the constant number to apply the Hessian shifting technique, and the ability to solve rendezvous problems. In this research a way to make user-independent the choice of the constant number to apply the Hessian shifting technique and a systematic method to solve rendezvous problem applying a continuation technique is proposed. One of the future tasks to improve any kind of DDP algorithm is the investigation of heuristic techniques or methods to select the check tolerances for the acceptance of a DDP iteration.

The "quasi-independence" theory presented in this research explains that the DDP algorithm based on orbital elements is very efficient in solving orbital transfer problems when all endpoint constraints apart from the one related to the fast variable (mean anomaly, true anomaly, eccentric anomaly) are imposed. This feature suggests several future opportunities for the design of both planetary and interplanetary missions in the field of averaged dynamics. When average dynam-

ics is considered the fast variable is no longer present making the DDP optimisation technique a perfect choice to solve problems as de-orbiting and re-entry of a spacecraft. The averaged dynamics fields presents also semi-analytical formulations which are modelling orbital perturbations, which can be inserted in the parametrised structure of the proposed DDP algorithm thanks to their formulations.

A last subject of investigation is the application of DDP optimisation technique to problems using relative orbital elements. Such problems are based on small variations of the orbital elements which should fit with the assumptions used to derive the backward difference equations representing the necessary conditions of optimality of the DDP algorithm.

Bibliography

- [1] R. Jehn, D. Garcia, J. Schoenmaekers, and V. Companys, “Trajectory design for bepi-colombo based on navigation requirements,” *Journal of Aerospace Engineering Sciences and Applications*, vol. 4, pp. 1–9, January 2012.
- [2] C. D. Edwards Jr., B. Arnold, R. DePaula, G. Kazz, C. Lee, and G. Noreen, “Relay communications strategies for mars exploration through 2020,” *Acta Astronautica*, vol. 59, pp. 310–318, April 2006.
- [3] T. N. Edelbaum, “Optimum power-limited orbit transfer in strong gravity fields,” *AIAA Journal*, vol. 3, no. 5, pp. 921–925, May 1965.
- [4] C. A. Shoemaker, G. J. Whiffen, and J. A. Denu, *Optimizing Groundwater Remediation Including Weighted Feedback for Incorporating Model Uncertainty*. Dordrecht: Springer Netherlands, 1995, pp. 123–126.
- [5] G. J. Whiffen and C. A. Shoemaker, “Nonlinear weighted feedback control of groundwater remediation under uncertainty,” *Water Resources Research*, vol. 29, no. 9, pp. 3277–3289, 1993.
- [6] D. M. Murray and S. J. Yakowitz, “Constrained differential dynamic programming and its application to multireservoir control,” *Water Resources Research*, vol. 15, no. 5, pp. 1017–1027, 1979.
- [7] “Dawn mission (nasa) homepage,” <https://solarsystem.nasa.gov/missions/dawn/overview/>, accessed: 2023-01-04.
- [8] G. 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*, 2006, pp. 1–12.
- [9] J. T. Betts, “Survey of numerical methods for trajectory optimization,” *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193–207, 1998.
- [10] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. London: Academic Press Inc., 1981.
- [11] Y. Gao and C. Kluever, *Low-Thrust Interplanetary Orbit Transfers Using Hybrid Trajectory Optimization Method with Multiple Shooting*.

-
- [12] J. T. Betts and P. D. Frank, "A sparse nonlinear optimization algorithm," *Journal of Optimization Theory and Applications*, vol. 82, no. 3, pp. 519–541, September 1994.
- [13] P. E. Gill, W. Murray, and M. A. Saunders, "Snopt: An sqp algorithm for large-scale constrained optimization," *SIAM Journal of Optimization*, vol. 47, no. 1, pp. 99–131, February 2005.
- [14] A. Watcher and L. T. Biegler, "On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming," *Mathematical Programming, Series A*, vol. 106, no. 1, pp. 25–27, April 2005.
- [15] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, "User's guide for npsol 5.0: a fortran package for nonlinear programming," Department of Management Science and Engineering (MS&E), Stanford, California, Tech. Rep. Technical Report SOL 86-1, July 1998.
- [16] R. H. Byrd, J. Nocedal, and R. A. Waltz, *Knitro: An Integrated Package for Nonlinear Optimization*. Boston, MA: Springer US, 2006, pp. 35–59.
- [17] A. Rao, D. Benson, C. Darby, M. Patterson, C. Francolin, I. Sanders, and G. Huntington, "Algorithm 902: Gpops, a matlab software for solving multiple-phase optimal control problem," *ACM Transactions on Mathematical Software*, vol. 37, no. 2, pp. 1–39, April 2010.
- [18] V. Becerra, "Solving complex optimal control problems at no cost with psopt," *2010 IEEE International Symposium on Computer-Aided Control System Design*, pp. 1391–1396, 2010.
- [19] R. Vanderbei, "Loqo: An interior point code for quadratic programming," *Optimization Methods and Software*, vol. 11, no. 1-4, pp. 451–484, March 2002.
- [20] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, 2nd ed. Society for Industrial and Applied Mathematics, 2010. [Online]. Available: <https://epubs.siam.org/doi/abs/10.1137/1.9780898718577>
- [21] C. Hargraves and S. Paris, "Direct trajectory optimization using nonlinear programming and collocation," *Journal of Guidance, Control, and Dynamics*, vol. 10, no. 4, pp. 338–342, 1987.
- [22] S. Tang and B. A. Conway, "Optimization of low-thrust interplanetary trajectories using collocation and nonlinear programming," *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 3, pp. 599–604, 1995.
- [23] P. Gath and K. Well, *Trajectory optimization using a combination of direct multiple shooting and collocation*. [Online]. Available: <https://arc.aiaa.org/doi/abs/10.2514/6.2001-4047>
- [24] F. Fahroo and I. M. Ross, "Direct trajectory optimization by a chebyshev pseudospectral method," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 1, pp. 160–166, 2002.

-
- [25] A. L. Herman and B. A. Conway, "Direct optimization using collocation based on high-order gauss-lobatto quadrature rules," *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 3, pp. 592–599, 1996.
- [26] D. J. Somavarapu, E. W. Wahl, A. Westfall, and K. Turkoglu, *Trajectory Optimization Procedure for Interplanetary Transfers based on Direct Collocation Method*. [Online]. Available: <https://arc.aiaa.org/doi/abs/10.2514/6.2016-1911>
- [27] M. Vasile and F. Bernelli-Zazzera, "Optimizing low-thrust and gravity assist maneuvers to design interplanetary trajectories." *Journal of Astronautical Sciences*, vol. 51, no. 1, pp. 13–25, 2003.
- [28] V. M. and K. Howell, *Global Low-Thrust Trajectory Optimization Through Hybridization of a Genetic Algorithm and a Direct Method*. [Online]. Available: <https://arc.aiaa.org/doi/abs/10.2514/6.2008-6614>
- [29] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. New York, NY, USA: John Wiley & Sons, 1987.
- [30] G. Colasurdo, D. Pastrone, and L. Casalino, *Optimization of rocket ascent trajectories using an indirect procedure*. [Online]. Available: <https://arc.aiaa.org/doi/abs/10.2514/6.1995-3323>
- [31] F. Fahroo and I. Ross, *Trajectory optimization by indirect spectral collocation methods*. [Online]. Available: <https://arc.aiaa.org/doi/abs/10.2514/6.2000-4028>
- [32] M. R. Sentinella and L. Casalino, *Genetic Algorithm and Indirect Method Coupling for Low-Thrust Trajectory Optimization*. [Online]. Available: <https://arc.aiaa.org/doi/abs/10.2514/6.2006-4468>
- [33] C. L. Ranieri and C. A. Ocampo, "Indirect optimization of spiral trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 6, pp. 1360–1366, 2006.
- [34] M. La Mantia and L. Casalino, "Indirect optimization of low-thrust capture trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 4, pp. 1011–1014, 2006.
- [35] K. A. Bokelmann and R. P. Russell, "Optimization of impulsive europa capture trajectories using primer vector theory," *The Journal of the Astronautical Sciences*, vol. 67, pp. 485–510, June 2019.
- [36] L. Casalino, G. Colasurdo, and M. R. Sentinella, *Low-Thrust Trajectories to Mercury with Multiple Gravity Assists*.
- [37] L. Casalino and F. Simeoni, "Indirect optimization of asteroid deflection missions with electric propulsion," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 2, pp. 423–433, 2012.
- [38] D. Mayne, "A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems," *International Journal of Control*, vol. 3, no. 1, pp. 85–95, 1966.

-
- [39] D. H. Jacobson and D. Q. Mayne, *Differential Dynamic Programming*. American Elsevier Publishing Company, 1970.
- [40] S. B. Gershwin and D. H. Jacobson, "A discrete-time differential dynamic programming algorithm with application to optimal orbit transfer," *AIAA Journal*, vol. 8, no. 9, pp. 1616–1626, 1970. [Online]. Available: <https://doi.org/10.2514/3.5955>
- [41] *Algorithms and computational techniques in differential dynamic programming*. New York: Academic Press, 1985, vol. 31, p. 1989.
- [42] C. Colombo, M. Vasile, and G. Radice, "Optimal low-thrust trajectories to asteroids through an algorithm based on differential dynamic programming," *Celestial Mechanics and Dynamical Astronomy*, vol. 105, no. 1, pp. 75–112, 2009.
- [43] G. Lantoine and R. P. Russell, "A hybrid differential dynamic programming algorithm for constrained optimal control problems. part 1: Theory," *Journal of Optimization Theory and Applications*, vol. 154, no. 2, pp. 382–417, 2012.
- [44] M. M., "Hybrid differential dynamic programming algorithm for low-thrust trajectory design using exact high-order transition maps," Master's thesis, Politecnico di Milano, 2018.
- [45] N. Ozaki, S. Campagnola, R. Funase, and C. H. Yam, "Stochastic differential dynamic programming with unscented transform for low-thrust trajectory design," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 2, pp. 377–387, 2018. [Online]. Available: <https://doi.org/10.2514/1.G002367>
- [46] S. J. Yakowitz and B. Rutherford, "Computational aspects of discrete-time optimal control," *Applied Mathematics and Computations*, vol. 15, no. 1, pp. 29–45, 1984.
- [47] A. E. Petropoulos and J. A. Siims, "A review of some exact solutions to the planar equations of motion of a thrusting spacecraft," in *Proceedings of the 2nd International Symposium on Low Thrust Trajectories*, Toulouse, 2022.
- [48] P. De Pascale and M. Vasile, "Preliminary design of low-thrust multiple gravity-assist trajectories," *Journal of Spacecraft and Rockets*, vol. 43, no. 5, pp. 1065–1076, 2006.
- [49] B. J. Wall and B. A. Conway, "Shape-based approach to low-thrust rendezvous trajectory design," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 1, pp. 95–101, 2009.
- [50] B. Wall, *Shape-Based Approximation Method for Low-Thrust Trajectory Optimization*. [Online]. Available: <https://arc.aiaa.org/doi/abs/10.2514/6.2008-6616>
- [51] D. M. Novak and M. Vasile, "Improved shaping approach to the preliminary design of low-thrust trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 1, pp. 128–147, 2011.
- [52] E. Vellutini and G. Avanzini, "Shape-based design of low-thrust trajectories to cislunar lagrangian point," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 4, pp. 1329–1335, 2014.

-
- [53] J. T. Betts, "Very low-thrust trajectory optimization using a direct sqp method," *Journal of Computational and Applied Mathematics*, vol. 120, no. 1, pp. 27–40, 2000. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0377042700003010>
- [54] A. Petropoulos, *Low-Thrust Orbit Transfers Using Candidate Lyapunov Functions with a Mechanism for Coasting*.
- [55] Y. Gao and C. A. Kluever, "Analytic orbital averaging technique for computing tangential-thrust trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 6, pp. 1320–1323, 2005.
- [56] H. Liu and B. H. Tongue, "Indirect spacecraft trajectory optimization using modified equinoctial elements," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 2, pp. 619–623, 2010.
- [57] F. Zuiani, M. Vasile, A. Palmas, and G. Avanzini, "Direct transcription of low-thrust trajectories with finite trajectory elements," *Acta Astronautica*, vol. 72, pp. 108–120, 2012.
- [58] M. Pontani, "Optimal space trajectories with multiple coast arcs using modified equinoctial elements," *Journal of Optimization Theory and Applications*, vol. 191, pp. 545–574, 2021.
- [59] J. D. Aziz, J. S. Parker, D. J. Scheeres, and J. A. Englander, "Low-thrust many-revolution trajectory optimization via differential dynamic programming and a sundman transformation," *The Journal of the Astronautical Sciences*, vol. 65, no. 2, pp. 205–228, 2018.
- [60] P. R. Escobal, *Methods of Orbit Determination*. Malabar, Florida: Krieger, 1965, ch. 4, pp. 162–174.
- [61] D. A. Vallado, *Fundamentals of Astrodynamics and Applications*, 4th ed., ser. Space Technology Library. Hawthorne: Microcosm, 2013.
- [62] S. Z. Fixler, "Umbra and penumbra eclipse factors for satellite orbits," *AIAA Journal*, vol. 2, no. 8, pp. 1455–1457, August 1964.
- [63] J. D. Kraft, "A solution for satellite orbit time in umbra and penumbra with application to a lunar satellite mission analysis," Goddard Space Flight Center, Tech. Rep. N66-17237, November 1965.
- [64] C. Colombo and C. McInnes, "Orbit design for future spacechip swarm missions in a planetary atmosphere," *Acta Astronautica*, vol. 75, pp. 25–41, 2012. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0094576512000161>
- [65] C. R. Ortiz Longo and S. L. Rickman, "A method for the calculation of the spacecraft umbra and penumbra shadow terminator points," Lyndon B. Johnson Space Center, Tech. Rep. 3547, April 1995.
- [66] O. Montenbruck and E. Gill, *Satellite orbits: models, methods and applications*. Berlin: Springer, 2012.
- [67] P. E. Dreher, "Satellite shadow time," George C. Marshall Space Flight Center, Tech. Rep. TMX-53389, October 1965.

-
- [68] D. Vokrouhlicky, P. Farinella, and F. Mignard, "Solar radiation pressure perturbations for earth satellites: Effects of the earth polar flattening on the shadow structure and the penumbra transitions," *Astronomy and Astrophysics*, vol. 307, pp. 635–644, 1996.
- [69] S. Adhya, A. Sibthorpe, and M. Ziebart, "Oblate earth eclipse state algorithm for low-earth-orbiting satellites," *Journal of Spacecraft*, vol. 41, pp. 157–159, 2003.
- [70] D. Vokrouhlicky, P. Farinella, and F. Mignard, "Solar radiation pressure perturbations for earth satellites: A complete theory including penumbra transitions," *Astronomy and Astrophysics*, vol. 280, pp. 295–312, 1993.
- [71] J. Woodburn, "Effects of eclipse boundary crossings on the numerical integration of orbit trajectories," in *Proceedings of the Astrodynamics Specialist Conference*, Colorado, Denver, CO, August 2000 2000, pp. 116–122.
- [72] M. Nugnes and C. Colombo, "Low-thrust trajectory optimisation through differential dynamic programming method based on keplerian orbital elements," in *70th International Astronautical Congress (IAC)*, Washington D.C., 2019.
- [73] —, "A new methodology for the solution to the stiffness problem applied to low-thrust trajectory optimisation in terms of orbital elements using the differential dynamic programming," in *31st AAS/AIAA Space Flight Mechanics Meeting*, 2021.
- [74] M. Nugnes and C. C., "Low-thrust trajectory optimisation problem mathematical theory using differential dynamic programming based on orbital elements," submitted to *Journal of Guidance, Control, and Dynamics* for peer-review. Expected date: 2023.
- [75] M. Nugnes and C. Colombo, "New analytical method for eclipse entry/exit positions determination considering a conical shadow and an oblate earth surface," in *2022 AAS/AIAA Astrodynamics Specialist Conference*, Charlotte, 2022.
- [76] M. Nugnes and C. C., "Analytical determination of eclipse entry and exit points considering a conical shadow and oblate earth," submitted to *Journal of Guidance, Control, and Dynamics* for peer-review. Expected date: 2023.
- [77] M. Nugnes and C. Colombo, "Multi-revolution low-thrust trajectory optimisation using differential dynamic programming in orbital element formulation," in *73rd International Astronautical Congress (IAC)*, Paris, 2022.
- [78] R. H. Battin, *An Introduction to the Mathematics and Methods of Astrodynamics*. AIAA Educational Series, 1999.
- [79] H. D. Curtis, *Orbital Mechanics for Engineering Students*, 3rd ed. Elsevier Aerospace Engineering Series, 2014.
- [80] M. J. H. Walker and J. Ireland, B. Owens, "A set of modified equinoctial orbital elements," *Celestial Mechanics*, vol. 36, pp. 409–419, 1985.
- [81] P. Prince and J. R. Dormand, "High order embedded runge-kutta formulae," *Journal of Computational and Applied Mathematics*, vol. 7, no. 1, pp. 67–75, 1981. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0771050X81900103>

- [82] R. S. Park and D. J. Scheeres, “Nonlinear semi-analytic methods for trajectory estimation,” *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 6, pp. 1668–1676, 2007.
- [83] R. Bellman, *Dynamic Programming*. Princeton University Press, 1957.
- [84] L.-Z. Liao and C. Shoemaker, “Convergence in unconstrained discrete-time differential dynamic programming,” *IEEE Transactions on Automatic Control*, vol. 36, no. 6, pp. 692–706, 1991.
- [85] R. B. Schnabel and E. Eskow, “A revised modified cholesky factorization algorithm,” *SIAM Journal on Optimization*, vol. 9, no. 4, pp. 1135–1148, 1999.
- [86] J. D. Aziz and D. J. Scheeres, “Sundman-transformed differential dynamic programming with modified equinoctial elements,” *The Journal of the Astronautical Sciences*, vol. 66, no. 4, pp. 419–445, 2019.
- [87] M. Majji, J. L. Junkins, and J. D. Turner, “A high order method for estimation of dynamic systems,” *The Journal of the Astronautical Sciences*, vol. 56, no. 3, pp. 401–440, 2008.
- [88] D. A. Maturi, M. Z. Ullah, S. Ahmad, and F. Ahmad, “An efficient computation of effective ground range using an oblate earth model,” *Abstract and Applied Analysis*, vol. 2014, August 2014.

APPENDIX A

Gauss' planetary equations in different reference frames and orbital elements set

A.1 Gauss equations in terms of classic Keplerian orbital elements

In this subsection the expressions of Gauss planetary equations in different reference frames in terms of classic Keplerian elements are reported. Gauss' equation in reference frame $[\hat{\mathbf{r}}, \hat{\boldsymbol{\vartheta}}, \hat{\mathbf{h}}]$ in terms of the eccentric anomaly, ν :

$$\begin{aligned}\frac{da}{dt} &= 2\sqrt{\frac{a^3}{\mu(1-e^2)}} \left[e \sin \nu \frac{u_r}{m} + (1 + e \cos \nu) \frac{u_\vartheta}{m} \right] \\ \frac{de}{dt} &= \sqrt{\frac{a(1-e^2)}{\mu}} \left[\sin \nu \frac{u_r}{m} + \frac{e + 2e \cos \nu + e \cos^2 \nu}{1 + e \cos \nu} \frac{u_n}{m} \right] \\ \frac{di}{dt} &= \sqrt{\frac{a(1-e^2)}{\mu}} \frac{\cos(\omega + \nu)}{1 + e \cos \nu} \frac{u_h}{m} \\ \frac{d\Omega}{dt} &= \sqrt{\frac{a(1-e^2)}{\mu}} \frac{\sin(\omega + \nu)}{\sin i (1 + e \cos \nu)} \frac{u_h}{m} \\ \frac{d\omega}{dt} &= \sqrt{\frac{a(1-e^2)}{\mu}} \left\{ \frac{1}{e} \left[-\cos \nu \frac{u_r}{m} + \frac{2e + e \cos \nu}{1 + e \cos \nu} \frac{u_\vartheta}{m} \right] - \frac{\cos i \sin(\omega + \nu)}{\sin i (1 + e \cos \nu)} \frac{u_h}{m} \right\} \\ \frac{d\nu}{dt} &= \sqrt{\frac{\mu}{[a(1-e^2)]^3}} (1 + e \cos \nu)^2 - \sqrt{\frac{a(1-e^2)}{\mu}} \frac{1}{e} \left[-\cos \nu \frac{u_r}{m} + \frac{2 + e \cos \nu}{1 + e \cos \nu} \sin \nu \frac{u_\vartheta}{m} \right] \\ \frac{dm}{dt} &= -\frac{\sqrt{u_r^2 + u_\vartheta^2 + u_h^2}}{Isp \cdot g_0}\end{aligned}$$

Gauss' equation in reference frame $[\hat{\mathbf{r}}, \hat{\boldsymbol{\vartheta}}, \hat{\mathbf{h}}]$ in terms of the eccentric anomaly, E :

$$\begin{aligned}
 \frac{da}{dt} &= 2\sqrt{\frac{a^3}{\mu}} \frac{1}{1 - e \cos E} \left[e \sin E \frac{u_r}{m} + \sqrt{1 - e^2} \frac{u_{\vartheta}}{m} \right] \\
 \frac{de}{dt} &= \sqrt{\frac{a(1 - e^2)}{\mu}} \frac{1}{1 - e \cos E} \left[\sqrt{1 - e^2} \sin E \frac{u_r}{m} + (2 \cos E - e - e \cos^2 E) \frac{u_{\vartheta}}{m} \right] \\
 \frac{di}{dt} &= \sqrt{\frac{a}{\mu(1 - e^2)}} \left[\cos \omega (\cos E - e) - \sqrt{1 - e^2} \sin E \sin \omega \right] \frac{u_h}{m} \\
 \frac{d\Omega}{dt} &= \sqrt{\frac{a}{\mu}} \frac{1}{\sqrt{1 - e^2} \sin i} \left[\sin \omega (\cos E - e) + \sqrt{1 - e^2} \cos \omega \sin E \right] \frac{u_h}{m} \\
 \frac{d\omega}{dt} &= \sqrt{\frac{a}{\mu}} \left\{ \frac{1}{e(1 - e \cos E)} \left[-\sqrt{1 - e^2} (\cos E - e) \frac{u_r}{m} + (2 - e \cos E - e^2) \sin E \frac{u_{\vartheta}}{m} \right] + \right. \\
 &\quad \left. - \frac{\cos i}{\sin i} \frac{1}{\sqrt{1 - e^2}} \left[\sin \omega (\cos E - e) + \sqrt{1 - e^2} \cos \omega \sin E \right] \frac{u_h}{m} \right\} \\
 \frac{dE}{dt} &= \sqrt{\frac{\mu}{a^3}} \frac{1}{1 - e \cos E} + \sqrt{\frac{a}{\mu}} \frac{1}{e(1 - e \cos E)} \left[(\cos E - 2e + e^2 \cos E) \frac{u_r}{m} + \right. \\
 &\quad \left. - \sqrt{1 - e^2} (2 - e \cos E) \sin E \frac{u_{\vartheta}}{m} \right] \\
 \frac{dm}{dt} &= -\frac{\sqrt{u_r^2 + u_{\vartheta}^2 + u_h^2}}{Isp \cdot g_0}
 \end{aligned}$$

Gauss' equation in reference frame $[\hat{\mathbf{t}}, \hat{\mathbf{n}}, \hat{\mathbf{h}}]$ in terms of the eccentric anomaly, E :

$$\begin{aligned}
 \frac{da}{dt} &= 2\sqrt{\frac{a^3}{\mu}} \sqrt{\frac{1 + e \cos E}{1 - e \cos E}} \frac{u_t}{m} \\
 \frac{de}{dt} &= \sqrt{\frac{a(1 - e^2)}{\mu}} \sqrt{\frac{1 - e \cos E}{1 + e \cos E}} \left[2\sqrt{1 - e^2} \frac{\cos E}{1 - e \cos E} \frac{u_t}{m} - \sin E \frac{u_n}{m} \right] \\
 \frac{di}{dt} &= \sqrt{\frac{a}{\mu(1 - e^2)}} \left[\cos \omega (\cos E - e) - \sqrt{1 - e^2} \sin E \sin \omega \right] \frac{u_h}{m} \\
 \frac{d\Omega}{dt} &= \sqrt{\frac{a}{\mu}} \frac{1}{\sqrt{1 - e^2} \sin i} \left[\sin \omega (\cos E - e) + \sqrt{1 - e^2} \cos \omega \sin E \right] \frac{u_h}{m} \\
 \frac{d\omega}{dt} &= \sqrt{\frac{a}{\mu}} \left\{ \frac{1}{e} \sqrt{\frac{1 - e \cos E}{1 + e \cos E}} \left[2\sqrt{1 - e^2} \frac{\sin E}{1 - e \cos E} \frac{u_t}{m} + (e + \cos E) \frac{u_n}{m} \right] + \right. \\
 &\quad \left. - \frac{\cos i}{\sin i} \frac{1}{\sqrt{1 - e^2}} \left[\sin \omega (\cos E - e) + \sqrt{1 - e^2} \cos \omega \sin E \right] \frac{u_h}{m} \right\} \\
 \frac{dE}{dt} &= \sqrt{\frac{\mu}{a^3}} \frac{1}{1 - e \cos E} - \sqrt{\frac{a}{\mu}} \frac{1}{e} \sqrt{\frac{1 - e \cos E}{1 + e \cos E}} \left[2 \frac{\sin E}{1 - e \cos E} \frac{u_t}{m} + \sqrt{1 - e^2} \cos E \frac{u_n}{m} \right] \\
 \frac{dm}{dt} &= -\frac{\sqrt{u_t^2 + u_n^2 + u_h^2}}{Isp \cdot g_0}
 \end{aligned}$$

A.2 Gauss equations in terms of modified equinoctial elements

In this section the expressions of Gauss planetary equations in different reference frames in terms of modified equinoctial elements are reported. Gauss' equation in reference frame $[\hat{\mathbf{r}}, \hat{\boldsymbol{\vartheta}}, \hat{\mathbf{h}}]$ in terms of the true longitude, L :

$$\begin{aligned}
\frac{da}{dt} &= 2\sqrt{\frac{a^3}{\mu(1-f^2-g^2)}} \left[(f \sin L - g \cos L) \frac{u_r}{m} + (1 + f \cos L + g \sin L) \frac{u_\vartheta}{m} \right] \\
\frac{df}{dt} &= \sqrt{\frac{a(1-f^2-g^2)}{\mu}} \left\{ \sin L \frac{u_r}{m} + \frac{1}{1+f \cos L + g \sin L} [f + \cos L(2 + f \cos L + \right. \\
&\quad \left. + g \sin L)] \frac{u_\vartheta}{m} + \frac{g}{1+f \cos L + g \sin L} (k \cos L - h \sin L) \frac{u_h}{m} \right\} \\
\frac{dg}{dt} &= \sqrt{\frac{a(1-f^2-g^2)}{\mu}} \left\{ -\cos L \frac{u_r}{m} + \frac{1}{1+f \cos L + g \sin L} [g + \sin L(2 + f \cos L + \right. \\
&\quad \left. + g \sin L)] \frac{u_\vartheta}{m} + \frac{g}{1+f \cos L + g \sin L} (h \sin L - k \cos L) \frac{u_h}{m} \right\} \\
\frac{dh}{dt} &= \frac{1}{2} \sqrt{\frac{a(1-f^2-g^2)}{\mu}} \frac{1+h^2+k^2}{1+f \cos L + g \sin L} \cos L \frac{u_h}{m} \\
\frac{dk}{dt} &= \frac{1}{2} \sqrt{\frac{a(1-f^2-g^2)}{\mu}} \frac{1+h^2+k^2}{1+f \cos L + g \sin L} \sin L \frac{u_h}{m} \\
\frac{dL}{dt} &= \sqrt{\frac{\mu}{[a(1-f^2-g^2)]^3}} (1+f \cos L + g \sin L)^2 + \\
&\quad + \sqrt{\frac{a(1-f^2-g^2)}{\mu}} \frac{h \sin L - k \cos L}{1+f \cos L + g \sin L} \frac{u_h}{m} \\
\frac{dm}{dt} &= -\frac{\sqrt{u_r^2 + u_\vartheta^2 + u_h^2}}{Isp \cdot g_0}
\end{aligned}$$

Gauss' equation in reference frame $[\hat{\mathbf{r}}, \hat{\boldsymbol{\vartheta}}, \hat{\mathbf{h}}]$ in terms of the eccentric longitude, F :

$$\begin{aligned}
 \frac{da}{dt} &= 2\sqrt{\frac{a^3}{\mu}} \frac{1}{1 - f \cos F - g \sin F} \left[(f \sin F - g \cos F) \frac{u_r}{m} + \sqrt{1 - f^2 - g^2} \frac{u_\vartheta}{m} \right] \\
 \frac{df}{dt} &= \sqrt{\frac{a}{\mu}} \frac{1}{1 - f \cos F - g \sin F} \left\{ \sqrt{1 - f^2 - g^2} \left[-g + \sin F \left(1 - \frac{f^2}{1 + \sqrt{1 - f^2 - g^2}} \right) \right. \right. \\
 &\quad \left. \left. + \cos F \frac{fg}{1 + \sqrt{1 - f^2 - g^2}} \right] \frac{u_r}{m} + \left[-f \sqrt{1 - f^2 - g^2} + \cos F (-g^2 + \right. \right. \\
 &\quad \left. \left. + 2 \left(1 - \frac{f^2}{1 + \sqrt{1 - f^2 - g^2}} \right) \right) + fg \sin F \left(1 - \frac{2}{1 + \sqrt{1 - f^2 - g^2}} \right) \right. \right. \\
 &\quad \left. \left. + g \sin F \cos F \left(-1 + 2 \frac{f^2}{1 + \sqrt{1 - f^2 - g^2}} \right) + f \cos^2 F \left(-1 + \frac{f^2}{1 + \sqrt{1 - f^2 - g^2}} \right) \right. \right. \\
 &\quad \left. \left. + g \frac{fg}{1 + \sqrt{1 - f^2 - g^2}} \sin^2 F \right] \frac{u_\vartheta}{m} - g \frac{1 - f \cos F - g \sin F}{\sqrt{1 - f^2 - g^2}} [(fk - gh) + \right. \\
 &\quad \left. + \cos F \left(h \frac{fg}{1 + \sqrt{1 - f^2 - g^2}} - k \left(1 - \frac{g^2}{1 + \sqrt{1 - f^2 - g^2}} \right) \right) \right. \\
 &\quad \left. \left. + \sin F \left(h \left(1 - \frac{f^2}{1 + \sqrt{1 - f^2 - g^2}} \right) - k \frac{fg}{1 + \sqrt{1 - f^2 - g^2}} \right) \right] \frac{u_h}{m} \right\} \\
 \frac{dg}{dt} &= \sqrt{\frac{a}{\mu}} \frac{1}{1 - f \cos F - g \sin F} \left\{ \sqrt{1 - f^2 - g^2} \left[f - \cos F \left(1 - \frac{g^2}{1 + \sqrt{1 - f^2 - g^2}} \right) \right. \right. \\
 &\quad \left. \left. - \sin F \frac{fg}{1 + \sqrt{1 - f^2 - g^2}} \right] \frac{u_r}{m} + \left[-g \sqrt{1 - f^2 - g^2} + \sin F (-f^2 + \right. \right. \\
 &\quad \left. \left. + 2 \left(1 - \frac{g^2}{1 + \sqrt{1 - f^2 - g^2}} \right) \right) + fg \cos F \left(1 - \frac{2}{1 + \sqrt{1 - f^2 - g^2}} \right) + \right. \\
 &\quad \left. \left. + f \sin F \cos F \left(-1 + 2 \frac{g^2}{1 + \sqrt{1 - f^2 - g^2}} \right) - g \sin^2 F \left(1 - \frac{g^2}{1 + \sqrt{1 - f^2 - g^2}} \right) \right. \right. \\
 &\quad \left. \left. + f \frac{fg}{1 + \sqrt{1 - f^2 - g^2}} \cos^2 F \right] \frac{u_\vartheta}{m} + f \frac{1 - f \cos F - g \sin F}{\sqrt{1 - f^2 - g^2}} [(fk - gh) + \right. \\
 &\quad \left. + \cos F \left(h \frac{fg}{1 + \sqrt{1 - f^2 - g^2}} - k \left(1 - \frac{g^2}{1 + \sqrt{1 - f^2 - g^2}} \right) \right) + \right. \\
 &\quad \left. \left. + \sin F \left(h \left(1 - \frac{f^2}{1 + \sqrt{1 - f^2 - g^2}} \right) - k \frac{fg}{1 + \sqrt{1 - f^2 - g^2}} \right) \right] \frac{u_h}{m} \right\}
 \end{aligned}$$

$$\begin{aligned}
 \frac{dh}{dt} &= \frac{1}{2} \sqrt{\frac{a}{\mu(1-f^2-g^2)}} (1+h^2+k^2) \left[-f + \cos F \left(1 - \frac{g^2}{1+\sqrt{1-f^2-g^2}} \right) + \right. \\
 &\quad \left. + \sin F \frac{fg}{1+\sqrt{1-f^2-g^2}} \right] \frac{u_h}{m} \\
 \frac{dk}{dt} &= \frac{1}{2} \sqrt{\frac{a}{\mu(1-f^2-g^2)}} (1+h^2+k^2) \left[-g + \sin F \left(1 - \frac{f^2}{1+\sqrt{1-f^2-g^2}} \right) + \right. \\
 &\quad \left. + \cos F \frac{fg}{1+\sqrt{1-f^2-g^2}} \right] \frac{u_h}{m} \\
 \frac{dF}{dt} &= \sqrt{\frac{\mu}{a^3}} \frac{1}{1-f \cos F - g \sin F} + \sqrt{\frac{a}{\mu}} \frac{1}{1-f \cos F - g \sin F} \left\{ [(f \cos F + \right. \\
 &\quad \left. + g \sin F) \left(1 + \frac{1}{1+\sqrt{1-f^2-g^2}} \right) - (2 - \sqrt{1-f^2-g^2})] \frac{u_r}{m} + \right. \\
 &\quad \left. + \left[\left(1 - \frac{2}{1+\sqrt{1-f^2-g^2}} \right) (g \cos F - f \sin F) + \frac{fg}{1+\sqrt{1-f^2-g^2}} (\cos^2 F + \right. \right. \\
 &\quad \left. \left. - \sin^2 F) + \sin F \cos F \left(\left(1 - \frac{f^2}{1+\sqrt{1-f^2-g^2}} \right) - (1 + \right. \right. \right. \\
 &\quad \left. \left. \left. - \frac{g^2}{1+\sqrt{1-f^2-g^2}} \right) \right) \right] \frac{u_\vartheta}{m} + \frac{1-f \cos F - g \sin F}{\sqrt{1-f^2-g^2}} [(fk - gh) + \right. \\
 &\quad \left. + \cos F \left(h \frac{fg}{1+\sqrt{1-f^2-g^2}} - k \left(1 - \frac{g^2}{1+\sqrt{1-f^2-g^2}} \right) \right) + \right. \\
 &\quad \left. + \sin F \left(h \left(1 - \frac{f^2}{1+\sqrt{1-f^2-g^2}} \right) - k \frac{fg}{1+\sqrt{1-f^2-g^2}} \right) \right] \frac{u_h}{m} \Big\} \\
 \frac{dm}{dt} &= - \frac{\sqrt{u_r^2 + u_\vartheta^2 + u_h^2}}{Isp \cdot g_0}
 \end{aligned}$$

Gauss' equation in reference frame $[\hat{\mathbf{t}}, \hat{\mathbf{n}}, \hat{\mathbf{h}}]$ in terms of the true longitude, L :

$$\begin{aligned}
 \frac{da}{dt} &= 2\sqrt{\frac{a^3}{\mu}} \sqrt{\frac{1 + 2f \cos L + 2g \sin L + f^2 + g^2}{1 - f^2 - g^2}} \frac{u_t}{m} \\
 \frac{df}{dt} &= \sqrt{\frac{a}{\mu}} \sqrt{\frac{1 - f^2 - g^2}{1 + 2f \cos L + 2g \sin L + f^2 + g^2}} \left\{ 2(f + \cos L) \frac{u_t}{m} + \right. \\
 &\quad \left. - \left[2g + \frac{\sin L(1 - f^2 - g^2)}{1 + f \cos L + g \sin L} \right] \frac{u_n}{m} \right\} - \sqrt{\frac{a(1 - f^2 - g^2)}{\mu}} g \frac{h \sin L - k \cos L}{1 + f \cos L + g \sin L} \frac{u_h}{m} \\
 \frac{dg}{dt} &= \sqrt{\frac{a}{\mu}} \sqrt{\frac{1 - f^2 - g^2}{1 + 2f \cos L + 2g \sin L + f^2 + g^2}} \left\{ 2(g + \sin L) \frac{u_t}{m} + \right. \\
 &\quad \left. \left[2f + \frac{\cos L(1 - f^2 - g^2)}{1 + f \cos L + g \sin L} \right] \frac{u_n}{m} \right\} + \sqrt{\frac{a(1 - f^2 - g^2)}{\mu}} f \frac{h \sin L - k \cos L}{1 + f \cos L + g \sin L} \frac{u_h}{m} \\
 \frac{dh}{dt} &= \frac{1}{2} \sqrt{\frac{a(1 - f^2 - g^2)}{\mu}} \frac{1 + h^2 + k^2}{1 + f \cos L + g \sin L} \cos L \frac{u_h}{m} \\
 \frac{dk}{dt} &= \frac{1}{2} \sqrt{\frac{a(1 - f^2 - g^2)}{\mu}} \frac{1 + h^2 + k^2}{1 + f \cos L + g \sin L} \sin L \frac{u_h}{m} \\
 \frac{dL}{dt} &= \sqrt{\frac{\mu}{[a(1 - f^2 - g^2)]^3}} (1 + f \cos L + g \sin L)^2 + \\
 &\quad + \sqrt{\frac{a(1 - f^2 - g^2)}{\mu}} \frac{h \sin L - k \cos L}{1 + f \cos L + g \sin L} \frac{u_h}{m} \\
 \frac{dm}{dt} &= -\frac{\sqrt{u_t^2 + u_n^2 + u_h^2}}{Isp \cdot g_0}
 \end{aligned}$$

Gauss' equation in reference frame $[\hat{\mathbf{t}}, \hat{\mathbf{n}}, \hat{\mathbf{h}}]$ in terms of the eccentric longitude, F :

$$\begin{aligned}
 \frac{da}{dt} &= 2\sqrt{\frac{a^3}{\mu}} \sqrt{\frac{1+f\cos F+g\sin F}{1-f\cos F-g\sin F}} \frac{u_t}{m} \\
 \frac{df}{dt} &= \sqrt{\frac{a}{\mu}} \frac{1-f\cos F-g\sin F}{1+f\cos F+g\sin F} \left\{ \left[2\frac{\sqrt{1-f^2-g^2}}{1-f\cos F-g\sin F} (\cos F (1+ \right. \right. \\
 &\quad \left. \left. -\frac{f^2}{1+\sqrt{1-f^2-g^2}}) - \sin F \frac{fg}{1+\sqrt{1-f^2-g^2}} \right] \frac{u_t}{m} - [g+ \right. \\
 &\quad \left. + \cos F \frac{fg}{1+\sqrt{1-f^2-g^2}} + \sin F \left(1 - \frac{f^2}{1+\sqrt{1-f^2-g^2}} \right) \right] \frac{u_n}{m} \right\} + \\
 &\quad - \sqrt{\frac{a}{\mu}} \frac{g}{\sqrt{1-f^2-g^2}} \left\{ (fk - gh) + \cos F \left[h \frac{fg}{1+\sqrt{1-f^2-g^2}} - k (1+ \right. \right. \\
 &\quad \left. \left. -\frac{g^2}{1+\sqrt{1-f^2-g^2}}) \right] + \sin F \left[h \left(1 - \frac{f^2}{1+\sqrt{1-f^2-g^2}} \right) + \right. \right. \\
 &\quad \left. \left. -k \frac{fg}{1+\sqrt{1-f^2-g^2}} \right] \frac{u_h}{m} \right\} \\
 \frac{dg}{dt} &= \sqrt{\frac{a}{\mu}} \frac{1-f\cos F-g\sin F}{1+f\cos F+g\sin F} \left\{ \left[2\frac{\sqrt{1-f^2-g^2}}{1-f\cos F-g\sin F} (\sin F (1+ \right. \right. \\
 &\quad \left. \left. -\frac{g^2}{1+\sqrt{1-f^2-g^2}}) - \cos F \frac{fg}{1+\sqrt{1-f^2-g^2}} \right] \frac{u_t}{m} + [f \right. \\
 &\quad \left. + \sin F \frac{fg}{1+\sqrt{1-f^2-g^2}} + \cos F \left(1 - \frac{g^2}{1+\sqrt{1-f^2-g^2}} \right) \right] \frac{u_n}{m} \right\} + \\
 &\quad + \sqrt{\frac{a}{\mu}} \frac{f}{\sqrt{1-f^2-g^2}} \left\{ (fk - gh) + \cos F \left[h \frac{fg}{1+\sqrt{1-f^2-g^2}} - k (1+ \right. \right. \\
 &\quad \left. \left. -\frac{g^2}{1+\sqrt{1-f^2-g^2}}) \right] + \sin F \left[h \left(1 - \frac{f^2}{1+\sqrt{1-f^2-g^2}} \right) + \right. \right. \\
 &\quad \left. \left. -k \frac{fg}{1+\sqrt{1-f^2-g^2}} \right] \frac{u_h}{m} \right\} \\
 \frac{dh}{dt} &= \frac{1}{2} \sqrt{\frac{a}{\mu(1-f^2-g^2)}} (1+h^2+k^2) \left[-f + \cos F \left(1 - \frac{g^2}{1+\sqrt{1-f^2-g^2}} \right) + \right. \\
 &\quad \left. \sin F \frac{fg}{1+\sqrt{1-f^2-g^2}} \right] \frac{u_h}{m}
 \end{aligned}$$

$$\begin{aligned}
\frac{dk}{dt} &= \frac{1}{2} \sqrt{\frac{a}{\mu(1-f^2-g^2)}} (1+h^2+k^2) \left[-g + \sin F \left(1 - \frac{f^2}{1 + \sqrt{1-f^2-g^2}} \right) + \right. \\
&\quad \left. + \cos F \frac{fg}{1 + \sqrt{1-f^2-g^2}} \right] \frac{u_h}{m} \\
\frac{dF}{dt} &= \sqrt{\frac{\mu}{a^3}} \frac{1}{1-f \cos F - g \sin F} - \sqrt{\frac{a}{\mu}} \sqrt{\frac{1-f \cos F - g \sin F}{1+f \cos F + g \sin F}} \frac{1}{1 + \sqrt{1-f^2-g^2}} \{ [\\
&\quad \frac{2}{1-f \cos F - g \sin F} (f \sin F - g \cos F)] \frac{u_t}{m} + \\
&\quad - \left(1 + \sqrt{1-f^2-g^2} + f \cos F + g \sin F \right) \frac{u_n}{m} \} + \sqrt{\frac{a}{\mu(1-f^2-g^2)}} \{ (fk - gh) + \\
&\quad + \cos F \left[h \frac{fg}{1 + \sqrt{1-f^2-g^2}} - k \left(1 - \frac{g^2}{1 + \sqrt{1-f^2-g^2}} \right) \right] + \\
&\quad + \sin F \left[h \left(1 - \frac{f^2}{1 + \sqrt{1-f^2-g^2}} \right) - k \frac{fg}{1 + \sqrt{1-f^2-g^2}} \right] \} \frac{u_h}{m} \\
\frac{dm}{dt} &= - \frac{\sqrt{u_t^2 + u_n^2 + u_h^2}}{Isp \cdot g_0}
\end{aligned}$$

APPENDIX B

Cardan's method to solve quartic equations

In this appendix the general analytical solution to a quartic equation is presented. Such procedure is applied to Eq. (7.19) to get the solution representing the entry and exit points for the umbra and penumbra regions. The general form of the quartic equation is:

$$C_1x^4 + C_2x^3 + C_3x^2 + C_4x + C_5 = 0 \quad (\text{B.1})$$

The first step is to rewrite the quartic equation so that the first term has a unit coefficient:

$$\begin{aligned} x^4 + \alpha x^3 + \beta x^2 + \gamma x + \delta &= 0 \\ \alpha &= C_2/C_1 \\ \beta &= C_3/C_1 \\ \gamma &= C_4/C_1 \\ \delta &= C_5/C_1 \end{aligned} \quad (\text{B.2})$$

Define the variable h as:

$$h = -\frac{\alpha}{4} \quad (\text{B.3})$$

Rewrite the quartic equation in a polynomial form that lacks the x^3 term:

$$\begin{aligned} x^4 + Px^2 + Qx + R &= 0 \\ P &= 6h^2 + 3\alpha h + \beta \\ Q &= 4h^3 + 3\alpha h^2 + 2\beta h + \gamma \\ R &= h^4 + \alpha h^3 + \beta h^2 + \gamma h + \delta \end{aligned} \quad (\text{B.4})$$

The solution to the equation depends on the Q parameter. Indeed, if $Q = 0$ the quartic equation becomes a bi-quadratic equation that can be solved by substitution of variables setting $y = x^2$. If the parameter Q is different from zero, the following cubic equation is solved:

$$\begin{aligned}
t^3 + ut^2 + vt + w &= 0 \\
u &= 2P \\
v &= P^2 - 4R \\
w &= -Q^2
\end{aligned} \tag{B.5}$$

The solution to the cubic equation is obtained applying Cardan's solution which writes the cubic equation in a "depressed" form that lacks the t^2 term and applying a substitution of variable.

$$\begin{aligned}
Z^3 + mZ + n &= 0 \\
t &= Z - \frac{u}{3} \\
m &= \frac{1}{3}(3v - u^2) \\
n &= \frac{1}{27}(2u^3 - 9uv + 27w)
\end{aligned} \tag{B.6}$$

A discriminant, Δ is introduced to take into account the number of real solutions and complex solutions associated with the cubic equation.

$$\Delta = \frac{m^3}{27} + \frac{n^2}{4} \tag{B.7}$$

The solution to the cubic equation are different according to the sign and value of the discriminant, Δ . If the discriminant is equal to 0 the cubic equation admits 3 real roots computed as follows:

$$\begin{aligned}
Z_1 &= 2\sqrt[3]{-\frac{n}{2}} \\
Z_2 &= \sqrt[3]{\frac{n}{2}} \\
Z_3 &= Z_2
\end{aligned} \tag{B.8}$$

If the discriminant is negative the cubic equation will also admit 3 real solutions that are computed using the following expressions:

$$\begin{aligned}
E_0 &= 2\sqrt{-\frac{m}{3}} \\
\cos \phi &= -\frac{n}{2\sqrt{-\frac{m^3}{27}}} \\
\sin \phi &= \sqrt{1 - \cos^2 \phi} \\
\phi &= \text{atan2}(\sin \phi, \cos \phi) \\
Z_1 &= E_0 \cos\left(\frac{\phi}{3}\right) \\
Z_2 &= E_0 \cos\left(\frac{\phi}{3} + \frac{2}{3}\pi\right) \\
Z_3 &= E_0 \cos\left(\frac{\phi}{3} + \frac{4}{3}\pi\right)
\end{aligned} \tag{B.9}$$

The last case is when the discriminant is positive. In this case the cubic equation admits one real solution and two complex solutions.

$$\begin{aligned}
Z_1 &= \sqrt[3]{-\frac{n}{2} + \sqrt{\Delta}} + \sqrt[3]{-\frac{n}{2} - \sqrt{\Delta}} \\
Z_2 &= -\frac{Z_1}{2} + \frac{\sqrt{-3}}{2} \left(\sqrt[3]{-\frac{n}{2} + \sqrt{\Delta}} + \sqrt[3]{-\frac{n}{2} - \sqrt{\Delta}} \right) \\
Z_3 &= -\frac{Z_1}{2} - \frac{\sqrt{-3}}{2} \left(\sqrt[3]{-\frac{n}{2} + \sqrt{\Delta}} + \sqrt[3]{-\frac{n}{2} - \sqrt{\Delta}} \right)
\end{aligned} \tag{B.10}$$

Once the depressed cubic equation is solved and the original roots t_1 , t_2 and t_3 are obtained, the solutions of the quartic equation are computed as follows:

$$\begin{aligned}
\xi &= \frac{1}{2} \left(P + \max[t_1, t_2, t_3] - \frac{Q}{\sqrt{\max[t_1, t_2, t_3]}} \right) \\
\zeta &= \frac{1}{2} \left(P + \max[t_1, t_2, t_3] + \frac{Q}{\sqrt{\max[t_1, t_2, t_3]}} \right) \\
k^2 + \sqrt{\max[t_1, t_2, t_3]}k + \xi &= 0 \\
k^2 - \sqrt{\max[t_1, t_2, t_3]}k + \zeta &= 0 \\
x_i &= k_i + h
\end{aligned} \tag{B.11}$$