

**POLITECNICO DI TORINO**

Dipartimento di Ingegneria Meccanica e Aerospaziale

**Corso di Laurea Magistrale  
in Ingegneria Aerospaziale**

Tesi di Laurea Magistrale

**Design and Optimization of Low Thrust  
Maneuvers in Interplanetary Missions**



**Relatore**

Prof. Casalino Lorenzo

**Tutor Aziendali**

Ing. Cosimo Chiarelli

Ing. Marco Berga

**Candidato**

*Nicola Savaglia*

Marzo 2018



# Indice

<b>Abstract</b>	<b>3</b>
English Version	3
Versione italiana	5
<b>Lista delle Figure</b>	<b>7</b>
<b>Lista delle Tabelle</b>	<b>9</b>
<b>Lista delle Abbreviazioni</b>	<b>10</b>
<b>1 Introduzione</b>	<b>11</b>
1.1 Thales Alenia Space	11
1.2 Introduzione ai vantaggi della Low Thrust	17
1.3 Space Mission and Electric Propulsion	18
1.4 Classificazione Low Thrust Propulsor	22
1.5 Low Thrust Trajectory Tools	26
<b>2 Elementi di Meccanica Orbitale</b>	<b>27</b>
2.1 Problema dei due corpi	27
2.2 Approssimazione Patched Conic	30
2.3 Perturbazioni	32
2.4 Analogie tra Low Thrust ed accelerazione perturbativa	37
2.5 Metodi di modellizzazione	39
2.6 Scelta del modello	41
2.7 Introduzione all'ottimizzazione – Risoluzione di problemi sovradimensionati	48
<b>3 Algoritmi di ottimizzazione</b>	<b>49</b>
3.1 Overview	49
3.2 Formulazione matematica generale	50
3.3 Metodo Indiretto	51

---

<b>3.4</b>	<b>Metodo Diretto</b>	<b>56</b>
3.4.1	Equality constraint	60
3.4.2	Inequality constraint	62
<b>3.5</b>	<b>Metodo di ottimizzazione utilizzato – SQP Method</b>	<b>64</b>
<b>4</b>	<b><i>Applicazione alla missione Mars Sample Return</i></b>	<b>68</b>
<b>4.1</b>	<b>Introduzione codice</b>	<b>68</b>
4.1.1	Formulazione matematica DOPLOT	69
4.1.2	Parametri della modellizzazione	71
4.1.3	Output DOPLOT	72
<b>4.2</b>	<b>Parametri Metodo SQP</b>	<b>73</b>
4.2.1	Analisi adimensionale	75
4.2.2	StepTolerance	77
4.2.3	ConstraintTolerance	80
4.2.4	OptimalityTolerance	81
4.2.5	Numero archi	84
<b>4.3</b>	<b>SEP – Solar Electric Propulsion</b>	<b>86</b>
<b>4.4</b>	<b>Validazione</b>	<b>91</b>
<b>5</b>	<b><i>Conclusione e miglioramenti futuri</i></b>	<b>101</b>
<b>6</b>	<b><i>Bibliografia</i></b>	<b>104</b>

# Abstract

## English Version

My final thesis is entitled “Design and Optimization of low thrust maneuvers in interplanetary missions” and it has been drafted during my internship in Thales Alenia Space Italia. Main point of the thesis is the creation of a tool that designs and optimizes the low thrust maneuvers. The software, which I named DOpLoTT – Design and Optimization of Low-Thrust Trajectory can analyse one interplanetary mission about two given planets or target orbit elements and ToF (departure and arrival date assigned).

Nowadays, the low-thrust maneuvers are always considered in the primary phase of the space mission project. After the first missions like NASA’s Deep Space 1 or ESA’s SMART-1 that have shown the effectiveness of low-thrust systems as primary propulsion devices, the electric propulsion has been improving and the advantages of this use are higher than in the past. This explains why a lot of missions like Hayabusa, LISA and Mars Sample Return have used (or will use) the low thrust as primary propulsion. For this reason, in the team of mission analysis, it is always more useful to own a tool that analyses the infinity world of the low-thrust trajectory.

Designing the trajectory is very important in the low-thrust maneuvers because the characteristic of them is that the thrust vector is continuous over time (not impulsive) and variable instant by instant (in module and direction). For this reason, the trajectory was divided in a finite number of arches, so that it is possible to analyse each arch with the approximation of Keplerian model.

DOPLOTT exploits the optimization tool of Matlab. In particular, it has been used the function *fmincon* that implements some direct method of optimization. In addition, it was chosen the Sequential Quadratic Programming that allows to calculate the minimum of a function in presence of equality and inequality constraints. DOPLOTT obtains the trend of the thrust in each arch and the linked total DV that will be the lowest between all trajectories that respect the constraints. The constraints are the state vectors of the target planet, and the maximum values of the thruster parameters.

Moreover, it is possible to set the type of thruster between SEP – Solar Electric Propulsion (depending on the distance to the Sun) and NEP – Nuclear Electric Propulsion (independent on the distance to the Sun). In this way when the SEP is chosen, the constraints of the maxim value of thrust change in relation of the distance between S/C and Sun instant by instant.

The tool has been validated with the Mission Analysis Guidelines of Mars Sample Return. DOPLOTT has calculated the trajectories analysed by the software DITAN of ESA with a velocity error of about  $\pm 3$  m/s and propellant mass error of about  $\pm 0.3$  kg.

In conclusion, it has been optimized the computational cost reducing the duration of the analysis. In order to obtain that, it has been used the dimensionless parameters, and carefully chosen the setting of *fmincon* function with some options like optimality tolerance, constraint tolerance and other.

## Versione italiana

Questa tesi, dal titolo "Progettazione e ottimizzazione di manovre a bassa spinta in ambito di missioni interplanetarie", è stata redatta in Thales Alenia Space Italia durante il periodo di stage. Il principale risultato raggiunto durante questo lavoro di tesi è stato la creazione di uno strumento che progetta e ottimizza le manovre a bassa spinta. Il software, denominato DOpLoTT - Design and Optimization of Low-Thrust Trajectory, permette di analizzare una missione interplanetaria tra due determinati pianeti o punti del sistema solare definiti da precisi parametri orbitali e in un definito ToF (data di partenza e arrivo assegnata).

Al giorno d'oggi, l'utilizzo di una propulsione primaria elettrica è sempre tenuto in considerazione nelle fasi iniziali di progetto delle missioni spaziali. Dopo le prime missioni come Deep Space 1 della NASA o SMART-1 dell'ESA che hanno dimostrato l'efficacia dei sistemi a bassa spinta come dispositivi di propulsione primaria, le prestazioni di un propulsore elettrico sono migliorate, ed i benefici di questo utilizzo sono più consistenti rispetto al passato. Questo spiega perché molte missioni come Hayabusa, LISA e Mars Sample Return hanno usato (o useranno) la bassa spinta come propulsione primaria. Per questo motivo, nei team di Mission Analysis, è sempre più utile possedere uno strumento che analizzi l'infinito mondo della traiettoria a bassa spinta.

Un'accurata modellizzazione della traiettoria è molto importante nelle manovre a bassa spinta. In queste manovre, il vettore di spinta è continuo nel tempo (non impulsivo) e variabile istante per istante (nel modulo e nella direzione). Per questo motivo, la traiettoria è stata divisa in un numero finito di archi, in modo che sia possibile analizzare ogni arco con l'approssimazione del modello Kepleriano.

DOPLOTT sfrutta lo strumento di ottimizzazione presente in Matlab. In particolare, è stata utilizzata la funzione *fmincon* che implementa alcuni algoritmi di ottimizzazione tra la classe dei metodi denominati metodi diretti. In particolare, è stato utilizzato il metodo SQP – Sequential Quadratic Programming che consente di calcolare il minimo di una funzione in presenza di vincoli non lineari. DOPLOTT ricava il profilo di spinta ad ogni arco e il  $\Delta V$  totale che sarà il più basso tra tutte le traiettorie che rispettano i vincoli. Questi ultimi sono i vettori di stato del pianeta target e i valori massimi dei parametri del propulsore.

Inoltre, è possibile impostare il tipo di propulsore utilizzato tra un SEP – Solar Electric Propulsion (il cui modulo della spinta è funzione della distanza dal Sole) e un NEP – Nuclear Electric Propulsion (indipendente dalla distanza dal Sole). La scelta del SEP, produce una variazione della spinta massima che il propulsore può fornire ad ogni singolo arco.

DOPLOTT è stato convalidato grazie ai dati ricavati dal MAG ESA della missione Mars Sample Return. Con il suddetto software sono state confrontate le traiettorie ottimizzate in

output, con quelle analizzate dal software DITAN dell'ESA. Con questo confronto è stato riscontrato un errore tra le due analisi di  $\pm 3 \text{ m/s}$  sul  $\Delta V$  complessivo dell'intera manovra e un errore sulla massa del propellente utilizzato di  $\pm 0.3 \text{ kg}$ .

In conclusione, è stato ottimizzato il costo computazionale del software, ottenendo una riduzione della durata delle analisi. Per fare ciò, dopo aver adimensionalizzato il modello, è stata scelta con cura l'impostazione della funzione *fmincon*, ad esempio modificando la scelta delle tolleranze con una modifica di alcuni parametri come il *OptimalityTolerance* o il *ConstraintTolerance*.

## Lista delle Figure

Figura 1.1 – Logo azienda Thales Alenia Space	11
Figura 1.2 – Cargo Cygnus	12
Figura 1.3 – Astronauta S. Cristoforetti all'interno del modulo Cupola	13
Figura 1.4 – Veicolo spaziale IXV in esposizione (aeroporto di Torino)	13
Figura 1.5 – EXOMARS: Ricostruzione virtuale rilascio lander Schiaparelli	14
Figura 1.6 – BepiColombo: ricostruzione virtuale avvicinamento <sup>6</sup>	15
Figura 1.7 – Complessa traiettoria a bassa spinta, missione BepiColombo	16
Figura 1.8 – Missione Dawn (software MYSTIC): simulazione orbitale	20
Figura 1.9 – Configurazione Arcogetto	23
Figura 1.10 – Configurazione propulsori ad effetto Hall	24
Figura 1.11 – Propulsore NSTAR	25
Figura 1.12 – Struttura della traiettoria nelle	26
Figura 2.1 – Differenti orbite in relazione alla velocità di immissione	28
Figura 2.2 – Parametri orbitali	29
Figura 2.3 – Sistema di riferimento RTN	34
Figura 2.4 – Schema a blocchi algoritmo che sfrutta le Lagrange's Planetary Equations	35
Figura 2.5 – Modellizzazione della traiettoria nel Solver Integral	39
Figura 2.6 – Modellizzazione della traiettoria nel Solver Impulsive Node	40
Figura 2.7 – Modellizzazione della traiettoria nel Solver Impulsive Middle	41
Figura 2.8 – Confronto risolutori: modulo posizione (in alto) e modulo velocità (in basso)	42
Figura 2.9 – Confronto risolutori: durata delle analisi	43
Figura 2.10 – Confronto risolutori impulsivi: traiettoria	44
Figura 2.11 – Confronto risolutori impulsivi: durata delle analisi	45
Figura 2.12 – Confronto risolutori impulsivi: differenza tra i moduli della velocità	46
Figura 2.13 – Modello arco <i>i</i> -esimo	47
Figura 2.14 – Vettore della spinta nel sistema di riferimento RTN	47
Figura 3.1 – Discretizzazione vs. formulazione continua	56
Figura 3.2 – Metodo di Newton: rappresentazione geometrica	57
Figura 3.3 – Multiple Shooting	65
Figura 4.1 – Componenti del vettore <i>i</i> -esimo della spinta	69
Figura 4.2 – Iterazioni e StepTolerance	73
Figura 4.3 – Variazione del numero di archi	85
Figura 4.4 – Curva caratteristica di un generico SEP	87
Figura 4.5 – SEP: confronto tra le due implementazioni	89
Figura 4.6 – Output Validation Case: software DITAN - ESA	93
Figura 4.7 – Output Validation Case: traiettoria	94
Figura 4.8 – Output Validation Case: andamento distanza dal Sole	95
Figura 4.9 – Output Validation Case: profilo della spinta	95
Figura 4.10 – Output Validation Case: andamento delle direzioni della spinta	96
Figura 4.11 – Output analisi NEP: traiettoria	99

<i>Figura 4.12 – Output analisi NEP: profilo della spinta</i>	99
<i>Figura 4.13 – Output analisi NEP: andamento direzioni della spinta</i>	100
<i>Figura 4.14 – Output analisi NEP: andamento distanza dal Sole</i>	100
<i>Figura 5.1 – Analisi dati metodo iterativo (primi risultati DOPLOTT)</i>	101
<i>Figura 5.2 – Esempio di Pork Chop manovre impulsive</i>	102

## Lista delle Tabelle

<i>Tabella 2.1 – Parametri orbitali classici</i>	30
<i>Tabella 2.2 – SOI pianeti del sistema solare</i>	31
<i>Tabella 4.1 – Analisi dimensionale vs. analisi adimensionale</i>	76
<i>Tabella 4.2 – ExitFlag</i>	77
<i>Tabella 4.3 – Analisi al variare dello StepTolerance</i>	79
<i>Tabella 4.4 – ConstraintTolerance ed errori associati</i>	80
<i>Tabella 4.5 – Analisi al variare del ConstraintTolerance</i>	81
<i>Tabella 4.6 – Analisi al variare dello OptimalityTolerance (ConstraintTolerance <math>10^{-5}</math>)</i>	82
<i>Tabella 4.7 – Analisi al variare dello OptimalityTolerance (ConstraintTolerance <math>10^{-4}</math>)</i>	83
<i>Tabella 4.8 – Settaggio Validation Case</i>	91
<i>Tabella 4.9 – Input Validation Case</i>	91
<i>Tabella 4.10 – Risultati Validation Case: software DITAN - ESA</i>	92
<i>Tabella 4.11 – Risultati Validation Case: software DOPLOTT</i>	94
<i>Tabella 4.12 – Validation Case: errori sui vettori di stato</i>	97
<i>Tabella 4.13 – Validation Case: confronto</i>	97
<i>Tabella 4.14 – Analisi NEP</i>	98

## Lista delle Abbreviazioni

<b>S/C</b>	Spacecraft
<b>ESA</b>	European Space Agency
<b>NASA</b>	National Aeronautics and Space Administration
<b>JAXA</b>	Japan Aerospace eXploration Agency
<b>ASI</b>	Agenzia Spaziale Italiana
<b>TAS</b>	Thales Alenia Space
<b>TASI</b>	Thales Alenia Space Italia
<b>RTN</b>	Radial Tangenzial Normal
<b>DAE</b>	Differential Algebraic Equations
<b>NLP</b>	Non Linear Programming
<b>LP</b>	Linear Programming
<b>QP</b>	Quadratic Programming
<b>KT</b>	Kuhn – Tucker system
<b>KKT</b>	Karush – Kuhn – Tucher system
<b>SQP</b>	Sequential Quadratic Programming
<b>BFGS</b>	Broyden – Fletcher – Goldfarb – Shanno algorithm
<b>DOPLOTT</b>	Design and Optimization of Low – Thrust Trajectory
<b>MAG</b>	Mission Analysis Guidelines
<b>MSR</b>	Mars Sample Return mission
<b>SOI</b>	Sphere Of Influence
<b>ToF</b>	Time of Flight
<b>NAIF</b>	Navigation and Ancillary Information Facility
<b>JPL</b>	Jet Propulsion Laboratory
<b>AU</b>	Astronomical Units
<b>SEP</b>	Solar Electric Propulsion
<b>NEP</b>	Nuclear Electric Propulsion

# 1 Introduzione

## 1.1 Thales Alenia Space

“Thales Alenia Space believes in space as humankind’s new horizon, which will enable to build a better, more sustainable life on Earth“<sup>1</sup>- Thales Alenia Space crede nello spazio come nuovo orizzonte dell'umanità, che permetterà di costruire una vita migliore e più sostenibile sulla Terra.



*Figura 1.1 – Logo azienda Thales Alenia Space*

Da una veloce ricerca online per acquisire informazioni su Thales Alenia Space, questa è la frase che mi ha maggiormente entusiasmato. Ho avuto la fortuna di trascorrere tutto il mio periodo di ricerca nella sede torinese e penso sia la frase che più di tutte rispecchia l'azienda e la passione che i numerosi dipendenti mettono nel proprio lavoro, consapevoli dell'utilità che da esso scaturisce.

Nella mia esperienza da tesista non ho sicuramente avuto contatti con tutte le realtà aziendali, ma quello che ho notato lavorando con il team di Mission Analysis è la condivisione dell'idea dello spazio come il futuro di ogni miglioramento sul pianeta Terra. Credere in questa affermazione penso abbia creato un ambiente interno molto produttivo, permettendo negli anni di realizzare progetti all'avanguardia, diventando leader internazionale per la realizzazione di infrastrutture orbitanti, sistemi satellitari adibiti alla telecomunicazione, navigazione, ma anche all'osservazione e monitoraggio del pianeta Terra. La società inoltre collabora con le maggiori industrie spaziali internazionali (fra tutte l'ESA, la NASA e l'ASI) nei progetti di missioni per esplorazione spaziale.

---

<sup>1</sup> Estratto dal sito web di Thales Group nella sezione Space <https://www.thalesgroup.com/en/global/activities/space>

Nel 1990, dalla fusione di Aeritalia Sistemi Spaziali e Selenia Spazio, nasce la Alenia Spazio, società del gruppo Finmeccanica, specializzata nel realizzare componenti meccanici ed elettronici per missioni spaziali. Nel 2005 la Alenia Spazio si unì con l'azienda francese Alcatel Space dando vita a una nuova società denominata Alcatel Alenia Space. Il 5 aprile 2006 Alcatel decise di vendere la sua partecipazione in Alcatel Alenia Space al gruppo Thales. L'Unione Europea diede l'autorizzazione definitiva all'operazione il 10 aprile 2007. Nasce quindi una joint venture tra Thales con il 67% e Leonardo (gruppo Finmeccanica) con il 33%.

Negli anni la suddetta azienda ha ottenuto enormi successi. Il progetto Cygnus è uno di questi, tutt'ora attivo, prevede la realizzazione di veicoli di rifornimento, privi della capacità di rientro atmosferico, che dal 2011 (anno in cui venne ritirato dal servizio lo Space Shuttle) trasporta rifornimenti alla ISS.



*Figura 1.2 – Cargo Cygnus<sup>2</sup>*

Sempre riguardo la Stazione Spaziale Internazionale, TAS ne vanta la realizzazione di oltre la metà del volume pressurizzato come ad esempio la costruzione dei moduli della stazione denominati Harmony (Node 2) e Tranquility (Node 3) e il modulo europeo Columbus, oltre alla struttura di osservazione esterna nota come Cupola.

---

<sup>2</sup> Fonte Thales Alenia Space <https://www.thalesgroup.com>



*Figura 1.3 – Astronauta S. Cristoforetti all'interno del modulo Cupola<sup>3</sup>*

Thales Alenia Space ha sfruttato la sua ineguagliata esperienza nelle infrastrutture orbitali e nel trasporto spaziale per affrontare molte sfide, tra cui il veicolo spaziale IXV, lanciato nel febbraio 2015, la cui missione consisteva nel verificare la capacità di compiere un rientro atmosferico controllato da orbita terrestre; è stato un enorme successo che aprirà la strada a future missioni di rientro in Europa.

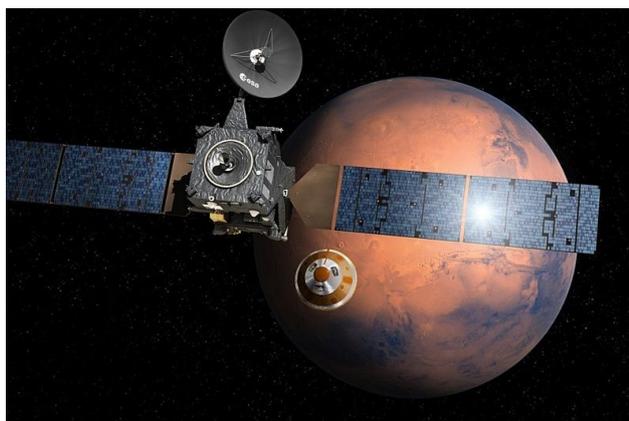


*Figura 1.4 – Veicolo spaziale IXV in esposizione (aeroporto di Torino)<sup>4</sup>*

<sup>3</sup> Fonte NASA [https://ares.jsc.nasa.gov/image\\_science/](https://ares.jsc.nasa.gov/image_science/)

<sup>4</sup> Fonte ANSA <http://www.ansa.it>

Ma non si tratta solo di infrastrutture orbitali, oggi TASI è la Prime Contractor del progetto ExoMars. Si tratta di un progetto composto da due missioni, entrambe con l'obiettivo di cercare biotracce su Marte, con la possibilità quindi di confermare definitivamente la presenza di vita sul Pianeta Rosso. La prima venne lanciata il 14 marzo 2016 e fu un parziale successo, poiché il lander Schiaparelli andò distrutto il 19 ottobre 2016 nel tentativo di posarsi sulla superficie di Marte. Il Trace Gas Orbiter (TGO), dotato di strumenti per l'analisi dei gas atmosferici e la mappatura delle loro fonti e da cui venne sganciato il lander, venne inserito in orbita marziana il 19 ottobre 2016 ed è tutt'ora in orbita circolare a 400km di quota. Nella missione 2020, la joint venture italo-francese è incaricata di sviluppare il sistema di navigazione e guida per i moduli orbitali e di discesa, nonché la progettazione del rover marziano dotato di un laboratorio di analisi in loco. Questo laboratorio è dotato di un perforatore, in grado di perforare due metri di profondità nel suolo marziano e di rimuovere i campioni.



*Figura 1.5 – EXOMARS: Ricostruzione virtuale rilascio lander Schiaparelli <sup>5</sup>*

Come per ExoMars, Thales Alenia Space ebbe un ruolo di guida nella recente missione Rosetta (da cui il nome della sonda) con lo scopo di effettuare l'atterraggio sulla cometa 67P/Churyumov-Gerasimenko, ad opera del lander Philae, ed effettuare studi sulla composizione. Nello specifico TAS ebbe in carico l'assemblaggio, l'integrazione e il collaudo dell'intera sonda Rosetta.

Tra le future missioni in cui Thales Alenia Space sta dando il proprio contributo si può annoverare la missione BepiColombo. Si tratta di una missione del programma di esplorazione di Mercurio da parte dell'ESA in collaborazione con l'agenzia spaziale giapponese JAXA. La missione si compone di due sonde distinte che opereranno autonomamente una volta giunta in orbita di Mercurio (oltre a un modulo che condurrà a

---

<sup>5</sup> Fonte Intelligent Aerospace <http://www.intelligent-aerospace.com>

destinazione la missione): la Mercury Planetary Orbiter (MPO) progettata e costruita sotto la guida dell'Agenzia Spaziale Europea (ESA), la Mercury Magnetospheric Orbiter (MMO) progettata e costruita sotto la guida dell'Agenzia Spaziale Giapponese (JAXA) e il modulo di trasferimento denominato Mercury Transfer Module (MTM) utile unicamente durante la fase di viaggio e per l'inserimento in orbita.

Nello specifico TAS ha realizzato parti tecnologiche di rilievo, come l'antenna ad alto guadagno per le comunicazioni verso Terra, oltre al trasponditore in banda X, il computer



**Figura 1.6** – *BepiColombo: ricostruzione virtuale avvicinamento*<sup>6</sup>

di bordo, la memoria di massa e l'esperimento di Radio Scienza e sarà inoltre responsabile della fase di assemblaggio, integrazione e test. Per la cronaca, si stima che il lancio avverrà entro la fine del 2018, e prevede l'utilizzo di una propulsione solare elettrica, che porterà lo spacecraft a destinazione entro il dicembre 2025 sfruttando per ben nove volte nel suo percorso la spinta gravitazionale della Terra, di Venere (due gravity assists) e di Mercurio (sei gravity assists).

La prossima sfida scientifica è il programma europeo Euclid, progettato per aiutare la nostra comprensione della materia oscura. Thales Alenia Space Italia ricopre il ruolo di primo contraente del satellite per conto dell'ESA ed è a capo di un consorzio per la sua realizzazione. Qualità e stabilità di forma del sistema ottico, velocità e completezza del rilevamento, puntamento accurato e stabile nonché affidabilità del sistema di trasmissione dati, sono i requisiti fondamentali del progetto proposto da Thales Alenia Space Italia. Sulla base dell'esperienza acquisita con la piattaforma di Herschel-Planck (formata da due dei più grandi osservatori spaziali costruiti in Europa nel 2009, per i quali TAS ne fu responsabile), che ha dimostrato eccellenti prestazioni in orbita, la società italo-francese integrerà anche un innovativo sistema di puntamento, agile e preciso, e telecomunicazioni di nuova generazione. La missione, che ha l'obiettivo di studiare l'origine dell'espansione dell'Universo, effettuerà la mappatura ad alta precisione del cielo nelle bande del visibile e del vicino infrarosso.

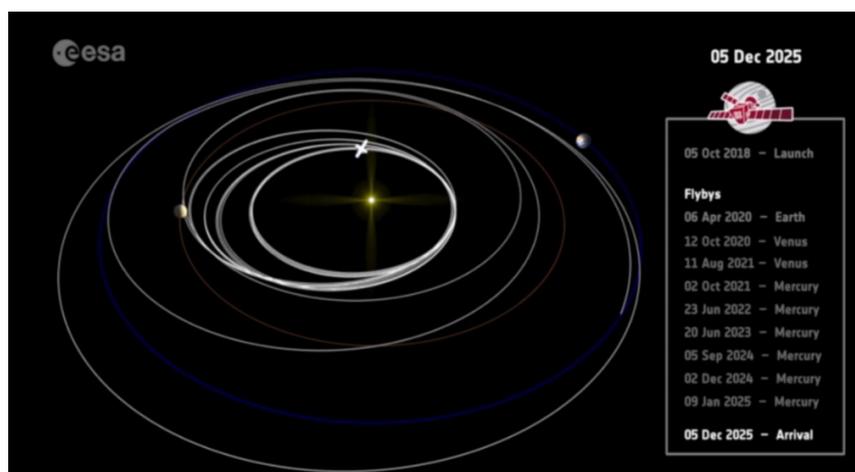
Il satellite, che verrà lanciato nel 2020, sarà dotato di un telescopio di 1.2 metri di diametro, e con i dati ricavati, sarà possibile misurare la struttura su larga scala dell'Universo. Verranno così studiate l'energia e la materia oscura, ingredienti fondamentali ma ignoti

<sup>6</sup> Fonte ESA <http://sci.esa.int/bepicolombo>

dell'odierno "Modello Standard" della cosmologia fisica. In questo modello, solo il 4% della massa ed energia dell'universo è fatto di materia ordinaria (protoni e neutroni). Il resto è invisibile (materia oscura), e l'universo stesso si sta espandendo a ritmo accelerato sotto l'azione di una fonte di energia finora sconosciuta (energia oscura).

Da sottolineare inoltre l'investimento che Thales Alenia Space effettua ogni anno sulla ricerca scientifica, come si nota dalla recente creazione del team di Mission Analysis. Ogni anno vengono infatti investiti fondi nella ricerca in tale ambito per la creazione di tools utili alla realizzazione di missioni interplanetarie (e non solo). Lo scopo è quello di rendere l'azienda autonoma a 360° su quello che concerne la realizzazione di qualsiasi missione spaziale, avendo la possibilità di realizzare studi su traiettorie e analisi orbitali che fino a qualche anno fa erano sempre state demandate ad esterni lasciando ai dipendenti di Thales Alenia Space il merito di verificarne la validità.

Tuttavia, oggi sono sempre più le missioni in cui viene valutata la possibilità di utilizzare propulsione primaria a bassa spinta, che come verrà spiegato nel proseguo della tesi permette di migliorarne alcune prestazioni a discapito di altre. Su questo ambito l'azienda ha deciso di investire, ed è così che è nata per me la possibilità di svolgere questo lavoro di tesi che verterà sulla realizzazione di uno strumento utile a modellizzare ed ottimizzare traiettorie a bassa spinta in ambito di missioni interplanetarie.



**Figura 1.7** – Complessa traiettoria a bassa spinta, missione BepiColombo <sup>7</sup>

<sup>7</sup> Fonte ESA <http://sci.esa.int/bepicolombo>

## 1.2 Introduzione ai vantaggi della Low Thrust

L'utilizzo delle manovre a bassa spinta è di notevole impatto sul design della missione. Ma cosa si intende per bassa spinta? Secondo la definizione estrapolata da un testo accademico [2], per indicare una "bassa spinta" si fa riferimento al rapporto spinta\peso. Questo metodo permette di classificare, almeno approssimativamente, l'entità della spinta fornita dal propulsore. Sia  $W_s$  il peso locale dello S/C calcolato secondo la relazione:

$$W_s = m \frac{\mu}{r^2} \quad (1.1)$$

Con  $m$  la massa istantanea del velivolo,  $r$  la distanza dal centro di massa del corpo principale e  $\mu$  il parametro gravitazione del corpo principale.

Sia  $T$  il modulo della spinta massima che può fornire il propulsore in analisi, si parlerà di bassa spinta quando:

$$\frac{T}{W_s} \leq 10^{-5} \quad (1.2)$$

In tali condizioni le analisi in gioco assegnate al team di Mission Analysis risultano notevolmente più complesse in quanto cade l'ipotesi di orbite kepleriane in presenza di manovre impulsive<sup>8</sup>.

La propulsione elettrica è un tipico esempio di propulsione a bassa spinta. L'utilizzo di manovre a bassa spinta determina un aumento sostanziale della durata della missione, in quanto per fornire un analogo incremento di energia (o analogamente un  $\Delta V$ ) rispetto a manovre impulsive serviranno manovre continue più durature nel tempo. L'utilizzo del propulsore a bassa spinta in modo continuo crea una variazione infinitesima dei parametri orbitali istante per istante; ciò spiega la caratteristica traiettoria a spirale delle manovre a bassa spinta.

Di contro, viene migliorata una delle limitazioni presente in qualsiasi missione spaziale e in particolare nelle missioni interplanetarie, nonché la massa utile al lancio. In generale i propulsori elettrici sono caratterizzati da impulsi specifici notevolmente elevati, ciò permette a parità di missione di ridurre notevolmente la massa di propellente utile a realizzarla. È possibile dimostrare tale osservazione mediante un'attenta analisi dell'equazione del razzo di Tsiolkovsky:

$$\Delta v = g_0 I_{sp} \ln \left( \frac{m_0}{m_f} \right) \quad (1.3)$$

---

<sup>8</sup> Verrà approfondito l'argomento nei capitoli successivi.

Con  $I_{sp}$  l'impulso specifico del propulsore in analisi,  $g_0 I_{sp} = c$  nonchè la velocità efficace di scarico tipica del propulsore,  $m_0$  la massa iniziale al tempo  $t_0$ ,  $m_f$  la massa finale al tempo  $t_f$  e  $\Delta v$  la variazione di velocità ottenibile con tale propulsore attivo per un intervallo di tempo  $t_f - t_0$ .

Rielaborando la relazione esplicitando le masse (forma più adatta ai fini suddetti), si ottiene:

$$m_f = m_i \left[ 1 - e^{\left( \frac{-\Delta v}{g_0 I_{sp}} \right)} \right] \quad (1.4)$$

Sia  $m_i - m_f = m_p$  con  $m_p$  la massa di propellente a bordo; è noto che a parità di missione e quindi di  $\Delta v$ , avere un elevato valore di  $I_{sp}$  permette di minimizzare la massa di propellente a bordo.

### 1.3 Space Mission and Electric Propulsion

L'utilizzo di propulsori a bassa spinta per le missioni spaziali interplanetarie è ancora oggi una possibilità spesso scartata a causa delle complessità che ne derivano. Tuttavia, ad oggi sono numerosissime le missioni che hanno sfruttato propulsori a bassa spinta per le proprie esplorazioni interplanetarie.

Il primo velivolo spaziale che utilizzò la propulsione elettrica come propulsione primaria fu il Deep Space 1 lanciato dalla NASA il 25 ottobre 1998 dalla base di Cape Canaveral mediante un vettore Delta II nell'ambito del New Millennium Program. Questa missione rimase nella storia per i numerosi successi ottenuti. L'obiettivo principale della missione era la sperimentazione di dodici nuove tecnologie rivoluzionarie. La sfida maggiore era installare a bordo della sonda un nuovo sistema di propulsione che utilizzasse lo xeno come propellente, un gas incolore, inodore e quattro volte più leggero dell'aria. Al momento del lancio la sonda imbarcava 82 kg di Xe. La propulsione primaria avveniva quindi mediante l'utilizzo di un propulsore a ioni (il cui funzionamento verrà trattato nei capitoli successivi) denominato NSTAR.

NSTAR è un acronimo che indica NASA SEP (Solar Electric Propulsion) Technology Application Readiness, un programma in collaborazione tra Glenn Research Center e Jet Propulsion Laboratory iniziato nel novembre 1992 con l'obiettivo di progettare, realizzare e validare la propulsione a ioni a bassa spinta per uso spaziale.

Gli ingegneri del team di Mission Analysis dovettero per la prima volta approcciarsi alla bassa spinta e la progettazione delle traiettorie richiese l'applicazione di tecniche fino ad allora poco utilizzate. La destinazione, utile a testare le nuove tecnologie, era l'asteroide

9969 Braille, un piccolo corpo celeste di 2.2 km di diametro orbitante attorno la fascia principale degli asteroidi situata tra Marte e Giove con il quale la sonda Deep Space 1 ebbe un incontro ravvicinato il 29 luglio 1999 ad appena 27 km di distanza.

La prima fase di test avvenne con successo, e la missione primaria terminò il 18 settembre 1999, ma la NASA decise di estenderla fino all'ottobre 2001 con l'obiettivo di realizzare due nuovi incontri: con la cometa Wilson-Harrington nel gennaio 2001 (poi non effettuato) e con la cometa Borrelly nel settembre 2001. A fine settembre 2001, a 230.000 km dalla Terra, nelle vicinanze dell'orbita di Marte, la sonda ha attraversato la chioma della cometa, raccogliendo numerose immagini dell'astro.

Nel complesso l'IPS (Ion Propulsion System) accumulò 16.265 ore di funzionamento e consumò circa il 90% del propellente, fornendo un  $\Delta V$  di 4.3 km/s. Dopo i tentativi falliti iniziali di avviamento, l'IPS ha prontamente avviato la spinta su tutti i 199 tentativi. Per la cronaca anche le altre 12 tecnologie superarono a pieno i test.

Questo importante successo diede il via ad un serie di missioni con propulsione elettrica primaria delle quali ne presenteremo solo alcune più ragguardevoli.

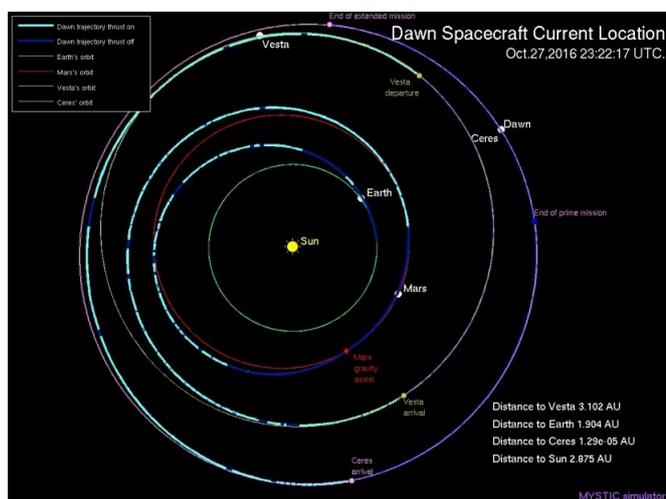
Il 9 maggio 2003, per opera della JAXA (Agenzia Spaziale Giapponese), venne lanciata la sonda Hayabusa (letteralmente falco pellegrino) con lo scopo di atterrare sull'asteroide 25143 Itokawa e prelevare dei campioni di suolo da riportare sulla Terra. La propulsione primaria anche questa volta fu affidata ad un propulsore a ioni con propellente lo Xenon. Dopo una prima fase di controllo del propulsore ionico, il 25 giugno il sistema propulsivo iniziò ad operare alla massima potenza fino al raggiungimento della destinazione verso la metà di settembre del 2005; un funzionamento quasi continuo per tutta la traiettoria ad eccezione ad esempio del 31 marzo 2004 in cui l'attività del propulsore ionico fu interrotta in previsione della manovra di fionda gravitazionale con la Terra. Il 28 agosto, si passò dalla propulsione elettrica alla propulsione chimica, predisposta per le manovre in prossimità dell'asteroide. La missione non avvenne in assenza di complicazioni; durante i 30 minuti di permanenza sul corpo celeste, il malfunzionamento di uno dei dispositivi di frantumazione non permise l'ingresso nella capsula di una grande quantità di materiale, inoltre durante la trasferta di rientro 2 dei 3 giroscopi si guastarono compromettendo il risultato finale della missione, ma nonostante tutto il rientro avvenne il 13 giugno 2010 in Australia. La conferma della presenza di minute particelle di asteroide nel contenitore interno della sonda, in data 6 luglio 2010, ha sottolineato il buon risultato della missione accompagnato da dati sconvolgenti sulla propulsione a ioni con ben 40.000 ore di funzionamento che hanno permesso un'accelerazione complessiva di 2.2 km/s.

Sulle orme dei successi di tale missione, il 3 dicembre 2014 è stata lanciata la sonda Hayabusa 2 con lo scopo di raggiungere l'asteroide Ryugu e prelevare dei campioni di suolo da riportare sulla Terra. La propulsione è affidata a quattro propulsori ionici che utilizzano xenon e forniscono una spinta complessiva di 28 mN, con un impulso specifico di 2080 s;

complessivamente, forniranno un  $\Delta V$  di 2 km/s. La sonda è dotata inoltre di 12 propulsori chimici ad idrazina in grado di fornire 20 N di spinta per le manovre di correzione di rotta e quelle in prossimità della Terra. La sonda si trova ad oggi a circa 5 000 000 km dall'obiettivo, e l'arrivo a destinazione è previsto per il luglio 2018 con un rientro in atmosfera terrestre dal 2020.

Come non citare inoltre la sonda SMART-1 dell'Agenzia Spaziale Europea facente parte delle missioni SMART dall'acronimo Small Missions for Advanced Research in Technology, lanciata il 27 settembre 2003 dal Centre Spatial Guyanais nella Guyana Francese ad opera del vettore Ariane 5. L'obiettivo della missione era testare nuove tecnologie tra cui il propulsore ionico alimentato dai pannelli fotovoltaici con propellente lo Xenon che sarebbe servito a portare a destinazione la sonda in orbita lunare. La missione si concluse il 3 settembre 2006, al momento dello schianto della sonda con la superficie lunare nella regione denominata Lacus Excellentiae (Lago dell'Eccellenza).

Tra le missioni americane ad opera della NASA, è d'obbligo citare la più moderna missione DAWN lanciata il 27 settembre 2007 presso la Cape Canaveral Air Force Station Space Launch ad opera del vettore Delta II. L'obiettivo della missione era quello di analizzare e migliorare la comprensione dei processi che portarono alla formazione del sistema solare, e per fare ciò si decise di raggiungere ed esaminare il pianeta nano Cerere e l'asteroide Vesta.



**Figura 1.8** – Missione Dawn (software MYSTIC<sup>9</sup>): simulazione orbitale <sup>10</sup>

<sup>9</sup> MYSTIC è un software di design e visualizzazione, sviluppato dal JPL della NASA, che fornisce un'ottimizzazione di traiettorie a bassa spinta.

<sup>10</sup> Fonte NASA <https://dawn.jpl.nasa.gov>

Fu la prima sonda in assoluto ad orbitare attorno a due corpi celesti sfruttando inoltre la spinta gravitazionale del pianeta Marte. La propulsione primaria venne affidata a tre propulsori a ioni derivanti proprio dalla prima missione Deep Space 1, aventi una spinta massima di 90 mN e un impulso specifico di 3100s. Ad oggi la missione è ancora operativa. Il 16 luglio 2011, dopo 2.8 miliardi di km ed il consumo di 252 kg di propellente (xeno), la sonda venne dichiarata ufficialmente in orbita attorno a Vesta, ma la traiettoria spiraleggiante di avvicinamento al planetoido finì il 22 luglio, con una distanza tra esso e la sonda di circa 5 200 km. Dopo un anno di acquisizione dati di Vesta, la sonda lasciò il nano pianeta raggiungendo Cerere il 6 marzo del 2015 ed è tutt'ora in orbita attorno all'asteroide.

Un'altra missione di esplorazione spaziale su cui l'agenzia spaziale europea sta investendo, in collaborazione con JAXA, è la missione BepiColombo. Come detto in precedenza si compone di due sonde distinte che opereranno autonomamente una volta giunta in orbita di Mercurio: la Mercury Planetary Orbiter (MPO) e la Mercury Magnetospheric Orbiter (MMO). Il modulo di trasferimento è denominato Mercury Transfer Module (MTM) utile unicamente durante la fase di viaggio e per l'inserimento in orbita. La traiettoria di BepiColombo è realizzata mediante l'utilizzo di una propulsione elettrica solare, raggiungendo Mercurio con una bassa velocità relativa combinando archi di propulsione a bassa spinta e fly-by della Terra, Venere e Mercurio. Il lancio avverrà il 5 ottobre 2018 mediante il vettore Ariane 5 e raggiungerà Mercurio dopo circa 7.2 anni di crociera. La propulsione elettrica (propulsione a ioni capace di fornire una spinta massima di 290 mN) verrà impiegata fino all'inserimento della SOI di Mercurio; la cattura e l'immissione in orbita su Mercurio avverrà mediante l'utilizzo di un propulsore a propellente chimico.

Una missione che verrà spesso citata in questo testo, poiché utilizzata successivamente per la fase di validazione delle analisi, è il progetto Mars Sample Return. Si tratta di uno studio in fase embrionale che prevede il prelevamento di campioni di roccia e polvere dalla superficie marziana, e il rientro sul pianeta Terra. La realizzazione di questa missione sarà sicuramente un passo avanti verso la realizzazione di future missioni di esplorazione umana su terreno marziano. Per svolgere questa missione pionieristica saranno necessarie numerose nuove tecnologie: il sistema di atterraggio, il veicolo di risalita, il sistema di rendez-vous nell'orbita di Marte e il veicolo o capsula di rientro sulla Terra. Per tale motivo è richiesto uno sforzo globale, con una particolare collaborazione tra l'ESA e la NASA, nonché la partecipazione di altre agenzie spaziali. Dai primi studi, si è sempre più diffusa la possibilità di utilizzare uno scenario che preveda una propulsione primaria elettrica per ambe due i trasferimenti (Terra-Marte e il rientro Marte-Terra).

Una breve presentazione di alcune delle missioni che hanno sfruttato, stanno sfruttando e che si pensa sfrutteranno in futuro una propulsione primaria elettrica, permette di addentrarci sempre più nel vasto, complesso ma affascinante mondo della propulsione a bassa spinta.

## 1.4 Classificazione Low Thrust Propulsor

Come è stato possibile osservare dalla lettura del precedente paragrafo, il propulsore maggiormente utilizzato tra le ultime missioni spaziali a bassa spinta è il propulsore a ioni. Seguirà un breve elenco dei vari propulsori a bassa spinta di uso comune con associate le principali caratteristiche prestazionali.

Come si evince nella parte iniziale di questo capitolo, l'obiettivo di qualsiasi sviluppo in ambito di propulsione è quello di massimizzare il più possibile l'impulso specifico caratteristico del propulsore. Fino ad ora, aumentare l'impulso specifico ha sempre implicato una spinta massima del propulsore relativamente bassa.

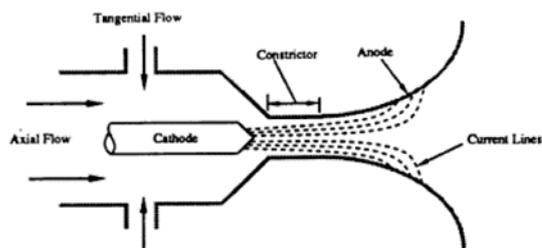
Ad oggi, due dei principali sistemi di propulsione a bassa spinta sono la propulsione elettrica e le vele solari; la prima è assolutamente quella con alle spalle più anni di ricerca. La propulsione elettrica fa uso di energia da una fonte esterna, normalmente solare.

Distinguiamo tre diverse tipologie di propulsori elettrici, classificati in base al metodo di accelerazione del propellente:

- Elettrotermico (Resistojets, Arcjets)
- Elettrostatico (propulsori ionici, propulsori ad effetto Hall)
- Elettromagnetico (PPT, PIT, MPD)

Nei propulsori di propulsione elettrotermica, il funzionamento si basa sull'effetto Joule, mediante il quale l'energia elettrica viene convertita in calore che viene fornito al fluido di lavoro in una camera di riscaldamento aumentandone la velocità di scarico. Nei Resistojets questa conversione avviene mediante l'utilizzo di una resistenza elettrica (che può essere o non essere a diretto contatto con il propellente utilizzato, rispettivamente si parlerà di Direct Contact o di Sealed Cavity). Tra i fluidi di lavoro, in termini prestazionali sarebbe conveniente l'utilizzo dell'idrogeno (per la sua ridotta massa molecolare) ma a causa della sua elevata criogenicità, si tende ad utilizzare ammoniaca ( $\text{NH}_3$ ) e idrazina ( $\text{N}_2\text{H}_4$ ).

La principale limitazione dei Resistogetti è la presenza della resistenza elettrica, con i conseguenti limiti strutturali sul valore della temperatura. I tipici valori di impulso specifico che ne derivano sono tra i 500s e 350s, 40% superiori all'equivalente chimico.



**Figura 1.9** – Configurazione Arcogetto<sup>11</sup>

Gli Arcjets sono un'altra forma di propulsione elettrotermica la cui struttura è presentata in figura. In questa tipologia di propulsori il propellente si riscalda passando attraverso un arco elettrico generato tra i due elettrodi. Problemi si pongono tuttavia per la stabilizzazione dell'arco elettrico, per la quale si può ricorrere ad un moto di swirl (rotazione) imposto al fluido propellente che lambisce gli elettrodi, o ad una strizione dell'arco. L'impulso specifico è in genere compreso tra i 500 e i 600 s utilizzando idrazina, e si raggiungono valori anche di 800 s con l'utilizzo di ammoniacca.

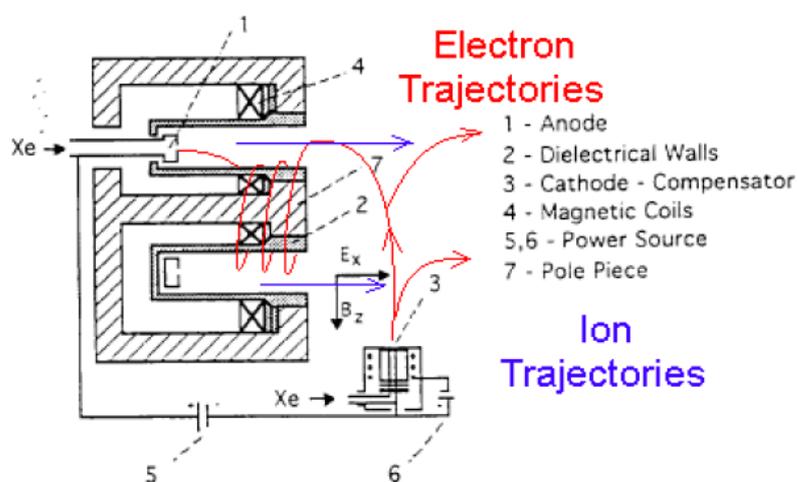
I principali propulsori elettrostatici sono i propulsori a griglia elettrostatica e i propulsori a effetto Hall, in cui le molecole ionizzate del gas operativo vengono direttamente accelerate mediante un campo elettrico. Con questa tipologia di propulsori non si presentano i problemi strutturali legati alla temperatura e per tale motivo si riescono ad ottenere impulsi specifici molto più elevati nell'intervallo di 3000-4500 s.

I propulsori ionici a griglia elettrostatica usano comunemente gas di xeno che viene ionizzato bombardandolo con elettroni provenienti da un catodo rovente. I ioni positivi generati all'interno della camera di ionizzazione, si diffondono poi attraverso la griglia schermo nella zona successiva delimitata dalla griglia acceleratrice. Queste due griglie sono poste ad una opportuna differenza di potenziale che causa l'accelerazione degli ioni a velocità elevata, i quali, attraversando la griglia acceleratrice, generano la spinta. Un altro catodo nella parte esterna del motore emette altri elettroni che si combinano con gli ioni per neutralizzarli. Questo per evitare che il propulsore si carichi negativamente, con un conseguente ritorno del raggio di ioni verso il veicolo che ne annulla la spinta.

Uno dei problemi di questi propulsori è dovuto alla limitazione della legge di Child secondo la quale vi è un valore massimo di corrente che può passare tra due elettrodi nel caso in cui ad essere accelerati siano ad esempio solo ioni.

<sup>11</sup> Fonte vedere riferimento bibliografico [\[24\]](#)

Per ovviare a tale problema i propulsori ad effetto Hall utilizzano un plasma quasi neutro (stesso numero di ioni ed elettroni) per ionizzare il propellente. Accelerare lo stesso numero di ioni ed elettroni causerebbe un contributo nullo alla propulsione e per tale motivo è necessario inserire un campo magnetico relativamente elevato da determinare un parametro di Hall maggiore di uno. La configurazione di questi propulsori permette di generare un campo magnetico  $B$  radiale, tramite bobine magnetiche interne ed esterne. Viene anche generato un campo elettrico  $E$  assiale e l'effetto combinato di questi campi genera l'effetto Hall, che confina gli elettroni in una direzione perpendicolare al campo elettrico  $E$  ed al campo magnetico  $B$  seguendo una traiettoria cicloidale. Il loro principio è usare un plasma rotante di elettroni utili a ionizzare un propellente iniettato attraverso l'anodo. Gli ioni però sono troppo pesanti per essere significativamente influenzati dall'effetto Hall (raggio della cicloide non contenuto nella macchina). Di conseguenza accelerano assialmente sotto l'influenza del campo elettrico ed escono dal motore ad alta velocità, producendo spinta. I propulsori ad effetto Hall in genere forniscono una spinta nella regione di 100 mN e un impulso specifico tra i 1500 e i 1700 s. Uno schema operativo è mostrato in figura 1.10 .

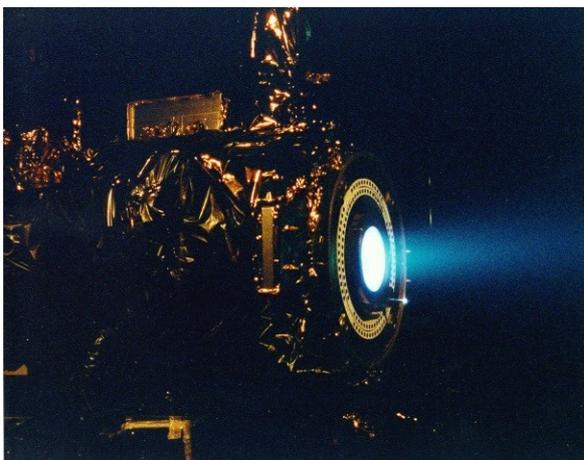


**Figura 1.10** – Configurazione propulsori ad effetto Hall<sup>12</sup>

Il sistema di propulsione solare elettrica del veicolo spaziale SMART-1 dell'Agenzia Spaziale Europea era ad esempio un propulsore a effetto Hall, lo Snecma PPS-1350-G1. Durante il suo funzionamento di poco più di 13 mesi con 289 accensioni, ha consumato circa 58.8 kg di xeno, producendo un  $\Delta V$  di 2737 m/s.

<sup>12</sup> Fonte vedere riferimento bibliografico [17]

Come non citare il propulsore a ioni denominato NSTAR utilizzato nelle missioni Deep Space 1 e Dawn. Utilizzando Xenon come propellente fornisce una spinta di 92 mN ad un impulso specifico di 3300 s richiedendo una potenza di 2.3 kW. Durante la fine del 2012 la



NASA presentò il progetto NEXT, un propulsore a ione frutto dell'evoluzione del precedente NSTAR. NASA Evolutionary Xenon Thruster – NEXT ha superato a pieno i test di funzionamento, rimanendo attivo per 43.000 ore (circa 5 anni), e gestendo oltre 770 kg di propellente Xenon. Alla massima potenza richiesta (circa 10kW) fornisce una spinta di 364 mN ad un impulso specifico di 3900 s.

*Figura 1.11 – Propulsore NSTAR<sup>13</sup>*

I dispositivi di propulsione elettromagnetica sfruttano campi magnetici, autoindotti o applicati, per accelerare il plasma, tipicamente mediante la forza di Lorentz ( $J \times B$ ) in cui l'accelerazione fornita da tale forza è proporzionale al prodotto vettoriale tra la densità di corrente elettrica e il campo magnetico. Un tipico esempio sono i propulsori MPD (Magneto Plasma Dynamic). Il gas (solitamente Litio, Idrogeno o Argon) entra nella camera principale dove viene ionizzato in plasma dal campo elettrico tra l'anodo e il catodo. La corrente che si genera tra essi crea un campo magnetico attorno al catodo che attraversa il campo elettrico, accelerando di conseguenza il plasma a causa della forza di Lorentz.

La maggior parte di questi propulsori è proposta per livelli di potenza elevati, quasi sempre più alte di 100 kW, e a causa della tecnologia in nostro possesso (servirebbero generatori nucleari) non hanno guadagnato alcuna esperienza di volo pur se caratterizzati da valori di impulso specifico molto alti anche dell'ordine di 6000 s.

L'altro tipo di sistema di propulsione a bassa spinta in fase di sviluppo è la vela solare, che utilizza direttamente la pressione della radiazione solare per fornire la spinta. I fotoni vengono riflessi dalla vela e le collisioni generano un effetto propulsivo. Sono caratterizzati da spinta notevolmente ridotte dell'ordine di qualche micron Newton, rendendole di utilizzo non molto frequente tra le missioni spaziali anche se ormai ampiamente testate.

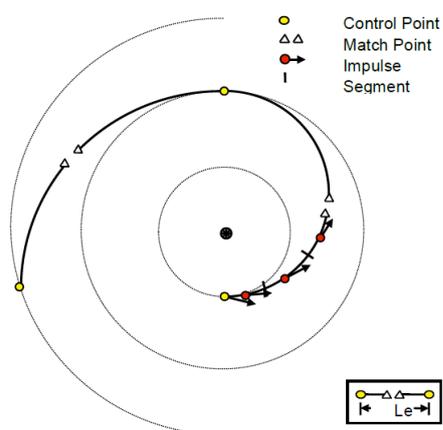
<sup>13</sup> Fonte NASA JPL <https://www.nasa.gov>

## 1.5 Low Thrust Trajectory Tools

Come è possibile comprendere dal vasto numero di missioni con propulsione primaria a bassa spinta ad oggi sviluppate, risulta di fondamentale importanza per le aziende spaziali approcciarsi alle analisi di queste traiettorie mediante opportuni tools che modellizzino ed ottimizzino il problema in ambito di missioni interplanetarie (e non solo).

In questo paragrafo verranno presentati due software progettati ed utilizzati dall'ESA e dalla NASA in ambito della Mission Analysis per l'ottimizzazione di traiettorie a bassa spinta.

MALTO (Mission Analysis Low-Thrust Optimizer) è un software per la progettazione preliminare e l'ottimizzazione di traiettorie interplanetarie a bassa spinta. Con questo software viene modellizzata la manovra continua a bassa spinta con una serie di manovre impulsive.



L'entità della spinta su ogni segmento è limitata dalla potenza disponibile (ad esempio fornita dai pannelli solari in caso di propulsione solare e quindi dalla distanza dal Sole), dal modello del motore e dalla durata del segmento. Durante l'ottimizzazione, MALTO propaga una traiettoria in avanti e all'indietro da punti di controllo (ad esempio pianeti); viene applicata la continuità nei "Match Point", vincolando velocità e posizione dello S/C in quei punti.

**Figura 1.12** – Struttura della traiettoria nelle analisi del software MALTO<sup>14</sup>

DITAN (Direct Interplanetary Trajectory Analysis) è un software generico per l'analisi e la progettazione della traiettoria spaziale sviluppato dal Politecnico di Milano su commissione dell'ESA. L'algoritmo principale risolve un problema di controllo ottimale vincolato e non lineare sfruttando un metodo diretto, nello specifico il metodo SQP ed è adottato per ottimizzare traiettorie nello spazio, in particolare quelle basate su propulsione a bassa spinta anche in presenza di gravity assist.

<sup>14</sup> Fonte vedere riferimento bibliografico [21]

## 2 Elementi di Meccanica Orbitale

### 2.1 Problema dei due corpi

Il modello kepleriano fa riferimento ad una condizione ideale valida nel campo della meccanica classica, in cui si è in presenza di un corpo principale attrattore di massa  $M$  e supposto puntiforme, e un corpo secondario di massa  $m$ , anch'esso puntiforme, in cui l'unica forza in gioco è l'attrazione gravitazionale  $F$ , definita dalla seguente relazione:

$$\bar{F} = -G \frac{Mm}{r^3} \bar{r} \quad (2.1)$$

dove il vettore  $\mathbf{r}$  è il vettore che congiunge i due corpi, mentre  $G$  è la Costante di Gravitazione Universale, che nel Sistema Internazionale, assume il valore

$$G = 6.6726 \times 10^{-11} \frac{m^3}{kg \ s^2} \quad (2.2)$$

In seguito ad opportuni passaggi algebrici a partire dalla relazione precedente, è possibile formulare l'equazione del moto in ipotesi di moto kepleriano.

$$\ddot{\bar{r}} = -G \frac{M + m}{r^3} \bar{r} \quad (2.3)$$

In ipotesi di satelliti artificiali il corpo primario ha solitamente una massa più grande di quella del secondario, potendo così effettuare la seguente assunzione.

$$M + m \approx M \quad (2.4)$$

L'equazione del moto diventa quindi la seguente.

$$\ddot{\bar{r}} = -\frac{\mu}{r^3} \bar{r} \quad (2.5)$$

Da cui è possibile introdurre la quantità  $\mu$  che prende il nome di Parametro Gravitazionale, che nel caso della Terra assume il valore:

$$\mu_{\oplus} = GM_{\oplus} = 3.986 \times 10^{14} \frac{m^3}{s^2} \quad (2.6)$$

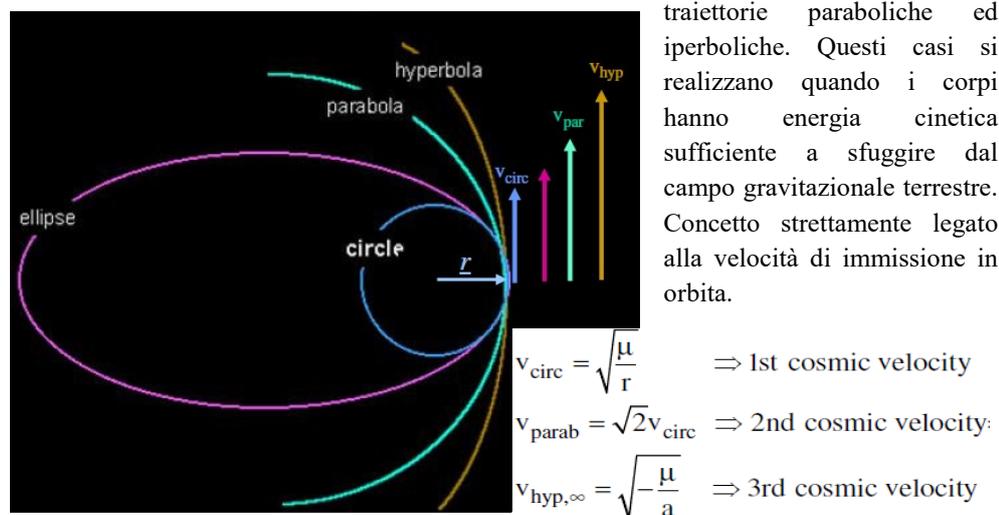
La risoluzione del problema dei due corpi, nonché del moto kepleriano, permette di ottenere in forma chiusa la traiettoria nello spazio del corpo secondario, orbitante attorno al corpo principale. Le deduzioni cui si perviene applicando queste ipotesi e sviluppando i facili passaggi matematici sono riassunte nelle tre leggi di Keplero. Adattate al nostro ambito, esse affermano che:

1. I satelliti si muovono su piani fissi dello spazio e la loro traiettoria è un'ellisse in cui la Terra occupa uno dei fuochi;
2. La superficie spazzata dalla congiungente Terra - satellite nell'unità di tempo è costante;
3. Il periodo orbitale è direttamente proporzionale alla distanza media dal centro della Terra.

Queste leggi furono formulate tra il 1609 e il 1619 dal matematico tedesco Johannes Kepler (Giovanni Keplero) e descrivono il moto dei pianeti del Sistema Solare, qui nel testo ricondotte al moto tra un satellite artificiale e la Terra.

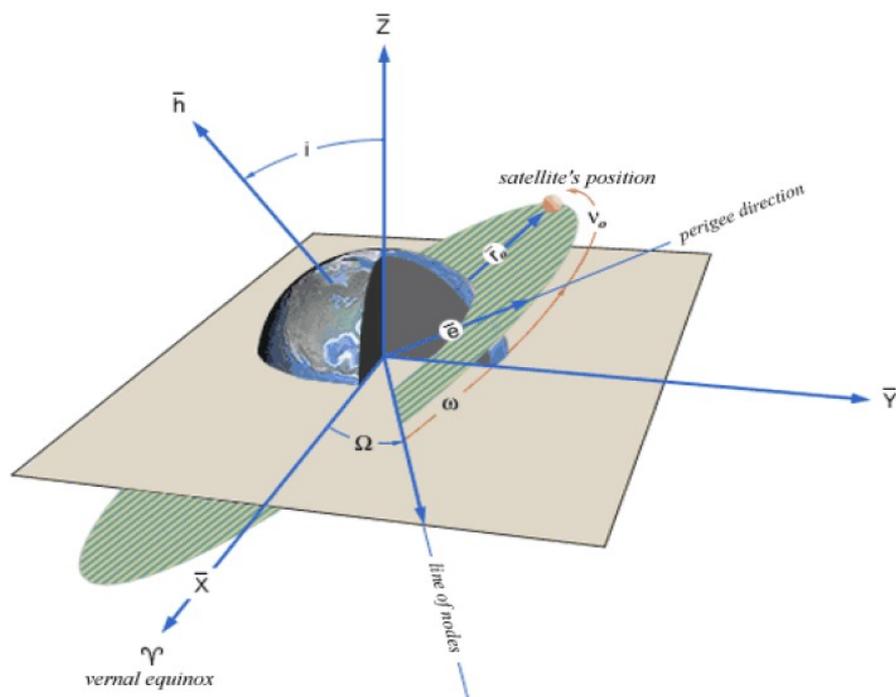
Il primo dato a saltare all'occhio è il fatto che le traiettorie siano fisse nello spazio. In teoria, una volta inserito un satellite su una certa orbita, questo continua a muoversi su di essa indefinitamente. In questo senso, esaurita la fase del lancio, non è necessario l'ausilio di alcun propulsore per perpetuare il moto.

In secondo luogo, si può notare come la forma dell'orbita non sia casuale. Per i casi di nostro interesse si tratta di un'ellisse e le caratteristiche geometriche di questa curva dipendono dalle condizioni iniziali di moto. In generale, esiste anche la possibilità di avere



**Figura 2.1** – Differenti orbite in relazione alla velocità di immissione

Ogni orbita è caratterizzata da sei parametri orbitali che istante per istante permettono di conoscere velocità e posizione del satellite nello spazio. Questi ultimi prendono il nome di vettori di stato. I parametri orbitali classici, che è possibile osservare in figura, sono schematizzati nella tabella 2.1 .



**Figura 2.2 – Parametri orbitali<sup>15</sup>**

<sup>15</sup> Fonte vedere riferimento bibliografico [3]

*Tabella 2.1 – Parametri orbitali classici*

Simbolo	Nome	Significato
<b>a</b>	Semiasse maggiore	Parametro dimensionale dell'orbita
<b>e</b>	Eccentricità	Parametro che definisce la forma dell'orbita
<b>i</b>	Inclinazione	Inclinazione del piano orbitale rispetto al piano di riferimento
<b><math>\Omega</math></b>	Ascensione retta del nodo ascendente	Angolo tra l'asse X del sistema di riferimento e la linea dei nodi (intersezione tra piano di riferimento e piano orbitale)
<b><math>\omega</math></b>	Argomento del perigeo	Angolo tra la linea dei nodi e la direzione del perigeo
<b><math>\Theta</math></b>	Anomalia vera	Posizione dello S/C sull'orbita a partire dal perigeo

## 2.2 Approssimazione Patched Conic

Nel caso di missioni interplanetarie, il modello kepleriano risulterebbe troppo semplificato poiché lungo la traiettoria interplanetaria lo S/C subisce l'influenza dei vari corpi celesti in maniera non trascurabile. Con lo scopo di ottenere una metodologia che mantenga le semplificazioni ottenute dal modello kepleriano, si è soliti attuare in prima analisi il metodo delle coniche raccordate.

Il patched conic approximation o metodo delle coniche raccordate, consiste nel suddividere la missione in una serie di fasi, nelle quali il satellite si trova soggetto principalmente all'azione gravitazionale di uno specifico corpo celeste. Questo concetto permette di applicare in ciascuna fase il modello kepleriano, semplificando notevolmente la risoluzione che potrà essere trattata in termini analitici.

Ipotizzando ad esempio una trasferta Terra-Marte, è possibile suddividerla in tre fasi:

- Fase di fuga: mediante un opportuno sistema propulsivo lo S/C si allontana dall'orbita terrestre, ma risentendo maggiormente dell'attrazione gravitazionale terrestre;
- Fase eliocentrica: lo S/C si trova in orbita di trasferimento attorno al Sole in avvicinamento al pianeta target;

- Fase di cattura: questa fase ha inizio quando lo S/C si avvicina al pianeta target, ad una distanza tale da risentire maggiormente dell'attrazione gravitazionale marziana.

Poter applicare il modello kepleriano ad ogni fase, permette per ognuna di queste di definire la traiettoria in modo univoco. Gli archi di conica della traiettoria associati ad ogni fase dovranno essere opportunamente raccordati. Ciò richiede l'utilizzo di ulteriori ipotesi semplificative; risulta necessario introdurre il concetto di sfera di influenza.

Detta anche sfera di Hill, dall'astronomo americano George William Hill che la definì, indica la porzione di spazio sferico attorno ad un corpo celeste, nella quale un corpo secondario orbitante risulta interamente soggetto all'influenza gravitazionale del corpo celeste (potendo trascurare le perturbazioni di terzi corpi).

Riconducendosi all'esempio precedente, la fase di fuga è contenuta all'interno della sfera di influenza terrestre. Le estremità di questa sfera sono caratterizzate dalla condizione secondo la quale il rapporto dell'accelerazione perturbativa della Terra sullo S/C fratto l'accelerazione gravitazionale primaria del Sole sullo S/C sia uguale al rapporto dell'accelerazione perturbativa del Sole sullo S/C fratto l'accelerazione gravitazionale primaria della Terra sullo S/C. Oltre il valore del raggio di Hill il Sole avrà influenza maggiore sullo S/C, addentrandosi nella fase eliocentrica della missione in questione.

Il concetto della sfera di influenza è una notevole semplificazione del problema in quanto nelle zone di raccordo si passa improvvisamente dall'azione gravitazionale di un corpo celeste a quella di un altro. Sarebbe opportuno considerare la transizione che nella realtà è molto graduale ma ciò complicherebbe notevolmente il problema facendo cadere le semplificazioni del modello kepleriano.

*Tabella 2.2 – SOI pianeti del sistema solare*

Pianeta	Raggio della sfera d'influenza
	[km]
<b>Mercurio</b>	112 500
<b>Venere</b>	616 400
<b>Terra</b>	924 600
<b>Marte</b>	577 400
<b>Giove</b>	48 223 000
<b>Saturno</b>	54 432 000
<b>Urano</b>	51 792 000
<b>Nettuno</b>	86 668 000
<b>Plutone</b>	3 130 000

Di notevole importanza è anche l'assunzione di un opportuno sistema di riferimento che permette una semplificazione del problema. Come è immediato aspettarsi, in ogni fase risulta opportuno scegliere un sistema di riferimento centrato nel corpo principale (legato al concetto di unica attrazione gravitazionale prima introdotto). Dal punto di vista di un ipotetico osservatore solidale al Sole (fase eliocentrica), le sfere di influenza risultano sostanzialmente puntiformi, coincidenti con il baricentro del corpo celeste a cui si riferiscono. Se invece si paragona il raggio della sfera di Hill di un pianeta, con una sua dimensione caratteristica (ad esempio il suo raggio medio), tale sfera apparirebbe enorme. Quindi, in un sistema di riferimento centrato sul pianeta di riferimento, si può assumere che le estremità della sfera di Hill siano poste all'infinito. Nel complesso, ciò permette di "normalizzare" le equazioni semplificandone la risoluzione.

In generale comunque il metodo delle coniche raccordate permette di mantenere una sufficiente accuratezza pur garantendo una elevata semplicità delle analisi e per tale motivo è ampiamente utilizzato nelle analisi preliminari di missione.

## 2.3 Perturbazioni

Il modello kepleriano sta alla base di assunzioni semplificative che rendono il modello ideale. Nella realtà la presenza di disturbi esterni (radiazioni solari, resistenza aerodinamica, asimmetrie del campo gravitazionale o presenza di terzi corpi) allontana la traiettoria reale da quella kepleriana. Ciò complica notevolmente la risoluzione. Con la presenza di una generica forza perturbativa non è possibile ottenere soluzioni in forma chiusa ed è necessario ricorrere a metodi di integrazione numerica.

Il primo studio a riguardo risale al passaggio della cometa di Halley nel 1910, la cui traiettoria venne studiata nel 1910 per opera di Philip Herbert Cowell che, insieme al collega astronomo Andrew C. D. Crommelin studiarono l'orbita della cometa integrando nel tempo le equazioni del moto in un sistema di riferimento eliocentrico. Questo metodo, denominato successivamente metodo di Cowell, è il più semplice e intuitivo per determinare la posizione  $r(t)$  e la velocità  $v(t)$  di un satellite in presenza di effetti perturbativi. Il metodo di Cowell prevede di integrare numericamente le equazioni del moto in un sistema di riferimento cartesiano centrato nel baricentro del corpo attrattore primario.

Si tratta di un sistema di due equazioni differenziali vettoriali non lineari del primo ordine:

$$\begin{cases} \dot{\mathbf{r}} = \mathbf{v} \\ \dot{\mathbf{v}} = -\frac{\mu}{r^3}\mathbf{\bar{r}} + \mathbf{a}_p \end{cases} \quad (2.7)$$

Indicando con  $\mathbf{a}_p$  l'accelerazione perturbativa complessiva e le cui condizioni all'istante iniziale  $t_0 = 0$  siano le seguenti:

$$\begin{cases} \mathbf{r}(t_0) = \mathbf{r}_0 \\ \mathbf{v}(t_0) = \mathbf{v}_0 \end{cases} \quad (2.8)$$

potendo così integrare numericamente nel tempo per  $t \geq t_0$ .

Un ulteriore metodo utilizzabile è la risoluzione delle *Lagrange's planetary equations* nonché le derivate temporali dei sei parametri orbitali in funzione di un'accelerazione di disturbo.

$$\dot{a} = \frac{da}{dt} = \frac{2\sqrt{a^3}}{\sqrt{\mu(1-e^2)}} [e \sin \theta a_{pR} + (1 + e \cos \theta) a_{pT}] \quad (2.9)$$

$$\dot{e} = \frac{de}{dt} = \sqrt{\frac{a(1-e^2)}{\mu}} [\sin \theta a_{pR} + (\cos \theta + \cos E) a_{pT}] \quad (2.10)$$

$$\dot{i} = \frac{di}{dt} = \frac{\sqrt{a(1-e^2)}}{\sqrt{\mu}(1-e \cos \theta)} \sin(\omega + \theta) a_{pN} \quad (2.11)$$

$$\dot{\Omega} = \frac{d\Omega}{dt} = \frac{\sqrt{a(1-e^2)} \sin(\omega + \theta)}{\sqrt{\mu}(1-e \cos \theta) \sin i} a_{pN} \quad (2.12)$$

$$\dot{\omega} = \frac{d\omega}{dt} = \sqrt{\frac{a(1-e^2)}{\mu}} \left[ -\frac{\sin(\omega + \theta) \cot i}{i + e \cos \theta} a_{pN} + \frac{2 + e \cos \theta}{e(1 + e \cos \theta)} \sin \theta a_{pT} - \frac{\cos \theta}{e} a_{pR} \right] \quad (2.13)$$

$$\dot{\theta} = \frac{d\theta}{dt} = \frac{h}{r^2} + \frac{1}{eh} [p \cos \theta a_{pR} + (p + r) \sin \theta a_{pT}] \quad (2.14)$$

Avendo indicato con  $p$  il *semilatus rectum*.

Riguardo l'accelerazione perturbativa è necessario effettuare alcune considerazioni. Come si osserva in figura 2.3, il vettore dell'accelerazione perturbativa  $\mathbf{a}_p$  può essere scomposto nelle varie componenti relative al sistema di riferimento RTN.

$$\mathbf{a}_p = a_{pR}\hat{\mathbf{R}} + a_{pT}\hat{\mathbf{T}} + a_{pN}\hat{\mathbf{N}} \quad (2.15)$$

$a_{pR}$ ,  $a_{pT}$ ,  $a_{pN}$  indicano le tre componenti del vettore  $\mathbf{a}_p$  che possono considerarsi note.

$\hat{\mathbf{R}}$ ,  $\hat{\mathbf{T}}$ ,  $\hat{\mathbf{N}}$  sono i versori del sistema di riferimento RTN (*Radial Transverse Normal*), la cui origine coincide istante per istante con il baricentro dello S/C. In particolare, il versore  $\hat{\mathbf{R}}$  è diretto lungo la radiale locale e punta verso l'esterno (rispetto al corpo principale),  $\hat{\mathbf{T}}$  è concorde con la direzione della velocità dello S/C in quell'istante, ed  $\hat{\mathbf{N}}$  completa la terna.

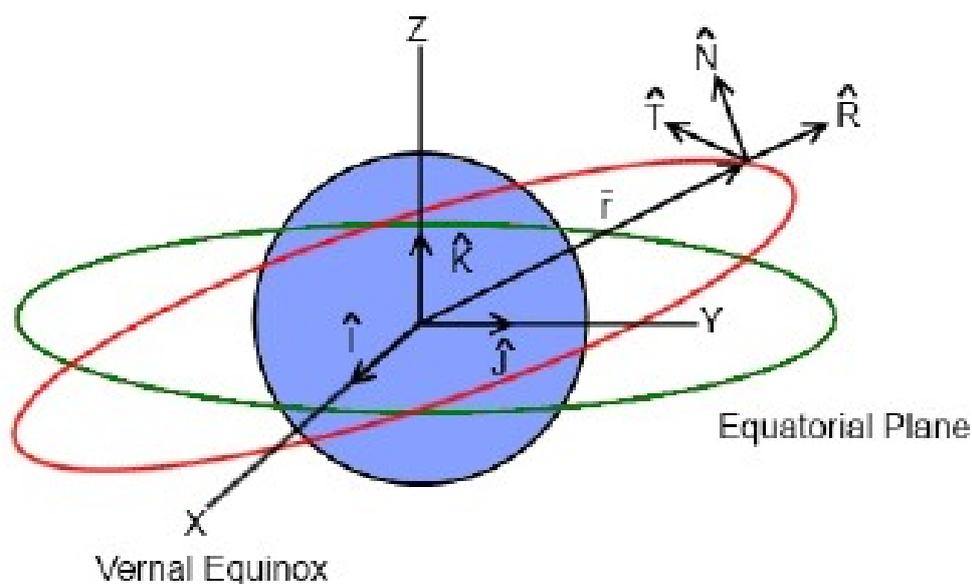
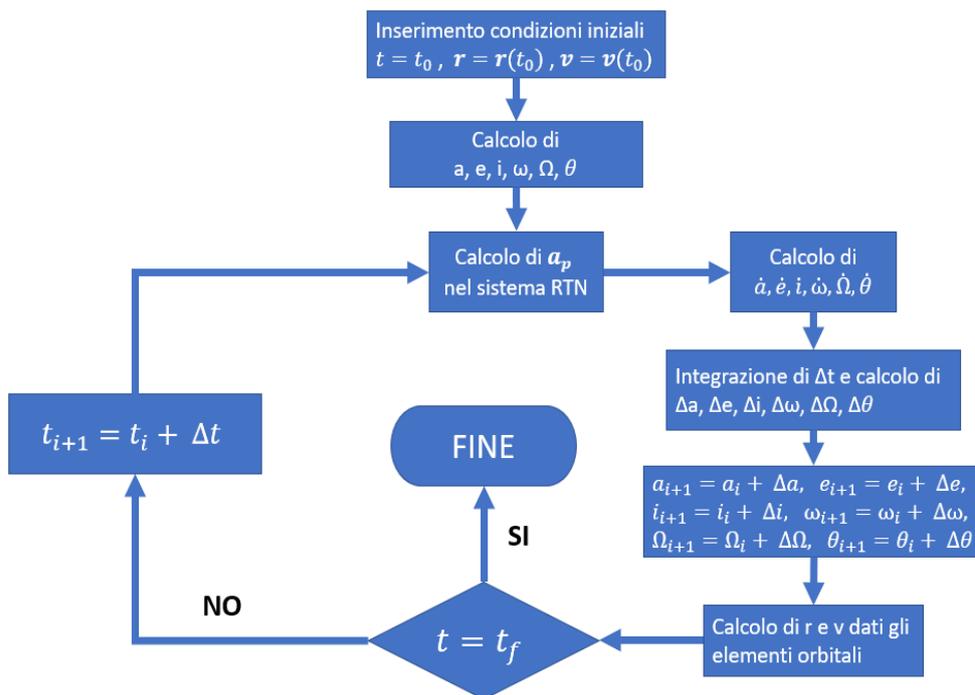


Figura 2.3 – Sistema di riferimento RTN

Utilizzando le equazioni di perturbazione precedentemente elencate è possibile applicare un algoritmo che definisca istante per istante la posizione del satellite lungo l'orbita perturbata. La figura 2.4 illustra l'algoritmo in questione.



**Figura 2.4** – Schema a blocchi algoritmo che sfrutta le Lagrange's Planetary Equations

Le formule appena introdotte hanno però alcune limitazioni. Osservando le varie equazioni si nota che presenterebbero delle singolarità in alcune specifiche condizioni. Per tale motivo questo sistema di equazioni è accettabile solo nel caso di orbite ellittiche ( $e < 1$ ) e non è valido per i casi particolari di orbite equatoriali ( $i = 0$ ) e circolari ( $e = 0$ ).

Un metodo risolutivo impiegato per la prima volta in anni più recenti (1970/1985) per ovviare a questo problema fu l'utilizzo di nuovi parametri orbitali, riconducibili ai parametri orbitali classici ma con la possibilità di trattare anche orbite iperboliche e circolari, con l'unica singolarità in presenza di inclinazioni di  $180^\circ$ .

Questi parametri furono chiamati “*modified equinoctial orbital elements*”, le cui relazioni in funzione dei parametri classici sono le seguenti:

$$p = a(1 - e^2) \quad (2.16)$$

$$g = e \sin(\omega + \Omega) \quad (2.17)$$

$$f = e \cos(\omega + \Omega) \quad (2.18)$$

$$k = \tan\left(\frac{i}{2}\right) \sin \Omega \quad (2.19)$$

$$h = \tan\left(\frac{i}{2}\right) \cos \Omega \quad (2.20)$$

$$L = \Omega + \omega + \theta \quad (2.21)$$

Noti i “*modified equinoctial orbital elements*”, è possibile ricavare i parametri orbitali classici dalle seguenti relazioni:

$$a = \frac{p}{1 - f^2 - g^2} \quad (2.22)$$

$$e = \sqrt{f^2 + g^2} \quad (2.23)$$

$$i = 2 \tan^{-1}\left(\sqrt{f^2 + g^2}\right) \quad (2.24)$$

$$\omega = \tan^{-1}\left(\frac{g}{f}\right) - \tan^{-1}\left(\frac{k}{h}\right) \quad (2.25)$$

$$\Omega = \tan^{-1}\left(\frac{k}{h}\right) \quad (2.26)$$

$$\theta = L - (\Omega + \omega) = L - \tan^{-1}\left(\frac{g}{f}\right) \quad (2.27)$$

Analogamente alle *Lagrange's planetary equations* è possibile calcolare le derivate temporali dei parametri orbitali equinoziali modificati e attraverso un algoritmo simile a quello precedentemente introdotto è possibile calcolare i vettori di stato dello S/C istante per istante.

Il sistema di equazioni differenziali di primo ordine dei parametri equinoziali in funzione dell'accelerazione perturbativa è il seguente:

$$\dot{p} = \frac{dp}{dt} = \frac{2p}{\omega} \frac{p}{h} a_{pT} \quad (2.28)$$

$$\dot{g} = \frac{dg}{dt} = \frac{r}{h} \left\{ -\frac{p}{r} \cos L a_{pR} + \left[ g + \left( 1 + \frac{p}{r} \right) \sin L \right] a_{pT} - f(k \cos L - h \sin L) a_{pN} \right\} \quad (2.29)$$

$$\dot{f} = \frac{df}{dt} = \frac{r}{h} \left\{ \frac{p}{r} \sin L a_{pR} + \left[ f + \left( 1 + \frac{p}{r} \right) \cos L \right] a_{pT} + g(k \cos L - h \sin L) a_{pN} \right\} \quad (2.30)$$

$$\dot{k} = \frac{dk}{dt} = \frac{r}{2h} (1 + k^2 + h^2) \sin L a_{pN} \quad (2.31)$$

$$\dot{h} = \frac{dh}{dt} = \frac{r}{2h} (1 + k^2 + h^2) \cos L a_{pN} \quad (2.32)$$

$$\dot{L} = \frac{dL}{dt} = \sqrt{\frac{\mu}{a^3}} - \frac{r}{h} (k \cos L - h \sin L) a_{pN} \quad (2.33)$$

Avendo indicato con  $\bar{\mathbf{h}} = \bar{\mathbf{r}} \times \bar{\mathbf{v}}$  il momento della quantità di moto per unità di massa dell'orbita.

## 2.4 Analogie tra Low Thrust ed accelerazione perturbativa

Come visto nel capitolo precedente, con la definizione di propulsore a bassa spinta si indica un propulsore la cui spinta massima rispetti la seguente condizione:

$$\frac{T}{W_s} \leq 10^{-5} \quad (2.34)$$

con

$$W_s = m \frac{\mu}{r^2} \quad (2.35)$$

Con questa assunzione, il modulo della spinta risulta molto basso se confrontato all'attrazione gravitazionale esercitata dal corpo principale. Per questo motivo si può trattare la propulsione come una sorta di accelerazione perturbativa che discosta la traiettoria da quella kepleriana ideale. Con tali considerazioni, è possibile effettuare uno studio delle manovre a bassa spinta riconducendoci allo studio appena presentato sugli effetti perturbativi.

La modellizzazione di traiettorie a bassa spinta richiede però un approccio leggermente più accurato che richiede non poche complicazioni. In primis la presenza di effetti propulsivi induce un consumo di propellente e quindi una riduzione di massa che va "aggiornata" ad ogni intervallo di integrazione. Qui di seguito il sistema di equazioni del moto in presenza di propulsione a bassa spinta:

$$\begin{cases} \frac{d\bar{r}}{dt} = \bar{v} \\ \frac{d\bar{v}}{dt} = -\frac{\mu}{r^3}\bar{r} + \bar{a}_p + \frac{\bar{T}}{m} \\ \frac{dm}{dt} = -\frac{T}{g_0 I_{sp}} \end{cases} \quad (2.36)$$

Avendo indicato con  $c = g_0 I_{sp}$  la velocità efficace di scarico caratteristica del propulsore data dal rapporto tra l'accelerazione gravitazionale media al livello del mare  $g_0$  e l'impulso specifico  $I_{sp}$  del propulsore.

Un'ulteriore complicazione sta nella peculiarità dei propulsori a bassa spinta in cui è possibile variare l'intensità e la direzione della spinta istante per istante (vettore  $\bar{T}$ ). Poter modellizzare fedelmente questa variazione renderebbe i calcoli troppo onerosi. Ad oggi, per ovviare a questo problema si tende a suddividere la traiettoria a bassa spinta in tanti sotto archi, ognuno dei quali avrà intensità e direzione della spinta costanti. In questo modo si riesce a ricondurre l'analisi ad un numero finito di variabili e sarà la scelta dell'ampiezza dell'arco (legato alle potenzialità del calcolatore) a dare il valore di accuratezza cercato.

Avere un vettore spinta costante per l'intero arco, permette anche di ricondurre l'intera propulsione ad una spinta impulsiva che fornisca nel complesso lo stesso  $\Delta V$  della low thrust.

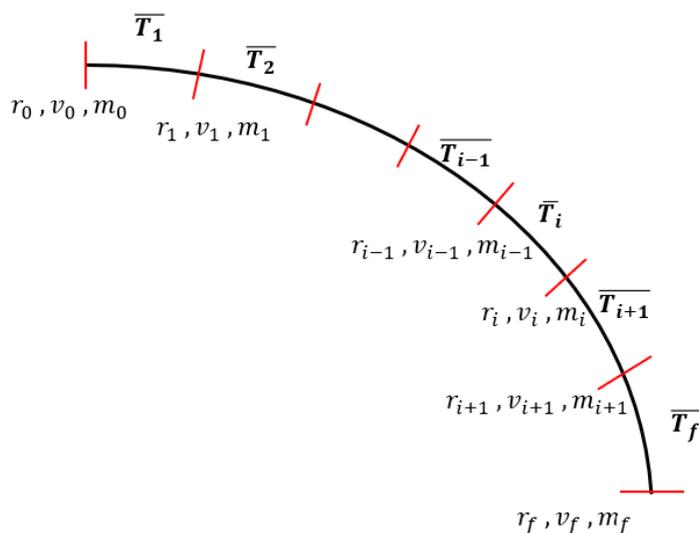
In questa tesi sono stati presi in considerazione tre possibili metodi di modellizzazione che verranno presentati nel proseguo del capitolo.

## 2.5 Metodi di modellizzazione

Un'accurata modellizzazione delle traiettorie a bassa spinta è di fondamentale importanza per una successiva analisi di ottimizzazione di quest'ultima. In questa tesi sono stati presi in esame tre possibili modelli; in ognuno è stata sfruttata la semplificazione di suddividere la traiettoria finale in tanti sotto archi, mantenendo costante modulo e direzione della spinta. Nel proseguo, verrà presentato lo specifico modello in termini di singolo arco. Verranno chiamati nodi il punto iniziale e finale di ogni arco, assumendo per continuità che l'ultimo nodo dell'arco  $i$ -esimo coincida con il primo nodo dell'arco  $i+1$ .

Ogni modello avrà come input posizione, velocità e massa al nodo iniziale, il vettore di spinta (con modulo e direzioni) e l'ampiezza dell'arco in secondi. Gli output saranno invece i vettori di stato e la massa al nodo finale. Le modifiche apportate sui tre modelli riguardano il modo di processare i dati all'interno del singolo arco, come è possibile osservare nel proseguo del capitolo.

### 1. Integrazione parametri orbitali – Solver Integral



**Figura 2.5** – Modellizzazione della traiettoria nel Solver Integral

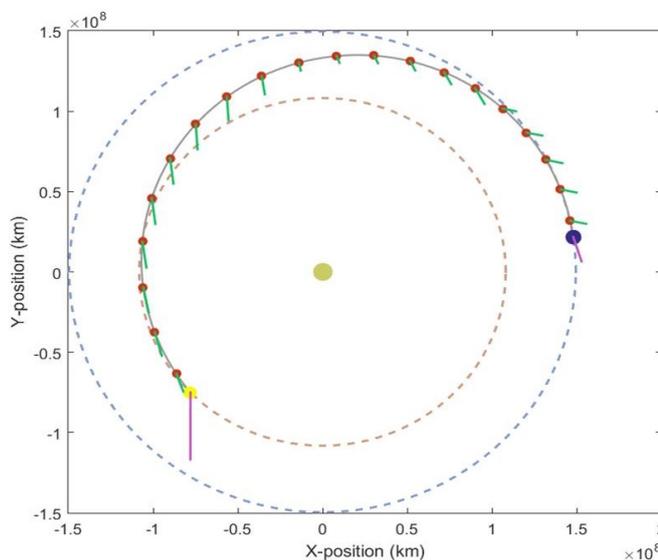
In questo modello gli output si ricavano integrando per tutta l'ampiezza dell'arco le *Lagrange's planetary equations* in funzione dei *modified equinoctial parameters*. Per l'integrazione si utilizza un metodo numerico di Runge Kutta di 4° ordine.

## 2. Modello impulsivo ai nodi – Solver Impulsive Node

In questo modello gli output vengono ricavati approssimando la low thrust ad una manovra impulsiva<sup>16</sup>. Nello specifico si avrà un primo impulso (metà del  $\Delta V$  complessivo) applicato al primo nodo e l'altra metà applicata al secondo nodo. All'interno del nodo la propagazione avviene mediante l'integrazione delle *Lagrange's planetary equations*. Con queste assunzioni le equazioni risultano notevolmente semplificate. Assumere una spinta impulsiva solo ai nodi equivale ad annullare i termini perturbativi nelle Lagrange's planetary equations. Ciò fa sì che tutti i parametri si mantengono costanti (modello kepleriano) e l'unico termine che varia è la posizione lungo l'orbita cioè l'anomalia vera  $\theta$ . Il calcolo della propagazione non richiede quindi alcuna integrazione ma un semplice calcolo algebrico.

$$\dot{\theta} = \frac{d\theta}{dt} = \frac{h}{r^2} + \frac{1}{eh} [p \cos v a_{pR} + (p + r) \sin v a_{pT}] = \frac{h}{r^2} \quad (2.37)$$

Nella figura 2.6 si può osservare una generica traiettoria tra due orbite circolari, in cui il  $\Delta V$  impulsivo associato ad ogni arco viene applicato su ogni nodo.



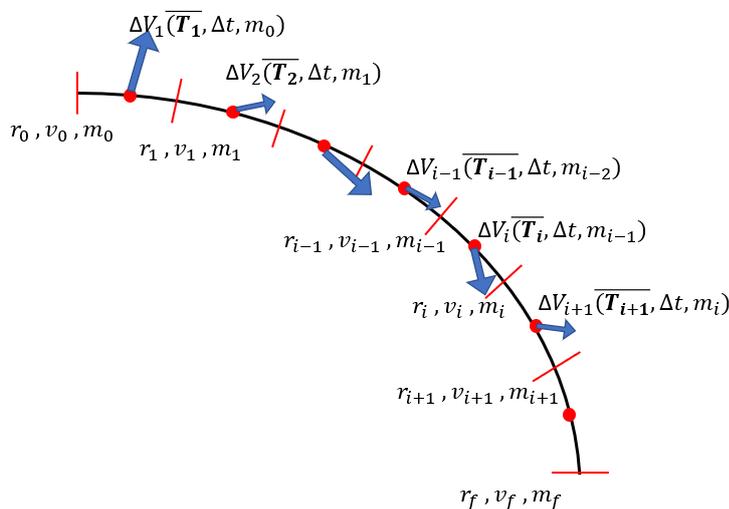
**Figura 2.6** – Modellizzazione della traiettoria nel Solver Impulsive Node<sup>17</sup>

<sup>16</sup> Concettualmente analogo al passaggio da carico distribuito a carico concentrato in termini strutturali.

<sup>17</sup> Fonte vedere riferimento bibliografico [12]

### 3. Modello impulsivo al nodo centrale – Solver Impulsive Middle

Questo modello risulta concettualmente analogo al precedente in termini di propulsione e propagazione, l'unica differenza sta nell'assumere un terzo nodo posto al punto medio dell'arco e applicare la manovra impulsiva interamente al nodo suddetto.



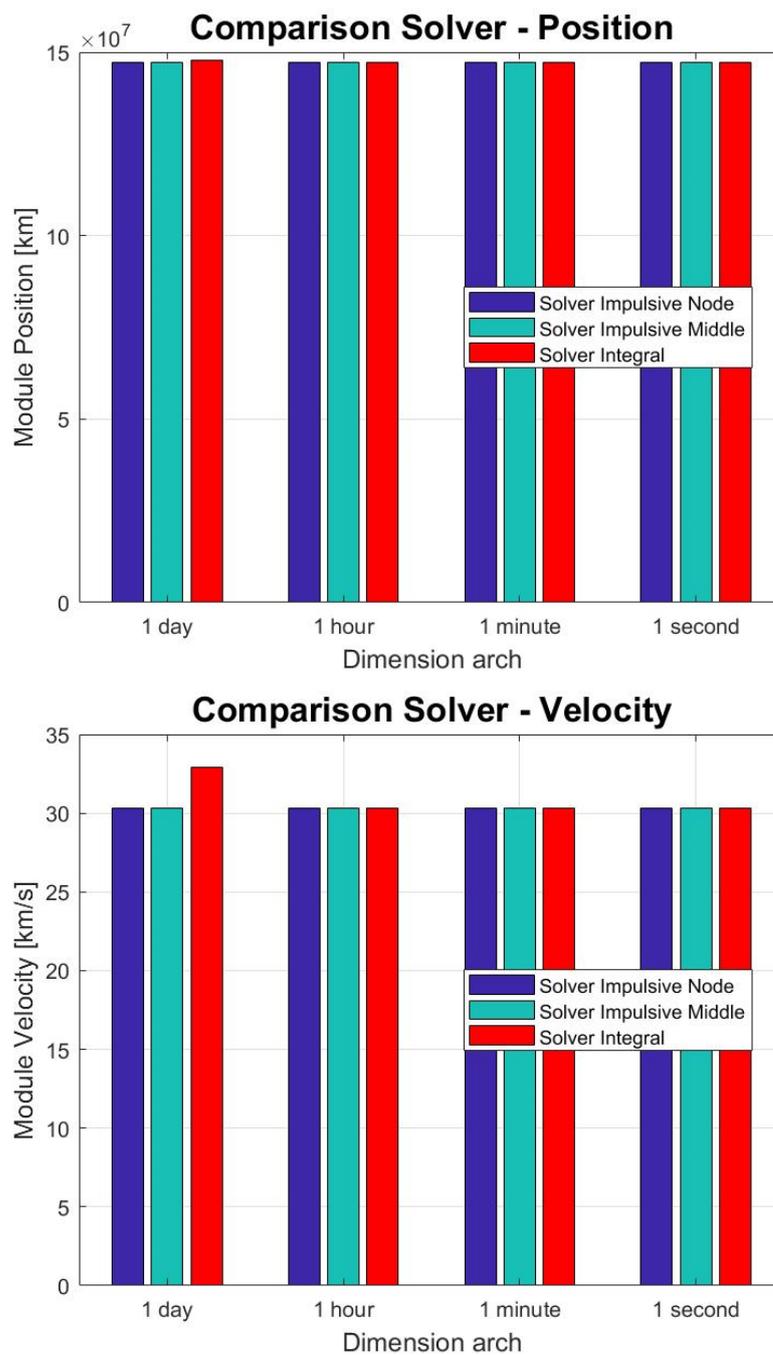
*Figura 2.7 – Modellizzazione della traiettoria nel Solver Impulsive Middle*

## 2.6 Scelta del modello

Al fine di valutare il più adatto risolutore da utilizzare nello studio di ottimizzazione, sono state effettuate delle analisi per confrontare i tre modelli.

Il primo studio è stato effettuato sul singolo arco, confrontando i vettori di stato in uscita dal risolutore, al variare della dimensione dell'arco. Sono state utilizzate 4 dimensioni di riferimento (1 secondo, 60 secondi, 3600 secondi, 86400 secondi).

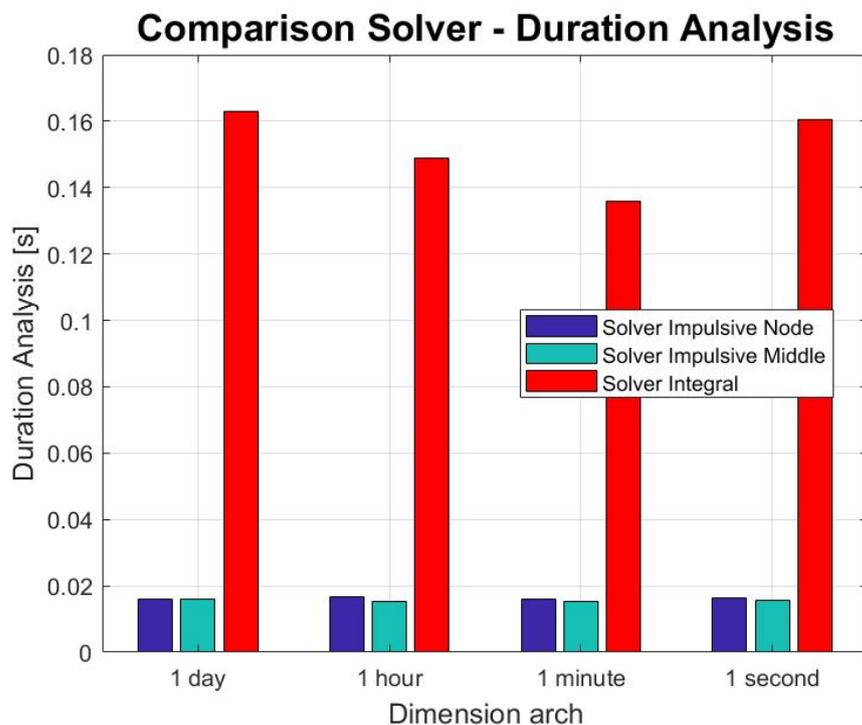
Si è riscontrata un'analogia tra i risultati dei tre risolutori. Come si osserva nei due grafici seguenti, i valori di velocità ricavati sono analoghi, eccetto durante l'utilizzo dell'arco di 86400 secondi.



**Figura 2.8** – Confronto risolutori: modulo posizione (in alto) e modulo velocità (in basso)

L'utilizzo del "Solver Integral" è consigliato per archi di dimensioni ridotte poiché l'approssimazione numerica utilizzata per la risoluzione del sistema di equazioni differenziali di primo ordine (metodo numerico di Runge Kutta di 4° ordine) mantiene un'adeguata accuratezza se implementata per archi di piccole dimensioni.

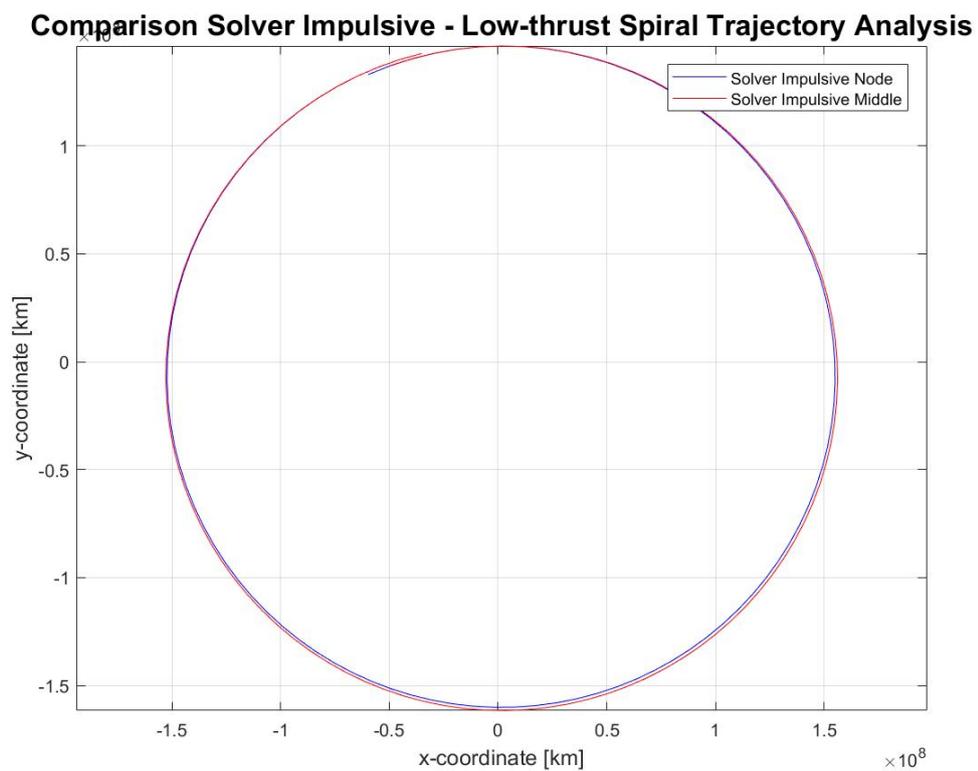
Lo svantaggio principale nell'utilizzo di questo risolutore è però l'elevato costo computazionale. Come si osserva nel grafico seguente, si riscontra una durata delle analisi sul singolo arco ben 10 volte maggiore dei rispettivi risolutori impulsivi.



**Figura 2.9** – Confronto risolutori: durata delle analisi

Nella seconda parte di questo studio, sono stati confrontati i due modelli impulsivi per una modellizzazione più complessa. È stata effettuata l'analisi per una traiettoria suddivisa in 100 archi. Sono stati quindi confrontati i risolutori "Solver Impulsive Node" e "Solver Impulsive Middle" per una manovra a bassa spinta avente una spinta costante di 0.290 N e un impulso specifico costante di 4010 s.

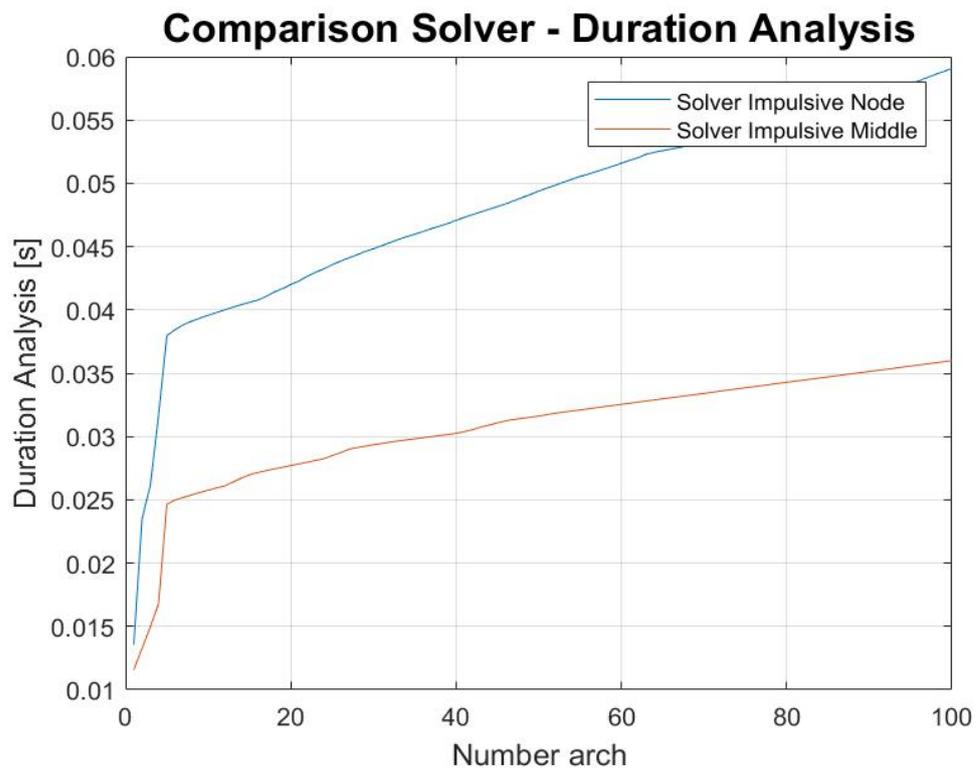
Come si osserva dal grafico in figura 2.10 rappresentante la traiettoria a spirale caratteristica delle low thrust, i due risolutori danno con una buona approssimazione dei risultati analoghi.



*Figura 2.10 – Confronto risolutori impulsivi: traiettoria*

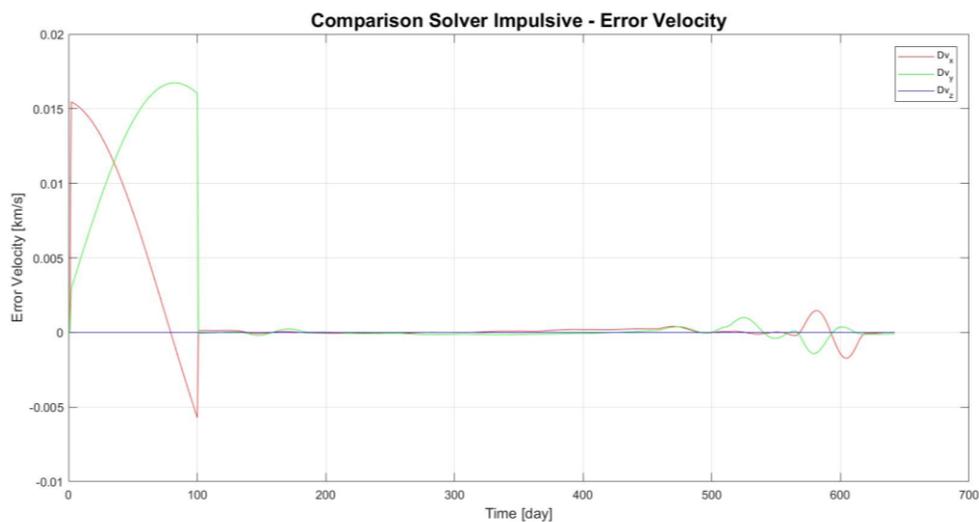
Andando maggiormente nel dettaglio è stato però possibile valutare alcune differenze. La prima differenza riguarda la durata delle analisi.

Come si osserva nel grafico seguente, la durata delle analisi risulta circa il doppio per il risolutore “Solver Impulsive Node”; se pur con valori trascurabili in confronto alla modellizzazione con l’integrazione.



*Figura 2.11 – Confronto risolutori impulsivi: durata delle analisi*

Ma la differenza sostanziale, che causerebbe non poche complicazioni nella successiva fase di ottimizzazione, è rappresentata nel grafico seguente, di figura 2.12 .

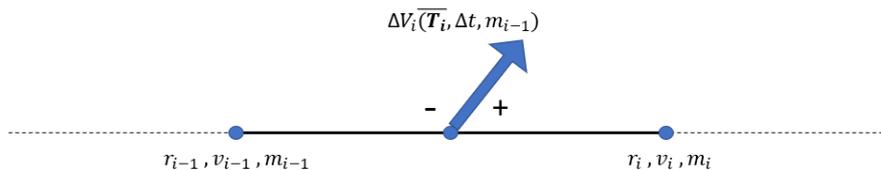


**Figura 2.12** – Confronto risolutori impulsivi: differenza tra i moduli della velocità

Confrontando le componenti dei vettori di velocità di ogni singolo arco, si può osservare un elevato errore nel primo arco per poi annullarsi subito dopo. Ciò è dovuto proprio alla diversa modellizzazione sfruttata. Nello stesso nodo (cioè quello finale dell'arco  $i$ -esimo e quello iniziale dell'arco  $i+1$ ) si avrebbero due porzioni di spinta (metà del  $\Delta V$  complessivo) relative a due integrazioni successive. Ciò potrebbe complicare notevolmente la successiva fase di ottimizzazione.

Per tutti questi motivi appena presentati, la scelta è ricaduta sul modello che approssima la manovra a bassa spinta con una manovra impulsiva effettuata al nodo centrale dell'arco; nello specifico il risolutore “*Solver Impulsive Middle*”.

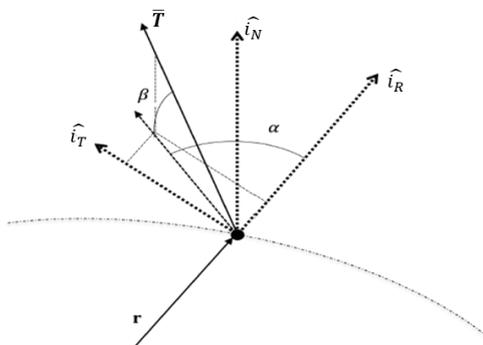
Analizzando il singolo arco, il sistema di equazioni che caratterizza il suddetto modello è il seguente:



**Figura 2.13** – Modello arco *i*-esimo

$$\left\{ \begin{array}{l} r_{i-1}^- = r_{i-1}^-(a_{i-1}, e_{i-1}, i_{i-1}, \omega_{i-1}, \Omega_{i-1}, \theta_{i-1} + \Delta\theta^-) \\ v_{i-1}^- = v_{i-1}^-(a_{i-1}, e_{i-1}, i_{i-1}, \omega_{i-1}, \Omega_{i-1}, \theta_{i-1} + \Delta\theta^-) \\ \\ r_i^+ = r_{i-1}^- \\ v_i^+ = v_{i-1}^- + \Delta V_i(T_i, \alpha_i, \beta_i, \Delta t_i, m_{i-1}) \\ \\ r_i = r_i^+(a_i, e_i, i_i, \omega_i, \Omega_i, \theta_i + \Delta\theta^+) \\ v_i = v_i^+(a_i, e_i, i_i, \omega_i, \Omega_i, \theta_i + \Delta\theta^+) \\ \\ m_i = m_{i-1} - \frac{T_i}{g_0 I_{sp}} \Delta t_i \end{array} \right. \quad (2.38)$$

Opportunamente riformulato si tratta di un sistema di 7 equazioni in 11 incognite ( $\bar{r}_i, \bar{v}_i, m_i, T_i, \alpha_i, \beta_i, \Delta t_i$ ). Ricordare che  $\bar{r}_i, \bar{v}_i$  sono vettori da tre componenti, quindi un totale di 6 variabili.



**Figura 2.14** – Vettore della spinta nel sistema di riferimento RTN

Avendo indicato con  $\alpha_i, \beta_i$  le direzioni rispetto al sistema di riferimento RTN del vettore di spinta  $T_i$ . Nello specifico il vettore risulta composto dalle seguenti componenti:

$$\mathbf{T}_i = T_i [\cos \alpha_i \cos \beta_i, \sin \alpha_i \cos \beta_i, \sin \beta_i] \quad (2.39)$$

Considerando una traiettoria suddivisa in un numero  $N$  di archi con  $N$  numero intero naturale, è necessario introdurre la condizione di congruenza tra i nodi centrali. Ciò coincide con l'affermare che negli  $N-1$  nodi centrali, gli output di un arco coincideranno con gli input dell'arco successivo.

Un problema di questo tipo non avrà una soluzione in forma chiusa. Si tratta di un sistema di  $7N$  equazioni in  $11N$  variabili. Considerando il caso che verrà trattato in questo studio che prevede la dimensione degli archi assegnata ( $N$  variabili) e i vettori di stato del pianeta target assegnati (6 variabili). In definitiva si tratta di un sistema non risolvibile in forma chiusa formato da  $7N$  equazioni in  $10N - 6$  variabili.

Per poter risolvere analiticamente tale sistema è necessario apportare alcune semplificazioni per ridurre il numero di variabili e chiudere il sistema. Una di queste potrebbe essere mantenere il modulo della spinta  $T_i$  variabile 3 volte durante l'intera traiettoria (passando da  $N$  variabili ad 1 variabile), la direzione  $\beta_i$  costante negli  $N$  archi (passando da  $N$  variabili ad 1 variabile) e la direzione  $\alpha_i$  variabile 2 volte durante l'intera traiettoria (anche in questo caso riducendo le variabili da  $N$  a 2). Queste semplificazioni permettono di ottenere un sistema di  $7N$  equazioni in  $7N$  incognite risolvibile in forma chiusa.

## 2.7 Introduzione all'ottimizzazione – Risoluzione di problemi sovradimensionati

Un modo sicuramente meno accurato (che non prevede alcuna soluzione in forma chiusa), ma che non richiede assunzioni semplificative, è l'utilizzo degli algoritmi di ottimizzazione. Ottimizzare una certa funzione che nello specifico sarà il consumo di propellente o la durata della missione, nel rispetto di opportuni vincoli (ad esempio posizione di partenza e di arrivo) sono concetti che stanno alla base del metodo di ottimizzazione che verrà utilizzato nel proseguo delle analisi. Addentrarsi nel dettaglio dei processi di ottimizzazione e nei concetti matematici che ne conseguono, non è lo scopo della presente tesi. Tuttavia, nel prossimo capitolo verranno presentati alcuni dei possibili metodi di ottimizzazione con un particolare richiamo al metodo utilizzato nella tesi.

In generale quindi, ridurre il numero di variabili in gioco rende l'analisi poco veritiera, ed è per tale motivo che sfruttando i potenti algoritmi di ottimizzazione (ad oggi ampiamente analizzati ed implementati) è stato possibile mantenere il numero complessivo di variabili, omettendo qualsiasi ulteriore ipotesi semplificativa.

## 3 Algoritmi di ottimizzazione

### 3.1 Overview

L'ottimizzazione è una pratica utilizzata in moltissimi ambiti e consiste nello sfruttare uno specifico algoritmo che riesca a ricavare il minimo di una funzione scalare a più variabili, chiamata funzione obiettivo. Tale funzione risulta inserita nel contesto di uno specifico "problema" che in precedenza sarà stato opportunamente modellizzato.

Alcune delle funzioni suddette sono ad esempio il peso dell'automobile, il rendimento di un sistema meccanico (in questo caso si parlerà di massimizzarlo), la dissipazione di potenza in un processo, ma in generale si potrebbe estendere a qualsiasi ambito. Sono tutti concetti che vanno inseriti in un contesto di lavoro opportunamente modellizzato, e mediante uno dei tanti algoritmi di ottimizzazione sarà possibile valutare i valori delle variabili che permettano la condizione di ottimo.

Ovviamente la funzione obiettivo potrà avere relazioni matematiche differenti in base alla natura del problema. Nel paragrafo successivo si vedrà nel dettaglio questo concetto introducendo la formulazione matematica generale del problema.

Nella maggior parte dei casi le variabili in gioco non saranno libere di assumere un qualsiasi valore ma saranno vincolate ad un determinato valore, o comprese tra due valori limite; si parlerà rispettivamente di *equality constraints* e di *inequality constraints*. Questo concetto è abbastanza banale immaginando ad esempio che qualora la massa sia una variabile in gioco, risulterebbe surreale far analizzare all'ottimizzatore masse minori di zero. La seguente classificazione si può applicare alla singola variabile (si avrà quindi una variabile fissata nel caso di un *equality constraint*, mentre si parlerà di variabile vincolata agli estremi nel caso di *inequality constraint*) o ad una combinazione lineare o non lineare di più variabili.

I due principali metodi di ottimizzazione sono i metodi diretti e i metodi indiretti che verranno presentati dettagliatamente nel proseguo del capitolo. In termini generali, il metodo diretto e nello specifico il *Non Linear Programming* permette di ricavare la condizione di ottimo discretizzando il "problema" in un elevato numero di parametri, utili a descrivere i controlli interni al sistema. Il metodo indiretto e nello specifico il processo di *Optimal Control* può essere visto come una estensione del NLP in cui vengono utilizzate un numero infinito di variabili. In questo processo si determinano i controlli tramite la soluzione di un problema differenziale ai limiti, risolto tramite successive integrazioni.

Come presentato in precedenza, il processo di ottimizzazione implementato in questo lavoro di tesi permetterà di ricavare una specifica traiettoria a bassa spinta con il più basso consumo di propellente, e che rispetti i vincoli imposti. Volendo presentare alcuni dei vincoli utilizzati, avremo ad esempio il ToF, cioè la durata complessivo della traiettoria che

risulta fissata (la somma di tutti i  $\Delta t$  associati ai vari archi dovrà essere uguale al ToF finale). Un ulteriore vincolo, ad esempio, verrà assegnato agli  $N$  moduli della spinta che risulteranno vincolati in funzione del propulsore utilizzato. Mentre tra i *Non Linear Equality Constraints* vi saranno i vettori di stato del pianeta target di arrivo (che verrà assegnato in input). Nel successivo capitolo, in cui verrà presentata l'implementazione del codice, si rivedranno nel dettaglio questi concetti.

## 3.2 Formulazione matematica generale

Il generico sistema a cui si applica la teoria del controllo ottimale è descritto da un vettore di variabili di stato  $x$ . Le equazioni differenziali che ne descrivono l'evoluzione tra gli istanti iniziale e finale (contorni esterni) sono funzioni di  $x$ , del vettore dei controlli  $u$  e della variabile indipendente  $t$  (tempo), e hanno la generica forma:

$$\dot{x} = f(x(t), u(t), p, t) \quad (3.1)$$

Nella dinamica del problema potrebbero presentarsi anche dei parametri  $p$  non dipendenti da  $t$ . Nel complesso parleremo di un sistema di  $n_x$  equazioni differenziali (con  $n_x$  il numero di componenti del vettore delle variabili di stato e  $n_u$  il numero di componenti del vettore delle variabili di controllo).

Le condizioni al contorno al tempo iniziale  $t_0$  e al tempo finale  $t_f$  sono le seguenti:

$$\psi_{0l} \leq \psi(x(t_0), u(t_0), p, t_0) \leq \psi_{0u} \quad \text{con} \quad \psi(x(t_0), u(t_0), p, t_0) = \psi_0 \quad (3.2)$$

$$\psi_{fl} \leq \psi(x(t_f), u(t_f), p, t_f) \leq \psi_{fu} \quad \text{con} \quad \psi(x(t_f), u(t_f), p, t_f) = \psi_f \quad (3.3)$$

L'obiettivo è quello di determinare la variabile di stato  $x^*$  e la corrispondente legge di controllo ottimale  $u^*$  che minimizzi (o massimizzi) la funzione obiettivo. La formulazione generale di questa funzione prende il nome di Formulazione di Bolza e può essere espressa nella forma:

$$J = \varphi(x_0, x_f, t_0, t_f) + \int_{t_0}^{t_f} \Phi(x, \dot{x}, t) dt \quad (3.4)$$

Nella seguente formulazione la funzione obiettivo  $J$  risulterà dipendente sia da una quantità calcolata in un determinato istante (solitamente al tempo finale  $t_f$ ), ma anche da come la variabile  $x$  e la sua evoluzione nel tempo  $\dot{x}$ , variano tra l'istante iniziale  $t_0$  e l'istante finale  $t_f$ .

Nello specifico si potrà distinguere tra:

- Formulazione di Meyer in cui  $\Phi = 0$

$$J = \varphi(\mathbf{x}_0, \mathbf{x}_f, t_0, t_f) \quad (3.5)$$

- Formulazione di Lagrange in cui  $\varphi = 0$

$$J = \int_{t_0}^{t_f} \Phi(\mathbf{x}, \dot{\mathbf{x}}, t) dt \quad (3.6)$$

La condizione di ottimo dovrà quindi rispettare il sistema di equazioni differenziale e le rispettive condizioni al contorno, ma anche un set di equazioni algebriche legate ai vincoli. In termini generali, evitando la classificazione del paragrafo precedente, si possono racchiudere in un'unica equazione vettoriale:

$$\mathbf{g}_l \leq \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}, t) \leq \mathbf{g}_u \quad (3.7)$$

con  $\mathbf{g}$  il vettore che racchiude tutti i vincoli di dimensione  $n_g$ .

Inoltre, dovranno essere rispettati gli *Upper Bound Constraints* e i *Lower Bound Constraints* delle variabili di stato e delle variabili di controllo:

$$\mathbf{x}_l \leq \mathbf{x}(t) \leq \mathbf{x}_u \quad (3.8)$$

$$\mathbf{u}_l \leq \mathbf{u}(t) \leq \mathbf{u}_u \quad (3.9)$$

### 3.3 Metodo Indiretto

Come osservato nel paragrafo precedente, un problema di ottimizzazione, nel caso più generale può presentare variabili di stato e variabili di controllo come funzioni continue del tempo.

$$\mathbf{x}(t), \mathbf{u}(t)$$

Il metodo indiretto prevede quindi la risoluzione di un sistema di equazioni differenziali con assegnate condizioni al contorno. Vanno quindi ricavate le condizioni necessarie di ottimo da rispettare. Si procede partendo da una soluzione tentativo, e successivamente valutando gli errori effettuati su tali condizioni di ottimo e sulle condizioni al contorno. Successivamente si sostituisce la soluzione di tentativo con l'obiettivo di ridurre questi errori.

Un tipico metodo indiretto è il metodo del controllo ottimale che permette la risoluzione di un problema di ottimo seppur di dimensioni infinite. I metodi pratici per risolvere il problema del controllo ottimale prevedono comunque di convertire (processo di trascrizione) il problema in vari sotto-problemi di dimensione finite e successivamente svolgere una serie di iterazioni risolutive basate sul metodo di Newton (si vedrà nel dettaglio tale metodo nel paragrafo successivo).

Ultimato il processo si valuta l'accuratezza del risultato dato dall'approssimazione a dimensioni finite e, se necessario, si ripetono i passaggi di trascrizione e ottimizzazione.

A tal fine si trascrive la funzione obiettivo introducendo i moltiplicatori di Lagrange, costanti  $\mu$  associati alle condizioni al contorno, e variabili  $\lambda$ , detti anche variabili aggiunte, associati alle equazioni differenziali delle variabili di stato:

$$J^* = \varphi + v^T \psi + \int_{t_0}^{t_f} [\Phi + \lambda^T (f - \dot{x})] dt \quad (3.10)$$

Da notare che  $J^* = J$  per qualsiasi valore di  $v$  e  $\lambda$  se le equazioni di stato e le condizioni al contorno sono rispettate.

Mediante l'utilizzo di un'espansione del primo ordine della funzione obiettivo modificata e differenziando, si ottiene la variazione prima del  $J^*$ , data dalla seguente relazione:

$$\begin{aligned} dJ^* = & \left( \frac{\partial \varphi}{\partial t_f} + v^T \frac{\partial \psi}{\partial t_f} + \Phi_f + \lambda_f^T (f_f - \dot{x}_f) \right) dt_f + \\ & + \left( \frac{\partial \varphi}{\partial t_0} + v^T \frac{\partial \psi}{\partial t_0} + \Phi_0 + \lambda_0^T (f_0 - \dot{x}_0) \right) dt_0 + \\ & + \left( \frac{\partial \varphi}{\partial x_f} + v^T \frac{\partial \psi}{\partial x_f} \right) dx_f + \left( \frac{\partial \varphi}{\partial x_0} + v^T \frac{\partial \psi}{\partial x_0} \right) dx_0 + \\ & + \int_{t_0}^{t_f} \left[ \frac{\partial (\Phi + \lambda^T f)}{\partial x} \delta x + \frac{\partial (\Phi + \lambda^T f)}{\partial u} \delta u - \lambda^T \delta \dot{x} \right] dt \end{aligned} \quad (3.11)$$

Risulta conveniente introdurre l'Hamiltoniano dato dalla seguente relazione:

$$H = \Phi + \lambda^T f \quad (3.12)$$

Sostituendo opportunamente:

$$\begin{aligned}
 dJ^* = & \left( \frac{\partial \varphi}{\partial t_f} + v^T \frac{\partial \psi}{\partial t_f} + H_f \right) dt_f + \left( \frac{\partial \varphi}{\partial t_0} + v^T \frac{\partial \psi}{\partial t_0} - H_0 \right) dt_0 + \\
 & + \left( -\lambda_f^T + \frac{\partial \varphi}{\partial x_f} + v^T \frac{\partial \psi}{\partial x_f} \right) dx_f + \left( \lambda_f^T + \frac{\partial \varphi}{\partial x_0} + v^T \frac{\partial \psi}{\partial x_0} \right) dx_0 + \\
 & + \int_{t_0}^{t_f} \left[ \left( \frac{\partial H}{\partial x} + \lambda^T \right) \delta x + \frac{\partial H}{\partial u} \delta u \right] dt
 \end{aligned} \tag{3.13}$$

La condizione (necessaria) di ottimo prescrive la stazionarietà della funzione obiettivo modificata e quindi richiede che la sua variazione prima per qualsiasi scelta di variazione  $\delta x$ ,  $\delta u$ ,  $dx_0$ ,  $dx_f$ ,  $dt_0$ ,  $dt_f$ , sia uguale a zero.

$$dJ^* = 0 \tag{3.14}$$

Da tale condizione di stazionarietà del  $J^*$ , si estrapolano altre due condizioni necessarie nonché le equazioni di Eulero-Lagrange (funzioni differenziali delle variabili aggiunte)

$$\frac{d\lambda}{dt} = - \left( \frac{\partial \mathbf{H}}{\partial x} \right)^T \tag{3.15}$$

e le equazioni di controllo ottimale (funzioni algebriche delle variabili di controllo)

$$\left( \frac{\partial H}{\partial u} \right)^T = 0 \tag{3.16}$$

Risulta conveniente introdurre una opportuna funzione ausiliaria:

$$\Psi = \varphi + v^T \psi \tag{3.17}$$

utile a definire le condizioni di trasversalità

$$\lambda(t_f) = \Psi_x^T|_{t=t_f} \tag{3.18}$$

$$0 = (\Psi_t + \mathbf{H})|_{t=t_f} \tag{3.19}$$

Il set completo di condizioni necessarie è costituito da un sistema di equazioni algebrico-differenziali DAE – *Differential Algebraic Equations*

$$\begin{cases} \dot{\mathbf{x}} = f(\mathbf{x}(t), \mathbf{u}(t), t) \\ \frac{d\lambda}{dt} = -\left(\frac{\partial \mathbf{H}}{\partial \mathbf{x}}\right)^T \\ \left(\frac{\partial \mathbf{H}}{\partial \mathbf{u}}\right)^T = 0 \end{cases} \quad (3.20)$$

le cui condizioni al contorno sono le seguenti:

$$\psi_{0l} \leq \psi_0 \leq \psi_{0u} \quad (3.21)$$

$$\psi_{fl} \leq \psi_f \leq \psi_{fu} \quad (3.22)$$

$$\lambda(t_f) = \Psi_x^T|_{t=t_f} \quad (3.23)$$

$$0 = (\Psi_t + \mathbf{H})|_{t=t_f} \quad (3.24)$$

In presenza di *equality constraints* il sistema DAE si complica dovendo aggiungere la relazione che regola i vincoli

$$\mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t) = 0 \quad (3.25)$$

In tali condizioni, si avranno due possibilità a seconda del rango della matrice  $\mathbf{g}_u$ , cioè la matrice delle derivate parziali dei vincoli  $\mathbf{g}$  rispetto le variabili di controllo  $\mathbf{u}$ . Se tale matrice avrà rango massimo, avremo un DAE di indice uno, e l'equazione appena introdotta prende il nome di *control variable equality constraint*. In questa circostanza l'Hamiltoniano avrà la seguente formulazione:

$$\mathbf{H} = \Phi + \lambda^T \mathbf{f} + \mu^T \mathbf{g} \quad (3.26)$$

La seconda possibilità è che la matrice  $\mathbf{g}_u$  non sia di rango massimo. In questo caso, differenziando l'equazione del vincolo in funzione del tempo si ottiene:

$$0 = \mathbf{g}_x \dot{\mathbf{x}} + \mathbf{g}_u \dot{\mathbf{u}} + \mathbf{g}_t = \mathbf{g}' \quad (3.27)$$

Utilizzando la nuova funzione  $\mathbf{g}'$  si valuta nuovamente il rango della matrice  $\mathbf{g}'_u$ . Anche in questo caso, se questa matrice avrà rango massimo, allora il DAE da risolvere avrà indice 2, in caso contrario viene ripetuto il procedimento. Da notare che la presenza di un indice elevato aumenta la probabilità di errori numerici.

Non sempre però il rispetto delle condizioni necessarie di ottimo definisce le variabili di controllo ottimali. Un esempio è la relazione  $H_u^T = 0$  che definisce la condizione di controllo ottimale solo se la matrice  $H_{uu}$  sia non singolare; in caso contrario la variabile di controllo  $u$  non risulta definita in modo univoco e si parlerà di arco singolare.

Un'ulteriore complicazione potrebbe verificarsi in presenza degli *inequality constraints*:

$$\mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t) \geq 0 \quad (3.28)$$

La differenza principale nella modalità di approcciarsi a questi vincoli è che, a differenza degli *equality constraints* che devono essere soddisfatti per tutto l'intervallo di tempo, gli *inequality constraints* possono essere attivi ( $\mathbf{g} = 0$ ) o inattivi ( $\mathbf{g} > 0$ ). Andando nel dettaglio vi potranno essere degli archi della traiettoria in cui il vincolo risulta attivo e quindi vincolati, ed altri archi non vincolati. Per gli archi vincolati valgono le relazioni già presentate in precedenza per gli *equality constraints*. Una complicazione, potrebbe presentarsi poiché il numero di sotto archi vincolati non è noto a priori, così come la collocazione dei punti di giunzione in quanto non si conosce il momento in cui si verifica la transizione da arco vincolato a non vincolato.

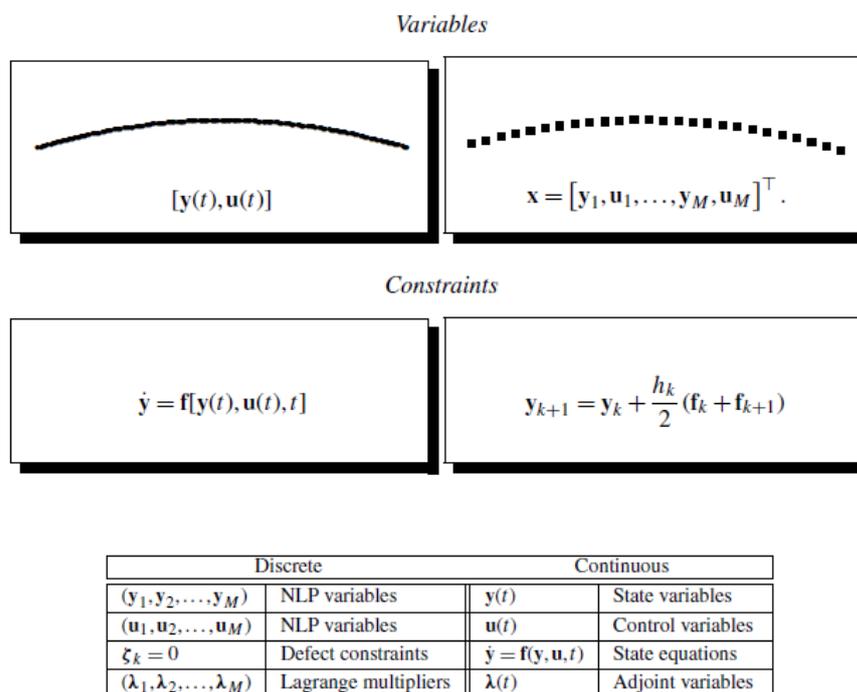
Nel complesso i vantaggi di questo metodo sono la capacità di trattare problemi complessi, un più semplice trattamento dei vincoli e un'elevata robustezza. Di contro presenta una scarsa accuratezza, un elevato costo computazionale. Inoltre, una scelta non adeguata della soluzione iniziale di tentativo potrebbe far ricavare una soluzione finale subottimale.

In conclusione, l'applicazione di un metodo indiretto prevede la risoluzione del sistema DAE appena presentato mediante un'opportuna approssimazione numerica, quale può essere ad esempio quella ottenibile dall'applicazione del metodo di Runge-Kutta. In ultima analisi, quindi il problema continuo andrà comunque discretizzato ai fini della risoluzione del sistema di equazioni differenziali. Impropiamente viene considerato il metodo indiretto come un metodo applicabile solo per problemi di tipo continuo, distinguendolo dal metodo diretto che prevede la risoluzione di problemi a variabili discrete. Come è stato possibile osservare, il metodo del controllo ottimale prevedere il raggiungimento della soluzione ottimale, senza mai calcolare la funzione obiettivo da minimizzare. Si procede valutando gli errori sulle condizioni di ottimo appena presentate e modificando la soluzione tentativo ai fini di ridurre questi errori. Nel prossimo paragrafo verranno presentati i metodi diretti che in contrasto al metodo appena presentato, ricava la soluzione ottimale calcolando direttamente la funzione obiettivo ad ogni iterazione e valutandone l'andamento (da qui il nome del metodo).

### 3.4 Metodo Diretto

A differenza dei metodi indiretti, questi processi di ottimizzazione prevedono la presenza di un numero finito di variabili. L'andamento delle variabili e dei vincoli annessi non si presenta in forma continua ma in forma discreta. In generale quasi tutti i metodi numerici utilizzati per risolvere un problema di ottimo, prevedono un processo iterativo suddividendo il problema in tanti sotto-problemi di dimensioni finite (discretizzazione del problema con un numero di incognite finito).

Un tipico metodo diretto è il *Non Linear Programming* o metodo NLP. Il seguente metodo permette di trovare un numero finito di variabili tali da ottimizzare una specifica funzione obiettivo senza violare una serie di vincoli. Tra i metodi NLP vi sono alcuni casi particolari come il metodo della programmazione lineare (LP), della programmazione quadratica (QP) e il metodo dei minimi quadrati.



**Figura 3.1** – Discretizzazione vs. formulazione continua<sup>18</sup>

<sup>18</sup> Fonte vedere riferimento bibliografico [5]

L'approccio fondamentale per la maggior parte degli schemi iterativi utilizzati in questi metodi è stato suggerito oltre 300 anni fa da Newton. Esso formulò un metodo, chiamato metodo di Newton, che permette mediante un processo iterativo di calcolare le soluzioni di una funzione detti zeri o radici della funzione. Si consideri ad esempio un caso banale ad una variabile ad esempio la relazione di un vincolo scritta nella forma:

$$c(x) = 0 \quad (3.29)$$

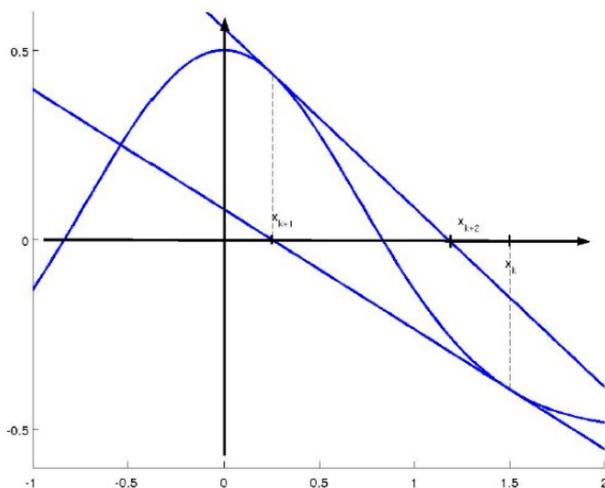
Partendo da una soluzione di tentativo  $x_k$  si approssima questa funzione non lineare del vincolo con i primi due termini dell'espansione in serie di Taylor rispetto la soluzione  $x_k$ . Questo produce un'approssimazione lineare della funzione scritta nella forma:

$$c(x) = c(x_k) + c'(x_k)(x - x_k) \quad (3.30)$$

In termini geometrici è stata approssimata localmente la funzione  $c(x)$  con una retta passante per il punto  $(x_k, c(x_k))$ .

Basterà quindi trovare la radice di tale retta, ottenendo il punto  $x_{k+1}$ , come è possibile osservare in figura 3.2 .

$$x_{k+1} = x_k - \frac{c(x_k)}{c'(x_k)} \quad (3.31)$$



**Figura 3.2** – Metodo di Newton: rappresentazione geometrica

Si ripete l'iterazione fino a quando non si trova la radice  $x^*$  della funzione di partenza. Questo metodo prevede l'utilizzo della funzione e della sua derivata ad ogni iterazione.

Per ovviare al seguente problema si approssima la derivata mediante il metodo delle differenze finite:

$$c'(x) = \frac{c(x_k) - c(x_{k-1})}{x_k - x_{k-1}} = \frac{\Delta c}{\Delta x} \quad (3.32)$$

Con tale correzione il metodo prende il nome di metodo di Newton-like o metodo delle secanti.

Questo concetto viene ampliato anche nel caso in cui la radice  $x^*$  che si vuole ottenere è la soluzione del problema di ottimo, ad esempio il minimo della funzione obiettivo  $J(x)$ .

Come primo passo va approssimata la funzione con una espansione in serie di Taylor troncata al terzo ordine.

$$J(x) = J(x_k) + J'(x_k)(x - x_k) + \frac{1}{2}J''(x_k)(x - x_k)^2 \quad (3.33)$$

Non è possibile utilizzare un'approssimazione lineare, poiché non presenterebbe alcuno punto stazionario di minimo. Per trovare il punto di minimo è necessario porre a zero la derivata prima della funzione appena approssimata:

$$\frac{dJ(x)}{dx} = J'(x) = J'(x_k) + J''(x_k)(x - x_k) = 0 \quad (3.34)$$

La cui risoluzione prevede un processo iterativo analogo a quello presentato in precedenza:

$$x_{k+1} = x_k - \frac{J'(x_k)}{J''(x_k)} \quad (3.35)$$

Visto in altri termini, ad ogni passo iterativo viene calcolato l'incremento da apportare alla soluzione di tentativo per ripetere l'iterazione.

$$p = -\frac{J'(x_k)}{J''(x_k)} \quad \rightarrow \quad x_{k+1} = x_k + p \quad (3.36)$$

Da notare che la risoluzione di questo metodo permette di trovare il punto stazionario senza distinzione tra minimo e massimo. Per fare ciò, è necessario valutare il valore della derivata seconda della funzione  $J(x)$ .

Nello specifico, si avrà un punto di minimo se sono rispettate le seguenti condizioni necessarie:

$$J'(x^*) = 0 \quad (3.37)$$

$$J''(x^*) \geq 0 \quad (3.38)$$

Si parlerà di condizioni sufficienti se  $J''(x^*)$  sarà strettamente maggiore di zero. Il rispetto di una o dell'altra condizione sta alla base della differenza tra un minimo locale e un minimo globale.

Riportando il seguente metodo in ambito multi dimensionale, in ipotesi di ottimizzazione non vincolata, è possibile introdurre la seguente formulazione matematica.

Sia  $\mathbf{x} = [x_1, \dots, x_k, \dots, x_n]^T$  e conseguentemente  $J(\mathbf{x}) = J(x_1, \dots, x_k, \dots, x_n)$  e sia  $\tilde{\mathbf{x}}$  la soluzione tentativo di partenza:

$$J(\mathbf{x}) = J(\tilde{\mathbf{x}}) + \mathbf{g}^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T \mathbf{H} \mathbf{p} \quad (3.39)$$

Avendo indicato con  $\mathbf{p}$  il vettore che definisce la *search direction*:

$$\mathbf{p} = \mathbf{x} - \tilde{\mathbf{x}} \quad (3.40)$$

mentre con  $\mathbf{g}$  e con  $\mathbf{H}$  rispettivamente il gradiente e la matrice Hessiana della funzione obiettivo:

$$\mathbf{g}(\mathbf{x}) = \nabla_{\mathbf{x}} J = \begin{bmatrix} \frac{\partial J}{\partial x_1} \\ \vdots \\ \frac{\partial J}{\partial x_n} \end{bmatrix} \quad (3.41)$$

$$\mathbf{H} = \nabla_{\mathbf{x}\mathbf{x}}^2 J = \begin{bmatrix} \frac{\partial^2 J}{\partial x_1^2} & \frac{\partial^2 J}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 J}{\partial x_1 \partial x_n} \\ \vdots & \ddots & & \vdots \\ \frac{\partial^2 J}{\partial x_n \partial x_1} & \frac{\partial^2 J}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 J}{\partial x_n^2} \end{bmatrix} \quad (3.42)$$

Per completezza verrà introdotta anche la matrice Jacobiana, presente ad esempio nel caso del vettore dei vincoli  $c(x)$ :

$$\mathbf{G}(x) = \frac{\partial c}{\partial x} = \begin{bmatrix} \frac{\partial c_1}{\partial x_1} & \frac{\partial c_1}{\partial x_2} & \dots & \frac{\partial c_1}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial c_n}{\partial x_1} & \frac{\partial c_n}{\partial x_2} & \dots & \frac{\partial c_n}{\partial x_n} \end{bmatrix} \quad (3.43)$$

Varranno quindi le stesse condizioni necessarie riguardanti i valori del gradiente e dell'Hessiana calcolati nella condizione di ottimo  $x^*$ .

$$g(x^*) = 0 \quad (3.44)$$

$$\mathbf{p}^T \mathbf{H}(x^*) \mathbf{p} \geq 0 \quad (3.45)$$

Da notare che il rispetto della condizione di ottimo  $g(x^*) = 0$  può essere analizzata applicando il metodo di Newton. In questo modo si otterrà:

$$g + \mathbf{H}\mathbf{p} = 0 \quad (3.46)$$

Da cui è possibile calcolare la *search direction*:

$$\mathbf{p} = -\frac{g}{\mathbf{H}} \quad (3.47)$$

Come nel caso ad una variabile, le derivate verranno approssimate con un opportuno metodo numerico, primo tra tutti il metodo alle differenze finite, che sfrutta i risultati dell'iterazione precedente per valutare le derivate delle varie grandezze (siano esse contenute in matrici Jacobiane, Hessiane, o gradienti).

### 3.4.1 Equality constraint

In presenza di *equality constraints* è necessario introdurre alcuni concetti, tra cui la Lagrangiana. Consideriamo una funzione obiettivo  $J(x)$  soggetta ad *equality constraint* del tipo:

$$c(x) = 0 \quad (3.48)$$

con  $x$  vettore delle variabili di stato formato da  $n_x$  componenti e  $c$  il vettore dei vincoli di dimensione  $n_c \leq n_x$ .

In tali condizioni, l'approccio più semplice è introdurre la Lagrangiana che è definita dalla seguente relazione:

$$L(x, \lambda) = J(x) - \lambda^T c(x) = J(x) - \sum_{i=1}^m \lambda_i c_i(x) \quad (3.49)$$

con  $\lambda$  il vettore degli  $n_c$  moltiplicatori di Lagrange.

In modo analogo ai casi non vincolati presentati in precedenza, le condizioni necessarie di ottimo rimangono le stesse, con l'unica differenza che le variabili in gioco saranno sia il vettore delle variabili di stato sia il vettore dei moltiplicatori di Lagrange. Le condizioni necessarie affinché il punto  $(x^*, \lambda^*)$  sia di ottimo sono:

$$\nabla_x L(x^*, \lambda^*) = 0 \quad (3.50)$$

$$\nabla_\lambda L(x^*, \lambda^*) = 0 \quad (3.51)$$

Questi gradienti sono facilmente ricavabili e possono essere scritti nella seguente forma:

$$\nabla_x L = g - G^T \lambda = \nabla J - \sum_{i=1}^m \lambda_i \nabla c_i \quad (3.52)$$

$$\nabla_\lambda L = -c(x) \quad (3.53)$$

Come è stato presentato nel paragrafo precedente, queste condizioni permettono di definire un punto stazionario della Lagrangiana senza poter definire con esattezza se si tratta di un punto di massimo o minimo. Questa condizione è ricavabile analizzando la curvatura della funzione da ottimizzare e per far ciò è necessario introdurre l'Hessiana della Lagrangiana che è data dalla seguente relazione:

$$H_L = \nabla_{xx}^2 L = \nabla_{xx}^2 J - \sum_{i=1}^m \lambda_i \nabla_{xx}^2 c_i \quad (3.54)$$

L'ulteriore condizione necessaria affinché il punto sia di minimo è la seguente:

$$v^T H_L v > 0 \quad (3.55)$$

Da questa relazione si nota una tra le sostanziali differenze con il metodo non vincolato. Mentre nel caso non vincolato si richiede che la curvatura sia positiva in tutte le direzioni  $\mathbf{p}$ , nel caso vincolato la condizione è più restrittiva proprio perché è necessario che vengano

soddisfatti anche i vincoli. Infatti, l'equazione appena presentata va applicata solo nelle direzioni  $\mathbf{v}$  cioè tutte quelle direzioni tangenti allo spazio contenente i vincoli.

In analogia con le precedenti analisi, le due relazioni che annullano i gradienti possono essere risolte applicando il metodo di Newton, quindi approssimando le due relazioni con una espansione di Taylor troncata al primo ordine. Queste due condizioni così approssimate possono essere scritte nella forma seguente che prende il nome di Kuhn-Tucker (KT) o Karush-Kuhn-Tucker (KKT) system:

$$\begin{bmatrix} H_L & G^T \\ G & 0 \end{bmatrix} \begin{bmatrix} -\mathbf{p} \\ \tilde{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ c \end{bmatrix} \quad (3.56)$$

Ricordando che  $\mathbf{p}$  indica la *search direction* ( $\mathbf{p} = \mathbf{x} - \tilde{\mathbf{x}}$ ) e  $\tilde{\lambda}$  è il vettore dei moltiplicatori di Lagrange al nuovo valore di tentativo  $\tilde{\mathbf{x}}$  dell'iterazione.

Un ulteriore metodo per valutare la direzione  $\mathbf{p}$  è minimizzare la seguente forma quadratica:

$$\mathbf{g}^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T H \mathbf{p} \quad (3.57)$$

soggetta ai seguenti vincoli lineari

$$G \mathbf{p} = c \quad (3.58)$$

### 3.4.2 Inequality constraint

Si consideri adesso il caso in presenza di *inequality constraints* della forma:

$$\mathbf{c}(\mathbf{x}) \geq 0 \quad (3.59)$$

Da notare che in questo caso non è strettamente necessario che le componenti del vettore dei vincoli siano minori o uguali alle componenti del vettore delle variabili di stato.

In questo ambito è necessario fare due distinzioni; ricordando che con  $\mathbf{x}^*$  si considera la soluzione ottimale, si avrà:

- Vincoli attivi cioè quei vincoli che vengono soddisfatti da questa relazione:

$$\mathbf{c}(\mathbf{x}^*) = 0 \quad (3.60)$$

- Vincoli inattivi cioè quei vincoli che vengono soddisfatti da quest'altra relazione:

$$\mathbf{c}(\mathbf{x}) > 0 \quad (3.61)$$

Risulta ovvio notare che i vincoli attivi possono essere trattati come *equality constraints*, ma risulta altrettanto ovvio valutare quali di questi *inequality constraints* possono essere trattati come attivi e quali come inattivi. Ogni algoritmo di ottimizzazione prevede un meccanismo per valutare questa distinzione chiamato *active set strategy*.

Un algoritmo spesso utilizzato tra i metodi diretti è il *Quadratic Programming*. Il problema QP prevede una funzione obiettivo quadratica e dei vincoli lineari (*equality* e *inequality*). In questi casi, l'algoritmo che si utilizza è il seguente:

1. Calcolare il minimo della funzione quadratica obiettivo che rispetti i vincoli (*equality* più gli *inequality* attivi); viene fatto risolvendo il sistema KKT prima presentato.
2. Scegliere la *search direction*  $p$  più ampia possibili che non violi i vincoli inattivi

$$\tilde{x} = x + \alpha p \quad (3.62)$$

Il tutto quindi ricade nella scelta del parametro  $\alpha$  che viene chiamato *steplenght* ed avrà un valore compreso tra 0 e 1.

3. Nel caso in cui la scelta dello *steplenght* è ricaduta su un valore minore di 1, bisogna aggiungere il vincolo che limita  $\alpha$  tra i vincoli attivi e riiniziare il processo;
4. Nel caso in cui è stato utilizzato  $\alpha = 1$  bisogna controllare il segno dei moltiplicatori di Lagrange:
  - Se tutti gli *inequality constraints* hanno moltiplicatori di Lagrange di segno positivo, è possibile terminare il processo;
  - In caso contrario bisogna eliminare tra i vincoli attivi quello che ha associato il moltiplicatore di Lagrange con il valore negativo più ampio e ritornare allo step 1.

Come visto per gli archi singolari, vi sono molte complicazione nell'utilizzo di questi metodi, che introducono non poche carenze nella capacità di convergenza del metodo. Per questo motivo sono state studiate varie strategie che prendono il nome di *Globalization Strategies*. Esse hanno il compito di valutare i problemi nell'applicazione del metodo di Newton e correggerne le carenze all'iterazione successiva.

La strategia più comune è quella che prevede l'utilizzo della funzione di merito  $M$ . Un tipico esempio è l'utilizzo della funzione di merito per applicare il *Line Search Method* che prevede di "aggiustare" lo *steplenght* ad ogni iterazione al fine di ottenere una "sufficiente" riduzione della funzione di merito che sarà una combinazione tra la funzione obiettivo e le violazioni dei vincoli (opportunamente pesati con relative costanti). La stesura di questa funzione è ampiamente complessa e vi sono vari approcci a riguardo (ad esempio la funzione di merito lagrangiana aumentata, e altre varianti proposte nel mondo accademico).

Un differente approccio è il *Trust Region Method* utilizzato per modificare la *search direction*  $p$  ottenuta dall'applicazione del metodo di Newton. In termini generali, questo metodo permette di effettuare una modifica delle variabili di stato all'interno di un range limite di correzione  $\Delta x$  posto all'interno di una regione di raggio  $\delta$  definito *trust radius*.

Nel complesso i metodi diretti presentano un'elevata accuratezza, un basso costo computazionale e quindi una bassa durata delle analisi. Di contro spesso prevedono un approccio preliminare più accurato nel trattare alcune equazioni e alcuni vincoli più complessi. Inoltre, come per i metodi indiretti, presentano il problema legato alla scelta della soluzione tentativo di partenza che potrebbe causare la convergenza verso una soluzione sub-ottimale.

### 3.5 Metodo di ottimizzazione utilizzato – SQP Method

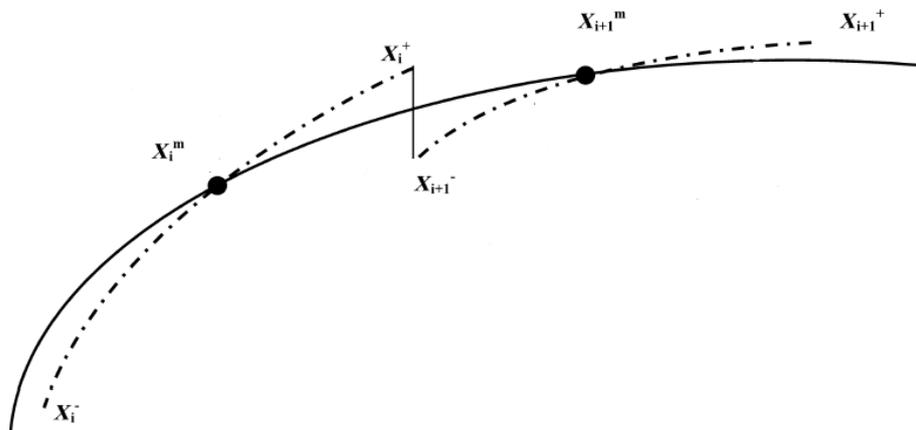
Prima di introdurre il metodo utilizzato nelle analisi svolte in questo lavoro di tesi, per completezza vengono presentati due metodi alternativi di approccio al problema analogamente per un metodo diretto o indiretto.

Come visto nel capitolo 2, il problema delle Low Thrust, che è un tipico problema a variabili continue, è stato discretizzato suddividendo la traiettoria in  $N$  archi, ed in ogni arco è stato possibile utilizzare variabili discrete come valori di spinta e direzione della Low Thrust costanti. Mediante il risolutore *Solver Impulsive Middle* queste variabili, ad esempio le componenti della bassa spinta, sono state ricondotte ad una manovra impulsiva applicata al nodo centrale di ogni arco.

Il metodo diretto, così come il metodo indiretto, prevedono però due approcci differenti. Suddividendo il problema in sotto archi, è possibile imporre la continuità tra il nodo finale dell'arco  $i$ -esimo e il nodo iniziale nell'arco  $i+1$ . Quello che è stato fatto nelle analisi è imporre tale condizioni di continuità già in fase di modellizzazione, senza che le seguenti equazioni influiscano il processo di ottimizzazione. Si parla in questo caso di metodo *Direct Shooting*.

Un ulteriore approccio è inserire tali condizioni di continuità all'interno del processo di ottimizzazione. In questo contesto si consideri ad esempio di inserire tra le variabili di stato anche i vettori di stato al nodo iniziale di ogni arco (componenti dei vettori posizione e/o velocità). Il processo di ottimizzazione prevede quindi di modificare le condizioni iniziali di ogni arco, e valutare l'errore tra posizione iniziale di tentativo e posizione finale ottenuta dal risolutore per ogni singolo arco.

È possibile osservare questo concetto nella figura 3.3 in cui  $x_i^-$  e  $x_{i+1}^-$  indicano la posizione di “tentativo” (cioè fornita dall’ottimizzatore) del rispettivo arco. Inoltre, si notano le variabili  $x_i^m$  e  $x_{i+1}^m$  che coincidono con i vettori della thrust al nodo centrale (anch’esse variabili di “tentativo” ricavate dall’ottimizzatore ad ogni iterazioni), e le variabili  $x_i^+$  e  $x_{i+1}^+$  che sono gli output del risolutore.



**Figura 3.3** – Multiple Shooting<sup>19</sup>

Sarà compito dell’ottimizzatore regolare l’errore sulla posizione degli archi nel rispetto dei vincoli

$$x_i^+ = x_{i+1}^- \quad (3.63)$$

Questo metodo è chiamato *Multiple Shooting*. Viene spesso utilizzato perché permette in molti casi una convergenza più veloce, ma di contro presenta un numero maggiore di variabili in gioco. In questo studio è stato utilizzato un *Direct Shooting*, tenendo in considerazione la futura possibilità di implementare un *Multiple Shooting* per analizzarne le differenze in termini di accuratezza, di costo computazionale e durata delle analisi.

L’aver modellizzato il problema in questo modo, induce l’utilizzo di un metodo diretto che, come visto in precedenza, si presta maggiormente a problemi con variabili discrete. Nello specifico l’algoritmo utilizzato è il *Sequential Quadratic Programming* o SQP method. Questo metodo è basato su una approssimazione quadratica della funzione obiettivo e una approssimazione lineare dei vincoli, e prevede la risoluzione di un problema NLP tramite la risoluzione di una successione di sotto problemi QP. Precisamente, si risolve una

<sup>19</sup> Fonte vedere riferimento bibliografico [6]

successione di problemi di programmazione quadratica, in cui l'Hessiana della funzione obiettivo da minimizzare corrisponde con la Hessiana della Lagrangiana (prima presentata) ed i vincoli sono l'approssimazione del primo ordine dei vincoli effettivi.

Questo algoritmo prevede l'utilizzo delle *Globalization Strategies*, e per tale motivo la convergenza si ottiene minimizzando simultaneamente anche una opportuna funzione di merito (sfruttando il metodo della *Line Search*).

L'implementazione di questo metodo è stata effettuata mediante l'utilizzo di una funzione di Matlab denominata *fmincon*. Con la suddetta funzione è possibile risolvere un problema di ottimo (valutazione del minimo) per un problema vincolato non lineare scritto nella forma:

$$\min_x J(x) \text{ tale che } \begin{cases} c(x) \leq 0 \\ ceq(x) = 0 \\ A x \leq b \\ Aeq x = beq \\ lb \leq x \leq ub \end{cases} \quad (3.64)$$

È stata quindi utilizzata la seguente funzione, avendo settato l'utilizzo dell'algoritmo SQP per la risoluzione di questo problema NLP.

Per completezza si introducono nel dettaglio alcune relazioni utilizzate dalla *fmincon* nella risoluzione dell'algoritmo SQP. Un esempio è la relazione utilizzata per approssimare l'Hessiana della Lagrangiana alla  $k$ -esima iterazione che viene calcolata mediante il metodo BFGS. Si utilizza infatti la seguente Hessiana modificata data dalla relazione:

$$H_{k+1} = H_k + \frac{q_k q_k^T}{q_k^T p_k} - \frac{H_k p_k p_k^T H_k^T}{p_k^T H_k p_k} \quad (3.65)$$

Avendo indicato con  $p_k$  la *search direction* all'iterazione  $k$ -esima cioè  $p_k = x_{k+1} - x_k$ , e con  $q_k$  la differenza tra i gradienti della Lagrangiana all'iterazione  $k$ -esima, definito dalla relazione:

$$\begin{aligned} q_k &= \nabla_x L_{k+1} - \nabla_x L_k = \\ &= \left( \nabla J(x_{k+1}) + \sum_{i=1}^m \lambda_i \nabla c_i(x_{k+1}) \right) - \left( \nabla J(x_k) + \sum_{i=1}^m \lambda_i \nabla c_i(x_k) \right) \end{aligned} \quad (3.66)$$

La necessità di avere una matrice Hessiana definita positiva ad ogni iterazione, è garantita mantenendo  $q_k^T s_k > 0$  ad ogni iterazione. Qualora ciò non avvenisse, il primo passo è dimezzare il valore del componente di  $q_k$  maggiormente negativo, fino a garantire la condizione  $q_k^T s_k > 0$ . Ciò avviene fino a quando  $q_k^T s_k$  è maggiore o uguale ad una piccola

tolleranza negativa (legata ad un parametro denominato *ConstraintTolerance* che verrà presentato nel capitolo successivo).

Se, dopo questa procedura  $q_k^T s_k$  non è ancora positivo, viene effettuata una modifica forzata del  $q_k$  aggiungendo un vettore  $v$  moltiplicato per uno scalare costante  $w$ :

$$q_k = q_k + wv \quad (3.67)$$

Dove  $v_i = \nabla c_i(x_{k+1}) c_i(x_{k+1}) - \nabla c_i(x_k) c_i(x_k)$ , mentre  $w$  è uno scalare che va aumentato sistematicamente fino a quando non viene rispettata la condizione  $q_k^T s_k > 0$ .

La funzione di merito utilizzata è invece la seguente:

$$M(x) = J(x) + \sum_{i=1}^{m_e} r_i c_i(x) + \sum_{i=m_e+1}^m r_i \max[0, c_i(x)] \quad (3.68)$$

Avendo indicato con  $m_e$  il numero di vincoli dati dalla somma tra *equality constraints* e *inequality constraints* attivi, ed  $m$  il numero di vincoli complessivo. Il parametro  $r$  indica il parametro di penalità che “pesa” la correzione dell’errore sui vari vincoli al fine di effettuare un’adeguata scelta dello *steplenght*  $\alpha$ .

$$r_i = (r_{k+1})_i = \max_i \left\{ \lambda_i, \frac{(r_k)_i + \lambda_i}{2} \right\} \quad i = 1, \dots, m \quad (3.69)$$

Nel capitolo successivo verrà presentato il software implementato durante il lavoro di tesi. Verranno introdotte le modalità con cui è stata sfruttata la funzione *fmincon* e le specifiche analisi effettuate per un corretto settaggio dell’ottimizzatore. In conclusione, verrà presentata la modalità di validazione del software mediante l’utilizzo dei dati ricavati da uno studio dell’Agenzia Spaziale Europea, svolto durante la fase di progetto della missione Mars Sample Return.

## 4 Applicazione alla missione Mars Sample Return

### 4.1 Introduzione codice

Con l'obiettivo di sviluppare un codice di ottimizzazione per il problema delle manovre a bassa spinta, è stato utilizzato l'*Optimization Tool* di Matlab e nello specifico la funzione *fmincon*. Sfruttando la seguente funzione che risolve problemi NLP è stato possibile creare un software per modellizzare e ottimizzare manovre a bassa spinta in ambito di missioni interplanetarie. Il codice che verrà presentato nel seguente capitolo è stato denominato DOPLLOT – Design and OPTimization of LOW-thrust Trajectory.

Nel proseguo del capitolo verranno presentate le formulazioni matematiche e i concetti teorici che hanno permesso l'implementazione del codice. Successivamente sono state effettuate delle analisi per valutare il corretto settaggio della funzione *fmincon* al fine di migliorare la convergenza del codice. Il tutto è stato effettuato sfruttando come dati di riferimento quelli estrapolati dal MAG ESA della missione Mars Sample Return. Proprio il documento di Mission Analysis Guidelines della missione MSR redatto dall'ESA in data 28/03/2017 è stato fondamentale per la validazione del software DOPLLOT.

Come presentato nel capitolo introduttivo, MSR è una missione che prevedrà il prelevamento di campioni di roccia e polvere dalla superficie marziana, e il rientro sul pianeta Terra. La realizzazione di questa missione sarà sicuramente un passo avanti verso la realizzazione di future missioni di esplorazione umana su terreno marziano. Dai primi studi, si è sempre più diffusa la possibilità di utilizzare uno scenario che preveda una propulsione primaria elettrica per ambe due i trasferimenti (Terra-Marte e il rientro Marte-Terra). Per la validazione del software sono state utilizzate le analisi sulle possibili traiettorie di rientro Marte-Terra.

La formulazione matematica generale associata alla funzione *fmincon* per la risoluzione di ottimo può essere riassunta nella seguente formulazione 3.64, già vista nel capitolo 3:

$$\min_x J(x) \text{ tale che } \begin{cases} c(x) \leq 0 \\ ceq(x) = 0 \\ A x \leq b \\ Aeq x = beq \\ lb \leq x \leq ub \end{cases} \quad (3.64)$$

Nel prossimo paragrafo verrà presentata l'implementazione specifica per il caso in esame in relazione alla formulazione appena presentata.

### 4.1.1 Formulazione matematica DOPLLOT

Sfruttando il risolutore *Solver Impulsive Middle*, è stata suddivisa la traiettoria complessiva in un numero finito  $N$  di archi. Ad ogni arco  $N$ -esimo si considera un valore di spinta costante e viene approssimato ad una spinta impulsiva al nodo centrale dell'arco.

La funzione da ottimizzare sarà la somma di questi impulsi, nonché il  $\Delta V$  complessivo associato a questa traiettoria.

$$J(x) = \Delta V_{Tot} = \sum_{i=1}^N \Delta V_i \quad (4.1)$$

Con  $\Delta V_i$  viene indicato l'impulso associato al singolo arco, il quale è dato dalla relazione:

$$\Delta V_i = J(x) = \Delta V_{Tot} = \sum_{i=1}^N \Delta V_i \quad (4.2)$$

Il risolutore *Solver Impulsive Middle* prevede l'ipotesi semplificativa secondo la quale le componenti del vettore delle variabili di stato e la massa, risultano costanti per ogni arco. Ciò permette di semplificare l'equazione 4.2, ottenendo la seguente relazione algebrica per l'impulso  $i$ -esimo.

$$\Delta V_i = \frac{x_{3i-2}}{m_i} \Delta t_i \quad (4.3)$$

avendo indicato con  $x$  il vettore di stato dell'ottimizzazione le cui componenti associate all' $i$ -esimo arco sono le seguenti:

$$x = (T, \alpha, \beta)_i \quad \text{con} \quad 1 \leq i \leq N \quad (4.4)$$

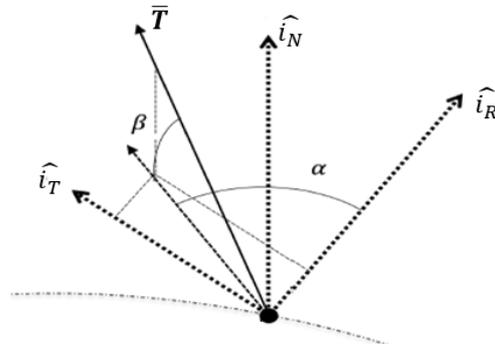


Figura 4.1 – Componenti del vettore  $i$ -esimo della spinta

Come si può osservare in figura 4.1, queste componenti sono riferite al sistema di riferimento RTN, dove la componente T indica il modulo del vettore di spinta,  $\alpha$  la direzione sul piano orbitale ( $\alpha = 0$  coincide con la direzione radiale quindi concorde con la velocità dello S/C) e  $\beta$  l'angolo fuori dal piano.

La variabile di stato sarà quindi formata da  $3N$  componenti. La componente  $x_{3i-2}$  con  $1 \leq i \leq N$  indica i moduli della spinta ad ogni arco. Questo valore, diviso il valore della massa  $m$  definisce l'accelerazione associata a quella spinta. Andrà quindi moltiplicata per la durata dell'arco considerato  $\Delta t_i$  ricavando il valore del  $\Delta V_i$  dell'arco  $i$ -esimo. Da sottolineare che il valore del  $\Delta t$  è previsto in input al codice, in modo tale che  $\Delta t = \Delta t_i$  cioè archi di ampiezza costante lungo la traiettoria.

Per un corretto processo di ottimizzazione è di fondamentale importanza la scelta dei vincoli. Sono stati quindi scelti dei *Non Linear Equality Constraints* per fissare il target di arrivo della traiettoria. È stato imposto che i vettori di stato (posizione e velocità) relativi alla condizione di arrivo del pianeta target siano uguali ai rispettivi vettori di stato ricavati dall'ottimizzatore (posizione e velocità dell'ultimo nodo della traiettoria). Rispettivamente saranno presenti 6 *Non Linear Equality Constraints*, 3 per le componenti della velocità e 3 per le componenti dei vincoli.

$$ceq(x) = \begin{bmatrix} r_N(1) - r_f(1) \\ r_N(2) - r_f(2) \\ r_N(3) - r_f(3) \\ v_N(1) - v_f(1) \\ v_N(2) - v_f(2) \\ v_N(3) - v_f(3) \end{bmatrix} = 0 \quad (4.5)$$

Avendo indicato con  $r_N, v_N$  i vettori posizione e velocità in uscita dall'ottimizzatore all' $N$ -esimo arco, ed  $r_f, v_f$  i vettori di posizione e velocità del target alla data di arrivo.

Ulteriori vincoli introdotti sono i *Lower Bound Constraints* e gli *Upper Bound Constraints* cioè i valori al contorno delle variabili di stato. Sono state considerati i seguenti valori limite per le tre componenti del vettore  $x_i$  associate all'arco  $i$ -esimo.

$$0 \leq T_i \leq T_{max} \quad (4.6)$$

$$-90^\circ \leq \beta_i \leq +90^\circ \quad (4.7)$$

$$-180^\circ \leq \alpha_i \leq +180^\circ \quad (4.8)$$

Non è stato invece necessario introdurre dei *Linear Equality Constraints* e dei *Linear Inequality Constraints*. Rispettivamente si tratta dei vincoli scritti nella forma  $Aeq\ x = beq$  e nella forma  $Ax \leq b$ . Analogamente per gli i vincoli non lineari non è stato necessario introdurre dei Non Linear Inequality ( $c(x) \leq 0$ ).

#### 4.1.2 Parametri della modellizzazione

Scelta la formulazione matematica che verrà utilizzata per il problema da ottimizzare, risulta necessario presentare quali siano i parametri che controllano la modellizzazione utilizzata.

Il primo passo è la scelta del sistema di riferimento. In relazione a tale argomento andrebbe fatto un accorgimento. In fase di validazione sono state effettuate le analisi solo per la fase eliocentrica, in modo tale da poter considerare le SOI dei singoli pianeti puntiformi. Qualora si volesse analizzare traiettorie all'interno della SOI di altri pianeti, sarà necessario modificare il sistema di riferimento, con una conseguente modifica della costante gravitazionale  $\mu$ . Un esempio è svolgere le analisi all'interno della SOI terrestre e quindi passare da  $\mu_{\odot} = 132712440018 \frac{km^3}{s^2}$  a  $\mu_{\oplus} = 398600.4415 \frac{km^3}{s^2}$ .

Per completezza il sistema di riferimento utilizzato è un sistema eliocentrico avente come piano di riferimento il piano dell'eclittica all'epoca J2000 (corrispondente all'1° gennaio 2000 UT 12:00).

Il secondo passo è ricavare i vettori di stato associati ai due pianeti di partenza e di arrivo alle rispettive date di riferimento. Per fare ciò è stato sfruttato un software denominato SPICE. Questo strumento è stato realizzato dal Navigation and Ancillary Information Facility – NAIF della NASA, su richiesta del Jet Propulsion Laboratory – JPL, laboratorio NASA fondato nel 1936 con la funzione principale di progettare veicoli spaziali robotici per missioni interplanetarie, sebbene conduca anche missioni di orbita terrestre e di astronomia. Mediante la lettura di specifici file<sup>20</sup> è possibile ricavate i vettori di stato del corpo celeste richiesto in una specifica data. Le funzionalità del software SPICE sono molto più complesse e specifiche del semplice utilizzo svolto in questo lavoro di tesi e per tale motivo si consiglia al lettore un approfondimento [7]. Indirettamente risulta quindi scelto il Time of Flight – ToF della missione da analizzare che verrà suddivisa in un numero finito di archi di ampiezza  $\Delta t$ .

---

<sup>20</sup> chiamati “kernel” e al cui interno sono presenti tutti i dati relativi alle effemeridi.

Per una corretta modellizzazione è fondamentale conoscere alcuni parametri caratteristici del propulsore e dello S/C, qui di seguito elencati:

- Massa iniziale dello S/C (in  $10^3$  grammi);
- Massima spinta che può fornire il propulsore alla massima potenza (in Newton);
- Valore dell'impulso specifico caratteristico del propulsore che in prima approssimazione viene considerato costante (in secondi);
- Vincoli fisici sulla direzione della spinta rispetto alla massima ampiezza prima introdotta nei *Lower e Upper Bound Constraints* (in gradi).

### 4.1.3 Output DOPLLOT

Mediante l'applicazione di questo software sarà quindi possibile ricavare la traiettoria che permette di raggiungere la condizione di target con il minor consumo possibile di propellente. Verranno quindi forniti in output tutti i dati relativi ai parametri fondamentali per ogni sotto arco (massa, posizione, velocità, parametri orbitali, modulo e direzione della spinta ed errore sui vincoli). In termini grafici si otterranno quattro plot: la traiettoria ottimale in fase eliocentrica, il profilo della spinta T allo scorrere della traiettoria, l'analogo andamento per la distanza tra lo S/C e il Sole, ed infine l'andamento delle direzioni di elevazione  $\alpha$  e sul piano orbitale  $\beta$  per l'intera manovra. Si vedranno nel dettaglio questi grafici nel paragrafo conclusivo nel quale si presenterà la fase di validazione del software.

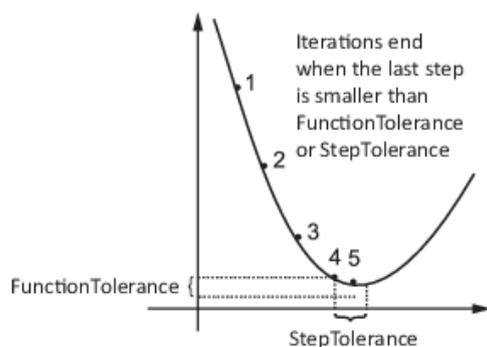
## 4.2 Parametri Metodo SQP

Come osservato nel capitolo precedente, gli algoritmi di ottimizzazione si basano su processi iterativi. Ad ogni iterazione vi è una costante applicazione del metodo di Newton e un continuo calcolo di derivate mediante l'approssimazione del metodo delle differenze finite. La convergenza del metodo SQP verrà garantita se le condizioni di ottimo e il rispetto dei vincoli, avvengono con un errore al di sotto di opportune tolleranze. Sarà quindi necessario scegliere con particolare attenzione queste tolleranze da utilizzare. In termini implementativi questo richiede un opportuno settaggio della funzione *fmincon*, in modo tale da indirizzare il codice verso un'adeguata accuratezza in funzione del problema da ottimizzare.

In questo paragrafo verranno presentate le principali tolleranze da modificare, e le rispettive analisi effettuate per una scelta opportuna.

- **StepTolerance**

Questo valore rappresenta il limite inferiore della variazione di  $x$  ad ogni iterazione, cioè la norma infinito del vettore  $x_i - x_{i+1}$ .



**Figura 4.2** – Iterazioni e StepTolerance<sup>21</sup>

Se il risolutore tenta di eseguire un passo inferiore a *StepTolerance*, l'algoritmo utilizzato prevede un blocco delle iterazioni. Spesso questa tolleranza è utilizzata come limite relativo, ovvero le iterazioni finiscono quando

$$|x_i - x_{i+1}| < \text{StepTolerance} \cdot (1 + |x_i|) \quad (4.9)$$

---

<sup>21</sup> Fonte Documentazione Matlab <https://it.mathworks.com/help/optim/ug/tolerances-and-stopping-criteria.html>

- **ConstraintTolerance**

Questo parametro indica un limite superiore dell'errore associato a qualsiasi vincolo (*Inequality* o *Equality*). Se ad esempio all'iterazione k-esima si avranno dei punti in cui

$$c(x_k) > \text{ConstraintTolerance} \quad (4.10)$$

oppure

$$ceq(x_k) > \text{ConstraintTolerance} \quad (4.11)$$

il risolutore segnala che i vincoli sono stati violati. Da notare che a differenza della tolleranza sullo step il metodo SQP continua le iterazioni pur se i vincoli sono stati violati. Come visto nel capitolo precedente, questo parametro a parità di qualsiasi dato causa una variazione del "percorso" che il codice seguirà durante le varie iterazioni per raggiungere la condizione di ottimo.

- **OptimalityTolerance**

Questo valore è una tolleranza per la misura del *First Order Optimality*, se quest'ultimo valore è inferiore a *OptimalityTolerance*, le iterazioni si bloccano. Il *First Order Optimality* è un indice che definisce una condizione di ottimo necessaria ma non sufficiente. Se la variabile  $x$  definisce un punto di minimo della funzione  $J(x)$ , questo valore dovrà essere nullo, ma di contro se il *First Order Optimality* è nullo non si ha necessariamente un minimo.

Spesso si utilizzano valori relativi. In questo caso le iterazioni si bloccano se:

$$\text{First Order Optimality} < \mu \text{OptimalityTolerance} \quad (4.12)$$

dove  $\mu$  è la norma infinito (nonché il massimo) del gradiente della funzione obiettivo calcolato al punto iniziale di tentativo  $x_0$ . Si avrà  $\mu = 1$  quando la norma del gradiente è minore di 1.

Il First Order Optimality (in relazione alla formulazione 3.64 presentata nel paragrafo precedente) è definito come il massimo delle due norme infinito seguenti:

$$\begin{aligned} \|\nabla_x L(x, \lambda)\| = & \|\nabla_x J(x) + A^T \lambda_{ineqlin} + Aeq^T \lambda_{eqlin} + \\ & + \sum \lambda_{ineqnonlin_i} \nabla c_i(x) + \sum \lambda_{eqnonlin_i} \nabla ceq_i(x)\| \end{aligned} \quad (4.13)$$

$$\begin{aligned} & \left\| \begin{array}{ll} |l_i - x_i| \lambda_{lower_i}, & |x_i - u_i| \lambda_{upper_i}, \\ (Ax - b)_i \lambda_{ineqlin_i}, & |c_i(x)| \lambda_{ineqnonlin_i} \end{array} \right\| \end{aligned} \quad (4.14)$$

### 4.2.1 Analisi adimensionale

Prima di addentrarsi nel particolare delle tolleranze appena presentate, è utile introdurre un concetto fondamentale riguardo la modellizzazione del problema. Il dover effettuare le analisi con valori numerici aventi ordini di grandezza notevolmente differenti tra loro<sup>22</sup>, potrebbe causare delle difficoltà nelle analisi ritardando la convergenza. Per questo motivo è stata introdotta la possibilità di adimensionalizzare il problema sfruttando le seguenti grandezze di riferimento:

- Distanza media Terra – Sole (Unità Astronomica – AU):  

$$r_{rif} = 1.4965978707 \cdot 10^8 \text{ km}$$
- Costante gravitazione solare:  $\mu_{rif} = \mu_{\odot} = 132712440018 \frac{\text{km}^3}{\text{s}^2}$
- Velocità circolare della Terra:  $v_{rif} = \sqrt{\frac{\mu_{\odot}}{r_{rif}}} = 29.7785 \frac{\text{km}}{\text{s}}$
- Tempo dato dalla relazione:  $t_{rif} = \frac{r_{rif}}{v_{rif}} = 5.025761411 \cdot 10^6 \text{ s}$
- Massa iniziale dello S/C:  $m_{rif} = m_0$
- Spinta data dalla relazione:  $T_{rif} = m_i \frac{r_{rif}}{t_{rif}^2} = 11.2578 \text{ N}$

Queste assunzioni permettono di lavorare con grandezze prossime all'unità. Per valutarne la differenza tra l'analisi adimensionale appena presentata e l'analisi dimensionale, sono stati confrontati i risultati ottenuti dall'ottimizzatore.

Per le analisi suddette sono state utilizzate le tolleranze di default cioè uguali a  $10^{-6}$  e il valore della variabile di stato iniziale di tentativo  $x_0 = 0$ . È stato schematizzato il risultato di tale confronto nella tabella 4.1, avendo indicato con “Errori sui Non Linear Equality Constraints” il valore delle 6 componenti dei  $ceq(x)$  presentati nell'equazione 4.5 .

---

<sup>22</sup> In ambito interplanetario si hanno ordini di grandezza di  $10^8$  per le distanze in km e 10 per le velocità in km/s.

**Tabella 4.1** – *Analisi dimensionale vs. analisi adimensionale*

Tipologia Analisi		<i>Dimensionale</i>	<i>Adimensionale</i>
<b>Numero Finale di Iterazioni</b>	<i>[-]</i>	312	346
<b>Valore Funzione Obiettivo</b>	<i>[km/s]</i>	5.81482	5.81475
<b>First Order Optimality</b>	<i>[-]</i>	18.17	0.693
<b>Tempo Complessivo Analisi</b>	<i>[s]</i>	977.085	562.735
<b>Errori sui Non Linear Equality Constraints – Vettori di stato condizione target</b>			
<b>Componenti vettore di stato – Posizione</b>			
<b>X</b>	<i>[km]</i>	1327.90296	0.519020
<b>Y</b>	<i>[km]</i>	677.64354	0.338088
<b>Z</b>	<i>[km]</i>	8.308499	0.012018
<b>Componenti vettore di stato – Velocità</b>			
<b>X</b>	<i>[km/s]</i>	0.000087	$\approx 10^{-8}$
<b>Y</b>	<i>[km/s]</i>	0.000286	$\approx 10^{-8}$
<b>Z</b>	<i>[km/s]</i>	$\approx 10^{-6}$	$\approx 10^{-10}$

Questo confronto permette di estrapolare utili considerazioni. In primis è possibile osservare che le soluzioni ricavate sono praticamente analoghe (come si osserva nel valore della funzione obiettivo ottimizzata e nel confronto tra i plot in output nelle due analisi). Per l'analisi adimensionale si riscontra un numero maggiore di iterazioni per raggiungere la convergenza, ma di contro è stata ottenuta una riduzione del 40% della durata delle analisi.

Ma la differenza fondamentale riguarda l'accuratezza del risultato. Un indice che dimostra questa affermazione è il *First Order Optimality*, il quale è ben due ordini di grandezza più piccolo. Il massimo errore nell'analisi adimensionale è di 1327.9 km sulla prima componente del vettore della posizione. Nell'analisi dimensionale la stessa componente avrà un errore nell'ordine dei 500 m, ma se rapportato in termini adimensionali il massimo errore che legge il risolutore sarà associato alla velocità (valore trascurabile di circa  $10^{-2}$  mm/s).

Se ne conclude che l'analisi adimensionale permette di ottenere una soluzione maggiormente accurata ma con un risparmio nella durata delle analisi e per tale motivo è stata scelta per l'implementazione del codice.

### 4.2.2 StepTolerance

La prima analisi effettuata riguarda la scelta dello *StepTolerance*. Per valutare l'influenza che una variazione di questo parametro ha sul susseguirsi delle iterazioni, viene variato il valore dello *StepTolerance* a partire dal valore di default che corrisponde a  $10^{-6}$ . Analogamente tutte le altre tolleranze vengono mantenute uguali a questo valore di default.

Prima di commentare i dati ricavati dalla suddetta analisi è necessario sottolineare che il processo iterativo potrà bloccarsi per cause differenti. Il parametro che ci permette di valutare tale causa è denominato *ExitFlag*. Si tratta di un valore numerico associato alla motivazione per la quale le iterazioni si bloccano.

Nella tabella 4.2 è possibile prendere visione di tale cause e del relativo valore numerico associato.

**Tabella 4.2** – *ExitFlag*

EXITFLAG	
2	Change in x was less than options.StepTolerance and maximum constraint violation was less than options.ConstraintTolerance.
1	First-order optimality measure was less than options.OptimalityTolerance, and maximum constraint violation was less than options.ConstraintTolerance.
0	Number of iterations exceeded options. MaxIterations or number of function evaluations exceeded options.MaxFunctionEvaluations.
-1	Stopped by an output function or plot function.
-2	No feasible point was found.

In generale questo parametro permette di riconoscere la causa della fine delle iterazioni. Il metodo SQP prevede il riconoscimento di un punto di ottimo solo se verranno rispettate le condizioni di ottimo e i vincoli. Quando gli errori su queste condizioni sono al di sotto delle tolleranze appena presentate, il valore ricavato sarà quello di ottimo (*ExitFlag* = 1). La scelta delle tolleranze è di fondamentale importanza per la convergenza. Tuttavia, pur con valori di tolleranza non adeguati, l'algoritmo prevede l'uscita dalle iterazioni anche per altre cause. Come detto nel paragrafo 4.2 potrebbe esservi il blocco se non vengono rispettati i limiti imposti dalla *StepTolerance* (*ExitFlag* = 2) o se il numero di iterazioni eccede il valore limite (*ExitFlag* = 0).

Un dato che va particolarmente attenzionato è il valore di  $ExitFlag = -1$ . In questa situazione le iterazioni si bloccano per un motivo diverso dai precedenti e che potrà essere modificato caso per caso. In termini implementativi, ciò prevede l'utilizzo di una particolare funzione, che verrà inserita internamente all'ottimizzatore, e forzerà il blocco delle iterazioni qualora vengano rispettate particolari condizioni. Questa funzione prende il nome di *output function*.

Nel caso in esame è stata introdotta una *output function* che prevede l'uscita dalle iterazioni se vengono rispettate le seguenti condizioni:

1. Il valore di *StepLenght*  $\alpha$  avrà valore minore di  $10^{-15}$  che si ritiene essere un valore oltre al quale non si ha più influenza sul risultato (ricordando che la precisione di macchina è di  $10^{-16}$ );
2. Il valore della funzione obiettivo non varia rispetto all'iterazione precedente di un valore di  $10^{-8}$  che in termini dimensionali corrisponde ad una variazione di 0.001 mm/s;
3. La condizione 1 e la condizione 2 dovranno presentarsi consecutivamente per 50 iterazioni.

Da sottolineare che la condizione 2 e la condizione 3 sono solo delle modalità per rafforzare la prima condizione. Ciò permette di riconoscere che per tali presupposti l'uscita dalle iterazioni avverrà se le analisi risultano bloccate nell'intorno di un valore. In questa situazione, tale valore della variabile di stato è ritenuto un minimo della funzione obiettivo, ma che non rispetta le tolleranze sui vincoli e/o sul First Order Optimality.

Questa possibilità ricade nella problematica di riuscire a settare in modo opportuno il codice, ed inoltre potrebbe essere dovuta all'eventualità che l'algoritmo SQP possa trovare un valore sub-ottimale a causa di una errata scelta della condizione iniziale di tentativo  $x_0$ .

L'analisi sulla scelta dello StepTolerance è stata riassunta nella tabella 4.3 .

**Tabella 4.3** – Analisi al variare dello StepTolerance

<b>Step Tolerance</b>	<b>Iter Tot</b>	<b>Exit Flag</b>	<b><math>J(x)</math></b>	<b>FirstOrder Optimality</b>	<b><math>t_{analysis}</math></b>
<i>[-]</i>	<i>[-]</i>	<i>[-]</i>	<i>[km/s]</i>	<i>[-]</i>	<i>[s]</i>
$10^{-6}$	288	2	5.815	0.01883	640.25
$10^{-7}$	288	2	5.815	0.01883	550.25
$10^{-8}$	288	2	5.815	0.01883	544.25
$10^{-9}$	288	2	5.815	0.01883	524.24
$10^{-10}$	288	2	5.815	0.01883	601.86
$10^{-12}$	290	2	5.815	0.01883	635.69
$10^{-13}$	290	2	5.815	0.01883	594.66
$10^{-15}$	290	2	5.815	0.01883	567.70
$10^{-20}$	3371	0	5.815	0.6929	9939.9
0	346	-1	5.815	0.6929	1275.2

<b>Step Tolerance</b>	<b>Error Position</b>			<b>Error Velocity</b>		
	<b>X</b>	<b>Y</b>	<b>Z</b>	<b>X</b>	<b>Y</b>	<b>Z</b>
<i>[-]</i>	<i>[km]</i>	<i>[km]</i>	<i>[km]</i>	<i>[km/s]</i>	<i>[km/s]</i>	<i>[km/s]</i>
$10^{-6}$	0.939	0.659	0.015	$10^{-8}$	$10^{-8}$	$10^{-10}$
$10^{-7}$	0.553	0.365	0.012	$10^{-8}$	$10^{-8}$	$10^{-10}$
$10^{-8}$	0.522	0.342	0.012	$10^{-8}$	$10^{-8}$	$10^{-10}$
$10^{-9}$	0.519	0.339	0.012	$10^{-8}$	$10^{-8}$	$10^{-10}$
$10^{-10}$	0.518	0.339	0.012	$10^{-8}$	$10^{-8}$	$10^{-10}$
$10^{-12}$	0.518	0.339	0.012	$10^{-8}$	$10^{-8}$	$10^{-10}$
$10^{-13}$	0.518	0.339	0.012	$10^{-8}$	$10^{-8}$	$10^{-10}$
$10^{-15}$	0.518	0.339	0.012	$10^{-8}$	$10^{-8}$	$10^{-10}$
$10^{-20}$	0.518	0.339	0.012	$10^{-8}$	$10^{-8}$	$10^{-10}$
0	0.519	0.338	0.012	$10^{-8}$	$10^{-8}$	$10^{-10}$

Da questa analisi si è potuto osservare che la scelta dello *StepTolerance* non influenza il susseguirsi delle analisi. Si nota infatti che alla 288° iterazione il codice ha già trovato una soluzione ottimale ma che non rispetta i parametri del *OptimalityTolerance*. Quando infatti si utilizza un valore di *StepTolerance* molto piccolo di  $10^{-20}$  il codice continua le analisi per oltre 3000 iterazioni (senza riscontrare nessuna sostanziale modifica già dalla 290°). In questo caso l'uscita dalle analisi si verifica poiché viene superato il numero limite di iterazioni. L'ultima prova viene fatta annullando la tolleranza sullo step (*StepTolerance* = 0) ed inserendo la *output function* precedentemente presentata. Si nota infatti che tra l'ultima analisi è la penultima vi è una differenza sulla durata delle iterazioni di quasi il 900%, pur mantenendo un'accuratezza molto elevata. Noto che la condizione di ottimo è solita raggiungersi quando viene utilizzato uno step molto piccolo, e alla luce di questi risultati appena presentati, nel proseguo delle analisi verrà annullata la tolleranza sullo step ed introdotta la *output function* così da evitare che le iterazioni si blocchino troppo presto.

### 4.2.3 ConstraintTolerance

Come enunciato nel capitolo precedente, questo parametro è di fondamentale importanza poiché influenza l'applicazione dell'algorithm di ottimizzazione; infatti una sua modifica può causare una variazione dei dati in uscita ad ogni iterazione. È stato analizzato il risultato dell'ottimizzazione al variare del *ConstraintTolerance* (tra  $10^{-6}$  e  $10^{-2}$ ). Da sottolineare che i risultati in uscita dall'ottimizzatore sono tutti associati ad un *ExitFlag* = -1 (presenza della *output function*).

La presenza di una tolleranza sui vincoli equivale ad imporre un limite sull'errore accettati dall'ottimizzatore in termini di posizione e velocità rispetto alla condizione target reale ricavata dall'effemeridi. Si osserva ad esempio che assegnare un valore di *ConstraintTolerance* di  $10^{-4}$  equivale ad accettare soluzioni aventi errori sulla posizione al massimo di  $10^4$  km e sulla velocità di 1 m/s.

I risultati sopra citati sono stati schematizzati nella tabella 4.4.

**Tabella 4.4** – *ConstraintTolerance ed errori associati*

<b>Constraint Tolerance</b>	<b>Max Error Velocity</b>	<b>Max Error Position</b>
<i>[-]</i>	<i>[m/s]</i>	<i>[km]</i>
$10^{-6}$	$10^{-2}$	$10^2$
$10^{-5}$	$10^{-1}$	$10^3$
$10^{-4}$	1	$10^4$
$10^{-3}$	10	$10^5$
$10^{-2}$	$10^2$	$10^6$

**Tabella 4.5** – *Analisi al variare del ConstraintTolerance*

Constraint Tolerance	Iter Tot	$J(x)$	Error Position			Error Velocity		
			X	Y	Z	X	Y	Z
$[-]$	$[-]$	$[km/s]$	$[km]$	$[km]$	$[km]$	$[km/s]$	$[km/s]$	$[km/s]$
$10^{-6}$	346	5.8148	0.519	0.338	0.012	$10^{-8}$	$10^{-8}$	$10^{-10}$
$10^{-5}$	378	5.8145	3248.6	1355.2	25.623	0.0002	0.0006	$10^{-6}$
$10^{-4}$	344	5.8149	11181.1	6396.9	98.373	0.0008	0.0025	$10^{-5}$
$10^{-3}$	303	5.8227	77485.8	59389.6	2411.6	0.0045	0.0203	$10^{-5}$
$10^{-2}$	192	7.9826	60551350	5103923	16595.5	1.6285	11.657	0.1528

Da una valutazione accurata di questi dati sono state attenzionate due possibili scelte. La scelta di un valore del *ConstraintTolerance* compreso tra  $10^{-4}$  e  $10^{-5}$  prevede errori massimi sulla velocità e sulla posizione ampiamente accettabili in ambito di missioni interplanetarie. Nello specifico si tratta di errori sulla velocità compresi tra 0.1 e 1 m/s e sulle posizioni tra 1000 e 10000 km. Va però ricordato che l'ottimizzatore permette di ricavare il punto di minimo solo se vengono rispettati sia il *ConstraintTolerance* che lo *OptimalityTolerance*. Per tale motivo l'ultima analisi prevede il confronto per questi due valori di *ConstraintTolerance* al variare dell'*OptimalityTolerance*.

#### 4.2.4 OptimalityTolerance

Il parametro *OptimalityTolerance* è una tolleranza sul *First Order Optimality*. L'annullamento del *First Order Optimality*, la cui formulazione matematica è stata presentata in precedenza, indica la condizione necessaria ma non sufficiente per definire un punto di minimo. Nella seguente analisi è stato valutato l'andamento del risultato al variare dell'*OptimalityTolerance* per i due valori di *ConstraintTolerance* prima definiti.

Nel proseguo del paragrafo sono stati tabellati tutti i dati caratteristici di ogni analisi.

**Tabella 4.6** – Analisi al variare dello OptimalityTolerance (ConstraintTolerance  $10^{-5}$ )

Constraint Tolerance	Max Error Velocity	Max Error Position
$[-]$	$[m/s]$	$[km]$
$10^{-5}$	$10^{-1}$	$10^3$

Optimality Tolerance	Iter Tot	Exit Flag	$J(x)$	FirstOrder Optimality	$t_{analysis}$
$[-]$	$[-]$	$[-]$	$[km/s]$	$[-]$	$[s]$
$10^{-6}$	378	-1	5.8145	0.215	633.3
$10^{-5}$	378	-1	5.8145	0.215	629.1
$10^{-4}$	378	-1	5.8145	0.215	620.4
$10^{-3}$	378	-1	5.8145	0.215	613.0
$10^{-2}$	378	-1	5.8145	0.215	613.0
$10^{-1}$	264	1	5.8157	0.0589	418.8

Optimality Tolerance	Error Position			Error Velocity		
	X	Y	Z	X	Y	Z
$[-]$	$[km]$	$[km]$	$[km]$	$[km/s]$	$[km/s]$	$[km/s]$
$10^{-6}$	3248.6	1355.19	25.63	0.00023	0.0006	$10^{-6}$
$10^{-5}$	3248.6	1355.19	25.63	0.00023	0.0006	$10^{-6}$
$10^{-4}$	3248.6	1355.19	25.63	0.00023	0.0006	$10^{-6}$
$10^{-3}$	3248.6	1355.19	25.63	0.00023	0.0006	$10^{-6}$
$10^{-2}$	3248.6	1355.19	25.63	0.00023	0.0006	$10^{-6}$
$10^{-1}$	584.481	299.65	12.805	$10^{-5}$	0.0001	$10^{-6}$

**Tabella 4.7** – Analisi al variare dello OptimalityTolerance (ConstraintTolerance  $10^{-4}$ )

Constraint Tolerance	Max Error Velocity	Max Error Position
$[-]$	$[m/s]$	$[km]$
$10^{-4}$	1	$10^4$

Optimality Tolerance	Iter Tot	Exit Flag	$J(x)$	FirstOrder Optimality	$t_{analysis}$
$[-]$	$[-]$	$[-]$	$[km/s]$	$[-]$	$[s]$
$10^{-6}$	344	-1	5.8149	4517	561.3
$10^{-5}$	344	-1	5.8149	4517	591.9
$10^{-4}$	344	-1	5.8149	4517	555.9
$10^{-3}$	344	-1	5.8149	4517	536.9
$10^{-2}$	281	1	5.8156	0.00703	441.4
$10^{-1}$	273	1	5.8611	0.08802	448.1

Optimality Tolerance	Error Position			Error Velocity		
	X	Y	Z	X	Y	Z
$[-]$	$[km]$	$[km]$	$[km]$	$[km/s]$	$[km/s]$	$[km/s]$
$10^{-6}$	11180.8	6396.9	98.373	0.0008	0.0025	$10^{-5}$
$10^{-5}$	11180.8	6396.9	98.373	0.0008	0.0025	$10^{-5}$
$10^{-4}$	11180.8	6396.9	98.373	0.0008	0.0025	$10^{-5}$
$10^{-3}$	11180.8	6396.9	98.373	0.0008	0.0025	$10^{-5}$
$10^{-2}$	599.57	332.03	8.504	$10^{-5}$	0.0001	$10^{-6}$
$10^{-1}$	8469.8	4586.8	172.74	0.0003	0.0019	$10^{-5}$

La scelta del *OptimalityTolerance* ha richiesto un'analisi maggiormente accurata. Il primo dato importante è la durata delle analisi che risulta parzialmente minore nel caso in cui si utilizza un *ConstraintTolerance* di  $10^{-4}$ . Ridurre la tolleranza sui vincoli permette di raggiungere la convergenza in un minor numero di iterazioni. Tuttavia, la strada che viene seguito per raggiungere questo obiettivo permette comunque di avere un risultato ugualmente accurato seppur con una tolleranza sui vincoli meno restrigente.

Fatta tale premessa, si osserva nella seconda tabella che gli unici due valori di *OptimalityTolerance* che permettono la convergenza al valore ottimale (*ExitFlag* = 1) sono quelli utilizzati nelle ultime due analisi. È stata però riscontrata, nel caso di *OptimalityTolerance* = 0.1 una convergenza verso un valore della funzione obiettivo ben maggiore dei casi precedenti (scarto maggiore di 50 m/s). Riguardo l'accuratezza, se confrontata con gli ordini di grandezza caratteristici delle missioni interplanetarie risulta accettabile ma se confrontata con le altre analisi si osserva un peggioramento. Proprio per questo motivo è stato scelto come settaggio ottimale della funzione *fmincon* un *OptimalityTolerance* non al di sotto dei  $10^{-2}$  e un *ConstraintTolerance* =  $10^{-4}$ .

È doveroso precisare che la scelta ottimale di questi parametri può variare in funzione del caso analizzato. La scelta più accurata dipenderà dall'esperienza dell'operatore. Questa modifica verrà fatta ad esempio nella parte finale del capitolo in cui si implementerà il problema della bassa spinta con una parziale variazione riguardante l'andamento della spinta in funzione della distanza dal Sole.

#### 4.2.5 Numero archi

Scelti i valori ottimali delle tolleranze, è stata fatta un'ulteriore analisi al variare dell'ampiezza del singolo arco. Variare questo parametro corrisponde variare il numero complessivo di archi che compone la traiettoria. Sono stati quindi confrontati due parametri fondamentali nonché il valore della funzione ottimizzata (cioè il  $\Delta V$  totale dell'intera traiettoria) e la durata delle analisi. Ricordando che la durata della missione analizzata è di 650 giorni, l'andamento ottenuto per questi due parametri si può osservare in figura 4.3 .

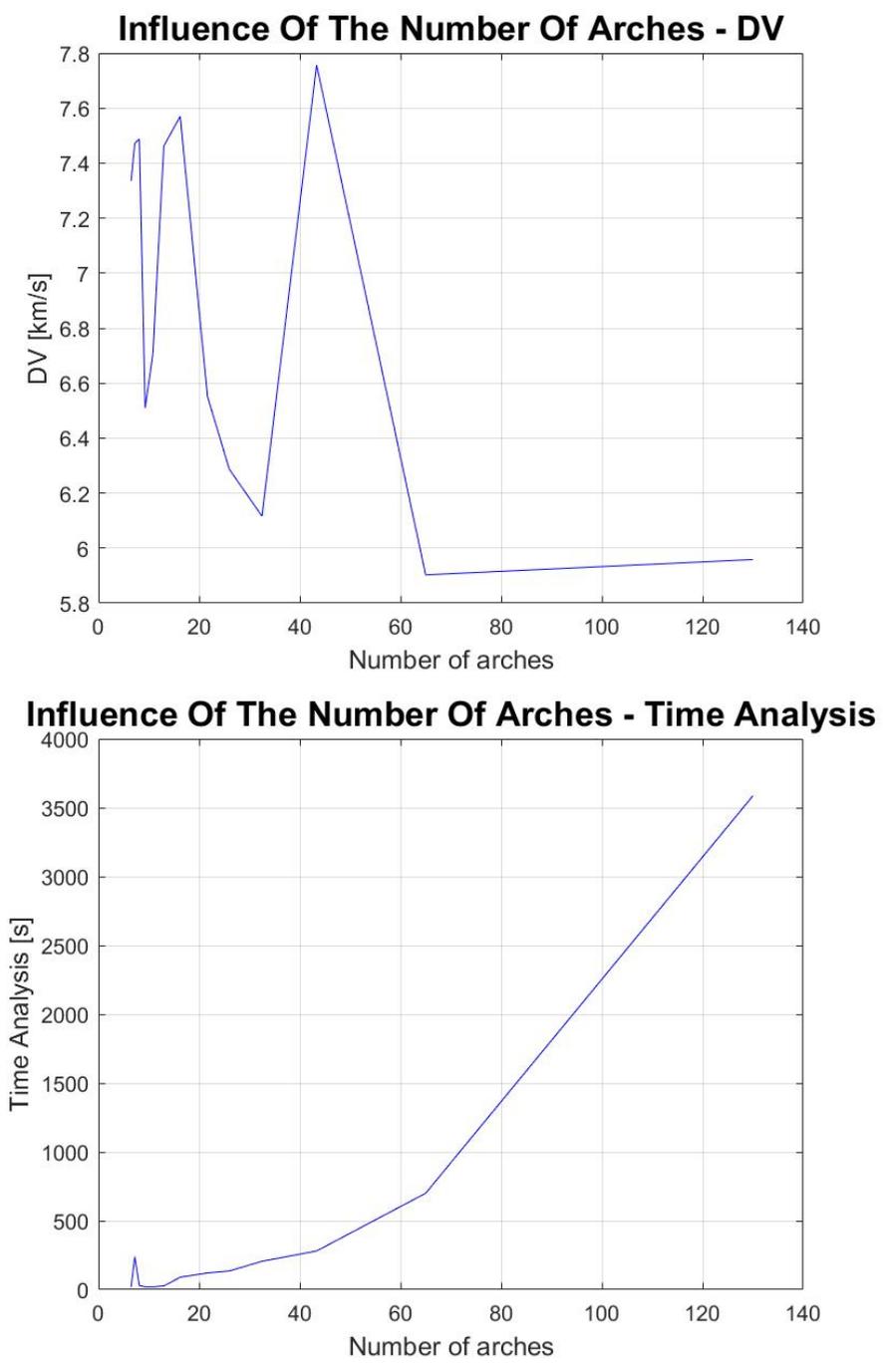


Figura 4.3 – Variazione del numero di archi

Si nota che al diminuire dell'ampiezza dell'arco le analisi tendono a convergere verso un valore della funzione obiettivo che sarà quello ottimale. Ciò però avviene a discapito della durata delle analisi che aumenta esponenzialmente all'aumentare del numero di archi e quindi delle variabili in gioco. È stato scelto come valore di compromesso un arco di 10 giorni che permette una buona accuratezza del risultato pur con una durata delle analisi accettabile.

### 4.3 SEP – Solar Electric Propulsion

L'implementazione utilizzata nelle analisi precedenti prevedeva un valore massimo della spinta che viene garantito per tutta la durata della traiettoria. In termini matematici, questo concetto equivale ad aver assegnato un *Upper Bound Constraint* per il modulo della spinta.

$$0 \leq T_i \leq T_{max} \quad (4.15)$$

con  $1 < i < N$  ed  $N$  numero totale di archi con cui viene suddivisa la traiettoria.

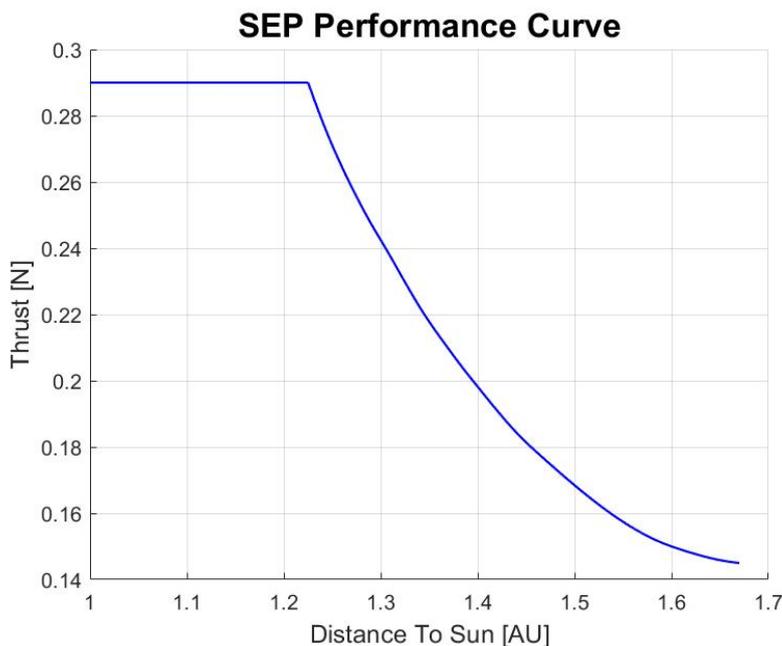
Per garantire la spinta massima per tutta la durata della missione, sarebbe necessario un generatore che possa fornire la massima potenza costantemente. In prima approssimazione questo concetto può essere garantito solo dall'utilizzo di un NEP – Nuclear Electric Propulsion. Il requisito fondamentale di un sistema di propulsione elettrica nucleare NEP è di convertire l'energia termica del reattore nucleare in energia elettrica da poter fornire al propulsore. Queste tipologie di generatori non sono ampiamente utilizzate tra le missioni spaziali, principalmente per la difficoltà di testare generatori nucleari in ambiente terrestre a causa di restrizioni politiche ed ambientali.

La tipologia di propulsione maggiormente utilizzata tra le missioni spaziali interplanetarie è la Solar Electric Propulsion – SEP. Questi propulsori prevedono l'utilizzo di pannelli solari come generatori di potenza elettrica. Questo strumento è ampiamente influenzato dalla distanza dal Sole, poiché il funzionamento dei pannelli solari dipende dalla intensità delle radiazioni solari, le quali sono inversamente proporzionali alla distanza dal Sole. Implementare questo concetto nell'algoritmo di ottimizzazione appena presentato, richiede sostanziali modifiche nella formulazione matematica correlata.

In questa ottica, sono stati sfruttati due possibili approcci nell'implementare l'influenza che questa distanza ha sul modulo della spinta in presenza di un Solar Electric Propulsion.

L'implementazione del SEP ha necessitato l'introduzione di una specifica funzione. Ogni propulsore elettrico genera una spinta in funzione alla potenza fornita. Tale potenza nel caso dei SEP è funzione della distanza dal Sole. Per questa ragione la curva caratteristica di ogni SEP è analoga a quella presentata nel grafico seguente.

In figura 4.4 si osserva che l'andamento della spinta in funzione della distanza dal Sole è costante fino a circa 1.25 AU per poi assumere un andamento quadratico decrescente con l'aumentare della distanza dal Sole.



**Figura 4.4** – Curva caratteristica di un generico SEP<sup>23</sup>

Saranno quindi necessarie le seguenti modifiche:

1. Introdurre una funzione, denominata  $lim\_thrust\_AU$ , che permette di ricavare la massima spinta che può fornire il propulsore in relazione alla distanza dal Sole. La relazione matematica associata al grafico di figura 4.4 è la seguente:

$$\begin{aligned}
 lim\_thrust\_AU &= \\
 &= \begin{cases} 0.290 & \text{if } r_i \leq 1.25 \text{ AU} \\ 0.6864 r_i^2 - 2.3011 r_i + 2.0745 & \text{if } r_i > 1.25 \text{ AU} \end{cases} \quad [N] \quad (4.16)
 \end{aligned}$$

<sup>23</sup> Fonte vedere riferimento bibliografico [\[25\]](#)

2. Inserire dei *Non Linear Inequality Constraint* che permettano internamente al risolutore di vincolare la spinta ad ogni arco in relazione alla posizione iniziale dell'arco i-esimo  $r_i$ .

In conclusione, questa nuova implementazione prevedrebbe la seguente formulazione dei vincoli:

- *Non Linear Equality Constraints:*

$$ceq(x) = \begin{bmatrix} r_N(1) - r_f(1) \\ r_N(2) - r_f(2) \\ r_N(3) - r_f(3) \\ v_N(1) - v_f(1) \\ v_N(2) - v_f(2) \\ v_N(3) - v_f(3) \end{bmatrix} = 0 \quad (4.17)$$

- *Non Linear Inequality Constraints:*

$$c(x) = [x_{3i-2} - T_{max_i}] \leq 0 \quad (4.18)$$

- *Lower Bound Constraints e Upper Bound Constraints:*

$$0 \leq T_i \leq T_{max} \quad (4.19)$$

$$-90^\circ \leq \beta_i \leq +90^\circ \quad (4.20)$$

$$-180^\circ \leq \alpha_i \leq +180^\circ \quad (4.21)$$

con  $T_i = x_{3i-2}$      $\beta_i = x_{3i-1}$      $\alpha_i = x_{3i}$

avendo indicato con  $T_{max}$  la massima spinta che può fornire il propulsore alla massima potenza e con  $T_{max_i}$  la massima spinta che il propulsore può fornire nell'arco i-esimo in relazione alla distanza dal Sole; si avrà sempre  $T_{max} \geq T_{max_i}$ .

Come è possibile osservare nelle equazioni 4.18 e 4.19, questa scelta non è applicabile. Si nota che la variabile  $x_{3i-2} = T_i$  risulterebbe vincolata due volte. La Lagrangiana avrà all'interno il contributo di ambe due i vincoli pur se la variabile vincolata è la stessa. Questa condizione falserebbe le analisi di ottimizzazione e per tale motivo è stata utilizzata un'implementazione alternativa.

Per ovviare alla complicazione appena presentate, è stata modificata la modellizzazione delle traiettorie a bassa spinta. La modifica sostanziale riguarda la sostituzione delle componenti 3i-2 del vettore della variabile di stato  $x$ .

Queste componenti riguardavano il valore del modulo della spinta, la quale verrà sostituita con un valore percentuale dal 0 al 100%. Questa percentuale equivale alla porzione di spinta utile ad ottimizzare la manovra ed ottenere la traiettoria finale con il minimo consumo di propellente. Pertanto, la modifica della formulazione matematica verrà fatta come segue:

- *Lower e Upper Bound Constraints:*

$$0 \leq x_{3i-2} \leq 1 \quad (4.22)$$

- Spinta massima i-esimo arco:

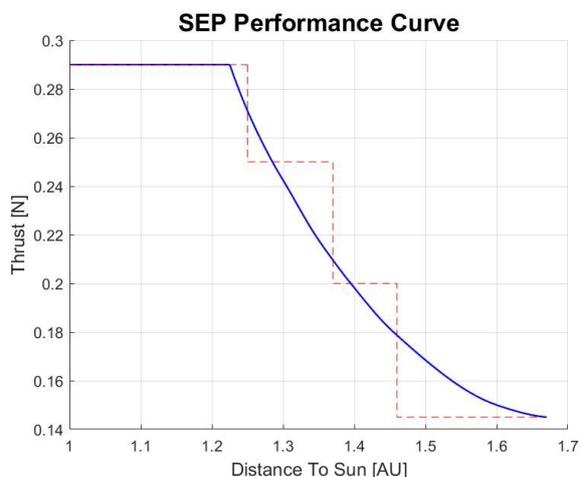
$$T_i = x_{3i-2} \cdot \lim\_thrust\_AU(r_i) \quad (4.23)$$

- Funzione obiettivo:

$$J(x) = \sum_{i=1}^N \frac{x_{3i-2} \cdot \lim\_thrust\_AU(r_i)}{m_i} \Delta t_i \quad (4.24)$$

Attuando questa seconda metodologia implementativa è stato possibile eliminare i *Non Linear Inequality Constraints*, i quali sono una tipologia di vincoli che rallenta notevolmente le analisi.

Un'ulteriore implementazione che non presenta i vantaggi di quella appena presentata prevede l'utilizzo di un andamento lineare a gradini della curva caratteristica del propulsore.



**Figura 4.5** – SEP: confronto tra le due implementazioni

In relazione al grafico, questo concetto espresso in termini matematici è il seguente:

$$T_{max_i} = 0.290 N \quad se \quad r_i < 1.25 AU \quad (4.25)$$

$$T_{max_i} = 0.250 N \quad se \quad 1.25 AU \leq r_i < 1.37 AU \quad (4.26)$$

$$T_{max_i} = 0.200 N \quad se \quad 1.37 AU \leq r_i < 1.46 AU \quad (4.27)$$

$$T_{max_i} = 0.145 N \quad se \quad r_i \geq 1.46 AU \quad (4.28)$$

L'utilizzo di questa semplificazione ha però causato delle singolarità nelle analisi e per questo motivo è stata abbandonata. Aver approssimato la curva caratteristica del propulsore con un andamento lineare a gradini causa molte complicazioni all'interno dell'algoritmo di ottimizzazione. Il metodo SQP effettua il calcolo dei gradienti mediante l'approssimazione alle differenze finite. Il gradiente di una determinata grandezza viene quindi approssimato valutando l'incremento che tale grandezza avrà rispetto l'iterazione precedente. Sfruttando questo andamento lineare a gradini è possibile ricadere in una situazione che falsa notevolmente le analisi.

Per motivare questa affermazione è stato considerato il caso in cui alla k-esima iterazione il codice prevede come soluzione di tentativo una traiettoria che avrà nell'i-esimo arco una spinta massima di 0.250 N. Con l'applicazione del metodo SQP verrà calcolata una nuova soluzione tentativo per la k+1 iterazione. In linea di massima si parla sempre di variazioni molto piccole del vettore della spinta per ogni arco  $(T_i, \beta_i, \alpha_i)$ . Questa variazione potrebbe causare un lieve cambiamento della posizione dell'arco i-esimo causando un salto nella curva a gradino. Nel caso in cui si trovasse nell'intorno di 1.37 AU di distanza dal Sole, basterebbe una variazione positiva infinitesima per causare una variazione del valore limite della spinta da 0.250 N a 0.200 N, compromettendo notevolmente le analisi. In questa circostanza una variazione dell'ordine di  $10^{-6}$  delle variabili di stato, farà variare l'errore sul vincolo da un valore prossimo allo zero (circa  $10^{-13}$ ), ad un valore dell'ordine di  $10^{-2}$ . Questo crea delle discontinuità nell'andamento del First Order Optimality che compromettono la convergenza dell'ottimizzatore.

Nel paragrafo seguente verranno quindi presentate tutte le analisi di validazione effettuate confrontando i risultati ricavati dal codice DOPLLOT con quelli fornite nel MAG ESA della missione Mars Sample Return (la cui data di pubblicazione risale al 28/03/2017).

## 4.4 Validazione

In questo paragrafo verranno presentate le analisi di validazione del software DOPLLOT. Sono state utilizzate le informazioni ricavate dal MAG ESA della missione Mars Sample Return per riprodurre alcune delle traiettorie proposte nel documento.

Il settaggio utilizzato, scelto mediante una serie di accurate analisi appena presentate, è il seguente:

*Tabella 4.8 – Settaggio Validation Case*

<b><i>StepTolerance</i></b>	0	<i>[-]</i>
<b><i>ConstraintTolerance</i></b>	$10^{-4}$	<i>[-]</i>
<b><i>OptimalityTolerance</i></b>	$10^{-3} \div 10^{-4}$	<i>[-]</i>
<b><i>Ampiezza archi</i></b>	10	<i>[giorno]</i>

Gli input del software DOPLLOT sono i seguenti:

*Tabella 4.9 – Input Validation Case*

<b>Reference Frame</b>	Eliocentrico – Ecliptic J2000	<i>[-]</i>
<b>Departure Planet</b>	Mars	<i>[-]</i>
<b>Arrival Planet</b>	Earth	<i>[-]</i>
<b>S/C Mass</b>	1900	<i>[kg]</i>
<b>Max Thrust</b>	0.290	<i>[N]</i>
<b>Mechanical Constraints Of Directions</b>	0°	<i>[deg]</i>
<b>Specific Impulse</b>	4010	<i>[s]</i>

Il Validation Case è il seguente:

**Tabella 4.10** – Risultati Validation Case: software DITAN - ESA

<b>Validation Case</b>	Mars - Earth Transfer 2034 (Case ME2034-EP-SC2)	[-]
<b>Software</b>	DITAN	[-]
<b>Departure Date</b>	2034 – 10 – 02 UTC 00:00	[-]
<b>Hyperbolic Excess Velocity <math>v_{\infty DEP}</math></b>	0	[km/s]
<b>Arrival Date</b>	2036 – 07 – 13 UTC 00:00	[-]
<b>Hyperbolic Excess Velocity <math>v_{\infty ARR}</math></b>	0	[kms[/s]
<b>Transfer Time</b>	650	[day]
<b>Mass Propellant</b>	264.9	[kg]
<b>Total DV</b>	5.906	[km/s]

È stata quindi considerata la trasferta di rientro Marte Terra in data 2 ottobre 2034 e arrivo il 13 luglio 2036 (ToF di 650 giorni). Il codice prevede l'analisi della fase eliocentrica e considera le sfere di influenza dei due pianeti puntiformi. Pertanto, con tale implementazione, è prevista l'analisi solo nel caso in cui la velocità di eccesso iperbolico  $v_{\infty}$  sia nulla.

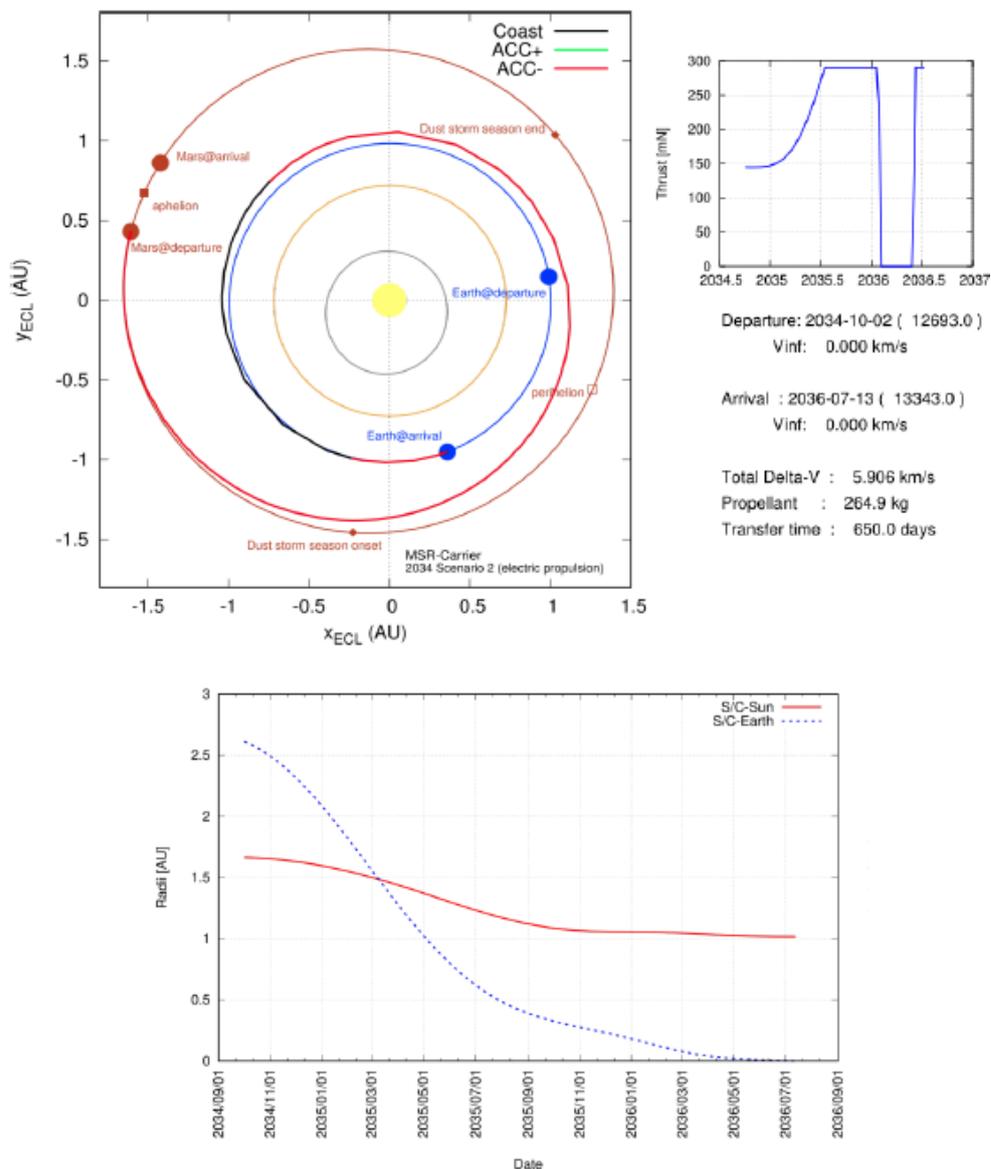


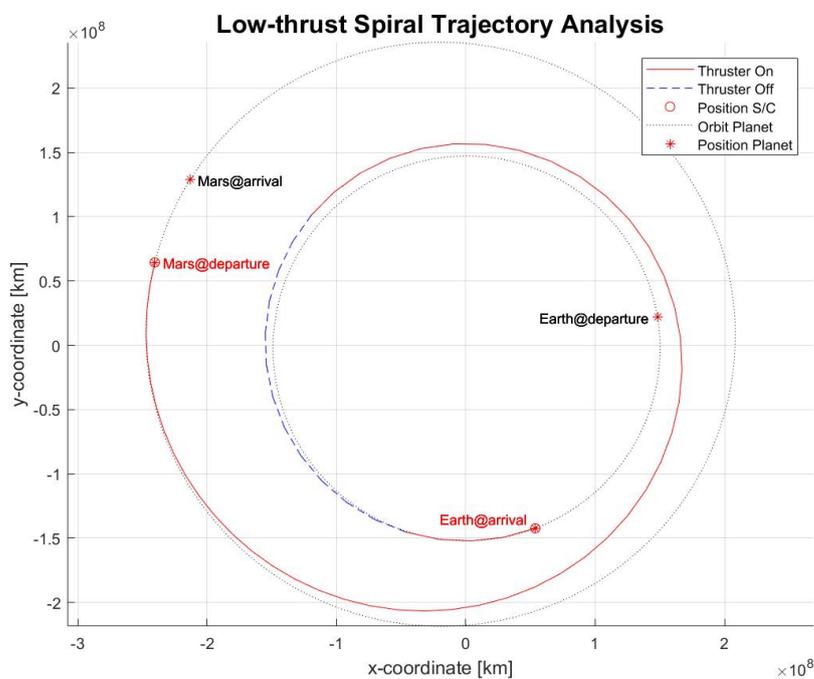
Figura 4.6 – Output Validation Case: software DITAN - ESA<sup>24</sup>

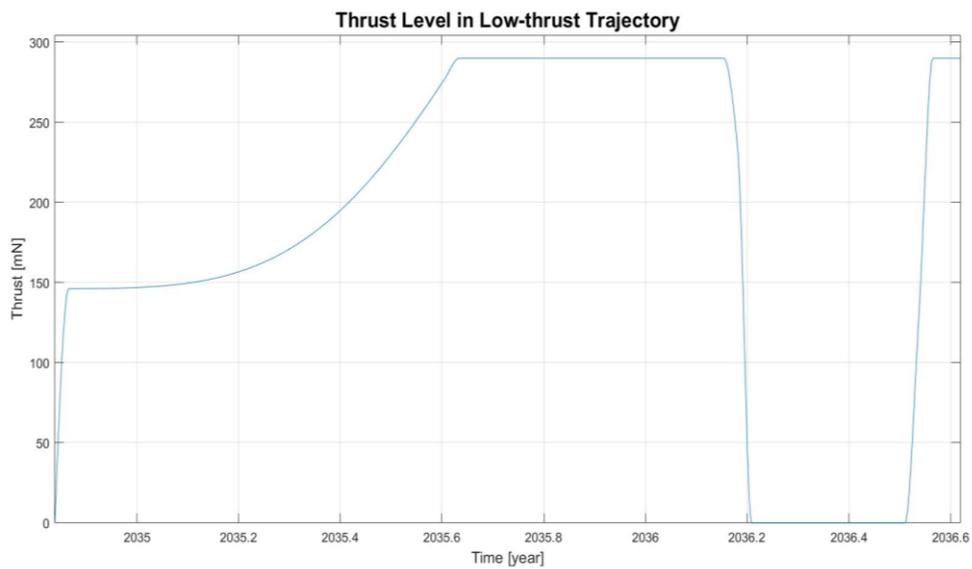
Mediante il software DOPLLOT sono stati ottenuti i risultati presentati nella tabella 4.11 .

<sup>24</sup> Fonte vedere riferimento bibliografico [25]

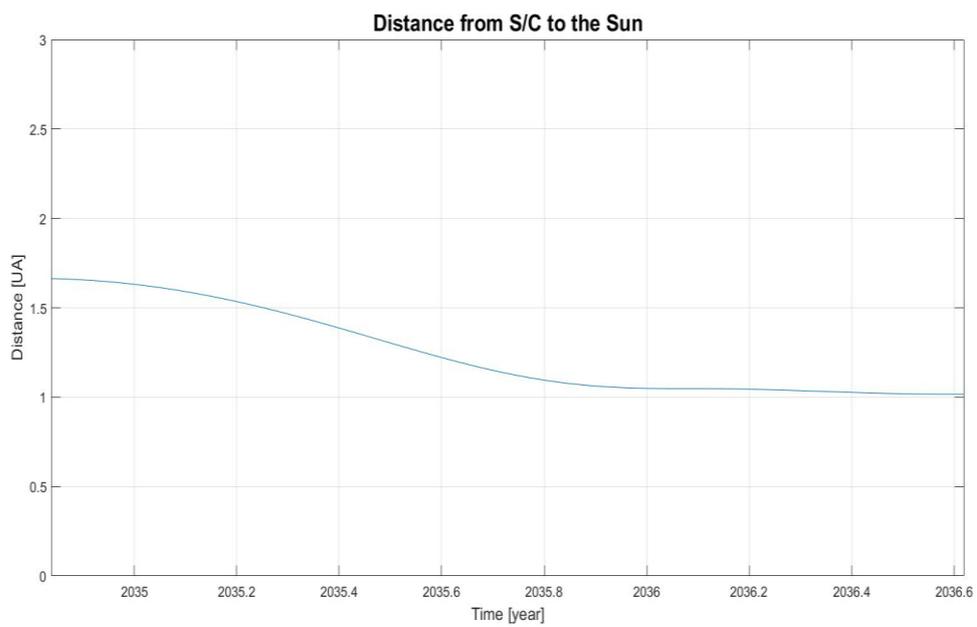
**Tabella 4.11** – Risultati Validation Case: software DOPLLOT

Software	DOPLOTT – SEP		[-]
Departure Date	2034 – 10 – 02 UTC 00:00		[-]
Hyperbolic Excess Velocity $v_{\infty DEP}$	0		[km/s]
Arrival Date	2036 – 07 – 13 UTC 00:00		[-]
Hyperbolic Excess Velocity $v_{\infty ARR}$	0		[km/s]
Transfer Time	650		[day]
OptimalityTolerance	$10^{-3}$	$10^{-4}$	[-]
Mass Propellant	265.198	265.202	[kg]
Total DV	5.9026	5.9027	[km/s]
Time Analysis	656.9	34291.5	[s]

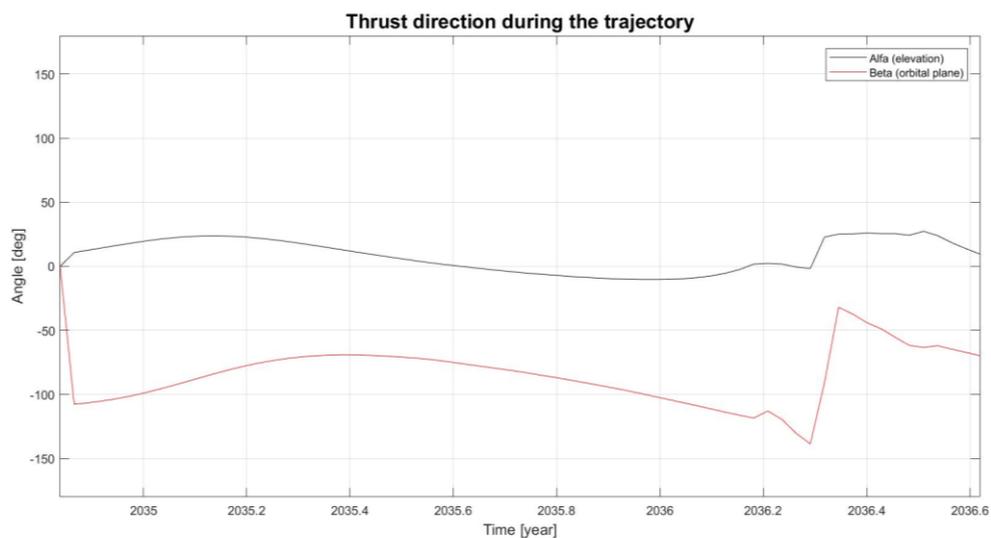
**Figura 4.7** – Output Validation Case: traiettoria



*Figura 4.9 – Output Validation Case: profilo della spinta*



*Figura 4.8 – Output Validation Case: andamento distanza dal Sole*



**Figura 4.10** – Output Validation Case: andamento delle direzioni della spinta

Ricordando la formulazione 4.19 dei *Non Linear Equality Constraints*

$$ceq(x) = \begin{bmatrix} r_N(1) - r_f(1) \\ r_N(2) - r_f(2) \\ r_N(3) - r_f(3) \\ v_N(1) - v_f(1) \\ v_N(2) - v_f(2) \\ v_N(3) - v_f(3) \end{bmatrix} = 0 \quad (4.19)$$

è stata formulata la seguente tabella in cui vi sono gli errori di posizioni e velocità che lo S/C avrà al tempo finale rispetto ai vettori di stato ricavati dall'effemeridi.

**Tabella 4.12** – *Validation Case: errori sui vettori di stato*

<b>Errori vettore di stato</b>			
<b>OptimalityTolerance</b>	$10^{-3}$	$10^{-4}$	<b>[-]</b>
$\mathbf{r}_N(1) - \mathbf{r}_f(1)$	11097.062	1850.203	[km]
$\mathbf{r}_N(2) - \mathbf{r}_f(2)$	5748.749	889.437	[km]
$\mathbf{r}_N(3) - \mathbf{r}_f(3)$	73.845	18.829	[km]
$\mathbf{v}_N(1) - \mathbf{v}_f(1)$	0.00075	0.00013	[km/s]
$\mathbf{v}_N(2) - \mathbf{v}_f(2)$	0.0024	0.00036	[km/s]
$\mathbf{v}_N(3) - \mathbf{v}_f(3)$	$10^{-5}$	$10^{-6}$	[km/s]

Da un'attenta analisi dei dati si osserva che la scelta dell'*OptimalityTolerance* non influisce sulla validazione del software, dal quale si ottiene per ambe due i valori un risultato analogo a quello visualizzabile nel MAG ESA.

Le uniche differenze si notano in termini di durata delle analisi che risulta quasi due ordini di grandezza maggiore nel caso di un *OptimalityTolerance* di  $10^{-4}$ . Questo dato però non è giustificato dall'accuratezza del risultato in relazione agli errori sui vincoli. Come si può osservare nella tabella 4.2, gli errori si mantengono accettabili in prima approssimazione per tutti e due i valori della tolleranza sul *First Order Optimality*.

Nella tabella 4.13 è stata riassunta l'analisi di validazione del software DOPLOTT.

**Tabella 4.13** – *Validation Case: confronto*

<b>Departure Date</b>	2034 – 10 – 02 UTC 00:00			<b>[-]</b>
<b>Hyperbolic Excess Velocity</b> $v_{\infty_{DEP}}$	0			[km/s]
<b>Arrival Date</b>	2036 – 07 – 13 UTC 00:00			<b>[-]</b>
<b>Hyperbolic Excess Velocity</b> $v_{\infty_{ARR}}$	0			[km/s]
<b>Transfer Time</b>	650			[day]
<b>Software</b>	<b>DITAN</b>	<b>DOPLOTT</b>	<b>Difference</b>	<b>[-]</b>
<b>Mass Propellant</b>	264.9	265.2	0.3	[kg]
<b>Total DV</b>	5.906	5.903	0.003	[km/s]

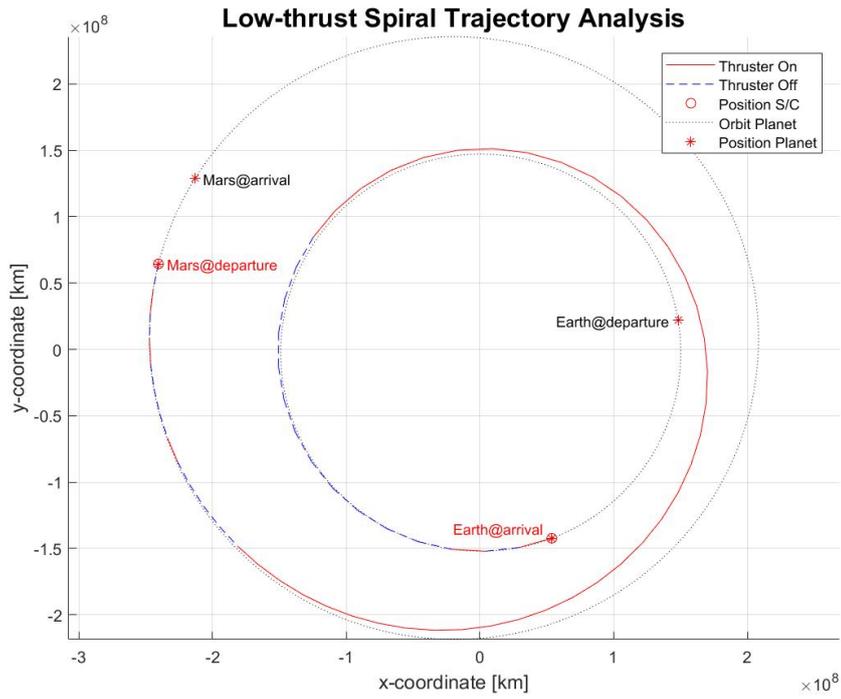
Per completezza è stata effettuata la stessa analisi in ipotesi di utilizzo di un NEP cioè rimuovendo la dipendenza della spinta dalla distanza dal Sole. I risultati non sono stati validati in quanto non sono presenti alcune analisi a riguardo.

I risultati ottenuti rispecchiano però la realtà del problema, cioè quello che comprensibilmente ci si aspetterebbe. Garantire una spinta massima costante permette l'ottimizzazione di una traiettoria più economica come si osserva nella tabella 4.14 . In questa tabella sono presenti la quantità di propellente consumato e il  $\Delta V$  totale necessario per compiere l'intera missione. Successivamente, verranno presentati i 4 grafici ricavati in output dal software DOPLOTT. Da notare che è stato utilizzato un settaggio analogo al caso precedente.

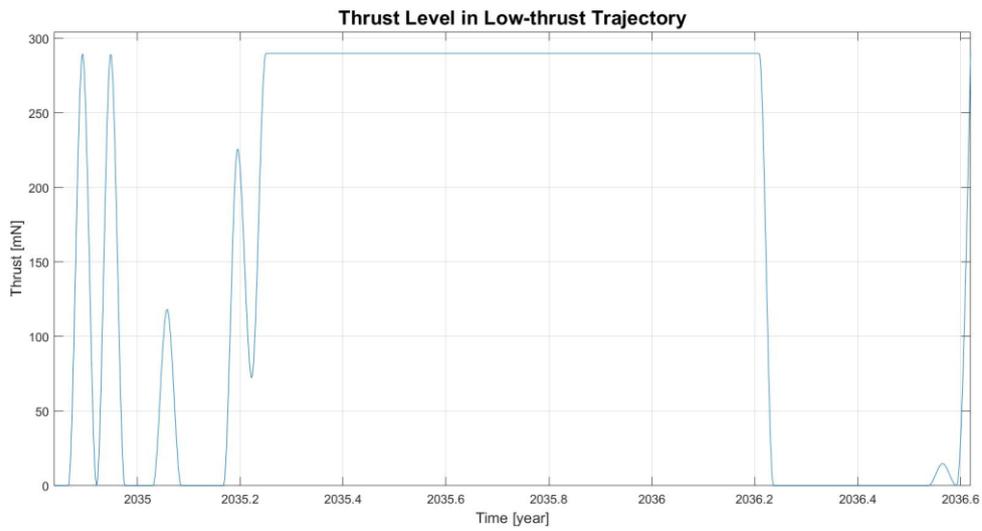
**Tabella 4.14 – Analisi NEP**

<b>Software</b>	DOPLOTT – NEP	<i>[-]</i>
<b>Departure Date</b>	2034 – 10 – 02 UTC 00:00	<i>[-]</i>
<b>Hyperbolic Excess Velocity <math>v_{\infty_{DEP}}</math></b>	0	<i>[km/s]</i>
<b>Arrival Date</b>	2036 – 07 – 13 UTC 00:00	<i>[-]</i>
<b>Hyperbolic Excess Velocity <math>v_{\infty_{ARR}}</math></b>	0	<i>[km/s]</i>
<b>Transfer Time</b>	650	<i>[day]</i>
<b>Mass Propellant</b>	257.975	<i>[kg]</i>
<b>Total DV</b>	5.7282	<i>[km/s]</i>
<b>Time Analysis</b>	3834	<i>[s]</i>

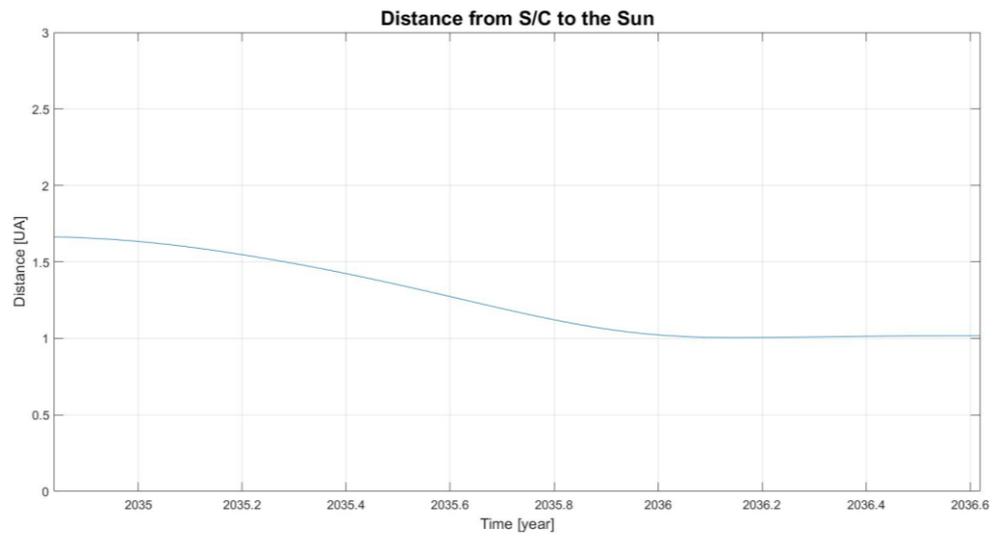
<b>Errori vettore di stato</b>		
$r_N(1) - r_f(1)$	0.019	<i>[km]</i>
$r_N(2) - r_f(2)$	0.073	<i>[km]</i>
$r_N(3) - r_f(3)$	0.025	<i>[km]</i>
$v_N(1) - v_f(1)$	$10^{-8}$	<i>[km/s]</i>
$v_N(2) - v_f(2)$	$10^{-9}$	<i>[km/s]</i>
$v_N(3) - v_f(3)$	$10^{-9}$	<i>[km/s]</i>



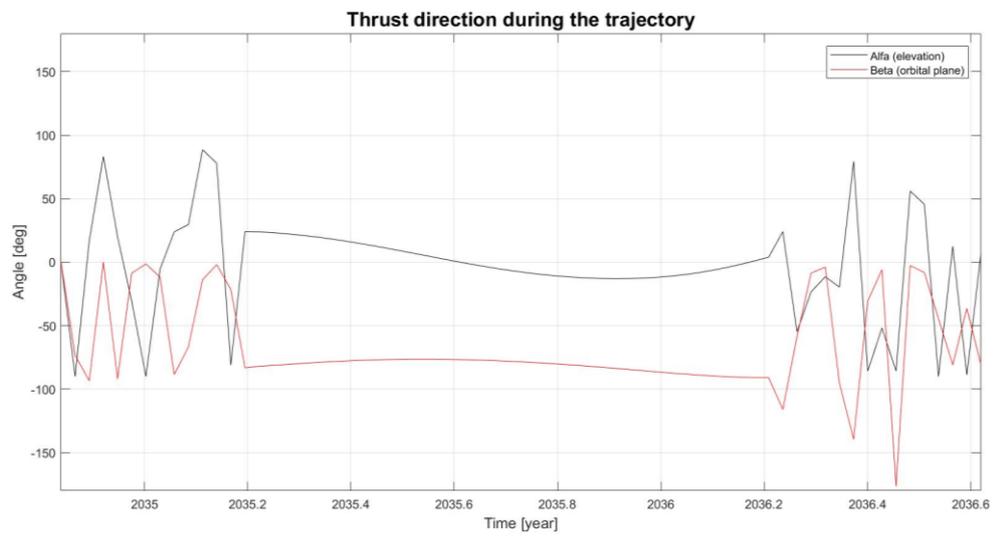
**Figura 4.11** – Output analisi NEP: traiettoria



**Figura 4.12** – Output analisi NEP: profilo della spinta



**Figura 4.14** – Output analisi NEP: andamento distanza dal Sole



**Figura 4.13** – Output analisi NEP: andamento direzioni della spinta

## 5 Conclusione e miglioramenti futuri

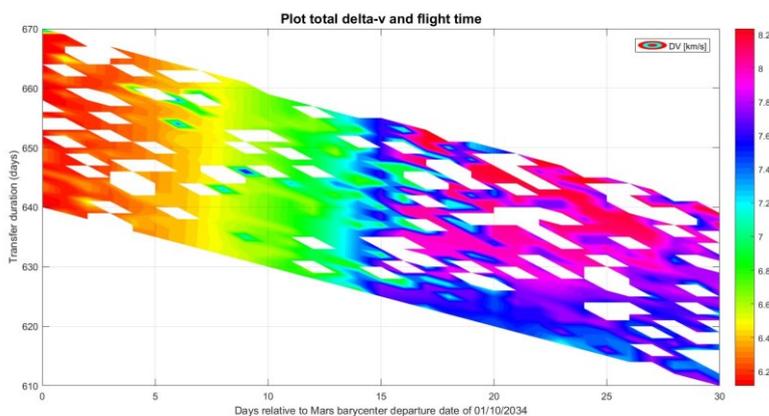
L'utilizzo delle manovre a bassa spinta in ambito di missioni interplanetarie è sempre più spesso preso in considerazione nella progettazione di future missioni per l'esplorazione del sistema solare. L'implementazione di questo problema prevede un'analisi molto più accurata rispetto al caso in cui si scelga di utilizzare come propulsione primaria la propulsione chimica. Ciò può generare notevoli vantaggi dal punto di vista del costo della missione a discapito della durata.

Lo studio effettuato in questa tesi ha permesso di definire un metodo per la modellizzazione e ottimizzazione di tali manovre. Le innumerevoli variabili presenti in un problema di manovre a bassa spinta, nonché la complessità degli algoritmi di ottimizzazione, causa l'impossibilità di seguire un unico approccio universale per la risoluzione di questo problema.

Da un primo approfondimento è stato possibile implementare un software che, fissate alcune assunzioni semplificative, permette di ottimizzare le manovre a bassa spinta. Il software DOPLLOT permette di ottimizzare traiettorie a bassa spinta avendo però fissato il ToF della missione (nonché data di partenza e di arrivo).

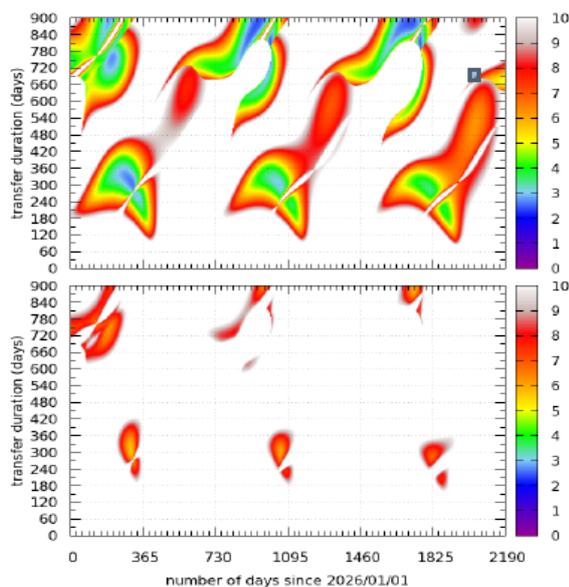
Il primo passo è stato quindi quello di definire tra tutte le traiettorie possibili a bassa spinta, quelle che prevedono un minor consumo di propellente. In ambito della Mission Analysis, l'obiettivo fondamentale è invece quello di valutare la traiettoria ottimale (minimo consumo di propellente) tra tutte le varie date possibili nel rispetto dei requisiti di missione.

I futuri miglioramenti del codice verranno effettuati con lo scopo di realizzare le analisi suddette. Un metodo parzialmente implementato si collega ai concetti legati ai Pork Chop.



**Figura 5.1** – Analisi dati metodo iterativo (primi risultati DOPLLOT)

Mediante un metodo iterativo, si analizzano tutte le traiettorie al variare della data di partenza e di arrivo (comprese tra due date limite) e se ne salvano i dati fondamentali.



**Figura 5.2** – Esempio di Pork Chop manovre impulsive<sup>25</sup>

Come si può osservare in figura 5.2, questo permette di rappresentare graficamente questi risultati, così da valutare le date in cui effettuare la missione con il minor consumo di propellente possibile. L'aspetto negativo di questo metodo è la durata delle analisi. Come visto nel capitolo precedente, l'analisi di validazione ha impiegato 10 minuti per raggiungere la convergenza. Tale valore andrebbe moltiplicato per ogni iterazione, ottenendo per uno studio di 2/3 mesi analisi dalla durata di parecchie ore.

Per risolvere questo problema sarà possibile valutare differenti approcci come ad esempio:

- Suddividere la traiettoria con archi di ampiezza maggiore così da ridurre le variabili in gioco e ridurre il costo computazionale a discapito dell'accuratezza. Successivamente restringere il campo ed effettuare analisi sempre più accurate nell'intorno delle condizioni ottimali.
- Sfruttare il concetto secondo il quale, dato un valore iniziale di tentativo prossimo al punto di minimo, il software convergerà più rapidamente. Per questa ragione, si potrebbe sfruttare come valore iniziale di tentativo al giorno di partenza  $n$ , la soluzione dell'analisi al giorno precedente. In prima approssimazione, le posizioni

<sup>25</sup> Fonte vedere riferimento bibliografico [10]

ricavate dall'effemeridi nell'arco di pochi giorni non si modificano in modo considerevole; utilizzare un approccio del genere permette di sfruttare una prima condizione iniziale di tentativo prossima a quella di ottimo.

Un metodo più complesso, che permette di ottenere le date in cui poter svolgere la traiettoria ottimale, prevede di sfruttare i potenti algoritmi di ottimizzazione. Con questo approccio non saranno però note le traiettorie nell'intorno del dato in questione.

Questo metodo prevede di inserire tra le variabili di stato anche le date caratteristiche della missione. In questo caso inserendo anche gli opportuni vincoli su queste variabili, sarà l'ottimizzatore stesso a valutarne la condizione ottimale tra tutte le date possibili.

Un'ulteriore modifica riguarda i possibili metodi di ottimizzazione. Uno studio utile potrebbe essere quello di confrontare il metodo Direct Shooting utilizzato nel software DOPLOT con un metodo Multiple Shooting come presentato nel capitolo precedente. Si potranno confrontare i tempi necessari al raggiungimento della convergenza, anche in funzione dell'aggiunta delle date tra le variabili.

Il software DOPLOT analizza qualsiasi traiettoria in assenza di perturbazioni. Un aggiornamento del codice, che ne migliorerebbe l'accuratezza, è l'inserimento delle perturbazioni legate alle radiazioni solari e alla presenza degli altri pianeti. Modificare il modello di propagazione della traiettoria renderebbe le analisi più veritiere.

Il fine ultimo è quello di un software che permetta l'ottimizzazione di traiettorie a bassa spinta nel minor tempo possibile e con la maggiore accuratezza, implementando tutte le modifiche appena presentate, e ad esempio anche l'influenza che, uno o più fly-by all'interno del sistema solare potrebbe avere sulla scelta della traiettoria ottimale.

## 6 Bibliografia

- [1] D. D. M. J. E. W. Roger R. Bate, *Fundamentals of ASTRODYNAMICS*, New York: Dover Publications, Inc, 2016.
- [2] A. A. Q. Giovanni Mengali, *Fondamenti di meccanica del volo spaziale*, Pisa: Pisa university press, 2013.
- [3] G. Marco, *SATELLITI ARTIFICIALI*, eBook: Simplicissimus.
- [4] S. N. F. Jon A. Sims, «Preliminary Design of Low-Thrust Interplanetary Missions,» ResearchGate, 2000.
- [5] T. B. John, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, Philadelphia: Society for Industrial and Applied Mathematics, 2010.
- [6] A. P. E. V. G. Avanzini, «Solution of Low-Thrust Lambert Problem with Perturbative Expansions of Equinoctial Elements,» *Journal of guidance, control and dynamics*, Pennsylvania, 2015.
- [7] NAIF, «SPICE Tutorial Package,» JPL - NASA, 2018.
- [8] S. Kemble, *Interplanetary Mission Analysis and Design*, Springer - Praxis Books, 2006.
- [9] J. D. P. Giorgio Fasano, *Modeling and Optimization in Space Engineering*, Turin: Springer, 2013.
- [10] R. Biesbroek, *Lunar and Interplanetary Trajectories*, Springer - Praxis, 2016.
- [11] D. Y. O. a. D. Landau, «A Simple Semi-Analytic Model for Optimum Specific Impulse Interplanetary Low Thrust Trajectories,» *The 32nd International Electric Propulsion Conference*, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 91109, USA, September 2011.
- [12] S. Boone, «A simplified low-thrust trajectory optimization framework with gravity assists,» [Online]. Available: [http://ccar.colorado.edu/asen5050/projects/projects\\_2015/Students/Boone\\_Spencer/extension.html](http://ccar.colorado.edu/asen5050/projects/projects_2015/Students/Boone_Spencer/extension.html).
- [13] C. Weiland, *Computational Space Flight Mechanics*, Heidelberg: Springer, 2010.
- [14] D. A. Vallado, *Fundamentals of astrodynamics and applications*, McGraw-Hill, 1997.

- 
- [15] J. Betts, «Survey of Numerical Methods for Trajectory Optimization,» *Journal of Guidance, Control and Dynamics*, Boeing Information and Support Services, Seattle, Washington 98124-2207.
- [16] J. Betts, «Very low-thrust trajectory optimization using a direct SQP method,» *Journal of Computational and Applied Mathematics*, Seattle, 1999.
- [17] A. Gallimore, «A Preliminary Investigation of Hall Thruster Technology,» NASA, CR-97-205777, 1997.
- [18] J. T. B. a. W. P. Huffman, «Application of Sparse Nonlinear Programming to Trajectory Optimization,» *Journal of Guidance, Control, and Dynamics*, Boeing Computer Services, Seattle, Washington 98124.
- [19] J. T. B. a. W. P. Huffman, «Path-Constrained Trajectory Optimization Using Sparse Sequential Quadratic Programming,» *Journal of Guidance, Control, and Dynamics*, Boeing Computer Services, Seattle, Washington 98124.
- [20] V. L. C. Jennifer J. Hargens, «Low-Thrust Interplanetary Mission Design Using Differential Inclusion,» *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Monterey, California, August 2002.
- [21] P. A. F. E. A. R. M. A. V. a. T. D. K. Jon A. Sims, «Implementation of a Low-Thrust Trajectory Optimization Algorithm for Preliminary Design,» *Jet Propulsion Laboratory*, Pasadena, California.
- [22] L. C. W. D. S. E. H. a. Z. A. M. M. C. Bartholomew-Biggs, «The Solution of some difficult problems in Low-Thrust Interplanetary Trajectory Optimization,» *Optimal Control Applications & Methods*, The Haijeld Polytechnic, HatJeld, Hertfordshire, 1988.
- [23] M. V. P. De Pascale, «Preliminary Design of Low-Thrust Multiple Gravity-Assist Trajectories,» *Journal of Spacecraft and Rockets*, 2006.
- [24] G. Pawlas, «Numerical Modeling of a Vortex Stabilized Arcjet,» NASA, CR-190797, 1992.
- [25] M. K. A. Payez, «Mars Sample Return Carrier - Mission Analysis Guidelines,» ESOC, 28/03/2017.
- [26] D. N. M. S. J. Edwards, «Response Surface Regressions for Low-Thrust Interplanetary Mission Design,» *SPACE Conferences and Exposition*, Long Beach, California, September 2016.
- [27] JAXA, «HAYABUSA - A technology Demonstrator for Sample and Return,» [Online]. Available: [http://global.jaxa.jp/projects/sat/muses\\_c/files/hayabusa\\_return.pdf](http://global.jaxa.jp/projects/sat/muses_c/files/hayabusa_return.pdf).

- [28] M. D. Rayman, «The Successful Conclusion of the Deep Space 1 Mission: important results without a flashy title,» Space Technology 23,, Pasadena, CA 91109 USA, 2003.
- [29] M. D. Rayman, «The Deep Space 1 Extended Mission: challenges in preparing for an encounter with comet Borrelly,» Acta Astronautica 51, Pasadena, CA 91109 USA, 2002.

*Sono profondamente grato al Prof. Ing. Lorenzo Casalino per i suoi preziosi consigli e per il tempo dedicatomi. Durante questo periodo di tesi, ho potuto apprezzare la sua professionalità e competenza che mi ha permesso di apprendere e di appassionarmi sempre più a questo fantastico mondo della meccanica orbitale.*

*Ringrazio inoltre la “Thales Alenia Space Italia” nella figura dell’Ing. Chiarelli Cosimo per avermi concesso, durante tutto il periodo di collaborazione, la disponibilità dei loro uffici e di qualsiasi altro mezzo utile alla realizzazione di questo lavoro di tesi.*

*Ringrazio l’Ing. Marco Berga, responsabile del team di Mission Analysis in TASI, per i suoi utili suggerimenti e con il quale ho condiviso ogni problematica e traguardo raggiunto.*

*Un grazie particolare all’Ing. Andrea D’Ottavio per l’infinita pazienza durante le mie più svariate domande; in bocca al lupo per la nuova avventura.*

*Il ringraziamento più grande va però alla mia famiglia senza la quale nulla sarebbe stato possibile, e a Raffa e Ale, due punti di riferimento nella mia vita. Grazie anche a te Giulia, ti aspettiamo con gioia.*