

POLITECNICO DI TORINO

SCHOOL OF ENGINEERING

Master's Degree Course in Aerospace Engineering

Master's Degree Thesis

**Fuel-Optimal Lander Trajectory for
Lunar Soft-Precision Landing**

Optimal Control Theory



Supervisor

Prof. Lorenzo Casalino

Candidate

Marco Grammatico

A.Y. 2020 - 2021

This work is subject to the Creative Commons Licence

Abstract

A recent and renewed interest for Moon exploration is currently fostering the development and improvement of many technologies in the space industry. Future lunar missions will call for precise and robust performance of guidance systems in order to land both robots and humans safely and accurately on the Moon's surface. To achieve a successful landing, a rigorous and well-integrated Guidance, Navigation and Control (GN&C) system must be employed. In particular, the landing guidance system is required to transfer a spacecraft from an orbit to a desired target on the planetary surface with a soft approach while meeting several constraints necessary for safe operations. In addition, due to current launch vehicles limitations, it is evident that minimizing the usage of propellant would make possible to carry more payload on-board. On top of that, the landing site may change in real-time in order to avoid previously undetected hazards, which gradually become ever more visible while getting closer to the surface. The Apollo program relied heavily on astronauts' eyesight and the avoidance maneuvers were performed through manual control. However, for missions with poor lighting conditions, like in lunar polar regions, such approach would make things more arduous; for robotic missions, this is off the table.

This thesis studies the lunar powered descent guidance trajectory taking inspiration from Apollo Lunar Module concept of operations, taking into account a fuel-optimization strategy, while constraints on the initial state, final state, and bounded control variables must be considered and solidly met. An indirect optimization method based on *Pontryagin's Maximum Principle* is applied to find the optimal guidance laws. Then the resulting *Boundary Value Problem* is numerically solved to get the solution for a group of test cases, in which both downrange and crossrange variations are illustrated to simulate possible diversion maneuver.

Table of Contents

List of Tables	III
List of Figures	IV
1 Introduction to Moon Exploration	1
1.1 Historical Overview	1
1.1.1 Pioneering Forays Into Space	1
1.1.2 Humans on the Moon	2
1.1.3 Rediscovered Interest	4
1.2 Recent Missions and Future Perspectives	4
1.2.1 European Glance	5
1.2.2 A Worldwide Ambition	6
1.2.3 The Futuristic Roadmap	8
2 Powered Descent and Landing	11
2.1 State of Art and Trends	11
2.2 Mission Description	14
3 Optimal Control Problem	17
3.1 What is Optimization	17
3.2 Survey of Methods	19
3.3 Optimal Control Theory	21
3.4 Boundary Value Problem	27
4 Model Definition	31
4.1 Assumptions	31
4.2 Objective Function	33
4.3 Equations of Motion	34
4.4 Controls	35
4.5 Boundary Conditions and Constraints	36
5 Application of Optimal Control	38
5.1 Two-Dimensional Motion Case	42

6 Results	46
6.1 Cases with Reference Values	47
6.1.1 No Crossrange Cases - Comparison	50
6.1.2 Crossrange Cases - Comparison	59
7 Conclusions	70
Bibliography	72

List of Tables

6.1	Scaling Factors	47
6.2	Parameter and Variables of the BVP	48
6.3	B.C. and Initial Tentative Variables	48
6.4	Test Cases	49
6.5	Final Mass - Comparison	69
6.6	Propellant Consumption - Comparison	69

List of Figures

1.1	Lunar Roving Vehicle (LRV), also called Moon buggy, is an electric vehicle designed to expand astronauts' range of exploration on the low-gravity lunar surface	3
1.2	Life on Mars (<i>concept art</i>) - credits by NASA	7
1.3	Explorers on the moon (<i>concept art</i>) - by David Hardy	10
2.1	GNC landing system architecture with HDA in-the-loop [16]	13
2.2	Main Phases from Circular Orbit to Landing Site	14
2.3	Reference Scenario (<i>not in scale</i>) [2]	16
4.1	Downrange and Crossrange	34
4.2	Flight Reference System	35
4.3	Ground Reference System	36
5.1	Thrust Reference Frame	38
5.2	Lander 3D-Model	41
5.3	X-Z plane of Motion	42
6.1	V_z as a function of time - <i>NoCross</i> Comparison	52
6.2	V_x as a function of time - <i>NoCross</i> Comparison	53
6.3	R_z as a function of R_x (2-D) - <i>NoCross</i> Comparison	54
6.4	R_z as a function of R_x (3-D) - <i>NoCross</i> Comparison	55
6.5	Mass m as a function of time - <i>NoCross</i> Comparison	56
6.6	Thrust angle ψ as a function of time - <i>NoCross</i> Comparison	57
6.7	Thrust angle ψ and ϕ and their respective R.S.	58
6.8	V_z as a function of time - <i>Cross</i> Comparison	60
6.9	V_y as a function of time - <i>Cross</i> Comparison	62
6.10	V_x as a function of time - <i>Cross</i> Comparison	63
6.11	R_z as a function of R_x (2-D) - <i>Cross</i> Comparison	64
6.12	R_z as a function of R_x (3-D) - <i>Cross</i> Comparison	65
6.13	Mass m as a function of time - <i>Cross</i> Comparison	66
6.14	Thrust angle ψ as a function of time - <i>Cross</i> Comparison	67
6.15	Thrust angle ϕ as a function of time - <i>Cross</i> Comparison	68

1

Introduction to Moon Exploration

1.1 Historical Overview

For however long humans have looked skyward, the Moon has held and fed imagination for millennia. We have been always fascinated by our cosmic partner's mottled and cratered, wondering about its origins and impact on the universe.

To begin with, telescopes honed our perspectives of its bumps, ridges, and deserted lava seas. Then, at that point, we got closer, thanks to further advanced technologies like satellites and landers. At last, during the twentieth century, humans reached Earth's only natural satellite for the first time and saw its surface up close, marking a giant step for humankind. From that point forward, a lot of spacecrafts have investigated our nearest celestial neighbor, dipping low over its dusty fields and surveying its curious unknown sides. Presently, following sixty years of exploration, we are once again planning to send humans to the lunar surface.

1.1.1 Pioneering Forays Into Space

The earliest signs of interest for lunar exploration were born during Cold War, when the U.S. and Soviet Union started to send unmanned spacecraft to orbit and land on the moon. The Soviets scored a first success in January 1959, when Luna 1, a small sphere loaded with antennas conceived to be an impactor, turned into the first spacecraft to escape Earth's gravity. Albeit a glitch in the ground-based control system caused a fault in the upper stage rocket's burn time, and the spacecraft missed the Moon by 5900 km. At long last, in 1959, Luna 2 became the first spacecraft to

touch the moon's surface. The site where it crashed was the Mare Imbrium (*Sea of Rains*), an immense lava plain situated in the Imbrium Basin on the Moon, one of the bigger craters in the Solar System. That very year, a third Luna mission caught the first, even though blurry, images of the more remote side of the moon, which made possible notice that the rough highland terrain is strongly different from the smoother basins on the side facing the Earth. Then, the U.S. got into the game with nine NASA Ranger spacecraft that were launched between 1961 and 1965, and gave researchers the first close-up perspectives of the moon's surface. The Ranger missions were designed with spacecraft engineered to streak toward the moon and capture as many images as possible prior to crashing onto its surface. A series of accidents, be that as it may, led to the failure of the first six flights.

By 1964, images taken from Ranger 7 till Ranger 9, had enabled a more prominent insight about the moon's rough landscape and the likely difficulties of finding a smooth landing site for humans. In 1966, after 7 years from the first lunar surface contact-impact, the Soviet spacecraft Luna 9 is considered as the first vehicle to land safely on the lunar soil. Supplied with scientific instrumentation and com-systems, the small spacecraft took ground-level photos of lunar panorama. Soon thereafter, Luna 10 was launched, turning into the first spacecraft to effectively orbit the moon.

NASA likewise landed a spacecraft on the moon's surface that year with the first of its seven Surveyor spacecrafts, equipped with cameras to inspect the moon's surface and innovative technologies to analyze lunar soil. The Surveyor program actually proved that manned landings on the Moon were possible thanks to the collected results on lunar dust: if the dust was too deep, then no astronaut could land. Over the two years that followed, NASA scheduled five Lunar Orbiter missions that were planned to circle the moon and map its surface in anticipation of a definitive goal: landing astronauts on the surface. These orbiters have taken images of around 99% of the moon's surface, highlighting potential landing destinations and paving the way for a giant leap forward in space exploration [1, 7, 11].

1.1.2 Humans on the Moon

At that point, NASA was dealing with the fulfillment of a presidential promise: In 1961, President John F. Kennedy committed the United States in landing an individual on the moon before the decade was over. With that in mind, the Apollo space program was planned and launched, being undoubtedly the most costly spaceflight endeavor ever. The otherwise called Project Apollo ended in 1972, with a total of nine missions and 24 astronauts which orbited or touched the moon.

The most emblematic of those was Apollo 11, on which occasion humans had stepped on another world for the very first time: on July 20, 1969, Neil Armstrong and Edwin "Buzz" Aldrin landed in the Sea of Tranquility inside the Lunar Module Eagle (LM-5), while astronaut Michael Collins remained orbiting over the moon in the command module Columbia (CM-107). Armstrong, who impressed the first "*bootprints*" on moon's surface, famously stated: "That's one small step for a man, one giant leap for mankind". The pair remained on the moon's surface for 21 hours

and 36 minutes before rendezvousing with Collins and making their way back to the Earth [18]. Every mission after Apollo 11 set new landmarks in space travel and lunar exploration. Four months after the first humans got to the moon, Apollo 12 touched down successfully once again, this time performing an even more precise landing.

Apollo 13 barely avoided a near-catastrophe when on-board oxygen tanks exploded in April 1970, obliging the entire crew to abort the mission, missing the third planned moon landing. All three astronauts survived. During the following mission, which took place in January 1971 with Apollo 14, commander Alan Shepard set another record for the farthest distance traveled on the moon surface during the two planned walks, totalizing about 3 kilometers. He even threw and soared for miles two golf balls with the help of the light gravity of the moon and a makeshift 6-iron club. Apollo 15, launched in July 1971, was the first of the last three missions fit for a more extended permanence on the moon. Over the span of three days spent on the lunar surface, achievements included collecting few kilograms of lunar samples and traveling over 27 kilometers in the first piloted moon buggy (Fig. 1.1). Apollo 16 and Apollo 17 in 1972 were the two latest manned missions to the moon, and Russia's Luna-24 crewless spacecraft in 1976 was the last to land until the next century. The sample collection achieved during these lunar explorations led to a remarkable amounts of knowledge about the geology and historical formation of the Earth's natural satellite [1, 29].

After the striking achievements of the 1960s and 1970s, the major space organizations turned their focus somewhere else for quite a few years. Up until this point, only 12 humans (all Americans and all men) have set foot on the moon.

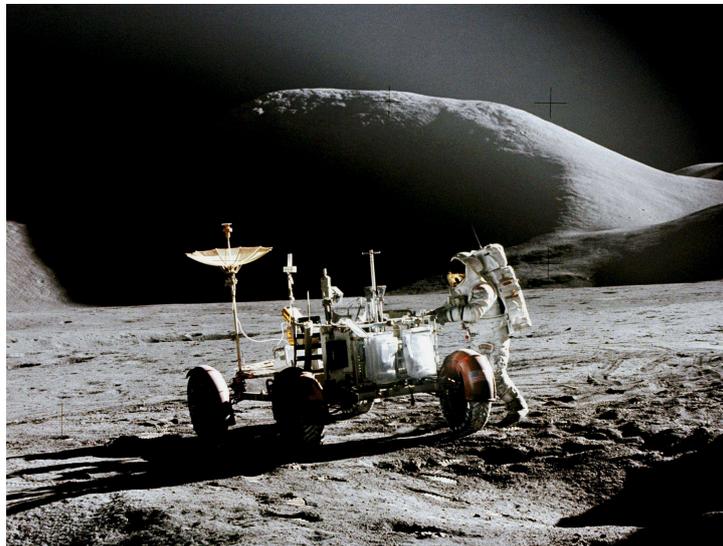


Fig. 1.1: Lunar Roving Vehicle (LRV), also called Moon buggy, is an electric vehicle designed to expand astronauts' range of exploration on the low-gravity lunar surface

1.1.3 Rediscovered Interest

It was not until 1994 that the moon returned into focus for the U.S., with a joint mission between NASA and the Strategic Defense Initiative Organization. The Clementine spacecraft mapped the moon's surface in wavelengths other than visible light, from ultraviolet to infrared. These observations were meant to obtain multi-spectral imaging of the entire lunar surface, assessing the surface mineralogy of the Moon and collecting data on gravity properties. A bistatic radar experiment was also performed to look for proof of lunar water at the Moon's poles. An initial analysis of the magnitude and polarization of the reflected signals suggested the presence of volatile ices, assuming then that water ice was contained. In 1999, the Lunar Prospector orbited the moon, confirming Clementine's first evidences of ice at the lunar poles, representing a vital asset that could be fundamental for any long-term lunar settlement. The mission's end was terrific: Prospector crashed into the moon, aiming to generate a plume that could be promptly inspected for evidence of water ice, however none was observed. Notwithstanding, after ten years, NASA's LCROSS (Lunar Crater Observation and Sensing Satellite) repeated this kind of analysis and discovered proof for water in a shadowed locale close to the moon's south pole, called Cabeus.

Since 2009, the Lunar Reconnaissance Orbiter (LRO) has taken high-resolution 3-D maps of the lunar surface at 100-meter resolution and 98.2% coverage (excluding polar areas in deep shadow), determining safe landing spots, finding plausible resources on the Moon, characterizing the radiation levels, and showing new advancements. LRO and LCROSS were launched as part of the United States's Vision for Space Exploration program. Afterward, Between 2011 and 2012, GRAIL (Gravity Recovery and Interior Laboratory) mission was launched, supported by two twin small spacecraft GRAIL A (Ebb) and GRAIL B (Flow), belonging to the wider NASA's Discovery Program. The two tandem spacecraft mapped the moon's gravitational field and its interior structure before intentionally crashing into a region near the lunar north pole. The knowledge acquired helped understand the history of the evolution of the terrestrial planets and improved computational capabilities in lunar orbits [1, 11].

1.2 Recent Missions and Future Perspectives

Till now it can be noticed that NASA was historically the most active space agency, especially regarding moon exploration, working and acting as a protagonist in the industry, but these days NASA is not the lone space agency with an uncommon surging interest in the moon. Within the most recent twenty years, lunar exploration grew into an undeniably international and even commercial interest, with public space agency as ESA (European Space Agency), the Japanese JAXA or the CSA (Canadian Space Agency) and also private space sector such as the American SpaceX.

1.2.1 European Glance

The European Space Agency is the main actor in the space industry when it comes to the old country. Established in 1975 and headquartered in Paris, ESA is an intergovernmental organization of 22 member states and has a worldwide cooperative staff. There are four different projects where ESA is involved in lunar exploration. The first and really significant is a collaboration with Russian space agency Roscosmos on a mission called Lunar Resource Lander (or Luna 27). This will be the first European hardware to go to the surface of the moon, and ESA is going to have a vital role by providing the precision GNC system to avoid the wrong craters and land in the right place on the south polar regions of the moon, with a system called PILOT (Precise Intelligent Landing using On-board Technology). Polar regions are gaining a lot of interest than in the past, where only equatorial regions were explored (like in Apollo missions). The south pole region is particularly appealing because it is a unique environment where craters preserve a surprising amount of water and other volatile substances. Such craters are called "*cold traps*", because their shattered interior areas are so cold that they are capable of freezing (trapping) volatiles.

The spin axis of the Moon is nearly perpendicular to the ecliptic plane (i.e. the plane in which the Moon orbits around the sun) so that the sun always appears just at the horizon at the poles. The lunar terrain is rugged and diversified, and because of the craters and mountains of the Moon, some areas near the pole are in near-permanent sunlight while others are constantly shadowed and dark. Consequently, this combination of properties generates extremely cold temperatures which allow any volatile substance, like water, to accumulate and deposit there. The lunar south pole region is much richer of such craters than the north area, the reason why space missions are focused on the former one. The presence of ice water would be an essential resource for next explorers and for a future sustainable human-settled outpost, which could take advantage of the large period of time where the mountain peaks near the pole are illuminated to provide solar energy.

So within the Lunar Resource Lander mission, once landed on the moon's surface thanks to PILOT, the so-called PROSPECT package will take action. It consists of a percussion drill (ProSEED) designed to collect permafrost regolith and ice samples and bring them up to an on-board laboratory (ProSPA) inside the spacecraft, consisting of fifteen instruments overall. The instrumentation won't only analyze lunar samples and possible existence of water, but also plasma in the exosphere, dust and seismic activity. The second project in which ESA is deeply involved in a collaboration with NASA colleagues is Orion (also known as Orion MPCV - Multi-Purpose Crew Vehicle), which is a class of partially reusable space capsule for NASA's manned space programs. The capsule consists of a Crew Module designed by Lockheed Martin and the European Service Module (ESM) designed by ESA and Airbus Defence and Space, which will provide the power, the profile propulsion, the water, the oxygen and all the supplies needed by the astronauts. Orion will be used for multi-purpose missions, such as Artemis 1 or for a future Deep Space Gateway, a space station orbiting around the moon in which Orion could be docked.

A third project aims to extract, collect and process resources directly on the Moon (and potentially on other astronomical objects) and convert them into usable products and services: ISRU (In-Situ Resource Utilization). ISRU could provide materials for propellants, general constructions, life support or energy. This practice is a possible path for reducing the mass and cost of payloads and space architectures by minimizing the materials from and to the Earth. A mission to explore lunar resources could be a reality from 2025. The main and first goal will be drinkable water and breathable oxygen on the Moon. The last one project is the Heracles mission, which could take off in 2028 to gain knowledge on human-robotic interaction. The idea consists of sending a relatively sophisticated lander or rover to the south polar regions of the moon, with the know-how of acquiring samples of unexplored regions scientifically exciting and then bring back those samples to the lander vehicle which would have an ascent craft bolted to the top of it that would take off and go back into the lunar orbit performing a rendezvous with the Deep Space Gateway. So the whole mission would be operated by the astronauts aboard the Gateway lunar station, with the samples coming back to them and then forwarded to the home planet with the first available Orion spacecraft. This mission has been already discussed with Japanese Space Agency (JAXA) and the Canadian Space Agency (CSA), finding a concrete collaboration to make it possible [14].

1.2.2 A Worldwide Ambition

In 2007, Japan launched its first lunar orbiter SELENE which orbited the Moon for only about two years before a planned crash set off the end of the mission. China, as far as concerns, launched its first lunar spacecraft named Chang'e-1 around the same time window, producing a high definition 3D map that would constitute a reference for future soft landings. India then followed in 2008 with the Chandrayaan Programme. By 2013, China turned into the third nation to successfully soft-touch the lunar surface, when its Chang'e-3 spacecraft deployed the Yutu rover. More breakthroughs were achieved in 2019, some more effective than others. In January, Chinese lander Yutu-2, marked the history by becoming the first rover to land on the furthest side of the moon. In the interim, India's second lunar orbiter, Chandrayaan-2, tried to deploy on the lunar surface the Vikram small lander, but failed that year (a new attempt is scheduled by 2021). In April 2019, Israel aimed for the moon with the launch of its Beresheet spacecraft, being more a demonstrator of a small robotic lunar lander with the objective of promoting careers in science and technology. Sadly, even though the spacecraft achieved lunar orbit, it crashed and failed its endeavor to land. Dissimilar to other spacecraft that preceded it, Beresheet was constructed mainly with private subsidies, proclaiming a new period of lunar exploration wherein private owned business are hoping to take the reins from governments. Private spaceflight is nothing new. In the United States, commercial companies played a role in the aerospace industry from the get-go: since the 1960s, NASA has relied on private contractors to build spacecraft for each significant human spaceflight program, beginning with Project Mercury and proceeding until the present.



Fig. 1.2: Life on Mars (*concept art*) - credits by NASA

Today NASA is collaborating with astounding commercial spaceflight organizations to foster both robotic and manned landers for lunar and Martian exploration; among those companies there are SpaceX, Blue Origin, and Astrobotic. Amazon CEO Jeff Bezos along with its Blue Origin company has declared the objective of setting up a lunar base near the south pole where people could work and live. SpaceX which determined a new worldview by making possible reusable rockets, has been running regular cargo resupply missions to the ISS since 2012. In May 2020, the company's Crew Dragon spacecraft ferried NASA astronauts Doug Hurley and Bob Behnken to the ISS, becoming the first crewed mission to launch from the U.S. in almost ten years. SpaceX is additionally developing the Starship spacecraft, someday capable of shipping astronauts to the moon and Mars (Fig. 1.2). The same company is also planning to bring a new class of space tourists to lunar orbit, likewise several organizations are concurrently planning, such as Blue Origin and Virgin Galactic, which are now investing money into sub-orbital space travel industry. To not be overshadowed by the commercial sector, NASA is planning its own ambitious return to the moon. The NASA's Artemis Program, a sister to the renowned Apollo Project, work towards putting the first woman, and the next man, on the moon by 2024. NASA's Orion space capsule represents the pillar of Artemis Program, and it is currently being developed. If Artemis goes well, then the not-so-distant future may likely see NASA and partners engineering and creating a brand-new space station in lunar orbit (the

Deep Space Gateway) that could serve as a springboard to destinations on the moon's surface and beyond [1, 11, 28].

1.2.3 The Futuristic Roadmap

It's just begun what could be called the 21st-century space race, the one that could potentially lead to 10-minute space vacations, orbiting space hotels, and settle humans on the moon and Mars. The idea of a spacial civilization, and particularly a sustainable lunar outpost, has long inspired humankind to imagine what daily life may be like on our satellite or somewhere else. But what is commonly overlooked while fantasizing about the future is the necessary path to get there. This path begins with the first lunar spaceport, which will be crucial for the construction of a moon colony. The first spaceport will allow humans to land and launch from the same location, unlike the six crewed Apollo missions did. This presents a series of challenges never faced before.

The first lunar spaceport will need several missions to take shape, starting with autonomous cargo and infrastructure missions, prior to the first humans' arrival, adequate tools, supplies, and habitable areas in place will all be necessary to assure the safety and productivity of the first crewed mission. Expect these autonomous cargo missions to be one-way trips designed to withstand the lunar debris ejected by several nearby landings, including the first manned mission. Without the return-trip requirement, more of the payload can be devoted to cargo instead of return propellant. While the first moon base's exact location is not yet known, it could be assumed that the primary habitat zone will be near an area of scientific interest, rich in minable resources or other scientific significance. Due to the variability of lunar regolith at different locations, each possible site will possess its own design characteristics and challenges. For example, the thickness of the regolith can vary from 3 meters on mare surfaces to around 10 meters on highland regions, so it means that more excavation process may be required in some places than other to reach local bedrock and stability of surrounding soils. Earlier research and space agencies' current direction both target the lunar south pole as a possible first site for a permanent lunar base. The craters near the south pole have a certain number of permanently shadowed regions with possible significant ice deposits, providing a source for other valuable elements, as previously stated.

On Earth, launching and landing rockets requires extensive and high-cost infrastructure. To engineering and build spaceports is necessary to adequate space, facilities, propellants, high-pressure gasses, water, electricity, communications, telemetry, and logistical operations to support both a launch and a base. In order to support a lunar spaceport, several parts of a whole network of structures will be necessary to guarantee the capabilities to routinely land and relaunch vehicles, a much different vision than the Apollo single-use design. While the thrust magnitude required for launch is much smaller than on Earth, the pad's material will likely be much different requiring alternative maintenance and construction timelines. Once crewed missions begin, there will be the need for various types of vehicle processing, including

payload integration and refueling, which should be able to operate concurrently on the lunar surface.

Initial launches and landings will have to supply their own fuel at extreme cost, as well as provide the lunar surface with equipment and resources to support future missions. As time goes by, subsequent launches from the lunar surface would gradually and massively benefit from fuel produced and stored on the moon. The appropriate tools and resources to construct these facilities could be delivered over multiple iterative round-trips. That's why first lunar spaceport's layout will need to consider operational logistics more heavily. Two main factors will lead to provide flown-in equipment meant to serve as many purposes as possible in order to reduce operational difficulties: a not yet well organized and distributed lunar population, and the environment itself which makes every single task harder than on Earth.

Safety in engineering processes is another aspect at the forefront, and one useful observation could be the explosive safety on the lunar surface. Due to its atmosphere-less characteristic, blast pressure-waves do not propagate in space, so explosive safety distances may be much smaller on the moon. Even so, the threat from debris ejected by an explosion or a launch still remains, with a further-reaching capability the moon has in contrast to the Earth. The first lunar spaceport will need a co-located launch and landing pad on a flat, clean and hard surface with protective barriers. The barriers limit the potential hazard for debris to impact equipment and structures in the surrounding areas, avoiding possible damages and keeping the low-angle debris from reaching orbit. From the first lunar spaceport's earliest days, a reliable energy source will be necessary for almost all operations. The easiest, safest solution for space power generation is the solar energy, which has served well space exploration to date. But the lunar day can keep certain areas of the moon in shadows for extended periods of time, mostly in the polar regions, so it doesn't represent the perfect viable option. Lunar range temperature is furthermore wide and highly sensitive, and can vary from around 120°C at the equator to -180°C just before sunrise. This huge swing in temperature on the surface could lead to a sort of geothermal energy source. Since the lunar surface is a great insulator, heat exchangers could be exposed to boiling temperatures during the day to help generate power. However, the long days and nights would limit the use of these systems to the 14-day in and out of operation, which means alternative energy sources – such as a Fission Power System – may become a more workable long-term solution. The lunar power distribution grid will likely require a mixture of sources to create reliable, sustainable energy supply. The baseline power requirements on a lunar spaceport could also go through changes because of climatic differences if compared to those on Earth.

Human factor engineering and requirements will experience a serious evolution and adaptation. A lunar spaceport offers an opportunity to build from the past 50 years of space exploration and test the best innovative theories for long-term habitation. In this scenario, 3D printing technologies could be a game changer in developing large-scale space-based construction systems, giving also a cutting-edge tool to achieve in-situ building and maintenance autonomy. When constructing lunar



Fig. 1.3: Explorers on the moon (*concept art*) - by David Hardy

habitats, human factor engineers will need to consider things that are afterthoughts on Earth. For example, door sizes will need to be reassessed, as lunar circulation between rooms may require different heights. In chambers the habitat's internal dimensions will vary from those where inhabitants wear casual clothing to where inhabitants wear spacesuits. The lunar voyagers' expeditions must also be considered. Mobility will be both a limiting factor and a driving requirement to plan and build a future moonbase. How far an astronaut can reasonably walk will not only determine requirements for our first crew but could determine what is expected in the future missions. In addition to protecting the inhabitants physically, protecting the mental health of long-term colonists is also a key factor. Preserving the natural landscape of the lunar surface could be a possible solution: the human mind could find solace in the undeveloped wilderness that laid before settling, just like our ancestors did on Earth [35].

With each new data acquisition and discovery, the long-considered vision of a lunar civilization is continuously modified with no clear sight of where it will end up. One thing is certain, however: regardless of the settlement location, the nations involved, or how far into the future it happens, it will all start with a spaceport, that's why engineering autonomous soft-landing capabilities play a crucial role in this day and age. Sooner or later, what today may seem as a simple visionary picture is going to turn the moon into a real launchpad to the universe, establishing a new flagship for all mankind.

2

Powered Descent and Landing

2.1 State of Art and Trends

A wide range of key developments can currently be pointed out in Guidance, Navigation and Control (GNC) systems for planetary landers, among which Hazard Detection and Avoidance (HDA) systems and Visual Navigation stand out. In this section these systems are shown to disclose some more details, taking into consideration a general survey of the prerequisites on trajectory shaping guidance for current and future missions. Considering recorded data, the empirical safe landing chance was uniquely about 80% as of the last decade. Thanks to sub-meter resolution satellite images and scans of the Moon's surface obtained lately by NASA's Lunar Reconnaissance Orbiter, remarkable progresses on automated hazard detection have been made. To some degree, hazards like slopes, boulders, craters or shadows can be detected by the gathered orbital measurements. However, the resolution is still not enough to identify smaller hazardous terrain features, like medium-small rocks, that could jeopardize touchdown success. In such a scenario, employing automatic HDA technologies would be a beneficial solution to increase the probability of safe landing. The system instrumentation should have real-time capabilities in order to characterize the sensed terrain in terms of hazardous elements within the bounds of the designated landing area at any moment. Lastly, the system must be able to decide whether the targeted landing spot is safe, and if necessary, command a diversion to a more secure zone (safe-site determination capability) [16].

Vision-based technologies allow for more precise landings than ever before. Overall these navigation technologies can be split in two main categories: relative and absolute vision-based navigation. The latter works by comparing camera images to a on-board map database in support of the estimation of the inertial state. Given

that HDA can dictate diversions to new landing safer sites, the relative position of the vehicle with respect to the landing site gains in importance over the inertial state, so the distance between the vehicle itself and the landing target needs to be known even more precisely. As a result, relative vision-based navigation is employed in the final landing phase, so the vehicle position is calculated over the features captured by the on-board camera images of the landing target. In other words, relative navigation is performed by comparing information from successive real-time captured images, while absolute navigation is achieved by matching the perceived environment with a vehicle-carried pre-installed map. Thanks to the contribution of these technologies, the reachable increase on landing precision can be in the order of hundreds of meters. Furthermore, if foregoing landed surface data for relative navigation were available, the precision may be increased up to tens of meters or even single-digit meter. This degree of precision differs dramatically from the kilometer-sized landing ellipses, that represented the best feasible result for earlier autonomous lunar landings. The combination of visual navigation and HDA systems broaden the range of mission scenarios, including their success. Anyway, all that glitters is not gold, and the need to keep a visual to the landing site during the entire phase of descent, coupled with possible retargetings and diversions, pose new challenges for autonomous landers. HDA and vision-based landing have a major role in the design of the guidance system, like trajectory shaping, and they influence it in several ways. First of all, new landing sensors place additional constraints, and a few extra details are worth mentioning:

- *Pinpoint-landing Capability*: guidance mode must be able to target accurately a specific site on the planetary surface with the best achievable precision. Not all algorithms are necessarily capable of it.
- *Adaptivity*: being that HDA system can order diversions to alternative landing sites – in case of detecting unsafe or safer site – autonomous guidance shall be capable of re-planning the trajectory to the new locked on target in real-time.
- *Constrained Trajectory Generation*: to meet the requirements of the whole landing phase, a constrained trajectory is generated. One of the path constraint could be that the landing site must be kept in the sensors' field of view, in its turn translated into limits on the viewing angle, and thus in considering the glide slope component. That leads to another fundamental constraint which is the lander attitude. In addition, thrust limitations cannot be left out, especially when retargeting maneuvers require engines to throttle up.
- *Fuel Optimality*: the generated constrained-trajectory shall be fuel-optimized in order to maximize the payload mass. This feature is significantly important, mostly when combined with HDA, because diversion maneuvers may require extra usage of propellant.
- *Computational Efficiency*: powerful space-grade computers have been developed, but the on-board data processing and the computational power needed

for HDA and visual navigation might put a heavy load on the CPU and memory. For this reason, guidance workload will be limited in terms of computational power availability, and algorithms must work seamlessly, ensuring maximum reliability.

- *Robustness*: in a HDA activity scenario, robustness is a crucial property to ensure a high safe-landing probability. The algorithm must guarantee operational suitability and meet the required executing performance against uncertainties, presence of invalid inputs and stressful environmental conditions.

These questions become clearer when guidance is contextualized within the other GNC system elements. An example of a HDA architecture is shown in Fig. 2.1. Instruments map the surface, and subsequently assess the safety of targeted sites by creating risk maps. Alongside, a simplified guidance shapes the trajectory and computes the reachability, called fuel map, to establish the range of feasible diversions. Fusing both maps, the piloting algorithm either approves the current landing site or sets a new one. If the direct interface of guidance receives the retargeting command a new trajectory is then computed. Constraint-handling is continuously kept active by guidance computer.

Having more clearly understood the requirements for an advanced guidance and now that the research area has been identified, the problem formulation can be circumscribed to the mission scenario under evaluation: The powered descent guidance problem for pinpoint landing is defined as finding the fuel-optimal trajectory that takes a lander with a given initial state (position and velocity) to a prescribed final state in a uniform gravity field, taking into account constraints of a different kind [16, 24].

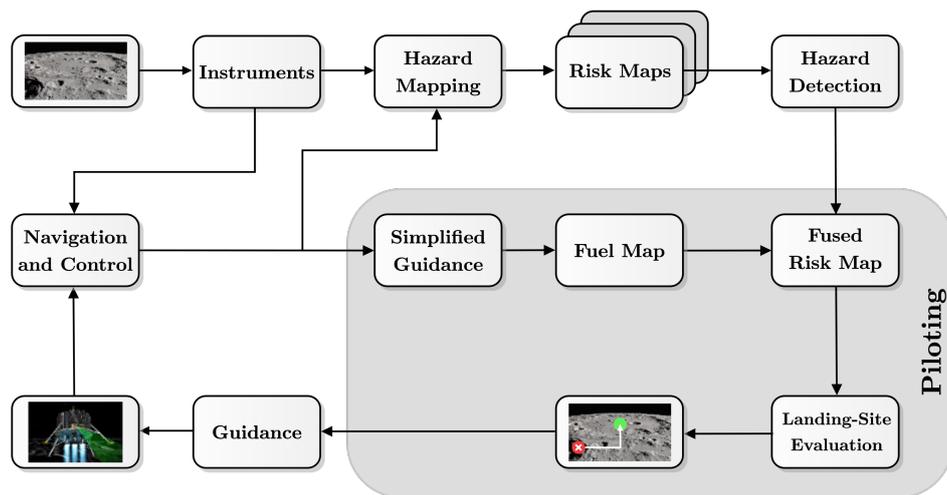


Fig. 2.1: GNC landing system architecture with HDA in-the-loop [16]

2.2 Mission Description

The vehicle is required to gently touch down with high accuracy on the moon's surface, while meeting all the imposed and necessary constraints on the dynamic system during the whole descent phase. A certain accuracy grade will probably be required, just think of outpost kind of mission, in which assets and equipments may be already settled on the Moon, or merely the ability to avoid unsafe landing areas. This prompts to a significant progress in capabilities over the Apollo Lunar Module, particularly in route precision. Likewise advantageous would be the ability to detect and avoid hazards, such as rocks and craters, provided by the HDA systems previously mentioned. In this sense, advances in sensor capability come to the aid of sharp detection, highlighting surface features of any kind. The trajectory guidance must ensure dynamic tridimensional retargeting to avoid potential hazards detected by the sensing systems. So not only downrange distance but also crossrange capability is desirable to be implemented within this framework. Generally, the trajectory shape must consider space and attitude constraints for allowing the sensors to see the targeted area, as well as time and altitude margin for calibrating the trajectory based on the real-time information acquired by the sensors [33].

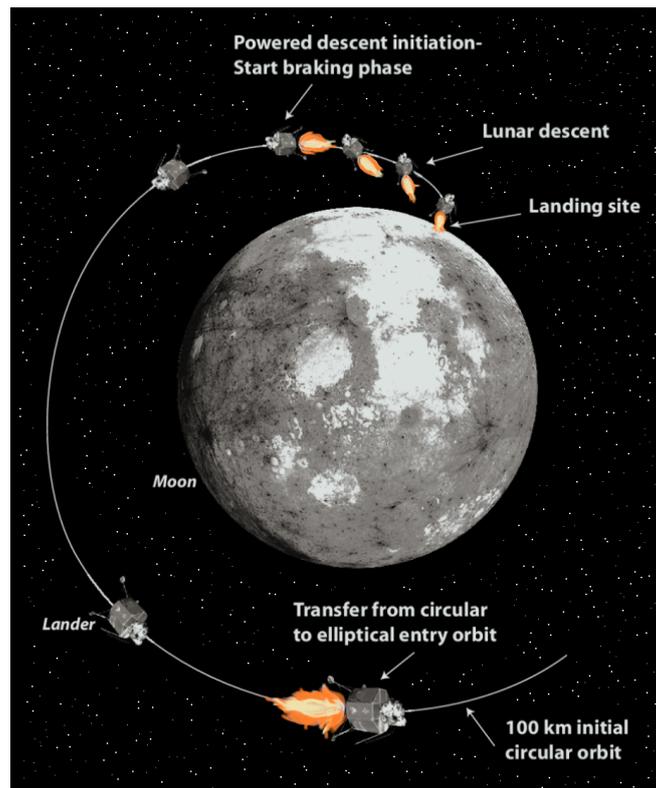


Fig. 2.2: Main Phases from Circular Orbit to Landing Site

A propellant optimized trajectory is also useful, taking into account a given cost function for landing site selection, it can result in a better payload capability and also in a safe estimation of the remaining translational motion of the vehicle. Some extra considerations and features about guidance may include site authorization and landing abort with back to orbit recovery. Guidance algorithms are usually customized to a specific mission stage, this is why are not large-scale suitable. A reference frame of a generic landing sequence is drawn up to provide a general outline. This scenario is based on the heritage of previous missions, most notably Apollo Missions, and the terms presented hereinafter are thus common terminologies appropriate for typical landings on low-gravity bodies with a thin atmosphere. The general phases are shown in Fig. 2.2. The mission is assumed to start in a generic initial orbit, be it an interplanetary hyperbolic transfer orbit or an elliptical/circular parking orbit around the Moon. Taking as a reference the Apollo Program, the vehicle was first put into a circular lunar parking orbit before starting the descent. Given the initial position and a targeted landing point on the surface, a landing algorithm estimates the most convenient Time of Ignition (TIG) at which to begin the deorbit (descent) burn. In correspondence of the Descent Orbit Insertion (DOI), the descent is started and inserts the spacecraft into a Hohmann-transfer orbit with a lowered periapsis above the landing area. When the vehicle is located to an altitude of about 15 km above the lunar surface, Powered Descent Phase starts and consists of 3 main sub-phases: *Braking Phase*, *Approach Phase*, *Terminal Vertical Descent*.

Most of the time, such as the Apollo Lunar Module, the main *Braking Phase* is started minimum at 500 km downrange with the Powered Descent Initiation (PDI) maneuver. It is at this point that most of the orbital velocity is removed, slowing it down from thousands to tens meters per second. As soon as the landing site (LS) enter into the Field of View (FOV) of the Hazard Mapping (HM) hardware, High gate (HG) is reached. When HG is crossed, engines throttle back and the vehicle starts evaluating the best spot to land, Approach gate (AG) is reached. The subsequent phase is then called the *Approach Phase*, and may include retargetings, i.e. diversions to safer landing areas executed only if the initial landing site would be found to be unsafe. As soon as view on LS is lost, low gate (LG) is reached. All the horizontal velocity should be cancelled when the vehicle is a couple of meters above the surface. From here, the vehicle descends at constant speed to the surface, till touchdown (TD). As soon as AG point is crossed, the vehicle starts the most crucial landing operation: HDA is activated, the landing site is targeted, thrust and attitude thus constantly adjusted to reach it (pinpoint landing). The main Braking Phase is commonly performed at full thrust in the optimal direction, with minimal steering. In this way velocity drops in the most efficient manner possible by firing engines against the velocity vector, enabling the so-called retro propulsion. The terminal Vertical Phase can be easily controlled as well. The most critical phase in terms of safety, landing precision, and room for improvements on the propellant usage is thus the Approach Phase. Bear in mind that lunar guidance algorithms may be also adapted for descent phase on Mars or asteroids [2, 3, 16].

3

Optimal Control Problem

3.1 What is Optimization

Principles of optimality rule our regular day-to-day existence. Every system around us, be it natural or artificial, that has a bearing on our lives, seeks to function optimally, i.e. it attempts to maximize or minimize some given function under certain conditions. This can be observed in both microscopic and macroscopic circumstances like atomic bonds whose structure aims to minimize the total potential energy of molecules. Neural networks in human body consider threshold limit of transferred information to calibrate and minimize neuron wired-connections. Flower petals are placed in order to maximize their number as a function of their size and the available area, while birds optimize their wings' shape in real time during flight. In these examples, nature is who decide, using evolutionary (trial and error mechanisms - law of the fittest) process. As Leonhard Euler once wrote:

“[...] nothing at all takes place in the universe in which some rule of maximum or minimum does not appear.”

Optimization is often used to mean improvement, but mathematically it is a much more precise concept: finding the best possible solution by changing variables that can be controlled, often subject to a preset of constraints. Optimization is a widespread recourse because it is applicable in all domains and because we can all identify with a desire to make things better. Any problem where a decision needs to be made can be cast as an optimization problem.

Artificial systems, such as robotics or logistic chains, are meant by engineers to represent mathematically the objective and constraints form of functions and

equations. Using such mathematical model of the reality, real optimization problem can be then solved by exploiting algorithms provided by mathematics and computer science coding. Solution is then output by a set of discrete values of decision variables, called optimal control variables. Once completed the process, undoubtedly nothing better is possible to achieve for the declared form of objective function and constraints, and for the real and concurrent state of the system.

Historically, optimization problems started dealing with static models, i.e. problems with no time horizon; in other words without any sort of dynamics. This kind of approach was only spacial-based and gave rise to what is known as **Calculus of Variations**. Variational problems include finite-dimensional and infinite-dimensional frameworks. While the former is about the search of maxima and minima - collectively called *extrema* - of a given function, the latter framework opens to a wider family of possibilities, where the domain shifts from a generic function $y(x)$ to a *functional* $J(y)$, mathematically defined as a function on a space of functions. This shifting ushers in a set of problems where a certain functional has to be minimized or maximized over a given family of curves: this can be also seen as the choice of a path - $y^*(x)$ - from a given family of admissible paths $y(x)$ so as to minimize or maximize the value of a certain functional $J(y^*)$, called *cost or objective functional* if using optimization terminology. This approach deals with path optimization but still not in the setting of control system. The transition from Calculus of Variations (static model) to **Optimal Control Theory** (dynamic model) can be summarized as follows: rather than considering the curves as given a priori, let's imagine a particle moving in the space and "drawing a trace" of its motion. The choice of the slope $y'(x)$ at each point on the curve can be thought of as an infinitesimal decision, or control. The generated curve is thus a trajectory made by a simple control system, which can be expressed as $y' = u(t)$, where $u(t)$ represents the control variable. In order to steadily minimize or maximize the overall cost function, optimal control decisions must be taken along the curve at each point.

Once we adopt the dynamic viewpoint, it is opportune to parameterize the curves by time t rather than by the spatial variable x , so the approach becomes time-based. Apart from being more intuitive, this new formulation is definitely more illustrative because it allows us to distinguish between two geometrically identical curves crossed with different speeds, for instance. Even the formal and conceptual representation of the functional changes, since we are no longer working with "a priori" curves, it makes more sense to write $J(u)$ instead of $J(y)$, given that the actual path is now determined by the choice of controls which in turn are functions of time. For the sake of completeness it should be clarified that the cost depends on the initial data, on the final time as well as on the control, so it would be more appropriate to write $J(t_0, x_0, t_f, u)$ but we write $J(u)$ for a better clarity while pointing out that the cost is being minimized or maximized over the space of control functions. Note that even if the cost function does not depend explicitly on u , the cost J depends on the control u through x , state variables which describes the trajectory that a given control regulates. As can be seen, the new problem formulation is more general in

another aspect: it avoids the implicit assumption made in calculus of variations that admissible curves are just graphs of functions of x .

Having switched to a dynamical framework, optimal problems will be generally based on dynamical or control systems, so from this point onward terminology changes, the time t will be used as the independent variable, $x = (x_1, \dots, x_n)^T$ for the (dependent) state variables, $\dot{x} = (\dot{x}_1, \dots, \dot{x}_n)^T$ for their time derivatives, and $u = (u_1, \dots, u_m)^T$ for the controls, which will take values in some control set or domain [10, 25, 27].

3.2 Survey of Methods

It is of remarkable interest to notice that a wide variety of problems of optimal control configuration emerges in distinct fields of engineering, computer science, economics, finance, astronomy, physics, structural and molecular biology, medicine, and material science. Such problems are based on finding of optimal control strategy which minimizes energy consumption or resource usage inside a production chain, maximizes sales profit, or leads to optimal model or structure identification. Remember that a minimization problem can be converted in a maximization one by simply inverting the sign of the objective functional.

It has been clarified that an optimization problem consists in finding the control law that maximizes or minimizes a particular performance index. Considering a space mission scenario, consumptions of propellant play a big role in influencing the costs of orbital transfers and maneuvers, so minimize the quantity of propellant necessary for the maneuver or, equivalently, maximizing the final mass of the vehicle (fixed the initial one) becomes crucial. Derek Frank Lawden laid the foundations of the optimization of space trajectories, in his pioneering work during the 1960s. In particular he was the first to coin the term "primer vector" to refer to the adjoint variables in the costate equation associated with the velocity vector, pointing out their fundamental connection to optimal thrust: this concept will be better elucidated in the next chapters, with a more in-depth analysis tools [10].

Finding the minimum-fuel (or max-final mass) trajectory with a set of path constraints is a classical space engineering optimal control problem. Many efforts have been devoted to theoretically analyze this problem. Exact or approximate analytical solutions may be derived depending on the dynamics formulation, but such analytical solutions, when available, are of limited scope due to the high amount of simplifications needed, and therefore are of scant interest. In order to solve real-world applications a variety of numerical methods have been developed which can be distinguished in three main categories: *Direct Methods*, *Indirect Methods* and *Evolutionary Algorithms*. There also exists a fourth category which strives to combine direct and indirect methods in order to benefit from their respective qualities, and for this reason they are called *Hybrid Methods*.

- *Direct methods* transform the optimal control problem in a nonlinear programming (NLP) problem. The trajectory is discretized, and state and control variables are generally approximated by a piecewise constant parameterization for each trajectory stretch (or arc). So the whole problem turns into a parameter optimization problem and many discretizing approaches can be envisioned depending on the nature of the problem (Convex Optimization, Dynamic Programming, Quadratic Programming, Fractional Programming). The resulting large-scale optimization problem is then solved by a nonlinear (NLP) software. Direct methods handle easily any problem formulation (dynamics and constraints) with reduced programming effort. Despite that, due to the problem large size (the method requires a very large number of parameters for an accurate trajectory description), they may be computationally expensive and possibly inaccurate.
- *Indirect methods* transform the optimal control problem into a Boundary Value Problem (BVP) through the application of the Optimal Control Theory. The trajectory can be split in arcs here too, only if necessary, and especially when the problem has to deal with discontinuities. Unlike direct methods, indirect methods deal with continuous-form of state and control variables. The optimal control law is determined from the Pontryagin's Maximum Principle (PMP). The problem unknowns are the initial costates that must be found in order to satisfy all the imposed boundary conditions and constraints. The small problem size makes the indirect approach attractive, indeed the method needs a small amount of parameters to describe a trajectory and limited calculation times, it is highly accurate but numerical issues arise due to the high sensitivity to the initial guess, and need possibly to control discontinuities along the trajectory.

N.B.: The term **direct** approach comes from the fact that you directly work with the continuous optimal control problem by discretizing the integral and all time-dependent functions (ODE, constraints, etc.) in order to obtain a static nonlinear program. For this nonlinear program you can construct optimality conditions given by the KKT (Karush-Kuhn-Tucker) equations. So first you discretize the problem with a set of parameters, then you optimize the resulting nonlinear static program to obtain the (approximate) optimal solution of the optimal control problem. In the **indirect** approach, you first look at the optimality conditions of the continuous-time optimal control problem which leads you to the BVP through the Euler-Lagrange equations (which, again, only constitute a necessary condition and come from PMP). This BVP is still continuous. So in order to solve it, you discretize it in time to obtain an approximate numerical solution.

To recap in a few words: the direct approach is a "first discretize, then optimize" method, while the indirect approach is a backward process, i.e. a "first optimize, then discretize" method.

- *Evolutionary Algorithms* use mechanisms inspired by biological evolution, such as reproduction, mutation, recombination, and selection. Candidate solutions to the optimization problem have the role of individuals in a population, while the fitness function determines the quality of the solutions. Then the evolution process starts, taking advantage of the large populations of solutions, and the final aim is to move towards the global optimum according to specific rules. There are different techniques to achieve the result, and the most popular type of EA is Genetic Algorithm (GA).
- *Hybrid methods* attempt to incorporate both direct and indirect methods useful properties. A direct method is first used to build a good initial guess, and an indirect method is applied to yield an accurate convergence. Amongst the numerous hybrid techniques applied to trajectory optimization, can be mentioned: Impulsive Solution Guess, Multiple shooting, Response Surface Methodology and Dynamic Programming based on the Hamilton-Jacobi-Bellman equation.

Among all these categories, one of the most appreciated method in space engineering field is the indirect approach, and it will be the method employed hereinafter. Despite its drawbacks of being sensitive in terms of convergence (the tentative solution typically needs to be sufficiently close to the optimal one to be qualified as good) and variables discontinuities management is not trivial, it offers a high numerical precision, a cost-effective framework and a significant theoretical content [9].

As already mentioned, indirect methods are based on the Optimal Control Theory which, in turn, has its roots in the Calculus of Variations principles. OCT will be here presented in the form that best suits the application to the optimization of space trajectories. The optimal differential equations, which will constitute the BVP, will come from the application of the Optimal Control Theory and the Pontryagin's Maximum Principle (PMP) [5, 30, 36].

3.3 Optimal Control Theory

Recalling the notation before introduced¹, the optimal control theory is applied to a generic system described by a given vector of state variables $\mathbf{x}(t)$ and a vector of control variables $\mathbf{u}(t)$, both generally continuous function of *time*. The differential equations based on the state and control variables describe the time evolution of $\mathbf{x}(t)$ and $\mathbf{u}(t)$ between the initial and final states (*outer boundaries*). The differential equations take the generic form of:

$$\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt} = f(\mathbf{x}, \mathbf{u}, t) \quad (3.1)$$

¹To make the reading process clearer, it will be referred to vector quantities with bold text.

Where appropriate, a good practice is to divide the trajectory into a number of n sub-intervals, called arcs, inside of which the variables are still continuous. The generic j -th sub-interval starts at $t_{(j-1)_+}$ and ends at t_{j_-} , consequently $x_{(j-1)_+}$ and x_{j_-} are the values assumed by the state variables in those instants. The signs '-' and '+' denote the values assumed respectively right before and right after a specific point: this framework allows to take into account possible discontinuities of the state variables (e.g. impulsive maneuver), of the time (e.g. residence time in a fly-by) or internal constraints, that take place among the junction points between each contiguous arc (*inner boundaries*).

Boundary conditions are then defined, being generally non-linear and assuming a mixed-type form, i.e. where values of the state variables and time are involved both in the outer and inner contours. The assigned boundary conditions are gathered in a designated vector introduced below:

$$\chi(\mathbf{x}_{(j-1)_+}, \mathbf{x}_{j_-}, t_{(j-1)_+}, t_{j_-}) = 0 \quad j = 1, \dots, n \quad (3.2)$$

The optimization problem is based on the research of the control law that maximize or minimize a given functional. Such functional represents the so-called objective functional which expresses costs or benefits of a process which one wants to either avoid or reach, and its generic form is the following:

$$J = \varphi(\mathbf{x}_{(j-1)_+}, \mathbf{x}_{j_-}, t_{(j-1)_+}, t_{j_-}) + \sum_j \int_{t_{(j-1)_+}}^{t_{j_-}} \Phi(\mathbf{x}(t), \mathbf{u}(t), t) dt \quad j = 1, \dots, n \quad (3.3)$$

The contracted form, i.e. with no sub-interval division, of the expression above might be more immediate for the reader and it is here mentioned:

$$J = \varphi(x_{t_f}, t_f) + \int_{t_0}^{t_f} \Phi(\mathbf{x}(t), \mathbf{u}(t), t) dt \quad (3.4)$$

where x_{t_f} and t_f are the final (or terminal) state and time, while φ is the *terminal cost* and Φ is the *running cost* (or Lagrangian). The expression of the objective functional Eq. (3.4) is known as *Bolza form*. There are two important special cases of the Bolza form. The first one is the *Lagrange form*, in which there is no terminal cost: $\varphi = 0$. This formulation comes from the Calculus of Variations, but here with a dynamical adaptation. The second special case is the *Mayer form*, in which there is no running cost: $\Phi = 0$. This last formulation is here preferred and will be used to define a specific performance index later on.

It is necessary to consider also that optimal problems might be likely and realistically subject to given *constraints* that narrow the operational field and therefore the optimization search space, as well as state equations and boundary conditions. Different equality and inequality constraints may be considered to bound the values of decision as well as state variables and controls to respect safety or environmental safeguards, to state certain setpoints in control loop, and so on. These requirements

(state equations, B.C. and constraints) can be handled by the *method of Lagrange multipliers*. This method reduces the constrained problem to a new, unconstrained optimization problem with additional variables. The additional variables are known as Lagrange multipliers. The Lagrange multiplier is an extra scalar variable, so the number of degrees of freedom of the problem has increased, but the plus side is that now simple, unconstrained minimization techniques can be applied to the composite (*augmented* by definition) functional. So let's introduce the multipliers: adjoint variables λ associated with state equations and adjoint constants μ associated with the boundary conditions. Now it is convenient to apply them to the functional that can be reformulated as follows:

$$J^* = \varphi + \mu^T \chi + \sum_j \int_{t_{(j-1)+}}^{t_{j-}} \left(\Phi + \lambda^T (\mathbf{f} - \dot{\mathbf{x}}) \right) dt \quad j = 1, \dots, n \quad (3.5)$$

equivalent to:

$$J^* = \varphi + \mu^T \chi + \int_{t_0}^{t_f} \left(\Phi + \lambda^T (\mathbf{f} - \dot{\mathbf{x}}) \right) dt \quad (3.6)$$

which it is called the *augmented (cost) functional*. Note the same size of adjoint variables vector λ^T and states vector \mathbf{x} : that is why adjoint variables are sometimes also called *co-state variables*.

Both functional J and J^* are functions of: time t , state variables \mathbf{x} and their time derivatives $\dot{\mathbf{x}}$ (if sub-intervals division is considered, J and J^* depends on the values assumed by state variables and time at the endpoints of each arc, i.e. on \mathbf{x}_j and t_j), and finally controls \mathbf{u} . It is quite evident that as long as boundary conditions ($\chi = 0$) and state equations ($\dot{\mathbf{x}} = \mathbf{f}$) are both satisfied, the functional J and its augmented form J^* - and therefore their respective extremal² values - will be exactly the same (whatever the values of μ and λ). To get rid of the dependence on the time derivatives of state variables $\dot{\mathbf{x}}$, an integration by parts is carried out in J^* at this stage:

$$J^* = \varphi + \mu^T \chi + \sum_j \left(\lambda_{(j-1)+}^T \mathbf{x}_{(j-1)+} - \lambda_{j-}^T \mathbf{x}_{j-} \right) + \sum_j \int_{t_{(j-1)+}}^{t_{j-}} \left(\Phi + \lambda^T \mathbf{f} - \dot{\lambda}^T \mathbf{x} \right) dt \quad j = 1, \dots, n \quad (3.7)$$

Let's define the *Hamiltonian* before proceeding:

$$H = \Phi + \lambda^T \mathbf{f} \quad (3.8)$$

By picking up the Eq. (3.7) and differentiating it, the first variation of the augmented function is obtained³:

²An extremal is a function that makes a functional an extremum. Trajectories satisfying the Euler-Lagrange equation are called extremals (of the functional J). Since the E-L equation is only a necessary condition for optimality, not every extremal is automatically an extremum.

³Square brackets stand for a matrix.

$$\begin{aligned}
 \delta J^* = & \left(-H_{(j-1)_+} + \frac{\partial \varphi}{\partial t_{(j-1)_+}} + \boldsymbol{\mu}^T \frac{\partial \boldsymbol{\chi}}{\partial t_{(j-1)_+}} \right) \delta t_{(j-1)_+} \\
 & + \left(H_{j_-} + \frac{\partial \varphi}{\partial t_{j_-}} + \boldsymbol{\mu}^T \frac{\partial \boldsymbol{\chi}}{\partial t_{j_-}} \right) \delta t_{j_-} \\
 & + \left(\boldsymbol{\lambda}_{(j-1)_+}^T + \frac{\partial \varphi}{\partial \mathbf{x}_{(j-1)_+}} + \boldsymbol{\mu}^T \left[\frac{\partial \boldsymbol{\chi}}{\partial \mathbf{x}_{(j-1)_+}} \right] \right) \delta \mathbf{x}_{(j-1)_+} \\
 & + \left(-\boldsymbol{\lambda}_{j_-}^T + \frac{\partial \varphi}{\partial \mathbf{x}_{j_-}} + \boldsymbol{\mu}^T \left[\frac{\partial \boldsymbol{\chi}}{\partial \mathbf{x}_{j_-}} \right] \right) \delta \mathbf{x}_{j_-} \\
 & + \sum_j \int_{t_{(j-1)_+}}^{t_{j_-}} \left(\left(\frac{\partial H}{\partial \mathbf{x}} + \dot{\boldsymbol{\lambda}}^T \right) \delta \mathbf{x} + \frac{\partial H}{\partial \mathbf{u}} \delta \mathbf{u} \right) dt \quad j = 1, \dots, n \quad (3.9)
 \end{aligned}$$

The *necessary* condition for optimality requires that the differential of the functional must be zero at optimum: $\delta J^* = 0$ (*stationary condition*). So δJ^* must be nullified for any arbitrary and admissible variation - called also perturbation - consistent with state equations, B.C. and constraints: δx , δu , $\delta x_{(j-1)_+}$, δx_{j_-} , $\delta t_{(j-1)_+}$, δt_{j_-} . The introduction of multipliers (adjoint variables and constants) and their proper selection allows to cancel out each term - or coefficient - in brackets in Eq. (3.9) at once, ensuring the stationary condition of the functional. Let's start the extrapolation process of the necessary condition for optimality:

The **Euler-Lagrange equation for the adjoint variables** λ is obtained by nullifying the coefficient of the variation δx :

$$\frac{d\boldsymbol{\lambda}}{dt} = - \left(\frac{\partial H}{\partial \mathbf{x}} \right)^T \quad (3.10)$$

It can be shown to be equivalent to the Euler-Lagrange necessary condition for optimality derived from the Calculus of Variations:

$$\frac{\partial \Phi}{\partial \mathbf{x}} - \frac{1}{dt} \frac{\partial \Phi}{\partial \dot{\mathbf{x}}} = 0$$

The **algebraic equation for controls** u is obtained by nullifying the coefficient of the variation δu :

$$\left(\frac{\partial H}{\partial \mathbf{u}} \right)^T = 0 \quad (3.11)$$

Before proceeding, it is worthwhile to make some observations with regards to the results just achieved. First of all, the control law Eq. (3.11) is pretty much independent from finding the extrema (maxima or minima) of J . The same characteristic will be also valid for B.C. as shown later on. In addition, if a control variable is subject to constraints, the equation may not provide the optimal controls. This may be the case when the thrust magnitude can vary between a minimum or a maximum value

(cases where controls depend on time or state variables are not taken into account, just explicit and constant expressions shall be allowed). Therefore, it would make no sense to look for a solution that requires a thrust level outside the constraints. An admissible control is defined as a control that does not violate any constraint, and the subset of admissible controls is called *admissible region*. The **Pontryagin's Maximum Principle** states that the optimal control must maximize the Hamiltonian H (if J^* is to be maximized). Basically two possibilities may occur:

- the optimal control is given by Eq. (3.11) if the control is in the admissible region, thus the constraints are actually not involved (control locally not constrained);
- the optimal control is at the boundary of the admissible region, meaning that the control assumes its maximum or minimum value if Eq. (3.11) yields a control outside of the admissible region (constrained control).

In particular, if the Hamiltonian H is linear with respect to a control variable, i.e. the Eq. (3.8) doesn't yield a well-determined and explicit expression of the control, two other possibilities may occur:

- if the coefficient of the control in the Hamiltonian H is not null, H is maximized either for the maximum value of the control if the coefficient is positive or for the minimum value of the control if the coefficient is negative, according to the Pontryagin's Maximum Principle;
- if the coefficient of the control in the Hamiltonian H is null for a finite interval of time, a *singular arc* arises and it is necessary to set all the successive time derivatives of the coefficient equal to zero, until one of the controls appears specifically in one of them; the optimal control is determined by setting such time derivative (the one that explicitly gives the control) equal to zero (it is well known that the order of time derivative is always even for this purpose, and its half value sets the order of the singular arc).

Finally, the **boundary conditions for optimality** are determined by nullifying the coefficients of the remaining variations $\delta \mathbf{x}_{j-}$, $\delta \mathbf{x}_{j+}$, δt_{j-} , δt_{j+} , leading to:

$$-\lambda_{j-}^T + \frac{\partial \varphi}{\partial \mathbf{x}_{j-}} + \boldsymbol{\mu}^T \left[\frac{\partial \mathcal{X}}{\partial \mathbf{x}_{j-}} \right] = 0 \quad j = 1, \dots, n \quad (3.12)$$

$$\lambda_{j+}^T + \frac{\partial \varphi}{\partial \mathbf{x}_{j+}} + \boldsymbol{\mu}^T \left[\frac{\partial \mathcal{X}}{\partial \mathbf{x}_{j+}} \right] = 0 \quad j = 0, \dots, n-1 \quad (3.13)$$

$$H_{j-} + \frac{\partial \varphi}{\partial t_{j-}} + \boldsymbol{\mu}^T \frac{\partial \mathcal{X}}{\partial t_{j-}} = 0 \quad j = 1, \dots, n \quad (3.14)$$

$$-H_{j+} + \frac{\partial \varphi}{\partial t_{j+}} + \boldsymbol{\mu}^T \frac{\partial \mathcal{X}}{\partial t_{j+}} = 0 \quad j = 0, \dots, n-1 \quad (3.15)$$

The above B.C. for optimality are calculated for a generic j -th arc, ranging from the final extreme of $(j - 1)$ -th sub-interval and the initial extreme of j -th sub-interval. The subscripts j_- and j_+ stand respectively for the values assumed by point j just before and just after its position. This distinction is crucial because as previously mentioned, it allows to consider possible discontinuities in the junction points between sub-intervals. Considering the whole trajectory, Eqs. (3.12) and (3.14) have no meaning at the starting point, while Eqs. (3.13) and (3.15) have no meaning at the final point.

If the generic state variable x is characterized by particular boundary conditions, Eqs. (3.12) and (3.13) yield particular boundary conditions for optimality for the corresponding adjoint variable λ_x :

- if the value of x is given at the starting time ($\chi = 0$ contains the equation $x_0 - a = 0$ with a given value for a), the corresponding adjoint variable λ_{x_0} is free, it means that it does not appear as a boundary condition for optimality and it can assume any value; the same happens for a given value for x at the final time;
- if the initial value x_0 appears in neither the function φ nor in the boundary conditions, the corresponding adjoint variable is null at the initial time ($\lambda_{x_0} = 0$); the same happens for the analogous situation at the final time;
- if the state variable is continuous and its value is not explicitly set to a value at the inner boundary j ($\chi = 0$ contains the equation $x_{j_+} = x_{j_-}$), the corresponding adjoint variable is continuous ($\lambda_{x_{j_+}} = \lambda_{x_{j_-}}$);
- if the state variable is continuous and its value is explicitly set to a value at the inner boundary j ($\chi = 0$ contains the equations $x_{j_+} = x_{j_-} = a$), the corresponding adjoint variable has a free discontinuity, that is the value of $\lambda_{x_{j_+}}$ is independent from that of $\lambda_{x_{j_-}}$ and it has to be determined by the optimization procedure.

Likewise, if H is not an explicit function of time t , in some cases Eqs. (3.14) and (3.15) yield specific boundary conditions for optimality:

- if the initial time t_0 appears explicitly in neither the boundary conditions nor the function φ , the Hamiltonian is null at the initial time ($H_0 = 0$); analogously, the Hamiltonian is null at the final time if t_f appears explicitly in neither χ nor φ ;
- if the intermediate time t_j does not explicitly appear in the function φ (it appears only in the boundary condition for the time continuity $t_{j_+} = t_{j_-}$), the Hamiltonian is continuous at the inner boundary j ($H_{j_+} = H_{j_-}$);
- if the intermediate time t_j is explicitly assigned (it appears in the boundary conditions as $t_{j_+} = t_{j_-} = a$), the Hamiltonian has a free discontinuity at the inner boundary j .

By canceling out the adjoint constants μ from Eqs. (3.12) ÷ (3.15), the resulting boundary conditions for optimality, and the boundary conditions on the state variables given by Eq. (3.2), can be collected altogether in the following vector:

$$\sigma(\mathbf{x}_{(j-1)_+}, \mathbf{x}_{j_-}, \boldsymbol{\lambda}_{(j-1)_+}, \boldsymbol{\lambda}_{j_-}, t_{(j-1)_+}, t_{j_-}) = 0 \quad (3.16)$$

Hence, Eqs. (3.1), (3.10), (3.11) and (3.16) define the so-called *Multi-Point Boundary Value Problem* (MPBVP) [8].

3.4 Boundary Value Problem

The application of the Optimal Control Theory to the system (3.1) generally produces a MPBVP (in the case of one interval of integration a *Two-Point Boundary Value Problem*). Equations (3.1) and (3.10) are the differential equations of the MPBVP and the controls are determined by Eq. (3.11). The solution to this problem is obtained by looking for the initial values of the unknown variables such that the integration of the differential equations satisfies the boundary conditions of Eq. (3.16). In particular, the interval of integration is split in sub-intervals and different sub-intervals might be characterized by different differential equations. Generally, the duration of each sub-interval is unknown and the boundary conditions may be non-linear and may involve both outer and inner boundaries. In addition, variables may be discontinuous at the inner boundaries and their values may be unknown after a discontinuity. It is evident that the main challenge that comes from indirect methods is actually the search for a solution of the MPBVP that emerges from their application. It is from here that the discretization process starts.

In order to deal with the unknown duration of the sub-intervals of integration, a change of independent variable is applied and, for each sub-interval j , time is replaced with:

$$\varepsilon = j - 1 + \frac{t - t_{j-1}}{t_j - t_{j-1}} = j - 1 + \frac{t - t_{j-1}}{\tau_j} \quad (3.17)$$

where $\tau_j = t_j - t_{j-1}$ is the duration (generally unknown) of the sub-interval. By doing so, the extremes of the integration of each sub-interval are fixed and correspond to consecutive integer values of the new independent variable ε at the boundaries (inner and outer ones). The description of the *shooting method* for the solution of the MPBVP is given by referring to the generic system:

$$\frac{dy}{dt} = \mathbf{f}^*(\mathbf{y}, t) \quad (3.18)$$

where the state variables and the adjoint variables are now directly grouped in the vector $\mathbf{y} = (\mathbf{x}, \boldsymbol{\lambda})$ while controls \mathbf{u} are substituted by the expression defined by Eq. (3.11). Since the problem is also defined by constant parameters, such as the duration τ_j of the sub-intervals and eventual values of the variables after a discontinuity, it is convenient to introduce a new vector $\mathbf{z} = (\mathbf{y}, \mathbf{c})$ that groups

together the state variables, the adjoint variables and the vector \mathbf{c} of the constant parameters. Moreover, the replacement of time with the new independent variable ε yields:

$$\frac{dz}{d\varepsilon} = \mathbf{f}(\mathbf{z}, \varepsilon) \quad (3.19)$$

which can be made explicit by writing:

$$\frac{dz}{d\varepsilon} = \left(\frac{dy}{d\varepsilon}, \frac{dc}{d\varepsilon} \right) \quad (3.20)$$

where

$$\frac{dy}{d\varepsilon} = \tau_j \frac{dy}{dt} \quad (3.21)$$

$$\frac{dc}{d\varepsilon} = 0 \quad (3.22)$$

The boundary conditions are generally expressed by grouping together imposed B.C. and B.C. for optimality in the following vector:

$$\Psi(\mathbf{s}) = 0 \quad (3.23)$$

where \mathbf{s} is a vector that contains the values that the (state and adjoint) variables assume at the inner and outer boundaries ($\varepsilon = 0, 1, \dots, n$), as well as the unknown parameters:

$$\mathbf{s} = (\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_n, \mathbf{c}) \quad (3.24)$$

Some of the initial values of the variables are unknown. As already stated, the solution to the MPBVP is obtained with an iterative numerical method that looks for the initial values such that the integration of the differential equations satisfies Eq. (3.23). The method is here described by assuming none of the initial values is known. The r -th iteration is started by integrating Eq. (3.19) with the initial values \mathbf{p}^r derived from the previous iteration. It is set:

$$\mathbf{z}(0) = \mathbf{p}^r \quad (3.25)$$

The integration is carried out for the entire trajectory (if r is the first iteration, tentative values for \mathbf{p}^r are chosen). The values of the state variables are determined at each boundary and the errors on the boundary conditions Ψ^r are calculated at each r -th iteration every time the integration process ends.

A variation $\Delta \mathbf{p}$ results in a variation of the errors on the boundary conditions equal to:

$$\Delta \Psi = \left[\frac{\partial \Psi}{\partial \mathbf{p}} \right] \Delta \mathbf{p} \quad (3.26)$$

where higher than first order terms are neglected. Since the objective is reducing the errors on boundary conditions to zero, the goal for each next iteration is achieving $\Delta\Psi = -\Psi^r$. In virtue of this observation, the initial values are progressively corrected by a quantity equal to:

$$\Delta\mathbf{p} = \mathbf{p}^{r+1} - \mathbf{p}^r = - \left[\frac{\partial\Psi}{\partial\mathbf{p}} \right]^{-1} \Psi^r \quad (3.27)$$

where iteration $r + 1$ is started by integrating the differential equations with initial values, indeed $\mathbf{p}^{r+1} = \mathbf{p}^r + \Delta\mathbf{p}$. The iterations are performed until the boundary conditions are satisfied with the wanted precision. The matrix from Eq. (3.26) can be calculated as the product of two other matrices:

$$\left[\frac{\partial\Psi}{\partial\mathbf{p}} \right] = \left[\frac{\partial\Psi}{\partial\mathbf{s}} \right] \left[\frac{\partial\mathbf{s}}{\partial\mathbf{p}} \right] \quad (3.28)$$

The gradient of boundary conditions with respect to vector \mathbf{s} is easily obtained by analytical derivation. On the other hand, the derivative of vector \mathbf{s} with respect to vector of initial values \mathbf{p} is equivalent to the matrix which contains the derivative of vector \mathbf{z} calculated at (inner and outer) boundaries $\varepsilon = (0, 1, \dots, n)$:

$$[\mathbf{g}(\varepsilon)] = \left[\frac{\partial\mathbf{z}}{\partial\mathbf{p}} \right] \quad (3.29)$$

obtained by taking the derivative of the system in Eq. (3.19) with respect to the vector of initial values \mathbf{p} leads to:

$$[\dot{\mathbf{g}}(\varepsilon)] = \frac{d}{d\varepsilon} \left[\frac{\partial\mathbf{z}}{\partial\mathbf{p}} \right] = \left[\frac{\partial}{\partial\mathbf{p}} \left(\frac{d\mathbf{z}}{d\varepsilon} \right) \right] = \left[\frac{\partial\mathbf{f}}{\partial\mathbf{p}} \right] \quad (3.30)$$

Defining the Jacobian matrix from Eq. (3.19), i.e. $[\partial\mathbf{f}/\partial\mathbf{z}]$, the above Eq. (3.30) can be further developed:

$$[\dot{\mathbf{g}}(\varepsilon)] = \left[\frac{\partial\mathbf{f}}{\partial\mathbf{z}} \right] \left[\frac{\partial\mathbf{z}}{\partial\mathbf{p}} \right] = \left[\frac{\partial\mathbf{f}}{\partial\mathbf{z}} \right] [\mathbf{g}(\varepsilon)] \quad (3.31)$$

The initial values for the homogeneous system (3.31) are easily obtained by taking the derivative of Eq. (3.25) with respect to vector \mathbf{p} , and thus obtaining the identity matrix:

$$[\mathbf{g}(0)] = \left[\frac{\partial\mathbf{z}(0)}{\partial\mathbf{p}} \right] = [\mathbf{I}] \quad (3.32)$$

This method allows to deal with discontinuities of variables. Indeed, if a discontinuity occurs at generic boundary j , vector \mathbf{z} and matrix $[\mathbf{g}]$ can be updated through the below relation of \mathbf{h} that relates the values of the variables before and after the discontinuity:

$$\mathbf{z}_{j_+} = \mathbf{h}(\mathbf{z}_{j_-}) \quad (3.33)$$

$$[\mathbf{g}_{j_+}] = \left[\frac{\partial \mathbf{h}}{\partial \mathbf{z}} \right] [\mathbf{g}_{j_-}] \quad (3.34)$$

That is the reason why when vector \mathbf{s} has been defined, the distinction between vectors \mathbf{y}_{j_+} and \mathbf{y}_{j_-} has been neglected: one is a known function of the other thanks to \mathbf{h} and vector \mathbf{c} . If some of the initial values of the variables are known, the problem is simplified, the vector \mathbf{p} just contains the estimation of unknown initial values of the variables $\mathbf{z}(0)$, while vector Ψ only contains the boundary conditions that are not explicit at the initial time (hence the term *implicit conditions*).

Since the procedure described above for the determination of the error gradient matrix $[\partial\Psi/\partial\mathbf{p}]$ takes a heavy analytical effort and the computational time may be quite long, another alternative method can be easily adopted. It is about evaluating numerically the error gradient matrix: row i of the matrix is obtained by perturbing the i -th component of \mathbf{p} by a small variation $\Delta\mathbf{p}$ and subsequently by integrating the Eq. (3.19). By doing so for each component of \mathbf{p} it is possible to evaluate the variation of the errors $\Delta\Psi(\Delta\mathbf{p})$ and the corresponding i -th row as $\Delta\Psi^T/\Delta\mathbf{p}$ by a linearization procedure. Empirical values for $\Delta\mathbf{p}$ are in the order of $10^{-6} \div 10^{-7}$. Although this other method is rather faster than the one described earlier, it may not ensure numerical convergence. Its implementation is therefore dependent on the complexity and sensitivity of the problem.

Such numerical procedure might be also used to calculate the Jacobian matrix $[\partial\mathbf{f}/\partial\mathbf{z}]$ and the matrix $[\partial\Psi/\partial\mathbf{s}]$: however analytical procedure is here preferred and a comparison between analytical and numerical results of both matrices is anyway implemented and performed to verify accuracy and precision [8, 9].

4

Model Definition

Consider a spacecraft, be it a lander or a rover, close to a planetary object. To define a proper descent trajectory, it is demanded to move the vehicle from a known initial position and velocity state through a given constraint frame, which will bring to the specified final position and velocity state. The goal is to end the whole maneuver in an optimal manner, for which the objective function is yet to be defined. Being the purpose the development of a guidance model, a series of assumptions will be made.

4.1 Assumptions

Assumption 1: Atmosphere-related forces can be neglected.

For celestial objects without atmosphere such as the moon or minor planets known as asteroids, this assumption is very solid and plausible. For this reason, relevant aerodynamic forces cannot be generated and will not be considered.

Assumption 2:: Planetary object assumed to be flat.

During the approaching and landing phases, the position of the space vehicle along the trajectory is so close to the moon surface that it is possible to neglect its curvature.

Assumption 3: Gravitational acceleration is constant.

Due to the fact that the radial distance from the center of the planetary object

during the descent changes marginally, the gravitational acceleration can be reasonably assumed constant. It will be demonstrated that the resulting classic bilinear tangent steering law is based on the same simplification of a uniform gravitational field. In other words, in the case of a planetary landing, distances like downrange and altitude, are quite small in relation to the planet's radius. Therefore, the assumption of a constant gravity field with flat ground is pretty reasonable.

Assumption 4: The rotational motion of the celestial body can be neglected.

Alongside the constant gravitational acceleration assumption, neglecting rotation makes feasible modelling the planetary object as flat.

Assumption 5: A perfectly expanded chemical rocket engine is carried by the vehicle to generate a proper thrust force.

The vehicle is able to generate a thrust force by expelling propellant through the usage of a chemical rocket engine. The thrust force is directly proportional to the rate of propellant flow, the exit speed of the propellant from the rocket engine, the difference between the exterior atmospheric pressure and the pressure at the rocket engine nozzle exit, and the area of the nozzle exit, described mathematically by:

$$T = -\dot{m}V_{\text{exit}} + (p_{\infty} - p_{\text{exit}})A_{\text{exit}} \quad (4.1)$$

where:

- T = thrust magnitude
- \dot{m} = mass flow rate
- V_{exit} = nozzle exit velocity
- p_{∞} = atmospheric pressure
- p_{exit} = nozzle exit pressure
- A_{exit} = nozzle exit area

Under the assumption that the rocket nozzle is perfectly expanded (i.e. $p_{\text{exit}} = p_{\infty}$), the thrust magnitude and the mass flow rate are linked by:

$$T = -\dot{m}V_{\text{exit}} = -\dot{m}g_o I_{\text{sp}} \quad (4.2)$$

where:

- g_o = gravity at the Earth's surface
- I_{sp} = specific impulse of rocket

Rearranging Eq. (4.2) results in:

$$\dot{m} = -\frac{T}{g_o I_{\text{sp}}} = -\frac{T}{c} \quad (4.3)$$

Assumption 6: The thrust magnitude is bounded.

The maximum rate at which propellant mass can be expelled from the chemical rocket engine correspond to the thrust upper limit, while the absolute minimum rate should be zero. Nonetheless, there might be operating status where the minimum rate is nonzero, just to guarantee safety in power supply the propulsion system by not shutting down the engines. So limits on propellant mass flow rate are directly converted into limits on thrust magnitude.

$$0 \leq T_{\min} \leq T \leq T_{\max} \quad (4.4)$$

Assumption 7: The thrust direction has no limits and can be changed instantaneously.

With the aim of developing a guidance algorithm, authentic control system limitations on attitude rates, piloting functions and accelerations are ignored. In any case, it is notable that these properties cannot be disregarded when testing an integrated GN&C system. These kind of assumptions are pretty common in the early stages of a guidance study as shown by J.R. Rea in his work [31].

4.2 Objective Function

Since this is an optimization problem, it is necessary to define the objective function or *functional* (i.e. a real-valued function on a space of functions). Indeed, among all the successful or admissible control variables which could satisfy a given landing problem with its boundary conditions and constraints, from an engineering point of view, optimization gives a truly useful design principle, so it is appropriate and convenient to find the set of control variables which, not only match the whole problem dynamics and conditions, but also minimize or maximize a given objective function J . In this case, it is desired to find the descent and landing trajectory to minimize the fuel consumption through the employment of retrofired velocity as soon as the powered descent phase starts. Therefore, the minimum fuel problem is equal to the maximum final landing mass problem. With that in mind, the optimal trajectory to maximize the final landing mass is designed, and the objective function for this problem is defined as follows:

$$J = m_f \quad (4.5)$$

with the aim to

maximize J

where:

m_f = final landing mass

The form chosen to define J corresponds to the Mayer form $J = \varphi(x_{(t_f)}, t_f)$, here preferred, and it represents the so-called *terminal cost*. This form allows for optimization of some final condition criterion and can be used for many optimal control tasks. Remember that Mayer form is derived by putting the Lagrangian term or *running cost* equal to zero. The sum of both cost terms is known as Bolza form. The *functional* J can be thought of as the *profit* to be maximized by the related and desired set of optimal controls.

4.3 Equations of Motion

A cartesian coordinate system will be used. Let the Z-axis be along the vertical direction with the origin at an altitude of zero. The X-axis and Y-axis form the horizontal plane, and represent the *downrange* and the *crossrange* respectively (Fig. 4.1). The only forces acting on the vehicle are the force due to gravity and its own thrust force.

$$\mathbf{F} = m\mathbf{g} + T\hat{\mathbf{u}} = m\dot{\mathbf{v}} = m\ddot{\mathbf{r}} \quad (4.6)$$

where:

$\mathbf{r} = R_x\hat{\mathbf{i}} + R_y\hat{\mathbf{j}} + R_z\hat{\mathbf{k}}$	radius vector
$\mathbf{v} = V_x\hat{\mathbf{i}} + V_y\hat{\mathbf{j}} + V_z\hat{\mathbf{k}}$	velocity vector
$\mathbf{g} = -g\hat{\mathbf{k}}$	gravity acceleration
$\hat{\mathbf{u}} = u_x + u_y + u_z$	unit thrust direction vector

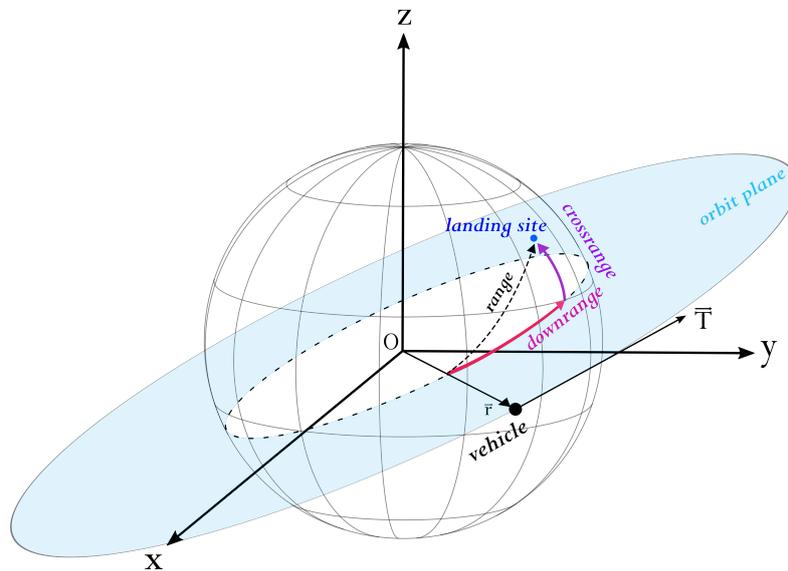


Fig. 4.1: Downrange and Crossrange

Keep in mind that a planetary landing is characterized by fast dynamics. The expected time of flight is in the order of magnitude of minutes, and the mass is supposed to significantly change during the maneuver. A three-dimensional frame is chosen to shape the dynamics.

Dynamics are described by the following set of equations:

$$\begin{cases} \dot{\mathbf{r}} = \frac{d\mathbf{r}}{dt} = \mathbf{v} \\ \dot{\mathbf{v}} = \frac{d\mathbf{v}}{dt} = \mathbf{g} + \frac{\mathbf{T}}{m} = \mathbf{g} + \mathbf{a}_T \\ \dot{m} = \frac{dm}{dt} = -\frac{T}{g_o I_{sp}} = -\frac{T}{c} \end{cases} \quad (4.7)$$

4.4 Controls

It is necessary to find the thrust vector time history or control profile $\mathbf{T}(t)$ throughout the entire trajectory path (from the initial to the target final states) that achieves the desired fuel optimal maneuver, integrating all with the constraints dictated by the chosen system architecture. In the dynamic system, the thrust vector T acts as a control variable. The main thruster is supposed to be tightly docked to the spacecraft lower part. Hence, the direction of the thrust vector is instantly determined by the spacecraft attitude. The spacecraft attitude is described by an supplemental reference system, called Flight Reference System (FRS), constituted by the unit vector $\hat{\mathbf{u}} = [u_x, u_y, u_z]^T$, centered in the center of mass of the spacecraft. The u_x axis points toward the downrange direction, the u_z axis points upwards, and the u_y axis forms a right-handed triad. It is through all these last, along the previous, assumptions that it can be stated that $\hat{\mathbf{u}}$ is another control variable and corresponds to the unit thrust direction vector.

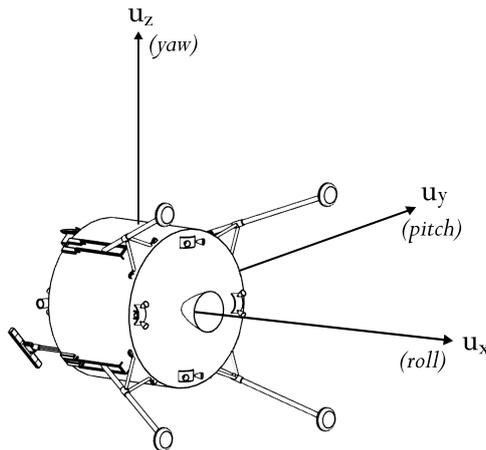


Fig. 4.2: Flight Reference System

4.5 Boundary Conditions and Constraints

It is a reasonable requirement to bound the problem with specific initial and final conditions, in addition to physic and engineering constraints. Uppercase letters symbolize vector components.

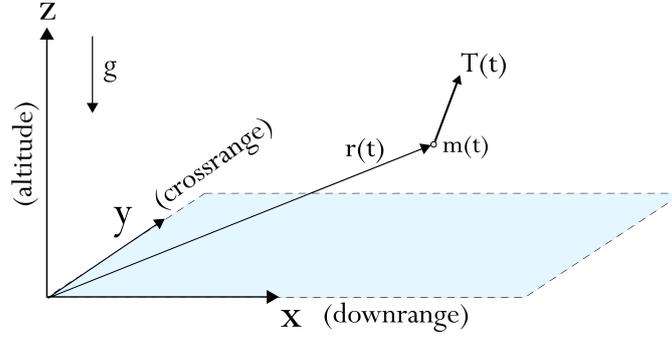


Fig. 4.3: Ground Reference System

Let's define the initial conditions:

$$\chi_0 = \mu^T \begin{cases} (R_{x_0} - R_{x_{0s}}) = 0 \\ (R_{y_0} - R_{y_{0s}}) = 0 \\ (R_{z_0} - R_{z_{0s}}) = 0 \\ (V_{x_0} - V_{x_{0s}}) = 0 \\ (V_{y_0} - V_{y_{0s}}) = 0 \\ (V_{z_0} - V_{z_{0s}}) = 0 \\ (m_0 - m_{0s}) = 0 \end{cases} \quad (4.8)$$

followed by the final conditions:

$$\chi_f = \mu^T \begin{cases} (R_{x_f} - R_{x_{fs}}) = 0 \xrightarrow{R_{x_{fs}}=0} (R_{x_f} = 0) \\ (R_{y_f} - R_{y_{fs}}) = 0 \xrightarrow{R_{y_{fs}}=0} (R_{y_f} = 0) \\ (R_{z_f} - R_{z_{fs}}) = 0 \xrightarrow{R_{z_{fs}}=0} (R_{z_f} = 0) \\ (V_{x_f} - V_{x_{fs}}) = 0 \xrightarrow{V_{x_{fs}}=0} (V_{x_f} = 0) \\ (V_{y_f} - V_{y_{fs}}) = 0 \xrightarrow{V_{y_{fs}}=0} (V_{y_f} = 0) \\ (V_{z_f} - V_{z_{fs}}) = 0 \xrightarrow{V_{z_{fs}}=0} (V_{z_f} = 0) \\ m_f \text{ free} \end{cases} \quad (4.9)$$

where:

- $\boldsymbol{\mu}^T$ = adjoint constants associated with B.C.
- \mathbf{r}_0 = initial position vector
- \mathbf{r}_f = final position vector
- \mathbf{v}_0 = initial velocity vector
- \mathbf{v}_f = final velocity vector
- \square_s = specified value

The system of Eqs. (4.9) represents the case with no crossrange component, indeed V_{yfs} is set to zero, but it must be taken into account that it can be a non-zero value in case of crossrange diversion maneuver. The trajectory might be divided up into three arcs, associated with the *Breaking Phase*, *Approach Phase* and *Terminal Vertical Descent* respectively. It could be useful or even necessary if some internal conditions must be introduced at certain points, some of which may concern attitude feedback. Another fundamental variable to be defined is the **time**. In this case the initial time t_0 is fixed, while the final time t_f is set free.

A constraint, already described in the assumptions, involves the thrust magnitude T . The domain has already been depicted by the Eq. (4.4) and the thrust magnitude limits are reported again:

$$0 \leq T_{\min} \leq T \leq T_{\max}$$

To definitely complete the whole picture is necessary to point out a feature related to the unit thrust direction vector \hat{u} . By definition it must be a unit vector, so mathematically we can express this condition through its components by imposing:

$$u_x^2 + u_y^2 + u_z^2 = 1 \quad (4.10)$$

Now the problem can be solved with the techniques of the optimal control theory previously introduced in Chapter 3.

5

Application of Optimal Control

The problem can be now stated mathematically as an optimal control problem. Two angles ϕ and ψ^1 are introduced to properly project the direction of the thrust vector \mathbf{T} along its three components: $T_x = Tu_x$, $T_y = Tu_y$ and $T_z = Tu_z$.

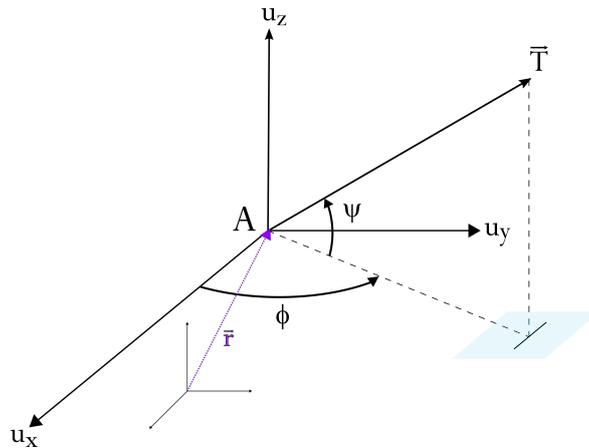


Fig. 5.1: Thrust Reference Frame

Remind that the main objective of an optimal control problem is to find the set of control variables (T, \hat{u}) capable of

maximizing $J = m_f$

¹Note that angle ψ is not equal to B.C. vector Ψ defined in Chapter 3.

subjected to the dynamic Eqs. (4.7), hereinafter expressed in their scalar form with explicit components:

$$\begin{cases} \dot{R}_x = V_x \\ \dot{R}_y = V_y \\ \dot{R}_z = V_z \\ \dot{V}_x = \frac{T}{m} u_x + \cancel{g_x}^0 = \frac{T}{m} \cos \psi \cos \phi \\ \dot{V}_y = \frac{T}{m} u_y + \cancel{g_y}^0 = \frac{T}{m} \cos \psi \sin \phi \\ \dot{V}_z = \frac{T}{m} u_z + g_z = \frac{T}{m} \sin \psi - g \\ \dot{m} = -\frac{T}{c} \end{cases} \quad (5.1)$$

The first function to be defined is the *Hamiltonian*, given by:

$$H = \lambda_r \mathbf{v} + \lambda_v \left(\frac{T}{m} \hat{\mathbf{u}} + \mathbf{g} \right) + \lambda_m \left(-\frac{T}{c} \right) \quad (5.2)$$

that can be rewritten as follows to highlight each component:

$$\begin{aligned} H &= \lambda_{R_x} V_x + \lambda_{R_y} V_y + \lambda_{R_z} V_z \\ &\quad + \lambda_{V_x} \left(\frac{T}{m} \cos \psi \cos \phi \right) \\ &\quad + \lambda_{V_y} \left(\frac{T}{m} \cos \psi \sin \phi \right) \\ &\quad + \lambda_{V_z} \left(\frac{T}{m} \sin \psi - g \right) \\ &\quad + \lambda_m \left(-\frac{T}{c} \right) \end{aligned} \quad (5.3)$$

The next step is about the calculus of the Lagrange multipliers λ , specifically those defined as *adjoint* or *costate variables*. Their explicit expressions are given by the Eulero-Lagrange differential equations, here adapted to the model and derived below:

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

where:

$$\mathbf{x} = [\mathbf{r}, \mathbf{v}, m]^T \quad \text{state variable vector}$$

1° State Variable: $x = \mathbf{r}$

$$\frac{d\boldsymbol{\lambda}}{dt} = -\left(\frac{\partial H}{\partial \mathbf{r}}\right)^T \rightarrow \begin{cases} \dot{\lambda}_{R_x} = 0 \\ \dot{\lambda}_{R_y} = 0 \\ \dot{\lambda}_{R_z} = 0 \end{cases} \quad (5.4)$$

2° State Variable: $x = \mathbf{v}$

$$\frac{d\boldsymbol{\lambda}}{dt} = -\left(\frac{\partial H}{\partial \mathbf{v}}\right)^T \rightarrow \begin{cases} \dot{\lambda}_{V_x} = -\lambda_{R_x} \\ \dot{\lambda}_{V_y} = -\lambda_{R_y} \\ \dot{\lambda}_{V_z} = -\lambda_{R_z} \end{cases} \quad (5.5)$$

3° State Variable: $x = m$

$$\frac{d\boldsymbol{\lambda}}{dt} = -\left(\frac{\partial H}{\partial m}\right)^T \rightarrow \begin{cases} \dot{\lambda}_m = \lambda_{V_x} \left(\frac{T}{m^2} \cos \psi \cos \phi\right) \\ \quad + \lambda_{V_y} \left(\frac{T}{m^2} \cos \psi \sin \phi\right) \\ \quad + \lambda_{V_z} \left(\frac{T}{m^2} \sin \psi\right) \end{cases} \quad (5.6)$$

First of all, it is noticeable that Euler-Lagrange differential equations do not depend on the objective function J . In addition, it is possible to reshuffle the equations by introducing what has been defined *primer vector*² $\mathbf{p}(t)$ by D.F. Lawden [23]. Since Pontryagin's Maximum Principle is applied, it is well-known that the main goal is the calculus of those control variables that maximize H , as functional J is what it has to be maximized. The control variables of the current problem are $[T, \hat{u}]$, that is to say, thrust magnitude and thrust direction. By inspection, the Hamiltonian H of equation Eq. (5.2) is maximized over the choice of thrust direction by aligning the unit thrust direction vector \hat{u} parallel to the adjoint variable λ_v . Mathematically describable as:

$$\mathbf{p}(t) = \boldsymbol{\lambda}_v(t) \quad (5.7)$$

The optimal unit thrust direction vector is then in the direction of the primer vector, so:

²For impulsive trajectories, the primer vector determines the times and positions of the thrust impulses that maximize the final mass. For continuous thrust trajectories, both the optimal thrust direction and the optimal thrust magnitude as functions of time are established by the primer vector.

$$\hat{\mathbf{u}} = \frac{\mathbf{p}(t)}{\|\mathbf{p}(t)\|} = \frac{\lambda_v(t)}{\|\lambda_v(t)\|} \quad (5.8)$$

Now let's consider each component of the thrust vector $\mathbf{T} = T\hat{\mathbf{u}}$, bearing in mind what has been just stated in Eq. (5.8), it brings about:

$$\begin{aligned} u_x &= \frac{T_x}{\|T\|} = \frac{T \cos \psi \cos \phi}{T} = \frac{\lambda_{V_x}}{\|\lambda_v\|} \\ u_y &= \frac{T_y}{\|T\|} = \frac{T \cos \psi \sin \phi}{T} = \frac{\lambda_{V_y}}{\|\lambda_v\|} \\ u_z &= \frac{T_z}{\|T\|} = \frac{T \sin \psi}{T} = \frac{\lambda_{V_z}}{\|\lambda_v\|} \end{aligned} \quad (5.9)$$

Recalling also that:

$$\|\lambda_v\|^2 = \lambda_{V_x}^2 + \lambda_{V_y}^2 + \lambda_{V_z}^2$$

It is now possible to redefine Eq. (5.6) as follows:

$$\frac{d\lambda}{dt} = -\left(\frac{\partial H}{\partial m}\right)^T \rightarrow \begin{cases} \dot{\lambda}_m = \frac{\lambda_{V_x}^2}{\|\lambda_v\|} \left(\frac{T}{m^2}\right) \\ \quad + \frac{\lambda_{V_y}^2}{\|\lambda_v\|} \left(\frac{T}{m^2}\right) \\ \quad + \frac{\lambda_{V_z}^2}{\|\lambda_v\|} \left(\frac{T}{m^2}\right) \\ = \|\lambda_v\| \left(\frac{T}{m^2}\right) \end{cases} \quad (5.10)$$



Fig. 5.2: Lander 3D-Model

5.1 Two-Dimensional Motion Case

Let's imagine that the vehicle motion is temporarily confined to the X-Z plane. In other words, the crossrange parameter is neglected and not considered at the moment. Due to the fact that crossrange, i.e. Y-axis, is not an active direction of motion:

$$\alpha = 0^\circ$$

$$\phi = 0^\circ$$

By imposing these two stricter requirements, all the equations seen so far can be further simplified. It means that neither the Y-component of the position vector R_y nor the Y-direction of thrust vector T_y are allowed, just as V_y . By doing so, all the vectors are deprived of their Y-component, and therefore their size is reduced from three components to two components.

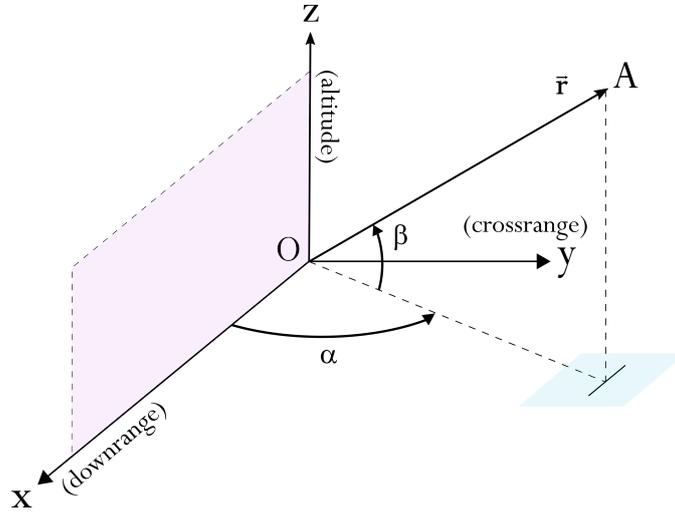


Fig. 5.3: X-Z plane of Motion

The dynamic model is reduced and becomes:

$$\begin{cases} \dot{R}_x = V_x \\ \dot{R}_z = V_z \\ \dot{V}_x = \frac{T}{m} u_x = \frac{T}{m} \cos \psi \\ \dot{V}_z = \frac{T}{m} u_z + g_z = \frac{T}{m} \sin \psi - g \\ \dot{m} = -\frac{T}{c} \end{cases} \quad (5.11)$$

The Hamiltonian is rearranged as follows:

$$\begin{aligned}
 H &= \lambda_{R_x} V_x + \lambda_{R_z} V_z \\
 &+ \lambda_{V_x} \left(\frac{T}{m} \cos \psi \right) \\
 &+ \lambda_{V_z} \left(\frac{T}{m} \sin \psi - g \right) \\
 &+ \lambda_m \left(-\frac{T}{c} \right)
 \end{aligned} \tag{5.12}$$

Considerations made about *primer vector* are still valid, in a nutshell:

$$\begin{aligned}
 u_x &= \frac{T_x}{\|T\|} = \frac{T \cos \psi}{T} = \frac{\lambda_{V_x}}{\|\lambda_v\|} \\
 u_z &= \frac{T_z}{\|T\|} = \frac{T \sin \psi}{T} = \frac{\lambda_{V_z}}{\|\lambda_v\|}
 \end{aligned} \tag{5.13}$$

Paying also attention to the costate variable λ_v , which is also reduced in size:

$$\|\lambda_v\|^2 = \lambda_{V_x}^2 + \lambda_{V_z}^2$$

It is now possible to go ahead with the calculus of adjoint variables λ , in the same manner it has been done in the previous section, but this time the Euler-Lagrange equation leads to a slightly different and shorter representation:

1° State Variable: $x = \mathbf{r}$

$$\frac{d\lambda}{dt} = -\left(\frac{\partial H}{\partial \mathbf{r}}\right)^T \longrightarrow \begin{cases} \dot{\lambda}_{R_x} = 0 \\ \dot{\lambda}_{R_z} = 0 \end{cases} \tag{5.14}$$

2° State Variable: $x = \mathbf{v}$

$$\frac{d\lambda}{dt} = -\left(\frac{\partial H}{\partial \mathbf{v}}\right)^T \longrightarrow \begin{cases} \dot{\lambda}_{V_x} = -\lambda_{R_x} \\ \dot{\lambda}_{V_z} = -\lambda_{R_z} \end{cases} \tag{5.15}$$

3° State Variable: $x = \mathbf{m}$

$$\frac{d\lambda}{dt} = -\left(\frac{\partial H}{\partial \mathbf{m}}\right)^T \longrightarrow \begin{cases} \dot{\lambda}_m = \lambda_{V_x} \left(\frac{T}{m^2} \cos \psi \right) \\ \quad + \lambda_{V_z} \left(\frac{T}{m^2} \sin \psi \right) \end{cases} \tag{5.16}$$

According to Optimal Control Theory, not only Euler-Lagrange equations must be satisfied, but also the algebraic equation of controls, displayed just below:

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

where:

$$\mathbf{u} = [\psi, T]^T \quad \text{control variables vector}$$

Indeed it is absolutely legit assuming that the thrust direction $\hat{u} = [u_x, u_z]^T$ can be controlled by ψ angle, which in turn controls the lander motion on the X-Z plane. Taking as a reference the Hamiltonian in the Eq. (5.12), it yields to:

$$\boxed{\text{1° Control Variable: } u = \psi}$$

$$\left(\frac{\partial H}{\partial \psi}\right)^T = 0 \longrightarrow \left\{ -\frac{T}{m} \left(\lambda_{V_x} \sin \psi - \lambda_{V_z} \cos \psi \right) = 0 \right. \quad (5.17)$$

Solving the above equation leads to the sought control law, formulated as follows:

$$\boxed{\tan \psi^* = \frac{\lambda_{V_z}}{\lambda_{V_x}}} \quad (5.18)$$

It has been achieved the so-called *Bilinear Tangent Steering Law*, in short, the law that represents the optimal value \square^* of the control variable ψ , in case of two-dimensional motion, throughout the whole trajectory for each instant of time.

$$\boxed{\text{2° Control Variable: } u = T}$$

Along with ψ , the other control variable is the thrust magnitude T , which is bounded by its own admissible region. Now looking at the Hamiltonian, it is clear that its derivative is linear with respect to T , hence when the algebraic equation of controls is used, the control itself does not appear explicitly and it cannot be calculated directly. Let's rearrange the Hamiltonian for a moment, applying also the definitions from the Eqs. (5.13), which brings to:

$$\begin{aligned} H &= \lambda_{R_x} V_x + \lambda_{R_z} V_z \\ &+ \left[\frac{\lambda_{V_x}^2 + \lambda_{V_z}^2}{\|\lambda_v\|} \right] \left(\frac{T}{m} \right) \\ &+ \lambda_m \left(-\frac{T}{c} \right) - \frac{\lambda_{V_z}}{g} \end{aligned} \quad (5.19)$$

And finally, collecting properly T as a common factor:

$$\begin{aligned}
 H &= \lambda_{R_x} V_x + \lambda_{R_z} V_z \\
 &+ \left(\frac{\lambda_v}{m} - \frac{\lambda_m}{c} \right) T \\
 &- \frac{\lambda_{V_z}}{g}
 \end{aligned} \tag{5.20}$$

Given that the coefficient associated with the control variable T is not null, then the control T falls under the category of what are defined *bang-bang controls*. Based on what Pontryagin's Maximum Principle states, it can be claimed that:

$$H \text{ is maximized by } \longrightarrow \begin{cases} T = T_{max} & \text{if } SF > 0 \\ T = T_{min} & \text{if } SF < 0 \end{cases} \tag{5.21}$$

where:

$$SF = \left(\frac{\lambda_v}{m} - \frac{\lambda_m}{c} \right) \tag{5.22}$$

and it constitutes the so-called *Switching Function*. It is meant as the optimal time balance between how much thrust must be used, thus when ignite and shut-off engines, and consequently how much mass is convenient to consume or save along the whole trajectory according to the entire set of necessary conditions for optimality. Regarding the **Three-Dimensional Motion Case**, at the beginning of the chapter it has been proved that all the steps are basically the same. Algebraic control variable equations have been calculated just for 2-D case for convenience, but with a proper transposition formulas maintain the same expression and meaning even for 3-D case. The analytical solution has been omitted due to its high complexity. Numerical methods are definitely preferred and convenient to solve all the differential equations that comes from *OCT* application. Just remind that in a 3-D framework, beyond thrust magnitude T and ψ , the other control variable will be ϕ , given that crossrange is now an active direction of motion. So all the y -component are restored, and thrust angles can be written as follows³ (see Fig. 5.1):

$$\tan \psi^* = \frac{\lambda_{V_z}}{\sqrt{\lambda_{V_x}^2 + \lambda_{V_y}^2}} \quad \tan \phi^* = \frac{\lambda_{V_y}}{\lambda_{V_x}} \tag{5.23}$$

The new control variables vector will be $\mathbf{u} = [\psi, \phi, T]^T$. As generally applies with 2-D case, even for 3-D case the related algebraic equations of controls derived from *OCT* will provide the optimal control values for each instant of time.

³Note that if $\lambda_{R_y} = 0$ and $\lambda_{V_y} = 0$, then 2-D case is re-established.

6

Results

The current chapter presents the results for a case study regarding a mid-size lander designed to deliver payloads of up to 300 kg. The mission concept starts in an initial 100 km circular low lunar orbit: since we are dealing with the Powered Descent Phase, i.e. the final phase of flight leading to touchdown, it is necessary to know the velocity at which the lander approaches the moon surface. Therefore a *Hohmann*-transfer orbit with a lowered periapsis (*perilune* since it is the Moon) of around 15 km has been considered as the nominal value for the PDI altitude, at which the expected velocity is equal to 1.685 km/s, with the lander still horizontally oriented. The Braking Phase is generally started at around 500 km downrange from the landing site with the Powered Descent Initiation (PDI), during which the early introduced initial speed is then gradually reduced down to zero for a safe touchdown. At the beginning of the descent, it is assumed that the mass of the lander is equal to 1500 kg, while the propulsion system is supposed to be able to deliver up to 5000 N of thrust with a specific impulse of almost 290 s. Mass, thrust and altitude values draw inspiration directly from Apollo's Lunar Module, besides reducing its weight and thrust availability by a factor of ten, leading to 1:10 scale mission concept, which still remain a full-fledged reasonable scenario. Indeed, the LM-5 weighed about 15000 kg and its engines were able to deliver about 45000 N of thrust power, starting the PDI from an altitude of 15 km above the lunar surface.

The numerical results are obtained by following the procedure described in Section 3.4 and by integrating the differential equations with a variable-step variable order method based on the *Adams-Moulton* method [6]. Variables normalization has been adopted, given that a faster and easier convergence can be obtained when values have similar orders of magnitude. The desired precision for the B.C. is set to 10^{-7} : thereby the maximum error $E_{max} = \max_i \Psi_i$ must be kept below such value. In

accordance with the procedure described in Eq. (3.27) the initial tentative solution is corrected by a quantity $\Delta \mathbf{p}$ computed by neglecting higher than first order terms. Such linearization procedure may lead to values for $\Delta \mathbf{p}$ that would make the error increase instead of decreasing it. In order to avoid to move away from the wanted solution, a relaxation empiric factor K_1 is used, so that:

$$\mathbf{p}^{r+1} = \mathbf{p}^r + K_1 \Delta \mathbf{p} \quad (6.1)$$

where K_1 ranges from 0.1 to 0 depending on whether the initial tentative solution is closer or farther from the desired one. By so doing, such relaxation factor reduces the standard correction $\Delta \mathbf{p}$ and the integration is thus more controlled: since the boundary conditions error can grow after the first iterations, the introduction of this factor allows this behavior but stops the procedure if the error is growing too much. Subsequently, if the maximum error E_{\max}^{r+1} of the iteration $r + 1$ is lower than a multiple of the maximum error E_{\max}^r of the iteration r , the code proceeds with a new iteration. Basically, a new iteration is started if $E_{\max}^{r+1} < K_2 E_{\max}^r$ with $K_2 = 2 \div 3$. On the other hand, if the error is too big with respect to the previous iteration ($E_{\max}^{r+1} > K_2 E_{\max}^r$) a bisection of the correction is performed:

$$\mathbf{p}^{r+1} = \mathbf{p}^r + K_1 \frac{\Delta \mathbf{p}}{2} \quad (6.2)$$

The iteration is then repeated and the new maximum error is compared to the one of the previous iteration. If necessary, the iteration and thus the bisection strategy is repeated up to a maximum of five times, so if and once reached, then the procedure is stopped due to the no convergence.

6.1 Cases with Reference Values

First of all, let's introduce the scaling factors that have been used to properly set up variables to the integration process:

Table 6.1: Scaling Factors

Variables	Units	Values
m_{ref}	[kg]	1500
r_{ref}	[km]	15
v_{ref}	[km/s]	1
a_{ref}	[km/s ²]	$v_{ref}^2 / r_{ref} = 1/15 \approx 0.067$
t_{ref}	[s]	$r_{ref} / v_{ref} = 15$
T_{ref}	[N]	$m_{ref} \cdot a_{ref} \cdot 1000 = 10^5$

As already mentioned, all the outputs provided by the code algorithms are scaled by those factors shown in (Table 6.1) because of variables normalization, which makes the integration calculus more robust and reliable, rearranging quantities around the

same order of magnitude. It means that to get the actual variable's values, above factors must be multiplied by the related quantities once the integration is over and the solutions printed. Taking a look at the units of measure chosen, the gravity acceleration of the moon must be converted to $g_{\alpha} = 0.001623 [km/s^2]$ to ensure unit consistency. The Boundary Value Problem under study consists of one parameter $KP = 1$ and fourteen variables $NY = 14$, where $KP \equiv t_f$ stands for the duration of the whole trajectory arc, while NY variables have been already illustrated in the previous chapters. Therefore, the total size of the BVP is equal to $N = KP + NY = 15$. To solve the problem, same number - 15 - of B.C. are needed, be them explicit-initial or implicit-final ones (Table 6.3). On the other hand, variables are declared and implemented through the differential equations derived from OCT application.

Table 6.2: Parameter and Variables of the BVP

Parameter: KP	Variables: NY
t_f	R_x, R_y, R_z
	V_x, V_y, V_z
	m
	$\lambda_{R_x}, \lambda_{R_y}, \lambda_{R_z}$
	$\lambda_{V_x}, \lambda_{V_y}, \lambda_{V_z}$
	λ_m

Once defined B.C. and differential equations, i.e. Eqs. (3.1), (3.10), (3.11) and (3.16),

Table 6.3: B.C. and Initial Tentative Variables

Explicit Boundary Conditions	Implicit Boundary Conditions	Unknown Initial Values
m_0, λ_{m_0} ¹	H_f	t_f
R_{x_0}	R_{x_f}	$\lambda_{R_{x_0}}$
R_{y_0}	R_{y_f}	$\lambda_{R_{y_0}}$
R_{z_0}	R_{z_f}	$\lambda_{R_{z_0}}$
V_{x_0}	V_{x_f}	$\lambda_{V_{x_0}}$
V_{y_0}	V_{y_f}	$\lambda_{V_{y_0}}$
V_{z_0}	V_{z_f}	$\lambda_{V_{z_0}}$

¹It actually substitutes $\lambda_{m_f} = 1$, derived from the first transversality condition $\lambda_{x_f} = \partial\varphi/\partial x_f$ where the state variable $x_f = m_f$. This interchange is possible thanks to the arbitrary scaling of the problem, where a generic α parameter will guarantee problem consistency allowing for $\lambda_{m_0} = 1$ for any actual value of λ_{m_0} , while $\lambda_{m_f} = 1 \cdot \alpha$. This trick allows to increase the known initial values at the beginning, removing one extra tentative value. The second transversality condition $H_f = -\partial\varphi/\partial t_f$ leads to $H_f = 0$.

one of the major challenge in a BVP is finding the fittest initial tentative solution to start the integration, due to the high sensitivity of the shooting methods. So the unknown initial values are those for which a tentative guess is required to run the code (Table 6.3). To get started, it was decided to modify the final code, called **bound x-range: n1**, changing in it a specific condition: instead of requiring the state variable R_{x_f} to be null, the associated co-state variable $\lambda_{R_{x_f}}$ has been the one put equal to zero, giving rise to what has been called **free x-range: n2**. By doing so, $\lambda_{R_{x_f}}$ is zero, and the final value of downrange R_{x_f} is thus set free. An additional - preliminary - simplification was made also on $n2$, where not only $R_{x_f} = 0$ has been replaced with $\lambda_{R_{x_f}} = 0$, but also $V_{x_f} = 0$, putting $\lambda_{V_{x_f}} = 0$ instead. These further edit generates a *free-fall* test case, called $n3$, from which it is easier to guess time duration t_f and verify earlier control profiles. In this way all the convergent solutions of $n3$ may be used as initial guesses for $n2$, and the same procedure can consequently be done between $n2$ and $n1$, setting up a backward strategy. Thanks to this knack, the probability to find sub-optimal solution is significantly reduced.

Anyway, $n3$ case has been omitted, while $n2$ has been plotted and used as a benchmark to verify the accuracy of $n1$. When dealing with $n2$, if we want to calculate the initial state variable R_{x_i} that ensures convergence to $R_{x_f} = 0$ (being it free), what needs just to be done is computing the difference between initial and final position at the first $n2$ run, getting $k = R_{x_i} - R_{x_f}$. Then, if k is used as the new initial position, convergence to zero is ensured, and $x_f - x_0$ is the optimal downrange: having said that, it is legit to expect that $n1(k) \cong n2(k)$. Once explained the meaning of $n1$, $n2$ and the k parameter, let's introduce the chosen test cases, summarized in Table 6.4. The first three tests, named *NoCross#1/#2/#3*, do not consider any crossrange motion, while test cases *Cross#1/#2* do. Variations on thrust, altitude, downrange and crossrange have been performed to highlight behavior profiles of V_x , V_y , V_z , *mass*, thrust angles ψ , ϕ and trajectory shapes, carrying out in the next sections a detailed comparison to shed light on differences, similarities, pros and cons.

Table 6.4: Test Cases

Parameters		NoCross#1	NoCross#2	NoCross#3	Cross#1	Cross#2
mass	[kg]	1500	1500	1500	1500	1500
thrust	[N]	5000	5000	10000	5000	10000
altitude	[km]	15	10	15	15	15
n2(k): $x_f - x_0$...	373.00	375.00	210.56	373.01	210.56
						209.49
						207.14
n1(k): $x_f - x_0$...	373.00	375.00	210.56	373.01	/
n1(k-2): $x_f - x_0 + 30$...	403.00	405.00	240.56	403.01	/
n1(k-4): $x_f - x_0 + 60$...	433.00	435.00	/	433.01	/
n1(k+1.5): $x_f - x_0 - 22.5$...	/	/	188.06	/	/
diversion j	[km]	/	/	/	± 1.5	$\pm 1.5/3/4.5$

6.1.1 No Crossrange Cases - Comparison

The present section contains the trend-plots of the variables involved with regard to *NoCross* cases, wherein Y -components of the dynamic model are missing, therefore it is examined a temporarily bounded 2-D motion case on X - Z plane. Bear in mind that Z -positive direction is assumed for increasing altitude values, while X -positive direction is assumed to be from negative toward positive downrange values. For each group, the first graph of third stands for the nominal case *NoCross#1*, while a couple of data edits were applied to plot the remaining two *NoCross#2/#3*, in order to investigate profiles tendencies and variations.

The first evident and foreseeable element regards trajectory time duration: no matter what, largest downrange starting point values ($k-2$, $k-4$, ...) correspond to longest time-path, as long as final landing site point R_{x_f} is fixed. The first group of plots (Fig. 6.1) represents the time evolution of the Z -component velocity during the descent arc. One element that stands out when looking at $V_z(t)$ is the different pattern among the curves. Firstly let's consider the first two cases, #1 and #2, which differs only on altitude starting point (which in turn influences a bit the k parameter, for which lower altitude needs slightly larger downrange values to converge): with initial high downrange-distance values from the landing site, such as ($k-2$, $k-4$) trials, curves follow a more linear development. On the contrary, if the trajectory has at its disposal less downrange distance to brake, descent and landing, e.g. k trial, $V_z(t)$ curve turns into a parabolic trend. This means that ($k-2$, $k-4$) test cases, having more room to perform, hold a first trajectory stretch with engines shut-off [$SF < 0$]: that is why at the very beginning V_z increases linearly, being uniquely under the action of lunar gravity acceleration, heading toward negative values (velocity vector is indeed oriented in the opposite direction in respect to Z -axis positive direction, because lander is going down in any case). Note that when lander starts PDI stage (t_0 time), it is horizontally oriented, so the only velocity component initially present is the X -component non-zero value, equal to $V_x = 1.685$ km/s as anticipated, while $V_{z_0} = 0$. Observe that [$SF < 0$] condition, i.e. $T = T_{min} = 0$, lasts longer accordingly to downrange starting point position: $k-2$ starts engines [$SF > 0$] before $k-4$ case. Indeed, looking at *NoCross#1/#2/#3*, regardless of differences on altitude values or thrust power, the Switching Function works in the same manner: the further lander starts PDI, the longer engines are kept off. That is also reflected on $V_x(t)$, where higher downrange starting point records higher velocity at a given and equal time instant. In fact, the guidance algorithm try to save propellant as much as possible, igniting engines when strictly necessary. If downrange starting point is barely sufficient to land the vehicle safely, engines are kept on throughout the entire trajectory, as with k case. Whereas there is not enough space to properly carry out landing, algorithm do not converge to the designated point because of thrust power limits (reason why #1/#2 do not have any $k+\Delta$, being k their upper limit).

Things quite change when #3 case is executed: having available twice the thrust power (from 5000 N to 10000 N), curves behave slightly different. First of all, trajectory time duration drops drastically, be almost halved, and k parameter is dramatically

reduced. So dynamics are much faster and curves' graphic scale influenced by them. $V_z(t)$ curves appears parabolic even though the first part of the trajectory is crossed with engines shut down [$SF < 0$].

The fact that this time a first linear drop of V_z is not recorded in #3, may be attributable to those rapid evolution which "stretch out" curves, in addition to a different thrust angle direction ψ , which represents trust-angle on X - Z plane, as shown in Fig. 6.6: this time, #3 thrust direction ranges from $\psi = 92^\circ$ to $\psi = 69^\circ$, meaning that the algorithm tends to firstly slow down V_x component and then V_z , given that #3 has a much lower braking distance, while #1 and #2 cases behave inversely. Actually, looking at Fig. 6.6 and Fig. 6.7, it is possible to notice that #1 case: $\psi = 59^\circ$ to $\psi = 72^\circ$ while #2 case: $\psi = 56^\circ$ to $\psi = 74^\circ$, which results in higher values of V_z in #1 case other than #2. In other words, if thrust engines are throttled up, angle-thrust direction play a fundamental role in deciding where and how to focus thrust-power availability. Lower values of ψ in #2 at the beginning, basically result in a more vertical thrust direction than #1 case, which in turn results in a powerful braking along Z -axis: that is obvious having #2 less vertical space to brake. However, it must be paid always attention to SF , although ψ ranges differ, it must always be checked if engines are either on or off: so the thrust pointing direction must be always taken into consideration along with SF 's sign. Moreover V_z magnitude in #3 is lower on average, so the descent phase takes less time and it is performed with a more slowly fall than #1/#2 cases, thanks to the higher net thrust #3 has got. Just a little note: *NoCross* cases develop their motion on X - Z plane, so if we consider ϕ angle (i.e. thrust direction along X - Y plane), it will be constantly oriented to -90° . It means that thrust direction permanently lies in X - Z plane, pointing toward X -negative values, i.e. thrust angle ϕ is exactly opposite to V_x direction and that is right if braking is what we are looking for. Furthermore, the guidance algorithm provides a *bang-bang* control of the kind: T_{max} or T_{min} - T_{max} for nearly all cases, while *NoCross#3* ($k+1.5$) is the only one to record a T_{max} - T_{min} - T_{max} control profile, although its T_{min} phase² occurs for a short stretch. Remarkable is the fact that k case in #1/#2 is a T_{max} constant profile only, while in #3 it records a T_{min} - T_{max} control framework, justified by the fact that the higher net thrust availability enables a powerful braking, allowing for an unpowered initial stretch.

All these features can be further and clearer verified through mass time-evolution plots (Fig. 6.5), where engines shut-off is equal to none propellant usage, so $m(t)$ curves appears constant right there. The same consideration can be done for V_x (Fig. 6.2), being not directly affected by the effect of gravity, unlike V_z , since V_x keeps its value constant till $SF < 0$. Given that a soft-landing is required, both V_x and V_z must be zero at the landing point, so at a certain stage of descent trajectory the braking is triggered and $SF < 0$ condition switches to $SF > 0$, which stands for $T = T_{max}$, proving a *bang-bang* control framework. This turnaround can be seen in

² T_{min} phase stands for *coasting phase* in the jargon of space trajectories.

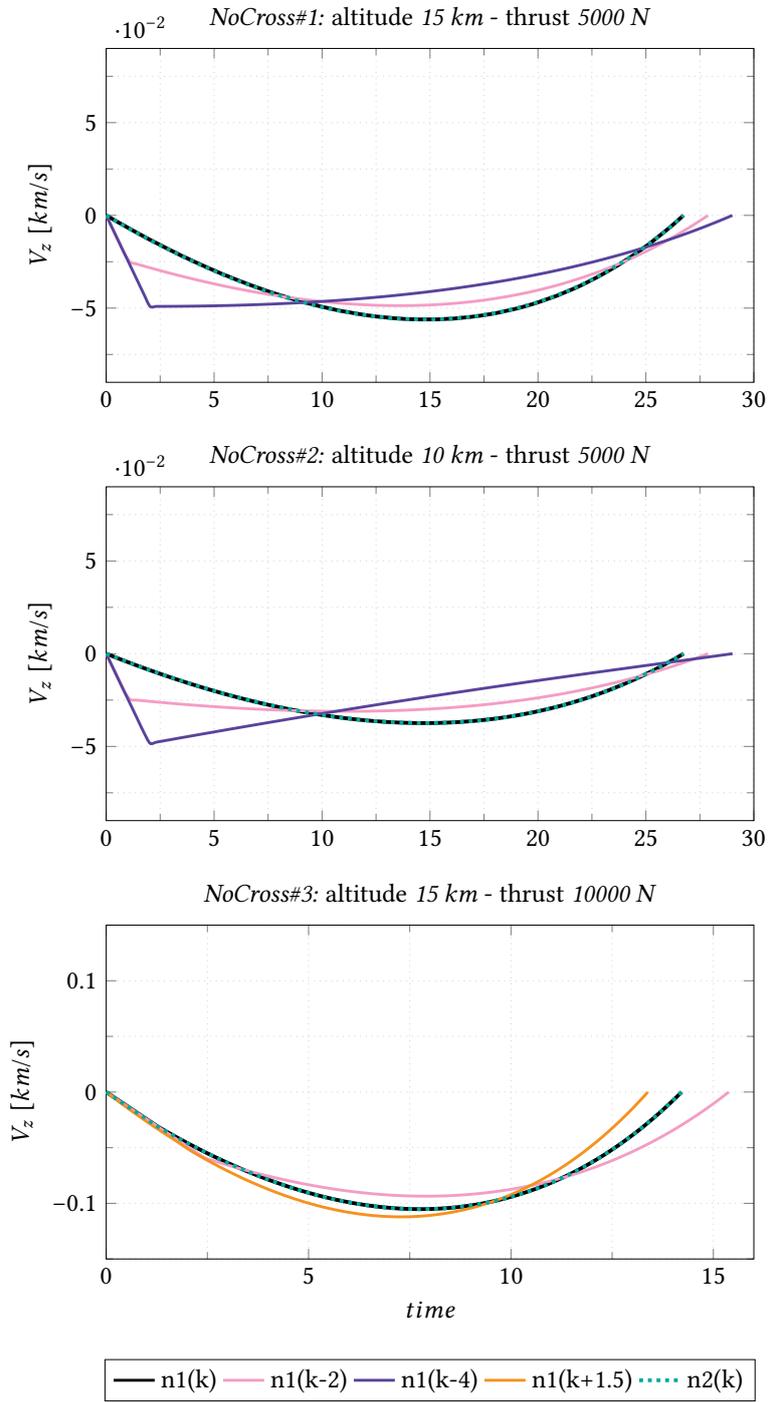


Fig. 6.1: V_z as a function of time - NoCross Comparison

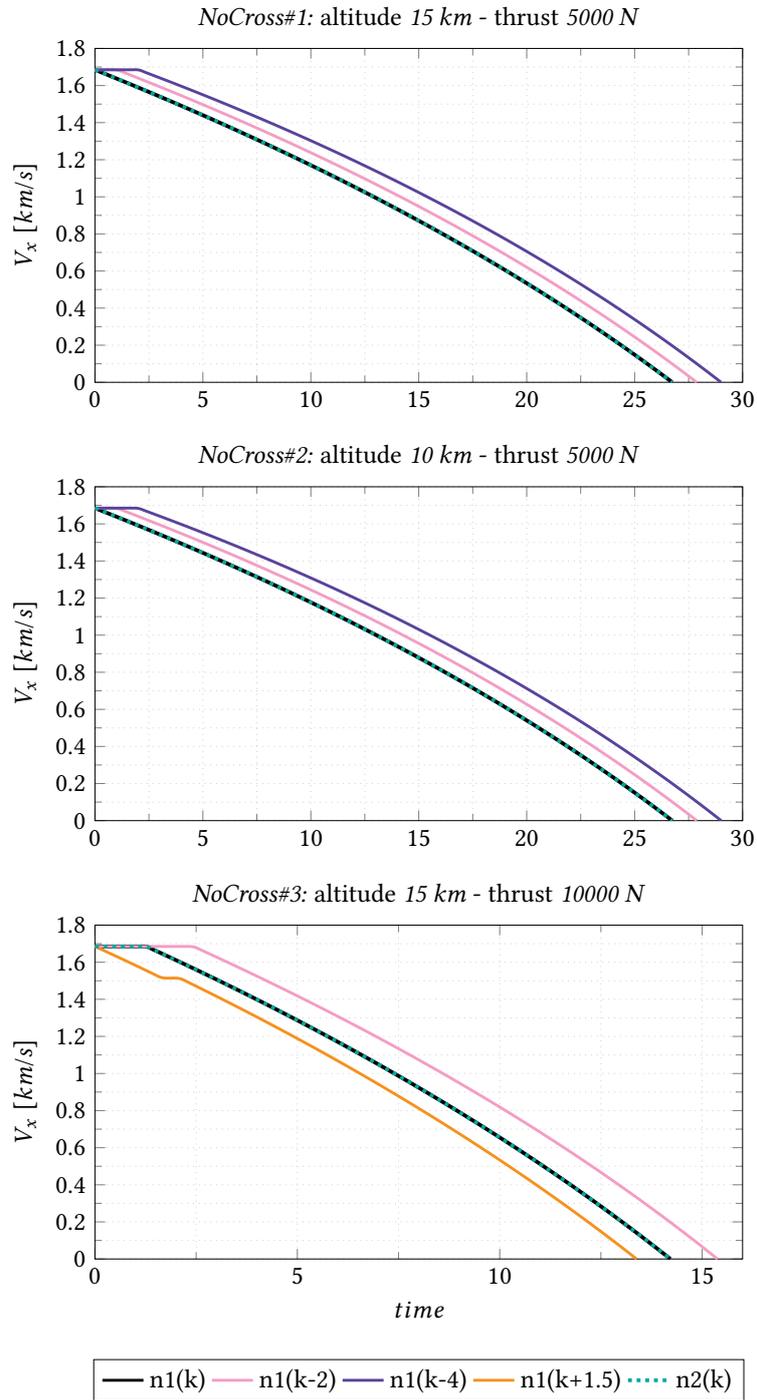


Fig. 6.2: V_x as a function of time - *NoCross* Comparison

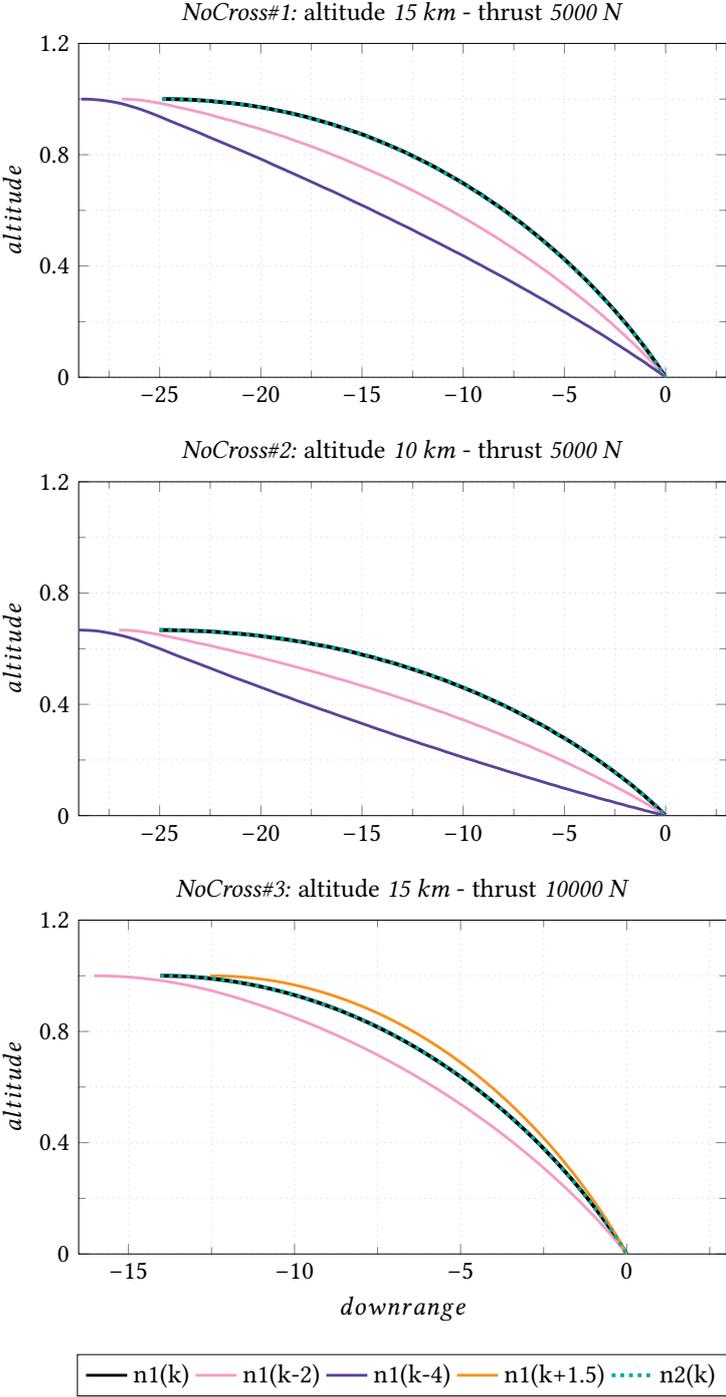


Fig. 6.3: R_z as a function of R_x (2-D) - NoCross Comparison

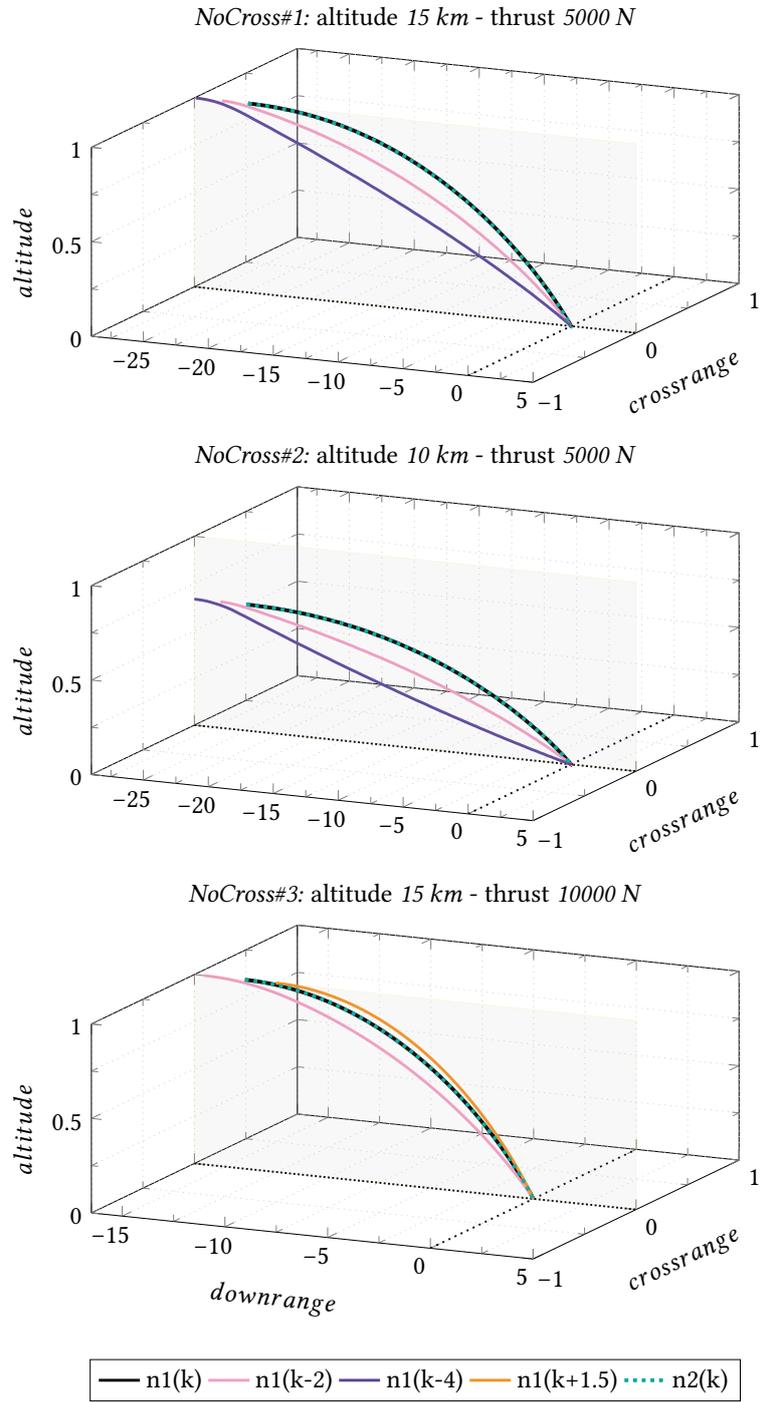


Fig. 6.4: R_z as a function of R_x (3-D) - NoCross Comparison

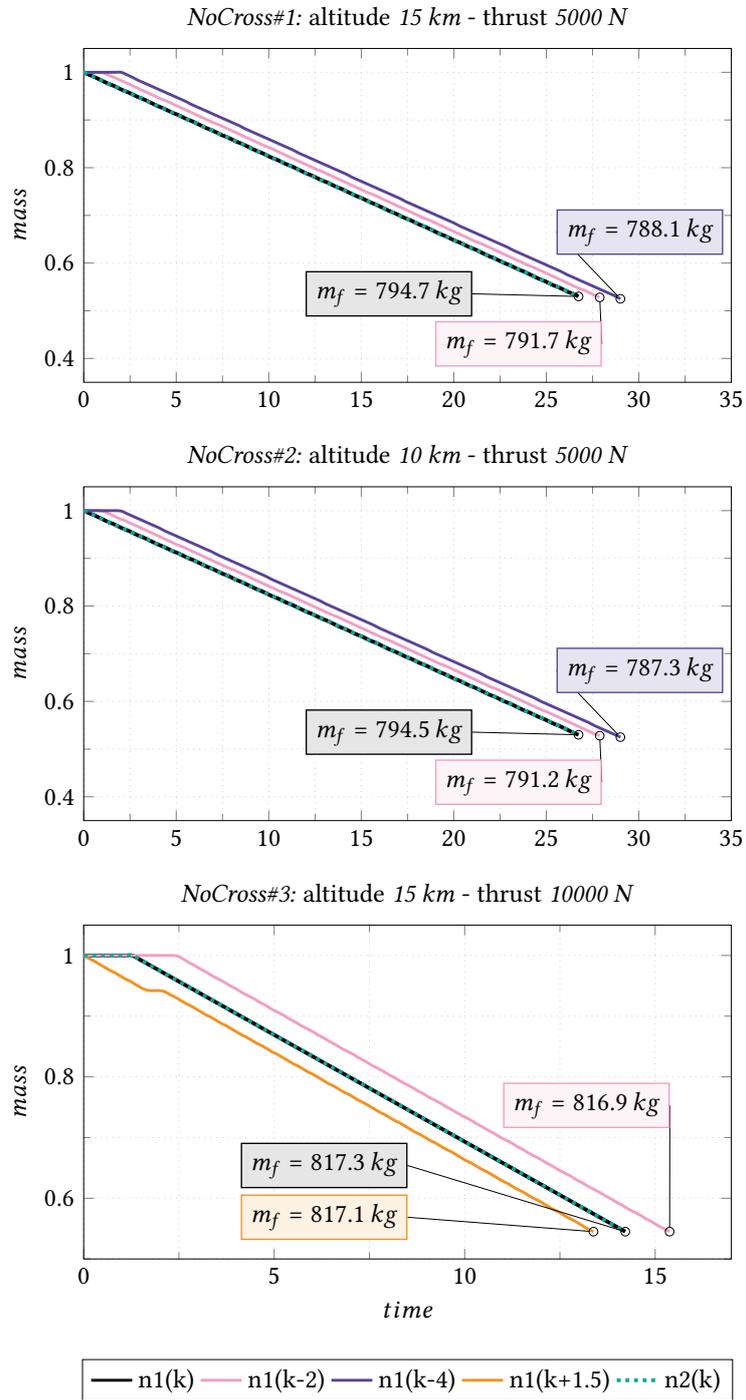


Fig. 6.5: Mass m as a function of time - NoCross Comparison

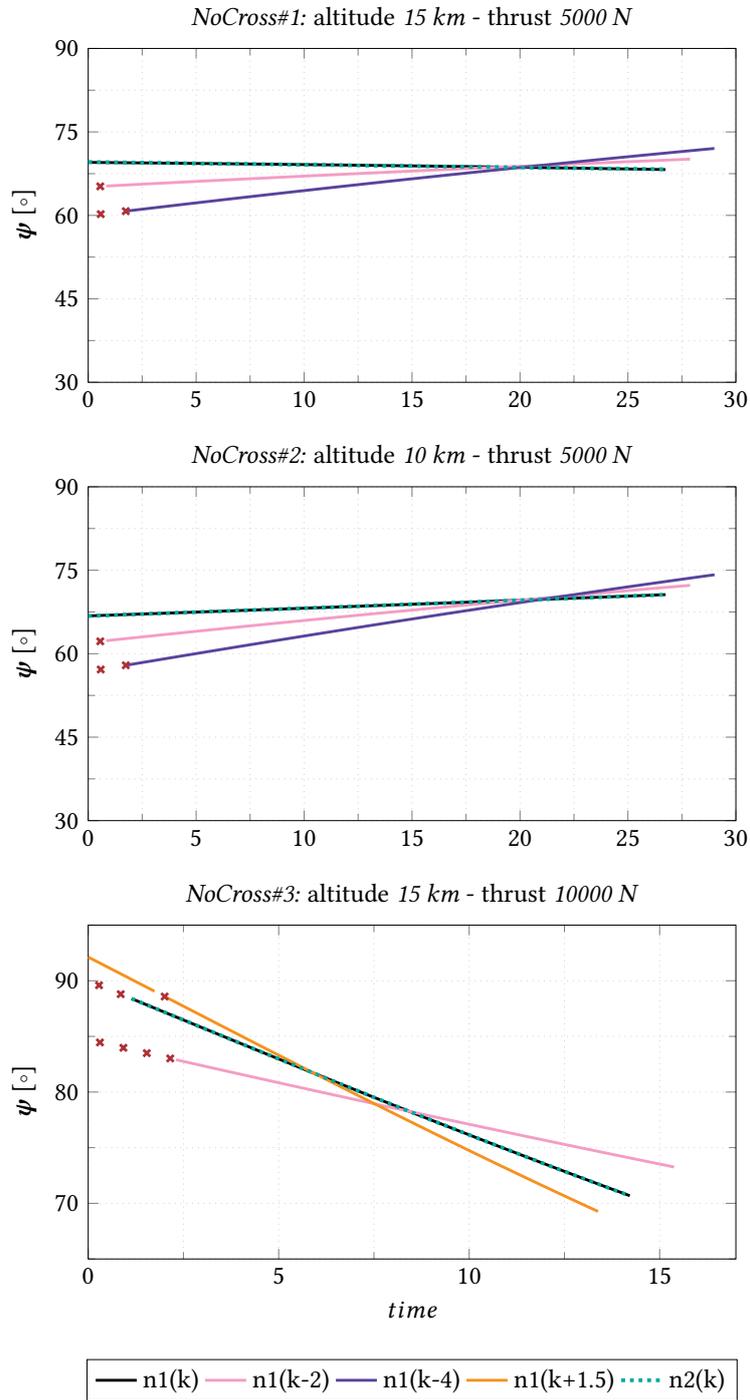


Fig. 6.6: Thrust angle ψ as a function of time - NoCross Comparison

(Fig. 6.1), when the negative peak of $k-2$, $k-4$ is reached and slope changes abruptly from negative to positive. Less immediate are k trial and the whole *NoCross#3* set, where the changing in slope sign is not indicative of the moment when SF switches, in the former case because there is no SF switch at all and ψ angle is kept approximately constant, in the latter case because time, thrust and angles management is radically different as already pointed out. Anyway, additional indicators to figure out when $SF > 0$, is always through $m(t)$ and $V_x(t)$, when curves simply start going downward. Talking about mass (Fig. 6.5), and considering #1 and #2 cases, differences recorded in $m(t)$ are minimal, #2 shows a bit higher propellant consumption, likely because linked to a superior k parameter and a lower altitude than #1 case, indeed net thrust power is the same and variables' profile quite similar among them. With regards to #3, the double net thrust available improves mass saving, being the braking more efficient, significantly reducing space and time. Nevertheless, a doubling of thrust power without any variation on vehicle weight fits into an ideal scenario, given that generally speaking, an increase in power corresponds to an increase in weight.

Last but not least, trajectory shapes have been drawn to highlight some path features and to provide a clear vision on how space vehicle stirs around space (Fig. 6.3, 6.4). It is possible to notice that, more negative is the value of k parameter, more the downrange starting point moves backward as expected. Moreover, landing point site $R_x = R_z = 0$ is guaranteed. Cases #1 and #2 have got same amount of thrust, but different altitude values, so #2 trajectory starts in a lower altitude point than #1, appearing more flattened. On the other hand, #3 test case follows a more arched trajectory with a higher radius of curvature, thanks to powerful engines and to a different management of thrust direction, preferring slowing down X-component of velocity first, and then its Z-component, while #1 and #2 cases follow a reverse tendency as already pointed out.

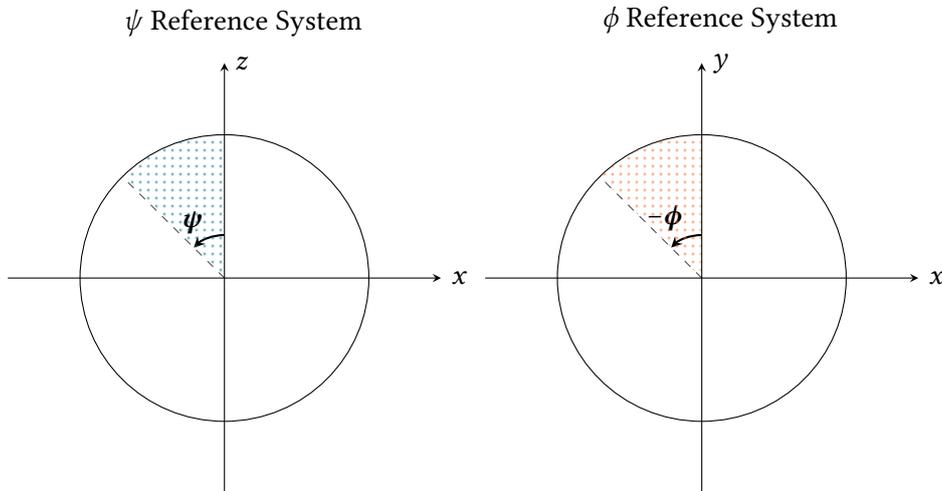


Fig. 6.7: Thrust angle ψ and ϕ and their respective R.S.

6.1.2 Crossrange Cases - Comparison

Basically, the same algorithm guidance principles and pattern illustrated in the previous section are still valid for the *Cross* cases. So where switching function changes sign, the same variations of *NoCross* are experienced. However, given that *Y*-axis is now an operative direction of motion, two new active variables are added to the plot set: $V_y(t)$ and $\phi(t)$, where the former represents the *Y*-component of velocity vector \mathbf{v} , considered positive when westbound with regard to the FRS (Fig. 4.2), while the latter represents the thrust-angle on *X*-*Y* plane, see Fig. 5.1 and Fig. 6.7. To begin with, the *Cross#1* case differs from the *NoCross#1* case only by one specific initial data input: a final crossrange value of $R_{y_f} = j = 1.5$ km was entered to simulate a diversion maneuver and see how the guidance algorithm reacts. Basically, all the trends of the variables that come into play remain almost identical to the *NoCross#1* case, in fact it is sufficient to manage the direction of the thrust in a proper way to arrive at the designated deviation point without affecting the propellant consumption, landing essentially with a mass equal to the case without crossrange.

It is also true that a deviation equal to $j = 1.5$ km is not excessively demanding, but it could certainly be decisive if a hazard is seen or an unforeseen situation arises. As can be seen from Fig. 6.9, the lateral velocities to which the vehicle is subjected are pondered and relatively low, indeed a parabolic trend is achieved, so no maneuver is carried out brusquely by the guidance algorithm. This characteristic can be further confirmed by the graph showing the $\phi(t)$ trend, where the variations of the thrust angle on the *X*-*Y* plane are contained in a narrow range: it starts from about -89° to reach almost -91° with a smooth slope. Looking at Fig. 6.15 we can therefore deduce that the values assumed by ϕ initially correspond to a first phase in which the deviation towards the *Y*-positive is searched and performed. Subsequently we gradually move towards slightly lower values, exceeding -90° and thus ensuring braking. In fact, lower values of -90° place thrust on the *X*-*Y* plane with opposite direction to V_{xy} , so that it counteracts velocity and ultimately cancels its modulus. With regard to *Cross#1*, crossrange values higher than $j = 1.5$ km have been tested, but without succeeding in converging due to a limited thrust power.

Therefore, it was decided to develop a second test case, i.e. *Cross#2*, where the dynamic model was provided with an available thrust equal to twice the nominal case, in a manner similar to what was done in the previous section with *NoCross#3*. It has been assumed in fact that, as it occurred for the *NoCross#3* cases, it is possible to ensure a higher trajectory control with a higher thrust, leaving initial mass unvaried, meanwhile keeping ensuring convergence. Once done, it was found that the guidance algorithm is capable of appropriately landing the spacecraft for crossrange values well above those guaranteed by case *Cross#1*. Moreover, as it happened for the *NoCross* cases, if the thrust is doubled, time and covered distances are halved, that is why also the k values must be updated with respect to case *Cross#1*: double thrust allows to start the braking phase closer to the landing site. To evaluate the effects on crossrange variations, the following tests have been considered: $j, j + 1.5, j + 3$, for which there is a slight alteration of the k parameter, i.e. the initial downrange

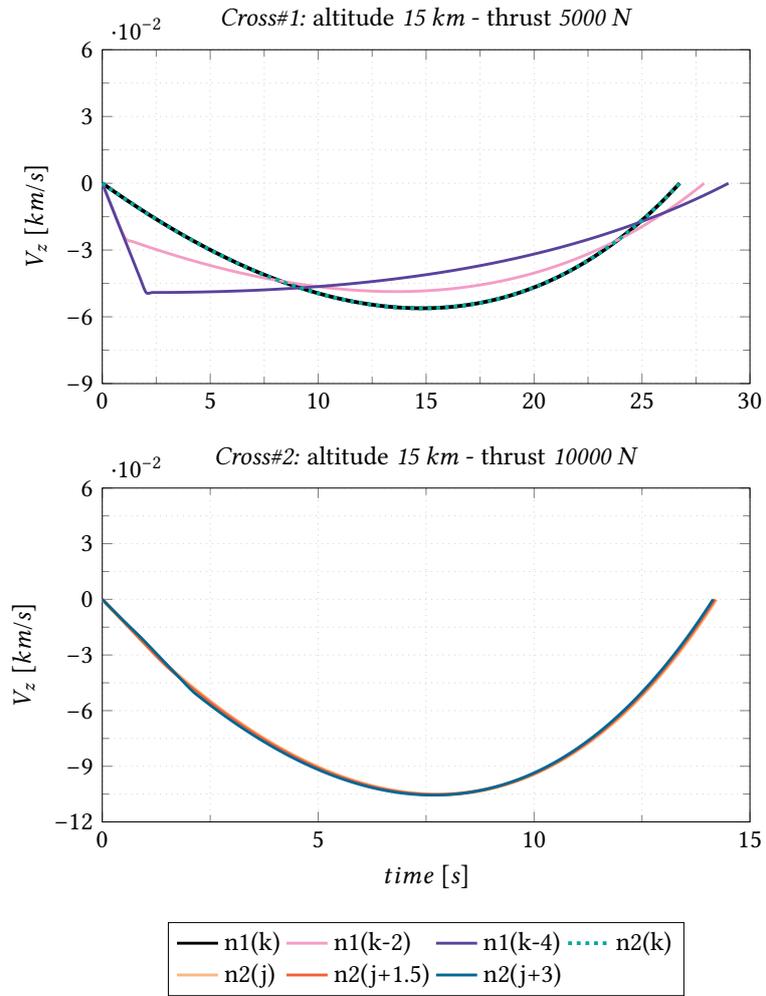


Fig. 6.8: V_z as a function of time - Cross Comparison

parameter, which can be considered almost constant, but we will explain later why these small variations occur.

Starting from the graph $V_z(t)$ - *Cross#2* - (Fig. 6.8), the rate of descent with respect to the *NoCross#3* case has remained unchanged, the descent is carried out with around the same speed, and even for higher values of crossrange, there is no particular influence on $V_z(t)$. However, if we start considering $V_y(t)$, variations caused by j parameter are much more significant. When referring to (Fig. 6.9), it is possible to notice that lower crossrange values correspond to lower speeds, and vice versa: likewise, increasing j parameter leads to higher peaks. On the other hand, when dealing with $V_x(t)$ (Fig. 6.10), apart from initial variations due to the effect of the switching function, the values become equal and merge from a certain point onwards. Moreover, V_y and V_x , lying on the X - Y plane, are affected by the variations caused by the switching function without any type of conditioning, while V_z is not always representative of SF variations, due to the action of gravity acceleration, which still influences the trend, even when $SF < 0$. Hence, for both $V_x(t)$ and $V_y(t)$, when engines are off ($SF < 0$) the speed is kept constant and a straight line is displayed in the graphs. Control profiles also change, according to the amount of deviation: for j test case a T_{min} - T_{max} profile is recorded, while for $j + 1.5$ and $j + 3$ a T_{max} - T_{min} - T_{max} control profile regulates trajectory. Control profiles can be further verified through $m(t)$, $\psi(t)$ and $\phi(t)$ plots.

As mentioned, variations on mass are quite indicative of how guidance algorithm is managing dynamics, and as might be expected, the less extensive the crossrange deviation is, the higher the final mass at touchdown will be. However, the values do not differ much between $j, \dots, j + 3$. On the other hand, the differences in *Cross#2* are more pronounced than in case *Cross#1* (Fig. 6.13), and we have found the same reason in the *NoCross* cases: more net thrust, in addition to shortening times and spaces, provides more effective braking, ensuring higher propellant savings. As for the trajectory, the higher the crossrange value j , the more the trajectory is reduced, both in duration and in downrange: this is a direct consequence of the fact that, assuming everything else being kept unchanged, if thrust and mass are limited and limiting elements, the energy balance dictates that the higher the gain is in crossrange, the less it can be in downrange. This is why the trajectory, as the crossrange value increases, curves shift slightly forward as can be seen from the enlargement in Fig. 6.11. If we did not have the end point fixed (R_{x_f} free), and the curves all started from the same position (k fixed), then the curves would move backwards, respecting the same principle.

Finally, let's consider the thrust angles of case *Cross#2* (having already said that case *Cross#1* trends are pretty similar to the *NoCross* #1/#2 cases): the same basic principles as for case *Cross#1* apply. As far as ψ angle in *Cross#2* is considered, i.e. the direction of the thrust in the X - Z plane, the trend is similar to what we have already seen for the *NoCross* cases. The high available thrust and a reduced downrange space drive the guidance algorithm to initially execute and concentrate most of the braking on the X -component of the speed, which is the most relevant component of velocity

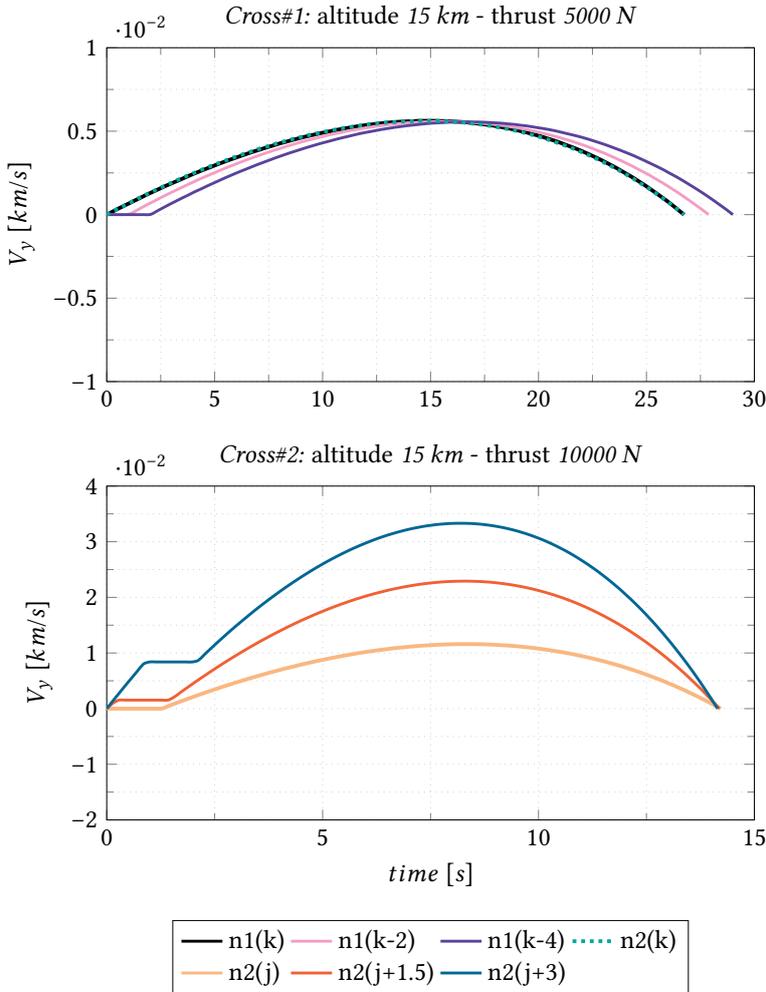


Fig. 6.9: V_y as a function of time - Cross Comparison

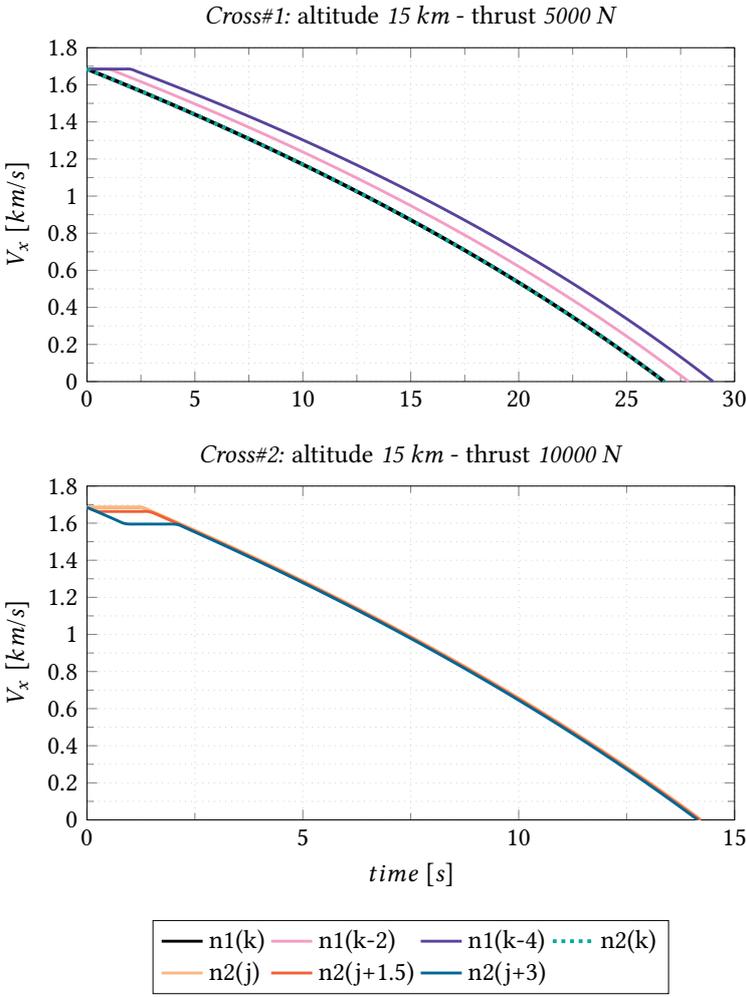


Fig. 6.10: V_x as a function of time - Cross Comparison

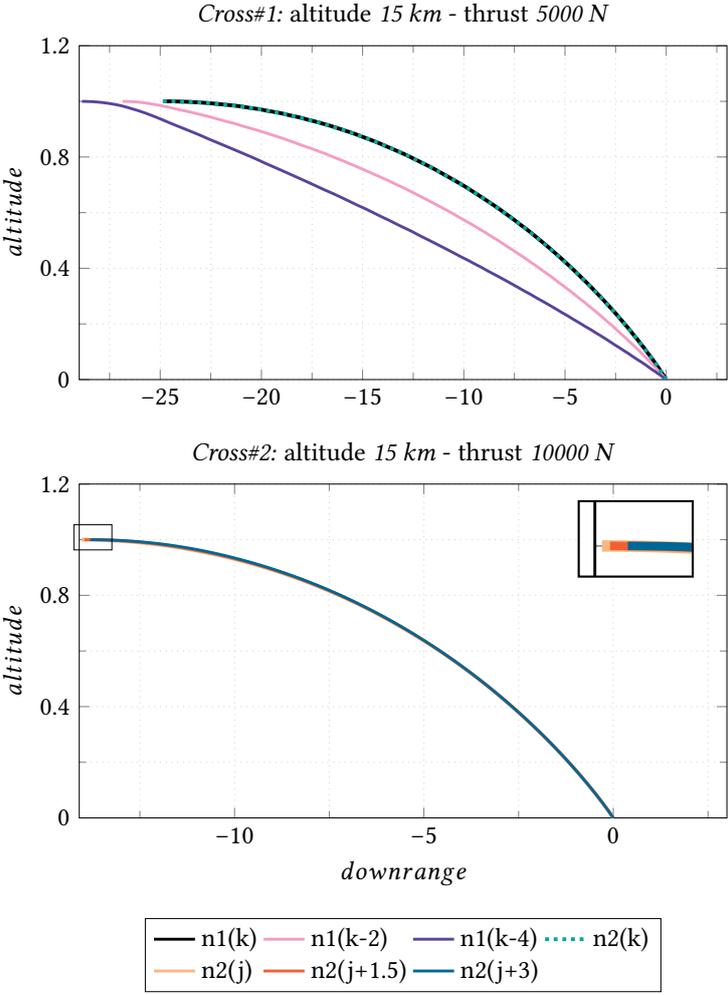


Fig. 6.11: R_z as a function of R_x (2-D) - Cross Comparison

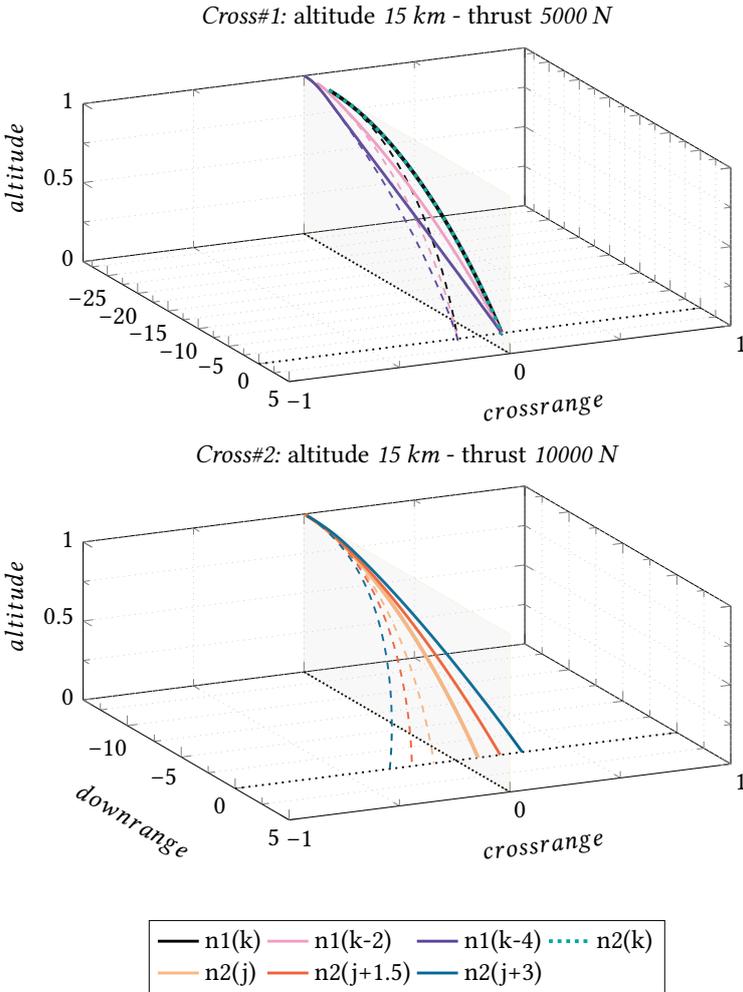


Fig. 6.12: R_z as a function of R_x (3-D) - Cross Comparison

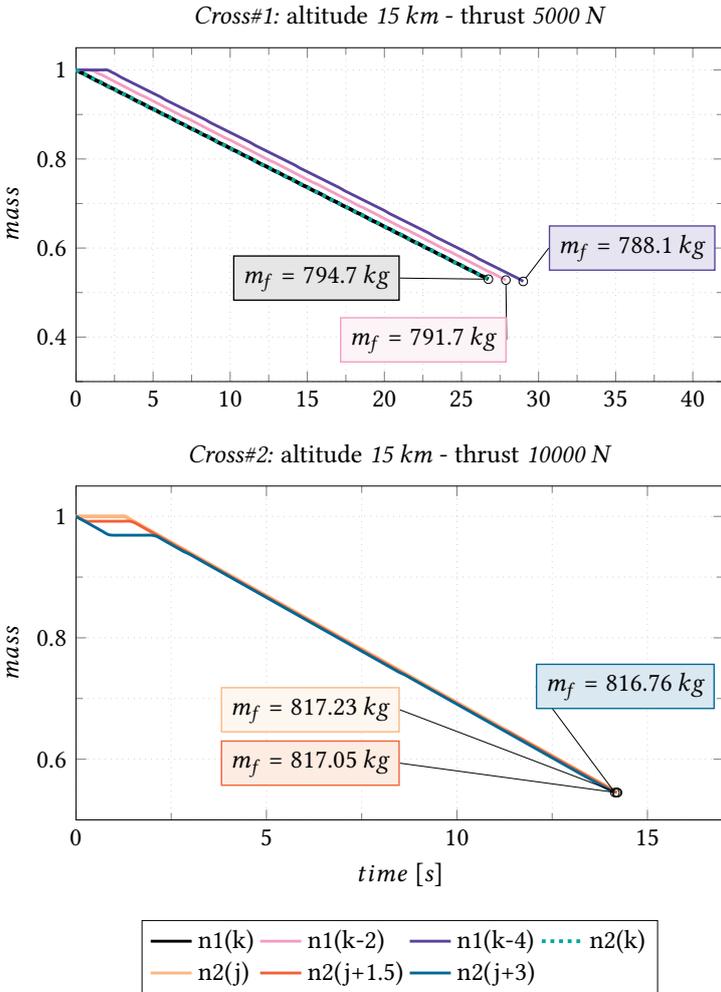


Fig. 6.13: Mass m as a function of time - Cross Comparison

at the starting point for every single test case³. As proof of fact, ψ values are around 90° at the very beginning, and then reach about 70° : it means that an increasing portion of thrust is shifted along X -axis toward Z -axis to ensure a proper braking as we approach touchdown. Noteworthy (see magnification frame in Fig. 6.14) is the perfect correspondence of the times when the motors are switched on or off with the previously mentioned control profiles. The slope of the $\psi(t)$ curve is constant and roughly the same if compared to the three cases $j, j + 1.5, j + 3$, so only smooth changes of thrust direction occur along the X - Z plane.

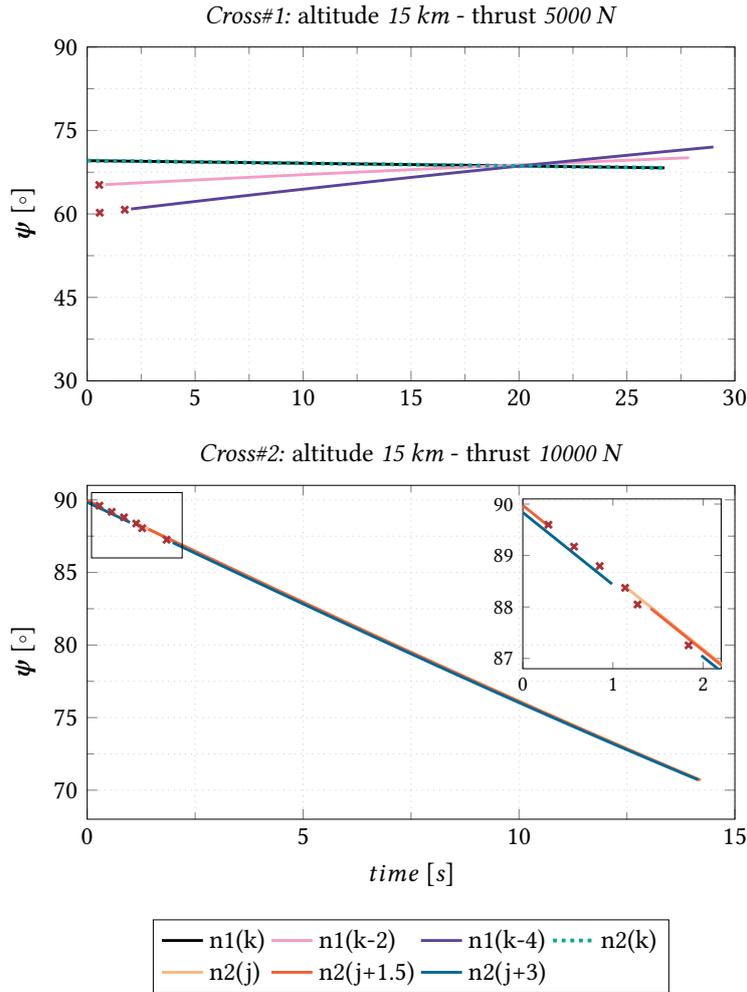


Fig. 6.14: Thrust angle ψ as a function of time - Cross Comparison

Similarly, thrust angle ϕ in *Cross#2*, which lies on the X - Y plane, shows an analogous trend to *Cross#1* regardless of the variation operated on j and thrust: i.e. it

³Note that V_x is the only velocity component generally influenced by both ψ and ϕ .

starts from a certain value and goes down towards more and more negative values, even if unlike Cross#1, Cross#2 is characterized by less gradual and more intense maneuvers on X-Y, motivated by the greater slope of the curves, in turn influenced by greater crossrange values. Taking $j + 3$ into consideration, it records the greatest excursion, starting from -84° and arriving at almost -94° . The greater the excursion, the greater the deviation commanded along Y. Moreover, it is possible to notice a certain symmetry of ϕ values with respect to the value of -90° . This means that the crossrange deviation towards Y generally starts at the beginning of the trajectory with $\phi > -90^\circ$, and gradually moves towards $\phi < -90^\circ$ in order to place thrust in the opposite direction of V_{xy} , with the goal of slowing V_y down and bringing it to zero. Through $\phi(t)$ plot, as well as $\psi(t)$, it is possible to visualize the moments of the trajectory in which the engines switch on or off ($SF \geq 0$), making it possible to deduce the control profile of each scenario.

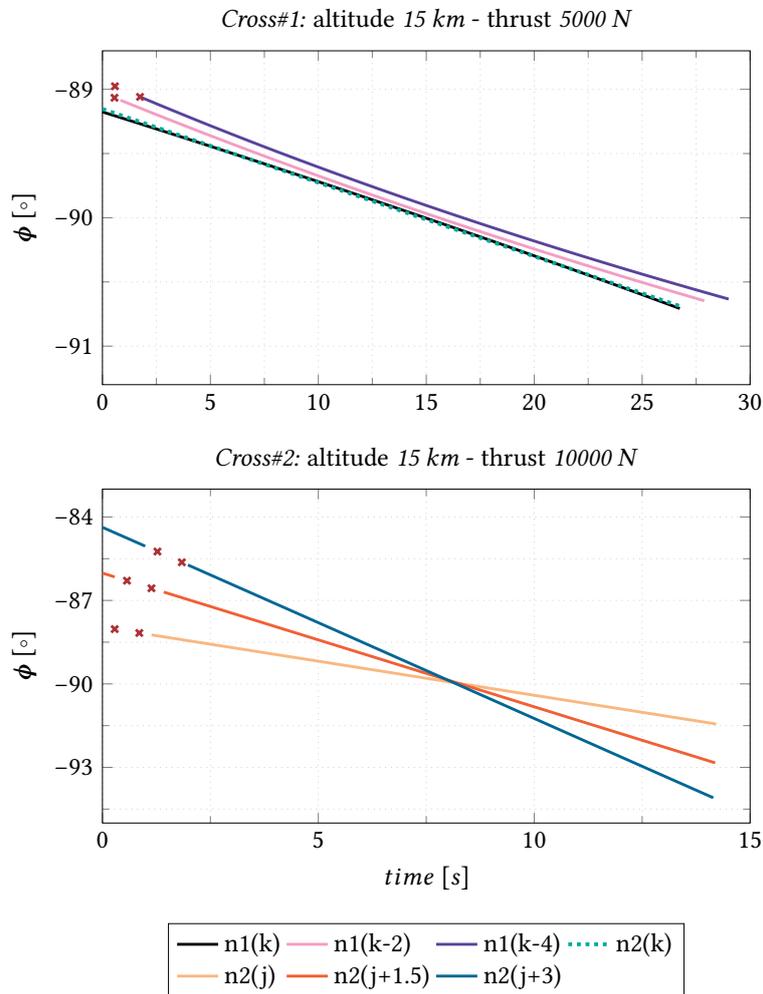


Fig. 6.15: Thrust angle ϕ as a function of time - Cross Comparison

Table 6.5: Final Mass - Comparison

Final Mass [kg]					
Downrange [km]	NoCross#1	NoCross#2	NoCross#3	Cross#1	Cross#2
n2: $x_f - x_0$	794.7	794.5	817.3	794.7	817.23 817.05 816.76
n1: $x_f - x_0$	794.7	794.5	817.3	794.7	/
n1: $x_f - x_0 + 30$	791.7	791.2	816.9	791.7	/
n1: $x_f - x_0 + 60$	788.1	787.3	/	788.1	/
n1: $x_f - x_0 - 22.5$	/	/	817.1	/	/

Note: Cross#2 top-down values refer to j, j+1.5, j+3 crossrange diversion [km] respectively.

Table 6.6: Propellant Consumption - Comparison

Propellant Consumption [kg]					
Downrange [km]	NoCross#1	NoCross#2	NoCross#3	Cross#1	Cross#2
n2: $x_f - x_0$	705.3	705.5	682.7	705.3	682.77 682.95 683.24
n1: $x_f - x_0$	705.3	705.5	682.7	705.3	/
n1: $x_f - x_0 + 30$	708.3	708.8	683.1	708.3	/
n1: $x_f - x_0 + 60$	711.9	712.7	/	711.9	/
n1: $x_f - x_0 - 22.5$	/	/	682.9	/	/

Note: Cross#2 top-down values refer to j, j+1.5, j+3 crossrange diversion [km] respectively.

7

Conclusions

The Moon has awakened the interest of the scientific community in recent times, both for latest discoveries on ice reserves present in its poles, and for its great potential as a stepping stone for a spaceport capable of supporting future missions of interplanetary nature. In this context, the retrofired descent trajectory on the lunar soil thus represents a significant technological challenge, due to the complexity of physics behind it and the high safety standards required by the maneuver.

In addition, if fuel saving is fundamental for the maximization of the on-board payload, optimization techniques play a primary role, always in the respect of mission constraints and requirements. After illustrating the preliminary assumptions and presenting the resulting approximate mathematical model, an indirect optimization method was applied, yielding to a boundary value problem, finally solved through shooting techniques. Measurements on the impact of mass, thrust and altitude variations were conducted to understand the response of the optimal guidance algorithm, once defined the landing target. Results showed the high incidence of thrust power on control profiles, revealing for higher thrust levels longer coasting phases. The latter are often useful and necessary for the identification of a safe landing site during the approach phase, even though variations on path times may affect mass expenditure. Cross-range diversion maneuvers were also performed to highlight how the guidance system reacts in the event that the autonomous and integrated GN&C system detects hazards, commanding a change of direction.

Again, thrust levels bound and regulate times and distances that space vehicles can cover. By supplying twice the thrust level of the nominal case under study, further targets can be reached. Mass consumption ranges around acceptable values, since the increase in mass flow consumed and exhausted is balanced by a shortening of the descent times and a higher braking efficiency (the more thrust available spacecraft

has at its disposal, the longer the braking phase can be delayed).

Although the physical model used in this work is not particularly accurate, due to a couple of simplification, this study is still a solid starting point for future developments, since the method used is particularly suitable for integrating more sophisticated features and constraints: it is possible to imagine the implementation of 6 *DOF* (Degrees of Freedom), providing spacecraft attitude controls by introducing rotations around spacecraft's center of gravity (roll, pitch and yaw). In this way it would be feasible to constrain the vehicle orientation to certain points or arcs of the trajectory, in order to correctly manage the glideslope visibility, taking into account possible pilot windows or tracking cameras. Other additions on the guidance algorithm could include: constraints on deceleration levels for possible manned missions where astronauts welfare is relevant, and variations on orbital parameters.

Bibliography

- [1] A Brief History of Moon Exploration. URL: <https://www.nationalgeographic.com/science/article/moon-exploration>.
- [2] Apollo 11 Flight Plan. 1969. URL: https://www.hq.nasa.gov/alsj/a11/a11ftpln_final_reformat.pdf.
- [3] *Apollo Lunar Landing Mission Symposium*. Manned Spacecraft Center, Houston TX, 1966. NASA, National Aeronautics and Space Administration. URL: https://www.hq.nasa.gov/alsj/LunarLandingMissionSymposium1966_1978075303.pdf.
- [4] R.R. Bate, D.D. Mueller, and J.E. White. *Fundamentals of Astrodynamics*. Dover Books on Aeronautical Engineering Series. Dover Publications, 1971.
- [5] John T. Betts. Survey of Numerical Methods for Trajectory Optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193–207, 1998.
- [6] William E. Boyce and Richard C. DiPrima. *Elementary Differential Equations and Boundary Value Problems*. Wiley, Hoboken, NJ, 2017.
- [7] James D. Burke. Chapter 56 - Planetary Exploration Missions. In *Encyclopedia of the Solar System (Third Edition)*, pages 1205–1222. Elsevier, Boston, third edition edition, 2014.
- [8] Guido Colasurdo and Lorenzo Casalino. Indirect Methods for the Optimization of Spacecraft Trajectories. In *Modeling and Optimization in Space Engineering*, pages 141–158. Springer, New York, 2013.
- [9] Guido Colasurdo and Lorenzo Casalino. Tentative Solutions for Indirect Optimization of Spacecraft Trajectories. In *Space Engineering: Modeling and Optimization with Case Studies*, pages 87–102. Springer, Cham, 2016.
- [10] Bruce A. Conway. *Spacecraft Trajectory Optimization*. Cambridge Aerospace Series. Cambridge University Press, 2010.
- [11] Ian A. Crawford, Katherine H. Joy, and Mahesh Anand. Chapter 25 - Lunar Exploration. In *Encyclopedia of the Solar System (Third Edition)*, pages 555–579. Elsevier, Boston, third edition edition, 2014.
- [12] Howard D. Curtis. *Orbital Mechanics for Engineering Students*. Butterworth-Heinemann, Amsterdam, 2020.

- [13] Cho Dong-Hyun, Kim Donghoon, and Leeghim Henzeh. Optimal Lunar Landing Trajectory Design for Hybrid Engine. *Mathematical Problems in Engineering*, 2015:1–8, 2015.
- [14] ESA website - Exploration of the Moon. URL: <https://exploration.esa.int/web/moon/>.
- [15] *Lunar Landing Systems Study*, 1997. ESA, European Space Agency. URL: https://nebula.esa.int/sites/default/files/neb_study/223/C9558ExS.pdf.
- [16] Ingo Gerth and Erwin Mooij. Guidance for Autonomous Precision Landing on Atmosphereless Bodies. In *AIAA Guidance, Navigation, and Control Conference*, 2014.
- [17] J. Guo and C. Han. Design of Guidance Laws for Lunar Pinpoint Soft Landing. In *TU Delft Institutional Repository*, 2009.
- [18] David Harland. *Exploring the Moon: The Apollo Expeditions*. Springer Praxis Pub, New York, London, Chichester, UK, 2008.
- [19] Gerald R. Hintz. *Orbital Mechanics and Astrodynamics: Techniques and Tools for Space Missions*. Springer, Cham, 2015.
- [20] Chu Huiping, Ma Lin, Wang Kexin, Shao Zhijiang, and Song Zhengyu. Trajectory Optimization for Lunar Soft Landing with Complex Constraints. *Advances in Space Research*, 60(9):2060–2076, 2017.
- [21] Dario Izzo, Nicolás Weiss, and Tobias Seidl. Constant-Optic-Flow Lunar Landing: Optimality and Guidance. *Journal of Guidance, Control, and Dynamics*, 34(5):1383–1395, 2011.
- [22] D.E. Kirk. *Optimal Control Theory: An Introduction*. Dover Books on Electrical Engineering Series. Dover Publications, 2004.
- [23] D.F. Lawden. *Optimal Trajectories for Space Navigation*. Butterworths, London, 1963.
- [24] Shuang Li and Zhang Liu. Autonomous Navigation and Guidance Scheme for Precise and Safe Planetary Landing. *Aircraft Engineering and Aerospace Technology*, 81:516–521, October 2009.
- [25] Daniel Liberzon. *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton University Press, 2011.
- [26] Jack Macki and Aaron Strauss. *Introduction to Optimal Control Theory*. Springer, 1982.
- [27] Joaquim R.A. Martins and Andrew Ning. *Engineering Design Optimization*. Cambridge University Press, 2021.
- [28] NASA - Jet Propulsion Laboratory website. URL: <https://www.jpl.nasa.gov>.
- [29] Richard W. Orloff and David Harland. *Apollo: The Definitive Sourcebook*. Springer, New York, 2006.

- [30] Anil V. Rao. A Survey of Numerical Methods for Optimal Control, 2009. Preprint Version.
- [31] Jeremy Ryan Rea. *An Investigation of Fuel Optimal Terminal Descent*. PhD thesis, University of Texas at Austin, 2009.
- [32] Lawrence F. Shampine and Marilyn K. Gordon. *Computer Solution of Ordinary Differential Equations: The Initial Value Problem*. W.H. Freeman, San Francisco, 1975.
- [33] *Powered Descent Trajectory Guidance and Some Considerations for Human Lunar Landing*. Breckenridge, Colorado, 2007. 30th Annual AAS Guidance and Control Conference, American Astronautical Society - Rocky Mountain Section. URL: <https://ntrs.nasa.gov/citations/20070004584>.
- [34] Jason L. Speyer and David H. Jacobson. *Primer on Optimal Control Theory*. Society for Industrial and Applied Mathematics, Philadelphia, 2010.
- [35] Paul D. Spudis. *The Value of the Moon: How to Explore, Live, and Prosper in Space Using the Moon's Resources*. Smithsonian Books, Washington, DC, 2016.
- [36] E. Trélat. Optimal Control and Applications to Aerospace: Some Results and Challenges. *Journal of Optimization Theory and Applications*, 154:713–758, 2012.
- [37] David Vallado. *Fundamentals of Astrodynamics and Applications*. Microcosm Press & Springer, Hawthorne, CA, 2013.