OPTIMAL LUNAR LANDING AND RETARGETING USING A HYBRID CONTROL STRATEGY

Daniel R. Wibben,[*] Roberto Furfaro[†], Ricardo G. Sanfelice[‡]

A novel non-linear spacecraft guidance scheme utilizing a hybrid controller for pinpoint lunar landing and retargeting is presented. The development of this algorithm is motivated by a) the desire to satisfy more stringent landing accuracies required by future lunar mission architectures, and b) the interest in analyzing the ability of the system to perform retargeting maneuvers during the descent to the lunar surface. Based on Hybrid System theory, the proposed Hybrid Guidance algorithm utilizes both a global and local controller to bring the lander safely to the desired target on the lunar surface with zero velocity in a finite time. The hybrid system approach utilizes the fact that the logic and behavior of switching guidance laws is inherent in the definition of the system. The presented case of a hybrid system utilizes a global controller that implements an optimal guidance law augmented with a sliding mode to bring the lander from an initial state to a predetermined reference trajectory and an LQR-based local controller to bring the lander to the desired point on the lunar surface. The individual controllers are shown to be stable in their respective regions. The behavior and performance of the Hybrid Guidance Law (HGL) is examined in a set of Monte Carlo simulations under realistic conditions. Results demonstrate the capability of the hybrid guidance law to reach the desired target point on the lunar surface with low residual guidance errors. Further, the Hybrid Guidance Law has been applied to the problem of retargeting in order to examine the performance of the algorithm under such conditions.

INTRODUCTION

The problem of achieving pinpoint landing accuracy on the lunar surface presents new challenges which may require the development of novel and more advanced guidance algorithms. Such new class of guidance algorithms must bring the spacecraft to the lunar surface at the desired point with zero velocity with unprecedented precision to meet new, more stringent landing requirements. The lander's Guidance, Navigation, and Control (GNC) system implements several on-board functions to bring the spacecraft safely to the lunar surface and with the right orientation. Most of the guidance algorithms currently available date back to the Apollo-era[1,2]. Mostly based on linear control theory, these algorithms may not be able to satisfy the higher degree of flexibility imposed by future mission architectures (e.g. ability to land anywhere on the lunar surface, ability to perform retargeting in real-time). Over the past two decades,

[*] Graduate Student, Department of Systems and Industrial Engineering, University of Arizona, 1127 E. James E. Roger Way, Tucson, Arizona, 85721, USA

[†] Assistant Professor, Department of Systems and Industrial Engineering, University of Arizona, 1127 E. James E. Roger Way, Tucson, Arizona, 85721, USA

[‡] Assistant Professor, Department of Aerospace and Mechanical Engineering, University of Arizona, 1130 N. Mountain Ave., Tucson, AZ 85721.

advancements in non-linear control theory have brought about innovative and more robust guidance laws for missiles. For example, Yanushevsky et. Al. showed that a Lyapunov approach can be effectively employed to determine a guidance law that yields superior performance in missile targeting when compared to the more conventional Proportional Navigation (PN) guidance laws.[3] Hybrid control theory, another recent advancement, allows for the modeling of systems that incorporate both continuous and discrete dynamics, gaining new insight on the behavior of dynamical systems.[4] The problem of lunar landing features only continuous time dynamics, however the combination of multiple continuous time controllers introduces discrete behavior into the system. The use of multiple controllers may allow for the utilization of controllers that work well only in certain regions, i.e. the combination of a controller that works well globally with one that is more efficient near the desired target point.[5] However, very little has been done to apply such non-linear methods to the development of landing algorithms for precision and/or pinpoint lunar landing. For example, Chomel and Bishop proposed a targeting program capable of generating on-line reference trajectories based on analytical gravity-turn solutions and a real-time non-linear guidance algorithm based on Lyapunov second methods.[6] Furfaro et. al. proposed a set of non-linear guidance algorithms based on recent advancements of sliding control theory.[7] In both cases, the guidance problem utilized only a single guidance law, which may not have the flexibility or may be less optimal than a combination of proper controllers. In addition, using a single guidance law may limit the number of potential landing sites based on the reachability of the system whereas multiple controllers can utilize the 'catch-and-throw' methodology of certain controllers 'throwing' the system open-loop to a neighborhood that allows for successful landing near the desired target point.[8]

In this paper, we introduce a novel, robust guidance algorithm for lunar pinpoint landing that incorporates a hybrid control strategy. The algorithm, called the Hybrid Guidance Law (HGL) utilizes the idea of a switching system, which combines both local and global controllers, for the landing guidance problem. The switching system is one that has different continuous dynamics based on a switching signal. A switching logic between the two control laws is implemented in a hybrid controller to develop a more robust guidance law, with the potential for better performance than what can be achieved with a single guidance law. This is closely related to the "throw-and-catch" control strategy, which uses a hybrid approach by combining local controllers to steer trajectories to the desired point and a global controller that is capable of steering trajectories to a neighborhood of the desired point so that the local controller may be used.[8] The chosen guidance laws are two that have been seen previously in the literature and are familiar to the authors. First, the global guidance law uses an algorithm named Optimal Sliding Guidance (OSG). This algorithm determines an optimal acceleration command and augments it with a sliding mode to provide robustness against perturbations.[7] The OSG law is considered as the 'throw' portion of the throw-and-catch control strategy. The guidance law 'throws' the lander from an initial state to a state near a pre-defined reference trajectory. The second law is based on a Linear Quadratic Regular that follows the reference trajectory. This is the 'catch' part of the throw-and-catch control strategy as this guidance law 'catches' the lander and forces the system to track the reference trajectory to the target point. Both of these guidance laws have been previously and more thoroughly developed elsewhere, so this paper will only introduce the individual control laws to the reader. A set of Monte Carlo simulations is included to demonstrate the performance of the guidance algorithm under realistic conditions. Further, in the scenario that information learned during the descent show that the original site is not satisfactory for landing due to any number of conditions, including safety hazards such as boulders, it may be necessary for the guidance algorithm to actively target a different landing site. The inherent ability of the guidance law to guide the lander to a new landing site is analyzed in an additional set of Monte Carlo simulations.

GUIDANCE PROBLEM FORMULATION

We consider the lunar descent and landing guidance problem that can be formulated as follows: given the current state of the spacecraft, determine a real-time acceleration command program that brings the spacecraft to the target point on the lunar surface with zero velocity.

Guidance Model: Equations of Motion

The fundamental equations of motion of a spacecraft moving in the lunar gravitational field can be described using Newton's law. In a drag-free central force field, the only forces acting on the body are the gravitational force from the moon and the thrust forces generated by the vehicle's propulsion system, as shown in Figure 1.

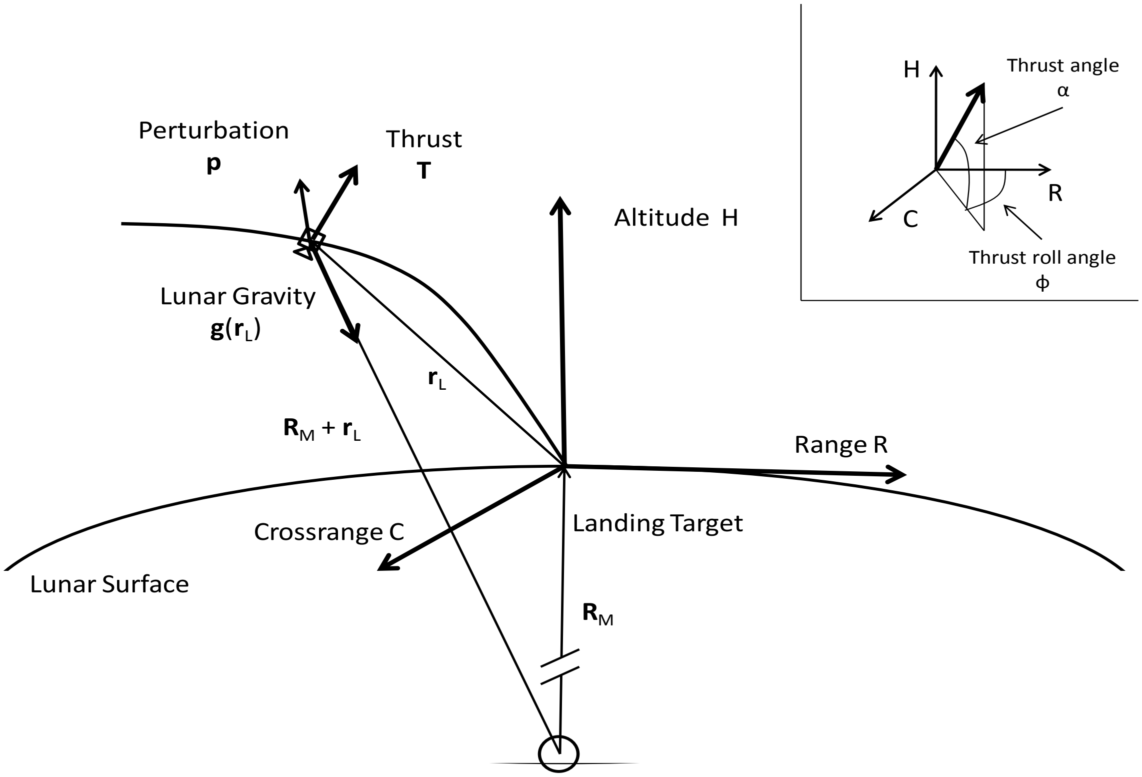

Figure 1. Guidance Reference Frame and Free-Body Force Diagram for the Lunar Lander during the Powered Descent to the Designated Target

Assuming a system with variable mass, the equations of motion can be written as follows:

$$\dot{\boldsymbol{r}}_L = \boldsymbol{v}_L \tag{1}$$

$$\dot{\boldsymbol{v}}_L = \boldsymbol{g}(\boldsymbol{r_L}) + \frac{\boldsymbol{T}}{m_L} = \boldsymbol{g}(\boldsymbol{r_L}) + \boldsymbol{a_c} \tag{2}$$

$$\frac{d}{dt}m_L = -\frac{||\boldsymbol{T}||}{I_{sp}g_0} \tag{3}$$

Here, $\boldsymbol{r}_L$ and $\boldsymbol{v}_L$ are, respectively, the position and velocity of the lander with respect to a coordinate system with origin on the lunar surface, $\boldsymbol{a}_c$ is the commanded acceleration vector, $\boldsymbol{g}_L(\boldsymbol{r})$ represents the gravitational acceleration vector of the moon, $\boldsymbol{T}$ is the commanded thrust vector, m_L is the mass of the lander, I_{sp} is the specific impulse of the spacecraft's thrusters, and g_0 is the constant gravitational parameter. If $\boldsymbol{r}_L = \left[r_x, r_y, r_z\right]^T$ and $\boldsymbol{v}_L = \left[v_x, v_y, v_z\right]^T$, where the *x*, *y*, and *z* directions represent the respective coordinates of the lander, the equations of motions can be written by components as follows:

$$r_x = v_x \tag{4}$$

$$r_y = v_y \tag{5}$$

$$r_z = v_z \tag{6}$$

$$v_x = a_{c_x} \tag{7}$$

$$v_y = a_{c_y} \tag{8}$$

$$v_z = -g_L + a_{c_z} \tag{9}$$

Clearly, the considered mathematical model is a 3-DOF model with varying mass. This model is employed to simulate spacecraft descent dynamics driven by the proposed guidance laws which require the formulation of an appropriate guidance model as discussed in the next sections.

HYBRID LANDING GUIDANCE CONTROL LAW DEVELOPMENT

Generally, the dynamics of the system are such that the guidance law can be formulated in a hybrid framework: the system can utilize two different continuous time controllers expressed in a single set of equations to provide flexibility and performance that are not possible with just one controller. The local controller will have the capability to steer trajectories to the desired final target on the lunar surface from a particular reference point while the global controller will be able to steer all trajectories to this reference point. Figure 2 helps to clarify the type of switching system being described. In the specific case of this paper, globally the lander will utilize the Optimal Sliding Guidance (OSG) until it has reached a point that is close to the reference trajectory, at which point it will switch to the Linear Quadratic Regulator (LQR) control-based guidance, which is the local controller. Each of these guidance laws will be formally introduced in later sections.

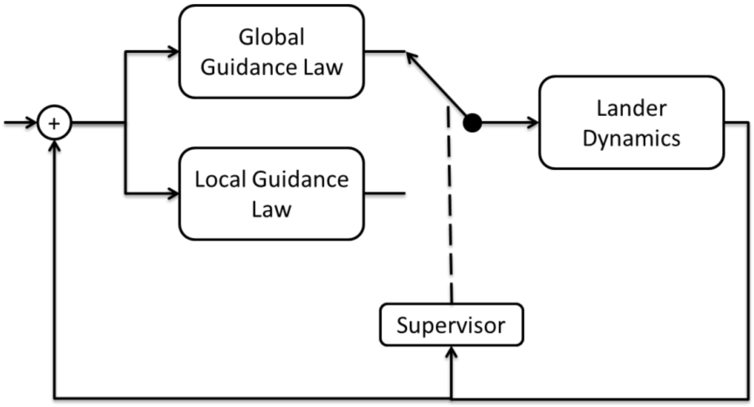

Figure 2. Closed-loop system combining local and global guidance laws

In order to model the switching behavior of the guidance laws in a proper hybrid system framework, the dynamics of the system must be expressed solely as functions of the state variables. Therefore, new state variables are introduced. First, the guidance model used for the derivation of the guidance laws will use a constant mass system, i.e. Eq. (3) does not apply. Due to the fact that the lander will be taken to the target point on the lunar surface if it tracks the reference trajectory exactly, the states for position and velocity are set to be the error difference between the current state and the desired state on the reference trajectory. Next, due to the time dependence of the global guidance law, a timer is introduced. Finally, a switching variable is included to model the switching behavior of the system. The new hybrid system state is defined as follows:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{r}_L - \boldsymbol{R}_{RG} \\ \boldsymbol{v}_L - \boldsymbol{V}_{RG} \\ q \\ \tau \end{bmatrix} = \begin{bmatrix} \boldsymbol{\varepsilon}_r \\ \boldsymbol{\varepsilon}_v \\ q \\ \tau \end{bmatrix} \tag{10}$$

where the new state variable q is the switching logic variable, τ is the timer, position and velocity are as defined in Eq. (1-2), and $\boldsymbol{R}_{RG}$ and $\boldsymbol{V}_{RG}$ are the position and velocity of the reference trajectory (defined in a later section). The value of q specifies which guidance law is currently being used, with $q \in \{1,2\}$ where $q = 1$ represents the global law and $q = 2$ represents the local. This switching property of the variable q introduces the discrete time dynamics into the system. The new state now leads to the formal definition of the hybrid system $\boldsymbol{\mathcal{H}}$.

Formal Hybrid System Definition

For this problem, the hybrid system is defined as:

$$\boldsymbol{\mathcal{H}} = (C, \boldsymbol{F}, D, \boldsymbol{G}) \tag{11}$$

where $\boldsymbol{F}$ is the flow map, $\boldsymbol{G}$ is the jump map, C is the flow set, and D is the jump set. That is, C is defined as the set of states of $\boldsymbol{x}$ in which the system will follow the continuous time dynamics defined by $\boldsymbol{F}$. Likewise, D and $\boldsymbol{G}$ are defined similarly for the discrete time dynamics. In the sense of the landing guidance problem, this generally translates to D defining the states in which the guidance law that is used by the system will change.

The flow map $\boldsymbol{F}$ follows the equations of motion seen in Eq. (1-2) to model the continuous time dynamics of the system, with a slight augmentation due to the new state variables defined in Eq. (10).

$$\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{a_c}) := \begin{bmatrix} \mathbf{v}_{\mathrm{L}} - \boldsymbol{V}_{RG} \\ \boldsymbol{g}_L + \boldsymbol{a}_{c_q} - \boldsymbol{A}_{RG} \\ 0 \\ 2 - q \end{bmatrix} \tag{12}$$

where $\boldsymbol{A}_{RG}$ is the acceleration defined by the reference trajectory, and the commanded acceleration, $\boldsymbol{a}_{c_q}$, will change depending on the value of the switching variable. Note that q does not change in continuous time due to the chosen controller remaining constant between jumps. The addition of the timer variable τ allows for the use of time varying controllers in the hybrid framework. By recasting them as functions of the timer variable τ as opposed to t, the system is then only dependent on state variables, the state of the reference trajectory, the input $\boldsymbol{a}_c$, and the constant value for the lunar gravity. The form of the dynamics of τ are such that while the global controller is in use, τ has equivalent dynamics to t, while it will be set to zero when the local controller is being used. This is possible specifically due to the local controller that has been chosen and prevents τ from becoming unbounded. By forming the system in this way, the standard analysis used for hybrid systems is applicable.

Next the jump map $\boldsymbol{G}$, i.e. the discrete dynamics of the system**,** is defined. When the state of the lander is in the set D (defined later in this section), the system will jump according to the dynamics of $\boldsymbol{G}$**.** During a jump, the only variables in the system that experience discrete dynamics are the switching variable q, as the system switches between the separate guidance laws during jumps, and the timer τ,which is reset to zero. The value of $3 - q$ is used so that when the global controller is being used $(q = 1)$ and the system jumps, the new value of q $(q = 2)$ will represent the local controller, and vice versa.

$$\boldsymbol{G}(\boldsymbol{x}) \coloneqq \begin{bmatrix} \boldsymbol{\varepsilon}_{\mathrm{r}} \\ \boldsymbol{\varepsilon}_{v} \\ 3 - q \\ 0 \end{bmatrix} \tag{13}$$

Finally, the flow and jump set, C and D, respectively, are defined. In the definition of the lunar landing problem, there are two clear criteria that define where the system will flow, i.e. follow continuous dynamics as defined by Eq. (12) that are based on the usage of the guidance law. In order for the global guidance law to be used $(q = 1)$, the state of the lander must be far from the reference trajectory. In addition, due to the form of the global guidance law chosen, a constraint must be included on the timer τ. This leads to the definition of the flow set for the global controller as:

$$C_1 = \{\boldsymbol{x}\colon \|\boldsymbol{\varepsilon}_r\| \geq \Lambda_{r_1}, \|\boldsymbol{\varepsilon}_v\| \geq \Lambda_{v_1}, q = 1, t_f - \tau \geq \delta\} \tag{14}$$

Here, Λ_{r_1} and Λ_{v_1} are parameters that define the distance of the state from the reference trajectory at which the guidance law will switch, t_f is a parameter that defines the final time for convergence of the global guidance law, and $0 < \delta \ll 1$ is a parameter that prevents the global guidance law from becoming undefined. Similarly, a flow set for the use of the local controller can be defined with the knowledge that it will be used when the state is near the reference trajectory:

$$C_2 = \{\boldsymbol{x}\colon \|\boldsymbol{\varepsilon}_r\| \leq \Lambda_{r_2}, \|\boldsymbol{\varepsilon}_v\| \leq \Lambda_{v_2}, q = 2, \tau = 0\} \tag{15}$$

Here, $\Lambda_{r_2} > \Lambda_{r_1}$ and $\Lambda_{v_2} > \Lambda_{v_1}$ are parameters that define the distance from the reference trajectory the state is allowed to stray while using the local controller. The constraints on Λ_{r_2} and Λ_{v_2} are in place to allow for hysteresis between the two sets, i.e. as the jump map G does not change the lander state (position and velocity), this prevents constant switching. The complete system flow set is then defined as the union between these two sets:

$$C = C_1 \cup C_2 \tag{16}$$

Using the same logic, it is easy to define the jump sets D_1 and D_2 as the complement to the flow sets, describing the states at which the system will switch between the guidance laws:

$$D_1 = \{\boldsymbol{x}\colon \|\boldsymbol{\varepsilon}_r\| \leq \Lambda_{r_1}, \|\boldsymbol{\varepsilon}_v\| \leq \Lambda_{v_1}, q = 1, t_f - \mathrm{\tau} < \delta\} \tag{17}$$

$$D_2 = \{\boldsymbol{x}\colon \|\boldsymbol{\varepsilon}_r\| \geq \Lambda_{r_2}, \|\boldsymbol{\varepsilon}_v\| \geq \Lambda_{v_2}, q = 2, \tau = 0\} \tag{18}$$

$$D = D_1 \cup D_2 \tag{19}$$

In the nominal case, the system will flow, following the equations of motion, under the influence of the global guidance law until the lander's state is sufficiently close to the predetermined state of the reference trajectory. At this time, the system will jump and change the guidance scheme to that of the local law and the lander will track the reference trajectory to the desired target point. However, the hybrid system is set up in such a way that it has the capability to deal with off-nominal conditions. In particular, if the lander is forced to target a new position on the lunar surface during the descent due to hazards or other undesired conditions at the initial landing site, the hybrid system will utilize the global guidance law to enter a neighborhood of a reference trajectory that will bring the spacecraft to the secondary landing site.

Let us now introduce both the local and global guidance laws that are chosen for demonstration in this paper. These laws are used simply to provide an example of laws that can be used in the hybrid system framework, and are chosen due to their familiarity to the authors.

GLOBAL AND LOCAL GUIDANCE LAWS DEVELOPMENT

The goal of this paper is to demonstrate the performance of a guidance law based on hybrid control theory, specifically in the event of a necessary retargeting maneuver. This will be done through the use of two separate guidance algorithms: one for use globally and one that will be used when the lander's state is near a pre-determined reference trajectory. The chosen global guidance law will follow a ZEM-ZEV Optimal Sliding Guidance (OSG) approach, while the chosen local law used when near the reference trajectory will be a LQR-based guidance law. Both of these guidance laws, as well as the details on the formulation of the reference trajectory, are introducted in the following sections. Note that in the development of the individual controllers, all functions that are traditionally a function of time are expressed here as a function of the hybrid system timer variable τ in accordance with the definition of the hybrid system seen in the previous section.

Optimal Reference Trajectory Definition

The development of both guidance laws involve defining a reference trajectory; as a target state for the global guidance law and as a reference to track for the local guidance law. The chosen reference trajectory is determined and found numerically by solving the minimum-fuel optimal landing problem via pseudo-spectral methods. The minimum-fuel optimal guidance problem can be formulated as follows[9]:

Minimum-Fuel Problem: Find the thrust program that minimizes the following cost function (negative of the lander's final mass):

$$\max_{t_f,\boldsymbol{a_c}} m_L(t_f) = \min_{t_f,\boldsymbol{a_c}} \int_0^{t_f} ||\boldsymbol{a_c}||dt \tag{20}$$

Subject to the following constraints (equations of motion):

$$\ddot{\boldsymbol{r}}_L = -\boldsymbol{g} + \frac{\mathbf{T}}{m_L} \tag{21}$$

$$\frac{d}{dt}m_L = -\frac{||\mathbf{T}||}{I_{sp}g_0} \tag{22}$$

And the following boundary conditions and additional constraints:

$$0 < T_{min} < \|\boldsymbol{T}\| < T_{max} \tag{23}$$

$$\boldsymbol{r}_L(0) = \boldsymbol{r}_{L0}, \quad \boldsymbol{v}_L(0) = \dot{\boldsymbol{r}}_L(0) = \boldsymbol{v}_{L0} \tag{24}$$

$$\boldsymbol{r}_L(t_F) = \boldsymbol{r}_{LF}, \quad \boldsymbol{v}_L(t_F) = \dot{\boldsymbol{r}}_L(t_F) = \boldsymbol{v}_{LF} \tag{25}$$

$$m_L(0) = m_{Lwet} \tag{26}$$

This is equivalent to minimizing the amount of propellant used during the descent. Here, the thrust is limited to operate between a minimum value (T_{min}) and a maximum value (T_{max}). The problem formulated in Eq. (20)-(26) does not have an analytical solution and must be solved numerically. To obtain the open-loop, fuel-optimal thrust program, the General Pseudospectral Optimal Control Software (GPOPS) has been employed.[10] GPOPS is an open-source optimal control software that implements Gauss and Radau hp-adaptive pseudospectral methods. After formulating the landing problem as described above, the software allows the direct transcription of the continuous-time, fuel-optimal control problem to a finite-dimensional Nonlinear Programming Problem (NLP). In GPOPS, the resulting NLP is solved using the SNOPT solver.[11] The pseudospectral approach is very powerful as it allows one to approximate both state and control using a basis of lagrange polynomials. Moreover, the dynamics is collocated at the Legendre-Gauss-Radau points. The use of global polynomials coupled with Gauss quadrature collocation points is known to provide accurate approximations that converge exponentially to continuous problems with smooth solutions. This open-loop optimal trajectory is used for the definition of the reference values seen in the formulation of the Hybrid Guidance Law (Eq. 10 and 12).

Global Guidance Law: Optimal Sliding Guidance (OSG)

The development of the Optimal Sliding Guidance (OSG) Law seen here is explained more thoroughly in Furfaro, et.al.[7]

The OSG algorithm is designed by combining some known results from optimal control theory as applied to the landing problem with relatively recent advancements in non-linear sliding control theory. Proper development of the sliding-based guidance algorithm requires the definition of an appropriate guidance model, which is seen in a 3-DOF framework in Eq. (4-9). These equations can be integrated from knowledge of the current position and velocity at time τ to determine the position and velocity at a specified final time, t_f:

$$\boldsymbol{r}_L(t_f) = \boldsymbol{r}_L(\tau) + \boldsymbol{v}_L(\tau)t_{go} + \int_{\tau}^{t_f} (t_f - \mathrm{t})(\boldsymbol{g}_L + \boldsymbol{a}_c(\mathrm{t}))d\mathrm{t} \tag{27}$$

$$\boldsymbol{v}_L(t_f) = \boldsymbol{v}_L(\tau) + \int_{\tau}^{t_f} (\boldsymbol{g}_L + \boldsymbol{a}_c(\mathrm{t}))d\mathrm{t} \tag{28}$$

Here, $t_{go} = t_f - \tau$ is the time-to-go. Next, we define the following quantities:

Definition #1. Given the time τ, we define the Zero-Effort Miss (ZEM) as the distance (vector) the lander will miss the target point if no acceleration command (guidance) is generated after τ:

$$\boldsymbol{ZEM}(\tau) = \boldsymbol{r}_{Lf} - \boldsymbol{r}_L(t_f) \qquad , \boldsymbol{a}_c(\mathrm{t}) = \boldsymbol{0}, \mathrm{t} \in [\tau, t_f] \tag{29}$$

Definition #2. Given the time τ, we define the Zero-Effort Velocity (ZEV) as the error in velocity at the final time, if no acceleration command (guidance) is generated after τ:

$$\boldsymbol{ZEV}(\tau) = \boldsymbol{v}_{Lf} - \boldsymbol{v}_L\left(t_f\right) \qquad , \boldsymbol{a}_c(\mathrm{t}) = \boldsymbol{0}, \mathrm{t} \in \left[\tau, t_f\right] \tag{30}$$

Here, $\boldsymbol{r}_{L_f}$ and $\boldsymbol{v}_{L_f}$ are fixed parameters that define the desired target state. The basis of the algorithm development is the ability to generate an optimal guidance law as a function of ZEM and ZEV. One of the key pieces is the ability to obtain a closed loop guidance law that minimizes the overall guidance effort, i.e. a guidance law that minimizes the overall acceleration command. The optimal problem can be formulated as follows:

Given the current position and velocity, $\mathbf{r}_\mathrm{L}$ *and* $\mathbf{v}_\mathrm{L}$, *as initial conditions, and the final desired conditions,* $\mathbf{r}_\mathrm{Lf}$ *and* $\mathbf{v}_\mathrm{Lf}$, *find the* $\mathbf{a}_\mathrm{c}(\tau)$ *as a function of* $\mathbf{ZEM}(\tau)$ *and* $\mathbf{ZEV}(\tau)$ *that minimizes the following performance index:*

$$J(\boldsymbol{a}_c) = \int_{\tau}^{t_f} \boldsymbol{a}_c(\mathrm{t})^T \boldsymbol{a}_c(\mathrm{t}) d\mathrm{t} \tag{31}$$

Subject to the equations of motion as physical constraints.

The acceleration command is assumed to be unconstrained, i.e. the thrust generated by the propulsion system is unbounded. It is found that the acceleration command is linear in time[7], i.e.:

$$\boldsymbol{a}_c(\tau) = \boldsymbol{A}_1\tau - \boldsymbol{A}_2 \tag{32}$$

Finally, the optimal acceleration command can be expressed as a function of $\boldsymbol{ZEM}(\tau)$, $\boldsymbol{ZEV}(\tau)$, and t_{go} as follows:

$$\boldsymbol{a}_c(\tau) = \frac{k_R}{t_{go}^2}\boldsymbol{ZEM}(\tau) + \frac{k_V}{t_{go}}\boldsymbol{ZEV}(\tau) \tag{33}$$

Here $\mathrm{k_R}$ = 6, and $\mathrm{k_V}$ = -2 are the optimal guidance gains[7]. This guidance law can also be written in terms of the error state, $\boldsymbol{\varepsilon}_r$ and $\boldsymbol{\varepsilon}_v$, as presented in Eq. (10) by using the definitions of $\boldsymbol{ZEM}$ and $\boldsymbol{ZEV}$.[11]

$$\boldsymbol{a}_c(\tau) = \frac{k_1}{t_{go}^2}\ \boldsymbol{\varepsilon}_r + \frac{k_2}{t_{go}}\boldsymbol{\varepsilon}_v - \boldsymbol{g} \tag{34}$$

The mathematical expression of the acceleration command is fairly simple and may be attractive for direct implementation on the on-board guidance computer. However, the optimal guidance, as derived, does not account for unmodeled disturbances which may negatively affect performance. In order to make the optimal control law robust against perturbations, we choose to integrate it with a non-linear sliding control mode to produce a robust guidance algorithm.

In order to implement the sliding control approach into the optimal guidance framework and derive the Optimal Sliding Guidance (OSG) equations, we begin by defining a sliding surface as a function of $\boldsymbol{\varepsilon}_r$ and $\boldsymbol{\varepsilon}_v$ as follows:

$$\boldsymbol{s} = \boldsymbol{\varepsilon}_v + \tilde{\lambda}\boldsymbol{\varepsilon}_r = \frac{\boldsymbol{\varepsilon}_v}{\tilde{\lambda}} + \boldsymbol{\varepsilon}_r = \boldsymbol{0} \tag{35}$$

Clearly, the surface goes to the null value as $\boldsymbol{\varepsilon}_r$ and $\boldsymbol{\varepsilon}_v$ both approach zero. Subsequently, the idea is to construct the guidance law in such a way that the system is always driven to the sliding surface. Therefore, we consider the dynamics of the sliding surface, i.e. take the derivative of Eq. (35) and substitute the definitions of $\boldsymbol{\varepsilon}_r$ and $\boldsymbol{\varepsilon}_v$:

$$\frac{d}{dt}\boldsymbol{s} = \frac{d}{d\tau}\boldsymbol{s} * \frac{d}{dt}\tau = \dot{\boldsymbol{\varepsilon}}_v + \tilde{\lambda}\dot{\boldsymbol{\varepsilon}}_r = \boldsymbol{g} + \boldsymbol{a}_c + \tilde{\lambda}\boldsymbol{\varepsilon}_v \tag{36}$$

If the optimal $\boldsymbol{a}_c$, as shown in Eq. (34) is substituted into Eq. (36), we obtain:

$$\frac{d}{dt}\boldsymbol{s} = \frac{k_2 + \tilde{\lambda}t_{go}}{t_{go}}\boldsymbol{\varepsilon}_v + \frac{k_1}{t_{go}^2}\boldsymbol{\varepsilon}_r = -K(\tau)\left(\boldsymbol{\varepsilon}_v + \tilde{\lambda}\boldsymbol{\varepsilon}_r\right) = -K(\tau)\boldsymbol{s} \tag{37}$$

The following relationships between the parameters can be easily found:

$$K(\tau) = -\frac{k_2 + \tilde{\lambda}t_{go}}{t_{go}} \tag{38}$$

$$\tilde{\lambda}K(\tau) = -\frac{k_1}{t_{go}^2} \tag{39}$$

$$t_{go}^2\tilde{\lambda}^2 + k_2 t_{go}\tilde{\lambda} - k_1 = 0 \tag{40}$$

This provides for two possible values of $\tilde{\lambda}$. The sliding mode is incorporated into the optimal guidance law to guarantee that the sliding surface behaves as follows:

$$\frac{d}{dt}s = -K(\tau)s - \Phi sign(s) \tag{41}$$

Here, $\Phi = const > 0$. By incorporating the sliding mode, the OSG equations are subsequently determined:

$$\boldsymbol{a}_c(\tau) = \frac{k_1}{t_{go}^2}\boldsymbol{\varepsilon}_r + \frac{k_2}{t_{go}}\boldsymbol{\varepsilon}_v - \boldsymbol{g} - \frac{\Phi}{t_{go}}sign(\boldsymbol{s}) \tag{42}$$

This guidance law can now be shown to be globally stable through the use of Lyapunov's second method using the specific case of $\boldsymbol{v}_{Lf} = \boldsymbol{0}$**.** Consider the following quadratic function as a Lyapunov candidate:

$$V = \frac{1}{2}s^T s = \frac{1}{2}\left(\boldsymbol{\varepsilon}_v + \tilde{\lambda}\boldsymbol{\varepsilon}_r\right)^T\left(\boldsymbol{\varepsilon}_v + \tilde{\lambda}\boldsymbol{\varepsilon}_r\right) \tag{43}$$

Differentiating with respect to time, we obtain:

$$\frac{d}{dt}V = \frac{d}{d\tau}V * \frac{d}{dt}\tau = \boldsymbol{s}^T\frac{d}{d\tau}\boldsymbol{s} = \boldsymbol{s}^T\left(\dot{\boldsymbol{\varepsilon}}_v + \tilde{\lambda}\dot{\boldsymbol{\varepsilon}}_r\right) \tag{44}$$

Inserting the expressions for the derivative of $\boldsymbol{\varepsilon}_r$ and $\boldsymbol{\varepsilon}_v$:

$$\begin{aligned} \frac{d}{d\tau}V = \boldsymbol{s}^T\frac{d}{d\tau}\boldsymbol{s} = \boldsymbol{s}^T\left(\frac{k_2+\tilde{\lambda}t_{go}}{t_{go}}\varepsilon_v + \frac{k_1}{t_{go}^2}\varepsilon_r - \frac{\Phi}{t_{go}}sign(\boldsymbol{s}) + \boldsymbol{p}(\tau)\right) = \\ = \boldsymbol{s}^T\left(\left(\frac{k_2 t_{go}\tilde{\lambda} + t_{go}^2\tilde{\lambda}^2}{\tilde{\lambda}t_{go}^2}\boldsymbol{\varepsilon}_v + \frac{k_1}{t_{go}^2}\boldsymbol{\varepsilon}_r\right) - \frac{\Phi}{t_{go}}sign(\boldsymbol{s}) + \boldsymbol{p}(\tau)\right) = \end{aligned} \tag{45}$$

$$= s^T \left(\left(\frac{k_1}{\lambda t_{go}^2} \varepsilon_v + \frac{k_1}{t_{go}^2} \varepsilon_r \right) - \frac{\Phi}{t_{go}} sign(s) + \boldsymbol{p}(\tau) \right) =$$

$$= \frac{k_1}{t_{go}^2} \boldsymbol{s}^T \boldsymbol{s} + \left(- \frac{\Phi}{t_{go}} sign(\boldsymbol{s}) + \boldsymbol{p}(\tau) \right)$$

Here, $\boldsymbol{p}(\tau)$ represents a vector of unmodeled dynamics and perturbations. These are included in the development of the guidance law to prove stability against perturbations. Now, substituting $k_1 = -6$ and assuming that $\Phi > \|\mathbf{p}\|$ we get:

$$\frac{d}{d\tau} V = -\frac{6}{t_{go}^2} \|\boldsymbol{s}\|^2 - \boldsymbol{s}^T \left(\frac{\Phi}{t_{go}} sign(\boldsymbol{s}) + \boldsymbol{p}(\tau) \right) \leq 0 \tag{46}$$

This ensures global stability for the OSG for all $t_{go} > \delta$, as defined in the flow set, C (Eq. (14-16)).

Local Guidance Law: Linear Quadratic Regulator (LQR)

In order to develop a LQR based guidance law, the system must first be linearized.[12] This is done by taking a Taylor expansion of the dynamics about the reference trajectory:

$$\Delta \dot{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{y}_n, \boldsymbol{u}_n) + \left. \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}} \right|_{\mathbf{y}_\mathrm{n}, \mathbf{u}_\mathrm{n}} \Delta \boldsymbol{y} + \left. \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}} \right|_{\mathbf{y}_\mathrm{n}, \mathbf{u}_\mathrm{n}} \Delta \boldsymbol{u} + \mathcal{O}(\|\Delta \boldsymbol{y}\|^2, \|\Delta \boldsymbol{u}\|^2) \tag{47}$$

where $\boldsymbol{y}, \boldsymbol{u}$ are the current state vector and input (i.e. acceleration vector), $\boldsymbol{y}_n, \boldsymbol{u}_n$ are the state and acceleration vector input on the reference trajectory, and the last term represents higher order terms in the expansion, which are ignored here. Eq. (47) can be re-written as

$$\Delta \dot{\boldsymbol{y}} = A \Delta \boldsymbol{y} + B \Delta \boldsymbol{u} \tag{48}$$

where A and B are defined as the derivatives of the equations of motion with respect to $\boldsymbol{y}$ and $\boldsymbol{u}$ evaluated on the reference trajectory, respectively, as seen in Eq. (47). Notice that Eq. (48) takes the form of a linear equation, so the work in order to prove the stability of the system is done in the usual manner. If $\Delta \boldsymbol{u}$ is defined as a linear feedback controller, the system takes the form:

$$\Delta \boldsymbol{u}(\boldsymbol{y}) = \mathbf{a_c} = -k(\boldsymbol{y} - \boldsymbol{y}_n) \rightarrow \Delta \dot{\boldsymbol{y}} = (A - Bk) \Delta \boldsymbol{y} = A_c \Delta \boldsymbol{y} \tag{49}$$

where k is the gain of the feedback controller, and can be found such that the system is locally stable, i.e. all eigenvalues of the matrix A_c have negative real parts.

Next we introduce the LQR approach, i.e. define a quadratic optimal control problem, which is defined as:

Find $\Delta \boldsymbol{u} = -k \Delta \boldsymbol{y}$ (i.e. find k) that minimizes the following performance index:

$$J = \int_0^\infty (\Delta \boldsymbol{y}^T Q \Delta \boldsymbol{y} + \Delta \boldsymbol{u}^T R \Delta \boldsymbol{u}) d\tau = \int_0^\infty \Delta \boldsymbol{y}^T (Q + k^T R k) \Delta \boldsymbol{y} \, d\tau \tag{50}$$

Subject to Eq. (48) as a physical constraint.

In Eq. (50), Q and R are defined as positive definite matrices that determine the relative importance of accuracy (landing error) and effort (commanded acceleration, i.e. propellant mass used). If the following, from Eq. (50), is set to be true:

$$\boldsymbol{y}^T(Q + k^T Rk)\boldsymbol{y} = -\frac{d}{d\tau}(\boldsymbol{y}^T P\boldsymbol{y}) \tag{51}$$

then the following is also true:

$$-(Q + k^T Rk) = (A - Bk)^T P + P(A - Bk) \tag{52}$$

Eq. (52) essentially states that for a given k such that $A - Bk$ is stable, then there exists a matrix P such that the condition in Eq. (52) is true. In other words, P satisfies what is known as the Reduced Riccati Equation, i.e.:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{53}$$

If Eq. (53) is held true, the optimal problem shown in Eq. (50) can be solved analytically, with the solution for k being such that

$$k = R^{-1}B^T P \tag{54}$$

Thus, the controller gain k is found such that it not only sovles the optimal problem, but also insures local asymptotic stability for the linearized system. Details on the proof of stability of a LQR controller can be found in many linear control system textbooks and is not included here.[12]

RESULTS

Hybrid Guidance Law for Lunar Descent and Landing

Generally, any properly designed guidance algorithm is expected to perform well under ideal conditions. However, a test campaign must be planned to verify that the proposed guidance algorithm works under realistic conditions. The guidance routines are therefore tested using a more realistic model to verify their performance for real-time implementation. A 3-DOF model that simulates the translational dynamics of the landing vehicle as shown in Eq. (1-3) has been implemented in a MATLAB® environment for Monte Carlo analysis. The model includes: 1) a more realistic model of the moon spherical gravitational field that account for the moon's non-flat surface; 2) a linearly time-varying mass model with a nominal mass flow-rate subjected to perturbations; and 3) a random perturbation acceleration that accounts for unmodeled dynamics.

Table 1. Monte Carlo Simulation Perturbation Values

Initial Condition	Mean Value	Standard Deviation
X-Axis Position	$0\ m$	$100\ m$
Y-Axis Position	$-9000\ m$	$100\ m$
Z-Axis Position	$3000\ m$	$100\ m$
Velocity Magnitude	$180\ m/sec$	$60\ m/sec$
Flight Path Angle	$1.77\ rad$	$20\ mrad$
Crossing Angle	$1.57\ rad$	$20\ mrad$

Mass	$1900\ kg$	N/A
Mass Flow Rate	0%	20%
Thrust Angle Bias	$0\ rad$	$30\ mrad$
Roll Angle Bias	$0\ rad$	$50\ mrad$
Disturbing Acceleration	$0\ m/sec^2$	$Uniform\ Distribution$

The mass of the lander is nominally assumed to be $1900\ kg$ with a specific impulse of $292\ sec$. The lander is assumed to be capable of a maximum allowable thrust of 30 kN. The initial conditions for the optimal reference trajectory are set to be $r_L(0) = [0, -7500, 2500]^T$. These conditions are very close to the ideal entry point of the trajectory designed for the Apollo approach phase. The nominal lander entry point, which is the simulation initial condition, is set to be $\boldsymbol{r}_L(0) = [150, -9000, 3000]^T$. This allows the guidance law to begin with the global ZEM/ZEV law and then switch to the local law for purpose of demonstration. Here, it is assumed that an initial de-orbiting maneuver (assuming the lander is initially parked in a lunar orbit) is followed by a braking phase with an ad-hoc guidance routine that targets the ideal nominal entry point that sets the stage for the terminal phase that guides the lander to the desired position on the lunar surface. Clearly, because of guidance errors during the de-orbiting and braking maneuvers, the initial conditions for the terminal guidance are not the nominal. A set of Monte Carlo simulations is conducted assuming a dispersion of the initial conditions as reported in Table 1. All dispersions in the initial position and velocity have been drawn from Gaussian distributions with prescribed mean and standard deviation. Moreover, as reported in the same table, perturbations were introduced in both thrust magnitude and direction simulating effects of fluctuating mass flow rate and misalignment in the thrust direction. A random disturbing acceleration (uniform distribution with maximum of 20% of the overall acceleration vector) has been introduced in the lander dynamics to further verify the robustness of the proposed algorithm.

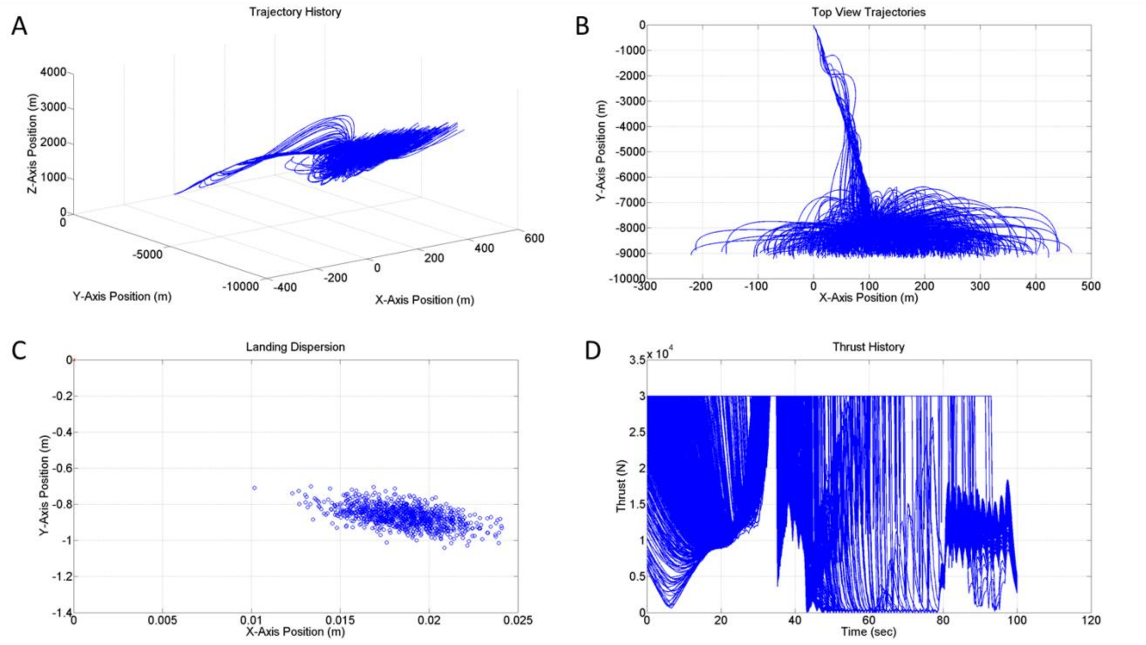

Figure 4. Monte Carlo histories for the Hybrid Guidance algorithm simulations. a) 3-D trajectories of the descending lander. b) Top-down view of the trajectory histories. c) Final Lander Position Error Dispersion. d) Thrust command histories.

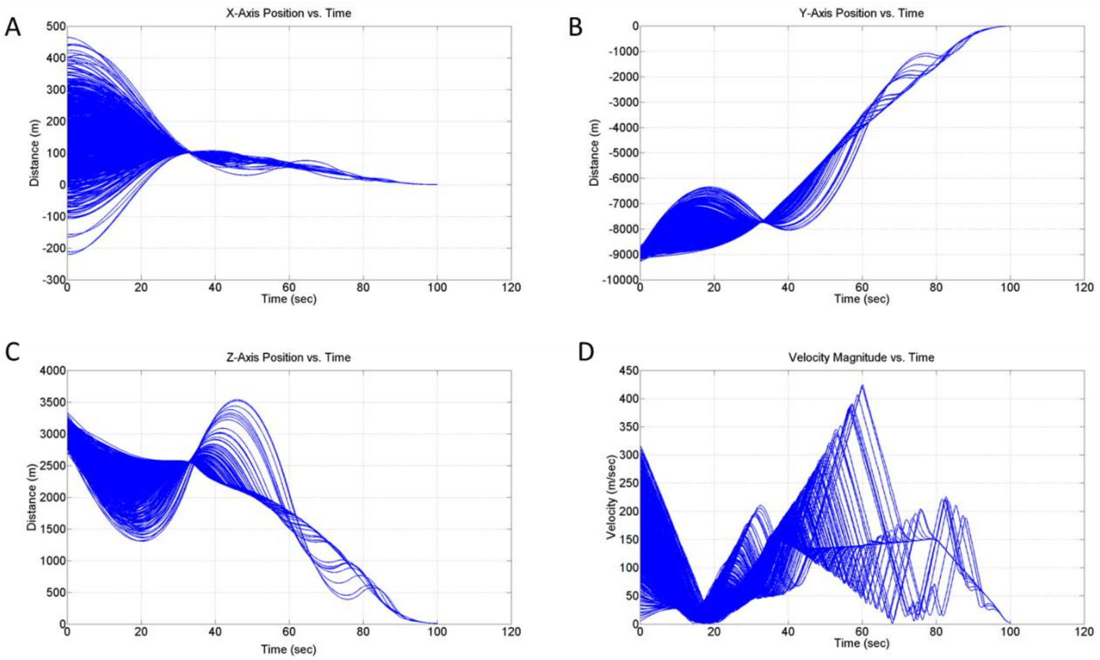

Figure 5. Monte Carlo state history results for the Hybrid Guidance Monte Carlo simulations. a) X-Axis Position History b)Y-Axis Position History c)Z-Axis Position History d)Velocity Magnitude History

A Monte Carlo analysis has been conducted by running 1000 simulations of the guidance algorithm in the 3-DOF simulation framework. The Hybrid Guidance algorithm requires a reference trajectory; it targets a point on the trajectory such that the lander will reach the reference trajectory at a preplanned time-to-go. This allows a trajectory planner to plan the final approach to the landing site and include any desired specific constraints on the trajectory of the descent. In the presented simulations, the algorithm was asked to target a reference trajectory that ended in a point that is located at an altitude of 10 m above the desired landing point located at the origin of the reference frame. The algorithm also targets a final velocity of zero in all axes (soft landing). The Hybrid Guidance algorithm does not target a point directly on the surface to account for additional final maneuvers that may be required to a) divert for surface hazards avoidance (e.g. big rocks or uneven surface on desired landing point) and b) adjust the lander attitude for vertical descent. Figure 4 and Figure 5 show the state history of the trajectory for the 1000 Monte Carlo simulations of the Hybrid Guidance algorithm. The guidance parameters employed in the simulations are reported in Table 2 and the terminal state statistics are reported in Table 3.

Table 2. Hybrid Guidance Parameters

Guidance Parameter	Value
Position gain, $\boldsymbol{k_1}$	6
Velocity gain, $\boldsymbol{k_2}$	-2
Sliding parameter, $\boldsymbol{\Phi}$	0.4
$\boldsymbol{Q}$	$5I_{6x6}$

$\boldsymbol{R}$	$5I_{3x3}$
$\boldsymbol{\Lambda_{r_1}}$	$10\ m$
$\boldsymbol{\Lambda_{v_1}}$	$1\ m/sec$
$\boldsymbol{\Lambda_{r_2}}$	$15\ m$
$\boldsymbol{\Lambda_{v_2}}$	$1.5\ m/s$

Table 3. Monte Carlo Final State Error Statistics

	Nominal	**Mean**	**Standard Deviation**
X-Axis Position (m)	0	0.0185	0.0022
Y-Axis Position (m)	0	-0.8634	0.0608
Z-Axis Position (m)	10	10	0
Velocity Magnitude (m/sec)	0	1.7686	0.0418

Generally, the algorithm performs very well. Figure 4c shows the landing dispersion that highlights the precision capabilities of the Hybrid Guidance algorithm. On the average, the desired target point was achieved with accuracy within a few centimeters. Notably, there is a bias found in the y-axis component of the terminal position. This is most likely due to the interpolation of the optimal reference trajectory which is causing the final position to be slightly offset from the origin. Regardless, the algorithm performs well from a precision point of view.

Importantly, all the final landing points resulting from the 1000 simulated guided trajectories fall within a dispersion characterized by mean position error of 0.864 m with a standard deviation of .061 m, assuming a normal distribution. The statistics of the terminal velocity magnitude of the lander are seen in Table 3. The acceleration command generated by the Hybrid Guidance algorithm is generally higher at the beginning of the landing descent, reaching its maximum close to when the guidance law switch occurs. However, as can be seen in Fig. 4d, the thrust is very near the maximum allowable level for many of the simulations. On average, the thrust level decreases dramatically once the lander has reached the reference trajectory and is using the local LQR-based guidance. This is due to the fact that the only effort necessary at this point is minor corrections to track the trajectory locally. The acceleration peaks as it approaches the point on the reference trajectory, most likely due to the LQR controller quickly accounting for the error in state of the lander directly after the guidance switch occurs. This large peak in control activity then quickly brings the lander onto the reference trajectory, at which point the commanded acceleration has a much lower magnitude. In general, even with a maximum thrust limit applied, the guidance law still performs well demonstrating low residual errors in position and velocity.

Hybrid Guidance Law for Retargeting

Modern landing algorithms should have the flexibility and capability to react in real-time to mission critical decision that may enforce an alternative target for landing on the surface of a planetary body. Indeed, during the descent, one may decide that the current targeted site is no longer desirable due to safety or scientific reasons. After selecting a new site, the guidance law must be able to adapt and safely bring the lander to the new location. In a more conventional guidance algorithm, a new trajectory must be developed that will take the lander from its current state to the new target. Indeed, Chomel and Bishop [6] showed that their proposed algorithm is capable of effectively retargeting the landing site while en route to the lunar surface. Once the new landing point was selected, their algorithm computed a new trajectory assuming that the downrange was the shortest distance between the vehicle's current position and the desired final target site. Once the states were properly defined, the guidance algorithm autonomously converged to the new trajectory.

The proposed Hybrid Guidance Law has the inherent ability to guide the lander to a new landing site if a decision to change from the original location is made. Importantly, the same optimal trajectory can be used, as it can simply be translated from the original landing point to the new location. Further, when the landing location is updated, the Hybrid Guidance inherently switches to the global guidance law in order to bring the lander to the neighborhood of the new translated reference trajectory, following the logic presented in the definition of the flow and jump sets, Eq. (14-19). Clearly, retargeting can be easily implemented by simply shifting the target position (the final velocity is assumed to be zero as before), and subsequently the reference trajectory, and letting the algorithm generate the acceleration command required to drive the lander to the new location. An example of such a retargeting scenario is shown in Fig. 6 and 7.

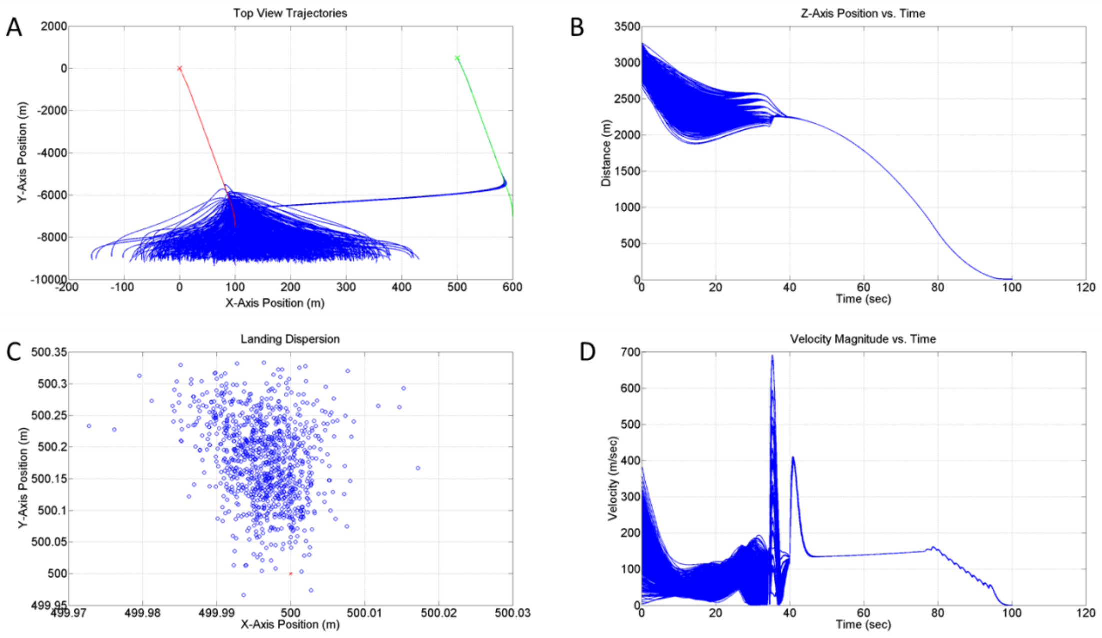

Figure 6. Monte Carlo state history results for the Hybrid Guidance Monte Carlo simulations. a) Top View Trajectory History with Original Reference Trajectory (red) and New Reference Trajectory (green) b)Z-Axis Position History c)Final Landing Position Dispersion d)Velocity Magnitude History

In this case, the desired landing point is moved 500 m in the X-axis direction and 500 m in the Y-axis position. The guidance system is initially asked to drive the lander toward the original site for the first 40 seconds of the descent. At this time, the lander has generally converged to the original reference trajectory and is using the local guidance law. This is also when the new target location is specified and the guidance is required to target the new location. Importantly, the full time of the simulation was not increased for this set of simulations, as the algorithm was able to quickly converge to the translated reference trajectory after the target landing point was changed. Additionally, the maximum allowable thrust limit that was applied to the previous results has been removed for purpose of demonstration of the guidance law to adapt and target the new location successfully.

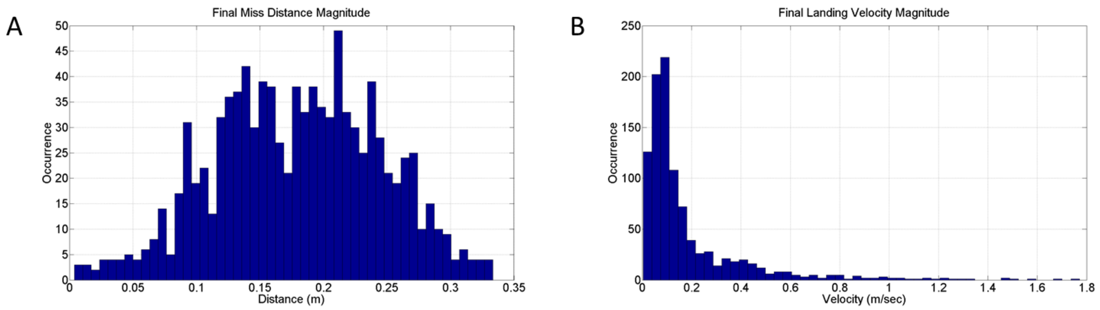

Figure 7. Monte Carlo Result Statistics. A) Final Miss Distance Magnitude Statistics; B) Residual Velocity Error Magnitude Statistics

A set of 1000 Monte Carlo simulations have been implemented to verify the ability of the proposed guidance law to actively retarget a new landing site. Figure 6 shows the performance of algorithm when retargeting, while the terminal state statistics are shown in Fig. 7. These statistics are also reported in Table 4. Note that Fig. 6a includes the two reference trajectories, both for the original site in red and the new site in green. As can be seen, the algorithm accurately tracks both of these trajectories when necessary, bringing the lander very close to the new desired target location. As can be seen by these results the algorithm is quite adept at retargeting to a new landing site while descending to the lunar surface.

Table 4. Retargeting Monte Carlo Final State Error Statistics

	Nominal	Mean	Mean Error	Standard Deviation
X-Axis Position (m)	500	499.9964	-0.0036	0.0045
Y-Axis Position (m)	500	500.1791	0.1791	0.0652
Z-Axis Position (m)	10	10	0	0
Velocity Magnitude (m/sec)	0	0.1870	0.1870	0.2348

Generally, each case of the simulation successfully converged to the original reference trajectory for a short amount of time before switching back to using the global ZEM/ZEV guidance law to target the new reference trajectory. At this point, the trajectories successfully converge to the new trajectory and track it to the desired target point. Under the condition of retargeting, the algorithm is shown to perform very well. The results in Fig. 6 and 7 show that the residual error in both position and velocity are near zero for all cases. The final velocity has a maximum magnitude of 1.77 m/sec. While some cases do feature significant residual velocity, the values are still low, with most cases being much lower than the maximum, as can be seen by the mean error in Table 4. Despite these errors, the algorithm is still shown to be not only capable, but accurate at effectively retargeting the lander to a new location on the lunar surface.

CONCLUSIONS

The guidance algorithm responsible for driving the Apollo lander in its journey toward the Moon has shown to be effective in accomplishing its goal, i.e. take the three astronauts on-board safely to the lunar surface. Nevertheless, a new class of guidance algorithms must be developed to satisfy more stringent requirements imposed by a new desire to explore the Moon with an unprecedented degree of flexibility. Such algorithms should have both a) the ability to land the spacecraft with more stringent precision and b) increased flexibility to meet new mission requirements. In this paper, a hybrid guidance algorithm was presented that may be an excellent option to satisfy both of these requirements. The local controller is an LQR controller algorithm that generally comprises of two major elements, i.e. targeting algorithm (optimal open-loop trajectory generation) and real-time guidance (trajectory tracking). The global controller breaks that paradigm, using a formalism borrowed from recent advancements in non-linear, higher-order sliding mode control theory which generates an acceleration command that requires only knowledge of the current lander state and the desired (final) state on the reference trajectory. The algorithm is tested by running multiple sets of Monte Carlo simulations, which show that the hybrid guidance law is quite effective in driving the lander to the desired position with very minimal residual guidance error, and that they are robust against large perturbations. Importantly, the Hybrid Guidance Law is shown to work well with a guidance loop running at 10 Hz. Further analysis was performed to examine the capability of the algorithm to actively retarget a different landing site during the descent. An additional set of Monte Carlo simulations show that the algorithm is quite capable of successfully targeting a new site if the original location is deemed unacceptable for landing.

While the application and simulation scenarios presented provide a representation of the capability of the application of hybrid control schemes to the spacecraft landing problem, it is by no means limited to the example provided. The presented hybrid system provides a large amount of flexibility, and as such, there is still quite a large amount of research and exploration that can be done into the true potential of using such a framework for spacecraft landing guidance. Future efforts will involve the incorporation of other guidance schemes and landing scenarios into the proposed hybrid framework, such as asteroid proximity operations or terminal powered landing guidance on Mars. In addition, further analysis is necessary to test the limits of the flexibility of the hybrid framework, such as the inclusion of multiple guidance schemes on-board that are used in regions where they are the most optimal from a fuel-usage standpoint.

REFERENCES

[1] Klumpp, A.R., “Apollo Guidance, Navigation, and Control: Apollo Lunar-Descent Guidance”, Massachusetts Inst. Of Technology, Charles Stark Draper Lab, TR R-695, Cambridge, MA, June 1971.

[2] Klumpp, A.R., “Apollo Lunar Descent Guidance”, *Automatica*, Vol. 10, Issue 2, 133-146, 1974.

[3] Yanushevsky, R., and Boord, W., “New Approach to Guidance Law Design,” *Journal of Guidance, Control and Dynamics*, Vol. 28, No. 1, 2005, pp. 162-166.

[4] Goebel, Rafal, Sanfelice, Ricardo G., and Teel, Andrew R., “Hybrid Dynamical Systems,” *IEEE Control Systems Magazine*, April 2009, pp. 28-93.

[5] Goebel, R., Sanfelice, R.G., and Teel, A.R., “Hybrid Dynamics Systems: Modeling, Stability and Robustness”, Princeton University Press, 2012

[6] Chomel, C., T., and Bishop, R., H., “Analytical Lunar Descent Algorithm”, *Journal of Guidance, Control, and Dynamics*, 32, 3, 915-927, 2009.

[7] R. Furfaro, S. Selnick, M. L. Cupples, and M. W. Cribb, “Non-Linear Sliding Guidance Algorithms for Precision Lunar Landing,” *Proceedings of the 21st AAS/AIAA Space Flight Mechanics Meeting* (AAS 11-167), 2011.

[8] Sanfelice, Ricardo G., and Teel, Andrew R., “A “Throw-and-Catch” Hybrid Control Strategy for Robust Global Stabilization of Nonlinear Systems”, Proceedings of the 2007 American Control Conference, pp. 3470-3475, 2007.

[9]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, 2007, pp. 1353–1366.

[10]A. V., Rao, D. A., Benson, C., Darby, M. A., Patterson, C., Francolin, I., Sanders, et al. Algorithm 902: GPOPS, a MATLAB software for solving multiple phase optimal control problems using the Gauss pseudospectral method. ACM Transactions on Mathematical Software, 37(2), 2010, 22:1-22:39.

[11]P.E. Gill, M.A. Saunders, and W. Murray. SNOPT: An SQP algorithm for large scale constrained optimization. Technical Report NA 96-2, University of California, San Diego, 1996.

[12]Guo, Yanning G., Hawkin, Matt, and Wie, Bong, “Optimal Feedback Guidance Algorithms for Planetary Landing and Asteroid Intercept”, *Proceedings of the 2011 AAS/AIAA Astrodynamics Specialist Conference* (AAS 11-588), 2011.

[13]Bryson Jr. , Arthur E., *Control of Spacecraft and Aircraft*, Princeton University Press, Princeton, NJ, 1994, pp. 317-328.