Politecnico di Milano
Aerospace Science and Technology Department
Doctoral Programme In Aerospace Engineering

Hazard Detection and Avoidance Systems for Autonomous Planetary Landing

Doctoral Dissertation of:
Paolo Lunghi

Supervisor:
Prof. Michèle Lavagna

Tutor:
Prof. Lorenzo Dozio

The Chair of the Doctoral Program:
Prof. Luigi Vigevano

Year 2017 – Cycle XXIX

"Gravity," said Dirk with a slightly dismissed shrug, "yes, there was that as well, I suppose. Though that, of course, was merely a discovery. It was there to be discovered."... "You see?" he said dropping his cigarette butt, "They even keep it on at weekends. Someone was bound to notice sooner or later. But the catflap ... ah, there is a very different matter. Invention, pure creative invention. It is a door within a door, you see."

DOUGLAS ADAMS, Dirk Gently's Holistic Detective Agency

Ringraziamenti

Crema, 3 Gennaio 2017

E con questo si è arrivati alla fine. La fine di una tappa, intendiamoci. Parafrasando con un po' di libertà Robert Jordan, la ruota non smette mai di girare: si tratta però di mettercela sempre tutta per fare un po' meglio ad ogni giro. Quest'ultimo giro è stato un po' faticoso, ma pieno di soddisfazioni.

Non posso che cominciare con un grande grazie alla prof. Lavagna. Il giorno della mia laurea non sospettavo minimamente che di lì in capo a trenta giorni avrei inviato una proposta per un dottorato. Mi ha messo lei la pulce nell'orecchio: e nei più di tre anni successivi non mi ha mai fatto mancare la sua fiducia.

A costo di prenderla un po' alla lunga, tutto è cominciato più di trent'anni fa, quando due persone in particolare hanno iniziato a leggermi fumetti e libri sul divano, innescando quella curiosità di capire come funziona il mondo intorno a noi che mi ha portato fino a qui: grazie Papà, grazie Mamma. Avete cominciato tutto voi.

E grazie Sorelle che gli avete sempre dato corda: da quando Mari mi insegnava l'alfabeto, a quando Claudia mi raccontava l'Odissea, a quando prendevo in prestito il libro di Scienze della Terra di Daniela. In quanto "sorelle grandi" mi avete sempre aperto tante strade: e quando sono diventato un po' più grande anche io, mi avete sempre sostenuto, non risparmiandovi di prendermi in disparte e avvertirmi di quando sbagliavo (i momenti più importanti: facile sostenere quando tutto va bene!).

Grazie Cognati che sostenete le sorelle: a Simone (ci sono un po' di uscite in montagna arretrate da recuperare), a Alberto che mi tiene informato sulle migliori cantine in prossima visita, a Angelo sempre disponibile nonostante la distanza.

Per non parlare degli straordinari Nipoti: Michele, Milena, Andrea, Giulia, Arianna, Marta, Edoardo, Davide, Alice, se dovessi dire qualcosa per ognuno di voi, dovrei scrivere una seconda tesi, ma mi raccomando, non fate riposare troppo quel vostro zio che ha in testa solo lo spazio, fategli domande su domande che altrimenti si

adagia troppo! E quando ne saprete più di lui sarà il vostro turno di soddisfare un bel po' di curiosità da parte sua.

Grazie a tutti coloro con i quali ho trascorso il dottorato, il valore aggiunto di questi anni. Grazie a Federica, la mia "vicina di banco" e di risate, e a Filippo: avete iniziato ad ospitarmi quando ero un pendolare lontano e non avevo più treni, siete finiti per essere vicini di casa e amici veri. A proposito, non ho potuto farci tutta la tesi, ma come vedete non ho resistito: il Courier dovevo usarlo da qualche parte... Grazie a Mirco, se non ci fossi stato sarei ancora a cercare di raccapezzarmi sull'utilizzo di certi moduli di un certo innominabile software, detieni una quota non indifferente della mia gratitudine, nonché un credito di birre davvero notevole. Grazie a Fabio, con cui ho iniziato, andando a caccia di scrivanie libere in dipartimento, e con cui ora sto finendo il dottorato, ora si inizia a fare sul serio; grazie a Michele, Barbara, Andrea e Francesca, indimenticabili compagni di ufficio e protagonisti di altrettante indimenticabili e surreali discussioni in pausa pranzo, che mi mancano davvero; ad Alessandro, che ci ha reso tutti un po' più edotti sui modi di cucinare il porco; a Jakub, dirimpettaio di scrivania che non ho mai visto entrare in ufficio senza un cubitale sorriso; al Parri, che si vede di rado, ma che quando compare dice sempre cose sagaci (perfino le battute brutte!); a Louis, che sembra sempre tanto tranquillo, ma che poi quando meno te lo aspetti tira fuori la battuta fulminante.

Grazie a Roberto, che fin dalla mia "nascita accademica" mi dà sempre idee e consigli importanti, e a Pierluigi, con cui ho iniziato a collaborare da relativamente poco ma che mi da consigli non meno preziosi: abbiamo ancora tanti lavori in sospeso!

Grazie ai miei amici di sempre, Ste, Toga, Betto: stiamo prendendo ognuno la sua strada, ogni tanto si scompare per un bel po', ma siamo sempre tutti lì quando serve.

Grazie Bonni, paziente come pochi quando nei periodi di maggior pressione il tuo coinquilino sembrava sparire di colpo per giorni, lasciando la casa in abbandono.

Grazie alla mia famiglia surrogata in California: prima di tutto i Palladini della ...., Ale, vicino di banco senza il quale al mio arrivo non avrei saputo da che parte girarmi, Maggia e le discussioni sulle assurdità della vita americana, Biasio (questa volta la i te la meriti), Enrico si, anche tu (:P), Minno, il padrone di casa delle migliori grigliate e conoscitore di tutti i migliori posti del Sud della California, ma anche di Milano; e poi a stretto giro Elisa, che con Ale ha garantito la mia sopravvivenza, Thalia, Valentina, Ellen, Francesca, George: i mesi che sono stato tra di voi mi avete fatto sentire come a casa, 9714 km e non sentirli: state attenti che ritorno (si, è una minaccia!). Non potevo capitare meglio.

Grazie ai miei amici reali dei mondi immaginari: quelli dei mondi immaginari dal vivo, i Sigilli, gli Aurei e gli Arcani: Elisa,

Canciano, Vale, Lazza, Davide, Luca, Denise, Tesla; e quelli dei mondi immaginari di carta: Stefano (e fatta la tesi, presto arriverà anche il nuovo PG!), Renato, Fabio, Federico, Chris, Alessandro. Per non parlare di Riccardo, trasversale a un po' tutti questi mondi, reali e immaginari, fin da quando si è cominciato con l'ingegneria. Ogni volta che sono venuto con voi su Numea, o che sono stato trascinato con voi per il mondo come una banderuola dall'ARCAM Corporation, mi avete permesso di staccare completamente e ricaricare le pile: mi avete evitato un numero di esaurimenti nervosi esprimibile solo in numeri di Graham.

La cosa più importante che ho imparato in questi anni forse è questa: che le cose difficili non si fanno mai veramente da soli. Non so se ho banalmente riscoperto la gravità l'ennesima volta, o se ho finalmente inventato la porticina del gatto, e se qui dentro c'è qualcosa di brutto o sbagliato è farina del mio sacco, ma per il resto... gran bel lavoro di gruppo, gente.

Grazie.

Paolo

Abstract

PRECISE and autonomous landing capability is a key feature for the next space systems generation. The possibility to carry out Hazard Detection and Avoidance would allow both absolute and relative correction maneuvers, dramatically increasing the robustness and the flexibility of the mission. A novel guidance algorithm is presented: the trajectory is modeled as a polynomial of minimum degree required to satisfy the boundary constraints, leaving a reduced set of parameter free to be optimized. The novelty of the proposed method lies in the fact that it allows the computation of large diversions, with a suboptimization of the fuel consumption, satisfying at the same time all the constraints imposed by the system in terms of control torques, thrust, and allowed hovering area. Only 2 or 3 optimization variables are needed, making the algorithm light enough to run on-board. The flexibility of the guidance is addressed with two different applications, a lunar landing and the close approach to an asteroid. An ad hoc optimizer is also developed, based on Differential Algebra, capable to solve the guidance optimization in a fast and reliable way. Objective and constraints are modeled as low order Taylor maps. The general features of the functions are easily got, leading in a few iterations to the optimal solution, due to the property of the Taylor series to converge to the true value in proximity to the expansion point. An innovative hazard detection and target selection algorithm is also proposed. The capability of Artificial Neural Networks (ANNs) to extrapolate underlying rules in complex datasets is exploited to obtain an automatic classifier that builds a hazard map of the landing area, basing on a single image. Manual establishment of heuristic correlations between image and terrain features is no longer required, leaving to the ANN training the task to identify these correlations automatically; the process is run off-line, while only the trained network runs on-board, with a minimal computational burden. A target selection algorithm exploits the map to locate and rank the candidate landing sites following safety and reachability criteria. A coherent and effective dataset for rigorous training and test is generated with a realistic simulation tool. The network showed the ability to select a safe landing site in 100 % of cases.

Contents

List of Figures V

List of Tables IX

Glossary XI

1 Introduction 1
1.1 Autonomous Landing GNC Chain 5
1.2 Previous Works 7
1.3 Dissertation Overview and Main Contributions to the Field 8
1.4 Bibliographic Disclaimer 10

2 Landing Guidance 13
2.1 General Approach 17
2.2 Planetary Landing: Problem Formulation 19
2.2.1 Problem Statement 19
2.2.2 Parametric Trajectory Formulation 20
2.2.3 Trajectory Constraints 24
2.2.4 Optimization Problem 28
2.3 Asteroid/Small Body Landing: Problem Formulation . 29
2.3.1 Problem Statement 29
2.3.2 Parametric Trajectory Formulation 31
2.3.3 Trajectory Constraints 33
2.3.4 Optimization Problem 36
2.4 Optimization Algorithms 37
2.4.1 Modified Compass Search 38
2.4.2 Differential Algebra Optimization 39
2.5 Planetary Landing Simulation and Test 46
2.5.1 MCS - Algorithm Tuning 47
2.5.2 MCS - Objective and Constraints Functions . . . 49

2.5.3 DA Optimization - Algorithm Tuning 51
2.5.4 DA Optimization - Objective and Constraints Functions 52
2.5.5 MCS vs DA Optimization Comparison 52
2.5.6 Landing Simulation: Nominal Navigation Errors 58
2.5.7 Landing Simulation: Sensitivity to Navigation Errors 60
2.6 Asteroid Landing Simulation and Test 64
2.6.1 Algorithm Performance Estimation 65
2.6.2 Landing Simulation: Exact Measures 65
2.6.3 Landing Simulation: Navigation Errors 67
2.6.4 Landing Simulation: Waypoint Trajectory 71

3 Hazard Detection 75
3.1 Nominal HDA Maneuver 77
3.2 System Architecture 79
3.2.1 Input and Preprocessing 80
3.2.2 Information Extraction and ANN Input Assembly 81
3.2.3 Artificial Neural Network and H-Map Computation 84
3.2.4 Target Selection 85
3.3 Network Training 88
3.3.1 Artificial Images Generation: SECRET-PLAN . 89
3.3.2 Ground Truth Solution 90
3.4 Performance Assessment 93
3.4.1 Real Images 96
3.4.2 Profiling 101
3.4.3 Sensitivity to Uncertainties 103

4 Conclusion 105
4.1 Roadmap for Future Research 107

Bibliography 111

List of Figures

1.1 Autonomous Landing system functional architecture. . 6

2.1 Ground reference system. 19
2.2 Flight reference system. 21
2.3 Body-fixed reference system. 22
2.4 Glide-slope constraint. 27
2.5 Body-fixed asteroid reference frame. 30
2.6 Initial control acceleration. 32
2.7 Glide-slope and minimum altitude constraints. 35
2.8 Analogy between floating point numbers and DA representation of functions. 40
2.9 Lunar landing: computational time as a function of approximation order N. 48
2.10 Lunar landing: position error at landing as a function of approximation order N. 49
2.11 Lunar landing: objective function example. 50
2.12 Lunar landing: MCS optimization example case. . . . 50
2.13 Lunar landing: DA Effective Area 51
2.14 Lunar landing: DA optimization example cases. 53
2.15 Lunar landing: Attainable area and fuel consumption comparison. 55
2.16 Lunar landing: comparison to true optimal solution example, retargeting maneuver. 56
2.17 Lunar landing: comparison to true optimal solution example, nominal maneuver. 57
2.18 Lunar landing: DA optimization computation time . . 58
2.19 Lunar landing simulation ($M = 1000$): 3D Landing trajectories. 61
2.20 Lunar landing simulation: dispersion in final position and velocity, horizontal components. 62

2.21 Lunar landing simulation: dispersion in final position and velocity, vertical components. 62
2.22 Lunar landing simulation: final attitude dispersion. . . 63
2.23 Lunar landing simulation: navigation errors effect. . . 63
2.24 NEA landing: algorithm computation time. 65
2.25 NEA landing: attainable area and fuel consumption. . 66
2.26 NEA landing: logical schematic of the GNC System. . 67
2.27 NEA landing simulation, no navigation errors, 3D trajectories. 68
2.28 NEA landing simulation, no navigation errors, position distribution. 68
2.29 NEA landing simulation, no navigation errors, velocity distribution. 69
2.30 NEA landing simulation, navigation errors, final position. 70
2.31 NEA landing simulation, navigation errors: final velocity. 70
2.32 NEA landing simulation, waypoint, 3D trajectories. . . 71
2.33 NEA landing simulation, waypoint, final position. . . . 72
2.34 NEA landing simulation, waypoint, final velocity. . . . 72
2.35 NEA landing simulation: single maneuver vs waypoint comparison. 73

3.1 Nominal Hazard Detection and Avoidance maneuver. . 79
3.2 Hazard detection system logical scheme 80
3.3 Multilayer feed-forward network structure. 85
3.4 Cascade network structure. 86
3.5 Cascade network structure. 90
3.6 Ground truth solution computation. 92
3.7 Computed hazard map. 93
3.8 $z_{\max}$ threshold tuning. Safety and Completeness ratios 95
3.9 $z_{\max}$ threshold tuning. Overall performance index. . . 96
3.10 Target ranking and selection. Multilayer ANN. 97
3.11 Target ranking and selection. Cascade ANN. 97
3.12 Lunar real surface image application. 99
3.13 Comet 67P Churyumov-Gerasimenko, original frame and hazard maps. 100
3.14 Hazard Detection, computation time breakdown. . . . 102

4.1 PoliMi landing simulation facility. Comparison between a real image and its correspondent artificial rendered with SECRET-PLAN. 109

List of Tables

2.1 Lunar landing simulation: lander architecture assumptions. 47
2.2 Lunar landing nominal case. 47
2.3 Complete lunar landing simulation: initial and final conditions for Monte Carlo (MC) analysis. 59
2.4 Lunar landing simulation: IMU performance properties. 59
2.5 1999 RQ_{36} “Bennu” nominal parameters. 64
2.6 NEA landing: MC parameters, initial and target states. 64

3.1 Camera model parameters for artificial images renderings. 90

Glossary

ACS	Attitude Control System.
ALHAT	Autonomous Landing Hazard Avoidance Technology.
ANN	Artificial Neural Network.
BRF	Body-fixed Reference Frame.
CGL	Cebyshev-Gauss-Lobatto.
CLS	Candidate Landing Site.
DA	Differential Algebra.
DEM	Digital Elevation Model.
EDL	Entry, Descent, and Landing.
ESA	European Space Agency.
ESM	European Service Module.
FP	Floating Point.
FRS	Flight Reference System.
GNC	Guidance, Navigation, and Control.
GRS	Ground Reference System.
HDA	Hazard Detection and Avoidance.
IMU	Inertial Measurement Unit.
ISRO	Indian Space Research Organisation.
IVP	Initial Value Problem.
LIDAR	Light Detection And Ranging.
MC	Monte Carlo.
MCS	Modified Compass Search.
MER	Mars Exploration Rover.
MSE	Mean Square Error.
NASA	National Aeronautics and Space Administration.
NEA	Near Earth Asteroid.
NLP	Non Linear Programming.
NLS	Nominal Landing Site.

NPAL	Navigation for Planetary Approach and Landing.
ODE	Ordinary Differential Equations.
PILOT	Precise Intelligent Landing using On-board Technology.
PROSPECT	Platform for Resource Observation and in-Situ Prospecting in support of Exploration, Commercial exploitation & Transportation.
PWPF	Pulse Width Pulse Frequency.
SECRET-PLAN	SElenitic Camera with Ray-traced Environment and Terrain for Planetary LANding Simulator.
SINPLEX	Small Integrated Navigation system for PLanetary EXploration.
SLS	Space Launch System.
TGO	Trace Gas Orbiter.
TLS	Target Landing Site.
TRL	Technology Readiness Level.

"I didn't know you could fly a plane."
"Fly, yes. Land, no."

Indiana Jones and the Last Crusade

1 Introduction

Certainly, landing is one of the most critical tasks that could be involved in a space mission. The Entry, Descent, and Landing (EDL) can be often considered as a mission bottleneck: a failure encountered in this phase would lead with high probability to the complete loss of the spacecraft. And even today, despite of the relatively large number of successes attained in past years, landing is complex and difficult, as denoted by the recent failure of the European Space Agency (ESA) lander module Schiapparelli on Mars [1].

Moreover, it is going to be even more important in the next future: in fact, a renewed interest in Solar System exploration has brought in the last years to the design of several missions involving landing maneuvers. Schiapparelli itself was just a single component of the more articulated ESA/Roscosmos ExoMars program: together with it, the Trace Gas Orbiter (TGO) was successfully released in a high eccentric Mars orbit, with the purpose to search for signatures of biological processes in the Martian atmosphere. The huge amount of data collected by the lander during the descent will be exploited to identify criticalities and improve the project of a second mission, scheduled for the 2020 with the aim to deliver a rover on the surface of the planet [2]. And beyond the ExoMars program, a Mars sample return mission has been already included as a flagship mission in the

ESA's Aurora program to take place in the timeframe 2020-2025 [3].

Mars is a privileged target for scientific missions. NASA has nowadays a long heritage in successful landings, begun with the Viking program in the Seventies [4], continued more recently with the Mars Pathfinder in 1997 [5], the two Mars Exploration Rover missions in 2004 [6, 7], the Phoenix lander [8–12], and culminated with the landing of the rover Curiosity in August 2012 [13, 14]. Also NASA is planning the landing of another rover for the 2020, in the larger context of a Mars sample return [15].

Also human space flight has returned to be a topic of discussion: NASA is planning to bring humans back in space with the development of the Space Launch System (SLS), whose maiden flight is scheduled for 2017 [16, 17]. ESA will supply the Orion/MPCV European Service Module (ESM) for the unmanned Exploration Mission-1, including ground and flight operation support [18]. Targets for the subsequent manned Exploration-2 and 3 missions are under study, including Near Earth Asteroids (NEAs) and the Moon as possible destinations [19]. Provisions for the construction and delivery of a second ESM have been taken. The SLS/Orion system is going to be the basis for future manned missions to Mars after 2030 [20]. The human exploration of Mars is nowadays considered as one of the great objectives of the next decades, and it has started to attract also private space companies and capitals [21].

Together with Mars, the Moon still remains a main destination for exploration. Besides scientific research, our satellite is considered also as a convenient environment to test and develop technologies required to support future interplanetary missions [22]. ESA has conducted several studies concerning a possible unmanned lunar lander [23]: although the program was subsequently put on hold, the technologies developed in that frame are going to be tested in a joint program with the Russian Roscosmos agency, on Luna-Glob and Luna-Resurs landers programs [24–26], with the Precise Intelligent Landing using On-board Technology (PILOT) and the Platform for Resource Observation and in-Situ Prospecting in support of Exploration, Commercial exploitation & Transportation (PROSPECT) experiments. Also, new actors appeared on the scene in the last years: in 2013, the Chinese mission Chang'e-3 successfully delivered a spacecraft carrying a rover on the lunar soil with a soft landing maneuver [27], while the Indian Space Research Organisation (ISRO) is planning to do the same with

the Chandrayaan-2 mission in the next few years [28].

In addition to missions toward planets and their moons, there is a strong interest in visiting small bodies as asteroids and comets. A typical high-autonomy scenario in this case is the close approach to a low-gravity object, finalized to either touch and go operations or landing. The ESA Rosetta probe, launched in March 2004, has performed a rendezvous with the comet 67P/Churyumov-Gerasimenko in August 2014 [29]. The release of the lander Philae, with the objective to collect and analyze on-board samples of comet's soil, has been successfully performed the next 12th November [30]. The OSIRIS-REx spacecraft, launched by NASA in 2016, is traveling to the NEA Bennu, to study it in detail, and bring back a sample to Earth [31]. MarcoPolo-R, a project with similar objectives, has been studied by ESA as M-class candidate mission for the launch in 2022 [32], while in the FY2014 budget proposal, NASA has included a plan to robotically capture a small NEA and redirect it safely to a stable orbit in the Earth-moon system where astronauts can visit and explore it [33, 34]. ESA and NASA carried out a joint study called AIDA, including ESA's AIM and NASA's DART spacecraft, to rendezvous with the Didymos binary asteroid [35]. AIM will rendezvous and study the asteroid system. Then, it will release a lander named MASCOT-2 to one of the bodies, followed by two or more CubeSats with the purpose to perform high-resolution visual, thermal, and radar mapping of the smaller body (called Didymoon) to build detailed maps of its surface and interior [36–38]. Finally, it will observe closely the impact between Didymoon and the DART spacecraft, to characterize the effects of the collision on the orbit and on the internal structure of the secondary body. Unfortunately, the mission has been suspended at the last ESA Ministerial Council held in Lucerne, Switzerland, in December 2016 due to problems in budget allocation [39]. Nevertheless, investigation about small bodies in the Solar System still remains a topic of interest for the agency.

All the examples mentioned above share the common problem of designing a landing on a celestial body. In these cases, safety is the main driver in mission analysis and design process.

During the last decades, several improvements in automatic landing precision have been achieved [14], but high uncertainties in attainable position at touchdown still impose severe requirements on the selection of the landing site. The traditional selection process is very

complex, with the strong limitation of fitting the absolute landing site dispersion ellipse in a safe area that can be tens of kilometers wide [40–46]. A powered descent phase is usually introduced at the end of the landing maneuver to guide the spacecraft to the target final altitude with null velocity, with the aim to ensure a safe touchdown, but with no control on the horizontal final position. The landing accuracy relies mainly on the precision in the determination of the states at the beginning of the entire maneuver [13]. These limitations are not going to be acceptable for the next space systems generation. Often, scientifically relevant sites on the celestial body of interest are associated with hazardous terrain features or confined in small areas; in other cases there is no possibility to completely characterize an interesting region in advance with the required accuracy. Moreover, in the case of planetary landing, the short duration of the maneuver, together with telecommunications delays, makes a continuous control from the ground impossible. In case of proximity maneuvers around low gravity bodies, the long duration of the operations allows a certain degree of remote control: the operation sequence can be designed and corrected step by step over a timespan of days. But even in these cases, high accuracy is still extremely difficult without an on-board autonomous guidance system [45, 47], while efficient counteraction to unexpected events or failures is impossible at all, as demonstrated by the uncontrolled bounce of the ESA lander Philae during the landing on the comet 67P [48, 49].

In addition, in the next future reusable launch vehicles are going to become a key factor to achieve a simple and affordable access to space. Between 2015 and 2016 the SpaceX Falcon 9 was the first commercial launcher able to put in orbit its payload and to come back to Earth successfully [21, 50]. In prospect of permanent human settlements not only on Earth, but also on the Moon and other planets like Mars, landing capabilities with a precision higher than tens of meters at maximum are going to become a common requirement.

This is why precise and autonomous landing capability is a key feature for the next space systems generation. The possibility to adapt the trajectory during the descent would reduce the landing dispersion, making possible the execution of both absolute and relative correction maneuvers. At the same time, in conjunction with the capability to distinguish hazardous from safe landing areas, the safety criteria during the mission analysis could be relaxed, leaving to the system

the task of the Hazard Detection and Avoidance (HDA), dramatically increasing in this way the robustness and the flexibility of the future exploration missions.

1.1 Autonomous Landing GNC Chain

An autonomous landing system with HDA capabilities should be able to scan the area around the landing site, to verify if the nominal target can be reached with the required level of safety and, if not, to seek for an alternative safe and reachable one. Then, a new landing path toward the updated target should be computed, followed by the execution of the divert maneuver. The main difference from a classical "blind" landing system is the presence of an autonomous Guidance, Navigation, and Control (GNC) chain, made up by three core components:

- *Adaptive Guidance* subsystem.
- *Hazard Detection* subsystem;
- *Relative Navigation* subsystem.

Plus, a landing site selector algorithm has the role to join hazard detection and guidance modules. The overall system functional architecture is shown in Figure 1.1.

Adaptive Guidance Once a new target is defined, this system recomputes on the fly a new reference trajectory. This can be achieved by solving on-board an optimum control problem. Being the fuel mass a major part of the total vehicle mass, minimizing the fuel consumption is one of the most suitable criteria in determining an efficient strategy. Brute force optimization is not suitable for an on-board computation, due to the limited hardware and time available. Efficient trajectory formulations and optimization techniques must be developed in order to find a feasible trajectory with a computation time of the same order of magnitude of the control update frequency.

Hazard Detection and Target Selection This system (divided in 2 smaller subsystems in Figure 1.1 for the sake of clarity) analyzes the nominal landing site area in search of a safe place to land. In case of visible wavelength images, shadows, slopes and terrain roughness

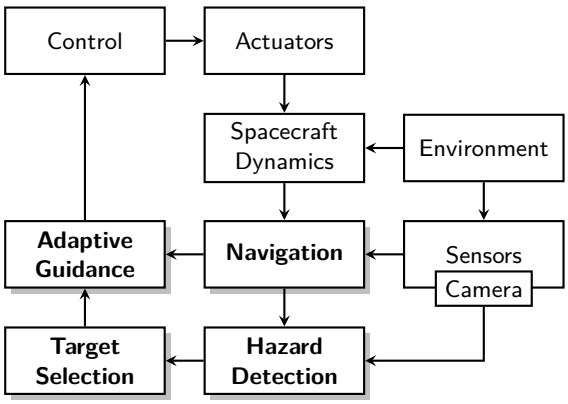


Figure 1.1: *Autonomous Landing system functional architecture. The core components of the HDA GNC chain are highlighted. Being vision-based systems one of the major topics of this work, the role of the navigation camera in the sensors block is highlighted as well, although not the only possible option.*

shall be detected, in order to exclude hazardous terrain. Other types of sensors, like Light Detection And Ranging (LIDAR), can provide also direct measures of distances and slopes. Once located the safe areas, the possible sites are ranked, following safety and reachability criteria, and the best one is selected as new target. Any additional criteria due to specific mission objectives (scientific targets, illumination etc. etc.) can be considered in addition. Hazard detection performances impose some requirements on the descent trajectory, in terms of visibility of the nominal landing site area. Efficient image processing algorithms and/or dedicated hardware are required, in order to achieve performances compatibles with on-board computation.

Precision Relative Navigation Relative navigation subsystem has the role to identify the states of the system with respect to the ground. To perform HDA tasks, an accuracy of the same order of magnitude of the dimension of the potential hazards is required. The accuracy guaranteed by classical navigation systems is too low for precision landing purposes. HDA capabilities require a high relative precision in the determination of position and speed, needed by the guidance in order to re-compute the trajectory. Navigation errors are the main source of inaccuracy in landing site attainment, for the knowledge of the initial states with respect to the target is required for a proper

guidance computation. Also, the hazard detection module requires at least some basic telemetry to correctly estimate size and position of suitable landing sites.

One of the most promising technologies to achieve such results is the family of vision-based algorithms. Cameras are widely used in space: they are relatively cheap and can have a very small form factor with light weight and reduced power consumption, with a proven flight heritage. A HDA system based on cameras could be tailored to different mission categories, from the smallest and cheapest to the most complex and largest. For this motivation a vision-based system is taken as reference configuration, preferred to more complex and expensive systems, e.g. LIDAR based GNC.

1.2 Previous Works

Currently, no complete HDA system has ever flown in a space mission. Anyway, in the last decade, different research programs have faced the problem. Different methods and technologies have been proposed, but in most of the cases additional effort is still required in order to increase the Technology Readiness Level (TRL) of these technologies up to a level suitable for their exploitation in the next years.

Initiated in 2001, the ESA Navigation for Planetary Approach and Landing (NPAL) project [51] has promoted the development of a feature-tracking based visual navigation system. Dedicated hardware for features extraction and tracking has been tested on the ESA Precision Landing GNC Test Facility [52]. In the frame of the ESA STARTIGER initiative, a multi-copter carrier platform was developed [53]. This carrier platform demonstrated a safe lowering and deployment of a rover mock-up to demonstrate the autonomous landing of an exploration rover on a planetary surface. A feature-tracking filter has been used for relative navigation, together with a very simple hazard detection system based on shadows and texture maps. The main goal of the Small Integrated Navigation system for PLanetary EXploration (SINPLEX) project, promoted by DLR, is to develop an innovative navigation system for exploration missions which include a landing and/or a rendezvous and capture/docking phase with a mass which is significantly lower than conventional systems [54]. Multiple sensors data fusion (exploiting cameras, inertial

sensors and LIDAR) has been tested on unmanned aerial vehicles for the validation of space oriented navigation technologies, in the context of the PERIGEO project, founded by the Spanish Centre for the Development of Industrial Technology [55]. Laser altimeter, Doppler LIDAR and flash LIDAR are instead the core sensors adopted in the NASA Autonomous Landing Hazard Avoidance Technology (ALHAT) project for both relative navigation and hazard detection purposes, started in 2006. The system has in part been tested in flight on helicopters [56] and on the rocket-propulsive terrestrial testbed Morpheus vehicle [57].

1.3 Dissertation Overview and Main Contributions to the Field

This work focuses on the two most innovative components of the autonomous landing GNC chain: the *Adaptive Guidance* and the *Hazard Detection* subsystems. A target ranking and selection algorithm has been developed as well, to properly estimate the system performances. A vision-based system, with a monocular visible wavelength camera as main sensor, has been assumed as main configuration.

The problem of visual navigation has been widely studied in recent years in the field of the robotics [58–61], although several further improvements are still needed to effectively adapt existing algorithms to space applications. The required level of accuracy is achieved by fusing data of usual navigation sensors (inertial sensors and laser/radar altimeters) with the tracking of landmarks obtained by cameras. Research on this topic has been carried out in parallel with this work, but it is not been included here.

A light and fast guidance algorithm is presented in Chapter 2. On-ground Trajectory optimization for landing is indeed a well known topic since years, for mission analysis purposes. On the contrary, the adaptation of the trajectory on-board, to autonomously cope with dispersions, navigation errors or ordered diversions for hazard avoidance is still an open point. In this work, a trajectory computation method is proposed that:

- Allows the (sub)optimization of the fuel consumption;
- Can compute large diversions;

- Ensures the satisfaction of all the constraints imposed by the system in terms of allowed hovering area, maximum control torques, maximum and minimum thrust;
- Is light enough to run on-board, without intervention from ground.

The trajectory is modeled as a polynomial of minimum degree required to satisfy the boundary constraints. A reduced set of parameters (including the time of flight, the initial thrust magnitude, and possible additional parameters that are problem dependent) is left free for a constrained optimization able to minimize the fuel consumption, satisfying at the same time all the other constraints imposed by the actual spacecraft capabilities, and leaving enough margins to further corrections in case of multiple retargetings. Classical guidances (like the one adopted for the Apollo mission, described in [62]) do not consider path constraints like attitude control torques magnitude, or minimum thrust magnitude. Only small diversions can be considered to avoid infeasibility with reasonable confidence. Then, many algorithms are not suitable for hazard detection and avoidance tasks [63]. On the contrary, a very recent class of guidance algorithms, based on pseudospectral collocation associated with convex optimization is able to find nearly optimal solutions to large diversions, but involving tens (or even hundreds) of optimization variables [64, 65]. An intermediate approach is here proposed, leading to the computation of suboptimal solutions for large diversions with only 2 or 3 optimization variables. The flexibility of the proposed method is addressed with two different applications, a lunar landing and the close approach to a low gravity small celestial body in the NEA category. The robustness of the formulation is demonstrated through Monte Carlo simulations coupled with a very basic optimizer.

Moreover, a novel, more complex optimizer, based on Differential Algebra (DA), is developed, capable to solve the optimization problem in a fast and reliable way. The objective and constraints functions are modeled as low order Taylor Maps. The general features of the function are easily got, leading in a few iterations in the neighborhood of the optimal solution. There, the current point is refined, exploiting the property of the Taylor series to converge to the true function value in proximity to the expansion point.

In Chapter 3 a novel hazard detection and target selection algo-

rithm is expounded. The capability of Artificial Neural Networks to extrapolate underlying rules in complex datasets is exploited to obtain an automatic classifier that, given a single image of the landing area taken by a monocular navigation camera, is able to build a hazard map of the terrain surrounding the target. This method removes the need to manually establish heuristic correlations between image features (e.g intensity mean and variance, etc.) and terrain physical features (like slopes and rocks), leaving to the Artificial Neural Network (ANN) training process the task to identify these correlations automatically; the training is completely run off-line, before flight, while only the trained network runs on-board, with a minimum computational burden. Based on the hazard map, the target selection algorithm locates and ranks the candidate landing sites following safety and reachability criteria, leading to the selection of the best target. A visual simulation tool is also developed to build a coherent and effective image dataset used for the networks training.

Conclusions are drawn in Chapter 4. Obtained results are discussed, and a possible research roadmap toward a full integration and testing of the algorithms developed is outlined in the end, in prospect of a future practical exploitation in real space systems.

1.4 Bibliographic Disclaimer

During the years of my PhD, I presented updates of my work in many conferences and I had also the possibility to publish part of them in peer reviewed journals. Therefore, most of the work presented in this thesis has already been published in different articles. The most significant are listed below here:

- P. Lunghi, M. Lavagna, and R. Armellin, "A semi-analytical guidance algorithm for autonomous landing," *Advances in Space Research*, vol. 55, no. 11, pp. 2719–2738, 2015, DOI: `10.1016/j.asr.2015.02.022`
- P. Lunghi, M. Ciarambino, and M. Lavagna, "A multilayer perceptron hazard detector for vision-based autonomous planetary landing," *Advances in Space Research*, vol. 58, no. 1, pp. 131–144, 2016, DOI: `10.1016/j.asr.2016.04.012`
- P. Lunghi, R. Armellin, P. Di Lizia, and M. Lavagna, "Semi-analytical adaptive guidance computation based on differential algebra for autonomous planetary landing," in *26th AAS/AIAA Space Flight Mechanics Meeting*, vol. 158 of *Advances in the Astronautical Sciences*, pp. 2003–2022, Univelt, San Diego, CA, Jan. 2016

Other publications in which part of the work already appeared are:

- P. Lunghi, M. Lavagna, and R. Armellin, "Semi-analytical adaptive guidance algorithm for fast retargeting maneuvers computation during planetary descent and landing," in *12th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, ESA/ESTEC, Noordwijk, The Netherlands, May 2013
- P. Lunghi, M. Lavagna, and R. Armellin, "Adaptive semi-analytical guidance for autonomous planetary landing," in *64th International Astronautical Congress (IAC)*, Beijing, China, Oct. 2013
- P. Lunghi, M. Lavagna, and R. Armellin, "Semi-analytical guidance algorithm for autonomous close approach to non-cooperative low-gravity targets," in *24th AAS/AIAA Space Flight Mechanics Meeting*, vol. 152 of *Advances in the Astronautical Sciences*, pp. 731–746, Univelt, San Diego, CA, Jan. 2014
- P. Lunghi and M. Lavagna, "A neural network based hazard detection algorithm for planetary landing," in *9th International ESA Conference on Guidance, Navigation, and Control Systems*, Porto, Portugal, Jun. 2014
- P. Lunghi, M. Ciarambino, and M. Lavagna, "Vision-based hazard detection with artificial neural networks for autonomous planetary landing," in *13th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, ESA/ESTEC, Noordwijk, The Netherlands, May 2015
- P. Lunghi, M. Ciarambino, and M. Lavagna, "A multilayer perceptron hazard detector for vision-based autonomous planetary landing," in *AAS/AIAA Astrodynamics Specialist Conference 2015*, vol. 156 of *Advances in the Astronautical Sciences*, pp. 1633–1650, Univelt, San Diego, CA, Aug. 2015
- P. Lunghi, M. Ciarambino, and M. Lavagna, "Simulation facility for vision-based planetary landing systems," in *23th Conference of Italian Association of Aeronautics and Astronautics - AIDAA*, Nov. 2015
- P. Lunghi, M. Ciarambino, and M. Lavagna, "A new test facility for vision-based hazard detection and avoidance systems for planetary landing maneuvers," in *66th International Astronautical Congress (IAC)*, Jerusalem, Israel, Oct. 2015
- P. Lunghi, M. Ciarambino, and M. Lavagna, "A hazard detection and avoidance system for autonomous planetary landing," in *23th Conference of Italian Association of Aeronautics and Astronautics - AIDAA*, Torino, Italy, Nov. 2015
- P. Lunghi, M. Ciarambino, L. Losi, and M. Lavagna, "Development, validation and test of optical based algorithms for autonomous planetary landing," in *6th International Conference on Astrodynamics Tools and Techniques (ICATT)*, Darmstadt, Germany, Mar. 2016

Richard pushed the phone back into its cradle and slammed his car into reverse for twenty yards to have another look at the sign-post by the road junction he'd just sped past in the mist. He had extracted himself from the Cambridge one-way system by the usual method, which involved going round and round it faster and faster until he achieved a sort of escape velocity and flew off at a tangent in a random direction, which he was now trying to identify and correct for.

Douglas Adams, Dirk Gently's Holistic Detective Agency

2 Landing Guidance

Uncertainties in the determination of the system states can propagate during the landing, making the spacecraft drift with respect to its nominal path. The capability to continuously correct the trajectory on-board would improve both relative and absolute precision, also in case of nominal maneuver without safety diversions. In case of a hazard avoidance maneuver, a completely new trajectory must be generated.

In this chapter, the problem of the adaptive guidance for spacecraft landing is analyzed. The guidance problem consists in finding a feasible trajectory that drives a spacecraft, starting from a given set of initial states, to a correspondent set of desired final states (the target). The control profile of the spacecraft is defined by a time history of the thrust vector. Being the control variables continuous, the subspace of the feasible solutions is generally infinite (except for the particular case in which there is just one feasible solution): an additional criterion is then required to select the best solution in the feasibility domain.

In this work, a fuel optimization criterion is followed. The minimization of the propellant consumption is a goal of every space mission, as it allows a reduction in launch mass or an increase in payload, and thus in the scientific return of the mission. In the

specific case of the landing guidance, a fuel optimal approach in HDA computation contributes to maximize the attainable landing area, consequently increasing the chances to find a safe landing site. Furthermore, it improves the possibility to perform additional subsequent diversions, if requested by hazard detection as the altitude decreases and the ground is analyzed with better resolution, or in case of unexpected events. That is why propellant minimization can be considered as an ideal criterion in the design of diversion trajectories. Since brute force numerical optimization usually implies heavy computation with no guarantee of convergence, particular care is required in finding efficient and robust formulations and optimization methods, to obtain a reliable algorithm capable to run autonomously on-board a spacecraft.

Different approaches to the problem have been adopted during the years. At the dawn of the spaceflight era, the available computational capability imposed strong limitations over the complexity of GNC algorithms: a trajectory based on a quartic polynomial in time, with no optimization involved, was used during the Apollo missions [62]. This class of methods allows to find a closed-form, explicit solution, fast and easy to be implemented on-board, but is not able to take into account any additional constraint except for boundary conditions [63]. Without the certainty that the solution is always feasible, only limited corrections along the nominal path are allowed: the probability to find an unfeasible solution is unpredictable, but it is higher as the requested diversion increases. Despite of these drawbacks, the simplicity of this method made it very popular: a derivative of the Apollo lunar descent guidance has been still considered in recent years for the Mars Exploration Rover (MER) missions [79], while another variant of this explicit scheme based on a polynomial formulation of the acceleration – called *E-Guidance* – has been recently considered to accomplish HDA tasks [80].

As the available computational capabilities increased, various other approaches to obtain both numerical and approximate optimal solutions of the pinpoint landing terminal guidance problem have been proposed. In [81] the first-order necessary conditions for the problem are developed, and it is shown that the optimal thrust profile has a maximum-minimum-maximum structure. Direct numerical methods for trajectory optimization have been widely investigated, not requiring the explicit consideration of the necessary conditions

and with better convergence properties [82]. These methods have been used together with Chebyshev pseudospectral techniques, to allow the reduction of the number of the optimization variables [83]. Also convex programming has been proposed to guarantee the convergence of the optimization; this approach, coupled with direct collocation methods, has proved that the size of the region of initial states for which there exist feasible trajectories can be increased drastically (more than twice) compared to the traditional polynomial-based guidance approaches, but at the price of a higher computational cost, due to the high number (up to hundreds) of optimization variables involved [64]. This method has been coupled with a minimum-landing-error approach, in order to compute a landing trajectory even in case a feasible solution for the selected landing site is not found [65].

In the case of asteroids and comets, landing and close proximity operations present some peculiarities, due to their small size and irregular shape. In particular, the gravitational acceleration is very weak and variable in function of the relative position of the spacecraft with respect to the target. Due to that, orbits are generally complex and non periodic, and stable only in certain regions [84]. A detailed characterization of the target allows pure ballistic landing trajectories to achieve high precision, but unmodeled dynamics and unpredictable events can lead to uncontrolled bounces and multiple touchdowns with undesired consequences [45, 48]. Zero Emission Effort/Zero Emission Velocity guidance has been proved to produce a good approximation of the fuel-optimal trajectory in close proximity maneuvers around asteroids [85], and it has been applied together with high-order sliding mode control to increase robustness to disturbances and unmodeled dynamics [86, 87]. Convex optimization has been adapted also for low gravity bodies: also in this case, the solution obtained is near to the true theoretical optimum, but with a large number of optimization variables [88].

A guidance algorithm capable to dynamically recompute and correct the landing trajectory during the descent is here developed, allowing the on-board choice of the landing site, as required by systems that have to operate in full autonomy. An innovative semi-analytical approach is proposed: the trajectory is parameterized in a polynomial form, depending only on a few parameters that can be efficiently optimized by fast optimization algorithms. Traditional closed-form guidance schemes (such as Apollo guidance, E-guidance)

are sub-optimal and do not include explicitly path constraints, potentially leading to infeasible trajectories. On the other hand, fully numerical methods, although extremely flexible in terms of optimality and constraints evaluation, require the handling of hundreds of optimization variables or complex gradient based optimization techniques, and thus they are computationally intensive [63]. With the proposed approach, the feasibility region is increased with respect to traditional polynomial algorithms, avoiding at the same time the higher computational cost of complex optimization methods, being the number of variables to be optimized very low. Furthermore, additional parameters allow us to include path constraints usually not taken into account by traditional guidance algorithms. As a result a simple, efficient, and nearly-optimal guidance law is obtained. The proposed method is flexible enough to be tailored on problems with very different time scales: as demonstration, it is here applied to both the problems of a planetary and an asteroidal landing.

Specific ad-hoc optimizers are developed to obtain a fast computation, as required by on-board systems. At first, a very simple and basic derivative-free method is tested, not for an in-flight use, but to assess the effectiveness and robustness of the proposed formulation. Then, a more complex method based on DA is developed. In DA the usual algebraic operators are extended from real numbers to functions, modeled as their Taylor expansions around a selected point, up to an arbitrary order. The three components of the acceleration are expressed as DA quantities, expanded around the nominal trajectory followed by the lander at the retargeting epoch: in this case the DA formulation leads to an exact representation of the acceleration profile, due to its polynomial nature. From the acceleration history, the mass trend is easily obtained through integration, leading to a DA representation of the objective and the constraints as functions of the two optimization variables around the expansion point). Such a representation carries additional information regarding the quantity that it represents: not only its value at the expansion point, but also its sensitivity with respect the variation of the parameters that concur to its determination (e.g. to the optimization variables).

The chapter is organized as follows: in Section 2.1 the general logic of the proposed algorithm is presented. Then, it is formalized in Section 2.2 for the planetary landing case, and in Section 2.3 for a NEA close approach. Section 2.4 expounds the optimization

algorithms developed to solve the problem in a fast and reliable way. Finally the results obtained in simulations of a lunar and an asteroidal landing are discussed in Sections 2.5 and 2.6; Monte Carlo simulations are exploited to assess the effectiveness of the proposed method.

2.1 General Approach

The retargeting problem, as part of a HDA system, involves the last part of the landing phase only. Hazard avoidance maneuvers take place in the last few kilometers before the touchdown, and a powered descent is assumed.

The proposed algorithm presents a general approach that can be tailored on the specific problem, from slow (low-gravity objects) to fast (planetary landing) dynamics. In this section the generic procedure of the adaptive guidance is presented. In the next sections, the method here expounded is applied to write the actual guidance laws for two very different cases, a planetary landing and a soft landing on a small asteroid-like target in a low gravity environment.

The implemented guidance is based on the following scheme:

1. The system translational dynamics are identified and expressed in the general form:

$$\begin{cases} \dot{\mathbf{r}} = \mathbf{v} \\ \dot{\mathbf{v}} = \mathbf{f}(\mathbf{r}, \mathbf{v}, m, \mathbf{T}) \\ \dot{m} = g(\mathbf{T}) \end{cases} \tag{2.1}$$

   where $\mathbf{r}$ is the position vector, $\mathbf{v}$ is the velocity vector, m is the mass of the spacecraft, and $\mathbf{T}$ is the thrust vector. $\mathbf{f}(\mathbf{r}, \mathbf{v}, m, \mathbf{T})$ and $g(\mathbf{T})$ are generic functions of states and thrust.

2. The boundary constraints are defined. It is assumed that the full spacecraft states $\mathbf{r}_0$, $\mathbf{v}_0$, and m_0 are known at the time t_0 when the retargeting is ordered. At final time t_f constraints on both position $\mathbf{r}_\mathrm{f}$ and velocity $\mathbf{v}_\mathrm{f}$ are considered. Additional boundary constraints on initial and final acceleration can arise from the actual system architecture, depending on propulsion and attitude control systems requirements. Initial acceleration

is expressed as function of initial thrust magnitude and, when needed, initial spacecraft attitude.

3. The acceleration profile is expressed in a polynomial form in time, of minimum order to satisfy the boundary constraints. By inverse dynamics, a complete control profile is obtained, function of time-of-flight and possible additional parameters. These parameters are problem-dependent and can vary for different applications, as described in the subsequent sections.

4. The problem is reduced to finding the values of these parameters, according to any additional constraint not implicitly satisfied by the polynomial formulation, minimizing the fuel consumption. Representing as $\mathbf{x}$ the vector of optimization parameters, the cost function is $f(\mathbf{x}) = m(t_0) - m(t_\mathrm{f})$, and the problem can be expressed in the form:

$$\min_{\mathbf{x}} f(\mathbf{x}) \text{ such that } \begin{cases} \mathbf{x}_L \leq \mathbf{x} \leq \mathbf{x}_U \\ \mathbf{c}_L \leq \mathbf{c}(\mathbf{x}) \leq \mathbf{c}_U \end{cases} \tag{2.2}$$

The search space for the optimization variables is defined by upper and lower bounds, $\mathbf{x}_U$ and $\mathbf{x}_L$ respectively. These are called *Box Constraints*. The elements of $\mathbf{c}(\mathbf{x})$ in Equation (2.2) are generally nonlinear functions of the optimization variables, also bounded between lower and upper limits $\mathbf{c}_L$ and $\mathbf{c}_U$. These constraints need to be satisfied during all the landing maneuver, and they are called *Path Constraints*.

In writing the actual guidance law, these general steps are tailored to the specific case:

- The dynamic system is written;
- The proper assumptions about the spacecraft architecture and mission requirements are made to derive the actual boundary, box, and path constraints;
- The remaining unknown parameters are identified as optimization variables, and the resulting optimization problem is formalized.

2.2 Planetary Landing: Problem Formulation

A planetary landing is characterized by fast dynamics. The expected time of flight is in the order of magnitude of 1 min [63, 64], and the mass is expected to significantly change during the maneuver.

2.2.1 Problem Statement

In the case of a planetary landing, distances, for both downrange and altitude, are small compared to the planet's radius; thus, the assumption of a constant gravity field with flat ground is appropriate. This assumption is widely used and accepted in the development of terminal guidances for planetary landing [62–64, 79, 80]. Furthermore, aerodynamic forces are neglected. In fact the eventual presence of atmosphere (especially with low density, as in the case of Mars) could be negligible due to the relative low velocity (on the order of $100\,\mathrm{m\,s^{-1}}$), and shape of the spacecraft (without lifting surfaces), and the associated forces can be then treated as disturbances [64].

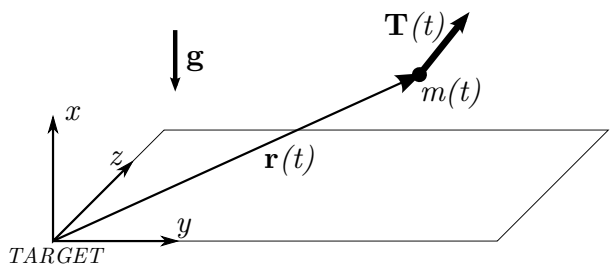


Figure 2.1: *Ground reference system.*

The translational dynamics of the spacecraft are expressed in a Ground Reference System (GRS) (see Fig. 2.1), where x is the altitude, y is called the *Downrange* direction and z is the *Crossrange* direction. The dynamics is described by the equations

$$\begin{cases} \dot{\mathbf{r}} = \mathbf{v} \\ \dot{\mathbf{v}} = \dfrac{\mathbf{T}}{m} + \mathbf{g} \\ \dot{m} = -\dfrac{T}{I_{\mathrm{sp}} g_0} \end{cases} \tag{2.3}$$

where $\mathbf{g}$ is the constant acceleration of gravity vector of the planet, I_{sp} the specific impulse of the main engine, and g_0 the standard gravity acceleration on Earth. The thrust net magnitude is indicated with $T = \|\mathbf{T}\|$.

In this system, the thrust vector acts as control variable. The mass equation is linked to the control acceleration by the thrust-to-mass ratio $\mathbf{P}$:

$$\mathbf{P} = \mathbf{T}/m = \dot{\mathbf{v}} - \mathbf{g} \tag{2.4}$$

Then, the mass equation in system (2.3) can be rewritten as

$$\dot{m} = -\frac{P}{I_{\mathrm{sp}} g_0} m \tag{2.5}$$

which is a first order linear ordinary differential equation whose solution is

$$m(t) = m_0 \exp\left(-\int_{t_0}^{t} \frac{P(\tau)}{I_{\mathrm{sp}} g_0} d\tau \right) \tag{2.6}$$

At the time t_0 the initial states $\mathbf{r}_0$, $\mathbf{v}_0$ and m_0 are supposed to be known. At the end of the maneuver, at time t_{f}, final states $\mathbf{r}_{\mathrm{f}}$ and $\mathbf{v}_{\mathrm{f}}$ are required. Then, the optimal guidance problem is to find a control profile $\mathbf{T}(t)$, to bring the system from the initial to the target final states, that maximizes the final mass compatibly with all the constraints imposed by the actual system architecture.

2.2.2 Parametric Trajectory Formulation

The main thruster is assumed to be rigidly connected to the spacecraft body. Thus, the direction of the thrust vector is determined directly by the spacecraft attitude. The spacecraft attitude is expressed relatively to an auxiliary reference system, called Flight Reference System (FRS), defined by the unit vectors $[\mathbf{x}_{\mathrm{f}}\ \mathbf{y}_{\mathrm{f}}\ \mathbf{z}_{\mathrm{f}}]^T$ (see Fig. 2.2), centered at the center of mass of the spacecraft, the $\mathbf{x}_{\mathrm{f}}$ axis pointing toward the downrange direction (y in GRS), the $\mathbf{z}_{\mathrm{f}}$ axis pointing downwards, an the $\mathbf{y}_{\mathrm{f}}$ axis forming a right-handed triad.

Attitude is defined as the rotation from FRS to the Body-fixed Reference Frame (BRF). The body axes $[\mathbf{x}_b\ \mathbf{y}_b\ \mathbf{z}_b]^T$ are assumed to be defined as in Fig. 2.3, where the $\mathbf{x}_{\mathrm{b}}$ direction is called *Roll*

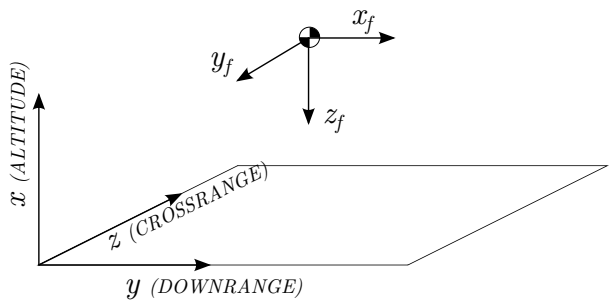


Figure 2.2: *Flight reference system.*

axis, $\mathbf{y}_\mathrm{b}$ is the *Pitch* axis and $\mathbf{z}_\mathrm{b}$ is the *Yaw* axis. The rotation is expressed in Euler angles, in the 231 form, where θ (pitch angle) is the first rotation around $\mathbf{y}_\mathrm{b}$, ψ (yaw angle) is the second rotation about $\mathbf{z}_\mathrm{b}$, and ϕ (roll angle) is the third rotation around $\mathbf{x}_\mathrm{b}$. The 231 form is preferred to the more traditional 321, because it avoids the presence of singularities in the angles determination, in the field of application of the landing phase (it is assumed that the thrust vector is never required to have a downward component). The attitude with respect to the flight reference frame is expressed by the direction cosine matrix

$$\mathrm{A_b} = \begin{bmatrix} c\psi c\theta & s\psi & -c\psi s\theta \\ -c\phi s\psi c\theta + s\phi s\theta & c\phi c\psi & c\phi s\psi s\theta + s\phi c\theta \\ s\phi s\psi c\theta + c\phi s\theta & -s\phi c\psi & -s\phi s\psi s\theta + c\phi c\theta \end{bmatrix} \tag{2.7}$$

where c and s are the abbreviated forms for cos and sin.

The rotation of the flight reference frame with respect to the ground is constant and expressed by the matrix

$$\mathrm{A_f} = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \tag{2.8}$$

Following these assumptions, the thrust vector can be represented as:

$$\mathbf{T}(t) = \mathrm{A_f}^T \mathrm{A_b}^T \begin{bmatrix} -T(t) \\ 0 \\ 0 \end{bmatrix} = -T(t) \begin{bmatrix} \cos\psi(t)\sin\theta(t) \\ \cos\psi(t)\cos\theta(t) \\ -\sin\psi(t) \end{bmatrix} \tag{2.9}$$

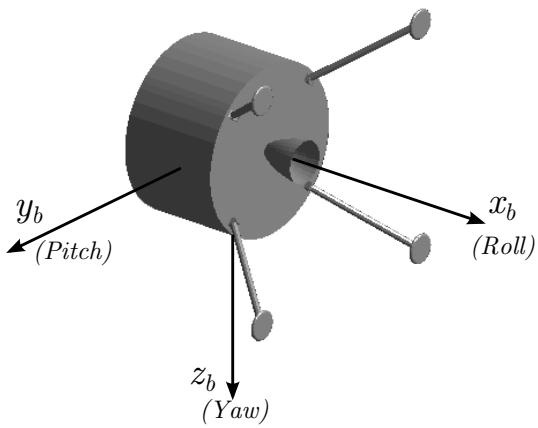


Figure 2.3: *Body-fixed reference system.*

By substituting Equation (2.9) in Equations (2.3) the system can be written in its scalar form as:

$$\begin{cases} \dot{x} = v_{\mathrm{x}} \\ \dot{y} = v_{\mathrm{y}} \\ \dot{z} = v_{\mathrm{z}} \\ \dot{v}_{\mathrm{x}} = -T\dfrac{\cos\psi\sin\theta}{m} + g_{\mathrm{x}} \\ \dot{v}_{\mathrm{y}} = -T\dfrac{\cos\psi\cos\theta}{m} \\ \dot{v}_{\mathrm{z}} = T\dfrac{\sin\psi}{m} \\ \dot{m} = -\dfrac{T}{I_{\mathrm{sp}}g_0} \end{cases} \tag{2.10}$$

Equations (2.10) show that the system is not affected by the roll angle ϕ. Due to this fact, ϕ is always considered as null. It must be taken into account also the fact that the sensors used by navigation and hazard detection would be probably required to maintain a relatively stable pointing during the landing. The condition of null roll angle makes easier the design of the spacecraft configuration. In this form, the control variables consist of the thrust magnitude T,

the pitch angle θ and the yaw angle ψ; anyway, the attitude profile is not a completely free parameter: pitch and yaw angles at time t_0 are known and fixed. This imposes an additional boundary constraint on the initial acceleration, which now depends only on the initial thrust magnitude:

$$\dot{\mathbf{v}}_0 = -\frac{T(t_0)}{m_0}\begin{bmatrix}\cos\psi_0\sin\theta_0\\ \cos\psi_0\cos\theta_0\\ -\sin\psi_0\end{bmatrix} + \begin{bmatrix}g_{\mathrm{x}}\\ 0\\ 0\end{bmatrix} \tag{2.11}$$

Moreover, at the end of the maneuver, the lander's attitude is required to be aligned with the local vertical on the Target Landing Site (TLS). This boundary constraint is expressed through the equation

$$\dot{\mathbf{v}}(t_{\mathrm{f}}) \times \hat{\mathbf{n}}_{\mathrm{LS}} = \mathbf{0} \tag{2.12}$$

where $\hat{\mathbf{n}}_{\mathrm{LS}}$ is the unit vector normal to the planetary surface at the TLS. In case of flat surface, $\hat{\mathbf{n}}_{\mathrm{LS}}$ is aligned with the x axis of the ground reference frame (see Fig. 2.1), and Equation (2.12) reduces to

$$\dot{v}_{\mathrm{y}}(t_{\mathrm{f}}) = \dot{v}_{\mathrm{z}}(t_{\mathrm{f}}) = 0 \tag{2.13}$$

No boundary constraint is put on the vertical component of the control acceleration at the final time t_{f}. This is because it is assumed that the main thruster can be turned off instantaneously. This is practically true, since the actuation of space thrusters is much faster than the spacecraft dynamics [89, 90]. Then, total of 17 boundary constraints are available for position, velocity and acceleration components: 6 on initial states, 3 on initial acceleration (function of initial thrust magnitude), 6 on target final states and 2 on final acceleration due to final attitude requirements:

$$\begin{cases}\mathbf{r}(t_0) = \mathbf{r}_0\\ \mathbf{v}(t_0) = \mathbf{v}_0\\ \dot{\mathbf{v}}(t_0) = \mathbf{f}(T_0, \theta_0, \psi_0)\\ \mathbf{r}(t_{\mathrm{f}}) = \mathbf{r}_{\mathrm{f}}\\ \mathbf{v}(t_{\mathrm{f}}) = \mathbf{v}_{\mathrm{f}}\\ \dot{\mathbf{v}}(t_{\mathrm{f}}) = [\mathit{free}, 0, 0]^T\end{cases} \tag{2.14}$$

The 3 components of the acceleration can be expressed in a polynomial form. The minimum order needed to satisfy boundary constraints is 2 for the vertical axis, 3 for the horizontal components:

$$\dot{\mathbf{v}}(t) = \begin{bmatrix} \dot{v}_{\mathrm{x}} \\ \dot{v}_{\mathrm{y}} \\ \dot{v}_{\mathrm{z}} \end{bmatrix} = \begin{bmatrix} \dot{v}_{0\mathrm{x}} + c_{1\mathrm{x}}t + c_{2\mathrm{x}}t^2 \\ \dot{v}_{0\mathrm{y}} + c_{1\mathrm{y}}t + c_{2\mathrm{y}}t^2 + c_{3\mathrm{y}}t^3 \\ \dot{v}_{0\mathrm{z}} + c_{1\mathrm{z}}t + c_{2\mathrm{z}}t^2 + c_{3\mathrm{z}}t^3 \end{bmatrix} \tag{2.15}$$

Integrating the acceleration two times, and applying boundary constraints, the trajectory becomes function of t_{f} and T_0. Once the acceleration profile is defined, the thrust-to-mass ratio can be obtained from Equation (2.4) and the thrust profile is:

$$\mathbf{T} = m\mathbf{P} \tag{2.16}$$

The mass profile is obtained by solving Equation (2.6). The analytical calculation of the integral exponent is complex, but can be easily attainable through numerical integration. The smooth polynomial profile of the control acceleration makes the numerical integration very efficient and precise even with a low amount of function evaluations, suitable for on-board computation. From the thrust unit vector $\hat{\mathbf{n}}_{\mathrm{T}} = \mathbf{T}/\|\mathbf{T}\|$ a complete guidance profile, in terms of Euler angles and thrust magnitude, is obtained, function of initial thrust magnitude T_0 and final time t_{f}:

$$\begin{cases} \theta = \tan^{-1}\left(\hat{n}_{\mathrm{Tx}}/\hat{n}_{\mathrm{Ty}}\right) \quad -\pi \leq \theta \leq 0 \\ \psi = \tan^{-1}\left(\hat{n}_{\mathrm{Tz}}\left(\hat{n}_{\mathrm{Tx}}^2 + \hat{n}_{\mathrm{Ty}}^2\right)^{-0.5}\right) \quad -\dfrac{\pi}{2} \leq \psi \leq \dfrac{\pi}{2} \\ \phi = 0 \end{cases} \tag{2.17}$$

The free parameter t_{f} can be replaced by the time-of-flight $t_{\mathrm{tof}} = t_{\mathrm{f}} - t_0$.

2.2.3 Trajectory Constraints

Box constraints, path constraints and any other additional constraint not implicitly satisfied by the polynomial formulation have the general form of Equation (2.2). For autonomous landing purposes, throttleable engines are required [62, 64, 89, 90]. The initial thrust magnitude is bounded to the thrust actually available on-board:

$$0 < T_{\min} \leq T_0 \leq T_{\max} \tag{2.18}$$

while the time-of-flight must lie between its lower and upper limit:

$$0 < t_{\min} \leq t_{\mathrm{f}} \leq t_{\max} \tag{2.19}$$

where

$$t_{\max} = m_{\mathrm{fuel}} \frac{I_{\mathrm{sp}} g_0}{T_{\min}}, \quad t_{\min} = \left(\frac{2 r_{0\mathrm{x}}}{T_{\max}/m_{\mathrm{dry}} - \|\mathbf{g}\|} \right)^{0.5} \tag{2.20}$$

The theoretical $t_{\max}$ is determined by the amount of fuel on board m_{fuel}, whereas $t_{\min}$ corresponds to the time required by the lander to reach the ground with maximum thrust pointing downward. The lower bound $t_{\min}$ does not corresponds to a feasible soft landing maneuver, but it is adopted as a theoretical lower limit to exclude singularities arising towards $t_{\mathrm{f}} = 0$. In the actual implementation, the inverse time-of-flight $\tau_{\mathrm{f}} = 1/t_{\mathrm{f}}$ is adopted instead of t_{f} as optimization variable, with corresponding $\tau_{\min} = 1/t_{\max}$ and $\tau_{\max} = 1/t_{\min}$. This choice depends exclusively on code efficiency purposes: all the considerations made so far remain valid, for there is no difference in using the time or its inverse (once the change of variable is done correctly).

From here onward, the notation $\mathbf{x} = [\tau_{\mathrm{f}}, T_0]^T$ is adopted for the optimization variables vector. All the trajectory constraints are rewritten in the form of $g(\mathbf{x}) \leq 0$; in order to achieve better convergence properties, they are also scaled to assume a value between -1 and 0 inside the feasible domain. In the case of thrust magnitude and time-of-flight, this leads to the normalized inequalities:

$$\frac{T_0 - T_{\max}}{T_{\max} - T_{\min}} \leq 0 \tag{2.21}$$

$$\frac{T_{\min} - T_0}{T_{\max} - T_{\min}} \leq 0 \tag{2.22}$$

$$\frac{\tau_{\mathrm{f}} - \tau_{\max}}{\tau_{\max} - \tau_{\min}} \leq 0 \tag{2.23}$$

$$\frac{\tau_{\min} - \tau_{\mathrm{f}}}{\tau_{\max} - \tau_{\min}} \leq 0 \tag{2.24}$$

Boundaries on thrust are applied also as path constraint:

$$\frac{T(t) - T_{\max}}{T_{\max} - T_{\min}} \leq 0 \tag{2.25}$$

$$\frac{T_{\min} - T(t)}{T_{\max} - T_{\min}} \leq 0 \tag{2.26}$$

The angular velocity of the spacecraft is limited by the actual control torques M_{Cmax} given by the Attitude Control System (ACS). The extrapolation of the exact torques from angles is not immediate, due to the coupled terms in the attitude dynamics. The objective is to characterize such a rotational rate constraint without coupling the problem to the rotational dynamics, to save computation time. Torques are approximated by the decoupled term due to the angular acceleration, which is a sufficiently accurate approximation in case of small angles and low angular speed. Exploiting this approximation leads to the following (normalized) inequality:

$$\left(\frac{I_{\max}}{M_{\mathrm{Cmax}}}\right)^2 \|\dot{\boldsymbol{\omega}}(t)\|^2 - 1 \leq 0 \tag{2.27}$$

in which $\dot{\boldsymbol{\omega}}$ is the derivative of the rotational velocity vector, and $I_{\max}$ is the maximum moment of inertia at initial time t_0. In this way, the on-board calculation of inertia properties can be avoided, and a safety margin in the torques calculation is introduced. The assumption of null roll angle implies that the spacecraft does not rotate around the control acceleration vector. Thus, this vector and the rotational velocity vector are perpendicular with respect to each other. It is then possible to obtain $\boldsymbol{\omega}$ from the relation:

$$\dot{\mathbf{a}} = \dot{a}\hat{\mathbf{a}} + \boldsymbol{\omega} \times \mathbf{a} = \frac{\dot{\mathbf{a}} \cdot \mathbf{a}}{\|\mathbf{a}\|} \frac{\mathbf{a}}{\|\mathbf{a}\|} + \boldsymbol{\omega} \times \mathbf{a} \tag{2.28}$$

where $\mathbf{a} = \dot{\mathbf{v}} - \mathbf{g}$ is the control acceleration vector (whose derivative $\dot{\mathbf{a}}$ is known exactly, being a polynomial in time), $\dot{a}$ is the derivative of the control acceleration modulus and $\hat{\mathbf{a}}$ is the control acceleration unit vector. The rotational rate vector can be then computed as:

$$\boldsymbol{\omega} = \frac{\left[\left(\frac{\dot{\mathbf{a}} \cdot \mathbf{a}}{\|\mathbf{a}\|^2}\right)\mathbf{a} - \dot{\mathbf{a}}\right] \times \mathbf{a}}{\|\mathbf{a}\|^2} = \frac{\mathbf{a} \times \dot{\mathbf{a}}}{\|\mathbf{a}\|^2} \tag{2.29}$$

In a feasible landing path, altitude is always greater than zero. This constraint can be improved considering a *Glide-Slope Constraint*. In this case the lander is required to remain in a cone defined by the maximum slope angle $\delta_{\max}$, as showed in Fig. 2.4. This constraint has a dual purpose: it assures that the the lander does not penetrate the ground, even in presence of bulky terrain features near the landing

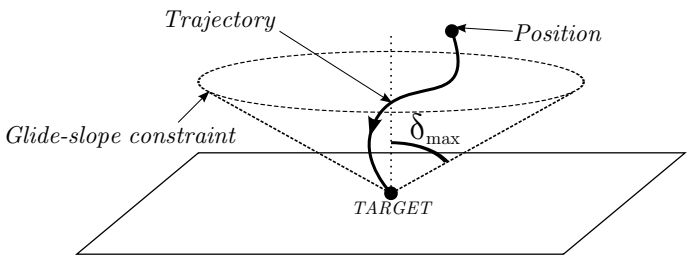


Figure 2.4: *Glide-slope constraint.*

site; at the same time it limits the angle of view on the target. In fact, the performances of vision-based navigation systems depend on inclination at which the target is observed [91, 92]. Following [64], the constraint takes the form

$$-\infty \leq \|\mathrm{S_g}\mathbf{r}(t)\| + \mathbf{c}_\mathrm{g}^T\mathbf{r}(t) \leq 0 \tag{2.30}$$

where

$$\mathrm{S_g} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.31}$$

$$\mathbf{c}_\mathrm{g}^T = \begin{bmatrix} -\tan\delta_\mathrm{max} & 0 & 0 \end{bmatrix} \tag{2.32}$$

The constraint can be rearranged and normalized as:

$$\frac{r_\mathrm{y}^2(t) + r_\mathrm{z}^2(t)}{r_\mathrm{x}^2(t)\ \tan^2(\delta_\mathrm{max})} - 1 \leq 0 \tag{2.33}$$

Path constraints need to be satisfied at every time instant during the landing. Here, they are evaluated discretely at Cebyshev-Gauss-Lobatto (CGL) points: once the values at these points are known, constraints can be reconstructed over the entire domain with Cebyshev pseudospectral collocation [93]. The values computed at CGL points are exploited also to compute unknown derivative terms ($\dot{\omega}$) through the Chebyshev differentiation matrix [93]. The evaluation of the control acceleration at CGL point is included in the numerical integration of the equation (2.6), to save computation time.

Finally, the polynomial formulation does not explicitly consider any constraint on final mass, which must be included between the initial value and the spacecraft dry mass. Since the mass trend is strictly monotone (by problem construction: the thrust magnitude cannot

be negative, an then the mass can only decrease) the evaluation of the maximum mass constraint is redundant: the only constraint with respect the minimum mass is verified. The initial mass value is still exploited to obtain the normalized relation:

$$\frac{m_{\mathrm{dry}} - m(t_{\mathrm{f}})}{m_0 - m_{\mathrm{dry}}} \leq 0 \tag{2.34}$$

2.2.4 Optimization Problem

The optimization problem for planetary landing takes the form:

$$\arg\min_{T_0, \tau_{\mathrm{f}}} f(T_0, \tau_{\mathrm{f}}), \quad f(T_0, \tau_{\mathrm{f}}) = -m(t_{\mathrm{f}}) \tag{2.35}$$

in the domain defined by the inequalities:

$$\frac{T_0 - T_{\max}}{T_{\max} - T_{\min}} \leq 0 \tag{2.36}$$

$$\frac{T_{\min} - T_0}{T_{\max} - T_{\min}} \leq 0 \tag{2.37}$$

$$\frac{\tau_{\mathrm{f}} - \tau_{\max}}{\tau_{\max} - \tau_{\min}} \leq 0 \tag{2.38}$$

$$\frac{\tau_{\min} - \tau_{\mathrm{f}}}{\tau_{\max} - \tau_{\min}} \leq 0 \tag{2.39}$$

subject to constraints:

$$\frac{T(t) - T_{\max}}{T_{\max} - T_{\min}} \leq 0 \tag{2.40}$$

$$\frac{T_{\min} - T(t)}{T_{\max} - T_{\min}} \leq 0 \tag{2.41}$$

$$\left(\frac{I_{\max}}{M_{\mathrm{Cmax}}}\right)^2 \|\dot{\boldsymbol{\omega}}(t)\|^2 - 1 \leq 0 \tag{2.42}$$

$$\frac{r_{\mathrm{y}}^2(t) + r_{\mathrm{z}}^2(t)}{r_{\mathrm{x}}^2(t)\ \tan^2(\delta_{\max})} - 1 \leq 0 \tag{2.43}$$

$$\frac{m_{\mathrm{dry}} - m(t_{\mathrm{f}})}{m_0 - m_{\mathrm{dry}}} \leq 0 \tag{2.44}$$

Denoting as $\mathbf{g}(\mathbf{x})$ the vector of the constraints defined by Equations (2.36–2.44), the problem can be written in the compact form:

$$\arg\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{subject to} \quad \mathbf{g}(\mathbf{x}) \leq 0 \tag{2.45}$$

The optimization could be solved with any Non Linear Programming (NLP) solver: the choice of this solver has a huge impact over the final convergence properties and computational time.

2.3 Asteroid/Small Body Landing: Problem Formulation

Relatively low thrust and slow dynamics are typical of maneuvers in low-gravity environment. The maneuver here presented is suitable for both landing and close approach to low-gravity objects such as NEA. The expected time is in the order of magnitude of several thousands of seconds, but with a limited change in mass [86–88].

2.3.1 Problem Statement

The motion of the spacecraft is modeled in an asteroid-fixed Cartesian frame, centered in the center of mass of the asteroid. Assuming the asteroid rotational rate as constant, the dynamics are described (using the same notation of the planetary landing case) by the well known equations of motion for uniform rotating frames

$$\begin{cases} \dot{\mathbf{r}} = \mathbf{v} \\ \mathbf{a} = \dot{\mathbf{v}} + 2\boldsymbol{\omega} \times \mathbf{v} + \boldsymbol{\omega} \times \boldsymbol{\omega} \times \mathbf{r} \\ \dot{m} = -\dfrac{T}{I_{\mathrm{sp}} g_0} \end{cases} \tag{2.46}$$

where $\mathbf{a}$ is the acceleration vector and $\boldsymbol{\omega}$ is the asteroid rotational rate vector. A restricted two-body model is considered: the spacecraft mass is assumed to be negligible compared to the asteroid. The adopted reference frame is represented in Fig. 2.5.

The acceleration vector acting on the spacecraft consists of different contributions

$$\mathbf{a} = \mathbf{g}(\mathbf{r}) + \mathbf{a}_{\mathrm{c}} + \mathbf{d} \tag{2.47}$$

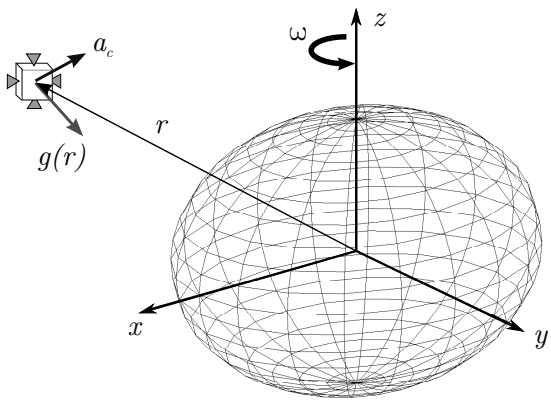


Figure 2.5: *Body-fixed asteroid reference frame.*

in which $\mathbf{g}(\mathbf{r})$ is the gravitational acceleration, function of the position in the asteroid reference frame, $\mathbf{a}_\mathrm{c}$ is the control acceleration and $\mathbf{d}$ is a term that includes disturbances (such as solar pressure or gravity terms not included in the adopted gravity model).

The asteroid is modeled as a tri-axial ellipsoid with uniform density ρ. This allows us to analytically evaluate the gravitational component of the acceleration as the gradient of its potential field $V_g(\mathbf{r})$ [94]:

$$\mathbf{g}(\mathbf{r}) = -\nabla\Big(V_g(\mathbf{r})\Big) \tag{2.48}$$

Although asteroids can have more complex motion, with non constant rotational rate vector, the adopted model well adapts to the class of objects target of current and near future missions, like 1999 RQ_{36} Bennu [31,86,87], 162173 Ryugu [95], and Didymos (which is a binary system, in which both the primary and the secondary bodies are assumed to be uniformly rotating ellipsoids, [36, 38]).

Even in case of an asteroid with high irregular shape, at a certain distance the gravity acceleration tends to be well described by a point mass model [96]. The adopted tri-axial ellipsoid is a higher order model, capable to properly describe the field in a wider region of space. However, to cope with extremely complex shapes at short distances, more complex approaches, such as a polyhedron shape model [97], could always be considered without impacting on the proposed guidance algorithm, for it is capable to handle any gravity model in the form $\mathbf{a}_\mathrm{g} = \mathbf{g}(\mathbf{r})$. Assuming the asteroid's rotational

rate vector aligned with the z axis, results $\boldsymbol{\omega} = [0, 0, \omega]^T$. Then, the dynamical system can be written in scalar form as:

$$\begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{z} = v_z \\ \dot{v}_x = 2\omega v_y + \omega^2 x - \dfrac{\partial V_g}{\partial x} + a_{\mathrm{cx}} + d_x \\ \dot{v}_y = -2\omega v_x + \omega^2 y - \dfrac{\partial V_g}{\partial y} + a_{\mathrm{cy}} + d_y \\ \dot{v}_z = -\dfrac{\partial V_g}{\partial z} + a_{\mathrm{cz}} + d_z \end{cases} \tag{2.49}$$

As the planetary landing case, the mass equation is linked to the control acceleration that corresponds to the thrust-to-mass ratio:

$$\mathbf{a}_{\mathrm{c}} = \mathbf{T}/m = \mathbf{P} \tag{2.50}$$

The mass versus time trend is then evaluated with the same solution of Equation (2.6).

2.3.2 Parametric Trajectory Formulation

From Equations. (2.49) and (2.47), the initial derivative of the velocity depends on the initial control acceleration, which is determined by the initial thrust vector $\mathbf{T}_0$ (disturbances are not taken into account in the guidance algorithm). In order to minimize the number of free parameters of the polynomial guidance the initial thrust vector is constrained on the plane defined by $\mathbf{r}_0$ and $\mathbf{r}_f$, as shown in Fig. 2.6.

First a local frame defined by the direction cosines matrix A_0 is defined

$$\mathrm{A}_0 = [\hat{\mathbf{x}}_0 \quad \hat{\mathbf{y}}_0 \quad \hat{\mathbf{z}}_0]^T \tag{2.51}$$

where

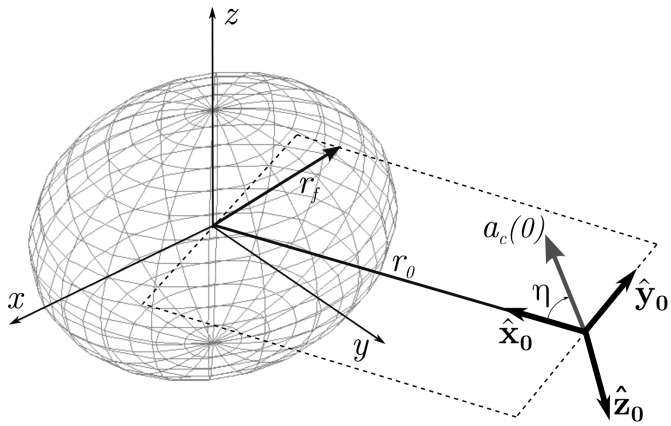


Figure 2.6: *Initial control acceleration.*

$$\hat{\mathbf{x}}_0 = -\frac{\mathbf{r}_0}{\|\mathbf{r}_0\|} \tag{2.52}$$

$$\hat{\mathbf{z}}_0 = \frac{\mathbf{r}_\mathrm{f} \times \mathbf{r}_0}{\|\mathbf{r}_\mathrm{f} \times \mathbf{r}_0\|} \tag{2.53}$$

$$\hat{\mathbf{y}}_0 = \hat{\mathbf{z}}_0 \times \hat{\mathbf{x}}_0 \tag{2.54}$$

This local frame is aligned with the spacecraft-asteroid direction, and its xy plane contains $\mathbf{r}_0$ and $\mathbf{r}_\mathrm{f}$. By defining a second matrix describing a rotation η_0 around $\hat{\mathbf{z}}_0$

$$\mathrm{A}_{\eta_0} = \begin{bmatrix} \cos\eta_0 & \sin\eta_0 & 0 \\ -\sin\eta_0 & \cos\eta_0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.55}$$

The initial thrust vector $\mathbf{T}_0$ can be expressed as function of initial thrust magnitude T_0 and initial angle of thrust η_0 only

$$\mathbf{T}_0 = \mathrm{A}_0^T \mathrm{A}_{\eta_0}^T \begin{bmatrix} T_0 & 0 & 0 \end{bmatrix}^T \tag{2.56}$$

The problem is characterized by a set of 15 boundary constraints: 6 on initial states, 3 on initial control acceleration and 6 on the desired final states

$$\begin{cases} \mathbf{r}(t_0) = \mathbf{r}_0 \\ \mathbf{v}(t_0) = \mathbf{v}_0 \\ \dot{\mathbf{v}}(t_0) = \mathbf{f}(T_0, \eta_0) \\ \mathbf{r}(t_\mathrm{f}) = \mathbf{r}_\mathrm{f} \\ \mathbf{v}(t_\mathrm{f}) = \mathbf{v}_\mathrm{f} \end{cases} \tag{2.57}$$

These constraints are satisfied by expressing the acceleration in a polynomial form. The minimum order needed to satisfy boundary constraints is 3. If $t_0 = 0$:

$$\dot{\mathbf{v}}(t) = \dot{\mathbf{v}}_0 + \mathbf{c}_1 t + \mathbf{c}_2 t^2 + \mathbf{c}_3 t^3 \tag{2.58}$$

By integrating Equation (2.58) as needed, and solving for the boundary constraints, a fully defined trajectory can be determined, depending on 3 parameters: time-of-flight t_tof, initial thrust magnitude T_0, and initial angle of thrust η_0. By solving acceleration equations in the system (2.49) for $\mathbf{a}_c$ a complete control acceleration profile is obtained.

2.3.3 Trajectory Constraints

The search space for the time of flight is defined by

$$0 \leq t_\mathrm{tof} \leq \sqrt{\frac{2r_0^2 h_0}{\mu}} \tag{2.59}$$

in which h_0 is the initial altitude over the asteroid, and μ is its gravitational parameter

$$\mu = G\rho\frac{4\pi}{3}abc \tag{2.60}$$

with G the universal gravitational constant, and a, b, c semi-axes of the ellipsoid. The adopted upper bound represents the time that the spacecraft would take to cover a distance equal to h_0 in free fall, if subject to a constant acceleration of gravity equal to $\mathbf{g}(\mathbf{r}_0)$. Indeed, this time is not associated with a plausible trajectory, but it has the proper order of magnitude (2-3 times the optimal t_tof) for an efficient optimum search.

The initial thrust magnitude is bounded to the thrust available on-board:

$$-T_{\max} \leq T_0 \leq T_{\max} \tag{2.61}$$

The initial thrust angle bounds should be large enough to cover every direction in the plane:

$$-\frac{\pi}{2} \leq \eta_0 \leq \frac{\pi}{2} \tag{2.62}$$

During the landing the required thrust magnitude cannot exceed the limit imposed by the engine on board. Since the control action is evaluated in terms of acceleration, the corresponding thrust should depend on the actual spacecraft mass, according to Newton's second law. In order to simplify the evaluation of the constraint and reduce the computational burden, in the actual algorithm the constant value of the initial mass is used. This assumption ensures the respect of the original constraint (since the mass cannot never increase), without being too restrictive, for the expected fuel consumption in low gravity environment is relatively small compared to the spacecraft mass (less than 1 %, see [87] and [88] and the results in Section 2.6):

$$0 \leq \|\mathbf{a}_{\mathrm{c}}(t)\| \leq \frac{T_{\max}}{m_0} \leq \frac{T_{\max}}{m(t)} \tag{2.63}$$

Also in this case, a *Glide-Slope Constraint* is considered. The spacecraft is required to remain in a cone, pointing at the TLS and defined by the maximum slope angle $\delta_{\max}$, as showed in Fig. 2.7.

Due to the small dimension of the target, it is possible that the maneuver starts from a position that does not satisfy this constraint. In this case it is required that the spacecraft remains over a minimum altitude as long as it doesn't enter into the cone. The j-th general constraint on trajectory shape can be represented in the form

$$-\infty \leq \|\mathrm{S}_j\mathbf{r}(t) - \mathbf{b}_j\| + \mathbf{c}_j^T\mathbf{r}(t) + a_j \leq 0 \tag{2.64}$$

where $\mathrm{S}_j \in \mathbb{R}^{3\times 3}$, $\mathbf{b}_j \in \mathbb{R}^3$, $\mathbf{c}_j \in \mathbb{R}^3$ and $a_j \in \mathbb{R}$. In the case of the glide-slope cone, we have

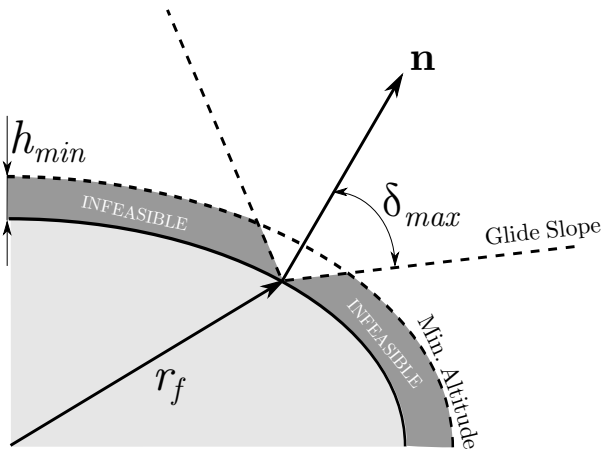


Figure 2.7: *Glide-slope and minimum altitude constraints.*

$$\mathrm{S_g} = \mathrm{I} - \mathbf{n}\mathbf{n}^T \tag{2.65}$$

$$\mathbf{b_g} = \mathrm{S_g}\mathbf{r_f} \tag{2.66}$$

$$\mathbf{c_g}^T = -\tan(\delta_{\max})\mathbf{n}^T \tag{2.67}$$

$$a_{\mathrm{g}} = -\mathbf{c_g}^T\mathbf{r_f}, \tag{2.68}$$

where $\mathbf{n}$ is the unit vector normal to the ground at the TLS. The constraint on minimum altitude $h_{\min}$ for a tri-axial ellipsoid can be expressed with

$$\mathrm{S_h} = -\operatorname{diag}\Big([\beta\gamma, \alpha\gamma, \alpha\beta]\Big) \tag{2.69}$$

$$\mathbf{b_h} = 0 \tag{2.70}$$

$$\mathbf{c_h}^T = 0 \tag{2.71}$$

$$a_{\mathrm{h}} = -\alpha\beta\gamma \tag{2.72}$$

where $\alpha = a + h_{\min}$, $\beta = b + h_{\min}$ and $\gamma = c + h_{\min}$. Glide-slope cone and minimum altitude can be bounded in a single trajectory constraint through the inequality

$$-\infty \leq \min(C_g, C_h) \leq 0 \tag{2.73}$$

where

$$C_{\mathrm{g}} = \|\mathrm{S_g}\mathbf{r}(t) - \mathbf{b}_{\mathrm{g}}\| + \mathbf{c}_{\mathrm{g}}^T\mathbf{r}(t) + a_{\mathrm{g}} \tag{2.74}$$

$$C_{\mathrm{h}} = \|\mathrm{S_h}\mathbf{r}(t)\| + a_{\mathrm{g}} \tag{2.75}$$

Path constraints need to be satisfied at every time instant during the landing. Pseudospectral techniques allow us to evaluate them discretely at CGL points.

Also in this case, the additional constraint on final mass of Equation (2.34) is required:

$$m_{\mathrm{dry}} \leq m(t_{\mathrm{f}}) \leq m_0 \tag{2.76}$$

2.3.4 Optimization Problem

The generic optimization problem (2.2) for asteroidal landing is now expressed as:

$$\underset{T_0, t_{\mathrm{tof}}, \eta_0}{\arg\min}\, f(T_0, t_{\mathrm{tof}}, \eta_0), \quad f(T_0, t_{\mathrm{tof}}, \eta_0) = -m(t_{\mathrm{f}}) \tag{2.77}$$

in the domain defined by the inequalities:

$$0 \leq t_{\mathrm{tof}} \leq \sqrt{\frac{2r_0^2 h_0}{\mu}} \tag{2.78}$$

$$-T_{\max} \leq T_0 \leq T_{\max} \tag{2.79}$$

$$-\frac{\pi}{2} \leq \eta_0 \leq \frac{\pi}{2} \tag{2.80}$$

subject to constraints:

$$0 \leq \|\mathbf{a}_{\mathrm{c}}(t)\| \leq \frac{T_{\max}}{m_0} \tag{2.81}$$

$$-\infty \leq \min(C_g, C_h) \leq 0 \tag{2.82}$$

$$m_{\mathrm{dry}} \leq m(t_{\mathrm{f}}) \leq m_0 \tag{2.83}$$

As in the case stated in Sec. 2.2.4 the problem can be solved with different NLP solvers which strongly affect the actual performances of the system. A detailed discussion about optimizers is expounded in the next section.

2.4 Optimization Algorithms

In the selection and development of the optimization algorithms to solve the guidance problem, three criteria have been followed:

- *Convergence toward feasibility.* The optimizer is required to find at least a feasible solution, if one exists.
- *Computation speed.* The optimization should be fast, to have a system compatible with on-board performances.
- *Quality of the solution.* The solution found should be as close as possible to the optimal one.

The first two properties are mandatory: the system must ensure at least a feasible solution (if it exists) to guarantee the safety of the spacecraft. At the same time, an algorithm unable to run in a timespan of the same magnitude of the system control update frequency, with the limited computational resources on board a spacecraft, would be of limited usefulness in prospect of a practical implementation. On the other hand, the quality of the solution found can be seen as an index of the performance obtained by the system.

The convergence of the optimization toward a correct solution depends also on the proper formulation of the problem. Objective and constraint functions can be written in different forms: a good choice minimizes the number of local minima and leads to a smooth, well scaled problem, easier to be optimized. To assess the effectiveness of the proposed formulation, the system was initially tested with a solver as simple as possible. A compass search algorithm, modified to handle also nonlinear constraints, has been selected for this task. Then, a more accurate algorithm, based on Differential Algebra, has been developed. The capability to model entire functions as well as real numbers as DA variables makes these techniques very effective when applied to optimization, giving implicitly additional information about the sensitivity of the objective and constraint functions to the variation of the optimization variables.

2.4.1 Modified Compass Search

In the context of autonomous adaptive guidance, fast computation must be privileged, in prospect of a real-time implementation for on-board hardware. Derivative-free optimization methods are attractive, because they don't require any differentiation of the cost function, treating it as a "black-box".

As first attempt, a *Compass Search Method* has been adopted. The algorithm is not meant to be actually used in flight, but it has been chosen to assess the effectiveness of the proposed formulation even with a basic optimization algorithm. Since this method is suitable only for unconstrained problems, some modifications have been introduced to handle also non linear constraints. Only the modifications applied to constraints handling are here described. For a detailed description of the classical compass search method, see [98].

First, the optimization variables are normalized, to give them the same relative weight in the optimization:

$$\tilde{\mathbf{x}} = \frac{\mathbf{x} - \mathbf{x}_L}{\mathbf{x}_U - \mathbf{x}_L} \quad \Leftrightarrow \quad \mathbf{x} = \tilde{\mathbf{x}}(\mathbf{x}_U - \mathbf{x}_L) + \mathbf{x}_L \tag{2.84}$$

Normalized optimization variables can vary between 0 and 1. Then, a feasibility function $F(\tilde{\mathbf{x}})$ is created, defined as

$$\Phi(\tilde{\mathbf{x}}) = \sum_{j=0}^{N_C} \frac{1}{w_{\mathrm{F}j}} \max(0, g_j) \tag{2.85}$$

where g_j are the components of a generalized constraints vector $\mathbf{g}(\tilde{\mathbf{x}})$, and $\mathbf{w}_\mathrm{F}$ is a vector of weights, that normalize different constraints that can have different orders of magnitude. In the planetary landing formulation, the constraints are already in generalized form, and a vector of unitary weights is adopted. In the case of asteroid landing, instead, constraints are not normalized: this operation can be handled also during the optimization. Following the same notation of Equation 2.2, the vectors of generalized constraints and weights are computed as follows:

$$\mathbf{g}(\tilde{\mathbf{x}}) = \begin{bmatrix} \mathbf{c}_L - \mathbf{c}(\tilde{\mathbf{x}}) \\ \mathbf{c}(\tilde{\mathbf{x}}) - \mathbf{c}_U \\ 0 - \tilde{\mathbf{x}} \\ \tilde{\mathbf{x}} - 1 \end{bmatrix}, \quad \mathbf{w}_\mathrm{F} = \begin{bmatrix} \mathbf{c}_U - \mathbf{c}_L \\ \mathbf{c}_U - \mathbf{c}_L \\ \mathbf{x}_U - \mathbf{x}_L \\ \mathbf{x}_U - \mathbf{x}_L \end{bmatrix} \tag{2.86}$$

Note that the glide-slope lower bound of Equation (2.64), and consequently the corresponding weight, is infinite. An improper constraint evaluation is avoided by setting this weight to a value with the correct order of magnitude: in this case, $h_{\min}$ is adopted. A feasible set of optimization variables $\tilde{\mathbf{x}}$ corresponds to a null value of the feasibility function. On the contrary, in case of infeasibility, $\Phi(\tilde{\mathbf{x}}) > 0$.

The optimization algorithm operates in two phases. Firstly, an unconstrained compass search on the function $\Phi(\tilde{\mathbf{x}})$ is performed. The search is stopped when a feasible point is found ($\Phi(\tilde{\mathbf{x}}) = 0$), or when the iteration limit is reached. In this case, the problem is classified as infeasible. If the fist step is successful the algorithm keeps solving for the optimum through an unconstrained search on the modified cost function $f(\tilde{\mathbf{x}})$, defined as

$$f(\tilde{\mathbf{x}}) = -m(\tau_{\mathrm{f}}) + \xi \operatorname{sgn}\Big(F(\tilde{\mathbf{x}})\Big) \tag{2.87}$$

where $-m(\tau_{\mathrm{f}})$ is the original cost function of the problem (2.2), and ξ is a number certainly greater than the maximum value that the cost function can assume. From here onward, this optimization method will be identified as Modified Compass Search (MCS).

2.4.2 Differential Algebra Optimization

DA techniques were devised to attempt solving analytical problems through an algebraic approach [99]. Historically, the treatment of functions in numerics has been based on the treatment of numbers, and the classical numerical algorithms are based on the mere evaluation of functions at specific points. DA techniques rely on the observation that it is possible to extract more information on a function rather than its mere values. The basic idea is to bring the treatment of functions and the operations on them to computer environment in a similar manner as the treatment of real numbers.

Referring to Figure 2.8, consider two real numbers a and b. Their transformation into the floating point representation, $\overline{a}$ and $\overline{b}$ respectively, is performed to operate on them in a computer environment. Then, given any operation $*$ in the set of real numbers, an adjoint operation $\circledast$ is defined in the set of Floating Point (FP) numbers so that the diagram in Figure 2.8 commutes (the diagram commutes approximately in practice due to truncation errors). Consequently,

$$\begin{tikzcd} a, b \in R \arrow[r, "\mathcal{T}"] \arrow[d, "*"'] & \overline{a}, \overline{b} \in FP \arrow[d, "\circledast"] \\ a * b \arrow[r, "\mathcal{T}"'] & \overline{a} \circledast \overline{b} \end{tikzcd} \qquad \begin{tikzcd} f, g \arrow[r, "\mathcal{P}"] \arrow[d, "*"'] & F, G \arrow[d, "\circledast"] \\ f * g \arrow[r, "\mathcal{P}"'] & F \circledast G \end{tikzcd}$$

Figure 2.8: *Analogy between the floating point representation of real numbers in a computer environment (left) and the introduction of the algebra of Taylor polynomials in the differential algebraic framework (right).*

transforming the real numbers a and b into their FP representation and operating on them in the set of FP numbers returns the same result as carrying out the operation in the set of real numbers, and then transforming the achieved result in its FP representation. In a similar way, let us suppose two k differentiable functions f and g in n variables are given. In the framework of differential algebra, the computer operates on them using their k-th order Taylor expansions, F and G respectively. Therefore, the transformation of real numbers in their FP representation is now substituted by the extraction of the k-th order Taylor expansions of f and g. For each operation in the space of k differentiable functions, an adjoint operation in the space of Taylor polynomials is defined so that the corresponding diagram commutes; i.e., extracting the Taylor expansions of f and g and operating on them in the space of Taylor polynomials (labeled as ${}_kD_n$) returns the same result as operating on f and g in the original space and then extracting the Taylor expansion of the resulting function.

High-Order Expansion Solution of an ODE

Differential algebra allows the derivatives of any function f of n variables to be computed up to an arbitrary order k, along with the function evaluation. This has an important consequence when the numerical integration of an Ordinary Differential Equations (ODE) is performed by means of an arbitrary integration scheme. Any integration scheme is based on algebraic operations, involving the evaluation of the ODE right hand side at several integration points. Therefore, carrying out all the evaluations in the DA framework allows differential algebra to compute the arbitrary order expansion

of the flow of a general ODE with respect to the initial condition.

Without loss of generality, consider the scalar Initial Value Problem (IVP)

$$\begin{cases} \dot{x} = f(x,t) \\ x(t_0) = x_0 \end{cases} \tag{2.88}$$

Starting from the DA representation of the initial condition x_0, Differential Algebra allows us to compute the Taylor expansion of the IVP with respect to the initial condition at the final time t_f. Replace the point initial condition x_0 by the DA representative of its identity function up to order k, which is a $(k+1)$-tuple of Taylor coefficients. As only the first two coefficients, corresponding to the constant part and the first derivative respectively, are non zero, the DA variable $[x_0]$ can be written as $x_0 + \delta x_0$, in which x_0 is the reference point for the expansion. If all the operations of the numerical integration scheme are carried out in the framework of Differential Algebra, the solution x_i is approximated, at each fixed time step t_i, as a Taylor expansion in x_0. For the sake of clarity, consider the forward Euler's scheme

$$x_i = x_{i-1} + f(x_{i-1})\Delta t \tag{2.89}$$

and substitute the initial value with the DA identity $[x_0] = x_0 + \delta x_0$. At the first time step we have

$$[x_1] = [x_0] + f([x_0]) \cdot \Delta t \tag{2.90}$$

If the function f is evaluated in the DA framework, the output of the first step, $[x_1]$, is the k-th order Taylor expansion of the solution of the IVP in x_0 at $t = t_1$. Note that, as a result of the DA evaluation of $f([x_0])$, the $(k+1)$-tuple $[x_1]$ may include several non zero coefficients corresponding to high-order terms in δx_0. The previous procedure can be inferred through the subsequent steps. The result of the final step is the k-th order Taylor expansion of the solution in x_0 at the final time t_f. Thus, the solution of the IVP can be approximated, at each time step t_i, as a k-th order Taylor expansion in x_0 in a fixed amount of effort.

From here onward, this result will be expressed as $[x_i] = \mathcal{P}_x(\delta x_0)$, in which the square brackets remind that the output is a DA variable, $\mathcal{P}$ indicates a Taylor map or polynomial, the subscript the quantity

represented by the Taylor expansion, and the δ reminds that the Taylor expansion is function of the variation with respect to the reference values. Note that the expansion of the solution of the IVP can be easily obtained also with respect to any parameter q that appears in the dynamics model. In this case also the parameter has to be initialized as a DA variable, i.e. $[q] = q + \delta q$, and the solution at time t_i is $[x_i] = \mathcal{P}_x(\delta x_0, \delta q)$. The conversion of standard integration schemes to their DA counterparts is straightforward for explicit solvers, substituting operations between real numbers with those on DA objects. In addition, whenever the integration scheme involves step size control, an appropriate measure of the accuracy of the Taylor expansion of the flow needs to be included.

The main advantage of the DA-based approach is that there is no need to write and integrate variational equations in order to obtain high order expansions of the flow. This result is basically obtained by the substitution of operations between real numbers with those on DA numbers, and therefore the method is ODE independent. The availability of the resulting Taylor maps is exploited in this work to implement a fast and efficient method for non linear constrained optimization.

Optimization Based on Taylor Map Inversion

As in the case of MCS, the search for the optimum is divided in 2 subsequent phases. First, the algorithm searches for a feasible point (*feasibility* phase); once feasibility is achieved, the solution is refined to gain the optimum (*optimality* phase). The feasibility function $\Phi(\mathbf{x})$ is defined as:

$$\Phi(\mathbf{x}) = \sum_i w_i \Big(g_i(\mathbf{x}) \Big)^2 \tag{2.91}$$

where the weight w_i corresponds to the Heaviside step function of the i-th constraint:

$$w_i = H\Big(g_i(\mathbf{x}) \Big) \tag{2.92}$$

introduced to consider only active constraints. Given $\Phi(\mathbf{x})$, the feasibility phase goes through the following steps:

1. The constraints are evaluated at the current point $\mathbf{x}_j$, to identify active constraints and weights w_{ij}. The first guess $\mathbf{x}_0$ is

here assumed fixed in a predefined point inside the optimization domain. In a real scenario, the actual lander state at the retargeting epoch can be used as first guess to further reduce the iterations required for the feasibility phase. If the point is feasible, the algorithm passes to the optimality phase.

2. The feasibility function is expanded as a DA variable:

$$\Phi_j = \Phi(\mathbf{x}_j) \quad \longrightarrow \quad [\Phi_j] = \sum_i \mathcal{P}_{w_i g_i(\mathbf{x}_j)}(\delta\mathbf{x}_j) = \mathcal{P}_{\Phi_j}(\delta\mathbf{x}_j) \tag{2.93}$$

where $\mathcal{P}_{w_i g_i(\mathbf{x}_j)}$ denotes the Taylor expansion of the i-th constraint about the reference point $\mathbf{x}_j$ with respect to the optimization variables. To save additional computation time, when a constraint is evaluated in multiple points (e.g. the path constraints at CGL points) only the most violated is included in the computation of $[\Phi_j]$.

3. The Taylor expansion of the gradient of Φ_j with respect to the optimization variables is obtained by derivation of $[\Phi_j]$. Being DA variables polynomials maps, derivation is a straightforward process, available as a common operator in the DA domain. The resulting 2-dimensional DA variable maps the value of the partial derivatives with respect the variation of $\mathbf{x}$:

$$[\nabla\Phi_j] = \mathcal{P}_{\nabla\Phi_j}(\delta\mathbf{x}_j) \tag{2.94}$$

4. The gradient map is inverted: first, the constant part of the gradient (corresponding to the real value of the gradient at the expansion point $\mathbf{x}_j$) is subtracted: the new map obtained describes the variation of the gradient as a function of the variation of the expansion point:

$$[\delta\nabla\Phi_j] = \mathcal{P}_{\nabla\Phi_j}(\delta\mathbf{x}_j) - \mathcal{P}_{\nabla\Phi_j}(\delta\mathbf{x}_j)\Big|_{\delta\mathbf{x}_j=0} = \mathcal{P}_{\delta\nabla\Phi_j}(\delta\mathbf{x}_j) \tag{2.95}$$

The inversion of the map describes the variation of the expansion point as a function of a variation of the partial derivatives [99]:

$$[\delta \mathbf{x}_j] = \mathcal{P}^{-1}_{\delta\nabla\Phi_j}(\delta\nabla\Phi_j) \tag{2.96}$$

The inverted map is evaluated at $-\mathcal{P}_{\nabla\Phi_j}(\delta\mathbf{x}_j)\Big|_{\delta\mathbf{x}_j=0}$ to obtain the correction step to move from $\mathbf{x}_j$ to the point of the domain that cancels the Taylor approximation of the gradient in Equation 2.94. Once evaluated, the results is not more a DA variable, but a vector of two reals:

$$\Delta\mathbf{x}_{\text{step}} = \mathcal{P}^{-1}_{\delta\nabla\Phi_j}\left(-\mathcal{P}_{\nabla\Phi_j}(\delta\mathbf{x}_j)\Big|_{\delta\mathbf{x}_j=0}\right) \tag{2.97}$$

5. The expansion point is updated with the relation:

$$\mathbf{x}_{j+1} = \mathbf{x}_j + \alpha\,\Delta\mathbf{x}_{\text{step}} \tag{2.98}$$

where the correction parameter α depends on the length of the computed step:

$$\alpha = \frac{a}{\|\Delta\mathbf{x}\|} + b \tag{2.99}$$

where a, $b \in \mathbb{R}^+$. This relation allows us to to reduce the step size for unacceptably larger $\delta\mathbf{x}_{\text{step}}$, while imposing at the same time a minimum step size. This correction is necessary to achieve a feasible point: in fact, the minimum of the feasibility function Φ lies on the boundary of the feasible domain. Iterating from the infeasible side of the equation without including α would force the optimization process to get arbitrarily close to the feasibility region without reaching it. Once the expansion point is updated, the algorithm starts a new iteration from step 1. If a predefined maximum number of iterations is reached without achieving feasibility, or the requested step length is zero, the process is interrupted and the requested retargeting is classified as infeasible.

In the optimality phase, the solution computed in the feasibility step is refined towards the optimum. The optimality function corresponds to the original objective function modified as:

$$f(\mathbf{x}) = -m(\tau_{\mathrm{f}}) + \frac{1}{t}\sum_i \log\Big(-g_i(\mathbf{x})\Big) \tag{2.100}$$

where the second term of the summation is a logarithmic barrier that forces the solution to remain in the feasible domain. The optimality phase works as follows:

1. The optimality function is expanded as a DA variable:

$$f_j = f(\mathbf{x}_j) \quad \longrightarrow \quad [f_j] = \mathcal{P}_f(\delta\mathbf{x}_j) \tag{2.101}$$

   The solution of the equation (2.6) is expanded with an explicit Runge-Kutta 4th order scheme along the CGL points, as described in the previous sections. At the first step, the scaling parameter of the logarithmic barrier t is initialized at a certain value $t_0 \in \mathbb{R}^+$. In this case, all the constraints $g_i(\mathbf{x})$ are expanded in the DA function. Again, to save computation time, if the same constraint is evaluated in multiple CGL points, only the most violated is included.

2. The gradient of the optimality map $[\nabla f_j]$ is computed and inverted with the same procedure described in the feasibility step, to obtain the correction vector $\Delta\mathbf{x}_{\mathrm{step}}$. The current point is updated:

$$\mathbf{x}_{j+1} = \mathbf{x}_j + \Delta\mathbf{x}_{\mathrm{step}} \tag{2.102}$$

   as well as the logarithmic barrier scaling factor:

$$t_{j+1} = kt_j \tag{2.103}$$

   with $k > 1$.

3. When operating with real numbers, the logarithmic barrier tends to infinity while approaching the constraints, avoiding to step outside the feasibility region. The DA representation of the logarithm is a polynomial, that generally assumes a finite value on the feasibility boundary (unless the expansion point is exactly on the boundary itself). Thus, the computed $\Delta\mathbf{x}_{\mathrm{step}}$ may turn out to point $\mathbf{x}_j$ towards an infeasible region. To avoid this, the

constraints are evaluated at the new point: if the new point is feasible, the next iteration restarts from the step 1; if not, a golden search is performed along the segment $\mathbf{x}_j - \mathbf{x}_{j+1}$ until feasibility condition is met.

The optimality step ends when the step size goes below a predefined tolerance (and the solution found is flagged as optimal), or when a maximum of iterations is reached. In this last case, the solution is probably suboptimal, nevertheless feasible, and then the algorithm throws a warning.

2.5 Planetary Landing Simulation and Test

The performances of the guidance algorithm are assessed thought MC simulations with variable number of samples M. As representative of a planetary landing application, a lunar landing is chosen as test case. In order to make realistic assumptions on spacecraft architecture, the *ESA Lunar Lander* mission is taken as reference. Originally planned for launch in 2018 and designed for landing near the Moon's south pole, the mission's primary objectives include the demonstration of safe precision landing technology as part of preparations for participation to future human exploration of the Moon [89]. Later, the project was put on hold at the 2012 ESA Ministerial Council, but the technology developed in the context of Lunar Lander phase B1 could be exploited for future cooperations in the area of Lunar Exploration with Russia. The Luna-Resource Lander mission, planned by Roscosmos for 2017, could be a testing platform for European precision landing technology, with the proposed Hazard Detection and Avoidance Experiment and the Visual Absolute/Relative Terrain Navigation Experiment (VNE) [25].

The powered descend is assumed divided in 3 different phases [100]:

- *Main Brake.* In this phase, starting from 15 km of altitude at the perilune of a transfer orbit, the thrust is constant at maximum value, and most of the orbital velocity is dropped. At the end of this phase, the spacecraft begins a pitch maneuver and the hazard detection system starts to work.

- *Approach.* Between 2500 and 1500 m of altitude, the thrust is reduced to get maneuverability. In this phase the thrust is variable and retargeting can be commanded.

Table 2.1: *Lunar landing simulation: lander architecture assumptions.*

Feature	Value	UoM
Wet mass m_{wet}	1500	kg
Wet matrix of inertia	$\text{diag}(1650, 1500, 1500)$	$\text{kg}\,\text{m}^2$
Dry mass m_{dry}	790	kg
Dry matrix of inertia	$\text{diag}(845, 675, 675)$	$\text{kg}\,\text{m}^2$
I_{sp}	325	s
I_{max}	1000	$\text{kg}\,\text{m}^2$
T_{min}	1000	N
T_{max}	2320	N
M_{Cmax}	40	$\text{N}\,\text{m}$

Table 2.2: *Lunar landing nominal case (*$\mathbf{e}$ *denotes the vector of Euler angles).*

Quantity	Value	Units	Quantity	Value	Units
$\mathbf{r}_0$	$[2000, -1062.27, 0]$	m	$\mathbf{r}_{\text{f}}$	$[30, 0, 0]$	m
$\mathbf{v}_0$	$[-35, 30, 0]$	$\text{m}\,\text{s}^{-1}$	$\mathbf{v}_{\text{f}}$	$[-1.5, 0, 0]$	$\text{m}\,\text{s}^{-1}$
$\mathbf{e}_0$	$[-55, 0, 0]$	deg	$\mathbf{e}_{\text{f}}$	$[-90, 0, 0]$	deg
m_0	865	kg			

- *Terminal Descent.* Once reached the vertical onto the TLS, at 30 m altitude, the lander performs a vertical descent at the constant speed of $1.5\,\text{m}\,\text{s}^{-1}$ until touchdown.

HDA tasks take place in the approach phase. Assumptions on lander architecture are summarized in Table 2.1, while the nominal boundary conditions of a typical lunar landing maneuver, used for all the simulation here presented, are synthesized in Table 2.2.

2.5.1 MCS - Algorithm Tuning

The polynomial formulation of the landing trajectory produces inherently exact states at the maneuver end. On the other hand, pseudospectal methods are used to carry out the integral of Equation (2.6), necessary for mass computation. These methods involve the selection of an order of approximation N (the number of discrete points at which the approximated function is evaluated) which affects

both precision and computational speed of the algorithm: higher degree improves the precision of the path evaluation, but slows the computation (by increasing the calculation time of single iterations).

The computational efficiency is assessed with a MC run with $M = 1 \times 10^5$ for values of $N = [10, 15, 20, 25, 30]$. The MCS algorithm is written in Matlab® code (with no speedups like mex functions or any other compiled element). In this simulation only the retargeting is considered, and the subsequent simulation of the diversion maneuver is not included. The retargeting is assumed to be ordered from the nominal landing path, at a random altitude between 500 m and 2000 m, with a random ordered diversion between −2000 m and +2000 m, independently in both downrange and crossrange directions. Figure 2.9 shows that the computation time is very stable, with very low dispersion.

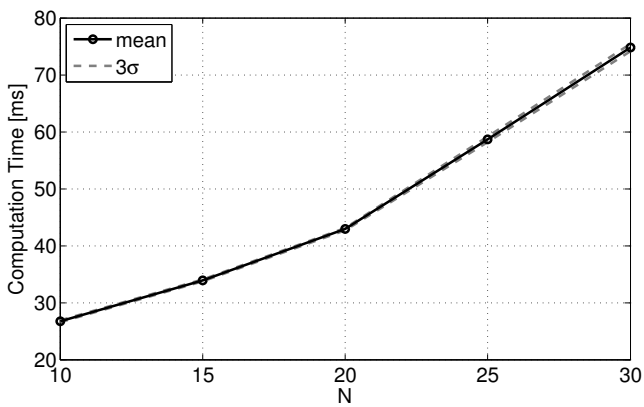


Figure 2.9: *Lunar landing: computational time as a function of approximation order N.*

The estimation of the algorithm precision is slightly different. The control profile calculated by the algorithm, reconstructed with Equations (2.17) and (2.9), is applied to the system (2.10) and integrated trough a traditional Runge-Kutta (4,5) method. The result is compared to the desired one to obtain the error on final position and speed. For each value of N, a set of 10 000 feasible points are considered. The initial altitude is constant at 2000 m, in order to maximize dispersion due to the approximation of the guidance profile reconstruction. The commanded diversion is random between −2000 m and +2000 m in both the horizontal directions. Resulting

errors for position at landing are shown in Fig. 2.10. The polynomial approximation imposes on the trajectory a smooth profile, on which the pseudospectral approximation is very effective. The error is small, and it can considered negligible from $N = 20$ onward. Then, 20 has been taken as nominal value for N.

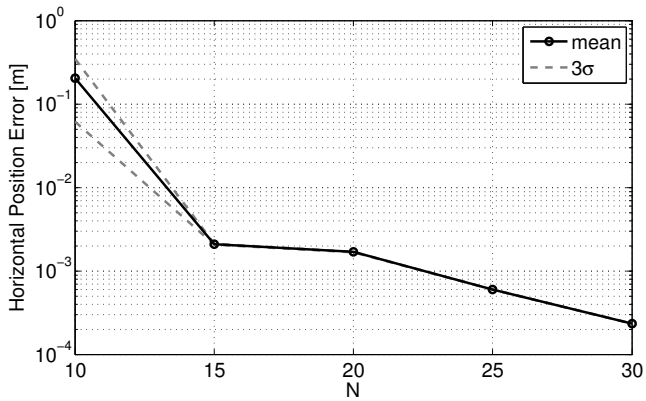


Figure 2.10: *Lunar landing: position error at landing as a function of approximation order N.*

2.5.2 MCS - Objective and Constraints Functions

Planetary landing maneuver requires the optimization of only two parameters: is then possible to visualize graphically the objective and the constraint functions. Due to the polynomial formulation, the objective function maintains a very smooth shape, efficient to be handled by NLP solvers. Figure 2.11 reports a typical example. The initial conditions are the same reported in Table 2.2, except for the initial position that is $\mathbf{r}_0 = [2000 \;\; -562.3 \;\; 1000]^T$ m. The function is mainly dependent on time-of-flight, while the dependency on the initial thrust magnitude is limited.

In Figure 2.12 the different infeasible regions associated to each nonlinear constraint are shown, and the solution found is compared to the actual global optimum (computed solving the optimization problem with the nonlinear programming solver SNOPT). Note that the absence of local minima together with a compact, although non-convex, feasibility region produces an easy-to-solve optimization problem.

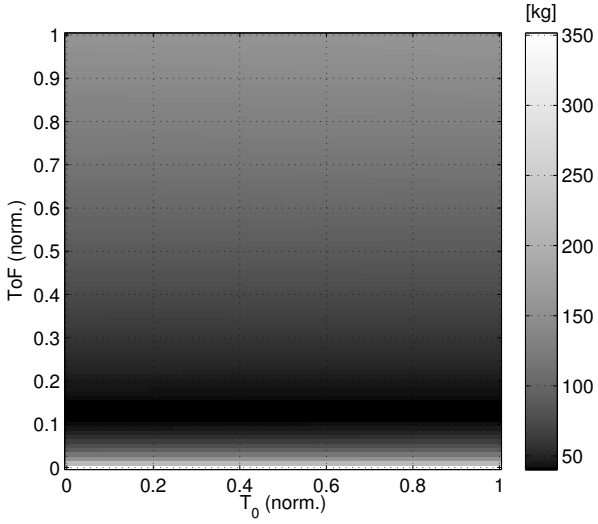


Figure 2.11: *Lunar landing: objective function example.*

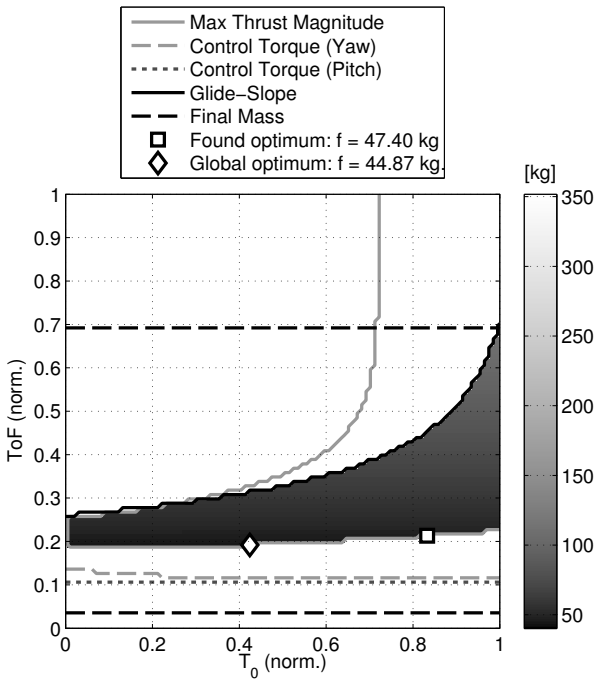
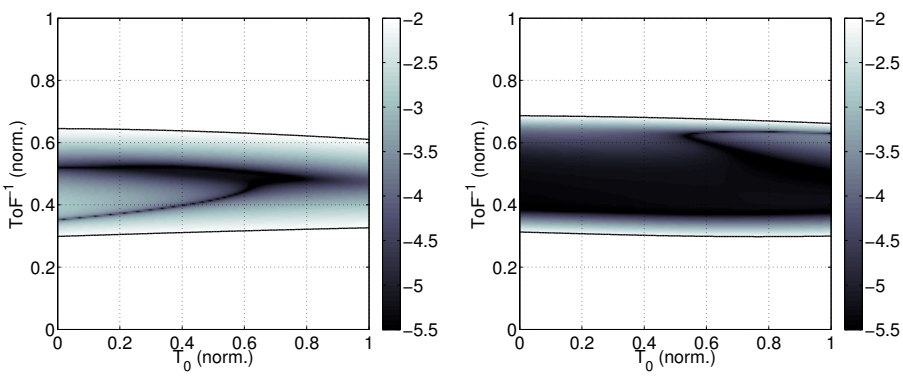


Figure 2.12: *Lunar landing: nonlinear constraints and solution found example, MCS optimization. The solution found is* 5.64 % *far than global optimum.*

2.5.3 DA Optimization - Algorithm Tuning

In the DA optimization process, the solution of the Equation (2.6) is obtained by propagating the initial system states with an explicit Runge-Kutta (4) scheme. Pseudospectal differentiation is exploited only to evaluate the angular acceleration in the constraint (2.27). The consequence is that the choice of the pseudospectral differentiation order has a softer impact in the precision of the results with respect the case of the MCS algorithm. In all the simulations here presented with the DA optimization, $N = 15$ is adopted.

On the other hand, the selection of the proper DA expansion order is crucial, trading between precision and computational performances. Some preliminary tests on the DA representation of the objective function were carried out to identify the most suitable value. The solution evaluated numerically on a grid spanning the entire optimization domain was compared to the solution obtained with the evaluation of the Taylor polynomials. Expansion orders from 2 to 15 were tested, with different expansion points and maneuver cases. Once a predefined relative accuracy threshold of 1% is selected, the *effective area* is defined as the part of the domain in which the error between the DA expansion and the numerical evaluation is within the threshold.



(a) *DA order 2* **(b)** *DA order 15*

Figure 2.13: *Effective area of the DA representation. The inside of the area is colored following the log_{10} of the error with respect to the real number computation.*

It is observed that increasing the expansion order leads to a more accurate representation of the objective function around the

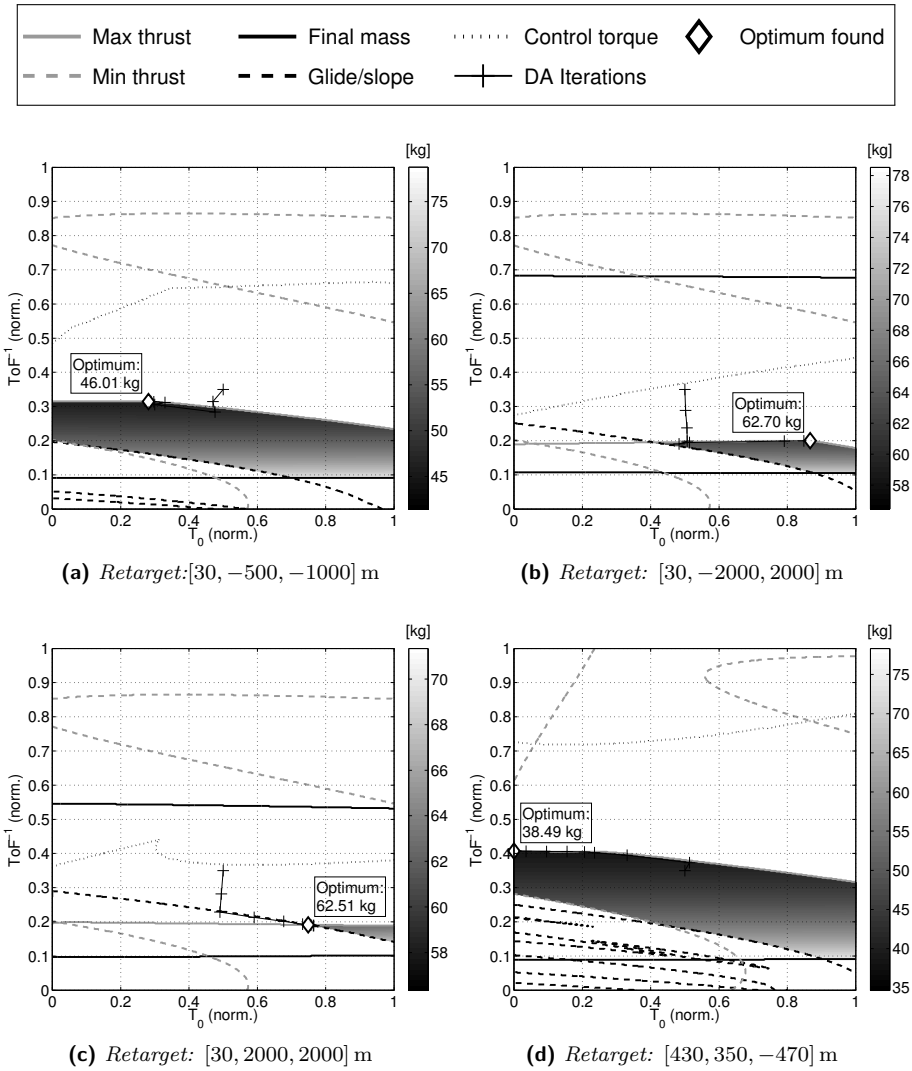
expansion point (and that proves the correct implementation of the DA computation), but without a significant increment of the size of the effective area (see Figure 2.13, in which a comparison between order 2 and 15 is presented). This is due to the peculiar shape of the objective function, that has a general simple shape but with subtle changes of slope that makes it difficult to be represented globally. In an optimization context, it is more efficient to quickly catch the global features of the objective function, avoiding local minima, and then to refine the solution locally toward the optimum: for all the simulations here presented, the lowest possible order (i.e. order 2) is adopted. In practice, all the simulations here presented that involve the DA optimizer are written in C++ code. Differential Algebra is implemented with the Dinamica DACE library [101].

2.5.4 DA Optimization - Objective and Constraints Functions

In Figure 2.14 the optimization process for 4 retargeting cases is shown in detail. Each image represents the optimization domain, with constraints boundaries and the objective function. The optimization steps are depicted in black. The initial conditions are always the same, of Table 2.2. In each case, the final target position vector $\mathbf{r}_\mathrm{f}$ is changed. The DA optimization algorithm efficiently detects active constraints and reaches the true optimum solution, even in cases of small feasible area and multiple active constraints, such as the case of Figure 2.14c.

2.5.5 MCS vs DA Optimization Comparison

A MC simulation is exploited also to assess the algorithm performances in terms of attainable landing area and fuel consumption. Starting from the initial conditions summarized in Table 2.2, a series of 1×10^5 random diversions between $\pm 4000\,\mathrm{m}$ along both the horizontal axes is ordered at an altitude of 2000 m from a nominal trajectory. The attainable landing area can be obtained by correlating optimization results together with the coordinates of the TLSs. The same simulation is run with both the MCS and the DA optimization algorithms: the results are shown in Figure 2.15. Only the points classified as feasible are shown, colored following the value of the objective function of the solution found. The system is able to compute a feasible landing path in an approximately circular landing



(a) *Retarget:*$[30, -500, -1000]$ m

(b) *Retarget:* $[30, -2000, 2000]$ m

(c) *Retarget:* $[30, 2000, 2000]$ m

(d) *Retarget:* $[430, 350, -470]$ m

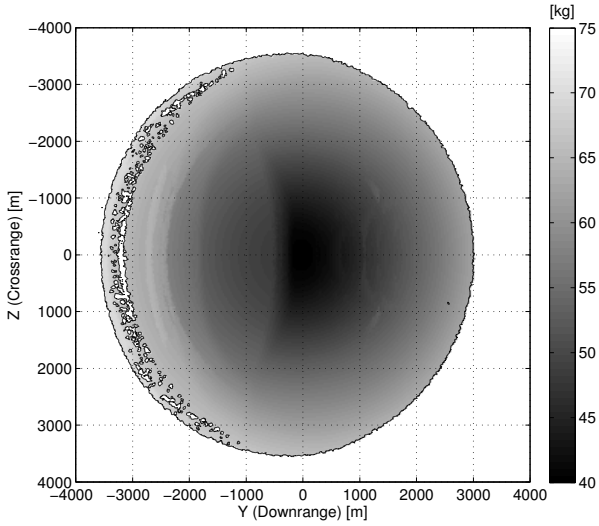
Figure 2.14: *DA Single optimization cases. The algorithm is effective in tracking the boundaries of the feasible area and to efficiently detect the activation of different constraints at each iteration.*

area of radius larger than 2500 m centered at the nominal landing site (at the origin of the figure), a performance better than what is required for similar scenarios [100, 102].

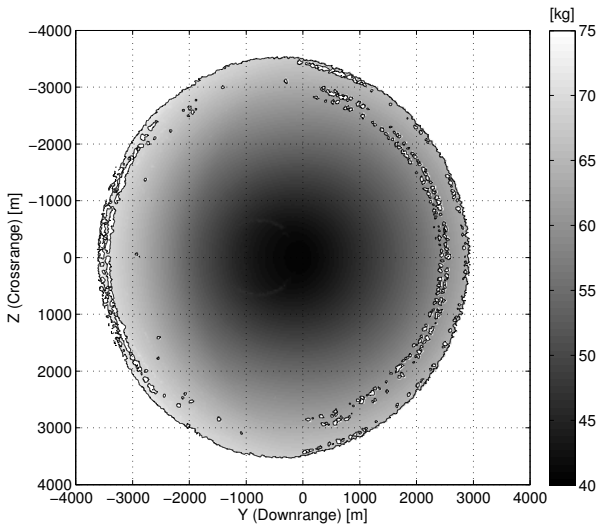
While the attainable area is practically the same with both the optimizers, the quality of the solutions found is different. In the case of the MCS algorithm, the fuel consumption presents sudden variations, especially in cases in which the lander is required to perform an additional brake along the downrange direction. On the contrary, the solution found with the DA optimization is smooth over the entire attainable area. This is an indicator of the fact that the MCS is not always able to get the optimum, and stops the algorithm at a sub-optimal point. Instead, the DA optimization gets always the best possible trajectory. This phenomenon was already visible comparing the optimization processes in Figures 2.12 and 2.14, and it is even more evident with a comparison between the performances of the semi-analytical guidance with the true optimal solution.

In fact, the adopted polynomial form actually limits the shape that the trajectory can assume. Then, the optimal parameterized solution generally differs with respect to the true optimal, which is known to have a bang-bang solution [81]. In order to estimate the distance from the optimum, the proposed algorithm is compared with an open-loop numerical solution, computed through a pseudospectral collocation method (using Tomlab/PROPT® optimization software). Figure 2.16 refers to the same case adopted in Figure 2.11 and Fig. 2.12. The solution found by the semi-analytical guidance optimized with the MCS method is labeled "SAGuid (MCS)"; also the true optimal polynomial solution is included (labeled "SAGuid (optimal)"), computed by optimizing the semi-analytical guidance with SNOPT. Thrust, attitude, and mass profiles are showed. Actually, the solution found with the DA optimization algorithm is not reported in the figure; in all the simulations carried out, the largest difference between the solution found with SNOPT, and the solution found with the DA optimization is less than 0.02 %,

The correlation between the time-of-flight and the optimality of the solution, together with the discontinuous structure of the true optimal thrust profile are clearly visible. Also, it can be seen as polynomial solutions are approximations of the true optimal. In this specific case, the solution found with MCS is 11 % higher than the true theoretical optimum, while the distance reduces to only 2.5 %



(a) *MCS*



(b) *DA Optimization*

Figure 2.15: *Lunar landing: Attainable area and fuel consumption comparison. The quality of the solution found by DA optimization is clearly better than the one found by the MCS algorithm.*

for the polynomial optimum solution found by the DA optimization and SNOPT.

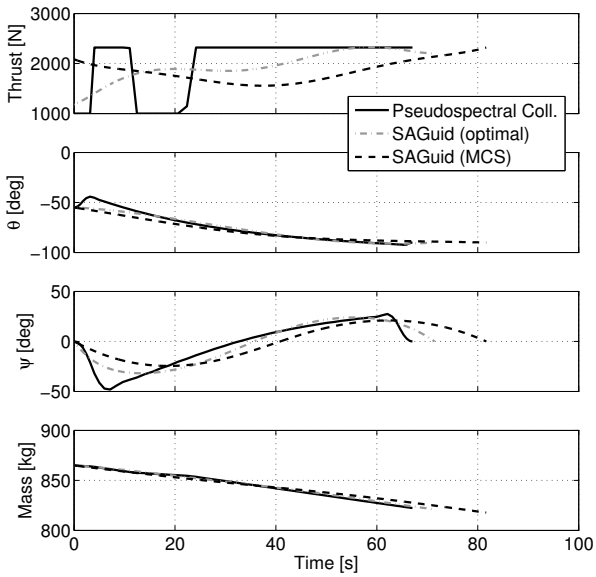


Figure 2.16: *Lunar landing: comparison to true optimal solution example, retargeting maneuver. Polynomial solutions approximate the true optimal. The propellant consumption found by MCS is 11% higher than the solution computed by pseudospectral collocation; the polynomial optimum found by both the DA and SNOPT differs by only* 2.5 %.

In some cases, also the MCS is able to get the true polynomial optimal solution: in Figure 2.17 the same optimality comparison for a nominal approach phase (see Table 2.2) is shown. In this case the MCS, the DA optimizer, and SNOPT find the same solution, that differs by only 3.97 % with respect to the theoretical optimum.

The simulation of Figure 2.15b was exploited also to obtain an estimation of the computation time for the DA optimizer. All the simulations were tested on a Intel® Core™ i7-2630QM CPU at 2 GHz of frequency. Figure 2.18 reports the results obtained: the histogram at the top reports the time dispersion for feasible cases. The mean computation time is 25.23 ms with a standard deviation of 7.16 ms. This implies a 3σ computation time below 46.71 ms. In case of an infeasible retarget is ordered (Figure 2.18 bottom), the systems performs a fixed number of iterations until either the maximum iteration number (fixed at 30 for the simulations here presented),

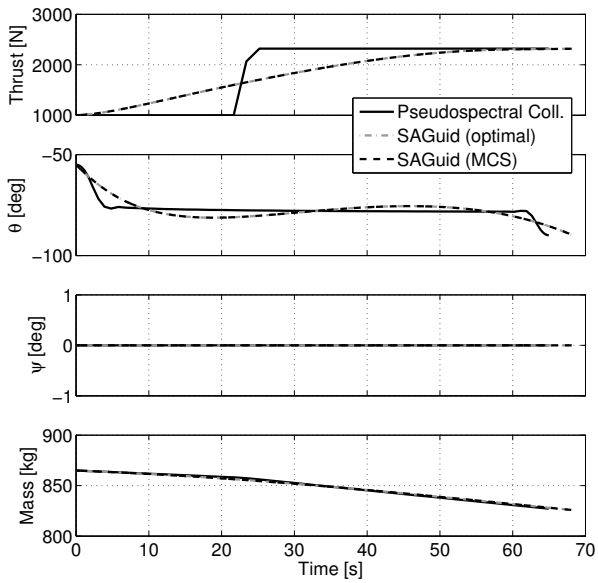


Figure 2.17: *Lunar landing: comparison to true optimal solution example, nominal maneuver. The propellant consumption found by MCS and DA optimization corresponds to the true optimal parameterized solution, 3.97% higher than the theoretical optimum.*

or a stationary point (with null step length) is reached. In most of the cases, the algorithm stops due to maximum iteration limit, and then the computation time is quite constant, with a mean equal to 33.94 ms. The quite low dispersion (STD = 4.20 ms) is mainly due to the cases in which a stationary point in the feasibility function is reached. Comparing these results with the performances achieved by the MCS (see Figure 2.9), it could appear that the DA optimizer is even faster. This is not true, being the routine written in C++ language instead of Matlab code. Nevertheless, in both feasible and infeasible region, the DA algorithm is still very fast. In a real application, the degradation of the performances due to the algorithm implementation on space grade CPUs (much slower than a PC) is expected to be counterbalanced by the wide, further improvements still possible in code optimization. Then, the guidance is expected to be compatible with on-board computation, being the expected control frequency in the order of magnitude of 10 Hz (equivalent to a control update period of 100 ms).

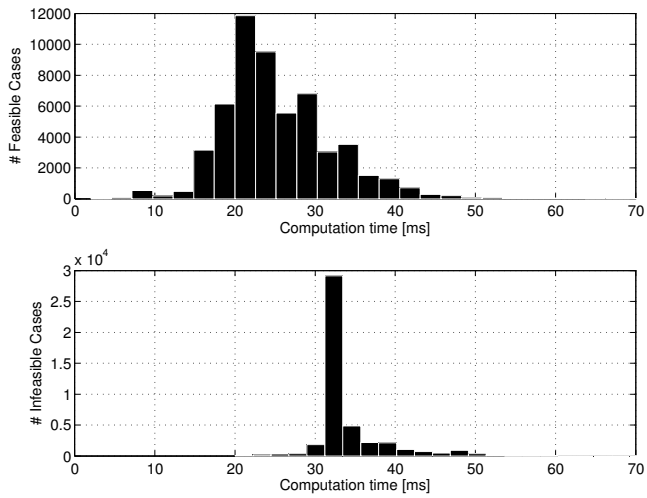


Figure 2.18: *Lunar landing: DA optimization computation time*

2.5.6 Landing Simulation: Nominal Navigation Errors

The guidance algorithm is tested in a 7DoF (three-dimensional rototranslation with variable mass) retargeting simulator of a lunar landing, realized in Matlab® and Simulink® environment.

In this full simulation, the maneuver is optimized with the MCS algorithm: being the most critical of the considered algorithms, the results here presented can be considered as a lower bound for the system performance.

It is supposed that a nominal trajectory is known, obtained through traditional optimization methods. The simulation covers the approach phase from 2000 m altitude to the beginning of the terminal descent onto the TLS, over a timespan of the order of magnitude of 70 s. In order to test both effectiveness and diversion capability of the algorithm, a MC simulation with $M = 1000$ is adopted. Table 2.3 shows initial and final boundary constraints imposed, together with dispersion added to initial conditions, to include uncertainties at the end of the main brake phase. Larger dispersions are considered for the horizontal components of the initial position. In fact, all the considered dynamics are relative, from the spacecraft with respect to the landing site. During trajectory computation, ordering a retargeting or shifting the initial position of the same magnitude toward the opposite direction are equivalent (actually, shifting the initial

Table 2.3: *Complete lunar landing simulation: initial and final conditions for MC analysis.*

Condition	Nominal value	1σ	UoM
Initial mass m_0	865	± 10	kg
Initial position $\mathbf{r}_0$	$[2000\ \ -1060\ 0]^T$	$\pm[30\ 600\ 600]^T$	m
Initial speed $\mathbf{v}_0$	$[-35\ 30\ 0]^T$	$\pm[0.5\ 0.5\ 0.5]^T$	$\mathrm{m\,s^{-1}}$
Initial pitch angle θ_0	-55	± 5	deg
Initial yaw angle ψ_0	0	± 5	deg
Target position $\mathbf{r}_\mathrm{f}$	$[30\ 0\ 0]^T$	-	m
Target speed $\mathbf{v}_\mathrm{f}$	$[-1.5\ 0\ 0]^T$	-	$\mathrm{m\,s^{-1}}$
Target pitch angle θ_f	-90	-	deg
Target yaw angle ψ_f	0	-	deg

Table 2.4: *Lunar landing simulation: IMU performance properties.*

Property	Value	UoM
Scale factor	1	ppm
Misalignment Error	170	µrad
Bias Error	0.005	deg/h
ARW noise density	0.005	$\mathrm{deg}/\sqrt{\mathrm{h}}$

position is exactly what the guidance algorithm does). Thus, the introduction of larger dispersions in the horizontal components of the initial position allows us also to evaluate retargeting capabilities, including both position uncertainties and random ordered diversions.

To properly test the guidance algorithm, it is applied to a lander model more complex than the one adopted inside the guidance itself. A full 7DoF model is considered for the spacecraft. The lander is considered as a rigid body, with an inertia matrix linearly variable with the mass (within the boundaries described in Table 2.1. Realistic disturbances and navigation errors are added. A disturbance torque is introduced by a 10 mm thrust misalignment from the spacecraft center of mass. Errors in the states passed to the guidance block are included to emulate a real navigation system. Attitude is supposed to be estimated by an Inertial Measurement Unit (IMU), whose performances are summarized in Table 2.4.

The presence of a vision-based navigation system is assumed to estimate position and speed. These systems make use of a radar or laser altimeter to estimate the altitude with which the images taken by cameras are resized to the proper scale. Since altimeters absolute error increases with the distance from the ground, the error in the estimate is modeled as a Gaussian random error with zero mean and standard deviation varying linearly with the altitude. The values adopted as reference are $\pm 25\,\mathrm{m}$ and $\pm 0.4\,\mathrm{m\,s^{-1}}$ (1σ) at 2000 m altitude (they are both assumed to be null at zero altitude). The guidance subsystem recalculates the trajectory every 5 s, to cope with measure dispersion. From the guidance profile, at every update of the control system, target quaternions and angular velocities are computed, and a Proportional Integral Derivative controller is used to calculate theoretical control torques. The attitude is assumed to be controlled by a cluster of chemical thrusters able to supply a constant torque of $\pm 40\,\mathrm{N\,m}$ on each axis. Theoretical control torques are processed by a Pulse Width Pulse Frequency (PWPF) modulator that commands thrusters firings. The considered guidance and control systems update rate is 20 Hz.

Figure 2.19 shows the obtained 3D trajectories. Dispersions in position and velocity, for their horizontal (Fig. 2.20) and vertical (Fig. 2.21) components are reported. Figure 2.22 shows the obtained final attitude distribution. Overall, the system attains a good performance despite of the uncertainties. The dimensions of the obtained 3σ ellipse are comparable to a possible lander footprint [89], giving to the system hazard avoidance capabilities.

2.5.7 Landing Simulation: Sensitivity to Navigation Errors

Additional MC runs are exploited to assess the sensitivity of the system to navigation errors. The same initial and target conditions of Table 2.3 are assumed, except for the value of the navigation errors standard deviation at the maneuver start. A sample number $M = 300$ is adopted. The following values are considered for position and velocity estimation:

- *Pessimistic Case* (PC): Position error: $\pm 45\,\mathrm{m}$; speed error: $\pm 0.4\,\mathrm{m\,s^{-1}}$ (1σ);
- *Optimistic Case* (OC): Position error: $\pm 10\,\mathrm{m}$; speed error: $\pm 0.4\,\mathrm{m\,s^{-1}}$ (1σ);

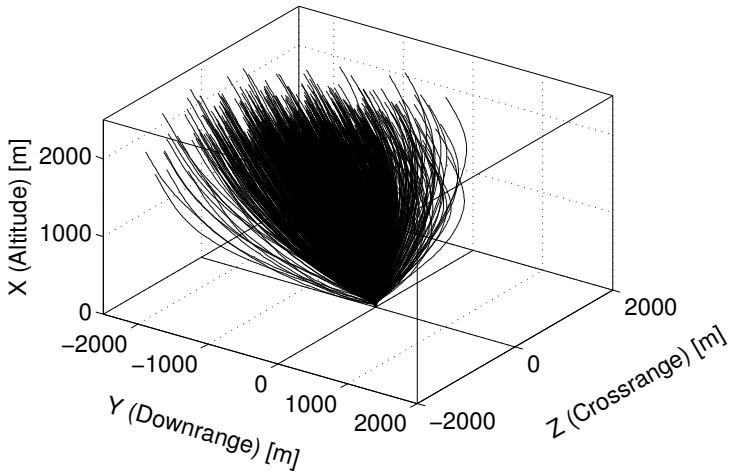
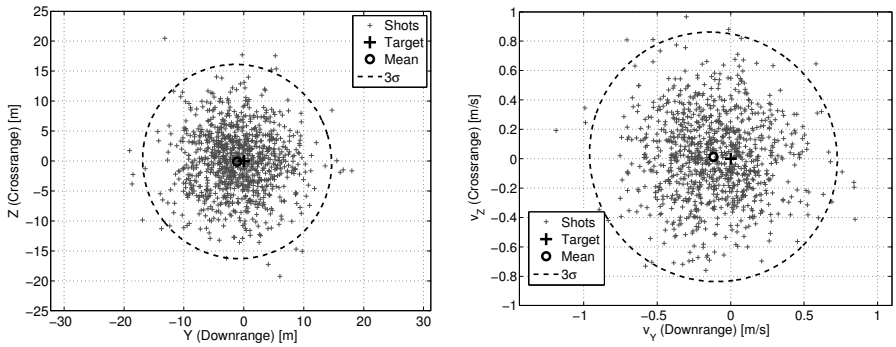


Figure 2.19: *Lunar landing simulation ($M = 1000$): 3D Landing trajectories.*

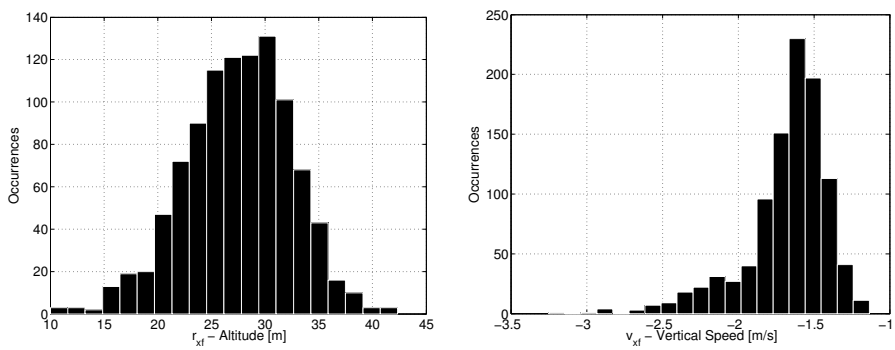
- *Exact Case* (EC): Ideal measures with no navigation errors.

As for the nominal case (presented in the previous section) navigation errors are supposed to decrease linearly with the altitude, and they are assumed to be null at touchdown. The EC is considered to highlight the effect of the navigation system over the final landing accuracy. Figure (2.23) shows a comparison of the results obtained for the 3 cases, including also the *Nominal Case* (NC) presented in section 2.5.6. The obtained 3σ dispersion ellipses for final horizontal position and velocity are presented. It can be seen that dispersion due to control system only is at least one order of magnitude lower than the one due to navigation. This proves that landing precision is mainly affected by navigation errors.



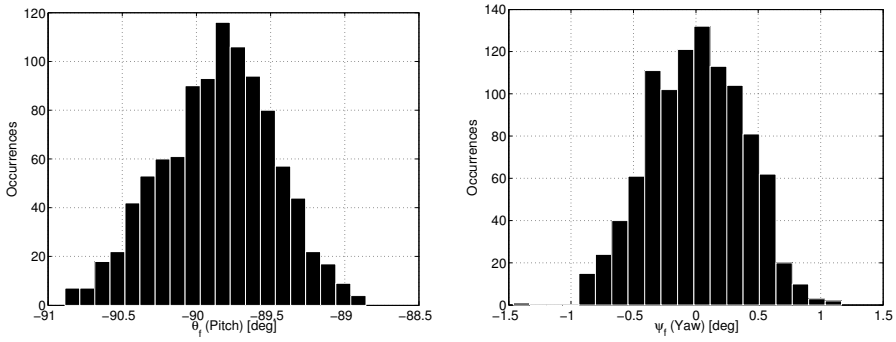
(a) *Position.* **(b)** *Horizontal velocity.*

Figure 2.20: *Lunar landing simulation: dispersion in final position and velocity, horizontal components.*



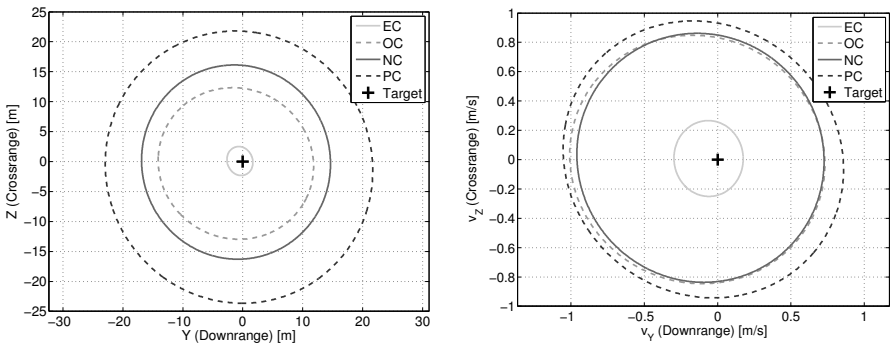
(a) *Final altitude.* **(b)** *Final vertical speed.*

Figure 2.21: *Lunar landing simulation: dispersion in final position and velocity, vertical components.*



(a) *Pitch angle.* **(b)** *Yaw angle.*

Figure 2.22: *Lunar landing simulation: final attitude dispersion.*



(a) *Position.* **(b)** *Velocity.*

Figure 2.23: *Lunar landing MC simulation: navigation errors effect.* 3σ *dispersion ellipses at touchdown for position (a) and velocity (b).*

Table 2.5: *1999 RQ_{36} "Bennu" nominal parameters.*

Feature	Value	UoM
Major semi-axis, a	350	m
Intermediate semi-axis, b	287	m
Minor semi-axis, c	250	m
Density, ρ	1400	$\mathrm{kg\,m^{-3}}$
Rotational rate, ω	4.04×10^{-4}	$\mathrm{rad\,s^{-1}}$

Table 2.6: *NEA landing: MC parameters, initial and target states.*

Condition	Nominal value	1σ	UoM
Initial Position, $\mathbf{r}_0$	$[1500, 0, 0]^T$	$\pm[50, 100, 100]^T$	m
Initial Velocity, $\mathbf{v}_0$	$[0, 0, 0]^T$	$\pm[0.1, 0.1, 0.1]^T$	$\mathrm{m\,s^{-1}}$
Initial Mass, m_0	750	-	kg
Target Position, $\mathbf{r}_\mathrm{f}$	$[0, 290, 0]^T$	-	m
Target Velocity, $\mathbf{v}_\mathrm{f}$	$[0, -0.1, 0]^T$	-	$\mathrm{m\,s^{-1}}$
Specific Impulse, I_sp	315	-	s
Max Available Thrust, T_max	10	-	N
Dry Mass, m_dry	740	-	kg
Asteroid Density, ρ	1400	$\pm 10\%$	$\mathrm{kg\,m^{-3}}$
Asteroid Rotational rate, ω	4.04×10^{-4}	$\pm 10\%$	$\mathrm{rad\,s^{-1}}$

2.6 Asteroid Landing Simulation and Test

A landing on the asteroid 1999 RQ_{36} "Bennu", target of the mission OSIRIS-REx, recently launched by NASA and scheduled to reach its target in 2019 [31, 86], is selected as NEA application test. Table 2.5 summarizes the assumed asteroid nominal parameters.

The case of an equatorial landing is here presented. An equatorial target has the additional challenge of its rotation, that the lander is required to follow. Polar trajectories do not have this challenge. The spacecraft is supposed to start at a near hovering condition; the target state is on the vertical over the selected landing site, at 3 m of altitude, with a vertical speed of $0.1\,\mathrm{m\,s^{-1}}$ toward the ground and a null horizontal speed. Adopted parameters, initial and target states, common to all the simulations here presented, are summarized in Table 2.6. Only the MCS optimization algorithm has been tested for NEA landing simulations.

2.6.1 Algorithm Performance Estimation

The assessment of the computational performance is performed through a MC simulation, with the same parameters of Table 2.6 and $M = 10000$. Figure 2.24 shows that the additional optimization variable causes an increase of the computation time up to one order of magnitude, compared to the planetary landing case. Anyway, due to the relatively large time-of-flight typical of maneuvers in low-gravity environment, it remains compatible with maneuver requirements.

The estimation of attainable landing area and fuel consumption is carried out considering the nominal initial conditions of Table 2.6. The landing site is varied over a regular grid of 1° resolution in both latitude and longitude. From Fig. 2.25 it is shown that the spacecraft can reach any site on the NEA surface, and that the fuel consumption presents an asymmetry in longitude, due to the asteroid rotational rate.

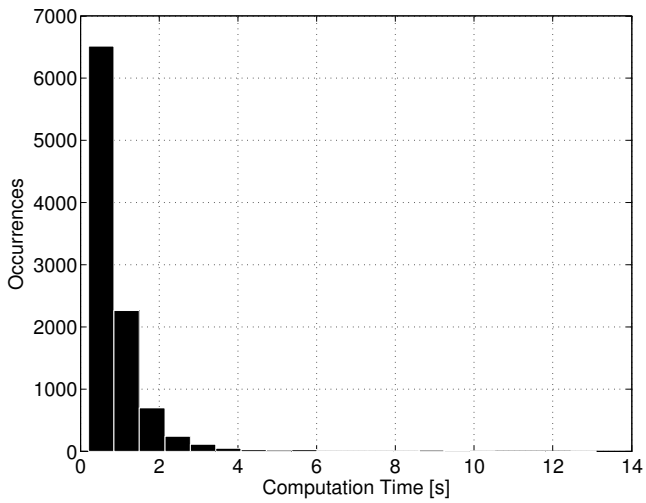


Figure 2.24: *NEA landing: algorithm computation time.*

2.6.2 Landing Simulation: Exact Measures

Due to the weak gravitational acceleration involved, in a NEA landing case the theoretical thrust can assume very low values (also for long times) that could be not attainable by traditional propulsion systems. Thus, it is assumed that the thrust is supplied by the same system

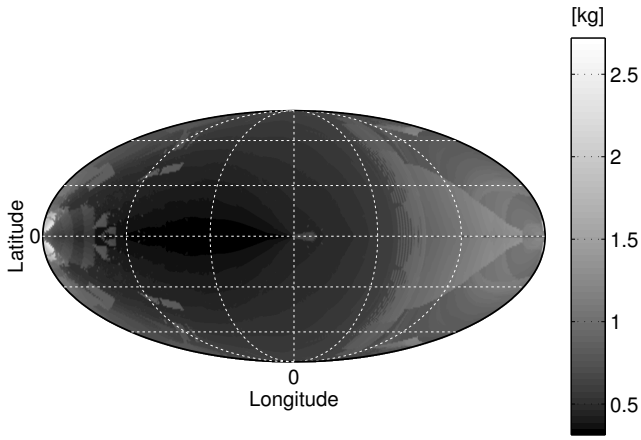


Figure 2.25: *NEA landing: attainable area and fuel consumption.*

of chemical thrusters used by ACS, filtered by a PWPF modulation system.

Sharing of the propulsion system is made possible by the slow dynamics of both attitude and thrust control systems. During the landing maneuver, the spacecraft is simply required to point toward the asteroid center of mass. The actual GNC system architecture is represented in Figure 2.26: the navigation system determines position $\mathbf{r}$, velocity $\mathbf{v}$, attitude quaternions $\mathbf{q}$ and rotational rate vector $\boldsymbol{\omega}$. Attitude control system computes the control torques $\mathbf{M}_c$, while the adaptive guidance system provides the control thrust vector $\mathbf{T}_c$. Their actuation is fused together by PWPF modulation in a unique thruster activation scheme. This configuration presents several advantages:

- The 3 components of the thrust vector can be generated independently, in body axes, leaving the spacecraft free to assume any attitude imposed by vision-based navigation.
- There is no need of additional dedicated devices devoted to low-thrust.
- No additional constraints are imposed over high-trust propulsion system (devoted to large scale orbital control), in terms of thrust throttleability or minimum thrust level.

As a result of this architecture, attitude and propulsion are assumed as independent and the simulation is reduced to 4 degrees

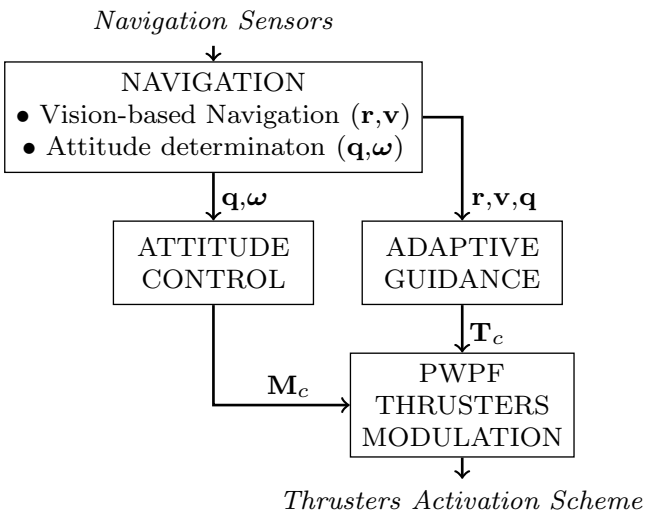


Figure 2.26: *NEA landing: logical schematic of the GNC System.*

of freedom (3 translations and the variable mass). The full landing simulation is realized in Matlab® and Simulink® environment, with MCS as optimization algorithm. As the landing site gets closer, the trajectory is updated by additional runs of the algorithm, performed at 1000, 500, 300, 200 and 150 m from the target. In this way, dispersion due to modulation is compensated. In this article an example of equatorial landing is reported: asteroid and lander data, together with initial conditions, TLS, and their relative dispersions, are the same reported in Tables 2.5 and 2.6. MC simulations with $M = 300$ are run.

As visible from Figures 2.27-2.29, the modulation of the thrust introduces a certain error in the attained position over the landing site. Anyway this error remains into acceptable limits, with an obtained final maximum accuracy of 8 m (3σ) from the target.

2.6.3 Landing Simulation: Navigation Errors

In the GNC system schematic represented in Fig. 2.26, is possible to see how navigation errors influence trajectory calculation. At the time the trajectory is recomputed, errors in position and velocity determination affects directly the obtained path. Moreover, since the thrust profile obtained from the optimization is expressed in asteroid reference frame, a conversion in spacecraft body-fixed frame

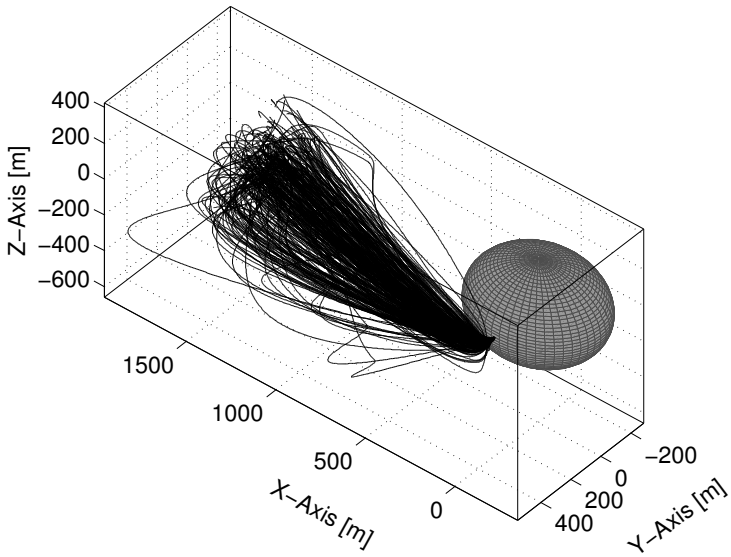
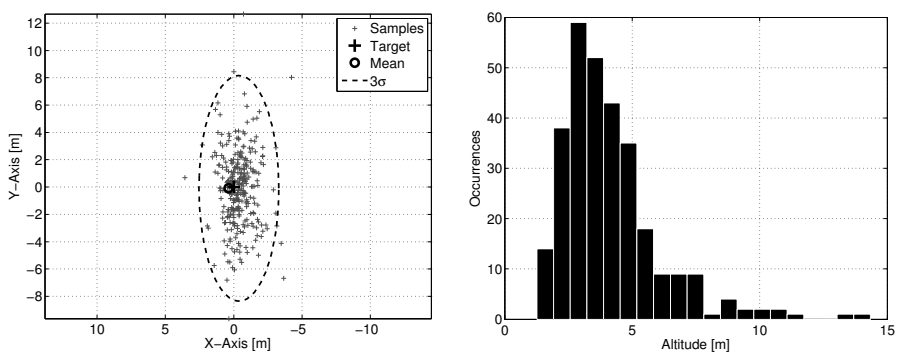
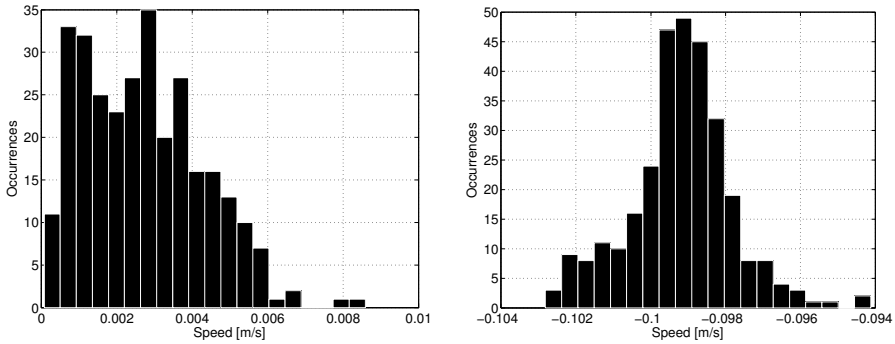


Figure 2.27: *NEA landing MC simulation ($M = 300$), no navigation errors: 3D landing trajectories.*



(a) *Horizontal components.* **(b)** *Altitude.*

Figure 2.28: *NEA landing MC simulation, no navigation errors: final position distribution.*



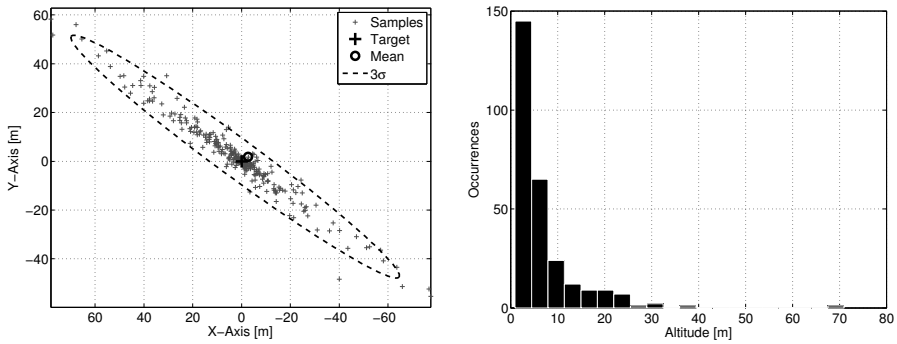
(a) *Horizontal components.* **(b)** *Vertical component.*

Figure 2.29: *NEA landing MC simulation, no navigation errors: final velocity distribution.*

is required at every control timestep to properly command the actuators. Errors in attitude determination affect the direction of the actual thrust, introducing additional errors in attained states at the maneuver's end.

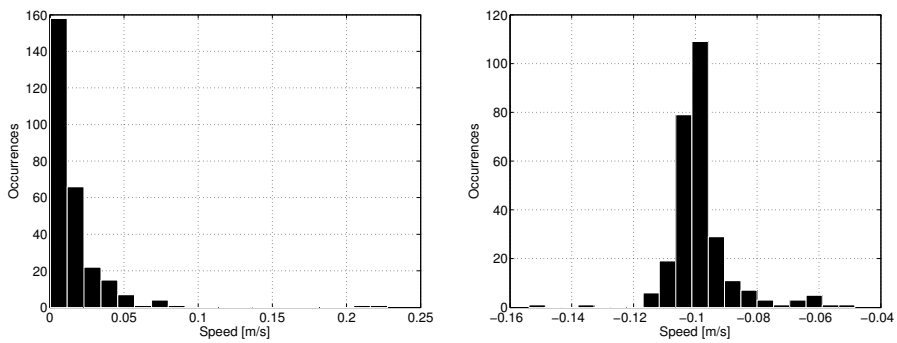
Assuming the presence of a visual-based navigation system, errors in position and velocity are modeled as Gaussian errors, with zero mean and variable standard deviation proportional to the distance between the asteroid and the spacecraft. The values of 25 m and $0.1\,\mathrm{m\,s^{-1}}$, at the reference distance of 2000 m are adopted (they are both considered null on the surface of the asteroid). The presence of a star tracker is considered for attitude determination. The attitude error is considered as Gaussian with a bias (mean) rotation of 5 arcsec and a standard deviation of 3 arcsec around each axis. As in the previous case, dispersion at touchdown is limited by updating the trajectory as the spacecraft gets closer to the target, at the same target distances of the first simulation. An MC run with $M = 300$ is carried out.

Due to the relative long time requested by the maneuver, together with the applied open-loop control, errors in states determination at the retargeting epoch propagate up to potentially unacceptable values, especially for position (while a good precision in velocity is preserved), as shown in Figures 2.30 and 2.31. In particular, the error obtained in final position is almost of the same order of magnitude of the asteroid's size itself, and cannot be accepted.



(a) *Horizontal components.* **(b)** *Altitude.*

Figure 2.30: *NEA landing MC simulation ($M = 300$), navigation errors effect: final position distribution.*



(a) *Horizontal components.* **(b)** *Vertical component.*

Figure 2.31: *NEA landing MC simulation, navigation errors effect: final velocity distribution.*

2.6.4 Landing Simulation: Waypoint Trajectory

As possible method to regain precision at landing, the introduction of a waypoint along the trajectory is investigated. The trajectory computation is split into two concatenate maneuvers: in the first one, the target of the trajectory optimization is not the TLS, but the point 250 m above it, along the local vertical direction. Once this first maneuver is ended, the system performs a second optimization toward the final target. A third MC with $M = 300$ is adopted.

A level of precision of the same order of magnitude of the of the case without navigation errors is recovered, as visible by comparing Fig. 2.32 – 2.34 with the correspondent Fig. 2.27 – 2.29. This result is achieved without a significant impact on propellant consumption as shown in Fig. 2.35.

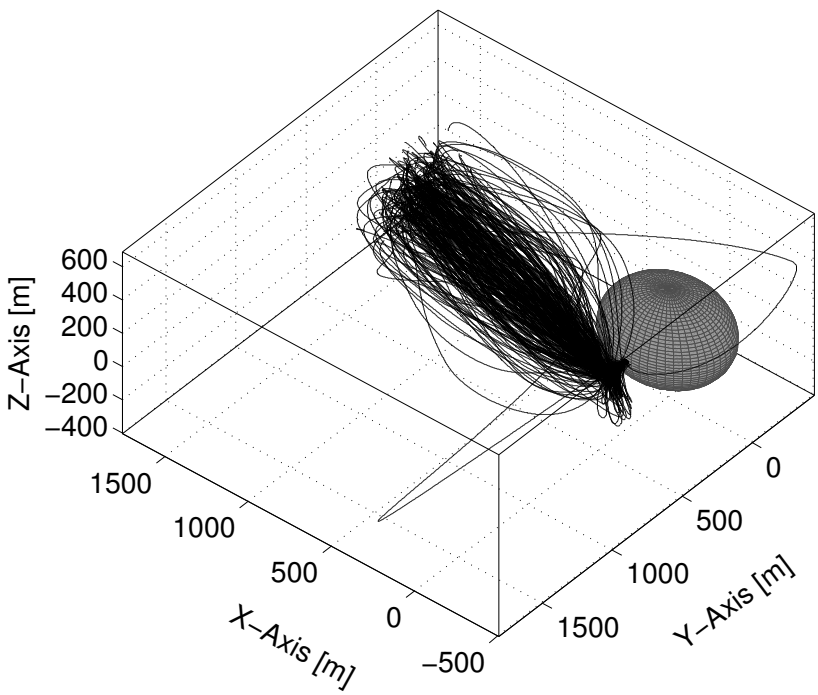
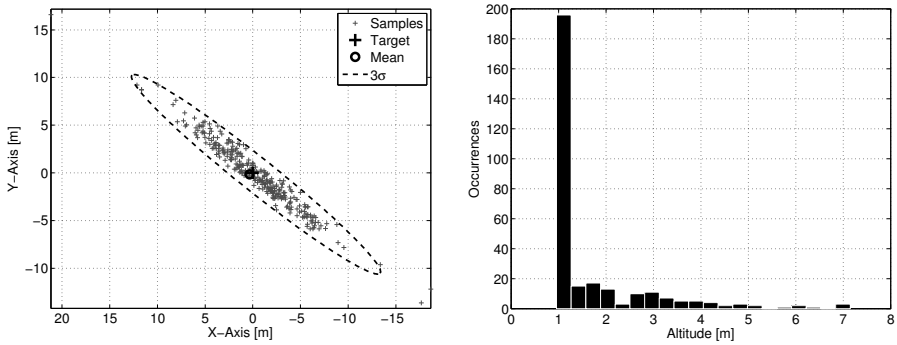
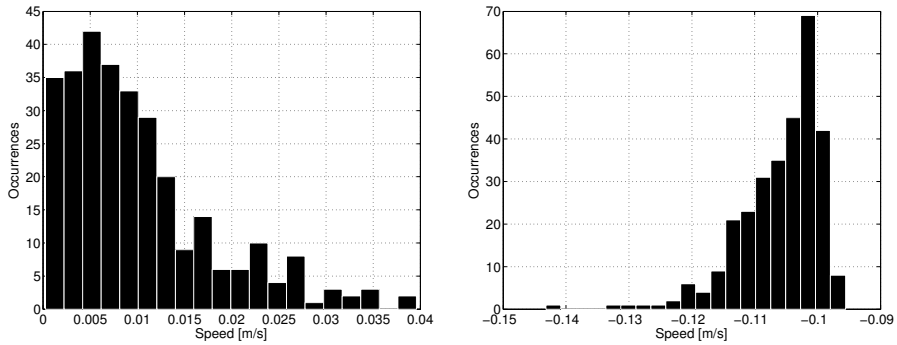


Figure 2.32: *NEA landing MC simulation ($M = 300$), waypoint improved maneuver case: 3D landing trajectories.*



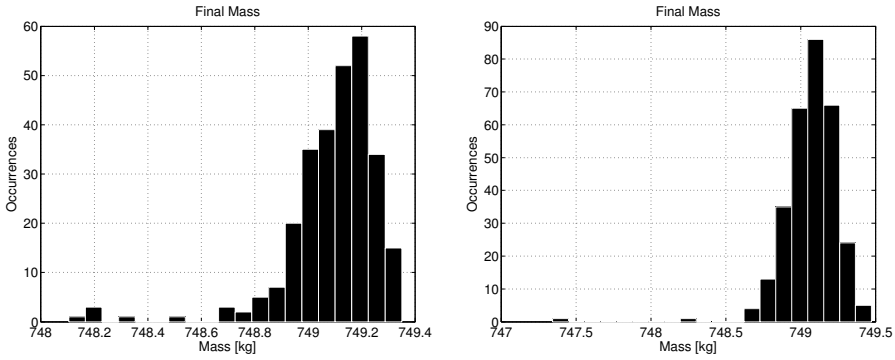
(a) *Horizontal components.* **(b)** *Altitude.*

Figure 2.33: *NEA landing MC simulation, waypoint improved maneuver case: final position distribution.*



(a) *Horizontal components.* **(b)** *Vertical component.*

Figure 2.34: *NEA landing MC simulation, waypoint improved maneuver case: final velocity distribution.*



(a) *Single maneuver case.* **(b)** *Waypoint case.*

Figure 2.35: *NEA landing MC simulation: comparison between single maneuver and waypoint improved trajectory.*

I heard that your sailors have very similar experiences while they traverse your seas and discern some distant island or coast lying on the horizon. The far-off land may have bays, forelands, angles in and out to any number and extent; yet at a distance you see none of these (unless indeed your sun shines bright upon them revealing the projections and retirements by means of light and shade), nothing but a grey unbroken line upon the water.

Edwin A. Abbott, Flatland

3 Hazard Detection

Locate safe landing areas is a very complex task. The actual definition of "safety" itself is largely determined by the architecture of the specific spacecraft, in terms of landing mechanisms (gears, legs, airbags, etc.) and masses distribution, but it also depends on relative factors, like the sensors field of view and the capabilities of the attitude and propulsion subsystems to reach the candidate target from the specific set of states the system has at the time of the retargeting.

Early studies on HDA systems exploited very simple principles: in [103] local variance over an intensity image is considered as a way to estimate surface roughness, together with surface major irregularities detection performed by a scanning ranging laser. Later, the development of more powerful systems and specialized hardware paved the way to the development of more complex and accurate hazard detection methods. In the frame of the ALHAT project, carried out by NASA since 2006, extensive studies have been conducted on the hazard estimation based on a Digital Elevation Model (DEM) obtained by active ranging sensors, such as Doppler LIDAR and flash LIDAR, as shown by [56]. A proposal to include also scientific criteria in the selection process is done by [104] exploiting soft computing techniques. Other methods to reconstruct a DEM of the landing

area through image processing techniques, such as shape from shading [105], stereo-vision [106] and shadow analysis [107] have been investigated.

Four main criteria concur to determine if a landing site can be classified as safe:

- *Sensors field of view.*
  Areas that cannot be analyzed by the sensors system should be classified a priori as unsafe; considering systems based on visual information, areas in shadows are included in this category.

- *Surface roughness.*
  The actual architecture of the lander touchdown system (legs, airbags), determines which are the maximum allowed dimensions of local obstacles that maintain the probability to get damages below tolerable values.

- *Slopes.*
  The maximum slope that the lander can handle without danger of capsizing. This value should be determined together with the maximum surface roughness with a proper margin of safety basing on the actual lander architecture.

- *Size of the safe area.*
  The landing site dimension must be compatible with the lander footprint plus expected uncertainties due to the GNC system.

Plus, also if a target is found safe, it could be impossible to be reached, due to the limited control authority of the spacecraft. Then, also the probability to find a feasible trajectory to the target should be taken into account in the selection process.

A novel hazard detection algorithm, based on ANN is proposed in this work. ANNs appear particularly attractive for their generalization properties: in fact, once trained with proper data, this kind of systems is able to autonomously determine "fading" rules that describe the phenomenon under investigation [108]. This property is very relevant for hazard detection. In fact, during algorithms development, it is impossible to consider in advance all the types of terrain morphological structures that a landing spacecraft could potentially deal with during operations. In the concept here proposed, the training process is carried out on ground, while only the trained

network runs in flight. The ANNs working principle relies on a long series of elementary mathematical operations (sums and multiplications), giving them a high computational efficiency, compatible with real-time systems.

It is assumed that the system receives as input only images from a monocular navigation camera and some basic telemetry, including spacecraft altitude and attitude. The former is required to estimate the scale factor of the observed scene, needed to correctly estimate the size of the possible landing sites, while the latter is exploited to correct distortions due to inclined views. The aim is to demonstrate the robustness and the effectiveness of a neural networks based system with minimal available information. In a real case, ANNs can be provided with additional input from different sources (LIDAR, feature tracking systems, stereo cameras etc.), making the system even more effective.

The system is coupled with a landing site selection algorithm that ranks the candidate sites following criteria of safety level, overall dimension of the landing area (to add additional margin to cope with possible landing dispersion), and distance from the nominal landing site (to maximize the probability to find a feasible trajectory toward the new selected target).

The chapter is structured as following: the assumption about how the HDA tasks are accomplished during a planetary landing are expounded in Section 3.1. Then, in Section 3.2, the system architecture is described; two different structures of ANNs are considered. The generation of ground truth models for system training and validation are explained in Section 3.3, and obtained results and performances are assessed in Section 3.4.

3.1 Nominal HDA Maneuver

Some assumptions about the operations sequence during a planetary landing maneuver are here expounded. As seen in Chapter 2, HDA operations are assumed to be executed by a landing spacecraft during the so-called *Approach* phase, starting at an altitude between 2500 and 1000 m, as the nominal landing site comes into sensors' field of view, depending on the landing strategy and on the specific target celestial body. The actual required performances and operative conditions of a hazard detection system largely depend on the specific operations

planned to be performed in this phase.

For this work, a lunar landing is considered. The assumed operation sequence is derived from real mission requirements (see [14, 64, 79, 100, 102]), and it is summarized in Figure 3.1. Following, it is described in detail:

1. HDA maneuvers starts at the beginning of the Approach Phase. At this point, called *HDA High Gate*, the lander is assumed to have a near down-looking camera, a vertical downward velocity in the range $15-30\,\mathrm{m\,s^{-1}}$, and horizontal velocity (downrange) $5-15\,\mathrm{m\,s^{-1}}$.

2. The system performs a *Large Scale Hazard Avoidance Maneuver*:

   (a) The system scans the landing area and builds a large-scale hazard map. A new target landing site is selected.

   (b) The landing trajectory is computed and a diversion maneuver is commanded. The final target of this maneuver is a point, called *HDA Low Gate*, located on the vertical to the target, at altitude $250-500\,\mathrm{m}$. The target velocities are $10-5\,\mathrm{m\,s^{-1}}$ and $2-4\,\mathrm{m\,s^{-1}}$ respectively in vertical and horizontal directions, with vertical attitude.

3. Then, the system performs a *Small Scale Hazard Avoidance Maneuver*:

   (a) The system scans again the landing area, building a small-scale hazard map. If required by hazard detection, the target landing site is updated.

   (b) The trajectory is updated and if needed a new diversion maneuver is commanded. The target point, called *Terminal Gate*, is located at the target landing site, at altitude $30-50\,\mathrm{m}$. At the Terminal Gate the lander is required to have a vertical attitude, with a null horizontal velocity ($\leq 1\,\mathrm{m\,s^{-1}}$) and a small vertical speed $\leq 3\,\mathrm{m\,s^{-1}}$.

4. At the Terminal Gate the Approach phase ends and the *Terminal Descent* phase starts. In this phase, the lander follows a vertical trajectory at constant speed until touch down.

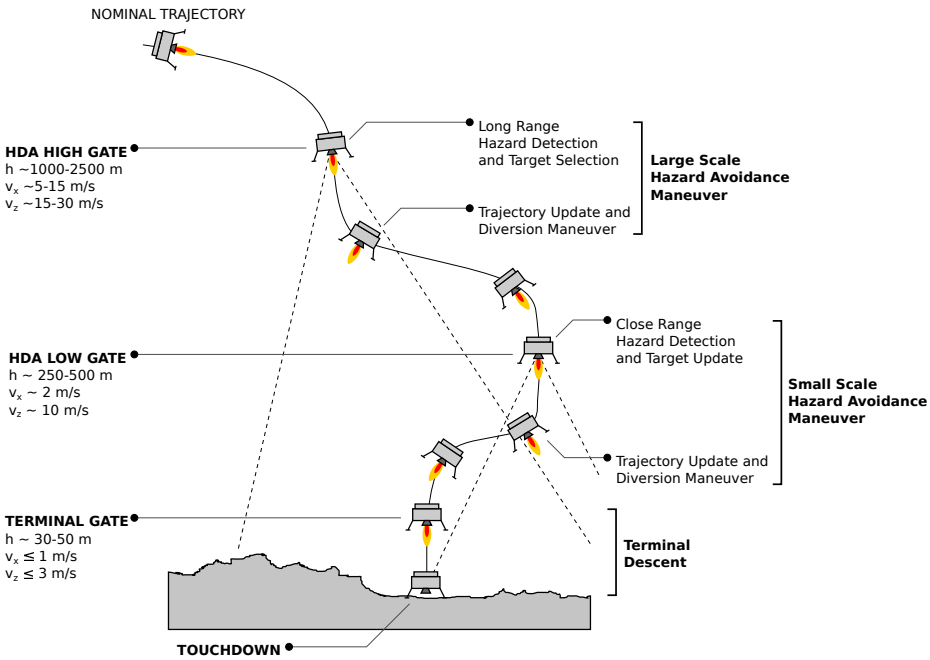


Figure 3.1: *Nominal Hazard Detection and Avoidance maneuver.*

3.2 System Architecture

In Fig. 3.2 the logical scheme of the hazard detection system is shown. For each position of the landing area, this system assigns a hazard index, giving a measure of the safety of that position if chosen as landing site. Hazard index can assume any value between 0 (completely safe) and 1 (absolutely hazardous). The hazard detection consists in 4 stages:

1. Preprocessing: the raw image is acquired. The image is rectified to compensate the perspective distortion in case of inclined view.

2. Indexes extraction: image is segmented at different scales and low level information (e.g. local mean, variance, etc.) is extracted.

3. The extracted information is processed by an artificial neural network and arranged in a hazard map.

4. TLS search. Computed hazard map is exploited to select the most attractive landing site.

In the following sections each stage is expounded in details.

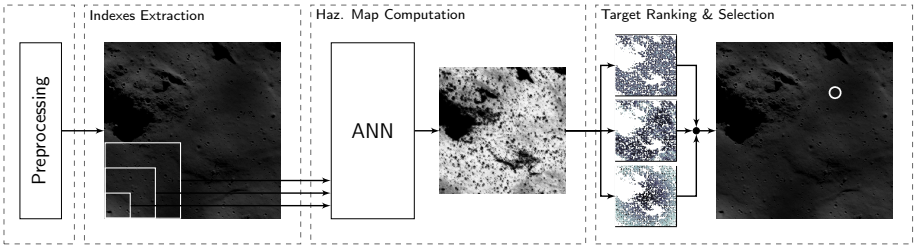


Figure 3.2: *Hazard Detection system logical scheme. An artificial neural network estimates hazard index value from elementary information extracted from the image at different scales. Different ranking criteria are fused together to select the best target.*

3.2.1 Input and Preprocessing

Grayscale (8 bit, single channel) images have been considered as system input. An image size of 1024×1024 pixels (compatible with most of the present camera devices for use in space) has been adopted.

Due to the kind of the considered Approach maneuver, it is assumed that HDA systems operate in a near vertical attitude. Small deviations from nadir pointing are corrected by the application of a perspective transformation [109] just before the hazard map computation.

In the case of an inclined view, the pixel size is not uniform across the image: more distant objects appear smaller. In the perspective correction process the frame is rectified with a linear coordinates transformation and then re-sampled to maintain a uniform pixel dimension. This process is required to proper carry out the subsequent computation, but no additional information is actually recovered. Then, the image resolution *before* resampling should be high enough to allow the system to correctly detect hazardous features, in every part of the frame. The resolution requirement is different at different altitudes: for a large scale maneuver, the pixel size should not exceed the order of magnitude of the lander footprint, while for lower altitudes, corresponding to a low scale hazard detection, it should not be greater than the largest obstacle dimension handleable by the landing system. Assuming a 60° field of view pinhole camera and a 1024×1024 pixels sensor, the pixel size for a perfect down-looking

attitude is equal to 2.25 m uniformly on the whole image, taken from 2000 m altitude. For an inclined view, this value increases (at worst) up to 2.88 m for a deviation from nadir of 10°. After this value, the pixel size becomes quickly very large, increasing proportionally to the tangent of the view angle. Considering also that, with the same inclination, the pixel size increases linearly with the altitude, and assuming a medium size lander with a 3 m footprint diameter, angles up to 10° are considered acceptable.

This type of transformation assumes that attitude information is known by the system, and that the scene can be considered as flat. Errors in attitude and altitude estimation, and presence of orographic reliefs introduce approximations whose impact over the system is discussed more deeply in Section 3.4.3. If additional information is available, e.g. a terrain shape model by a vision-based navigation system, more complex corrections can be applied [110].

Plus, in order to perform hazard detection and avoidance tasks, the landing site region is required to come into the sensor's field of view with sufficient time and control margin to maximize the lander divert capabilities. This requirement excludes too inclined trajectories [79], like the ones exploited during the Apollo missions [62], limiting in this way the maximum view angle during the HDA phase.

3.2.2 Information Extraction and ANN Input Assembly

Indexes extraction is a key feature for a correct hazard detection with neural networks. It consists in the extraction of low level information from the raw image, with the aim to reduce the space of the data in which neural network can detect morphological features of the real world. The smaller the space, the more effective the network training process, together with a reduction of the system's complexity. At the same time, an inappropriate choice of the indexes could lead to an excessive loss of information, with consequent loss of accuracy in the results.

A single-channel image can be considered as a discrete two-dimensional function where the intensity on the ith pixel I_i depends on the two spatial coordinates:

$$I_i = f(x_i, y_i). \tag{3.1}$$

Then, it is possible to obtain derivative information with the

appropriate discrete filters. Zeroth order information allows to have a reference value from which the network can understand the general brightness of the area analyzed in the acquired image. First and second derivatives are instead indicators of the presence of specific features on planetary surface detecting variations of pixel intensity. Four different quantities were selected: mean, standard deviation, gradient (1st derivative) and Laplacian of Gaussian (estimation of the 2nd derivative). All the selected entities can be computed by the application of different linear filters to the image. In this work, only the most basic quantities have been considered, trying to maintain the computational complexity at a minimum. More sophisticated techniques to extract information from images actually exist, such as independent component analysis [111], principal component analysis [112], and Gabor filters [113]. Despite their high efficiency in carrying information, they are also more demanding in terms of computational burden: their study is left for future developments (see Section 4 for a more complete discussion).

Each quantity can be computed in two different ways: by segmentation or globally. In the first case, the input image is segmented in sub-windows: the index is computed for the whole bounce of pixels included in each of them. The resulting matrix of indexes has a dimension equal to the size of the input divided by the size of the adopted window. In the global case, the index is instead computed on the whole image, by applying a linear filter with a sliding kernel of predefined dimension. Then, the result is a matrix of the same size of the input: the desired final size is obtained by downsampling the image trough Gaussian pyramid in order to match the size of the desired scale.

The knowledge of the same kind of information at different scales allows neural networks to better understand depth and relative distances as described by [114]. The scale can be varied by varying the segmentation window size or the number of subsequent downsamplings. Five scales, 2, 4, 8, 16, and 32 pixels (corresponding to a downsampling pyramid levels number from 1 to 5 for global indexes) have been initially selected: after some preliminary evaluations, scales 2 and 32 have been discarded. The former because of the greater computation required, the latter for too excessive loss of information.

Combining 4 quantities, 2 extraction methods and the 3 remaining scales, a total of 24 indexes are available. In a try and error process,

they were tested in 12 different configurations. In this work, only the most effective is presented: the minimum RMS error obtained at the end of ANNs training phase, measured on the validation set (see Section 3.3), was adopted as criterion to estimate the effectiveness of the computation. In the actual implementation all the 3 different scales, called simply *small*, *medium* and *large* from here onward, respectively of $s_S = 4$, $s_M = 8$, $s_L = 16$ pixel, are included, leading to a final hazard map of 256×256 pixel in size. For a down-looking camera with a 60° angle of view, this corresponds to a resolution of 9.02 meters per pixel, for images taken from an altitude of 2000 m (representative of a HDA High Gate altitude), decreasing to a value of 1.80 meters per pixel for images taken from 400 m. This value is of the same order of magnitude of a realistic lander footprint, making the final hazard map resolution appropriate for an efficient landing site selection.

Mean μ and standard deviation σ are computed by segmentation at each of the considered scales. The two statistical indexes are defined as:

$$\mu = \frac{\sum_{i=1}^{N} I_i}{N}, \tag{3.2}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} (I_i - \mu)^2}{N-1}}, \tag{3.3}$$

where I_i corresponds to the intensity of the i-th pixel, N is the number of pixels inside the considered image window. Image gradient (*Grad*) and Laplacian of Gaussian (*LoG*) are computed by global filtering and downsampling. *Grad* is approximated through an expanded 5×5 Prewitt kernel for both horizontal and vertical directions [115]. Then, the square root of the sum of the square of every element of directional gradients yields the total image gradient. Laplacian of Gaussian is a second order operator widely used as edge detector [116, Ch. 4.2]. It combines a Gaussian smoothing with the Laplacian operator and its general formulation in continuous space is:

$$LoG(x, y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2 + y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{3.4}$$

where x, y represents image coordinates, centered at the current point, while the filter standard deviation σ determines the characteristic

length at which the filter tends to reject noise. Here it has been implemented through a discrete linear 5×5 kernel.

In addiction to these indexes, also *Sun inclination angle*, defined as the angle of the sun above the local horizon, is assembled in the input matrix. This is necessary to make the neural network able to correctly distinguish between sharp and blunt shadows. Summarizing, the value of each hazard map pixel is computed from a vector of 13 component: μ, σ, Grad and LoG for each of the 3 considered scales, plus the Sun inclination angle. Each component is normalized before it is passed to the next stage. Eventually, assembling of the whole input is concluded expanding indexes relative to bigger image windows and higher downsample levels because of their intrinsic smaller size, with a nearest neighbor criterion.

3.2.3 Artificial Neural Network and H-Map Computation

After assembling, input is processed by an Artificial Neural Network. In this work, two alternative architectures are considered. A *Feedforward Multilayer Neural Network*, and a *Cascade Neural Network*.

Feedforward multilayer (shown in Figure 3.3) is the most simple type of ANN, widely used in function regression and pattern recognition [117, 118]. It consists in an input layer, one or more hidden layers, and an output layer. The input is processed by each layer sequentially; each layer is fully connected with the subsequent one. The network architecture (number of hidden layers and neurons for each layer) is a priori determined; the training process consists in an optimization (by error backpropagation) of the weights of each neuron. The main criticality of feedforward multilayer networks is the choice of a suitable architecture: too complex schemes make the system prone to overfitting phenomena and slow down the optimization process. In this work, in order to determine the minimal effective size of the net, progressively more complex architectures have been tested: the final scheme consists in a single hidden layer made up by 15 neurons. Addition of further neurons produces no significant performance improvement.

In cascade networks, each layer is made up by a single neuron: the input for each layer is the original input of the network, plus the output of each previous hidden layer (see Figure 3.4). In this structure,

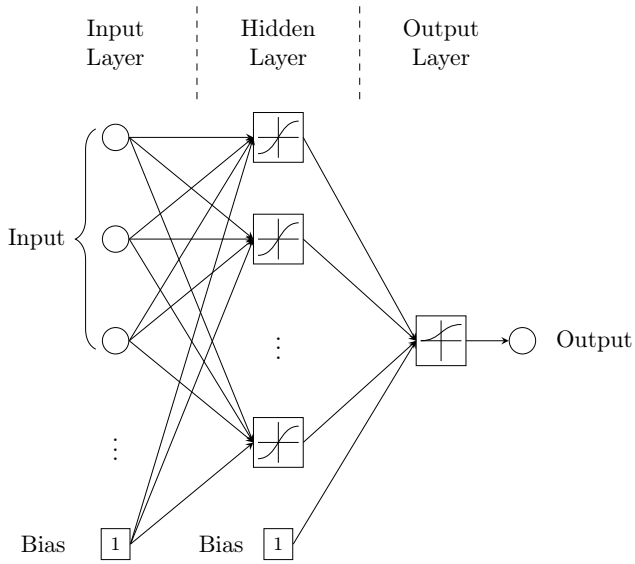


Figure 3.3: *Multilayer feed-forward network structure. The network structure is predetermined and only neurons' weights are trained.*

hidden neurons are part of the optimization: at the beginning of the training the net has only input and output layers, and hidden units are progressively added, leading to a near-optimal architecture [119].

In the selection of the neural network architecture, the same criteria followed for the choice of the indexes are followed: simpler solutions are privileged, to assess the possibility to obtain an efficient hazard detector with the minimum of the complexity. While multilayer network is the most simple type of ANN, the cascade network is a first attempt to introduce an optimal architecture maintaining a minimal structure. The two architectures share some characteristics. Hyperbolic tangent is used as hidden layer activation function, making the networks able to handle nonlinearities [120]. Since the desired output is limited in the range $[0, 1]$, the logarithmic sigmoid is adopted as output function. At the end of the computation, a very light blur filter is applied on the hazard map to relate nearby pixels.

3.2.4 Target Selection

Once the hazard map is computed, the system seeks a safe landing site. Possible sites are classified and ranked according to the following drivers:

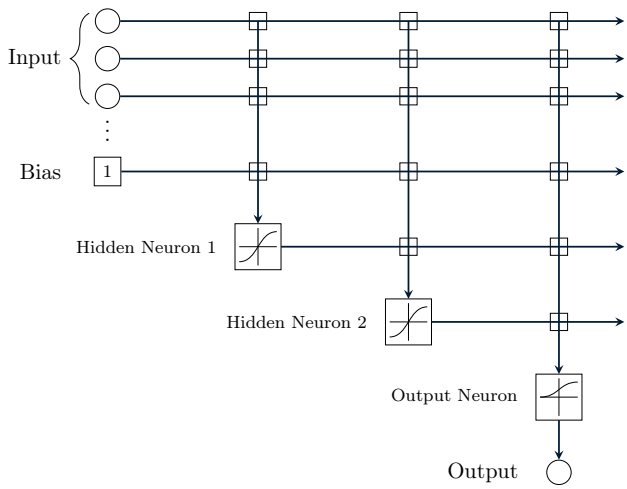


Figure 3.4: *Cascade network structure. Hidden neurons are progressively added during training. Each white square represents a weight.*

- Minimum hazard level;
- Maximum landing area;
- Minimum distance from nominal landing site (required to maximize the probability to find a landing site actually reachable by the lander divert capabilities).

For each of these principles, a specific index is assigned to each landing site candidate. Then, the three indexes are fused in a unique score exploited to create a global landing site ranking. The image reference frame is considered. This frame is centered in the pixel at the upper-left image corner, aligned with the image borders. Distances in image reference frame can be expressed in pixel units (the position of each pixel corresponds to the number of its column and row, numerated from 0) or in real units (meters). The transformation between them is a simple scaling conversion:

$$\mathbf{r} = d_{\text{res}} s_{\text{s}} \mathbf{x}, \tag{3.5}$$

where $\mathbf{r}^T$ is the image position vector expressed in meters, $\mathbf{x}^T$ is the same vector expressed in pixels units, d_{res} is the original image resolution (after the perspective correction), expressed in meters per pixel, and s_{s} is the length, in pixels, of the "small" window

considered by neural networks during hazard map computation. A sufficiently good estimation of the image resolution is required to correctly scale the scene and then estimate the dimension of the areas available for the landing. Then, at least information about the altitude, camera field of view, and attitude (the minimum required for image perspective correction and resolution estimation) are assumed to be available to the system.
The following procedure is adopted:

1. The hazard map is thresholded at the maximum tolerable level of hazard index for a safe landing site, denoted as $z_{\max}$. All the pixels above the threshold are classified as unsafe. Pixels under this level are initially classified as Candidate Landing Site (CLS).

2. For each pixel at coordinates $\mathbf{x}^T = [i, j]$ with $z_{ij} \leq z_{\max}$ the *Size Score* $r_{\mathrm{CLS}ij}$ is computed as the distance from the nearest unsafe pixel.

3. A safe landing site is required to respect a minimum dimension requirement. Modeled as a circle, its radius is required to be:

$$r_{\mathrm{CLS}ij} \geq r_{\min} = \frac{d_{\mathrm{foot}}}{2} + e_{\mathrm{gnc}}, \tag{3.6}$$

   where d_{foot} is the lander footprint diameter, and e_{gnc} is the expected landing error due to navigation imprecision, with the desired level of confidence. Then, all CLSs that do not respect this constraint are discharged.

4. For each remaining CLS the *Diversion Score* $d_{\mathrm{CLS}ij}$ is computed as the distance from the Nominal Landing Site (NLS):

$$d_{\mathrm{CLS}ij} = \|\mathbf{r}_{\mathrm{CLS}ij} - \mathbf{r}_{\mathrm{NLS}}\|, \tag{3.7}$$

   where $\mathbf{r}_{\mathrm{CLS}ij}$ is the metric position of the CLS at image coordinates, and $\mathbf{r}_{\mathrm{NLS}}$ is the metric position of the NLS.

5. The *Safety Score* $z_{\mathrm{CLS}ij}$ of each CLS is obtained by the mean of the hazard index of the pixels contained in the circle centered at the CLS_{ij} of radius $r_{\mathrm{CLS}ij}$.

6. The three scores are normalized to make their values to the same order of magnitude. Than, a global "landability" score $l_{\mathrm{CLS}ij}$ is obtained as:

$$l_{\mathrm{CLS}ij} = \mathbf{w}^T \begin{bmatrix} \tilde{r}_{\mathrm{CLS}ij} \\ 1 - \tilde{d}_{\mathrm{CLS}ij} \\ 1 - \tilde{z}_{\mathrm{CLS}ij} \end{bmatrix}, \tag{3.8}$$

   where $\mathbf{w}^T$ is a vector of weights, introduced to give to the user the faculty to confer more relative importance to one index with respect to others. Symbols marked with a tilde stand for normalized values.

7. Finally, the CLS with the larger global score is selected as TLS.

3.3 Network Training

ANNs performance depend widely on the completeness and coherence of the dataset used to train the network. In the specific case of HDA systems, the type of training set must be also tailored on the celestial body target of the mission. In this work, a lunar landing case is considered.

True lunar images present several criticalities: additional data required to obtain the corresponding ground truth solution (in particular an affordable 3D model that can be associated with the images) are only available for a very limited number of sites. On the contrary artificial images make possible an objective and precise ground truth reconstruction, being all the setting and the 3D model used for image generation completely known. Despite that, an high level of photorealism is required to preserve coherence.

This latter approach has been selected for this work: a dataset of 98 artificial images has been generated and exploited for training and testing purposes. Images are taken from an altitude of 2000 m, a value inside the interval in which the HDA system is required to operate (see Section 3.1), and they are divided into 3 subsets: training, validation and test. The training set (67 images) is exploited to directly optimize the networks weights with a backpropagation algorithm; network overfitting is avoided using an early stopping method: the training is stopped when the RMS error, evaluated on the validation set (23

images), reaches a minimum. The test set, made up of 8 images, is exploited to assess the system performances.

3.3.1 Artificial Images Generation: SECRET-PLAN

The use of artificial images gives a complete control over the scene. For this motivation, it is very attractive for the generation of a training dataset, that should be as complete as possible in terms of terrain features and illumination conditions. On the other hand, a lack of realism in image generation lead to incoherent results, not representative of real operative conditions. A particular care is then required in the image generation process. Some affordable solutions are available on the market, like PANGU from University of Dundee / ESA [121], and Surrender by Airbus Defense and Space [122], but unfortunately the access to the software is often complex or expensive. For the purpose of this work, a new tool for the generation of realistic images, representative of a dataset taken by a monocular navigation camera during a lunar landing maneuver, has been developed. The tool has named SElenitic Camera with Ray-traced Environment and Terrain for Planetary LANding Simulator (SECRET-PLAN).

The tool is able to create artificial lunar landscapes from scratch, generated by fractal noise, or to refine existing DEMs up to the desired level of detail. High resolution DEMs of the Moon obtained from LROC data[1], with a variable resolution between 2 and 5 m/point, have been used as starting point for the creation of the dataset. First, the DEM resolution is improved up to 0.3 m/point, by adding fractal noise, boulders, and craters to small to be visible with the original resolution [123]. Craters deposition respects the statistical distribution observed on the real lunar surface, as well as the real craters formation process [124], while craters morphology follows empirical morphometric relations obtained from lunar imagery [125]. A texture proportional to the local roughness is provided together with the refined DEM. The output of SECRET-PLAN is a complete 3D model, together with a set of parameters regarding illumination conditions, camera model, lander position and attitude. Finally, the camera frame is rendered in *POV-Ray*[2] with the desired settings.

[1] Courtesy of NASA and Arizona State University. http://wms.lroc.asu.edu/lroc/rdr_product_select, last visit on: January 3, 2017.

[2] Persistence of Vision Raytracer (Version 3.7) [Computer software]. Retrieved from http://www.povray.org/download/

Ray-tracing techniques, followed in the rendering process, allow the attainment of the most realistic results, reproducing the same physical phenomena of reflection and diffusion of light that happens in the reality. A pinhole camera model, with a 60° angle of view has been adopted. Table 3.1 summarizes the assumed camera parameters. An example of image generated by SECRET-PLAN is shown in Figure 3.5.

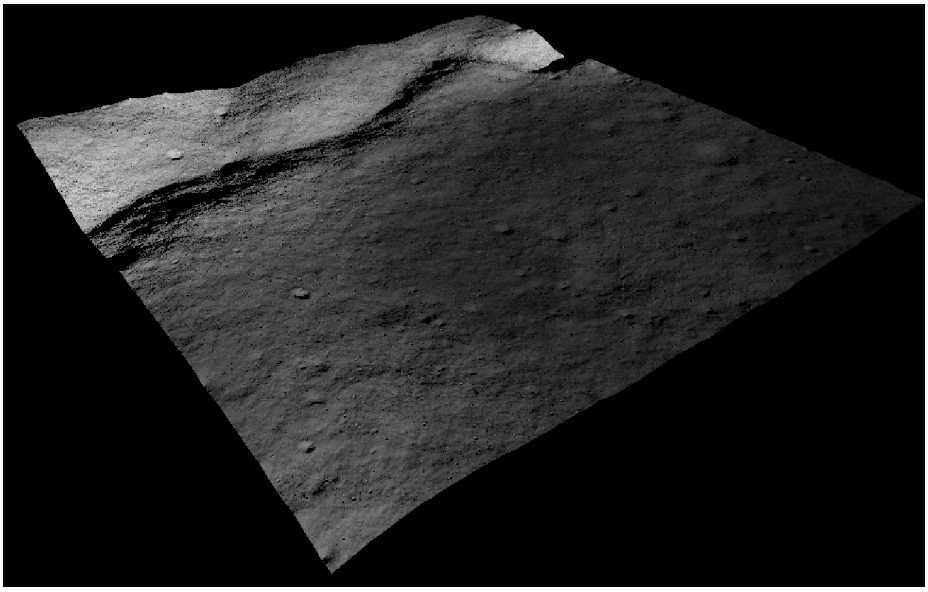


Figure 3.5: *SECRET-PLAN example image.* $4 \times 4\,\mathrm{km}$ *patch of the Planck Crater floor on the Moon.*

Table 3.1: *Camera model parameters for artificial images renderings.*

Model	Pinhole
Resolution	1024×1024 pixel
Angle	60°
Color	8 bit grayscale

3.3.2 Ground Truth Solution

Slopes and roughness can be extracted directly from DEM data. For each DEM point, a surrounding circular window with diameter

equal to the lander footprint is considered: the slope is computed as the inclination of the mean plane obtained by a least squares approximation of the points in the window. The plane is expressed by the equation:

$$Z = aX + bY + c, \tag{3.9}$$

where X and Y are the coordinates of the points in the window, and Z is the altitude. Then, the plane inclination S is obtained as:

$$S = \tan^{-1}\Big(\sqrt{a^2 + b^2}\Big). \tag{3.10}$$

Roughness R is estimated as the difference between the maximum and the minimum deviation of the window points from the mean plane:

$$R = \max\Big(Z_i - (aX_i + bY_i + c)\Big) - \min\Big(Z_i - (aX_i + bY_i + c)\Big), \tag{3.11}$$

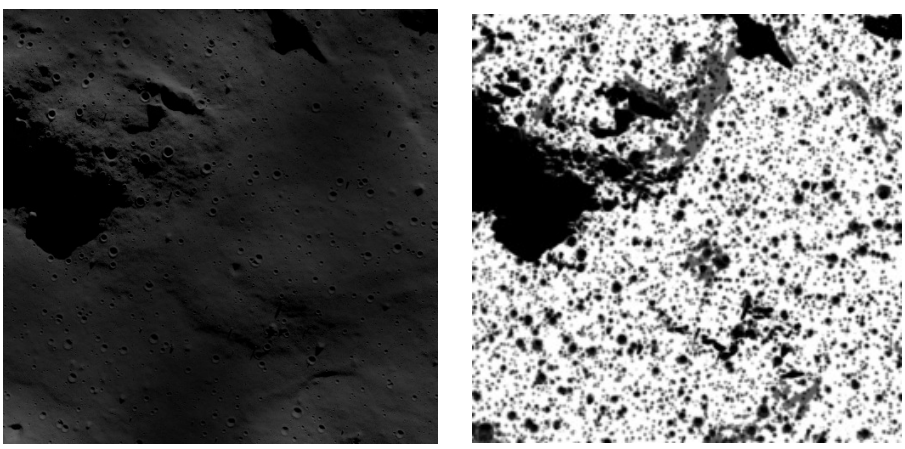
where the subscript i denotes the i-th DEM point inside the window. Once slopes and roughness maps are available, they are converted in camera image coordinates by a perspective transformation, computed through rendering software. Then, they are exploited to obtain the correspondent ground-truth hazard map. Each point is considered safe if respect the following conditions:

- $S \leq S_{\max}$,
- $R \leq R_{\max}$,
- the point is not in shadow.

Shadow map can be obtained through the same rendering software adopted for the image generation: a rendering of the scene with the terrain model textured in uniform white, with no reflections and no environmental light, produces a boolean map in which sunlit regions are perfectly white and shadows are perfectly black. Practically, the process has been exploited only for the first images to automatically identify an intensity threshold, in order to obtain shadow maps by simply thresholding the camera image histogram (with a consistent speedup of the training set generation process). At each pixel of the camera image is assigned the hazard index 0 (perfectly safe) if

respects all the conditions mentioned above. Pixels in shadow are considered as out of the sensor range, and are then considered as completely unsafe (hazard index 3). Hazard index 1 is assigned to those pixels that fail only one of the tests on slope and roughness, while the value 2 is assigned to those ones that fail both the tests. Then, the obtained hazard map is normalized to bring back the hazard index in the interval $[0, 1]$.

At this stage the ground truth hazard map has the same resolution of the camera image. In the last step, the map size is decreased up to the resolution computed by ANNs (256×256 for a 1024×1024 pixel frame) by applying a Gaussian pyramid. The downsampling process increases the hazard map smoothness, making easier the ANNs training process (ANNs are less effective in reproducing discontinuous functions). Figure 3.6a reports an example of artificial image, obtained from a real DEM of the Lowell crater floor, while the correspondent ground truth hazard map is depicted in Figure 3.6b. Based on this hazard map, is possible to compute the true safety and ranking of landing sites with the algorithm presented in the previous section, with $z_{\max} = 0.3$.

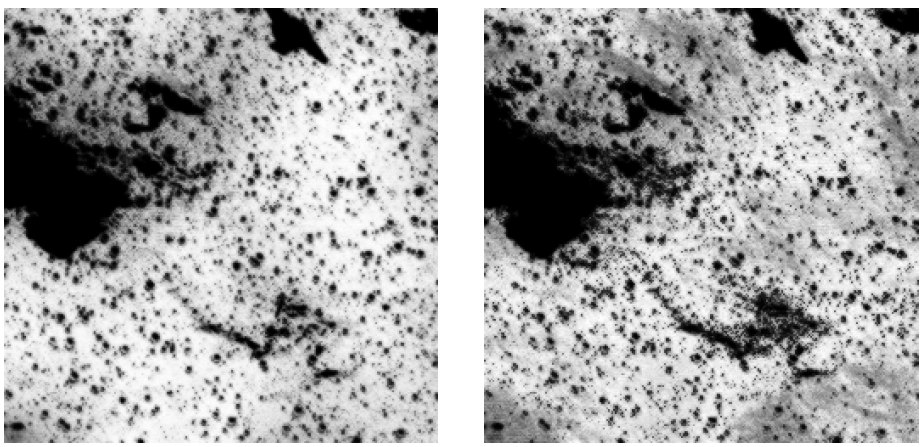


(a) *Original image.* **(b)** *Ground truth hazard map.*

Figure 3.6: *Ground truth solution computation. Artificial image, Lowell crater floor DEM. Original image and ground truth map. Safe areas are depicted in white ($z = 0$), while black represents shadowed regions ($z = 1$). Intermediate gray levels represents unsafe regions failing one single or both safety tests (roughness and slope).*

3.4 Performance Assessment

The hazard detector is written in C++ code. The OpenCV library[3] has been used for image preprocessing tasks. The training of the neural networks has been done with the FANN C++ library [126]. The system performance is verified by comparison with the ground truth solution of a test set, which consists of four landscapes with two different sun inclination angles (15° and 80°) for a total of 8 images. A footprint $d_{\text{foot}} = 3\,\text{m}$ and a navigation error $e_{\text{gnc}} = 15\,\text{m}$ (3σ) have been considered. The training ended for the two network with a very similar value of Mean Square Error (MSE) over the test set, equal respectively to 0.01940 (multilayer) and 0.02039 (cascade).



(a) *Multilayer network.* **(b)** *Cascade Network.*

Figure 3.7: *Computed hazard map. Artificial image, Lowell crater floor DEM. Safe areas are bright (low z).*

ANNs are not expected to exactly reproduce the original ground truth hazard map; instead an approximation of them is expected. In Figure 3.7 an example of hazard map computed by the two architectures, relative to the same input image of Figure 3.6, is shown. It is possible to see how in both cases all the large scale hazardous features are correctly detected; the network response tends to be conservative, with a mean hazard index higher than the ground truth solution. It can be seen how the cascade architecture tends

[3]The Open source Computer Vision library v2.7. Retrieved at: http://opencv.org/. Last visit: January 3, 2017

to give sharper results, while multilayer tends to smooth the hazard index. For these reasons, the safety threshold $z_{\max}$ used in the actual system is not required to be equal to the value used in ground truth computation: on the contrary, its value should be tailored to the specific architecture. Once compared with the ground truth, a landing site can be classified as:

- *True Positive* (TP): a site correctly classified as safe;
- *False Positive* (FP): an unsafe site, erroneously considered safe;
- *False Negative* (FN): a safe site, seen as unsafe;
- *True Negative* (TN): a correctly recognized unsafe site.

It this then possible to define the *Safety Ratio* r_S as the ratio between the number of TP landing sites and the total number of sites classified as safe:

$$r_S = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{3.12}$$

while the *Correctness Ratio* r_C is the ratio between the number of the correctly found sites and the total number of true safe landing sites in the image:

$$r_C = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{3.13}$$

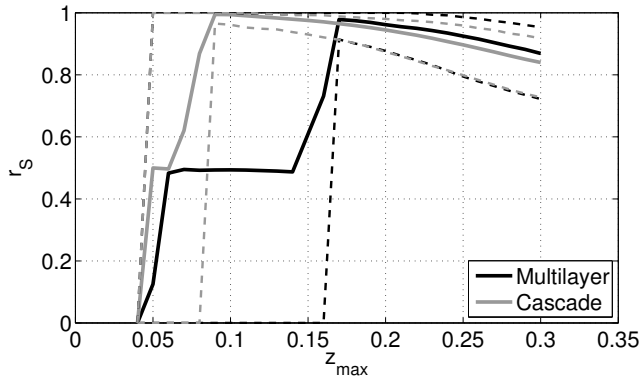
The probability to select an unsafe target is minimized by maximizing r_S, while the maximization of r_C increase the available landing area. The system performances can be expressed in a unique index J defined as:

$$J = r_S^5 \, r_C^{1/5}, \tag{3.14}$$

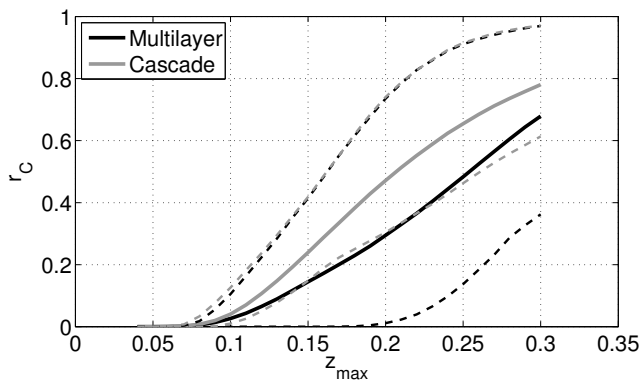
where the exponents 5 and 1/5 are introduced to give more relative importance to landing safety, which is the main driver in the performance assessment.

Values of $z_{\max}$ from 0.04 to 0.30 have been tested on the hazard maps computed by the architectures under exams on the test set images: obtained r_S and r_C are shown in Figure 3.8. A null ratio value mean that no landing site has been found for the specific threshold value. Looking at the compound performance index J in

Figure 3.9, it is easy to identify the best threshold values of 0.23 and 0.17 respectively for the multilayer and the cascade networks.



(a) *Safety Ratio.*



(b) *Completeness Ratio.*

Figure 3.8: *Architectures comparison, z_{max} threshold tuning. Safety and Completeness ratios define the overall system performance. Solid lines are mean values, dashed lines are lower and upper boundaries.*

Once the threshold is selected, it is possible to compare the performances of the two architectures. In sites ranking, a weight vector $\mathbf{w}^T = [0.6\ 0.3\ 0.1]$ is adopted. The heavier weight (0.6) has been assigned to the size score r_{CLS}: a larger available area increase the robustness of the system with respect to navigation uncertainties. The intermediate weight corresponds to the diversion score d_{CLS}, in order to maximize the probability that a feasible trajectory is found by the guidance system, minimizing the requested diversion; the lightest weight is linked to the mean hazard index, being all

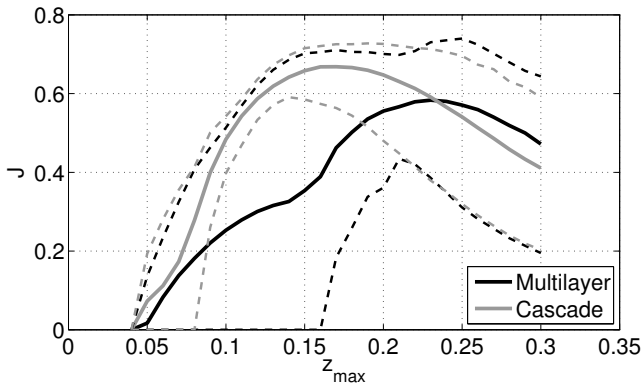


Figure 3.9: *z_{max} threshold tuning. Overall performance index (higher is better). Solid lines are mean values, dashed lines are lower and upper boundaries.*

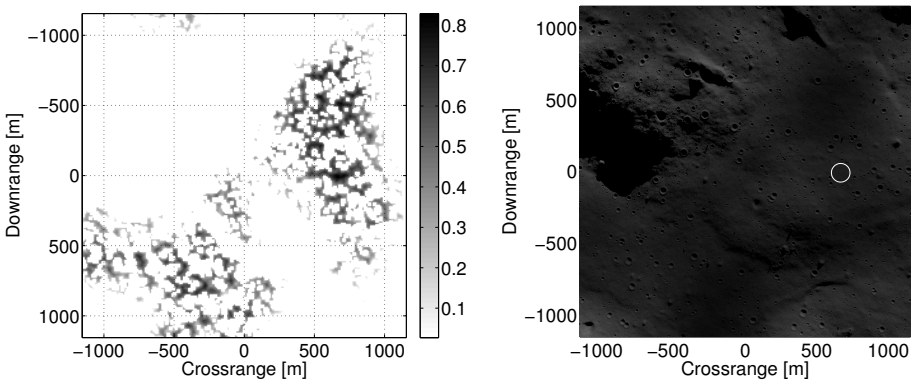
the candidate targets already under the threshold $z_{\max}$, and then considered as safe. Following the obtained results are summarized:

- Always a True Positive is selected as first Target Landing site;
- The mean ranking of the first FP is 695 (cascade) and 460 (multilayer), allowing the system to always find a backup landing site if required by the guidance system (worst case on single image: 39 and 38, respectively).

Globally, the cascade architecture obtain a higher score principally due to a higher safety ratio (0.9649 with respect to 0.9430 obtained by multilayer architecture). Figures 3.10 and 3.11 show the landing site ranking, and the final selected target, for the original image of Figure 3.6, computed by the two architectures under test. It is possible to see how the sharper hazard map obtained by cascade networks allows to find a larger number of landing sites also with the application of a smaller threshold value.

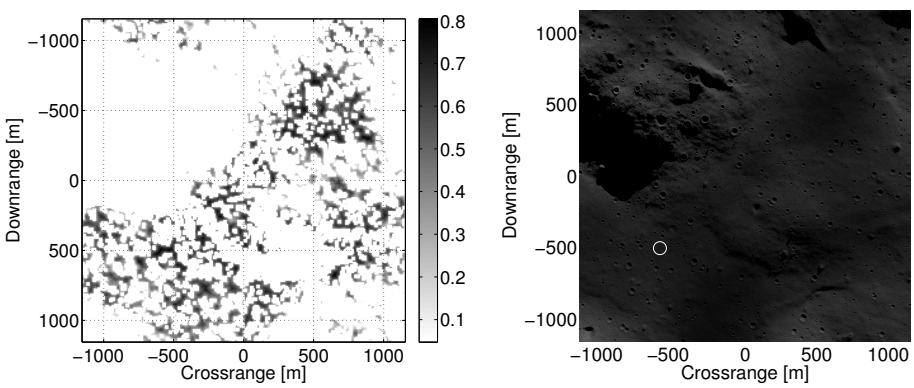
3.4.1 Real Images

The system has been also tested on real lunar images and photos taken by Rosetta mission of the 67P/Churyumov–Gerasimenko. Being unknown the Sun inclination angle, it has been briefly hypothesized looking at the photos. Also the attitude of the spacecraft at the time of the shots is a missing data (considering also that a true "local



(a) *Landing sites ranking.* **(b)** *Selected target.*

Figure 3.10: *Target ranking and selection. Multilayer ANN ($z_{max} = 0.23$).*



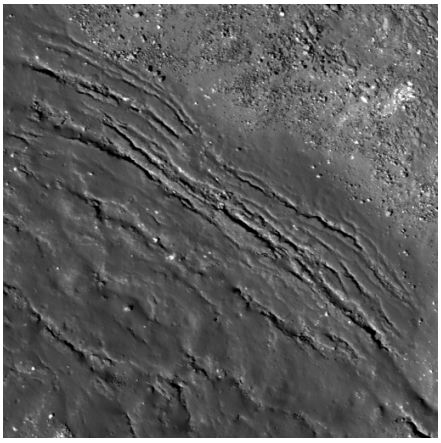
(a) *Landing sites ranking.* **(b)** *Selected target.*

Figure 3.11: *Target ranking and selection. Cascade ANN ($z_{max} = 0.17$).*

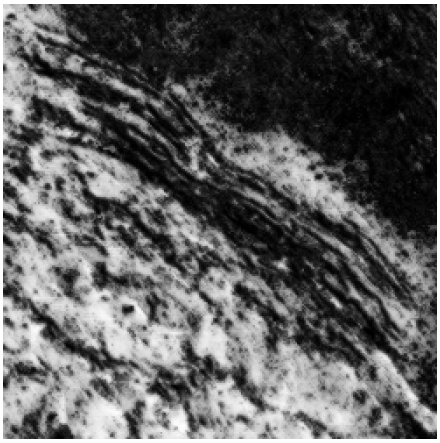
horizon" cannot be determined in the case of the 67P photos) the images are then assumed to be taken with a perfect down-looking pointing, and no perspective correction is applied. Moreover, in these images there is no ground truth to quantitatively test the hazard maps with. Thus, results are to be intended just as an example of the ANNs generalization capabilities, and must be not intended as a valuable result of the hazard detection system. Anyway, the choice of the photos was dictated by the presence of relevant morphological features, that could have challenged the system.

Moon In Fig. 3.12a, taken by LROC Narrow Angle Camera, depicting part of the Larmor Q crater floor, it is possible to spot some fractures on the surface in the lower left hand side half, while the rest of the image is characterized by diffuse roughness due to craters. In its relative hazard maps (Figures 3.12b and 3.12c), the neural networks seem to have qualitatively understood the terrain features, assigning a distributed high hazard value to the rough region at top right hand side and about maximum value precisely where fractures are located. The higher sharpness of cascade networks w.r.t. the smoother maps computed by multilayer architecture is clearly visible.

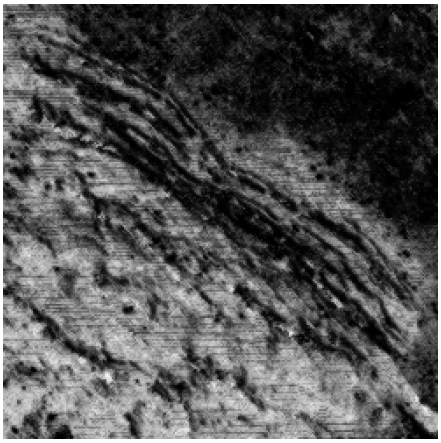
67P/Churyumov–Gerasimenko The great interest of both the scientific community and companies in small celestial bodies pushed to test the same hazard system used for lunar images on 67P/C-G. Not many suitable images are available for the purpose, and even less are equipped with data the neural network should need to be as much efficient as it can. A test on the Imhotep region is presented in Fig. 3.13a. This area is composed by many well distinct features: a planar plateau with sharp boulders and rifts, developing from the center to the top of the picture, surrounded by an irregular area full of craters and high sloped sides. In the relative computed hazard map (Figures 3.13b and 3.13c), the system seems to have qualitatively understood hazard trends of the various areas: bright colors (safe) for the planar area apart from the irregularities, gray and black (unsafe) for the most of the rest. The very low albedo of this particular image is sometimes interpreted as shadow by the cascade system, which give more conservative results.



(a) *Original frame.*

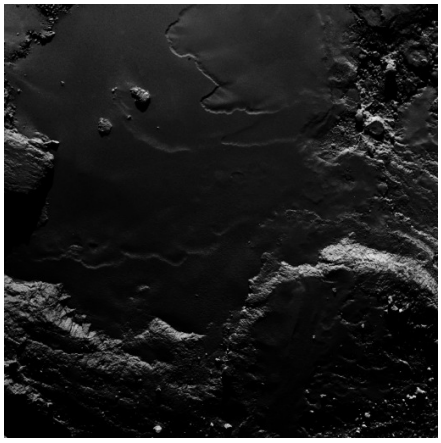


(b) *Haz. map, multilayer network.*

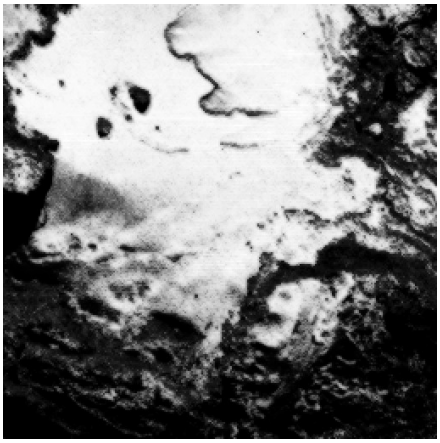


(c) *Haz. map, cascade network.*

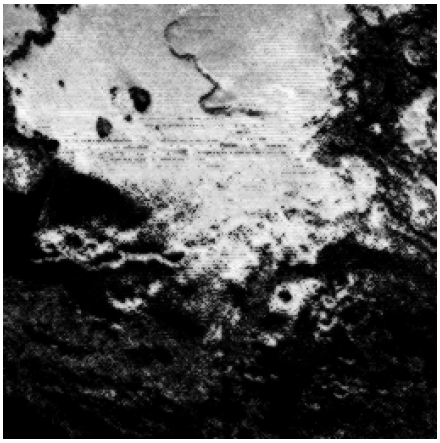
Figure 3.12: *Lunar real surface image application, Larmor Q crater floor, NAC frame M151726155R, courtesy of NASA/GSFC/ASU.*



(a) *Original frame.*



(b) *Haz. map, multilayer network.*



(c) *Haz. map, cascade network.*

Figure 3.13: *Comet 67P Churyumov-Gerasimenko, Imhotep region. Original frame and hazard maps (Photo: ESA).*

3.4.2 Profiling

To properly estimate the computational weight of the proposed HDA system, a profiling analysis has been carried out. *Gperftools*, a tool released by Google under BSD license, has been selected as main profiler method. Results have been cross-checked in two independent ways: each subroutine execution has been measured with the high resolution clock of the standard C++ *chrono* library, while with the *GNU/time* command has been exploited to verify the overall computation. All tests have been performed on a AMD A10-7700K APU, running 64 bit Ubuntu 14.04 GNU/Linux operative system. In each profiling test, the hazard detector runs in a cycle for 1000 times, while the sampling frequency has been set to 250 Hz (the highest possible value) to maximize the precision in runtime estimation. In order to avoid modern processors' automatic multi-core computation, the system has been forced to run in single-thread configuration.

The estimation of the global computation time has a relative usefulness, being the performance of a real space hardware running with a real-time operative system far different with respect to a common PC, with diversities that are not completely a priori predictable. On the other hand, it allows to evaluate the relative computational weight of the subroutines that the algorithm is made up of, and to identify possible bottlenecks.

Gperftools registered 108 148 hits at 250 Hz, for a total time of 432.59 s, while the correspondent CPU time resulted 432.67 s. Taking into account the possible overhead that can affect measurements differently with the two methods, the values are comparable. Figure 3.14 shows the breakdown of the computation time over the different algorithm stages: the principal bottleneck is identified in indexes extraction, that requires more than the 49% or the total runtime. This result agrees with the expected: image processing algorithms, that constitute most of the task, are computationally expensive.

The sensitivity of the hazard map computation time from the actual network architecture is weak. ANNs are computationally efficient and the number of operations to be performed in this stage has the same order of magnitude in both feedforward multilayer and cascade neural network.

The computation time demanded by preprocessing, indexes extrac-

tion, and hazard map computation is practically constant, operating these subroutines on the whole image indifferently. It is not the same for target ranking and selection, since the number of sites classified as safe is determined by both the input image and the adopted threshold $z_{\max}$. To estimate the theoretical maximum required time, a profiling test with $z_{\max} = 1$ has been performed. In this condition, all the image is considered as a potential target and performances decrease drastically, as can be seen looking at the dashed line in Figure 3.14. This case is highly improbable, solely exploited to identify a theoretical upper boundary of the computational burden: the actual mean time measured during profiling test (solid gray in the graph) is much lower, and target selection resulted the fastest of the four algorithm's stages. All the algorithm bottlenecks are located in potentially high parallel tasks: recent developments of dedicated space qualified hardware, based on high performance parallel units (such as Field Programmable Gate Arrays) allow to expect further improvements in real applications up to real-time performances, with a speedup greater than 100 times with respect a full software implementation [127, 128].

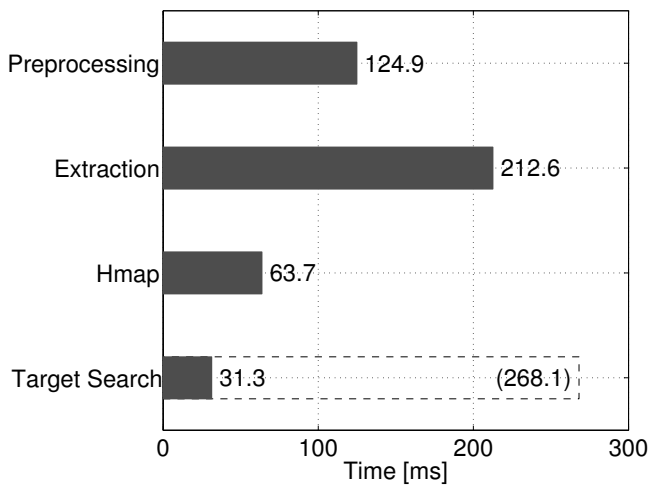


Figure 3.14: *Computation time breakdown. It is easy to see the bottleneck of the indexes extraction stage. Theoretical maximum time required by target search is shown in dashed line.*

3.4.3 Sensitivity to Uncertainties

The proposed system requires additional information about the altitude and the attitude of the spacecraft: in this section it is briefly discussed how uncertainties in this information could affect the proposed hazard detection system. The attitude measure is involved in the perspective correction during the preprocessing phase: in this case, the deformation of the scene is not uniform on the image, but increasing with the inclination at which a certain region is seen (proportional to the tangent of the angle). Assuming a maximum allowed inclination from the vertical equal to 15° and a 60° FoV for the camera, the maximum inclination at which an area can appear in the input image is 45°. The expected attitude estimation accuracy is far better than 1°, relying on star trackers and inertial measurement units [92]. Taking a 1 degree error as a worst case scenario, the corresponding maximum relative error in lengths estimation would be 3.55 %.

The altitude value is required for the computation of the image resolution d_{res}: uncertainties in the altitude estimation introduces a scaling error of the entire scene, with a linear relationship. In real systems the expected accuracy in position (and then altitude) estimation is better than 1 %, using pure vision-based navigation. Conjugated with laser range measures, an accuracy in position better than 0.2 m is expected [56, 129]: this is equivalent to an error less than 0.2 % from an altitude of 1000 m. The errors introduced by altitude measurement are then at least one order of magnitude lower with respect to other sources of uncertainty.

Also unknown orographic reliefs introduce errors in distances estimation: a certain level of distortion is introduced by the perspective correction, applied with the flat scene approximation, especially near the edges of the scene. A certain amount of non uniform scaling error is also introduced, due to the deviation from the scene mean plane that alters the actual distance of the terrain with respect to the camera. ANNs proved to be effective in correctly identifying the slopes, while including the size of the landing area in the target ranking criteria copes with scaling errors, privileging wider target. During the tests, all the points selected as target resulted several times larger than the minimum required. Terrain models with deviations from the mean plane up to ±10 % (with respect to the altitude) have been

included in the training and test set.

And the story ends
Insanity said coldly
Still waiting for the chance
So out of nowhere it will rise
Oh, and another journey starts

Blind Guardian

4 Conclusion

One significant step toward the implementation of a full autonomous landing GNC chain for space applications is presented in this work. Attention has been put on two of the most innovative tasks that such a system is required to accomplish: adaptive trajectory computation and hazard detection.

Adaptive Guidance. A retargeting algorithm for spacecraft landing, capable of updating and correcting a landing trajectory almost to the touchdown, is presented. A novel approach based on the inclusion of free parameters in a classical polynomial formulation is proposed, in order to improve flexibility in the landing site choice, and to consider additional non linear constraints during the descent, such as thrust magnitude and attitude control torques boundaries. The resulting algorithm has light computational load, and maintains a high divert capability even with the use of a simple optimization algorithm. A more complex optimizer based on Differential Algebra and specifically developed for the landing optimization showed the capability to get the true optimum solution of the parameterized problem still maintaining a low computational burden.

The flexibility and the robustness of the proposed approach have been tested by applying it in retargeting simulations of two very

different cases, a lunar landing and a landing over a NEA, characterized by time scales and dynamics separated by at least 2 orders of magnitude. Monte Carlo simulations have been exploited to assess the algorithm retargeting capabilities.

The enforcement of the additional constraints has been verified by complete landing MC simulations. The proposed guidance has been applied to a lander simulator, including perturbed states (introduced in order to emulate navigation system errors), a simple control system, and pulsed actuators. The guidance algorithm resulted able to find a feasible landing trajectory in all the tested cases, with attainable landing areas larger than what is expected to be required in future missions.

It has been observed that accuracy at touchdown is mainly affected by navigation errors. Their impact can be mitigated by updating the landing trajectory several times during the descent. This strategy has proved effective especially in fast maneuvers, as in the tested lunar landing case. On the other hand, it has been observed that in the case of slow maneuvers (as in the NEA landing simulation) errors can propagate more easily. Introducing an opportune waypoint on the landing trajectory, a high level of precision is recovered, with a negligible effect on fuel consumption. In these situations an accurate project of the retargeting phase and of the guidance logic (waypoints, update frequency, open or close loop control between two consecutive updates) is required.

Hazard Detection and Target Selection. A new Hazard Detection and Avoidance algorithm, based on Artificial Neural Networks, is proposed as well. A deep analysis to detect which information can be extracted from the original image and exploited as network input has been carried out. The extraction of the most informative indexes from the original image reduces the dimension of the neural network input space, allowing a precise classification with a light network architecture, maintaining in this way a low computational weight. A fully objective training and validation method has been developed, in order to avoid any dependency of the system performance from the operator's choices during the training phase and to have an affordable estimation of the system capabilities. Two possible ANNs types have been tested; also if a safe landing site was selected in the 100 % of the test cases, the cascade networks ability to autonomously get a

suboptimal structure makes them more effective in false positives rejection than standard multilayer neural networks. More complex ANNs architectures are still possible, with additional margins of improvement.

A primary role in detection accuracy is played by the information extraction stage: a proper selection of the input makes the neural networks training process faster and more effective. In this field, several improvements are still possible, and some options are under study. Discrete wavelet transform is a promising technique, allowing to distinguish between high an low frequency content of the image recursively at multiple scales; also other representations, such as independent component analysis and principal component analysis, are potentially more efficient as information carriers with respect the quantities here considered. However, most of these methods require more computation: a trade-off between index complexity and needed number of indexes should be carried out to maximize the system performance, minimizing at the same time the overall computational cost of the system. Nevertheless, the selection of the indexes remains dependent on the user choice, and the verification of the effectiveness of a specific combination is a very slow process, requiring a complete training of the system to be properly estimated. In this context, the application of machine learning techniques appears attractive, to automatically obtain an optimal minimal representation suitable for hazard detection in place of the indexes vector. Methods that make use of unsupervised learning processes, such as autoencoders, convolutional neural network and self-organized maps, appear as the most promising options, and are currently under study.

Finally, the next step in system validation will require a deeper testing. A wider test set will be considered, with more terrain types and variations in environmental condition. Plus, the robustness of the system with respect possible image quality degradation will be assessed.

4.1 Roadmap for Future Research

Toward the implementation of a complete GNC chain for autonomous landing, the integration of the two developed systems with a relative navigation algorithm is the most logical next step. The navigation subsystem operates as the link between Hazard Detection and

Adaptive Guidance: the coordinates of the selected target must be identified in the real world, together with the current states of the system with respect to the updated landing site. The Adaptive Guidance module must then be fed with these data to properly compute the new trajectory.

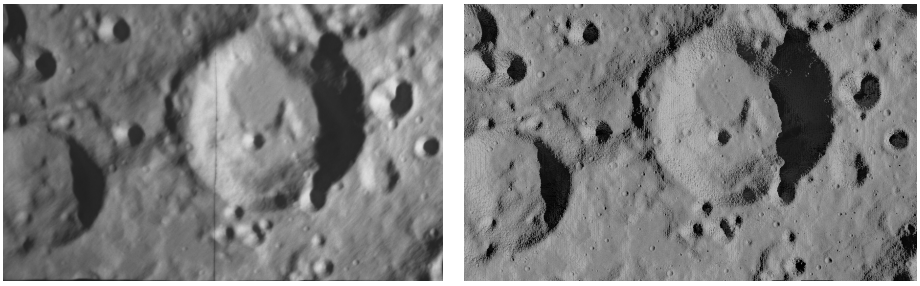
Not included in this work, a parallel research line about the application of vision-based navigation to the spacecraft landing problem has been carried out in last years, and it is still ongoing. Preliminary results indicate that a proper data fusion between different sensors is required, mixing classical measurements (laser/radar altimeter, inertial measurements) with information available by new methods (e.g features tracking by cameras, reconstructed DEM by LIDAR). The geometrical characteristics of the problem (small rotations, and high flatness of the scene, especially in the case of planetary landings maneuvers) make the problem hard to be solved by pure visual information, introducing some mathematical indeterminacies, for the motion of optical features in the camera field of view is similar in both pure translation and pure small rotation. The introduction of a navigation filter is then required.

To prosecute the development of the proposed algorithms towards real applications, an intensive test campaign is required, with both software-in-the-loop and hardware-in-the-loop simulations. Validation campaigns, using Monte Carlo simulations coupled with high fidelity landing simulators, are needed to ensure the functionality of the routines in different conditions. A robust handling of the exceptions must be provided to cope with unexpected event and non nominal conditions.

Several improvements are needed also in prospect of a hardware porting. At the current state, the routines implementation leaves room to significant code optimization. The routines should be tested on a real-time operative system, and a rigorous profiling should be performed to identify bottlenecks and estimate a realistic computation time.

Finally, a fast and repeatable method to generate realistic input is necessary to achieve the required level of reliability. For this purpose, a new landing simulation facility for optical GNC systems is under development at the PoliMi Aerospace Science and Technology Department premises [1].

[1] The author acknowledges the support received by the ESA Directorate of Human Spaceflight



(a) *Real image* **(b)** *Artificial rendering*

Figure 4.1: *PoliMi landing simulation facility. Comparison between a real image and its correspondent artificial rendered with SECRET-PLAN.*

The facility consists in a robotic arm, which carries a suite of sensors (mainly a camera and a range detector simulating a laser altimeter), moving them over a diorama reproducing the lunar landscape. The terrain simulator, coupled with a light system mimicking the sunlight, provides a scaled environment for the simulation of lunar landing maneuvers. The system is designed to work in both open and closed loop, along predefined trajectories or guided by guidance and control algorithms, with the spacecraft dynamics simulated in parallel on a dedicated unit that control the arm pose and velocity. Figure 4.1 shows a comparison between a real image, taken by the facility's camera, and a correspondent benchmark image rendered with SECRET-PLAN. The simulator is currently entering a phase of integration and functional tests before the beginning of the operations, scheduled by the end of January 2017.

and Robotic Exploration, and by the Space Robotics Programs Office of Leonardo-Finmeccanica in the setting of the landing simulation facility. The activity is carried out under ESA contract.

Bibliography

[1] "Schiapparelli descent data: decoding underway - ESA press release," http://exploration.esa.int/mars/58475-schiaparelli-descent-data-decoding-underway/. Last visit: 10th Nov 2016.

[2] J. L. Vago, O. G. Witasse, H. Svedhem, P. Baglioni, A. Haldemann, G. Gianfiglio, T. Blancquaert, D. McCoy, and R. de Groot, "ESA ExoMars program: The next step in exploring Mars," *Solar System Research*, vol. 49, no. 7, pp. 518–528, 2015, DOI: 10.1134/S0038094615070199.

[3] U. Derz and W. Seboldt, "Mars sample return mission architectures utilizing low thrust propulsion," *Acta Astronautica*, vol. 77, pp. 83–96, 2012, DOI: 10.1016/j.actaastro.2012.03.017.

[4] G. A. Soffen, "The viking project," *Journal of Geophysical Research*, vol. 82, no. 28, pp. 3959–3970, 1977, DOI: 10.1029/JS082i028p03959.

[5] D. A. Spencer, R. C. Blanchard, R. D. Braun, P. H. Kallemeyn, and S. W. Thurman, "Mars Pathfinder Entry, Descent, and Landing reconstruction," *Journal of Spacecraft and Rockets*, vol. 36, no. 3, pp. 357–366, 1999, DOI: 10.2514/2.3478.

[6] R. Roncoli and J. Ludwinski, "Mission design overview for the Mars Exploration Rover mission," in *AIAA Guidance, Navigation, and Control Conference*, Monterey, CA, Aug. 2002, DOI: 10.2514/6.2002-4823. AIAA Paper 2002-4823.

[7] J. Crisp, M. Adler, J. Matijevic, S. Squyres, R. Arvidson, and D. Kass, "Mars Exploration Rover mission," *Journal of Geophysical Research E: Planets*, vol. 108, no. 12, pp. ROV 2–1–ROV 2–17, 2003.

[8] P. N. Desai, J. L. Prince, E. M. Queen, M. M. Schoenenberger, J. R. Cruz, and M. R. Grover, "Entry, descent, and landing performance of the Mars Phoenix Lander," *Journal of Spacecraft and Rockets*, vol. 48, no. 5, pp. 798–808, 2011, DOI: 10.2514/1.48239.

[9] J. L. Prince, P. N. Desai, E. M. Queen, and M. R. Grover, "Mars Phoenix Entry, Descent, and Landing simulation design and modeling analysis," *Journal of Spacecraft and Rockets*, vol. 48, no. 5, pp. 756–764, 2011, DOI: 10.2514/1.46561.

[10] E. M. Queen, J. L. Prince, and P. N. Desai, "Multibody modeling and simulation for Mars Phoenix Entry, Descent, and Landing," *Journal of Spacecraft and Rockets*, vol. 48, no. 5, pp. 765–771, 2011, DOI: 10.2514/1.46918.

[11] R. C. Blanchard and P. N. Desai, "Mars Phoenix Entry, Descent, and Landing trajectory and atmosphere reconstruction," *Journal of Spacecraft and Rockets*, vol. 48, no. 5, pp. 809–822, 2011, DOI: 10.2514/1.46274.

[12] M. R. Grover, B. D. Cichy, and P. N. Desai, "Overview of the Phoenix Entry, Descent, and Landing system architecture," *Journal of Spacecraft and Rockets*, vol. 48, no. 5, pp. 706–712, 2011, DOI: 10.2514/1.46548.

[13] R. Prakash, P. D. Burkhart, A. Chen, K. A. Comeaux, C. S. Guernsey, D. M. Kipp, L. V. Lorenzoni, G. F. Mendeck, R. W. Powell, T. P. Rivellini, A. M. S. Martin, S. W. Sell, A. D. Steltzner, and D. W. Way, "Mars Science Laboratory Entry, Descent, and Landing system overview," in *IEEE Aerospace Conference 2008*, pp. 1–18, Big Sky, MT, Mar. 2008, DOI: 10.1109/AERO.2008.4526283.

[14] M. S. Martin, G. F. Mendeck, P. B. Brugarolas, G. Singh, F. Serricchio, S. W. Lee, E. C. Wong, and J. C. Essmiller, "In-flight experience of the Mars Science Laboratory Guidance, Navigation, and Control system for Entry, Descent, and Landing," *CEAS Space Journal*, vol. 7, no. 2, pp. 119–142, 2015, DOI: 10.1007/s12567-015-0091-3.

[15] A. A. Gonzales and C. R. Stoker, "An efficient approach for Mars Sample Return using emerging commercial capabilities," *Acta Astronautica*, vol. 123, pp. 16–25, 2016, DOI: 10.1016/j.actaastro.2016.02.013.

[16] C. E. Singer, "NASA's Space Launch System marks critical design review," in *SpaceOps Conferences*, Daejeon, Korea, May 2016, DOI: 10.2514/6.2016-2529. AIAA Paper 2016-2529.

[17] A. A. Schorr and S. D. Creech, "Space Launch System spacecraft and payload elements: Making progress toward first launch," in *AIAA SPACE Forum*, Long Beach, CA, sep 2016, DOI: 10.2514/6.2016-5418. AIAA Paper 2016-5418.

[18] P. F. Marshall and S. D. Norris, "Orion project status," in *AIAA SPACE 2013 Conference and Exposition*, San Diego, CA, Sep. 2013, DOI: 10.2514/6.2013-5476. AIAA Paper 2013-5476.

[19] J. Gutkowski, T. Dawn, and R. Jedrey, "Evolution of Orion mission design for exploration mission 1 and 2," in *39th Annual AAS Rocky Mountain Section Guidance and Control Conference*, vol. 157 of *Advances in the Astronautical Sciences*, pp. 697–708, Univelt, San Diego, CA, Feb. 2016.

[20] J.-M. Salotti, "Robust, affordable, semi-direct Mars mission," *Acta Astronautica*, vol. 127, pp. 235–248, 2016, DOI: 10.1016/j.actaastro.2016.06.004.

[21] E. Musk, "Making humans a multiplanetary species," http://www.spacex.com/mars (last visit: 29th Dec. 2016), Sep. 2016. Special keynote at 67th International Astronautical Congress (IAC), Guadalajara, Mexico.

[22] B. Donahue, D. Burks, and D. Cooper, "Exploration opportunities enabled by the space launch system," in *2016 IEEE Aerospace Conference*, pp. 1–11, Big Sky, MT, Mar. 2016, DOI: 10.1109/AERO.2016.7500788.

[23] J. D. Carpenter, R. Fisackerly, D. De Rosa, and B. Houdou, "Scientific preparations for lunar exploration with the European Lunar Lander," *Planetary and Space Science*, vol. 74, no. 1, pp. 208–223, 2012, DOI: 10.1016/j.pss.2012.07.024.

[24] B. Gardini, L. Zelenyi, V. Khartov, A. Lukiyanchikov, I. Mitrofanov, A. Pradier, C. Philippe, and J. Carpenter, "ESA-Roscosmos strategy for moon exploration," in *62nd International Astronautical Congress (IAC)*, Napoli, Italy, Sep. 2012.

[25] B. Gardini, "OUTLOOK after the CM 2012," in *European Commission Workshop on Space Science and Exploration*, Madrid, Spain, Feb. 2013.

[26] R. Fisackerly, J. Carpenter, G. Visentin, B. Houdou, B. Patti, F. Rizzi, P. Magnani, and A. Fumagalli, "Accessing, drilling and operating at the Lunar south pole: Status of European plans and activities," in *Earth and Space 2014: Engineering*

for Extreme Environments - Proceedings of the 14th Biennial International Conference on Engineering, Science, Construction, and Operations in Challenging Environments, pp. 237–246, 2014, DOI: 10.1061/9780784479179.027.

[27] C. Li, J. Liu, X. Ren, W. Zuo, X. Tan, W. Wen, H. Li, L. Mu, Y. Su, H. Zhang, J. Yan, and Z. Ouyang, "The Chang'e 3 mission overview," *Space Science Reviews*, vol. 190, no. 1, pp. 85–101, 2015, DOI: 10.1007/s11214-014-0134-7.

[28] P. Chauhan, A. Bhardwaj, S. Kumar, P. Kaur, and N. Bhandari, "Understanding our celestial neighbors: An Indian perspective in planetary sciences and exploration," *Proceedings of the Indian National Science Academy*, vol. 82, no. 3, pp. 403–423, 2016.

[29] "Catching a Comet," *Science Special Issue*, vol. 347, Jan. 2015.

[30] K. Geurts, C. Fantinati, S. Ulamec, and R. Willnecker, "Rosetta lander: On-comet operations preparation and planning," in *SpaceOps 2014 Conference*, Pasadena, CA, May 2014, DOI: 10.2514/6.2014-1752. AIAA Paper 2014-1752.

[31] J. Gal-Edd and A. Cheuvront, "The OSIRIS-REx asteroid sample return: Mission operations design," in *13th International Conference on Space Operations*, Pasadena, CA, May 2014, DOI: 10.2514/6.2014-1721. AIAA Paper 2014-1721.

[32] P. Michel, M. A. Barucci, A. F. Cheng, H. Böhnhardt, J. R. Brucato, E. Dotto, P. Ehrenfreund, I. A. Franchi, S. F. Green, L.-M. Lara, B. Marty, D. Koschny, and D. Agnolon, "MarcoPolo-R: Near-earth asteroid sample return mission selected for the assessment study phase of the ESA program cosmic vision," *Acta Astronautica*, vol. 93, no. 0, pp. 530–538, 2014, DOI: 10.1016/j.actaastro.2012.05.030.

[33] D. Reeves, D. Mazanek, B. Cichy, S. Broschart, and K. DeWeese, "Asteroid redirect mission proximity operations for reference target asteroid 2008 EV5," in *39th Annual AAS Rocky Mountain Section Guidance and Control Conference*, vol. 157, pp. 655–666, San Diego, CA, 2016.

[34] G. L. Condon and J. Williams, "Asteroid Redirect crewed mission nominal design and performance," in *SpaceOps 2014 Conference*, Pasadena, CA, May 2014, DOI: 10.2514/6.2014-1696. AIAA Paper 2014-1696.

[35] A. F. Cheng, "AIDA: Test of asteroid deflection by spacecraft impact," in *44th Lunar and Planetary Science Conference*, The Woodlands, TX, Mar. 2013.

[36] A. Galvez, I. Carnelli, M. Fontaine, and C. Corral Van Damme, "Asteroid Impact Mission (AIM) & Deflection assessment," in *European Planetary Science Congress 2012*, Madrid, Spain, Sep. 2012.

[37] A. Galvez, I. Carnelli, M. Khan, W. Martens, P. Michel, S. Ulamec, and A. Hriscu, "Asteroid investigation mission: the European contribution the the AIDA EU-US cooperation," in *24th International Symposium on Space Flight Dynamics*, Laurel, MD, May 2014.

[38] M. Küppers, I. Carnelli, A. Galvez, K. Mellab, and P. Michel, "The Asteroid Impact Mission (AIM)," in *European Planetary Science Congress 2015*, Nantes, France, Sep. 2015.

[39] "ESA council at ministerial level 2016: success, tinged with a bit of disappointment," http://blogs.esa.int/janwoerner/2016/12/04/esa-council-at-ministerial-level-2016-success-tinged-with-a-bit-of-disappointment/. Last visit: 20th Dec 2016.

[40] H. Masursky and N. L. Crabill, "The Viking landing sites: Selection and certification," *Science*, vol. 193, no. 4255, pp. 809–812, 1976, DOI: 10.1126/science.193.4255.809.

[41] R. Arvidson, D. Adams, G. Bonfiglio, P. Christensen, S. Cull, M. Golombek, J. Guinn, E. Guinness, T. Heet, R. Kirk, A. Knudson, M. Malin, M. Mellon, A. McEwen, A. Mushkin, T. Parker, F. Seelos, K. Seelos, P. Smith, D. Spencer, T. Stein, and L. Tamppari, "Mars Exploration Program 2007 Phoenix landing site selection and characteristics," *Journal of Geophysical Research: Planets*, vol. 113, no. E3, 2008, DOI: 10.1029/2007JE003021.

[42] D. A. Spencer, D. S. Adams, E. Bonfiglio, M. Golombek, R. Arvidson, and K. Seelos, "Phoenix landing site hazard assessment and selection," *Journal of Spacecraft and Rockets*, vol. 46, no. 6, pp. 1196–1201, 2009, DOI: 10.2514/1.43932.

[43] M. Golombek, J. Grant, D. Kipp, A. Vasavada, R. Kirk, R. Fergason, P. Bellutta, F. Calef, K. Larsen, Y. Katayama, A. Huertas, R. Beyer, A. Chen, T. Parker, B. Pollard, S. Lee, Y. Sun, R. Hoover, H. Sladek, J. Grotzinger, R. Welch, E. Noe Dobrea, J. Michalski, and M. Watkins, "Selection of the Mars Science Laboratory landing site," *Space Science Reviews*, vol. 170, no. 1, pp. 641–737, 2012, DOI: 10.1007/s11214-012-9916-y.

[44] S. Ulamec, J. Biele, A. Blazquez, B. Cozzoni, C. Delmas, C. Fantinati, P. Gaudon, K. Geurts, E. Jurado, O. Küchemann, V. Lommatsch, M. Maibaum, H. Sierks, and L. Witte, "Rosetta lander – Philae: Landing preparations," *Acta Astronautica*, vol. 107, pp. 79–86, 2015, DOI: 10.1016/j.actaastro.2014.11.019.

[45] A. Accomazzo, S. Lodiot, and V. Companys, "Rosetta mission operations for landing," *Acta Astronautica*, vol. 125, pp. 30–40, 2016, DOI: 10.1016/j.actaastro.2016.01.013. Rosetta and Philae at comet 67P/Churyumov-Gerasimenko.

[46] L. Witte, R. Roll, J. Biele, S. Ulamec, and E. Jurado, "Rosetta lander Philae – landing performance and touchdown safety assessment," *Acta Astronautica*, vol. 125, pp. 149–160, 2016, DOI: 10.1016/j.actaastro.2016.02.001. Rosetta and Philae at comet 67P/Churyumov-Gerasimenko.

[47] K. Berry, B. Sutter, A. May, K. Williams, B. Barbee, M. Beckman, and B. Williams, "OSIRIS-REx touch-and-go (TAG) mission design and analysis," in *36th Annual AAS Rocky Mountain Section Guidance and Control Conference*, vol. 149 of *Advances in the Astronautical Sciences*, pp. 667–678, Univelt, San Diego, CA, Feb. 2013.

[48] S. Ulamec, C. Fantinati, M. Maibaum, K. Geurts, J. Biele, S. Jansen, O. Küchemann, B. Cozzoni, F. Finke, V. Lommatsch, A. Moussi-Soffys, C. Delmas, and L. O'Rourke, "Rosetta lander – landing and operations on comet 67P/Churyumov–Gerasimenko," *Acta Astronautica*, vol. 125, pp. 80–91, 2016, DOI: 10.1016/j.actaastro.2015.11.029.

[49] E. Remetean, B. Dolives, F. Souvannavong, T. Germa, J. Ginestet, A. Torres, and T. Mousset, "Philae locating and science support by robotic vision techniques," *Acta Astronautica*, vol. 125, pp. 161–173, 2016, DOI: 10.1016/j.actaastro.2015.12.005.

[50] "CRS-8 launch and landing - SpaceX press release," http://www.spacex.com/news/2016/04/09/crs-8-launch-and-landing. Last visit: 10th Nov 2016.

[51] B. Frapard and S. Mancuso, "Vision navigation for european landers and the NPAL project," in *GNC 2005: 6th International ESA Conference on Guidance, Navigation & Control Systems*, Loutraki, Greece, Oct. 2005.

[52] G. P. Guizzo, R. Drai, N. Despré, and M. Manzano Jurado, "Flight tests results of the precision landing GNC test facility," in *GNC 2011: 8th International ESA Conference on Guidance, Navigation & Control Systems*, Karlovy Vary, Czech Republic, Jun. 2011.

[53] G. Hormigo, C. Gu, T. Lutz, S. Gardecki, F. Cordes, A. Böljes, J. Bolz, and P. De Maagt, "Startiger Dropter project: integrated closed-loop vision-aided navigation with hazard detection and avoidance," in *GNC 2014: 9th International ESA conference on Guidance, Navigation & Control Systems*, Porto, Portugal, Jun. 2014.

[54] S. Steffes, M. Dumke, D. Heise, M. Sagliano, M. Samaan, S. Theil, E. Boslooper, H. Oosterling, J. Schulte, D. Skaborn, S. Söderholm, S. Conticello, M. Esposito, Y. Yanson, B. Monna, F. Stelwagen, and R. Visee, "Target relative navigation results from hardware-in-the-loop tests using the Sinplex navigation system," in *AAS/AIAA Guidance, Navigation, and Control 2014*, vol. 151 of *Advances in the Astronautical Sciences*, pp. 171–184, Univelt, San Diego, CA, 2014.

[55] J. Montaño, P. Molina, E. Angelats, I. Colomina, and A. Latorre, "PERIGEO: image processing techniques development & testing for multimodal absolute & relative navigation," in *GNC 2014: 9th International ESA conference on Guidance, Navigation & Control Systems*, Porto, Portugal, Jun. 2014.

[56] N. Trawny, J. M. Carson, A. Huertas, M. E. Luna, V. E. Robak, A. E. Johnson, K. E. Martin, and C. Y. Villalpando, "Helicopter flight testing of a real-time hazard detection system for safe lunar landing," in *AIAA SPACE 2013 Conference and Exposition*, San Diego, CA, 2013, DOI: 10.2514/6.2013-5313. AIAA Paper 2013-5313.

[57] J. M. Carson, N. Trawny, E. Robertson, V. E. Roback, D. Pierrottet, J. Devolites, J. Hart, and J. N. Estes, "Preparation and integration of ALHAT precision landing technology for Morpheus flight testing," in *AIAA SPACE 2014 Conferences and Exposition*, San Diego, CA, Aug. 2014, DOI: 10.2514/6.2014-4313. AIAA Paper 2014-4313.

[58] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, vol. 2, pp. 1403–1410, Oct. 2003, DOI: 10.1109/ICCV.2003.1238654.

[59] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 225–234, 2007.

[60] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *IEEE International Conference on Robotics and Automation*, pp. 15–22, 2014, DOI: 10.1109/ICRA.2014.6906584.

[61] R. Mur-Artal, J. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, 2015, DOI: 10.1109/TRO.2015.2463671.

[62] A. R. Klumpp, "Apollo lunar descent guidance," *Automatica*, vol. 10, no. 2, pp. 133–146, 1974, DOI: 10.1016/0005-1098(74)90019-3.

[63] I. Gerth and E. Mooij, "Guidance for autonomous precision landing on atmosphereless bodies," in *AIAA Guidance, Navigation, and Control Conference*, National Harbor, MD, Jan. 2014, DOI: 10.2514/6.2014-0088. AIAA Paper 2014-0088.

[64] B. Açikmeşe and S. R. Ploen, "Convex programming approach to powered descent guidance for Mars landing," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 5, pp. 1353–1366, 2007, DOI: 10.2514/1.27553.

[65] L. Blackmore, B. Açikmeşe, and D. P. Scharf, "Minimum-landing-error powered-descent guidance for Mars landing using convex optimization," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 4, pp. 1161–1171, 2010, DOI: 10.2514/1.47202.

[66] P. Lunghi, M. Lavagna, and R. Armellin, "A semi-analytical guidance algorithm for autonomous landing," *Advances in Space Research*, vol. 55, no. 11, pp. 2719–2738, 2015, DOI: 10.1016/j.asr.2015.02.022.

[67] P. Lunghi, M. Ciarambino, and M. Lavagna, "A multilayer perceptron hazard detector for vision-based autonomous planetary landing," *Advances in Space Research*, vol. 58, no. 1, pp. 131–144, 2016, DOI: 10.1016/j.asr.2016.04.012.

[68] P. Lunghi, R. Armellin, P. Di Lizia, and M. Lavagna, "Semi-analytical adaptive guidance computation based on differential algebra for autonomous planetary landing," in *26th AAS/AIAA Space Flight Mechanics Meeting*, vol. 158 of *Advances in the Astronautical Sciences*, pp. 2003–2022, Univelt, San Diego, CA, Jan. 2016.

[69] P. Lunghi, M. Lavagna, and R. Armellin, "Semi-analytical adaptive guidance algorithm for fast retargeting maneuvers computation during planetary descent and landing," in *12th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, ESA/ESTEC, Noordwijk, The Netherlands, May 2013.

[70] P. Lunghi, M. Lavagna, and R. Armellin, "Adaptive semi-analytical guidance for autonomous planetary landing," in *64th International Astronautical Congress (IAC)*, Beijing, China, Oct. 2013.

[71] P. Lunghi, M. Lavagna, and R. Armellin, "Semi-analytical guidance algorithm for autonomous close approach to non-cooperative low-gravity targets," in *24th AAS/AIAA Space Flight Mechanics Meeting*, vol. 152 of *Advances in the Astronautical Sciences*, pp. 731–746, Univelt, San Diego, CA, Jan. 2014.

[72] P. Lunghi and M. Lavagna, "A neural network based hazard detection algorithm for planetary landing," in *9th International ESA Conference on Guidance, Navigation, and Control Systems*, Porto, Portugal, Jun. 2014.

[73] P. Lunghi, M. Ciarambino, and M. Lavagna, "Vision-based hazard detection with artificial neural networks for autonomous planetary landing," in *13th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, ESA/ESTEC, Noordwijk, The Netherlands, May 2015.

[74] P. Lunghi, M. Ciarambino, and M. Lavagna, "A multilayer perceptron hazard detector for vision-based autonomous planetary landing," in *AAS/AIAA Astrodynamics Specialist Conference 2015*, vol. 156 of *Advances in the Astronautical Sciences*, pp. 1633–1650, Univelt, San Diego, CA, Aug. 2015.

[75] P. Lunghi, M. Ciarambino, and M. Lavagna, "Simulation facility for vision-based planetary landing systems," in *23th Conference of Italian Association of Aeronautics and Astronautics - AIDAA*, Nov. 2015.

[76] P. Lunghi, M. Ciarambino, and M. Lavagna, "A new test facility for vision-based hazard detection and avoidance systems for planetary landing maneuvers," in *66th International Astronautical Congress (IAC)*, Jerusalem, Israel, Oct. 2015.

[77] P. Lunghi, M. Ciarambino, and M. Lavagna, "A hazard detection and avoidance system for autonomous planetary landing," in *23th Conference of Italian Association of Aeronautics and Astronautics - AIDAA*, Torino, Italy, Nov. 2015.

[78] P. Lunghi, M. Ciarambino, L. Losi, and M. Lavagna, "Development, validation and test of optical based algorithms for autonomous planetary landing," in *6th International Conference on Astrodynamics Tools and Techniques (ICATT)*, Darmstadt, Germany, Mar. 2016.

[79] E. Wong, J. Masciarelli, and G. Singh, "Autonomous guidance and control design for hazard avoidance and safe landing on Mars," in *AIAA Atmospheric Flight Mechanics Conference*, Monterey, CA, Aug. 2002, DOI: 10.2514/6.2002-4619. AIAA Paper 2002-4619.

[80] B. Parreira, P. Rogata, E. Di Sotto, A. Caramagno, J. M. Rebordao, P. Motrena, and S. Mancuso, "Consolidated performance assessment of hazard avoidance techniques for vision based landing," in *AIAA Guidance, Navigation, and Control Conference*, Hilton Head, SC, Aug. 2007, DOI: 10.2514/6.2007-6854. AIAA Paper 2007-6854.

[81] U. Topcu, J. Casoliva, and K. D. Mease, "Fuel efficient powered descent guidance for Mars landing," in *AIAA Guidance, Navigation, and Control Conference*, San Francisco, CA, Aug. 2005, DOI: 10.2514/6.2005-6286. AIAA Paper 2005-6286.

[82] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193–207, 1998, DOI: 10.2514/2.4231.

[83] F. Fahroo and I. M. Ross, "Direct trajectory optimization by a Chebyshev pseudospectral method," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 1, pp. 160–166, 2002, DOI: 10.2514/2.4862.

[84] M. A. Lara and D. Scheeres, "Stability bounds for three-dimensional motion close to asteroids," *Journal of the Astronautical Sciences*, vol. 50, no. 4, pp. 389–409, 2003.

[85] M. Hawkins, Y. Guo, and B. Wie, "ZEM/ZEV feedback guidance application to fuel-efficient orbital maneuvers around an irregular-shaped asteroid," in *AIAA Guidance, Navigation, and Control Conference*, Minneapolis, MN, Aug. 2012, DOI: 10.2514/6.2012-5045. AIAA Paper 2012-5045.

[86] R. Furfaro, B. Gaudet, D. R. Wibben, and J. Simo, "Development of non-linear guidance algorithms for asteroids close-proximity operations," in *AIAA Guidance, Navigation, and Control Conference*, Boston, MA, 19–22 August 2013, DOI: 10.2514/6.2013-4711. AIAA Paper 2013-4711.

[87] R. Furfaro, D. Cersosimo, and D. Wibben, "Asteroid precision landing via multiple sliding surfaces guidance techniques," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 4, pp. 1075–1092, 2013, DOI: 10.2514/1.58246.

[88] R. Pinson and P. Lu, "Rapid generation of optimal asteroid powered trajectories via convex optimization," in *AAS/AIAA Astrodynamics Specialist Conference 2015*, vol. 156 of *Advances in the Astronautical Sciences*, Univelt, San Diego, CA, Aug. 2015.

[89] R. Fisackerly, A. Pradier, B. Gardini, B. Houdou, C. Philippe, D. De Rosa, and J. Carpenter, "The ESA Lunar Lander mission," in *AIAA SPACE 2011 Conference and Exposition*, Long Beach, CA, Sep. 2011, DOI: 10.2514/6.2011-7217. AIAA Paper 2011-7217.

[90] H. Ellerbrock, N. Henn, G. Hagemann, M. Riehle, and M. Kinnersley, "Chemical propulsion challenges for ESA's exploration missions and automatic return vehicle," in *47th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, San Diego, CA, jul 2011, DOI: 10.2514/6.2011-5767. AIAA Paper 2011-5767.

[91] G. Flandin, B. Polle, N. Despré, J. Lheritier, N. Perrimon, and P. Blanc-Paques, "Maturing vision based navigation solutions to space exploration," in *AIAA Guidance, Navigation, and Control Conference*, Toronto, Ontario Canada, Aug. 2010, DOI: 10.2514/6.2010-7601. AIAA Paper 2010-7601.

[92] J. Riedel, A. Vaughan, R. A. Werner, W. Tseng-Chan, S. Nolet, D. Myers, N. Mastrodemos, A. Lee, C. Grasso, T. Ely, and D. Bayard, "Optical navigation plan and strategy for the lunar lander Altair; OpNav for lunar and other crewed and robotic exploration applications," in *AIAA Guidance, Navigation, and Control Conference*, Toronto, Ontario Canada, Aug. 2010, DOI: 10.2514/6.2010-7719. AIAA Paper 2010-7719.

[93] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics*, Springer, New York, 1988.

[94] D. J. Scheeres, "Dynamics about uniformly rotating triaxial ellipsoids: Applications to asteroids," *Icarus*, vol. 110, no. 2, pp. 225–238, 1994, DOI: 10.1006/icar.1994.1118.

[95] Y. Tsuda, M. Yoshikawa, M. Abe, H. Minamino, and S. Nakazawa, "System design of the Hayabusa 2—asteroid sample return mission to 1999 JU3," *Acta Astronautica*, vol. 91, pp. 356–362, 2013, DOI: 10.1016/j.actaastro.2013.06.028.

[96] F. Ferrari, M. Lavagna, and K. C. Howell, "Dynamical model of binary asteroid systems through patched three-body problems," *Celestial Mechanics and Dynamical Astronomy*, vol. 125, no. 4, pp. 413–433, 2016, DOI: 10.1007/s10569-016-9688-x.

[97] R. A. Werner and D. J. Scheeres, "Exterior gravitation of a polyhedron derived and compared with harmonic and mascon gravitation representations of asteroid 4769 Castalia," *Celestial Mechanics and Dynamical Astronomy*, vol. 65, no. 3, pp. 313–334, 1996, DOI: 10.1007/BF00053511.

[98] T. Kolda, R. Lewis, and V. Torczon, "Optimization by direct search: New perspectives on some classical and modern methods," *SIAM Review*, vol. 45, no. 3, pp. 385–482, 2003, DOI: 10.1137/S003614450242889.

[99] M. Berz, *Modern Map Methods in Particle Beam Physics*, Academic Press, London, 1999.

[100] J. Delaune, D. De Rosa, and S. Hobbs, "Guidance and control system design for lunar descent and landing," in *AIAA Guidance, Navigation, and Control Conference*, Toronto, Ontario Canada, 2010, DOI: 10.2514/6.2010-8028. AIAA Paper 2010-8028.

[101] A. Morselli, M. Rasotto, P. Di Lizia, R. Armellin, M. Massari, and A. Wittig, "Nonlinear propagation of uncertainties in space dynamics based on taylor differential algebra," Tech. Rep. Contract no. 4000109643/13/NL/MH Final Report, European Space Agency, 2015.

[102] M. C. Johnson, "A parameterized approach to the design of lunar lander attitude controllers," in *AIAA Guidance, Navigation, and Control Conference*, Keystone, CO, 2006, DOI: 10.2514/6.2006-6564. AIAA Paper 2006-6564.

[103] H. Pien, "Autonomous hazard detection and avoidance for Mars exploration," in *8th AIAA Computing in Aerospace Conference*, Baltimora, MD, 1991, DOI: 10.2514/6.1991-3731. AIAA Paper 1991-3731.

[104] R. Furfaro, W. Fink, and J. S. Kargel, "Autonomous real-time landing site selection for Venus and Titan using evolutionary fuzzy cognitive maps," *Applied Soft Computing*, vol. 12, no. 12, pp. 3825–3839, 2012, DOI: 10.1016/j.asoc.2012.01.014.

[105] B. Parreira, E. Di Sotto, A. Caramagno, and J. Rebordão, "Hazard avoidance for planetary landing: GNC design and performance assessment," in *GNC 2008: 7th International ESA Conference on Guidance, Navigation & Control Systems*, Tralee, County Kerry, Ireland, Jun. 2008.

[106] S. Woicke and E. Mooij, "Stereo-vision algorithm for hazard detection during planetary landings," in *AIAA Guidance, Navigation, and Control Conference*, National Harbor, MD, Jan. 2014, DOI: 10.2514/6.2014-0272. AIAA Paper 2014-0272.

[107] L. Matthies, A. Huertas, Y. Cheng, and A. Johnson, "Landing hazard detection with stereo vision and shadow analysis," in *AIAA Infotech@Aerospace Conference*, Rohnert Park, CA, May 2007, DOI: 10.2514/6.2007-2835. AIAA Paper 2007-2835.

[108] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991, DOI: 10.1016/0893-6080(91)90009-T.

[109] M. S. Nixon and A. S. Aguado, *Feature Extraction and Image Processing*, Academic Press, London, 3rd ed., 2012.

[110] T. Toutin, "State-of-the-art of geometric correction of remote sensing data: A data fusion perspective," *International Journal of Image and Data Fusion*, vol. 2, no. 1, pp. 3–35, 2011, DOI: 10.1080/19479832.2010.539188.

[111] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, John Wiley & Sons, Inc., New York, 2001.

[112] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998, DOI: 10.1162/089976698300017467.

[113] J. G. Daugman, "Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 7, pp. 1169–1179, Jul 1988, DOI: 10.1109/29.1644.

[114] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," in *Advances in Neural Information Processing Systems 18*, pp. 1161–1168, MIT Press, Cambridge, MA, Dec. 2006.

[115] H. B. Kekre and S. M. Gharge, "Image segmentation using extended edge operator for mammographic images," *International Journal on Computer Science and Engineering*, vol. 2, no. 4, pp. 1086–1091, 2010.

[116] M. S. Nixon and A. S. Aguado, *Feature Extraction and Image Processing*, Academic Press, London, 2nd ed., 2008.

[117] K. Hornik, M. B. Stinchcombe, and H. L. White, "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," *Neural Networks*, vol. 3, no. 5, pp. 551–560, 1990, DOI: 10.1016/0893-6080(90)90005-6.

[118] M. M. Leshno, V. Lin, A. Pinkus, and S. Schocken, "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function," *Neural Networks*, vol. 6, no. 6, pp. 861–867, 1993, DOI: 10.1016/S0893-6080(05)80131-5.

[119] S. E. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," in *Advances in Neural Information Processing Systems 2*, edited by D. S. Touretzky, pp. 524–532, Morgan-Kaufmann, San Francisco, CA, 1990.

[120] H. Yonaba, F. Anctil, and V. Fortin, "Comparing sigmoid transfer functions for neural network multistep ahead streamflow forecasting," *Journal of Hydrologic Engineering*, vol. 15, no. 4, pp. 275–283, 2010, DOI: 10.1061/(ASCE)HE.1943-5584.0000188.

[121] S. Parkes, M. Dunstan, I. Martin, P. Mendham, and S. Mancuso, "Planet surface simulation for testing vision-based autonomous planetary landers," in *57th International Astronautical Congress (IAC)*, 2006.

[122] G. Jonniaux and D. Gherardi, "Robust extraction of navigation data from images for planetary approach and landing," in *GNC 2014: 9th International ESA conference on Guidance, Navigation & Control Systems*, Porto, Portugal, Jun. 2014.

[123] U. J. Shankar, S. Wen-Jong, T. B. Criss, and D. Adams, "Lunar terrain surface modeling for the ALHAT program," in *IEEE Aerospace Conference*, pp. 1–10, Big Sky, MT, Mar. 2008, DOI: 10.1109/AERO.2008.4526300.

[124] C. A. Cross and D. L. Fisher, "The computer simulation of lunar craters," *Monthly Notices of the Royal Astronomical Society*, vol. 139, no. 2, pp. 261–272, 1968, DOI: 10.1093/mnras/139.2.261.

[125] F. Hörz, R. Grieve, G. Heiken, P. Spudis, and A. Binder, *Lunar Surface Processes*, chap. 4, pp. 61–120, Cambridge University Press, Cambridge, England, 1991.

[126] S. Nissen, "Implementation of a Fast Artificial Neural Network library (FANN)," Tech. rep., Department of Computer Science - University of Copenhagen (DIKU), 2003.

[127] G. Capuano, M. Severi, E. Della Sala, R. Ascolese, C. Facchinetti, and F. Longo, "Compact and high-performance equipment for vision-based navigation," in *63rd International Astronautical Congress (IAC)*, Napoli, Italy, Oct. 2012.

[128] M. Dunstan and K. Hornbostel, "Image processing chip for relative navigation for lunar landing," in *9th International ESA Conference on Guidance, Navigation, and Control Systems*, Porto, Portugal, Jun. 2014.

[129] F. Amzajerdian, D. F. Pierrottet, G. D. Hines, L. B. Petway, and B. W. Barnes, "Doppler lidar sensor for precision navigation in GPS-deprived environment," *Proc. SPIE*, vol. 8731, pp. 87310G–87310G–6, 2013, DOI: 10.1117/12.2018359.