

# Vision-Based Navigation Using Map Relative Localization for Crewed Lunar Landing Missions

Abhinav Kamath\*,† and Jonathan Boyd\*

\*Department of Mechanical & Aerospace Engineering, University of California, Davis, CA 95616

†Center for Human/Robotics/Vehicle Integration and Performance, University of California, Davis, CA 95616

**Abstract**—With human missions to the Moon on the horizon, technology used to navigate and land safely is of critical importance. This paper reviews the state-of-the-art for terrain relative navigation (TRN) and its implementation on lunar landers. We propose a simulation of a lunar lander that employs map relative localization to achieve terrain relative navigation. We use this simulation to test the effect of map-relative localization errors on propellant consumption projections to land on the lunar surface, by implementing a fuel-optimal guidance algorithm. We then use a course-correction maneuver simulation to obtain precision landing guidance solutions, and to evaluate performance metrics in terms of the total propellant consumption and the minimum feasible time-of-flight.

**Index Terms**—lunar lander, terrain relative navigation, map relative localization, fuel-optimal guidance, precision landing, course-correction maneuvers

## I. INTRODUCTION

Ever since the Apollo missions, mankind has dreamed of returning to the Moon. The dream might become a reality in the near future with human lunar landing missions on the horizon. The new opportunity to put humans on the Moon presents the same problem that was faced by the Apollo missions - to accurately navigate in a GPS-denied environment. The Apollo missions relied on inertial measurement units (IMU) for approximate localization and extremely skilled pilots to safely land the spacecraft. Modern technology allows the implementation of more accurate navigation and autonomous landing strategies. The Altair Lunar Lander developed by NASA JPL in 2010 serves as a good example for potential sensors and systems on future landers.

Figure 1 shows that future landers will have far more sensors than just an IMU. The need for additional sensors arises from the fact that IMUs measure acceleration to estimate position. The measured acceleration is integrated twice to obtain position estimates, causing the error in estimation to grow quadratically. Employing a vision-based approach to navigation would help increase the accuracy of position estimation. Since terrain relative navigation (TRN) using visual-odometry measures the position directly, the growth in error is linear, as opposed to quadratic.

There are numerous other methods for implementing improved navigation, but the focus of this paper will be terrain relative navigation (TRN), implemented using a modified version of NASA JPL's Lander Vision System (LVS) that has been developed for the Mars 2020 mission. This system implements a monocular vision system with a Doppler Radar.

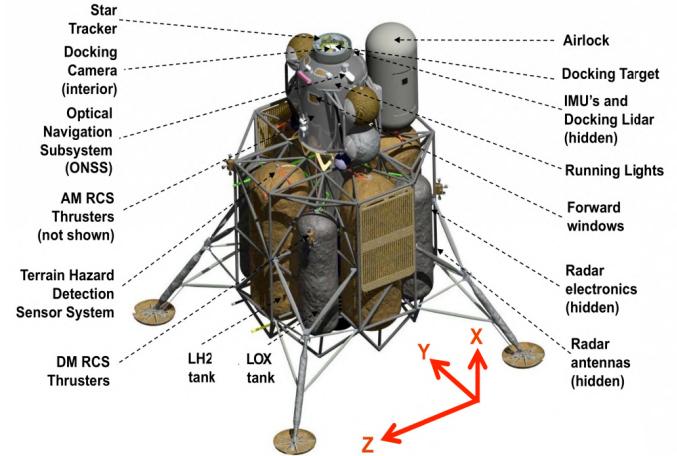


Fig. 1. Systems on the Altair Lunar Lander [1]

The Doppler Radar is used for altimetry, while the monocular vision system is used for egomotion, which is a method for computing the 3D motion of a camera within an environment [2]. The IMU and Radar can be used to seed the vision-based localization algorithm with the approximate initial position. A simulated vision system will be used to evaluate the impact of localization accuracy on propellant consumption projections.

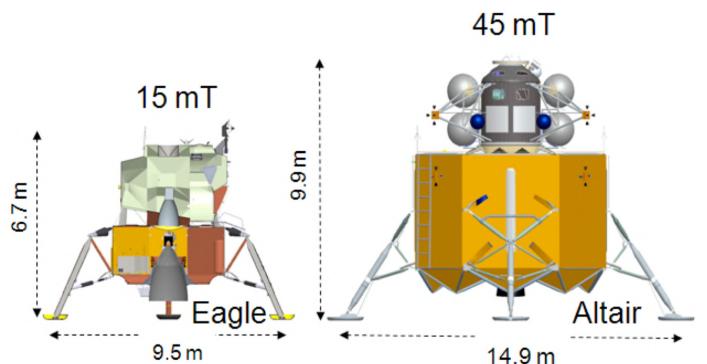


Fig. 2. The Apollo Eagle Lunar Lander & The Altair Lunar Lander [1]

## II. LITERATURE REVIEW

For our research question, we investigated localization sensors and algorithms to match the requirements of our chosen lunar lander vehicle model and mission.

### A. Localization Sensor Selection

Before the JPL monocular vision system was selected for this simulation, other egomotion strategies were reviewed. The starting point was IMU-only position estimation using dead-reckoning. This method is beneficial because it uses no additional sensors and calculations are simple and quick. Acceleration is simply integrated twice to get position, which results in a quadratically growing error in position estimation. To put this into perspective, the initial position estimation error for the Mars 2020 mission is expected to be 3.2 km before the vision system is used to reduce this error [3]. This error leads to extremely large landing ellipses which are not suitable for robotic or human spaceflight missions that require precision landing. Figure 3 shows the landing ellipses of Mars missions, to demonstrate the impact of position-estimation errors.

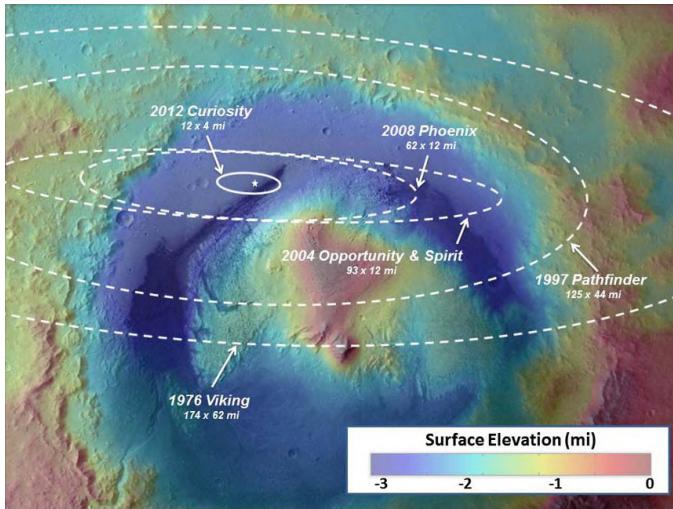


Fig. 3. Landing ellipses of Mars missions [4]

The next intuitive solution is the use of the Global Positioning System (GPS); however, this requires a vast network of satellites in orbit around the celestial body that the mission is exploring. There are no plans to establish such a network around the Moon in the near future. This makes the Moon a GPS-denied environment. Although there is no GPS network on the Moon, extensive lunar surface imaging and mapping are available from orbiters, such as the Lunar Reconnaissance Orbiter (LRO) [5]. This points to a TRN-based solution to localization because extensive feature databases and global maps can be generated prior to landing. The engineers behind the Altair Lunar Lander reached a similar conclusion, and a CAD model of the proposed optical navigation sensor is shown in figure 4.

Once a TRN approach to egomotion has been selected, it is necessary to select the appropriate sensor suite to implement. The decision comes down to either an active range sensor or a passive image sensor [6]. The active range sensor being considered is a 2D or 3D LIDAR system. The LIDAR benefits from robustness to illumination conditions to the point where it can be used with no light source. The main drawbacks of the LIDAR system are maturity, packaging and power consumption. Compared to a camera sensor, LIDAR is a less

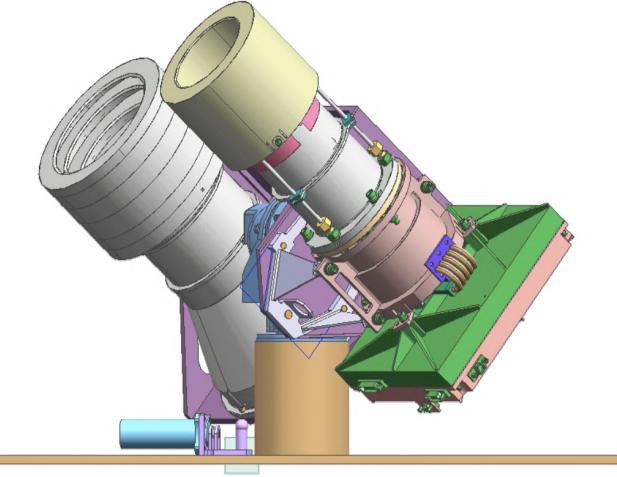


Fig. 4. Altair Optical Navigation Sensor System [1]

mature technology, and currently no LIDAR systems are rated at a technology readiness level (TRL) of nine for space travel [4]. Nine is the highest rating, and it means that the technology is flight proven. In contrast, LIDAR systems are only rated to a maximum TRL of four. This means that LIDAR systems have only been tested in laboratory settings. A LIDAR system is also larger and heavier than a camera sensor. This large mass increases the cost of launch significantly. In addition, LIDARs also consume more power.

**Precise Landing**  
**Terrain Relative Navigation (TRN)**

**Passive-Optical/Camera-Based**  
(requires illuminated terrain: applicable to most missions)

- **JPL Lander Vision System (LVS):** camera + IMU + dedicated computing **to be TRL 9 with Mars2020**
- **TRN solutions also available from APL, Draper & elsewhere** **in dev for multiple mission concepts**
- **JPL Intelligent Lander System (ILS)**  
**in dev for Europa Lander concept**

**HEO/STMD / SMD/STMD/other**

**Active/Lidar-based TRL 3-4**  
(dark/shadowed or illuminated terrain)  
**dev & tested in ALHAT**

Fig. 5. Current sensor systems that NASA employs for TRN [4]

The main downside of a camera based system is the need for illumination. This constraint does not affect our chosen application because all proposed manned missions to the Moon in the near future involve landing on the near side of the Moon. The impact of varied illumination intensity can also be mitigated by using advanced localization algorithms. Therefore, a camera based system was chosen for this project. Specifically, a monocular camera was chosen, in order to lower the mass of the system. The use of a monocular camera sensor does introduce a scaling problem as the camera system cannot

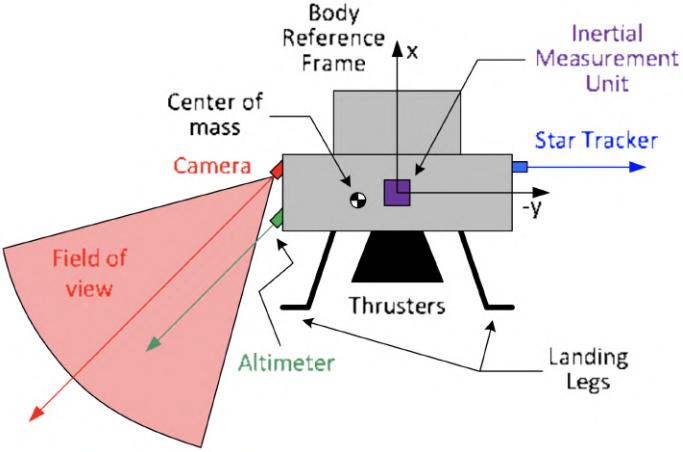


Fig. 6. Sensor placement on a generic lunar landing vehicle [7]

distinguish a far away large object from an up close smaller object. This problem can be addressed with either an additional sensor for altimetry or the use of advanced image-processing techniques.

#### B. Localization Algorithm Selection

Once a camera-based system has been chosen, it is necessary to choose an algorithm to process the captured image for visual-odometry. Image processing algorithms fall into two categories: algorithms to estimate position and attitude, and algorithms to estimate velocity (Table I). Four position estimation algorithms and two velocity estimation algorithms have been investigated in this paper. The position estimation algorithms are the *Crater Pattern Matching for Position Estimation (CPMPE)*, the *Scale Invariant Feature Transform (SIFT) Pattern Matching for Position Estimation*, *Onboard Image Reconstruction for Optical Navigation (OBIRON) - Surface Patch Correlation for Position Estimation* and *Image to Map Correlation for Position Estimation*. The velocity estimation algorithms are the *Descent Image Motion Estimation Subsystem (DIMES) - Consecutive Image Correlation for Velocity Estimation* and *Structure From Motion - Consecutive Image Correlation for Velocity and Attitude Rate Estimation* [6].

Algorithm	Required Inputs	Output Estimate
Crater Pattern Matching	Descent image Crater landmark DB	Absolute position Attitude
SIFT Pattern Matching	Descent image SIFT landmark DB	Absolute position Attitude
OBIRON Surface Patch Correlation	Dense image Descent image Attitude Altitude	Absolute position Attitude update
Image to Map Correlation	Map image Descent image Attitude Altitude	Absolute horizontal position

TABLE I  
COMPARISON OF POSITION LOCALIZATION ALGORITHMS [6]

Crater Pattern Matching for Position Estimation (CPMPE) was the algorithm originally chosen for our project. However, due to the complexity of implementing an encoded crater database, a simpler algorithm was chosen. The CPMPE algorithm works by first creating a crater database from previously collected images of the landing site. The system then compares descent images to this database, in order to determine the global position of the lander. This method can also be used to find the local position by comparing the crater locations in consecutive descent images. The crater pattern matching algorithm is convenient, as it returns position and attitude estimates without the need for additional sensors. It also is robust to varying lighting conditions. This robustness is due to the dark/bright pairing shown in figure 7. The main issue with the algorithm is that it mandates the existence of craters in the vision-field of the camera sensor along the flight path of the lander. This limitation would not be a concern for most lunar missions [8].

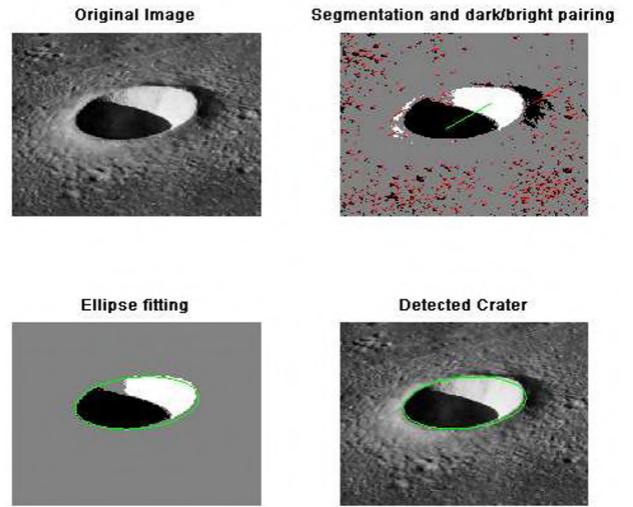


Fig. 7. Steps of crater detection algorithm [9]

Three other algorithms were considered for the project. The first of these algorithms was SIFT Pattern Matching for Position Estimation. It is identical to Crater Pattern Matching except that the features are generalized [10]. This generalization comes at the cost of increased sensitivity to light intensity and view angle. Because the Moon is heavily cratered, the generalized approach is not necessary. Next, OBIRON was investigated. The algorithm uses multiple overlapping orbital images to generate a 3D model of several patches of ground. The 3D models are then used to simulate lighting conditions and generate a simulated image of the ground patch. The descent images are then compared to the simulated image to get position estimates. This method was not selected because it requires altitude and attitude measurements in order to generate the simulated image [11]. The final algorithm considered was Image to Map Correlation for Position Estimation. A simple version of this algorithm was implemented for our simulation. This method first rectifies

the descent image to the same scale and orientation. Then sections of the descent image are compared to the orbital image. This method is currently used on cruise missiles. Map Correlation was not chosen because it requires altitude and attitude information for image rectification [12], [13].

Although the velocity of the vehicle can be determined by taking multiple position estimates over a known period of time, velocity estimation algorithms provide faster results without the need to reference the global map or feature database. This is done by comparing consecutive images and converting the image-shift to velocity. Although we considered crater matching for consecutive images to determine velocity, we determined that getting velocity from the image based algorithm was not necessary for our simulation. The other two methods previously mentioned have been extensively tested: *DIMES* on Mars rovers and *Structure from Motion* on a helicopter testbed [6].

### C. Guidance

Once the position and velocity of the lander are estimated within a range of ‘nominal’ error values, a guidance algorithm is needed to determine the optimal trajectory to the landing site. A multitude of guidance algorithms have been developed to implement fuel-optimal landing. Optimal Control theory has been extensively studied and applied in this regard [14]. The proposed guidance methodology for JPL’s Altair Lunar Lander involved Optimal Control solutions for fuel-efficient trajectory generation [15]. Modern spacecraft landing guidance algorithms rely heavily on Convex Optimization, which ensures a global optimal solution, given that the formulated problem is tractable and that it satisfies a set of well-defined rules [16]. SpaceX uses Convex Optimization in real-time for autonomous precision landing of the first-stage boosters of the Falcon 9 and Falcon Heavy rockets [17]. Engineers at JPL and SpaceX co-developed a real-time implementable fuel-optimal guidance algorithm for planetary pinpoint landing, called the G-FOLD algorithm [18]. As for Earth-based applications, real-time Convex Optimization has been employed to perform constrained motion planning for quadrotors, and this technology is pivotal in being able to operate high-performance aerial robots in populated spaces [19].

## III. PROBLEM FORMULATION

### A. The Lunar Lander

NASA JPL’s proposed vehicle, the Altair Lunar Lander, will be considered as our model lander.

The lander undocks from the Orion crew capsule in the Low Lunar Orbit (LLO), which is a 100 km circular orbit, and performs the Descent Orbit Insertion (DOI) maneuver. This marks the beginning of the descent phase. This maneuver puts the lander on an elliptical orbit that has a perilune (point on the orbit closest to the lunar surface) of 15.24 km (50,000 ft).

The powered-descent phase of lunar landing commences at this altitude, with the Powered Descent Initiation (PDI), and

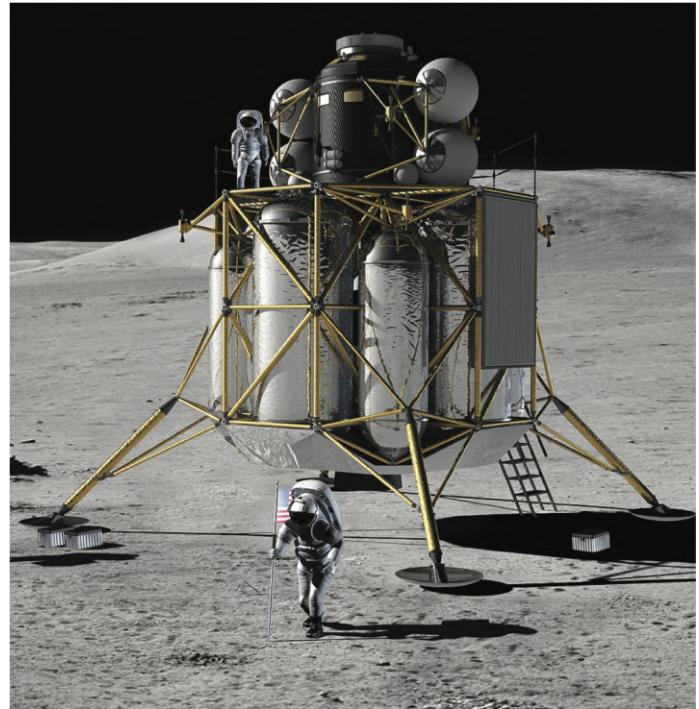


Fig. 8. A depiction of the Altair Lunar Lander on the Moon. Credit:NASA

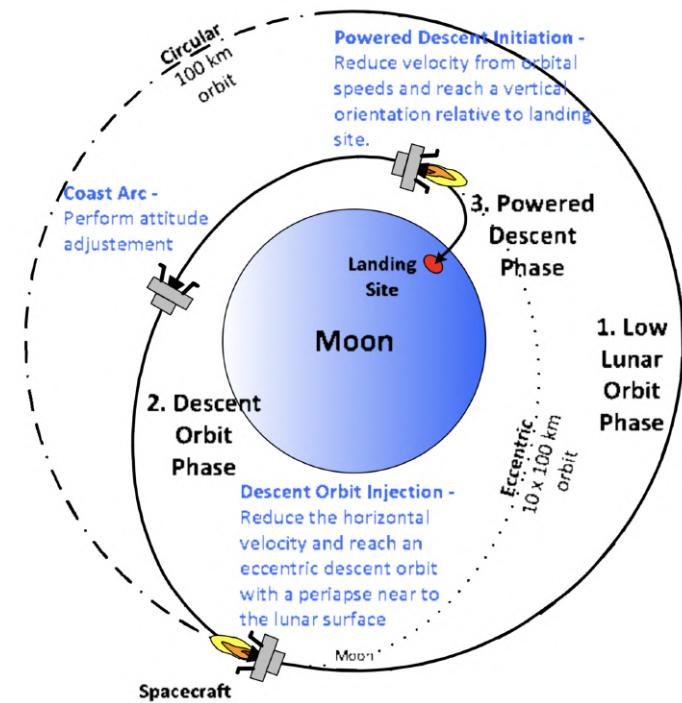


Fig. 9. The phases of lunar descent and landing [7]

consists of one continuous, throttled burn, until touchdown [20]. The subphases of powered-descent are as follows:

- 1) The Braking Burn (BB) Maneuver
- 2) The ‘Pitch-Up’ Maneuver
- 3) The Approach (visibility) Phase
- 4) The Terminal-Descent (TD) Phase

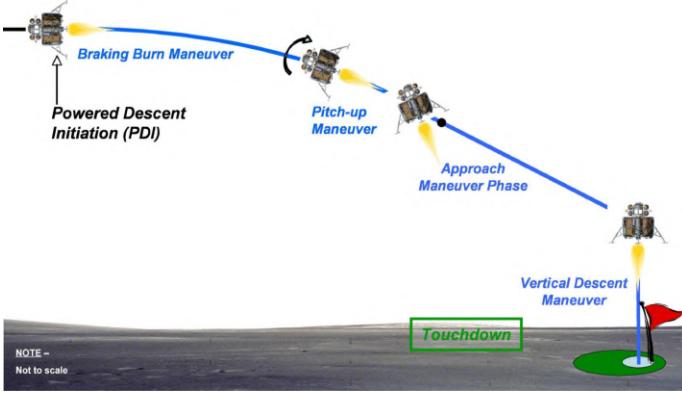


Fig. 10. Lunar Descent Subphases of NASA JPL’s Altair Vehicle. [21]

*1) The Braking Burn (BB) Maneuver:* The braking burn is initiated at the perilune of the descent orbit, which is at an altitude of 15.24 km. The main engine is fired at 92% full engine thrust, with the engine aligned with the lander’s velocity vector. This thrust margin is required to eliminate potential dispersions during the braking burn.

*2) The ‘Pitch-Up’ Maneuver:* After completion of the braking burn, the lander commences the ‘pitch-up’ maneuver to get the lander “nearly vertical”. The end of this phase marks the beginning of the approach (visibility) phase.

*3) The Approach (Visibility) Phase:* The engine is fired between 60% and 40% full engine thrust in the approach phase. Redesignation of the target landing site, if required, occurs in this phase. The approach subphase ends at an altitude of 30 meters directly above the final touchdown site, 30 seconds before touchdown.

*4) The Terminal-Descent (TD) Phase:* The TD phase is the final subphase in lunar landing (Fig. 11). It begins at 30 meters directly above the target landing site and continues the burn to get the lander to descend at a constant downward velocity of 1 m/s for the final 30 seconds. This ‘vertical-only’ descent would help mitigate instability effects relating to fuel-slosh in the tanks. The shutdown sequence is initiated 1 m above the surface to mitigate regolith kick-up at landing.

#### B. The Human Factor

Safety of the crew is of utmost importance in any human spaceflight mission. This hard requirement on safety drives the design of the lander and the landing trajectory. For example, the ‘pitch-up’ maneuver is executed at the beginning of the approach phase to provide the crew with better visibility to detect hazards around the landing site, giving them the ability to override the vehicle autonomy at their discretion. This maneuver would involve a change in the attitude of the spacecraft once the landing site enters the field-of-vision of the on-board camera sensor.

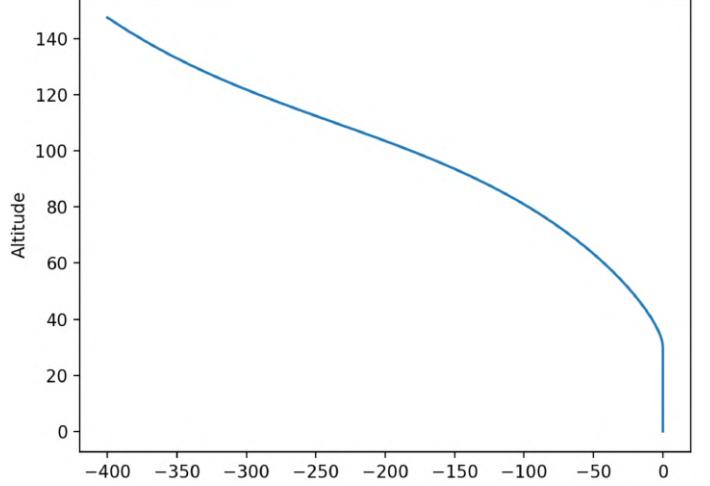


Fig. 11. Fuel-optimal trajectory generated by solving a convex optimization problem, depicting “vertical-only” terminal descent

#### C. Navigation

We propose a modified version of JPL’s Lander Vision System (LVS) for lunar landing missions.

Navigation of the lunar lander would be based on map relative localization (MRL). This method of localization requires an on-board lunar map. Images captured by the monocular camera sensor would then be matched with the on-board lunar map to get position estimates. With every captured image, the localization error reduces, and the on-board map is cropped accordingly to speed-up computation [3].

We also propose a novel approach to precision landing. The approach would involve storing an image of the landing site along with the on-board map. This would require *a priori* knowledge and selection of the target landing site, which can be made possible by means of an orbiting spacecraft such as the Lunar Reconnaissance Orbiter [5]. For the upcoming Artemis missions, this information can be collected by the Gateway [22].

The Mars 2020 mission will use the Lander Vision System (LVS) for lander state-estimation. The map relative localization (MRL) approach to terrain relative navigation (TRN) begins with a initial position estimate seeding from the on-board IMU. Mars 2020 will commence landing with a parachute phase, and the initial error is estimated to be around 3.2 km [3]. The LVS would then capture a series of six images, the first three for ‘coarse’ landmark matching and the last three for ‘fine’ landmark matching. With each image captured and processed, the localization error reduces, and it is estimated to reach a ‘nominal’ error margin of 40 m after six images, all in a time-frame of 10 seconds.

We presume that localization is performed by the lunar lander until the target landing site is detected by the on-board camera sensor (twice, in our simulation). This would require on-board storage of an *a priori* image of the site. At this point, the lander, which is now within a ‘nominal’ range of localization error values, will halt the localization procedure

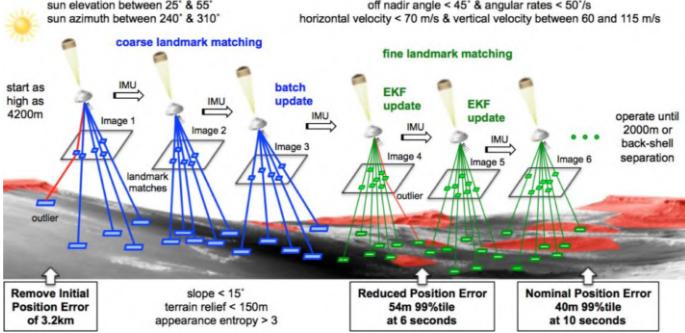


Fig. 12. Map Relative Localization (MRL) on the Mars 2020 lander [3]

and switch to precision landing guidance.

#### D. Guidance

The error in position estimation at the final localization point manifests as the error in the initial position in the landing guidance algorithm. Since the target landing site is now in the field-of-view of the camera sensor, the localization error becomes known to the lander. The trajectory to the target landing site is generated on-board, and this will inherently correct the positional error of the lander.

#### E. Objectives

In this project, we propose to develop a simulation to demonstrate vision-based terrain relative navigation (TRN) and fuel-optimal guidance for crewed lunar landing missions. The primary goal of this project is to answer the following questions:

- 1) How does the map-relative localization error affect propellant consumption projections?
- 2) How does the choice of the course-correction maneuver initiation point affect the total propellant consumption and the minimum feasible time-of-flight?

## IV. PROPOSED SIMULATION

### A. Simulation Environment

1) *Embodiment*: Embodiment of the simulated lander is captured by the physical limitations that are considered, such as its inertial properties and thrust bounds.

2) *Situatedness*: The situatedness of the simulate lander is significant as the camera sensor used for TRN is fixed relative to the spacecraft reference frame. This camera sensor needs to be facing the lunar surface for TRN to be effective.

#### 3) Assumptions:

- For this project, we consider only the monocular camera sensor for localization.
- The lander has one gimbaled main engine thruster, for simplicity of the control architecture.
- The target landing site is assumed to be obstacle-free and safe, as hazard detection and avoidance would involve a reactive control scheme in tandem with the deliberative control scheme, thus making the overall architecture hybrid.

- We assume that the lander tracks the guidance trajectory perfectly for the purposes of this project, as realistic trajectory tracking would involve complex low-level control algorithms.

#### 4) Constraints:

- The main engine can throttle between a minimum and a maximum value of thrust, and this is a limitation of the engine. Once the powered-descent phase commences, a continuous burn is initiated, and the engine cannot be turned off until touchdown. The lower-bound on thrust adds a non-convex constraint in the guidance algorithm. Lossless convexification of this non-convex constraint is needed to make the Convex Optimization problem tractable [23].
- The requirement that the landing trajectory should not involve surface impacts or subsurface flight, adds a glideslope constraint to the guidance problem. [23].
- The lander is constrained to a ‘vertical-only’ terminal-descent trajectory for the last 7 seconds of landing, in order to avoid instabilities due to fuel slosh and to ensure that the lander does not tip over on touchdown. The engine is turned off 1 meter above the lunar surface to mitigate regolith kick-up. The landing legs are designed to meet this requirement [15].

### B. Robotic Architecture

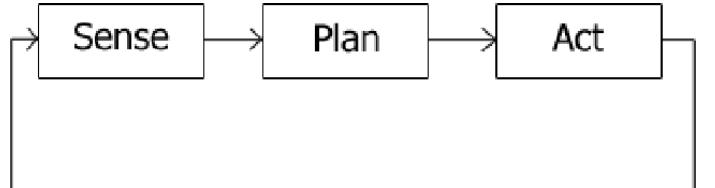


Fig. 13. The Deliberative Control Paradigm [24]

This project will implement a deliberative control architecture:

1) *Sense*: The lander ‘senses’ (estimates) its position using the on-board sensors and map relative localization (MRL), subject to localization errors. With consequent map relative localization, the position-estimation error is reduced.

2) *Plan*: After capturing and processing the fixed number of images and detecting the target landing site, the fuel-optimal landing guidance trajectory is generated. This is analogous to robotic path-planning.

3) *Act*: The lander follows the fuel-optimal trajectory by firing its thruster based on the input thrust commands and control system feedback, and lands on the lunar surface.

## V. NAVIGATION

The first part of our simulation involves map-relative localization of the lunar landing vehicle. This process involves an on-board lunar map and a high-fidelity environment to simulate the actual Moon. The descent imagery from the lunar lander is then matched with the on-board map using a robust template matching algorithm, in order to achieve

localization. In order to determine the density of craters in the chosen descent-patch, a pre-trained Convolutional Neural Network (CNN) is utilized. The obtained position estimates that are subject to localization errors are then used to update the guidance algorithm.

#### A. The Environment

A high-fidelity image of the Moon (10000x10000 [px]) from Lunar Reconnaissance Orbiter (LRO) imagery was considered to be the actual Moon for the purpose of the simulation. A screen capture of the image is shown in figure 14. One pixel on this image corresponds to 173.74 meters on the lunar surface. A descent patch is then chosen for the purposes of descent imagery.



Fig. 14. Screenshot of the high-resolution LRO image used to simulate the lunar environment in our simulation

#### B. The Sensor

The monocular camera sensor is chosen to have a field of view (FoV) of 179.8x179.8 [degrees] to ensure that the descent imagery contains sufficient information to run the navigation algorithms. We assume that the sensor has a resolution of 160x160 [px], to ensure fast map-relative localization computation. To ensure that the captured image closely represents a realistic image taken from a monocular camera sensor on a physical lunar landing vehicle, we assume that descent imagery is subject to Gaussian noise and motion-blur. We also assume that the descent image is not oriented with respect to the map. An example of simulated descent imagery is shown in 15.

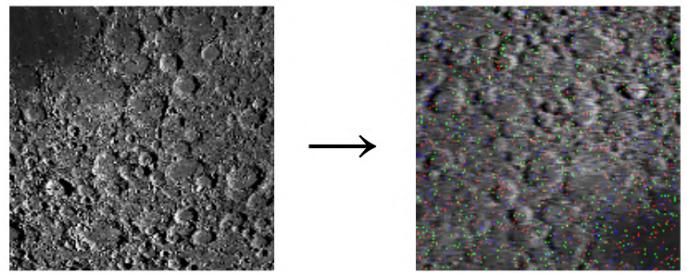


Fig. 15. Simulated Descent Imagery

#### C. The Map

The on-board map chosen for the simulation is an LRO image with a resolution of 1400x1400 [px] and a scale of 1.241 km/pixel. The chosen map is shown in figure 16. Selecting a relatively low resolution image for on-board processing facilitates lower storage capacity requirements and faster computation. The downside is that each pixel corresponds to 1.241 km on the lunar surface, and since the error in map-relative localization is based on error in pixel count, the surface relative localization error increases drastically with increase in the localization error pixel count.

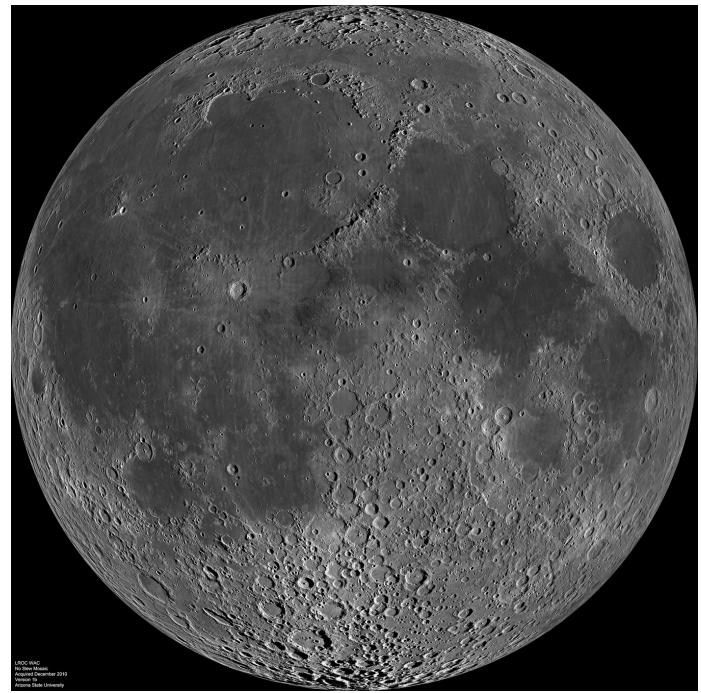


Fig. 16. Global map used in simulation

#### D. Robust Template Matching

We used a robust, scale and rotation invariant template matching algorithm to perform map relative localization. The template matching algorithm was implemented using the OpenCV *template matching* function. This function returns the correlation coefficient between by comparing each pixel in the captured image to the global map. To perform this

comparison, the descent image and the global map are preprocessed to convert both images from RGB to grayscale. The image and global map are also passed through the OpenCV *Canny* function to extract the edges from the images. Using the edges of the images increases the robustness of the template matching algorithm by reducing the number of false positive matches. The algorithm then sweeps the descent image template across the global map, one pixel at a time. The map is scaled through multiple levels and at each scale, the image is rotated in intervals of fifteen degrees (24 rotations per loop) to get the highest correlation between the global map and the descent image. The scaling loop increases the robustness of the algorithm to scale-mismatch between the descent image and the global map. The algorithm keeps track of the location with the highest correlation coefficient and provides this location as the predicted match [25]. Since the highest correlation scale and location are obtained, we assume that this algorithm would be efficient for both localization and altimetry measurements.

#### E. Crater Detection

A deep learning-based crater detection methodology was implemented to count the number of craters in the captured image. This was done using a TensorFlow-based pipeline called PyCDA, which utilizes a pre-trained Convolutional Neural Network for crater detection [26]. A sample crater detection output has been shown in figure 17.

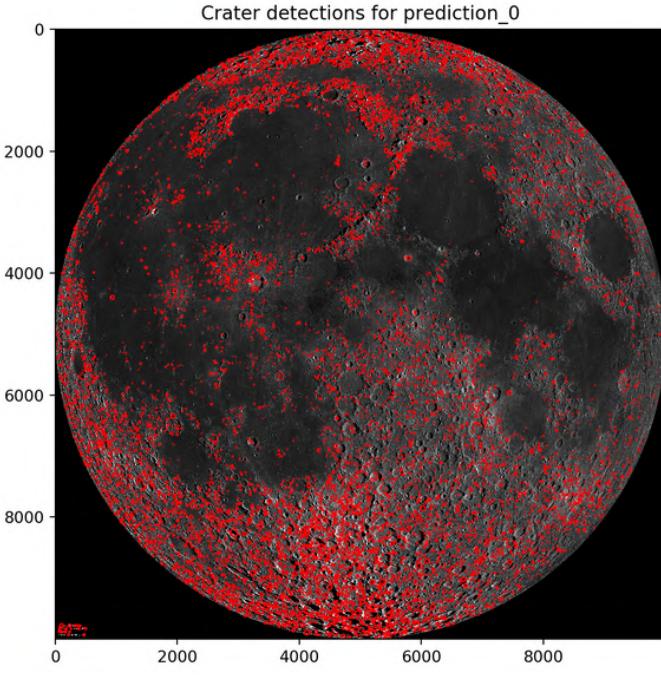


Fig. 17. A sample output to demonstrate the use of PyCDA for crater detection

## VI. GUIDANCE

### A. The Problem

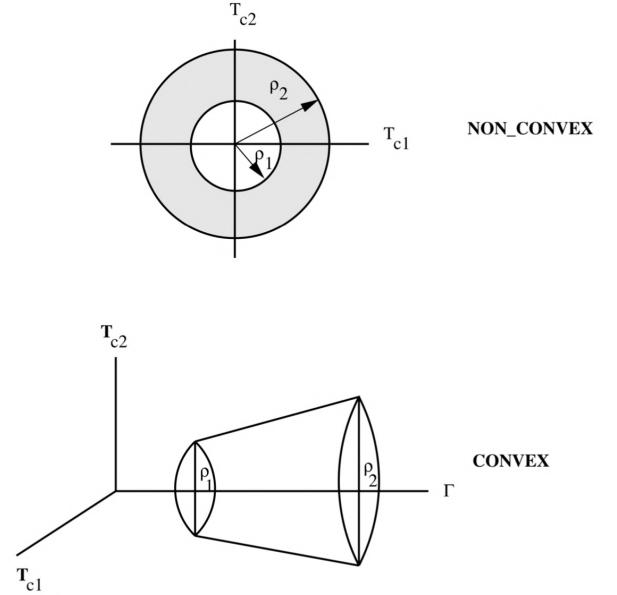
With the landing spacecraft's states having been estimated, a guidance algorithm is required to ensure that the control

system of the lander takes in a series of optimal thrust commands to accurately follow the desired position and velocity trajectories in order to land safely, having consumed the minimum possible amount of propellant.

This problem can be posed as an optimization problem subject to various constraints involving the dynamics of the landing spacecraft, thruster limitations, and physical constraints to ensure safe landing, with the amount of propellant consumed regarded as the 'cost' [27]. Moreover, this constrained optimization problem can be structured as one of convex optimization, and by obeying a set of well-defined rules to ensure problem tractability, a globally optimal solution can be guaranteed [16].

After powered-descent has been initiated, the thrusters can be shut down only at touchdown. The entirety of the powered-descent phase involves one continuous, throttled burn by the main engine. However, the lower-bound on thrust adds a nonconvex constraint, and problem reformulation is required to make the problem tractable.

Convexification of this constraint and reformulation of the problem has been proven to generate a global optimal solution, even with the introduction of a slack variable to lift the nonconvex lower thrust bound into a convex set of feasible controls (a convex cone), which essentially 'convexifies' the function. Since a global optimum for the relaxed problem also guarantees a global optimum for the original problem, based on mathematical proofs involving Hamiltonian analysis, the Transversality Condition and Pontryagin's Maximum Principle, this procedure is often referred to as "lossless" convexification [23].



Post lossless convexification, it is beneficial to reformulate the cost-function. Defining the new cost-function as the mass of the lander at touchdown as opposed to the total amount of propellant consumed, facilitates inclusion of the reformulated constraints [28].

### B. The Lander

The chosen vehicle, the Altair Lander, has a wet-mass of  $37045\text{kg}$ . Its dry-mass is  $10078\text{kg}$ , leaving it with  $26967\text{kg}$  of propellant. This is a hard constraint; the lander cannot physically consume more propellant than it is carrying on-board. The specific-impulse ( $I_{sp}$ ) of the vehicle is  $450.1\text{s}$ , and the main-engine of the vehicle is capable of generating a maximum of  $83000\text{N}$  of thrust.

### C. Constraints

Powered precision landing requires both the altitude and the velocity of the lander to reduce to zero exactly at touchdown. The engine is allowed to throttle between 30% and 80% the full-thrust value. The upper throttle limit is set to allow for a thrust-margin between the braking-burn set throttle and the maximum engine thrust [21].

Imposing vertical-only touchdown constraints for the final seconds of landing helps mitigate fuel-slosh effects and vehicle tip-over on touchdown.

A glideslope constraint ensures that the lander is at a safe distance from the surface at all points in the landing trajectory. It also enforces surface impact avoidance and does not allow the guidance algorithm to generate a trajectory that involves subsurface flight [23].

### D. The Algorithm

In order to solve the fuel-optimal precision landing problem, a Python-embedded modeling language for convex optimization problems called CVXPY has been used [29]. This modeling language was used in tandem with MOSEK, which is a proprietary large-scale optimization software [30]. Formulation of the guidance problem using CVXPY follows the basis of the G-FOLD algorithm developed at JPL [18], [31].

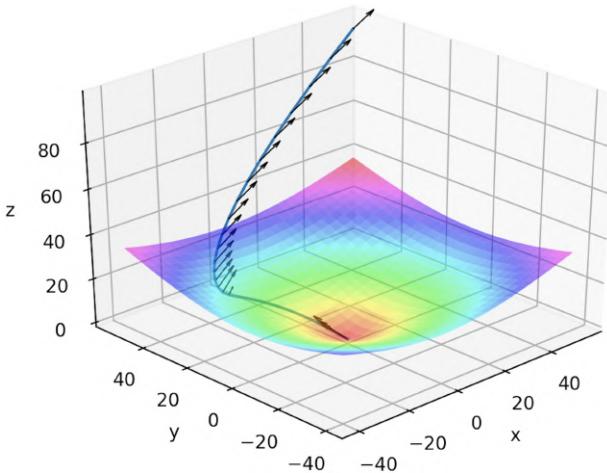


Fig. 19. A sample 3D fuel-optimal landing trajectory generated using CVXPY

## VII. SIMULATION

A flowchart describing the various steps in the simulation is shown in figure 20.

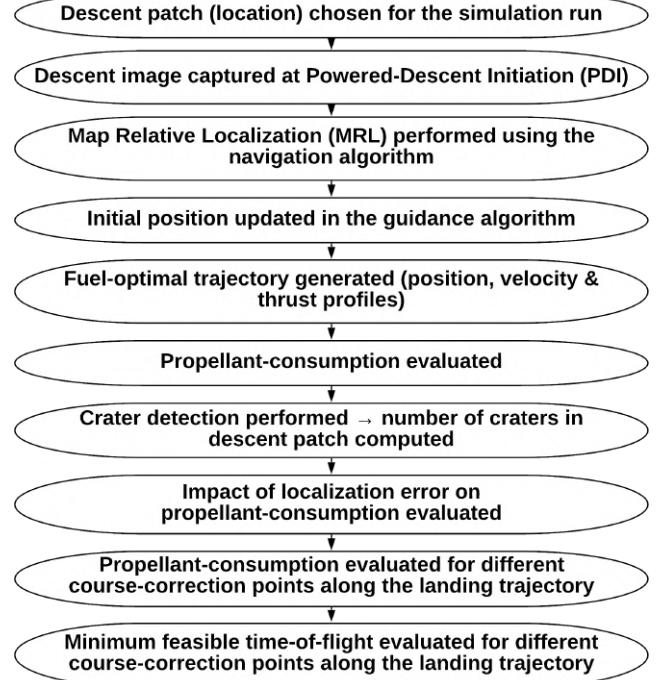


Fig. 20. Simulation Flow

### A. Terrain Relative Navigation

The first step in TRN is to select the decent patch used for localization. The area of the descent patch is calculated from the field of view (FoV) of the camera and the lander altitude at Powered-Descent Initiation (PDI). Map relative localization is performed using the on-board map. The MRL process has been depicted in figure 21. Once the algorithm is run, the newly estimated location is compared to the nominal location. The difference is converted into a positional estimate and the guidance algorithm is updated with a new initial position, accounting for the localization error.

Figure 21 shows an example of the template matching algorithm using a descent image pictured in the left corner to locate on the global map. This captured image includes Gaussian noise, motion-blur, and rotation and the navigation algorithm matches it with the global map with an error of one pixel. An error of one-pixel corresponds to 1.241 kilometers in positional error.

### B. Fuel-Optimal Guidance

The nominal initial position and velocity state conditions for the lander at PDI were chosen to closely match the values described in the Apollo 11 premission powered-descent event summary [32].

Note: The terms 'fuel' and 'propellant' have been used interchangeably in this paper.

It has been shown that the minimum feasible time-of-flight equals the fuel-optimal time-of-flight for 1 dimensional soft-landing problems [33]. For 2D powered-landing simulations, however, it has been observed that the

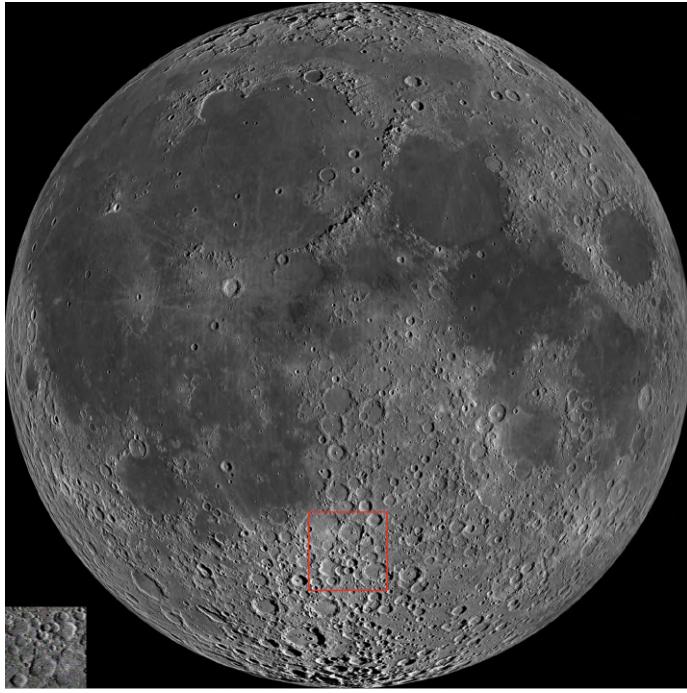


Fig. 21. Simulation of Map Relative Localization (MRL) using robust template matching

fuel-optimal time-of-flight is very close to the minimum feasible time-of-flight. It has also been observed that propellant consumption is a unimodal function (a function possessing a unique mode) of the time-of-flight [27]. Given this observation, a line search algorithm was implemented to evaluate this relationship in our simulation study.

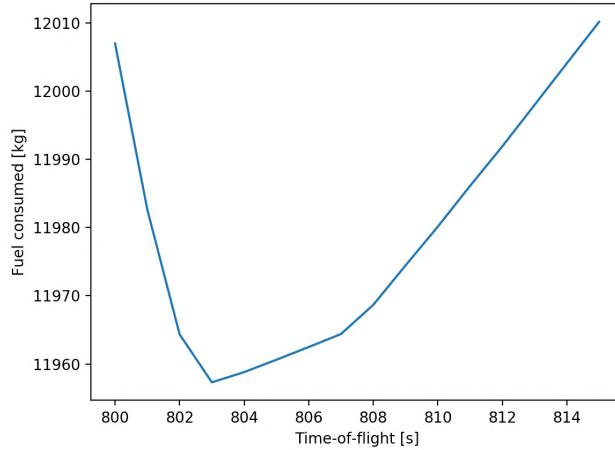


Fig. 22. Existence of a global-minimum for the propellant consumed vs. time-of-flight relationship (a convex function)

A line-search algorithm was implemented to compute the minimum feasible time-of-flight of the lander, given the initial states and subject to the constraints. For our simulation case, the minimum time-of-flight was found to be 799 seconds. Figure 22 shows the relationship between propellant

consumption and time-of-flight for our simulation. It is a unimodal function as expected, with a global minimum at 803 seconds, which is close to the minimum feasible time-of-flight of 799 seconds, also as expected.

The nominal (zero localization error) trajectory with initial position (downrange, altitude) at (-500, 15) [km] and initial velocity (horizontal, vertical) of (1100, -1) [m/s] has been shown in figure 23. The nominal landing trajectory was generated for the fuel-optimal time-of-flight case (803 seconds), and the propellant-consumption for the given initial conditions, lander model parameters and constraints is optimal. A total of 11957.309 [kg] was the projected fuel-consumption value for the nominal case.

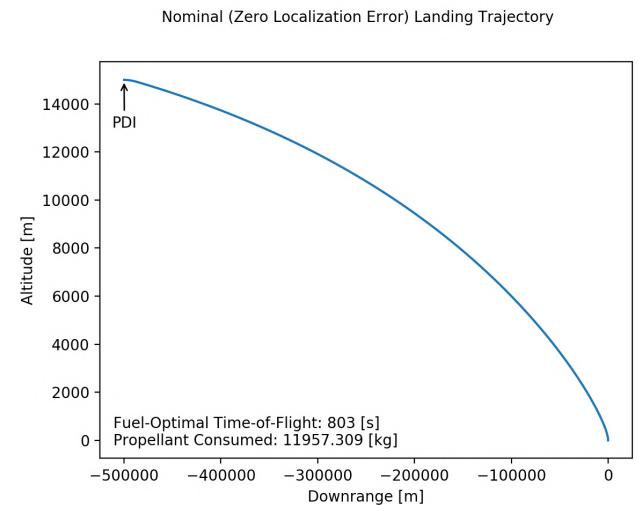


Fig. 23. Nominal Landing Trajectory

The new initial conditions obtained from the Map Relative Localization algorithm were then used to seed the guidance algorithm. The "actual" landing trajectory with the time-of-flight fixed at 803 seconds, subject to a map-relative localization error of 1 [px], has been shown in figure 24.

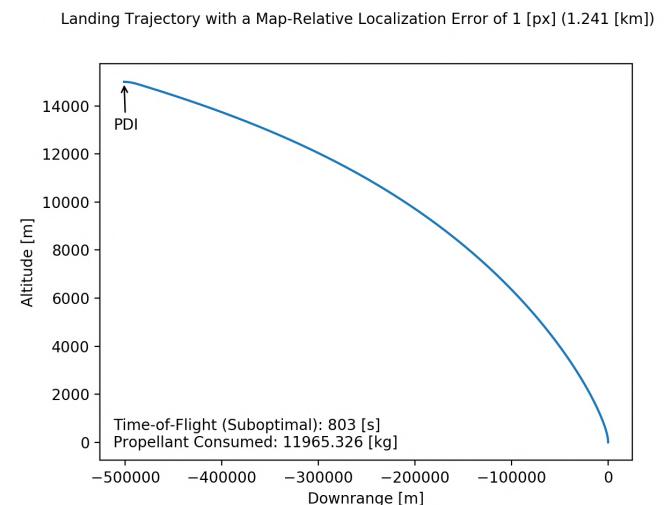


Fig. 24. Landing Trajectory with Localization Error

This evaluation depicts an additional propellant consumption projection of 8.017 [kg] on top of the nominal propellant consumption projection.

This localization error of 1 [px], however, as updated in the guidance algorithm, manifests in the form of an error of 1.241 [km] in position-estimation. The updated initial conditions are still map-relative, and the landing trajectory, if not corrected before touchdown, would result in a landing error of 1.241 [km] from the desired target site.

As a result, a course-correction maneuver is required to achieve precision landing.

### C. Precision Landing Guidance

In order to evaluate the best course-correction initiation point candidate along the nominal landing trajectory, five different points, equally spaced relative to the flight-time, were chosen. The course-correction maneuver would involve another sequence of Map Relative Localization (MRL) to reduce the localization error (to zero, for the purpose of our simulation). The different chosen points were spaced in time as opposed to altitude or downrange, in order to account for processing time that would be required to perform MRL.

The five different chosen course-correction initiation points along the nominal landing trajectory have been depicted in figure 25.

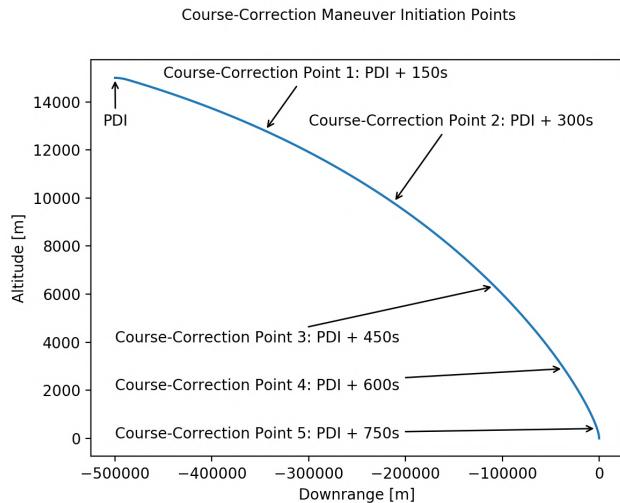


Fig. 25. Course-correction points (cases) chosen along the nominal landing trajectory

### D. The Course-Correction Maneuver

The individual course-correction cases seeded by the second and final MRL update, with the updated trajectories that include the required divert maneuvers, have been shown in figures 26, 27, 28, 29 and 30 respectively. After the points of course-correction initiation for the chosen cases, the updated landing trajectories were designed to be time-optimal in our simulation study. The resulting minimum feasible time-of-flight values and corresponding propellant consumption values have been depicted in the figures.

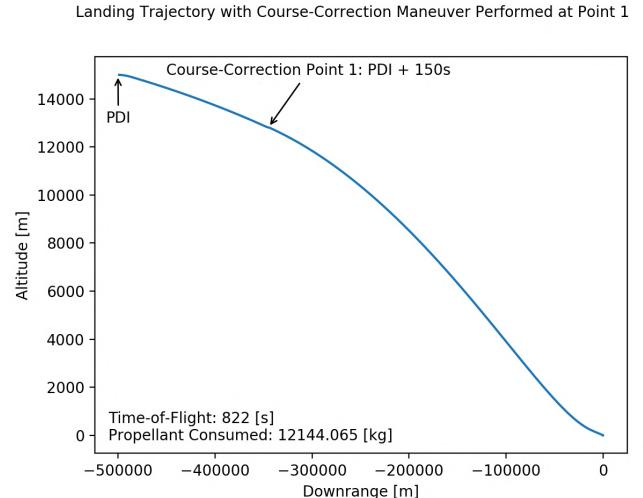


Fig. 26. Course-Correction Initiation Point 1

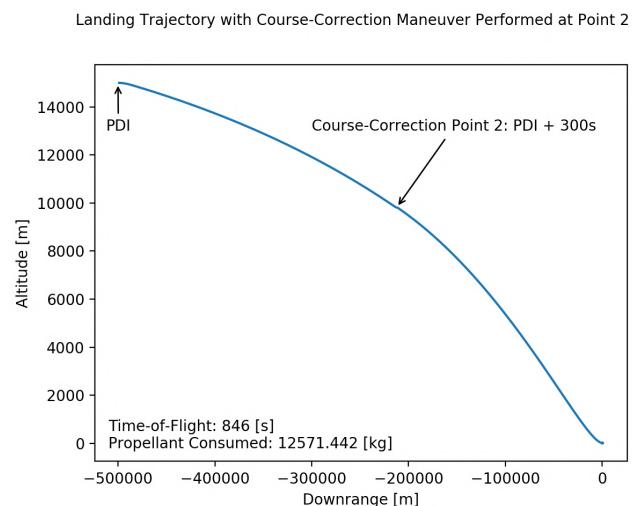


Fig. 27. Course-Correction Initiation Point 2

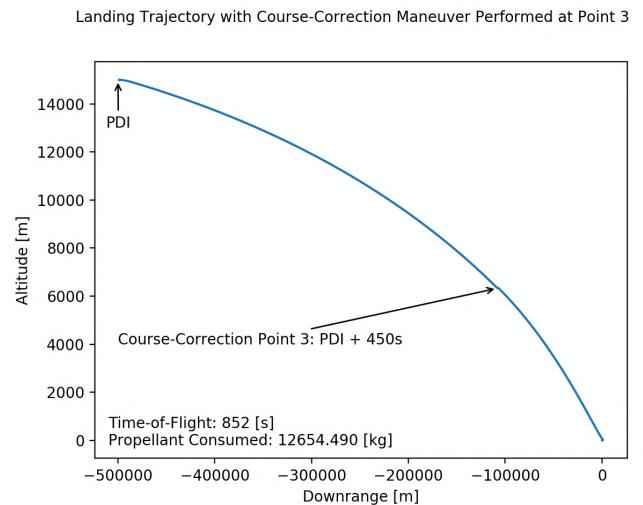


Fig. 28. Course-Correction Initiation Point 3

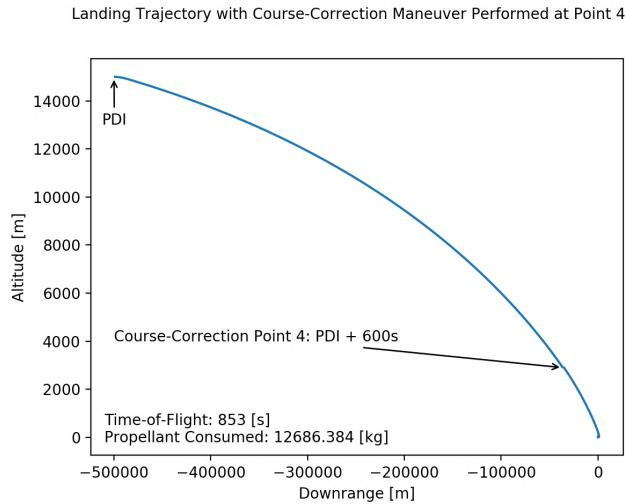


Fig. 29. Course-Correction Initiation Point 4

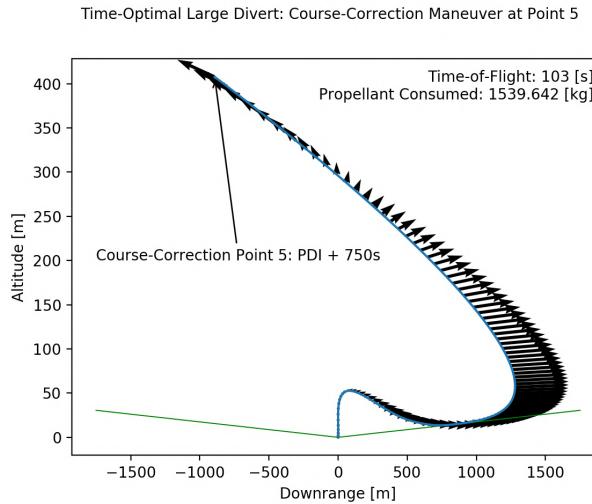


Fig. 31. Time-Optimal Large Divert Maneuver Performed at Course-Correction Initiation Point 5

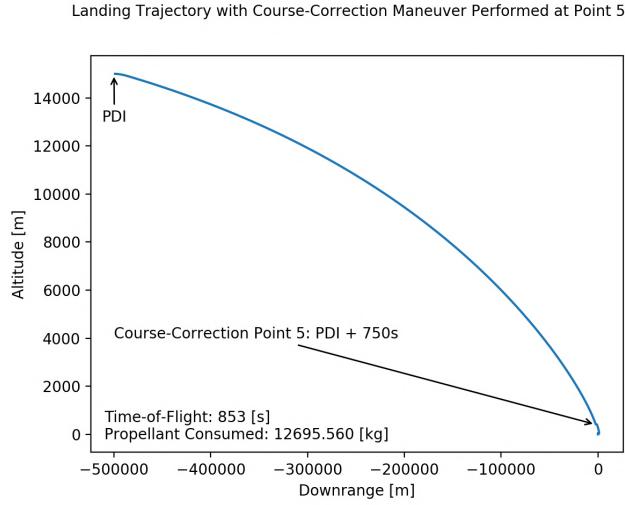


Fig. 30. Course-Correction Initiation Point 5

The trajectory overshoot and course-correction to achieve precision landing for the final case were particularly interesting. A "zoomed-in" version of the large course-correction divert maneuver has been shown in figure 31.

Figure 31 describes a time-optimal large course-correction divert maneuver, with the optimal thrust-command profile rendered along the landing trajectory. The chosen glideslope at a 1 degree inclination angle that describes the enforced constraint to avoid subsurface flight, and "vertical-only" terminal descent, are represented in the figure as well. It can be observed from the landing trajectory and the corresponding thrust-command profile that such a maneuver would not be well-suited for crewed lunar landing missions.

## VIII. RESULTS AND DISCUSSION

### A. Simulation Output

A screenshot of the output screen after running the simulation is shown in figure 32. It is observed that there is a monotonic decrease in the wet-mass estimates with increasing course-correction initiation points (in time).

```

Wet mass at correction point 1: 34914.595468356114 kg
Propellant consumed at correction point 1: 2130.404531643908 kg
Initial conditions at correction point 1: position, velocity
[-345127.31668656 12799.94957184]
[967.7163352 -17.76153298]

Wet mass at correction point 2: 32658.28643080762 kg
Propellant consumed at correction point 2: 4386.713569192401 kg
Initial conditions at correction point 2: position, velocity
[-212460.3977577 9800.3366802]
[795.88842796 -21.97925771]

Wet mass at correction point 3: 30401.904243792083 kg
Propellant consumed at correction point 3: 6643.095756207938 kg
Initial conditions at correction point 3: position, velocity
[-107925.92645045 6326.11552129]
[592.71617699 -23.77378757]

Wet mass at correction point 4: 28145.498270589553 kg
Propellant consumed at correction point 4: 8899.50172941047 kg
Initial conditions at correction point 4: position, velocity
[-36214.35519724 2898.71862314]
[358.12210351 -20.94233618]

Wet mass at correction point 5: 25889.082041647805 kg
Propellant consumed at correction point 5: 11155.917958352216 kg
Initial conditions at correction point 5: position, velocity
[-2133.69172449 407.78011924]
[90.59795127 -10.80132465]

```

Fig. 32. A screenshot of the simulation algorithm output, with the computed pre-correction conditions at the course-correction initiation points

### B. Crater Count Estimation

The results that we obtained for crater detection and count estimation proved to be inaccurate, as the images that we used in our simulation (173.74 meters/pixel) were of much lower resolution than the images originally used to train the PyCDA neural network (the algorithm has been tested on an image with a scale of 12.5 meters/pixel with fairly accurate results [26]). This resulted in a large number of false detections by the algorithm. An example case has been shown in figure 33.

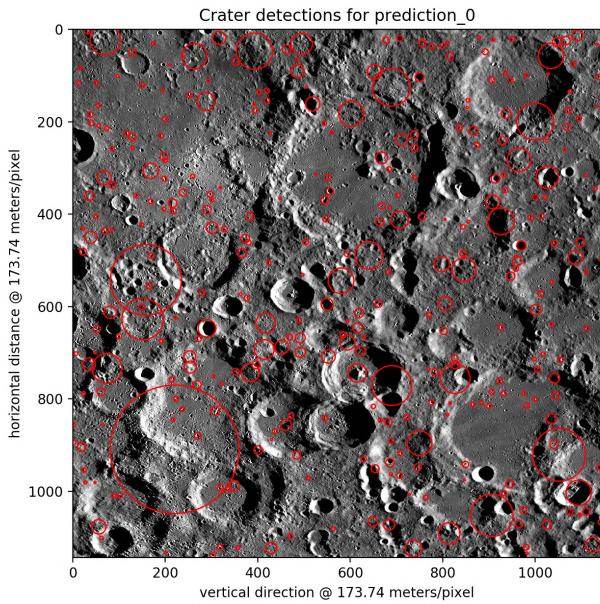


Fig. 33. Crater detection using PyCDA on the descent-patch: 328 craters were detected

### C. Performance Metric Evaluations

The simulation study demonstrates terrain relative navigation with a 1 pixel map-relative localization error. Figure 34 shows the trend in propellant consumption with increasing localization error in pixels, for a fixed time-of-flight of 803 seconds. The convex optimization problem becomes infeasible when the map-relative localization error is greater than 5 pixels.

Similarly, the trend in propellant consumption evaluated as a function of the localization error in terms of the position of the lander relative to the lunar surface is shown in figure 35.

Varying the course-correction initiation points along the nominal landing trajectory and evaluating the corresponding propellant consumption and minimum feasible time-of-flight values gave rise to interesting results.

Propellant consumption evaluated as a function of course-correction initiation point variation, as shown in figure 36, outlines a saturating trend, as the course-correction point is pushed closer to touchdown.

Time-of-flight evaluated as a function of course-correction initiation point variation, as shown in figure 37, however,

Propellant Consumption vs. Localization Error in Pixels

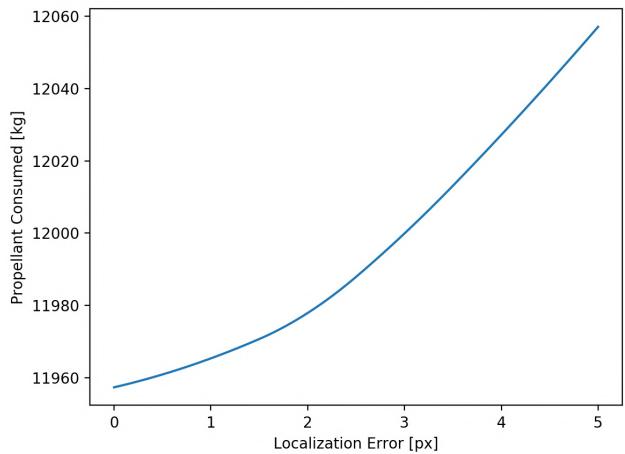


Fig. 34. Propellant Consumption vs. Localization Error [px]

Propellant Consumption vs. Localization Error in Meters

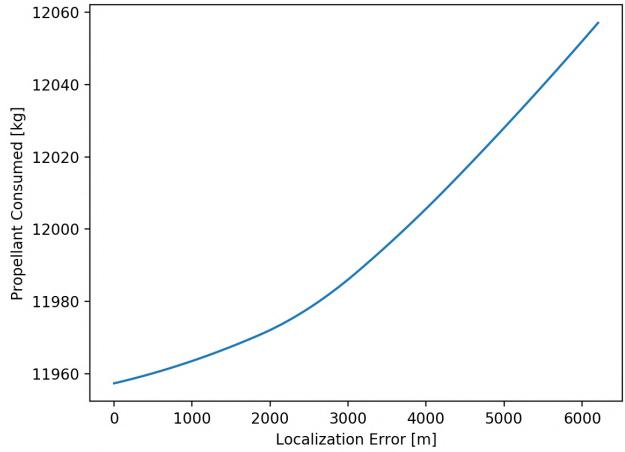


Fig. 35. Propellant Consumption vs. Localization Error [m]

Propellant Consumption vs. Course-Correction Point Along Nominal Trajectory

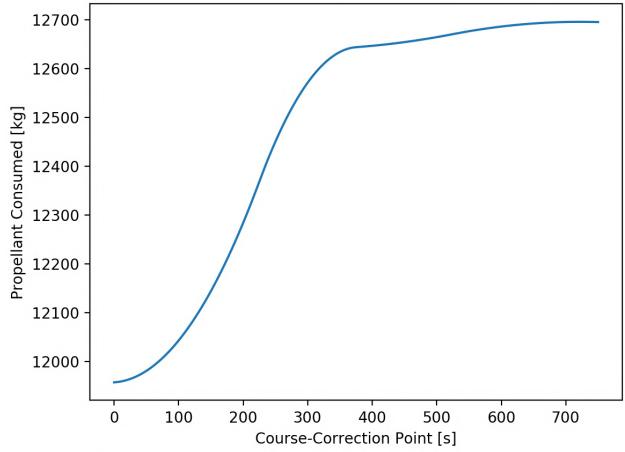


Fig. 36. Propellant Consumption vs. Course-Correction Initiation Point Along the Nominal Landing Trajectory

actually saturated at 853 seconds for course-correction points chosen at 600 seconds and 750 seconds along the nominal landing trajectory.

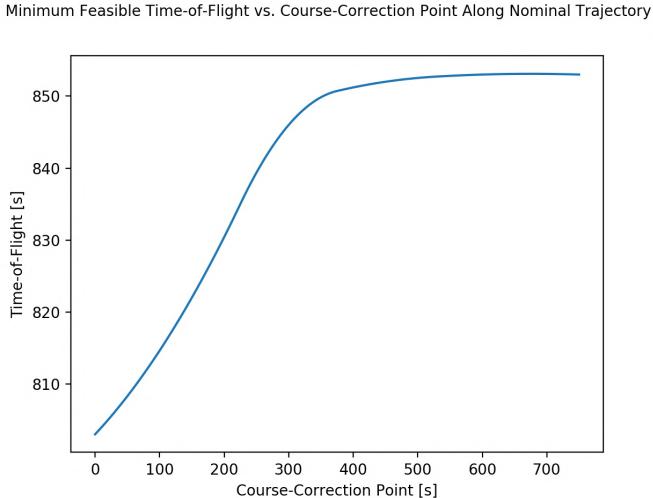


Fig. 37. Time-of-Flight vs. Course-Correction Initiation Point Along the Nominal Landing Trajectory

In both cases, however, it is evident that course-correction maneuvers performed as soon as possible in the descent path, result in the least propellant consumption values and minimum feasible time-of-flight values. These values are still greater than the values obtained for the nominal case.

## IX. FUTURE WORK

Having demonstrated static solutions for guidance and navigation, the next step would be to develop a high-fidelity simulation employing real-time guidance and navigation.

Such a simulation framework would also allow for the implementation of a deliberative and reactive hybrid control architecture by integrating hazard detection and avoidance (HDA) into the simulation environment. This can be executed using the same on-board monocular camera sensor, by implementing obstacle detection algorithms. The simulation can then be augmented to include the final phase of landing with the added obstacle avoidance capability of the lander.

An additional area of interest would be exploring more advanced localization algorithms. Specifically, robustness analysis and implementation of generalized-feature mapping algorithms would enable us to broaden the mission-scope.

## REFERENCES

- [1] A. Lee, T. Ely, R. Sostaric, A. Strahan, J. Riedel, M. Ingham, J. Wincentsen, and S. Sarani, "Preliminary design of the guidance, navigation, and control system of the altair lunar lander," in *AIAA Guidance, Navigation, and Control Conference*, 2010, p. 7717.
- [2] M. I. B. Rousso and S. Peleg, "Recovery of ego-motion using image stabilization," 1993.
- [3] A. Johnson, S. Aaron, J. Chang, Y. Cheng, J. Montgomery, S. Mohan, S. Schroeder, B. Tweddle, N. Trawny, and J. Zheng, "The lander vision system for mars 2020 entry descent and landing," 2017.
- [4] C. Restrepo, J. M. Carson III, and M. M. Munk, "Nasa splice project: Development and testing of precision landing gn&c technologies," 2018.
- [5] R. Vondrak, J. Keller, G. Chin, and J. Garvin, "Lunar reconnaissance orbiter (lro): Observations for lunar exploration and science," *Space science reviews*, vol. 150, no. 1-4, pp. 7–22, 2010.
- [6] A. E. Johnson and J. F. Montgomery, "Overview of terrain relative navigation approaches for precise lunar landing," in *2008 IEEE Aerospace Conference*. IEEE, 2008, pp. 1–10.
- [7] V. S. Bilodeau, S. Clerc, R. Drai, and J. de Lafontaine, "Optical navigation system for pin-point lunar landing," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 10535–10542, 2014.
- [8] Y. Cheng, A. E. Johnson, L. H. Matthies, and C. F. Olson, "Optical landmark detection for spacecraft navigation," 2003.
- [9] S. Clerc, M. Spigai, and V. Simard-Bilodeau, "A crater detection and identification algorithm for autonomous lunar landing," *IFAC Proceedings Volumes*, vol. 43, no. 16, pp. 527–532, 2010.
- [10] A. Ansar, "small body gn&c research report: Feature recognition algorithms," *Small Body Guidance Navigation and Control FY 2004 RTD Annual Report (Internal Document)*, pp. 151–171, 2004.
- [11] R. Gaskell, "Automated landmark identification for spacecraft navigation," 2001.
- [12] J. R. Carr and J. S. Sobek, "Digital scene matching area correlator (dsmac)," in *Image Processing For Missile Guidance*, vol. 238. International Society for Optics and Photonics, 1980, pp. 36–41.
- [13] A. Johnson, A. Ansar, L. Matthies, N. Trawny, A. Mourikis, and S. Roumeliotis, "A general approach to terrain relative navigation for planetary landing," in *AIAA Infotech@ Aerospace 2007 Conference and Exhibit*, 2007, p. 2854.
- [14] A. E. Bryson, *Applied optimal control: optimization, estimation and control*. Routledge, 2018.
- [15] A. Lee, "Fuel-efficient descent and landing guidance logic for a safe lunar touchdown," in *AIAA Guidance, Navigation, and Control Conference*, 2011, p. 6499.
- [16] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [17] L. Blackmore, "Autonomous precision landing of space rockets," in *Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2016 Symposium*. National Academies Press, 2017.
- [18] B. Acikmese, J. Casoliva, J. Carson, and L. Blackmore, "G-fold: A real-time implementable fuel optimal large divert guidance algorithm for planetary pinpoint landing," in *Concepts and Approaches for Mars Exploration*, vol. 1679, 2012.
- [19] M. Szmułk, C. A. Pascucci, D. Dueri, and B. Açıkmese, "Convexification and real-time on-board optimization for agile quad-rotor maneuvering and obstacle avoidance," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4862–4868.
- [20] T. A. Ely, M. Heyne, and J. E. Riedel, "Altair navigation during translunar cruise, lunar orbit, descent, and landing," *Journal of Spacecraft and Rockets*, vol. 49, no. 2, pp. 295–317, 2012.
- [21] L. Kos, T. Polsgrove, R. Sostaric, E. Braden, J. Sullivan, and T. Le, "Altair descent and ascent reference trajectory design and initial dispersion analyses," in *AIAA Guidance, Navigation, and Control Conference*, 2010, p. 7720.
- [22] J. Honeycutt, C. Cianciola, and L. Wooten, "Nasa's space launch system: Artemis 1 vehicle nears completion," 2019.
- [23] J. M. Carson, B. Açıkmese, and L. Blackmore, "Lossless convexification of powered-descent guidance with non-convex thrust bound and pointing constraints," in *Proceedings of the 2011 American Control Conference*. IEEE, 2011, pp. 2651–2656.
- [24] "European seventh framework programme fp 7-218086-collaborative project deliverable 2 . 7 proposed algorithms for mission planning for groups of uavs," 2010.
- [25] A. Rosebrock, "Multi-scale template matching using python and opencv," Oct 2018. [Online]. Available: <https://www.pyimagesearch.com/2015/01/26/multi-scale-template-matching-using-python-opencv/>
- [26] M. Klear, "Pycka documentation: Simple crater detection." [Online]. Available: <https://pycka.readthedocs.io/en/latest/index.html>
- [27] B. Acikmese and S. R. Ploen, "Convex programming approach to powered descent guidance for mars landing," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 5, pp. 1353–1366, 2007.
- [28] B. Açıkmese and L. Blackmore, "Lossless convexification of a class of optimal control problems with non-convex control constraints," *Automatica*, vol. 47, no. 2, pp. 341–347, 2011.
- [29] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.
- [30] M. ApS, *MOSEK Optimizer API for Python 9.0.105*, 2019. [Online]. Available: <https://docs.mosek.com/9.0/pythonapi/index.html>

- [31] N. S. Bhasin, W. Whittaker, and C. Atkeson, "Fuel-optimal spacecraft guidance for landing in planetary pits," Ph.D. dissertation, Carnegie Mellon University Pittsburgh, 2016.
- [32] F. V. Bennett, "Apollo experience report: mission planning for lunar module descent and ascent," 1972.
- [33] J. Meditch, "On the problem of optimal thrust programming for a lunar soft landing," *IEEE Transactions on Automatic Control*, vol. 9, no. 4, pp. 477–484, 1964.