Simulation of Autonomous Landing near a Plume Source in a Tiger Stripe Canyon on the South Pole of Enceladus

Kostas Konstantinidis, Manuel Thies, Julian Adler, Nico Hochberger, Martin Rudolph, Paul Dykta, Roger Förstner[1]

Abstract This paper presents a simulation tool for planetary landing operations near a plume source on the bottom of a tiger stripe canyon on the South Polar Terrain of Saturn's moon Enceladus, reports on its development status, and gives some first results derived by it.

Enceladus is a hot spot for astrobiology in the solar system. A spacecraft landing near one of the plume sources on the moons South Polar Terrain and deploying a melting probe to sample relatively shallow liquid water in the ice under that plume source would be able to look for signatures of life before they are degraded by exposure to the vacuum of space.

The lander would have to land under very challenging landing accuracy and safety conditions on an exceptionally challenging terrain.

To perform this challenging landing, a sophisticated landing Guidance, Navigation, and Control (GN&C) system would be necessary. To achieve the required accuracy, terrain relative navigation can use sensors such as optical and thermal cameras, LIDAR, etc. to navigate relative to detected terrain features. To ensure a safe landing the system must be able to assess if the originally planned landing site is safe, and if not to then autonomously command a retargeting to another safer spot. The guidance and control function must then calculate a viable trajectory and thrust arc to the newly chosen landing site.

To validate that the landing satisfies the accuracy and reliability requirements we are developing a tool in Matlab/Simulink to simulate the operation of the autonomous landing GN&C system. In this paper we describe the complete tool with all its planned features and we present its development status.

[1] Institute for Space Technology and Space Applications (ISTA), Bundeswehr University Munich, Germany

1 Introduction

1.1 Enceladus

Enceladus is currently seen as one of the prime candidates for hosting microbial life in the Solar system in the present day.

Enceladus is a small icy moon orbiting Saturn at a radius of 4 Saturn radii. Having a diameter of ≈ 500km, it was once believed too small to be active, but it has been found to be one of the most geologically dynamic objects in the Solar System [1]. It is characterized by a wide range of terrains, including old and young surfaces. It's most interesting area is at its south pole, where a geologically active province was identified by Cassini [2]. The most prominent feature at the South pole are four linear depressions, dubbed "tiger stripes" because of their appearance in the infrared. Multiple flybys of Cassini at Enceladus have shown that plumes of H_2O, including simple organic compounds [3], emanate via cryovolcanism from those "warm" fractures at the tiger stripes. Analysis of the plume material strongly implies that it originates from a global ocean below its icy crust [4], and its unique chemistry has fuelled speculation that Enceladus may harbour life [5].

1.2 Topography and Environment of the South Polar Terrain (SPT)

The South-Polar Terrain is host to the plume sources. The most prominent features characterizing the interior of the SPT as mentioned above are the "Tiger Stripe" valleys, which include 101 identified distinct vapor and ice jets which form the plumes towering above Enceladus (Fig. 2) [6]. The term "Tiger Stripe" describes a "V" shaped valley enclosed by two, nearly parallel ridges. These ridges are about 100–150 m high, while the valley is about 200–250 m deep. The total width of the formation is about 2–5 km. The South-Polar Terrain features 4 valleys (Damascus, Baghdad, Cairo and Alexandria Sulcus) separated from each other by 35 km wide plains. As one moves closer to a Tiger Stripe valley, the terrain rises with a relatively gentle gradient to the summits of the ridges. The terrain texture also changes from highly fractured to a more undulating one, covered with numerous icy blocks. Once over the lateral ridges, the slope is initially steeper but changes to a more moderate gradient the deeper one descends, with unconsolidated material sliding from steeper down to flatter sections, where material tends to accumulate. The valley floor is expected to be very narrow, in the order of 50–100 m [7], and interspersed with obstacles, such as narrow ridges and elongated domes. An indicative picture of the South-Polar Terrain at Enceladus can be seen

in Fig. 2. An elevation model of a characteristic canyon area on the SPT is shown in Fig. 3.

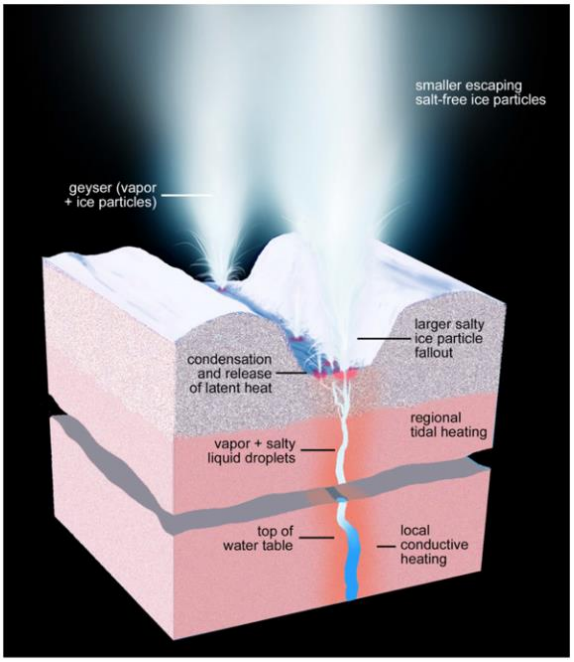

Fig. 1. Schematic illustrating current knowledge of the small-scale physical and thermal structure and processes relevant to Enceladus' geysering activity [6]

The spatial density of ice-block features in the SPT was estimated for blocks larger than 10 m in [8]. Block density was 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.

An important aspect is the texture of the surface material in the South-Polar Terrain. It is mainly the fallout from the plumes that modifies the texture: ice particles tend to accumulate more near the plumes and less further away. Studies have shown that nearer the plumes, particle deposition rates can reach up to 1 mm/year or more in the large scale and possibly more in smaller scales, indicating a deposit layer thickness of tens of meters if we assume that the plumes have been active in the past million years [9]. 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 also 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, where larger particles tend to fall nearer to the plumes [9]. 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 are translated to macroscopic properties of the surface material, namely: increased force transmission capacity due to the fine

grain size, non-consolidated layers of material and increased compressibility due to the low gravity.

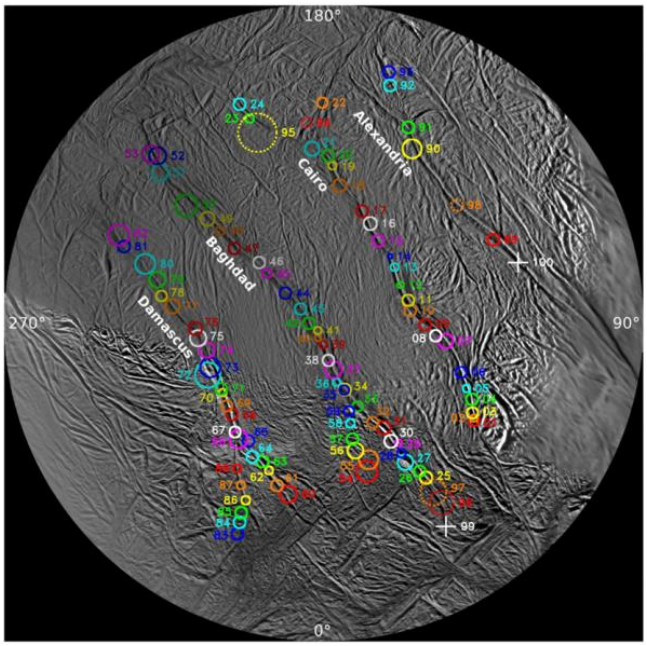

Fig. 2. Polar stereographic map of Enceladus' South-Polar Terrain (SPT) showing the location of 100 plume sources. The circles are the 2σ uncertainties [6]

The density of H_2O molecules in the vicinity of the plumes has been modelled in [10]. At the very mouth of the plume source the water vapor density is estimated in the order of 10^{22} molecules/m^3, dropping by 3 orders of magnitude within a distance of 100 m. To put these numbers into perspective it's instructive to compare them to the 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.

The fresh, clean ice that dominates the surface of Enceladus gives it the most reflective surface of any body in the Solar System (albedo of 0.99). Because of the high albedo characteristic to Enceladus, much sunlight is reflected off of its surface, resulting in a mean surface temperature at noon only reaches 75 K, somewhat colder than other Saturnian satellites [11]. It is estimated that the "tiger stripe" fractures, radiate heat at temperatures up to at least 167 K on the large scale [12]. Small scale (tens of meters) hot spots surrounding the plumes, can reach temperatures of up to 200K [6,13].

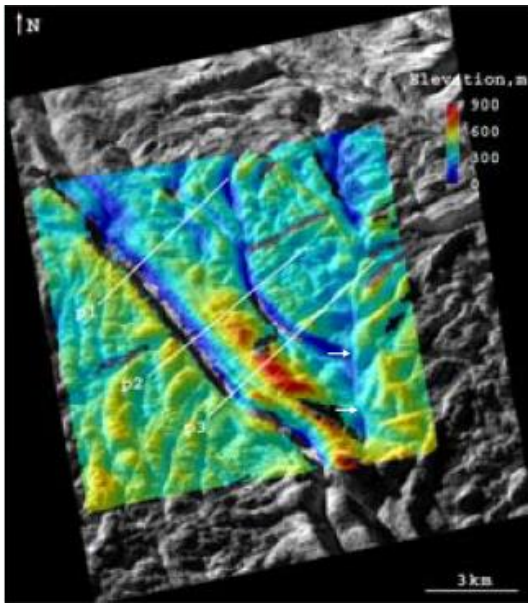

Fig. 3. Stereo-derived elevation model of characteristic canyon area on the SPT, on Baghdad Sulcus (76°S / 323°E) [7]

1.3 The Enceladus Explorer (EnEx) project and the Enceladus lander (EL) mission concept

Between 2012 and 2015 the joint research collaboration "Enceladus Explorer" (EnEx) funded by the German Space Administration (DLR), investigated the necessary technologies for a future exploration of the Saturnian moon Enceladus. The goal was the development of a terrestrial navigation system for a subglacial research probe. The EnEx consortium was led by FH Aachen University of Applied Sciences and consisted of eight German Universities.

The developed navigation solution was integrated into a research melting probe of the IceMole type, a melting probe concept which has been developed at FH Aachen University of Applied Sciences since 2008. The IceMole probe was validated and tested during field tests conducted as part of an ongoing collaboration between EnEx and MIDGE, a NSF funded Antarctic exploration initiative [14].

In the context of the EnEx project, the Institute of Space Technology & Space Applications (ISTA) of the Bundeswehr University Munich was responsible for the overall mission and system concept for a mission to land near a plume at Enceladus and deploy the IceMole there. The mission scenario for the extraterrestrial application of such a probe was studied, in order to determine the conditions for the complete design and extraterrestrial operation of the navigation sys-

tem in detail. The detailed Enceladus Lander (EL) mission concept created during the project is given in [15].

The EL mission concept has the aim to deploy a future version of the IceMole melting probe near one of the plumes of Enceladus. The EL mission comprises of three elements: an Orbiter, a Lander, and the IceMole. This Combined Spacecraft will transfer to Enceladus and communicate with Earth, with the Orbiter serving as the propulsion module. For the transfer to Enceladus, a nuclear reactor on the Lander provides power to the electric propulsion system. After orbital capture around Enceladus, the Orbiter will perform remote sensing of potential landing sites during the landing site reconnaissance phase. Once landing site reconnaissance is completed by the Orbiter, the Lander will separate and land precisely and safely near one of the plume sources on the bottom of a tiger stripe canyon. After Lander separation, the main function of the Orbiter will be to relay data between lander and Earth. Once landed, the IceMole will be deployed. The thermal melting head of the IceMole will be powered by the nuclear reactor. Once the IceMole reaches a depth of 100 m or more close to the wall of water-filled fracture, it will sample liquid water within the fracture for biosignatures.

1.3 Landing near a plume source

1.3.1 Landing Requirements

The drivers for landing accuracy are the width of the canyon floor, at around 100 m, and the desire to minimize the length of the tether connecting the IceMole to the Lander for power and communications purposes. Landing too close to the plume source should also be avoided to minimize planetary protection related concerns (see [16]). Taking these considerations into account, the target landing area shall be circle with a radius smaller than 50 m. Such a landing accuracy is categorized as pinpoint landing (see e.g. [17]).

As seen in Sec. 1.2, the landing area is also expected to be strewn with hazards such as ice blocks, cracks, domes, unconsolidated materials, etc. that are too small to be identified from orbit. Due to the uncertainty of the terrain, real-time hazard detection is necessary, as well as logic for the avoidance of landing hazards.

1.3.2 Landing Guidance, Navigation, and Control (GN&C) System and Operations

To give context to the following, the GN&C system used during descent and landing operations will be briefly described here.

Although a spacecraft may be equipped with a diverse set of sensors, there is typically no need for sophisticated hazard avoidance because the selected landing sites are mostly benign [18]. However in our case an autonomous safe and accurate landing must be performed on an extremely challenging terrain. Therefore the added robustness offered by multiple sensors is expected to be essential. We use therefore an optical camera and a Lidar.

Thermal cameras could be beneficial for landing on the SPT because of the need to land in total darkness and possibly effect of water vapor on Lidar. According to our calculations the vapor is too thin to cause such issues. Even if this is the case, a thermal camera could compensate for the weaknesses of Lidar when it comes to sensing non-opaque materials, such as e.g. ice. It can also take advantage of the high thermal contrast between the warm localized jet sources, and the surrounding colder ice and snow material.

Fig. 4 gives the top-level architecture of the GN&C system for landing. Terrain relative navigation instruments in conjunction with conventional navigation instruments help accurately navigate the lander during landing. During the final landing phases, data from terrain relative instruments is also used to detect and avoid landing hazards. Control of the calculated trajectory and attitude is then performed using the Lander's propulsion system.

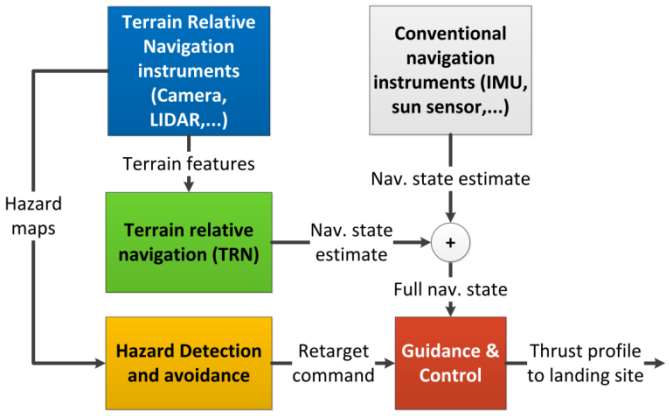

Fig. 4. Illustration of a top-level GN&C system architecture

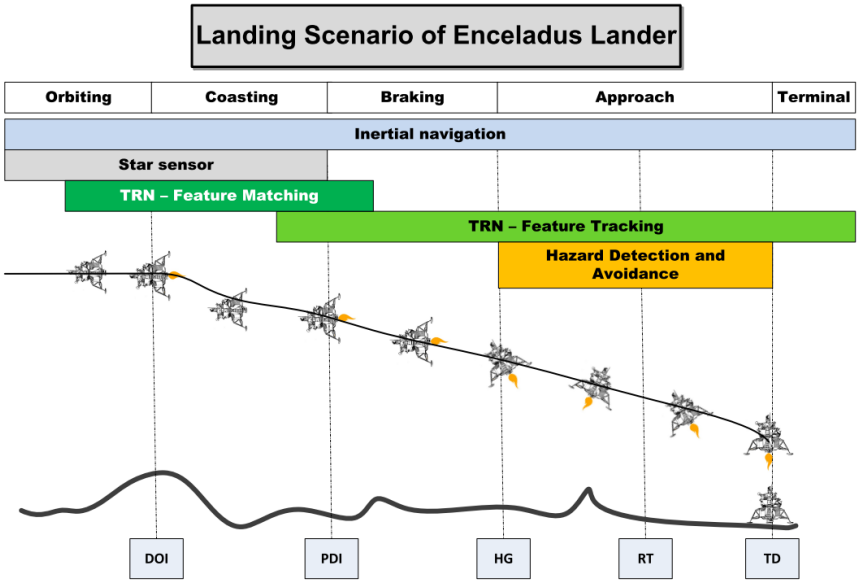

Fig. 5. Operations concept for landing phase (see text for explanation). Feature Matching refers to matching detected features to a feature database. Feature Tracking refers to tracking features from consecutive images

Fig. 5 illustrates the Descent and Landing sequence. The landing operational sequence was based on that of the ESA Lunar Lander mission [19][20][21]. The landing sequences of other icy moon landing missions with similar accuracy requirements were taken into account [22] [17].

The reference landing scenario is divided into separate phases:

- **Lander separation**: The reference scenario starts in the 200 km Enceladus Reconnaissance orbit, after separation between the Lander and the Orbiter has been successfully performed.
- **Coasting phase**: The Descent Orbit Insertion (DOI) marks the start of the descent phase. This is short burn by the main engines of the lander to go on a Hohmann transfer orbit and periapsis of 5 km. During the coasting phase up to periapsis, feature matching TRN is performed to increase precision of state estimation.
- **Braking phase**: At periapsis the Powered Descent Initiation (PDI) meaning that the main engine is turned on for the main braking phase most of the orbital velocity is eliminated, from orbital speeds to about 100 m/s. During or shortly after the main braking, the landing site will come into the field of view of the camera. This marks High Gate (HG).

- **Approach phase**: Once the Lander has achieved a sufficiently reduced velocity the engines begin to throttle down and the approach phase begins, starting targeted descent towards the selected landing spot. During this phase the Hazard Detection and Avoidance (HDA) function is active. If the designated landing spot proves unsuitable, retargeting can take place and a new landing spot within the Lander's reach selected. The spacecraft will perform feature tracking TRN, complemented with IMU measurements during this phase.
- **Touch down**: Once the Lander is above the landing site with no horizontal velocity and a vertical attitude at an altitude of 10-20 m (Terminal Gate, TG), it shuts it engines and slowly free-falls to a soft Touchdown (TD) on the surface.

In this work, we have identified the approach phase as the most critical, since it is the phase where critical hazard detection and avoidance takes place. Therefore, in the following we will focus only on this phase.

2 Landing Simulation Tool

To confirm the feasibility of landing on Enceladus with the set requirements, we are developing a tool in Matlab/Simulink to simulate the operation of the autonomous landing GN&C system. As mentioned above, the tool focuses on the approach phase, as the most critical. The four functions of the tool are described in detail in this chapter. In Ch. 3 we give a non-strictly formulated algorithm that we implemented in Matlab to create the tool. We also report on the development status of the tool.

The four blocks of the tool are meant to interact with each other in a closed loop: Terrain simulation generates a simulated terrain and produces simulated TRN sensor output. This output is in turn fed to the Terrain relative navigation block, which uses it to assess the navigational state of the lander with the required accuracy. The sensor output is also fed to the Hazard detection and avoidance block, so that it can calculate the landing safety of all candidate landing sites, and if necessary, command a retargeting to a landing site other than the nominal one. Finally, the Guidance and control block calculates feasible trajectories from the landers current state, to the chosen landing site. The blocks run in a loop for every timestep of the simulation that is completed when the lander touches down on the chosen landing site, or when it has failed to do so.

The tool is under development and not all its intended features are currently present. The development status of each individual block, as well as of the tool as a whole is given in Ch. 3.

2.1 Terrain Simulation Block

The target area for the EL lander is one of the plume sources on the bottom of a tiger stripe canyon on the south pole of Enceladus. As described in Sec. 1.2, this is a very challenging terrain to land on. The purpose of this tool block is to generate adequately realistic simulated terrains that can be used for the purpose of simulating sensor input to the TRN and HDA functions. These two functions are discussed later in this paper.

Although existing tools, such as Terragen[2], Grome[3] or Houdini[4] to name just a few, provide the required functionality; they also come with a vast amount of features that are not required in this context. Besides impeding performance, this typically also leads to a considerable amount of effort to be put into making these tools work with the already existing parts of the simulation.

PANGU[5] is a set of tools for modelling the surfaces of planetary bodies and can simulate TRN sensor measurements. These measurements can then be used for off-line and closed-loop simulations of planetary landing. This tool would be potentially beneficial to our work, it is however not freely available.

We have decided therefore to design a streamlined application specifically developed for and tailored to the existing tools and processes as a more viable choice in the current state of this work.

2.1.1 Terrain generation

A common approach to generate realistic terrains are fractal terrain generation algorithms, such as the Diamond-Square-Algorithm or the Square-Square-Algorithm as described in [23]. The implementation used in this project generates an equidistant grid of points in the x-z plane and successively calculates the elevation for each point in this grid using the average value of its surrounding and randomly generated offsets.

Despite being of roughly realistic shape, the generated terrains are typically rather edged, resembling harsh rocky areas. As Enceladus' surface is covered with snow and ice the generated terrain has to be smoothed out to some extent. This is achieved by using an iterative erosion algorithm as suggested by [24]. Depending on the number of iterations performed by the erosion algorithm, the terrain either remains edgy or becomes smooth, resembling a snow covered area.

---

[2] http://planetside.co.uk/

[3] http://www.quadsoftware.com/index.php?m=section&sec=product&subsec=editor

[4] https://www.sidefx.com/products/houdini/

[5] https://www.star-dundee.com/products/pangu-planet-and-asteroid-natural-scene-generation-utility

In addition to generating the terrain itself, the tool is also capable of creating and spreading obstacles, e.g. boulders or stones of various sizes, onto the surface.

A major shortcoming of this approach is that it is nearly impossible to control which kind of terrain is to be generated. In order to gain some control, the generated terrain is not used as is, but instead, an existing DEM of an area on Enceladus, or, as data on Enceladus is rather sparse, a DEM of a canyon like terrain on Earth is combined with the randomly generated terrain. By doing so, the basic shape of the Earth DEM will be retained while the resolution of the generated terrain improves the overall resolution and level of detail. Furthermore, additional information, or "terrain meta-data", e.g. temperature or composition of the surface can be added to the DEM.

2.1.2 Terrain Visualization

To achieve a convenient visualization of the generated terrain, the widely used graphics API OpenGL[6] is utilized. The visible surface is created by drawing a series of triangles with the points indicated by the generated grid as corners of these triangles. Using OpenGL's lighting and shading mechanics, the surface is illuminated according to its geometry and the lighting characteristics on Enceladus.

Depending on the slope of each triangle, the visualization uses a different texture. For areas with a low slope, the simulation uses a snowy texture, while higher slope areas are displayed using a texture resembling solid ice. Thus the accumulation of fallout on relatively flat surfaces is simulated. Also, the reflective behavior of the surface is tuned to match the texture. Snow covered areas feature a more diffuse way of reflecting light, while icy areas show a sharper, specular reflection behavior.

Further, more elaborate possibilities of OpenGL, such as the utilization of bump maps to provide the surface with higher resolution structure and shadowing make the visualization complete, providing a light and adequately accurate simulation of terrain similar to that of a canyon on the SPT of Enceladus.

2.1.3 Terrain Relative Sensor Output Simulation

The sensor simulation part of the tool is meant to create sensor data based on the DEM created during terrain generation and visualization. The sensor simulation is capable of emulating three kinds of sensors:

- Optical camera: Using OpenGL, a greyscale image of the scenery based on the lander's position and orientation is rendered and provided to the user.

[6] https://www.opengl.org/

- Lidar: Based on the underlying DEM, this sensor provides the user with an elevation map of the targeted area. This is done by generating simulated point clouds by ray-tracing each of the Lidar beams to the point where they intersect the DEM surface, and then by introducing some appropriate level of randomness to the location of each intersection point, to simulate measurement noise.
- Thermal camera: If the DEM contains thermal information, this sensor can provide an image of the thermal conditions on the surface, again by using OpenGL according to the underlying physical laws [25].

Upon startup, the sensor emulation tool creates a TCP/IP socket for each of the sensors. By connecting to one of these sockets, the user can request that sensor's data by providing the lander's position and the direction in which the sensor points at that time.

2.2 Terrain Relative Navigation (TRN) Block

Traditional landing approaches based on inertial sensing do not have the navigational precision to meet the high accuracy requirements described above. The purpose of TRN is to augment inertial navigation by providing position or bearing measurements relative to detected and then tracked landmarks [26]. In order to be able to land under any illumination conditions and benefit from the increased performance of multiple sensors, we will investigate here both an optical camera and a Lidar for TRN. For landing near a plume source on the SPT, we also plan to investigate any benefits to TRN that can be gained by using a thermal camera in order to take advantage of the particular thermal environment of the Tiger Stripe canyons and of individual plume sources. In this tool we will therefore use a multi-sensor Simultaneous Localization And Mapping (SLAM) approach for TRN.

Simultaneous localization and mapping (SLAM) is the problem of concurrently estimating in real time the structure of the surrounding world (the map), perceived by moving one or more "exteroceptive" (e.g. camera) as well as "proprioceptive" (e.g. IMU) sensors, while simultaneously getting localized in it [27].

Here we will follow the SLAM implementation as described in [28] and implemented in Joan Sola's open source SLAM Toolbox[7].

In EKF-SLAM, the map consists of a random state vector containing the robot pose and the currently mapped landmark positions:

$$X = \begin{bmatrix} R \\ M \end{bmatrix} \tag{1}$$

[7] http://www.iri.upc.edu/people/jsola/JoanSola/eng/toolbox.html

with

$$R = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{q} \end{bmatrix} \qquad M = \begin{bmatrix} \boldsymbol{p_1} \\ \vdots \\ \boldsymbol{p_n} \end{bmatrix}$$

where R is the robot state containing position and orientation and M is the set of landmark positions, all expressed in the same global reference frame. In the EKF framework, the a posteriori density is approximated by a Gaussian density with mean and covariances matrix defined by

$$\hat{X} = \begin{bmatrix} \hat{R} \\ \hat{M} \end{bmatrix} \qquad \boldsymbol{P} = \begin{bmatrix} \boldsymbol{P_{RR}} & \boldsymbol{P_{RM}} \\ \boldsymbol{P_{MR}} & \boldsymbol{P_{MM}} \end{bmatrix} \tag{2}$$

The objective of the SLAM system is to keep this pdf up-to-date when any of the following situations occurs:

1. The robot moves;
2. The robot perceives a landmark already existing in the map; and
3. The robot perceives a new landmark and decides to incorporate in the map.

These operations are described in the following paragraphs. In the following we use the common EKF formulation.

Robot motion: the prediction step

The evolution of the robot pose during one time step is described by the function

$$R^+ = \boldsymbol{f}(R, \boldsymbol{u}) \tag{3}$$

where $u \sim N\{\hat{u}; U\}$ is a vector of controls assumed to be Gaussian with mean $\hat{u}$ and covariances matrix U. From the EKF formulation we get the prediction step

$$\hat{R}^+ = \boldsymbol{f}(\hat{R}, \hat{\boldsymbol{u}}) \tag{4}$$

$$\boldsymbol{P}^+_{\boldsymbol{RR}} = \boldsymbol{F_R P_{RR} F_R^T} + \boldsymbol{F_u U F_u^T} \tag{5}$$

$$\boldsymbol{P}^+_{\boldsymbol{RM}} = \boldsymbol{F_R P_{RM}} \tag{6}$$

$$\boldsymbol{P}^+_{\boldsymbol{MM}} = \boldsymbol{P_{MM}} \tag{7}$$

where the Jacobian matrices are defined by

$$F_R = \left.\frac{\partial \boldsymbol{f}}{\partial R^T}\right|_{\hat{R},\hat{u}} \qquad F_u = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}^T}\right|_{\hat{R},\hat{u}} \tag{8}$$

Observations of existing landmarks: the correction step

The measure of a landmark i is described by the function

$$\boldsymbol{y_i} = \boldsymbol{h}(R, \boldsymbol{p}_i) + \upsilon \tag{9}$$

where $\upsilon \sim N\{0; R\}$ is a white Gaussian noise with covariances matrix R. The correction step at observation of landmark i is then

$$\boldsymbol{z}_i = \boldsymbol{y}_i - \boldsymbol{h}(\hat{R}, \hat{\boldsymbol{p}}_i) \tag{10}$$

$$\boldsymbol{Z}_i = \boldsymbol{H_i} \boldsymbol{P} \boldsymbol{H}_i^T + \boldsymbol{R} \tag{11}$$

$$\boldsymbol{K}_i = \boldsymbol{P} \boldsymbol{H}_i^T \cdot \boldsymbol{Z}_i^{-1} \tag{12}$$

$$\hat{X}^+ = \hat{X} + \boldsymbol{K_i} \cdot \boldsymbol{z_i} \tag{13}$$

$$\boldsymbol{P}^+ = \boldsymbol{P} - \boldsymbol{K_i} \boldsymbol{Z_i} \boldsymbol{K_i^T} \tag{14}$$

where the Jacobian matrix is defined by

$$\boldsymbol{H}_i = \left.\frac{\partial \boldsymbol{h}(R, \boldsymbol{p}_i)}{\partial X^T}\right|_{\hat{X}}$$

Landmark initialization

Initialization consists of stacking the new landmark position p into the map as

$$X^+ = \begin{bmatrix} X \\ \boldsymbol{p} \end{bmatrix} \tag{15}$$

and defining the pdf of this new state (the resulting map) conditioned to observation **y**. This task is easily performed from the first observation given by **y** = **h**(R, **p**) + υ as all the components of **p** are observed. The classic method performs the variable change

$$\boldsymbol{w} = \boldsymbol{h}(R, \boldsymbol{p}) \tag{16}$$

so the measurement is now y = w + υ. Then it defines the function g, inverse of h, in order to obtain an explicit expression of p

$$\boldsymbol{p} = \boldsymbol{g}(R, \boldsymbol{w}) \tag{17}$$

Assuming that P_{RR} and R are small enough we can approximate this expression with the Taylor series truncated at the linear terms

$$\boldsymbol{p} \approx \boldsymbol{g}(\hat{R}, \boldsymbol{y}) + \boldsymbol{G}_R(R - \hat{R}) + \boldsymbol{G}_W(\boldsymbol{w} - \boldsymbol{y}) \tag{18}$$

with the Jacobian matrices defined by

$$\boldsymbol{G}_R = \left.\frac{\partial \boldsymbol{g}}{\partial R^T}\right|_{\hat{R},y} \qquad \boldsymbol{G}_w = \left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{w}^T}\right|_{\hat{R},y} \tag{19}$$

Then p can be considered approximately Gaussian with mean and covariances matrices defined by

$$\hat{\boldsymbol{p}} = \boldsymbol{g}(\hat{R}, \boldsymbol{y}) \tag{20}$$

$$\boldsymbol{P}_{\boldsymbol{pX}} = \boldsymbol{G}_{\boldsymbol{R}} \boldsymbol{P}_{\boldsymbol{RX}} \tag{21}$$

$$\boldsymbol{P}_{\boldsymbol{pp}} = \boldsymbol{G}_{\boldsymbol{R}} \boldsymbol{P}_{\boldsymbol{RR}} \boldsymbol{G}_{\boldsymbol{R}}^T + \boldsymbol{G}_W \boldsymbol{R} \boldsymbol{G}_W^T$$

where $\boldsymbol{P}_{RX} = [\boldsymbol{P}_{RR} \quad \boldsymbol{P}_{RM}]$. The augmented map is finally specified by

$$\hat{X}^+ = \begin{bmatrix} \hat{X} \\ \hat{\boldsymbol{p}} \end{bmatrix} \qquad \boldsymbol{P}^+ = \begin{bmatrix} \boldsymbol{P} & \boldsymbol{P}_{pX}^T \\ \boldsymbol{P}_{pX} & \boldsymbol{P}_{pp} \end{bmatrix} \tag{22}$$

Its relatively high algorithmic complexity limits the usage of EKF-SLAM to moderately small maps. This has triggered an impressive amount of research on the filtering side of SLAM [27]. There are techniques alternative and closely related to SLAM, such as Visual Motion Estimation (VME) and Structure From Motion (SFM) [27]. Due to the relatively small size of the maps used, EKF-SLAM should be adequate for our application. However, when SLAM will have to be implemented on a real-time platform, one of the above may have to be used instead.

In the following, we discuss how features can be extracted from raw sensor data, for use in SLAM.

2.2.1 Feature extraction

A central role in SLAM is played by features (also known as landmarks or keypoints) detected by extroceptive sensors. But how are landmarks extracted from raw sensor data?

Various feature detection methods in 2D images exist. The general function of such methods is to extract distinctive features from 2D images and assign appropriate descriptors to those features, so that they can be matched between consecutive images.

The search for discrete image correspondences can be divided into three main steps. First, 'interest points' are *detected* at distinctive locations in the image, such as corners, blobs, and T-junctions. The most valuable property of an interest point detector is its repeatability, i.e. whether it reliably finds the same interest points under different viewing conditions. Next, the neighborhood of every interest point is *described* by a feature vector. This descriptor has to be distinctive and, at the

same time robust to noise, detection errors, and geometric and photometric deformations. Finally, the descriptor vectors are *matched* between different images [29].

Several feature detection algorithms for 2D images have been developed. Those of most interest for our application are briefly described in the following. The interested reader can investigate the mathematical details in the respective references.

SIFT (Scale-Invariant Feature Transform) is a rotation- and scale-invariant detection algorithm by Lowe [30]. The image features detected by SIFT have many properties that make them suitable for matching differing images of a terrain. The features are invariant to image scaling and rotation, and partially invariant to change in illumination and 3D camera viewpoint. In addition, the features are highly distinctive, which allows a single feature to be correctly matched with high probability against a large database of features, providing a basis for object and scene recognition. SIFT has a high recognition rate, and is also relatively fast. For real-time applications however, each one of the three steps (detection, description, matching) should be faster still [29].

The SURF (Speeded-Up Robust Features) algorithm is based on the same principles and steps as SIFT, but taking steps to reduce calculation time, as well as robustness [29].

Another promising feature matching algorithm is ORB (Oriented FAST and Rotated BRIEF). It builds on the FAST keypoint detector and the BRIEF descriptor; hence its name. ORB is two orders of magnitude faster than SIFT, while performing as well in many situations [31]. ORB has been used in planetary landing simulation, and its high performance for that application has been confirmed [32].

As we see from the above, as long as extroceptive sensor data can be represented as 2D images, they can be used with one of the above feature detection methods. Some considerations on each extroceptive sensor included in the EL system, as well as possible alternative approaches to feature detection, are given below

2.2.1.1 Optical Camera

Optical camera images are the most commonly used for feature detection and the above methods are most mature for optical image data. SURF appears to combine the best between speed and performance [29], something we have verified by our own implementation of the method.

2.2.1.2 Thermal Camera

The nature of thermal-infrared images makes them highly robust to changing lighting conditions and other environmental effects such as the presence of fog,

smoke and dust. Vision-systems which operate in this wavelength can therefore work effectively in difficult settings, or even in total darkness. Thermal-infrared SLAM could be used to help guide and localize robots and other vehicles when conditions cause other sensors to fail, or perform poorly. However, working in the thermal spectrum brings with it a range of challenges such as limited resolution, low SNR (Signal-Noise Ratio), data interruptions, and so on [33].

Approaches to detect features in thermal images have been proposed in [34] [35], and reviewed and compared in [33]. [33] concludes that SURF outperforms other descriptors for thermal images.

Not many SLAM implementations operating in the thermal-infrared can be found in the literature. [33] present a monocular SLAM system designed and tested for use in the thermal-infrared.

2.2.1.3 Lidar

As we have seen above, feature extraction is reasonably well developed in images. However, due to issues inherent to Lidar (noise, occlusion, etc.) such methods are less developed for Lidar [36]. Such methods are given for example in [37,38].

An application of Lidar TRN for planetary landing is presented in [39]. In it a process is described, where digital elevation maps generate by Lidar measurements are matched with a pre-existing elevation map of the relevant landing area, thus estimating the navigation state of the lander (absolute navigation as discussed above). Such a process can be adapted for a relative navigation approach, comparing and matching features between consecutive measurements of the Lidar. In that work however, no details are given on the exact methods and algorithms for feature detection, description, and matching in Lidar point clouds.

Another Lidar feature matching TRN approach that could be adapted for feature tracking TRN is given in [40]. In this work, Lidar point clouds are transformed to a 2D Lidar contour DEM. This contour DEM is then matched to a reference DEM using a floating-point correlation algorithm [41].

A simple approach we are currently using for our tool however is to represent elevation data as a grayscale 2D image, with higher pixel intensities representing higher altitudes. The SURF method can then be readily applied to those images.

2.3 Hazard Detection and Avoidance (HDA) Block

In order to achieve landing with an adequate reliability on the challenging terrain near the bottom of a tiger stripe canyon, a capable HDA function must be implemented. HDA can be broken down to three sub-functions (Fig. 6) [18]:

- **Hazard detection:** the sensors continuously capture data from which terrain features are extracted
- **Safety reasoning:** the extracted features are evaluated based on safety requirements
- **Landing reasoning**: if the original landing site is found to be hazardous, the most suitable landing site is determined.

In this work, we have adapted the Fuzzy rule based HDA method described in [42–44], where the detection step is performed by multiple heterogeneous sensors, and the safety and landing reasoning steps are performed using a fuzzy logic algorithm. A promising approach would also be to apply a hybrid system of multiple decision engines as proposed in [18]. Such work however is left for the future.

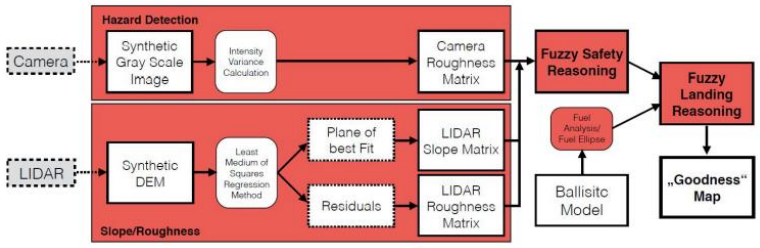

Fig. 6. Three-stage safe landing site determination process

2.3.1 Hazard Detection

Research for hazard avoidance technologies for unmanned space exploration has typically focused on a single sensor. For instance, a lidar model (incorporating parameters such as beam divergence and detector noise) and an algorithm to determine slope and roughness have been developed. Visible image algorithms have also been developed to determine slope and roughness from multiple images (using feature tracking and homography estimation), as well as hazard (rock and crater) detection based on image segmentation and shadow detection. The significance of the proposed approach lies in the use of multiple sensors to provide tolerance to single-sensor failure, as well as to utilize diverse sources of information for hazard assessment [18].

For hazard mapping, on-board sensors sense the terrain to detect hazards. Multiple sensors are used (lidar, optical camera) for robustness. A combination of terrain features derived directly from the sensor measurements (slope, roughness) are used together with terrain features obtained by landmark detection algorithms (ice blocks, ice cracks). A hazard map can thus be created from each instrument. A list of extracted features by each sensor is given in Table 1.

Table 1. Terrain features that can be detected by each TRN sensor

Optical Camera	Rocks/Ice blocks, Roughness, Craters[8], Ice cracks
Lidar	Slope, Roughness
Thermal Camera	Rocks/Ice blocks, Roughness

Each sensor produces independent measurements of the terrain. Terrain features are extracted from the sensor measurements and used to assess the safety of the terrain. Slope and roughness features are extracted from range data using a plane fitting technique. Least median of squares (LMedSq) regression is used to estimate the local plane parameters a, b, and c at location (x, y) of the range data. Given these parameters, the slope is obtained by

$$f_\theta^k(x,y) = \cos^{-1}\left(\frac{1}{\sqrt{a^2+b^2+1}}\right) \quad (23)$$

The roughness feature is simply the residual of the range data and the fitted plane at location (x, y).

$$f_e^L(x,y) = \left|d(x,y) - (ax+by+c)\right| \quad (24)$$

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 [43]:

$$f_3(x,y) = \frac{1}{|W|}\sum_{(i,j)\in W}[c(i,j)-\bar{c}_W]^2 \quad (25)$$

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

Hazard detection algorithms are applied to camera imagery to detect rocks. For the camera rocks are easily detectable due to high intensity changes within a given image. However, good lightning conditions are required to produce high intensity changes. At the current state of the work, the implementation of rock detection is left for the future. The algorithm we intent to apply is described in [45].

Obstacle and roughness features are expected to be extracted from thermal images using the same methods as in optical images. It remains to be investigated however whether the contrast of such features in thermal images will be enough to allow detection.

2.3.2 Safety reasoning

[8] The target landing area on the SPT of Enceladus is geologically very young and is expected to contain no craters. Equivalent algorithms can however be conceivably used to identify other characteristic terrain features relevant to the SPT such as cracks in the ice, etc.

Safety reasoning will perform the decision process to assess the landing safety of the investigated area. Principles of reasoning under uncertainty are used to assess landing safety based on terrain features observed by the sensors. [42] considers three methods: fuzzy set theory, Bayesian probability theory, and Dempster-Shafer belief theory. Here we consider only fuzzy set theory.

Fuzzy sets and conditional statements allow the system to manage heuristic rule-based knowledge, imprecise information from sensors, and the uncertainties in the knowledge about the environment. Also, fuzzy rule statements model the human expert's domain knowledge. Fuzzy logic rule evaluation involves only simple arithmetic calculations that can be performed very rapidly. Therefore, the computational time required to create a hazard map is quite manageable for a real-time decision system, making it feasible for landing operations [43].

For our tool we use the fuzzy reasoning approach put forth in [18]. Data uncertainty is handled through the level of membership to predefined datasets. The safety score s is represented with four linguistic labels {P, L, M, H}, which stand for *poor*, *low*, *moderate*, and *high*, respectively. The membership functions of the four terrain safety grades are shown in the top panel of Fig. 7. The parameters of these trapezoids are specified by a human expert.

The information flow in the fuzzy logic system for terrain safety assessment follows three basic stages. The terrain features first go through the fuzzification stage where their membership levels to predefined fuzzy sets are found. This information is then passed to the inferencing stage that consists of a set of linguistic rules for reasoning about the terrain safety based on the terrain features. These rules model the human expert's domain knowledge for terrain safety assessment expressed in plain English. Finally, the numeric (crisp) value of the terrain safety score is obtained in the defuzzification stage.

Safety scores are derived for each sensor separately. Camera, and lidar safety, s_C and s_L, respectively, are obtained by applying an FR set that maps terrain features to terrain safety. The linguistic labels for slope f_θ are {VF, F, I, VI}, which stand for very flat, flat, inclined, and very inclined, respectively. The linguistic labels for roughness f_e are {VS, S, R, VR}, which stand for very smooth, smooth, rough, and very rough, respectively. The linguistic labels for rock density (i.e. roughness) f_r are {VE, E, C, VC}, which stand for very empty, empty, crowded, and very crowded respectively. The terrain safety rules for each sensor are shown in the bottom center and bottom right panels of Fig. 7.

All rules in Fig. 7 are connected via the AND operator. For example, the first rule for the lidar is IF (f_θ^L is VI) AND (f_e^L is VR), THEN (s is P).

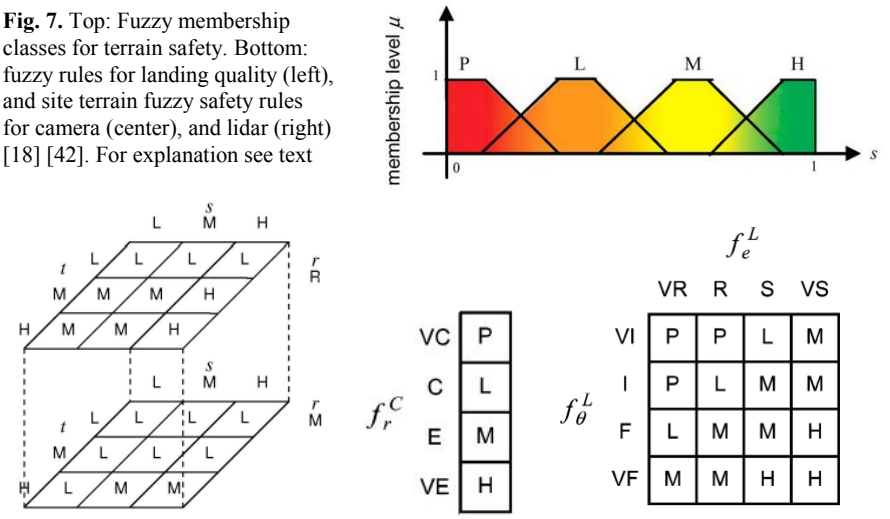

Fig. 7. Top: Fuzzy membership classes for terrain safety. Bottom: fuzzy rules for landing quality (left), and site terrain fuzzy safety rules for camera (center), and lidar (right) [18] [42]. For explanation see text

2.3.3 Landing reasoning

The goal of landing reasoning is to command a retargeting to a new landing site, if it is found by safety reasoning that the currently selected landing site is not suitable. To perform this decision, two additional factors apart from safety must be taken into account: the scientific interest of a new landing site, and its reachability according to the lander's descent trajectory, velocity and available fuel.

Landing site selection for a space exploration mission is generally a compromise between terrain safety and scientific return. When safety cannot be guaranteed, a potential site must be discarded regardless of its potential scientific impact. It is foreseen here to integrate several landing sites preferred by mission scientists before landing initialization, in order to influence the on-board site selection. These potential sites can then be used in conjunction with the on-board terrain safety assessment in order to select the best site during descent [42]. For the fuzzy rule determining scientific interest, four levels (or classes) of scientific return are considered: {N, L, M, H}, which stand for none, low, medium, and high, respectively. Before mission launch one of the regions is selected as the nominal landing site.

To assess reachability of a landing site, [42] suggest using ballistic analysis. This analysis gives an analytical equation of an elliptical landing footprint within which all points would be reachable by the lander in its current state. After simulations however we found that points which are designated as reachable by ballistic analysis are not always so when we tried to calculate a feasible trajectory using the guidance function (Ch. 2.4). We therefore updated the analytical equation for the landing ellipse, the new ellipse being significantly smaller than the one from [42].

For the respective fuzzy rule, three classes are considered: {U, M, R}, which stand for unreachable, marginally-reachable, and reachable: Points outside the original boundary are automatically unreachable. Points well within the landing ellipse are reachable. Points that lie close to the landing ellipse edge are marked as marginally-reachable.

2.3.4 Landing site "goodness" and retargeting

The three key factors for landing success (terrain safety, fuel consumption, and scientific return) can in turn be linked to a landing quality factor, or "goodness", using a new fuzzy rule set. To construct goodness map, the fuzzy safety and landing reasoning results are combined to a new fuzzy rule set, resulting in 144 rules that bring all maps into an interconnected relationship.

For defuzzification the terrain's linguistic values are again linked to specific numbers from 0 (very safe) to 8 (unsafe) that correspond to pre-defined landing site safety values.

If the initially chosen site is found to be unsafe a new landing site is selected and a retargeting commanded to the guidance function. Currently we simply retarget to the reachable spot with the highest goodness value. In the future we plan to apply the method used in [46], where the size of the largest convex cluster of good landing sites around a point is taken into account, i.e. the centers of larger convex areas of good landing sites are preferred.

2.4 Guidance and Control (G&C) Block

The general guidance problem can be formulated as a Two Point Boundary Value Problem (TPBVP), starting at the initial lander state and ending at the final lander state over the chosen landing site, within a specified time of flight and while respecting certain constraints. The aim of the guidance algorithm is then to solve for an acceleration profile that will take the vehicle from its initial state to the target state.

The vision-based navigation and hazard detection and avoidance technologies likely to fly on next-generation planetary landers strongly impact the guidance system. The most important emerging requirement is the need for path constraints to ensure robustness [21].

In [21] in total 24 guidance algorithms are listed and their suitability for precise and safe landing is extensively assessed. They looked into various categories of guidance algorithms including open loop, explicit/analytical, and numerical methods. They concluded that two guidance algorithms held the most promise: the primary choice was a guidance law based on convex optimization theory (Convex guidance). The secondary choice was an explicit guidance law called E-Guidance.

These results were verified by work in our institute [47]. [21] recommended a pre-screening of the selected algorithms, where a few candidates are implemented and compared for the particular test case under study. Following this logic we first implemented the easier to implement E-guidance. This algorithm however proved to be insufficient in various ways: it lacks an efficient way to implement descent angle constraints, it sometimes produces unfeasible trajectories, and the feasible divert distance is limited, particularly cross-range [48]. Identifying the above inadequacies, we implemented the more difficult to implement convex guidance as discussed below.

2.4.1 Convex guidance

Convex guidance was introduced in [49]. The particular implementation we are investigating here is called G-FOLD (Guidance for Fuel Optimal Large Divert) and is presented in [50] [51].

G-FOLD is an algorithm that is developed to compute, onboard in real-time, fuel optimal trajectories for large divert maneuvers necessary for planetary pinpoint or precision landing. The algorithm incorporates all relevant mission constraints and computes the global optimal trajectory. It 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 [50].

The goal of the planetary soft landing problem (depicted in Fig. 8) is to minimize fuel consumption (or equivalently maximize the remaining fuel at landing) (eq. (26)), subjected to the dynamics (eq.(27)) and state constraints (eq. (29)), while finding thrust inputs that take the spacecraft from an initial state to the target state, that are constrained in their minimum and maximum magnitude, as well as to their pointing (eq.(30)). The problem is to be solved based on given boundary conditions (eqs. (31) - (33)).

Problem 1 Non-Convex Minimum Fuel Landing Problem

$$\max_{t_f, \boldsymbol{T}_c} \; m(t_f), \text{ subject to:} \tag{26}$$

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + B\left(\boldsymbol{g} + \frac{\boldsymbol{T}_C(t)}{m}\right), \forall t \in [0, t_f] \tag{27}$$

$$\dot{m}(t) = -a\left\|\boldsymbol{T}_c(t)\right\|, \forall t \in [0, t_f], \tag{28}$$

$$x(t) \in \boldsymbol{X}, \forall t \in [0, t_f], \tag{29}$$

$$0 < \rho_1 < \|\boldsymbol{T}_c(t)\| \le \rho_2, \qquad \hat{\boldsymbol{n}}^T \boldsymbol{T}_c(t) \ge \|\boldsymbol{T}_c(t)\| \cos\theta, \tag{30}$$

$$m(0) = m_0 \tag{31}$$

$$\boldsymbol{r}(0) = \boldsymbol{r}_0, \qquad \dot{\boldsymbol{r}}(0) = \dot{\boldsymbol{r}}_0, \tag{32}$$

$$\boldsymbol{r}(t_f) = 0, \qquad \dot{\boldsymbol{r}}(t_f) = 0. \tag{33}$$

In the above problem X defines the state constraints and it is given by

$$\boldsymbol{X} = \{(\boldsymbol{r}, \dot{\boldsymbol{r}}) \in R^6 : \tan\gamma \boldsymbol{e}_1^T \boldsymbol{r} \ge \sqrt{(\boldsymbol{e}_2^T \boldsymbol{r})^2 + (\boldsymbol{e}_3^T \boldsymbol{r})^2}, \|\dot{\boldsymbol{r}}\| < V_{\max}, \left|\boldsymbol{e}_i^T \dot{\boldsymbol{r}}\right| \le V_h, i = 2,3\} \tag{34}$$

where V_{max} is maximum allowed velocity, V_h is the maximum allowable horizontal velocity, e_1 is the unit vector along the x-axis (vertical axis), e_2 and e_3 are unit vectors in y and z directions.

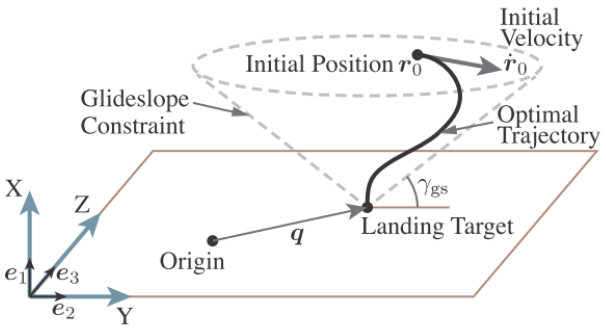

Fig. 8. Glideslope constraint in minimum landing error powered descent guidance problem. The glide slope constraint requires the spacecraft to remain in a cone defined by the minimum slope angle γ. In the minimum landing error case, the apex of the cone coincides with the landed position of the spacecraft, rather than the original target [50]

The control constraints in eq. (26) are non-convex. In order to convert the optimal control problem into an equivalent problem with convex control constraints, the "lossless convexification" method is to be applied.

The first step is to have convex relaxed control constraints.

Problem 2 Convex Relaxed Minimum Fuel Landing Problem

$$\max_{t_f, \boldsymbol{T}_c} \; m(t_f) \text{ , subject to (29), (31), (32) , and (33), and} \tag{35}$$

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + B\left(\boldsymbol{g} + \frac{\boldsymbol{T}_C(t)}{m}\right), \forall t \in [0, t_f] \tag{36}$$

$$\dot{m}(t) = -a\Gamma(t), \forall t \in [0, t_f] \tag{37}$$

$$x(t) \in \boldsymbol{X}, \forall t \in [0, t_f], \tag{38}$$

$$\|\boldsymbol{T}_c(t)\| \le \Gamma(t), \qquad 0 < \rho_1 \le \Gamma(t) \le \rho_2, \tag{39}$$

$$\hat{\boldsymbol{n}}^T \boldsymbol{T}_c(t) \ge \cos\theta\Gamma(t). \tag{40}$$

In the relaxed problem a slack variable 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. In summary, it can be shown that, since the relaxed control set includes the original one, and all the optimal solutions of Problem 2 satisfy the constraints of Problem 1, the optimal solutions of the relaxed problem are also optimal for the original one.

[52] presents a case study for the use of G-FOLD for landing inside of lunar pits. In this landing scenario that is similar to the one of EL, and even more demanding, the G-FOLD algorithm demonstrates impressive robustness.

3 Landing sim tool algorithm and tool development

The following table describes the algorithm we implemented for our landing simulation tool. The reader can refer to the respective chapters above to investigate the functions in the algorithm in more detail.

Algorithm 1 Landing sim tool algorithm

Initialize parameters

lander properties

Initial position and velocity (incl. uncertainties, not yet implemented)

Dry and propellant mass

Engine thrust and Isp

environmental parameters

Enceladus surface gravity

desired landing point

other function block parameters

SLAM (including SLAM update frequency)

guidance (including number of points for generated guidance trajecto ry. More points means more accurate trajectory, but longer calcula tion times)

HDA (including HDA frequency)

lander trajectory propagation options (including propagation step)

Pre-main loop operations

make map

load DEM (raw data)

simulate optical cam. input: apply shadows and texture to DEM, save DEM as (realistic looking) 3D terrain map

- simulate thermal cam. input: give “hotter” values to pixels near given plume sources, “colder” further away. Overlay resulting grey scale image on DEM
- simulate Lidar input: save DEM as greyscale 2D image.

calculate trajectory from initial lander state to nominal landing site (Convex guidance. Ch. 2.4.1)

- trajectory is discretized according to parameter chosen during initiali zation

run HDA for first time (in theory this step can be done off-line, from data obtained from orbit. HDA ran in the loop will then simply update these results)

- apply camera hazard mapping to pseudo optical image of terrain (eq. (25) (to entire image, camera FOV to be implemented)
- apply slope and roughness detection to DEM (eq. (23))
- calculate reachable ellipse (ballistic and guidance, Ch. 2.3.3) (ballistic might be useful in the future, as a constraint to the guidance prob lem, where the lander should not crash in non-nominal situations
- calculate scientific goodness of all map points, based on distance from pre-selected points of interest (plume sources)
- fuse all above generated maps to single landing site goodness map
- run retargeting function: if landing site un-safe, retarget to next best site (Ch. 2.3.4 (convex areas))
- output of HDA: new landing target site if retargeting commanded, and first hazard maps to be updated in the loop

Main loop (while altitude > 0, do)

perform Control function

- propagate lander trajectory according to thrust commanded in current guidance trajectory step

perform Navigation function (Ch. 2.2)

- Optical camera navigation
  - take screenshot of realistic looking 3D terrain map from real lander position and pose
  - extract SURF features from image
  - do SLAM
    - match features in current loop iteration with fea tures in previous iteration
    - correct lander and landmark position
    - initialize new and delete old landmarks
    - output: lander state estimation, and landmark po-sition estimation
- Lidar navigation
  - take screenshot of DEM elevation grayscale from real lander position and pose
  - extract SURF features from image
  - do SLAM (as in optical case)

output: lander state estimation, and LIDAR landmark position estimation

Thermal camera navigation

take screenshot of DEM overlaid with thermal image from real lander position and pose

extract SURF features from image

do SLAM (as in optical case)

output: lander state estimation, and thermal landmark position estimation

Inertial navigation

Simulated IMU measurements

State estimation fusion: EKF (not yet implemented)

fuse optical, LIDAR, thermal cam., and conventional nav. estimates to single lander state estimate, using EKF

perform HDA function (with frequency selected in input)

same function as run before loop

perform retargeting

If HDA commands, run Guidance function again to new landing site

Process, plot and save relevant output

Development of the tool is not yet complete. We have managed to implement the algorithm described above, albeit with the used functions in varying levels of maturity:

We have developed a first version of the **Terrain simulation block** as discussed in Ch. 2.1, and we are currently working on completing the described list of features. The implementation of rules for assigning terrain metadata (texture, temperature, others) are currently being investigated. Specularity properties of the terrain according to terrain type remain to be assigned, as well as realistic distributions of obstacles on the terrain. The implementation of shadows during terrain visualization is another important aspect we are currently working on. On sensor output simulation, the optical camera is already simulated. The Lidar simulation is currently under development. For the simulation of the thermal camera output, we are currently investigating the underlying physics to be simulated.

For the **TRN block**, we have adapted Joan Sola's open source SLAM Toolbox[9] written in Matlab [53]. Significant development of the TRN block is still needed to gain confidence in its results. A thermal camera or Lidar TRN algorithm also remains to be added in our tool. We plan to further investigate existing algorithms for fitness for our application, and implement the most promising.

A first version of the **hazard detection and avoidance block** has also been developed. Camera hazard, Lidar slope and roughness hazard maps have been implemented. These sensor hazard maps are then fused to an overall safety map, using fuzzy rulesets, as described in Ch. 2.3.2. This was done using Matlab's Fuzzy

[9] http://www.iri.upc.edu/people/jsola/JoanSola/eng/toolbox.html

Logic Toolbox[10]. A modified reachability ellipse as discussed in Ch. 2.3.3 can also be generated to result in the final fused hazard map. Since the shadowing and Lidar output simulation tools are currently missing from the terrain generation block, rock/ice block hazard detection by the optical camera could not be implemented here. This is a point we are currently working on, in parallel with the development of the aforementioned terrain generation block features. Similarly, thermal camera hazard detection has also not been yet implemented. We are currently investigating ways to take the scientific interest factor into account, in order to create the final landing quality map. We are also considering implementing multiple decision engines and fusing their output to get an improved hazard detection function, as proposed in [18].

For the **Guidance block** we implemented verified the G-FOLD algorithm in Matlab as described in [50][52] . To solve the SOCP problem we used the freely available FALCON tool[11], developed by TU Munich. To validate the implemented guidance algorithm and to demonstrate its capabilities, we recreated the results of [51]. The landing scenario presented there is especially challenging. Still our results agreed with those of [51]. The guidance algorithm performs satisfactorily, and we consider its development finalized for the purposes of our tool.

4 Outlook and future work

We have presented our autonomous planetary landing simulation tool, as will be applied to simulate a landing near a plume source in a tiger stripe canyon on the South Polar Terrain of Saturn's moon Enceladus. Development of the tool is ongoing: significant work has been done on each of the blocks constituting the tool (Terrain simulation, Terrain relative navigation, Hazard detection and avoidance, and Guidance and Control).

We aim to use this tool to simulate a landing on Enceladus with adequate fidelity, in order to demonstrate its feasibility and validate that the accuracy and safety requirements for this exceptionally challenging terrain are met. We also plan to use the tool to investigate novel approaches and propose improvements in all aspects of the landing process.

References

[1] J.S. Kargel, Enceladus: cosmic gymnast, volatile miniworld., Science. 311 (2006) 1389–1391. doi:10.1126/science.1124495.

[10] https://de.mathworks.com/products/fuzzy-logic/

[11] http://www.fsd.mw.tum.de/software/falcon-m

[2] C.C. Porco, P. Helfenstein, P.C. Thomas, A.P. Ingersoll, J. Wisdom, R. West, et al., Cassini observes the active south pole of Enceladus., Science. 311 (2006) 1393–1401. doi:10.1126/science.1123013.
[3] J.H. Waite Jr, W.S. Lewis, B. a. Magee, J.I. Lunine, W.B. McKinnon, C.R. Glein, et al., Liquid water on Enceladus from observations of ammonia and 40Ar in the plume, Nature. 460 (2009) 487–490. doi:10.1038/nature08153.
[4] D.A. Patthoff, S. a. Kattenhorn, A fracture history on Enceladus provides evidence for a global ocean, Geophys. Res. Lett. 38 (2011) n/a–n/a. doi:10.1029/2011GL048387.
[5] 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) 909–19. doi:10.1089/ast.2008.0265.
[6] 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).
[7] B. Giese, P. Helfenstein, P.C. Thomas, A.P. Ingersoll, J. Perry, G. Neukum, et al., The morphology of an active zone near Enceladus ' south pole and implications, 12 (2010) 11085.
[8] 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.
[9] 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.
[10] B. J. Hanna et al., Free Molecular and Collisional Studies of Enceladus' Water Vapor Plumes, 40th Lunar Planet. Sci. Conf. (2009).
[11] J.R. Spencer, J.C. Pearl, M. Segura, F.M. Flasar, A. Mamoutkine, P. Romani, et al., Cassini encounters Enceladus: background and the discovery of a south polar hot spot., Science. 311 (2006) 1401–1405. doi:10.1126/science.1121661.
[12] 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.
[13] O. Abramov, D. Raggio, P.M. Schenk, J.R. Spencer, TEMPERATURES OF VENTS WITHIN ENCELADUS' TIGER STRIPES, 46th Lunar Planet. Sci. Conf. (2015).
[14] NSF Award Abstract Web-page: MIDGE: Minimally Invasive Direct Glacial Exploration of Biogeochemistry, Hydrology and Glaciology of Blood Falls, McMurdo Dry Valleys, (n.d.).
[15] 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.
[16] K. Konstantinidis, C. Hager, Enceladus Scenario and Operations – Operations Concept, 2014.
[17] Europa Study 2012 Report, Europa Lander Mission, NASA JPL, 2012.
[18] H. Seraji, N. Serrano, A multisensor decision fusion system for terrain safety assessment, IEEE Trans. Robot. 25 (2009) 99–108. doi:10.1109/TRO.2008.2006705.
[19] J. Delaune, D. De Rosa, S. Hobbs, Guidance and Control system design for Lunar Descent and Landing, AIAA Guid. Navig. Control Conf. (2010) 1–10. doi:10.2514/6.2010-8028.
[20] E. Zaunick, D. Fischer, 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 June 21th, 2012.
[21] I. Gerth, E. Mooij, Guidance for Autonomous Precision Landing on Atmosphereless Bodies (under Review), AIAA J. (2014).
[22] A.I. Razzaghi, ed., Enceladus Flagship Mission Concept Study, NASA Goddard Space Flight Center, 2007.
[23] G.S.P. Miller, The Definition and Rendering of Terrain Maps, J. ACM. 20 (1986).
[24] J. Olsen, Realtime procedural terrain generation, Dep. Math. Comput. Sci. (…. (2004) 20. doi:10.1.1.366.6507.
[25] A.M.S. and A.M.T. Wasan R. Al-Azawi, MODELING AND SIMULATION FOR THE SENSOR PERFORMANCE OF THE THERMAL CAMERA, (2015).
[26] A.E. Johnson, J.F. Montgomery, Overview of Terrain Relative Navigation Approaches for Precise Lunar Landing, 2008 IEEE Aerosp. Conf. (2008) 1–10. doi:10.1109/AERO.2008.4526302.

[27] J. Solà-Ortega, Towards Visual Localization, Mapping and Moving Objects Tracking by a Mobile Robot: a Geometric and Probabilistic Approach - PhD Thesis, 2007.
[28] J. Sola, Simulataneous localization and mapping with the extended Kalman filter, Unpubl. Available Http//www. Joansola. eu/JoanSola/eng/JoanSola. Html. (2013) 1–35.
[29] H. Bay, T. Tuytelaars, L. Van Gool, A. Leonardis, H. Bischof, A. Pinz, SURF: Speeded Up Robust Features, Comput. Vis. ECCV 2006. 3951 (2006) 404–417. doi:10.1007/11744023_32.
[30] Lowe, Distinctive Image Features from Scale-Invariant Keypoints, (2004).
[31] E. Rublee, G. Bradski, ORB - an efficient alternative to SIFT or SURF, (2011). doi:10.1109/ICCV.2011.6126544.
[32] M. Ciarambino, P. Lunghi, L. Losi, M. Lavagna, Development, Validation and Test of Optical Based Algorithms for Autonomous Planetary Landing, Proc. ICATT 2016. (n.d.).
[33] S. Vidas, S. Sridharan, Hand-held monocular SLAM in thermal-infrared, 2012 12th Int. Conf. Control. Autom. Robot. Vision, ICARCV 2012. 2012 (2012) 859–864. doi:10.1109/ICARCV.2012.6485270.
[34] P. Christiansen, K. a R. Steen, R.N.Y. Jørgensen, H. Karstoft, Automated detection and recognition of wildlife using thermal cameras, Sensors (Basel). 14 (2014) 13778–13793. doi:10.3390/s140813778.
[35] L. Walker, D. Spencer, L. Walker, Thermal Imaging and Rapid Feature Detection for Small Satellites, (2010).
[36] N. Welsh, LIDAR Feature Extraction - Lecture, (2010).
[37] Michael Bosse and Robert Zlot, Keypoint design and evaluation for place recognition in 2D lidar maps., Rob. Auton. Syst. 57 (2009) 1211–1224.
[38] Y. Li, E.B. Olson., A General Purpose Feature Extractor for Light Detection and Ranging Data., Sensors. 10 (2010) 10356–10375.
[39] Y. Na, Y. Jung, H. Bang, FLASH-LIDAR BASED TERRAIN RELATIVE NAVIGATION FOR AUTONUMOUS PRECISION LUNAR LANDING, (n.d.) 1–9.
[40] A.E. Johnson, T.I. Ivanov, Analysis and Testing of a LIDAR-Based Approach to, (n.d.) 1–10.
[41] J.. Lewis, Fast Template Matching Vision Interface, in: Vis. Interface, Quebec City, CA, 2015: pp. 120–123.
[42] N. Serrano, H. Seraji, Landing Site Selection using Fuzzy Rule-Based Reasoning, (2007) 10–14.
[43] 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. doi:10.1109/AERO.2006.1655795.
[44] a. Howard, H. Seraji, Multi-sensor terrain classification for safe spacecraft landing, IEEE Trans. Aerosp. Electron. Syst. 40 (2004) 1122–1131. doi:10.1109/TAES.2004.1386868.
[45] A. Huertaslyang, C.W. Madisongjpl, A. Huertas, Y. Cheng, R. Madison, Passive Imaging Based Multi-cue Hazard Detection for Spacecraft Safe Landing ESTIMATIONqualified, (2006).
[46] C. Perez-Montenegro, PhD Thesis: Navigation, Guidance and Control for Planetary Landing, Politecnico di Torino, (2014). doi:10.6092/polito/porto/2557338.
[47] P. Dykta, Algorithms and sensor systems for planetary landing on Airless bodies - MSc Thesis, (2013).
[48] M. Thies, Simulation of Guidance and Hazard Detection and Avoidance for Autonomous Planetary Landing - MSc Thesis, (2016).
[49] B. Acikmese, S.R. Ploen, Convex Programming Approach to Powered Descent Guidance for Mars Landing, J. Guid. Control Dyn. 30 (2007) 1353. doi:10.2514/1.27553.
[50] B. Acıkmese, 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.
[51] B. Açıkmeşe, L. Blackmore, Lossless convexification of a class of non-convex optimal control problems for linear systems, Am. Control Conf. (ACC), 2010. (2010).
[52] N.S. Bhasin, W.R. Whittaker, C.G. Atkeson, Fuel-Optimal Spacecraft Guidance for Landing in Planetary Pits - PhD Thesis, (2016).

[53] J. Solà, T. Vidal-calleja, J. Sol, V.J. Civera, Impact of landmark parametrization on monocular EKF-SLAM with points and lines, Int. J. Comput. Vis. (2013).