der Bundeswehr
Universität München

Holistic design of a GN&C system for safe and precise autonomous landing in very challenging planetary terrains

M.Sc. Konstantinos Konstantinidis

Vollständiger Abdruck der von der Fakultät für Luft- und Raumfahrttechnik der Universität der Bundeswehr München zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender	:	Univ.-Prof. Dr. rer. nat. Matthias Gerdts
1. Berichterstatter	:	Prof. Dr.-Ing. Roger Förstner
2. Berichterstatter	:	Prof. Dr.-Ing. Hakan Kayal

Diese Dissertation wurde am 07.12.2018 bei der Universität der Bundeswehr München eingereicht und durch die Fakultät für Luft- und Raumfahrttechnik am 06.04.2019 angenommen.

Tag der Prüfung: 12.04.2019

Για το Νικόλα και την Ελένη

- Can you fly this plane and land it?
- Surely you can't be serious?
- I am serious. And don't call me Shirley.
Airplane!, 1980

Acknowledgements

The completion of this thesis would not have been possible without the people who helped and supported me during its writing.

I would like to thank my supervisor Prof. Roger Förstner for giving me the opportunity to work on such a fascinating project, his hands-off and wise guidance, and for instilling in me the principles of systems engineering. I would also like to thank my colleagues at ISTA for creating a warm and supportive working environment, and for their help during the writing of this thesis. I would like to thank in particular Graciela Gonzalez Peytavi for her extensive help on the subject of navigation and Robert Jakob for enlightening discussions on feature detection. I am also grateful to the students of Bundeswehr University that supported this project with their theses.

This work is dedicated to the memory of my father Nikos, who would have been very proud of his "dottore", to my mother Eleni, who imparted me her love of learning, and to my brother George who always inspired me with his strength of character. This achievement is owed to their unconditional love and support throughout the years. A big thank you goes out to the friends I had the luck to meet in Munich. You made my stay here unforgettable.

Finally, I thank Sagris for her love, patience, and caring support in the long days of work during the writing of this thesis.

Kurzfassung

Die Suche nach Leben im Sonnensystem ist eine der Hauptantriebskräfte für die Planetenforschung. Die Eismonde der Riesenplaneten sind die vielversprechendsten Ziele. Vor allem der Saturnmond Enceladus besitzt einen Ozean, der zunehmend die Voraussetzungen für das Entstehen mikrobiellen Lebens schafft. Darüber hinaus strömen Geysire aus seinem Südpol, die Wasser direkt aus dem Ozean in den Weltraum schleudern. Durch Wasserentnahmen unter den Geysiren, bevor mögliche Biosignaturen durch das Vakuum des Weltraums abgebaut werden, erhofft man sich vielverprechende Erkenntnisse.

Das vom Deutschen Zentrum für Luft- und Raumfahrt (DLR) geförderte Landerkonzept Enceladus Explorer (EnEx) zielt darauf ab, eine Schmelzsonde einzusetzen, um Flüssigkeitseinschlüsse unter den Geysiren zu untersuchen. Die Landung dort wäre aufgrund der rauen Canyon-Topographie, der polaren Lichtverhältnisse, der mit einer Schicht feinsten Schnees bedeckten Oberfläche und der strengen Planetary-Protection-Regeln außerordentlich schwierig. Jede Landung dort muss daher genau sein, um in der Nähe der Fahne für den Einsatz der Sonde zu landen, sicher, um alle oben genannten Gefahren zu erkennen und zu vermeiden, und autonom aufgrund der langen Signalrücklaufzeiten von der Erde zum Saturn. Frühere Landeeinsätze und -studien zeichnen sich durch blinde Landungen mit geringen Genauigkeitauf sicherem Gelände aus. Aktuelle Studien für eine autonome, sichere und genaue Landung konzentrieren sich auf den Mond und Mars und zielen auf vergelichsweise zu EnEx-Lander weniger herausfordernde Terrains ab.

In dieser Arbeit wird die kritischste Endphase der Landung untersucht. Ein detailliertes System- und Betriebskonzept für die Guidance, Navigation, und Control (GN&C) wird definiert, das die notwendigen Schlüsselfunktionen umfasst. Für das Sensing wird ein Satz interozeptiver inertialer und exterozeptiver Kameras sowie Lidar verwendet. Basierend auf den Eingaben dieser Sensoren wird ein EKF-SLAM-Ansatz verfolgt, um den Lander zu lokalisieren und die Umgebung abzubilden. Bei der Gefahrenerkennung und -vermeidung (Hazard Detection and Avoidance, HDA) verwendet ein Fuzzy-Logik-Ansatz den unsicheren Sensoreingang, um Gefahren zu erkennen und zu vermeiden. Für die Führung wird die konvexe Optimierung verwendet, um eine realisierbare optimale Trajektorie zum Ziellandeplatz zu berechnen.

Um die Landefähigkeit zu überprüfen und das Konzept weiter zu verfeinern, wurde ein Tool zur Landungssimulation entwickelt. Jede Funktion wurde separat validiert und analysiert, indem der jeweilige Block im Werkzeug mit Sensitivitäts- und Worst-Case-Analysen verwendet wurde. Aus den Ergebnissen wurden Kenntisse und Vorschläge für den System-optimierung abgeleitet.

Es wurden Schlussfolgerungen und Vorschläge für Änderungen und Verbesserungen gemacht. Schließlich wurde eine Monte-Carlo-Simulation mit allen Funktionen durchgeführt, um zu überprüfen, ob die Funktionen in einem vereinfachten geschlossenen Regelkreis ordnungsgemäß funktionieren, und um einige erste Erfolgsstatistiken zu schätzen. Die Ergebnisse wurden abgeschlossen und kommentiert und es werden Schritte für die nahe und ferne zukünftige Arbeit zur Erweiterung vorgeschlagen.

Abstract

The search for life in the solar system is one of the main driving forces behind planetary exploration. The icy moons of the giant planets are the most promising targets. Saturn's moon Enceladus in particular possesses an ocean increasingly likely to have the right conditions for microbial life to emerge. Furthermore, plumes are jetting from its south pole ejecting water directly from the ocean to space. The ability to sample water from under the plumes, before any potential biosignatures are degraded by the vacuum of space would be very desirable.

The German Aerospace Administration (DLR) funded Enceladus Explorer (EnEx) landing mission concept aims to deploy a melting probe to sample liquid pockets under the plumes for life. Landing there would be exceptionally challenging due to the rough canyonous topography, polar lighting conditions, the surface covered by a layer of superfine snow, and the strict planetary protection regulations. Any landing there must thus be accurate, to land near the plume for probe deployment, safe, to detect and avoid all the above hazards, and autonomous due to the long signal return times from Earth to Saturn. Past landing missions and studies, have landed blindly and with low accuracy on safe and flat terrains. Current studies for autonomous safe and accurate landing focus on the Moon or Mars, and target terrains that are still not as challenging as the one for the EnEx lander.

In this work the most critical final phase of landing is investigated. A detailed landing Guidance, Navigation, and Control (GN&C) system and operations concept is defined, encompassing the key functions necessary. For Sensing, a set of interoceptive inertial and exteroceptive cameras and lidar are used. Based on input from these sensors, an EKF-SLAM approach is followed to localize the lander and map the environment. For Hazard Detection and Avoidance (HDA), a fuzzy reasoning approach uses the uncertain sensor input to detect hazards and avoid them. For Guidance, convex optimization is used to calculate a feasible optimal trajectory to the target landing site.

To verify the landing feasibility and further refine the concept, a landing simulation tool was created. Each function was separately validated and analyzed using the respective block in the tool with sensitivity and worst case analyses. Conclusions and proposals for modifications and improvements were made. Finally, a closed loop Monte Carlo simulation with all functions was performed to verify that the functions work properly in a simplified closed loop and to estimate some initial mission success statistics. The results were concluded and commented and a steps for near and far future work to expand this are proposed.

Contents

Nomenclature

List of Acronyms

AG	Approach Gate
ALHAT	Autonomous precision Landing and Hazard Avoidance Technology
BN	Bayesian Network
BP	Bayesian Probability theory
COBALT	CoOperative Blending of Autonomous Landing Technologies
DAG	Directed Acyclic Graph
DCA	Downrange-Crossrange-Altitude
DEM	Digital Elevation Map
DLR	Deutsches Zentrum für Luft- und Raumfahrt
DOF	Degree Of Freedom
DOI	Descent Orbit Insertion
DS	Dempster-Shafer belief theory
DTM	Digital Terrain Model
EKF	Extended Kalman Filter
EL	Enceladus Lander
EnEx	Enceladus Explorer
ESA	European Space Agency
ESA LL	ESA Lunar Lander
FALCON	FSD optimAL CONtrol tool
FN	False Negative
FOV	Field of View
FP	False Positive
FRB-HDA	Fuzzy Rule Based HDA

FSD	Institute of Flight System Dynamics (TU-München)
GCA	Ground Coverage Area
GNC	Guidance, Navigation and Control
GSD	Ground Sample Distance
G-FOLD	Guidance for Fuel-Optimal Large Diverts
HD	Hazard Detection
HDA	Hazard Detection and Avoidance
HG	High Gate
HRN	Hazard Relative Navigation
ICP	Iterative Closest Point
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
ISTA	Institute for Space Technology and Space Applications
LCROSS	Lunar Crater Observation and Sensing Satellite
LG	Low Gate
LIDAR	Light Detection and Ranging
LMedSq	Least Median of Squares
Lunar MARE	Lunar Moon Age and Regolith Explorer
MC	Monte Carlo simulation
MEA	Main Engine Assembly
MF	Membership Function
NASA	National Aeronautics and Space Administration
NEP	Nuclear Electric Propulsion
PANGU	Planet and Asteroid Natural Scene Generation Utility
PDI	Powered Descent Initiation
PILOT	Precise and Intelligent Landing using On board Technology
PP	Planetary Protection
RDF	Right-Down-Front
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SOCP	Second Order Cone Problem

SPT	South Polar Terrain
TBD	To be determined
TD	Touchdown
TG	Terminal Gate
TM	True Measurement
ToF	Time-of-flight
TPBVP	Two-Point Boundary-Value Problem
TRL	Technical Readiness Level
TRN	Terrain Relative Navigation
TtW	Thrust-to-weight
UniBw	Bundeswehr University Munich

List of Symbols

General

t	$[s]$	Time
ρ_{planet}	$[\text{mol/m}^3]$	Atmospheric density of planet
ΔV	$[m/s]$	Change in velocity
m_{wet}	$[kg]$	Initial spacecraft mass with fuel
m_{dry}	$[kg]$	Spacecraft structure mass
g_0	$[m/s^2]$	Standard Earth gravity, $g_0 = 9.807 m/s^2$
g_{Enc}	$[m/s^2]$	Enceladus gravity, $g_{Enc} = 0.113 m/s^2$

Vision

$X^{\mathcal{F}}, Y^{\mathcal{F}}, Z^{\mathcal{F}}$	3D coordinates in $\mathcal{F}$ frame of a point in space
$X^{\mathcal{W}}, Y^{\mathcal{W}}, Z^{\mathcal{W}}$	3D coordinates in the World frame of a point in space
$\mathcal{I}_{RD}$	The oriented image frame: u-Right; v-Down.
$\mathbf{K}$	Intrinsic matrix.
$\mathcal{I}_{RD}$	The oriented image frame: u-Right; v-Down.
$\mathbf{P}_0$	Normalized projection matrix.

Navigation - SLAM

$\hat{x}$	An estimate of the random variable **x**.
$\hat{x} \sim \mathcal{N}\{\hat{x}; \mathbf{X}\}$	Shortcut for "**x** is Gaussian with mean $\hat{x}$ and covariances matrix **X**
$\mathcal{W}$,$\mathcal{R}$,$\mathcal{S}$	The world, robot/vehicle, and Sensor reference frames.
$\mathcal{R}$	State vector of the robot frame.
$\mathcal{M}$	State vector of the set of landmark positions,.

HDA

$f_f^S(x,y)$	Hazard information layer for feature f, sensed by sensor S
β_x	Sensor confidence parameter if x is sensor, or else generic scoring factor for a parameter x

Guidance and propulsion

$\mathbf{T}(t)$	$[N]$	Spacecraft thrust as a function of time
$\Gamma(t)$	[-]	Slack variable (thrust relaxation) as a function of time
α	$[s/m]$	Mass flow factor
θ	[°]	Thrust angle
γ_{gs}	[°]	Glideslope angle
ψ	[°]	Velocity magnitude constraint slope
I_{sp}	$[s]$	Specific impulse
t_f	$[s]$	Total time of fight
ρ_1, ρ_2	$[N]$	Spacecraft lower and upper thrust limits

List of Notations

x	Scalar

$\mathbf{x}$	Vector
$\mathbf{X}$	Matrix
$x(t)$	Dependent variable of time 't'
$\|\|\mathbf{x}\|\|$	Euclidean norm or 2-norm of a vector
$\dot{x}, \ddot{x}, ...$	Time derivatives

Chapter 1

Introduction

1.1 Astrobiology on the icy moons

The search for life in the solar system has been one of the main driving forces behind space exploration. The recent discoveries that the icy moons of the giant planets in our solar system contain vast oceans, has made them prime targets for that search. In particular, Jupiter's moon Europa and Saturn's moon Enceladus are currently the most promising candidates among the icy moons, as they appear to fulfill the basic requirements for them to host life: the heat that is generated by the tidal pull of the parent planet maintains a sub-glacial ocean in the liquid state and in direct contact with the rocky core of the moon, through which reactions critical for the creation of the building blocks of basic life as we understand it can occur. Exchange processes through the thick ice shells covering those moons, much like in the polar regions of Earth, mean that further chemicals needed for life are transported from the surface where they have been delivered by e.g. micro-meteoroids, all the way down to the ocean [1, 5, 4]. The chemical makeup of plume jets found to emanate from the south pole of Enceladus by the recently decommissioned Cassini spacecraft further point to a chemically rich ocean hospitable to microbial life [3]. There are strong indications that similar plumes also exist on Europa [6].

Hydrothermal vents in the ocean bottom of an icy moon are often suggested as a source for the emergence of potential microbial life there [3, 4, 8]. Any potentially present microbial life created there can then migrate and populate other hospitable niches in the ocean of the icy moon (Figure 1.1). Signs of that life can be carried closer to the surface and beyond by various transport methods. Glaciological processes in the ice shell slowly transfer ocean material upwards. Once on the surface, ocean material and any signs of possible life contained in it are degraded by the strong radiation surrounding the giant planets [9].

A more dramatic transport method is through channels that directly connect to the aforementioned plumes on the surface. Microbes from the ocean could be carried along with the ocean water, and then ejected to space via the plumes [10]. The microbes would remain in their unaffected state up to a certain depth under the plumes and could even form small microbial communities in pockets of liquid water close to the plume channel.

Any potential life forms will be heavily altered after they are ejected by the plumes, due to their exposure to vacuum, but signatures of life could be detectable in the plume material and in the deposits it may form on the surface of the icy moon [3].

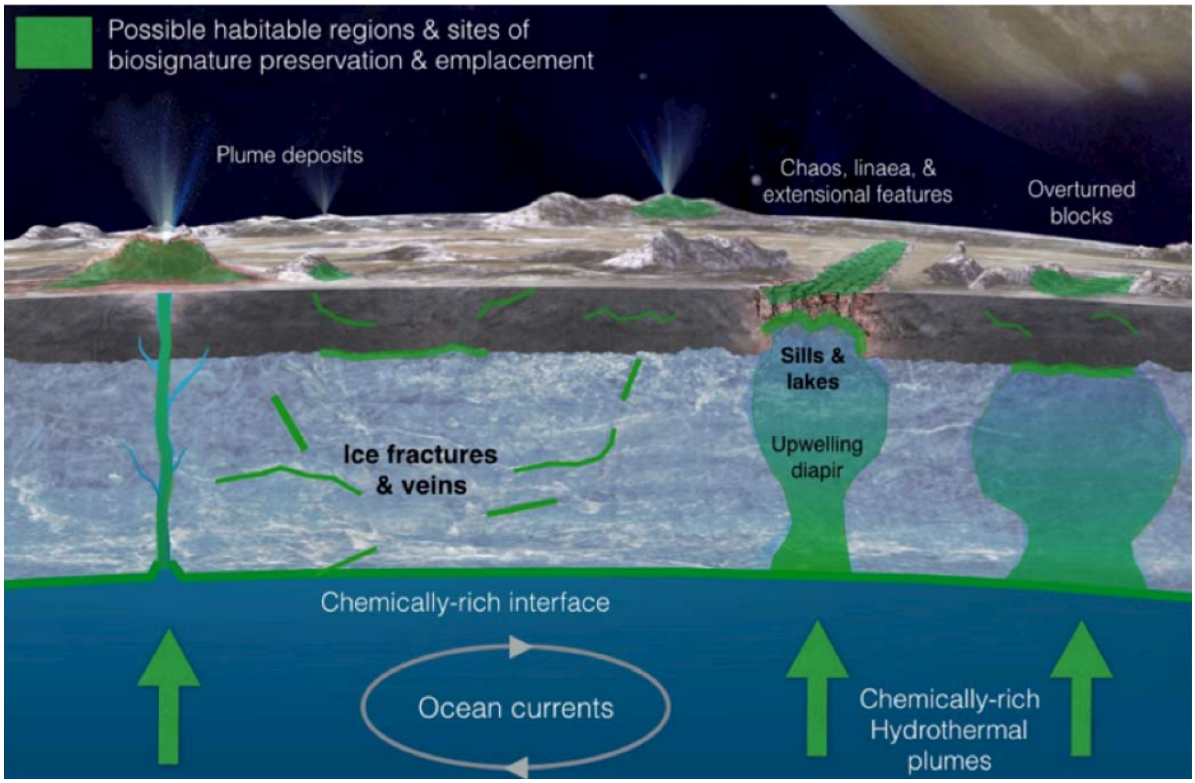

Figure 1.1: Potential habitable regions on an icy moon, and transport mechanisms of ocean material to the surface [2]. Although the figure refers to Europa, the concept is also valid for Enceladus.

1.2 Mission concepts for in-situ icy moons astrobiology

Each of the potential habitable regions and biosignature transfer and preservation areas shown in Figure 1.1 can be conceivably sampled in-situ for signatures of life.

An orbiter could fly through the plumes, capturing freshly ejected material for in-situ analysis or Earth return [11]. A lander could land near areas where material from the ocean has been deposited via glacial upwelling, by plume ejection, or by other means [2, 90]. Increasing in complexity, but also in science return potential, a probe can be deployed from a lander and melt through the ice to reach material deposits or even local liquid sub-glacial water reservoirs [38], or even climb down the plume channel, potentially all the way to the ocean [12]. Even more ambitious mission concepts aim to melt through the entire ice shell to deploy a submersible that will navigate the ocean, investigate the hypothesized hydrothermal vents themselves and characterize the hypothesized extant life there [13].

Interest in such missions is indeed growing, as evidenced e.g. the creation by NASA of an

Ocean Worlds Exploration Program (OWEP) to parallel the Mars Exploration Program (MEP) in the mid-term has been hinted at [14, 15]. The German Space Administration (DLR) has been investigating technologies for accessing environments relevant to the icy moons through the Explorer and other initiatives [16].

In the above mission concepts, an interesting outlier in cost-to-science-return can be said to exist: the ice shell beneath the plume sources of Enceladus is postulated to be 2 - 3 km thick, with liquid water potentially rising even to the surface [10], and forming liquid water pockets a few hundreds of meters under the surface. The combination of the relative ease of access to these water pockets and the freshness of the material contained in them makes them a very desirable target for in-situ astrobiology.

1.3 The Enceladus Explorer project and the EnEx lander concept

A mission concept for the in-situ astrobiological sampling of a relatively shallow water pocket ($\sim$ 200 m) under a plume source on Saturn's moon Enceladus, was developed between 2012 and 2015 at the Institute of Space Technology & Space Applications (ISTA) of Bundeswehr University Munich. The lander would land near a plume source and deploy a future version of the IceMole melting probe currently under development by FH Aachen [39]. This work was part of the joint research collaboration of eight German universities called Enceladus Explorer (EnEx) and funded by the German Space Administration (DLR). The main goal of the EnEx project was the development of a terrestrial navigation system for a subglacial research probe, to be applied and field-tested on the IceMole. The Enceladus Lander (EL) mission concept is presented in [38] in some detail and will be summarized in the following.

The mission concept comprises of three elements: an Orbiter, a Lander, and the IceMole.

The IceMole (Figure 1.2) is a novel, highly maneuverable sub-glacial probe combining melting and screwing. A differential melting head can be heated in different sides and hollow screw assures close-contact between the melting head and the ice, which optimizes conductive heat transfer into ice and, combined with the differential heating, aids steering in the desired direction. It is connected to the lander via a tether a few hundreds of meters long, un-spooled from its backside as it melts into the ice. Further significant elements of the IceMole is the subglacial navigation system described in [40] as well as its general operational autonomy [38]. Both of these areas have been under significant development within the original EnEx initiative and its follow-on projects [16].

The Lander is seen in Figure 1.3. Due to the large power requirements coming from the electrical melting head of the IceMole and the large distance from the Sun, a small 5 kWe nuclear reactor is chosen as the power source. A radiator is in turn necessary to radiate away the excess heat generated by the reactor. The radiation emitted by the nuclear reactor when it is active further drives the configuration of the lander, with lander electronics set on top of a mast and behind shielding material. The lander must be able to perform a very demanding autonomous planetary landing.

According to the EL concept, the nuclear reactor is also used to provide power to the

Orbiter during the interplanetary trip using Nuclear Electric Propulsion (NEP).

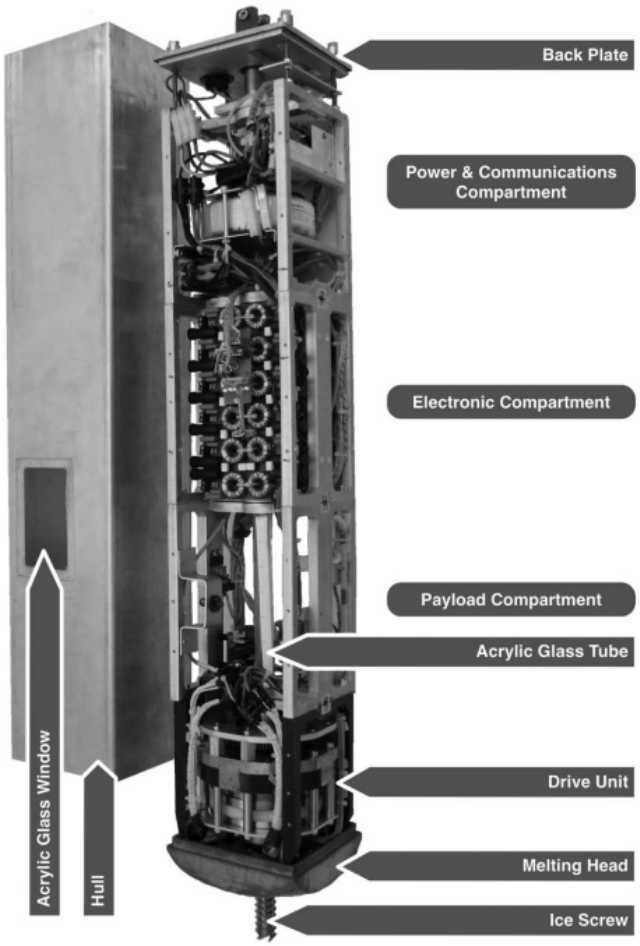

Figure 1.2: The IceMole 1 design as tested on the Morteratsch glacier, Switzerland (2010) [39]

.

After launch, the combined Orbiter-Lander-IceMole spacecraft will transfer to Enceladus and capture in orbit around it. It will then perform remote sensing of potential landing sites. Due to the strong gravitational influence of Saturn, orbits with inclinations higher than 60° are not stable for more than a few days, making the reconnaissance of polar latitudes operationally challenging [38].

Once landing site reconnaissance is completed by the Orbiter, the landing phase can start (Figure 1.3, left). After separation from the Orbiter, the Lander performs a burn to simultaneously lower its periapsis to 5 km above the south-pole and raise the inclination of its orbit from 60° to 90°. The Lander then coasts to periapsis, where it starts a second burn to dump most of its excess velocity. A ΔV of 180 m/s is needed for an impulsive landing maneuver at periapsis [90]. Once the Lander has achieved a sufficiently reduced velocity the engines begin to throttle down, and a targeted descent phase begins. Once the Lander reaches an altitude of 10–20 m, it shuts its engines and slowly free-falls to a soft landing on the surface.

After a precise and safe landing near one of the plume sources (Figure 1.4, right), the IceMole will be deployed and melt to a depth of $\sim$ 200 m to a sub-glacial liquid water pocket, which it will then sample for potential signatures of life, all the while communicating with the Earth via the Orbiter.

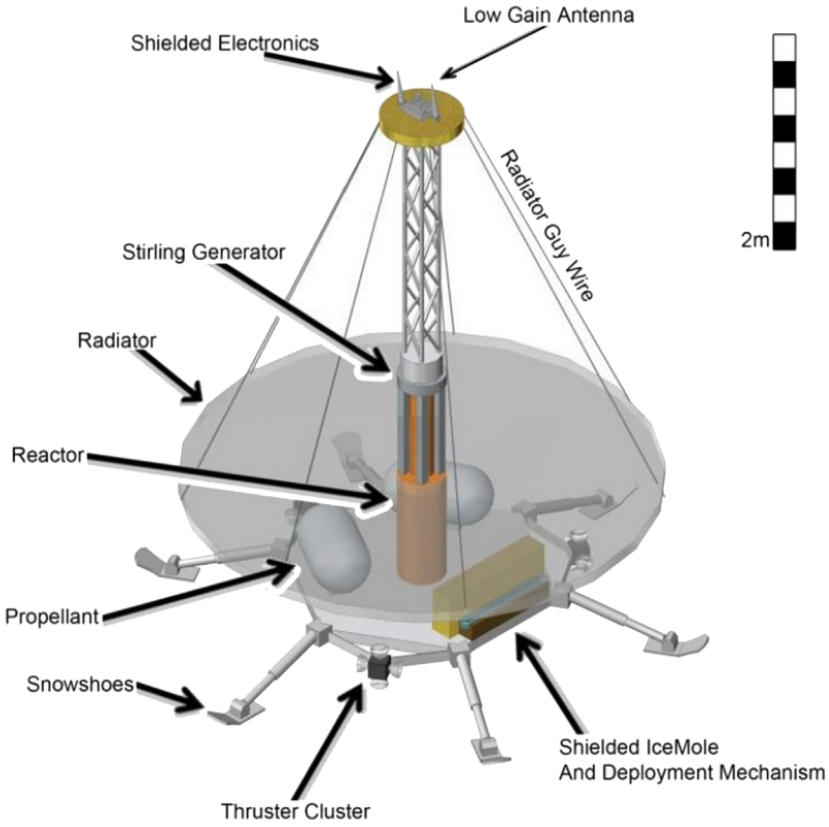

Figure 1.3: Preliminary design of the EL-Lander [38].

1.4 The environment near the plume sources of Enceladus

Enceladus is a small icy moon (diameter $\sim$ 500 km) orbiting Saturn at a radius of 4 Saturn radii. The moon is tidally locked, with identical rotational and orbital periods of 1.37 days. Its orbital eccentricity and inclination are close to zero. It has an approximate surface gravity of 0.113 m/s^2 [33, 17]. More details on Enceladus are given in [33, 17].

Figure 1.5 gives an overview of Enceladus, showing its rocky core in contact with the ocean, with hydrothermal vents on its bottom. Water channels lead from the ocean all the way to the surface where it is ejected to space via the plumes.

The plumes are located in the South-Polar Terrain (SPT) of Enceladus, and indeed 100 distinct of them have been detected emanating from the bottom of the so-called tiger stripe canyons (Damascus, Baghdad, Cairo and Alexandria Sulcus) as seen in Figure 1.6 [10]. Each canyon is roughly formed by ridges, about 100-150 m high while the valley is

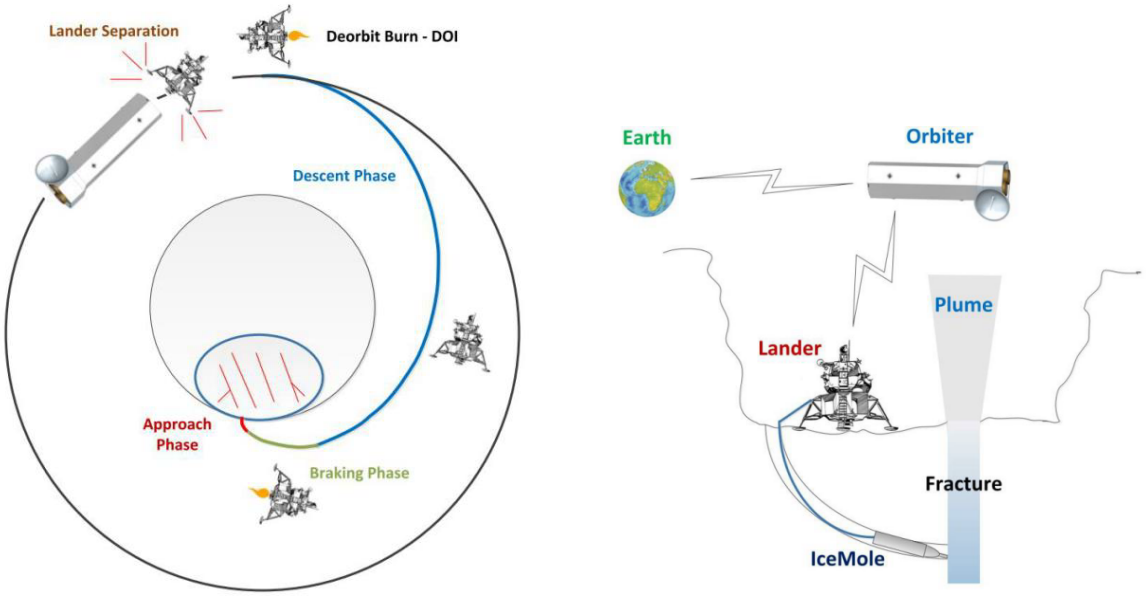

Figure 1.4: Concepts of operations for the EnEx landing (left) and the landed (right) operations phases.

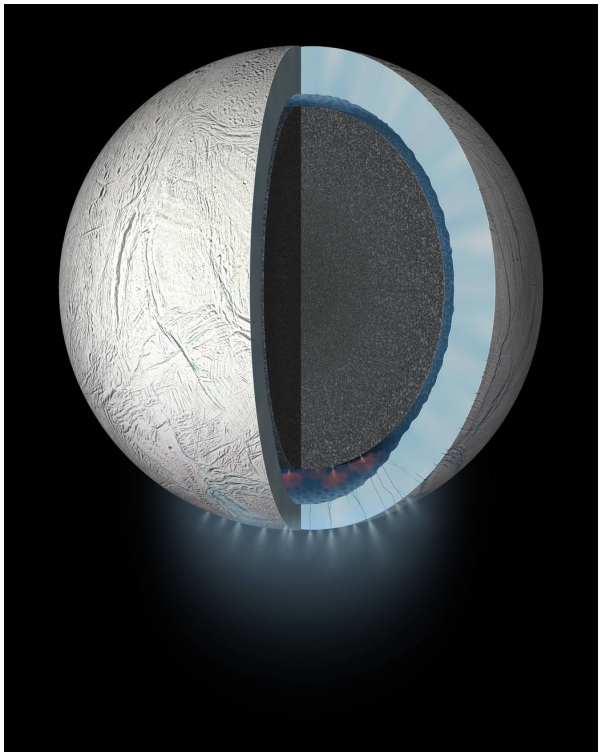

Figure 1.5: Cutaway view of the interior of Saturn's moon Enceladus and the plumes [7].

about 200-250 m deep and 50-100 m wide. The canyon faces have a maximum slope of $\sim 30°$[18]. The total width of the formation is about 2-5 km (Figure 1.7).

A close-up illustrating canyon topography around - and subsurface structure under - a plume source can be seen in Figure 1.8.

As seen in Figures 1.7 and 1.9, terrain on the SPT is interspersed with obstacles such as icy blocks. The spatial density of icy blocks features in the SPT was estimated for blocks larger than 10 m in [19] and found to be up to 1500 ± 450 blocks per square km. No distinct block distribution pattern was found with respect to the tiger-stripe flanks or jet sources. The resolution of existing maps is insufficient to deduce the distribution of smaller blocks. Other features typical to glacial terrains might exist, such as ice cracks etc.

An important aspect is the unusual texture characterizing the SPT. As ice particles are ejected by the plumes, larger ones fall near the plume sources while smaller ones fall further away or are even ejected into Saturn orbit (populating the E-ring). This results in the SPT terrain being covered in plume fallout. Studies have shown that nearer the plumes, particle deposition rates can reach up to 1 mm/year or more, indicating a deposit layer thickness of tens of meters (if the plumes can be assumed to have been active in the past million years) [20]. Still, exposed icy crust can also be encountered, especially on slopes on which less consolidated material has slid downward.

The mechanical behavior of the fallout is crucial to the understanding of the surface texture. In a first approach the fallout can be approximately treated as super-fine snow, comprising grains of about 7.5 μm outside the SPT, 40 μm in the vicinity of the Tiger Stripes and 100 μm or larger inside the valleys [20]. Grains are expected to have lost their crystalline shape due to collisions with the vent walls, and have a roughly round shape. These microscopic properties can result in increased force transmission capacity due to the fine grain size and non-consolidated layers of material and increased compressibility due to the low gravity. On the other hand, due to the absolute lack of humidity and very fine grain size, a possible analogue for it could be dry quicksand on Earth [32]. Alternatively, thermal cycling could have cause fallout particles to have sintered, resulting in a hard surface. Overall, a high uncertainty around the properties of the surface texture is assumed here. In the following, the plume fallout will be referred to as snow.

Another characteristic aspect it the polar latitude of the terrain in question. The SPT includes areas below a latitude of 60°S, meaning that it is characterized by polar illumination conditions. Due to the eccentricity of Saturn's orbit the total solar flux in orbit around Saturn varies between 17 W/m^2 and 13 W/m^2. Enceladus follows the seasonal cycle of Saturn: with Saturn's orbital period of 29.46 years about the Sun, each season on Enceladus lasts approximately 7 to 8 years. The next Saturnian southern summer period will begin in the year 2028 and last until beginning of 2036 with the peak in 2032. For the majority of the plume sources the maximum Sun angle during summer solstice does not exceed 40°. During the rest of the southern summer on Enceladus the illumination angles will be smaller, with many areas in permanent darkness due to the rough canyon topography. During the long southern winter the entire SPT will be in complete darkness [33, 17].

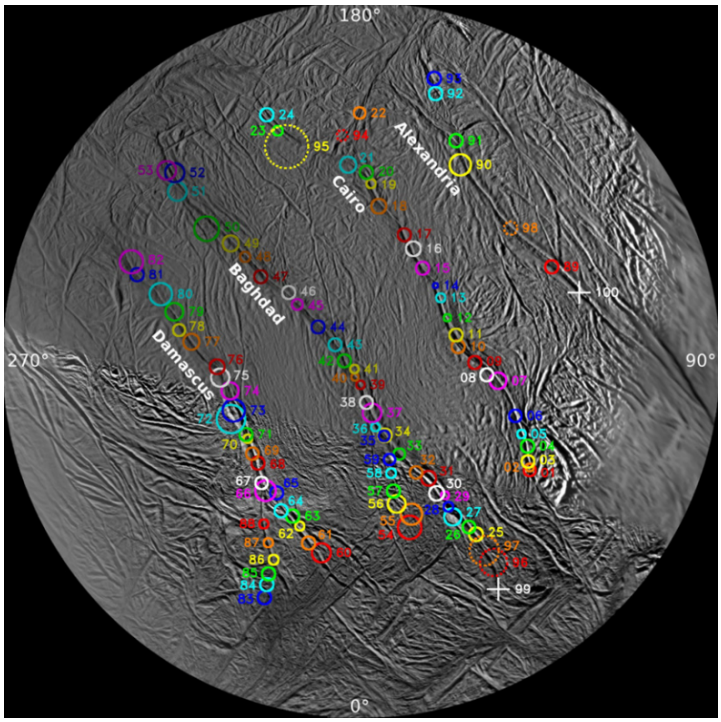

Figure 1.6: Map of Enceladus' south polar terrain (SPT) showing all 100 currently known plume sources. The circles are the corresponding 2σ location uncertainties [10]

.

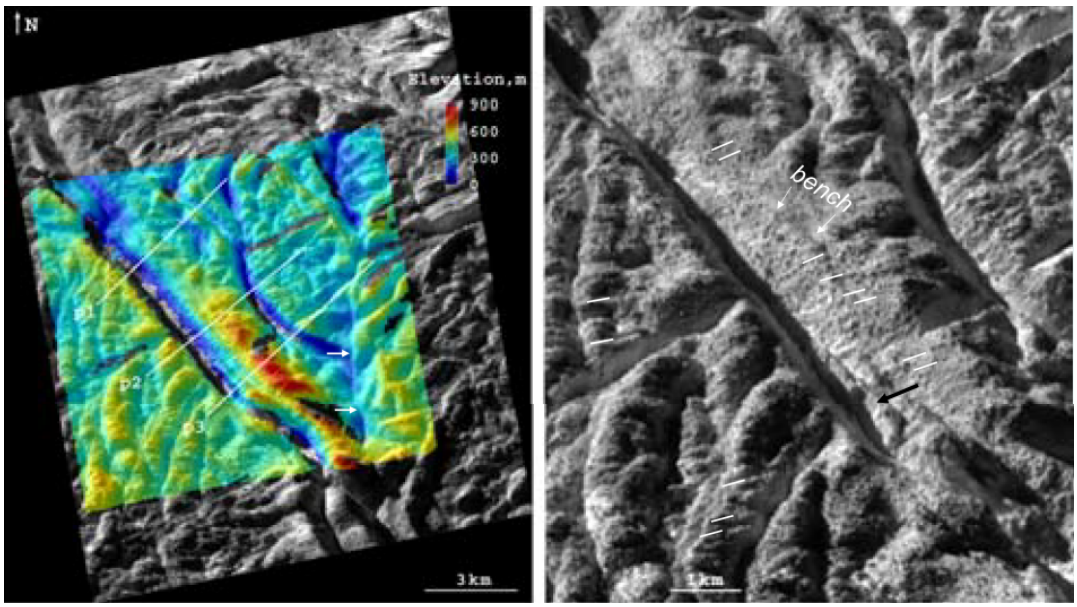

Figure 1.7: Left: Stereo-derived elevation model of characteristic canyon area on the SPT, on Baghdad Sulcus (76°S / 323°E, resolution: 150-300 m/pixel horizontally and 7-14 m/pixel vertically). Right: Enlarged image of the central-lower part of the previous image [18].

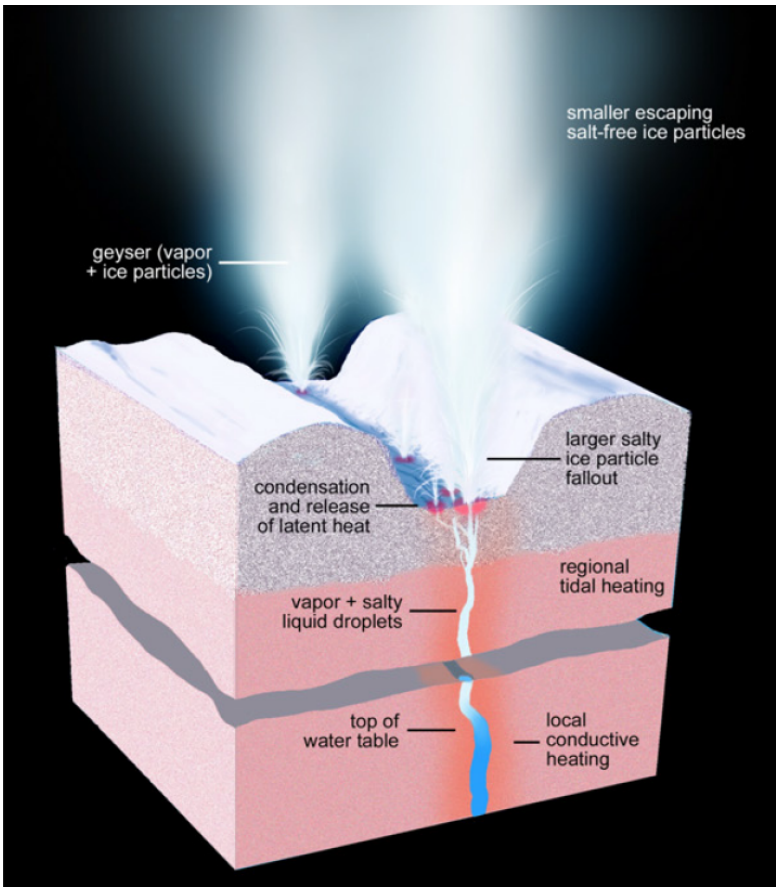

Figure 1.8: Schematic illustrating current knowledge of the small-scale physical and thermal structure and processes relevant to Enceladus' geysering activity [10].

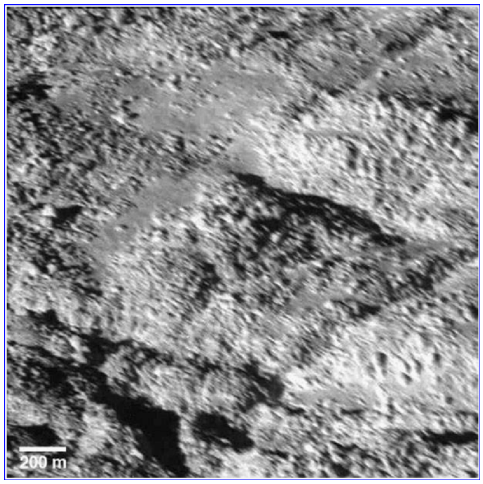

Figure 1.9: Blocky terrain just beyond the terminus of Damascus Sulcus (resolution: 4 m/pixel). Image is slightly smeared due to spacecraft motion [18].

The density of H_2O molecules in the vicinity of the plumes has been modelled in [34]. At the very mouth of the plume source the water vapor density is estimated at $\sim 10^{22}$ molecules/m^3, dropping by three orders of magnitude within a distance of 100 m. Comparing this to Earth's atmospheric density at sea level ($\rho_{Earth} = 10^{25}$ molecules/m^3), or to that of Mars ($\rho_{Mars} = 10^{23}$ molecules/m^3) it is seen that the plume density is negligible for engineering purposes.

The fresh, clean ice that dominates the surface makes Enceladus the body with the highest albedo in the solar system (~ 0.99).

Despite the icy surroundings, the bottom of the tiger stripe canyons are relatively warm. Due to water condensation, and the resulting release of heat at their mouths, the plume sources can act as concentrated hot spots on the canyon bottoms. This means that the canyons have temperatures of up to at least 167 K on the large scale [35], and can reach to temperatures up to 200K in concentrated areas (tens of meters) surrounding the plumes (Figures 1.6 and 1.10) [10, 36].

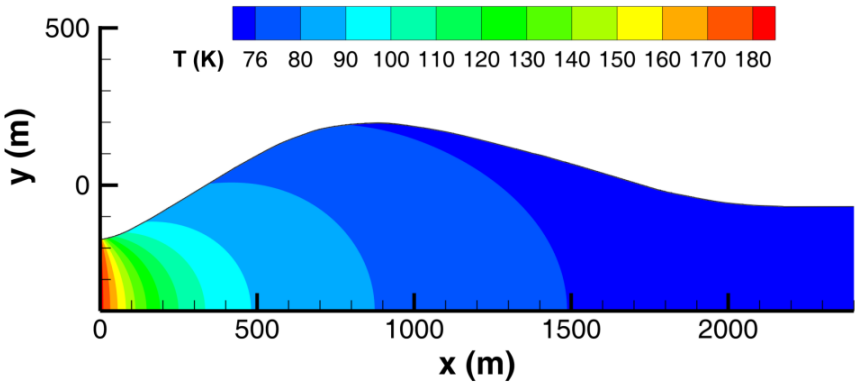

Figure 1.10: Model temperature distribution at an idealized tiger stripe, showing terrain temperature over a 2D cut-out of the canyon terrain near a plume source. Distance from the plume source and terrain elevation are denoted by x and y respectively. Plume source temperature is assumed to be 180 K [36].

Another interesting characteristic is the difference in thermal inertia of the two textures composing the surface. Ice has a relatively high thermal inertia, while loosely packed and dry snow has a very low one, comparable to the best industrial insulation materials [37]. It thus follows, that in the localized heat environment near the plume source, the surface temperature will vary significantly according to the local texture.

The SPT of Enceladus presents a restricting area from a planetary protection (PP) perspective. Active venting from fissures in the ice may lead directly downward into a liquid water environment and cracks at the surface may stay open to a few kilometers depth. The PP sensitivity of such a mission would be thus unprecedented, making PP one of the driving requirements. In addition, additional PP requirements concerning nuclear or chemical materials might be made for landing near a plume source. This means that among decontamination and other measures, an especially high landing reliability might be required.

Finally, even though Saturn possesses a trapped particle belt, radiation around it, and in particular in the vicinity of Enceladus is not as harsh as, e.g. that of Jupiter's Europa. Furthermore, Enceladus itself aborbs some of the particles around Saturn, slightly shielding spacecraft in low orbit around and on it [17].

1.5 Landing requirements definition

Arguably the most critical phase of any lander mission is the landing itself, where much of the mission/operational risk is concentrated. EnEx in particular, is a mission where a large majority of the science will be performed after landing, where landing must take place in an extremely challenging terrain, and where landing reliability must be ensured to a level even higher than that of past landing missions due to the strict planetary protection rules.

From the environmental, operational, and regulation aspects described in the previous sections, the requirements for landing near a plume source can be deduced. The deduction process is shown in Table 1.1. The first step includes decomposing the landing to three verbal characteristics it must possess: it must be accurate, safe, and reliable.

Focusing on each characteristic in turn, accuracy was first investigated. The lander must land within the canyon bottom (with a width of 50-100 m), and it must also land close enough so that the IceMole can reach its subglacial target under the plume source (depth of $\sim$ 200 m) without running out of tether. Based on analysis discussed in [38], a distance of 50 m from the plume source is deemed adequate.

The next characteristic investigated is landing safety. This can be broken down into further safety elements. The lander must not overturn on landing, so it must land on a slope within its landing tolerances. It must also avoid landing on obstacles, that might results in the lander exceeding its slope tolerances. A further reason for the lander to avoid obstacles is to not risking puncturing or otherwise damaging its underside. It must then avoid landing on obstacles that are bigger than a set tolerance. To further avoid damage or overturning, the velocity of the lander upon touchdown must not exceed a set limit. This limit is exacerbated for landing in a tiger stripe canyon due to the possibility of landing on snow and causing an avalanche that could overturn or cover the lander. Depending on the design of the lander and the touchdown method, as well as the eventual dynamical behavior of snow, snow cover could also be critical to landing safety, and should be potentially avoided. Furthermore, the lander must be able to localize itself accurately, so that it can reliably avoid any landing hazards and also avoid landing too close to the planetary-protection-sensitive area close to the plume source.

A final desirable landing characteristic is reliability. To consider the landing viable, a specific probability of success must be ensured. That number for landing missions is typically set in the order of 99% (see e.g. [21]). However in this case, as planetary protection is involved, it must be ensured that in any failure scenario the contamination probability will remain within acceptable levels. This is a complex requirement to verify here. In its place, a simple substitute requirement will be given, to not land to close to the plume source.

Table 1.1: Requirements traceability matrix for landing GN&C. Elements in italics refer are special or "soft" requirements.

Characteristic	Requirement	Reason
Accurate	Distance to plume <50 m	Canyon bottom size IceMole tether length
Safe	Velocity < 1 m/s in any direction	Lander tolerances
	Velocity < 0.25 m/s in any direction	Avoid avalanche on snow touchdown
	Detect slope > 15 deg.	Lander tolerances
	Land in darkness	Polar illum. and topography, long seasons (programmatic)
	Detect obstacles > 50 cm	Lander tolerances
	Prefer ice over snow	Texture, lander tolerances
Reliable	TBD success probability	Failure probability within limits
	Distance to plume >5 m	Avoid ocean contamination

The specific lander tolerances and other relevant characteristics for a successful landing are gathered in Table 1.2. As the EnEx design was preliminary, the lander characteristics used here are adopted from a few similar planetary landing missions with similar requirements and operational conditions (see also Section 1.6). An important characteristic is the tolerance of the lander to landing in snow. As it is very likely that the landing terrain will be largely covered by snow, and that landing on this snow will be of higher risk than landing on firm ice, a special or "soft" requirement is introduced here that the lander should "prefer" landing on ice over snow. This is not an ideally expressed requirement but it should be sufficient for the purposes of this work[1].

Table 1.2: Lander tolerances and characteristics, adopted from similar landing missions [91, 78]. Elements in italics originate from special or "soft" requirements.

Tolerance	Value
Survivable slope	15°
Max. survivable obstacle	50 cm
Max. survivable impact velocity (any direction)	1 m/s
Max. survivable impact velocity impact velocity on snow (any direction)	0.25 m/s
Lander footprint (circle diam.)	4 m
Able to land on snow	*Yes*

From the above information, some preliminary top-level functional and performance requirements for the lander, and in particular its GN&C system can be defined and

[1]Wherever this soft requirement is mention in this section, it is written in italics.

listed here. Each requirement is given an identifier corresponding to its the landing verbal characteristic it originated from (ACC for accurate, SAF for safe, and REL for reliable).

- **ACC001**: The lander shall land no further than 50 m from a plume source,
- **ACC002**: The lander shall land with a velocity of 0 and a velocity uncertainty of < 0.25 m/s,
- **ACC003**: The lander shall land on the designated landing spot with an accuracy of < 1 m relative to surrounding hazards,
- **SAF001**: The terrain slope in the lander footprint upon landing shall not exceed 15°,
- **SAF002**: No obstacle larger than 50 cm shall exist within the lander footprint upon landing,
- **SAF003**: *The lander should preferably land on ice rather than snow, other considerations being equal*,
- **SAF004**: The lander shall be able to land in darkness,
- **REL001**: The lander shall land no closer than 5 m from a plume source[2],

Due to the uncertainties in orbital maps and imperfect lander navigation (as will be also seen in Sections 2.2 and 2.2.5 respectively), it is impossible for an automated sequential landing with a pre-calculated trajectory to achieve the above requirements. It would be further impossible to remotely control the landing due to the ~ 3 hour two-way signal delay from Earth to Saturn. The following requirements then demand from the lander to perform certain functions autonomously on-board. The identifier assigned (AUT) stands for autonomy.

- **AUT001**: The lander shall sense terrain slope,
- **AUT002**: The lander shall sense terrain roughness and detect obstacles larger than 10 cm (reasonable assumption, for dangerous obstacles can be resolved),
- **AUT003**: *The lander should sense terrain snow cover*,
- **AUT004**: The lander shall be able to identify and land on a new suitable landing spot,
- **AUT005**: The lander shall be able to reach a new suitable landing spot,
- **AUT006**: The lander should be able to identify and localize the plume source to within 10 m 3σ if its not known to that accuracy before landing operations start,
- **AUT007**: The lander shall be able to sense all hazards also in darkness.

1.6 State-of-the-art for precise and safe planetary landing

Robotic planetary landing has a long history, starting with Luna 9 softly landing on the moon in 1966 [22], up to more recently to the Mars InSight lander [23]. Up to now, such robotic planetary landings have been automated, landing blindly while following either a timed pre-set event sequence, or using simple triggers, e.g. a certain altitude being

[2]This PP-related requirement is classified here provisionally as a reliability requirement.

reached as measured by an altimeter. The accuracy achievable is in the orders of 10-100 km for both Mars and Moon landings. To ensure a safe landing, large, flat areas are chosen as targets (e.g. [24]). This significantly restricts landing missions from accessing scientifically interesting areas and forces planetary rovers to drive long distances to reach such areas.

Of the currently planned missions, NASA's Mars2020 employs the most advanced landing GN&C system: it navigates during landing by comparing camera images to an a-priori map stored on-board, and performs simple avoidance maneuvers to pre-marked hazardous areas known to exist on that a-priori map [25].

Several novel landing mission concepts exist, that aim to land on planetary surfaces with requirements of safety and accuracy, similar to those of EnEx. The ESA Lunar Lander had the aim to land on the rim of the Shackleton crater near the lunar south pole, with an accuracy in the order of ~ 100 m and with autonomous hazard detection and avoidance. Apart from the preparatory science for human exploration, its goal was to demonstrate critical technologies for planetary landing and prove Europe's ability to land safely and precisely. The mission reached the B1 phase before being canceled in 2012 [91, 94]. NASA's Lunar MARE concept aims to land on a 100-meter radius region of smooth lunar maria terrain near Aristarchus crater for sample collection and dating, with an accuracy of $\sim 20m$. Due to the hazardousness of the area, a hazard detection and avoidance capability is included [78, 25]. Both these concepts carry a wide set of sensors including lidar and cameras to observe the terrain for navigation and hazard detection, and the capability for autonomous re-targeting to safe landing locations.

Technology development programs around autonomous precise and safe planetary landing are being performed by the agencies.

NASA is developing several aspects necessary with interconnected relevant projects [25]. Autonomous Landing Hazard Avoidance Technology (ALHAT) was a technology development program that combined autonomous guidance, navigation and control algorithms to characterize a landing surface while identifying and avoiding hazards in real time, and also navigate in reference to those hazards. It was extensively tested on helicopter flights and the Morpheus autonomous landing testbed, and has brought its HDA payload to a TRL of 6 [26, 25]. ALHAT has also been proposed for landing on Europa [28].The CoOperative Blending of Autonomous Landing Technologies (COBALT) program developed technologies for terrain relative navigation, with the goal of infusion into near-term robotic and future human missions. It has been also tested on landing testbed called Xodiac to reach a TRL of 5 [25].

ESA and Airbus are currently working on the Precise Intelligent Landing using On-board Technology (PILOT) landing system, further developing the terrain relative navigation and hazard detection and avoidance technologies of the ESA Lunar Lander [27]. The PILOT system is planned to be flown as an ESA contributed payload on the Russian Luna Resurs lunar lander mission in the coming years [29].

There are several icy moon landing mission concepts with strict accuracy and safety requirements [90, 38, 2], however no detailed, dedicated GN&C system has been proposed for them, except in terms of potential and partial re-use of technologies used e.g., for lunar landing as seen above. There are also not a lot of detailed studies about landing

in difficult planetary topography. An interesting study discusses the necessary guidance technology for landing in planetary pits [87].

Planetary landing success statistics have been relatively low, and a lot of effort is needed for ensure a high landing reliability, in particular for a complex autonomous system needed to satisfy the strict requirements discussed above. A dedicated study on landing reliability, in particular for safe and precise landing, was not found in the literature.

1.7 Top-level systems and operations concept for safe and accurate planetary landing

The requirements derived in Section 1.5 can be allocated to top-level functions of a landing GN&C system:

- **Sensing** senses elements of the surrounding environment and of the lander itself,
- **Navigation** estimates the navigational state of the lander,
- **Hazard Detection and Avoidance (HDA)** identifies hazards and command a retargeting to a new suitable landing spot if needed,
- **Guidance** calculates a feasible trajectory from the current lander position to the selected landing target, and
- **Actuation and control** implements the thrust commands given to move the lander along the desired trajectory.

A top-level system architecture for the landing GN&C seen in Figure 1.11.

A general operations concept was derived by studying missions with landing requirements similar to EnEx (mainly the ESA Lunar Lander [91, 92], and NASA Lunar MARE [78], for both of which detailed information exists) and is given in Figure 1.12. A typical safe and accurate planetary landing consists of a few specific phases. The navigational state of the lander is accurately estimated right before landing initiation, e.g. using ground measurements. To start the landing, a Deorbit Initiation Maneuver (DOI) places the lander on a trajectory with a periapsis with an altitude of a few km above the general landing area. Before the periapsis is reached, at an altitude in the order of ~ 10 km, the Powered descent phase starts which sheds the largest part of the kinetic energy

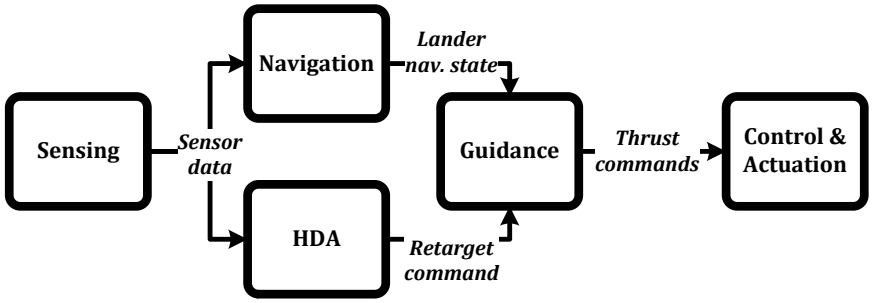

Figure 1.11: Top-level diagram of the landing GN&C system for the Enceladus Lander (EL).

by applying full thrust. During this phase the navigation state is propagated based on inertial navigation measurements. To reduce the navigational uncertainty relative to the surface, the navigation function uses e.g. on-board cameras to estimate the navigational state of the lander based on optical navigation methods. Once the nominal landing site comes in view and in range of the full sensor set on the lander (at a range of $\sim$ 2 km), the approach phase begins. During that phase, the landing terrain is sensed for hazards, the measurements are analyzed by the HDA function, and a command can be given to target a new, better landing site. Navigation further continues, now including the entire sensor set, e.g. a lidar. The guidance function can then calculate a trajectory and thrust profile to take the lander to the new landing site. The control and actuation function translates the thrust profile to actuator commands, and apply the commands on the on-board thrusters as required.

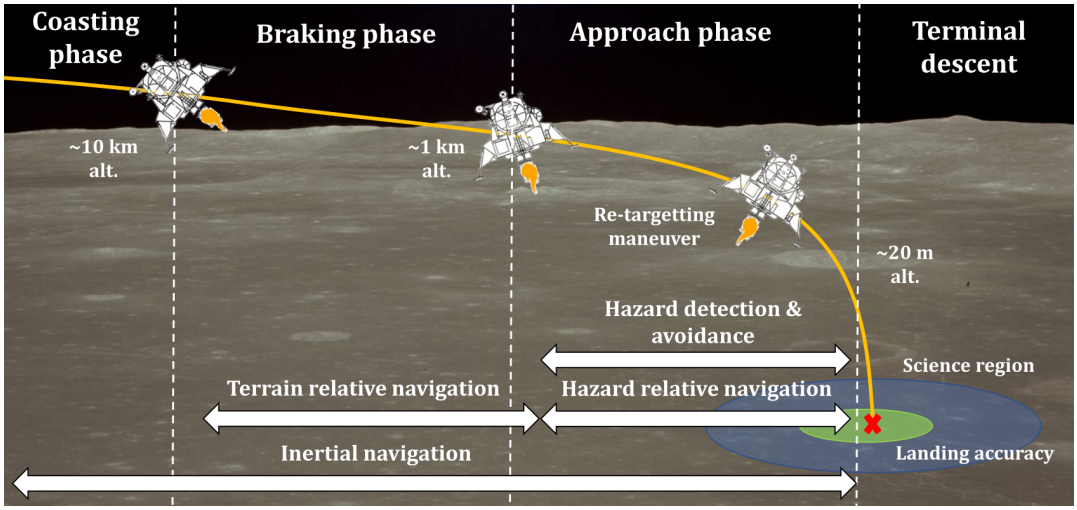

Figure 1.12: Generic concept of operations for safe and accurate planetary landing. Adapted from [78].

1.8 Goal of this thesis

From the information given in the previous section, the approach phase emerges as the most critical and complex operational phase of landing. In a span of a few minutes the landing GN&C system must apply the full extend of its capabilities to ensure a safe and accurate landing. A detailed closer look into the approach phase, and an extended effort into designing the needed system and operations would be central in furthering the development of the EnEx lander concept.

The mission concepts discussed in Section 1.6, deal with some of the issues facing the EnEx lander: landing accuracy and safety to a much higher degree than past landing mission and in addition, on significantly more challenging terrains. The EnEx lander would face the same challenges to an even greater degree: the landing would have to take place in very challenging topography, under polar illumination conditions on a terrain with an “exotic”, ultra-fine snow-covered texture. Landing reliability should exceed even the high requirements for other accurate landing mission concepts, due to the particularly strict planetary protection requirements. A crash near the target plume

source could introduce material from the lander to the ocean in timescales orders of magnitude smaller than the ones for a crash in any other part of Enceladus. On the other hand, particularities of the landing terrain such as the unusually rich thermal environment can be leveraged to extract additional information from sensed terrain.

The goal of this thesis then becomes the following: to investigate the approach phase of a safe and accurate landing near a plume source on the bottom of a tiger stripe canyon on Enceladus. First a detailed GN&C system and operations concept is to be defined, encompassing the key functions necessary: sensing, navigation, hazard detection and avoidance, and control and actuation. To verify the landing feasibility with the defined concept, and to further refine the design, a simulation tool was created incorporating each of the above top-level GN&C functions. The functions interact with each other in a closed loop and with simulated relevant elements of the landing environment. Using this tool, detailed analysis of each function is performed separately and modifications are recommended based on the results. Finally, a closed loop Monte Carlo simulation including all used functions is performed, and the run statistics analyzed.

The following analysis and the developed tool consider the lander a point mass with gravity and thrust vectors acting upon it, with the attitude dynamics of the vehicle not considered. This is a common and reasonable assumption made in similar analyses, as the attitude control takes place in far higher frequencies than that of the translational control - the two can thus be considered separately [81]. This approach also avoids incorporating attitude dynamics into the problem formulation, keeping the scope and complexity of the problem within reasonable bounds. Where necessary, the thrust pointing vector can be considered as replacement for lander attitude.

Chapter 2

Autonomous landing on Enceladus

In this section the underlying theory for each of the top-level functions for landing GN&C will be discussed and analyzed. Conclusions drawn from this analysis will help in further defining a detailed system and operations concept. It is reminded that this work does not consider attitude dynamics (Section 1.8). Only information relevant to translational dynamics will be given in this chapter.

2.1 Sensing

Sensors have the function to either sense particular elements of the lander navigation state (interoceptive), or elements of the external world (exteroceptive). Sensor types of interest will be described in this section.

2.1.1 Inertial Navigation System (INS)

An Inertial Navigation System (INS) uses its constituent accelerometer to measure the accelerations acting on the carrying vehicle, and by double integration estimate successively the vehicle velocity and position.

Real accelerometers are affected by several errors and noises (Figure 2.1). For an ideal accelerometer the input accelerations would correspond exactly to the output measurements; the function to map the first to the second would then be a straight line with a slope of 1. This is not the case for a real accelerometer: a bias (usually unknown) means that a non-zero measurement will be output even when no accelerations are acting on the sensor. A scale factor affects the measurement model line making slightly different than unity. A random noise is also always present in the measurement. After double integration, the noises lead to a random walk in the final position solution [41].

It is common to correct for these errors by filtering them (as in the Section 2.2.1). As will be seen further down in this chapter, this would mean that these drifts, some of them initially unknown (e.g. the bias), should appear in the dynamics and observation model equations, and be observed in the filtering process. This would increase the complexity of this work significantly. Considering the small timescales involved in the problem studied here, and the high quality of accelerometers used on-board spacecraft,

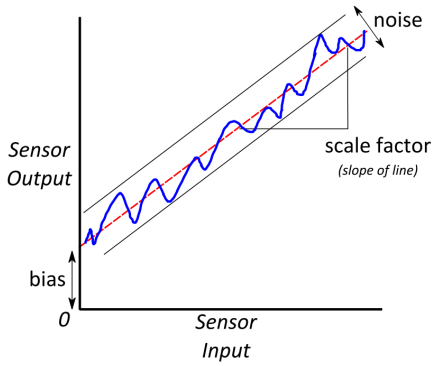

Figure 2.1: Most significant IMU Errors [41]

only the white noise, the simplest to integrate in the following filtering application, will be taken into consideration.

In the rest of this work, "INS" and "accelerometer" are used interchangeably.

2.1.2 Cameras

A camera sensor produces 2D projections of the 3D world. The simplest camera model is the pin-hole model (Figure 2.2). This model consists of the optical axis, and the focal plane perpendicular to the axis, with an infinitely small hole situated at the optical center, i.e. the intersection of the plane with the axis. A second parallel plane, called the image plane, is situated behind the focal plane at a distance f, called the focal distance. The point where the optical axis intersects the image plane is called the principal point. In a pin-hole camera, light rays coming from the external 3D world are projected into the 2D image plane in a straight line through the optical center. More advanced camera models can take into account ideal focused lenses, as well as real world lens distortion effects [46]. In this work however, only the simple pinhole camera model will be considered. According to the above, cameras can be defined as bearing-only sensors.

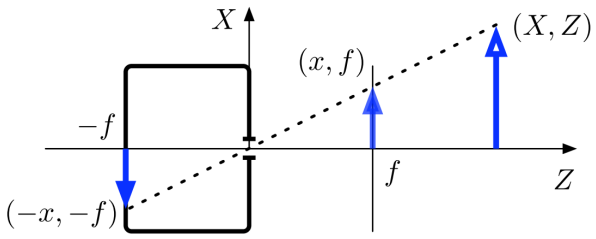

Figure 2.2: The pin-hole camera model [46].

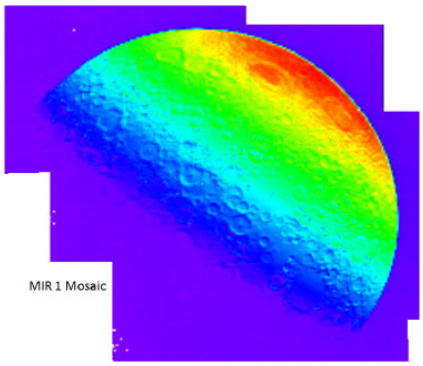

Figure 2.3: A thermographic image from the far-side of the moon taken by the mid-infrared camera on-board the LCROSS mission. Temperatures range from −100°C (blue) to 70°C (red) [43].

Cameras sensing in the visible spectrum - optical cameras - have been used in space since the beginning of the space age and are a basic component of GN&C systems for autonomous planetary landing (Section 1.6).

Similar to the optical camera, a thermal imaging camera uses infrared radiation, projecting the temperature of the sensed object in a 2D thermographic image (Figure 2.3)[3]. Although thermal cameras have been used or proposed to be used on-board spacecraft [42, 43], and models already exist for aerospace and defense applications [44], they have not so far not been proposed for autonomous planetary landing.

Camera images are composed by a finite number of pixels. By dividing the field-of-view of a camera, its angular resolution can be derived. The sharpness of a camera image also depends on the sensed wavelength, with optical cameras giving sharper images than thermal ones [45]. Image blur can be the result of incorrect focusing or movement during image creation.

2.1.3 Lidar

Rangefinder instruments (e.g. radar) estimate the distance to a surface by measuring the time it takes an emitted electromagnetic beam to return to the sensor head. LIght Detection And Ranging (LIDAR) sensors use light round-trip time-of-flight (ToF) to estimate the range to an impacted surface. For each measured point, the lidar produces a set of x,y coordinates on the sensor focal plane (similar to a camera) plus the range to the point, and the returned signal intensity [54]. An often used lidar variant is Flash LIDAR. Flash LIDARs simultaneously provide the range-and-bearing from the sensor to many points, generating a 3D point cloud representation of the observed object.

Lidar measurements are characterized by range and bearing errors, and a random unknown ranging bias. Furthermore, each lidar beam is not in reality an ideal ray, but defines a small cone, due to beam divergence, covering a beam footprint on ground impact. Multiple flash lidar measurements are usually taken in quick succession to reduce

[3]A distinction is to be made between thermal imaging cameras and night vision cameras, that simply intensify existing reflected visible light, see e.g. `https://www.flir.com/discover/ots/thermal-vs-night-vision/`.

range noise within acceptable limit, and methods exist for averaging out lidar biases (see e.g. [30]).

The sensing of the return pulses on a lidar is more commonly a discrete signal, depending on wiehter the incoming intensity surpasses a set threshold. It is possible however for some sensor head types to sample the entire reflected waveform of the return signal. This way, further physical properties of objects included in the diffraction cone may be derived with an analysis of the backscattered waveform. The shape of the returned waveform will depend on the roughness of the surface inside the diffraction cone of the beam in question and the reflectivity of the surface material. Methods to decompose the return signal waveform are given e.g. in [55] and are used among other applications, for forestry applications [56].

It is conceivable therefore that, given the right type of lidar sensor head and waveform decomposition methods, snow and ice can be distinguishable by lidar. This assumption is reinforced by two facts: first, snow will be significantly smoother than ice and will therefore produce a different waveform shape [49]. Second, and as can be seen in Figure 2.4, light in the visible wavelengths typically used by lidar is reflected differently by different types of ice and snow and more so for very fresh and pure snow. These two facts combine to reinforce the above assumption.

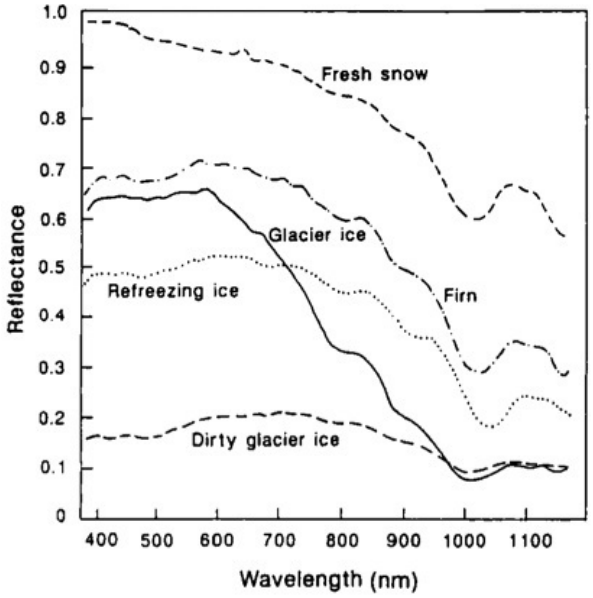

Figure 2.4: Spectral reflectance curves for various types of snow and ice [48]. Fresher and finer snow will have a higher reflectivity in all wavelengths than older and coarser snow (not shown in the figure).

A description of lidar used for planetary landing applications is given in [47]. The following assumption concerning lidar are made for this work: a range noise and a pointing error is considered for each lidar beam. Lidar rays are considered ideal rays, that can however detect terrain texture, albeit with a relatively high uncertainty.

2.2 Navigation

Navigation is defined as the task of estimating the state of the spacecraft from sensed data. Traditional landing approaches based on inertial sensing only, do not have the navigational precision to meet the high accuracy requirements (Section 1.6). In addition, and depending on the previous knowledge of the plume source location, the lander must be able to localize it with some accuracy (Section 1.5)

For lander navigation a sensor-aided inertial navigation approach is thus followed, where the measurements of the INS are corrected by those of exteroceptive sensors, namely optical and thermal cameras and lidar. A mapping element must be added for any sensed elements of the environment, such as a plume source, to be localized relative to the lander.

This process will be discussed here in detail. The approach to navigation taken here is based on references [46] and [50].

2.2.1 Filtering

The problem of navigation during landing under uncertain initial conditions and while using imperfect sensor measurements can be formulated as a filtering problem. Navigation filtering is defined as estimating the navigational state of a vehicle by making measurements of elements of that state. Several noises and uncertainties affect such a system: the vehicle is affected by random perturbations and has an uncertain initial state and sensors are inherently noisy. Furthermore, the way this system evolves with time, and the way the measurements relate to the current state of the vehicle are known [46].

Filtering is roughly made up of two steps:

- **Prediction/time update step**: the navigation state and related uncertainties are propagated to the next timestep according to the system model and corresponding probability distribution functions (pdfs) of its elements.
- **Correction/measurement update step**: the navigation state is corrected according to measurement of certain of its elements, according to the model and corresponding uncertainties for these measurements.

The key assumption commonly made in filtering problems is that all uncertainties are considered Gaussian.

Various solutions to the filtering problem exist (see e.g. discussion in [46]). Possibly the most widely used is the Kalman Filter and its variants. The driving assumption the Kalman Filter makes is the linearity of the evolution and observation equations. However, many practical systems have non-linear state update or measurement equations. Variants of the original Kalman Filter have been formulated to deal with these non-linearities. For the Extended Kalman Filter (EKF), this linearity constraint is relaxed by applying local linearizations around the most recent computed estimates [46].

2.2.2 Simultaneous Localization and Mapping (EKF-SLAM)

Apart from estimating the navigational state of a moving vehicle (in our case the lander), a map of landmarks can be constructed of the surrounding world by the vehicle using exteroceptive sensors. These landmarks can then be used at the same time to localize the vehicle. This process is known as Simultaneous Localization And Mapping (SLAM). The steps of this process are shown in Figure 2.5.

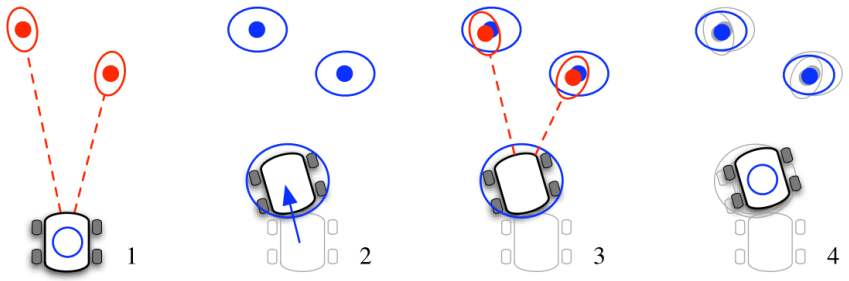

Figure 2.5: SLAM principle of operation for a single filter iteration. 1: Perception and initialization of new landmarks. 2: Prediction of vehicle motion & increase in position uncertainty. 3: Observation of already mapped landmarks from an uncertain vehicle position. 4: Correction of landmark positions and vehicle localization, with accompanying reduction in both vehicle and map uncertainties. Ellipses represent the uncertainty boundaries, and dotted lines the observations [46].

An Extended Kalman Filter can be used as the basis for a SLAM algorithm (EKF-SLAM). In this work, the EKF-SLAM formulation and notation given in [46] and [50] will be used[4].

In EKF-SLAM, the map[5] vector is defined as a state vector containing the vehicle pose (position, velocity, etc.) and the currently mapped landmark positions:

$$\mathbf{x} = \begin{bmatrix} \mathcal{R} \\ \mathcal{M} \end{bmatrix} = \begin{bmatrix} \mathcal{R} \\ \mathcal{L}_1 \\ \vdots \\ \mathcal{L}_n \end{bmatrix} \tag{2.1}$$

where $\mathcal{R}$ is the vehicle navigation state and $\mathcal{M} = (\mathcal{L}_1, \cdots, \mathcal{L}_n)$ are landmark states, with n the current number of landmarks in the map. The map in EKF is modeled as a Gaussian variable using the mean and the covariances matrix of the state vector,

[4]The reader is advised to familiarize themselves with the notation, as given in the preamble. The reader with little or no background in Kalman filters is referred to the simple and intuitive introduction given in [51], and an interactive tutorial in Matlab in [52].

[5]It should be kept in mind that the SLAM "map" used in navigation is a different than e.g. the hazard map used in HDA (Section 2.3).

denoted respectively by $\hat{\mathbf{x}}$ and $\mathbf{P}$:

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathcal{R}} \\ \hat{\mathcal{M}} \end{bmatrix} = \begin{bmatrix} \hat{R} \\ \hat{\mathcal{L}}_1 \\ \vdots \\ \hat{\mathcal{L}}_n \end{bmatrix} \qquad \mathbf{P} = \begin{bmatrix} \mathbf{P}_{\mathcal{R}\mathcal{R}} & \mathbf{P}_{\mathcal{R}\mathcal{M}} \\ \mathbf{P}_{\mathcal{M}\mathcal{R}} & \mathbf{P}_{\mathcal{M}\mathcal{M}} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{\mathcal{R}\mathcal{R}} & \mathbf{P}_{\mathcal{R}\mathcal{L}_1} & \cdots & \mathbf{P}_{\mathcal{R}\mathcal{L}_n} \\ \mathbf{P}_{\mathcal{L}_1\mathcal{R}} & \mathbf{P}_{\mathcal{L}_1\mathcal{L}_1} & \cdots & \mathbf{P}_{\mathcal{L}_1\mathcal{L}_n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{\mathcal{L}_n\mathcal{R}} & \mathbf{P}_{\mathcal{L}_n\mathcal{L}_1} & \cdots & \mathbf{P}_{\mathcal{L}_n\mathcal{L}_n} \end{bmatrix} \tag{2.2}$$

Elements across the diagonal of a covariance matrix $\mathbf{P}_{ij}$ are the variances σ_i^2 of the i-th element of the map vector, a measure of the uncertainty in their value (σ_i being the standard deviation of the value). Elements off the diagonal $\mathbf{P}_{ij}$ are the covariances, and represent the degree of correlation between the i-th state variable and the j-th state variable. The covariance matrix is always symmetrical ($\mathbf{P}_{ij} = \mathbf{P}_{ji}$).

The goal of EKF-SLAM it then to maintain the map $\{\hat{\mathbf{x}}, \mathbf{P}\}$ up-to-date when the vehicle moves, when it perceives a landmark it has already observed, and when it perceives a new landmark and decides to add it to the map. These steps correspond to the prediction and correction steps of the filtering described above, with the added step of landmark initialization.

2.2.2.1 Vehicle motion: the prediction step

The map can be initialized with state means and covariances of the navigation state of the vehicle, as well as of any known landmarks. Setting $\mathcal{R}$ as the state vector and $\mathbf{u}$ as the control vector, then the generic model update function is be given as

$$\mathcal{R}^+ = \mathbf{f}(\mathcal{R}, \mathbf{u}) \tag{2.3}$$

with $\mathbf{u} \sim N\{\hat{u}; \mathbf{U}\}$ a vector of controls assumed to be Gaussian with mean $\hat{u}$ and covariances matrix $\mathbf{U}$. The prediction step from the EKF formulation can be given as

$$\hat{\mathcal{R}}^+ = \mathbf{f}(\hat{\mathcal{R}}, \hat{\mathbf{u}}) \tag{2.4}$$

$$\mathbf{P}^+_{\mathcal{R}\mathcal{R}} = \mathbf{F}_{\mathcal{R}}\mathbf{P}_{\mathcal{R}\mathcal{R}}\mathbf{F}_{\mathcal{R}}^\top + \mathbf{F}_{\mathbf{u}}\mathbf{U}\mathbf{F}_{\mathbf{u}}^\top \tag{2.5}$$

$$\mathbf{P}^+_{\mathcal{R}\mathcal{M}} = \mathbf{F}_{\mathcal{R}}\mathbf{P}_{\mathcal{R}\mathcal{M}} \tag{2.6}$$

$$\mathbf{P}^+_{\mathcal{M}\mathcal{M}} = \mathbf{P}_{\mathcal{M}\mathcal{M}} \tag{2.7}$$

where the Jacobian matrices are defined by

$$\mathbf{F}_R = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{R}^\top}\right|_{\hat{R},\hat{\mathbf{u}}} \qquad \mathbf{F}_{\mathbf{u}} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}^\top}\right|_{\hat{R},\hat{\mathbf{u}}} \tag{2.8}$$

In short, during the prediction step the new vehicle nav. state is predicted (Equation 2.4), the part of the map covariance matrix concerning the vehicle state increases based on the map and control covariance matrices (Equation 2.5), the covariances between the vehicle and the landmarks are also updated (Equation 2.6), and finally the variances of the landmarks in map covariance matrix are left unchanged, as their state remains unaffected.

2.2.2.2 Observations of existing landmarks: the correction step

In EKF the generic observation function is written as:

$$\mathbf{y} = h(\mathbf{x}) + \nu \tag{2.9}$$

where $\mathbf{y}$ is the noisy measurement, $\mathbf{x}$ is the full state, $h()$ is the observation function and ν is the measurement noise[6]. The EKF correction step is then given as

$$\hat{\mathbf{z}} = \mathbf{y} - h(\hat{\mathbf{x}}) \tag{2.10}$$
$$\mathbf{Z} = \mathbf{R} + \mathbf{H_x}\mathbf{P}\mathbf{H}_x^\top \tag{2.11}$$
$$\mathbf{K} = \mathbf{P}\mathbf{H}_x^\top \mathbf{Z}^{-1} \tag{2.12}$$
$$\hat{\mathbf{x}}^+ = \hat{\mathbf{x}} + \mathbf{K}\hat{\mathbf{z}} \tag{2.13}$$
$$\mathbf{P}^+ = \mathbf{P} - \mathbf{K}\mathbf{Z}\mathbf{K}^\top \tag{2.14}$$

with the Jacobian $\mathbf{H_x} = \frac{\partial h(\hat{\mathbf{x}})}{\partial \mathbf{x}}$ and where $\mathbf{R}$ is the covariances matrix of the measurement noise. $\mathbf{z}$ is defined as the measurement innovation, and $\{\hat{\mathbf{z}}; \mathbf{Z}\}$ as the innovation's mean and covariances matrix. The innovation can be seen as the difference between the "real" and the "expected" measurement (Equation 2.10). Equation 2.12 is defined as the Kalman gain $\mathbf{K}$. Equations 2.13 and 2.14 express the update of the entire map (means and covariance matrix).

In SLAM, observations occur when a measurement of a particular landmark is taken by a sensor on-board the vehicle. When observing the landmark i, the observation model is given as

$$\mathbf{y_i} = h_i(\mathcal{R}, \mathcal{L}_i) + \nu \tag{2.15}$$

which does not depend on any landmark other than $\mathcal{L}_i$. The Jacobian $\mathbf{H_x}$ in EKF-SLAM is then

$$\mathbf{H_x} = \begin{bmatrix} \mathbf{H}_\mathcal{R} & 0 & \cdots & 0 & \mathbf{H}_{\mathcal{L}_1} & 0 & \cdots & 0 \end{bmatrix} \tag{2.16}$$

with $\mathbf{H}_\mathcal{R} = \frac{\partial h_i(\hat{\mathcal{R}}, \hat{\mathcal{L}}_i)}{\partial \mathcal{R}}$ and $\mathbf{H}_\mathcal{R} = \frac{\partial h_i(\hat{\mathcal{R}}, \hat{\mathcal{L}}_i)}{\partial \mathcal{L}_i}$. The set of correction equations then becomes

$$\hat{\mathbf{z}} = \mathbf{y}_i - h_i(\hat{\mathcal{R}}, \hat{\mathcal{L}}_i) \tag{2.17}$$
$$\mathbf{Z} = \begin{bmatrix} \mathbf{H}_\mathcal{R} & \mathbf{H}_{\mathcal{L}_i} \end{bmatrix} \begin{bmatrix} \mathbf{P}_{\mathcal{R}\mathcal{R}} & \mathbf{P}_{\mathcal{R}\mathcal{L}_i} \\ \mathbf{P}_{\mathcal{L}_i\mathcal{R}} & \mathbf{P}_{\mathcal{L}_i\mathcal{L}_i} \end{bmatrix} \begin{bmatrix} \mathbf{H}_\mathcal{R}^\top \\ \mathbf{H}_{\mathcal{L}_i}^\top \end{bmatrix} + \mathbf{R} \tag{2.18}$$
$$\mathbf{K} = \begin{bmatrix} \mathbf{P}_{\mathcal{R}\mathcal{R}} & \mathbf{P}_{\mathcal{R}\mathcal{L}_i} \\ \mathbf{P}_{\mathcal{L}_i\mathcal{R}} & \mathbf{P}_{\mathcal{L}_i\mathcal{L}_i} \end{bmatrix} \begin{bmatrix} \mathbf{H}_\mathcal{R}^\top \\ \mathbf{H}_{\mathcal{L}_1}^\top \end{bmatrix} \mathbf{Z}^{-1} \tag{2.19}$$
$$\hat{\mathbf{x}}^+ = \hat{\mathbf{x}} + \mathbf{K}\hat{\mathbf{z}} \tag{2.20}$$
$$\mathbf{P}^+ = \mathbf{P} - \mathbf{K}\mathbf{Z}\mathbf{K}^\top \tag{2.21}$$

This set of equations is applied each time a landmark is measured and updated.

[6]The common filtering notation ν for measurement noise should not be confused with the common dynamics notation $\mathbf{v}$ for velocity, as used below

2.2.2.3 Landmark initialization

Landmark initialization happens when landmarks that are not yet mapped are discovered by the vehicle, and are in turn incorporated in the map.

Landmark initialization is simple in cases where the sensor provides information about all the degrees of freedom of the new landmark, as when using range-and-bearing sensors. When this happens, the observation function $h()$ needs only to be inverted to compute the new landmark's state $\mathcal{L}_{n+1}$ from the vehicle state $\mathcal{R}$ and the observation y_{n+1}.

In cases where the sensor does not provide enough degrees of freedom for the function $h()$ to be invertible (e.g. when using bearing only sensors such as a monocular camera), this lacking information is introduced by a further parameter called a "prior". The inverse observation model $g() = h^{-1}()$ including a prior is

$$\mathcal{L}_{n+1} = g(\mathcal{R},\mathbf{y}_{n+1}, \mathbf{s}) \tag{2.22}$$

where the prior $\mathbf{s}$ is Gaussian with mean $\hat{\mathbf{s}}$ and covariances matrix $\mathbf{S}$[7]. The landmark's mean and all Jacobians can then be computed:

$$\hat{\mathcal{L}}_{n+1} = g(\hat{\mathcal{R}},\mathbf{y}_{n+1}, \hat{\mathbf{s}}) \tag{2.23}$$

$$\mathbf{G}_{\mathcal{R}} = \frac{\partial g(\hat{\mathcal{R}},\mathbf{y}_{n+1}, \hat{\mathbf{s}})}{\partial \mathcal{R}} \tag{2.24}$$

$$\mathbf{G}_{\mathbf{y}_{n+1}} = \frac{\partial g(\hat{\mathcal{R}},\mathbf{y}_{n+1}, \hat{\mathbf{s}})}{\partial \mathbf{y}_{n+1}} \tag{2.25}$$

$$\mathbf{G}_{\mathbf{s}} = \frac{\partial g(\hat{\mathcal{R}},\mathbf{y}_{n+1}, \hat{\mathbf{s}})}{\partial \mathbf{s}} \tag{2.26}$$

and then the landmark's variance $\mathbf{P}_{\mathcal{L}\mathcal{L}}$, and its covariance with the rest of the map are computed $\mathbf{P}_{\mathcal{L}\mathbf{x}}$

$$\mathbf{P}_{\mathcal{L}_{n+1}\mathcal{L}_{n+1}} = \mathbf{G}_{\mathcal{R}}\mathbf{P}_{\mathcal{R}\mathcal{R}}\mathbf{G}_{\mathcal{R}}^{\top} + \mathbf{G}_{y_{n+1}}\mathbf{R}\mathbf{G}_{y_{n+1}}^{\top} + \mathbf{G}_{\mathbf{s}}\mathbf{S}\mathbf{G}_{\mathbf{s}}^{\top} \tag{2.27}$$

$$\mathbf{P}_{\mathcal{L}\mathbf{x}} = \mathbf{G}_{\mathcal{R}}\mathbf{P}_{\mathcal{R}\mathbf{x}} = \mathbf{G}_{\mathcal{R}}\begin{bmatrix}\mathbf{P}_{\mathcal{R}\mathcal{R}} & \mathbf{P}_{\mathcal{R}\mathcal{M}}\end{bmatrix} \tag{2.28}$$

These results are then appended to the state mean and covariance matrix:

$$\hat{\mathbf{x}}^{+} = \begin{bmatrix}\hat{\mathbf{x}} \\ \hat{\mathcal{L}}_{n+1}\end{bmatrix} \tag{2.29}$$

$$\mathbf{P}^{+} = \begin{bmatrix}\mathbf{P} & \mathbf{P}_{\mathcal{L}\mathbf{x}}^{\top} \\ \mathbf{P}_{\mathcal{L}\mathbf{x}} & \mathbf{P}_{\mathcal{L}\mathcal{L}}\end{bmatrix} \tag{2.30}$$

The above method to introduce an invented prior to overcome the lack of initialization information for a landmark might seem trivial but it is not. With $\mathbf{s}$ being an unknown

[7]The initial values of the mean and uncertainty of the prior can be calculated using reasonable guesses based on the problem under study. In this work, a more robust approach is taken, where a reasonable value is taken for the mean and the initial uncertainty is set to a very large number.

parameter, setting upper and lower validity limit for it would break the linearity condition for the EKF filter and make it fail. Advanced ways to deal with this problem and introduce a proper prior are discussed in [46] and [53] and will not be discussed here. In the application of this approach later in this work (Section 3.4), the methodology as implemented by the authors of those papers will be applied.

An interesting approach for full landmark initialization is to use a pointable laser beam or scanning lidar to provide full range information for every landmark [31]. Although promising, this approach is left to be investigated in the future.

2.2.2.4 Initialization of a-priori known landmarks

A landmark detected in SLAM is previously largely unknown. Based on the discussions above concerning the meaning of covariance and of SLAM, it can be said that its estimated position in the SLAM map is correlated to the navigational state of the vehicle that detected it. The implication of this is that landmarks detected in SLAM cannot have an navigational uncertainty smaller than that of the vehicle that observed them. Further, the navigational uncertainty of the vehicle drop below its initial value only by observing previously unknown landmarks.

There can exist however landmarks whose position in the world is assumed to be a-priori known. These landmarks can be identified as such when observed by the lander. In contrast to the previous case, the position of the landmarks can then be seen as not correlated to the navigational state of the vehicle. It then follows that vehicle-landmark covariances in the covariance matrix can be set to zero.

According to the observations above, Equation 2.30 can be modified for the initialization of a-priori known landmarks. The mean value $\hat{\mathcal{L}}_{n+1}^{\text{A-priori}}$ can be directly assigned, e.g. from a landmark database and the covariance matrix of the SLAM map can be updated as:

$$\hat{\mathbf{x}}^{+} = \begin{bmatrix} \hat{\mathbf{x}} \\ \hat{\mathcal{L}}_{n+1}^{\text{A-priori}} \end{bmatrix} \tag{2.31}$$

$$\mathbf{P}^{+} = \begin{bmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{\mathcal{L}\mathcal{L}} \end{bmatrix} \tag{2.32}$$

where $\mathbf{0}$ are zero matrices of the appropriate size.

The practical meaning of this, is that by forcing the vehicle-landmark covariances to be zero, as can be seen in the equation set for the map correction after the observation of an existing landmark (Equations 2.17 to 2.21), the non-diagonal elements of the covariance matrix $\mathbf{P}$ will be zero, meaning that in the final covariance matrix correction in Equation 2.21, the covariances can be reduced to levels below the initial.

In more intuitive terms, a-priori known landmarks can be thought of as navigational anchor points. Once the vehicle gets hold of a few, it can reduce its navigation state uncertainty based on them. With the new reduced uncertainty it can then in turn reduce the uncertainty of the rest of the map. In addition, a-priori unknown points also serve the purpose to observe certain "differential" parts of the vehicle navigation state, like e.g. velocity.

2.2.3 Sensor models

Mathematical models of the measurements taken from the sensors discussed in Section 2.1 can be integrated into the above EKF-SLAM framework. In particular, a model for the INS can be used as update function $\mathbf{f}()$ used in the prediction step, and models and their corresponding inverse models for the exteroceptive camera and lidar can be used as the measurement $\mathbf{h}()$ and inverse measurement $\mathbf{g}()$ functions respectively. These sensor models will be described here.

2.2.3.1 Update model - INS accelerometer

In inertial-based navigation methods, an accelerometer is used to update the navigation state of a vehicle. The initial lander 3-DOF navigation state is comprised of the position and velocity vectors $\mathcal{R} = [\mathbf{p} \quad \mathbf{v}]^\top$. Its update function (eq. 2.3) $\mathbf{f}()$ can be based on basic dynamics equations:

$$\mathcal{R}^+ = \mathbf{f}(\mathcal{R},\mathbf{u}_{INS}) = \begin{bmatrix}\mathbf{p}^+\\ \mathbf{v}^+\end{bmatrix} = \begin{bmatrix}\mathbf{p} + T_s \cdot \mathbf{v} + \frac{1}{2}T_s^2 \cdot \mathbf{u}_{INS}\\ \mathbf{v} + T_s \cdot \mathbf{u}_{INS}\end{bmatrix} \tag{2.33}$$

where the control vector from Equation 2.3 has been replaced with the acceleration measured by the INS accelerometer, and T_s is the update time-step; in our case the accelerometer sampling time-step. The accelerometer measures the accelerations applied to the lander:

$$\mathbf{a}_{applied} = \mathbf{a}_{grav} + \mathbf{a}_{thrust} \tag{2.34}$$

where $\mathbf{a}_{grav}$ is the gravitational acceleration vector and $\mathbf{a}_{thrust}$ the acceleration vector applied by the thrusters. The gravity acceleration is assumed here to be perfectly known, and the thrust accelerations ideal[8]. The combined applied acceleration on the lander $\mathbf{a}_{applied}$ is then measured by the INS, adding the related measurement noises $\boldsymbol{\nu}_{INS}$

$$\mathbf{u}_{INS} = \mathbf{a}_{applied} + \boldsymbol{\nu}_{INS} \tag{2.35}$$

The elements $\mathbf{F}_\mathcal{R}$ and $\mathbf{F}_\mathbf{u}$ of the covariance matrix update equation (Equation 2.5) are according to Equations 2.8:

$$\mathbf{F}_\mathcal{R} = \begin{bmatrix}\mathbf{I} & T_s \cdot \mathbf{I}\\ \mathbf{0} & \mathbf{I}\end{bmatrix} \qquad and \qquad \mathbf{F}_\mathbf{u} = \begin{bmatrix}\frac{1}{2}T_s^2 \cdot \mathbf{I}\\ T_s \cdot \mathbf{I}\end{bmatrix} \tag{2.36}$$

$\mathbf{P}_{\mathcal{R}\mathcal{R}}$ below is the initial covariance matrix and $\mathbf{U}$ is the control covariance matrix:

$$\mathbf{P}_{\mathcal{R}\mathcal{R}} = \begin{bmatrix}\boldsymbol{\sigma}_\mathbf{x}^2 & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\sigma}_\mathbf{v}^2\end{bmatrix} \qquad and \qquad \mathbf{U} = \boldsymbol{\nu}_{INS}^2 \tag{2.37}$$

where in turn $\boldsymbol{\sigma}_\mathbf{x}$ and $\boldsymbol{\sigma}_\mathbf{v}$ are the standard deviations in position and velocity, and $\mathbf{0}$ and $\mathbf{I}$ are 3×3 zero and diagonal unit matrices respectively.

[8]These assumptions can be made in this work, considering a relatively small timescale of a few minutes is investigated.

2.2.3.2 Observation model – Camera/Bearing-only sensors

In camera-based navigation a measurement of a point landmark corresponds to the coordinates of the pixel where this landmark is projected in the image: $y_i = (u_i, v_i)$. The landmark should first be expressed in the camera focal reference frame and then projected to the camera sensor following the pin-hole camera principle. Defining the focal frame as $\mathcal{S}\{\text{RDF}\}$ (Right-Down-Front) with coordinates XYZ and the image frame as $\mathcal{I}\{\text{RD}\}$ (Right-Down) with coordinates xy, as seen in Figure 2.6, the pin-hole projection equation can be given as [46]:

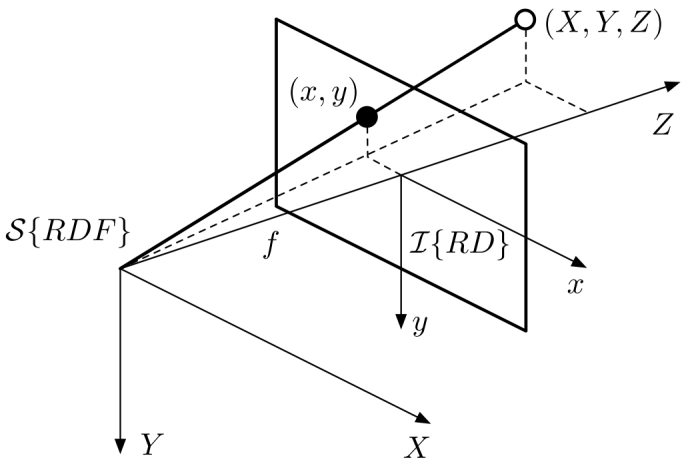

Figure 2.6: The camera RDF coordinate system, and the geometric principles of the projection of 3D points onto the camera focal plane. All points (X, Y, Z) in the back-projected ray of the image point (x, y) satisfy $[X, Y, Z]^\top = s[x,y,1]^\top$ with $s \in [0, \text{inf})$ [46].

$$h = s\underline{\mathbf{u}} = \mathbf{K}_s\mathbf{K}_f\mathbf{P}_0\underline{\mathbf{p}}^{\mathcal{S}} \tag{2.38}$$

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} s_u & 0 & u_0 \\ 0 & s_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.39}$$

In Equation 2.38, $\underline{\mathbf{p}}^{\mathcal{S}}$ are the homogeneous coordinates of the projected point in the sensor frame[9]. The coordinates are transformed from the world frame based on the equation $\underline{\mathbf{p}}^{\mathcal{S}} = \mathbf{H}^{\mathcal{WS}}\underline{\mathbf{p}}^{\mathcal{W}}$, where $\mathbf{H}^{\mathcal{WS}}$ is the homogeneous transformation matrix from the world frame to the sensor frame (see Appendix A.1). The matrix $\mathbf{P}_0$ is often referred to as the normalized projection matrix, representing the projection performed by the focal element into the image plane $Z = 1$ (in the focal frame). The matrix $\mathbf{K}_f$ represents a scaling from having the image plane at a distance $Z = f$ different to unity. $\mathbf{K}_s$ represents the conversion of the idealized projection to a pixel value.

In Equation 2.39, s_u and s_v are the number of pixels per unit length for each axis, u_0 and v_0 are the coordinates of the principal point, i.e. the point where the optical axis

[9]Homogeneous coordinates are often used in projective geometry as explained e.g. in [57]. For the their limited use in this work they can be considered as the usual coordinate vector padded with a 1

intersects the image plane, and f is the focal length of the camera. The values $a_u = f \cdot s_u$ and, $a_v = f \cdot s_v$, along with the pixel counts u_0 and v_0 of a given camera, are defined as the intrinsic camera parameters [46]. The parameter s of the projection ray connecting the projected to the world point (as seen in Figure 2.6) is the distance from the 3D point to the focal plane, and is called the depth of that point[10].

A useful concept in vision is the field-of-view (FOV) of a sensor, expressed in angle units and derived based on the intrinsic parameters for each axis as:

$$FOV_u = 2arctan\frac{u_0}{a_u} \qquad and \qquad FOV_v = 2arctan\frac{v_0}{a_v} \tag{2.40}$$

The inverse of the projection function is called back-projection. It maps pixel coordinates on the focal plane of a camera, back to 3D space. Following the inverse of the above process, the back projection equation is

$$g = \underline{\mathbf{p}}^{\mathcal{S}} = s\mathbf{K}^{-1}\underline{\mathbf{u}} \tag{2.41}$$

with

$$\mathbf{K}^{-1} = \begin{bmatrix} 1/a_u & 0 & -a_v u_0/a_u a_v \\ 0 & 1/a_v & -v_0/a_v \\ 0 & 0 & 1 \end{bmatrix} \tag{2.42}$$

The back-projected point can then be transformed to the world frame by the equation $\underline{\mathbf{p}}^{\mathcal{S}} = \mathbf{H}^{\mathcal{SW}}\underline{\mathbf{p}}^{\mathcal{W}}$, where $\mathbf{H}^{\mathcal{SW}} = \mathbf{H}^{\mathcal{WS}^{-1}}$.

2.2.3.3 Observation model – Lidar/Range-and-bearing sensors

A basic measurement model for the beam i of a lidar beam bundle can be simply given as

$$h_i = \underline{\mathbf{p}}_i^{\mathcal{S}} = \mathbf{H}^{\mathcal{WS}}\underline{\mathbf{p}}_i^{\mathcal{W}} + \mathbf{v}_i \tag{2.43}$$

where $\underline{\mathbf{p}}_i^{\mathcal{W}}$ are the coordinates of the impact point of beam i in the world frame, $\underline{\mathbf{p}}_i^{\mathcal{S}}$ the coordinates of the same point in the sensor frame, $\mathbf{H}^{\mathcal{WS}}$ is the frame transformation between the two (essentially a rotation and a translation, see Appendix A.1), and $\mathbf{v}_i$ is a zero-mean Gaussian noise, $\mathbf{v} \sim N\{0; \mathbf{R}_i\}$.

According to [54], assuming that the range error and bearing error from a lidar beam are small and uncorrelated, then it can be shown that the covariance for a single ranger measurement is described by

$$\mathbf{R}_i = \rho_i^2\sigma_\phi^2\mathbf{I} + \left(\frac{\sigma_\rho^2}{\rho_i^2} - \sigma_\phi^2\right)\underline{\mathbf{p}}^{\mathcal{S}}{}_i\underline{\mathbf{p}}^{\mathcal{S}}{}_i^{\top} \tag{2.44}$$

[10]This prior s, is the same one discussed in Section 2.2.2.3

where σ_ϕ is the standard deviation of the bearing error (in radians), σ_ρ is the standard deviation of the range error, $\rho_i = \|\underline{\mathbf{p}}^{\mathcal{S}}_{\mathbf{i}i}\| = \sqrt{\underline{\mathbf{p}}^{\mathcal{S}\top}_{\mathbf{i}i}\underline{\mathbf{p}}^{\mathcal{S}}_{\mathbf{i}i}}$, is the range from the sensor to the laser beam impact point, and as before, $\mathbf{I}$ and $\mathbf{0}$ are diagonal unit and zero 3×3 matrices respectively.

The inverse observation function is given as

$$g = \underline{\mathbf{p}}^{\mathcal{F}}_i = \mathbf{H}^{\mathcal{S}\mathcal{F}-1}\underline{\mathbf{p}}^{\mathcal{S}}_i + \mathbf{v}_i \tag{2.45}$$

2.2.4 Sensing the landing environment for navigation

As seen in Sections 1.4 and 2.1, there are three aspects of the environment that can provide information to the exteroceptive navigation sensors: the optical environment, the thermal environment, and the surface of Enceladus.

The optical and thermal environments can be sensed using the corresponding camera type. Strongly identifiable points in the resulting 2D images (e.g. corners, local intensity maxima, etc), can be uniquely associated to particular landmarks in 3D space. This step is called feature detection. By memorizing the neighborhoods of these features in the 2D image, a signature for a given feature can be created. In the succeeding feature matching step, it will be able for the same feature to be identified in different images. The same landmark can then be tracked across images. The process is illustrated in Figure 2.7. A matched feature can then serve as the camera measurement in the above SLAM formulation, to help navigate the lander and localize the observed landmark itself (Figure 2.8). Surveys of feature detection, description, and matching methods can be found in [59] and [60].

On Enceladus, salient features can be identified in visual images - such sensing however is restricted to illuminated areas. This restriction is non-existent for thermal images. Images from infrared cameras are usually less sharp than visual images making it more difficult to identify features in them [63]. Thermal cameras can also sense local sources of heat, such as the plume sources in the present case.

The surface of Enceladus can be sensed in a more straightforward way with range sensors. If a lidar is used, a point cloud is generated from every measurement, comprising the impact points of the laser beam with the surface, expressed in the lidar sensor frame. For most planetary landing applications, the terrain can be assumed to be flat in the large scale. An observation model for the lidar measurement can then calculate the lidar ray-surface intersection point by assuming that a plane more or less coincident with the planetary surface. For landing in a canyon terrain this assumption cannot be made, and a Digital Terrain Model (DTM) of the landing terrain is necessary to calculate the lidar beam impact points.

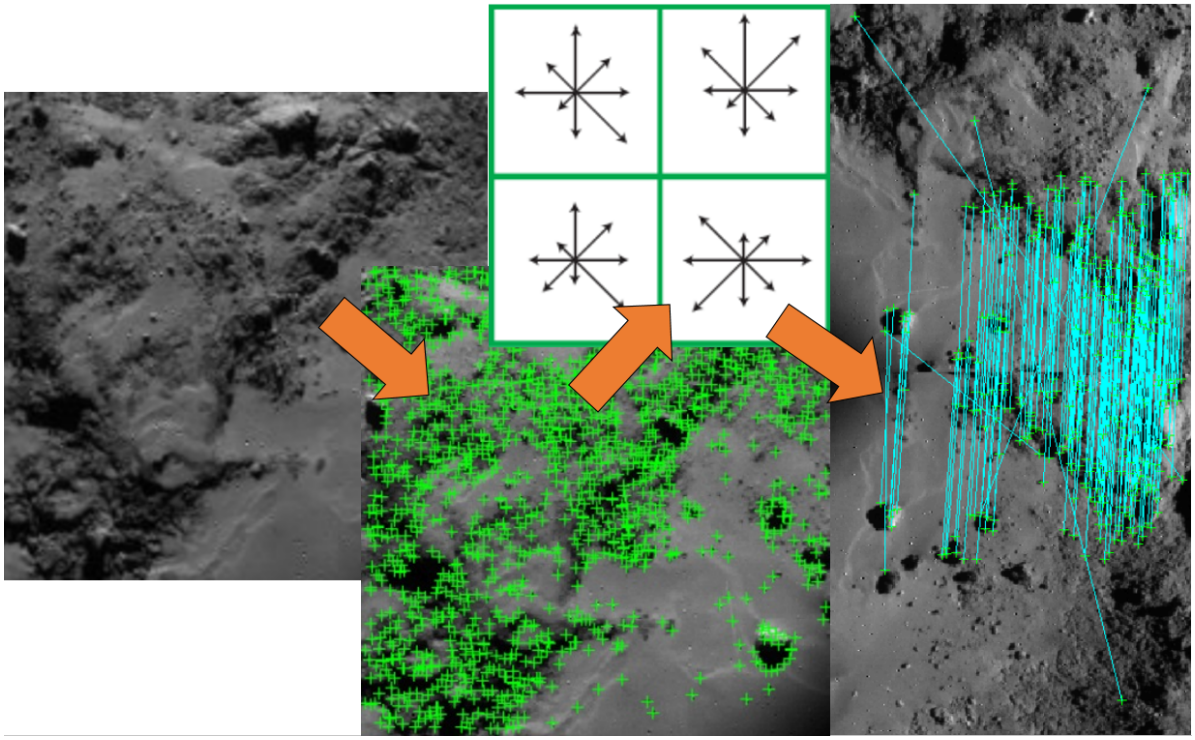

Figure 2.7: An illustration of feature detection, description, and matching in 2D images: salient features are detected in an image and a descriptor is assigned to each. In this case a bundle of vectors indicating the intensity gradient of pixels surrounding the detected feature is used as a descriptor (a simplified version of the SIFT descriptor [58]). Descriptors from different images can then be compared and matched. All three marked images were provided as shown by R. Jakob of Uni. Bundeswehr Munich.

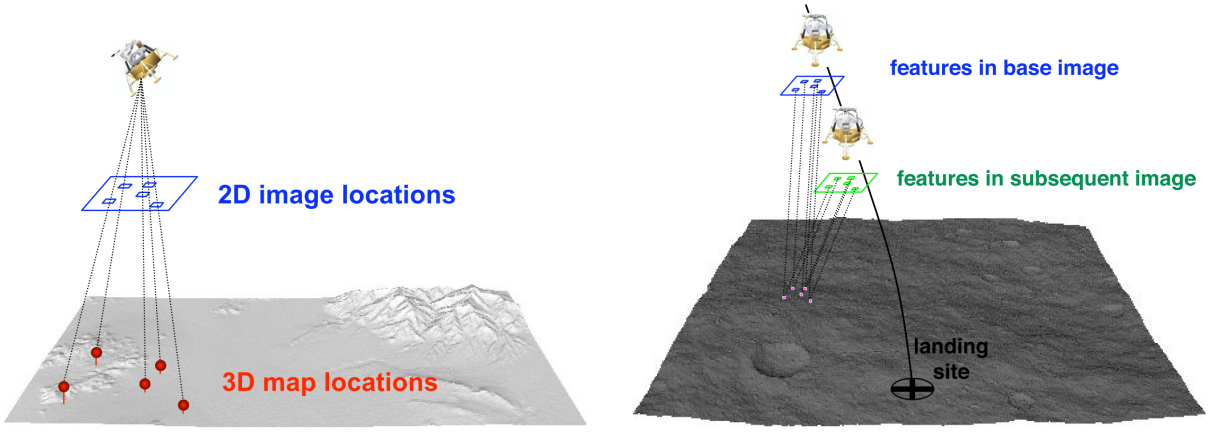

Figure 2.8: Correspondences between 2D image matched features and 3D map locations (left), and between matched features in a base image and a subsequent image (right) [61]

An alternative for lidar navigation is using point cloud registration to fit the point cloud generated by a lidar measurement to the DTM of the landing terrain. This way, the position of the lander relative to the DTM (assumed to be given in a planetary inertial coordinate frame) can be estimated. A popular registration method is the Iterative Closest Point (ICP) algorithm [66].

2.2.5 A-priori knowledge of the landing environment for navigation

As discussed in Section 2.2.2.4, landmarks on the surface of Enceladus can be broken down in two categories: a-priori known and a-priori unknown.

In the context of an accurate planetary landing, ground operators can prepare a database of a-priori known landmarks based on orbital reconnaissance before the landing starts. This is done following the process shown in Figure 2.9. In that process, a-priori known landmarks have been localized in the inertial frame of a planetary surface and given a corresponding uncertainty value. A descriptor has been assigned to each landmark so that it can be identified in further camera images. A Digital Terrain Model (DTM) has been first created by orbital reconnaissance data (from e.g. stereo imaging of the surface). The high resolution camera image corresponding to the region of the DTM is draped over it. Features detected in that image can then be back-projected onto the DTM, back to the distinctive landmarks on the surface that produced the features [62]. With this method however, features are generated only for regions for which images exist, and for the particular illumination conditions that the images were made in. To generate features particular to a wider range of illumination conditions, the DTM can be input to a high-performance graphics tool (e.g. PANGU [64]), and features can be identified and described in the generated virtual images for the expected illumination conditions and landing approach direction.

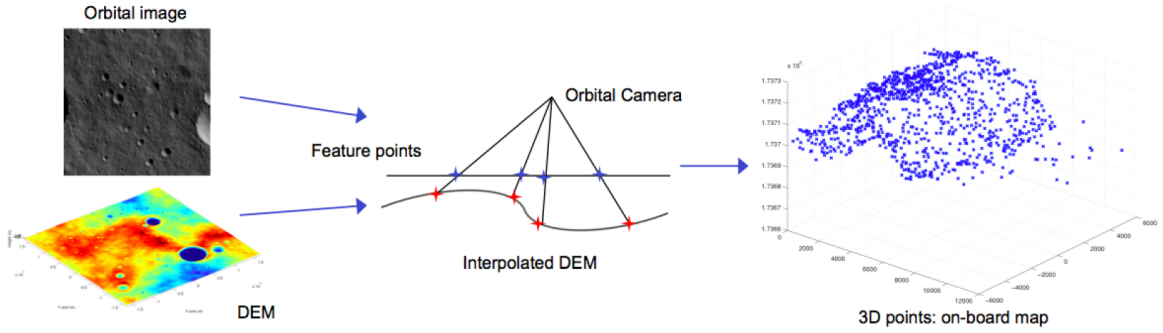

Figure 2.9: Process for the off-line generation of an a-priori known landmarks database [62]. DEM is used here in lieu of DTM. The blue line in the middle image represents the camera imaging plane. Rays originating from the focal point and crossing the imaging plane at the coordinates of the features will intersect the DTM at the locations of the landmarks the produced said features (right image).

The inertial location of identified features can be estimated, with the corresponding position uncertainties due to the resolution of the DTM, uncertainties in the planetary shape parameters, uncertainties in the correlation between the planetary shape model and the 2D camera images draped over it (Map-tie error), and others [62]. The inertial location mean values and uncertainties of a-priori known landmarks as well as a descriptor for the corresponding feature of each can be then made available to the lander before landing starts.

There is a minimum altitude down to which the descriptors of a-priori known points

would be detectable in images: at some point as the lander approaches the surface, the resolution of sensed images becomes higher than that of the terrain data that was used to generate the descriptors of the a-priori known features. After that point those descriptors become meaningless and the lander can no longer observe the a-priori known landmarks.

The best orbital images for e.g. the Moon and Mars can reach a ground sample distance (GSD)[11] of under ∼0.25 m. Due to the challenges due to the instability of polar orbits around Enceladus, and the polar illumination conditions and the general rough topography on the south polar terrain, it can be assumed that the resolution of the maps acquired from any orbital reconnaissance of that terrain will not be as high as that. As a general guide, a DTM can be derived with a resolution ∼4 times the pixel scale of the input images. For images with a resolution of 0.25 - 0.5 m/pixel, a DTM of 1-2 m can be generated with vertical precision in the tens of centimeters [65].

Another group of landmarks that can be localized a-priori are the plume sources themselves. Using optical imagery as above (see e.g. [10]), but also thermal measurements, the locations of plume sources on the planetary surface can be identified along with the corresponding uncertainties. Generating an identifiable optical feature descriptor for these plume source landmarks would be challenging due to the difficult illumination conditions on the canyon bottoms, but not impossible. The feature description and matching problem for thermal images of the plume sources can be simplified by assuming that during the final phases of landing, only a few plume sources will be visible, most likely only one. The detection and matching problem can then be simplified to matching the local intensity maxima within a certain pixel distance in a thermal image, with sensed surface temperatures above a given threshold.

For lidar navigation the landing terrain DTM discussed above can be used as the basis for an observation model, or for ICP registration. It is not certain however that the GSD of the DTM will be adequate for either application due to its relatively low GSD. Arguably, this is one of the reasons why ICP is more popular for rendezvous and close proximity satellite operations, where a 3D model of the target object is more readily available (see e.g. [66]).

2.2.6 System and operational considerations

The following preliminary navigation concept ties the information presented in this section together to give a detailed operations concept from the point of view of the navigation function for landing near a plume source on Enceladus.

An EKF-SLAM formulation is followed, with a SLAM map initialized with the lander state only. During operations features are detected, matched, and then initialized in the SLAM map. Detected features can be a-priori known, or a-priori unknown; they can also be detected by either the thermal or the optical camera. The SLAM map vector and covariance matrix are separated into separate “compartments” for the lander state

[11] In remote sensing, ground sample distance (GSD) in an image or DTM is the distance between pixels, measured on the ground. E.g., in an image with a one-meter GSD, adjacent pixels represent locations 1 meter apart on the ground.

and each landmark type: known visual, unknown visual, known thermal, and unknown thermal (Figure 2.10).

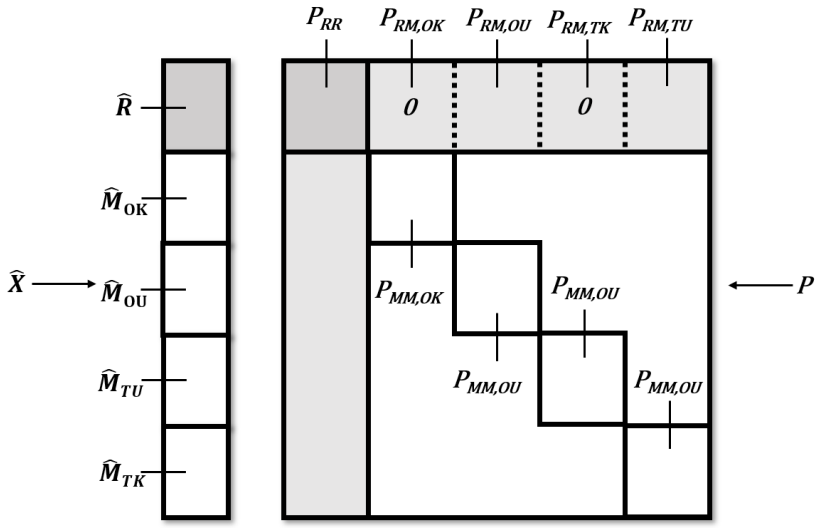

Figure 2.10: The EKF-SLAM map used in the EnEx lander navigation function. The initials for each compartment are: OK for optical (a-priori) known, OU for optical unknown, TK for thermal known, and TU for thermal unknown. See text in previous sections for further reference. Adapted from [46].

When an a-priori unknown feature is detected and matched, its off-diagonal covariances with the lander state are initialized normally. When an a-priori known landmark is detected, these covariances set to zero (sec. 2.2.2.4). When a landmark already in the SLAM map is observed, the correction step is performed. Depending on whether the detected landmark is a-priori known or unknown, the lander navigation state error can be reduced below its initial levels or not.

In the following, observation and navigation based on a-priori known points is called absolute navigation. On the contrary, navigation besed on a-priori unknown points is called relative navigation.

A plume source can be as an a-priori known or unknown thermal landmark on the map, depending on the accuracy of its position knowledge before landing start. With repeated observations it can be localized with increasing accuracy in the SLAM map. As the lander navigational state improves via absolute camera or lidar navigation, the accuracy of the plume source location will also improve.

The lidar can be used to give full observations of points on the surface. It should be noted that the points sensed by the lidar in the lidar sensor frame, after back-projected, are not located in the same inertial frame as those e.g. of the camera, but are defined in a new one. This frame is not identical to the World frame, and an extra transformation matrix should be found between the two if the lidar points are to be expressed in the World frame. This tranformation matrix can be given e.g. by ICP and registration methods, but as discussed in Section 2.2.5 this will very likely not be practical for the present case.

To overcome this issue, the new inertial frame defined by the back-projection of lidar points will be defined as the hazard frame. As will be seen later, this is adequate for the purposes of this study, where the main goal is to navigate relative to hazards (implicitly existing in the hazard frame as the name implies) and to the target plume source (which can be defined in the hazard frame as will be discussed later). Landmarks defined in the World frame, can no longer be used once a hazard frame has been defined. This approach of using a hazard frame is defined as hazard relative navigation, and is common in safe and accurate landing studies (e.g. [25]).

The lander and plume source position mean values and uncertainties can potentially be forwarded for use in other functions, e.g. the HDA for hazard evaluation and re-targeting.

2.3 Hazard detection and avoidance (HDA)

Landing safety is a key of planetary landing missions. In all planetary landing missions up to now the spacecraft follows a predetermined trajectory to its nominal landing site. Missions landing in less challenging terrains and with less strict accuracy requirements can afford this lack of capability, as a wide, flat, and safe landing area can be chosen to ensure a given landing success probability. This option however is not available to the EnEx lander that must land in an extremely challenging terrain, with a high degree of accuracy, and, due to the planetary protection sensitivity of the landing area, even stricter safety requirements than other similar landing concepts.

As seen in Section 1.5, the EnEx lander must be able to sense and assess terrain hazards and make the autonomous decision to select a new suitable landing site if necessary. A central feature of this approach is the imprecise and uncertain nature of the information coming from the sensors, due to environmental, sensor design, and other uncertainties.

In the following, the types of safety terrain characteristics that can be detected with the sensors presented in Section 2.1 will be discussed. Data fusion methods that can be used to provide a fused landing assessment based on the information available will then be investigated.

2.3.1 Terrain safety features

Each exteroceptive sensor type produces independent measurements of the terrain. For hazard sensing purposes the process of sensor output co-registration is beneficial: if the lidar and camera sensors are co-aligned with enough accuracy, and their angular resolutions coincide, an enhanced point cloud can be created from the combined illumination, thermal, and point cloud measurements by assigning one pixel from each camera to each point from the point cloud. As the fields-of-view (FOV) of each sensor might differ, the new co-registered enhanced point cloud will correspond to the smaller FOV.

The enhanced point cloud can then be used as the input to the hazard detection process. Safety feature values for each point in this enhanced point cloud can then be added as further information layers in the format of $f_f^S(x,y)$, where f in the subscript refers to a given feature as measured by the sensor S (either lidar - L, optical camera - C, or

thermal camera - T). This notation is adapted from [71].

Wherever features must be calculated within a certain area (e.g. the local slope), this area is the lander footprint (given in Table 1.2), i.e. a circle with a diameter of 4 m.

The point cloud produced by the lidar can be used to estimate the slope and roughness features. In [72] and [71], least median of squares (LMedSq) regression is used to estimate the local plane parameters a, b, and c at location (x, y) of the point cloud. Given these parameters, the slope (θ) is obtained by

$$f_{\theta}^{L}(x,y) = cos^{-1}\left(\frac{1}{\sqrt{a^2+b^2+1}}\right) \tag{2.46}$$

The roughness feature r is then the residual of the point cloud and the fitted plane at location (x, y)

$$f_{r}^{L}(x,y) = |d(x,y) - (ax + by + c)| \tag{2.47}$$

where $d(x, y)$ is the range data from the lidar.

A measure of terrain roughness as observed by the camera is obtained by computing the local intensity variance at each pixel [71, 72]:

$$f_{r}^{C}(x,y) = \frac{1}{|W|-1}\sum_{i,j\in W}|c(i,j)-\mu|^2 \qquad with \qquad \mu = \frac{1}{|W|}\sum_{i,j\in W}c(i,j) \tag{2.48}$$

where W is a sub-region of the camera intensity image $c(x, y)$, $|W|$ is the number of pixels in W, and μ is the mean intensity in sub-region W.

Individual obstacle detection from optical camera images can be seen as a clustering or image segmentation problem. The high saliency of either the illuminated faces or the shadows of obstacles in optical images can be taken advantage of to separate the image into "obstacle" and "non-obstacle" groups or clusters. In [73], the K-means clustering algorithm is applied to the obstacle detection problem. A boolean information layer can then be created based on the clustering analysis indicating the presence or not of an obstacle (o) for every point on the map:

$$f_{o}^{C}(x,y) = \begin{cases} 1, & \text{for } (x,y) \in \text{O} \\ 0, & \text{otherwise} \end{cases} \tag{2.49}$$

where O is the set of pixel locations in the camera image identified as obstacles.

As seen in Section 2.1.3, the strength and shape of lidar returns varies with the texture of the surface reflecting the beam. Smooth and reflective surfaces will give a sharper return signal, while rough and less reflective surfaces will tend to give a more spread out return signal; consequently snow covered surfaces will return sharper signal peaks than icy surfaces. It becomes then conceivable that lidar beams can act as texture sensors,

albeit potentially with high uncertainty, as it cannot always be clear whether a surface property or some other effect produced the observed return signal. In addition, different types of lidar have different resolutions of the return signal waveform (See Section 2.1.3). Here the simplification will be made that each lidar beam will directly return a binary snow/ice texture value. The boolean map describing the surface texture as measured by the lidar is then:

$$f_t^L(x,y) = \begin{cases} 1, & \text{for } (x,y) \in \text{S} \\ 0, & \text{otherwise} \end{cases} \tag{2.50}$$

where S is the set of pixel locations in the lidar input identified as snow-covered.

Using a thermal camera, the local thermal environment near a plume source can also be leveraged to extract information on the texture of the terrain. Heat originating from a plume source will warm the surrounding ice in a very localized manner (Section. 1.4). However, due to their significant differences in thermal inertia, the super-fine snow and the exposed ice will have a noticeable temperature difference. This temperature difference can be observable in thermal images.

To extract explicit texture information from thermal images, a two-step process is applied. First, the thermal background due to the plume source hot spot will be calculated. It can be assumed that by orbital reconnaissance and/or analysis, the location of the plume source will be roughly known, as will be the maximum surface temperature near it, the ambient surface temperature at some distance from it, and the surface temperature distribution around it. The simplification is further made that surface temperature near a plume source follows a Gaussian distribution with a standard deviation in the order of ∼30 m. Knowing the maximum and minimum temperatures, the plume source location, and the standard deviation of the heat "spread", a background temperature map can then be calculated. By subtracting the background temperature map from the thermal camera temperature map sensed during landing, a "corrected" image with pixel values indicating the difference from the local image background is given:

$$T_{corr}(x,y) = T_{background}(x,y) - T_{orig}(x,y) \tag{2.51}$$

Pixels with negative values in these corrected images will then correspond to warmer temperatures, and thus exposed ice, and positive values should accordingly correspond to snow:

$$f_t^T(x,y) = \begin{cases} 1, & \text{for } T_{corr}(x,y) \geq 0 \\ 0, & \text{for } T_{corr}(x,y) < 0 \end{cases} \tag{2.52}$$

An a-priori perfect knowledge of the variables necessary to create a background temperature image is assumed above. The effect of uncertainties in all of these parameters will be investigated in Section 4.2.3. Alternatively, some or all of these parameters can be estimated in real time according to measurements with the thermal camera. Another option would be to generate the background image "agnostically", i.e. without assuming

any related previous knowledge. Such a solution would be e.g. to generate the background thermal image by blurring the original temperature image. The thermal texture detection process is illustrated further down in this thesis, in Figure 4.30.

The terrain safety features detectable by each sensor are summarized in Table 2.1.

Table 2.1: Terrain features that can be detected by each exteroceptive sensor

Sensor	Slope	Roughness	Obstacles	Texture
Lidar	+	+		+
Optical camera		+	+	
Thermal camera				+

A critical parameter not related to safety but to the mission science requirements is the scientific interest of the map points. For the purposes of the landing mission concept discussed here, scientific interest is equated to the proximity of a point to a plume source. According to the landing requirements (Section 1.5), the landing should not be farther away than a given maximum distance r_{max} from the plume source, but also, for planetary protection reasons, not closer than a minimum r_{min}:

$$r_{min}^2 \leq (x - x_{PlS,i})^2 + (y - y_{PlS,i})^2 \leq r_{max}^2 \tag{2.53}$$

where ($x_{PlS,i}$,$y_{PlS,i}$) are the coordinates of the ith plume source in the map. As seen in Section 1.5, reasonable values for r_{min} and r_{max} are 5 and 50 m respectively.

2.3.2 Reasoning under uncertainty

Principles of reasoning under uncertainty can be used to first assess landing safety based on terrain features observed by the sensors, and then combine this with information on landing site scientific interest. Three different frameworks for representation of data uncertainty have gained traction for safe and precise planetary landing applications [71, 72, 74, 75, 76]: fuzzy set theory, Bayesian probability (BP) theory, and Dempster–Shafer (DS) belief theory.

Fuzzy logic is a form of logic in which the truth values of variables may be any real number between 0 and 1, in contrast to Boolean logic, where variables may only be the integer values 0 or 1. It is employed to handle the concept of partial truth, where the truth value may range between completely true and completely false [67]. Based on that logic, fuzzy rule-based reasoning uses a set of linguistic statements, or rules, defined by a human expert. Each rule is of the form $IF\ C$, THEN A, where the condition C is composed of fuzzy input variables (e.g. terrain safety, fuel consumption, scientific return) and fuzzy operands (e.g. AND, OR, NOT) and the action A is a fuzzy output variable (e.g. landing site quality). The rules are evaluated based on their membership to fuzzy sets. Unlike traditional boolean logic where full membership in a set is required, a fuzzy-based reasoning allows for degrees of membership in multiple sets [75].

Part of the appeal of fuzzy systems is that they can be used for approximate reasoning. This is particularly important when there is uncertainty in the reasoning process, in

addition to imprecision in the data [75]. An additional advantage is that fuzzy rule statements model the human expert's domain knowledge and are thus intuitive to work with. Fuzzy logic rule evaluation involves only simple arithmetic calculations and is thus computationally light [75]. However, a fuzzy logic reasoning engine and its component rules need to be "hand-crafted" and tuned with close supervision by an expert. Also the informal nature of the fuzzy rule-sets might make the formal verification of performance and traceability an issue down the line.

Bayesian Networks (BN) are probabilistic graphical models representing a set of variables and their conditional dependencies via a directed acyclic graph (DAG). In a DAG, the nodes represent variables and the links between nodes represent causal dependence between them. The direction of a link indicates causality, and thus a dependence relationship. Nodes that exist at the same level are considered conditionally independent [74]. In the landing safety example, a Bayesian network could represent the probabilistic relationships between landing safety and terrain features. Given certain features present in an area, the network can be used to compute the probabilities of safe landing in that area.

Where fuzzy logic above deals in a subjective and linguistic classification of probability, Bayesian networks formulate the uncertainties of a problem in an objective and numerical way. A more formal and traceable representation of probabilities is then given. The defining negative aspect of Bayesian inference is that it requires all unknowns to be represented by probability distributions, needing the definition of potentially largely unknown probabilities of random events [69].

The evidential reasoning algorithm is based on the Dempster–Shafter (DS) mathematical theory of evidence. DS is based on belief functions and plausible reasoning, which is used to combine separate pieces of information (evidence) to calculate the probability of an event [68]. In other words, DS allows for the consideration of uncertainties in the knowledge of probabilities in a problem. An example would be instructive: spinning a coin is known to be fair and a simpler probabilistic approach can be followed. Betting on the outcome of a fight between the world's greatest boxer and the world's greatest wrestler would be a more difficult probabilistic problem. In this second problem, where the probabilities are unknown, DS theory allows one to consider the confidence one has in the probabilities assigned to the various outcomes [69].

Evidence theory is defined as an extension of probability theory that explicitly accounts for lack of information. It allows the combination of multiple observations without the need to know about a priori or conditional probabilities as in the Bayesian approach. It thus presents a more robust approach while preserving the formality of the probabilistic approach. Disadvantages include the possible invalidity of the assumption made by DS, that the pieces of evidence are statistically independent from each other, and the high computational complexity [70]

Some previous works have opted to implement all three fusion engines and add an extra decision fusion step to get a combined, and thus more robust, result [72, 76]. However, due to its simplicity, intuitiveness, and smaller computational requirements, a fuzzy based landing reasoning approach was chosen for implementation in this work. It will be described it in the following.

2.3.3 Fuzzy-reasoning-based terrain landing assessment

Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The process is illustrated with an intuitive example in Figure 2.11. The input variables, given in crisp numerical values are first fuzzified (assigned a linguistic label) via the respective membership functions. Using the fuzzified inputs, the fuzzy rules are then evaluated using the logical-fuzzy operands and combined based on the implication method. The combined results are finally defuzzified, giving crisp numerical output values. The process is explained in more detail in [77].

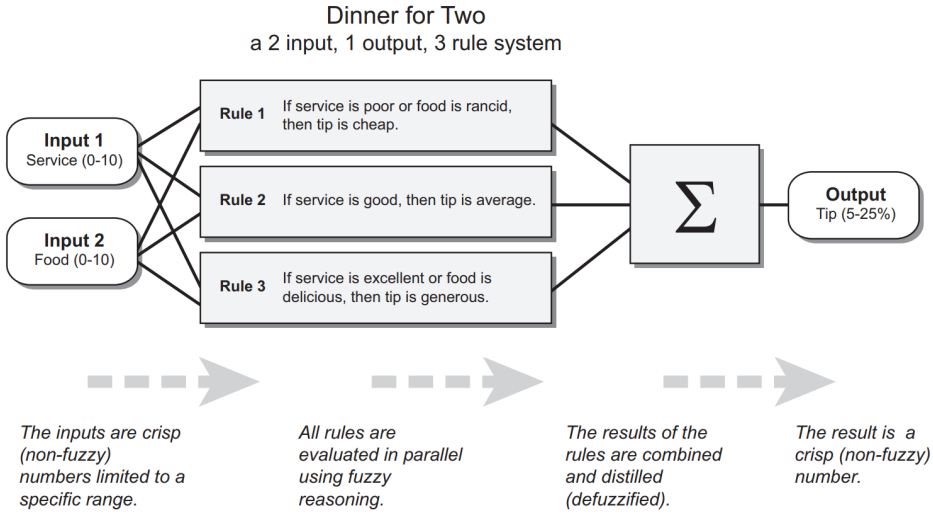

Figure 2.11: The fuzzy inference process, explained with an intuitive example use case: what tip is appropriate at a restaurant given service and food quality? [77].

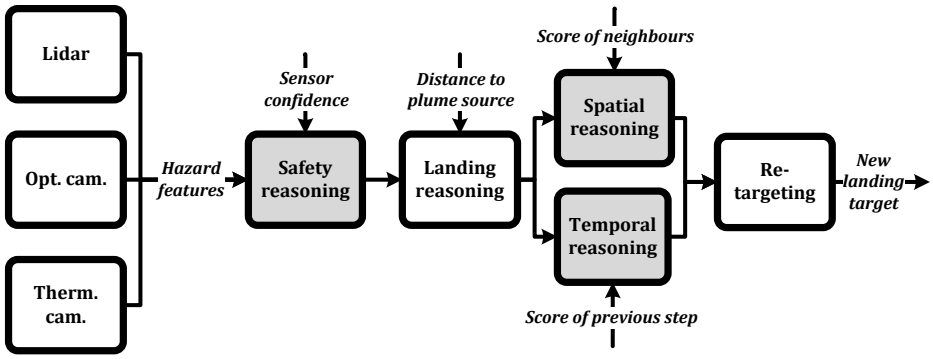

Figure 2.12: Block diagram of the selected fuzzy landing site selection approach. Phase where fuzzy reasoning is implemented are shown in grey. Adapted from [75]

A modified version of the fuzzy approach for landing reasoning given in [75] will be applied here. Determination of the best landing site is performed in three fuzzy reasoning phases as depicted in Figure 2.12.

In the first fuzzy step, the safety of the terrain is assessed using a set of rules to combine the safety extracted from the on-board sensors described in Section 2.3.1. The second

step involves fusing the combined safety information with information on the distance to the plume source. In the third phase, the landing quality of neighboring sites and landing quality at earlier times in the descent are combined using spatial and temporal rule sets. A final landing goodness score map is given for the input terrain at the end of this process. A re-targeting function then chooses the point in the map with the highest landing score, and commands a re-targeting to it. This process will be explained in detail in the following.

2.3.3.1 Fuzzy Safety Reasoning

Following the example of [71, 72, 75], the concept of a terrain safety score $s \in [0, 10]$ is used to represent a continuous, quantitative measure of the level of safety of a given point on the terrain. In a fuzzy context, the range of safety scores s can be classified into qualitative measures in the form of a linguistic fuzzy set with membership functions of adjacent grades having smooth and overlapping boundaries to avoid abrupt transitions. The fuzzy sets comprise the linguistic labels {HSAFE, MSAFE, MUNSAFE, HUNSAFE}, which stand for highly-safe, moderately-safe, moderately-unsafe, and highly-unsafe, respectively. The safety map for each sensor is obtained by applying the appropriate set of rules and membership functions. These fuzzy rules are listed in Tables 2.2 and 2.3 for the lidar and optical camera respectively.

Table 2.2: Lidar safety fuzzy rule set

Texture (pre-fused)	op.	**Roughness**	op.	**Slope**	**Terrain**
SNOW	and		and	STEEP	HUNSAFE
SNOW	and		and	SLOPED	HUNSAFE
SNOW	and		and	FLAT	MSAFE
ICE	and	VROUGH	or	STEEP	HUNSAFE
ICE	and	ROUGH	and	SLOPED	MUNSAFE
ICE	and	SMOOTH	and	SLOPED	MSAFE
ICE	and	ROUGH	and	FLAT	MSAFE
ICE	and	SMOOTH	and	FLAT	HSAFE

Table 2.3: Optical camera safety fuzzy rule set

Obstacle	op.	**Roughness**	**Terrain**
PRESENT	and		HUNSAFE
ABSENT	and	VROUGH	HUNSAFE
ABSENT	and	ROUGH	MUNSAFE
ABSENT	and	SMOOTH	MSAFE
ABSENT	and	VSMOOTH	HSAFE

For icy surfaces sensed by the lidar, a surface point is considered HUNSAFE if the area around it is VROUGH or if it has a slope that is STEEP. More benign combinations of roughness and slope will result in better safety scores. For snowy surfaces no roughness feature is considered, as it is assumed that snow covers over any terrain roughness. The

safety rating of snow-covered areas is downgraded compared to an equivalent ice covered area. This is due to the potential instability of loose snow on slopes, and the risk of sinking or otherwise unstable landing in flatter areas.

The texture information considered in the lidar rule table does not only consider information from that sensor, but also from the thermal camera. This is a more compact solution to having a separate thermal camera rules table comprising only of texture information. The two texture measurements are normalized according to the confidence parameters of each sensor:

$$f_t^{Fused}(x,y) = \frac{\beta_L f_t^L(x,y) + \beta_T f_t^T(x,y)}{\beta_L + \beta_T} \tag{2.54}$$

where $f_t^{Fused}(x,y)$ is the fused texture information, β_k is a confidence parameter for each sensor, and $f_t^k(x,y)$ is the texture measurement by each sensor (k can be L or T). A second processing step is performed on the texture map, applying a moving averaging window with a size equal to the lander footprint. The final combined texture map represents then the percentage of the lander footprint that is covered in snow for each point in the map.

The sensor confidence parameters $\beta_k \in [0,1]$ can themselves be fuzzy variables (as e.g. in [72]), or they can be set manually to favor measurements from one or the other sensor, provided that in the above equation $\beta_L + \beta_T = 1$. The second, simpler approach is used here.

For the camera, areas covered by obstacles are automatically considered HUNSAFE. Absent an obstacle, terrain safety is then dependent on the measured illumination variance as a proxy for terrain roughness.

The membership functions for the lidar and camera ruleset are given in Figures 2.13 and 2.14 respectively.

For the lidar, the slope was labeled as FLAT for values up to ∼10°, SLOPED from ∼8° to ∼18° and STEEP from ∼15° on. This gives preference to smaller slopes while respecting the ∼15° slope requirement given in Section 1.5. A similar approach is followed for the roughness parameter, where small roughness values as SMOOTH, values at around half the 0.5 m roughness requirement given in Section 1.5 are marked as ROUGH, and values around that requirement are marked as VROUGH. As texture in the lidar rules table is expressed as a snow cover percentage (Equation 2.54), the texture MFs have somewhat smooth slopes, with the texture characterization gently transitioning from ice to snow. The lidar safety score MF comprises of four trapezoidal functions reasonably distributed between zero and ten, with HUNSAFE centered around 0, MUNSAFE around 3, MSAFE around 6, and HSAFE around 9.

For the optical camera an essentially binary MF the obstacle MF is used for obstacles, due to the binary nature of the obstacle detection method. The illumination variance MFs were roughly calibrated to coincide with the roughness requirements in the same way as discussed for the lidar above. The camera safety output score MF is identical to that of the lidar.

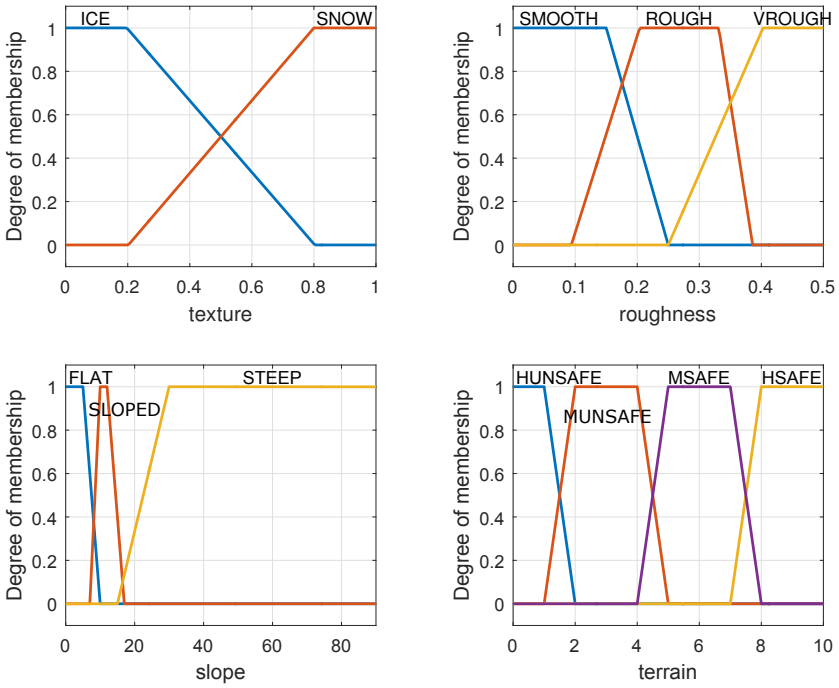

Figure 2.13: Membership functions for the lidar safety ruleset.

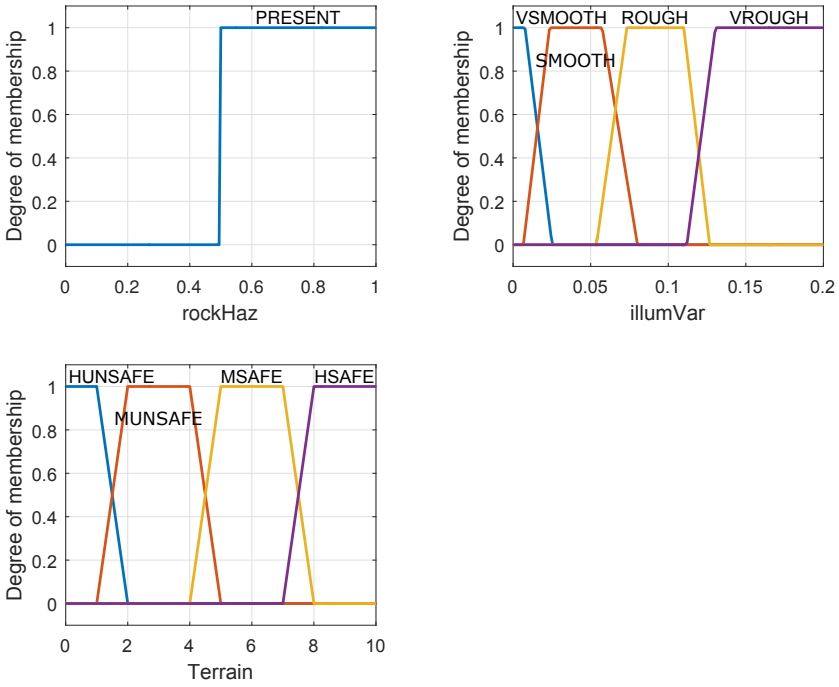

Figure 2.14: Membership functions for the optical camera safety ruleset.

Fuzzy rules and membership functions are also used to determine the sensor confidence values. Several relevant factors can be considered separately for each sensor, like e.g. illumination and angular velocity during a measurement for the optical camera, and the incidence angle and intensity of the return signal for lidar. Confidence factors can also be dependent on location in the map. For simplicity however, here only range to the center of sensed surface will be considered as a common confidence parameter for all sensors and map points.

Centroid defuzzification is finally used to produce a numerical safety score s_k each sensor. The safety score is a weighted combination of the degrees of membership to the fuzzy terrain safety classes

$$s_k(x,y) = \frac{\sum_i p_i A_i}{\sum_i A_i} \tag{2.55}$$

where $s_k(x,y)$ is the defuzzified terrain safety score for the sensor k (can be either L or C) at point (x,y) on the terrain, p_i is the peak value associated with the membership functions, and A_i is the area under the truncated membership function, as calculated during the implication step of the fuzzy assessment (see [77]). The resulting safety score is in the range [0,10].

The safety scores are then weighted according to the set sensor confidence scores ($\{\beta_L, \beta_C\} \in$ [0,1]) and are combined into a single fused safety score according to the equation

$$s_f(x,y) = \frac{\beta_L s_L(x,y) + \beta_C s_C(x,y)}{\beta_L + \beta_C} \tag{2.56}$$

It is reminded that the thermal camera information has been incorporated with its corresponding confidence in the lidar texture measurement. It should be pointed out that if equal confidence is assigned to each sensor, the final safety score will then be the sum of the optical camera and lidar scores divided by two.

2.3.3.2 Landing Reasoning

The second key factor for landing success is closeness to a plume source. According to Equation (2.53), the landing site must be between a minimum and a maximum distance from the plume source, set at 5 and 50 m respectively. To account for this in the fuzzy landing assessment process, a distance weighing factor is introduced, given as the difference of two sigmoidal functions[12]

$$\beta_d(x,y) = S_1(r; a_1,c_1) - S_2(r; a_2,c_2) \tag{2.57}$$

$$where \qquad S(r; a,c) = \frac{1}{1 + e^{-a(r-c)}} \tag{2.58}$$

$$and \qquad r = \sqrt{x^2 + y^2} \tag{2.59}$$

[12]See Matlab's *dsigmf* documentation page, `https://de.mathworks.com/help/fuzzy/dsigmf.html`

where, x is the distance of a map cell from the plume source and a and c are shaping parameters. After appropriately tuning the shaping parameters ($a_1 = 2.36$, $c_1 = 6.01$, $a_2 = 0.3958$, and $c_2 = 41.2$), β_d acquires the shape shown in Figure 2.15.

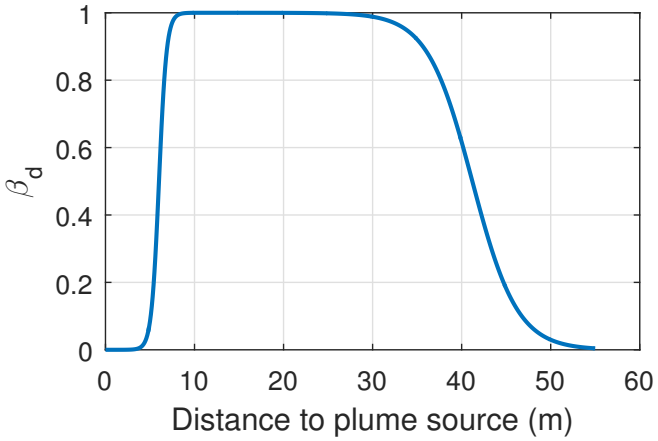

Figure 2.15: The distance to plume source weighting factor

The overall landing score of a given map cell is then given by

$$l_{map}(x,y) = \beta_d(x,y)s_f(x,y) \tag{2.60}$$

As seen, the distance factor maintains the score of map cells at distances between 5 and 30 m, while sharply reducing the score of points between 30 and 50 m and more so in the range of 0 to 5 m.

2.3.3.3 Fuzzy Spatial and Temporal Reasoning

The fuzzy reasoning process described up to this point is performed independently for each map cell (x,y). Realistically, however, the quality of a landing site is connected to the quality of its neighboring sites and does not change over time[75]. These observations can be used to further refine the landing assessment process with a final set of fuzzy reasoning processing as seen in Figure 2.16.

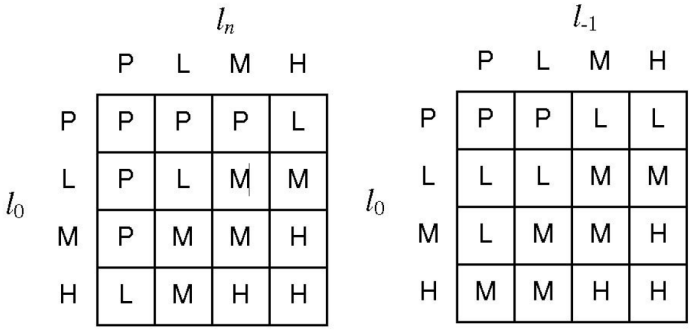

Figure 2.16: Fuzzy rulesets for spatial (left) and temporal (right) landing assessment [75]. P, L, M, and H stand for Poor, Low, Medium, and High landing goodness respectively.

In Figure 2.16, l_0 is the landing score at a given location and current point in time, l_{-1} is the landing score at a given location and previous point in time, and l_n, is the mean score of the map points within a set range around l_0 (e.g. at a range of 4 lander footprints). The spatial and temporal landing scores (l_s and l_t respectively) are calculated using a fuzzy assessment process, and are finally combined using a weighted average to arrive at a final landing score, l:

$$l(x,y) = a_s l_s + a_t l_t \tag{2.61}$$

where $a_s + a_t = 1$. The weights can be set equally or can be biased towards either the spatial score or the temporal score. The spatial and temporal rules-sets provide a further layer of uncertainty mitigation.

2.3.3.4 Re-targeting

The final landing score l takes into consideration the two key landing factors (terrain safety and scientific interest), as well as spatial and temporal information that further increases the robustness of the assessment. The selection of the optimal site for landing in a map then becomes about finding the point on the terrain with the highest score l. To avoid unnecessary re-targeting that would increase operational risk, a new landing site is selected only if the score of the currently selected site drops below a preset threshold.

2.3.4 System and operational considerations

The HDA function must be able to sense the hazards in order to satisfy landing safety requirements within the entire lander footprint. For meaningful HDA to take place in that sense, the ground sample distance of the HDA sensors must be small enough to resolve the area of the lander footprint. The lander would ideally be also able to sense as much of the candidate landing area for hazards as possible.

The critical parameters for HDA sensing are then the ground sample distance (GSD) and the ground coverage area (GCA)[13]. Both of these factors depend on the range from which an HDA measurement was taken. Further, the GSD depends on sensor angular resolution and the GCA depends on sensor FOV.

A reference HDA sensor suite can be defined here based on the HD lidar from the Lunar MARE mission concept/ALHAT project [78]. Its characteristics are given in Table 2.4. The dependence of GSD and GCA from range to the target surface given the sensor characteristics in Table 2.4, and assuming a nadir-pointing sensor can be calculated through basic trigonometry and is given in Figure 2.17.

[13]The ground coverage area is defined as the area on the surface that is visible by a sensor with a given FOV. It is expressed in terms of length of the side of the projection of the FOV on the surface

Table 2.4: Some characteristics of the chosen lidar (based on HD lidar from [78])

Characteristic	Value
Number of beams	128×128
FOV	6.6°
Angular resolution	0.05°

Considering a lander footprint with a diameter of ∼4 m and according to the comments above, the earliest meaningful HDA measurement will be for a GSD of ∼1 m. As seen in Figure 2.17 this will happen at a range from the target surface of ∼1100 m. From the same figure it can be seen that the GCA for this measurement is then ∼125 m, which nicely coincides with the potential landing area, i.e. the width of the bottom of the tiger stripe canyon containing the target plume source, with some margin.

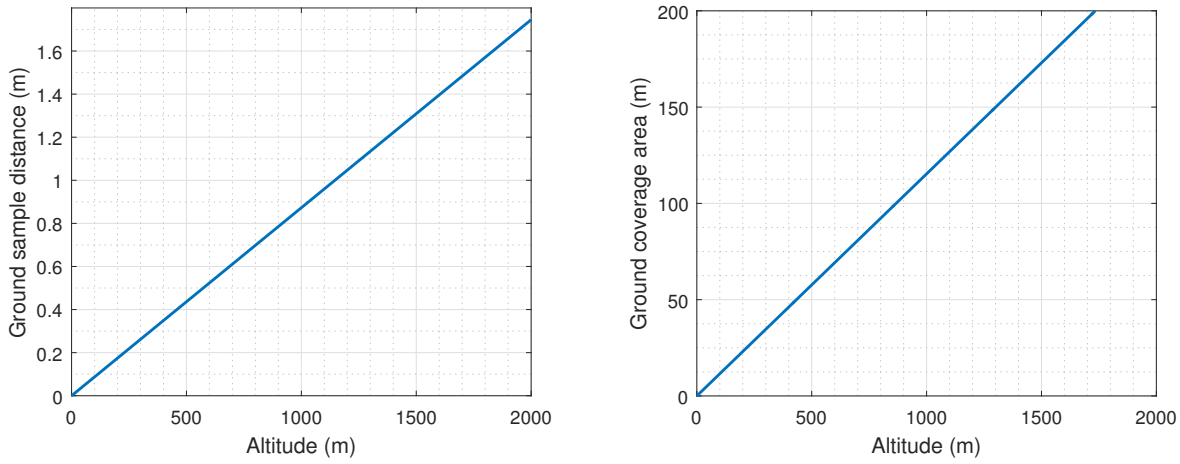

Figure 2.17: Ground sample distance (GSD, left) and ground coverage area (GCA, right) as a function of altitude from the target surface, assuming a nadir pointing sensor

Consecutive HDA measurements can be taken at set intervals as the lander approaches the landing target. These measurements will have increasingly smaller GSDs and GCAs. Constraints will be set on the timing of these measurements by the execution time needed for the HDA process to run on the on-board computer. For example, [78] gives an execution time for the NASA ALHAT HDA process of 5 s. Although not identical with the HDA process described above, this value will be used here as a first approximation.

There are also constraints setting a minimum reasonable range from target for HDA measurements. As the lander nears the surface, some fine snow might be kicked up by the lander's thrusters. This material can significantly interfere with lidar measurements, making them unreliable. Equally critically, and as seen in Figure 2.17, the GCA drops to sizes below that of the lander footprint. As the purpose of an HDA measurement is to command a re-targeting, it would be meaningless if the covered area contained no room for HDA to command a re-targeting to. The minimum meaningful GCA is then set to 10 m (the equivalent to ∼four lander footprints), which corresponds to an altitude of ∼100 m. This is set as the HDA cut-off altitude.

Taking the above into account, and assuming that an unnecessarily high number of HDA measurements is likely to increase operational complexity, a first value for the number of measurements is 4, taken at ranges of $[1100, 550, 250, 100]$ m corresponding to GCDs of $[1, 0.5, 0.2, 0.1]$ m and GCAs of $[125, 55, 20, 10]$ m (see Figure 2.17).

Concerning sensor pointing during landing, an agile approach is required, as each HDA measurement can possibly select a new target landing site. To provide this agility, a solution where the HDA sensor suite is mounted on a gimbaled platform is selected. Consecutive HDA measurements, with a smaller ground sample distance and a smaller ground coverage area will then be performed around this new landing target.

2.4 Guidance

The function of the guidance function is to solve the minimum-fuel planetary landing problem consisting of estimating the thrust profile to take the lander from its initial state, to a target location while minimizing the fuel consumed for these maneuvers [79].

2.4.1 Approaches to the planetary landing Guidance problem

An exhaustive survey of guidance laws for safe and accurate planetary landing is performed in [80]. Various guidance algorithms were compared for their optimality, computational efficiency, their ability to fulfill the requirements for landing with safety and precision, and their ability to incorporate the related landing constraints. Two approaches for landing come on top: Convex programming and Neural Guidance.

Convex guidance is based on a mathematical result known as lossless convexification of the associated optimal control problem, which allows for the formulation of the problem as a convex optimization problem and guarantees obtaining the global optimal solution when a feasible solution exists [81].

The main argument in favor of convex guidance is that, as the guidance problem must be solved in real time during landing, it is critical that any algorithms are guaranteed to converge to the global optimum. By using deterministic convergence criteria, convex guidance can achieve this. In addition, putting the problem in a convex formulation makes it very computationally efficient to solve. Landing constraints can be directly considered. The global optimality for a given solution is proven mathematically, so that validation of this optimality becomes straightforward. On the other hand, although relatively computationally efficient, this method still is more complex than simpler polynomial guidance approaches [80, 82].

A novel approach for planetary landing guidance is Neural guidance. This guidance method has a generally biomimetic origin, the underlying idea being to model neurons and connect them into a network. Such systems "learn" to perform tasks by considering examples. In the planetary soft landing case, networks must be trained by using models and sets of constraints and boundaries for hundreds or thousands of simulated landing trajectories [80]. Several approaches to neural guidance have been proposed [83, 84, 85].

Neural guidance has several advantages over more conventional guidance algorithms including improved performance (lower fuel consumption) and some ability for trajectory

modification in real time. It has been shown to be robust, accurate, and optimal for Mars landing cases [85]. A major drawback of neural guidance however is that its performance is limited by the input training cases. Although small deviations from the nominal mission parameters can be handled provided enough training cases, significant or unexpected changes in the mission parameters are not possible in real-time. In other words, the algorithm has no formal knowledge of the problem being solved, merely turning input values to working solutions, acting as a black box. This makes the algorithm very difficult to validate, and practically impossible to prove that the guidance solution is convergent [80].

Convex guidance optimization is chosen for further implementation in this work, mostly due to it being demonstrably optimal, its verifiability, and its operational robustness for real time application.

2.4.2 Convex planetary landing guidance

Convex optimization is a subfield of optimization that studies the problem of minimizing convex functions over convex sets[14]. Convex guidance was introduced in [82]. The particular implementation that will be investigated here is called G-FOLD (Guidance for Fuel Optimal Large Divert) and is presented in [81, 86]. The following is based on the formulation and notation used in [81].

The landing guidance function must calculate the thrust vector profile $\mathbf{T}_c$ and trajectory ($\mathbf{r}$, $\dot{\mathbf{r}}$) to take the lander from an initial position $\mathbf{r}_0$ and velocity $\dot{\mathbf{r}}_\mathbf{0}$ to a state of rest at the target location on a planetary surface while minimizing fuel use and respecting specific state and control constraints. The general problem is illustrated in Figure 2.18, and formulated in Equations 2.62 to 2.71.

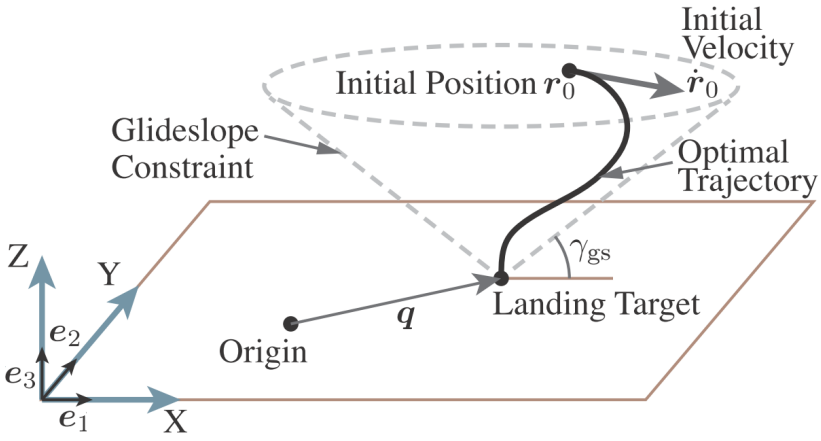

Figure 2.18: Illustration of the landing guidance problem (see text for description). Slightly adapted from [81].

[14]A real-valued function defined on an n-dimensional interval is called convex (or convex downward or concave upward) if the line segment between any two points on the graph of the function lies above or on the graph. A convex set is a subset of an affine space that is closed under convex combinations. More intuitively, in a Euclidean space, a convex region is a region where, for every pair of points within the region, every point on the straight line segment that joins the pair of points is also within the region [Wikipedia]

$$\max_{t_f, \mathbf{T}_c} m(t_f) \qquad \text{subject to} \tag{2.62}$$

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\Big(\mathbf{g} + \frac{\mathbf{T}_c(t)}{m(t)}\Big), \qquad \forall t \in [0,t_f] \tag{2.63}$$

$$\dot{m}(t) = -\alpha\|\mathbf{T}_c(t)\|, \qquad \forall t \in [0,t_f] \tag{2.64}$$

$$tan\gamma_{gs}\mathbf{e}_3^\top \mathbf{r}(t) \geq \sqrt{(\mathbf{e}_1^\top \mathbf{r}(t))^2 (\mathbf{e}_2^\top \mathbf{r}(t))^2}, \qquad \forall t \in [0,t_f] \tag{2.65}$$

$$||\dot{\mathbf{x}}(t)|| \leq tan(\psi)||\mathbf{x}(t)||, \qquad \forall t \in [0,t_f] \tag{2.66}$$

$$0 < \rho_1 < \|\mathbf{T}_c(t)\| < \rho_2 \tag{2.67}$$

$$\hat{\mathbf{n}}^\top \mathbf{T}_c(t) \geq \|\mathbf{T}_c(t)\| cos\theta \tag{2.68}$$

$$m(0) = m_{\text{wet}} \tag{2.69}$$

$$\mathbf{r}(0) = \mathbf{r}_0, \qquad \dot{\mathbf{r}}(0) = \dot{\mathbf{r}}_0 \tag{2.70}$$

$$\mathbf{r}(t_f) = \mathbf{0}, \qquad \dot{\mathbf{r}}(t_f) = \mathbf{0} \tag{2.71}$$

The optimization objective is expressed Equation 2.62: the lander must expend the minimum amount of propellant, with $m(t)$ being the time-variable lander wet mass (with m_0 the initial lander wet mass), and t_f the time of landing. The dynamics equations for the landing are given in Equation 2.63, where $\dot{\mathbf{x}}(t)) = (\mathbf{r}(t), \dot{\mathbf{r}}(t))$,

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \qquad \text{and} \qquad \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \tag{2.72}$$

$\mathbf{g}$ is the gravity vector, and $\mathbf{T}_c(t)$ is the time variant thrust vector. Equation 2.64 gives the propellant consumption rate where

$$\alpha = \frac{1}{I_{sp} g_0} \tag{2.73}$$

The optimized landing trajectory must respect state and control constraints. The first state constraint is given in Equation 2.65. This equation describes an inverse cone with a slope angle γ_{gs} with its apex on the landing target, within which the lander is allowed to fly. This is termed the glide slope cone constraint. The unit vectors for each of the axes are noted as $\mathbf{e}_{\{1,2,3\}}$. The second state constraint in Equation 2.66 keeps the velocity magnitude to increasingly lower levels as a function of the lander range-to-target, i.e. the lander is forced to be slower as it nears the target and the surface. This constraint is a line crossing zero in the velocity magnitude vs range space, with a slope of ψ. There are two constraints on the controls: first a thrust magnitude constraint is given in Equation 2.67, stating that the thrust magnitude must be between a minimum (ρ_1) and a maximum value (ρ_2). Second, Equation 2.68 requires that the angle between the thrust vector and the local vertical will not be larger than a given value θ. The control constraints are illustrated in Figure 2.19.

It should be noted here that the important requirement, that the final thrust vector of the lander should be pointing up-wards (representing an up-right attitude), was not implemented for tool-related technical issues. It should be considered in future iterations of this work.

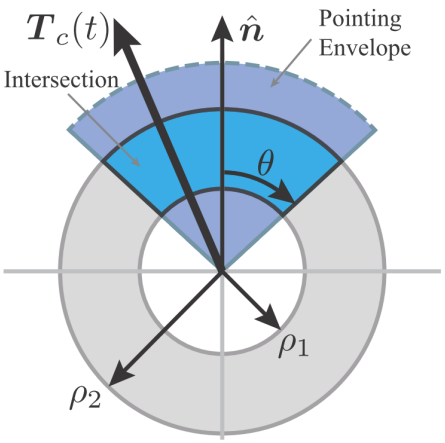

Figure 2.19: 2D representation of the thrust magnitude (ρ_1 - ρ_2) and angle (θ) constraints. The resulting allowed control space is non-convex [81].

Equations 2.69 to 2.71 give the initial and final boundary conditions for the optimization problem.

The element of the above formulation that does not allow for the problem to be formulated as a convex optimization problem are the non-convex thrust constraints, and in particular due to the thrust magnitude contraint: once started, the thrusters cannot be throttled below a certain level [79].

This issue is resolved in the G-FOLD algorithm by applying the "lossless convexification" by using a particular convex relaxation of the control constraints. In the relaxed problem a scalar slack variable $\Gamma(t)$ is introduced to lift the control space to a higher dimension (with one additional dimension) and relax the non-convex set of controls to a convex set. It can be shown that after this introduction the optimal solution to the relaxed problem is always a feasible and optimal solution to the original non-convex problem [82]. The mathematical foundation behind this finding is complex and extensive, and it falls out of the scope of this work to describe it in detail. The interested reader is directed to the main reference for G-FOLD [82] and the further work cited therein.

The new "convexified" problem is formulated as

$$\max_{t_f,\mathbf{T}_c,\Gamma} m(t_f) \qquad \text{subject to 2.65, 2.69, 2.70, 2.71, and} \tag{2.74}$$

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\Big(\mathbf{g} + \frac{\mathbf{T}_c(t)}{m}\Big), \qquad \forall t \in [0,t_f] \tag{2.75}$$

$$\dot{m}(t) = -\alpha\Gamma(t), \qquad \forall t \in [0,t_f] \tag{2.76}$$

$$\|\mathbf{T}_c(t)\| \leq \Gamma(t) \tag{2.77}$$

$$0 < \rho_1 \leq \Gamma(t) \leq \rho_2 \tag{2.78}$$

$$\hat{\mathbf{n}}^\top \mathbf{T}_c(t) \geq cos\theta\Gamma(t) \tag{2.79}$$

2.4.3 System and operational considerations

The guidance aspect of the landing operations concept can now be discussed in more detail. Guidance operations start at approach phase start. The lander is following a nominal pre-calculated trajectory, targeting a pre-selected nominal landing site near but not on the target plume source. However, due to navigational uncertainty, the nominal lander state at approach gate will be different than the real one. As the operations advance, the HDA function will likely command a re-targeting. In both cases either the starting or the ending point of the current guidance trajectory is no longer up-to-date. A new guidance trajectory will then have to be calculated.

In the case where the starting point of the current guidance trajectory is no longer considered valid, it would make sense for this new guidance calculation to happen when the navigational uncertainty has been reduced to within acceptable limits. This is most likely to happen after optical camera or lidar navigation measurements (Sections 2.2.3.2 and 2.2.3.3). An HDA assessment also takes place after each lidar measurement, it is thus likely that re-targeting will take place once the processing time for the HDA function has passed (5 s, see Section 2.3.4). It would make therefore sense for the guidance function to be executed after each lidar measurement and HDA assessment if a re-targeting is commanded by HDA.

The glide slope constraint can be used to enforce no-go areas around the target tiger stripe canyon. A possible approach would be to define two phases, one inside and one outside the canyon. While outside the canyon, the glide slope would be close to zero. Inside the canyon the glide slope constraint would match the slope of the canyon walls, with an additional buffer margin. By defining two separate no-go zones it is assured that each one is convex, and convex guidance can be applied. This is the approach followed e.g. by [87].

In this case however a single glide slope constraint would be preferable, as the lander will have to maintain visibility to the landing site, the plume source, and the entirety of the canyon bottom around them at all times. A single cone glide slope constraint will ensure that at no time will the landing site be obscured by local topography. As the canyon wall slope is 30 deg. (Section 1.4), a single glide slope constraint of $\gamma_{gs} = 40^\circ$ would allow for a safety buffer. Extending this cone to an altitude of $\sim$ 1500 m above the plume source, where the start point of the approach landing phase is assumed to be, the cone base has a diameter of $\sim$ 2000 m. This approach phase starting point should then be within that general area to safely reach the landing site while respecting the glide slope constraint. As will be seen later (Table 2.7), this level of lander position accuracy at the start of approach operations is easily achievable, thus further reinforcing the argument for using a single glide slope. The above points are illustrated in Figure 2.20.

A thrust angle constraint, as discussed above, can ensure that the lander attitude will not be allowed to deviate excessively from the local vertical during the final few minutes of the landing in the approach phase. A compromise between safety and agility can be achieved for $\theta = 45^\circ$. The velocity magnitude constraint is also implemented, with a velovity-vs-range slope empirically chosen as $\psi \sim 65^\circ$.

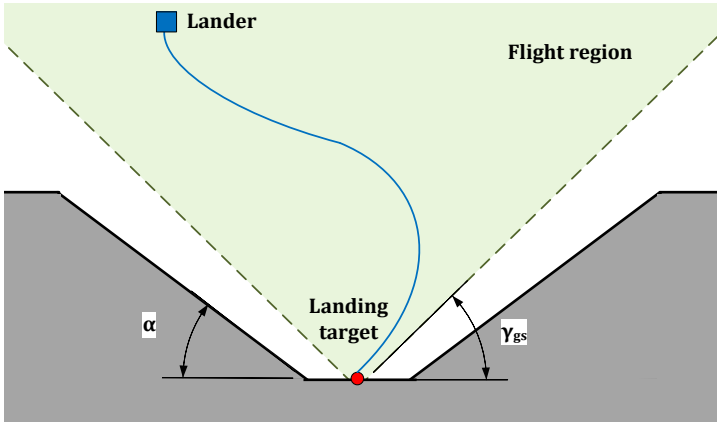

Figure 2.20: Glide slope cone flight region for landing on a tiger stripe canyon bottom (not to scale). The glide slope constraint cone with an angle $\gamma_{gs} \sim 40°$ would ensure adequate distance between the lander and the canyon faces (slope $\alpha \sim 30°$) for any landing target on the canyon bottom. The lander maintains visibility of the entire canyon bottom at all times. Adapted from [87]

2.5 Control and Actuation

The Control function in a GN&C system derives the control commands necessary to match the current vehicle state (position, velocity) with the desired path provided by the guidance function. These commands are in turn forwarded to the appropriate actuators that apply the commanded forces for trajectory control.

Control methods for planetary landing is a big field of research (see e.g. [88] for an introduction to existing control approaches). In this work however focus is given to the other functions of GN&C and ideal control will be assumed. The rest of this section will be focus on landing actuators, i.e. main thrusters for landing.

An initial detailed propulsion system design for the EnEx lander was performed in [89] and is summarized in [38]. The propulsion system given there is a monopropellant system working on blowdown mode. A Main Engine Assembly (MEA) of thrusters is mounted at the bottom of the Lander so that the thrust vector runs through the lander's center of gravity.

A critical parameter for the design of propulsion systems for planetary landing is thrust-to-weight (TtW), meaning how much larger the maximum thrust achievable by the thrusters is, compared to the weight of the lander on the target planet, i.e. the mass of the lander times the local gravity. Intuitively this parameter can be seen as a metric of lander maneuverability, as higher control accelerations can be implemented on the lander to quickly change its trajectory. Previous landing missions and existing mission concepts have had a thrust-to-weight ranging from a little above 1 [90], to ~ 2 [79], to ~ 5 [78]. As the EnEx lander has very challenging landing requirements, it is assumed that a high level of maneuvrability will be necessary and a TtW ratio of 5 is chosen. The resulting maximum thrust (rounded up) would then be:

$$T_{max} = TtW * m_{\text{wet}} * g_{\text{Encel}} \simeq 1118N \tag{2.80}$$

where, T_{max} is the maximum thrust magnitude, TtW is the thust-to-weight ratio, m_{wet} is the lander wet mass at approach phase start and $g_{Encel} = 0.113m/s^2$ is the gravitational acceleration magnitude on Enceladus. An interesting observation is that, despite the large wet mass of the lander, the low gravity on Enceladus means that a relatively low-thrust engine is needed, compared e.g. to the ESA Lunar Lander concept [91]. The I_{sp} for this propulsion system is given as 220s.

Propulsion systems can be typically throttled between a minimum and a maximum value. For this work these thrust limits are set between $\rho_1 = 30\%$ and $\rho_2 = 80\%$ of T_{max} [82, 86, 87]. Alternatively, pulse modulation, i.e. rapidly pulsing engines at varying frequencies, can be implemented to the same effect [91].

It is further assumed that a guidance-generated thrust vector is ideally applied on a point-mass lander, without considering noises in the magnitude and direction of thrust applied by actuators.

2.6 Baseline GN&C system and operations

By combining information and analysis performed in the above previous sections in this chapter, a detailed baseline system and operations concept can be defined for the approach phase of the landing.

The system is comprised of the sensors and actuators with the characteristics given in Tables 2.5 and 2.6 respectively.

Table 2.5: Sensor parameters and noises (see text for description). Lidar is always considered to be in range during the approach phase. See text for discussion on the thermal camera pixel error.

Ext. Sensor	Resolution	FOV	Angular resolution	Pixel error	Pointing error	Range noise
Optical camera	1024×1024	55°	0.05°	1	0.05°	-
Thermal camera	1024×1024	55°	0.05°	$[2, 10]$*	0.05°	-
Lidar	128×128	6.6°	0.05°	-	0.05°	0.3% (multiplicative)

Int. Sensor	Sampling frequency	White noise
INS accelerometer	100 Hz	$10^{-5}m/s^2$ 1σ

For the optical and thermal camera, typical parameters used in typical space cameras are assigned. A pointing error is assigned, attributed to mounting errors and similar effects. The camera pixel error corresponds to the angular resolution of a pixel, and represents the minimum angular accuracy achievable by a camera due to the fact that camera images are composed of a finite amount of pixels. Two pixel error values are assigned for the thermal camera. The first one is used for generic observations with the thermal camera and accounts for the lower image sharpness of thermal images. The

second even higher pixel error value is used when observing a plume source hot spot in the thermal image. Due to its nature (somewhat spread out local maximum in the thermal image) it will be difficult to pinpoint. Centroiding methods similar to the ones used in star trackers for sub-pixel pinpointing can be applied [93], but this is left to future iterations of this work.

For the INS accelerometer, typical measurement frequency value is used and a white noise value is chosen to produce a reasonable drift in lander position as seen in other similar works (see e.g. [92]).

All three exteroceptive sensors are assumed to be perfectly co-aligned and placed on a gimbaled platform able to point freely in any direction. All sensors have the same angular resolution so that their measurements can be directly co-registered for use in the HDA function.

The characteristics of the propulsion subsystem and the guidance constraints are summarized in Table 2.6.

Table 2.6: Propulsion system and guidance constraint characteristics.

Characteristic	**Value**
T_{max}	$1118N$
Thrust range	30% to 80% of T_{max}
ρ_1	356 N
ρ_2	949 N
I_{sp}	220 s
Fuel (approach phase)	100 kg
Max. thrust angle	45°
Glide slope angle	40°
Velocity magnitude slope	65°

The initial and final conditions for the landing approach phase start are given in Table 2.7 along with other necessary starting parameters. The initial navigation uncertainties will be inherited by the final state in the main braking phase (Section 1.7). To get an estimate for realistic navigation uncertainty values at the start of approach operations for the EnEx lander, previous lunar landing studies [78, 92, 94, 95] with similar landing requirements and navigation function operations were referred to. The nominal position is selected somewhat arbitrarily, so that the lander starts at a range of ∼2 km from the landing target, just as the lidar comes within range of the surface, which marks the start of the approach phase. The initial velocity is approximated based on the fact that to null all velocity relative to the surface of Enceladus over the plume sources, an impulsive landing maneuver of 180m/s is necessary (Section 1.3). Based on that number, it can be assumed that most of that ΔV is performed during the main braking phase, with $\sim$ 30 m/s leftover at approach phase start. Similar nominal position and velocity values at approach phase start are also given e.g. in [96]. The final conditions are dictated by the landing requirements (Section 1.5).

Table 2.7: Initial and final (target) navigation state and other conditions for the EnEx lander

Initial parameters (DCA frame[15])	**Nominal value**	**Uncertainty (3σ)**	**Comment**
Init. lander position, m	$[1000, 0, 1500]^\top$	$[100, 100, 100]^\top$	Worst case combination of values from [92] and [94]
Init. lander velocity, m/s	$[-20, 0, -20]^\top$	$[1, 1, 2]^\top$	see above
Init. plume source position, m	$[0, 0, 0]^\top$	$[10, 10, 10]^\top$	
Final lander position, m	$[0, 0, 20]^\top$	$< [1, 1, 1]^\top$	Uncert. relative to the plume source and local hazards
Final lander velocity, m/s	$[0, 0, 0]^\top$	$[0.25, 0.25, 0.25]^\top$	Uncert. of [94] divided by four (see text)
Lander dry mass, kg	2000	-	[38]
Lander wet mass, kg	2100	-	[38]
Enceladus grav. constant, m/s^2	0.113	-	

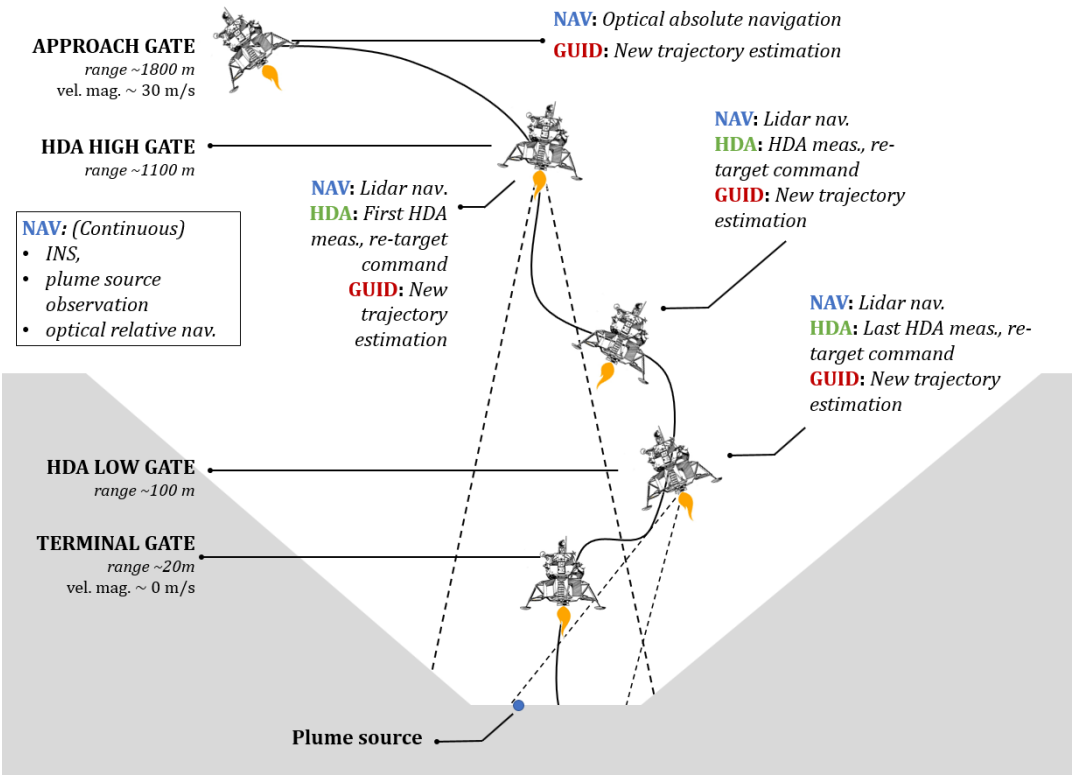

Figure 2.21: Baseline concept for landing operations for the EnEx lander (adapted from [96])

Based on the operational considerations for each GN&C function as described in the previous sections, and the operational sequences of similar missions [96, 92], a combined concept for the approach phase operations can be given, integrating all the functions described in this chapter. The operations concept is further illustrated in Figure 2.21.

1. **Approach Gate to High Gate**

   a) At approach gate (AG), the landing site comes in view of the sensors and is within range of the lidar ($\sim$ 2 km range from the landing site). An optical camera navigation measurement of a-priori known points is used for a significant reduction in the lander's navigation state uncertainty. A-priori known points are observed at set times down to a minimum altitude, after which no a-priori known features are recognizable to the lander cameras. Using the improved navigation knowledge, the gimbaled sensor set can be pointed to the nominal landing site.

   b) Based on this navigational uncertainty reduction, a new corrective guidance trajectory is calculated, from the newly estimated lander position to the nominal landing site. A large scale divert maneuver might be necessary, depending on the lander position accuracy before AG.

2. **HDA High Gate to HDA Low Gate**

   a) The first HDA measurement is taken at HDA High Gate ($\sim$ 1 km range from landing site). An HDA assessment is performed and after it is completed 5 seconds later, a new landing site is possibly selected. A few more (4) HDA measurements are taken at set times so that increasingly higher resolution HDA maps are produced. Each time a new landing site is selected, the sensor set turns on its gimbal to point to the new landing site.

   b) From the lidar measurement, the navigation function can define a new hazard frame. The lander navigation state is then estimated in this new hazard frame. If necessary, a thermal camera observes the plume source and localizes it on the new hazard frame, based on the EKF-SLAM formulation. As the thermal camera has a wide FOV, the plume source will be observable for almost the entire landing down to TG. If a thermal camera is not used, the position and uncertainty of the plume source on the hazard frame will be estimated based on its position and uncertainty on the world frame, plus the last lander position uncertainty value on the world frame, before the switch to the hazard frame. No more absolute camera navigation measurements are going to be taken after the frame switch to the hazard frame.

   c) Each time a re-targeting is commanded by HDA, guidance calculates a new trajectory to the new landing site. As the overall landing area is less than 100 m across, all diverts are expected to be small and any landing site within this area is assumed to be reachable by the lander.

3. **HDA Low Gate to Terminal Gate**

   a) At HDA Low Gate (LG), the last HDA measurement is taken, the final re-targeting maneuver can take place and the final guidance trajectory can be

calculated.

b) The lander follows the guidance trajectory to Terminal Gate (TG) where the approach phase terminates with the lander having reached a position of 20 m above a safe target landing site having a navigation state accuracy within the requirements.

4. **Terminal Descent**

a) The lander can then perform the Terminal Descent, either by free falling to the surface, or being gently lowered by a sky crane, depending on the design of the mission (Section 1.5).

Chapter 3

The Landing Simulation Tool

A simulation tool was put together implementing the functions described in the previous chapter, to achieve the goals, and respecting the assumptions and simplifications, described in Section 1.8. The tool was written in Matlab based on the elements discussed in Section 3.8, and is provisionally called the LandingSim tool.

3.1 Top Level Architecture

The tool comprises several simulation blocks, roughly corresponding to the top-level GN&C functions in Figure 1.11. The top level tool architecture is shown in Figure 3.1.

In a general description, the Sensor block senses elements of the lander and landing environment. Information from this sensing is passed on to the Navigation block that can estimate the lander navigation state and, if needed, the location of a plume source. Sensing information is also forwarded to the Hazard Detection and Avoidance (HDA) block that assesses the safety and proximity to a plume source of the sensed terrain. If an overall better landing site than the current is found in the HDA block, a re-target to this new site is commanded. The Guidance block will then estimate a viable landing trajectory, starting from the Navigation-estimated current lander position to the new HDA-commanded landing site. The thrust profile generated by Guidance will be applied on a point mass lander dynamics model along with the Enceladus gravitational force, and the updated position of the lander for the new simulation step will be given. The combined accelerations applied on the lander will be then forwarded to the navigation block thus closing the loop.

Figure 3.2 shows the representation of all the relevant elements in the tool. In more detail, the tool implements a 3 degrees-of-freedom (3-DOF) model, with a thrust acceleration vector applied on the point mass lander. Another unit vector represents the pointing direction for all sensors that are assumed to be on a gimbaled platform that can point freely. Each sensor has its own FOV and pixel count. A Digital Terrain Model (DTM) of the landing terrain can be input. The plume source is placed on the DTM, and a higher resolution area is created around it, further populated by obstacles. Sets of both a-priori known and a-priori unknown landmarks are interspersed randomly on the terrain. Terrain additional illumination, snow, and temperature layers can be generated.

The navigation function also tracks the estimated position of the lander.

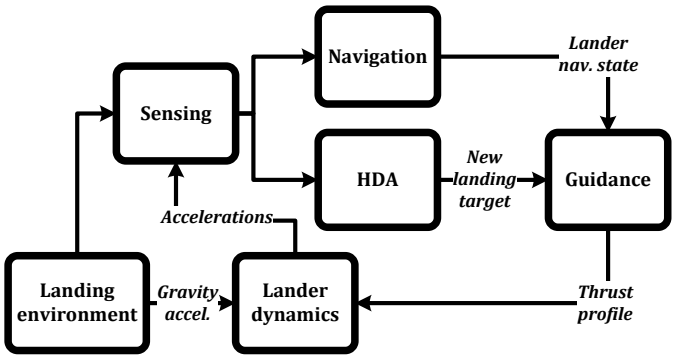

Figure 3.1: Top-level architecture of the LandingSim tool, comprising each of the major simulation blocks. The entire terrain is $\sim$ 500 m across, and the detailed area in around the plume source is $\sim$ 125 m across

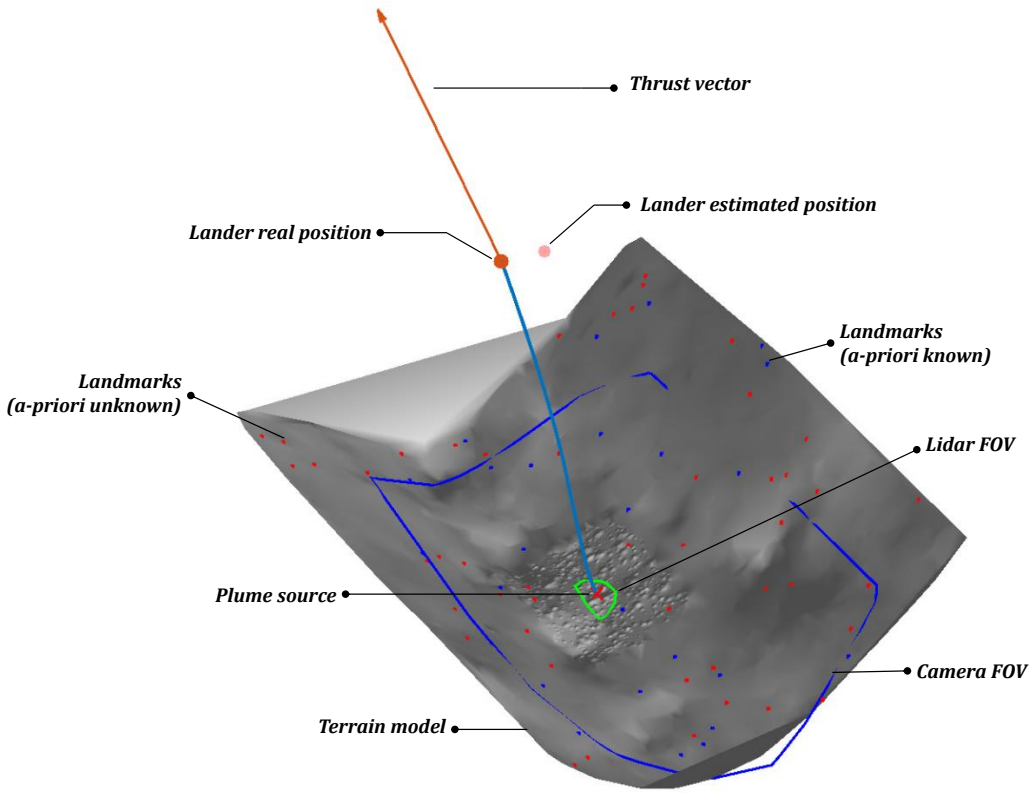

Figure 3.2: Illustration of all the relevant entities in the LandingSim tool

In the following each of the major simulation blocks will be described in detail.

3.2 Landing environment block

An abstracted version of the landing environment is simulated in the Landing environment block, based on the information given in Sec. 1.4. A central role in this block is played by the Digital Terrain Model (DTM) of the landing terrain. DTMs are a common format for 3D models of terrains. In the LandingSim tool, the input DTM is simply a list of terrain points expressed in the world frame (Appendix A.1). To generate a surface based on these points, Delaunay triangulation is applied [100]. A set of triangular "faces" is then generated, connecting the terrain model points, or "vertices" in threes. If the input terrain resolution is deemed too low, the tool has the capability to artificially increase this density by interpolating additional points on the current surface faces and noising their altitude values [101]. The magnitude of that noise is defined by a roughness parameter, representing natural terrain undulation in larger scales and small-scale roughness features in smaller scales.

The terrain input here is an Enceladus DTM[16] as seen in Figure 1.7 [18], with a GSD of ~ 100 m. A ~ 500 m part of the terrain was chosen and its resolution was increased as mentioned above. A point representing the plume source was placed in its center, and an area ~ 125 m across around it was further refined to a GSD of ~ 10 cm.

The next topography feature added is obstacles. A set number of obstacles can be randomly distributed on a terrain model, each obstacle described as a 2D Gaussian "bump":

$$h_i(x,y) = h_{i,max} exp\left(- \frac{x^2 + y^2}{2h_{i,max}^2} \right) \tag{3.1}$$

where $h_i(x,y)$ is the height of each point (x,y) comprising the i_{th} obstacle (starting from the center of the obstacle), $h_{i,max} \in [0.2, 10]$ m is the maximum height of the i_{th} obstacle, the limits being set between ∼half the lander roughness survivability requirement of 50cm and an arbitrary maximum reasonable height for an obstacle. As seen in the above equation, it is assumed that the width and height are equal for a given obstacle.

To simulate the snow-covered SPT surface, parts of the terrain can be designated as covered by the fine plume ejecta. It is assumed that the probability that the i_{th} face of the DTM, having a slope of α_i will be covered by snow follows the distribution

$$p_{snow,i}(\alpha_i; \mu_{snow}, \sigma_{snow}) = 1 - D(\alpha_i; \mu_{snow}, \sigma_{snow}) \tag{3.2}$$

$$\text{with} \qquad D(\alpha; \mu_{snow}, \sigma_{snow}) = \frac{1}{2}\left[1 + erf\left(\frac{\alpha - \mu_{snow}}{\sigma_{snow}\sqrt{2}}\right)\right] \tag{3.3}$$

where $D(x; \mu, \sigma)$ is the normal cumulative distribution function, erf is the Gauss error function, and μ_{snow} and σ_{snow} are values used to shape the snow cover distribution function. In particular, μ_{snow} represents a "cut-off" slope value for snow cover below which faces will be nominally snow covered and above which they will not be. In turn, σ_{snow} represents the 1σ uncertainty in that cut-off value. Here the rough assumption is

[16] Kindly provided by B. Giese of DLR-Cologne

made for $\mu_{snow} = 20°$ and $\sigma_{snow} = 10°$. The resulting distribution is plotted in Figure 3.3. Still, a few randomly sized and distributed no-snow areas can be defined. In those areas there is no snow present, irrespective of the local slope[17].

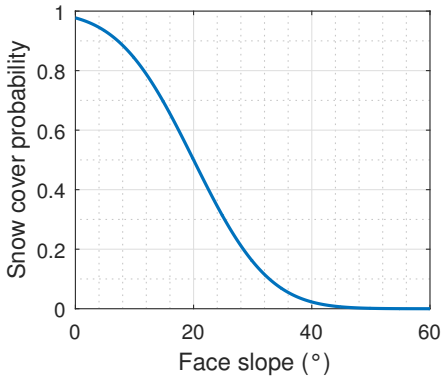

Figure 3.3: Probability of snow cover vs face slope ($\mu_{snow} = 20°$ and $\sigma_{snow} = 10°$)

Interesting points can be placed on the terrain model, first by defining their (x,y) coordinates and then by projecting them on the corresponding terrain face. The following interesting point groups are placed on the model:

- **Plume source**: A single plume source is placed on a selected point along the canyon bottom as discussed above.
- **A-priori known landmarks**: Points whose locations are assumed to be known before the start of the landing, along with associated uncertainties. These landmarks are distinctive enough for them to be identifiable as features in a camera image, as discussed in Sections 2.2.4 and 2.2.5 and then used for camera navigation. A further sub-grouping of these points can be between optical and thermal landmarks. The only type of thermal a-priori known landmark are the plume sources.
- **A-priori unknown landmarks**: Points that represent landmarks distinctive enough to be identifiable in sensor images, but whose position is a-priori unknown. Again, these point group can be sub-divided into optical and thermal landmarks. For the thermal camera, these points simply represent identifiable landmarks; it is assumed that no a-priori unknown plume sources exist[18].

An illumination layer can be added to the terrain model, by calculating the angle between each and the Sun direction, given in elevation and azimuth over the terrain. Faces facing away from the Sun are not illuminated. Shadows cast by the terrain are however not implemented due to the complexity of the corresponding algorithms and their computational cost [102], considering the limited added value to the model.

[17]The equation given here for the likelihood of a terrain model face being covered by snow is based purely on the author's intuition. In future work, a more accurate equation should be found to describe the snow cover of a given terrain.

[18]This is an important assumption for the design of the landing GN&C. The investigation of this case will be left for future work

The surface temperature of the terrain can also be estimated. The plume sources are a significant source of heat and produce local hot spots around them. The ice and snow comprising the local landscape possess distinct thermal properties, and thus can have slightly different temperatures (Section 1.4). In the tool, the ambient ice and plume source temperatures are set to $T_{amb} = 100K$ and $T_{PlS} = 180K$ respectively. The temperature distribution in the material around a plume source follows a Gaussian distribution with a maximum around the plume source (e.g. $180K$) and a standard deviation to define the "spread" of the hot spot of $\sigma_{PLS} = 30m$ [19]. To account for the lower thermal inertia of the very fine, powder-like snow, a factor by which the snow temperature is lowered is simply assigned, e.g. by $T_{mod,snow} = 10$ K). Due to the distance from the Sun and the polar latitudes, the contribution of Solar heat to the surface temperature can be ignored.

3.3 Sensor block

The main function of the Sensor block is to provide simulated interoceptive and exteroceptive measurements to be later used as input by the Navigation and HDA blocks. The structure of the Sensor block is shown in Figure 3.4.

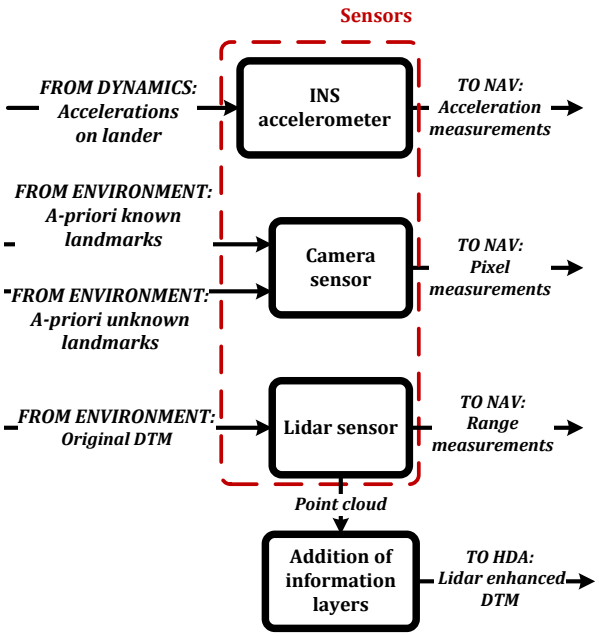

Figure 3.4: Architecture of the Sensor simulation block.

To simulate INS accelerometer measurements, the accelerations acting on the lander due to propulsive thrust and gravitational force, considered ideal, are added together and input into an accelerometer model. The INS accelerometer measurements are in turn noised (Section 2.2.3.1).

[19] This process can be seen as the inverse of that described in Section 2.3.1 for the generation of T_{backgr}

For the purposes of this work a perfect knowledge of the lander and target positions for exteroceptive sensor pointing purposes is assumed.

To simulate camera measurements for navigation purposes, a simplified version of the feature detection method described in Section 2.2.4 is used: the abstracted optical and thermal camera landmarks distributed on the surface of the terrain model are projected on the focal plane of the corresponding camera as described in Section 2.2.3.2. The resulting pixel measurement is the output camera measurement. Instead of a feature descriptor, the pixel measurement is accompanied by a unique numerical landmark identifier. Projected measurements that are outside the field-of-view of a camera are rejected.

To simulate lidar for navigation purposes, a number of beams, usually significantly less than the full beam count available, are back-projected from the lidar sensor, and their intersection points to the impact points are noised for corresponding range and pointing errors. The range to each point is then the lidar navigation measurement.

To simulate a lidar measurement for HDA purposes, the same process is performed for all lidar beams, resulting in a dense aligned grid of points. Due to the computational complexity of the above method and the resulting long execution times, a second, simpler method for lidar measurement simulation was also implemented. In this method, the points of the DTM that are inside the FOV of the lidar sensor are extracted, and are then down-sampled to a resolution according to the expected ground sample distance for the current range from the target (see Figure 2.17). The elevation of this point cloud is noised as needed. As will be seen later, this less accurate method will be adequate for generating input for the HDA function where a large number of points are involved, while the first method will be more useful to the Navigation function where accuracy is necessary and a smaller number of lidar rays are used. The noised lidar point cloud in both cases is given in the lidar sensor as well as the world frame.

As discussed in Section 2.3.1, information from camera sensors is co-registered with lidar clouds, to produce an enhanced lidar cloud to be used as input to HDA. To simulate this enhanced lidar cloud, illumination, thermal, and texture layers are added to the original point cloud produced by a lidar measurement. The illumination layer is created as described above for the landing terrain DTM. The lidar cloud is triangulated, and an illumination intensity value is added to a face according to the angle it forms with the Sun direction. To create the thermal layer, the temperature layer of the original DTM is sampled at the points of the lidar measurement, and the sensed temperatures are assigned to each lidar point. A similar process is used to simulate the texture layer. An enhanced lidar cloud with three additional information layers can thus be forwarded to the HDA block.

3.4 Navigation block

The Navigation block has the purpose to estimate the navigation state of the lander and, if needed, of the plume source. In the core of the Navigation block lies the EKF-SLAM approach described in Section 2.2.2. The architecture of the Navigation block is shown in Figure 3.5.

INS accelerometer readings are used as input to the state propagation function in the prediction step. Pixel measurements from camera sensors, as described above, represent detected camera features. An equally simplified process is used to attempt to match features from the current step with features from the previous step, whereby camera pixel measurements with the same numerical landmark identifier are assigned as matched, and the respective pixel measurement can be forwarded as input to the EKF-SLAM correction step. Unmatched features will go through the landmark initialization process. Each class of navigation-related points listed in Section 3.2 is assigned a designated part of the SLAM map for easier tracking.

Lidar measurements are primarily used to define a new, hazard relative frame, instead of being directly integrated to the EKF-SLAM approach (Section 2.2.3.3). To facilitate the current work, it is assumed that the new hazard frame is defined at the first lidar measurement, and that consecutive lidar measurements are made in that same frame. To be consistent with this approach, a-priori known points are not used again after that point. It is also assumed that the thermal camera measures the plume source hot spot in tandem with the lidar measurement, so that the plume source landmark can be maintained in the SLAM map.

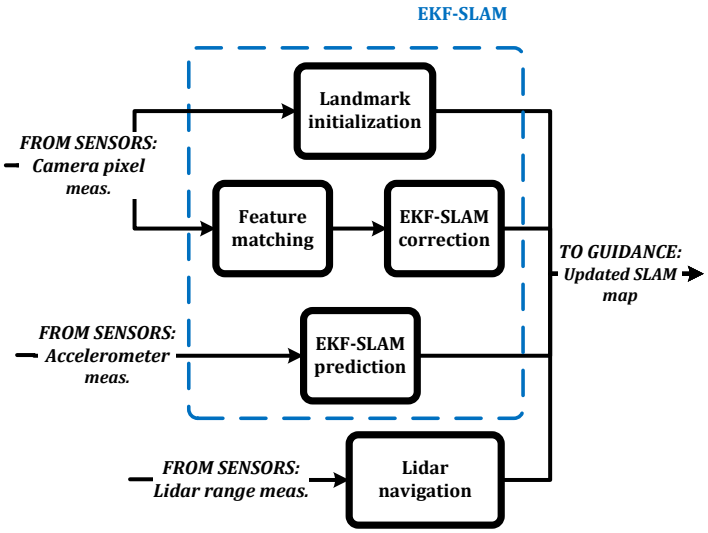

Figure 3.5: Architecture of the Navigation simulation block.

3.5 Hazard detection and avoidance block

The Hazard Detection and Avoidance block will evaluate the input co-registered sensor information for landing safety and proximity to the plume source and will provide a new landing target if the current one is found to be unsuitable. The process described in Section 2.3.3.1 was implemented and is summarized in Figure 3.6.

Sensor safety maps are input to the fuzzy safety reasoning step[20]. The resulting fused

[20]A note on obstacle detection: in this implementation, the K-means algorithm detects the illuminated faces of the obstacles only. To simplify that process, if the face of a particular obstacle is detected, then the entire obstacle is marked as detected.

safety map is fed to the next fuzzy landing reasoning step, that will calculate the overall "goodness" of the landing map, also considering proximity to a plume source. The proximity map simply measures the distance of each point of the HDA map to a plume source. The plume source location and the size of the no-go area around it can be fixed, or they can be informed by the plume source location and uncertainty estimated by the Navigation block. In this work however, the no-go area will be assumed fixed.

For the fuzzy spatial reasoning step, the landing goodness map is "blurred" by averaging the map values using a moving average filter of the size of a few lander footprints [107]. To get the temporal score of the map, the current landing goodness map is compared with the one from the previous HDA assessment. For the two maps to be comparable, the previous map (larger and less dense) is interpolated on the current map (smaller and denser). The two can then be compared point to point.

The final step in the HDA block, is the selection of a new target point. This takes place only if the landing score of the currently selected landing target falls below a set threshold. The highest scoring point from the final landing goodness map is then set as the new landing target.

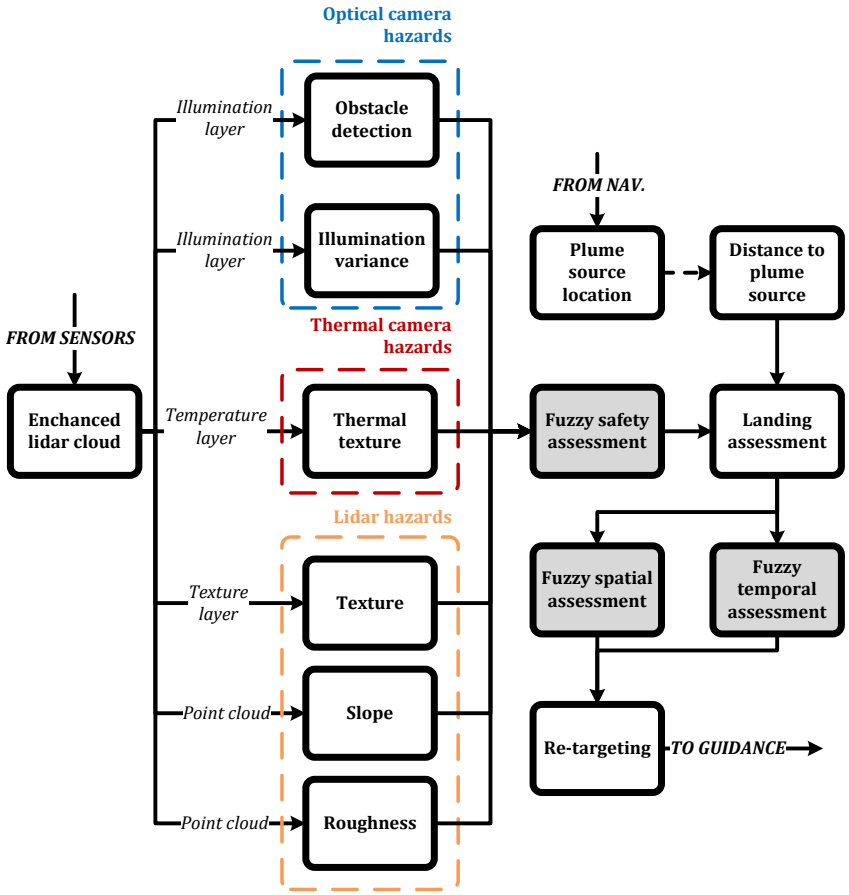

Figure 3.6: Architecture of the HDA simulation block. Gray boxes represent the Fuzzy reasoning blocks. The location of the plume source can be assumed to be known, or it can be given by the navigation function in real-time

3.6 Guidance block

The Guidance block uses a lander start state and an end state to calculate a viable trajectory and a thrust arc between the two. The general architecture of the Guidance simulation block is shown in Figure 3.7.

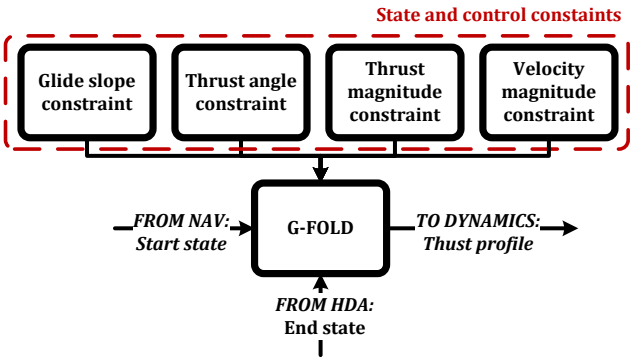

Figure 3.7: Architecture of the Guidance simulation block.

Several constraints must be satisfied: the calculated trajectory must remain within the glide slope cone. The angle between the local vertical and the calculated thrust vector must remain within the thrust angle constraint. The velocity magnitude must remain within its varying limits. Finally, the thrust magnitude must remain within set upper and lower limits.

The start and end states, the constraints and the lander dynamics are re-formulated in the G-FOLD problem formulation as given in Section 2.4.2. This convex optimal landing under constraints optimization problem is input into an optimal control problem solver. The solver outputs an optimal trajectory and thrust arc respecting the input constraints. Other useful parameters given is a "solution found" flag, the calculated propellant expenditure over time, etc.

The output thrust arc can then be fed to the dynamics block to be applied by the thrust actuators during landing, until a new guidance trajectory is calculated (e.g. after a re-targeting or other triggering event).

3.7 Lander Dynamics block

This block implements a point mass dynamic lander model, combining the gravitational and thrust accelerations from the environment and guidance blocks respectively. The accelerations are then added and forwarded to the INS accelerometer in the Navigation block, thus closing the loop of the simulation.

3.8 Software used

The LandingSim tool was entirely written in Matlab. Details will be briefly given concerning the specific toolboxes and functions used per block:

- **Environment block**: In-built Matlab graphics functions where used for the graphics-related processing in this block.
- **Sensors and Navigation**: A large part of the two blocks was put together based on the open-source Matlab SLAM-Toolbox [46, 53, 50] available for download at [103]. The EKF-SLAM approach used here was inspired by the SLAM-toolbox and its accompanying literature, and several lower level functions from that toolbox where used. Basic examples and lower level functions from the toolbox were heavily modified and combined into the present implementation.
- **Hazard Detection and Avoidance**: The Matlab in-built Fuzzy Logic Toolbox was extensively used for all fuzzy logic steps in this block [77, 104].
- **Guidance**: The G-FOLD convex optimization problem was implemented in the FALCON optimal control solver created by the Institute of Flight System Dynamics of the Munich Technical University (TUM), freely available for download at [105]. Another aerospace control problem solved with FALCON can be found e.g. in [106].

Chapter 4

Simulation Results

The performance and design results for the three central functions are presented in this chapter. Each individual function is first demonstrated and validated to work, and then its performance is assessed. Observations and modification proposals are made for each function. Finally, a combined simulation is performed.

The following types of analyses have been used in this work, applied per case as mentioned in the following sections:

- **Linear covariance analysis**: The complicated non-linear dynamics of a system are linearized and a state covariance matrix of the monitored parameters is carried forward during the run and updated and corrected at each appropriate simulation step [97, 98]. As this method involves an explicit formulation of the system uncertainties, a single simulation run can give a quick estimation of their evolution in time. However, more time should be dedicated in setting it up and it is less accurate than statistical methods.
- **Monte Carlo analysis**: A model of the system is simulated hundreds or thousands of times, while varying the uncertain initial conditions and system parameters. The results from the simulations are analyzed statistically. A Monte Carlo simulation can be useful for complex and non-linear systems where it is harder to know the underlying statistics, and potentially easier to set up. It is demands however long simulation run-times and a lot of computational resources.
- **Worst case analysis**: The worst-case values of the uncertain conditions or design parameters for a system are found, and a simulation of the system is performed with these values. It is then assumed that if the system performs successfully under these conditions, it will then be able to deal with all other initial conditions [99].

4.1 Navigation

In this section, the LandingSim tool will be used to design and investigate the performance of the navigation function.

4.1.1 Description of elements

As seen in Sections 2.2 and 3.4, the navigation function is built out of the following blocks:

- **Nav. state update** using a simple accelerometer model,
- **Absolute camera navigation** by feature matching to a-priori known optical and thermal (plume sources) landmarks for each respective camera type,
- **Relative camera navigation** by feature matching of detected a-priori unknown landmarks for each respective camera type,
- **Lidar navigation** by range estimation to multiple points on the surface, and
- **Plume source mapping** to improve the estimation of the location of the plume source, and to reduce its uncertainty to within acceptable values (if necessary).

The navigation function is built on the EKF-SLAM framework (Section 2.2.2). In short, the INS accelerometer predicts the navigation state of the lander, and the methods above are applied on the input of the relevant exteroceptive sensor to correct the navigation state of the lander, as well as of any observed landmarks. A SLAM map vector contains the navigation state of the lander and the landmarks, and the corresponding SLAM covariance matrix contains the variances and covariances of these vector elements.

For the purpose of testing and validating the Navigation function in this section and the HDA function in the following, a simple reference trajectory was generated by the guidance function as given in Table 4.10 below.

4.1.2 Validation and demonstration

To validate each of the navigation methods above, Monte Carlo runs were performed with the characteristics given in Table 4.1. For each run the navigation error was tracked during simulation, i.e. the difference between the real navigation state of the lander and that estimated by the navigation function. The navigation errors from each run were plotted together, along with the 3σ deviations of each navigation state element. By definition, the navigation errors should remain within the bounds set by the 3σ deviations $\sim 97\%$ of the time[21]. The navigation function, based on the principle of covariance analysis, can thus be validated using Monte Carlo simulations.

The simulation parameters common to all validation runs, and particular to each, are given in Table 4.1. Other relevant parameters are as given in Table 2.5. It must be noted here that the number of iterations in the Monte Carlo simulations (50) are not enough to produce a statistically significant validation, but is sufficient to demonstrate the proper functionality of the navigation function.

[21] In case a spread is observed in the estimated covariances (red dotted lines), this is due to the random selection of points to observe at each iteration.

Table 4.1: Navigation simulation parameters for validation. Elements of the first part of the table are common for all runs. Elements of the second part are relevant to each method being validated.

Common validation parameters	
Initial lander state and uncertainty	Same as Table 2.7
Number of Monte Carlo iterations	50
INS accel. sampling frequency	100 Hz
INS accel. meas. noise (1σ)	10^{-2} m/s^2
Abs. opt. nav. validation parameters	
Optical camera sampling period	10 s
Number of points tracked	10
Rel. nav. validation parameters	
Optical camera sampling period	5 s
Number of points tracked	10
Lidar parameters	
Number of measurements	4

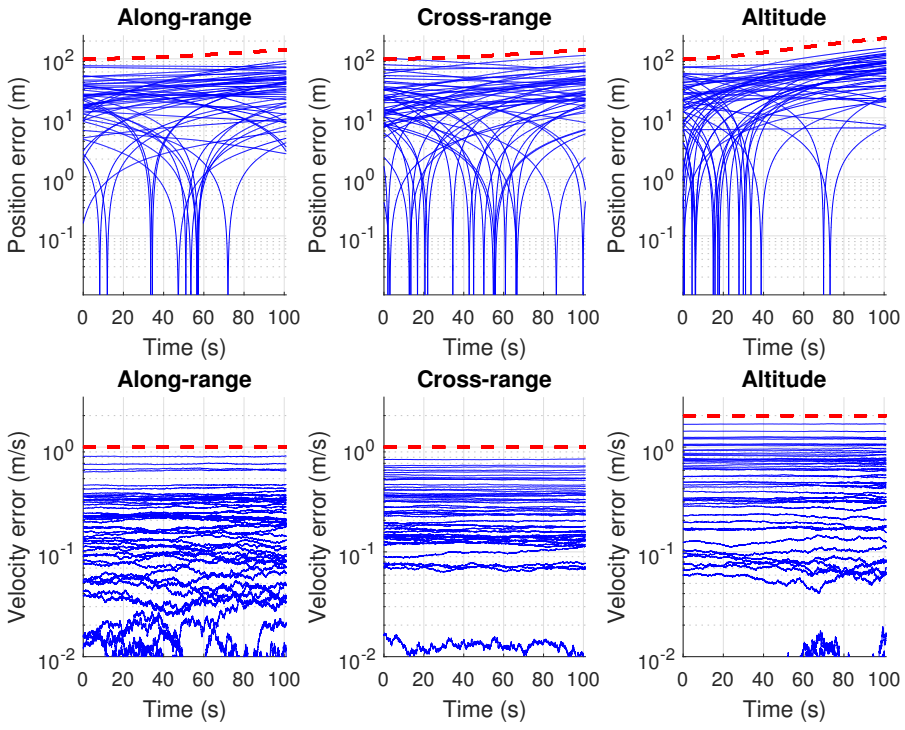

Figure 4.1: Validation simulation runs for the INS accelerometer Position and velocity errors of the MC runs are plotted in blue, while the 3σ covariances are given in red.

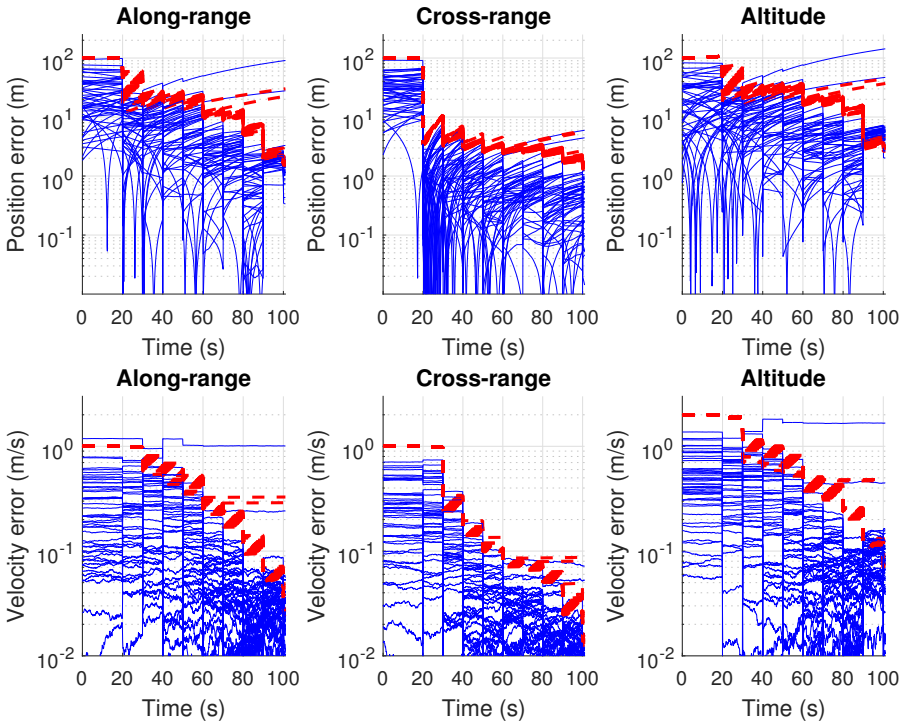

Figure 4.2: Validation simulation runs for absolute optical navigation. Position and velocity errors of the MC runs are plotted in blue, while the 3σ covariances are given in red.

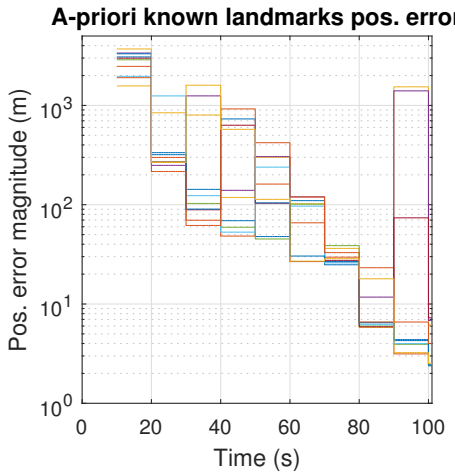

Figure 4.3: Position error of a-priori known landmarks during absolute optical navigation. Each landmark of the 10 tracked is represented by a different colored line.

Due to the large changes in scale in navigation error and uncertainty, in particular for the correction validation, the following plots are made in semi-logarithmic axes.

The validation results for the INS accelerometer are seen in Figure 4.1. The navigation state is updated according to the INS accelerometer equations at the given measurement rate, with no corrective measurements made. Due to the high accelerometer noise

assigned here, a significant drift in the covariance of all three components of the lander position. Almost no change is observed for the velocity covariances. All Monte Carlo-generated position and velocity errors are well within the 3σ covariance lines.

Next, the absolute optical camera navigation method was validated, with the results shown in Figure 4.2. The navigation state was being constantly updated as in the above case, but a correction step was applied every 10 s with measurements from an optical camera, observing 10 a-priori known landmarks at each measurement. The nav. state was then corrected through the EKF-SLAM formulation. It is assumed that 10 known landmarks can be found in every measurement of the optical camera, regardless of the lander altitude over the terrain (see Section 2.2.5)

A significant reduction both the position and the velocity uncertainties of the navigation state can be observed, mostly for the initial measurements and less so for the following ones. A more marked reduction is observed in the cross-range component, where little to no accelerations are applied (the nominal trajectory has no cross-range component). For some MC runs, the navigation error becomes larger than the corresponding covariance, something to be expected based on the fact that a 3σ covariance value covers $\sim 97\%$ of the cases. This increase in reduction for this component will be a feature of all the following validation runs.

It is seen in Figure 4.2 and other similar plots for other camera navigation cases in this section, that some covariance curves remain constant after a given point. This can be explained by the lack of any observable landmarks in the FOV of the camera. It is further seen that for the final part of the simulation, the estimated error consistently exceeds the one calculated by the EKF-SLAM. This discrepancy should be investigated in the future. However, it does not affect the performance of the current system, as absolute optical navigation is only used early in operations, where this effect is absent.

Figure 4.3 illustrates the evolution of the uncertainty in position that the tracked a-priori known landmarks have in the EKF-SLAM map. As seen, the uncertainty for each landmark is initialized at high, practically "infinite" values and is corrected with each observation. When tracked landmarks are lost from camera view, new ones take the place in the SLAM map, initialized with a high uncertainty (therefore the jumps in the plot). Overall, the lander and landmark position uncertainty are continually reduced with observation, with uncertainty tending towards zero.

Validation results for relative camera navigation, valid for both optical and thermal cameras, are seen in Figure 4.4. As discussed in Section 2.2.6, this method reduces the uncertainties in velocity, and at least negates the increase in position uncertainty. Figure 4.5 demonstrates the evolution of the position deviation of the tracked a-priori unknown landmarks. As expected, their uncertainty converges to the uncertainty of the lander position, as this is the minimum achievable, absent any other navigation methods.

Figure 4.6 illustrates the performance of navigation by observation of three a-priori known plume sources by the thermal camera. It is reminded here that the blurriness of detecting a hot spot in a thermal image is taken into account by setting the pixel error for the thermal camera to 10. It is seen, that even with three observed plume sources at all times, something that is not likely operationally due to the separation between plume sources on the SPT, the navigation performance is not sufficient. In addition, in

Figure 4.7, it is seen that the plume source itself is not adequately localized, due to the persistent high uncertainty in lander position.

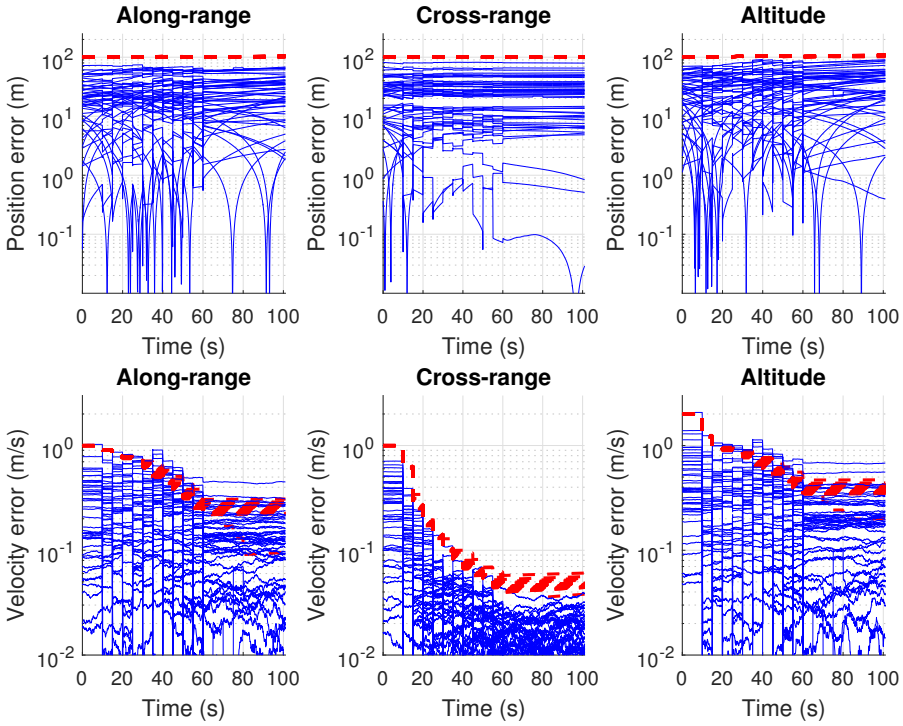

Figure 4.4: Validation simulation runs for relative camera navigation. Position and velocity errors of the MC runs are plotted in blue, while the 3σ covariances are given in red.

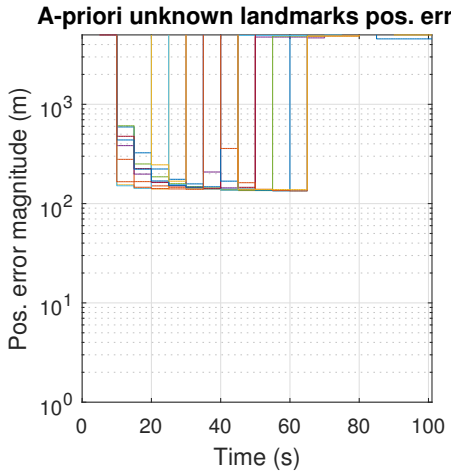

Figure 4.5: Position error of a-priori unknown landmarks during relative optical navigation. Each landmark of the 10 tracked is represented by a different colored line. Towards the end of the run, no individual landmark remains in the camera FOV for enough measurements to reduce its uncertainty.

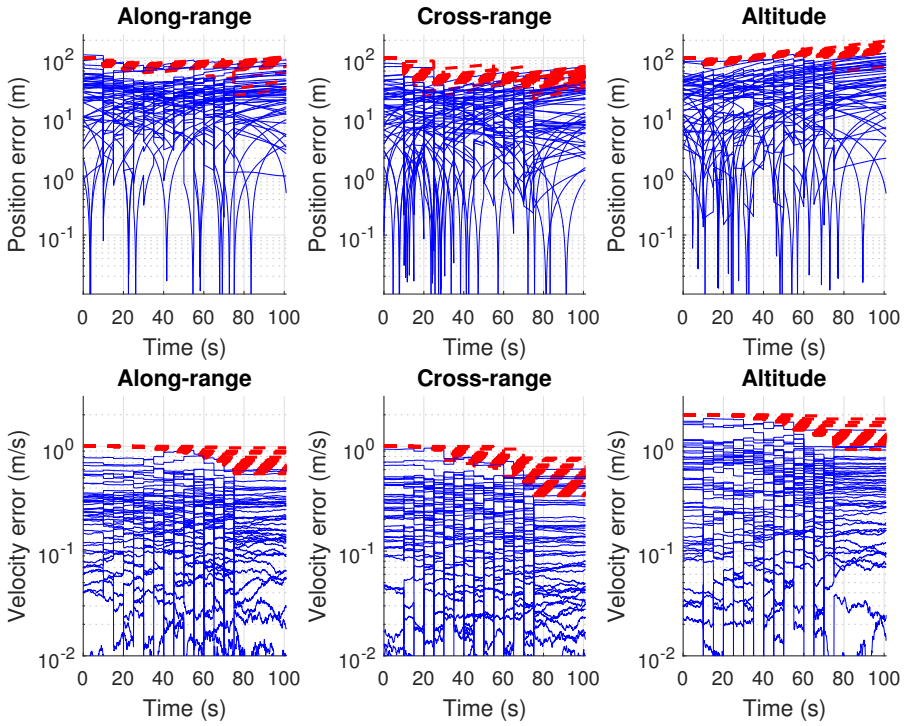

Figure 4.6: Validation simulation runs for absolute thermal navigation. Position and velocity errors of the MC runs are plotted in blue, while the 3σ covariances are given in red.

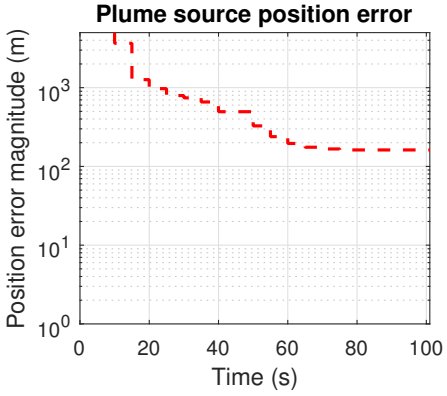

Figure 4.7: Position error of the a-priori known plume source landmark during absolute thermal navigation. The plume source is always in the thermal camera FOV, as it points towards it at all times.

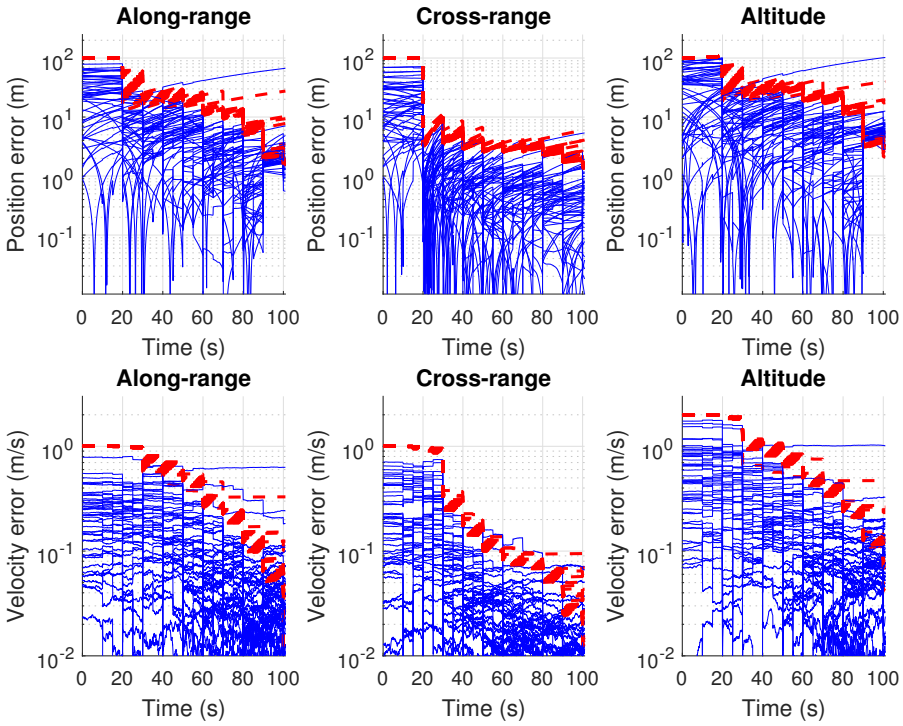

Figure 4.8: Validation simulation runs for combined camera navigation. Position and velocity errors of the MC runs are plotted in blue, while the 3σ covariances are given in red

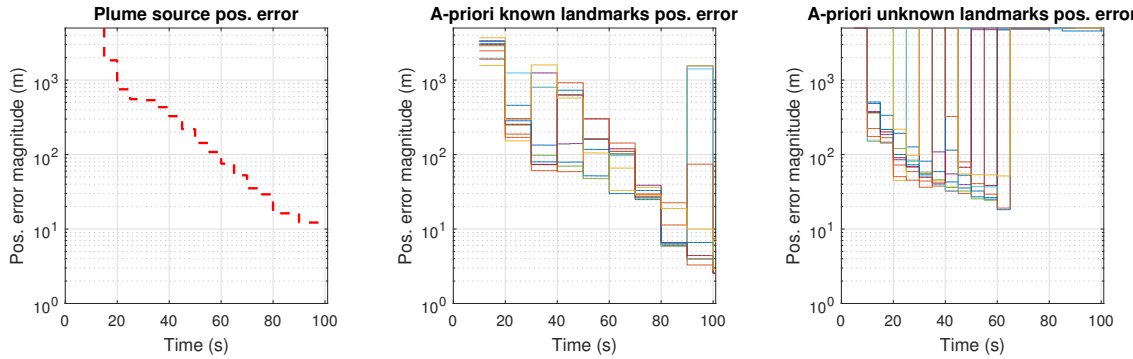

Figure 4.9: Position error of each landmark type (left to right: plume source, optical a-priori known, a-priori unknown).

Figures 4.8 and 4.9 show the evolution of lander and plume source navigation elements for a run combining all above camera navigation methods. The uncertainties of the navigation elements of the lander, plume source, and a-priori known and unknown landmarks are reduced progressively in concert.

For lidar navigation, four measurements are performed at set times, using 10×10 lidar beams. An impressive improvement is observed, even when using only a small proportion of the available number of beams (Figure 4.10)[22].

[22]This validation was performed on a different trajectory and with less iterations, but the validity of the results remains unaffected.

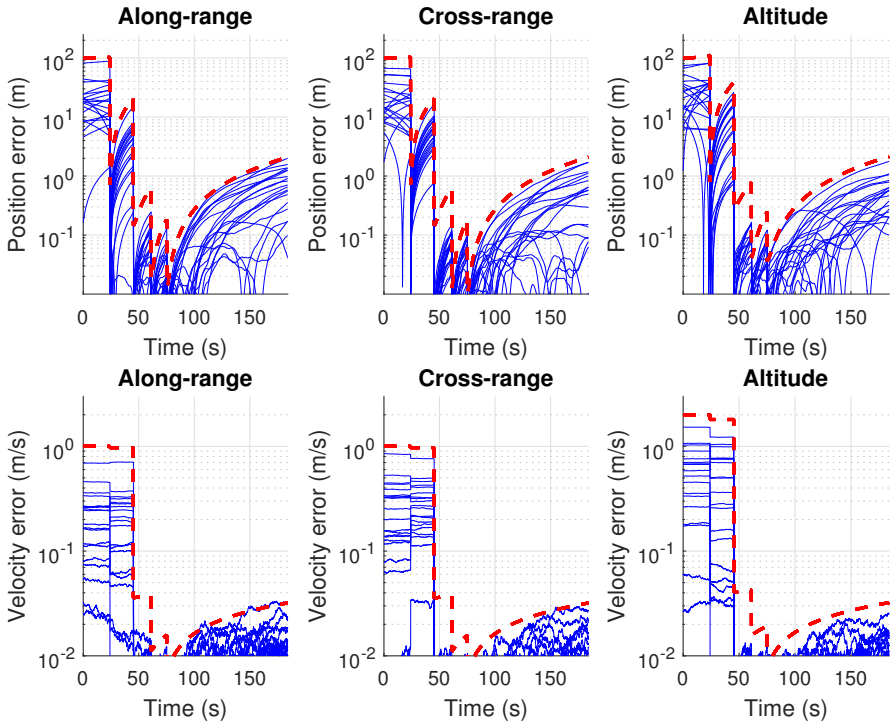

Figure 4.10: Validation simulation runs for lidar navigation. Position and velocity errors of the MC runs are plotted in blue, while the 3σ covariances are given in red.

4.1.3 Baseline navigation system and operations

Using the building blocks above, a navigation system and operation concept will be designed to reduce the initial lander navigation state and plume source position uncertainties to within the requirements (Section 1.5). The navigation requirements are summarized in Table 4.2.

Uncertainty must be first decreased in the world frame so that the assumption of a small difference between the world and hazard frame holds. Lidar can then further reduce uncertainties to the sub-meter accuracies needed for hazard avoidance. No more camera measurements will be taken in the new hazard frame, so the connection between the two frames is not necessary.

Absolute optical camera navigation will be first investigated. As mentioned in Section 2.2.5, there is a minimum altitude the optical camera can be used for absolute navigation and that altitude depends on the quality of the orbital reconnaissance before landing. Here it is assumed that that minimum altitude is at AG, for a range to the plume source of $\sim$ 1800 m, corresponding according to Figure 2.17 to a GSD of $\sim 1.6m$. This is a reasonable value compared to the 1-2m GSD expected as discussed in Section 2.2.5. Only one absolute camera measurement is to be performed then, taking place at AG.

Although lidar does a good job in reducing velocity uncertainties, and the INS accelerometer noise is low, relative navigation will still be performed to give robustness to the velocity estimation.

Two different broad landing scenarios can be identified for landing on the SPT. They are defined here, with their details given in Table 4.3:

- **Nominal operations**: The landing, defined as nominal, will take place at the height of Enceladan south summer, with good illumination conditions for all points of the landing terrain, with only part of the canyon bottom potentially in darkness. Absolute optical navigation can be performed at AG. A thermal camera is used for plume source tracking, and the optical camera is further used for relative navigation. Lidar navigation will start at at HDA HG (range = 1100 m) for sub-meter hazard frame accuracy. In case the plume source is not tracked but already known, no thermal tracking of it is necessary.
- **Landing in darkness**: The landing will take place during Enceladan winter, assuming no illumination. No absolute or relative navigation will be possible, even for some time into the braking phase, well before AG. This means that the lander navigation uncertainty will be increased (here assumed $\times 2$). Only the lidar and the thermal camera can then be used for navigation. Because of this, it will be desirable to perform lidar navigation as soon as possible (assumed at a range of $\sim$ 1500 m) to localize the lander in the hazard frame. As the camera is co-aligned with the lidar, the thermal imaged can be co-registered on the lidar cloud, and the plume source can be localized in the hazard frame. Everything relevant to the mission is then defined in the hazard frame, and the lander can proceed navigating with the lidar and thermal camera

The simulation results for the two scenarios are presented in Figure 4.11, where the magnitudes of the 3σ deviations are given for the lander position and velocity, and the plume source position estimations. The value of the mission requirement for each is also shown. It is seen that both scenarios have similar profiles in all three elements. Initially, a reduction in uncertainty due to the optical absolute navigation measurement takes place in the first scenario, and due to the first lidar measurement for the second scenario. The position uncertainties are gradually reduced, mainly due to the contribution of the lidar, and end up within the accuracy requirements. The uncertainties in velocity are more easily and quickly reduced. The uncertainties in the location of the plume are also slowly reduced down to the required levels. Both scenarios satisfy the landing position and velocity accuracy requirements, with the plume source accuracy requirement only marginally satisfied.

Table 4.2: Navigation function performance requirements.

Requirement	**Value**
Distance to plume source (3σ)	between $5m$ and $50m$
Lander Position accuracy (3σ, hazard frame)	$< 1m$
Lander Velocity accuracy (3σ, any component)	$0.25m/s$
Plume source position accuracy (3σ)	$10m$

Table 4.3: Nominal navigation parameters and requirements fulfillment. Initial and final/-target lander conditions are as given in the end of Chapter 2

Parameter	**Nominal**	**Landing in darkness**
Initial lander nav. uncertainties	1× of original	2× of original
Optical camera nav. parameters		
Absolute nav. measurement	at AG	-
Number of points tracked for abs. nav.	10	-
Relative nav. measurements	every 1 s	-
Number of points tracked for rel. nav.	10	-
Thermal camera nav. parameters		
Relative nav. measurements	-	every 1 s
Number of points tracked for rel. nav.	-	10
Plume source tracking measurement	every 5 s	every 5 s
Lidar nav. parameters		
Lidar nav. measurements, at ranges to target	[1100, 500, 200, 100] m	[1500, 1100, 500, 200, 100] m

An interesting effect is observed in the plume source position uncertainty plot, where the uncertainty remains fixed towards the end of operations. As the plume source for the reference scenario used here is always the pointing target for the sensor set, this is not to be attributed to the plume source going out of FOV, but the opposite: the plume source is in the center of the FOV, and as the lander approaches the target almost in a straight line towards the end of operations, there is almost no change in the pixel coordinates of the plume source. Its position estimate is therefore not improved, and the position uncertainty either.

The nominal scenario will be used in all the following analyses. It is improbable, due to several considerations, that landing will be attempted in the darkness of winter. Due to the long duration of winter on Enceladus however (7-8 Earth years, Section 1.4) it would be good to have the capability to be programmatically flexible.

4.1.4 Navigation sensitivity analysis

A sensitivity analysis will be performed to determine how system and operational parameters affect the navigation system performance. The parameters investigated for sensitivity are listed in Table 4.4. The values of critical parameters in the nominal scenario given above were varied and then the effect on performance was investigated. These effects were first visualized by jointly plotting the uncertainty magnitudes of lander position and velocity and of the plume source for all variation values. The effects were also given in the so-called sensitivity curves, where the values of the performance

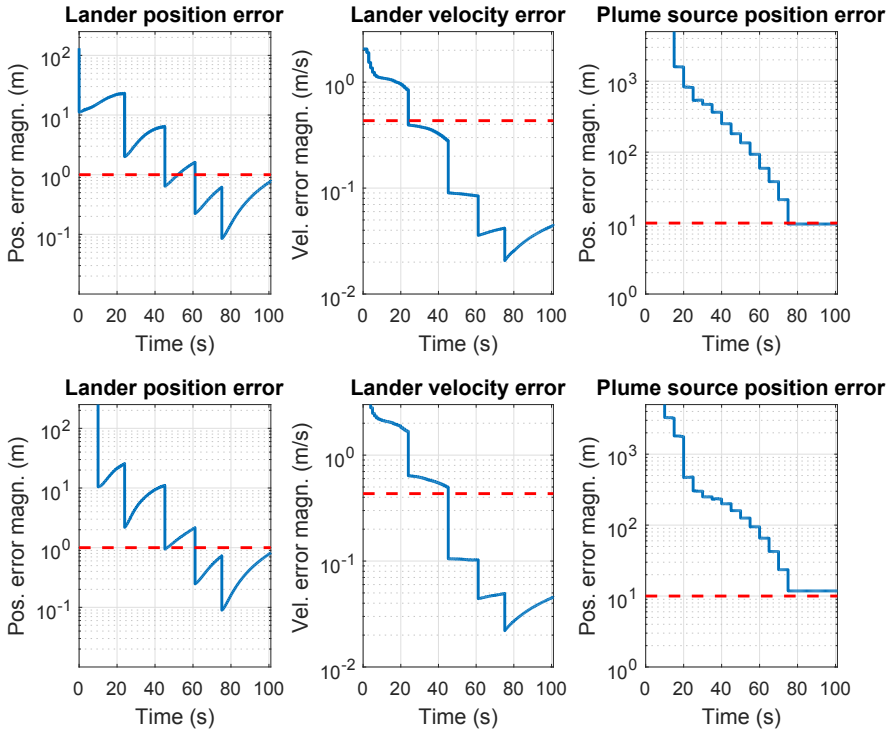

Figure 4.11: Performance profiles for the "nominal" (upper) and "landing in darkness" (lower) landing scenarios: magnitude of deviation in lander position and velocity, and plume source position (left to right). The respective requirements are marked with dotted red lines.

metrics where plotted against the varied values.

The magnitudes of the 3σ deviations fore lander position and velocity, and the plume source position are used in the following as performance metrics. The sensitivity analysis results are given in two types of plots in respect with these metrics. The varying parameters are given as blue dots on the sensitivity curves. Profile plots show the evolution of the metrics over time for each of these values of the varying parameters, and sensitivity curve plots plot the value of the varying parameters against the final values of the three navigation performance metrics at TG. The sensitivity profiles are plotted in shades of blue, with lighter shades representing higher values of the investigated varying parameter. The respective requirements as given in Table 4.2 are represented as dotted red horizontal lines.

The fulfillment of requirements at TG is important, but consistent performance during the entire approach phase is beneficial and adds robustness - both sets of plots are therefore indicative of performance.

First, sensitivity analysis will be performed to investigate system parameters and noises.

Figure 4.12 shows the sensitivity to the INS accelerometer noise. There a small variation in the profiles in position and velocity, especially in the beginning. Towards the end there is however no difference between the curves. The accuracy in plume source position seems to increase with low noise, however as can be seen in the respective profile plot, this result seems to be an artifact, due to the jump in the plume source certainty, that is assumed to be random. Similar behavior will be observed in some of the runs below

Table 4.4: Varying parameters in navigation sensitivity analysis. Initial and final/target lander conditions and sensor parameters are given in the end of Chapter 2.

System parameters	
INS accelerometer noise	$[0.1, 1, 10, 100]\times$ nominal
Lidar ranging noise	$[0.1, 0.5, 1, 2, 5]\times$ nominal
Lidar pointing error	$[0.1, 0.5, 1, 2, 5]\times$ nominal
Lidar beam number	$[2\times 2,\ 3\times 3,\ 5\times 5,\ 10\times 10,\ 20\times 20$
Optical cam. pixel error	$[0.1, 0.5, 1, 2, 5]\times$ nominal
Thermal cam. pixel error	$[0.1, 0.5, 1, 2, 5]\times$ nominal
Number of points tracked for abs. nav.	$[3, 5, 10, 20, 30]$
Number of points tracked for rel. nav.	$[3, 5, 10, 20, 30]$
Operational parameters	
Lander initial position uncertainty	$[0.1, 1, 10, 100]\times$ nominal
Lander initial velocity uncertainty	$[0.1, 1, 10, 100]\times$ nominal
INS accel. sampling frequency	$[10^{-3},\ 5*10^{-3},\ 10^{-2},\ 10^{-1}]$ Hz
Thermal cam. plume source tracking every	$[1, 2, 5, 10, 20]$ s
Optical cam. rel. nav. measurement every	$[1, 2, 5, 10, 20]$ s
Number of lidar nav. measurements	$[1, 2, 4, 6]$

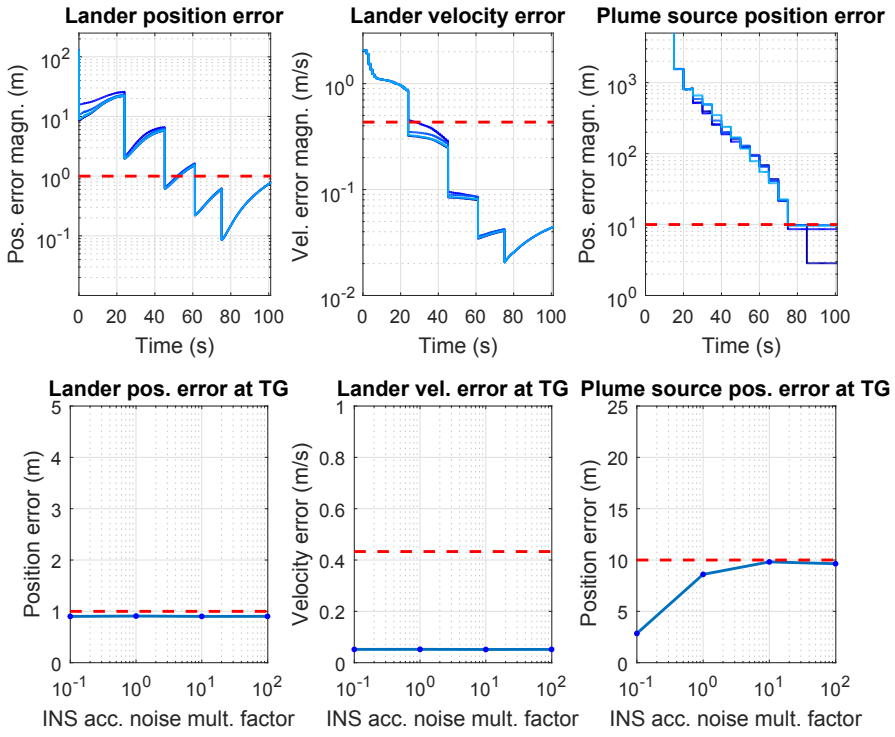

Figure 4.12: INS accelerometer sensitivity - Performance profiles and sensitivity curves. The blue lines are the 3σ deviations and the dotted red lines the respective requirements. See text for further explanation.

too. In those cases, the broader trends are investigated in this sensitivity curve.

Figure 4.13 shows the sensitivity of performance to the range noise levels of the lidar beams. In the profiles, a large effect is seen for the lander position and velocity uncertainties, which effect persists and is not corrected until TG. The result of that is that for noise levels larger than $1\times$ of the nominal, the final position uncertainty slightly exceeds the requirements. Velocity uncertainty remains well within, and plume source position uncertainty hover around the required values.

Figure 4.14 shows the sensitivity to the lidar pointing error. A small effect is seen in the early stages of operations for lander position and velocity uncertainty, but there are small effects remaining in the final values. A jump is seen in the plume source position sensitivity curve but it's interpreted as random as discussed above, the general trend for the plume source position uncertainty is also not changing.

Figure 4.15 shows sensitivity to the number of lidar beams used for navigation. The HD lidar used here (Table 2.4) has 128×128 beams, but only some of them are used for navigation purposes to save computational resources. The beam values given in the sensitivity figures' x-axis refer to the side length of the lidar point cloud, that is the number of beams used are $(n \times n)$. There is significant variation in the profiles throughout the landing for position and velocity. The final position values for small amounts of beam numbers exceed the requirements somewhat but not dramatically. In a relaxed requirements concept, the minimum acceptable value of $4\times$ can be used. A larger number of beams does not increase performance. Jumps are observed in the plume source position sensitivity plot, but it can be said that the general trend is stable.

Figure 4.16 shows sensitivity for the pointing error of the optical camera, used for absolute navigation at AG and for relative navigation at short intervals. This analysis corresponds to both an increased pixel error value and to a general pointing error.

A significant variation during operations is observed. An optical absolute navigation measurement of better quality can be said to give better head-start to the following lidar measurements. Due to the corrections from these succeeding lidar measurements, final values are minimally sensitive for all metrics.

Figure 4.17 shows sensitivity to the pointing error of the thermal camera. Again here pointing error is generically taken to mean pixel error and general pointing error. A dramatic effect is seen on plume source localization, which is logical, as an ability to better pin-point the feature corresponding to the plume source (hot spot/maximum intensity) means that the plume source landmark will be itself better localized in 3D space. If values can be achieved below the nominal 10 pixel error, a significant improvement will bring the uncertainty well under the requirement. This however might be challenging, as it is expected that hot-spots in the camera corresponding to plume sources will not be easy to pinpoint, meaning a high pixel error. Some sensitivity of the lander nav. state elements is also seen during operations but none in the final conditions at TG.

Figure 4.18 shows the sensitivity to the number of points tracked for optical absolute navigation. It can be considered as a design parameter or a metric of the quality of pre-landing orbital reconnaissance and the number of a-priori points available for sensing.

There is a significant difference on the lander position uncertainty right after the optical

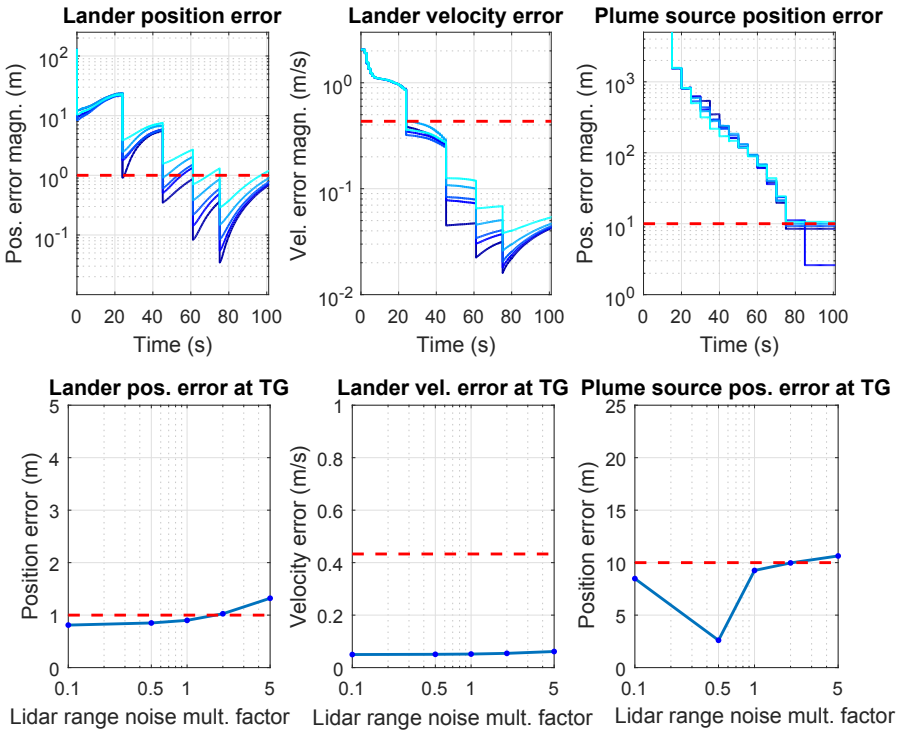

Figure 4.13: Lidar range noise sensitivity - Performance profiles and sensitivity curves. The blue lines are the 3σ deviations and the dotted red lines the respective requirements. See text for further explanation.

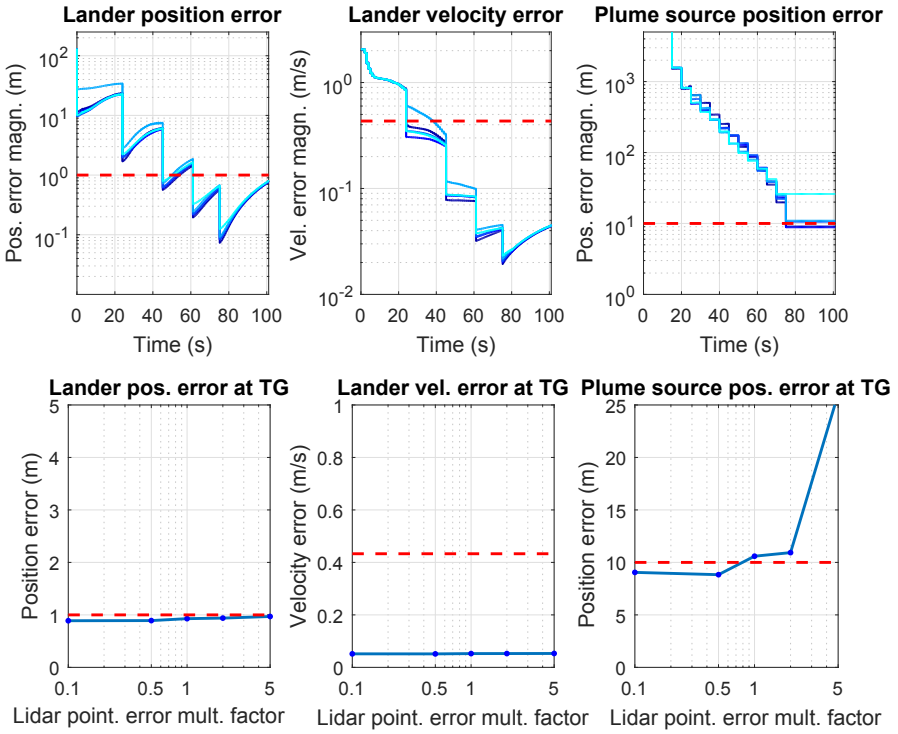

Figure 4.14: Lidar pointing error sensitivity - Performance profiles and sensitivity curves. The blue lines are the 3σ deviations and the dotted red lines the respective requirements. See text for further explanation.

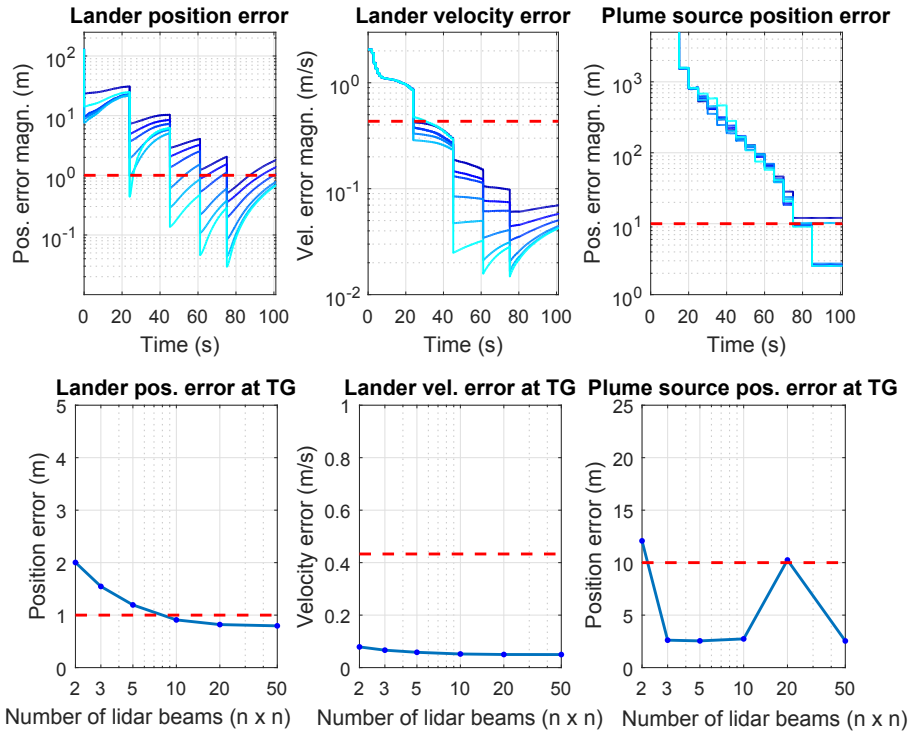

Figure 4.15: Lidar beam number sensitivity - Performance profiles and sensitivity curves. The blue lines are the 3σ deviations and the dotted red lines the respective requirements. See text for further explanation.

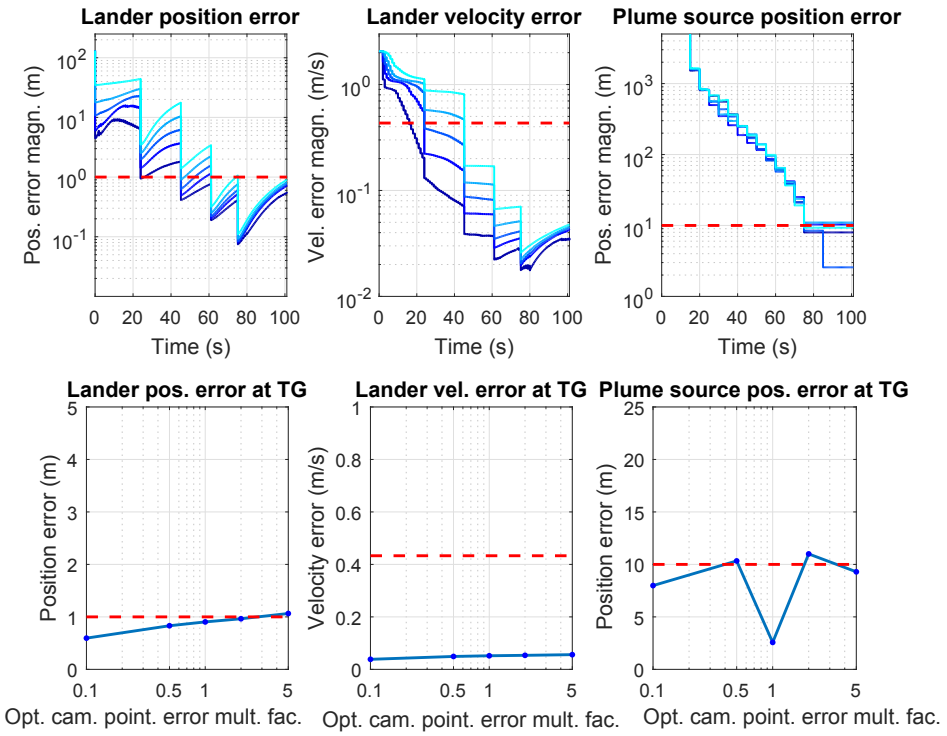

Figure 4.16: Optical camera pointing sensitivity - Performance profiles and sensitivity curves. The blue lines are the 3σ deviations and the dotted red lines the respective requirements. See text for further explanation.

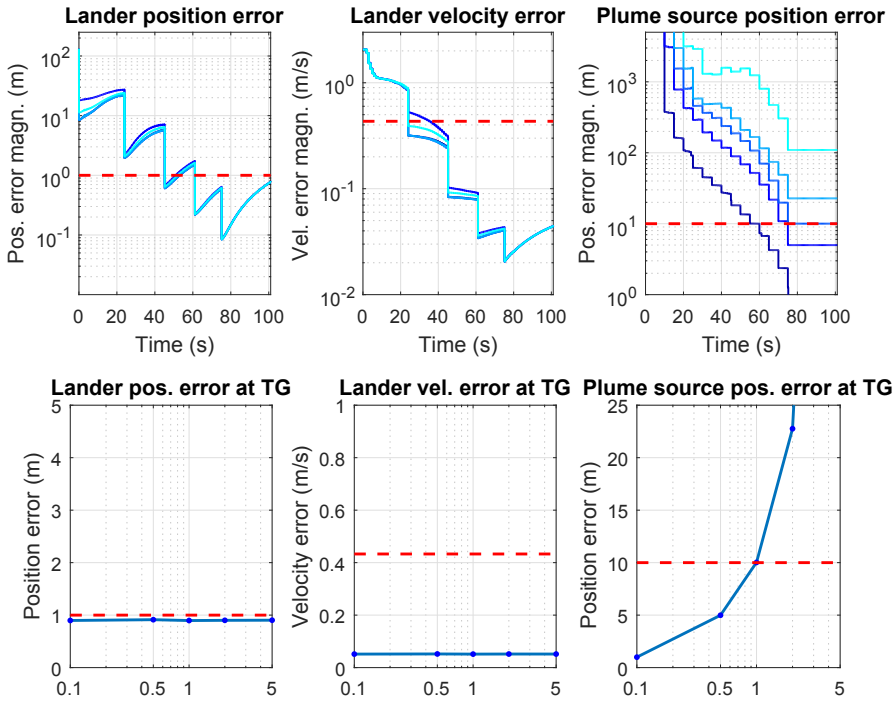

Figure 4.17: Thermal camera pointing error sensitivity - Performance profiles and sensitivity curves. The blue lines are the 3σ deviations and the dotted red lines the respective requirements. See text for further explanation.

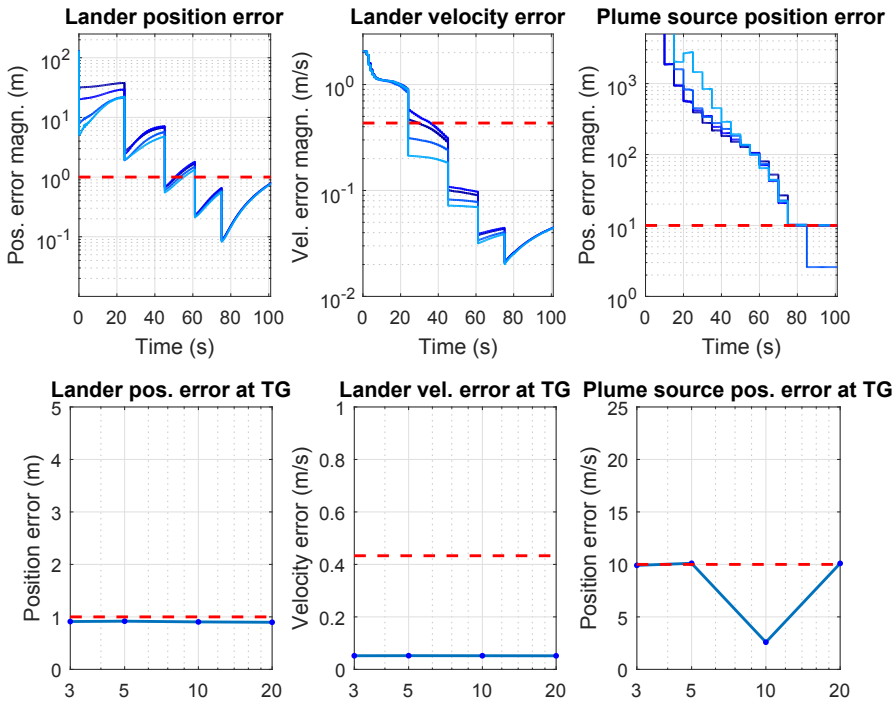

Figure 4.18: Sensitivity to number of points observed for optical absolute navigation - Performance profiles and sensitivity curves. The blue lines are the 3σ deviations and the dotted red lines the respective requirements. See text for further explanation.

absolute navigation measurement at AG, but the differences are equalized over time. The same happens with velocity and less markedly for the plume source location uncertainty. There is no sensitivity for final values at TG.

Figure 4.19 shows the sensitivity to the number of points tracked for optical relative navigation. This parameter mainly effects velocity, which is seen by the small effect in the velocity profile. Otherwise no sensitivity is observed for any of the metrics at TG.

The operational parameters will now be investigated.

Figure 4.20 shows the sensitivity to lander initial position. Lander position accuracy is corrected to approximately the same levels for all varying values after optical absolute navigation at AG. After that almost no variation is seen in the profiles. No sensitivity is seen in values at TG.

Figure 4.21 shows the sensitivity to lander initial velocity. Large initial dispersions in lander velocity are gradually corrected to coincide until the end of operations at TG. Some variation for the lander position at first stages that is eliminated by TG. No sensitivity for the plume source position uncertainty is observed.

Figure 4.22 shows the sensitivity to sampling frequency for the INS accelerometer. Some sensitivity is seen for the lander position uncertainty that exceeds requirements for higher noise levels, but not dramatically. Lander velocity uncertainty is quite sensitive but still remains well under the requirements. Some uncertain behavior is observed for the plume source position uncertainty but no particular trend can be said to be dominant.

Figure 4.23 shows the sensitivity to the number of lidar navigation measurements. Measurements in addition to the nominal four are timed roughly equidistantly in between them. A large effect is seen for lander position uncertainty: less measurements than the nominal significantly exceed the requirement, and more measurements than nominal do not add value. There is some sensitivity for velocity uncertainty but the values remain well within requirements in any case. Uncertain behavior is seen for the plume source position uncertainty, but no particular trend is observed.

Finally, Figure 4.24 shows the sensitivity to the measurement frequency of the thermal camera for plume source tracking. An impressive sensitivity is observed for the plume source position uncertainty. A higher measurement rate gives significant improvement, even with the original relatively low pixel accuracy of the thermal camera. On the contrary, a lower measurement rate ”breaks” the process and results in unacceptably high plume source position uncertainty estimates. Increasing the thermal camera measurement frequency is thus potentially a resource-efficient way to increase performance in plume source localization, in contrast e.g. with trying to decrease the thermal camera pixel error.

4.1.5 Results summary and initial conclusions

Informative observations on the above results can now be gathered, that will be helpful with the next design iteration of the navigation concept. In the following significant conclusions are summarized in bullet form, for easier reference

- Considering the lidar, the range noise has some effect on performance throughout

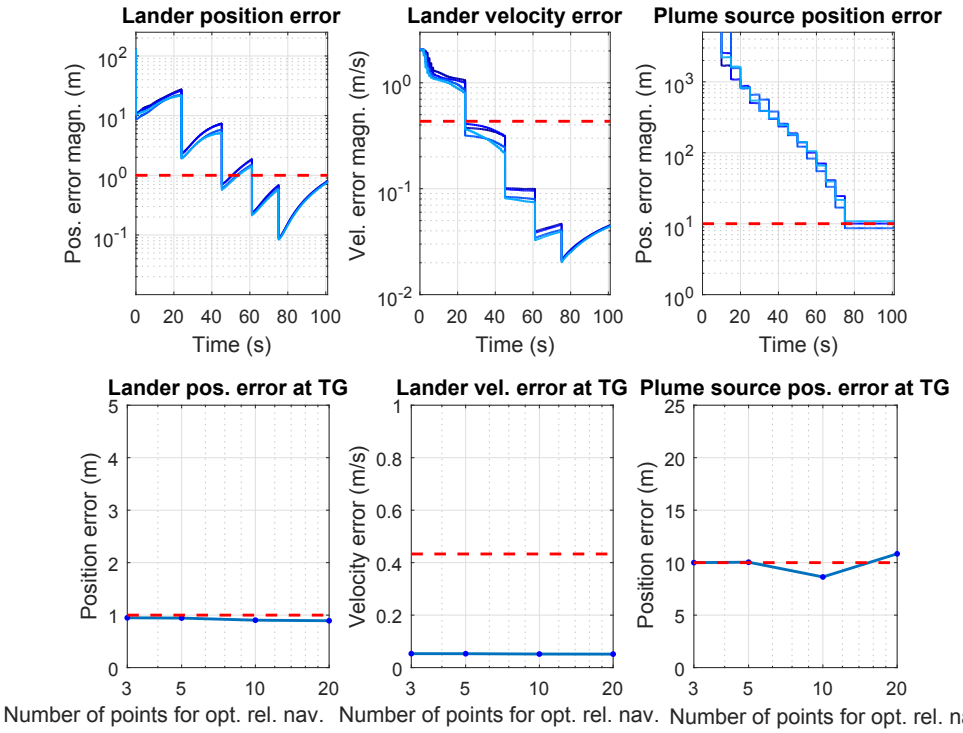

Figure 4.19: Sensitivity to number of points observed for optical relative navigation - Performance profiles and sensitivity curves. The blue lines are the 3σ deviations and the dotted red lines the respective requirements. See text for further explanation.

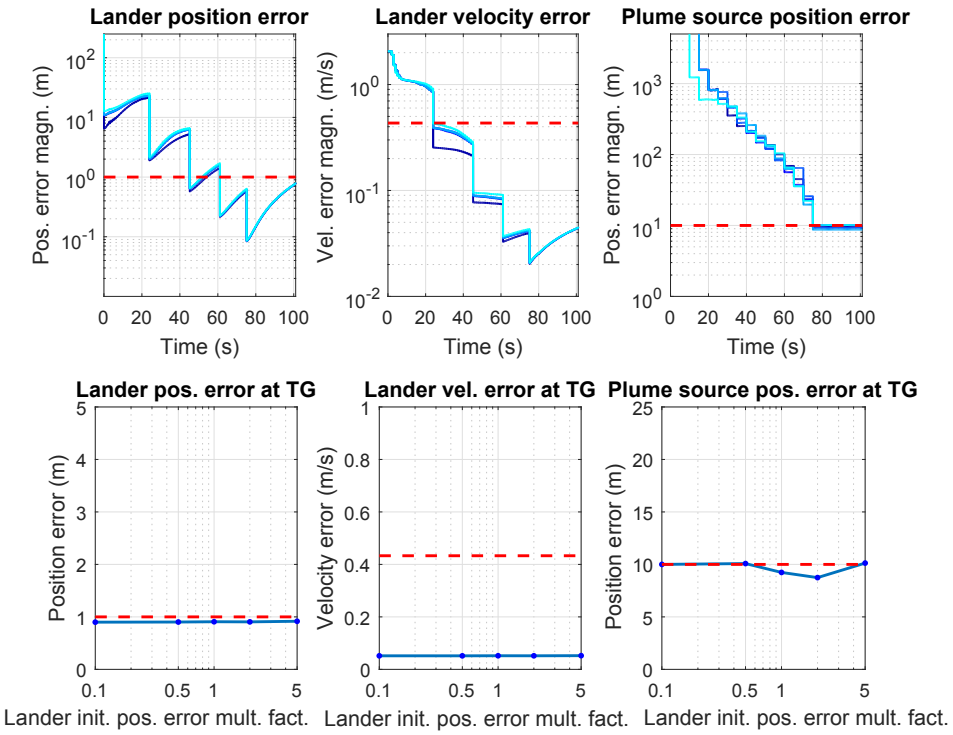

Figure 4.20: Initial lander position uncertainty sensitivity - Performance profiles and sensitivity curves. The blue lines are the 3σ deviations and the dotted red lines the respective requirements. See text for further explanation.

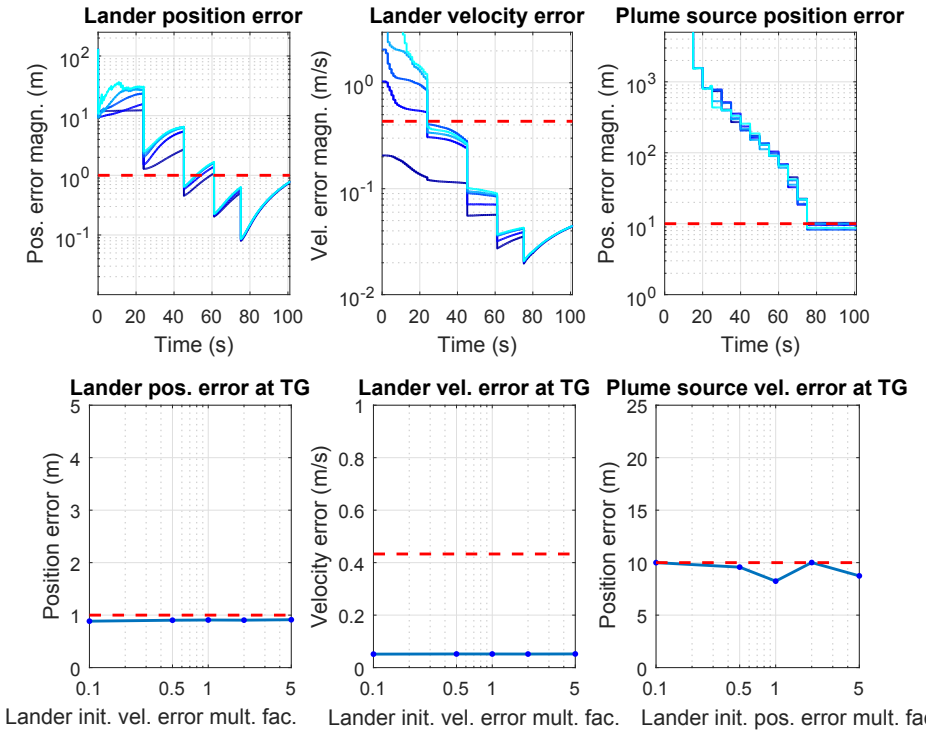

Figure 4.21: Initial lander velocity uncertainty sensitivity - Performance profiles and sensitivity curves. The blue lines are the 3σ deviations and the dotted red lines the respective requirements. See text for further explanation.

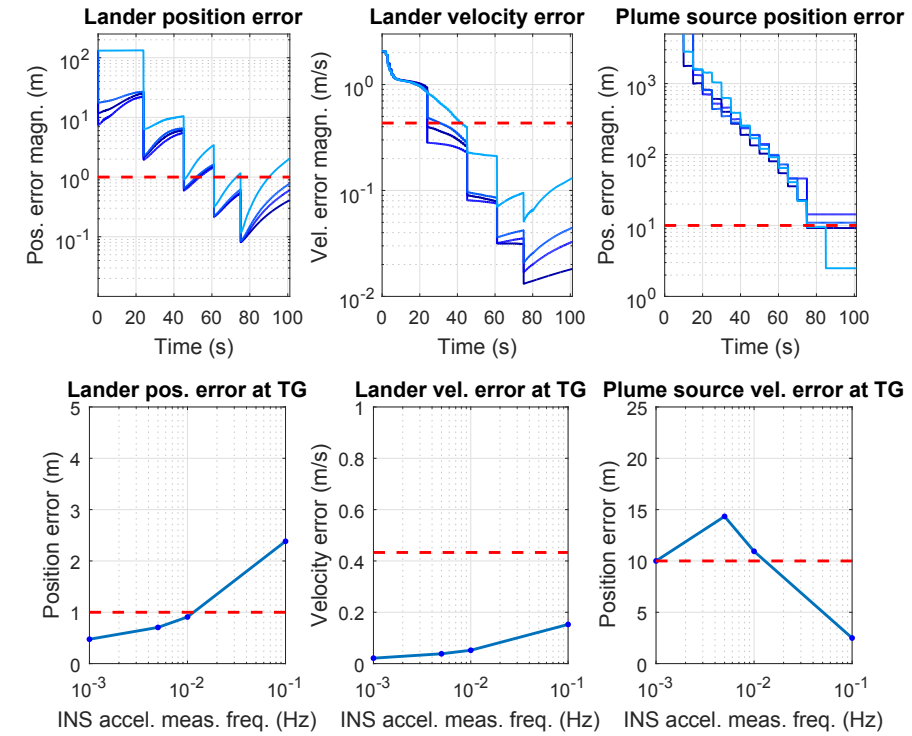

Figure 4.22: INS accelerometer measurement frequency sensitivity - Performance profiles and sensitivity curves. The blue lines are the 3σ deviations and the dotted red lines the respective requirements. See text for further explanation.

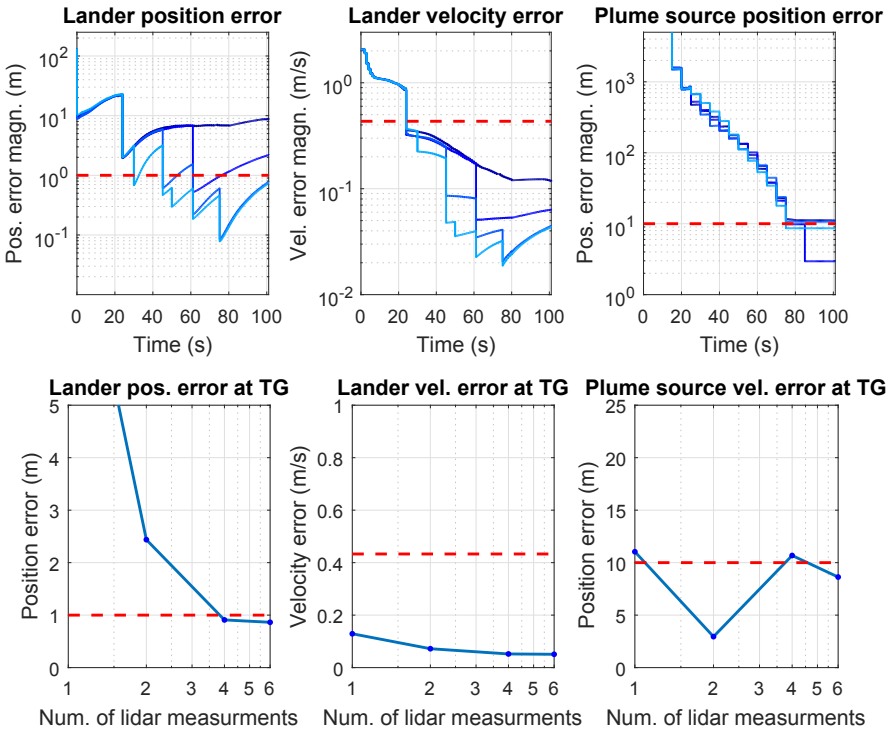

Figure 4.23: Number of lidar measurements sensitivity - Performance profiles and sensitivity curves. The blue lines are the 3σ deviations and the dotted red lines the respective requirements. See text for further explanation.

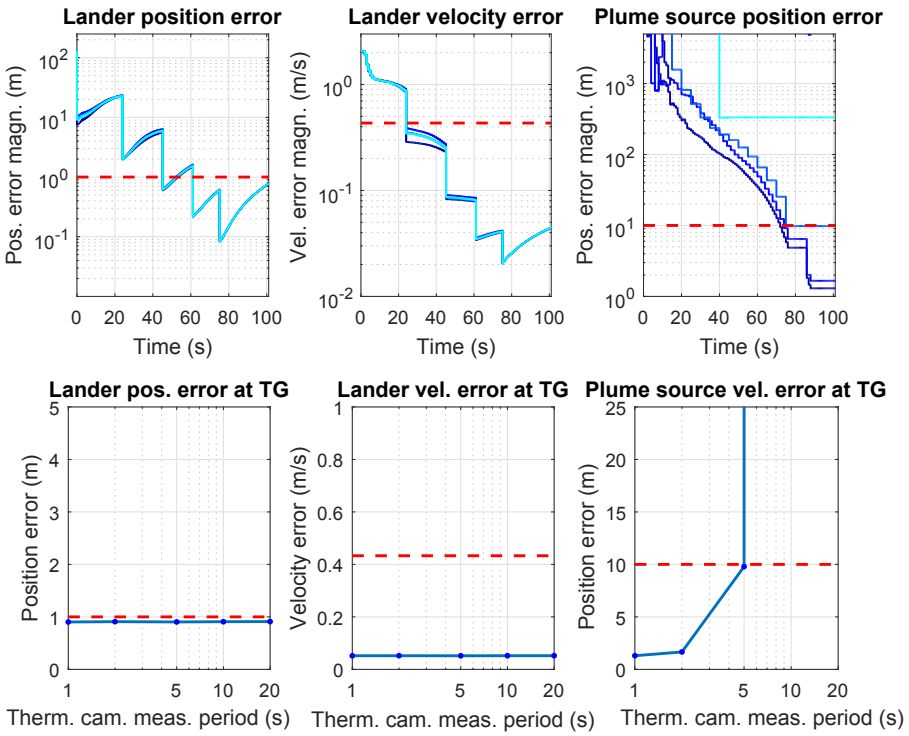

Figure 4.24: Thermal camera measurement frequency sensitivity - Performance profiles and sensitivity curves. The blue lines are the 3σ deviations and the dotted red lines the respective requirements. See text for further explanation.

operations and at TG, this effect is however not dramatic. The current relatively low 10×10 nominal number of beams is sufficient for accuracy. Although a lower number of beams would not violate the position accuracy requirement by much, it would be more robust to have a relatively larger number of beams for robustly defining the hazard frame with the lidar. The number of lidar measurements should not be decreased, and offers limited value if increased.

- Depending on the importance given to the performance of the last optical camera absolute navigation measurement, the generic pointing accuracy of the optical camera, expressed here via the pixel error, is important to navigation operations. The camera pointing error has a big effect on that measurement, as do the number of a-priori known points observed. To improve performance, considering the pointing error to be relatively low in all cases, better quality orbital reconnaissance would be necessary, resulting in more and better localized a-priori known points to be observed. Considering adding a further measurement at a lower altitude, it might a) not be possible due to the inability to create orbital maps with a GSD lower than the one assumed, and b) will possibly not add value, considering that the definition of a new hazard frame with the lidar might be a better solution.

- The main function of the thermal camera is observing and localizing the plume source. However this is done inaccurately mainly due to the high pixel uncertainty associated with detecting a relatively spread out hot spot with the thermal camera. Steps can be taken to improve this accuracy, e.g. by using centroiding methods for sub-pixel pinpointing, or somehow increasing the image quality of the thermal camera. A more resource-efficient way however it to increase the measurement frequency of the thermal camera. That way, even with a low resolution camera, simply detecting a local maximum, the plume source can be localized very accurately.

- In the reference scenario used for the study of the navigation function in this section, the plume source and the landing target were coincident. The thermal camera along with the entire gimbaled sensor set could always point at it, observe it and localize it. Somewhat unexpectedly this resulted in the unwanted effect that the plume source is in the center of the FOV, and not able to be well observed due to its staticness in the thermal camera image towards, as the lander approaches it in an almost straight line towards then end of the landing. In real operations, as the sensor set will be pointed to a target landing site different than the plume source, the plume source will be further than the center of the thermal image and will thus be more ”mobile” in consecutive thermal images, thus making the plume source localization better. Due to the wide FOV of the thermal camera, it will not go out of FOV until the very last stages of landing. Plume source localization in a more realistic scenario is thus expected to perform better.

- The navigation function can deal well with initial lander position uncertainty, as this uncertainty is immediately corrected by the optical absolute navigation measurement. Uncertainties in initial velocity however are more challenging, and are only corrected after several lidar measurements.

- INS accelerometer and optical relative navigation measurements are to be kept to

their respective nominal values. Neither is very impactful to the final result.

4.2 Hazard Detection and Avoidance

The results of the detailed analysis of the HDA function in the LandingSim tool will be discussed in this chapter.

4.2.1 Description of elements

The high-resolution 125 × 125 m slice of the DTM around the plume source on a tiger stripe canyon bottom as described in Section 3.2 was used here as a base map for the further HDA analysis. The characteristics of this terrain model are given in Table 4.5. The near and middle terrain models are given also in Figure 4.25 for easy reference. The illumination, snow cover, and temperature layers of the terrain are shown in Figure 4.26.

Table 4.6 summarizes the HDA-related landing requirements. Based on these a ground truth binary (suitable/unsuitable) landing safety map can be created for the baseline terrain model above. This map is shown in Figure 4.27. It should be noted that the "soft" requirement/goal about preferring a landing on ice rather than snow is not captured in this map.

Table 4.5: Terrain model parameters, as used in this section. Values in the table are based on the equations given in Sec. 3.2

Parameter	Value
Map size	125 m
Map resolution	10 cm
Number of obstacles	1000
Obstacle size limits	0.2 - 10 m
Sun angles	Azimuth = 0°, Elevation = 30°
Snow cover parameters	$\mu_{snow} = 20°$, $\sigma_{snow} = 10°$
Number of no-snow zones	20
Surface temperature parameters	T_{amb} = 100 K, T_{PlS} = 180 K, σ_{PlS} = 30 m, $T_{mod,snow}$ = 10 K

Table 4.6: HDA function performance requirements/Requirements on the lander footprint centered on the chosen landing site.

Requirement	Value
Max. slope	15°
Max. obstacle height/roughness	50 cm
Distance to plume source	$5 - 50$ m
Prefer ice rather than snow	

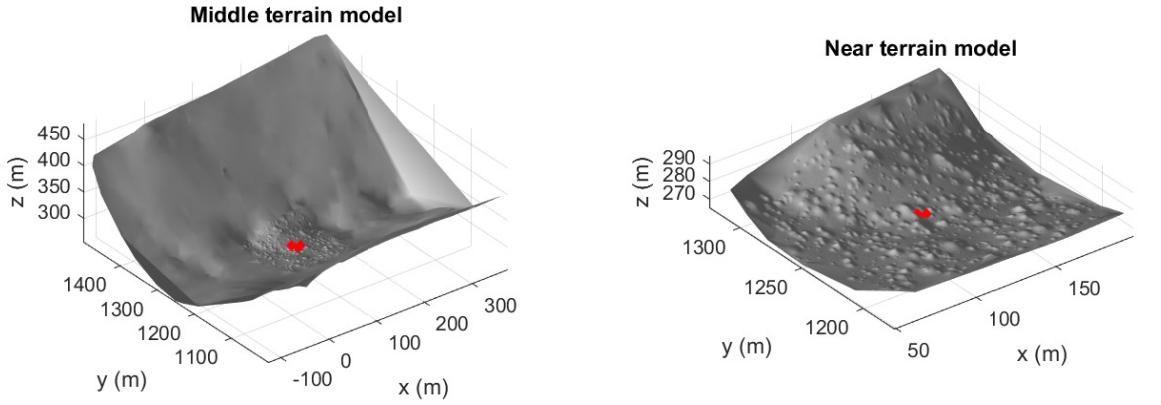

Figure 4.25: Illuminated middle and near terrain models. The plume source location is marked with a red x. The vertical face at the end of the canyon in the first image is an artifact of the triangulation process

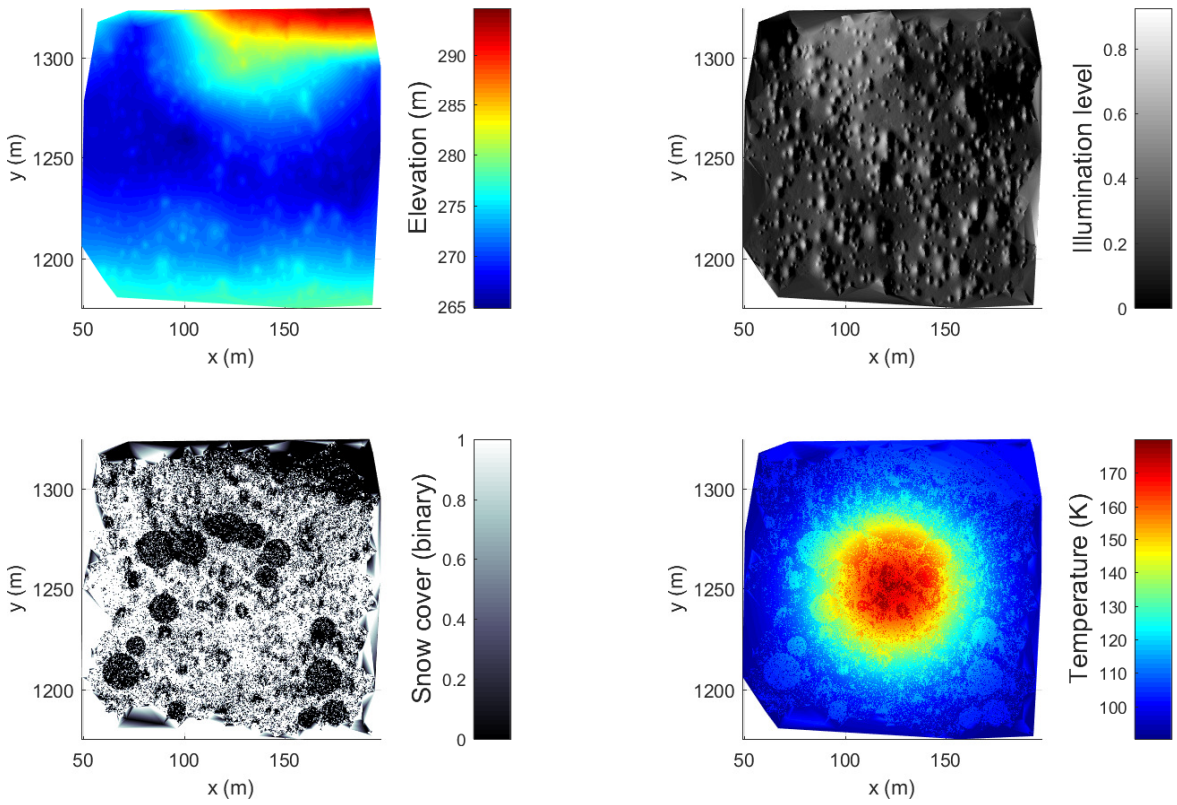

Figure 4.26: The input terrain model and its information layers: elevation (upper left), illumination (upper right), snow cover (lower left), and temperature (lower right).

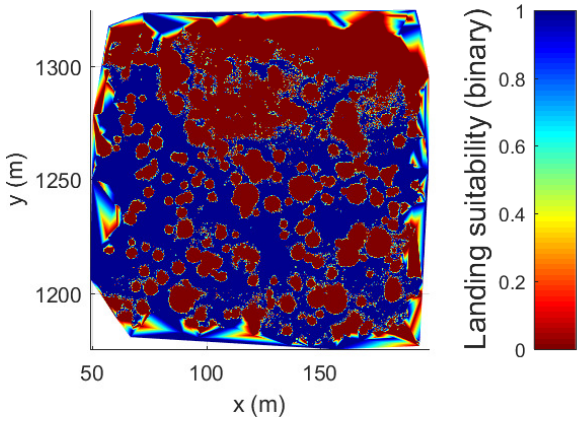

Figure 4.27: Ground truth binary (safe/unsafe) safety map. Blue areas are suitable for landing while red areas are not.

4.2.2 Validation and demonstration

To validate the Fuzzy-reasoning-based HDA process implemented here, it will be applied on the baseline terrain map and the results will be compared to the ground-truth landing suitability map. The successive steps of the HDA process as given in Section 2.3.3 can be seen in the following figures in this section.

Figures 4.28, 4.29, and 4.30 show the terrain hazard maps generated by the optical camera, the lidar and the thermal camera respectively. All sensor measurements are considered ideal (no noises or errors present). For the thermal camera, the thermal background map is created assuming perfect knowledge of the corresponding surface temperature characteristics, and the residual temperature map is extracted. Values lower than the mean on this last map are identified as the colder snow and values larger than the mean as the warmer ice. The fused thermal-lidar texture map is also given in Figure 4.30, combining the texture measurements of these two sensors. It is reminded that the score in this map represents the percentage of a lander footprint centered on each map point that is covered by snow.

The fused safety maps for each sensor, combining the safety features for each individual sensor, are shown in Figure 4.31. For the optical camera the obstacles are the main source of hazards, with illumination variance coming second. As slope is not being sensed, non-obstacle high slope areas are given high scores in the optical camera hazard map. For the lidar, the best scoring points are ones that are flat, smooth, and not covered in snow, followed by also smooth and flat but snow covered points.

The fused safety score for all sensors, the distance from the target plume source, and the overall landing goodness maps are given in Figure 4.32. The point in the final map with the best landing score is selected as the new landing target. It must be noted that the final landing score map has gone through the spatial but not the temporal assessment step. The effect of both these steps is to be demonstrated in Section 4.2.4.

To validate the implemented fuzzy-reasoning-based HDA method, the resulting landing safety map (Figure 4.32, upper left) will be compared with the binary ground truth

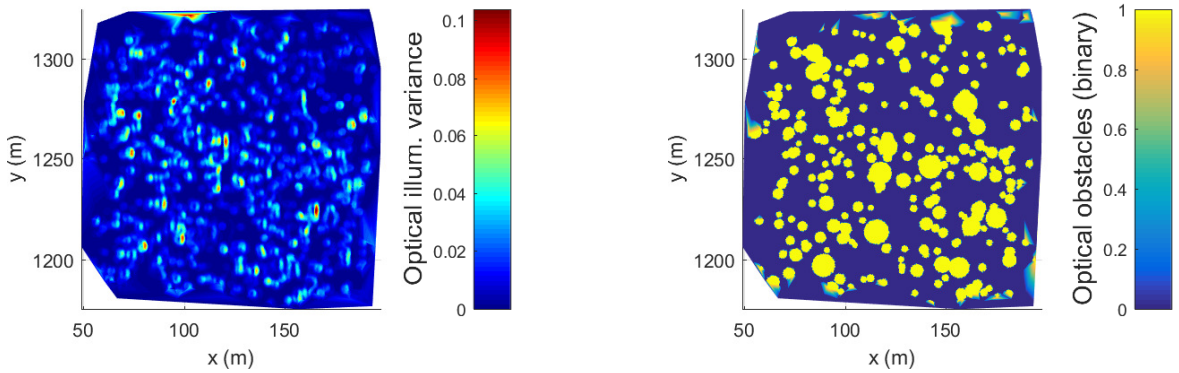

Figure 4.28: Maps for each safety feature sensed by the optical camera: illumination variance – a measure of terrain roughness (left), and obstacles (right).

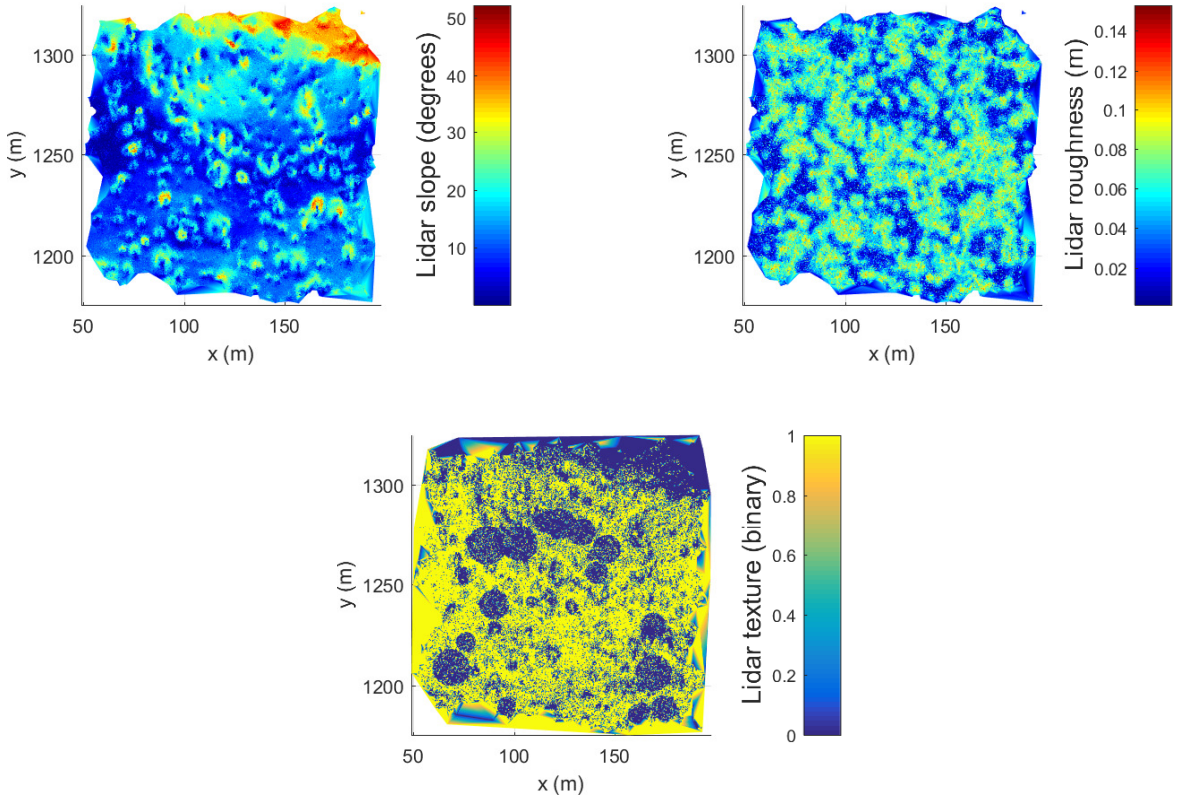

Figure 4.29: Maps for each safety feature sensed by the lidar: slope (upper left), roughness (upper right), and surface texture – snow coverage (lower).

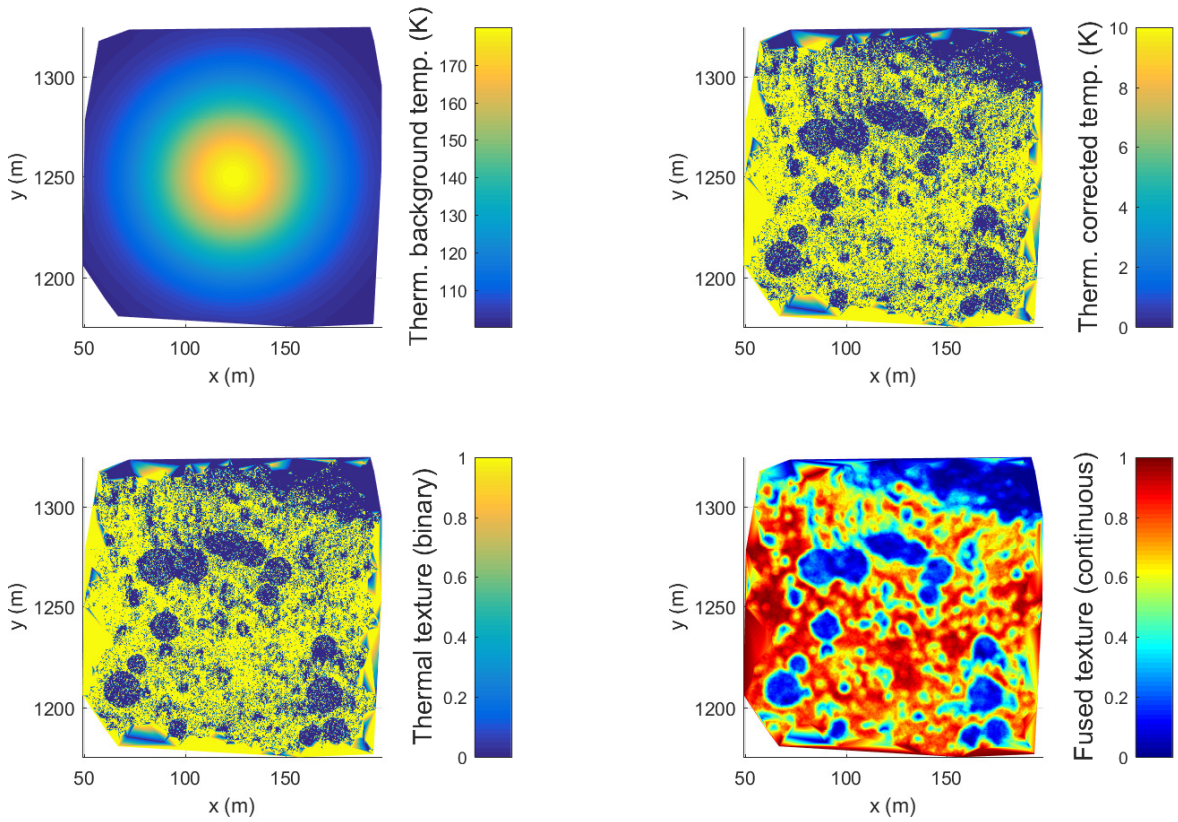

Figure 4.30: Maps for the thermal camera: thermal background due to heating from the plume source (upper left), residual thermal map (upper right), and surface texture – snow coverage (lower left). The fused and averaged thermal-lidar texture map is given here for completeness (lower right). In the plots where texture is concerned, 0 denotes ice, and 1 denotes snow.

safety map (Figure 4.27). The safety map is chosen for the comparison instead of e.g. the final landing goodness map, since it is the most important element of the HDA process, and it covers the entire landing area (compared to e.g. the final score map that covers a 50 m radius circle).

But how would one go about comparing the binary ground truth safety map, to the landing safety map scored continuously from 0 to 10? The method chosen is to convert the landing safety map from continuous to a binary good/bad score. To do this, a score threshold can be set above which landing sites are considered good, and below which bad. Given a threshold, and having converted the landing safety map to a binary above-good/below-bad map, it can then be compared to the ground truth landing suitability map. The results of this comparison for each point can be either of the three:

- **False Negative (FN)**: a point that is in reality unsuitable for landing, is given as suitable by the HDA process.
- **False Positive (FP)**: a point that is in reality suitable for landing, is given as unsuitable by the HDA process.
- **True Measurement (TM)**: the landing suitability of a point is given correctly by the HDA process.

"In reality" above, means the marking of a given point in the ground truth binary suitability map. In the above, the terms "positive" and "negative" refer then to hazardousness: a FN therefore falsely gives a negative hazardousness result while and FP falsely gives a positive hazardousness.

But what should the landing goodness score threshold be? To investigate this, the threshold was varied from 4 to 8, and the resulting binary map was compared to the ground truth binary goodness map for each value of the threshold. The percentages of false negatives, false positives, and true measurements where logged for each iteration and are presented as a plot in Figure 4.33. The proportion of the map that is marked as suitable for landing is also noted ("suit.").

As expected, by varying the score threshold the three above metrics also vary. It is

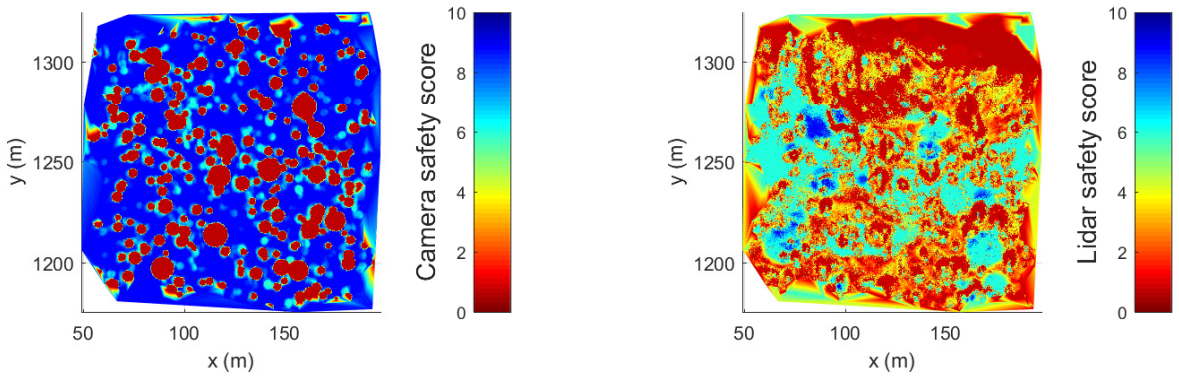

Figure 4.31: Fused safety maps for each individual sensor: camera safety map (left), lidar safety map (right).

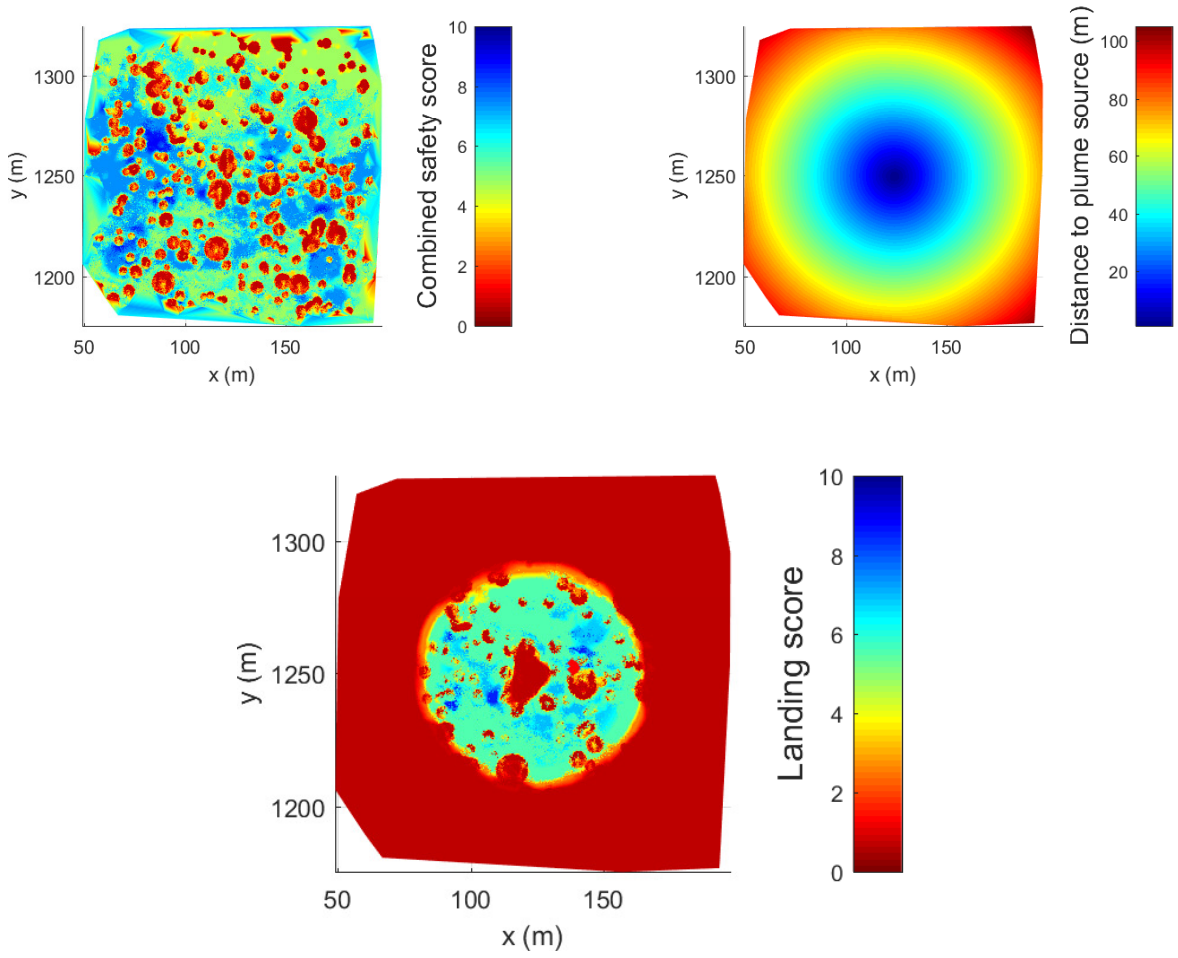

Figure 4.32: Fused safety score for all sensors (upper, left), distance from plume (upper, right), and final landing goodness score, with the selected best new landing site marked as a red x (lower, x is distinguishable slightly to the right of the center of the map).

preferred to select a threshold that maximizes the percentage of true measurements, therefore a value of 6.1 is selected. The resulting HDA and ground truth binary maps are compared with the results also shown in Figure 4.33. A significant agreement is noted between the two maps, with some false positives spread near lower slope part of the map, and some false negatives on the canyon slope. Very small areas of false negatives and positives can be seen in thin lines around the obstacles. False negatives are mostly present in high slope areas. It is assumed that this is due to the high scoring of these areas by the optical camera. On the contrary, false positives tend to be in lower slope areas, where the hazard landscape is more complex. This is because the lidar scores the low slope and roughness parts of larger obstacles as relatively safe, while both the camera and the ground truth map give them an "absolutely unsafe" marking.

The above FN, FP, and TM can be considered as the HDA performance metrics, as they readily describe the performance of the HDA process in identifying suitable and unsuitable areas for landing. In the above process, the metrics were in a sense "calibrated" to their optimal values for the ideal case, where there are no noises in the sensors, and the full resolution map is analyzed. These performance metrics for the ideal case can be seen in Table 4.7.

Table 4.7: HDA metrics for the ideal HDA measurement, for a threshold score of 6.1 (see discussion in text and Figure 4.33). These can be considered the optimal values for the HDA metrics (usefull for further analysis in this chapter).

HDA perf. metric	**Value**
False Negatives	8%
False Positives	7%
True Measurements	85%
Terrain suitable for landing	51%

It should be made clear that the values for the above performance metrics do not correspond to the likelihood of successful landing or any other straightforward mission success metric, but only imperfectly represent the agreement between the ground truth and the HDA maps. These metrics are to be used for comparison between different HDA measurements and will therefore be useful in the analysis below. The successful landing probability is expected to be significantly higher than the true measurement metric for the above ideal case, since the re-targeting function will not choose a random point with a score above the threshold, but the point in the map with the highest score. The actual mission success statistics will be calculated by Monte Carlo analysis later in this chapter (Section 4.4).

In summary, the main conclusions from the above validation analysis and calibration are:

a) any point with a fuzzy safety score above 6.1 can be nominally considered as safe for landing, and

b) the HDA process correctly assesses a large percentage of the ground truth map for safety and can be reliably used in further analysis.

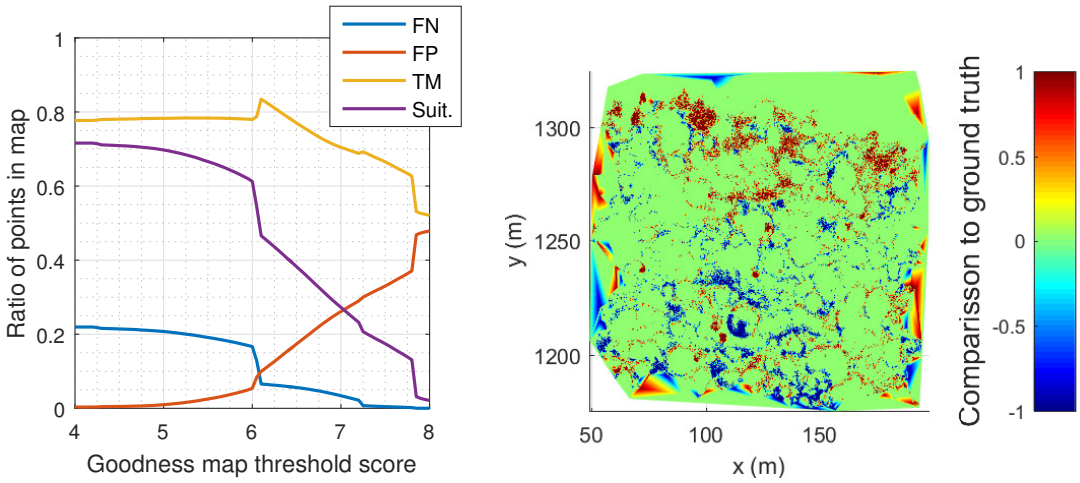

Figure 4.33: The HDA goodness threshold calibration plot (left, see text for further explanation), and the ground truth-HDA goodness comparison map for a threshold value set at 6.1 (right). Red (1) denotes false negatives, green (0) true measurements and blue (-1) false positives.

4.2.3 Sensitivity analyses

The effects of various parameters on the goodness of the results of the HDA function will be investigated. This is done by comparing the HDA maps generated from each test run with the original high resolution HDA map of the test terrain in the previous section (Figure 4.32, upper left)[23]. Where necessary for a direct comparison to be possible, the HDA maps from the test runs are interpolated to the high resolution of the original HDA map.

Table 4.8: Varying parameters in HDA sensitivity analysis.

System and error parameters	
Terrain model GSD/resolution (m)	$[0.1, 0.25, 0.5, 1, 2]$ m
Lidar range/elevation noise (1σ)	$[0, 0.1, 0.5, 1, 2]$ m
Lidar, probability of texture flip	$[0, 5, 10, 20, 50]\%$
Optical camera blurring (rolling window size)	$[0, 1, 2, 5, 10]$ m
Thermal camera, heat source assumed spread (1σ)	$[10, 20, 30, 50, 100]$ m
Thermal camera, assumed plume temperature	$[160, 170, 180, 190, 200]$ K
Thermal camera, plume source assumed location error	$[1, 5, 10, 20, 50]$ m

The first such sensitivity analysis is in respect to map GSD/resolution. For a given sensor with a certain angular resolution, this value depends only on the range to target that the measurement was taken (Figure 2.17). The effect of the measurement slant

[23]...and NOT with the binary goodness map as with the validation case.

angle[24] can be ignored here due to the small FOV of the lidar, and the relatively small slant angles for landing operations.

To investigate the dependence of HDA performance (as captured by the HDA metrics discussed in the previous section) on map resolution, the terrain map was down-sampled to various GSD/resolutions. The HDA process is then applied to each of these maps and the resulting safety map is interpolated. The interpolated map is subtracted from the nominal, high resolution safety map. This results in a new continuous comparison map (in contrast to the binary comparison map in the previous section) with scores ranging from -10 to $+10$. Negative values on this new comparison map indicate therefore an underestimation of hazardousness by the HDA function, and positive values an overestimation of hazardousness. Accordingly, a given map point is assigned as a true measurement if its score on the comparison map is between -1 and 1, as a false negative if its score is < -1, and as a true positive if its score is > 1. All this will become clearer during the presentation of the analyses in this section.

The sensitivity curves for the above resolution analysis are shown in Figure 4.34. As expected, larger ground sample distances will give a less accurate HDA map. For the nominal resolution of 0.1 m, the TM value is around 80%, and both FN and FP, are at around 10% each. The fact that, even for the nominal resolution, there is no perfect agreement between the ground truth safety map and the test map, can be attributed to the interpolation step that was performed on the map (essentially to interpolate it to its current resolution). By performing this step also on the nominal map, the effects of interpolation can be accounted for on the comparison process. The values of the performance metrics remain at those levels up to a map GSD of 0.5 m, which is the maximum acceptable obstacle size for the lander. For GSD above this, the HDA process starts "missing" hazards and therefore a small increase in FP, and dramatic increase in FN are observed.

A lower resolution map can also result from the non-nominal case, where some lidar beam return signals are not detected by the lidar head sensor. Assuming that the distribution of the beams whose signal is not returned is random, this results in a degradation in map resolution. For the extreme scenario where half of the beam return signals are not detected, this degradation is equivalent to the performance being downgraded to that of a higher GSD class. This effect will have significant consequences at farther measurement ranges where the sensitivity of performance to GSD is high, and less so for nearer ranges, where the performance curves of Figure 4.34 are linear.

Due to the high computational requirements of the HDA process, and to reduce the time needed for the following analyses to within reasonable levels, the 0.25 m GSD interpolated map was used as a basis for the rest of the sensitivity analyses.

Next, the sensitivity of the HDA assessment to the input of each of the sensors will be investigated. As lidar is essential for producing the point cloud basis for the HDA map and in detecting the majority of hazard characteristics, two cases will be investigated here: the absence of an optical camera, and the absence of any texture sensing by the lidar and the thermal camera. The comparison maps are seen in Figure 4.35. As noted before, the optical camera considers all non-obstacle and non-optically-rough areas

[24] Angle between horizon and lander-target line

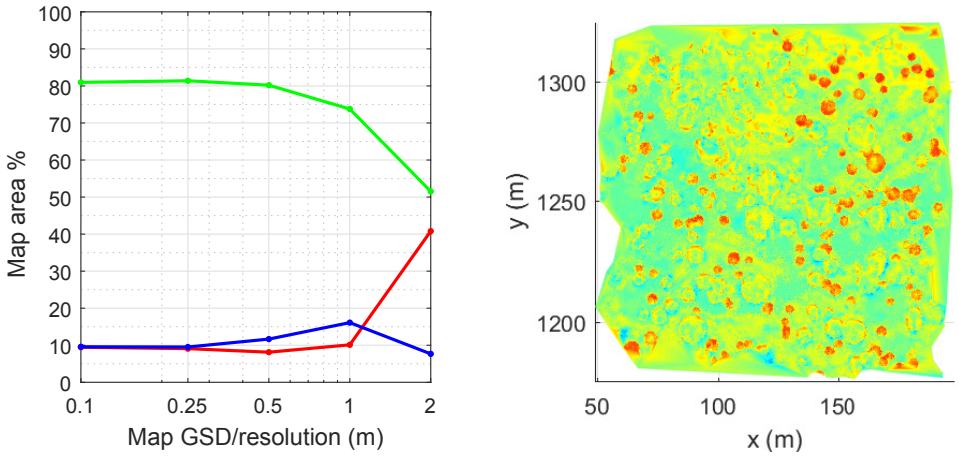

Figure 4.34: HDA sensitivity curves for a varying GSD/resolution of the input terrain model (left, x-scale is logarithmic, green line: TN, red line:FN, blue line: FP). The comparison map is given for the 2 m GSD case for illustration (right). On the comparison map, redder areas denote increasing hazard underestimation and bluer areas denote increasing hazard overestimation by the HDA function. The trend for smaller resolutions is for the hazard underestimation areas (yellow) to diminish.

as highly safe. It thus overestimates the safety of these areas that may include e.g. high slopes. Without the optical camera, the safety map becomes very similar to the lidar safety map in Figure 4.31. It can be argued that this map better represents the hazardousness of the terrain model. However, with no optical camera input, the redundancy in obstacle detection is lost. If the texture information fused together by the lidar and thermal camera is ignored, the safety of snow-covered areas is overestimated. Depending on the sensitivity of the terminal descent method used and the general lander design to landing on snow, this could be an acceptable choice to make.

The effect of noise in the elevation of values of the terrain model points will be then investigated. The main source of noise in elevation is lidar range noise. As the noise is multiplicative, measurements taken at larger ranges from target will produce higher elevation noises. In particular, for a lidar range from target at HDA HG of 1100 m, and the nominal multiplicative range noise value of 0.33%, an elevation noise of ~ 3.5 m 1σ is given, significantly higher than the size of the minimum acceptable terrain roughness (0.5 m).

Figure 4.36 illustrates that effect. A sharp drop in HDA performance is observed when deviating from the ideal, no-noise case. "Spikes" produced in the map due to increasing noise in elevation are interpreted by the HDA process as roughness hazards, thus increasing the FP significantly. FN remain relatively stable.

The effect of the uncertain nature of sensing the texture of the terrain with lidar is investigated next. To do this, a probability is assigned, that the texture of each point in the terrain model will flip (from snow to ice or vice versa). This probability is varied and the effects in HDA performance are presented in Figure 4.37. There it is seen, that even for the worst case where the lidar provides fully noised texture measurements, only

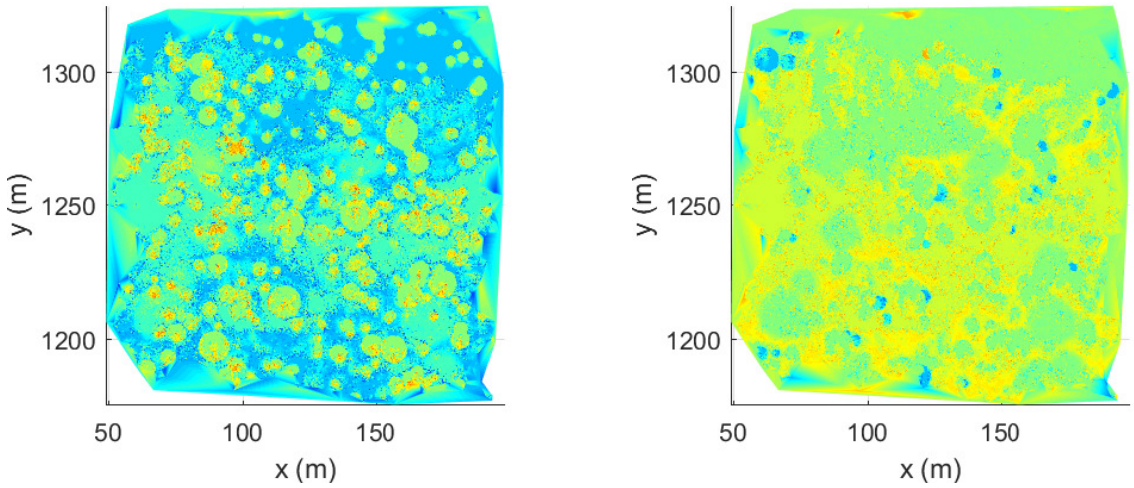

Figure 4.35: Comparison maps for HDA without opt. camera input (left), and without terrain texture input (right). Redder areas denote increasing hazard underestimation and bluer areas denote increasing hazard overestimation by the HDA function.

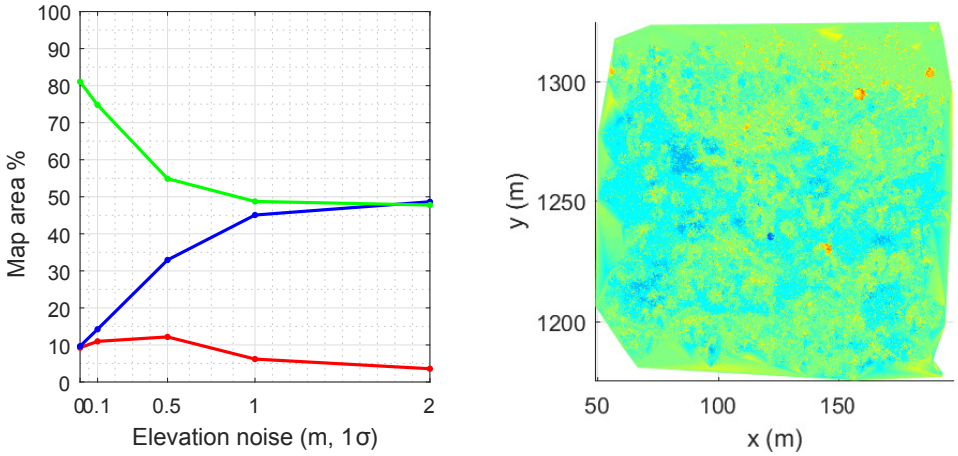

Figure 4.36: HDA sensitivity curves for a varying lidar range/elevation noise of the input terrain model (left, x-scale is linear, green line: TN, red line: FN, blue line: FP). The comparison map is given for the 1 m (1σ) case for illustration (right). On the comparison map, redder areas denote increasing hazard underestimation and bluer areas denote increasing hazard overestimation by the HDA function. The trend for lower noise levels is for the hazard overestimation areas (blue) to diminish.

a small effect on the overall safety score is noted, whereby the terrain safety is overestimated for areas covered in snow. This is thanks to the redundant texture measurements by the thermal camera, and the small contribution of texture to the total safety score.

Next, the effect of blurring in optical camera images is investigated. Blurring can occur for example when a picture is taken at high rotational velocities. To simulate optical image blurring, the nominal illumination layer was averaged using a rolling window with increasing sizes (whereby the illumination of a given point is assigned as the mean of the values in the neighborhood defined by the rolling window). The blurring effect is illustrated in Figure 4.38.

The sensitivity curves for varying blur levels in the illumination layer are seen in Figure 4.39. Due to image blur, smaller obstacles go undetected by the optical camera. Increasing blur levels increases the size of detectable obstacles until the obstacle detection process becomes useless. Accordingly, TM drop and FN increase in the sensitivity curves. It should be noted here that obstacles are still detected by the lidar as roughness and slope hazards. With the degradation of the optical obstacle detection though, obstacles are not scored as low as in the ground truth safety map.

The effects of uncertainty in the knowledge of the surface temperature related parameters used for the derivation of the texture by the thermal camera will be investigated next.

Starting with the assumed plume source hot spot spread, the HDA process was ran using various values around the nominal of 30 m, and the performance of the HDA function was plotted in the sensitivity plot of Figure 4.40. The effect of this mismatch between the assumed and the real spread of the hot spot around the plume source is that the corrected temperature after subtracting the assumed background temperature is either over, or underestimated, resulting in a ”ring” of falsely identified texture, with the size of the ring depending on the deviation of the assumed spread from the real one. This results in the false designation of the ring as snow or ice respectively.

The next parameter investigated was the assumed plume temperature, varying it also around the real value of 180 K. The results are seen in Figure 4.41 The effect is the creation of disks around the plume source where texture is falsely estimated as ice for assumed temperatures less than the real, and as snow for assumed temperatures larger than the real.

Finally, the effect of the uncertainty in the location of the plume source on HDA performance is investigated. To do this, the assumed plume source location was moved to the right on the x-axis by increasing values. As can be seen in Figure 4.42, an increasing error in assumed plume source location location results in two ”lobes” of increasing sizes, with the ahead lobe overestimating snow cover (as it gives higher surface temperatures) and the back lobe underestimating it.

The overall effect for the three parameters relating to the thermal camera texture detection, seen by the sensitivity curves in the above mentioned figures could be interpreted as small. As described though, there are strongly localized effects affecting the texture sensing of large areas on the map. In any case, the redundancy offered by the lidar texture sensing means that any erroneous sensing by the thermal camera is muted in the overall safety score maps.

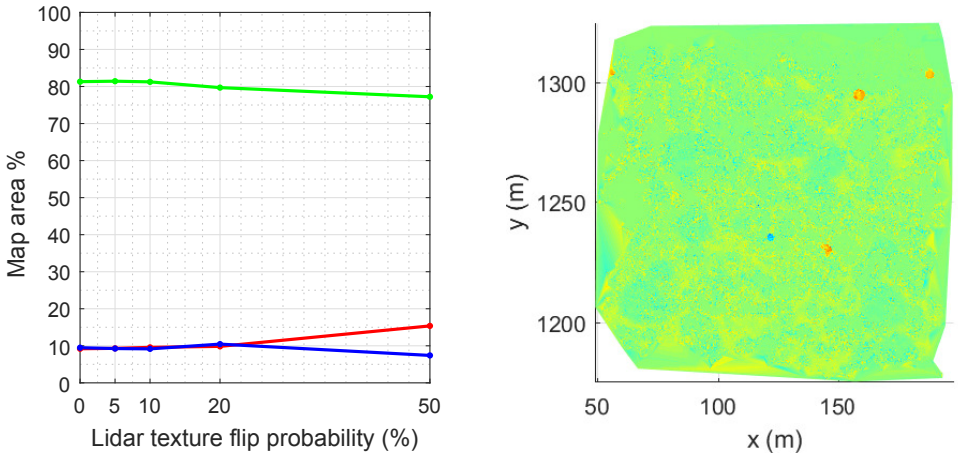

Figure 4.37: HDA sensitivity curves for a varying probability that a texture measurement of any point on the map will be flipped (left, x-scale is linear, green line: TN, red line: FN, blue line: FP). The comparison map is given for the 50% (1σ) case for illustration (right). On the comparison map, redder areas denote increasing hazard underestimation and bluer areas denote increasing hazard overestimation by the HDA function. The trend for lower noise levels is for the hazard underestimation areas (light yellow) to diminish.

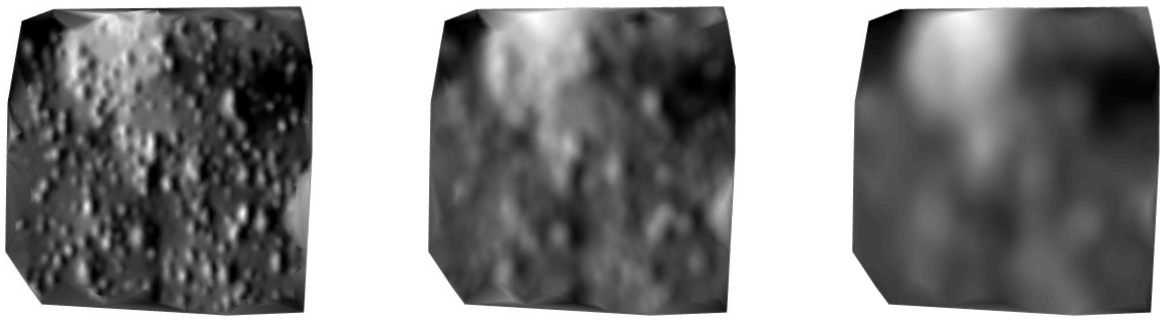

Figure 4.38: Illustration of the blurred illumination layer used for the blur sensitivity analysis, for rolling window sizes of 1, 2, and 5 m (left to right).

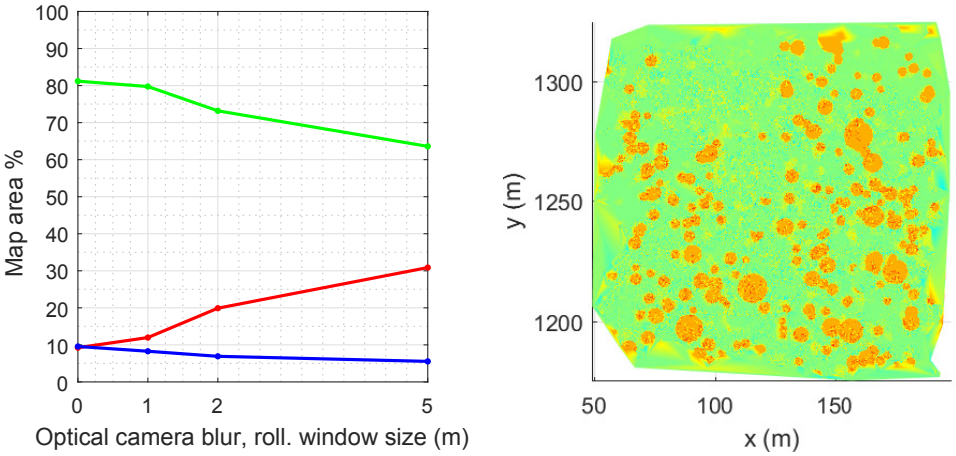

Figure 4.39: HDA sensitivity curves for a varying levels of optical image blurring, expressed in the size of an averaging rolling window, see text for further explanation (left, x-scale is linear, green line: TN, red line: FN, blue line: FP). The comparison map is given for the 5 m case for illustration (right). On the comparison map, redder areas denote increasing hazard underestimation and bluer areas denote increasing hazard overestimation by the HDA function. The trend for lower blur levels is for more and smaller obstacles to be identifiable.

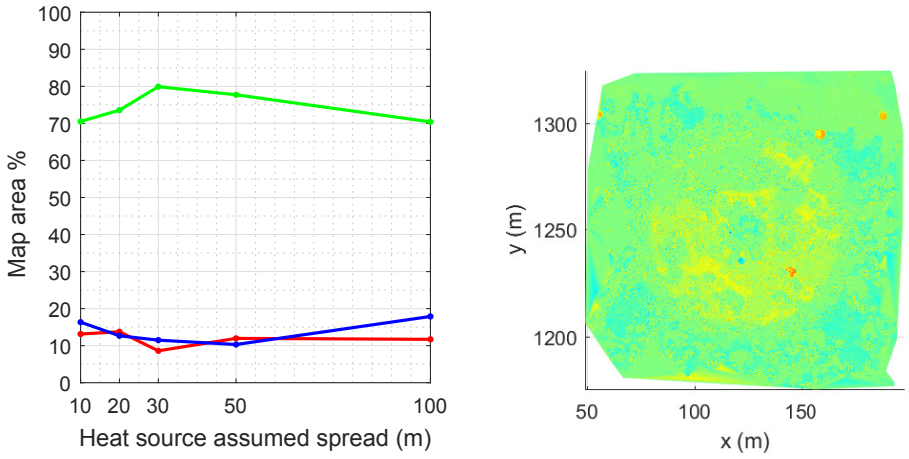

Figure 4.40: HDA sensitivity curves for a varying levels of the assumed spread of the hot spot around the plume source, (left, x-scale is linear, green line: TN, red line: FN, blue line: FP). The comparison map is given for the 10 m case for illustration (right). On the comparison map, redder areas denote increasing hazard underestimation and bluer areas denote increasing hazard overestimation by the HDA function. The effect of varying assumed spread around the nominal of 30 m, is for a ring of falsely sensed texture to appear.

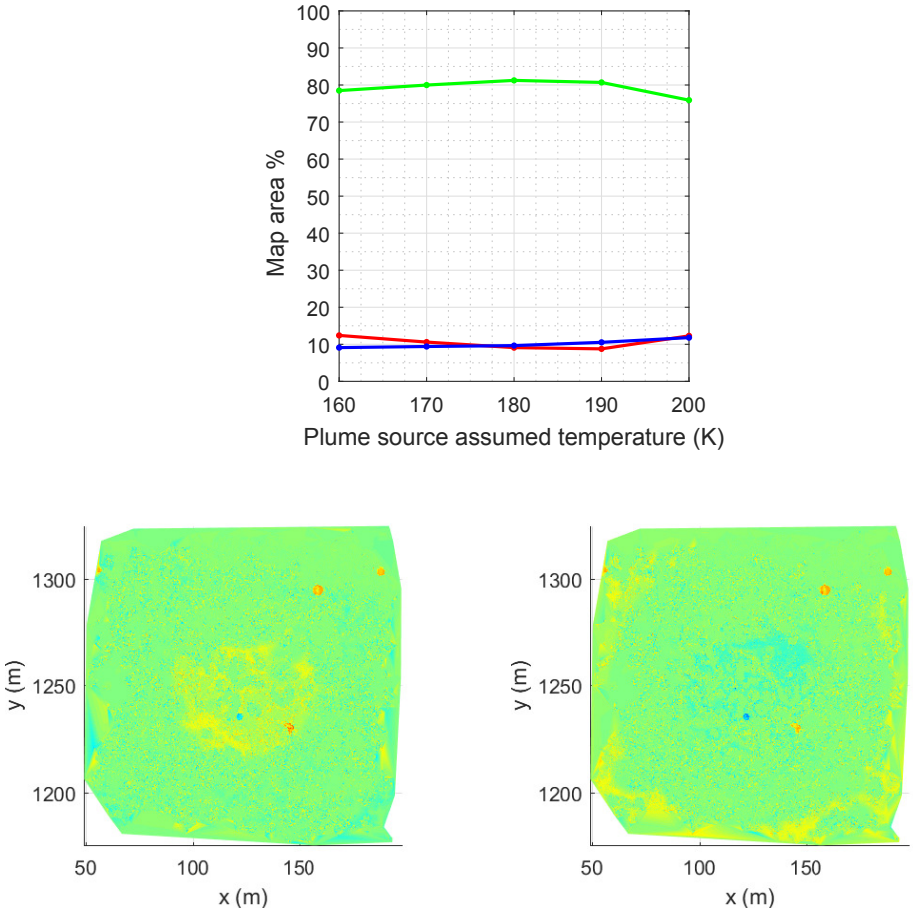

Figure 4.41: HDA sensitivity curves for a varying levels of the assumed plume source temperature (left, x-scale is linear, green line: TN, red line: FN, blue line: FP). The two comparison maps are given for the 160 and 200 K cases respectively for illustration (right). On the comparison map, redder areas denote increasing hazard underestimation and bluer areas denote increasing hazard overestimation by the HDA function. The trend is for lower assumed temperatures to produce a disk of mis-detected ice around the plume source, and for higher assumed temperatures, a disk of mis-detected snow.

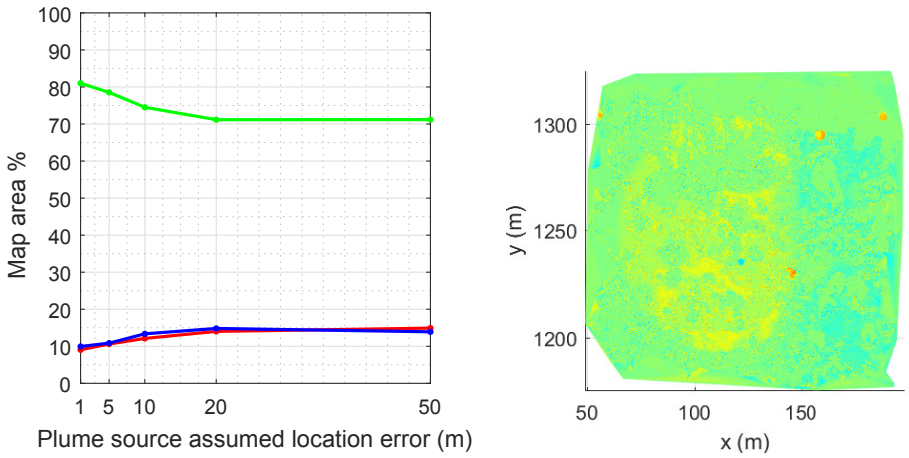

Figure 4.42: HDA sensitivity curves for a varying levels of error in the assumed plume source position (left, x-scale is linear, green line: TN, red line: FN, blue line: FP). The displacement here takes place on the x-axis. The comparison map is given for the 20 m case for illustration (right). On the comparison map, redder areas denote increasing hazard underestimation and bluer areas denote increasing hazard overestimation by the HDA function. The trend for increasing position errors is for the lobes of mis-detected texture to expand.

4.2.4 Simulation of nominal operations

After the sensitivity analyses above, nominal values where set for all parameters investigated above, based on reasonable assumptions for them (Table 4.9). Mainly, the lidar range error had to be reduced to 0.1× the nominal, so that reasonable roughness values are produced in early measurements. Using these values and based on the operational concept defined in Section 2.3.4, the nominal landing operations where simulated to further investigate the HDA function, with the lander following the reference landing trajectory given in Section 4.3 while assuming ideal navigation.

Four HDA measurements where made at the distances and timings given in Table 4.9. These measurements produced terrain models that where then processed by the HDA function to produce safety and landing score maps, and to select an new target for landing. This process is illustrated in Figure 4.43. Each panel of the figure illustrates the safety map generated by each HDA measurement. On each safety map, a circle is placed, representing the lander footprint centered around the new landing target selected by the re-targeting function (reminder: selected as the best score in the landing score map). Finally, a black square illustrates the projected FOV of the following HDA measurement, centered around the new landing target. As expected, earlier HDA measurements cover a larger area, but produce low safety scores mainly due to the high elevation noise. Scores increasingly grow for each measurement as the lander comes nearer to the surface. The lander footprint increasingly becomes a larger part of an HDA measurement map. A final landing site is chosen in the final HDA measurement.

Table 4.9: Nominal HDA parameters

Parameter	Value
Ranges to target	$[1100, 550, 200, 100]$ m
Time of measurement (after AG)	$[24, 45.2, 61.1, 75.2]$ s
Lidar range error	0.033% (1σ)
Lidar texture flip probability	10%
Lidar no beam return probability	5%
Optical camera blur	none
Thermal cam., hot spot assumed spread error	5 m (1σ)
Thermal cam., plume temp. uncertainty	5 K (1σ)
Thermal cam., plume source location uncertainty	5 m (1σ)

A closer look can also be taken at final steps of the HDA process, comprising the temporal and spatial assessment. These steps are illustrated in Figure 4.44, using the second HDA measurement of Figure 4.43 as basis. It can be seen there how the spatial and temporal steps further refine the overall landing score, assessing the goodness of neighboring points and the goodness of the same points in the previous measurement respectively. A new landing target is selected by picking the point in the map with the higher landing score in the final landing goodness map.

The evolution of HDA performance throughout landing operations is shown in Figure 4.45. As expected for earlier measurements, the FPs generated by false roughness measurements due to high noises in elevation dominate the performance. As the lander gets closer to the sensed surface, the ratio of TMs increase at the expense of FPs, and peaks at about 70%. FNs remain at the same levels throughout operations, at about 10%.

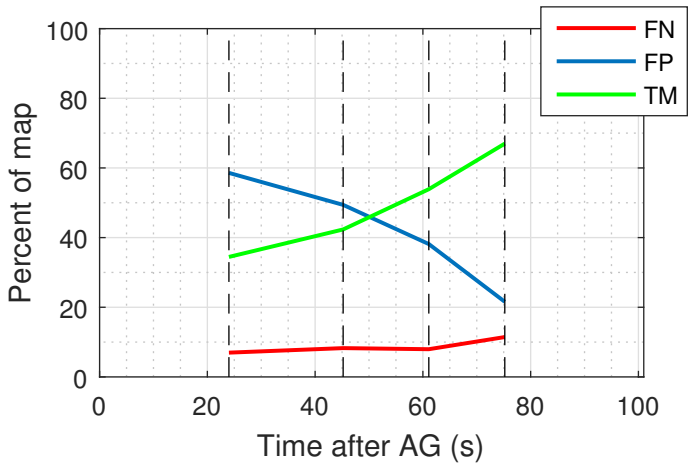

Figure 4.45: Evolution of HDA performance throughout the reference landing operations. Time is measured after AG. The times of HDA measurments are marked with dotted vertical lines.

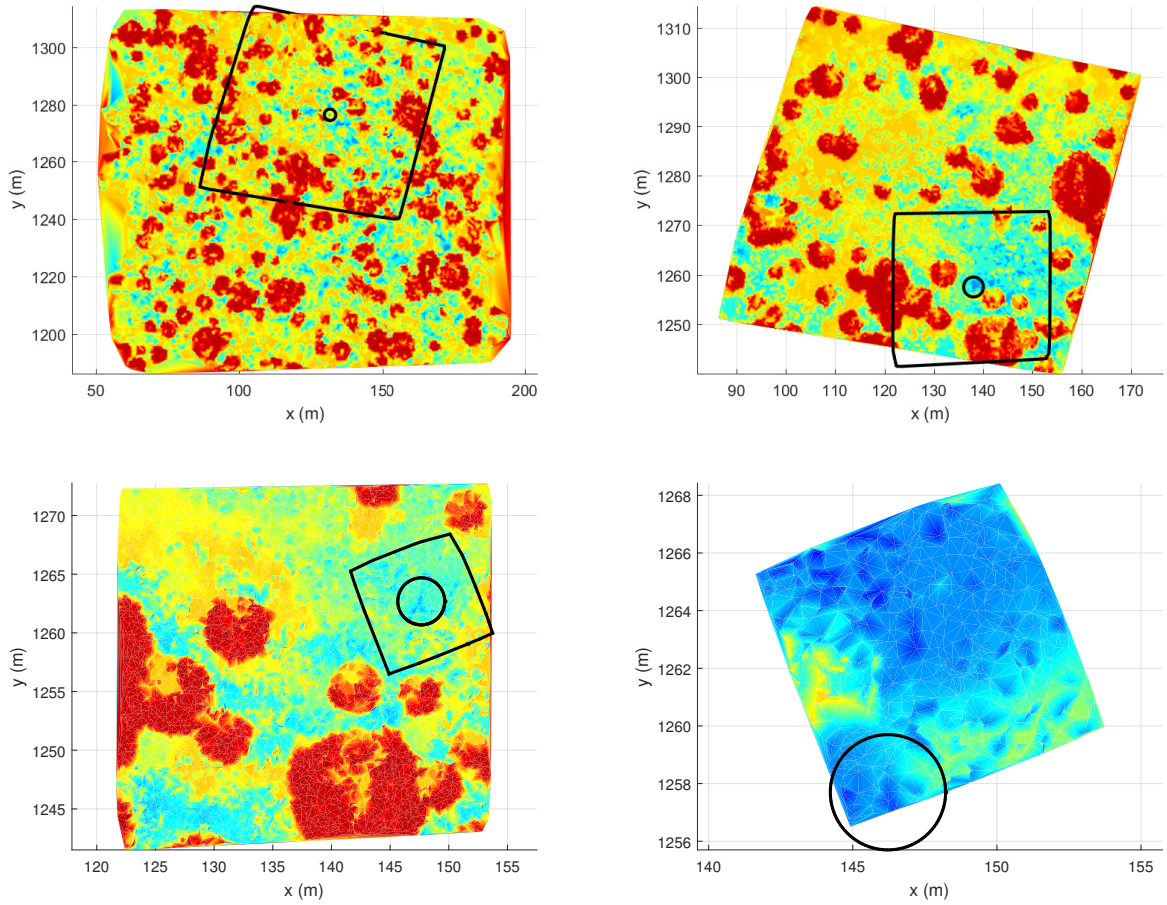

Figure 4.43: The four consecutive HDA measurements for the reference landing scenario. See text for details. The colorscale of these plots is omitted for compactness and is identical to that of Figure 4.32 (i.e. red is low and blue is high safety score).

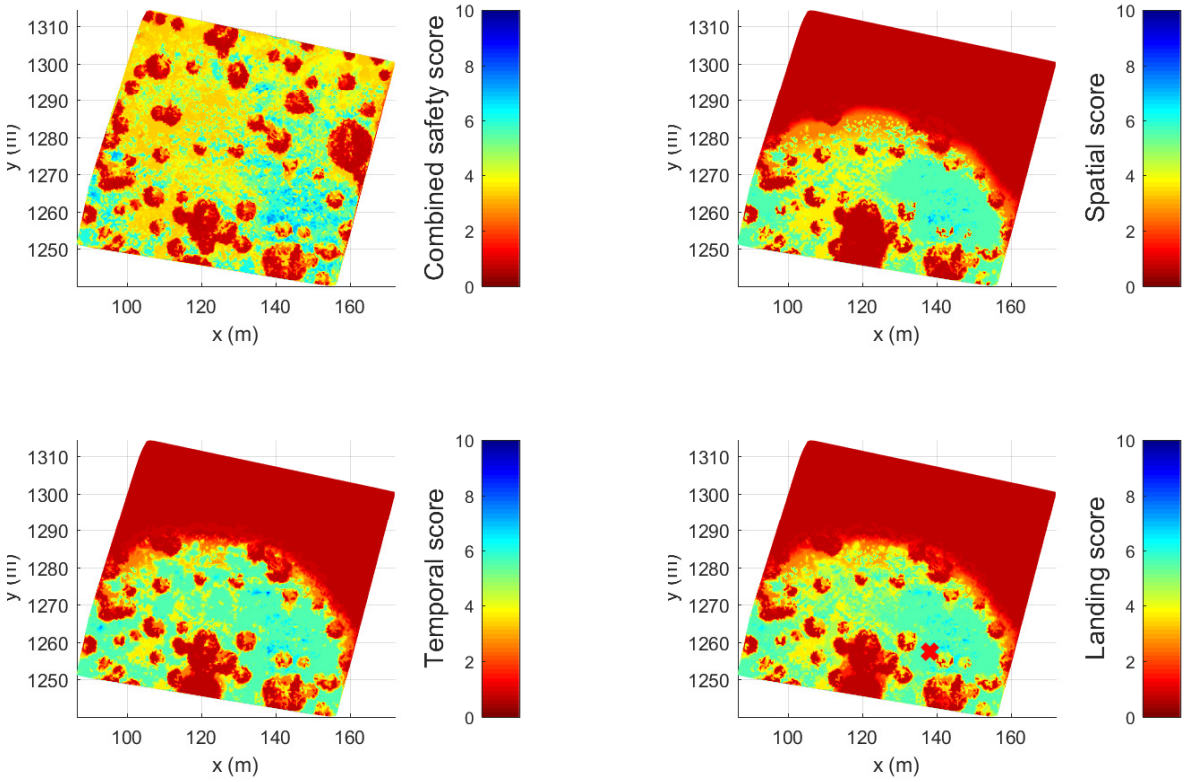

Figure 4.44: Demonstration of the final steps of the HDA process. The combined safety score map (upper left) goes through the spatial (upper right) and temporal (lower left) evaluation steps. The final landing score map is given in the lower right, with the highest scoring point marked with a red x.

4.2.5 Results summary and initial conclusions

We can now gather informative observations on the above results, that will be helpful with the next design iteration of the HDA concept. In the following significant conclusions will be summarized:

- Lidar is central for HDA and, in ideal conditions cover the entire range of hazards to be detected. It is also equally operable in darkness. However the optical and thermal cameras offer robustness in detecting obstacles, and terrain texture respectively. The first is more critical to mission success, while the usefulness of the second depends on the chosen strategy for touchdown, and how sensitive the lander design is to landing on snow.

- The range noise in the lidar beams should be reduced significantly, to levels of 10% of those taken from the literature, if meaningful hazard assessments are to be made with the early HDA measurements. Ways to achieve that (e.g. merging of succesive rapid lidar measurements) should be investigated in the future.

- The optical camera hazard map does not consider slope. Past safe and accurate landing missions targeted mostly terrains that are flat in the large scale, but this is not the case for EnEx. The hazardousness of a large part of the terrain is the underestimated by the camera. To correct this, the camera weight during the score combination of Equation 2.56 should be significantly lowered

- The method for texture sensing using the thermal camera is a good compensating measurement for the lidar, but it is not expected to be robust as a stand-alone method in its current form. However, it is a novel concept worth a further look in the future.

- Considering the example HDA run shown in Figure 4.43, a potentially hazardous scenario comes to mind: as the first HDA measurement has the lowest performance, it is conceivable for it to command a re-targeting to landing site that is not suitable, that is also in an area that is also unsuitable for landing. As the sensor suite will turn on its gimbal to point to that new point, future HDA measurements will be increasingly restrained to the area near the original unsuitable area. This means that the HDA function will be forced to select the "least of all evils" among a group of unsuitable points. Potential ways to deal with this is to further lower the range at which this first HDA measurement is taken, by setting a stricter requirement for the range error of the lidar, or by modifying the HDA operations concept, to search for a new target outside its current FOV if there is no point with a goodness score above a given threshold in its current landing goodness map. Each of these solutions comes with their challenges and advantages that should be investigated in the future.

- In this approach only basic scalar factors where used for sensor certainty. However, a more sophisticated approach would include fuzzy certainty factors taking into account range, illumination, lander navigation parameters etc. (as is done e.g. in [75]). The simplicity of the first approach should be compared with the robustness of the second.

4.3 Guidance and propulsion

In this section, the results of the guidance and propulsion relevant analyses performed with the LandingSim tool will be presented.

4.3.1 Description of elements

The characteristics of the EnEx lander and its propulsion system are given in Section 2.5, a description of guidance considerations in Section 2.4.3, and an operations concept for guidance in Section 2.6. In summary, the guidance function must create a landing trajectory from a given starting point to a given target point, while respecting glide slope, thrust magnitude and angle, and velocity magnitude constraints. The calculated control thrusts are then applied using the lander propulsion system. A certain amount of propellant has been assigned for the approach phase of the landing. It is reminded that it is assumed that the propulsion system thrusters apply the thrusts commanded by guidance ideally.

To further analyse the guidance and propulsion system, the guidance problem was simulated.

4.3.2 Validation and demonstation

Two landing scenarios were defined as baseline. Scenario 1 has been used as the reference trajectory in Sections 4.1 and 4.2 for the analysis of the navigation and HDA functions performed there. The initial and final conditions for both trajectories are given in Table 4.10.

Table 4.10: Initial and final conditions of reference trajectory used in the other simulation results sections and challengin scenario used for guidance function validation in this section. For guidance validation a generic inertial reference frame is assumed.

Parameter	Scenario 1 (nominal)	Scenario 2 (challenging)
Initial position	$[0, -1000, 1500]$ m	$[0, -1100, 1500]$ m
Initial velocity	$[0, 20, -20]$ m/s	$[20, -5, -15]$ m/s
Final position	$[0, 0, 0]$ m	$[0, 0, 0]$ m
Final velocity	$[0, 0, 0]$ m/s	$[0, 0, 0]$ m

The G-FOLD implementation in this work was validated against [79] for a Mars landing scenario, with the results of the validation given in [108]. Here, a landing with the initial conditions of Scenario 2 was simulated to demonstrate the performance of the G-FOLD guidance algorithm. These initial conditions where manually calibrated so that the landing target is barely reachable. A generic inertial frame is used for the purposes of this validation.

Figure 4.46 shows the calculated trajectory and thrust arcs for Scenario 2, and the projections of that trajectory on the z, y, and x planes respectively. A smooth trajectory is produced for this challenging scenario that respects the glideslope constraint (red dotted lines in the projection plots).

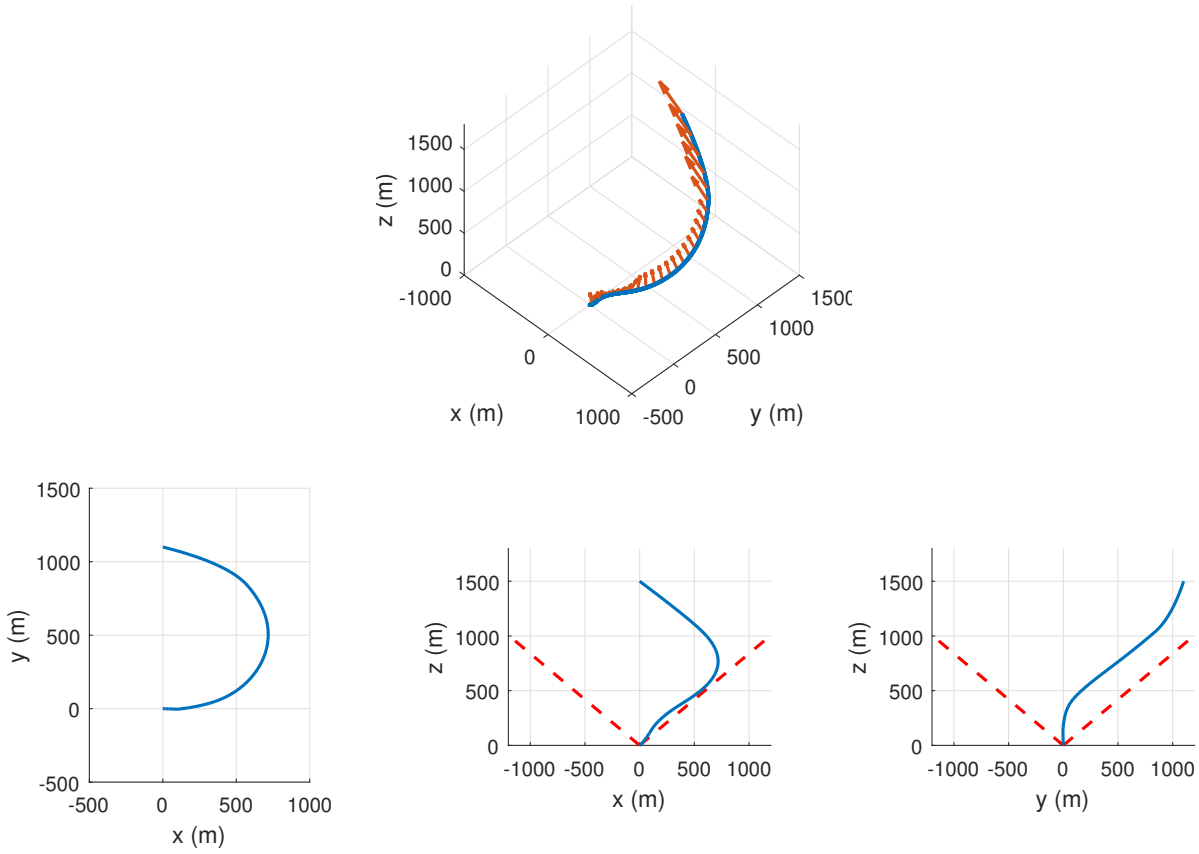

Figure 4.46: Demonstration of the trajectory calculated by G-FOLD for Scenario 2. The trajectory and thrust arcs (not in scale) are shown in 3D (upper), and the projections of the trajectory are shown on the z, the y, and the x planes (lower, left to right). The projected glideslope constraint cone is shown as red dotted lines.

Figure 4.47 shows the evolution of lander mass, the elevation angle of the line connecting the landing target and the lander, the velocity magnitude, thruster duty, and thrust angle. As seen, the lander expends around 60 kg of fuel, respects the glideslope and thrust angle constraint angles (black dotted lines in the respective plots). For the optimizer to converge easier to a solution, the velocity magnitude constraint (plotted vs range to landing target) had to be somewhat relaxed. The result is seen in the respective plot of Figure 4.47, where the velocity magnitude is allowed to be higher than the constraint. Perhaps due to this slight relaxation, the thrust magnitude also slightly violates the lower magnitude limits for a small part of the trajectory. This small violation is considered acceptable, also considering the extreme nature of scenario 2.

Figure 4.48 shows the evolution of lander position, velocity, and acceleration acting on the lander. They all evolve smoothly, except for sudden jumps in acceleration when the thrusters are turned to maximum or minimum thrust.

4.3.3 Sensitivity and worst-case analyses

Sensitivity analyses where performed for the guidance function. The guidance function system must be flexible in two senses: it must be able to reach the nominal landing target from varying initial lander state values, and also to reach any landing site commanded by the HDA re-targeting function. It must do so while minimizing the amount of fuel expended. The sensitivity of guidance to initial lander state conditions, and re-target point distance from nominal will be thus checked. The performance metrics will be the amount of propellant spent, and the distance from the nominal spot. The varying values are given in Table 4.11. Each parameter was varied separately.

The results are shown in Figure 4.49. It is seen that the guidance function can easily deal with highly variable initial and final conditions. The expenditure of propellant varies between $\sim$ 45 kg for the close-to-nominal cases, and $\sim$ 65 kg for the more extreme cases.

The above reassuring results are valid for the first guidance trajectory calculated at AG. The next trajectory to a newly selected point after the first HDA measurement will have to be calculated as the lander is already on this first highly optimized trajectory. The possible problem then for future re-targetings is then that the lander is no-longer

Table 4.11: Varying parameters in guidance sensitivity analysis. The parameters are varied between the minimum and maximum values for which guidance can produce a trajectory are given.

Initial and final lander state param.	
Final landing position, along-range	-1000 - 700 m
Final landing position, cross-range	-1200 - 1200 m
Initial landing position, along-range	-700 - 1000 m
Initial landing position, cross-range	-1200 - 1200 m
Initial lander velocity direction, rotation around altitude axis (left-right)	$-45°$ - 45°
Initial lander velocity direction, rotation around cross-range (up-down)	$-25°$ - 20°

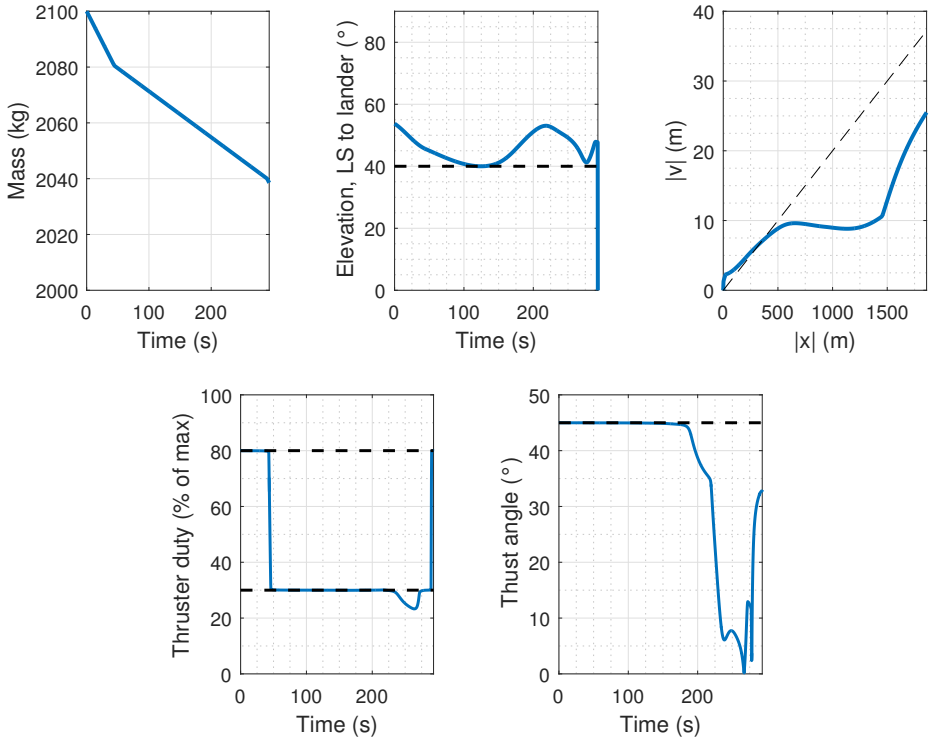

Figure 4.47: Evolution of lander mass, elevation angle between landing target and lander, velocity magnitude (upper, left to right), thrust magnitude/duty, and angle (lower, left to right). All parameters respect the set constraints (black dotted lines).

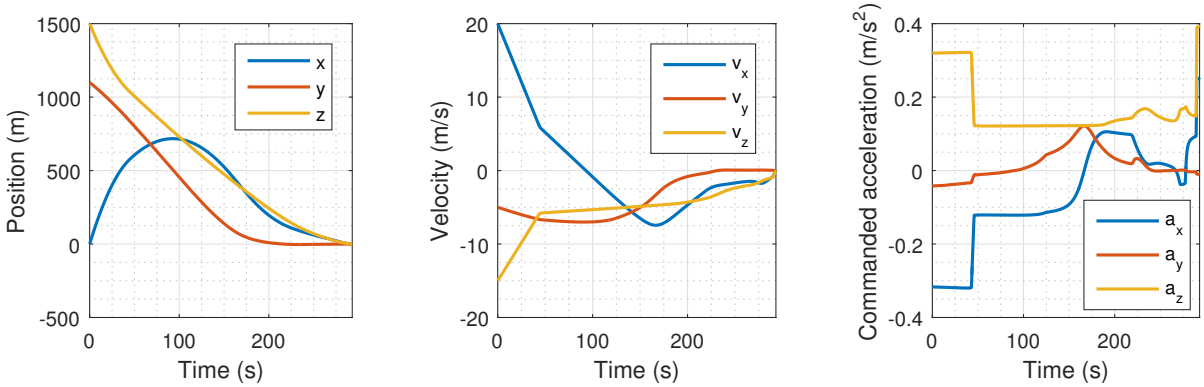

Figure 4.48: Time profiles of landing parameters for landing Scenario 2.

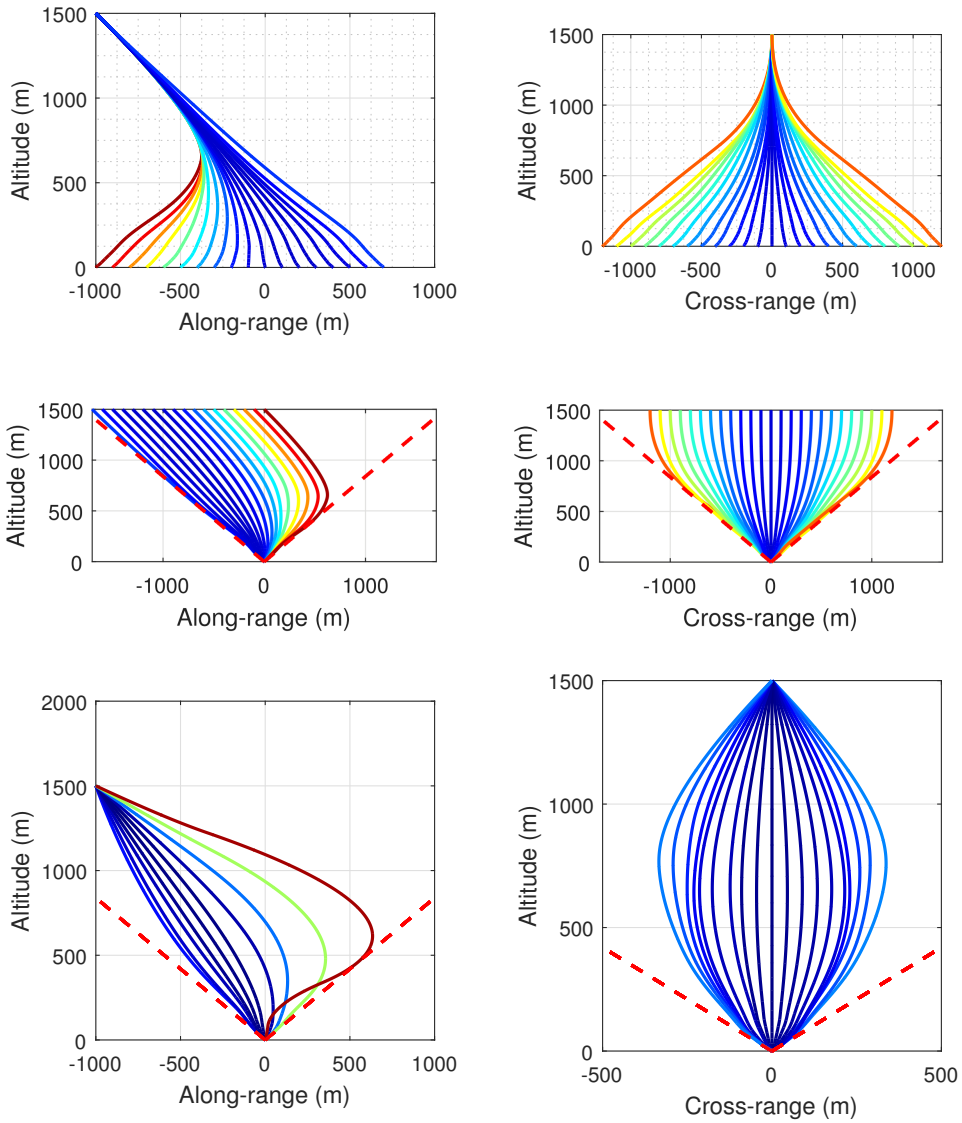

Figure 4.49: Sensitivity of guidance at AG to final landing site position along- and cross-range (upper, left, to right), initial lander position along- and cross-range (left, to right), and initial lander velocity rotated around the cross-range and the altitude axes (left, to right). Line color corresponds to fuel usage (45kg dark blue to 65 kg dark red).

flexible, as its already on an optimal trajectory to another landing site. In in other words, the lander is speeding towards a given landing site, with minimal flexibility and "left-over thrust" to divert to another.

A two-sided solution to the above problem was devised. Firstly, and as seen by the sensitivity analyses above, the guidance function somewhat "over-performs" for reachability at AG. A limit can then be set on the available thrust at this first, and slowly relaxed for the following maneuvers. Essentially this means that the engine will be "throttled-up" after each consecutive re-targeting. This relaxation was found to be insufficient after a few simulations and a second aspect to the approach was added, the gradual relaxation of constraints. Of the constraints given for guidance, arguably the more amenable to relaxation due to it being the least critical, is the velocity magnitude constraint. By setting a relaxation parameter > 0 for the respective constraint inequality in the guidance equations, and gradually increasing it after each maneuver, the necessary flexibility can be achieved by the lander.

To demonstrate and validate this approach a worst case analysis was performed. In this analysis the lander was initialized at its nominal nav. state and a trajectory to the nominal landing site was calculated by guidance. At the first re-targeting a new target was set at a worst case distance from the original. The reasonable worst-case assumption for re-targeting distance of $2\times$ the ground cover area (GCA) of the respective HDA measurement is made (see e.g. Figures 2.17 and 4.43). This is done for all four re-targeting maneuvers, for both the along- and the cross-range cases, with each new landing target being moved the appropriate distance to the opposite direction, resulting in a zig-zag overall landing trajectory. To allow for the needed increasing guidance flexibility, the thrust upper limit and velocity magnitude are gradually relaxed after each re-targeting maneuver. Due to their sensitive nature, the relaxations were performed manually during the analysis. In the future, a more formal and automated way will have to be found.

The results of this worst-case analysis are seen in Figures 4.50 and 4.51. The guidance trajectory successfully performs a re-targeting at every step. The gradual relaxation of thrust magnitude and velocity constraint is also visible in the corresponding plots after each re-targeting.

4.3.4 Results summary and initial conclusions

Information and conclusions from the above analysis can now be summarized here:

- The G-FOLD algorithm implemented is very robust and finds viable large divert trajectories for a very broad range of lander initial conditions and re-target distances, while respecting a wide array of constraints.
- It was found however during this analysis that this good performance at AG might be misleading. The real challenge to guidance are the multiple re-targeting maneuvers, each to be performed along an already optimal trajectory to another landing target. A new approach where thrust is held in reserve and the least critical constraint (velocity magnitude) is relaxed is introduced.
- Pending a more formal calibration of the re-targeting method discussed above, it

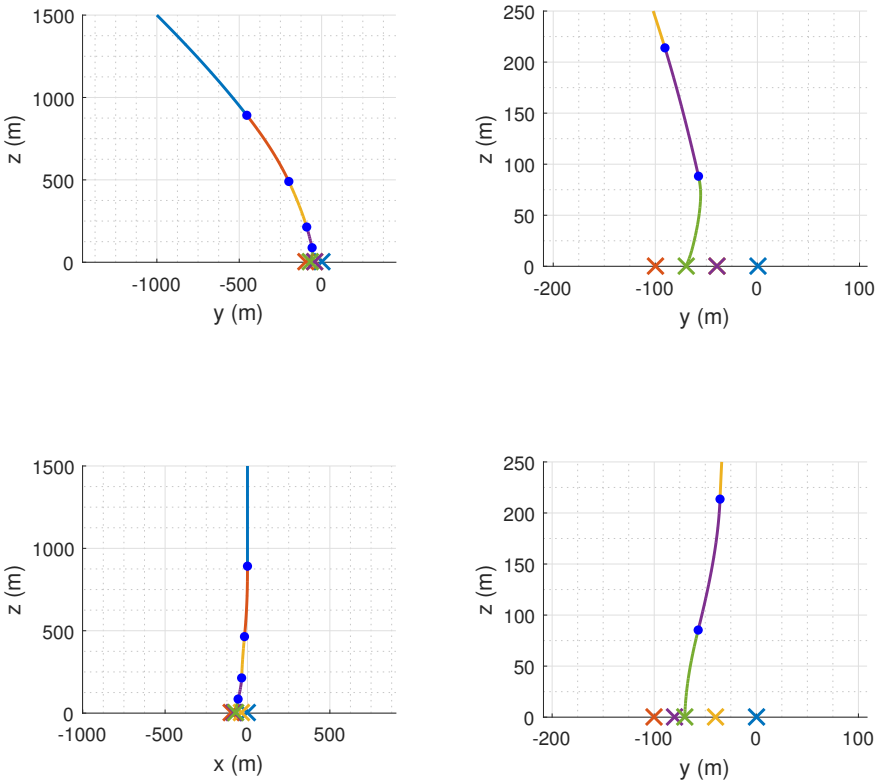

Figure 4.50: Zig-zagging guidance calculated worst-case trajectories for the along- (upper) and cross-range (lower) cases, and from a far (left) and near (right) perspective. Each target landing site has the same color with its respective trajectory segment. Re-target points are marked as blue dots.

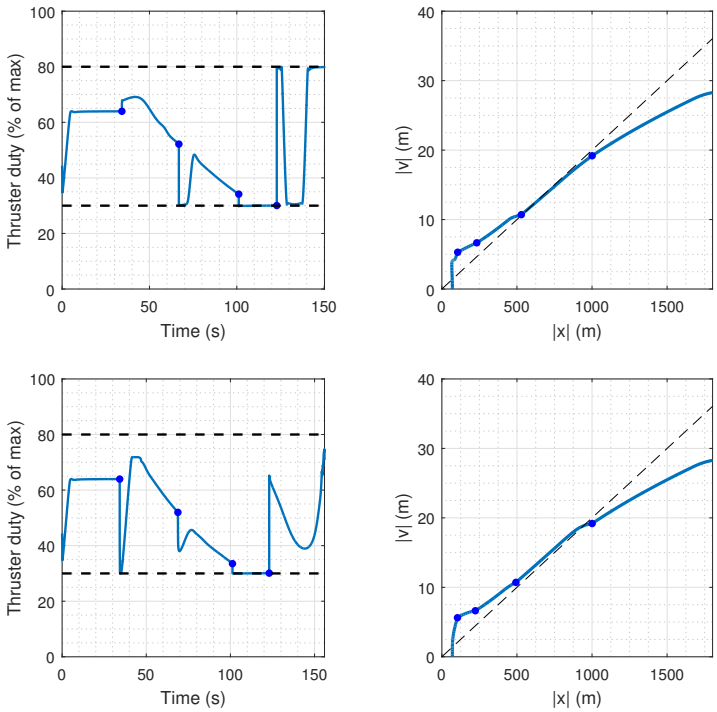

Figure 4.51: Thrust duty and velocity vs range to target plots for the worst-case trajectories of Figure 4.50. The maximum thrust and velocity magnitude constraints are relaxed after each re-targeting command (blue dots) to make the corresponding maneuver possible.

might be possible to use a smaller thrust-to-weight ratio for the thrusters, since they over-perform at AG. This is not expected to be the case however, since TtW will likely be a critical factor for the re-targeting maneuvers.

- The 100 kg of fuel initially allocated with ample margin in the original EnEx lander design, can be reduced to 65 kg for the worst case re-targeting scenario, plus a small margin.

4.4 Combined simulation results

To investigate the combined functionality of the LandingSim tool and get a first estimate of landing success probability, a closed-loop Monte Carlo simulation was performed. In short, for a single MC iteration, the lander started from its nominal position, and performed the navigation and HDA functions ase set out in Tables 4.3 (Nominal case) and 4.9 respectively. Due to technical issues with the solver used for the guidance function in a Monte Carlo environment, as each run required human supervision, it was assume that whatever target the HDA function commanded was reachable by guidance and with the fuel allocated. This assumption was shown in Section 4.3.3 to be reasonable. All parameters where noised as indicated by their respective deviation values. Major modifications for this simulation integrated from the conclusions drawn by the analyses performed in this chapter are listed in Table 4.12. 1000 MC iterations were performed.

Table 4.12: Modifications to the landing GN&C systems for the MC simulation.

Parameter	**Old nominal value**	**New modified value**
NAV: Thermal camera measurement frequency	5 s	2 s
HDA: Optical camera confidence parameter	1	0.2
HDA: Lidar range noise (cumulative, 1σ)	0.33%	0.033%
Monte Carlo iterations	1000	

The results of the Monte Carlo statistics are shown in Figures 4.52 to 4.54. In detail, Figure 4.52 demonstrates the fulfilment of navigation requirements and the correct function of the EKF-SLAM "machinery".

Figure 4.53 presents the landing success statistics. The overall landing success rate is $P_{\text{Success}} = 94.1\%$. Landing failures are primarily due to landing on high slope terrain, and can be connected to the small issues that the HDA function faced with high slope areas. Obstacles are nicely avoided and areas with little snow cover are mostly preferred. Seven runs ended within the critical no-go zone around the plume source.

Figure 4.54 further helps in interpreting the results. Successful landing spots are concentrated around specific map areas, nicely corresponding with the high scoring areas of the HDA safety and landing score maps (Figure 4.32). The chosen areas are also largely snow-free (compare to Figure 4.30). It is further seen that several failed landing areas are at the edges of obstacles, on the thin lines where the HDA function produced false negatives of hazardousness.

Overall, the landing success rate is high but still not within the $\sim 99\%$ success rate typical for landing missions. In addition, the worst-case scenario with significant consequences for planetary protection, with the lander entering the plume source no-go zone, has a probability of $\sim 1\%$. Further studies will need to be performed to investigate and improve both these metrics.

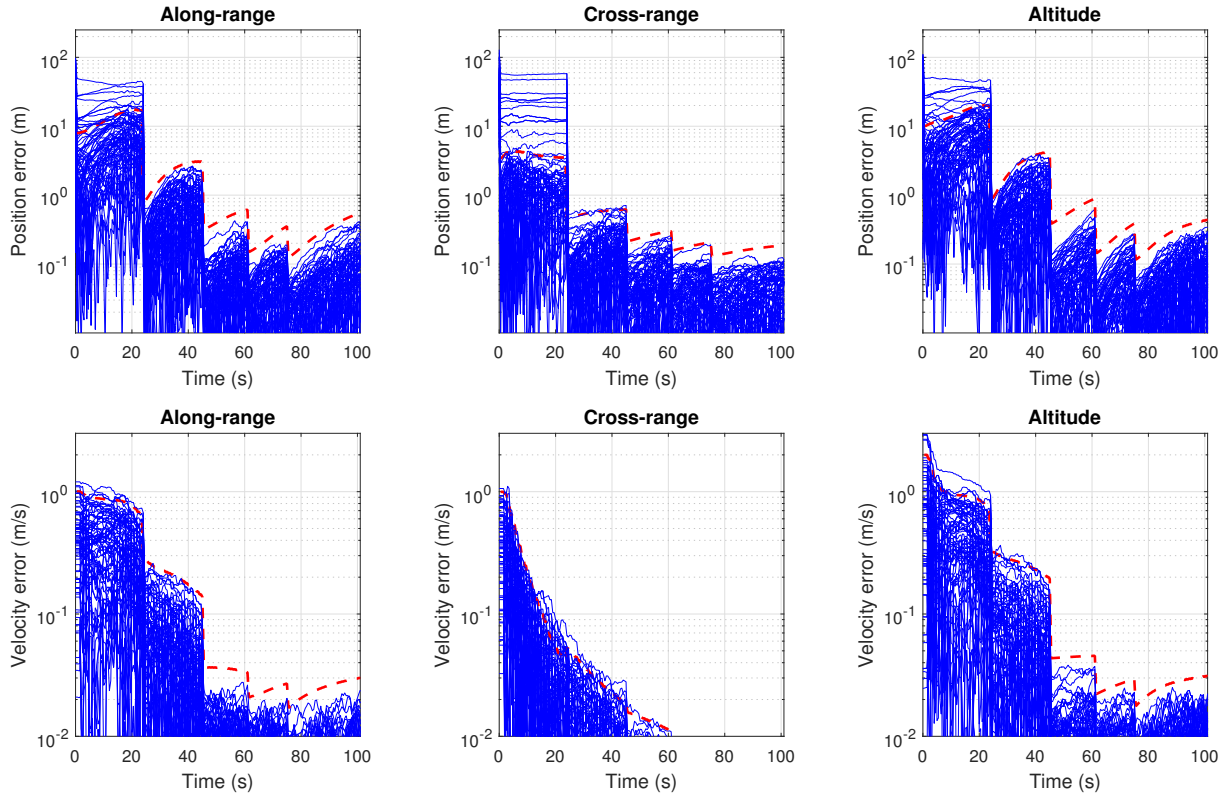

Figure 4.52: Distributions of landing success criteria from Monte Carlo run (1000 iterations)

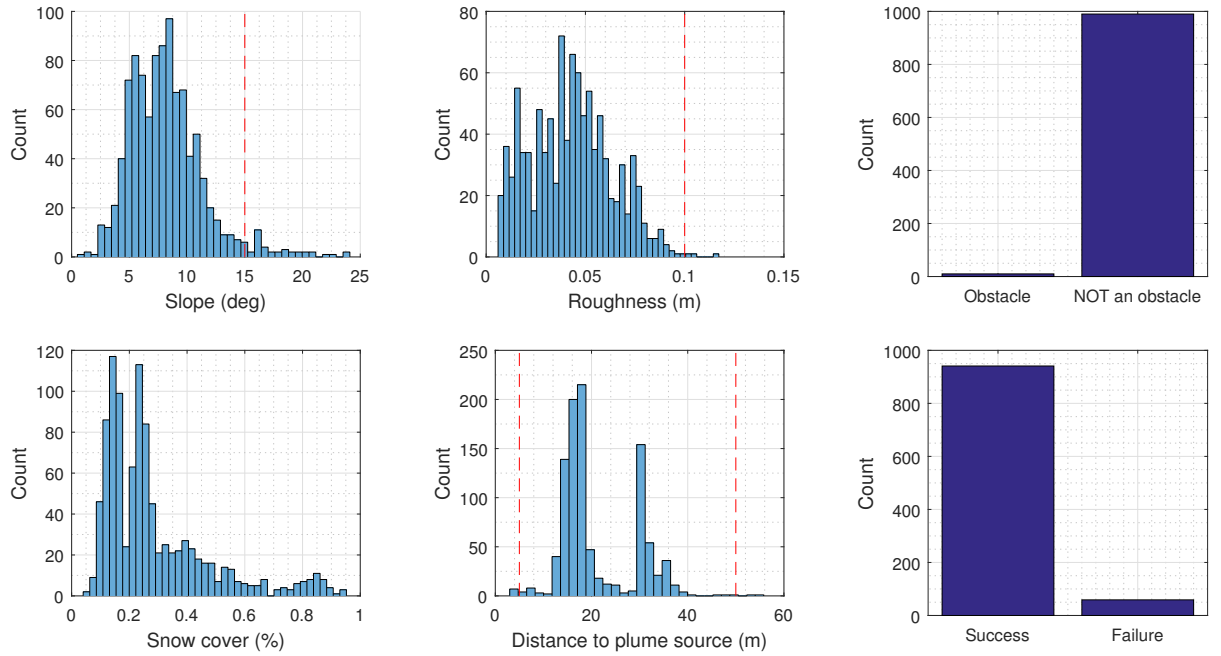

Figure 4.53: Distributions of landing success criteria. Requirements are marked with red dotted lines.

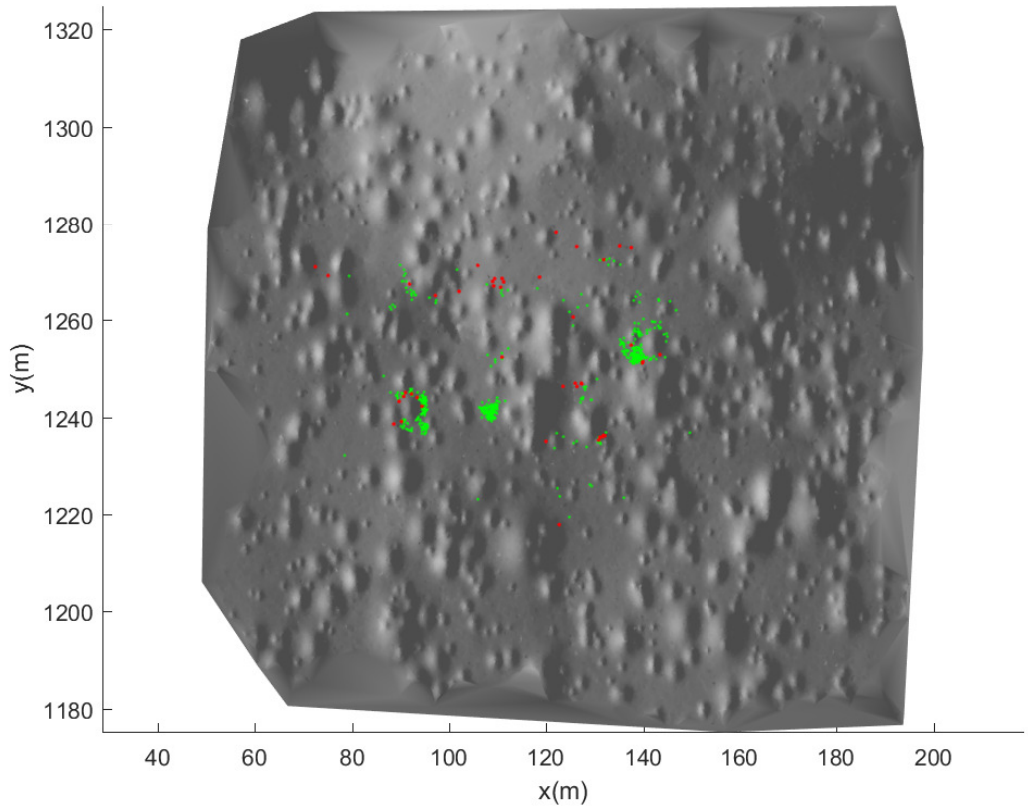

Figure 4.54: Final landing sites on the landing terrain. Green dots represent landing success and red landing failure. Red dots where plotted bigger for better visibility.

Chapter 5

Conclusions and Future Work

The search for life in the solar system is arguably one of the main driving motivations behind planetary exploration. Past, current, and planned orbiter remote sensing missions like the recently decommissioned Cassini spacecraft are showing with increasing convincingness that the icy moons of the giant planets, in particular Europa and Enceladus, are indeed habitable, containing all the necessary conditions to host microbial life. The space agencies are picking up on this trend and are planning near-, mid- and long-term missions to search for and characterize microbial life on the icy moons in-situ.

The EnEx lander, a particularly promising concept, aims to deploy a sub-glacial melting probe to access liquid water pockets in the ice under the plume sources on Saturn's moon Enceladus. This concept takes advantage of the opportunity presented by the plume sources, that bring fresh ocean material, potentially containing biosignatures, very close to the surface.

To achieve this, a landing of unprecedented challenge must be performed, that involves aspects not encountered by other, similarly ambitious mission concepts, including landing in a polar canyon, on a snow terrain with hazardous texture, and on a an extremely sensitive area from a planetary protection point of view.

The GN&C system for this landing must be extraordinarily capable and reliable. In this thesis the first steps towards the creation of this system are taken, in particular for the final critical landing approach phase were the full capabilities must be deployed.

In this work, the functions necessary were derived and investigated in detail: sensing, navigation, hazard detection and avoidance, and guidance. A detailed system and operations concept was defined, with a lot of useful insights from the analysis. The above functions were investigated using a simulation tool created for the purposes of this thesis, first individually and then in a closed loop Monte Carlo simulation. Some conclusions and observations from this analysis will be given here, and proposed steps for future work will be given.

Sensors found to be useful are the lidar and optical camera that are usually found in similarly advanced landing mission concepts. The unusual thermal environment and surface texture, however, make new sensing capabilities necessary, namely a lidar that is capable of sensing texture by analyzing the beam return signal and a thermal camera

that can be used for navigation and (indirectly) surface texture sensing.

These sensors are put to use for navigation and hazard detection. In navigation, a particularly novel and challenging problem, among others, it to be dealt with: the lander should land neither too far nor too close to the plume source. In parallel, the position of this plume source is likely to not be very accurately known before landing start, due to several difficulties in remotely sensing of the south pole of Enceladus.

An elegant way to deal with this that was implemented here is the SLAM method, by which the target point is localized while taking into account the uncertainty of the lander. This way all the critical uncertainties are quantified and taken into account. In this framework, further elements of the environment can be incorporated, like e.g. points whose location is known before landing start, that can assist with navigation. In this framework, the thermal camera, although observing a "spread out" target (the hot spot around the plume source), successfully localizes the plume source. The lidar for hazard relative navigation was found to be a very accurate and robust solution. The optical camera was essential only at the start of the approach phase.

In conclusion, the navigation approach used satisfied the requirements and was shown to be robust to noises. The EKF-SLAM approach used is elegant, and by explicitly formulating and tracking navigation uncertainties, also helps with understanding and verifying the system. Its true power will be better seen when tracking multiple uncertain points at one landing, e.g. in a landing scenario with multiple plume sources, some of them unknown.

The HDA function had to ensure a safe landing at a very challenging canyonous, snowy terrain. The most critical element for HDA was the lidar range noise. Due to this, the operational concept has a weak spot in the early HDA measurements, where the hazard map is "unclear" and a dangerous landing site might be selected there. As future measurements will take place in this original map, the HDA function might become "trapped" in selecting a bad landing site.

A simple and novel approach was developed to extract surface texture information by using thermal images and previous knowledge of the thermal environment. This method still needs to be made more robust and able to work as a stand-alone. It already works well as a complementary method for texture sensing to the uncertain lidar texture measurements. The method could be made to track certain relevant thermal parameters in real time, e.g. the spread of the plume source hot spot spread, and thus be made more robust.

Overall for HDA, the lidar is central and offers a robust base. The optical camera can offer dubious measurements and can be even misleading, since it cannot measure slope, a prevalent feature in a canyon. Whether a thermal camera is a worthy inclusion will depend whether the lander is robust to touchdown on snow, and to the actual hazardousness of the snow on the SPT for a lander.

The fuzzy framework implemented is intuitive and a smooth introduction in reasoning and autonomy under uncertain conditions. However, its relative informality and requirement for manual configuration make the work necessary for large scale analysis potentially tedious and difficult to validate. Probabilistic an DS methods should

be investigated in the future, possibly as a prelude to implementing a decision fusion algorithm for the three.

For guidance, the G-FOLD convex guidance algorithm implemented is clearly formulated, robust and can deal well with multiple constraints. Additional work is needed to implement further constraints to make the landing simulation even more realistic. A new concept is given for the unexpectedly critical part of guidance after each of many re-targeting maneuvers involving gradually increasing the thrust available, while relaxing the least critical constraints after each maneuver, so that the next one is feasible. More work is needed to develop and verify this concept, and make it more robust. In addition, more work is needed on the guidance block of the simulation tool so that it can be executed without human supervision and in Monte Carlo-type large scale simulations.

The closed-loop Monte Carlo simulations produce some first landing success statistics but more work is needed to implement more complex interactions between functions, and to make the guidance function work reliably. MC will be increasingly important for future work, in conjunction with the set of analyses primarily used in this thesis.

An interesting idea for future implementation resulting from this thesis is an alternative design for a more lean landing GN&C system involving only a lidar and a thermal camera. This system would make a landing possibly under any lighting conditions. In this concept, one or multiple plume sources can be initialized with bearing observations from the thermal camera and range observations of by the lidar. A co-registered lidar-thermal map can be used for HDA, assuming preferably that further hazard detection methods can be devised for the thermal camera. This concept would be particularly relevant in scenarios where texture has a critical role and more than one (including unknown) plume sources are expected.

Future work should also expand the studied operations back to DOI, add attitude into the conversation by implementing a 6-DOF approach, adding more noises and biases for the sensors, and more environmental details, including the implementation of e.g. a graphics engine such as Blender or PANGU [64], and so on.

Several longer term ideas for necessary technology development stem from this work.

Special sensors should be further developed satisfying the requirements derived here, in particular for texture sensing, i.e. a texture sensing lidar and a thermal camera, along with further development of the necessary algorithms, in particular for the second case.

In most cases, the GN&C problem is formulated in a deterministic way. The high uncertainties involved in the EnEx landing make a probabilistic/stochastic formulation of the GN&C system desirable. An idea would then be to expand the philosophy behind the SLAM approach to the other GN&C functions and express all relevant parameters and their interactions in a probabilistic way.

A big part of risk and un-reliability in complex autononous systems such as this one is the result not of random failures but of highly complicated interactions between the system elements. An approach to study these interactions and design for reliability from the earlier stages of design is given in [109]. This approach should be investigated in the future for EnEx potentially assisted by simulation tool such as the one developed here.

Finally, the algorithms and approaches developed here could eventually be flown on a landing testbed. A handy and inexpensive platform for such test are quadrocopters. An integrated simulation tool - quadrocopter platform can then serve as an end-to-end testbed for the rapid development of new technologies for landing.

Bibliography

[1] J. I. Lunine, Ocean worlds exploration, Acta Astronautica, Volume 131, 2017, Pages 123-130, https://doi.org/10.1016/j.actaastro.2016.11.017

[2] K.P. Hand, A.E. Murray, J.B.Brinckerhoff, et al., Report of the Europa Lander Science Definition Team (2017)

[3] C.P. McKay, C.C. Porco, T. Altheide, W.L. Davis, T. a Kral, The possible origin and persistence of life on Enceladus and detection of biomarkers in the plume, Astrobiology 8 (2008) 909–919, http://dx.doi.org/10.1089/ast.2008.0265

[4] C.D. Parkinson, M.-C. Liang, Y.L. Yung, J.L. Kirschivnk, Habitability of Enceladus: planetary conditions for life, Orig. Life Evol. Biosph. 38 (2008) 355–369, http://dx.doi.org/10.1007/s11084-008-9135-4.

[5] C. F. Chyba, C. B. Phillips, Possible ecosystems and the search for life on Europa, Proceedings of the National Academy of Sciences Jan 2001, 98 (3) 801-804; DOI: http://dx.doi.org/10.1073/pnas.98.3.801

[6] L. Roth, J. Saur, K.D. Retherford, D.F. Strobel, P.D. Feldman, M. McGrath, et al., Transient water vapor at Europa's south pole, Science 343 (2014) 171–174, http://dx.doi.org/10.1126/science.1247051.

[7] Feature on Enceladus, NASA website, https://www.nasa.gov/image-feature/jpl/pia20013/enceladus

[8] J.R. Spencer, F. Nimmo, Enceladus: an active ice world in the Saturn system, Annu. Rev. Earth Planet. Sci. 41 (2013) 693–717, http://dx.doi.org/10.1146/annurev-earth-050212-124025.

[9] R. Greenberg, Europa - The Ocean Moon, Search For An Alien Biosphere, Springer-Verlag Berlin Heidelberg (2005) http://dx.doi.org/10.1007/b138547

[10] C. Porco, D. DiNino, F. Nimmo, How the Geysers, Tidal Stresses, and Thermal Emission across the South Polar Terrain of Enceladus are Related, Astron. J. 148 (2014)

[11] P. Tsou, D. E. Brownlee, C. P. McKay, et al., LIFE: Life Investigation For Enceladus - A Sample Return Mission Concept in Search for Evidence of Life, Astrobiology, Volume 12, Number 8, 2012, http://dx.doi.org/10.1089/ast.2011.0813

[12] Masahiro Ono, L. Hall (Ed.), Journey to the Center of Icy Moons, Project description website, 2016, https://www.nasa.gov/feature/journey-to-the-center-of-icy-moons

[13] M. Peck et al., Soft-Robotic Rover with Electrodynamic Power Scavenging, NIAC Phase I Final Report, 2016, https://www.nasa.gov/sites/default/files/atoms/files/11-2015_phase_i_mason_peck_soft_robotic_rover_electrodynamic_power_scavenging.pdf

[14] B. Sherwood, J. Lunine, C. Sotin, T. Cwik, F. Naderi, Program options to explore ocean worlds, Acta Astronautica, Volume 143, 2018, Pages 285-296, https://doi.org/10.1016/j.actaastro.2017.11.047

[15] B. Sherwood, Strategic map for exploring the ocean-world Enceladus, Acta Astronautica, Volume 126, 2016, Pages 52-58, https://doi.org/10.1016/j.actaastro.2016.04.013

[16] O. Funke, DLR Explorer Initiatives - Enabling technologies for future robotic space exploration, Presentation at the 5th International Planetary Probe Workshop (IPPW-2018), University of Colorado Boulder, USA, https://www.colorado.edu/event/ippw2018/sites/default/files/attached-files/demotesting_12_funke_presid681_presslides_docid973.pdf

[17] A.z Szumski, K. Konstantinidis, R. Förstner, and B. Eissfeller, Enceladus' Environment and the Design of the Enceladus Ice-Probe Navigation System, AIAA SPACE Forum 2016, (AIAA 2016-5536), https://doi.org/10.2514/6.2016-5536.

[18] B. Giese, P. Helfenstein, P.C. Thomas, A.P. Ingersoll, J. Perry, G. Neukum, C. Porco., The morphology of an active zone near Enceladus' south pole, (2010)

[19] H.R. Martens, A.P. Ingersoll, S.P. Ewald, P. Helfenstein, B. Giese, Spatial distribution of ice blocks on Enceladus and implications for their origin and emplacement, Icarus. 245 (2015) 162–176 doi:10.1016/j.icarus.2014.09.035.

[20] S. Kempf, U. Beckmann, J. Schmidt, How the Enceladus dust plume feeds Saturn's E ring, Icarus. 206 (2010) 446–457. doi:10.1016/j.icarus.2009.09.016

[21] T. Hormigo, Safety and Reliability aspects of Planetary Descent and Landing (from a GNC Perspective), Presentation at the 3rd European Optimisation in Space Engineering Workshop, Glasgow, UK, 2015, https://www.turing-gateway.cam.ac.uk/sites/default/files/asset/doc/1606/Tiago%20Hormigo.pdf

[22] E. Howell, Luna 9: 1st Soft Landing on the Moon, Article on Space.com website, 2016, https://www.space.com/35116-luna-9.html

[23] InSight Landing animated infographic, NASA website, https://mars.nasa.gov/insight/entry-descent-landing/

[24] F. Westall, H. G. Edwards, L. Whyte, et al., Recommendation for the Narrowing of ExoMars 2018 Landing Site, ESA Report, 2014, http://exploration.esa.int/mars/54707-recommendation-for-the-narrowing-of-exomars-2018-landing-sites

[25] J. M. Carson, GN&C Technologies for Safe and Precise Landing, 10th Fault-Tolerant Spaceborne Computing Employing New Technologies, Albuquerque, NM; United States, 2017, https://ntrs.nasa.gov/search.jsp?R=20170004704

[26] T. P. Crain, R. H. Bishop, J. M. Carson, et al., Approach-Phase Precision Landing with Hazard Relative Navigation: Terrestrial Test Campaign Results of the Morpheus/ALHAT Project, AIAA Guidance, Navigation, and Control Conference, AIAA SciTech Forum, 2016, https://doi.org/10.2514/6.2016-0099

[27] J. Carpenter, ESA's plans for Lunar Exploration, On behalf of the ESA Lunar Exploration Team Directorate of Human Spaceflight and Operations, Presentation at the Anual Meeting of the Lunar Analysis Group, Laurel, Maryland, 2014, https://www.hou.usra.edu/meetings/leag2014/presentations/carpenter.pdf

[28] E. A. Robertson, J. M. Carson, F. Amzajerdian, et al., ALHAT GN&C Technologies for Safe and Precise Landing, An Overview of NDL,TRN and HD, 2016, http://forum.nasaspaceflight.com/index.php?action=dlattach;topic=39263.0;attach=1392779

[29] Mission scenario for the Luna-Resurs project, Russian space web site, http://www.russianspaceweb.com/luna_resurs_landing.html#pilot

[30] T. Y. Tang, D.J. Yoon, F. Pomerleau, T. D. Barfoot, Learning a Bias Correction for Lidar-only Motion Estimation, 15th Conference on Computer and Robot Vision (CRV), 2018, https://arxiv.org/abs/1801.04678

[31] B. Razgus, E. Mooij, D. Choukroun, Relative Navigation in Asteroid Missions: A Dual Quaternion Approach, AIAA Guidance, Navigation, and Control Conference, AIAA SciTech Forum, 2017, https://doi.org/10.2514/6.2017-1521

[32] S. Grenstad, Systems Engineering and Environmental Analyses for the Enceladus Lander Mission, Bachelor Thesis, Universität der Bundeswehr München, (2016)

[33] J.Oberst, H.Hussmann, B.Giese, F.Sohl, D.Shoji, A.Stark, K.Wickhusen, M.Wählisch,, Enceladus (and Europa) Geodetic Framework, EnEx Initiative Internal Report (2016)

[34] B. J. Hanna et al., Free Molecular and Collisional Studies of Enceladus' Water Vapor Plumes, 40th Lunar Planet. Sci. Conf. (2009).[10] B. J. Hanna et al., Free Molecular and Collisional Studies of Enceladus' Water Vapor Plumes, 40th Lunar Planet. Sci. Conf. (2009)

[35] J.R. Spencer, A.C. Barr, L.W. Esposito, P. Helfenstein, A.P. Ingersoll, R. Jaumann, et al., Enceladus: An Active Cryovolcanic Satellite, 2009. doi:10.1007/978-1-4020-9217-6

[36] O. Abramov, D. Raggio, P.M. Schenk, J.R. Spencer, Tempreatures of Vents within Enceladus' Tiger Stripes, 46th Lunar Planet. Sci. Conf. (2015)

[37] N. Calonne, F. Flin, S. Morin, B. Lesaffre, S. Rolland du Roscoat, and C. Geindreau, Numerical and experimental investigations of the effective thermal conductivity of snow, Geophysical Research Letters, Vol. 38, L23501, 2011 https://doi.org/10.1029/2011GL049234

[38] K. Konstantinidis, C.L. Flores Martinez, B. Dachwald, A. Ohndorf, P. Dykta, P. Bowitz, et al., A lander mission to probe subglacial water on Saturn's moon Enceladus for life, Acta Astronaut. 106 (2015) 63–89. doi:10.1016/j.actaastro.2014.09.012

[39] B. Dachwald, Mikucki, J., Tulaczyk, S., Digel, I., Espe, C., Feldmann, M., Francke, G., Kowalski, J., Xu, C. (2014). IceMole: A maneuverable probe for clean in situ analysis and sampling of subsurface ice and subglacial aquatic ecosystems. Annals of Glaciology, 55(65), 14-22. doi:10.3189/2014AoG65A004

[40] J. Kowalski, P. Linder, S. Zierke, B. von Wulfen, J. Clemens, K. Konstantinidis, et al., Navigation technology for exploration of glacier ice with maneuverable melting probes, Cold Regions Science and Technology, Volume 123, 2016, Pages 53-70, https://doi.org/10.1016/j.coldregions.2015.11.006.

[41] IMU Errors and Their Effects, Pamphlet, Novatel, 2014, https://www.novatel.com/assets/Documents/Bulletins/APN064.pdf

[42] M. S. Veto, P. R. Christensen, D.A. Spencer, The thermal camera for exploration, science, and imaging spacecraft (THESIS) for the PROX-1 microsat mission, 47th Lunar and Planetary Science Conference (2016), URL

[43] Mosaics of Lunar Swingby Images, NASA website for the LCROSS mission, https://www.nasa.gov/mission_pages/LCROSS/main/Swingby_Mosaics_story.html

[44] MIRICLE Performance Characteristics pamphlet, Thermoteknix Systems Limited, http://www.thermoteknix.com/wp-content/uploads/2013/03/miricle-thermal-imaging-modules.pdf

[45] Image sharpness from wavelength to wavelength, Cornell crash course in astronomy, http://spoon.astro.cornell.edu/crashcourse/sharpness.html

[46] J. Solà-Ortega, Towards Visual Localization, Mapping and Moving Objects Tracking by a Mobile Robot: a Geometric and Probabilistic Approach, PhD Thesis, Laboratoire d'Analyse et d'Architecture de Systemes du CNRS, 2007

[47] A. Farzin, D. F. Pierrottet, L. B. Petway, Glenn D. Hines, V.E. Roback, Lidar Systems for Precision Navigation and Safe Landing on Planetary Bodies, SPIE ISPDI 2011-International Symposium on Photoelectronic Detection and Imaging, Beijing, China, (2011), https://ntrs.nasa.gov/search.jsp?R=20110012163

[48] J.-G. Winther, Landsat TM derived and in situ summer reflectance of glaciers in Svalbard. Polar Research 12 (1): 37-55 (1993), doi:10.1111/j.1751-8369.1993.tb00421.x

[49] R.S. Lu, G. Y. Tian, On-line measurement of surface roughness by laser light scattering, Meas. Sci. Technol. 17 1496 (2006), https://doi.org/10.1088/0957-0233/17/6/030

[50] J. Solà-Ortega, Simultaneous localization and mapping with the extended Kalman filter, 2014, On-line tutorial, http://www.iri.upc.edu/people/jsola/JoanSola/objectes/curs_SLAM/SLAM2D/SLAM%20course.pdf

[51] T. Babb, How a Kalman filter works, in pictures, Bzarg blog, https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/

[52] j. Sola, Interactive Course on EKF and SLAM website, http://www.iri.upc.edu/people/jsola/JoanSola/eng/course.html

[53] J. Solà-Ortega, Teresa Vidal-Calleja, Javier Civera, Jose Maria Martinez Montiel, Impact of landmark parametrization on monocular EKF-SLAM with points and lines, International Journal on Computer Vision, (2012) 97: 339. https://doi.org/10.1007/s11263-011-0492-5

[54] J. L. Sell, A. Rhodes, J. Woods, J. A. Christian, Pose Performance of LIDAR-Based Navigation for Satellite Servicing, AIAA/AAS Astrodynamics Specialist Conference, AIAA SPACE Forum (AIAA 2014-4360), 2014, https://doi.org/10.2514/6.2014-4360

[55] J. Zhu, Z. Zhanga, X. Hu, Z. Li, Analysis and application of lidar waveform using a progressive waveform decomposition method, ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XXXVIII-5/W12, 2011, pp.31-36, http://adsabs.harvard.edu/abs/2011ISPAr3812W..31Z

[56] F. Pirotti, Analysis of full-waveform LiDAR data for forestry applications: a review of investigations and methods, iForest 4: 100-106, http://www.sisef.it/iforest/show.php?id=562

[57] T. Dalling, Explaining Homogeneous Coordinates & Projective Geometry, tomdalling.com blog post, 24 Feb, 2014, https://www.tomdalling.com/blog/modern-opengl/explaining-homogenous-coordinates-and-projective-geometry/

[58] D. G. Lowe, Object recognition from local scale-invariant features, Proceedings of the International Conference on Computer Vision. 2. pp. 1150–1157 (1999). doi:10.1109/ICCV.1999.790410

[59] A.I. Awad, M. Hassaballah (eds.), Image Feature Detectors and Descriptors, Studies in Computational Intelligence 630 (2016), https://doi.org/10.1007/978-3-319-28854-3_2

[60] Feature Detection and Description tutorial, Open Source Computer Vision (openCV) toolbox user's manual, https://docs.opencv.org/3.4/db/d27/tutorial_py_table_of_contents_feature2d.html

[61] A. Johnson, On-Board Real-Time Techniques for Safe and Precise Landing, Presentation at the Go For Lunar Landing Conference, Tempe, AZ, March 4th-5th, 2008

[62] J. Delaune, Vision-based Navigation for Pinpoint Planetary Landing on any Relief, PhD thesis, 2013, https://tel.archives-ouvertes.fr/tel-01021431/document

[63] J. Johansson, M. Solli, A. Maki, An Evaluation of Local Feature Detectors and Descriptors for Infrared Images, ECCV Workshops, 2016, https://doi.org/10.1007/978-3-319-49409-8_59

[64] PANGU - Planet and Asteroid Natural scene Generation Utility, STAR-Dundee.com website, https://www.star-dundee.com/products/pangu-planet-and-asteroid-natural-scene-generation-utility

[65] Overview of Digital Terrain Models (DTM), Webpage of the HiRISE instrument on-board Mars Reconnaissance Orbiter (MRO), LPL, The University of Arizona, https://www.uahirise.org/dtm/about.php

[66] F. Aghili, Robust Relative Navigation by Integration of ICP and Adaptive Kalman Filter Using Laser Scanner and IMU, IEEE/ASME Transactions on Mechatronics, Vol. 21, No. 4, 2016, https://ieeexplore.ieee.org/document/7442874

[67] V. Novak, I. Perfilieva, J. Mockor,Mathematical principles of fuzzy logic. Dordrecht: Kluwer Academic, 1999. ISBN 978-0-7923-8595-0.

[68] G. Shafer, A Mathematical Theory of Evidence, Princeton University Press, 1976

[69] A. Gelman, The Boxer, the Wrestler, and the Coin Flip, The American Statistician, Volume 60, 2006 - Issue 2, https://doi.org/10.1198/000313006X106190

[70] Q. Chen, W. Aickelin, Anomaly Detection Using the Dempster-Shafer Method, Available at SSRN, 2006, http://dx.doi.org/10.2139/ssrn.2831339

[71] H. Seraji, A Multisensor Decision Fusion System for Terrain Safety Assessment, IEEE Transactions on Robotics, Vol. 25, No. 1 (2009), https://doi.org/10.1109/TRO.2008.2006705

[72] N. Serrano, M. Bajracharya, A. Howard, H. Seraji, A Novel Tiered Sensor Fusion Approach for Terrain Characterization and Safe Landing Assessment, 2006 IEEE Aerosp. Conf. (2006) 1–10. https://doi.org/10.1109/AERO.2006.1655795

[73] A. Huertas, Yang Cheng and R. Madison, Passive imaging based multi-cue hazard detection for spacecraft safe landing, 2006 IEEE Aerospace Conference, Big Sky, MT (2006), https://doi.org/10.1109/AERO.2006.1655794

[74] N. Serrano, A Bayesian Framework for Landing Site Selection during Autonomous Spacecraft Descent, 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing , pp. 5112-5117 (2006), https://doi.org/10.1109/IROS.2006.282603

[75] N. Serrano, H. Seraji, Landing Site Selection using Fuzzy Rule-Based Reasoning, Proceedings 2007 IEEE International Conference on Robotics and Automation (2007), https://doi.org/10.1109/ROBOT.2007.364234

[76] F. Camara, J. Oliveira, T. Hormigo, J. Araujo, R. Ribeiro, A. Falcao, M. Gomes, O. Dubois-Matra, S. Vijendra, Data fusion strategies for hazard detection and safe site selection for planetary and small body landings, CEAS Space J 7:271–290 (2015), https://doi.org/10.1007/s12567-014-0072-y

[77] Fuzzy Logic Toolbox™for Use with MATLAB®, User's Guide, Version 2 (2018) https://www.mathworks.com/help/pdf_doc/fuzzy/fuzzy.pdf

[78] J. M. Carson III, A. E. Johnson, F. S. Anderson, G. L. Condon, et al., GN&C Subsystem Concept for Safe Precision Landing of the Proposed Lunar MARE Robotic

Science Mission, AIAA Guidance, Navigation, and Control Conference/SciTech Forum, San Diego, CA, United States (2016), http://dx.doi.org/10.2514/6.2016-0100

[79] L. Blackmore, B. Acikmese, 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), https://doi.org/10.2514/1.47202

[80] I. Gerth, Guidance and Control Methods for Safe and Precise Planetary Landing: A literature Survey, Internship report, TU Delft/ESA-ESTEC, Lunar Lander office (2013)

[81] B. Acikmese, M. Aung, J. Casoliva, Flight Testing of Trajectories Computed by G-FOLD: Fuel Optimal Large Divert Guidance Algorithm for Planetary Landing, AAS 13-386. (2013) 1–14

[82] B. Acikmese, S.R. Ploen, Convex Programming Approach to Powered Descent Guidance for Mars Landing, J. Guid. Control Dyn. 30, 1353 (2007), https://doi.org/10.2514/1.27553

[83] C. Sanchez-Sanchez, D. Izzo, D. Hennes, Optimal Real-time Landing using Deep Networks, 6th International Conference on Astrodynamics Tools and Techniques (ICATT) (2016), https://www.esa.int/gsp/ACT/doc/AI/pub/ACT-RPR-AI-2016-ICATT-optimal_landing_deep_networks.pdf

[84] P. Zhang, K. Liu, B. Zhao, Y. Li, A Computationally Inexpensive Optimal Guidance via Radial-Basis-Function Neural Network for Autonomous Soft Landing on Asteroids, PLoS ONE 10(9): e0137792 (2015), https://doi.org/10.1371/journal.pone.0137792

[85] G. Gelly, P Vernis, Neural Networks as a Guidance Solution for Soft-Landing and Aerocapture, AIAA Guidance, Navigation, and Control Conference, Guidance, Navigation, and Control and Co-located Conferences (2009), https://doi.org/10.2514/6.2009-5664

[86] B. Acikmese, L. Blackmore, Lossless convexification of a class of non-convex optimal control problems for linear systems, Am. Control Conf. (ACC), 2010. (2010)

[87] N.S. Bhasin, W.R. Whittaker, C.G. Atkeson, Fuel-Optimal Spacecraft Guidance for Landing in Planetary Pits, Master Thesis, Robotics Institute, Carnegie Mellon University (2016)

[88] J. Zhang, Control Algorithms and Flight Software Framework for a Spacecraft Guidance Navigation and Control System, Master's Thesis, The University of Texas at Austin, 2011

[89] M. Rudolph, Design of the Landing Propulsion and Nuclear Electric Propulsion Systems for the EnEx Mission, Bachelor's Thesis, Institute for Space Technology and Space Applications (ISTA), Bundeswehr University Munich (2014)

[90] A.I. Razzaghi (Ed.) et al., Enceladus Flagship Mission Concept Study, NASA Goddard Space Flight Center (2007)

[91] R. Fisackerly, A. Pradier, B. Gardini, B. Houdou, C. Philippe, D. De Rosa, J. Carpenter, The ESA Lunar Lander Mission, AIAA SPACE 2011 Conference & Exposition, Long Beach, California (2011)

[92] J. Ferreira, D. Esteves, J. Seabra. T. Hormigo, End-to-end GN&C for the Powered Descent and Safe, Precise Lunar Landing, Presentation at the 15th International Planetary Probe Workshop (IPPW-2018), https://www.colorado.edu/event/ippw2018/sites/default/files/attached-files/dltech_2_hormigo_presid501_presslides_docid1146.pdf

[93] C. Liebe, Accuracy performance of star trackers - a tutorial. IEEE Trans Aerosp Electron Syst 38(2):587–599, 2002

[94] E. Zaunick, D. Fischer, Dr. I. Ahrns, G. Orlando, B. Polle, E. Kervendal, Innovative Visual Navigation Solutions for ESA's Lunar Lander Mission, Presentation at the 9th International Planetary Probe Workshop, Toulouse (2012), https://websites.isae-supaero.fr/IMG/pdf/zaunick_lunarlander.pdf

[95] D. P. Christensen, Terrain-Relative and Beacon-Relative Navigation for Lunar Powered Descent and Landing, Master's thesis, Utah State University (2009), https://digitalcommons.usu.edu/etd/253

[96] P. Lunghi, Hazard Detection and Avoidance Systems for Autonomous Planetary Landing, PhD thesis, Politecnico di Milano (2017), https://www.politesi.polimi.it/bitstream/10589/131168/1/2017-01_PhD_Lunghi.pdf

[97] T. J. Moesser, Guidance and Navigation Linear Covariance Analysis for Lunar Powered Descent, Master's thesis, Utah State University, 2010, https://digitalcommons.usu.edu/etd/654/

[98] B. L. Visser, An Application of Linear Covariance Analysis to the Design of Responsive Near-Rendezvous Missions, Master's thesis, Massachusetts Institute of Technology, 2007, https://dspace.mit.edu/handle/1721.1/59696

[99] W. Wang, Worst-case Analysis of Space Systems, PhD thesis, University of Exeter, 2011, https://ore.exeter.ac.uk/repository/handle/10036/3550

[100] J.-F. Remacle, An introduction to mesh generation Part II: Delaunay-based mesh generation techniques, Class notes, Department of Civil Engineering, Université catholique de Louvain, Belgium, https://perso.uclouvain.be/vincent.legat/teaching/documents/meca2170-jfr-cours2.pdf

[101] T. McClure, Demonstration of Algorithmic Terrain Generation Tutorial webpage, https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/39559/versions/3/previews/html/terrain_generation_introduction.html?access_key=.

[102] J. Birn, Digital Lighting and Rendering, 2nd Edition, New Riders (2006)

[103] J. Sola, SLAM Toolbox for Matlab website, http://www.iri.upc.edu/people/jsola/JoanSola/eng/toolbox.html

[104] Mathwork's Fuzzy Logic Toolbox website, `https://www.mathworks.com/products/fuzzy-logic.html`

[105] TU Munich, Institute of flight dynamics, Open source FALCON optimizer website, `http://www.fsd.mw.tum.de/software/falcon-m/`

[106] J. Diepolder, S. Saboo, V. S. Akkinapalli, Flight Control Law Testing Using Optimal Control and Postoptimal Sensitivity Analysis, Advances in Aerospace Guidance, Navigation and Control, 2017, `https://link.springer.com/chapter/10.1007/978-3-319-65283-2_2`

[107] Stanford Artificial Intelligence Laboratory, Introduction to Computer Vision, Tutorial 1: Image Filtering (website) `http://ai.stanford.edu/~syyeung/cvweb/tutorial1.html`

[108] M. Thies, Simulation of Guidance and Hazard Detection and Avoidance for Autonomous Planetary Landing, Master's thesis, Institute for Space Technology and Space Applications (ISTA), Bundeswehr University Munich (2016)

[109] N. Leveson, Engineering a Safer World: Applying Systems Thinking to Safety, MIT Press (January 2012)

Appendices

Appendix A

Reference frames

A.1 Reference Frames for Navigation

In accordance to the approach followed in Section 2.2, the reference frames from the main source for the EKF-SLAM formulation [46] will be used to define the reference frames used in the navigation function and the transformations between them. The frame transformations are shown in Figure A.1.

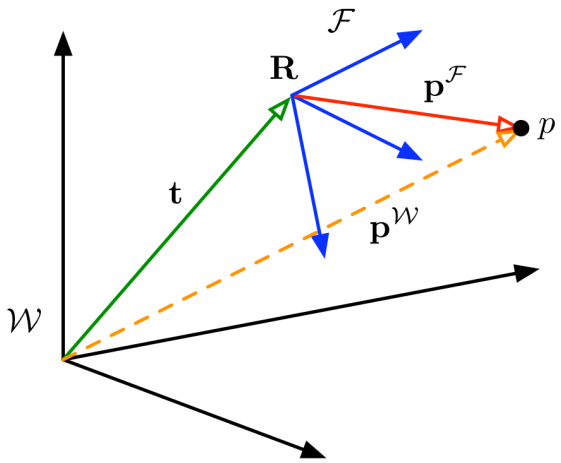

Figure A.1: Frame transformation in 3D. Each of the blue arrows is a column vector of the rotation matrix $\mathbf{R}$, corresponding to the orientation of the local frame $\mathcal{F}$ [46].

A point $\mathbf{p}$ in space can be expressed in the World frame $\mathcal{W}$ or in the local frame $\mathcal{F}$. Both expressions are related by the frame-transformation equations,

$$\mathbf{p}^{\mathcal{W}} = \mathbf{R}\mathbf{p}^{\mathcal{F}} + \mathbf{t} \tag{A.1}$$

$$\mathbf{p}^{\mathcal{F}} = \mathbf{R}^{\top}(\mathbf{p}^{\mathcal{W}} - \mathbf{t}) \tag{A.2}$$

where $\mathbf{R}$ is the rotation matrix associated with the orientation of the local frame $\mathcal{F}$ and $\mathbf{t}$ is the translation matrix of the local frame $\mathcal{F}$ relative to the World frame $\mathcal{W}$. The first expression is known as the ”from frame” transformation and the second as ”to frame”.

A.2 Reference Frames for Guidance

A reference frame often used in planetary landing is the Downrange[25]-Crossrange-Altitude (DCA) frame, illustrated in Figure A.2. It is defined based on the landing vehicle and its target. A ground surface is first defined as parallel to the local horizon and crossing the landing target. The downrange direction is then defined as parallel to the local horizon, starting from the sub-spacecraft point and pointing towards the landing target. The altitude direction is defined as the vertical to the local horizon, pointing from the sub-spacecraft point to the spacecraft. Finally, the cross-range direction completes the orthogonal system as seen in the figure. Negative down-range values are also referred to as up-range.

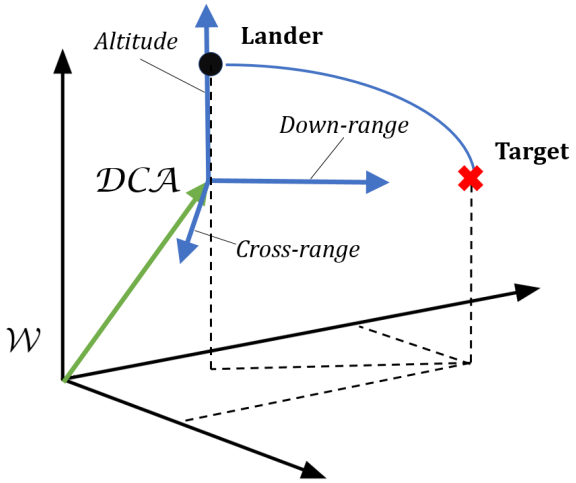

Figure A.2: The Downrange-Crossrange-Altitude (DCA) reference frame. Adapted from [46] based on Figure A.1.

[25]...or Along-range