Development of an Achievability Propellant Limit Algorithm for a Piloted, Lunar Lander

Carlos Pinedo,[1] Edward Zuzula,[2], Elliott Davis, [3] Jordan Dixon, [4] Torin K. Clark[5]
University of Colorado, Boulder, CO, 80309, USA

While all six Apollo lunar landings were successful, the multiple landing site redesignations and landing near hazards suggests crewed planetary landings can be challenging and demand high pilot workload. To assist with landing site selection, the concept of providing "achievability limit" information (i.e. where on the planetary surface is achievable with the energy remaining) to the pilot has been proposed. However, an approach to accurately estimate the achievability limit for a complex three-dimensional planetary landing task has not yet been developed. We developed an algorithm to estimate the achievability limit, generalizable to future planetary landings. The algorithm combines three components: 1) vehicle guidance laws, 2) vehicle and environmental dynamics, and 3) a simplified "crossover" pilot model. The algorithm performs multiple closed-loop numerical simulations to predict the propellant remaining to reach various landing points to identify the zero propellant remaining points that define the achievability limit area. Here we describe the algorithm and present a sensitivity analysis of input parameters to demonstrate its functionality for a lunar landing scenario. We envision the algorithm could run in real-time during a landing to provide achievability information for the crew, adding in safe landing point selection.

Nomenclature

$a_{x,y}$ = vehicle's translational acceleration in the x and y directions, m/s^2

ALD = Achievable Limit Display

---

[1] Graduate Student, Ann and H.J. Smead Aerospace Engineering Sciences, and AIAA Student Member.
[2] Undergraduate Student, Ann and H.J. Smead Aerospace Engineering Sciences.
[3] Undergraduate Student, Ann and H.J. Smead Aerospace Engineering Sciences.
[4] Graduate Student, Ann and H.J. Smead Aerospace Engineering Sciences.
[5] Assistant Professor, Ann and H.J. Smead Aerospace Engineering Sciences, and AIAA Member

$\delta_{r,p}$ = joystick deflection controlling either roll (r) or pitch (p), degrees

fps = feet per second

h = height above lunar equivalent ‘mean sea level’, m

Isp = specific impulse

K = feedback gain

LM = Lunar Module (Apollo)

LG = Low Gate

m = mass, kg

MPP = Maximum Propellant Point

MP = Mid-Point

θ = vehicle pitch, degrees

ϕ = vehicle roll, degrees

R = Range, m

TD = Terminal Descent

T = Thrust, N

V = Vehicle velocity over lunar surface, m/s

x = position in the x direction, Lunar-fixed (‘East/West’), m

y = position in the y direction, Lunar-fixed (‘North/South’), m

ZPP = Zero Propellant Point

Subscripts

c = control element

cmd = commanded variable

LP = Landing Point

p = pilot element

Superscripts

G = Guidance variable

* = reference trajectory variable

I.Introduction

The landing phase is one of the most challenging and mission critical phases for human planetary surface exploration missions (e.g. lunar landings as demonstrated by Apollo space program [1]). The Apollo lunar landing process began with the descent orbit insertion to efficiently reduce the orbit altitude from 111,120 km to 15 km for powered descent initiation [2]. The powered descent included three operational phases 1) braking phase: reduction of orbital velocity and guidance to 'high gate'; 2) approach phase: initial pilot visual (out the window) monitoring of approach to surface and guidance to 'low gate'; 3) landing phase: continuous pilot visual monitoring of landing site and when the pilot may take over from automatic control manually commanding the vehicle flight path and attitude [2]. The resultant trajectory facilitated vertical descent initiation from 45.72 m (150 feet) at 0.9 m/s (3 fps) and 'low gate' conditions of 152.4 m (500 ft.) altitude with 18.29 m/s (60 fps) forward velocity, 4.88 m/s (16 fps) vertical descent rate, and a 16º off vertical vehicle orientation [2]. This research focuses on the landing phase when pilot manual control is available and final landing point selection decisions are being made.

Crewed planetary landings require the operator(s) to determine a suitable landing point (defined below) and maintain control (supervisory or manual) of the vehicle throughout the entire landing process. The suitable landing point selection criteria includes points of scientific interest, vehicle performance, and hazard avoidance. The operator(s) will require detailed information on the presence of craters, boulders, terrain slope and surface roughness for hazard avoidance [3] and landing point redesignation [4]. Although some of this information is expected to be available pre-flight, either from previous un-crewed missions or orbital imaging, final landing zone assessment will ultimately be completed in real-time by the operators onboard the vehicle. The landing point selection is further constrained by time; 'low gate' to touchdown was typically around 2 minutes for Apollo [2], and the earlier a landing point redesignation occurs the lower the energy cost (ΔV). In this time constrained environment determining landing point achievability (ability to land given current vehicle energy state) is critical for safe landing point selection.

The six Apollo lunar landings all encountered potentially mission ending hazards (both recognized and unrecognized by the pilots in real-time), while managing diminishing available propellant in a unique landing environment [1]. The presence of these hazards at the original landing point required the operators to redesignate the landing point at least once during all missions and as many as 18 times during Apollo 15 [5]. In Apollo, the approach

and landing phases lasted approximately 185 seconds and required the crew members to monitor the automated systems, input commands to the Apollo guidance computer, identify navigational landmarks, select a landing point, and provide flight path and attitude control inputs in the final phase [5]. Cummings et al. [6] identified a series of limitations of the Apollo Lunar Module (LM) through interviews with Apollo astronauts and Apollo mission assessments which included: astronauts relied on significant mental calculation and rough estimation to identify possible landing areas, the geometric constraints of the LM windows limited visual landing point redesignation options, and a lack of lunar surface terrain information impacted landing accuracy. Some of the crewmembers encountered geographic disorientation during the pitch-over maneuver, which tilted the vehicle towards an upright position enabling the crewmembers to visually acquire the landing point. Pete Conrad, Apollo 12 mission commander, said "When the LM pitched up at 7,500 [ft.], I didn't have the foggiest idea where I was. There was 10,000 craters out there" [6]. Interviews with former Apollo astronauts highlight some of the challenges and the high workload encountered during the lunar landings. For many of the astronauts, propellant was the driving factor in determining the landing.

For future planetary landings higher quality imaging of the landing point is not guaranteed and human pilot capability and the physics of the landing task remain unchanged. The landing task will require crewmember collaboration with automated systems to identify the final landing point with minimal propellant available and time. Specifically, each second of flight requires propellant consumption and propellant allocated towards landing must be launched as "payload" from Earth. While automation can aid information synthesis and vehicle control during these complex flight operations, in accordance with NASA Procedural Requirement 3.3.2[1], human-rated spacecraft shall provide the crewmembers with the ability to override higher level automation and transition to manual control. This mode transition from supervisory control to manual control during lunar landings has been shown to increase subjective and objective workload and decrease performance [7]. Thus, success of crewed planetary landings will hinge on the precise coordination of human pilots with complex automated systems that facilitate automation mode transitions.

In order to improve landing point selection during manual control phases of planetary landings this paper introduces a novel algorithm to determine the achievability limit, providing the pilot with more detailed information

[1] NASA Procedural Requirement 3.3.2: "The crewed space system shall provide the capability for the crew to manually override higher level software control/automation (such as automated abort initiation, configuration change, and mode change) when the transition to manual control of the system will not cause a catastrophic event."

about where on the planetary surface they can reach with the remaining propellant/energy. The algorithm combines a model of vehicle guidance parameters (descent rate [$\dot{h}^G$], vehicle pitch [θ^G], and roll angle [ϕ^G]), the current vehicle state (position in three-dimensions, vehicle velocity [V_X, V_Y, and $\dot{h}$], vehicle pitch [θ] and roll [φ] angles, and vehicle propellant mass [m_{prop}]), and the environment, with a model of pilot-vehicle control behavior to provide an estimate on the achievable landing area during propellant-constrained planetary landings. Providing achievability limit information to the astronaut crewmembers (i.e., an Achievability Limit Display (ALD)) has been previously proposed as a means to reduce pilot workload, improve situation awareness, and increase safety [4, 5].

Since Apollo, various display designs have been proposed to explicitly provide processed information on the achievable landing area. The cockpit avionics upgrade (CAU) proposed for the Space Shuttle was an example of a modern energy management display that provided 'achievability' information to the pilots [8]. The CAU program horizontal situation entry display included an enclosed contour of the nominal landing footprint for an unpowered glide [8]. Another conceptional achievability limit design included an array of displays including a Landing Zone display with primary flight information, hazard information, and a redesignation mode [6]. However, the user had to sequence through various landing points comparing the propellant remaining gauge to assess achievability [6]. Additional landing point designation displays depicting propellant remaining contours and landing hazards have been designed [9]. One of these additional systems computed ratings for pre-selected landing points based on a weighted combination of surface slope, terrain roughness, ΔV, distance to hazards, and distance from points of interest [10]. A more recent study performed an experimental evaluation of an achievability contour display during simulated piloted lunar landings [5]. Potential landing point locations, a digital elevation map, and hazardous regions were depicted in a top-down egocentric moving map. As it was an initial assessment of the use of achievability contours the study did not include an algorithm that updated the achievability limit based on calculations, but rather utilized a lookup table and interpolation with a few characteristic parameters (e.g., vehicle position, velocity, and altitude).

To date the approach to predicting the achievability limit during lunar landing scenarios has relied on pre-determined landing ellipse information or a table lookup. The Space Shuttle CAU horizontal situation entry display was the only approach described above that incorporated real-time dynamically updated information on landing achievability. This information was critical to the orbiter because it behaved like a glider during the entry phase of flight with limited maneuverability and landing options [8]. The approaches for lunar landings described above did not dynamically incorporate the vehicle state or the planetary terrain in real time to provide timely information on the

achievability landing limit. Catastrophic failure (loss of crew or vehicle) during crewed planetary landings could occur if the display informs the operator that a landing point is achievable when it is not. To reduce this type of risk many of the approaches above were conservative in their estimate of the achievability limit. This conservative approach during a time and propellant constrained landing task may reduce the crews' landing point selection options. This may lead to suboptimal landing point selection or a decision to abort back to orbit when a suitable landing point was achievable but unrecognized. Providing accurate up-to-date information on the achievability limit is needed for safe and effective landing point selection for crewed planetary landing missions.

This paper introduces a novel achievability limit algorithm that incorporates the many factors that determine the achievable landing area. By incorporating a physical model of the vehicle and environment with a behavioral model of the pilot, a more accurate dynamically updated achievability landing limit can be determined. The goal of the algorithm is to provide the pilot in real-time achievability information, facilitating landing point redesignations with maximum achievable reach while ensuring safety. A secondary benefit is that by helping the pilot maximize achievable reach with the given propellant, less landing propellant mass may be required, saving on initial launch mass from Earth. In the following sections we present a detailed explanation of the algorithm, example predictions depicting the outcome of the algorithm, and a parameter sensitivity analysis to demonstrate the behavior of the achievability limit. We conclude with the next steps for future development and limitations of the algorithm.

II. Methods

Our achievability limit algorithm conducts numerical simulations from the current vehicle state to any designated landing point (i.e. location on the planetary surface) and provides the propellant required to land at that point. The set of landing points on the planetary surface which yield zero propellant remaining are identified as the "achievability limit." The zero propellant remaining landing points are combined to form a closed contour that defines the achievability landing area. This closed contour can be displayed on a moving map to provide the user with detailed information on where on the planetary surface the user can land given the current vehicle energy state.

The general algorithm approach is detailed below and we include, where applicable, the assumptions and parameters chosen for a lunar landing scenario implementation. The algorithm has been implemented in MATLAB, leveraging MATLAB built-in functions as identified.

General Algorithm Approach

Three key elements are defined in order to run a numerical simulation and predict the propellant required: vehicle and environmental dynamics, vehicle guidance laws, and a model of pilot behavior.

Vehicle and Environment Dynamics

The vehicle dynamics are defined by the vehicle design (i.e. inert mass), descent engine thruster characteristics (i.e. specific impulse, minimum and maximum thrust capability, and thrusting dynamics), and reaction control system capabilities (i.e. dynamics of attitude control and peak rates). To demonstrate the algorithm's functionality, we implemented the vehicle and environmental dynamics for a lunar landing scenario with the vehicle characteristics as defined by Bilimoria [11]. The vehicle body axes uses a standard aircraft convention with the origin at the center of mass (x-axis out the nose of the vehicle, y-axis out the right, and z-axis out the bottom). The vehicle includes a descent engine thruster with a specific impulse of 311 s, maximum thrust of 44,482 N, and an initial inert mass (m_{lander}) of 7,195 kg (493 slugs) and propellant mass (m_{prop}) of 730 kg (50 slugs).

The environmental dynamics consist of planetary gravity, atmospherics (e.g. density for drag calculations), and terrain. The algorithm was applied to a lunar landing scenario with negligible atmospheric drag and magnitude of gravity of g = 1.622 m/s^2 (5.32 ft/s^2) For the purposes of algorithm development, an example terrain map was created (3x3 km in size with terrain elevation that varied ±100 m, see Fig. 5) and no actual lunar geodetic information was used.

Guidance Laws

We used previously derived guidance law equations for lunar landing [11], but then modified them for a generalized framework. The guidance laws created by Bilimoria defined a reference trajectory and provided guidance cues (pitch and roll) to that reference trajectory for the pilot to attempt to track during manual control landings. The Apollo LM did not have any active guidance cues displayed to the astronauts. Appendix A includes a complete description of the guidance equations, derivatives, and modifications.

The reference trajectory used to determine the pilot guidance cues consisted of three modes: approach to terminal descent, terminal descent to touchdown, and hover. The reference trajectory was designed to guide the pilot from low gate (152.4 m [500 ft.] altitude and 411.5 m [1350 ft.] ground range) to the terminal descent point 45.7 m (150 ft.) above the landing point. During the transition from low gate to terminal descent the vertical descent rate ($\dot{h}$) decreases linearly from 4.88 m/s (16 fps) to 0.9 m/s (3 fps). The reference trajectory also slowly decreases the horizontal

velocities (V_X, V_Y) to 0 m/s. Once in the terminal descent mode vertical velocity is held constant at 0.9 m/s (3 fps) to landing (Fig 1).

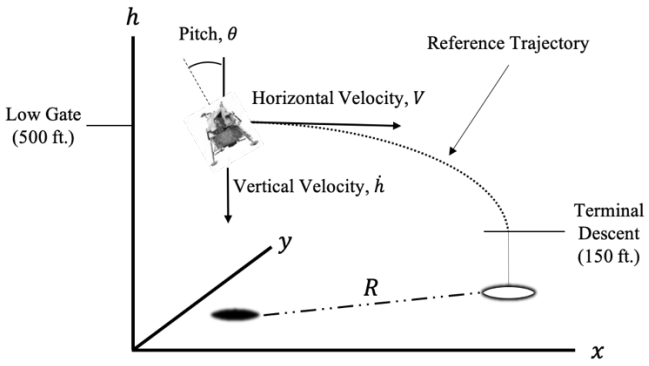

Fig. 1 Lunar landing flight profile and vehicle states

The reference trajectory in Bilimoria's work was pre-determined (based on Apollo) and had fixed parameters. Modifications were made to these equations to generalize this reference trajectory to account for any given starting position and changes to the landing point. In order to generalize, we made three main changes to the guidance and control law equations as implemented by Bilimoria. First, we added a hover "phase" (descent rate = 0 m/s) to the vertical reference trajectory, entered if the vehicle descended below 45.7 m (150 ft.) above the landing point elevation, but was farther than 15.2 m (50 ft.) horizontally from the landing point. Once the vehicle was within 15.2 m (50 ft.) horizontally from the landing point the hover would terminate and the terminal descent to touchdown phase would begin. Second, the reference trajectory could be recalculated mid-landing. The original equations provided guidance from the starting point to landing along the reference trajectory without changes to the landing point position and did not account for the pilot overshooting the landing point. The guidance equations were modified to address this scenario (see Appendix A). Third, the control law was modified to constrain the vehicle thrust profile between 4,448 N and 26,689 N (10-60% of the engine's maximum thrust). Each of the roll and pitch rates were limited to 5 degrees per second to replicate the expected thruster behavior of a reaction control system (RCS), which produce changes in vehicle attitude. Individual selection and firing of RCS jets was not modeled, instead the model uses aggregate forces and moments that would be generated by the combination of the RCS and descent engine thruster. In addition, we did not include vehicle yaw control (i.e., the vehicle remained facing in its initial orientation) since Bilimoria found that yaw inputs added little value while adding significant workload to the lunar landing task even for experienced astronaut pilots [11].

With these modifications implemented, the guidance equations provide the desired descent rate ($\dot{h}^G$), vehicle pitch (θ^G), and roll angles (ϕ^G) (see Appendix A). Pitch and roll angle control were modeled using pilot input via a rotation hand controller (i.e., joystick), while the descent rate was modeled as tasked to an automated control system. The guidance equations could be extended to include manual control of the descent rate, but currently only include the automated thrust control dynamics because manual throttle control was found to substantially increase an already high pilot workload task [11].

Pilot Model

A pilot model is required to estimate pilot reference trajectory tracking behavior from the current state to the landing point. For this we used a simplified version of the McRuer crossover model which describes pilot tracking behavior as a gain and time delay [12]. Although, actual pilot behavior is stochastic and varies during and across scenarios, the McRuer "crossover" model [12] has been effective at characterizing pilot tracking behavior. This crossover model Eq. (1) describes the open loop transfer function (Y_{OL}) as a product of the control element (Y_c) and pilot element (Y_p) and holds true for the outer loop when the control element transfer function includes the effects of all inner-loop closures.

$$Y_{OL}(s) = Y_p Y_c = \frac{\omega_c e^{-\tau s}}{s}; \quad near \omega_c \tag{1}$$

In Eq. (1), τ is the time delay associated with the pilot's perception, cognition, and motor responses (typically 0.2-0.3 seconds), s is the Laplace variable for time derivative, and ω_c is the crossover frequency. The above equation describes how operators tend to manipulate their control inputs such that $Y_p Y_c$ responds like a gain, a time delay, and an integrator in the region near the crossover frequency (ω_c).

The vehicle included a rate command attitude hold (RCAH) flight control system, in which the vehicle roll and pitch rate are proportional to pilot joystick deflections (δ), with attitude hold engaged when joystick deflections are zero. This control system with the crossover model above yields the following equations:

$$Y_c = \frac{\theta}{\delta} = \frac{K_c}{s} \quad ; \dot{\theta} = \delta K_c \tag{2}$$

And thus;

$$Y_p = \frac{\delta}{\theta^G - \theta} = K_p e^{-\tau s} \tag{3}$$

Where K_c is the proportionality constant for joystick deflection to commanded pitch/roll rate, and K_p is the proportionality constant corresponding to the pilot's control strategy. Rearranging the above equations in terms of control input (δ) and roll/pitch rate ($\dot{\theta}$) yields:

$$\delta = K_p e^{-\tau s}(\theta^G - \theta) \quad (4)$$

$$\dot{\theta} = K_c K_p e^{-\tau s}(\theta^G - \theta) \quad (5)$$

Preliminary analysis during algorithm development found that time delays between 0.0 to 1.0 seconds had limited impact (less than 1% change in propellant remaining) on the algorithm predictions and therefore, for simplicity, we removed it from the pilot model by setting $\tau = 0$ s (Eq. 6).

$$\dot{\theta} = K_c K_p (\theta^G - \theta) \quad (6)$$

During algorithm development, we also found the estimates of propellant required to be insensitive to gain (K_cK_p) values above 0.5 s – 1 s, and thus we set K_cK_p=0.5 s. Pitch and roll guidance cues are considered independent and cross-axis control interactions between pitch and roll guidance cues were not modeled (see Discussion).

The landing profile numerical estimation was conducted in MATLAB using the guidance laws, vehicle dynamics, and pilot model inputs described above. For the lunar scenario the landing point was defined in three dimensions (X_{LP}, Y_{LP}, and h_{LP}) where height was measured relative to a reference "mean sea-level" elevation. This accommodated changes in terrain elevation and resulted in h_{LP} having both positive and negative values.

Numerical Simulation

The numerical simulation begins with defining 1) initial vehicle state variables: vehicle position in three-dimensions (x, y, and h), vehicle velocity (V_X, V_Y, and $\dot{h}$), vehicle pitch (θ) and roll (φ) angles, and vehicle propellant mass (m_{prop}) and 2) a proposed landing point location (x_{LP}, y_{LP}, which define h_{LP}, the elevation of the planetary terrain at that location, relative to a "mean sea level") (Fig. 2A). The achievability limit algorithm uses the initial vehicle state and set of ordinary differential equations produced by the guidance laws, vehicle and environment dynamics, and pilot model to compute flight profiles. An ordinary differential equations solver (MATLAB function *ode45*) was used to compute a flight profile.

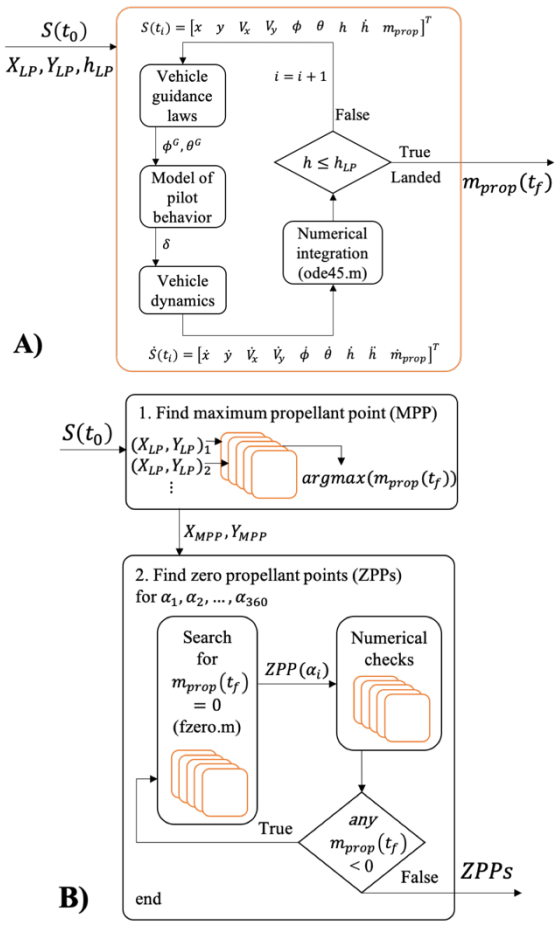

Fig. 2 Diagram of Achievability Limit Algorithm. Fig. 2a Numerical Simulation to estimate propellant at landing. Fig. 2b Achievability limit determination. When multiple numerical simulations (shown in orange) are performed, this is depicted by the stack of orange boxes.

The numerical simulation stops when the altitude of the lander reaches the elevation of the landing site ($h \leq h_{LP}$). The numerical simulation produces the eight vehicle state variables (x, y, h position, V_X, V_Y, $\dot{h}$, θ, and φ) as a function of time, and most critically, produces the vehicle propellant mass when reaching the landing point ($m_{prop}(t_f)$, Fig. 2A).

Generating the Achievability Limit

Multiple numerical simulations described above are performed from the initial vehicle state to various potential landing point locations in order to determine the points on the planetary surface which are reached with zero propellant remaining. In lieu of calculating the propellant required to land at every point on the planetary surface we developed a strategy to do this more efficiently (i.e., with a manageable number of numerical simulations). The algorithm first calculates the maximum propellant point (MPP) and then calculates zero propellant points (ZPPs) at a user defined angle range (0 – 360°) and increment. The angle used has its origin at the MPP with 0° directly east of the MPP and

positive angles in a counter-clockwise direction. While the achievability limit does not require determining the MPP, the portion of the algorithm that determines the ZPPs requires identifying a point on the planetary surface which was within the achievability limit area before beginning its estimation. If the achievability limit exists (i.e., there is some area on the planetary surface that can be achieved with the remaining propellant), the MPP is guaranteed to be within the achievability limit.

We assume the MPP is located along the direction of the vehicle's initial horizontal velocity vector. This assumption is violated when the vehicle has a large initial roll angle (see Discussion), but has the benefit of reducing a 2-D optimization problem to 1-D. To identify the point on the planetary surface along this direction which yields the maximum propellant remaining (MPP, Fig. 2B), we use the Nelder-Mead simplex algorithm (MATLAB's *fminsearch*) to find the maximum propellant remaining as a function of distance along this direction (Fig. 3). If the MPP has a positive propellant remaining value, there is an achievable area, and it is then used as the starting point for a 360-degree evaluation of zero propellant points at 1 degree increments.

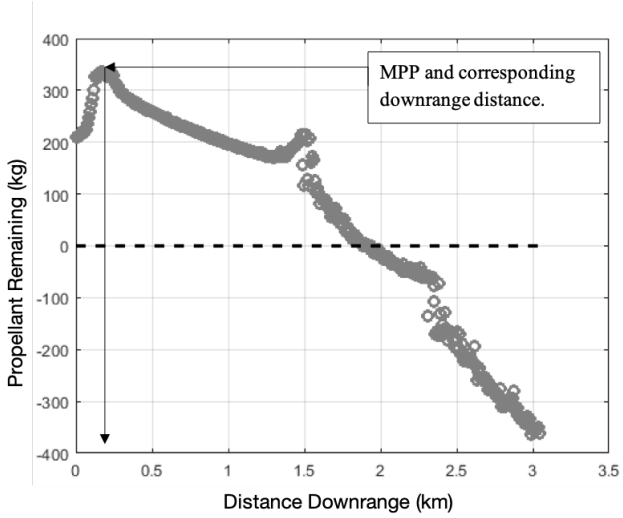

Fig. 3 Maximum Propellant Point Calculation. The downrange distance that yields the maximum propellant remaining (highlighted in the figure) is the MPP.

The ZPP evaluation begins by defining the horizontal angles, beginning at the MPP and moving radially outward, (0 to 360 degrees) and the angle increment (our default was 1 degree). A zero-finding methodology (leveraging MATLAB's *fzero*) was used to find the distance in each direction away from the MPP which resulted in zero propellant remaining. A maximum downrange distance of 3.05 km (10,000 ft.) was established to bound the algorithm and the tolerance was set to 5% of the evaluation distance. For each zero-propellant point evaluation, 100 points between the

ZPP and MPP were used for an elevation comparison. Those points with an elevation above the vehicle height are designated as unachievable and assigned a negative propellant value (i.e., -1) to categorize them as "not achievable". This elevation consideration is only conducted along the user specified angles from the MPP (Fig. 4). Critically, it does not consider the actual flight profile of the vehicle from its current position to the zero-propellant point.

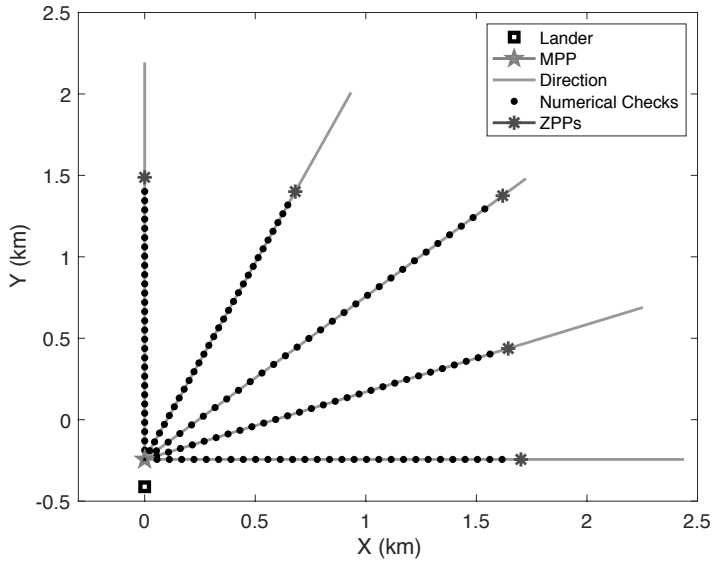

Fig. 4 Zero Propellant Point (ZPP) Calculation. For clarity, here we only show 5 directions in 22.5 degree increments in the upper right quadrant, but nominally the search is from 0 to 360 degrees in 1 degree increments.

There are additional scenarios that would yield an unachievable flight profile (defined below). If the algorithm encounters one of these scenarios, the propellant remaining for the flight simulation was again set -1, such that the search for ZPPs would treat the landing point as unachievable.

1- The landing point is outside of the limits of our predefined map.

2- Any landing points where the terrain elevation is higher than the initial vehicle height. The current guidance laws limit vehicle control to horizontal translation, vertical descent and hover, and thus does not allow for ascent to climb over terrain.

3- The vehicle's current altitude is below the elevation of the landing point. Again, since a climb profile is not implemented, the vehicle would not be able to reach the desired landing point location.

4- The vehicle's horizontal velocity is too high (>1 m/s) when reaching the landing point position and elevation. Limits are established for the horizontal velocity to ensure a safe landing.

Given the nature of the flight profiles, out along a given direction away from the MPP, there may be more than one crossing of zero propellant remaining. For example, in a given direction from the MPP, 0 - 1,000 m may yield positive propellant remaining, 1,000 – 1,100 m yields negative propellant (potentially due to terrain in that area), but then 1,100 – 1,200 m is positive again (i.e. achievable), while distance > 1,200 m are all negative. We aimed to identify the innermost ZPP, to avoid the achievability limit encompassing areas that were not achievable (i.e., identify 1,000 m instead of 1,200 m in the example above). To ensure that the first (closest to the MPP) ZPP was found, a series of numerical checks were performed along the search direction (black dots in Fig. 4 show examples of these numerical check locations). The check consists of simulated landings to a set number of distances between the MPP and the current ZPP, and evaluates the propellant remaining for each point. If any of the points yield a negative propellant remaining, then a closer downrange ZPP must exist. In the example above, if the original search for ZPPs yielded 1,200 m, but a numerical check was performed at 1,075 m, we would know 1,200 m is not the innermost ZPP. Using the previously calculated point as the new maximum distance (1,075 m in example), a new *fzero* evaluation is then performed until a ZPP is discovered (1,000 m in example). This sequence repeats until a ZPP is found where none of the numerical checks yield negative propellant (i.e., the innermost achievable limit is found in that direction). We used 30 inner numerical checks which aimed to yield accurate ZPP calculations while balancing computational requirements. The ZPP and numerical check process were then conducted out along each horizontal direction from the MPP at the predefined angle interval (Fig. 2B). Once all ZPPs were found, a polygon was created by direct connection of the ZPPs. This polygon thus describes the achievable limit for the vehicle.

Algorithm Evaluation

We aimed to assess the algorithm across a range of scenarios representative of those likely to be encountered during the final stages of landing, when achievability information is likely most critical. These scenarios defined the initial vehicle states and were motivated by the Apollo landing trajectory. This flight profile evaluation began at "low gate" (LG), a point along the landing trajectory following the pitch-over maneuver where there was visual assessment of the landing site, and was the transition point from automatic to manual control [2]. For this study two other points of interest along the trajectory were defined: terminal descent (TD), a point just after entering the constant velocity vertical descent from approximately directly above the predesignated landing site; and mid-point (MP), a point along the flight profile between LG and TD. These additional points of interest were included for a more robust analysis of

the potential benefit of providing an achievability limit, since a landing site redesignation may occur at any point in the landing phase. A complete list of initial vehicle states for all three starting scenarios is shown in Table 1.

Table 1 Initial Scenarios for ALD Generation

Parameter	Low Gate	Mid-Point	Terminal Descent
X Pos (m)	0	0	0
Y Pos (m)	-411.5	-93.3	-12.8
Vel X (m/s)	0	0	0
Vel Y (m/s)	18.3	7.2	1.5
Roll (deg)	0	0	0
Pitch (deg)	16	13.3	5.3
Height (m)	152.4	70.5	44.2
Descent Rate (m/s)	-4.9	-1.9	-0.75
Propellant (kg)	729.7	613.7	523.6

III. Results

The following sections will first provide six example achievability limits to showcase the algorithm output. We then compare the algorithm output (specifically the zero-propellant remaining contour) against the global propellant consumption calculations (depicted as top-down heat maps) to assess the accuracy of our ZPP search method (i.e., does the search method properly identify the zero propellant points?). Lastly, we present the results of a parameter sensitivity analysis, which consists of an evaluation of the effects of changing initial height (difference between vehicle altitude and landing site elevation), initial horizontal vehicle velocity, roll angle (ϕ, X-direction tilt), and pitch angle (θ, Y-direction tilt).

Example Achievability Limit Display

The achievability limit results from the lunar scenario is depicted below in Fig. 5. As described in the methods, results from three different initial scenarios are depicted: Low Gate, Mid-Point, and Terminal Descent (defined in Table 1). In each panel of Fig. 5 the center triangles represent the initial vehicle positions and the three achievability limits correspond to each of the initial scenarios (LG=red, MP=blue, TD=magenta). Panel A shows the results of the achievability algorithm without the presence of terrain (i.e., perfectly flat terrain), while panel B includes example terrain features (depicted by background topographic shading).

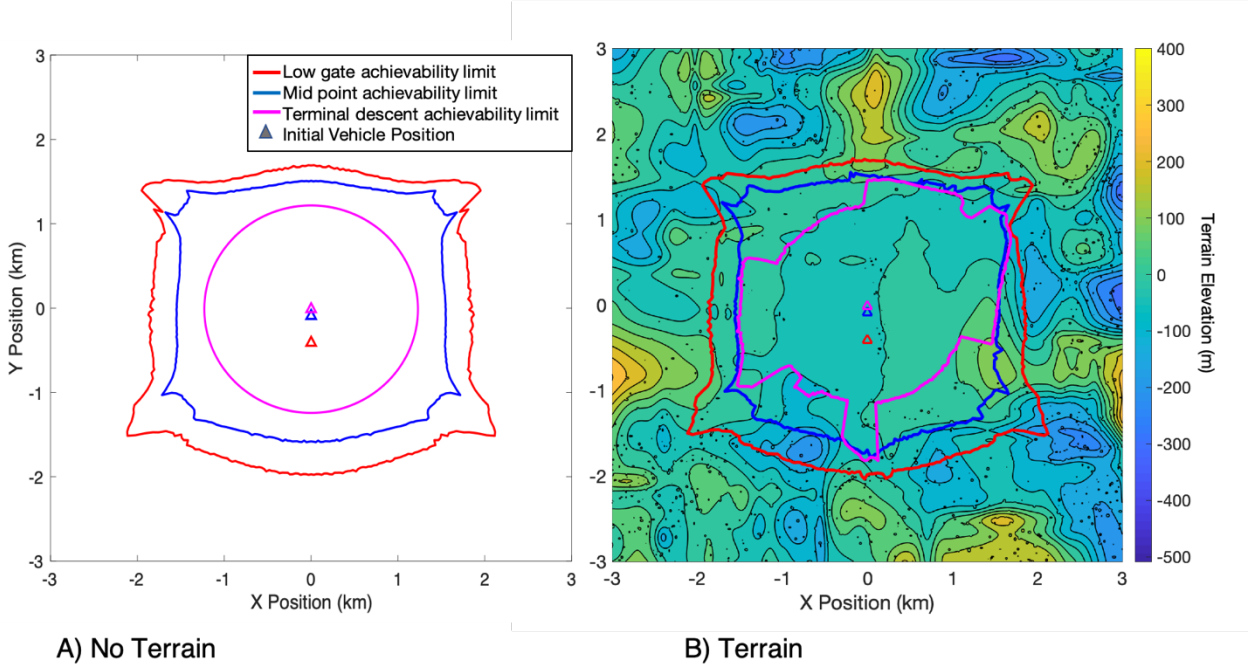

Fig. 5 Achievability Limit Display generation results for lunar scenario. Fig. 5A No terrain. Fig. 5B With example terrain shown as background topographic shading.

The achievability limit algorithm generates a contour line depicting an estimation for the zero-propellant remaining line. The area within the achievability limit line thus defines the achievable landing zone, while the area outside is unachievable. The outer contour corresponds to the Low Gate achievability limit zone. At Low Gate the vehicle is at its highest altitude (152.4 m (500 ft.)), farthest distance from the point of landing (411 m (1350 ft.)) and has the highest propellant remaining (729 kg (50 slugs)). This initial condition provides the largest achievable landing zone. Altitude, distance, and propellant remaining are incrementally decreased (Table 1) for the mid-point achievability limit (blue contour line), and terminal descent achievability limit (innermost contour line). Of note, all three achievability limit lines are shifted in the +Y-direction relative to the initial position (triangles in Fig. 5). This is due to all scenarios beginning with an initial horizontal velocity in the +Y-direction. Since the vehicle does not have an initial X-velocity component in any scenario, the achievability limit zone for the no-terrain scenarios are symmetric along the X-axis (Fig. 5A). For the Low Gate achievability limit, there are expanded regions along the 45º directions (e.g., up and to the right, up and to the left, etc.). This outcome, while non-intuitive, is produced by the vehicle having both pitch and roll angles at their respective limits (i.e., ±45 degrees) to achieve a maximum resultant horizontal velocity.

Figure 5B replicates the information in panel A, but now with an example terrain map. The terrain elevation information is marked at 100 m intervals. The contour for the LG and MP achievability limits is not substantially impacted by the terrain (compare red and blue contours in Fig. 5A vs. 5B). In these scenarios (LG and MP), the vehicle begins at a high enough altitude, such that the guidance equations adjust for the terrain elevation. However, for the TD achievability limit the polygon begins to exhibit an irregular, non-elliptical shape due to the terrain elevation. The higher terrain surrounding the vehicle actually increases the area of the achievability limit zone by decreasing the required height to descend (and thus descent time and propellant required for descent). This allows for greater horizontal travel for the same amount of initial propellant.

Propellant Remaining Validation

An analysis of propellant remaining for the lunar scenario was conducted to validate the achievability limit algorithm properly calculated the ZPPs. For each of the initial scenarios in Table 1, a propellant remaining map (i.e., the propellant remaining for each [X, Y] location, shown by color) was generated with an overlay of the algorithm achievability limit line (in white, Fig. 6).

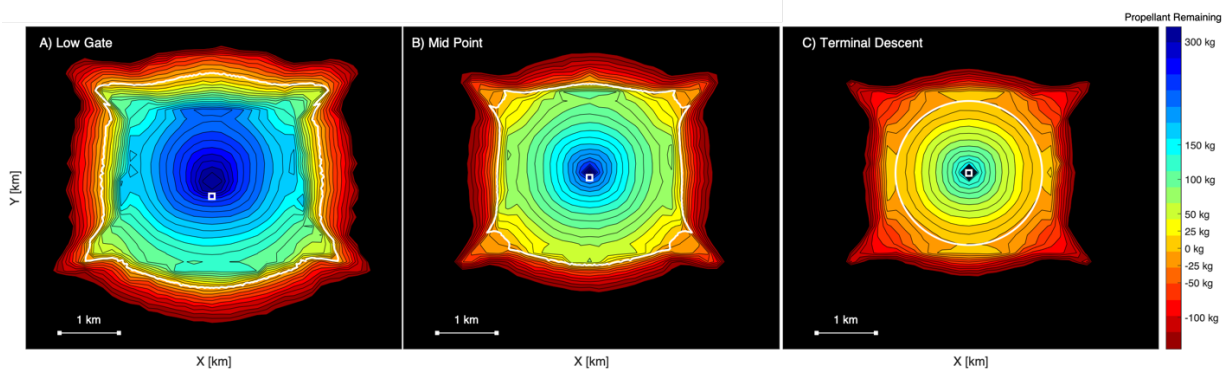

Fig. 6 Propellant remaining map for Low Gate, Mid-Point, and Terminal Descent for the no-terrain lunar scenario. White line corresponds to the algorithm generated zero propellant contour.

A numerical simulation was conducted for a landing at each point on the map out to ±3 km in X and Y directions, creating a color map of propellant remaining. Using the color bar as reference, the algorithm-derived achievability limit (white line) closely follows the locations where the propellant remaining was near 0 kg (within 2%). The 2% error was the worst case for only a few angles investigated, the error remains less than 1% for all other cases. Because calculating the propellant remaining to all points on the planetary surface is computationally demanding (requiring a numerical simulation to land at each [X, Y] position), the algorithm's search method is useful for reducing the

computational demand, while maintaining a close approximation of the zero-propellant remaining line. For example, the algorithm computes the achievability limit line in 17,196 numerical simulations for the LG scenario while the propellant remaining map depicted in Fig. 6A required 1,212,201 numerical simulations.

The propellant remaining maps in Fig. 6 demonstrate some interesting and non-intuitive behavior worth discussion. Generally, for the no-terrain scenario the propellant remaining decreases with increasing distance from the MPP (the MPP is depicted as the darkest blue location in the propellant remaining maps of Fig. 6, with propellant remaining decreasing depicted by the color bar). However, this is not always the case. For example, in Fig. 6A, traversing from the MPP in the ±X and ±Y directions (i.e., straight left, down, right, or up) there are points where the propellant remaining increases briefly (towards blue) before continuing to decrease (towards red). A two-dimensional visualization of this effect can be observed in Fig. 3 (at 1.5 km downrange) where the typical decrease in propellant remaining with increasing distance from the MPP is briefly interrupted by a slight increase. We analyzed the flight profiles to these locations and found this is due to a reduced time at hover, which in turn has a significant impact on propellant remaining since hover consumes the most propellant in the current implementation. The transition in and out of hover are a result of the specific guidance laws and vehicle dynamics implemented in the algorithm.

Parameter Sensitivity Analysis

A parameter analysis evaluation was conducted to further demonstrate the functionality of the algorithm. Four parameters were varied, relative to each of the initial scenarios described in Table 1, to understand their impact on the achievability limit area: initial vehicle height (difference between vehicle altitude and terrain elevation), horizontal forward velocity, roll angle (φ, X-direction tilt), and pitch angle (θ, Y-direction tilt). Table 2 show a list of the parameters and the values they were systematically varied to, while all other initial conditions were maintained for the specific scenario (LG, MP, or TD) defined in Table 1. For example, Fig. 7A shows achievability limits for the LG scenario, but when initial vehicle height was varied from 152.4m to 15, 30, 150, 300, or 500m. The algorithm input parameters were chosen to reflect the range of values that might be expected during typical landings. For example, vehicle roll and pitch span the range of ±45 degrees provided by the cue limits and the reference trajectory manipulates horizontal velocity across the lunar surface from 60 m/s down to 0 m/s.

Table 2 Parameter Sensitivity Analysis Values

Parameter	Values for Analysis				
Height (m)	15	30	150	300	500
Y-Velocity (m/s)	0	10	20	40	60
Roll (degrees)	-45	-23	0	23	45
Pitch (degrees)	-45	-23	0	23	45

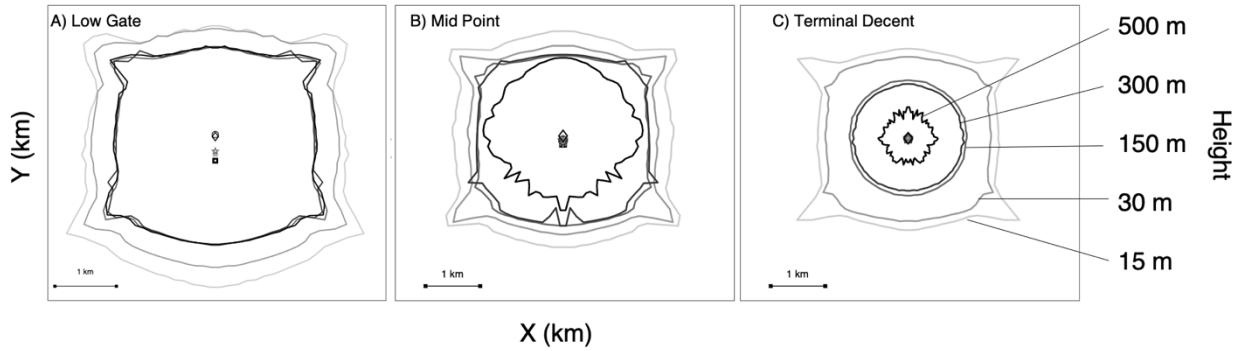

Fig. 7 Achievability Limit with varying height for Low-Gate, Mid-Point, and Terminal Descent, lunar scenario. Lighter colors correspond to a lower height, and darker colors to higher.

For each scenario (Fig. 7A (LG), 7B (MP), 7C (TD)), the vehicle height exhibited a non-linear effect on the achievability limit. For the LG scenario (Fig. 7A) no discernable benefit (increase in achievability limit zone) is gained over altitudes above 150 m (mid to dark gray contours are identical). However, comparing the achievability limit for 150 m, both 30 m and 15 m (lightest gray) show a discernable increase in the size of the achievability zone. When keeping the initial propellant constant, starting at this lower altitude allows the vehicle to use the extra propellant to increase its horizontal velocity, increasing the achievability limit zone. A similar non-linear effect of initial vehicle altitude is depicted for the MP and TD scenarios (Fig. 7B and 7C). The TD scenario (Fig. 7C) innermost contour depicts a worst-case scenario of high altitude (500 m) and low propellant (523.6 kg). The achievability limit is highly constrained in this case and most of the propellant onboard is required just to descend in a controlled manner and little propellant is available to enable any horizontal translation. The altitude where there is relatively no difference for the achievability limit, interacts with the other initial conditions: for LG (Fig. 7A) there is indifference >150 m, for MP (Fig. 7B) the indifference is from 150 - 300 m, and for TD (Fig. 7C) there is moderate indifference around 150 m. Any jagged characteristics seen in a contour are due to the tolerance settings in the algorithm and the angle increment chosen. Tighter tolerances and smaller angle increments result in smoother contour shapes but result in longer computation times.

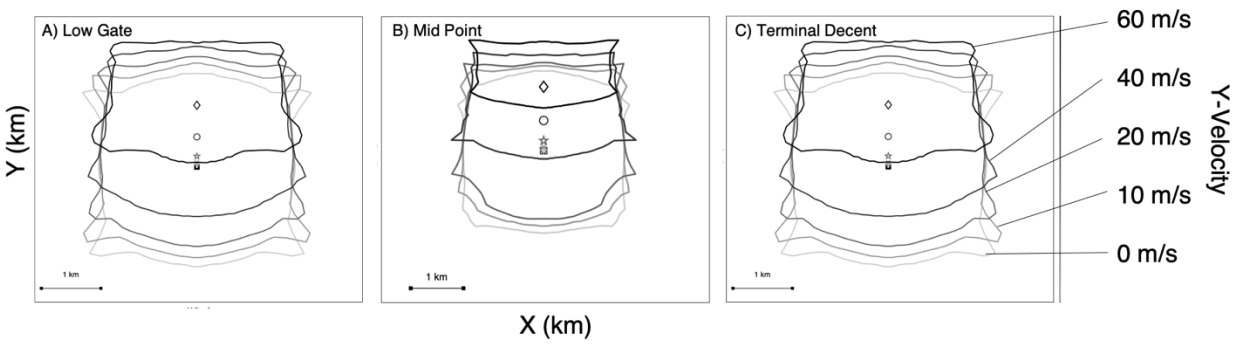

Fig. 8 Achievability Limit with varying initial velocity for Low-Gate, Mid-Point, and Terminal Descent, lunar scenario. Each contour line corresponds to the initial velocity, in the +Y-direction, in m/s.

Figure 8 depicts the impact of initial horizontal velocity on the achievability limit for each initial condition LG (Fig. 8A), MP (Fig. 8B), and TD (Fig. 8C). As expected, increasing the horizontal velocity (+Y-direction in this case) shifted the achievability limit along the velocity vector. However, the effect of horizontal velocity is non-linear. Initial horizontal velocities of 40 m/s and 60 m/s (darkest gray) had a substantial impact on the achievability limit. This is indicated in each panel with a substantial shift in the location of the MPP (depicted by the open circle for 40 m/s and diamond for 60 m/s). The MPP and subsequent achievability limit zones for 0 m/s, 10 m/s and 20 m/s (lighter grays) are relatively similar with only slight incremental changes. With the guidance equations and control laws we implemented, an initial velocity of 60 m/s is difficult to arrest in time to land directly below or behind your initial starting position. This reduces the overall size of the achievability limit zone..

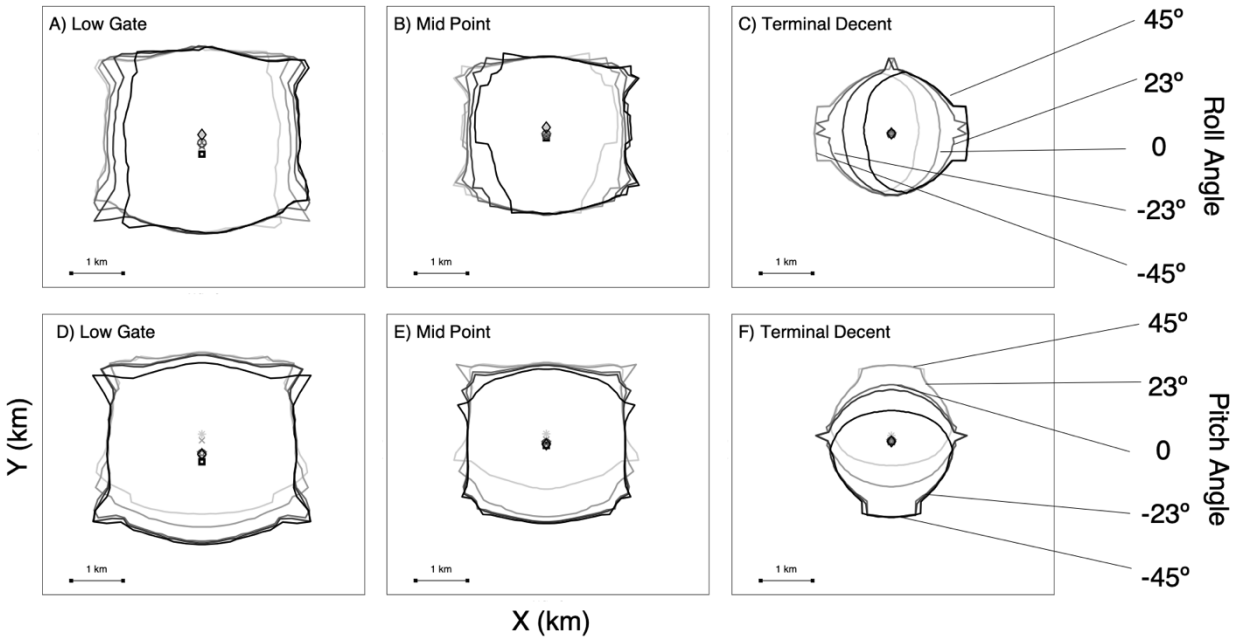

Fig. 9 Achievability Limit with varying vehicle pitch and roll for Low-Gate, Mid-Point, and Terminal Descent, lunar scenario. Roll right is positive and depicted as darker gray, pitch forward or nose down is positive and darker gray.

Finally, Fig. 9 depicts the impact of initial vehicle roll (Fig. 9A-C) or pitch (Fig. 9D-F) angle on the achievability limit for each initial condition LG (Fig. 9A and 9D), MP (Fig. 9B, and 9E), and TD (Fig. 9C and 9F). As expected, the initial vehicle tilt angle caused the achievability to shift in that direction: larger right roll corresponded to a shifting of the achievability limit to the right, and vice versa for left roll (Fig. 9A-C). Of note, the calculated MPP did not shift left or right with changing initial roll angle, as intuitively might be expected. This is a result of a limitation of the algorithm's MPP calculation (see Discussion). For pitch the achievability limit again behaved in an expected manner with the achievable zone shifting in the direction commensurate with the initial pitch angle. For the descent engine thruster vehicle, the pitch or roll angle causes a horizontal acceleration in the same direction. Thus, pitch and roll angle have a more substantial effect on the size of the achievability limit zone in the TD scenario (Fig. 9C and 9F). The upper limit of the guidance equations to command +/- 45 degrees yields initially unintuitive outcomes particularly for terminal descent (e.g., the boxy notch in light gray of panel F, corresponding to 45 degrees of pitch forward). The combination of lower initial propellant and altitude does not allow the vehicle to overcome large initial tilt angles and thus the achievability limit zone is highly biased in the direction of the vehicle's initial roll or pitch angle.

The results of the parameter analysis shown in Figs. 7-9 depict the impact of height, horizontal velocity, and pitch/roll angle on the achievability limit zone. Not depicted are any coupling effects caused by changing multiple

parameters simultaneously, beyond considering the LG, MP, and TD scenarios. Analyses where multiple parameters were varied further highlight the non-linear, complex, and often non-intuitive behavior of the achievable landing area.

IV. Discussion

This paper introduced a novel numerical algorithm to estimate the achievability limit of a planetary lander vehicle. While displaying achievability information has been proposed [5], to our knowledge this is the first effort to accurately predict that achievability limit, accounting for guidance laws, vehicle and environmental dynamics, and a model of pilot behavior. Planetary landings are highly propellant constrained. Specifically, to minimize Earth-launch costs, it is important to reduce planetary landing vehicle mass, of which propellant mass makes up a significant fraction. The achievability limit algorithm provides an accurate estimate of the achievable landing area (the area on the planetary surface that can be reached with positive propellant remaining), which in turn may benefit safety and performance of pilot control and decision making.

Key elements of the achievability limit algorithm

Our achievability limit algorithm approach (combining a pilot behavior model with physical models of the vehicle and environment) has been specifically designed such that it may be generalized to other planetary bodies and vehicle designs (with some limitations described below). Second, vehicle dynamics and subsequent guidance laws can be modified to correspond to a different vehicle design or guidance and control system. Thus, while we have demonstrated it for a lunar landing with a vehicle modeled after the Apollo LM design, it could readily be applied to Martian landings with dynamics of a future vehicle design. In fact, a simple extension to the Martian environment would just involve changing the magnitude of gravity in the algorithm's environmental model. In addition, Mars has a low density atmosphere, so a model of atmospheric drag could be included. However, the framework of the numerical simulation and search algorithm would remain identical. Further, the algorithm could be used during the design phase to understand the impact of different physical models (e.g. vehicle design and/or environmental conditions) and guidance algorithm choices on achievability and the dynamic behavior of the achievability limit. Third, the algorithm, through the propellant remaining calculation, search and verification process, reduces the calculations required to determine the boundary of the achievable limit (i.e., the ZPPs). This eliminates the computationally expensive need to calculate the propellant required for every point on the planetary surface (as shown in Fig. 7). Lastly, when used in real-time during a landing, the algorithm could accommodate changes in pilot behavior. Currently, the algorithm uses a simplified crossover model to capture the dynamics of pilot behavior. It is likely that

this simple linear model does not fully capture the complex dynamics of pilot responses. However, even if the pilot behavior differs from that expected in our model of the pilot Eq. (3), the algorithm would continue to update the depiction of the achievability limit zone using the new vehicle state resulting from the pilot's actual control inputs. This would help provide an accurate achievability limit in those instances when the pilot does not adequately follow the guidance cues or has other behaviors beyond the scope of our linear pilot model. This could occur if the pilot was maneuvering to avoid a hazardous area prior to updating the landing point in the guidance computer. The algorithm presented is unique from previous achievability limit displays in that it incorporates a model of human performance with the model of the physical system, performing numerical simulations to determine the feasibility of the landing site.

Non-intuitive (non-linear and interacting) behavior on the achievability limit

Through the parameter analysis (Figs. 7-9), we found that some of the key parameters that define the achievability limit (vehicle height, horizontal velocity, roll and pitch tilt) interact in a complex, non-intuitive manner. For example, in Fig. 5 the terrain has essentially no effect at higher altitudes, but a dramatic effect closer to landing (i.e., the Terminal Descent scenario). This complex interaction would make estimation of the achievable landing points difficult even for a well-trained pilot without some form of computational aid. Providing the pilot with a graphical display of the vehicle's achievable limit is expected to improve pilot performance during the planetary landing task by allowing the pilot to allocate cognitive and physical resources to other landing activities (e.g. communication with Earth or orbit assets, managing vehicle systems, or emergency situations) and using less cognitive resources in determining a suitable landing site. This graphical display of the achievability limit, when presented in real-time, may provide the pilot with, not only information about the achievability of their current landing site, but also indicate how long *any* potential landing site is likely to remain achievable before committing to an alternate site. The user does need to know the accuracy of the achievability limit information because depicting a landing site as achievable when it is not could lead to catastrophic failure (loss of vehicle, loss of crew, or loss of mission through abort back to orbit). At a minimum it would force a landing at a less desirable location once it becomes apparent the original planned location is actually not achievable. Alternatively, depicting a landing site as unachievable when it actually is may limit the crews' ability to meet mission objectives.

Limitations of the achievability algorithm

The current algorithm is limited to "descent-engine thruster"-type vehicles. Further, the guidance equations were kept consistent with those developed by Bilimoria [11], which did not include yaw. Yaw maneuvers have the benefit

of enabling the pilot's orientation to remain aligned with the spacecraft's velocity vector. For this reason, yaw maneuvers would help orient the pilot, but may not have a critical effect on the flight profile of the vehicle and subsequently the achievability limit zone. Additionally, the control and guidance laws as implemented do not allow the vehicle to ascend. Specifically, the vehicle is in either a descent or a vertical hover. Future iterations of the algorithm may incorporate an ability to ascend to allow the operators the ability to better avoid terrain hazards.

Next, we note the algorithm's calculation of the MPP, used as the starting point for the propellant remaining calculations, is not a global maximum of propellant remaining. Currently, the MPP is calculated along the direction of the vehicle's initial horizontal velocity vector. It is possible that the global MPP could exist along another direction. This will occur if the vehicle has a large initial roll orientation or potentially due to the terrain around the vehicle. As the MPP is only used as a starting point to find ZPPs, this limitation has no effective impact, as long as the point identified is within the theoretical achievability limit, which can then be used to search for ZPPs. However, if the MPP is negative propellant remaining, the algorithm assumes that no point on the planetary surface is achievable and no further points are considered.

Another limitation of the algorithm is the potential for a simulated flight to pass through terrain. During each ZPP calculation, the algorithm compares the current vehicle altitude to the terrain elevation at the zero propellant remaining evaluation points. However, in the algorithm we currently do not check each location along every potential flight trajectory to verify that it does not interact with the terrain at that location. This may result in the achievability limit containing areas that are achievable (in terms of propellant) but not physically feasible because they are achieved by flying through the terrain. With a terrain map, the algorithm could check whether the 3-D trajectory to each potential landing point passes through terrain, but to do this at a high spatial resolution is computationally expensive (i.e., requiring looking up and/or interpolating from the terrain map many times per simulated trajectory).

V.Conclusion

Here we presented a novel achievability landing limit approach that combined physical models of the vehicle and environment with a behavioral model of the pilot. The algorithm was implemented for a descent engine thrust lunar landing scenario, but could also be applied to hopper type vehicles which conduct repeated takeoffs and landings with constrained propellant. Additionally, the algorithm could be improved with the inclusion of onboard or offboard vehicle sensor information (LIDAR for terrain hazard detection, or satellite information for improved navigation) to

provide additional information on the *overall achievable area* (e.g., combining propellant achievable limits with hazardous terrain exclusion areas).

Knowing the achievable area enables safe selection or redesignation of the landing site and allows the vehicle to avoid unforeseen or newly discovered hazards: 1) terrain slope outside lander's safe touchdown limit, 2) rocky or rough terrain, 3) undesired geographic traits (non-traversable canyon between landing point and desired location for scientific study). The achievability limit information may also promote selection of landing sites closer to scientific points of interest and improve emergency abort-to-orbit decisions (e.g. in the event of a propellant leak, engine malfunction, control issue).

Here we have conceptualized and developed an algorithm to estimate the achievable limit for a propellant-constrained planetary lander, implemented the algorithm for a lunar landing with Apollo LM vehicle dynamics, and assessed the algorithm through simulations. Future experiments will aim to further validate the algorithm through human-in-the-loop testing. This future testing will help verify that the propellant remaining calculations provide an initial assessment of the achievability limit display, and provide feedback on the validity of our pilot behavior model. Furthermore, we aim to assess the impact of displaying achievability limit information on pilot workload and situation awareness. This will aid in quantifying the effect of the achievability limit display on utility, performance, and safety for landing site designation.

Appendix

The landing trajectory numerical simulation utilized the *ode45* MATLAB function with default tolerances. The output of the numerical simulation was vehicle state (state vector is listed below), most critically the propellant remaining at landing (m_{prop}). Described below is the set of equations use to compute the derivative of the state vector, enabling the numerical simulation. These equations are based upon the equations provided by Bilimoria [11], but have been generalized to apply to any situation. Where applicable, the specific values used to validate the achievability limit algorithm are included for the lunar landing scenario.

After Bilimoria [11], a generic descent engine lunar vehicle was modeled similar to the Apollo LM lunar lander. The lunar vehicle had an initial mass at Low Gate of 7925 kg (includes 730 kg of propellant). This mass varies with propellent consumption from the descent engine thruster. The moments of inertia and vehicle center of mass are considered constant during the landing phase of interest, Low Gate to touchdown. The descent engine thruster propellant has a specific impulse (Isp) of 311 seconds. During the landing phase the descent engine thrust is used to control the descent rate and to provide coarse horizontal trajectory corrections through vehicle pitch and roll changes. The descent engine is not gimballed and the thrust is assumed to pass directly through the vehicle's center of mass. Details of the control algorithm are provided in the main text and were modelled after Bilimora [11].

Vehicle State Vector

The vehicle state vector is defined by the nine variables described below. Of note, as in the Bilimoria equations, vehicle yaw (rotation about the z-axis) is not included in the algorithm.

Position	[y , x]	m (relative position from vehicle to landing site)
Velocity	[V_y, V_x]	m
Pitch	$[\theta]$	degrees
Roll	$[\phi]$	degrees
Altitude	$[h]$	m
Vertical Velocity	$[\dot{h}]$	m/s
Mass (prop + inert)	$[m]$	kg

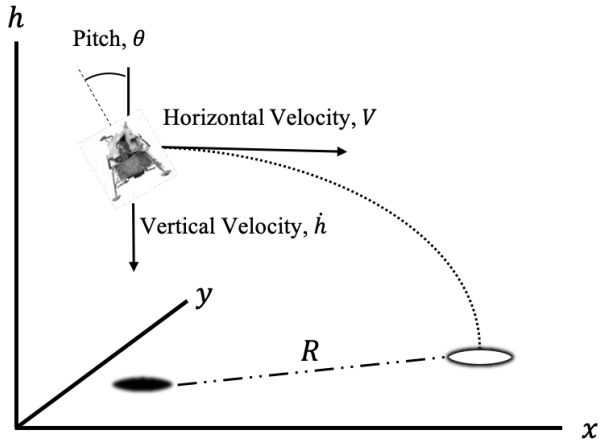

Fig. A1 Depiction of Vehicle States

In Bilimoria's implementation [11], the vehicle's initial range from the landing site (R_0) was determined by the landing scenario. The vehicle's range at any given point along the trajectory is $R = \sqrt{x^2 + y^2}$. To generalize the equations below, we added the following algorithm logic to handle R/R_0. In summary, to satisfy the guidance equations for the reference trajectory (h^*) and horizontal velocity (V_y^G and V_x^G) below, R/R_0 cannot equal e^2, due to the natural logarithm in those equations. This may be violated (R could become larger than R_0) when the pilot overshoots the original landing site, does not accurately follow the guidance cues, or redesignates to a new landing site. To account for this, the algorithm checks if $R/R_0 > 1$, and if exceed, then the algorithm resets $R_0 = R$, using the current range, and uses that going forward in the landing simulation. We use '1', to ensure that R_0 is reset prior to R/R_0 exceeding e^2. This addition ensures the simulation is stable for the scenarios described above (e.g., pilot overshoot) that may occur.

Constants for Guidance Equations

A series of constants are used throughout Bilimoria's equations (listed below, but provided as exact numbers in [11]). These constants were determined by the unique landing scenario chosen. To accommodate changes to the landing scenario, the constants were changed to be a function of the height at a specified low gate (h_{LG}) and terminal descent (h_{TD}), as well the respective vertical velocity for those regions ($\dot{h}_{LG}$) for low gate and ($\dot{h}_{TD}$) for terminal descent. For this study the descent rate decreases linearly from 5 m/s (16.4 fps) at LG altitude to 1 m/s (3 fps) at TD altitude and then remains constant at 1 m/s (3 fps) until touchdown. The heights at LG and TD were set to 152.4 m

(500 ft.) and 45.72 m (150 ft.) respectively. Future scenarios (e.g., Mars landings) could use this same framework for guidance equations, but now specify different values for the height and descent rates at LG and TD.

$$a = \left(\dot{h}_{LG} - \dot{h}_{TD}\right)/(h_{LG} - h_{TD}) \quad (A1)$$

$$b = -a \times h_{TD} + \dot{h}_{TD} \quad (A2)$$

$$C = -b/a \quad (A3)$$

$$k = h_{LG} - C \quad (A4)$$

$$q = \frac{(2\times a)}{ln(k/(h_{TD}-C))} \quad (A5)$$

Vertical Velocity Guidance

The guidance laws presented were developed by Bilimoria to guide the pilot along a reference trajectory from approach to touchdown [11]. The reference trajectory height (h^*) is a function of the initial conditions and is given by the following equation

$$h^* = C + \left(\frac{(h_{TD}-C)^{ln\sqrt{R/R_0}}}{k}\right)^{\frac{1}{\left(ln\sqrt{R/R_0}\right)-1}} \quad (A6)$$

The upper limit of h^* was set to 167.6 m (550 ft.), just above the low gate initial height of 167.6 m (500 ft.). In the case where the vehicle is not on the reference trajectory ($h \neq h^*$), but above the TD altitude, the commanded vertical velocity guidance is given by

$$\dot{h}^G = (a\, h^* + b) + (h^* - h)/\tau_h \quad (A7)$$

Where h is the current altitude, h^* is provided by Eq. (A6) and τ_h is $1/K_h$ = 25 seconds (K_h is the feedback gain). The feedback gains used in this study were the same ones used by Bilimoria. Vertical velocity guidance ($\dot{h}^G$) is set to a constant value of -1 m/s when h drops below the TD altitude.

Finally, to make the vertical guidance more robust, we added a third mode. In the case where the vehicle is below the TD altitude, but not within range to begin the descent ($R>R_{lim}$ where R_{lim} = 3 m), the vehicle will be guided to enter a hover, where vertical velocity $\dot{h}^G = 0$. Vertical velocity guidance ($\dot{h}^G$) is bounded by 0 and -10 m/s to limit the effects of large altitude errors.

Thrust Command

The descent engine thrust acts along the negative vehicle z-axis. The thrust command consists of the primary thrust (T^0_{cmd}), the force whose vertical component balances the vehicle's weight while compensating for roll and pitch, and secondary thrust (ΔT_{cmd}), the thrust increment derived from pilot input.

$$T^0_{cmd} = \frac{mg}{\cos\theta \cos\phi} \tag{A8}$$

$$\Delta T_{cmd} = \frac{m}{\cos\theta \cos\phi}\frac{(\dot{h}^G - \dot{h})}{\tau_T} \tag{A9}$$

During the approach phase from LG to touchdown thrust is controlled by a throttle between 10 and 60% of the maximum value of 10,000 lb (44,482 N). The descent rate is regulated within a deadband of 0.1 fps by a proportional feedback controller with a time constant $\tau_T = 1.5$ seconds. For stability, (ΔT_{cmd}) has a deadband in which it is 0 when the difference in $\dot{h}^G$ and $\dot{h}$ is less than 0.1 mps.

We assumed instantaneous thrust capability, such that actual thrust (T) is defined by

$$T = T_{cmd} = T^0_{cmd} + \Delta T_{cmd} \tag{A10}$$

Given the vertical guidance command and thrust, vertical acceleration can be determined by

$$\ddot{h} = \frac{T}{m}\cos\theta \cos\phi - g \tag{A11}$$

Vehicle Mass

The total vehicle mass changes as a function of thrust and the resulting propellant flow from the descent engine thruster where

$$\dot{m}_{fuel} = \frac{T}{I_{sp} g_0} \tag{A12}$$

And thus, the change in vehicle mass is simply the rate of the propellant mass burn

$$\dot{m} = -\dot{m}_{prop} \tag{A13}$$

Horizontal Velocity Guidance

Velocity guidance was determined with respect to the vehicle's horizontal position [x,y].

$$V^G_y = yq\left(1 - ln\left(\sqrt{R/R_0}\right)\right. \tag{A14}$$

$$V_x^G = xq(1 - ln(\sqrt{R/R_0}) \quad (A15)$$

And thus, the horizontal components of acceleration guidance when the vehicle is not on the reference trajectory $V \neq V^G$ is given by

$$a_y^G = \left(\frac{V_y V_y^G}{y} - \frac{yq\dot{R}}{2R}\right) + (V_y^G - V_y)/\tau_V \quad (A16)$$

$$a_x^G = \left(\frac{V_x V_x^G}{y} - \frac{yq\dot{R}}{2R}\right) + (V_x^G - V_y)/\tau_V \quad (A17)$$

Where τ_V = 1/K_v = 8 seconds. The acceleration component is set to zero when the vehicle is essentially directly above the landing site in each direction (i.e., |x| is less than 0.1 m, same for y).

Roll and Pitch Guidance

Tilting the vehicle creates an acceleration in the horizontal plane due to tilting of the descent engine thrust force. This impacts both roll and pitch guidance which are determined by

$$\phi^G = sin^{-1}\left(\frac{-m}{T}\left(a_y^G \sin\varphi\right) - a_x^G \cos\varphi\right) \quad (A18)$$

$$\theta^G = sin^{-1}\left(\frac{-m}{T\cos\phi^G}\left(a_y^G \cos\varphi\right) - a_x^G \sin\varphi\right) \quad (A19)$$

Roll and pitch guidance are constrained to ±45° to limit the effects of large trajectory errors.

Derivatives

The numerical simulation utilizes the above vehicle state equations and their derivatives below

$$\dot{y} = V_y \quad (A20)$$

$$\dot{x} = V_x \quad (A21)$$

$$\dot{V_y} = a_y = \frac{-T}{m}(\cos\phi \sin\theta \cos\varphi + \sin\phi \sin\varphi) \quad (A22)$$

$$\dot{V_x} = a_x = \frac{-T}{m}(\cos\phi \sin\theta \cos\varphi - \sin\phi \sin\varphi) \quad (A23)$$

$$\dot{\theta} = K_\theta(\theta^G - \theta) \quad (A24)$$

$$\dot{\phi} = K_\phi(\phi^G - \phi) \quad (A25)$$

Where $K_\theta = K_\phi = 0.5$ seconds. Both pitch and roll rate were limited to 5 degrees/second to simulate limited control authority of the reaction control system in producing attitude changes.

The equations above were used any time the algorithm needed to run a numerical simulation of the flight profile. For example, simulations were performed during the MPP calculation and during the zero propellant remaining function to calculate the ZPPs.

References

[1] Brady, T. and Paschall S., "The Challenge of Safe Lunar Landing," in *2010 IEEE Aerospace Conference*, Big Sky, MT, USA, 2010, pp. 1–14. doi: 10.1109/AERO.2010.5447029

[2] Bennet F., "Apollo Lunar Descent and Ascent Trajectories," presented at the *AIAA 8th Aerospace Sciences Meeting*, New York, NY, 1970

[3] Cohanim, B. E., and Collins, B. K. "Landing Point Designation Algorithm for Lunar Landing," *Journal of Spacecraft and Rockets*, Vol. 46, No. 4, pp. 858–864, Jul. 2009. doi: 10.2514/1.42002

[4] Paschall, S. C., Brady, T., Cohanim, B. E., and Sostaric, R. "A Self Contained Method for Safe & Precise Lunar Landing," in *2008 IEEE Aerospace Conference*, Big Sky, MT, USA, 2008, pp. 1–12. doi:10.1109/AERO.2008.4526298

[5] Stimpson, A. J., Clark, T. K., Young, L. R., Duda, K. R., and Oman, C. M. "Effects of an achievability display during simulated lunar landings," in *2011 Aerospace Conference*, Big Sky, USA, 2011, pp. 1–11. doi: 10.1109/AERO.2011.5747223

[6] Cummings, M., Wang, E., Smith, C., Marquez, J., Duppen, M., and Essama, S. "Conceptual Human-System Interface Design for a Lunar Access Vehicle," MIT HAL for Draper Labs, Cambridge, MA 2005.

[7] Hainley, C. J., Duda, K. R., Oman, C. M., and Natapoff, A. "Pilot Performance, Workload, and Situation Awareness During Lunar Landing Mode Transitions," *Journal Spacecraft and Rockets*, Vol. 50, No. 4, pp. 793–801, Jul. 2013. doi: 10.2514/1.A32267

[8] McCandless, J. W., McCann, R.S., Berumen, K.W., Gauvain, S.S., Palmer, V.J., Stahl, W.D., Hamilton, A.S. "Evaluation of the Space Shuttle Cockpit Avionics Upgrade (CAU) Displays," *Human Factors and Ergonomics Society*, Vol. 49, No. 1, pp. 10-14, 2005. doi: 10.1177/154193120504900104

[9] Chua, Z. K., Major, L. M., and Feigh, K. M. "Modeling Cockpit Interface Usage During Lunar Landing Redesignation," *International Symposium on Aviation Psychology,* Dayton, OH 2009.

[10] Needham, J. M., "Human-Automation Interaction for Lunar Landing Aimpoint Redesignation," MIT Draper Labs, Cambridge, MA 2008.

[11] Bilimoria, K. D. "Effects of Control Power and Guidance Cues on Lunar Lander Handling Qualities," *Journal of Spacecraft and Rockets*, Vol. 46, No. 6, pp. 1261–1271, Nov. 2009. doi: 10.2514/1.40187

[12] Mcruer, D. and Weir, D. H. "Theory of Manual Vehicular Control," *Ergonomics*, Vol. 12, No. 4, pp. 599–633, Jul. 1969. doi: 10.1080/00140136908931082