



**Politecnico
di Torino**

Politecnico di Torino

Corso di Laurea in Ingegneria Aerospaziale

A.A. 2023/2024

Sessione di Laurea: Dicembre 2024

**Ottimizzazione della traiettoria di
discesa sulla Luna
Metodo diretto**

Relatore:

Casalino Lorenzo

Candidata:

Franceschini Chiara
S306134

ABSTRACT

Questa tesi si occupa dell'ottimizzazione della traiettoria di discesa per un veicolo di atterraggio sulla superficie lunare. Lo scopo generale di questo lavoro è di ottenere le variabili di controllo ottimali che minimizzino il consumo di propellente e, di conseguenza, massimizzino la massa residua. Tale ottimizzazione consentirebbe un potenziale incremento del payload trasportabile, ampliando così le capacità della missione.

Partendo dalle radici dell'esplorazione lunare, il lavoro esplora la crescente attenzione verso la luna e in particolare verso gli studi di "soft-landing", che risultano importantissimi in un contesto in cui l'atterraggio autonomo diventa sempre più importante.

Tra i metodi di ottimizzazione esistenti, questa tesi impiega il metodo diretto, che suddivide la traiettoria in archi successivi e rappresenta i controlli tramite un elevato numero di parametri ottimizzabili. È stato realizzato un algoritmo in MATLAB che sfrutta la funzione `fmincon` per ottimizzare le traiettorie di discesa partendo da una soluzione iniziale di tentativo.

ABSTRACT

This thesis deals with the optimization of the descent trajectory for a landing vehicle on the lunar surface. The overall aim of this work is to obtain the optimal control variables that minimize propellant consumption and, consequently, maximize residual mass. Such optimization would allow a potential increase in the transportable payload, thus expanding the capabilities of the mission.

Starting from the roots of lunar exploration, the work explores the growing attention to the moon and in particular to "soft-landing" studies, which are very important in a context in which autonomous landing is becoming increasingly important.

Among the existing optimization methods, this thesis employs the direct method, which divides the trajectory into successive arcs and represents the controls through a large number of optimizeable parameters. An algorithm has been developed in MATLAB that uses the 'fmincon' function to optimize the descent trajectories starting from an initial retry solution.

SOMMARIO

Abstract	1
Abstract	2
Sommario	4
Introduzione	8
Capitolo 1	9
L'esplorazione della luna	9
Il programma Apollo	10
La Sospensione dell'Esplorazione e la Nuova Era Lunare	12
Il programma Artemis	13
Capitolo 2	20
Il problema della ottimizzazione	20
Metodo diretto e indiretto	21
Algoritmi di ottimizzazione lineare vincolata	22
Sequential Quadratic Programming	24
Capitolo 3	33
Il modello della discesa	33
Definizione del modello e ipotesi	35
Sistema di riferimento ed Equazioni	35
Variabili di controllo	37
Vincoli	37
Capitolo 4	38
Algoritmo e architettura dello strumento	38
Descrizione del codice	38
Risultati	41
Analisi con spinta costante (prova1.m)	41
Analisi con spinta costante e atterraggio verticale (spinta_var_con_t_att_fissa.m)	45
Angoli e spinta di atterraggio ottimale (spinta_var.m)	51
Sensibilità della spinta variabile al guess iniziale	56
Spinta variabile e atterraggio vincolato a 90°	62
Spinta variabile nella discesa al 95%Tmax (spinta_var_discesa.m)	63

Confronto delle velocità di atterraggio	73
Conclusioni	78
Lista delle figure	79
Lista delle tabelle	79
Bibliografia	80
Ringraziamenti	82

*A mia nonna,
che mi guarda dall'altro lato della Luna*

INTRODUZIONE

La luna ha da sempre affascinato l'uomo per la sua bellezza e misteriosità. Con il progresso tecnologico è diventata, durante la guerra fredda, oggetto di forte interesse scientifico sia da parte della sfera americana e che da quella sovietica.

Questo lavoro di tesi affronta un'iniziale panoramica sul contesto in cui è nata l'esplorazione lunare. Le prime missioni hanno le radici nel programma Apollo, con il quale, nel 1969, l'uomo poggiò per la prima volta i piedi sulla Luna.

Si passerà poi alle sfide attuali che riguardano la Luna. Al giorno d'oggi, settore privato e pubblico collaborano, in chiave internazionale, al fine di costruire programmi come Artemis. Gli obiettivi sono sempre più sfidanti e, per portarli a termine, si necessita di fondamenta solide e programmi ampi e ben strutturati.

Viene presentato, dunque, lo scopo di questo studio di tesi, ovvero l'ottimizzazione della traiettoria della discesa sulla luna. In particolare, dalla ipotesi di trovarsi su una orbita di parcheggio a 15km dalla superficie lunare, si analizza da questa quota quale sia la scelta ottimale delle variabili di controllo al fine di atterrare minimizzando i consumi.

Il modello adottato sarà un modello semplificato del problema reale di discesa, in particolare si farà riferimento a un modello bidimensionale e in cui si trascura la curvatura della superficie lunare.

L'ottimizzazione viene eseguita con il metodo diretto, un metodo vantaggioso in quanto semplice e intuitivo benché possa essere più costoso a livello computazionale se il problema è complesso.

Prima di riassumere i risultati ottenuti, sarà approfondito il modo in cui è stato strutturato il codice e in particolare la funzione di MATLAB chiamata 'fmincon'. Tale funzione, che risulta fondamentale in questo contesto, è stata utilizzata con l'algoritmo SQP (Sequential Quadratic Programming), di cui verrà fornita una spiegazione nel dettaglio.

Il codice ha dei limiti che verranno presentati, in particolare risulta molto sensibile ai guess iniziali sulle variabili. Nonostante questo limite, l'algoritmo rappresenta però un buon punto di partenza per analisi più approfondite e sviluppi futuri, anche se sarebbero necessarie migliorie per renderlo più robusto ed efficace.

L'ESPLORAZIONE DELLA LUNA

Fin dai tempi antichi, la luna, luminosa nel cielo, ha catturato l'interesse dell'uomo. La formazione della luna si pensa risalga a 4.5 miliardi di anni fa, di poco successiva alla terra. La sua orbita si colloca a una distanza media di circa 380 mila chilometri dalla Terra, con un periodo orbitale di 27 giorni, 7 ore e 43 minuti. Dalla superficie terrestre è possibile osservare ad occhio nudo le sue macchie scure e chiare, i suoi crateri e imperfezioni, che incuriosiscono sulla sua origine e trascorso travagliato. Il diverso colore di queste zone buie o luminose è dovuto al fatto che le prime, dette mari, sono costituite da rocce di basalto e altri materiali di colore scuro, mentre le seconde sono regioni con una altitudine molto più elevata rispetto a quelle dei mari che sfiorano anche i 9000 metri. Inoltre, la faccia visibile della Luna è costellata da centinaia di migliaia di crateri da impatto che hanno almeno mille metri di diametro. Il cratere lunare più importante però si trova sulla faccia nascosta, nella porzione meridionale. Si tratta del Polo Sud-Aitken, che ha un diametro di circa 2500 chilometri e una profondità di una decina di chilometri.

L'esplorazione della luna ci ha permesso di conoscere il suo lato sconosciuto, nel senso che, essendo in rotazione sincrona, la luna mostra verso la Terra sempre la medesima faccia.

Tale faccia, però, non appare sempre con la stessa forma, infatti, durante il moto orbitale, l'orientazione rispetto al sole porta a diverse fasi. L'uomo preistorico era interessato all'astronomia e le prime testimonianze risalgono al Paleolitico superiore. Il cielo stellato ha svelato ai nostri antenati le scadenze della natura. Sono state proprio le fasi lunari ad avere rivelato all'uomo il concetto del tempo anche tramite l'influenza che le fasi hanno sui raccolti e la fertilità. La luna risulta quindi indissolubilmente legata alla vita sulla Terra, regolando il ciclo riproduttivo di molte specie, influenzando sulle maree e sulla stabilità dell'asse di rotazione terrestre.

Il mese lunare, diversamente da quello terrestre, è di 29.5 giorni ed è caratterizzato da otto fasi principali. Quando la luna si trova tra la Terra e il Sole si ha la 'luna nuova', in cui il lato illuminato non è visibile dalla terra. Dopo la luna nuova, un piccolo spicchio diventa visibile e si è nella fase di 'luna crescente'. Nel momento in cui la metà destra della luna è visibile si è al 'primo quarto'. Seguirà la fase di 'gibbosa crescente', che si conclude con la 'luna piena', in cui la luna si trova, rispetto al sole, dalla parte opposta della terra. La 'gibbosa calante', 'ultimo quarto' e 'luna calante' sono le fasi successive e opposte a quelle citate.

Le fasi lunari, per tradizioni secolari contadine, si ritiene che abbiano effetti diversi sui raccolti e che convenga seminare in luna crescente e raccogliere in luna piena.



Figura 1: Fase lunare, gibbosa crescente [23]

Tra le caratteristiche più importanti sulla luna va segnalato che ha delle dimensioni così ridotte che risulta troppo piccola per trattenere un'atmosfera attorno a sé. Non soffia alcun vento e di conseguenza il lato illuminato dal sole supera i 100° , mentre il lato in ombra è più freddo di ogni punto della Terra, arriva fino a -180° [8].

Le nostre conoscenze sulla luna derivano in gran parte dalle informazioni raccolte da sonde orbitanti e veicoli lunari. Tali esplorazioni iniziarono tra gli anni Sessanta e Settanta. In precedenza, tra alcuni scienziati, circolava l'idea che, se uno spacecraft fosse atterrato sul suolo lunare sarebbe sprofondato nella sua superficie polverosa. Con l'avanzamento della tecnica e tecnologia si dimostrò un terreno solido, dando così il via all'esplorazione umana della Luna.

Il programma Apollo

L'esplorazione lunare ha le sue radici nella "Corsa allo Spazio" tra Stati Uniti e Unione Sovietica negli anni '50 e '60, in pieno contesto di Guerra Fredda. Il primo grande traguardo fu raggiunto nel gennaio 1959, quando la sonda sovietica Luna 1 volò a 6000 km dalla superficie lunare. Luna 2, nello stesso anno, divenne il primo oggetto umano a impattare sulla superficie lunare. Purtroppo, l'atterraggio non fu leggero, infatti Luna 2 si schiantò nel mare Imbrium. Pochi mesi dopo, Luna 3 trasmise le prime immagini del lato sconosciuto della Luna, fino ad allora mai osservato. La faccia nascosta, sempre opposta alla Terra, fu osservata grazie alle immagini che Luna 3 inviò sulla Terra nell'ottobre 1959. Queste missioni senza equipaggio rappresentavano la base per

le esplorazioni più ambiziose, e fornirono un'enorme quantità di conoscenze fondamentali per le missioni che seguirono.

La risposta degli Stati Uniti non tardò ad arrivare; a partire dal 1961 inviarono nove spacecraft NANA Ranger in soli quattro anni, fornendo per la prima volta immagini di prossimità. Questa serie di missioni partì con l'obiettivo di catturare più immagini possibili prima di schiantarsi sulla superficie lunare. Tramite il patrimonio fotografico degli Stati Uniti è stata compresa meglio la morfologia e sono risultate evidenti le irregolarità del suolo. L'uomo divenne consapevole della difficoltà di trovare un sito di atterraggio consono a missioni umane.

Fu il presidente John Kennedy a lanciare l'ambizioso obiettivo di portare sulla luna l'uomo entro la fine del decennio. Nel 1961 nacque così il programma Apollo. L'obiettivo principale del programma, oltre a far atterrare sulla luna e riportare sulla terra l'uomo, era quello di studiare il suolo e l'origine del sistema solare. Intanto, la concorrenza sovietica riuscì nel 1966 a far atterrare in sicurezza Luna 9, che disponeva di dispositivi di comunicazione e possibilità di raccogliere immagini dal suolo. Nello stesso anno Luna 10 diventò il primo orbiter e Surveyor della NASA e atterrò con l'obiettivo di esplorare e analizzare il suolo. Negli anni successivi la NASA mappò quasi la totalità della superficie lunare grazie alle cinque missioni Lunar Orbiter. Grazie a questo lavoro vennero individuati dei possibili siti di atterraggio potenzialmente adatti per portare l'uomo sul suolo lunare.

Intanto il programma Apollo, dopo una serie di missioni preparatorie, arrivò alla storica missione dell'Apollo 11 con la quale Neil Armstrong e Buzz Aldrin divennero parte della storia [22]. Prima dell'atterraggio del luglio 1969 vanno ricordate Apollo 1 del 1967, in cui morirono tre astronauti per un incendio a terra nel modulo di comando, Apollo 7 e 9 missioni in orbita terrestre per verificare il funzionamento in microgravità del CSM e LM e Apollo 8 e 10 che orbitano attorno alla luna testando l'atterraggio. Dopo Apollo 11, la NASA condusse altre sei missioni verso la Luna, cinque delle quali riuscirono a raggiungere la superficie. L'ultima missione, Apollo 17 nel 1972, concluse ufficialmente il programma e fu anche l'ultima occasione in cui l'uomo toccò la superficie lunare.

In totale, tra il 1969 e il 1972, ben 12 uomini passeggiarono sulla luna, avviando esperimenti scientifici e portando sulla terra quattro quintali di rocce. Nelle ultime missioni gli astronauti hanno anche esplorato a bordo di veicoli a batteria.

L'architettura del programma Apollo comprendeva il razzo Saturn V, il modulo di comando e servizio per trasportare gli astronauti e farli rientrare sulla terra, il modulo lunare con lo stadio di discesa (che conteneva il motore per l'atterraggio e le attrezzature per le operazioni lunari) e lo stadio di ascesa, usato per riportare gli astronauti dal suolo lunare all'orbita. Il Saturn V era un razzo multistadio, ovvero diviso in varie sezioni chiamate 'stadi', ognuna con il proprio sistema propulsivo in grado di accendersi in sequenza. Esaurito il propellente del primo stadio, essendo tale struttura non più utile ai fini della missione, si separava e seguiva l'accensione del secondo stadio e così via. Un razzo multistadio, rispetto a un razzo con un unico stadio, permette di portare un payload maggiore

in orbita in quanto elimina la massa dei serbatoi e delle strutture che non vengono più utilizzate.

Il Saturn V aveva delle dimensioni notevoli, vantava un diametro di 10m e una altezza di 110 m. Il peso superava di poco le 3000 tonnellate e forniva al decollo una spinta di 34 milioni di Newton; i motori del primo stadio dovevano fornire una spinta maggiore del peso dell'intero razzo al fine di farlo accelerare.

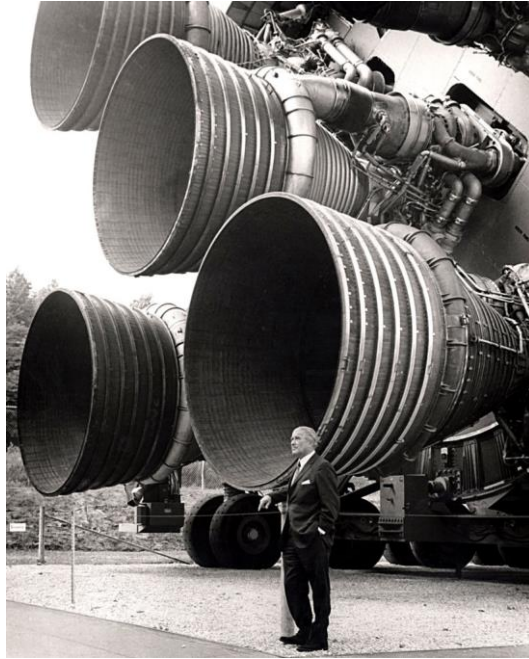


Figura 2: Motori del Saturn V [24]

Da allora, l'interesse per la Luna ha attraversato fasi alterne, ma oggi viviamo un periodo di rinnovato entusiasmo, motivato sia dalla possibilità di insediamenti umani sia dalle preziose risorse che la Luna potrebbe offrire per future missioni di esplorazione.

Tra le motivazioni per cui negli ultimi anni molti paesi si sono orientati verso le esplorazioni della Luna vi è la scoperta della presenza di ghiaccio sul fondo dei crateri nei pressi dei poli lunari. Essendo in zone d'ombra, non esposte ai raggi solari, questi ghiacci potrebbero essere lì da milioni di anni e potrebbero venire convertiti in acqua potabile e ossigeno dai futuri esploratori.

La Sospensione dell'Esplorazione e la Nuova Era Lunare

Le missioni lunari, dopo il grande entusiasmo della guerra fredda, hanno subito una battuta di arresto dovuta sia all'alto costo sia al fatto che la priorità passò ad altre tipologie di missioni spaziali. In particolare, le missioni umane, ovvero quelle che oltre al payload portavano anche dell'equipaggio, risultavano più

costose in quanto avevano dei requisiti sulla sicurezza più stringenti, oltre al fatto che richiedevano funzionalità in più per supportare la vita degli astronauti. Tuttavia, negli anni 90', sia la NASA che le agenzie spaziali europee ESA e giapponese JAXA, hanno fatto partire delle missioni robotiche che avevano come obiettivo sempre la luna. Con queste missioni si continuò a mappare e studiare il satellite. Dal novembre 2004 ha orbitato attorno alla luna SMART-1, che è stata la prima sonda lunare europea. Ha lavorato fino a settembre 2006 scattando immagini dettagliate della superficie e studiando le rocce. La missione si è conclusa con uno schianto della sonda sul suolo della luna che era stato programmato in fase di progettazione. Al giorno d'oggi, le diverse agenzie spaziali dei vari paesi collaborano a livello internazionale per realizzare missioni molto complesse con obiettivi ambiziosi, tra cui l'uso della luna come avamposto logistico per dirigersi su Marte. Oltre agli enti nazionali sono comparsi anche grandi aziende private. La NASA recentemente ha avviato il programma Artemis, il quale ambisce a riportare l'uomo sulla luna nei prossimi anni. Tale programma si propone di stabilire sulla luna una presenza umana sostenibile e di estrarre risorse che permettano di garantire autosufficienza.

Dall'altra parte del globo, la Cina, allo stesso tempo, è riuscita a portare sulla terra campioni di rocce grazie alla missione Chang'e 5. Anche l'India e altre nazioni stanno sviluppando missioni autonome per esplorare la luna.

Il futuro delle missioni sulla luna e, l'autonomia di quest'ultime, si deve far forza della robotica per esplorare e ridurre i costi delle missioni. Per far ciò è necessario vincere sfide tecniche come lo sviluppo di sistemi avanzati di navigazione e anche atterraggio. Grazie a questi progressi sarà possibile standardizzare tali missioni e renderle più facili in un contesto in cui l'obiettivo è avere degli insediamenti spaziali e facilitare quelle missioni che vogliono andare nello spazio profondo e avere una durata lunga.

Per esempio, il ghiaccio che è stato scoperto all'interno dei crateri polari potrebbe essere utilizzato per generare acqua potabile oppure potrebbe essere scomposto in ossigeno e idrogeno per utilizzarli come propellente per i razzi.

Il programma Artemis

Il programma della NASA Artemis vuole tornare sulla luna combattendo barriere di genere e di colore della pelle, infatti ambisce a far atterrare sia la prima donna che la prima persona di colore sul nostro satellite naturale. Le tecnologie utilizzate dal programma Artemis sono di ultima generazione e si propongono di esplorare più che mai la superficie lunare. Il programma ha sue radici nella NASA ma è molto ramificato tra partner commerciali che non sono solo degli Stati Uniti ma sono anche internazionali. Grazie a questa collaborazione, l'obiettivo è stabilire sulla luna il primo insediamento a lungo termine e costruire una stazione spaziale nell'orbita lunare. Inoltre, il passo successivo sarà quello di utilizzare le informazioni che si accumuleranno sulla luna per spingersi oltre, facendo il passo di inviare i primi astronauti su Marte.

Il termine Artemis deriva dalla dea Artemide, sorella di Apollo, dea della caccia e della luna. Artemis I è stata la prima missione del programma ed è avvenuta nel 2022. Durante Artemis I il nuovo mega razzo della NASA ruggì nella notte del 16 novembre. La capsula Orion partì per un viaggio oltre la luna e ritorno. Il volo fu senza uomini a bordo e fu il primo test di una serie al fine di rendere la missione sempre più complessa e capace di portare gli uomini sulla luna. Il decollo avvenne dal sito di lancio della NASA in Florida, ovvero il Kennedy Space Center. Una valutazione iniziale mostrò che il razzo funziona con precisione superando ogni aspettativa.

Per quanto riguarda il sistema di lancio principale progettato dalla NASA per il programma Artemis, ci si riferisce allo Space Launch System (SLS). Questo lanciatore super-heavy verrà utilizzato anche per i programmi futuri, eventualmente con direzione Marte. Il veicolo ha origine dalle ceneri di Constellation, il programma del presidente Bush che nel 2005 voleva sostituire l'obsoleta flotta degli Space Shuttle.

È un lanciatore non riutilizzabile che ha un costo per lancio che supera i 4 miliardi di dollari statunitensi. Aerojet Rocketdyne, Boeing e Orbital ATK sono stati i principali produttori. SLS presenta due stadi con 2 razzi ausiliari e ha dimensioni di oltre 100m di altezza e circa 8m di diametro.

I razzi ausiliari sono a propellente solido di percolorato di ammonio e polibutadiene acrilonitrile.

Il primo stadio, chiamato 'Core Stage', è dotato di quattro propulsori RS-25D/E con una spinta di 7440 kN, ha un impulso specifico di quattro secondi nel vuoto e utilizza il propellente idrogeno-ossigeno. Il primo stadio consiste di 5 parti ovvero la forward skin, il serbatoio di ossigeno liquido, il serbatoio intermedio, il serbatoio di idrogeno liquido e per finire il vano motori.

Il secondo stadio 'Interim Cryogenic Second Stage' ha un propulsore RL10-B-2 con una spinta di 110kN e funziona con lo stesso propellente del primo stadio. Questo stadio fornirà la grande spinta necessaria ad Orion per volare oltre la luna, prima di tornare verso la terra. Per missioni future di più lunga durata nello spazio profondo, l'ICPS verrà sostituito con un'altro più potente. Ritornando alla missione Artemis I, l'ICPS ha trasportato 13 satelliti, del tipo cubesat, nei vani vuoti della struttura.

Spostando l'attenzione invece sulla capsula Orion, questa è un veicolo spaziale progettato per trasportare equipaggio. Il veicolo è stato costruito dalla NASA in modo che sia parzialmente riutilizzabile, con una vita operativa fino ad un massimo di 10 voli circa. Come lo Space launch system, anche Orion è basata sul programma Constellation, secondo il quale era prevista la Crew Exploration Vehicle per portare gli astronauti sulla luna. I principali costruttori sono Lockheed Martin, per il modulo dell'equipaggio, e Airbus per il modulo di servizio.

La navicella permette un equipaggio da quattro a sei persone e il suo primo lancio risale al 2014, il quale però è stato solamente un volo di collaudo senza equipaggio. Orion, nel novembre 2022, ha volato verso la luna sempre

senza equipaggio, mentre il primo volo con equipaggio è previsto prossimamente.

Lo stile della struttura di Orion si ispira al modulo di comando e servizio del programma Apollo, però ovviamente con una tecnologia di ultima generazione. Orion si divide in due componenti principali: il modulo dell'equipaggio a forma di cono, e il modulo di servizio con i motori e i rifornimenti. La scelta di optare per soluzioni simili ad altre già testate è dovuta al fatto che in questo modo si riducano i rischi della missione.

Rispetto al programma Apollo, la capsula in questione contiene più astronauti rispetto ai tre che poterono viaggiare verso la luna in passato.



Figura 3: Decollo del SLS, 16 Novembre 2022 [25]

La missione Artemis I ha avuto il ruolo di apripista, testando il funzionamento del razzo SLS e della capsula Orion. Orion, è stata capace di orbitare alcuni giorni attorno alla luna con tutti gli strumenti di bordo verificati funzionanti.

La missione nello specifico, dopo il decollo, ha previsto che Orion fosse in orbita terrestre per alcune ore. Dopo 18 minuti dal lancio si aprirono le quattro ali dei pannelli solari del modulo di servizio europeo, al fine di caricare le batterie dello spacecraft. Circa 90 minuti dal lancio, l'ICPS si accese per accelerare Orion verso la luna, mentre 20 minuti dopo si separò dalla capsula. A questo punto ICPS rilasciò i payload secondari, ovvero dei satelliti di piccole dimensioni per testare nuove tecnologie a scopo scientifico. Il primo satellite ad essere rilasciato è stato Argomoon, un cubesat con lo scopo di scattare e inviare a terra le immagini dell'ICPS.

Intanto la capsula Orion e il modulo di servizio europeo continuarono il viaggio verso la luna, eseguendo correzioni di traiettoria con il motore dell'OMS. Furono necessari degli aggiustamenti per il flyby lunare, con il quale si raggiunse la distanza di circa 105.5 miglia dalla superficie. Il target del flyby fu l'inserzione nella orbita distante. Nel flyby propulso, il motore principale intervenne per una grande spinta lontano dalla luna, con un cambio di velocità di quasi 200m/s, per

inserirsi su un'orbita distante e retrograda attorno alla Luna. Orion realizzò metà rivoluzione su questa DRO, per una durata di 6 giorni a circa 44000 miglia dalla superficie lunare.

Terminata questa fase si riaccese il motore del modulo di servizio europeo, al fine di iniziare la partenza dall'orbita retrograda distante. Sfruttando una seconda volta il principio della fionda gravitazionale, Orion ripartì per il viaggio di ritorno verso il nostro pianeta.

L'OMS deve produrre un ΔV di 91 m/s per spostare Orion su una rotta a un raggio inferiore, più prossima alla Luna. L'OMS accelerò poi Orion in rotta verso la Terra, fornendo 282 m/s di variazione di velocità.

Nel dicembre del 2022 Orion rientrò nell'atmosfera terrestre con un ammaraggio nell'oceano Pacifico dopo una missione di quasi 26 giorni attorno alla luna. Il rientro fu molto critico perché la sonda raggiunse temperature pari alla metà di quelle che si trovano sulla superficie del sole. Nonostante questo, riuscì a rientrare grazie anche all'utilizzo di un paracadute, che frenò la fase finale della discesa. Lo spacecraft superò il record che era detenuto dall'Apollo 13 in quanto percorse 270.000 miglia di distanza, imponendosi come la sonda progettata per il trasporto umano ad essere arrivata più lontano.

La seconda missione del programma Artemis, ufficialmente Artemis II, sarà la prima con equipaggio. Al momento, il lancio con lo Space Launch System è programmato per settembre 2025. Artemis II prevede un sorvolo della luna e ritorno sulla terra. È dal 1972 che un veicolo spaziale con equipaggio non viaggia oltre l'orbita bassa lunare, tale missione verrà quindi ricordata come un traguardo storico. I membri dell'equipaggio saranno quattro: Gregory Wiseman, Victor Glover, Christina Koch, Jeremy Hansen. L'equipaggio quindi comprenderà la prima donna, la prima persona nera e la prima persona non statunitense a viaggiare oltre l'orbita bassa.

Il piano della missione prevede un sorvolo lunare di massimo 21 giorni. Rispetto alla missione Artemis I, il vettore sarà leggermente modificato nella variante Block 1. La traiettoria di trasferimento che dalla terra porta alla luna prevede più accensioni e infatti viene categorizzata come una 'multi-trans-lunar-injection' (MTLI). Eseguirà un flyby della luna ad un'altezza di 7500km dalla superficie lunare. Il ritorno, invece, prevede un viaggio di quattro giorni con numerose manovre correttive implementate dai motori ausiliari del modulo di servizio di Orion. Per quanto riguarda la fine della missione, è previsto un ammaraggio nell'Oceano Pacifico del veicolo di rientro con l'aiuto di paracaduti.

Il terzo volo pianificato della capsula Orion sarà la missione Artemis III. Questa missione durerà circa un mese e sarà provvista di equipaggio, però rispetto ad Artemis II, prevede allunaggio. L'ultimo allunaggio risale al lontano 1972, con l'equipaggio dell'apollo 17.

Artemis III era prevista, almeno inizialmente, per il 2025. In seguito ai ritardi nello sviluppo di Starship, la missione è stata ritardata dall'annuncio dei funzionari della NASA. Essi hanno definito come improbabile una missione di allunaggio con equipaggio, prima del 2027. Nell'agosto del 2023 è stato anche

affermato che potrebbe partire una missione senza equipaggio per un allunaggio di prova, mentre un allunaggio umano potrebbe essere gestito con una missione successiva, Artemis IV.

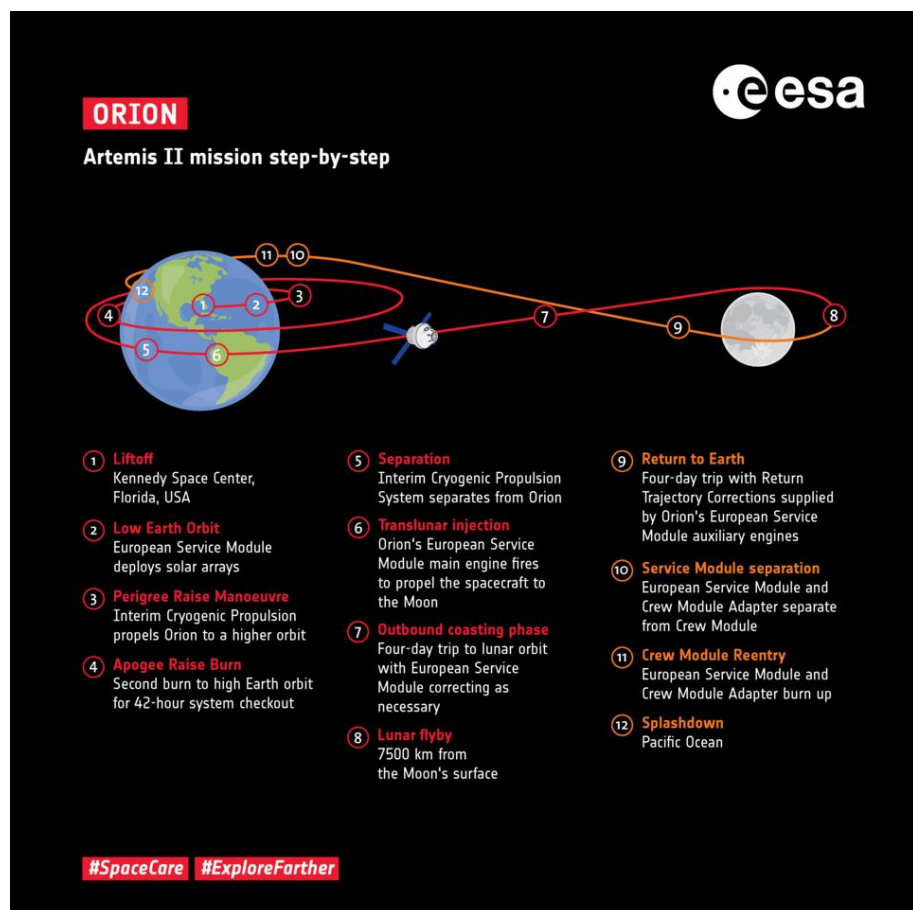
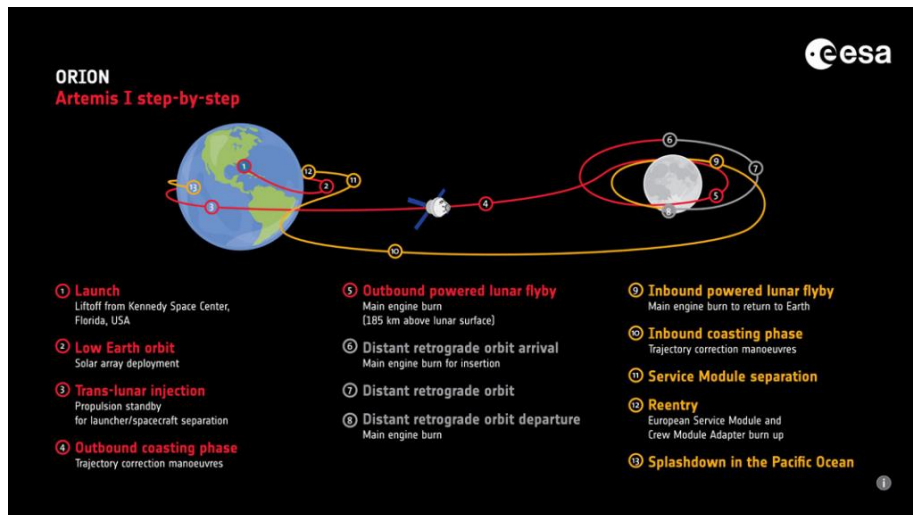
Artemis III ha l'intenzione di allunare la capsula nella regione del Polo Sud della luna. L'intenzione è quella di portare due astronauti sul suolo lunare e farli rimanere per circa una settimana. In particolare, il desiderio è quello di portare una donna sul satellite. I rimanenti due astronauti saranno a bordo del Gateway in orbita.

La passeggiata sul suolo lunare avrà come obiettivo quello di realizzare osservazioni scientifiche e campionare il ghiaccio che si è scoperto trovarsi al polo lunare. Prima del landing, verranno posizionate sulla zona di atterraggio delle attrezzature di supporto. Tra quest'ultime è previsto un Rover non pressurizzato che verrà utilizzato dagli astronauti per muoversi. Il Rover sarà controllato anche a distanza.

La missione Artemis III decollerà dal Kennedy Space Center, in Florida. Orion sarà lanciata dal Space Launch System (SLS) su una orbita bassa dove dispiegherà i pannelli solari, consentendo il funzionamento autonomo dei suoi sistemi. Per un certo periodo Orion rimarrà in una modalità di standby. Al momento opportuno avverrà l'iniezione translunare. Il razzo spingerà Orion verso la Luna dando il via alla fase di crociera. In questa fase manovre di correzione della traiettoria saranno necessarie per ottimizzare il viaggio. Una volta in prossimità della luna ci sarà un sorvolo, ovvero un passaggio ravvicinato alla superficie lunare. L'accensione del motore principale spingerà lo spacecraft nell'orbita lunare desiderata, tale da permettere l'aggancio al Gateway. Il Gateway servirà come punto di partenza per le operazioni future. Sarà distribuito in un'orbita eccentrica quasi rettilinea con un periodo di circa 7 giorni. Il vantaggio di questa orbita è il tempo minimo di blackout della comunicazione con la terra. L'orbita consentirebbe di raggiungere un'orbita polare bassa in mezza giornata con un ΔV di 730 m/s. Un altro vantaggio è che questa orbita richiederebbe solamente 10m/s di station keeping all'anno e cambiamenti di inclinazione non troppo costosi, in modo da raggiungere qualsiasi punto sulla superficie lunare. Gli astronauti utilizzeranno un lander per scendere sulla superficie della Luna e torneranno sul Gateway per prepararsi al viaggio di ritorno. Il modulo Orion eseguirà una serie di accensioni per lasciare l'orbita lunare e iniziare il rientro. Il modulo di servizio si separerà dalla capsula. L'ultima fase prevede il recupero degli astronauti e della capsula, che con paracaduti si tufferà nell'oceano Pacifico.

Data la complessità del programma e delle missioni viene riportata di seguito a scopo illustrativo una serie di immagini iconografiche dell'Esa, la quale spiega nel dettaglio in modo efficace e sintetico, le orbite scelte per effettuare la missione Artemis.

È opportuno notare che nella missione III, essendo la destinazione il polo sud, l'orbita di atterraggio del lander dovrà essere necessariamente un'orbita polare.



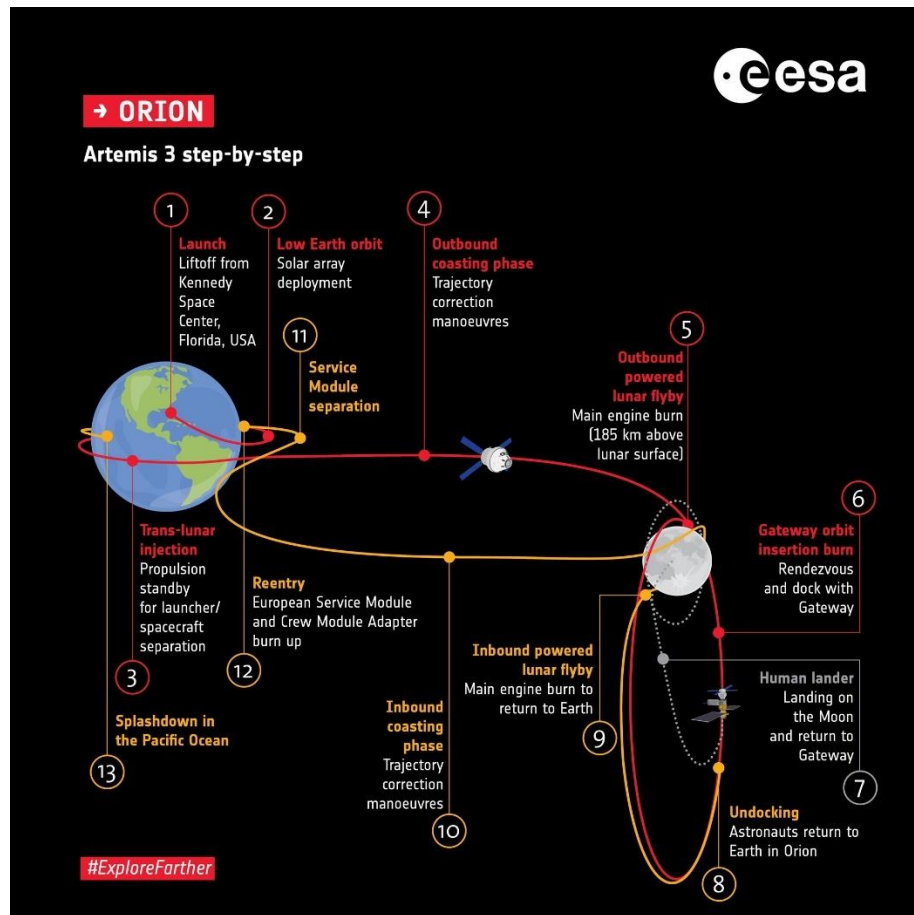


Figura 4: Immagini iconografiche sul programma Artemis, fonte NASA [26]

IL PROBLEMA DELLA OTTIMIZZAZIONE

In campo aerospaziale ci sono molte applicazioni in cui il problema è controllare lo spacecraft al fine di raggiungere un particolare obiettivo del controllo. In una missione di rendezvous e docking, per esempio, è necessario un controllo terminale della fase di docking. Il controllo terminale serve per portare lo spacecraft il più vicino possibile a uno stato terminale entro un determinato periodo di tempo. Questo si realizza nella pratica attuando una minimizzazione degli errori di posizione e atteggiamento al punto di docking (punto di attracco). Una seconda tipologia di controllo è il controllo del tempo minimo che si propone di raggiungere lo stato terminale più velocemente possibile [1]. Lo svantaggio di questo controllo rispetto a quello terminale è che è meno preciso e accurato nello stato finale a cui si arriva. Infatti, all'inizio il comando è impostato al valore massimo, mentre, dopo una durata specifica viene abbassato. Per completare la panoramica delle tipologie di controllo vanno citati il controllo di tracciamento e il rigetto delle perturbazioni. Il primo serve a forzare lo stato di uno spacecraft in modo che segua una traiettoria desiderata, quindi viene minimizzata la differenza tra la posizione desiderata e quella attuale. Viene usato per i lanciatori e nelle ultime fasi delle missioni di rendez-vous.

Il secondo, diversamente, serve a contrastare forze perturbative che potrebbero allontanare lo spacecraft dallo stato desiderato. Il controllore utilizza una funzione chiamata "rigetto delle perturbazioni" che elabora la perturbazione e fornisce comandi correttivi.

In questa tesi viene invece affrontato il problema del controllo minimo dell'energia o del minimo consumo di carburante. Questo è usato per trasferire la navicella spaziale da uno stato iniziale allo stato finale con il minimo consumo di propellente.

Per affrontare tali problemi esistono varie tecniche di controllo, di seguito verranno trattate le tre principali tecniche di controllo utilizzate nelle applicazioni industriali e aerospaziali. Il controllo ottimo, di cui esempi comuni sono il Controllo Predittivo (MPC) e il Controllo Lineare-Quadratico-Gaussiano (LQG), ottimizza un determinato "indice di costo", come minimizzare i consumi o il tempo. Il controllo ottimo si usa se il sistema è ben modellato e stabile e l'utilizzo ha il vantaggio di garantire stabilità in anello chiuso. Lo svantaggio è che variazioni improvvise non modellate costituiscono una limitazione alla sua efficacia.

Alternativamente esistono altre metodologie; il controllo robusto è utilizzato quando ci sono incertezze sui parametri del satellite (come massa e

inerzia) e le perturbazioni e i rumori sono sconosciuti. Tale controllo robusto si occupa di trovare una legge di controllo che mantenga la risposta del sistema e il segnale di errore entro tolleranze prescritte, nonostante le incertezze. La progettazione prevede un controller fisso che garantisce alte prestazioni anche in presenza di incertezze sugli input e output del veicolo spaziale. Le incertezze possono includere: effetti di disturbi sui parametri del veicolo spaziale, rumore di misura, errori di modellazione dovuti a non linearità e parametri variabili nel tempo.

Per ultimo, il controllo intelligente, utilizza vari approcci di intelligenza artificiale come reti neurali, logica fuzzy, apprendimento automatico, calcolo evolutivo e algoritmi genetici per controllare le dinamiche del veicolo spaziale. Il controller intelligente è in grado di mantenere un comportamento stabile in presenza di incertezze, che possono includere: incertezze nei modelli, ma anche cambiamenti ambientali imprevedibili, sensori incompleti o inaffidabili, oppure malfunzionamenti degli attuatori.

Metodo diretto e indiretto

Per risolvere problemi di controllo ottimo esistono due approcci principali per effettuare l'ottimizzazione: il metodo diretto e il metodo indiretto.

Il metodo diretto si basa sulla discretizzazione della traiettoria in un numero finito di archi, passando quindi da una traiettoria continua a una discretizzata. Trova applicazione in dinamiche spaziali e controlli ottimali, dove le condizioni al contorno e i vincoli possono essere gestiti direttamente. Il problema di ottimizzazione viene dunque ricondotto a un problema di programmazione non lineare (NLP). La dinamica complessa di una traiettoria spaziale si trasforma di fatto in un problema numerico. I principali vantaggi del metodo diretto sono semplicità e intuizione. Il metodo è relativamente semplice da implementare e permette una rappresentazione intuitiva chiara. Il metodo diretto ha due varianti:

- La collocazione diretta (direct collocation) utilizza punti di collocazione, o nodi, in cui le equazioni del moto vengono soddisfatte.
- Il metodo di Shooting lavora sulle condizioni iniziali e finali al fine di ottimizzarle per determinare la traiettoria migliore tra tutte le alternative.

Il metodo indiretto parte dal Principio del Minimo di Pontryagin, dal nome del matematico di origine russa da cui fu formulato. Per prima cosa vengono identificate le condizioni senza le quali i vincoli per lo stato e/o i controlli non sarebbero ottenibili. Nel momento in cui il principio è soddisfatto allora produce una condizione necessaria, ma non sufficiente, per dimostrare che la traiettoria scelta sia ottima.

Nella pratica il principio vuole massimizzare o minimizzare un obiettivo, che è costituito da una funzione detta la Hamiltoniana.

La formulazione del problema di costo ottimale che considera il problema di minimizzare una funzione obiettivo di costo definita come:

$$J = \int_{t_0}^{t_f} L(x(t), u(t), t) dt + \Phi(x(t_f))$$

Dove:

- $x(t) \in R^n$ è lo stato del sistema al tempo t ,
- $u(t) \in R^m$ è il controllo,
- $L(x, u, t)$ è il costo istantaneo, e
- $\Phi(x(t_f))$ è il costo finale, funzione dello stato finale $x(t_f)$

Un set di equazioni differenziali rappresenta la dinamica tramite un sistema con tanto di condizioni al contorno sia iniziali che finali.

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

Definito il vettore dei co-stati $\lambda(t) \in R^n$ si ha che sono soddisfatte per il controllo ottimale $u^*(t)$:

- L'equazione di stato: $\dot{x}(t) = \frac{\partial H}{\partial \lambda} = f(x(t), u(t), t)$
- Equazione dei co-stati: $\dot{\lambda}(t) = -\frac{\partial H}{\partial x}$
- Condizioni al bordo: $x(t_0) = x_0, x(t_f) = x_f$

Inoltre, il controllo ottimale deve minimizzare l'Hamiltoniana:

$$\min_u H(x(t), u, \lambda(t), t)$$

dove $H(x(t), u, \lambda(t), t) = L(x, u, t) + \lambda^T f(x, u, t)$

In questa tesi è stato scelto di realizzare la discesa sulla luna tramite il metodo diretto in quanto è più semplice e intuitivo benché possa essere più costoso a livello computazionale se il problema è complesso. D'altro canto, l'utilizzo di un metodo indiretto avrebbe permesso di accorciare i tempi di calcolo ma dall'altra parte sarebbe stato più complesso da implementare. Quindi, benché fornisca dei risultati più precisi, al contempo necessita della conoscenza approfondita della dinamica del sistema, dinamica che potrebbe essere trattata in modo più generale e meno dettagliato con un metodo diretto.

La scelta quindi di una trattazione tramite un metodo o l'altro è funzione delle esigenze del problema ma anche della capacità di calcolo di cui si è in possesso.

Algoritmi di ottimizzazione lineare vincolata

Si definisce il problema di minimizzazione vincolata come la ricerca del vettore x che è il minimo locale per una funzione scalare $f(x)$ soggetta ai vincoli.

$$\begin{aligned} \min_x f(x): & A * x \leq b, A_{eq} * x = B_{eq} \text{ (Vincoli lineari)} \\ & c(x) \leq 0, c_{eq}(x) = 0 \text{ (vincoli non lineari)} \\ & LB \leq X \leq UB \text{ (limiti)} \end{aligned}$$

Tra le diverse opzioni di ottimizzazione fornite da Matlab tramite la funzione `fmincon`, è possibile scegliere diverse opzioni di ottimizzazione tramite una specifica dell'algoritmo che si intende utilizzare.

La scelta dell'algoritmo di ottimizzazione può ricadere su:

- 'interior-point' (default)
- 'trust-region-reflective'
- 'sqp'
- 'sqp-legacy'
- 'active-set'

Si utilizza 'optimoptions' per selezionare l'opzione sull'algoritmo che si preferisce nella riga di comando. Per avere informazioni per eseguire una scelta opportuna dell'algoritmo esistono alcune raccomandazioni. La scelta in questo caso è ricaduta su 'sqp' in quanto capace di eseguire l'ottimizzazione con una maggiore velocità su problemi di piccole e medie dimensioni, rispetto all'algoritmo di default 'interior point'. 'Interior-point' supporta sia problemi di notevoli dimensioni, pure sparsi, sia problemi piccoli e densi. Inoltre, similmente a 'sqp', soddisfa i vincoli ad ogni iterazione e gestisce bene risultati NaN o Inf. 'Sqp' è simile, ma meno scalabile. La scelta non è ricaduta su 'sqp-legacy' in quanto è più lento e richiede più memoria. 'Active-set' è veloce per alcuni vincoli non regolari ma non è adatto su problemi di larga scala. Per finire, 'Trust-region-reflective' risulta efficiente per problemi specifici ma richiede gradienti e supporta solo vincoli semplici, lineari di uguaglianza.

Per la minimizzazione di funzioni non lineari il primo metodo è il metodo della regione di fiducia. Molti tra i metodi che sono inclusi nei solutori dell'Optimization Toolbox di Matlab hanno le proprie basi proprio sul concetto di regioni di fiducia. Si consideri un problema di minimizzazione non vincolata della funzione $f(x)$. Supponendo di essere in un punto x dello spazio e di voler trovare il punto dello spazio x per cui la funzione assume un valore inferiore. L'idea di base consiste nell'approssimazione della funzione $f(x)$ con una funzione q , più semplice, che rispecchi il comportamento di $f(x)$ in un intorno N di x . Tale intorno viene definito 'regione di fiducia'. Il passo s di tentativo è calcolato minimizzando la funzione nell'intorno N . Quindi il problema viene convertito in un sottoproblema nella regione di fiducia:

$$\min_s \{q(s), s \in N\}$$

Il punto corrente viene aggiornato ad essere:

$$x \rightarrow x + s$$

$$\text{se } f(x + s) < f(x)$$

Viceversa, il punto corrente rimane invariato e la regione di fiducia N viene ristretta, con conseguente ricalcolo del passo di tentativo s .

Sequential Quadratic Programming

La programmazione quadratica sequenziale fa parte degli algoritmi in grado di risolvere dei problemi di ottimizzazione non lineare. Questo algoritmo in particolare gestisce qualsiasi linearità, inclusa quella dei vincoli. SQP nacque nel 1963 e assembla il metodo Active Set con il metodo di Newton.

La formulazione del problema è la seguente:

$$\min_{x \in \mathbb{R}^n} f(x) \mid g(x) = 0 \text{ e } h(x) \leq 0$$

Di questa formulazione $f(x)$, $h(x)$ e $g(x)$ sono potenzialmente non lineari. La variabile x è un vettore di molte variabili per l'ottimizzazione, in quel caso $h(x)$ e $g(x)$ sono sistemi.

La programmazione quadratica sequenziale, per essere compresa al meglio, necessita la conoscenza di metodi alla base come il metodo dei moltiplicatori di Lagrange e le condizioni Karush, Kuhn e Tucker (KKT).

La funzione Lagrangiana è molto utile perché combina tutte le informazioni del problema in un'unica funzione. In tale funzione vengono utilizzati i moltiplicatori di Lagrange λ , per i vincoli di uguaglianza, e μ , per i vincoli di disuguaglianza. Dunque, l'inserimento di vincoli di uguaglianza o disuguaglianza in problemi di ottimizzazione viene risolto inserendo ulteriori gradi di libertà chiamati moltiplicatori di Lagrange. La risoluzione di una ottimizzazione vincolata avviene quindi con la ricerca di punti di stazionarietà della funzione lagrangiana, che coinvolge questi moltiplicatori di Lagrange aggiuntivi. Verrà approfondita la risoluzione numerica [27]. I sistemi di questa tipologia vengono trattati con il metodo di Newton, e il quadro che si genera viene chiamato 'programmazione quadratica sequenziale'.

Si considera la minimizzazione con vincolo di uguaglianza, omettendo per semplicità i vincoli di disuguaglianza:

$$\min_{x \in \mathbb{R}^n} f(x) \mid g(x) = 0$$

In cui $f: \mathbb{R}^n \rightarrow \mathbb{R}$, mentre $g: \mathbb{R}^n \rightarrow \mathbb{R}^m$

È stato analizzato che, sarebbe possibile risolvere matematicamente questo problema con una lagrangiana che è composta da una funzione di x e un vettore λ m -dimensionale dei moltiplicatori lagrangiani.

Si definisce quindi la lagrangiana:

$$L(x, \lambda) = f(x) + \lambda^T * g(x)$$

Con la condizione necessaria che il gradiente della lagrangiana sia nullo:

$$\nabla L(x, \lambda) = 0$$

Quindi basta una singola funzione che può essere ottimizzata, al fine di trovare i punti critici dove il gradiente della funzione è nullo. Questa procedura include ora λ , per i vincoli di uguaglianza, e μ , per i vincoli di disuguaglianza ora assenti. Il gradiente è scomponibile in due parti, se guardiamo rispettivamente il gradiente di L rispetto a x e poi rispetto a λ :

$$\nabla L(x, \lambda) = \begin{bmatrix} \nabla f(x) + J_g^T(x)\lambda \\ g(x) \end{bmatrix} = 0$$

Al sistema formato da questo gradiente viene dato l'appellativo di condizioni KKT. Compare in questa scrittura sia il gradiente di f che il Jacobiano, trasposto, della funzione g . Per risolvere questo sistema di due equazioni è necessario avere entrambe le componenti della matrice nulle.

La seconda condizione KKT è semplicemente una condizione di fattibilità, $h(x)$ erano vincolati a zero nella formulazione del problema.

La terza condizione KKT è complicata in quanto solamente l'insieme dei vincoli di disuguaglianza attiva deve soddisfare questa uguaglianza. L'insieme dei vincoli di disuguaglianza attiva viene indicato con h^* . Vengono definiti i vincoli di disuguaglianza attiva coloro che partecipano attivamente alla determinazione della soluzione ottimale. Tali vincoli saranno al limite dello zero. I vincoli di disuguaglianza che non sono neanche lontanamente vicini alla soluzione vengono invece definiti irrilevanti, e non fanno parte degli Active set.

Quindi, i moltiplicatori lagrangiani costituiscono la variazione della funzione obiettivo rispetto ad una variazione del vincolo. I vincoli inattivi possono essere rimossi dalla lagrangiana prima ancora che ne venga calcolato il gradiente.

Per risolvere le condizioni KKT si utilizza il metodo Active Set, dell'insieme dei vincoli attivi, utilizzando guess e verifiche per trovare i punti critici. Generalmente, si parte dalla ipotesi che ogni vincolo di disuguaglianza è inattivo. Una volta che si è risolto il sistema rimanente per x , va verificata la fattibilità. Nel caso alcuni vincoli fossero violati, questi vanno considerati attivi nella successiva iterazione. Inoltre, se qualche moltiplicatore risultasse negativo, i suoi vincoli andrebbero considerati inattivi nella successiva iterazione. Solo i problemi quadratici sono affrontabili con il metodo degli insiemi attivi perché le condizioni KKT sono lineari. La programmazione quadratica sequenziale supera questa limitazione incorporando il metodo di Newton per gestire funzioni altamente non lineari.

Il metodo di Newton si basa sull'idea principale di migliorare un'ipotesi in proporzione alla velocità con cui la funzione cambia all'ipotesi, e in modo inversamente proporzionale a come la funzione accelera al cambiare dell'ipotesi. Per capire intuitivamente il concetto, una funzione con un'inclinazione rapida non sarà vicina ad un punto critico e il miglioramento dell'ipotesi dovrebbe essere ampio. D'altro canto, una funzione con poca pendenza che tende a esaurirsi, è probabile che sia vicino ad un punto critico e quindi l'ipotesi deve essere migliorata di poco. Le iterazioni convergono al valore critico di qualsiasi funzione con step di miglioramenti che seguono la seguente formulazione:

$$x_{k+1} = x_k - \frac{\nabla f}{\nabla^2 f}$$

Il segno meno ha significato un importante: in prossimità del minimo, un gradiente positivo dovrebbe diminuire l'ipotesi o aumentarla, e la divergenza invece ha segno positivo. Se invece ci si trovasse vicino ad un massimo, un gradiente positivo dovrebbe, analogamente a prima, aumentare o meno l'ipotesi,

ma la divergenza è negativa. Questa considerazione è fondamentale nei problemi non convessi con diversi punti di massimo e minimo locale; in questi casi il metodo di Newton troverà il punto critico più vicino all'ipotesi di partenza. L'inserimento del metodo di Newton all'interno del metodo degli insiemi attivi trasforma il problema in forma matriciale.

Va notato che questo sistema di due equazioni è non lineare e quindi si può approssimare con il metodo di Newton.

Per utilizzare il metodo di Newton è necessario derivare il Jacobiano del gradiente di L .

Pensando al gradiente della Lagrangiana come un vettore di dimensione $(n + m)$, dove le prime n componenti corrispondono alle coordinate x , mentre le ultime m componenti corrispondono alle coordinate λ .

$$\nabla L(x, \lambda) = \begin{bmatrix} \nabla f(x) + \sum_{k=1}^m \lambda_k \nabla g_k(x) \\ g(x) \end{bmatrix} \in \mathbb{R}^{n+m}$$

Dovendo differenziare il gradiente della Lagrangiana sia rispetto a $x \in \mathbb{R}^n$ che a $\lambda \in \mathbb{R}^m$, possiamo gestire la derivazione suddividendola in due cicli su due indici i e j .

Per $i = 1, \dots, n$:

$$(\nabla L(x, \lambda))_i = \frac{\partial f(x)}{\partial x_i} + \sum_{k=1}^m \lambda_k \frac{\partial g_k(x)}{\partial x_i}$$

Derivando rispetto a x_j , per $i, j = 1, \dots, n$, si ottiene:

$$\frac{\partial}{\partial x_i} (\nabla L(x, \lambda))_i = \frac{\partial^2 f(x)}{\partial x_i \partial x_j} + \sum_{k=1}^m \lambda_k \frac{\partial^2 g_k(x)}{\partial x_i \partial x_j}$$

Dunque, il blocco $(n \times n)$ in alto a sinistra del Jacobiano di $\nabla L(x, \lambda)$ è:

$$B(x, \lambda) \equiv H_f(x) + \sum_{k=1}^m \lambda_k H_{g_k}(x)$$

Derivando $(\nabla L(x, \lambda))_i$ rispetto a λ_j , per $i = 1, \dots, n$, e $j = 1, \dots, m$, si ottiene:

$$\frac{\partial}{\partial \lambda_j} (\nabla L(x, \lambda))_i = \frac{\partial g_j(x)}{\partial x_i}$$

Dunque, il blocco $(n \times m)$ in alto a destra del Jacobiano di $\nabla L(x, \lambda)$ è:

$$J_g^T(x) \in \mathbb{R}^{n \times m}$$

Soffermandosi ora su $i = n + 1, \dots, n + m$, si ha

$$(\nabla L(x, \lambda))_i = g_i(x)$$

Derivando $(\nabla L(x, \lambda))_i$ rispetto a x_j , per $i = n + 1, \dots, n + m$, $j = 1, \dots, n$, si ottiene:

$$\frac{\partial}{\partial x_j} (\nabla L(x, \lambda))_i = \frac{\partial g_i(x)}{\partial x_j}$$

Quindi il blocco $(m \times n)$ in basso a sinistra del Jacobiano di $\nabla L(x, \lambda)$ è:

$$J_g(x) \in \mathbb{R}^{m \times n}$$

E, infine, il blocco $(m \times m)$ in basso a destra è semplicemente zero, dalla derivazione di $g_i(x)$ rispetto a λ_j .

Dunque, la matrice che si ottiene derivando $\nabla L(x, \lambda)$ è la seguente matrice jacobiana:

$$\begin{bmatrix} B(x, \lambda) & J_g^T(x) \\ J_g(x) & 0 \end{bmatrix} \in R^{(n+m) \times (n+m)}$$

Soffermandoci sulle dimensioni si ottiene una matrice quadrata:

$$\begin{bmatrix} (n \times n) & (n \times m) \\ (m \times n) & (m \times m) \end{bmatrix} \in R^{(n+m) \times (n+m)}$$

La struttura a blocchi 2x2 della matrice la rende una tra le matrici chiamate KKT. Le matrici con questa struttura sono state studiate da Karush, Kuhn e Tucker, i quali hanno fatto diversi lavori sulla ottimizzazione non lineare.

Applicando il metodo di Newton per risolvere il gradiente di $\nabla L(x, \lambda) = 0$, allora si procede con i passaggi seguenti:

$$\nabla L(x, \lambda) = \begin{bmatrix} B(x, \lambda) & J_g^T(x) \\ J_g(x) & 0 \end{bmatrix} \begin{bmatrix} s_k \\ \delta_k \end{bmatrix} = - \begin{bmatrix} \nabla f(x_k) + J_g^T(x_k) \lambda_k \\ g(x_k) \end{bmatrix} = 0$$

$$k = 0, 1, 2, \dots$$

Compaiono s_k e δ_k , di dimensione n e m rispettivamente.

Una volta trovati s_k e δ_k , è possibile usarli per aggiornare i valori di x_k e λ_k rispettivamente.

Qui $(s_k, \delta_k) \in R^{(n+m)}$ rappresenta il k -esimo passo di Newton.

Facendo un passo indietro, ora viene presentato un problema di minimizzazione vincolata con una forma specifica dove (x_k, λ_k) rappresentano lo step iterativo k , mentre s rappresenta un vettore di dimensione n .

L'equazione da minimizzare viene riportata di seguito ed è funzione di s :

$$\min_s \left\{ \frac{1}{2} s^T B(x_k, \lambda_k) s + s^T (\nabla f(x_k) + J_g^T(x_k) \lambda_k) \right\}$$

$$\text{tale che } J_g(x_k) s + g(x_k) = 0$$

Questa funzione obiettivo è quadratica in s e i vincoli sono lineari in s , in quanto si considerano x_k, λ_k sono costanti.

Quindi si potrebbe risolvere questo sistema scrivendo un lagrangiano $L_k(s, \delta)$, dove δ sono i moltiplicatori lagrangiani necessari per imporre i vincoli lineari.

Il problema di minimizzazione ha un Lagrangiano:

$$L_k(s, \delta) \equiv \frac{1}{2} s^T B(x_k, \lambda_k) s + s^T (\nabla f(x_k) + J_g^T(x_k) \lambda_k) + \delta^T (J_g(x_k) s + g(x_k))$$

Si procede calcolando il gradiente di $L_k(s, \delta)$ e imponendolo uguale a zero. Se si fa questo calcolo si scopre che si ottiene lo stesso sistema da risolvere al metodo di Newton al passaggio k -esimo. Questo ci fornisce una particolare informazione, ovvero che in ogni passaggio del metodo di Newton si sta risolvendo un problema di minimizzazione con una funzione obiettivo quadratica e vincoli lineari. Un problema di questo tipo viene definito un programma quadratico e questo motiva il nome SQP (programma quadratico sequenziale) in cui il metodo di Newton viene usato per risolvere il gradiente $\nabla L_k(s, \delta) = 0$.

Risolvere $\nabla L_k(s, \delta) = 0$ (condizione necessaria di primo ordine) fornisce un sistema lineare, esattamente lo stesso del k -esimo step di Newton. Quindi ad

ogni passo nel metodo di Newton si risolve un problema di minimizzazione, con obiettivo quadratico e vincoli lineari.

Il SQP è un metodo molto importante e ci sono molti aspetti da considerare per ottenere una implementazione che sia efficiente ed affidabile. Si possono risolvere sistemi lineari con il metodo di Newton in modo efficiente, sfruttando anche la struttura a blocchi 2x2. È possibile usare approssimazioni quasi-newton per la Hessiana oppure usare regioni di fiducia, ricerca della linea e voler aumentare la robustezza. Per garantire una implementazione corretta è opportuno scegliere attentamente il trattamento dei vincoli, sia di uguaglianza che disuguaglianza, durante il processo iterativo. Inoltre, potremmo affinare la selezione di buone ipotesi di partenza per i moltiplicatori di Lagrange λ .

La forza dell'algorithm sta nel fatto che i minimi della funzione obiettivo saranno gli stessi minimi della lagrangiana, in quanto la lagrangiana è uguale alla funzione obiettivo in un punto che soddisfa le condizioni che KKT. Poiché i moltiplicatori lagrangiani sono variabili aggiuntive, l'iterazione forma un sistema;

$$\begin{bmatrix} x_{k+1} \\ \lambda_{k+1} \\ \mu_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ \lambda_k \\ \mu_k \end{bmatrix} - \frac{\nabla L_k}{\nabla^2 L_k}$$

Dove, nel caso in cui siano presenti anche vincoli di disuguaglianza, il gradiente è:

$$\nabla L = \begin{bmatrix} \frac{\partial L}{\partial x} \\ \frac{\partial L}{\partial \lambda} \\ \frac{\partial L}{\partial \mu} \end{bmatrix} = \begin{bmatrix} \nabla f + \lambda \Delta g + \mu \nabla h^* \\ g \\ h^* \end{bmatrix}$$

Mentre l'Hessiana è:

$$\nabla^2 L = \begin{bmatrix} \nabla_{xx}^2 L & \nabla g & \nabla h \\ \nabla^T g & 0 & 0 \\ \nabla h^T & 0 & 0 \end{bmatrix}$$

Il passo di Newton $d_k = \frac{\nabla L_k}{\nabla^2 L_k}$ non può essere calcolato direttamente perché l'hessiana non è invertibile. Invece l'algorithm di programmazione quadratica sequenziale di base definisce una direzione di ricerca appropriata d_k , come soluzione del sottoproblema QP.

Il solutore PNL in MATLAB utilizza la formulazione mostrata di seguito; la minimizzazione vincolata è il problema di trovare un vettore x che sia un minimo locale per una funzione scalare $f(x)$ soggetta a vincoli sull'ammissibile x :

$$\begin{aligned} \min_{x \in \mathbb{R}^n} f(x) \quad & | \quad c(x) \leq 0, c_{eq}(x) = 0, \\ & A \cdot x \leq b, \\ & A_{eq} \cdot x = b_{eq}, \\ & lb \leq x \leq ub. \end{aligned}$$

Dove $f(x)$ è la funzione obiettivo da minimizzare con le variabili di progetto, lb e ub sono rispettivamente limite inferiore e il limite superiore. A_{eq} e b_{eq} sono le matrici che specificano i vincoli di uguaglianza, mentre A e b sono le matrici per i vincoli di disuguaglianza.

In Matlab, i vincoli non lineari sono codificati in una funzione separata con $c(x)$ come funzione di valori vettoriali con tutti i vincoli non lineari di disuguaglianza. $c_{eq}(x)$ è una funzione a valori vettoriali con tutti i vincoli non lineari di uguaglianza. Essi soddisfano le condizioni come mostrato di seguito:

$$c(x) \leq 0, c_{eq}(x) = 0$$

I problemi PNL sono iterativi, il che implica che inizino con una ipotesi iniziale come punto di partenza x_0 .

Initial guess: x_0

Il risolutore procede iterativamente dal punto di partenza seguendo il gradiente della funzione obiettivo e i vincoli per raggiungere il punto in cui il gradiente è uguale a zero.

Quindi, grazie a SQP, problemi di ottimizzazione vincolata possono essere riformulati come problemi di ottimizzazione non vincolata e rappresentati con la condizione di Karush-Kuhn-Tucker o KKT.

- La condizione necessaria del primo ordine che deve essere soddisfatta del minimo locale è che $\nabla f(x^*) = 0$.
- La condizione necessaria del secondo ordine che deve essere soddisfatta del minimo locale è che $\nabla^2 f(x^*)$ deve essere semidefinita positiva.

Tuttavia, per garantire che il punto ottimo ottenuto sia il minimo locale, deve venire soddisfatta anche la condizione sufficiente aggiuntiva del secondo ordine.

- La condizione sufficiente aggiuntiva del secondo ordine per il minimo locale è che $\nabla^2 f(x^*)$ deve essere definita positiva. Essendo $\nabla^2 f(x^*)$ coincidente con la matrice Hessiana, questo implica che la matrice Hessiana deve essere definita positiva.

L'interpretazione geometrica delle condizioni di ottimali del primo ordine.

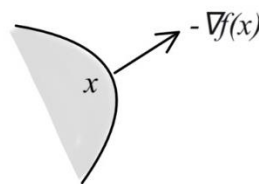


Figura 5: Interpretazione geometrica delle condizioni ottimali.

Alla base dell'algoritmo SQP c'è il metodo di Newton implementato per risolvere questo problema di ottimizzazione vincolata. In questo metodo, la funzione obiettivo viene approssimata da una funzione quadratica a ogni iterazione e questa approssimazione viene quindi ridotta al minimo e viene calcolata una direzione di discesa. L'approssimazione quadratica della funzione è data dalla formula seguente.

Infatti, precedentemente era stato chiarito che SQP definisce la direzione di ricerca appropriata $d_k = x - x_k$ alla iterazione k , dalla soluzione del sottoproblema QP:

$$\begin{aligned} f(x) &= f(x_k) + \nabla f(x_k)^T (x - x_k) + \frac{1}{2} (x - x_k)^T \nabla^2 L(x_k) (x - x_k) \\ h(x_k) + \nabla h(x_k)^T (x - x_k) &\geq 0 \\ g(x_k) + \nabla g(x_k)^T (x - x_k) &= 0 \end{aligned}$$

L'algoritmo SQP inizia scegliendo prima l'iterazione iniziale x_0 , λ_0 e μ_0 . Calcolando $\nabla^2 L(x_0, \lambda_0, \mu_0)$ e $\nabla L(x_0, \lambda_0, \mu_0)$, il sottoproblema QP risolve e trova il passo iniziale d_0 . Il passo iniziale viene utilizzato per aggiornare l'iterazione:

$$x_{k+1} = x_k - d_k$$

In particolare, dalla condizione necessaria del primo ordine $\nabla f(x) = 0$, si arriva a:

$$\nabla f(x) = \nabla f(x_k) + \nabla^2 f(x_k)(x - x_k) = 0$$

Che una volta risolta fornisce il valore successivo delle variabili di progettazione nell'iterazione successiva, applicando il passaggio d_k .

$$-\nabla f(x_k) = \nabla^2 f(x_k)(x - x_k)$$

$$x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$$

Dove $[\nabla^2 f(x_k)]^{-1}$ è la inversa della Hessiana e $d_k = [\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$.

Il d_k è il passo nella direzione di discesa, calcolabile come scritto sopra.

La visualizzazione del gradiente della funzione obiettivo, essendo antiparallelo al gradiente della funzione vincolo, è mostrata di seguito. Viene anche indicata la regione di ammissibilità.

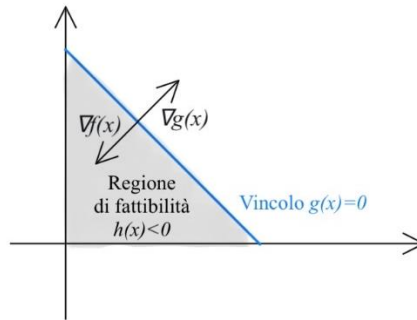


Figura 6: Gradienti e regioni di fattibilità

Per un problema con vincoli di disuguaglianza le condizioni di KKT verrebbero modificate per includere i vincoli di disuguaglianza nel problema di minimizzazione.

$$\min_{x \in \mathbb{R}^n} f(x) \mid h_i(x) \leq 0, i = 1, \dots, n$$

Ciò richiederebbe l'introduzione della condizione complementare di margine di flessibilità e delle condizioni di fattibilità duale.

- condizione KKT di fattibilità duale:

$$\lambda_i \geq 0 \text{ per ogni } i = 1, \dots, n$$

- condizione KKT complementare di margine di flessibilità

$$\lambda_i h(x^*) \geq 0 \text{ per ogni } i = 1, \dots, n$$

I vincoli di disuguaglianza possono essere convertiti in vincoli di uguaglianza tramite le variabili di margine (slack variables) s_1 e s_2 . Queste variabili sono definite positive, $s \geq 0$ e trasformano la disuguaglianza $Ax \leq b$ in una uguaglianza $Ax + s = b$. Se il problema ha un vincolo di disuguaglianza, potenzialmente non lineare $h(x) = x_1^2 + x_2 - 5 \leq 0$ allora il vincolo vuol dire che il valore della funzione può essere al massimo 0. Per semplificare viene introdotto una variabile slack, che rappresenta quanto manchi perché la disuguaglianza si trasformi in una uguaglianza. In pratica $h(x) + s = 0$ con la condizione $s \geq 0$.

Ad esempio, nel caso precedente, si può riscrivere il vincolo come

$$x_1^2 + x_2 - 5 + s = 0 \text{ con } s \geq 0$$

‘s’ quindi rappresenta il margine o scarto.

Se $h(x)$ è esattamente nulla, allora $s=0$, se $h(x)$ è negativa allora s la compensa per riportare tutto a 0.

Il problema a questo punto è risolvibile con il metodo dei moltiplicatori di Lagrange.

Per fare un altro esempio di implementazione si riportano le due equazioni seguenti, che rappresentano i vincoli del problema nella forma $h(x) + s = 0$:

$$\begin{aligned} x_2 - 2x_1 &= -s_1^2 \\ 1 - x_1 - x_2 &= -s_2^2 \end{aligned}$$

Se $s_1 = 0$ allora il primo vincolo è attivo, ovvero è soddisfatto, con uguaglianza $x_2 - 2x_1 = 0$. Se $s_1 > 0$, il primo vincolo è inattivo e $x_2 - 2x_1 < 0$. Ragionamento analogo vale per s_2 e il secondo vincolo.

Allora la Lagrangiana, che rinchiede sia l’obiettivo che i vincoli del problema, assume questa forma con i moltiplicatori KKT λ_i .

$$L = x_1^2 - x_2 + \lambda_1(x_2 - 2x_1 + s_1^2) + \lambda_2(1 - x_1 - x_2 + s_2^2)$$

Per trovare i minimi si deriva la lagrangiana rispetto a tutte le variabili. Le derivate parziali della lagrangiana rispetto a ciascuna variabili vanno settate a zero:

$$\begin{aligned} \nabla_{x_1} L = 0 &= 2x_1 - 2\lambda_1 - \lambda_2 \\ \nabla_{x_2} L = 0 &= -1 + \lambda_1 - \lambda_2 \\ \nabla_{\lambda_1} L = 0 &= x_2 - 2x_1 + s_1^2 \\ \nabla_{\lambda_2} L = 0 &= 1 - x_1 + s_2^2 \\ \nabla_{s_1} L = 0 &= \lambda_1 s_1 \\ \nabla_{s_2} L = 0 &= \lambda_2 s_2 \\ \lambda_1 &\geq 0, \lambda_2 \geq 0 \end{aligned}$$

L’ultima equazione, va aggiunta al sistema delle derivate, è la condizione complementare di margine di flessibilità. La condizione di complementarità $\lambda_i * s_i = 0$ significa che:

- Se $s_i > 0$ allora $\lambda_i = 0$ e il vincolo associato è inattivo.
- Se $\lambda_i > 0$ allora $s_i = 0$, il vincolo associato è attivo.

È una proprietà dei moltiplicatori di Lagrange nelle condizioni di KKT.

Poiché ci sono due vincoli, ogni combinazione delle slack e moltiplicatori fornisce un caso possibile. Va verificato ogni caso per vedere quale soddisfa le condizioni.

Da una analisi sulle equazioni emerge che λ è zero, o s è zero. Se s_1 è zero, allora il vincolo 1 è attivo; se λ_1 è zero, allora il vincolo 1 è inattivo. Questo implica quattro possibilità che è necessario testare una per volta. Viene riportato un esempio di come appaiono i vincoli attivi e inattivi.

Sulla sinistra appare il vincolo attivo, mentre sulla destra il vincolo inattivo.

- Caso 1: $\lambda_1 = \lambda_2 = 0$, l'equazione 2 è violata quindi non è possibile.
- Caso 2: $s_1 = s_2 = 0$, entrambi i vincoli sono attivi, il moltiplicatore di Lagrange è negativo, ovvero viene violata la condizione KKT di fattibilità duale.
- Caso 3: $\lambda_1 = s_2 = 0$, il vincolo uno è inattivo perché $s_1 > 0$, e il vincolo 2 è attivo $s_2 = 0$ e il moltiplicatore di Lagrange è negativo, ovvero viene violata la condizione KKT di fattibilità duale.
- Caso 4: $\lambda_2 = s_1 = 0$, la soluzione è valida, il vincolo 1 è attivo mentre il vincolo 2 è inattivo.

In questo caso si ottiene: $\lambda_1 = 1, s_2^2 = 2, x_1 = 1, x_2 = 2$;

Questo indica che il vincolo 1 è attivo e il vincolo 2 è inattivo, dando la soluzione ottimale. Le variabili $x_1 = 1, x_2 = 2$ minimizzano la funzione obiettivo rispettando i vincoli.

$s_1 = 0$, il primo vincolo è attivo
 $\lambda_1 = 1$, il primo vincolo è contribuisce alla soluzione
 $s_2^2 = 2, s_2 > 0$, il secondo vincolo è inattivo

L'immagine aiuta a visualizzare la differenza tra vincolo attivo (soddisfatto con uguaglianza) e vincolo inattivo (non vincola realmente la soluzione).

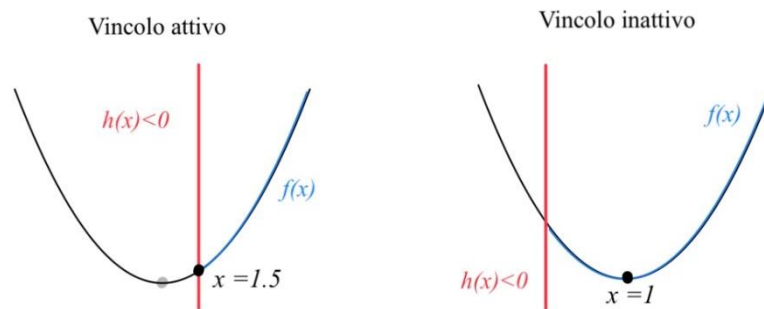


Figura 7: Vincoli attivi e inattivi.

IL MODELLO DELLA DISCESA

Il problema della discesa sulla luna è cercare di realizzare quello che è noto con il termine inglese “soft-landing”, ovvero un atterraggio, possibilmente autonomo, che permetta di toccare il suolo in modo morbido e sicuro, evitando schianti o compromettendo la missione.

La principale sfida nel successo di un atterraggio morbido è legata al fatto che non ci sia atmosfera attorno alla luna. Di conseguenza ricorrere a soluzioni come la frenata aerodinamica oppure il dispiegamento di un paracadute per rallentare la velocità di discesa risulta impossibile. L'unico metodo possibile per diminuire il rateo di discesa è frenare con il propulsore. In particolare, modificare il vettore della spinta, in modulo e direzione, è la soluzione che si adotta. Le missioni odierne, essendo molto complesse, richiedono soluzioni creative e una di queste è l'atterraggio autonomo e sicuro, che possa essere eseguito con poca attrezzatura e potenza di elaborazione a bordo, e al contempo sia efficiente in termini di consumi del carburante.

A livello qualitativo il problema può essere descritto come segue. Immaginando di aver percorso una trasferta di Hohmann, da una orbita terrestre a una orbita lunare, si vuole ora definire la traiettoria di discesa ottimizzata in modo da atterrare con minimi consumi. La velocità verticale al touchdown deve essere di circa 5m/s per garantire la sicurezza della attrezzatura scientifica e garantire un atterraggio delicato alla sonda. In genere viene utilizzata una orbita di parcheggio lunare da cui iniziare il processo di atterraggio. Il modulo di comando e servizio (CSM) e il lander si separano in modo sicuro nell'orbita di parcheggio (ad un'altezza di circa 100 km).

La velocità orbitale attorno alla luna deve essere neutralizzata da motori opportuni o propulsori adatti. Si prende come riferimento per l'inizio della discesa, una distanza approssimativa di 15 chilometri dalla superficie lunare. A partire da quella quota si avvia la procedura chiamata ‘frenatura propulsa’, in quanto sulla luna manca la atmosfera. È essenziale che la riduzione della velocità sia ottimale per i consumi al fine di massimizzare la massa, e quindi il payload, della sonda. A tale scopo la direzione della spinta deve essere modulata in modo opportuno. Il problema viene formulato con un problema di ottimizzazione del controllo con l'angolo di direzione della spinta come variabile di controllo, producendo questi percorsi di atterraggio morbidi ottimali per il carburante.

Dato il crescente interesse nella esplorazione dei pianeti del sistema solare, l'atterraggio morbido planetario è uno dei problemi nella teoria del controllo ottimale. I voli futuri andranno a concentrarsi principalmente sulla

individuazione di atterraggi morbidi, noti come atterraggi di precisione o atterraggi puntuali, su siti scientificamente interessanti. Si vuole rendere la procedura di atterraggio autonoma in modo che sia ripetibile e sicura per viaggi successivi.

L'atterraggio morbido, noto anche come stadio di discesa motorizzata, è l'ultimo stadio di un EDL planetario (EDL: entry, descent, and landing). Quello che viene preso di riferimento è lo schema di atterraggio del programma Apollo, che, infatti, parte da 15km e discende in modo convenzionale fino a 2.2km, in figura 8. In questo lavoro di tesi inizialmente verrà svolta una analisi semplificata che tratterà la traiettoria di discesa con una unica ottimizzazione fino alla superficie lunare, ovvero a quota zero, allontanandosi dunque dalla reale discesa di progetto di Apollo. Per terminare seguono poi le fasi di approccio e atterraggio.

L'obiettivo della trasferta di Hohmann è quello di abbassare efficacemente l'altitudine usando quasi una potenza nulla nel tratto dal 100 km ai 15 km (perilunio) in modo da essere pronti per l'inizio della discesa propulsa (PDI: Power descent initiation). Per raggiungere i requisiti operazionali, la progettazione della traiettoria per la discesa propulsa è in tre fasi: la fase di frenata, la fase di approccio e la fase di atterraggio. Seguendo la PDI, che inizia all'altitudine del perilunio, si ha la fase di frenata che è in gran parte destinata a rallentare la velocità dell'orbita. Durante la fase di frenata la altitudine passa da 15km a 2.2km. La fase di approccio serve a dare al pilota una vista della regione di atterraggio, consentendo di tenere d'occhio la superficie lunare d'approccio. La fase di approccio si colloca all'altitudine da 2.2km a 152m. Ultima, ma non per importanza, la fase di atterraggio, il cui obiettivo è fornire un volo a bassa velocità con un assetto perpendicolare alla superficie lunare. La transizione tra le tre fasi deve avvenire in modo morbido, da un angolo di assetto al successivo.

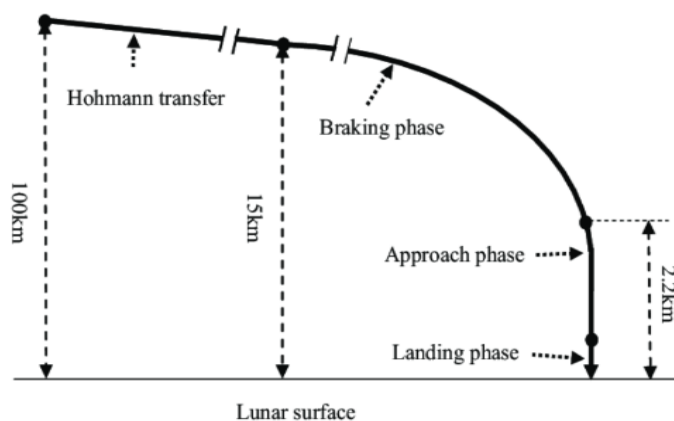


Figura 8: Fase di atterraggio lunare per la discesa di Apollo

Definizione del modello e ipotesi

Consideriamo uno spacecraft in avvicinamento alla luna. Si immagina conclusa la Hohmann in quanto la fase di interesse è quella di avvicinamento a partire dai 15km. Per definire una corretta traiettoria di discesa, è necessario spostare il veicolo da una posizione iniziale nota e da uno stato di velocità conosciuto attraverso un determinato controllo. L'obiettivo è quello di condurre l'intera manovra in modo ottimale. La definizione di ottimale è riferita ai consumi di carburante minimi. Essendo lo scopo lo sviluppo di un modello su Matlab, verranno elencate di seguito una serie di ipotesi.

- Ipotesi 1: Vengono trascurate le forze legate all'atmosfera. Tale ipotesi risulta molto solida e plausibile per i corpi celesti privi di atmosfera come la luna. Di conseguenza le forze aerodinamiche, come la portanza e la resistenza aerodinamica, non essendo generate non vengono incluse nel modello.
- Ipotesi 2: Si trascura la curvatura della luna. L'ipotesi risulta plausibile perché il modello partendo da 15km è così prossimo alla superficie lunare da rendere la curvatura della luna trascurabile.
- Ipotesi 3: L'accelerazione gravitazionale della luna è costante nel campo in considerazione. Questa semplificazione ha senso in quanto la distanza radiale dal centro della luna, durante la discesa, cambia marginalmente. In altre parole, se la traiettoria da ottimizzare è solamente la fase di atterraggio lunare o planetario, le distanze come il downrange e l'altitudine sono molto inferiori se paragonate al raggio del satellite.
- Ipotesi 4: Il moto rotatorio del corpo celeste può essere trascurato. L'ipotesi di accelerazione lunare costante, unita a quest'ultima semplificazione, permette di modellare l'oggetto planetario come piatto.
- Ipotesi 5: Si considera come propulsore un motore a razzo chimico. Il motore è in condizioni di perfetta espansione. Il veicolo, tramite il propulsore chimico, è in grado di generare una forza di spinta espellendo il propellente.
- Ipotesi 6: L'ampiezza della spinta è limitata. La motivazione di questa ipotesi è che la regolazione della spinta non conviene generalmente.
- Ipotesi 6: La direzione della spinta può variare.

Sistema di riferimento ed Equazioni

Il sistema di riferimento è un sistema di coordinate cartesiane x,y e z. L'asse z rappresenta l'altitudine e parte dall'origine del sistema di riferimento ed è diretto in verticale. L'asse x e y delimitano il piano orizzontale. Essendo assente l'atmosfera, le uniche forze che agiscono sullo spacecraft sono la forza causata dalla gravità lunare e la forza di spinta prodotta dal propulsore.

$$F = mg + T\hat{u} = m\dot{v} = m\ddot{r}$$

- $r = r_x \hat{i} + r_y \hat{j} + r_z \hat{k}$ raggio vettore
- $v = v_x \hat{i} + v_y \hat{j} + v_z \hat{k}$ vettore velocità
- $g = -g \hat{k}$ accelerazione gravitazionale
- $\hat{u} = u_x \hat{i} + u_y \hat{j} + u_z \hat{k}$ vettore di direzione di spinta unitaria

Un atterraggio planetario è caratterizzato da una dinamica veloce, nell'ordine di grandezza dei minuti; quindi, durante la manovra non dovrebbero esserci variazioni eccessive di massa.

La dinamica è descritta dal seguente insieme di equazioni.

$$\left\{ \begin{array}{l} \frac{dr_x}{dt} = v_x \\ \frac{dr_y}{dt} = v_y \\ \frac{dr_z}{dt} = v_z \\ \frac{dv_x}{dt} = \frac{T}{m} \cos \psi * \cos \phi \\ \frac{dv_y}{dt} = \frac{T}{m} \cos \psi * \sin \phi \\ \frac{dv_z}{dt} = \frac{T}{m} \sin \psi - g \\ \frac{dm}{dt} = -\frac{T}{c} \end{array} \right.$$

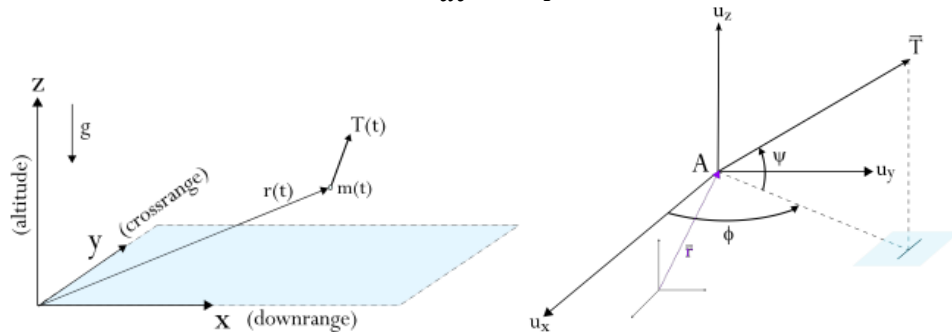


Figura 9: Sistemi di riferimento

Vengono introdotti due angoli ϕ e ψ per proiettare correttamente la direzione del vettore di spinta T lungo le sue tre componenti. In questo studio l'angolo ϕ è posto nullo, quindi si presuppone che la spinta non abbia componenti su y ma solo componenti lungo x e z .

$$\begin{aligned} T_x &= T \cos \psi * \cos \phi = T \cos \psi \\ T_y &= T \cos \psi * \sin \phi = 0 \\ T_z &= T \sin \psi \end{aligned}$$

Variabili di controllo

Le variabili di controllo potrebbero essere molteplici. In questo codice per semplicità vengono scelti gli angoli di spinta (in particolare quello nel piano verticale) e i tempi di realizzazione della manovra. Il tempo è trattato come variabile di controllo perché è impossibile determinarlo in partenza in quanto è funzione dei controlli stessi.

La storia temporale dell'angolo di spinta durante la discesa, dallo stato iniziale allo stato finale, consiste nella storia del controllo che minimizza i consumi di carburante. Per rendere la trattazione completa, verrà aggiunta un'analisi anche con il modulo della spinta variabile, tenuto costante inizialmente.

Vincoli

In questa prima fase vengono posti come vincoli l'atterraggio sulla superficie lunare (quindi $z=0$ per t_{fin}) e il vincolo che la componente verticale della velocità all'atterraggio sia nulla ($v_z=0$ per t_{fin}) in modo che si verifichi un "soft-landing" ovvero un atterraggio morbido che eviti danneggiamenti per impatti bruschi ad alta velocità. Le velocità di atterraggio devono essere basse, come 5m/s, per avere un atterraggio morbido. Con l'aiuto di dispositivi come le borse di atterraggio è possibile atterrare in modo morbido anche a velocità più elevate (velocità di 15m/s di Luna 9).

Verranno poi aggiunti dei vincoli aggiuntivi all'inizio della fase di atterraggio verticale, in particolare $v_x=0$.

ALGORITMO E ARCHITETTURA DELLO STRUMENTO

Questo capitolo descrive un approccio di ottimizzazione per la traiettoria di un satellite in fase di avvicinamento alla Luna. Lo scopo è minimizzare il consumo di massa propellente utilizzando metodi di controllo ottimale applicati tramite programmazione non lineare, gestendo vincoli e dinamiche complesse. L'implementazione è sviluppata in MATLAB e include una funzione obiettivo, vincoli e simulazioni con integrazione numerica delle equazioni del moto. L'algoritmo si struttura attorno alla funzione di ottimizzazione 'fmincon' di cui viene riportata la caratteristica formulazione. Essa trova il minimo vincolato di una funzione con più variabili e cerca di risolvere il problema nella forma:

$$\begin{aligned} \min_x f(x): Ax \leq b, A_{eq}x = B_{eq} & \text{ (Vincoli lineari)} \\ c(x) \leq 0, c_{eq}(x) = 0 & \text{ (vincoli non lineari)} \\ LB \leq X \leq UB & \text{ (limiti)} \end{aligned}$$

Come precedentemente spiegato, tramite opzioni di ottimizzazione è possibile scegliere l'algoritmo SQP.

I principali componenti di questa ottimizzazione sono:

- Funzione obiettivo: massimizzare la massa finale del satellite.
- Vincoli: rispettare le condizioni finali per velocità e posizione.
- Integrazione della Dinamica: gestire le forze di propulsione e gravità lunare.

Nella implementazione su Matlab la funzione obiettivo necessita di essere scritta come una funzione separata, tale che prenda in input un vettore x , contenente le variabili di ottimizzazione. L'output deve essere l'obiettivo della funzione. I vettori vanno creati definendo il valore superiore e inferiore. I limiti e i vincoli vanno specificati. I risultati che si ottengono dalla esecuzione del codice vengono visualizzati tramite il comando "iter-detailed". I risultati includono dettagli come i punti ottimali (x) e il valore della funzione obiettivo nel punto minimo. Viene inclusa anche una visualizzazione dei dettagli dell'iterazione con il numero delle valutazioni della funzione.

Descrizione del codice

Il codice utilizza variabili globali per definire costanti fisiche e condizioni di riferimento. Nella tabella seguente vengono riportate le variabili globali, dichiarate all'inizio del problema.

Variabile	Valore	Descrizione
G	$6.67 \times 10^{-11} \text{ m}^3/\text{kg}/\text{s}^2$	Costante gravitazionale universale
M_moon	$7.342 \times 10^{22} \text{ m}$	Massa della Luna
R_moon	$1.738 \times 10^6 \text{ m}$	Raggio medio della Luna
r_rif	15000 m	Riferimento per la distanza
v_rif	1000 m/s	Velocità di riferimento
m_rif	1600 kg	Massa di riferimento

Tabella 1: Variabili globali

Queste variabili globali sono condivise tra le varie funzioni del programma.

Le condizioni iniziali sono definite in `y0`, come un array di valori rappresentativi della posizione, velocità e massa del satellite al tempo $t=0$, successivamente resi adimensionali in `y0_adim`.

```
y0 = [-500; 0; 15000; sqrt(G*M_moon/(R_moon + r_rif)); 0; 0; 1600];
y0_adim = y0 ./ [r_rif; r_rif; r_rif; v_rif; v_rif; v_rif; m_rif];
```

Il codice si struttura su diverse funzioni.

- “Risolvi(problems)” è la funzione principale ed effettua l’ottimizzazione del controllo usando `fmincon` per minimizzare l’output della funzione obiettivo “objective”.

```
x = fmincon(@(allVar) objective(allVar, problem), allVarGuess, [], [], [], [], allVarLower, allVarUpper, @(allVar) constraintss(allVar, problem), options);
```

- “Objective” è la funzione obiettivo. La funzione obiettivo è costruita per massimizzare la massa finale del satellite.
- “Constrains” rappresenta i vincoli, i quali si assicurano che il satellite rispetti la posizione e velocità corrette. I vincoli vanno espressi tramite condizioni di uguaglianza e disuguaglianza.
- La funzione “satelliteODEE” contiene le equazioni del moto, tenendo conto della forza di gravità e della spinta propulsiva.

Di seguito viene riportato uno schema a blocchi per cercare di rendere, visivamente, la struttura del codice in modo più chiaro.

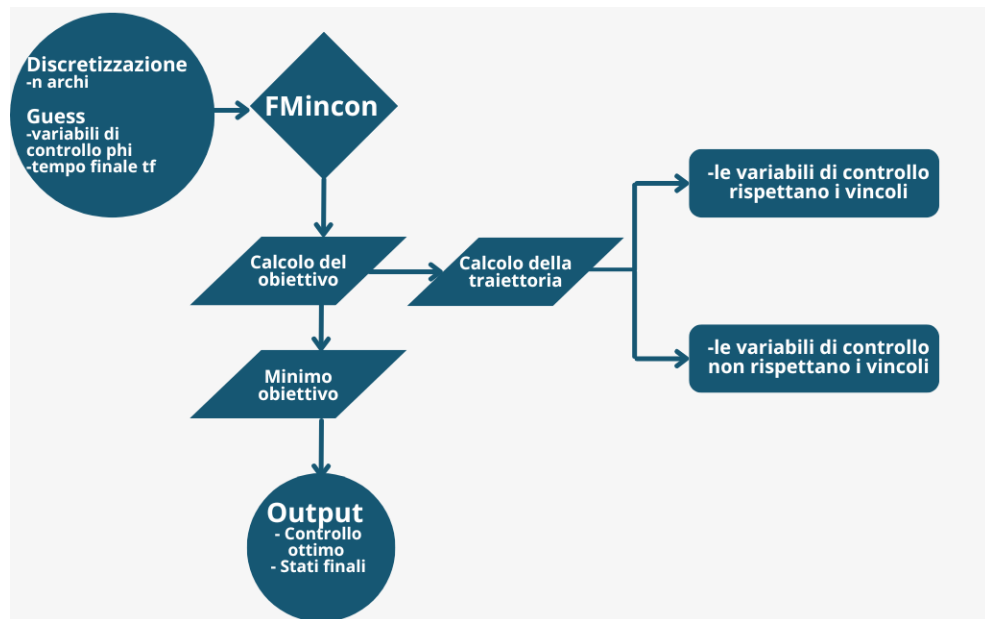


Figura 10: Struttura dell'algoritmo

Il codice, implementando il metodo diretto, impone una discretizzazione tramite n (nodi o archi) che permettono di analizzare la traiettoria in modo discreto a intervalli. Per ogni intervallo vengono calcolati i controlli e gli stati.

Le equazioni della dinamica del moto vengono adimensionalizzate in modo da lavorare con una variabile temporale ϵ adimensionale, che permette l'integrazione nel tempo in intervalli adimensionali.

Il tempo finale, inizialmente stimato con una variabile guess t_f , viene poi ripetutamente calcolato al fine di renderlo aderente con i valori di controllo scelti ad ogni iterazione.

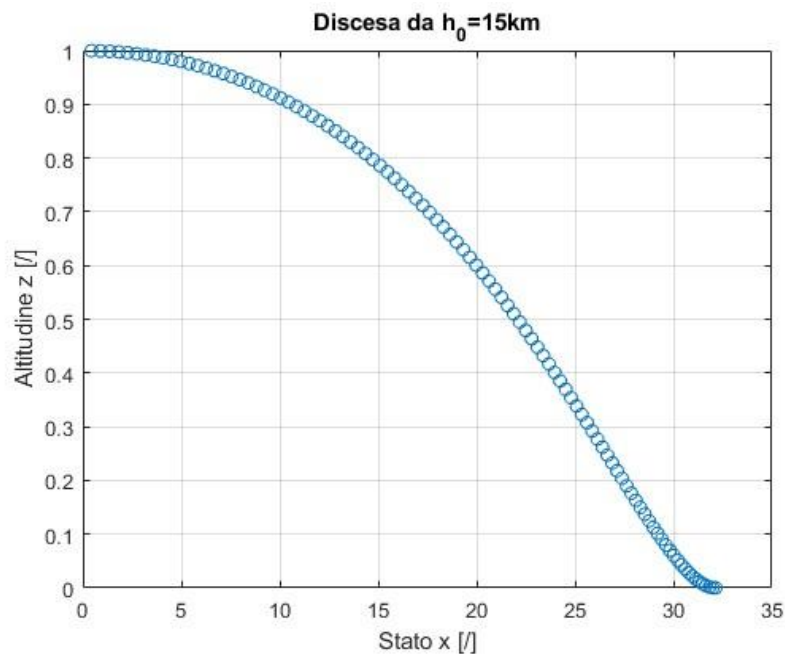
RISULTATI

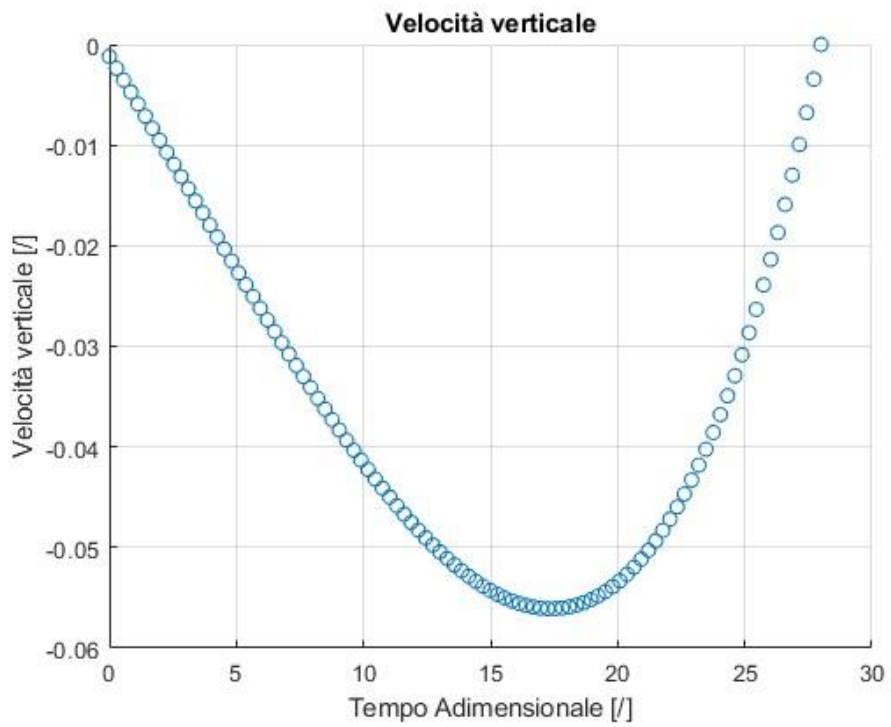
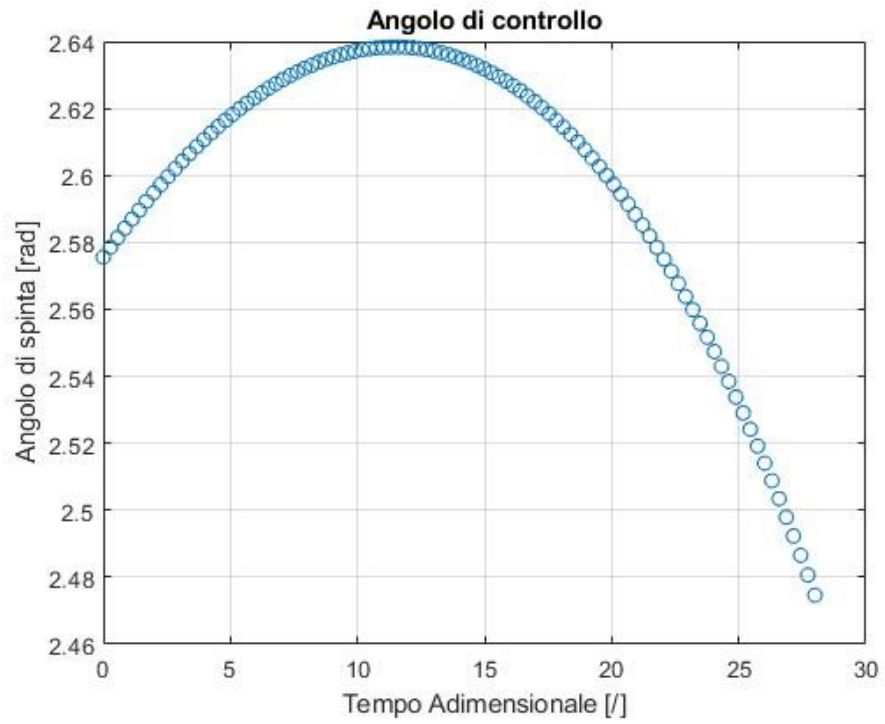
Analisi con spinta costante (prova1.m)

Di seguito vengono raccolti i risultati nel caso in cui la spinta venga mantenuta costante a 4000N, valore opportuno in questa fase. Si sceglie di minimizzare la massa finale come -mf, che equivale a massimizzare la massa finale +mf. La discesa si verifica correttamente, con velocità verticale nulla al suolo e con angoli ottimali in figura. La scelta delle variabili di tentativo iniziali, da cui parte la funzione fmincon, sono riportate in tabella.

Parametro	Valore
Tipo di analisi	Spinta costante a 4000N
Variabili di controllo	Angoli (uguess) e tempo finale
Vincoli	$V_z=0$ a $z=0$
Limiti sul tempo finale [s]	problems.tfMin = 375 (minimo) problems.tfMax = 600 (massimo)
Limiti sui controlli [rad]	problems.controlMin = [1.75] (minimo) problems.controlMax = [π] (massimo)
Stima iniziale dei controlli [rad]	problems.uguess(:, 1) = [2.4; 2.95]
Stima iniziale del tempo [s]	problems.tguess = [0; 420]
Numero di nodi	problems.nodes = 100

Tabella 2: Parametri della analisi con spinta costante





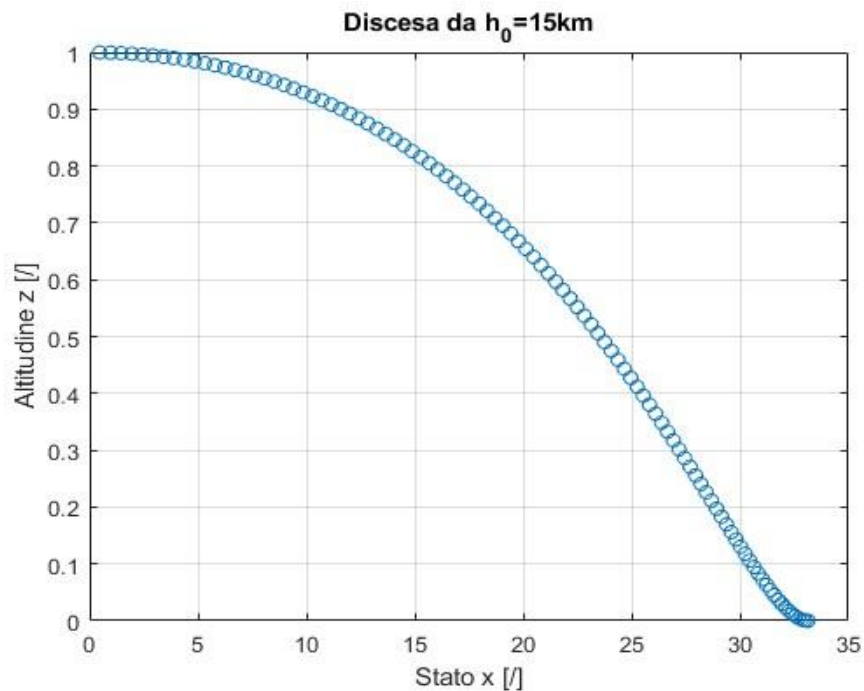
Da questa prima analisi, si ottiene che il tempo per discendere sulla luna sia di 419.9957s. La massa finale è 1009,3kg, ricordando che la massa di partenza è fissata a 1600kg a 15km dal suolo.

Ho riscontrato che il codice è molto sensibile ai guess iniziali. Modificando i parametri come segue i risultati cambiano come rappresentato di seguito.

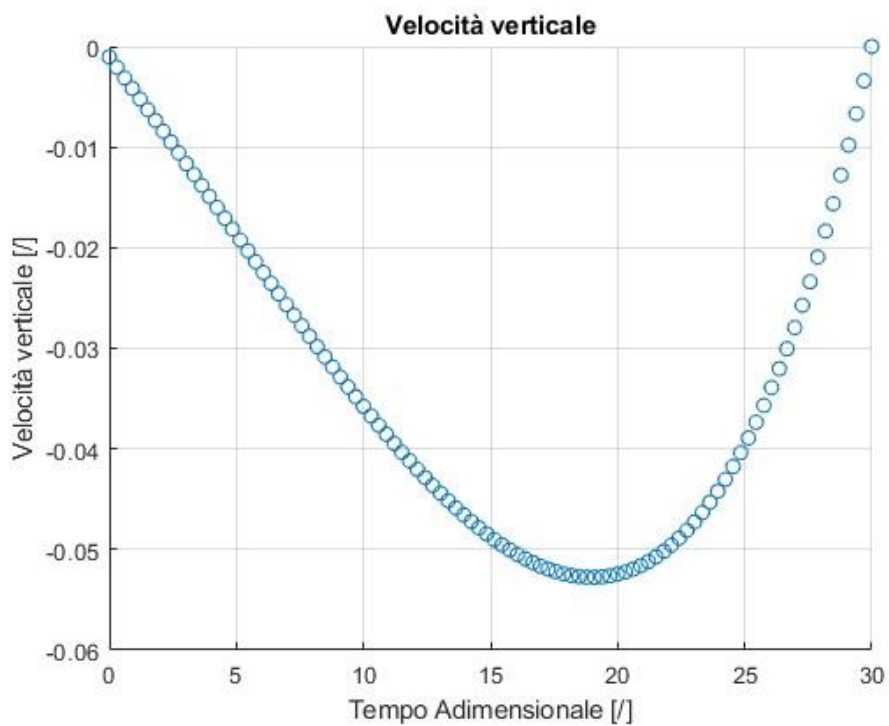
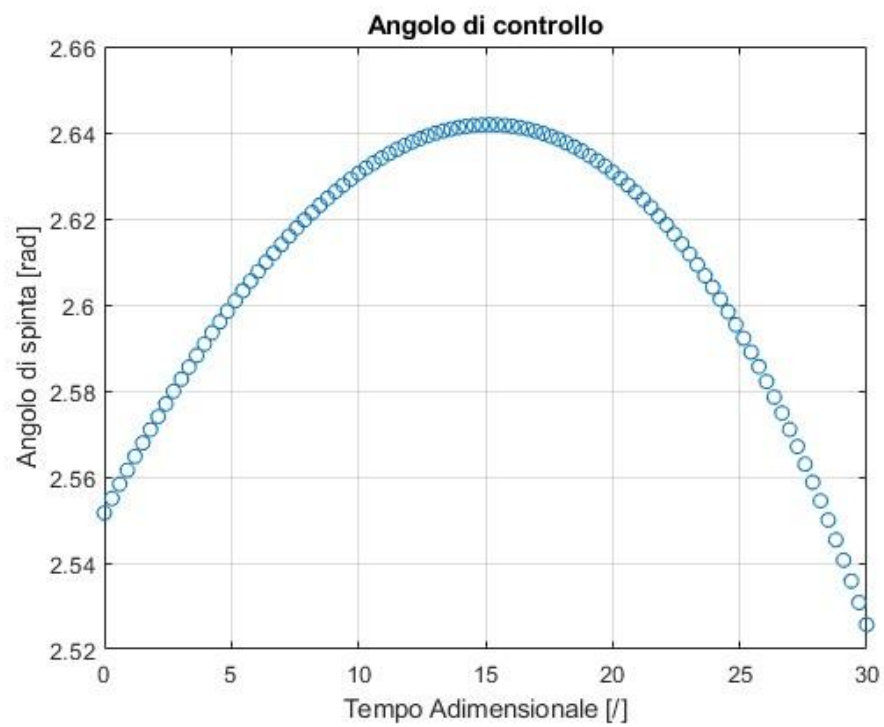
Parametro	Valore
Tipo di analisi	Spinta costante a 4000N
Variabili di controllo	Angoli (uguess) e tempo finale
Vincoli	$V_z=0$ a $z=0$
Limiti sul tempo finale [s]	problems.tfMin = 375 (minimo) problems.tfMax = 675 (massimo)
Limiti sui controlli [rad]	problems.controlMin = [2] (minimo) problems.controlMax = [π] (massimo)
Stima iniziale dei controlli [rad]	problems.uguess(:, 1) = [2.4; 2.95]
Stima iniziale del tempo [s]	problems.tguess = [0; 450]
Numero di nodi	problems.nodes = 100

Tabella 3: Modifica ai parametri della analisi con spinta costante

Con queste ipotesi modificate, il tempo finale della discesa si riduce a 449.8973s. Anche la massa finale si riduce di circa 30kg rispetto alla analisi precedente, raggiungendo 967.2kg



L'andamento del controllo, prima crescente e poi decrescente, non è facilmente spiegabile. Si può però commentare che il fatto che ci siano valori tendenzialmente più grandi all'inizio (spinta più verso $-z$ che verso $+z$) e più piccoli alla fine, ha senso. Dopotutto, si deve inizialmente guadagnare velocità verticale per scendere a quota nulla, e poi si deve annullare tale velocità verticale per evitare di impattare con troppa violenza sulla superficie lunare.



Da questa analisi emerge che

1. Il tempo finale è molto vicino al guess iniziale dello stesso.
2. Il controllo varia nel suo andamento in funzione del guess sul tempo.

Analisi con spinta costante e atterraggio verticale (spinta_var_con_t_att_fissa.m)

È stata spezzata la traiettoria in due porzioni, una che rappresentasse la discesa e una che rappresentasse l'atterraggio verticale. Viene trascurata la presenza di una fase di approccio, intermedia tra le due.

- da $t=0$ a $t=tf$ (discesa)
- da $t=tf$ a $t=tf1$ (atterraggio verticale)

L'integrazione, quindi, viene suddivisa su due intervalli temporali, il primo viene discretizzato su 100 intervalli, il secondo su 10 intervalli.

Per avere l'atterraggio verticale vengono mantenuti i vincoli già impostati nella trattazione precedente ($v_z=0$ a $z=0$) e viene aggiunto il vincolo sulla velocità orizzontale nulla all'inizio dell'atterraggio verticale ($v_x=0$ a $t=5s$ o $10s$).

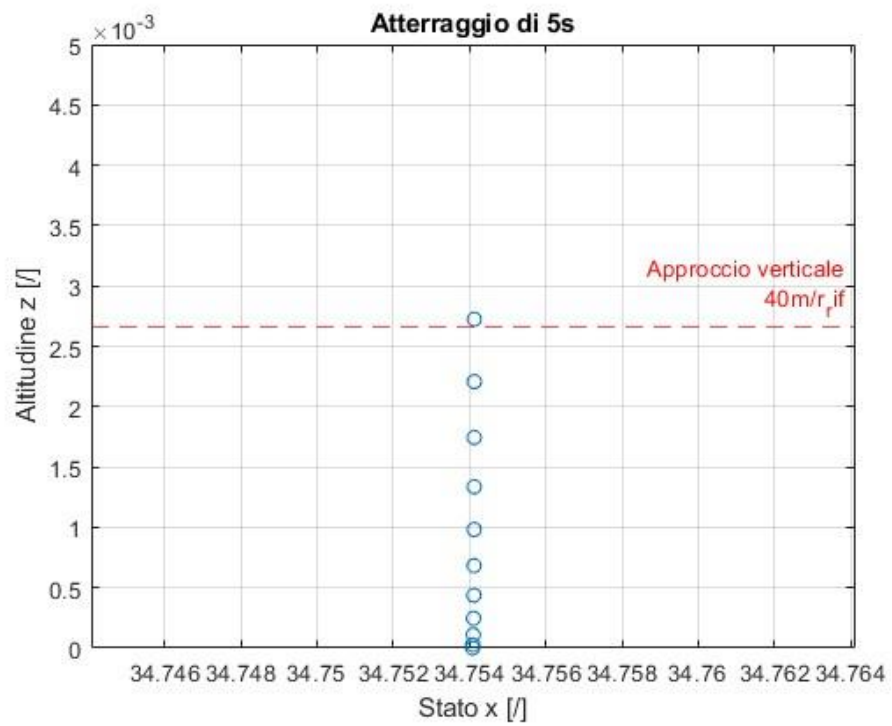
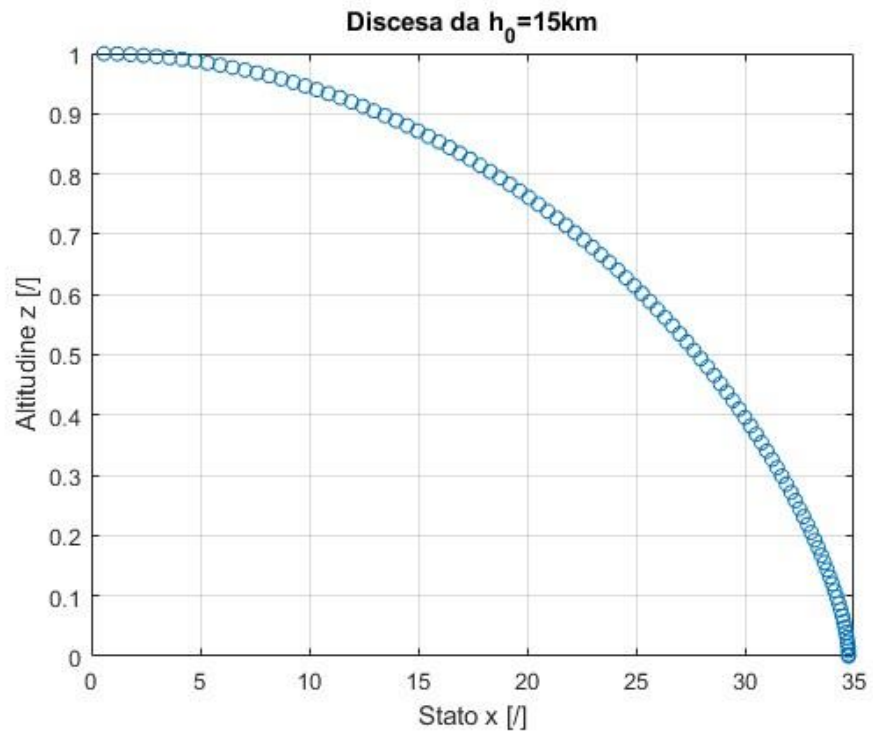
Per permettere al codice di funzionare correttamente, è stato inserito un vincolo di disequaglianza, prima assente, che impone che $tf < tf1$, ovvero che il tempo impiegato dall'inizio della discesa fino all'inizio dell'atterraggio verticale, sia inferiore al tempo complessivo per toccare il suolo lunare.

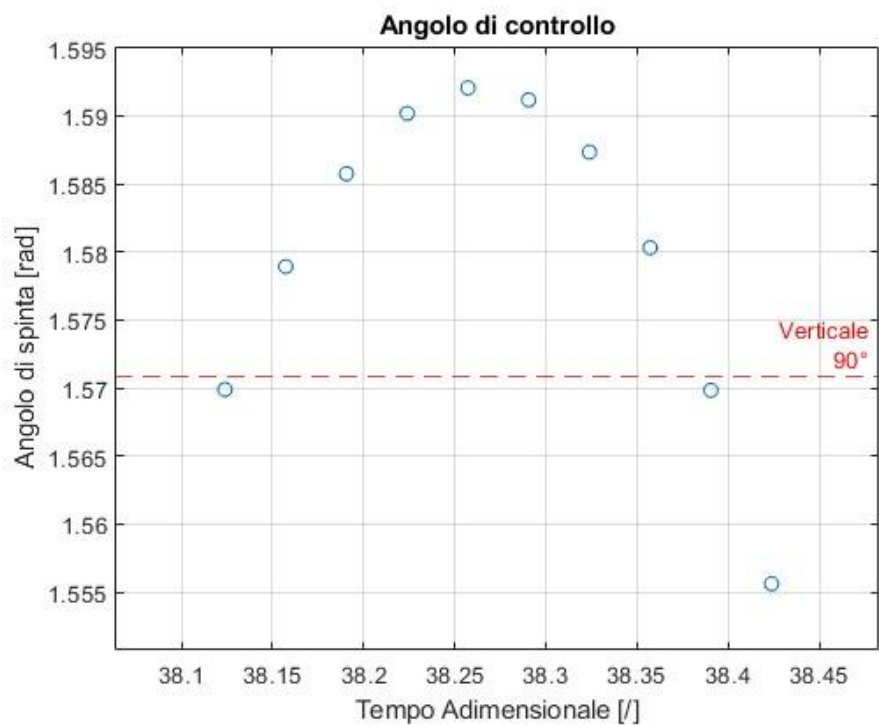
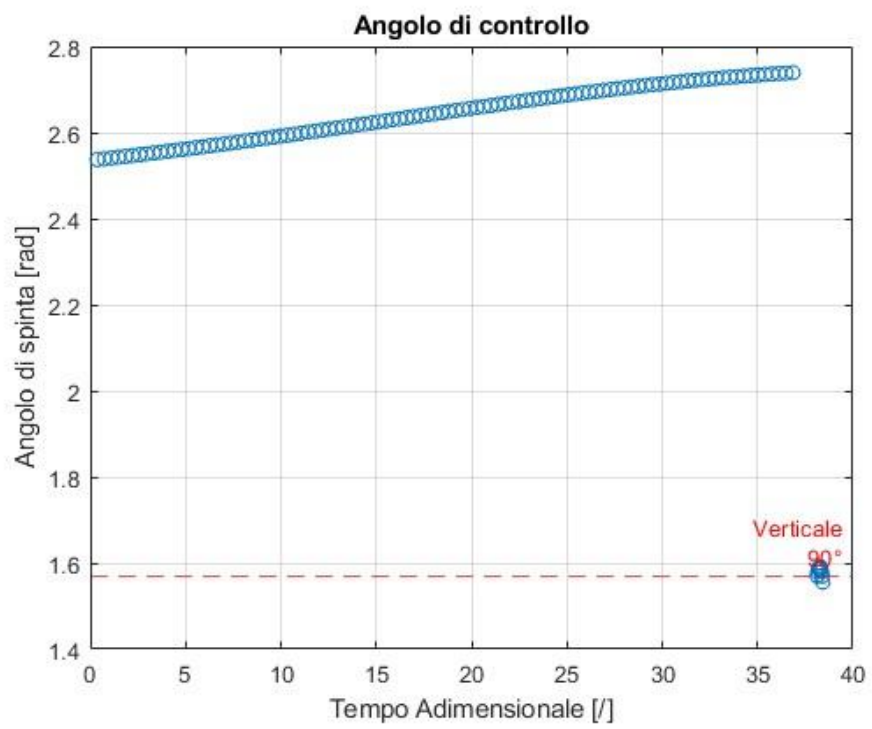
Tabella 4: Analisi con atterraggio verticale

Parametro	Valore
Tipo di analisi	Spinta costante a 4000N
Variabili di controllo	Angoli (u _{guess} +u _{1guess}) e tempi (tf e tf1)
Vincoli	$V_z=0$ a $z=0$, $V_x=0$ a $t=5s$, $tf < tf1$
Limiti sul tempo tf [/]	problems.tfMin = 30 (minimo)
	problems.tfMax = 40 (massimo)
Limiti sul tempo tf1 [/]	problems.tf1Min = 36 (minimo)
	problems.tf1Max = 41 (massimo)
Limiti sui controlli discesa[rad]	problems.controlMin = [2] (minimo)
	problems.controlMax = [π] (massimo)
Limiti sui controlli atterraggio[rad]	problems.control1Min = [1.4] (minimo)
	problems.control1Max = [1.7] (massimo)
Stima dei controlli discesa [rad]	problems.u _{guess} (:, 1) = [2.4; 2.95]
Stima dei controlli atterraggio [rad]	problems.u _{1guess} (:, 1) = [1.65; 1.56]
Stima del tempo di discesa [/]	problems.t _{guess} = [0; 30]
Stima del tempo di atterraggio [/]	problems.t _{1guess} = [0; 40]
Numero di nodi	problems.nodes = 100+10=110

I risultati vengono riportati nelle figure che seguono, prima vengono presentati i risultati con un atterraggio previsto di 5s e di seguito quelli con un atterraggio di 10s.

Atterraggio di 5s





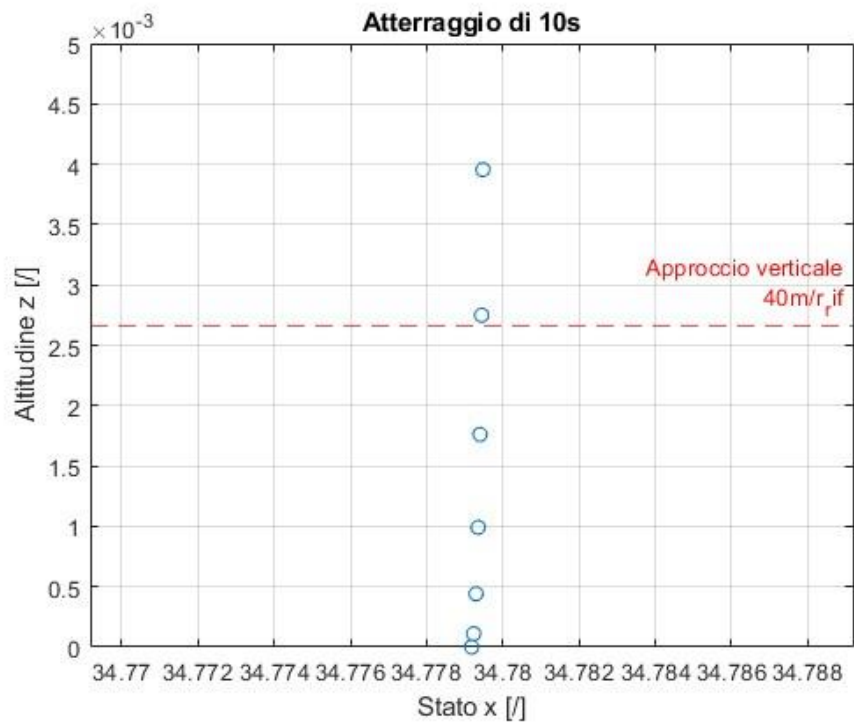
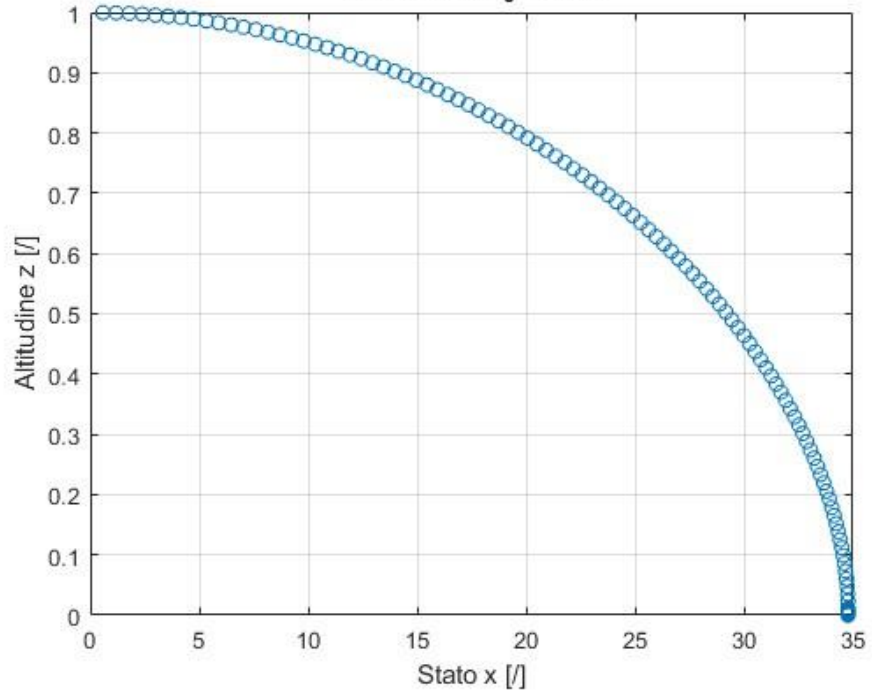
Con un tempo di atterraggio di 5s si ottengono i seguenti risultati:

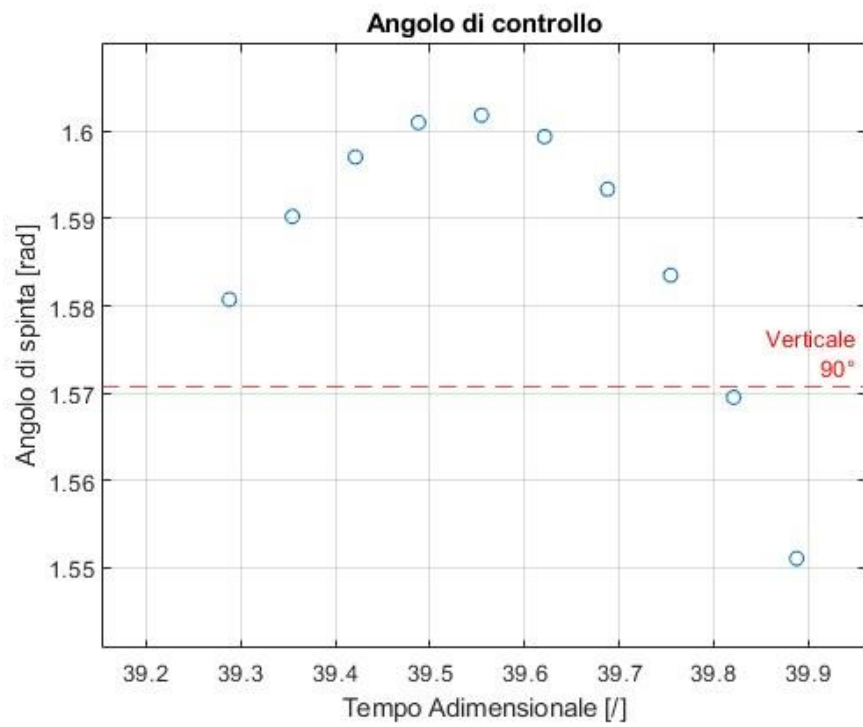
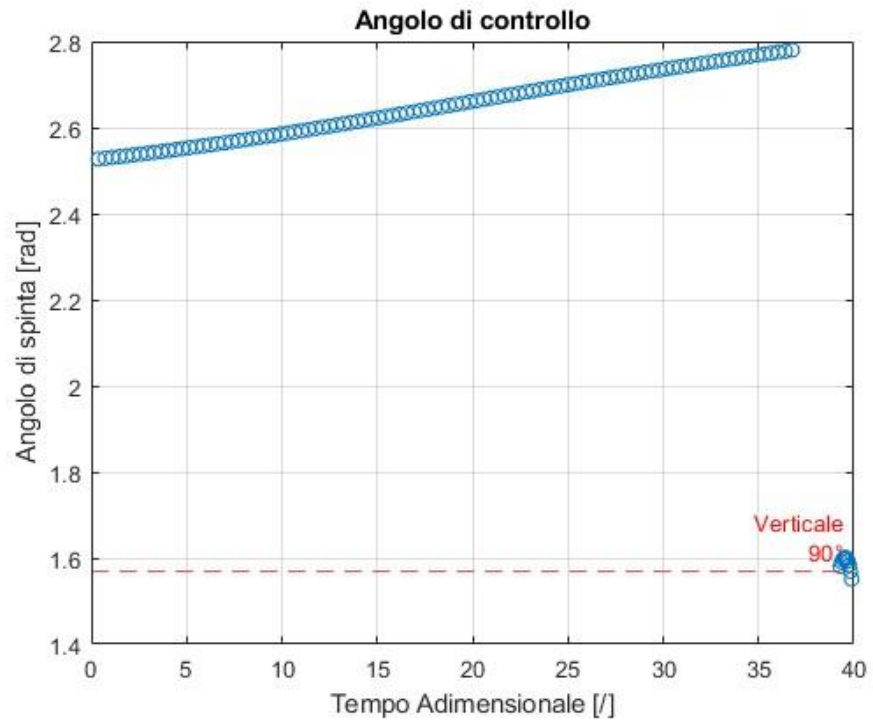
La massa finale è:	815.27 kg
Il tempo di discesa è:	552.93 s
Il tempo di touchdown è:	557.93 s

La velocità z si annulla ancora al tempo finale, come richiesto. Gli angoli di controllo nella fase di atterraggio non risultano precisamente uguali a novanta gradi. Dal codice risulta ottimale un controllo che oscilli attorno ai novanta gradi con un andamento ad U rovesciata. Traducendo gli angoli in gradi, risulta evidente, però, che questo controllo sia pressoché prossimo ad un controllo perpendicolare. Infatti, il valore massimo e minimo, in corrispondenza di 1.593rad e 1.555rad, rivelano che negli ultimi 5 secondi di atterraggio l'angolo della spinta ottimale vada variato tra 91.3° e 89.1° .

Atterraggio di 10s

Discesa da $h_0 = 15\text{km}$





La massa finale è:	810.18 kg
Il tempo di discesa è:	551.55 s
Il tempo di touchdown è:	561.55 s

Angoli e spinta di atterraggio ottimale (spinta_var.m)

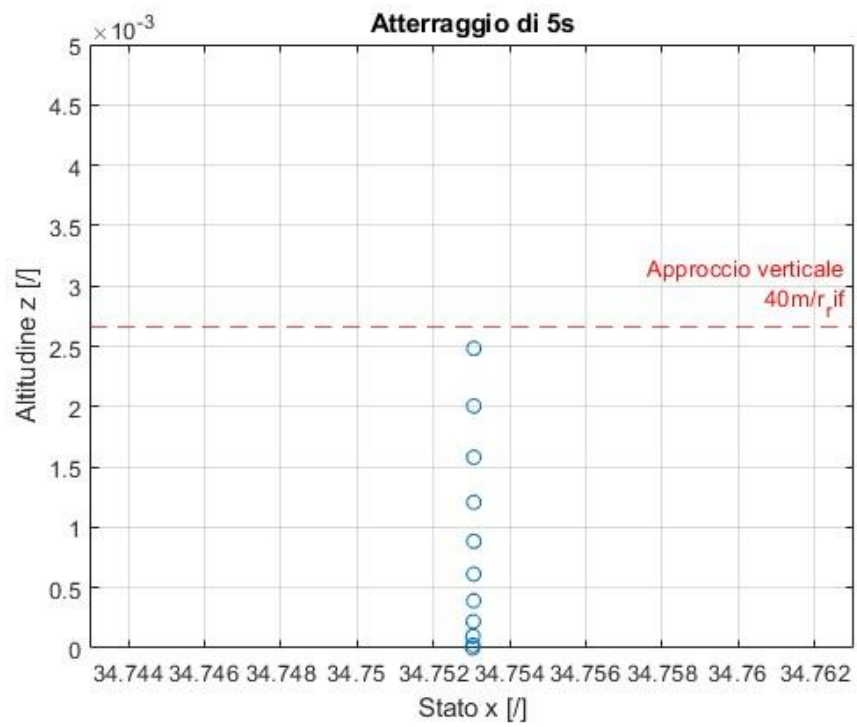
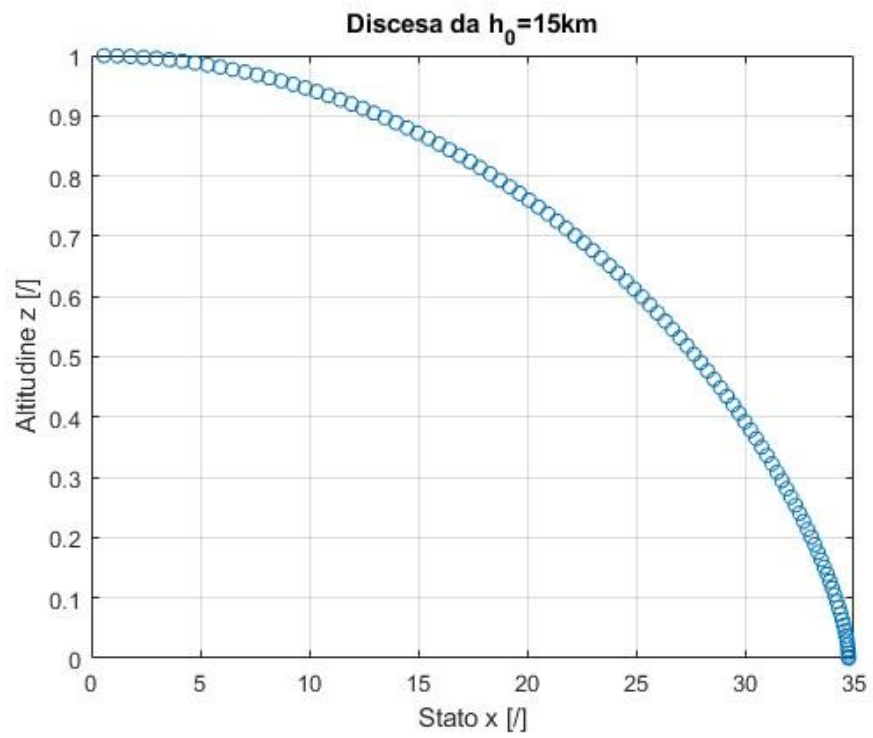
Questo paragrafo riassume, senza riportare tutte le analisi non funzionanti, il perché non convenga la spinta variabile nella fase finale.

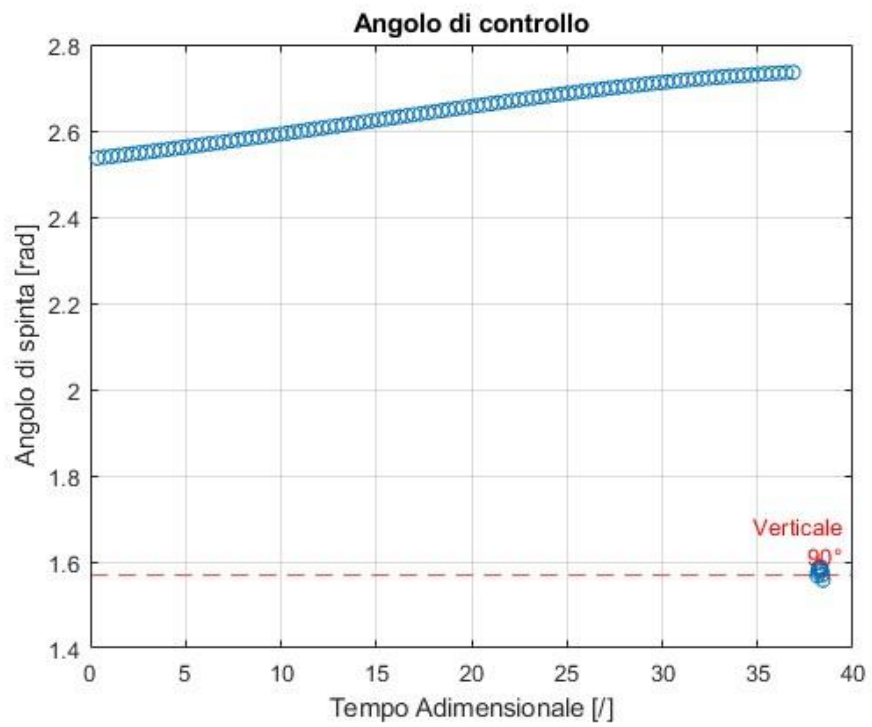
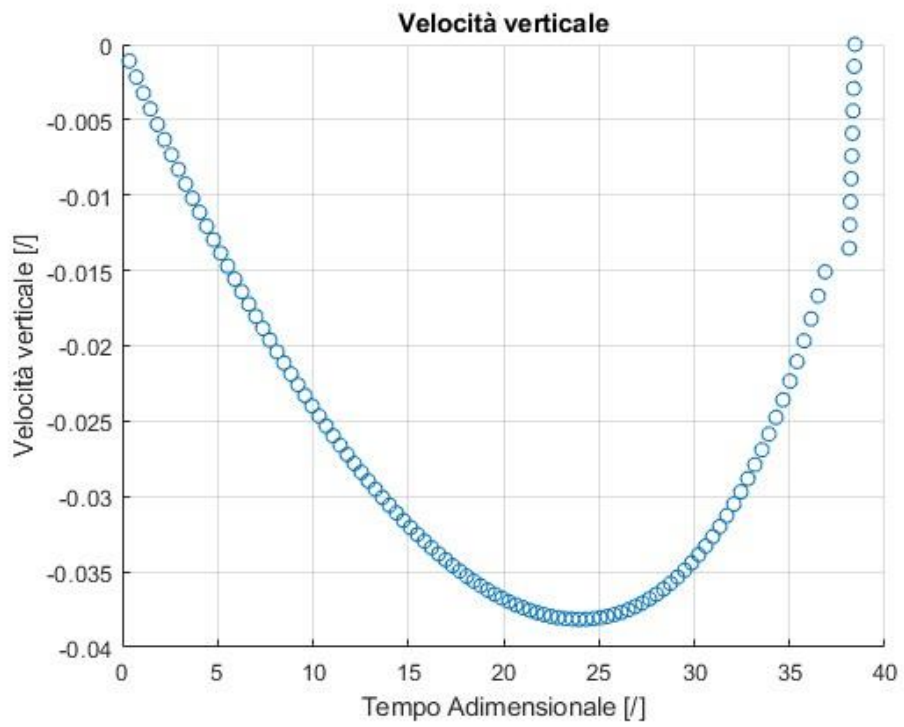
Si procede ad una analisi ora con spinta variabile nel tratto finale, considerando una possibile regolazione del 95% e 90% della spinta massima. La spinta, quindi, viene ora inserita come variabile di ottimizzazione aggiuntiva, oltre agli angoli di spinta. Le variabili totali in questo problema raggiungeranno il numero di 122 in totale.

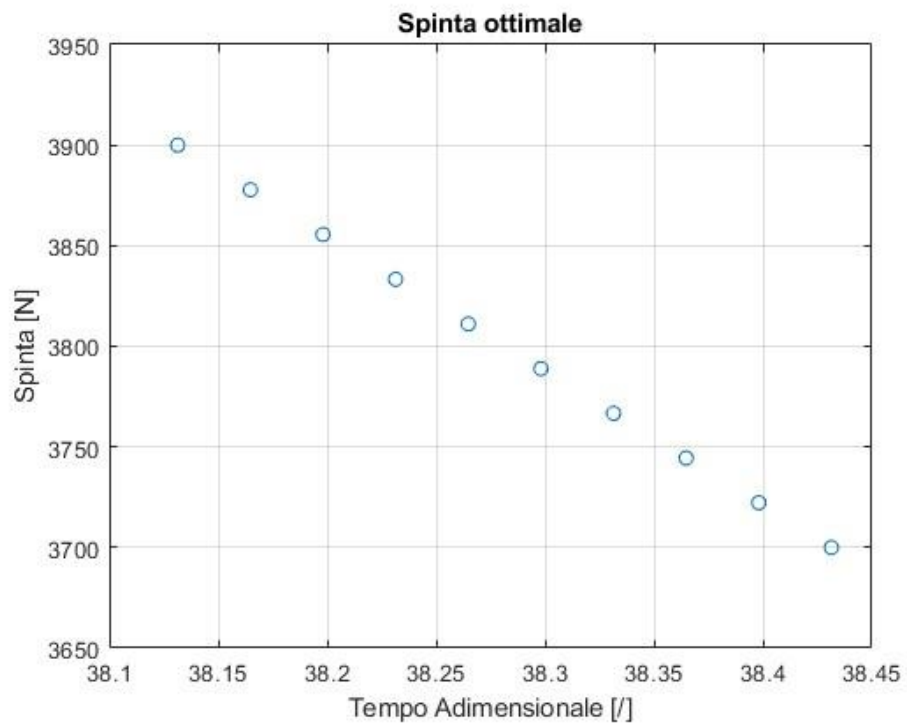
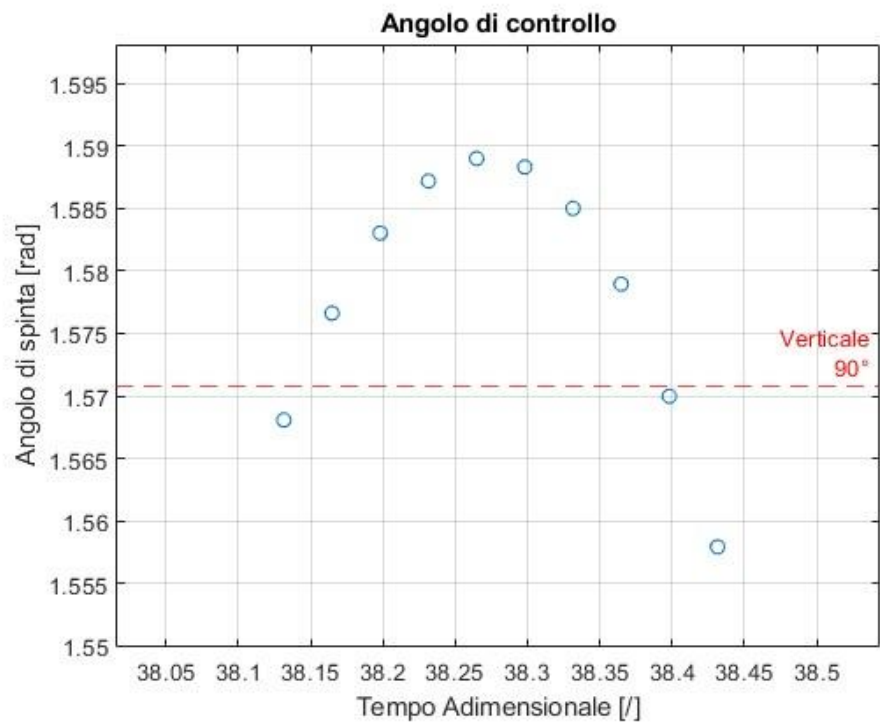
Tabella 5: Analisi con spinta variabile

Parametro	Valore
Tipo di analisi	Spinta di atterraggio variabile
Variabili di controllo	Angoli (u _{guess} +u _{1guess}), spinta di atterraggio, tempi (tf e tf1)
Vincoli	V _z =0 a z=0, V _x =0 a t=5s, tf<tf1
Limiti sulla spinta [/]	problems.spintaMin = 3600 (minimo) problems.spintaMax = 4000 (massimo)
Limiti sul tempo tf [/]	problems.tfMin = 30 (minimo) problems.tfMax = 40 (massimo)
Limiti sul tempo tf1 [/]	problems.tf1Min = 36 (minimo) problems.tf1Max = 41 (massimo)
Limiti controlli discesa[rad]	problems.controlMin = [2] (minimo) problems.controlMax = [π] (massimo)
Limiti controlli atterraggio[rad]	problems.control1Min = [1.4] (minimo) problems.control1Max = [1.7] (massimo)
Stima spinta atterraggio [N]	Problems.spintaguess=[3900; 3700]
Stima dei controlli discesa [rad]	problems.uguess(:, 1) = [2.4; 2.95]
Stima dei controlli atterraggio [rad]	problems.u1guess(:, 1) = [1.65; 1.56]
Stima del tempo di discesa [/]	problems.tguess = [0; 30]
Stima del tempo di atterraggio [/]	problems.t1guess = [0; 40]
Numero di nodi	problems.nodes = 110

Nelle immagini seguenti vengono raccolti i risultati in forma di grafici.







Si ottiene:

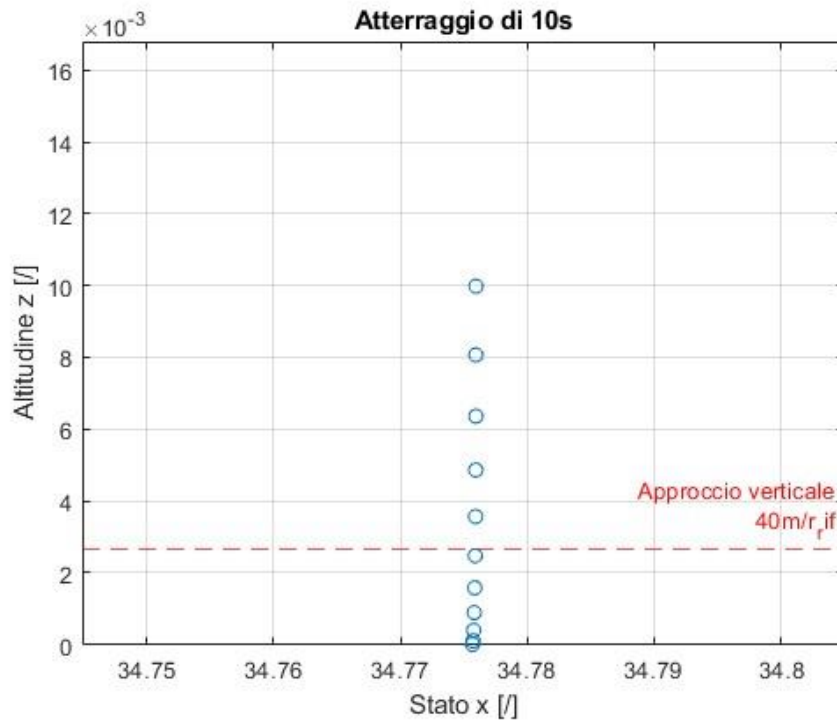
La massa finale è:	815.47 kg
Il tempo di discesa è:	553.04 s
Il tempo di touchdown è:	558.04 s

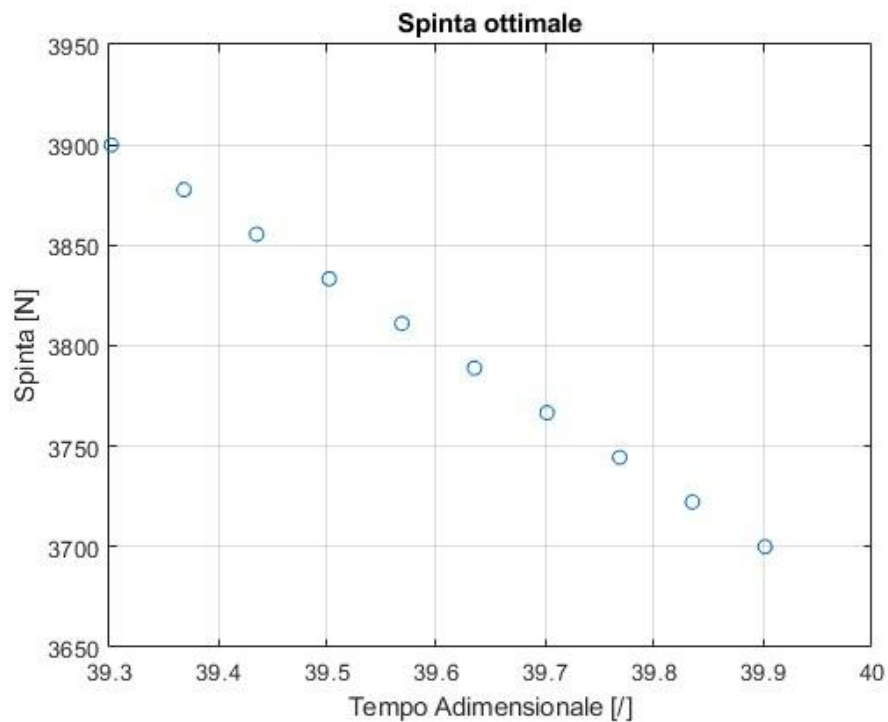
Inoltre, ottengo la spinta ottimizzata per il tratto finale diviso in 10 archi:

Spinta del tratto finale ottimizzata [N]
3900.0
3877.8
3855.6
3833.3
3811.1
3788.9
3766.7
3744.4
3722.2
3700.0

Come nel caso di spinta costante, i risultati del controllo finale risultano oscillare attorno al valore di 90° . Al fine di favorire una discesa più verticale, nella analisi sono stati limitati i limiti di variazione del controllo ad un intervallo più ristretto di quello impostato inizialmente, che era $[1 \text{ rad}, 2\text{rad}]$. La scelta della spinta variabile ci permette di osservare come sarebbe preferibile diminuire la spinta negli ultimi 5s.

Variando il tempo di atterraggio a 10s non si assiste a particolari modifiche; gli angoli di controllo e la traiettoria mantengono andamenti molto simili a quelli per 5s. Va invece testimoniato come, ovviamente, l'atterraggio verticale inizi a quote superiori ai 40m. La spinta ha un andamento invariato rispetto al precedente.



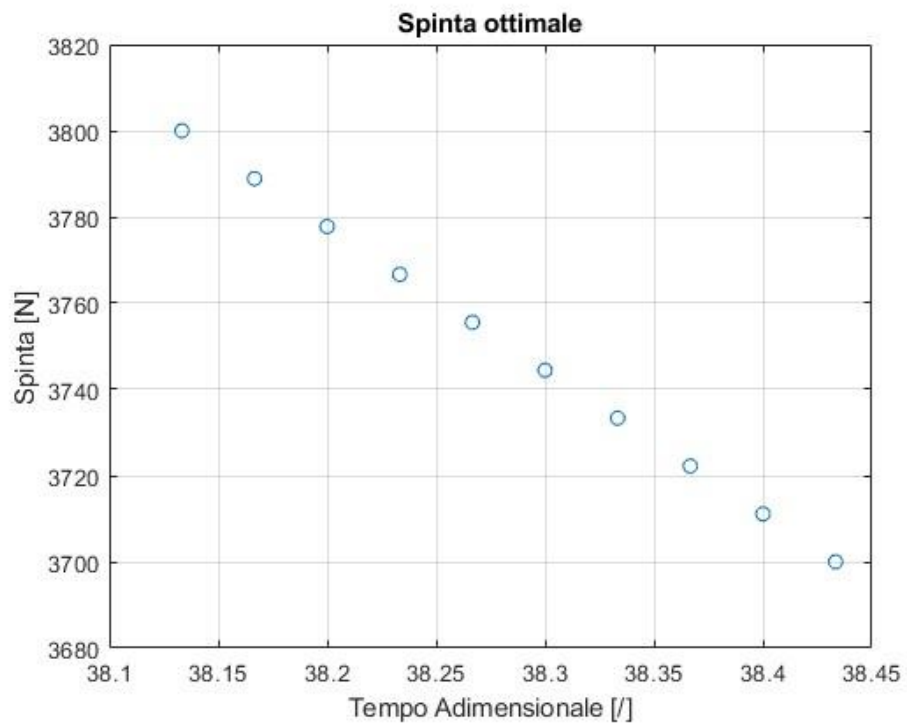
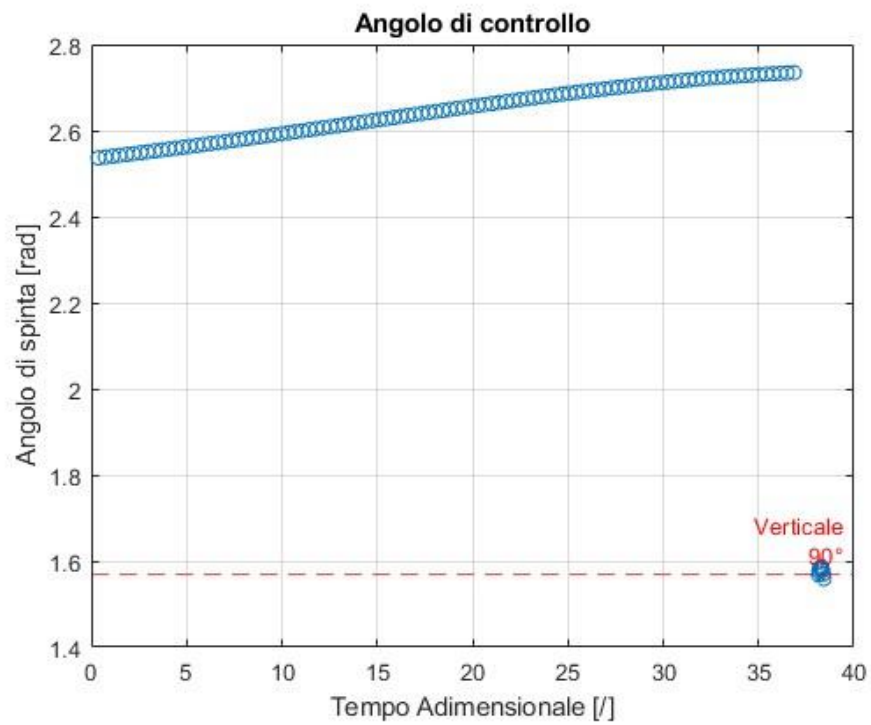


Sensibilità della spinta variabile al guess iniziale

Dato che risulta sospetto che in entrambi i casi la spinta risulti coincidente al guess iniziale, viene mantenuto lo stesso intervallo di vincolo sull'angolo di spinta finale ma viene cambiato il guess iniziale sulla spinta. L'obiettivo è verificare la sensibilità della analisi al guess iniziale sulla spinta. In particolare, le modifiche apportate risultano:

Guess modificati:	Spinta: 3800->3700
-------------------	--------------------

Risulta che la traiettoria non subisce modifiche rilevanti, e tantomeno i controlli risentono della modifica.



La spinta anche in questo caso risulta ottimale ridurla man mano che ci si avvicina alla superficie lunare. In generale, tentando altre ipotesi iniziali sulla

spinta, l'andamento decrescente della spinta ottimale nelle fasi finali risulta sempre analogo.

Al netto però dell'obiettivo di minimizzare il consumo una variazione della spinta negli ultimi 5s non riporta grandi migliorie sul valore finale della massa in quanto con una spinta costante di 4000N o 3600N, e condizioni iniziali riportate di seguito, si ottiene quanto segue.

Limiti controlli discesa[rad]	problems.controlMin = [2] (minimo)
	problems.controlMax = [π] (massimo)
Limiti controlli atterraggio[rad]	problems.control1Min = [1.4] (minimo)
	problems.control1Max = [1.7] (massimo)
Stima dei controlli discesa [rad]	problems.uguess(:, 1) = [2.4; 2.95]
Stima dei controlli atterraggio [rad]	problems.u1guess(:, 1) = [1.65; 1.56]
Stima del tempo di discesa [/]	problems.tguess = [0; 30]
Stima del tempo di atterraggio [/]	problems.t1guess = [0; 40]

Dal confronto si evince che la scelta di usare una spinta costante di 4000N è la scelta più onerosa in termini di carburante, mentre la spinta di 3600N costante permette di arrivare con una massa residua superiore. La spinta variabile fornisce risultati intermedi alle due casistiche. In generale, comunque, la massa residua non subisce grandi variazioni tra i tre casi.

Spinta	Costante a 4000N	Costante a 3600N	Variabile al 90% con spinta iniziale [3800:3700]N
massa finale	815.27 kg	815.67 kg	815.52 kg

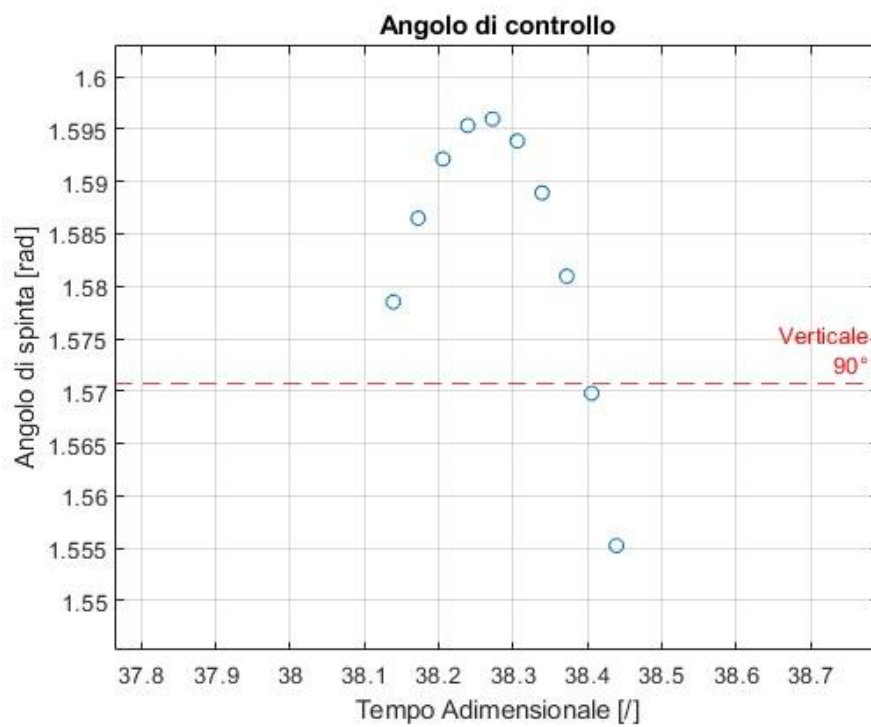
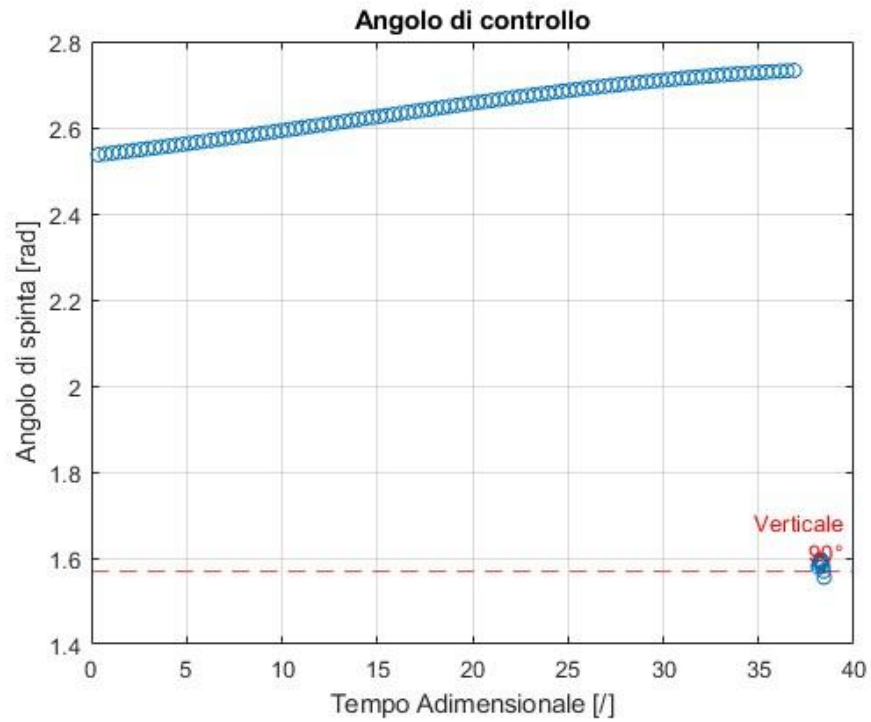
Per capire, perché, in effetti il codice con la spinta variabile non riesca a trovare come migliore la soluzione con 3600N costanti rispetto alla spinta variabile tra 3800N e 3700N, si è cercato di apportare delle migliorie all'algoritmo. In particolare, si è optato di proporre come guess iniziale sulla spinta il valore adimensionalizzato. Prima di questa modifica, l'adimensionalizzazione della spinta avveniva all'interno della funzione 'satelliteODEE.m'. In questo modo però l'algoritmo fmincon processava valori delle variabili di ottimizzazione di ordini di grandezza molto diversi, e questo genera problemi numerici.

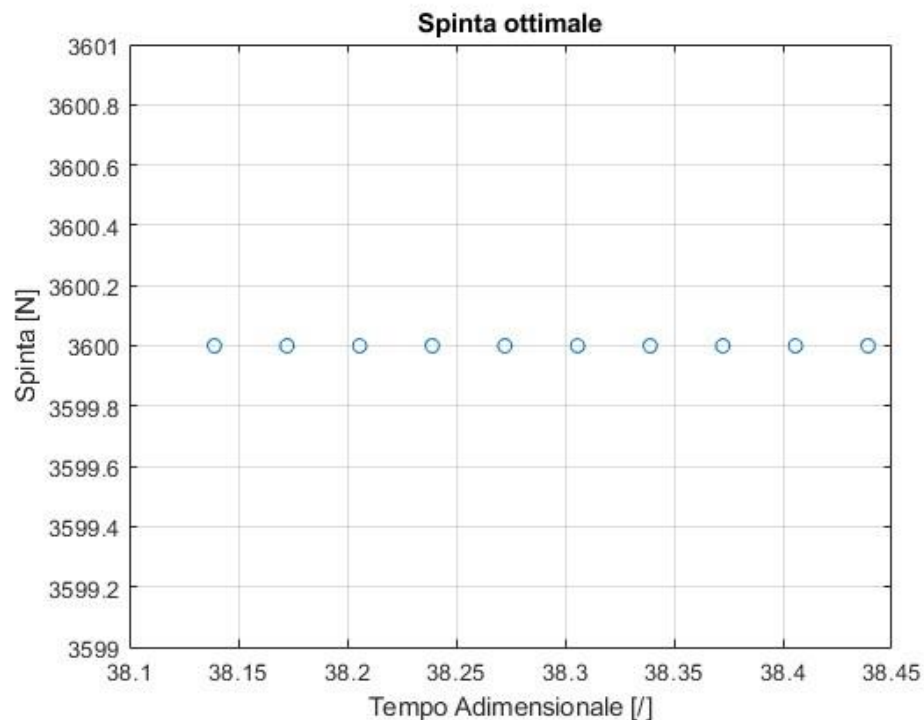
Questo accade perché gli algoritmi di ottimizzazione si basano su gradienti e passi iterativi, che rischiano di essere influenzati in modo negativo da queste differenze. In particolare, la matrice Hessiana potrebbe non risultare ben condizionata, causando un rallentamento della convergenza o soluzioni non corrette.

Al fine di risolvere questa problematica, viene scalata la variabile della spinta prima di inserirla su fmincon (spinta_var2.m). L'analisi con la spinta adimensionalizzata, realizzata con gli stessi input fa ottenere i seguenti risultati:

La massa finale è:	815.67 kg
Il tempo di discesa è:	553.15 s
Il tempo di touchdown è:	558.15 s

Grazie alla adimensionalizzazione, l'ottimizzatore riesce a evidenziare la soluzione ottimale di 3600N come spinta costante.





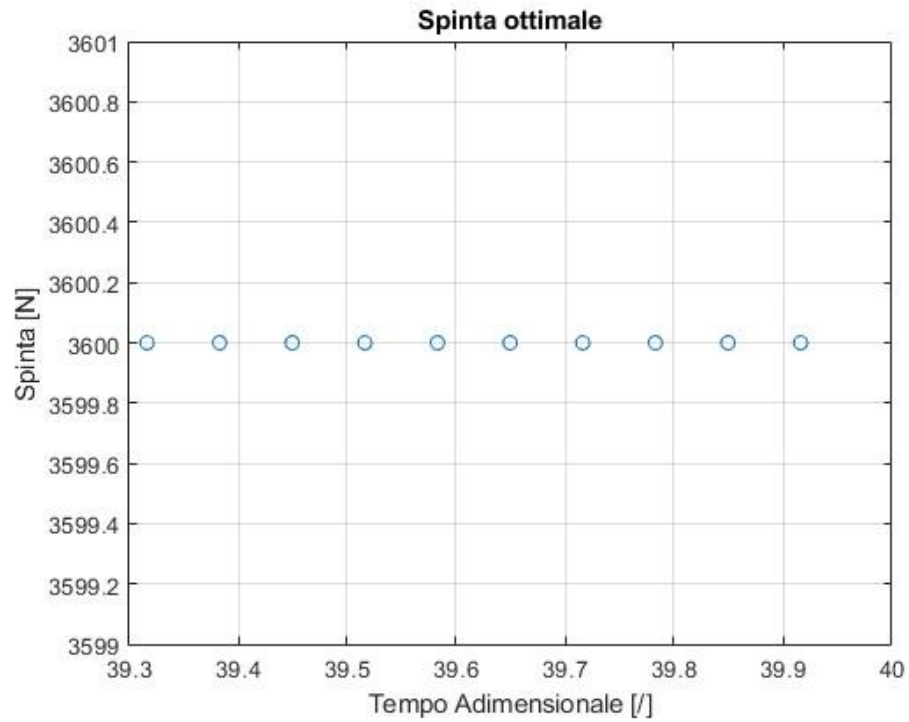
Con questa trattazione adimensionalizzata della spinta, l'algoritmo risulta più preciso perché lavora su ordini di grandezza più coerenti. Infatti, ritrova la spinta costante di 3600N, ovvero la spinta pari al limite inferiore di variazione T_{min} , come la migliore per i consumi.

A livello teorico la soluzione ottima è sempre spingere a quote basse, infatti per un dato ∇V , la velocità finale cresce con la spinta. Questo comporta che convenga spingere a quote basse, perché la spinta modifica l'energia in modo più efficace ad alte velocità.

Se si potesse regolare arbitrariamente, l'ottimo sarebbe tenere il motore spento per un po' e poi frenare con T_{max} . Si è cercato di ottenere questa soluzione tramite l'algoritmo allargando il campo di variazione sulla variabile della spinta. Modificando quindi la trattazione con $T_{min}=0$, il codice purtroppo non riesce a trovare la soluzione suggerita. Probabilmente il problema risiede ancora una volta in un problema numerico, la spinta adimensionalizzata raggiunge ordini di grandezza troppo diversi tra la spinta minima e la massima, compromettendo il funzionamento del modello.

Si è provato impostando $T_{min}=1000N$ e $T_{max}=4000 N$, il comportamento della spinta ottima continua a mantenersi costante a T_{min} . Potrebbe essere influente l'utilizzo di soli 10 punti discreti, studi successivi potrebbero soffermarsi su quanto influisca.

Riguardo il comportamento della spinta di atterraggio verticale, ottimizzando i consumi sull'intera traiettoria di circa 560 secondi, mi risulta che la spinta ottimale all'atterraggio è costante a 3600N variando il tempo di atterraggio tra 5 e 10 secondi.



L'atterraggio di 5s è però preferibile, infatti l'atterraggio di 10s, a pari spinta di 3600N, consente una massa finale inferiore di 811.02 kg. Il tempo di discesa è di 551.95s mentre il suolo si raggiunge dopo 561.95s.

Spinta variabile e atterraggio vincolato a 90°

Si procede ad una analisi sempre con spinta variabile nel tratto finale, imponendo nell'ultimo tratto angolo 90°. La spinta, quindi, è ancora inserita come variabile di ottimizzazione aggiuntiva. Viene scelto di mantenere invece l'angolo di spinta dell'ultimo tratto verticale pari a 90° per alleggerire il codice. Quindi gli ultimi 10 angoli, ora, sono eliminati dalla lista delle variabili di controllo, le quali si sono ridotte a 112.

Parametro	Valore
Tipo di analisi	Spinta di atterraggio variabile a 4000N
Variabili di controllo	Angoli (uguess), spinta di atterraggio, tempi (tf e tf1)
Vincoli	$V_z=0$ a $z=0$, $V_x=0$ a $t=5s$, $tf < tf1$
Limiti sulla spinta [/]	problems.spintaMin = 3600 (minimo) problems.spintaMax = 4000 (massimo)
Limiti sul tempo tf [/]	problems.tfMin = 30 (minimo) problems.tfMax = 40 (massimo)
Limiti sul tempo tf1 [/]	problems.tf1Min = 36 (minimo) problems.tf1Max = 41 (massimo)
Limiti controlli discesa[rad]	problems.controlMin = [2] (minimo) problems.controlMax = [π] (massimo)
Stima spinta atterraggio [N]	Problems.spintaguess=[3900; 3700]
Stima dei controlli discesa [rad]	problems.uguess(:, 1) = [2.4; 2.95]
Stima del tempo di discesa [/]	problems.tguess = [0; 30]
Stima del tempo di atterraggio [/]	problems.t1guess = [0; 40]
Numero di nodi	problems.nodes = 110

Da questa analisi si ritrovano gli stessi identici risultati precedenti, risulta ottimale la spinta di 3600N e risulta una massa finale di 815.67 kg. Questa analisi rende quindi evidente come le oscillazioni dell'angolo di spinta all'atterraggio siano così lievi da non portare benefici ulteriori sui consumi o sui tempi di atterraggio.

$m_{atterraggio}$
0.5134
0.5130
0.5125
0.5121
0.5117
0.5113
0.5109
0.5105
0.5101
0.5097

Con spinta che varia da 4000N a 3600N, è interessante sottolineare che negli ultimi 5s viene registrata una variazione della massa adimensionale di solo:

$$\Delta m_{\text{atterraggio}} = 0.5134 - 0.5097 = 0.0037$$

Nel tratto di discesa, in cui la spinta viene considerata costante pari a 4000N, dal tempo iniziale fino al tempo di 5s la variazione della massa adimensionale invece è di ben:

$$\Delta m_{\text{discesa}} = 1 - 0.5138 = 0.4862$$

Si suppone quindi che l'algoritmo, influenzato dalla sensibilità di 'fmincon', dai vincoli scelti e dalla configurazione dell'obiettivo di massimizzare la massa residua al touch down, non percepisca la spinta finale come rilevante.

Si potrebbe concludere da questa analisi che sarebbe preferibile lasciare la spinta costante nel tratto finale.

Per analisi successive si potrebbe:

- utilizzare un obiettivo diverso per il tratto finale e quindi due 'fmincon', di cui una per la discesa e una per l'atterraggio.
- aumentare la sensibilità dell'algoritmo con ConstraintTolerance, sebbene probabilmente i tempi di calcolo diventerebbero eccessivi.

Per l'atterraggio di 10s le considerazioni sono analoghe a quelle appena fatte, e i risultati sono gli stessi a meno di un decimale della massa finale, che è di 811.03 kg.

Spinta variabile nella discesa al 95%Tmax (spinta_var_discesa.m)

In questa sezione si realizza un'analisi con spinta variabile tra 95%Tmax e Tmax nel tratto di discesa, mantenendo spinta costante nel tratto di atterraggio.

Viene svolta l'analisi sia con 5s che con 10s di atterraggio verticale; la spinta iniziale viene adimensionalizzata per evitare problematiche di ordini di grandezza inconsistenti.

Tabella 6: Parametri dell'analisi con spinta variabile nella discesa

Parametro	Valore
Tipo di analisi	Spinta di discesa variabile
Variabili di controllo	Angoli (uguess+u1guess), spinta di discesa, tempi (tf e tf1)
Vincoli	$V_z=0$ a $z=0$, $V_x=0$ a $t=5s$, $tf < tf1$
Limiti sulla spinta [/]	problems.spintaMin = 3600 (minimo)
	problems.spintaMax = 4000 (massimo)
Limiti sul tempo tf [/]	problems.tfMin = 30 (minimo)
	problems.tfMax = 40 (massimo)
Limiti sul tempo tf1 [/]	problems.tf1Min = 36 (minimo)
	problems.tf1Max = 41 (massimo)
Limiti controlli discesa[rad]	problems.controlMin = [2] (minimo)
	problems.controlMax = [π] (massimo)
Limiti controlli atterraggio[rad]	problems.control1Min = [1.4] (minimo)

	problems.control1Max = [1.7] (massimo)
Stima spinta discesa [N]	Problems.spintaguess=[3900; 3700]
Stima dei controlli discesa [rad]	problems.uguess(:, 1) = [2.4; 2.95]
Stima dei controlli atterraggio [rad]	problems.u1guess(:, 1) = [1.65; 1.56]
Stima del tempo di discesa [/]	problems.tguess = [0; 30]
Stima del tempo di atterraggio [/]	problems.t1guess = [0; 40]
Numero di nodi	problems.nodes = 110

Inoltre, dato che la spinta adimensionalizzata ha un valore dell'ordine di grandezza di 10^{-2} , si è scelto di aumentare la rigidità del codice ai vincoli imposti ponendo un *constraint tolerance* e uno *step tolerance* di 10^{-6} .

```

optODE = odeset('RelTol', 1e-3, 'AbsTol', 1e-3);
problem.optODE = optODE;

% 9. Opzioni per l'ottimizzazione
options = optimoptions('fmincon', ...
    'Algorithm', 'sqp', ...
    'ConstraintTolerance', 1e-6, ...
    'StepTolerance', 1e-6, ...
    'Display', 'iter-detailed', ...
    'MaxIterations', 2000, ...
    'OptimalityTolerance', 1e-4, ...
    'FunctionTolerance', 1e-4);

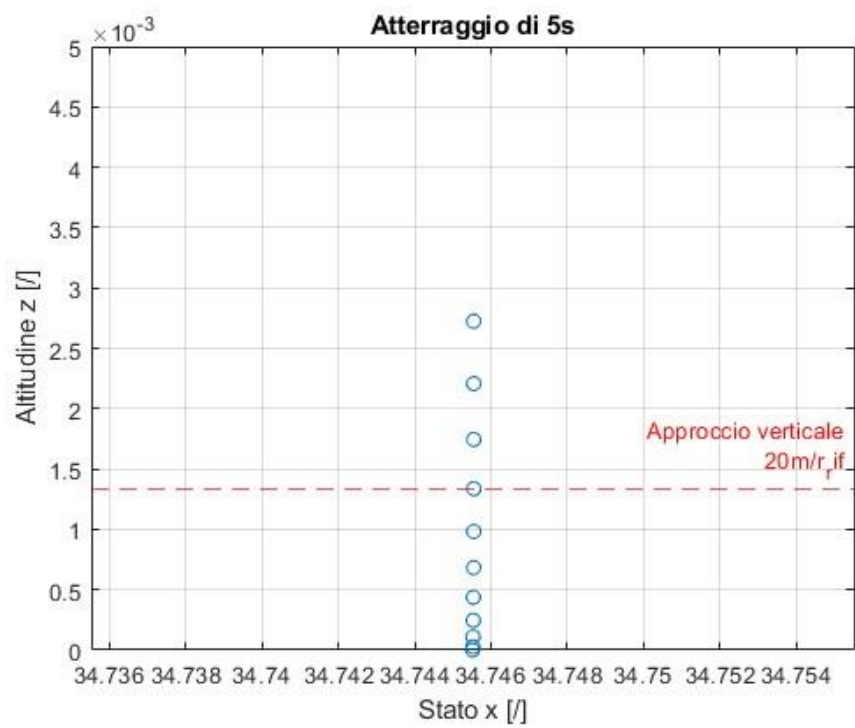
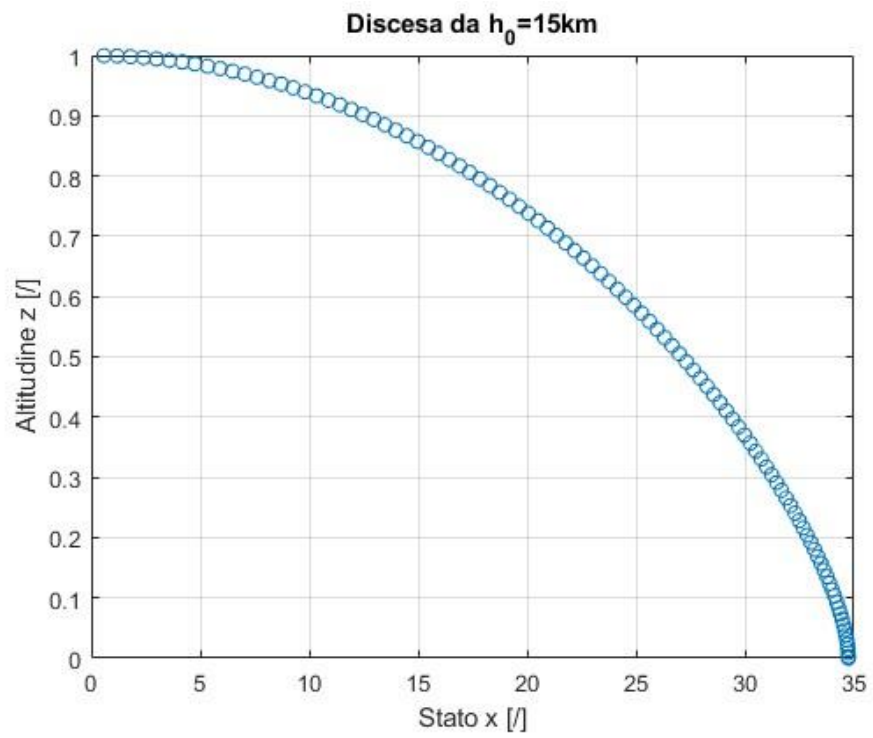
```

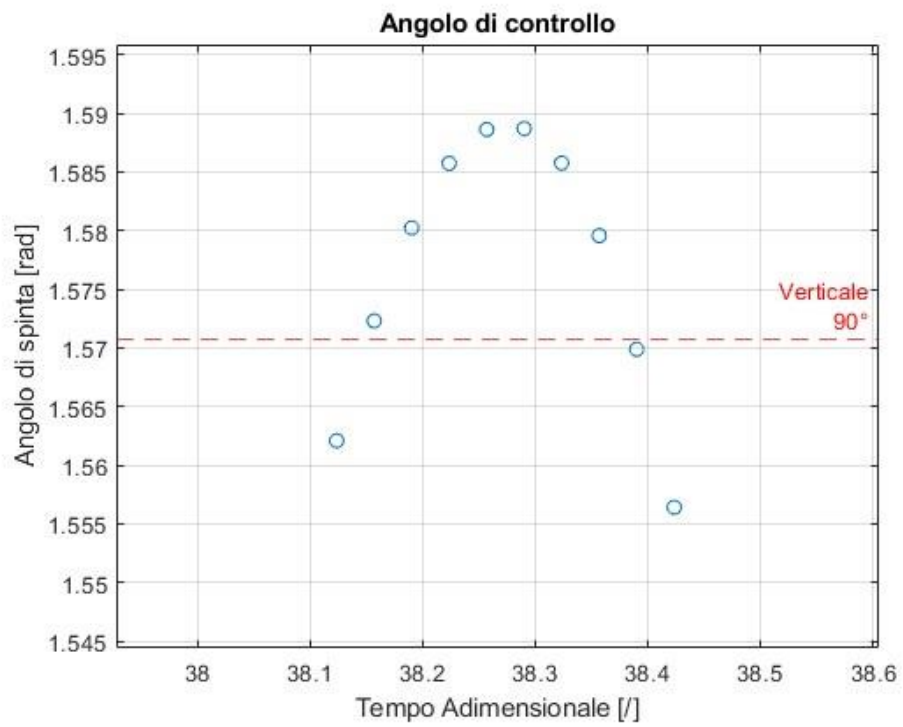
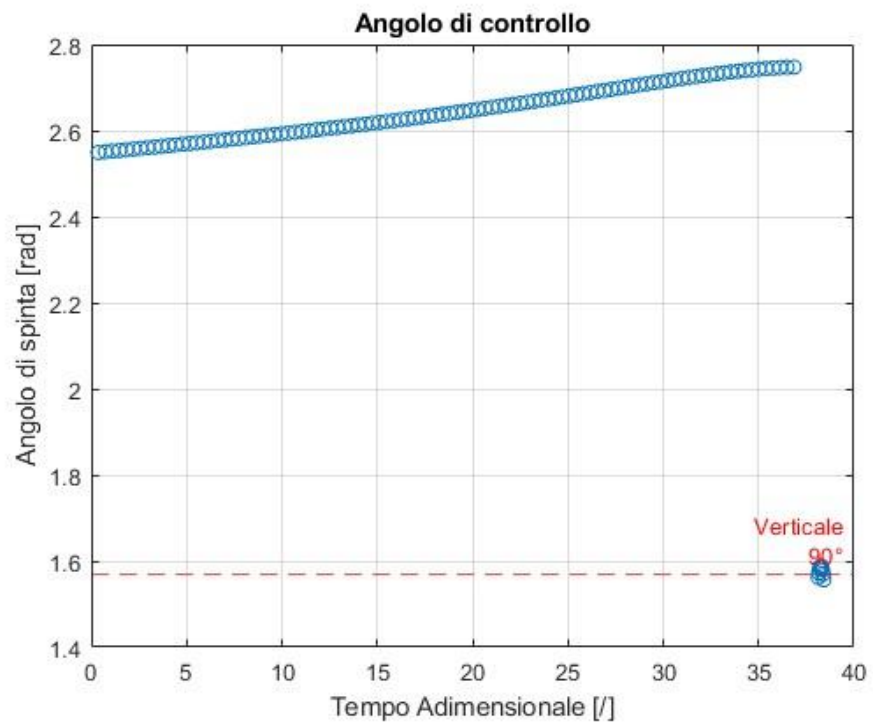
Di seguito vengono riportati i risultati per l'analisi con 5s:

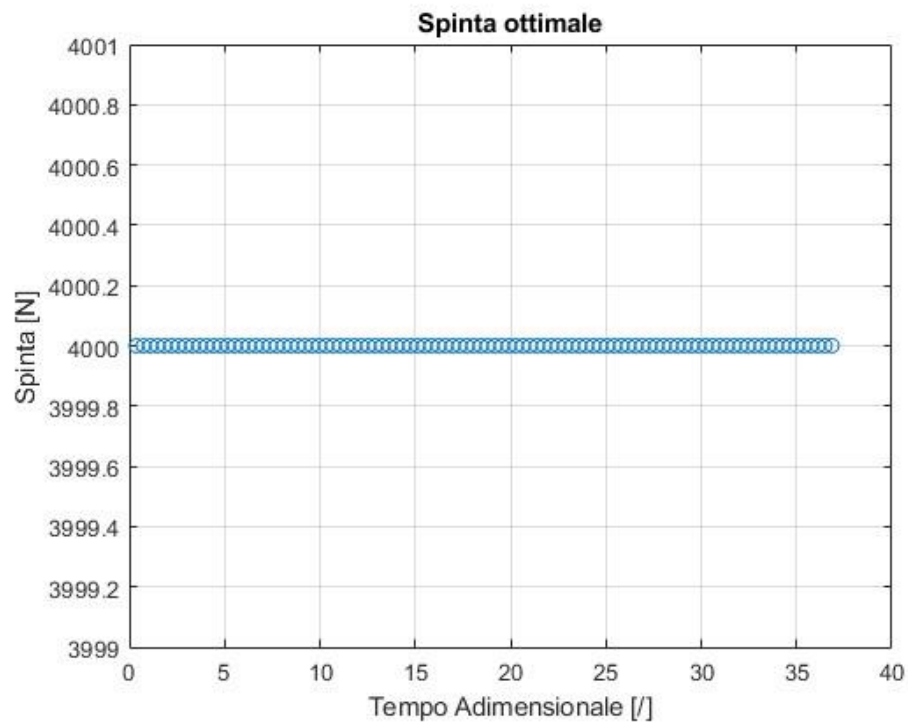
La massa finale è:	815.67 kg
Il tempo di discesa è:	553.15 s
Il tempo di touchdown è:	558.15 s

Anche in questo caso variare la spinta è superfluo. Regolare la spinta non è vantaggioso. È preferibile discendere, da 15km fino all'inizio dell'atterraggio verticale, mantenendo la spinta al valore costante e massimo, $T_{max}=4000N$.

Vengono riportare di seguito le immagini dei risultati.



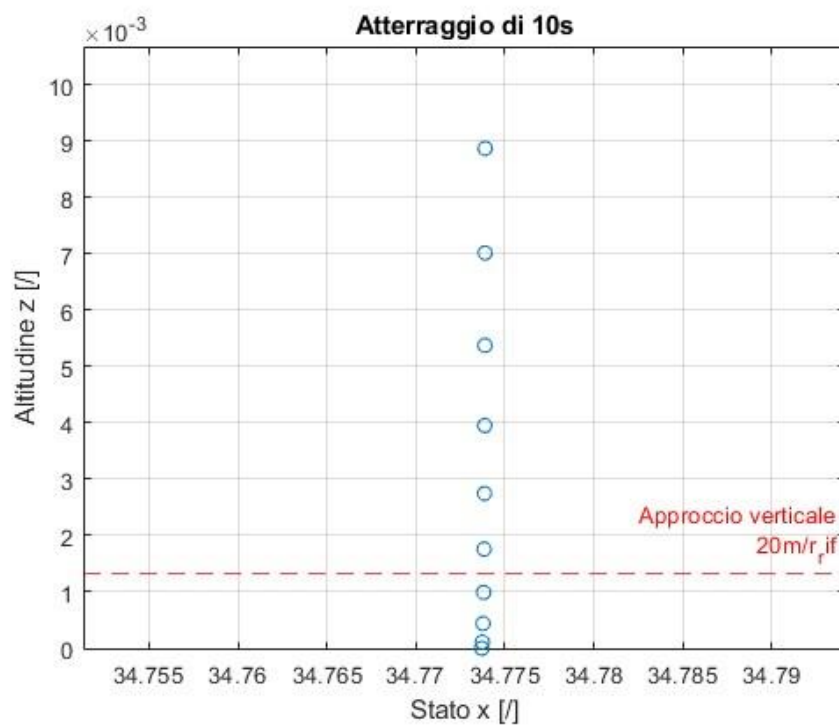
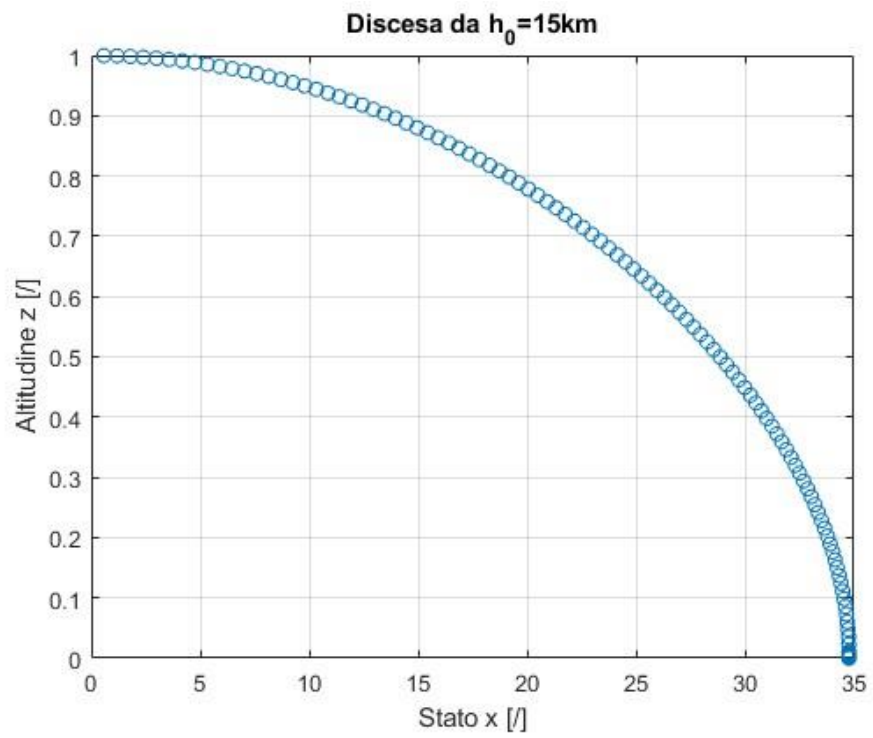


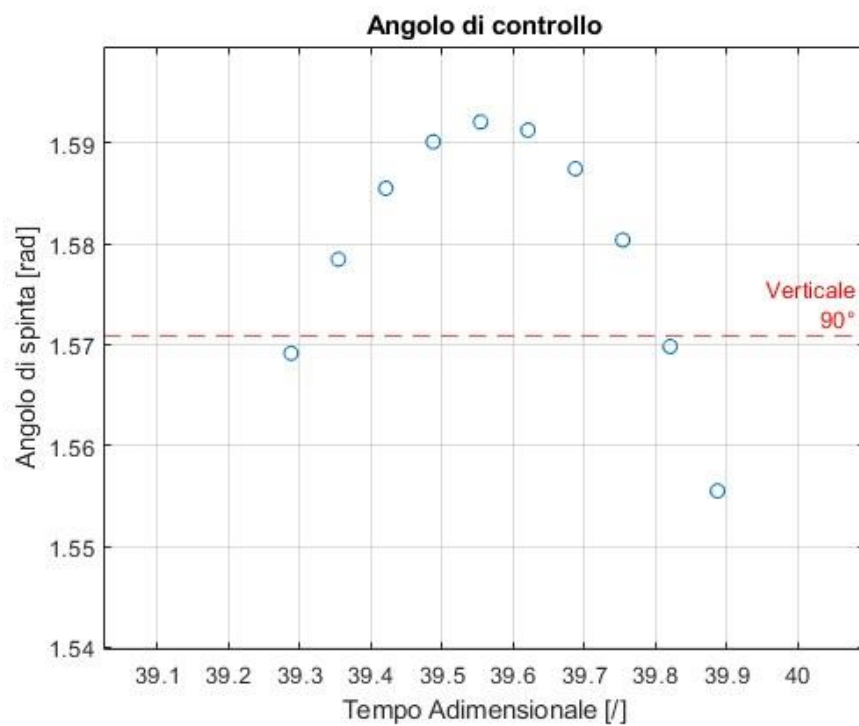
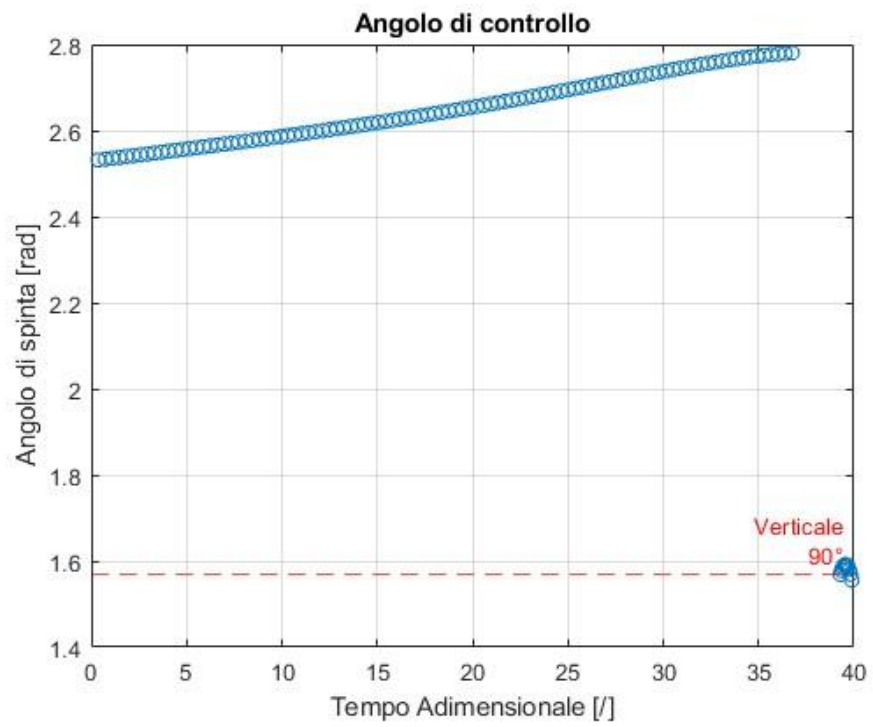


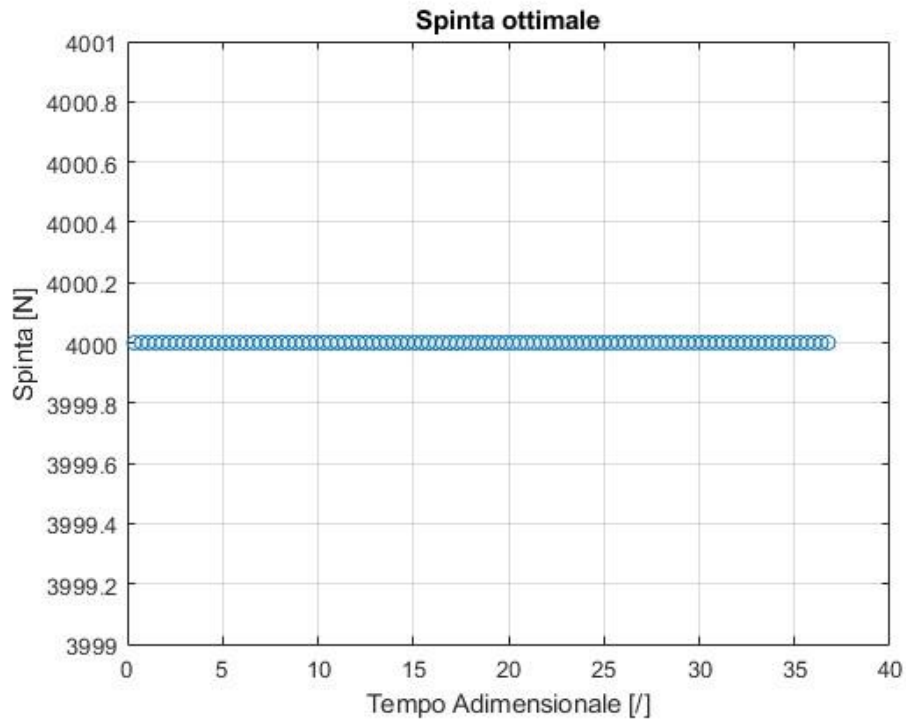
È possibile commentare che, nonostante la possibilità di variare la spinta tra 3800 N e 4000 N, conviene mantenere la spinta costante al valore massimo concesso nel range di spinta T_{max} . Di conseguenza regolare la spinta in fase di discesa non ha, secondo questo algoritmo, particolare vantaggio a livello di consumi.

Di seguito viene ripetuta l'analisi con 10s e vengono riportati i risultati, con uguali guess iniziali rispetto al caso precedente:

La massa finale è:	810.18 kg
Il tempo di discesa è:	551.55 s
Il tempo di touchdown è:	561.55 s



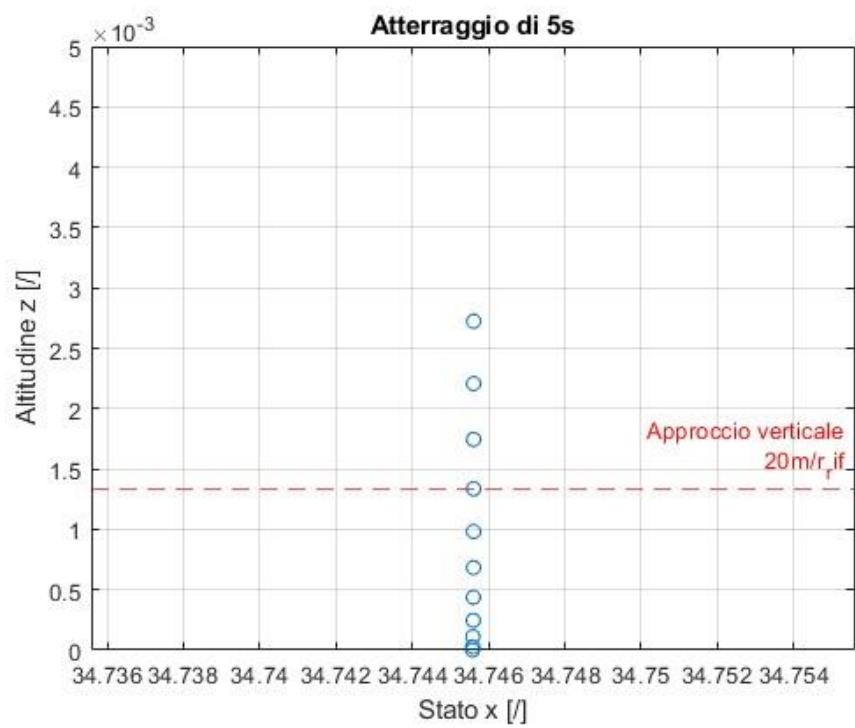
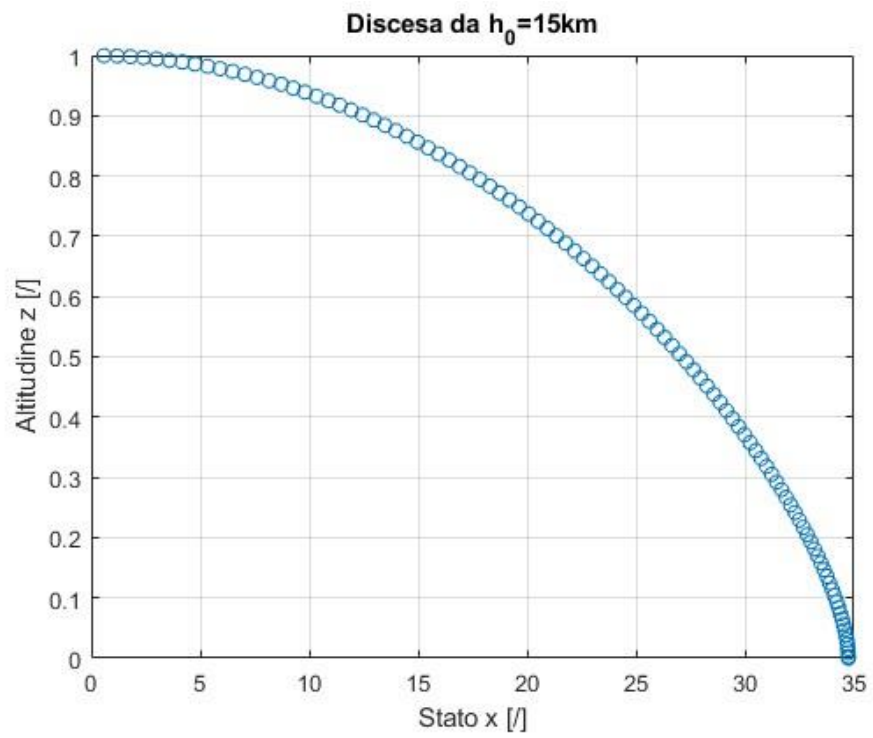


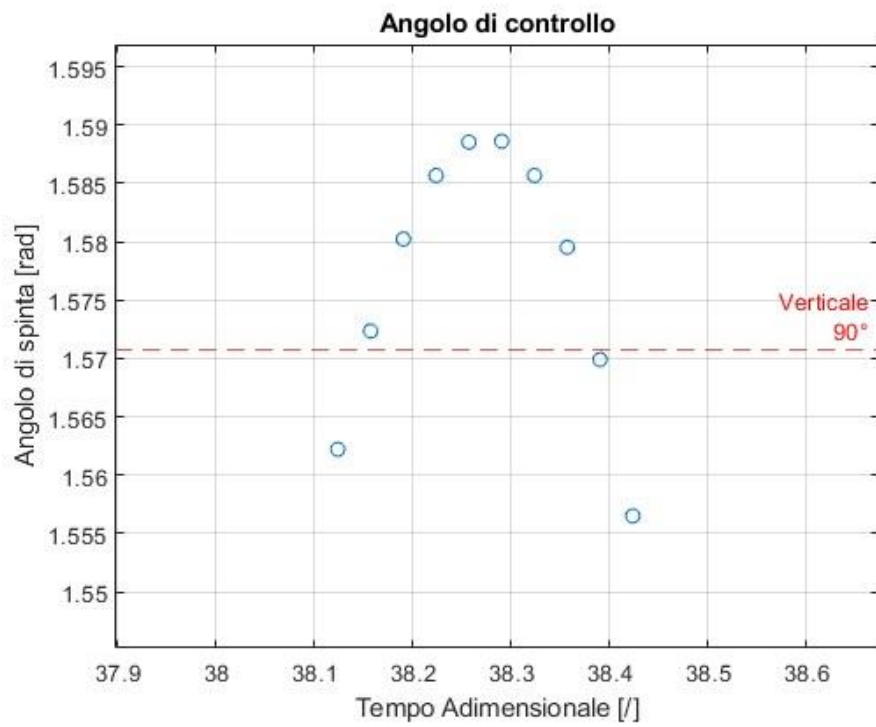
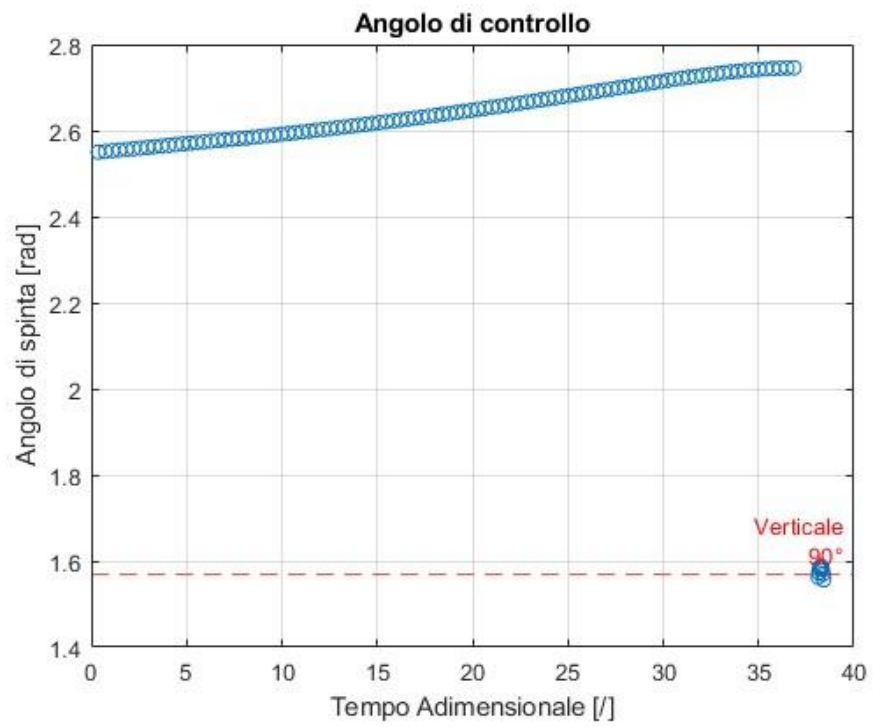


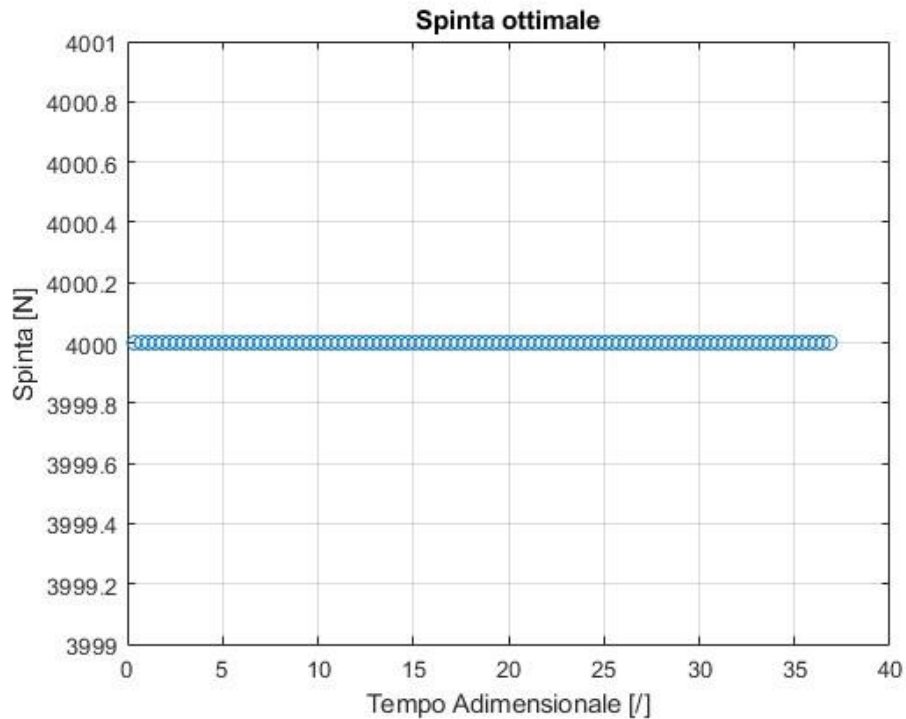
Anche con guess iniziali diversi dalle precedenti, viene dimostrato sempre che è ottimale percorrere la discesa con la spinta massima costante; sono stati eseguiti tentativi con con guess iniziale sulla spinta compreso tra [3850N e 3950N], al posto che [3900N e 3900N], e non sono state evidenti variazioni.

Allargando i limiti di variazione della spinta al 90% della spinta massima si ottengono sempre gli stessi risultati per la spinta ottimale, ovvero che nella fase di discesa convenga T_{max} . Viene riportata nelle seguenti immagini l'analisi con atterraggio di 5s. Si perviene a dei risultati leggermente diversi per via di piccole differenze dovute al calcolo numerico, in quanto non sono evidenti variazioni di comportamento.

La massa finale è:	815.28 kg
Il tempo di discesa è:	552.92 s
Il tempo di touchdown è:	557.92 s



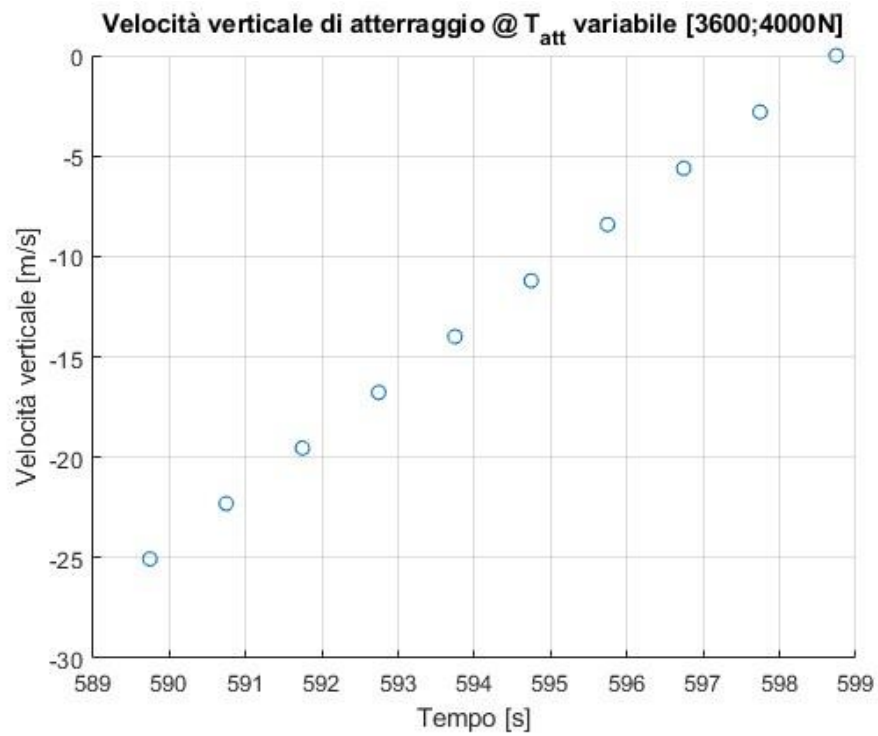




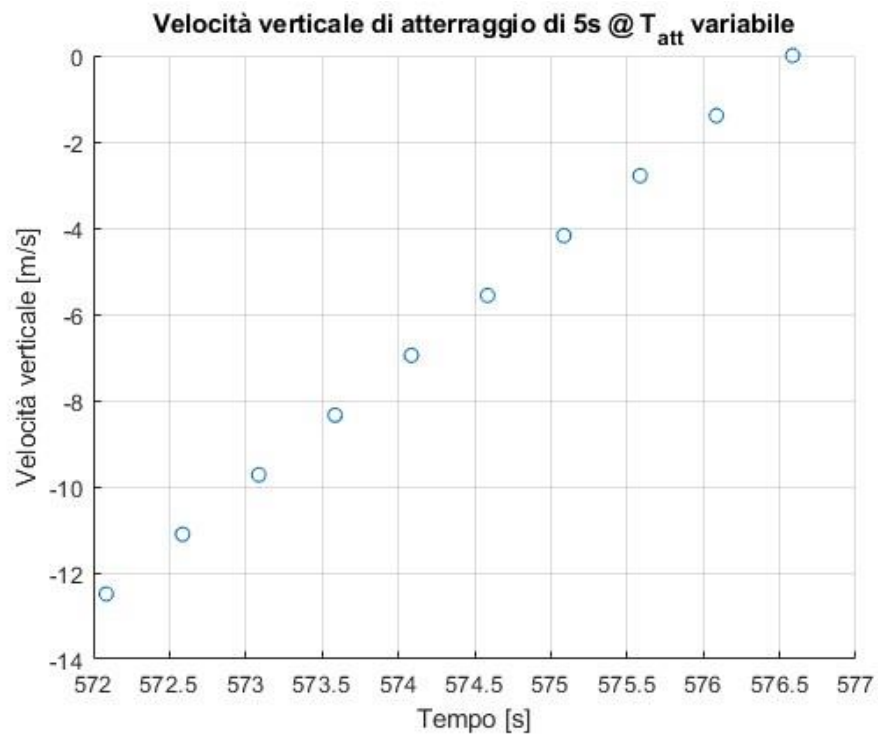
Confronto delle velocità di atterraggio

Nel problema del soft landing, è di vitale importanza la velocità di atterraggio verticale, che deve essere inferiore ai 5m/s per evitare danneggiamenti. Nel seguente paragrafo, si vuole soffermare l'attenzione su come vari la componente z della velocità di atterraggio, negli ultimi 10 intervalli di discretizzazione. Vengono quindi riportati dei grafici relativi alle analisi precedentemente effettuate, per capire come le variabili di ottimizzazione scelte nelle prove influiscano sulla rapidità dell'allunaggio.

L'analisi della tabella 5, con spinta variabile tra 3600N e 4000N nella fase di atterraggio, che aveva prodotto come risultato una spinta ottimale di atterraggio costante a T_{min} , produce una velocità z per gli ultimi 10s in figura.

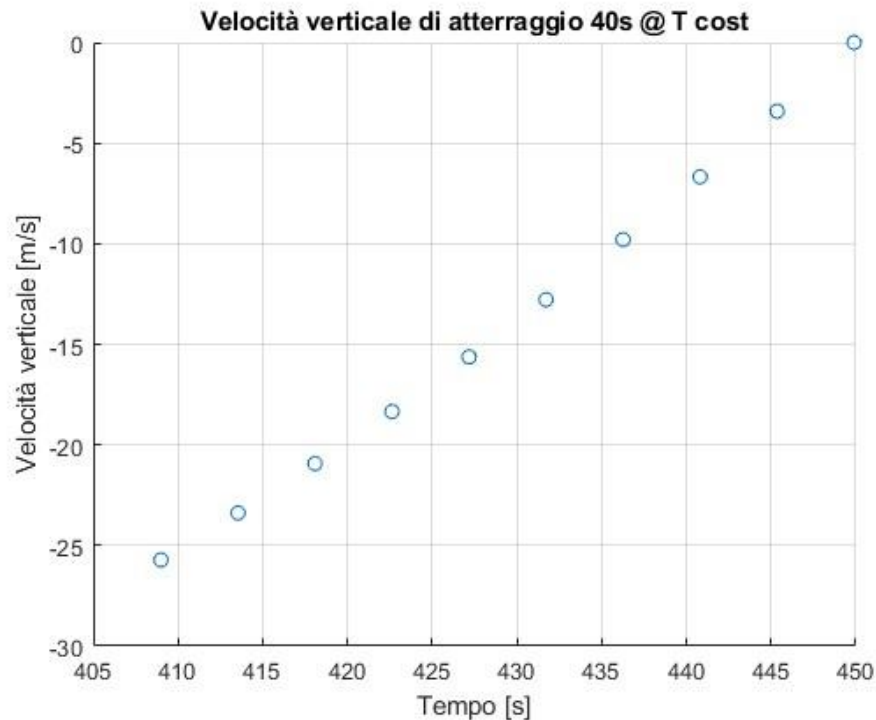


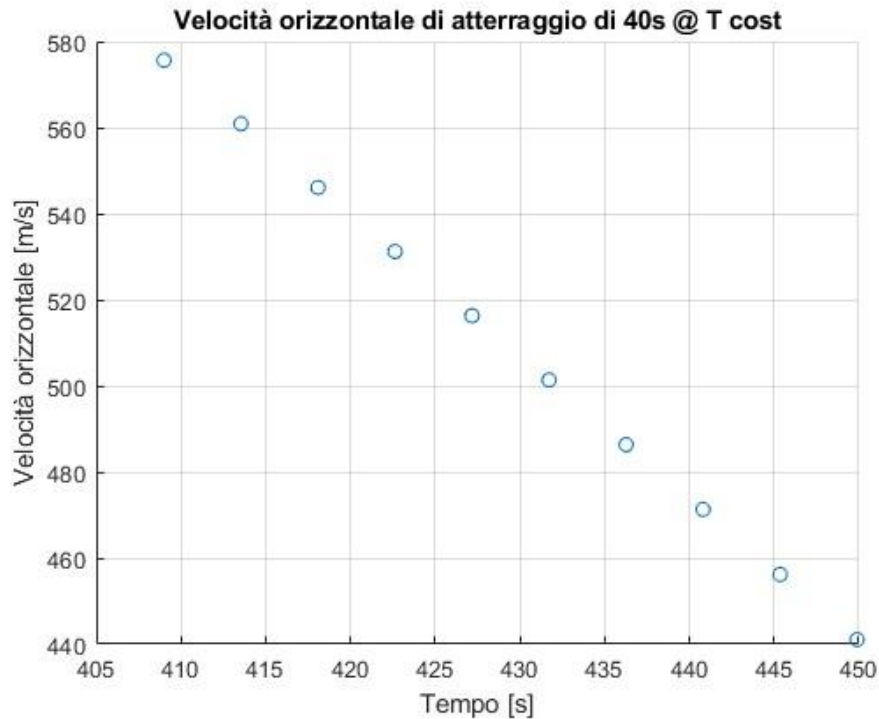
La medesima analisi, con atterraggio di 5s, fornisce invece velocità di atterraggio minori e quindi, dal punto di vista del soft landing, è preferibile atterrare con una traiettoria verticale più breve, in quanto la discesa sarà meno veloce.



Se invece si osserva la velocità finale nella analisi in tabella 6, in cui la spinta di discesa viene lasciata variabile, si ottengono risultati molto vicini a quelli precedenti perché il codice trova come spinta ottimale proprio quella massima, T_{max} . Di conseguenza, la spinta di discesa è costante a T_{max} , mentre l'atterraggio è fissata costante a 4000N per ipotesi, diversamente da prima che era a 3600N. Si perviene quindi a un risultato sulla velocità z analogo, in quanto è solo la spinta finale a variare tra le due prove.

Questi risultati sulla velocità, invece, si discostano da quelli ottenuti con la prima approssimazione della traiettoria con una sola discesa priva di atterraggio verticale. In tal caso, ci si riferisce alla tabella 2, con i cui input si perviene a una velocità che è distribuita sugli ultimi dieci intervalli in ben 40s, perché la traiettoria era stata discretizzata in 100 nodi, senza atterraggio finale. L'atterraggio non è verticale, ma che fornisce velocità verticali comunque basse. Il problema risiede nelle velocità orizzontali eccessive.





La scelta di lasciare libera da vincoli la velocità z all'inizio del tratto verticale potrebbe anche essere la spiegazione per cui l'algoritmo non dia importanza a questo tratto finale. Allunare perpendicolare al suolo viene percepito dal codice come una forzatura, la soluzione ottima non prevederebbe tale comportamento. Al posto di spingere a spinta uguale nelle due fasi, l'algoritmo preferisce frenare di più nella prima e usare poi una spinta minore nella seconda.

L'inserimento di un vincolo aggiuntivo sulla velocità z , all'inizio della fase di atterraggio, potrebbe migliorare i risultati del codice nella fase finale.

Tabella 7: Analisi con vincolo aggiuntivo

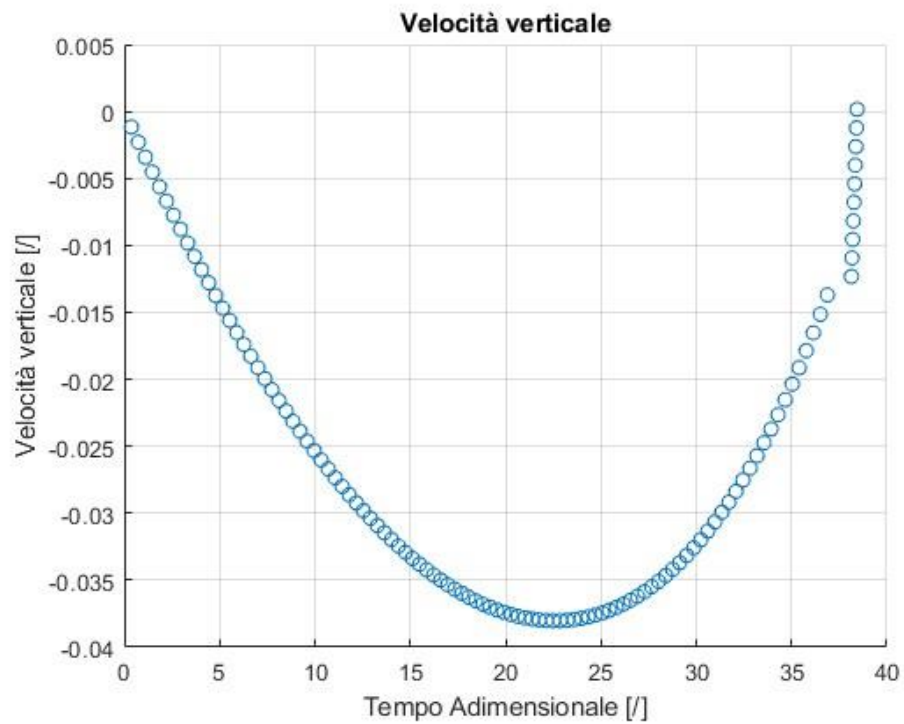
Parametro	Valore
Tipo di analisi	Spinta di atterraggio variabile
Variabili di controllo	Angoli ($u_{guess}+u_1$ guess), spinta di atterraggio, tempi (t_f e t_{f1})
Vincoli	$V_z=0$ a $z=0$, $V_x=0$ e $V_z=5\text{m/s}$ a $t=5\text{s}$, $t_f < t_{f1}$
Limiti sulla spinta [/]	problems.spintaMin = 3600 (minimo) problems.spintaMax = 4000 (massimo)
Limiti sul tempo t_f [/]	problems.tfMin = 30 (minimo) problems.tfMax = 40 (massimo)
Limiti sul tempo t_{f1} [/]	problems.tf1Min = 36 (minimo) problems.tf1Max = 41 (massimo)
Limiti controlli discesa[rad]	problems.controlMin = $[2]$ (minimo) problems.controlMax = $[\pi]$ (massimo)
Limiti controlli atterraggio[rad]	problems.control1Min = $[1.4]$ (minimo)

	problems.control1Max = [1.7] (massimo)
Stima spinta atterraggio [N]	Problems.spintaguess=[3900; 3700]
Stima dei controlli discesa [rad]	problems.uguess(:, 1) = [2.4; 2.95]
Stima dei controlli atterraggio [rad]	problems.u1guess(:, 1) = [1.65; 1.56]
Stima del tempo di discesa [/]	problems.tguess = [0; 30]
Stima del tempo di atterraggio [/]	problems.t1guess = [0; 40]
Numero di nodi	problems.nodes = 110

Di seguito viene presentato il risultato con l'imposizione del vincolo aggiuntivo:

$$V_z = 5 \frac{m}{s} \text{ al tempo di inizio atterraggio}$$

L'andamento della velocità risulta più continuo dopo l'imposizione del vincolo, ma non si hanno variazioni significative a livello di andamenti del controllo o della spinta ottimale.



CONCLUSIONI

La realizzazione dell'algoritmo per l'ottimizzazione della discesa sulla luna, con il metodo diretto, ha fatto emergere la sensibilità di questo ai valori iniziali di tentativo. Di conseguenza, benché l'ottimizzatore nasca con l'obiettivo di fornire risultati quantitativi e costituisca un ottimo punto di partenza, al momento non risulterebbe del tutto affidabile in quanto ne andrebbe aumentata la robustezza. I risultati evidenziano, comunque, angoli di controllo e tempistiche realistiche di atterraggio. La regolazione della spinta non risulta conveniente al fine di ridurre i consumi, e, al netto di tutte le analisi, converrebbe una traiettoria di discesa costante alla spinta massima. Le scelte sui vincoli imposti alla traiettoria, potrebbe anche essere la spiegazione per cui l'algoritmo non dia importanza alla fase di atterraggio finale. Allunare perpendicolare al suolo viene percepito dal codice come una forzatura, la soluzione ottima non prevederebbe tale comportamento. Al posto di spingere a spinta uguale nelle due fasi, l'algoritmo preferisce frenare di più nella prima con T_{max} e usare poi una spinta minore, T_{min} , nella seconda.

In futuro, l'algoritmo può essere migliorato utilizzando funzioni di Matlab più complesse, come *Multistart*. Questa funzione quando viene eseguita, riesce a trovare più soluzioni locali del problema, partendo da punti diversi. In questo modo sarebbe possibile superare il limite intrinseco di *Fmincon*, ovvero la sua sensibilità alle variabili di tentativo. Initial guesses errate possono portare a una convergenza lenta o a risultati subottimali.

Altri spunti di miglioramento potrebbero essere l'inclusione di *gates*, cioè punti lungo la discesa in cui deve venire soddisfatto un certo vincolo. Se si disponesse di potenze di calcolo maggiori e se si effettuassero le migliorie necessarie, sarebbe possibile rendere l'analisi molto più aderente alla realtà.

LISTA DELLE FIGURE

Figura 1: Fase lunare, gibbosa crescente [23].....	10
Figura 2: Motori del Saturn V [24].....	12
Figura 3: Decollo del SLS, 16 Novembre 2022 [25].....	15
Figura 4: Immagini iconografiche sul programma Artemis, fonte NASA [26]	19
Figura 5: Interpretazione geometrica delle condizioni ottimali.	29
Figura 6: Gradienti e ragioni di fattibilità.....	30
Figura 7: Vincoli attivi e inattivi.....	32
Figura 8: Fase di atterraggio lunare per la discesa di Apollo.....	34
Figura 9: Sistemi di riferimento	36
Figura 10: Struttura dell'algoritmo	40

LISTA DELLE TABELLE

Tabella 1: Variabili globali.....	39
Tabella 2: Parametri della analisi con spinta costante.....	41
Tabella 3: Modifica ai parametri della analisi con spinta costante	43
Tabella 4: Analisi con atterraggio verticale	45
Tabella 5: Analisi con spinta variabile.....	51
Tabella 6: Parametri dell'analisi con spinta variabile nella discesa	63
Tabella 7: Analisi con vincolo aggiuntivo	76

BIBIOGRAFIA

- [1] Fabrizio Stesina. *Attitude and Orbit Control System (AOCS) / Part 2 Sistemi Aerospaziali*. Codice 03GKZMT, DIMEAS.
- [2] Michael T. Heath. *Scientific Computing: An Introductory Survey*.
- [3] Lorenzo Casalino. *Access to Space and Advanced Rockets*. Politecnico di Torino.
- [4] Manoj Srinivasan. *Trajectory Optimization, a Brief Introduction*. Dipartimento di Ingegneria Meccanica, Ohio State University. Tutorial presentato al *Dynamic Walking 2010*, Media Lab, Massachusetts Institute of Technology, 8 luglio 2010. Disponibile su: <http://movement.osu.edu>.
- [5] Davide Fabro. *Optimization of Lunar Descent Trajectories with Direct Methods*. Relatore: Lorenzo Casalino. Corso di Laurea Magistrale in Ingegneria Aerospaziale, Politecnico di Torino, 2022.
- [6] Marco Grammatico. *Fuel-Optimal Lander Trajectory for Lunar Soft-Precision Landing*. Relatore: Lorenzo Casalino. Corso di Laurea Magistrale in Ingegneria Aerospaziale, Politecnico di Torino, 2021.
- [7] NASA. *View the Best Images from NASA's Artemis I Mission*. Disponibile su: <https://www.nasa.gov/humans-in-space/view-the-best-images-from-nasas-artemis-i-mission/>.
- [8] Wikipedia. *Luna*. Disponibile su: <https://it.wikipedia.org/wiki/Luna>.
- [9] European Space Agency. *Esplorazione lunare*. Disponibile su: https://www.esa.int/kids/it/imparare/L_Universo/Pianeti_e_satelliti/Esplorazione_lunare#:~:text=Hanno%20lasciato%20esperimenti%20scientifici%20sulla%20superficie%20e%20hanno,hanno%20deciso%20di%20riprendere%20le%20esplorazioni%20della%20Luna.
- [10] European Space Agency. *Moon Village*. Disponibile su: https://www.esa.int/About_Us/Ministerial_Council_2016/Moon_Village.
- [11] ResearchGate. *Fuel Optimum Mission Design for Lunar Soft Landing at a Specified Location: Comprehensive Review*. Disponibile su: https://www.researchgate.net/publication/365870186_Fuel_Optimum_Mission_Design_for_Lunar_Soft_Landing_at_a_Specified_Location_Comprehensive_Review.
- [12] MathWorks. *fmincon - Find Minimum of Constrained Nonlinear Multivariable Function*. Disponibile su: <https://it.mathworks.com/help/optim/ug/fmincon.html>.
- [13] Bing Video. *Nonlinear Constrained Optimization Using MATLAB's fmincon*. Pubblicato da Anuradha Viswanathan, 20 maggio 2022. Disponibile su: <https://www.bing.com/videos/riverview/relatedvideo?q=sequential+quadratic+programming+matlab+fmincon&&view=riverview&mmscn=mtsc&mid=B2ED50B51102FE381284B2ED50B51102FE381284&&aps=381&FORM=VMISOVR>.

- [14] MathWorks. *Optimization Toolbox*. Disponibile su: <https://it.mathworks.com/products/optimization.html>.
- [15] MathWorks. *ODE45 - Solve Nonstiff Differential Equations*. Disponibile su: <https://it.mathworks.com/help/matlab/ref/ode45.html>.
- [16] MathWorks. *ODE113 - Solve Nonstiff Differential Equations with Variable Step Size*. Disponibile su: <https://it.mathworks.com/help/matlab/ref/ode113.html>.
- [17] MathWorks. *Fmincon - Find minimum of constrained nonlinear multivariable function*. Disponibile su: <https://it.mathworks.com/help/optim/ug/fmincon.html>
- [18] MathWorks. *MultiStart - Solve Optimization Problems from Multiple Starting Points*. Disponibile su: <https://it.mathworks.com/help/gads/multistart.html>.
- [19] Wikipedia. *Sequential Quadratic Programming (SQP)*. Disponibile su: https://en.wikipedia.org/wiki/Sequential_quadratic_programming#:~:text=Sequential%20quadratic%20programming%20%28SQP%29%20is%20an%20iterative%20method,are%20twice%20continuously%20differentiable%2C%20but%20not%20necessarily%20convex.
- [20] MathWorks. *Constrained Nonlinear Optimization Algorithms*. Disponibile su: <https://it.mathworks.com/help/optim/ug/constrained-nonlinear-optimization-algorithms.html>.
- [21] Cornell University. *Sequential Quadratic Programming*. Disponibile su: https://optimization.cbe.cornell.edu/index.php?title=Sequential_quadratic_programming.
- [22] NASA. *Apollo 11 Mission Overview*. Disponibile su: <https://www.nasa.gov/mission/apollo-11/>.
- [23] Fausto Pino, astrofotografo, Foto della Luna 2024
- [24] Immagine dei motori del Saturn V, Cronache dal Silenzio. *Quanto era potente il motore del Saturn V?*
- [25] Il Sole 24 Ore. *Il ritorno dell'uomo sulla Luna: la missione Artemis 1 spiegata con una infografica*. Disponibile su: <https://www.infodata.ilsole24ore.com/2022/08/29/il-ritorno-delluomo-sulla-luna-la-missione-artemis-1-spiegata-con-una-infografica/>.
- [26] ESA Multimedia, Artemis Program, Step by Step, Disponibile su: https://www.esa.int/ESA_Multimedia/Images/2020/05/Artemis_3_step-by-step
- [27] Rycroft, C. (2022). *AM205 Video 4.10 - Sequential Quadratic Programming*. Disponibile su: <https://youtu.be/O-pTuBTShc4>

RINGRAZIAMENTI

Vorrei ringraziare il Prof. Lorenzo Casalino, per avermi guidato e supportato nella implementazione dell'algoritmo e nella trattazione di un argomento così interessante.

Inoltre, ci tengo a ringraziare tutte le persone che mi son state vicine durante il mio percorso di studi.

Per primi, ringrazio mia mamma e mio papà. Mia mamma, professoressa di matematica, mi ha trasmesso l'amore per le materie scientifiche e mi ha sempre stimolato, fin da bambina, a tenere alta l'asticella dei miei obiettivi e delle mie aspettative. È sempre stata la prima persona a cui ho voluto comunicare il successo di una bella valutazione, dopo che, magari, la sera prima dell'esame, mi spegneva la luce della camera mentre studiavo, per l'amore di farmi andare a riposare. Mio papà è sempre stato una luce guida per ogni mia scelta di percorso. Mi rifugiavo nei suoi consigli ogni volta che, prendere una decisione da sola, era troppo difficile. Ha saputo indirizzarmi in ogni momento, senza imporsi, e lasciandomi sempre libera di fare quello che meglio credevo senza giudizi. Con la sua positività e il suo instancabile sorriso, mi ha sempre fatto ridere e tirato su il morale in ogni situazione.

Ringrazio poi i miei tutti i miei parenti; i miei fratelli più piccoli, Arianna e Federico, a cui spero di aver trasmesso un buon esempio. Ringrazio Zia e Nonna Silvana, i cui occhi illuminati alla notizia della mia laurea, mi hanno commosso e ripagato di ogni sforzo. Far contente loro è la mia più grande ricompensa.

Vorrei ringraziare Nonna Carla, che purtroppo non c'è più, ma è stata fondamentale, sempre, e gli ultimi anni, benché non sentisse più, mi ascoltava ripetere, ore ed ore, qualsiasi materia pur di aiutarmi. So che è qui con me oggi.

Vorrei ringraziare il mio ragazzo, che mi regala ogni giorno la felicità di sentire che non mi manca più niente quando sono insieme a lui. Ringrazio Agnese, la mia più cara amica, infinitamente, per esserci vicine ogni giorno. Ringrazio poi Carlotta, con me fin dalla prima elementare, e Simona e Camilla, che mi fanno sorridere dalla prima media. Un ringraziamento va anche a Step, ormai dottoranda al Politecnico di Torino, Nicholas, Niccolò, Corinna e tutti i miei compagni di università e di liceo. Ringrazio, poi, senza citarli tutti, i miei amici ed amiche dalla pallavolo, da San Giusto e anche quelli conosciuti al mare. Ognuno di loro mi ha trasmesso qualcosa di speciale. Spero di condividere ancora tanti momenti felici con voi.

