Switching System Model for Pinpoint Lunar Landing Guidance Using a Hybrid Control Strategy

Daniel R. Wibben[1]
The University of Arizona, Tucson, AZ 85721

Roberto Furfaro[2]
The University of Arizona, Tucson, AZ 85721

and

Ricardo G. Sanfelice[3]
The University of Arizona, Tucson, AZ 85719

A novel non-linear spacecraft guidance scheme utilizing a hybrid controller for pinpoint lunar landing 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 the ability of a system with multiple controllers to provide robustness and performance that cannot be obtained with a single controller. 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 approach is used generally to provide flexibility; many stable controllers can be used for the global and local controllers in the hybrid framework, creating options that allow the algorithm to be tailored to meet mission requirements. The presented case 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, at which point the guidance law will switch to that of a 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. The simulations demonstrate that the HGL is very accurate and results in low residual guidance errors.

Nomenclature

A	=	linear system plant matrix
A_C	=	equivalent plant matrix
$\boldsymbol{a}_c$	=	commanded acceleration
$a_{c_x}, a_{c_y}, a_{c_z}$	=	components of commanded acceleration vector
B	=	linear system input matrix
C	=	flow set
C_1, C_2	=	flow set for global and local controllers
D	=	jump set
D_1, D_2	=	jump set for global and local controllers

[1] Graduate Student, Department of Systems and Industrial Engineering, 1127 E. James E. Rogers Way Tucson, AZ 85721, AIAA Student Member.
[2] Assistant Professor, Department of Systems and Industrial Engineering, 1127 E. James E. Rogers Way, Tucson, AZ 85721.
[3]Assistant Professor, Department of Aerospace and Mechanical Engineering, 1130 N. Mountain Ave., Tucson, AZ 85721.

$\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{a_c})$ = flow map
$\boldsymbol{G}(\boldsymbol{x})$ = jump map
$\boldsymbol{g}_L$ = lunar gravitational acceleration vector
g_L = lunar gravity constant
$\boldsymbol{\mathcal{H}}$ = hybrid system
K = sliding mode function
k = LQR guidance gain
k_r, k_v = ZEM and ZEV gains
k_1, k_2 = position and velocity error gains
P = LQR gain matrix satisfying the Reduced Ricatti equation
$\boldsymbol{p}$ = perturbations and unmodeled forces vector
Q = positive definite LQR gain matrix
q = switching variable
R = positive definite LQR gain matrix
$\boldsymbol{R}_{RG}, \boldsymbol{V}_{RG}, \boldsymbol{A}_{RG}$ = position, velocity, and acceleration vectors of reference trajectory
$\boldsymbol{R}_{TG}, \boldsymbol{V}_{TG}, \boldsymbol{A}_{TG}, \boldsymbol{J}_{TG}, \boldsymbol{S}_{TG}$ = position, velocity, acceleration, jerk, and snap vectors of target point
$\boldsymbol{r}_L$ = lander position vector in guidance reference frame
$\boldsymbol{r}_{L_f}$ = desired final position
r_x, r_y, r_z = components of lander position vector
$\boldsymbol{s}$ = sliding surface
t = time
t_f = final time
t_{go} = time-to-go
$\boldsymbol{u}$ = linear system control input
$\boldsymbol{u}_n$ = control thrust input on reference trajectory
V = Lyapunov candidate function
$\boldsymbol{v}_L$ = lander velocity vector in guidance reference frame
$\boldsymbol{v}_{L_f}$ = desired final velocity
v_x, v_y, v_z = components of lander velocity vector
$\boldsymbol{x}$ = hybrid state vector
$\boldsymbol{y}$ = state vector for LQR development
$\boldsymbol{y_n}$ = state vector on reference trajectory
$\boldsymbol{ZEM}$ = zero-effort miss
$\boldsymbol{ZEV}$ = zero-effort velocity
δ = time-to-go constraint
$\boldsymbol{\varepsilon}$ = composite error state (position and velocity)
$\boldsymbol{\varepsilon}_r$ = position error vector
$\boldsymbol{\varepsilon}_v$ = velocity error vector
$\Lambda_{r_1}, \Lambda_{r_2}$ = hybrid system position gains
$\Lambda_{v_1}, \Lambda_{v_2}$ = hybrid system velocity gains
$\tilde{\lambda}$ = sliding guidance law gain
Φ = sliding parameter matrix
τ = timer variable

I. 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, 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 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 time behavior into the system. The use of multiple controllers may allow for the use of controllers that work well only in certain regions, i.e. the combination of a controller that works 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 was treated solely as a continuous system and only a single chosen guidance law was used.

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 has it follow 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, while the focus of the paper is to detail the development of the full hybrid system and demonstrate how the two control laws are incorporated into one hybrid system. Also included is a set of results that demonstrate performance and investigate the robustness of the hybrid system under varying initial states through the use of a set of Monte Carlo simulations.

The paper is organized as follows. In section II, the guidance problem is formulated. The 3-DOF model used for both the development of the HGL and the simulations is presented. In section III, the notion of a hybrid system is presented and the hybrid framework for the lunar landing problem is developed. In section IV, the individual guidance laws that are used globally and locally are presented and defined for the reader. In section V, the HGL is implemented and its performance analyzed. A set of Monte Carlo simulations is conducted to verify the achieved landing accuracy when using the HGL in realistic scenarios. In section VI, the conclusions are reported.

II. 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.

A. 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.

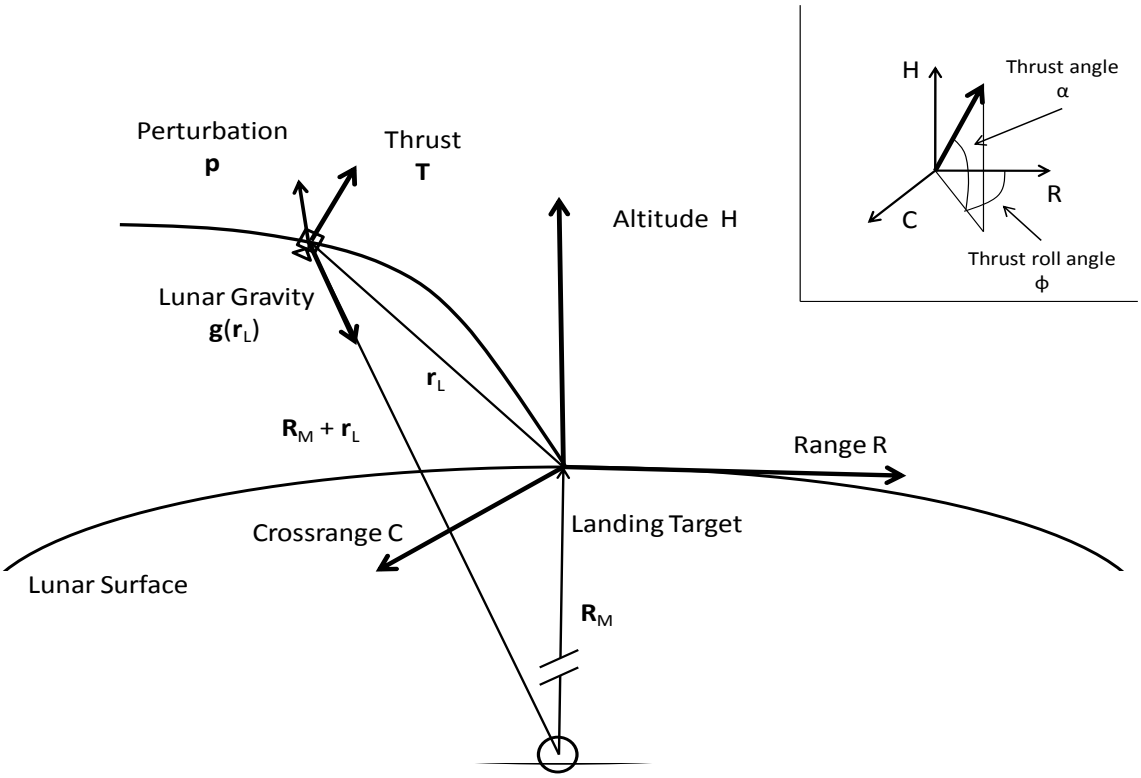

Figure 1. Guidance Reference Frame and Free-Body Force Diagram for the Lunar Lander during the Powered Descent to the Designated Target

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

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

$$\dot{\boldsymbol{v}}_L = \boldsymbol{g}_L + \boldsymbol{a}_{\mathrm{c}} \tag{2}$$

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, and $\boldsymbol{g}_L$ represents the constant gravitational acceleration vector of the moon. If $\boldsymbol{r}_L = [r_x, r_y, r_z]^T$ and $\boldsymbol{v}_L = [v_x, v_y, v_z]^T$, where the *x*, *y*, and *z* directions represent the crossrange, downrange, and altitude of the lander, respectively, the equations of motions can be written by components as follows:

$$\dot{r}_{\mathrm{x}} = v_x \tag{3}$$

$$\dot{r}_{\mathrm{y}} = v_y \tag{4}$$

$$\dot{r}_z = v_z \tag{5}$$

$$\dot{v}_x = a_{\mathrm{c}_x} \tag{6}$$

$$\dot{v}_y = \mathrm{a}_{c_y} \tag{7}$$

$$\dot{v}_z = -g_L + a_{\mathrm{c}_z} \tag{8}$$

Clearly, the considered mathematical model is a 3-DOF model with constant 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.

III. Hybrid Landing Guidance Control Law Development

Generally, the dynamics of the system are such that the guidance law can be formulated as a hybrid controller: the system will combine two different continuous time controllers to provide robustness and greater capabilities that are not possible with just one controller. The local controller will have the capability to steer trajectories to the desired final target point 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. Both of these guidance laws will be formally introduced in later sections.

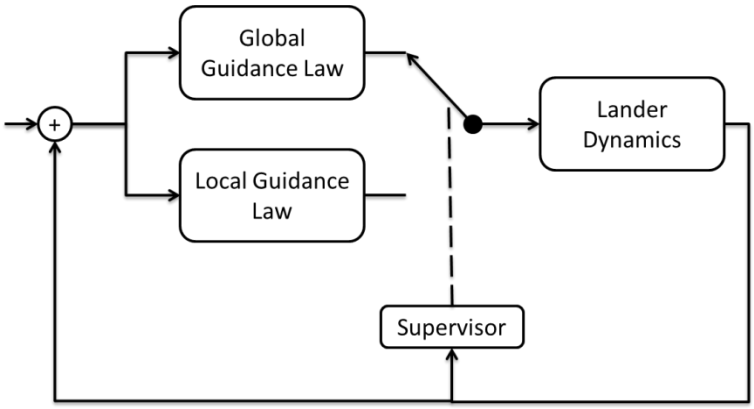

Figure 2. Closed-loop system combining local and global guidance laws

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, 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:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{r}_L - \boldsymbol{R}_{RG} \\ \boldsymbol{v}_L - \boldsymbol{V}_{RG} \\ q \\ \tau \end{bmatrix} = \begin{bmatrix} \boldsymbol{\varepsilon}_r \\ \boldsymbol{\varepsilon}_v \\ q \\ \tau \end{bmatrix} \tag{9}$$

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 Section IV.A). 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 $\boldsymbol{\mathcal{H}}$.

A. Formal Hybrid System Definition

For this problem, the hybrid system is defined as:

$$\boldsymbol{\mathcal{H}} = (C, \boldsymbol{F}, D, \boldsymbol{G}) \tag{10}$$

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. (9).

$$\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{a_c}) \coloneqq \begin{bmatrix} \mathbf{v}_\mathrm{L} - \boldsymbol{V}_{RG} \\ \boldsymbol{g}_L + \boldsymbol{a}_{c_q} - \boldsymbol{A}_{RG} \\ 0 \\ 2 - q \end{bmatrix} \tag{11}$$

where $\boldsymbol{A}_{RG}$ is the acceleration defined by the reference trajectory, which is formally defined in a later section 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, as the chosen controller is not updated 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 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 due to the local controller that has been chosen and prevents τ from being unbounded. By forming the system in this way, the analyses used for standard 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.

$$\boldsymbol{G}(\boldsymbol{x}) \coloneqq \begin{bmatrix} \boldsymbol{\varepsilon}_\mathrm{r} \\ \boldsymbol{\varepsilon}_v \\ 3 - q \\ 0 \end{bmatrix} \tag{12}$$

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. (11), 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 (see Section IV.B), a constraint must be included on the timer τ. This leads to the definition of the flow set for the global controller as:

$$C_1 = \{\boldsymbol{x}: \|\boldsymbol{\varepsilon}_r\| \geq \Lambda_{r_1}, \|\boldsymbol{\varepsilon}_v\| \geq \Lambda_{v_1}, q = 1, t_f - \tau \geq \delta\} \tag{13}$$

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:

$$C_2 = \{\boldsymbol{x}: \|\boldsymbol{\varepsilon}_r\| \leq \Lambda_{r_2}, \|\boldsymbol{\varepsilon}_v\| \leq \Lambda_{v_2}, q = 2, \tau = 0\} \tag{14}$$

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:

$$C = C_1 \cup C_2 \tag{15}$$

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:

$$D_1 = \{\boldsymbol{x}: \|\boldsymbol{\varepsilon}_r\| \leq \Lambda_{r_1}, \|\boldsymbol{\varepsilon}_v\| \leq \Lambda_{v_1}, q = 1, t_f - \tau < \delta\} \tag{16}$$

$$D_2 = \{\boldsymbol{x}: \|\boldsymbol{\varepsilon}_r\| \geq \Lambda_{r_2}, \|\boldsymbol{\varepsilon}_v\| \geq \Lambda_{v_2}, q = 2, \tau = 0\} \tag{17}$$

$$D = D_1 \cup D_2 \tag{18}$$

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.

The hybrid system that is defined here can be applied to the general lunar landing problem, with some small revision. Due to the form of the chosen global guidance law, the constraint on τ must be provided, but this may not be needed in the general case. As the system is generally not specific to any set of individual guidance laws, it provides a hybrid framework that can be used for the general case of the lunar landing problem. This provides a great amount of flexibility in allowing the designer to choose a combination of global and local controllers to meet specific mission requirements or provide more optimal solutions in certain regimes. For example, the system can be chosen such that the local controller will have little control activity so as to not contaminate the landing site, while the global controller will be more active to compensate and still bring the lander to the desired target point.

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.

IV. Global and Local Guidance Laws Development

The goal of this paper is to design an innovative guidance algorithm for precision lunar landing that explicitly uses a hybrid system approach. 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 nears 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 explained 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 section III.

A. 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 the one used by the Apollo lander and isdesigned with the desire to satisfy a two-point boundary value problem with a total of five degrees of freedom for each of the three components, such that the shape of the trajectory is constrained.[1,2] A quartic polynomial is the minimum order with which five constraints on the reference trajectory can be satisfied. With the reference trajectory evolving backwards in time from the target point, i.e. a Taylor expansion about the target point, it can be defined as:

$$\boldsymbol{R}_{RG}(\tau) = \boldsymbol{R}_{TG} + \boldsymbol{V}_{TG}\tau + \boldsymbol{A}_{TG}\frac{\tau^2}{2} + \boldsymbol{J}_{TG}\frac{\tau^3}{6} + \boldsymbol{S}_{TG}\frac{\tau^4}{24} \tag{19}$$

where $\boldsymbol{R}_{RG}$ is the position vector on the reference trajectory at time τ, and $\boldsymbol{R}_{TG}$, $\boldsymbol{V}_{TG}$, $\boldsymbol{A}_{TG}$, $\boldsymbol{J}_{TG}$, and $\boldsymbol{S}_{TG}$ are the target position, velocity, acceleration, jerk, and snap (first and second derivatives of acceleration, respectively). The reference velocity $\boldsymbol{V}_{RG}$ and reference acceleration $\boldsymbol{A}_{RG}$ are defined similarly by taking the first and second derivative of Eq. (19), respectively:

$$\boldsymbol{V}_{RG}(\tau) = \frac{d\boldsymbol{R}_{RG}}{d\tau} = \boldsymbol{V}_{TG} + \boldsymbol{A}_{TG}\tau + \boldsymbol{J}_{TG}\frac{\tau^2}{2} + \boldsymbol{S}_{TG}\frac{\tau^3}{6} \tag{20}$$

$$\boldsymbol{A}_{RG}(\tau) = \frac{d\boldsymbol{V}_{RG}}{d\tau} = \boldsymbol{A}_{TG} + \boldsymbol{J}_{TG}\tau + \boldsymbol{S}_{TG}\frac{\tau^2}{2} \tag{21}$$

Note that these derivatives are valid due to the fact that the dynamics of t and τ are equivalent for the global controller, as seen in Eq. (11). That is:

$$\frac{d}{dt}\big(\boldsymbol{R}_{RG}(\tau)\big) = \frac{d}{d\tau}\big(\boldsymbol{R}_{RG}(\tau)\big) * \frac{d}{dt}(\tau) = \frac{d}{d\tau}\big(\boldsymbol{R}_{RG}(\tau)\big) \tag{22}$$

B. 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 provlem() 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. (3-8). 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:

$$\boldsymbol{r}_L(t_f) = \boldsymbol{r}_L(\tau) + \boldsymbol{v}_L(\tau)t_{go} + \int_\tau^{t_f}(t_f - \mathrm{t})\big(\boldsymbol{g}_L + \boldsymbol{a}_c(\mathrm{t})\big)d\mathrm{t} \tag{23}$$

$$\boldsymbol{v}_L(t_f) = \boldsymbol{v}_L(\tau) + \int_\tau^{t_f}\big(\boldsymbol{g}_L + \boldsymbol{a}_c(\mathrm{t})\big)d\mathrm{t} \tag{24}$$

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:

$$\boldsymbol{ZEM}(\tau) = \boldsymbol{r}_{Lf} - \boldsymbol{r}_L(t_f) \qquad , \boldsymbol{a}_c(\mathrm{t}) = \boldsymbol{0}, \mathrm{t} \in [\tau, t_f] \tag{25}$$

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:

$$\boldsymbol{ZEV}(\tau) = \boldsymbol{v}_{Lf} - \boldsymbol{v}_L(t_f) \qquad , \boldsymbol{a}_c(\mathrm{t}) = \boldsymbol{0}, \mathrm{t} \in [\tau, t_f] \tag{26}$$

Here, $\boldsymbol{r}_{L_f}$ and $\boldsymbol{v}_{L_f}$ are fixed parameters that defined 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 and velocity,* $\mathbf{r}_{\mathrm{Lf}}$ *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:*

$$J(\boldsymbol{a}_c) = \int_\tau^{t_f} \boldsymbol{a}_c(\mathrm{t})^T \boldsymbol{a}_c(\mathrm{t})d\mathrm{t} \tag{27}$$

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.:

$$\boldsymbol{a}_c(\tau) = \boldsymbol{A}_1\tau - \boldsymbol{A}_2 \tag{28}$$

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

$$\boldsymbol{a}_c(\tau) = \frac{k_R}{t_{go}^2}\boldsymbol{ZEM}(\tau) + \frac{k_V}{t_{go}}\boldsymbol{ZEV}(\tau) \tag{29}$$

Here $k_R = 6$, and $k_V = -2$ 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. (9) by using the definitions of $\boldsymbol{ZEM}$ and $\boldsymbol{ZEV}$.[9]

$$\boldsymbol{a}_c(\tau) = \frac{k_1}{t_{go}^2}\,\boldsymbol{\varepsilon}_r + \frac{k_2}{t_{go}}\boldsymbol{\varepsilon}_v - \boldsymbol{g} \tag{30}$$

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:

$$\boldsymbol{s} = \boldsymbol{\varepsilon}_{\boldsymbol{v}} + \tilde{\lambda}\boldsymbol{\varepsilon}_r = \frac{\boldsymbol{\varepsilon}_v}{\tilde{\lambda}} + \boldsymbol{\varepsilon}_r = \boldsymbol{0} \tag{31}$$

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. (31) and substitute the definitions of $\boldsymbol{\varepsilon}_r$ and $\boldsymbol{\varepsilon}_v$:

$$\frac{d}{dt}\boldsymbol{s} = \frac{d}{d\tau}\boldsymbol{s} * \frac{d}{dt}\tau = \dot{\boldsymbol{\varepsilon}}_v + \tilde{\lambda}\dot{\boldsymbol{\varepsilon}}_r = \boldsymbol{g} + \boldsymbol{a}_c + \tilde{\lambda}\boldsymbol{\varepsilon}_v \tag{32}$$

If the optimal $\boldsymbol{a}_c$, as shown in Eq. (30) is substituted into Eq. (32), we obtain:

$$\frac{d}{dt}\boldsymbol{s} = \frac{k_2+\tilde{\lambda}t_{go}}{t_{go}}\boldsymbol{\varepsilon}_v + \frac{k_1}{t_{go}^2}\boldsymbol{\varepsilon}_r = -K(\tau)\big(\boldsymbol{\varepsilon}_v + \tilde{\lambda}\boldsymbol{\varepsilon}_r\big) = -K(\tau)\boldsymbol{s} \tag{33}$$

The following relationships between the parameters can be easily found:

$$K(\tau) = -\frac{k_2+\tilde{\lambda}t_{go}}{t_{go}} \tag{34}$$

$$\tilde{\lambda}K(\tau) = -\frac{k_1}{t_{go}^2} \tag{35}$$

$$t_{go}^2\tilde{\lambda}^2 + k_2 t_{go}\tilde{\lambda} - k_1 = 0 \tag{36}$$

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:

$$\frac{d}{dt}s = -K(\tau)s - \Phi sign(s) \tag{37}$$

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

$$\boldsymbol{a}_c(\tau) = \frac{k_1}{t_{go}^2}\boldsymbol{\varepsilon}_r + \frac{k_2}{t_{go}}\boldsymbol{\varepsilon}_v - \boldsymbol{g} - \frac{\Phi}{t_{go}}sign(\boldsymbol{s}) \tag{38}$$

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:

$$V = \frac{1}{2}s^T s = \frac{1}{2}\left(\boldsymbol{\varepsilon_v} + \tilde{\lambda}\boldsymbol{\varepsilon_r}\right)^T\left(\boldsymbol{\varepsilon_v} + \tilde{\lambda}\boldsymbol{\varepsilon_r}\right) \tag{39}$$

Differentiating with respect to time, we obtain:

$$\frac{d}{dt}V = \frac{d}{d\tau}V * \frac{d}{dt}\tau = \boldsymbol{s}^T\frac{d}{d\tau}\boldsymbol{s} = \boldsymbol{s}^T\left(\dot{\boldsymbol{\varepsilon}}_v + \tilde{\lambda}\dot{\boldsymbol{\varepsilon}}_r\right) \tag{40}$$

Inserting the expressions for the derivative of $\boldsymbol{\varepsilon_r}$ and $\boldsymbol{\varepsilon_v}$:

$$\begin{aligned}\frac{d}{d\tau}V = \boldsymbol{s}^T\frac{d}{d\tau}\boldsymbol{s} &= \boldsymbol{s}^T\left(\frac{k_2+\tilde{\lambda}t_{go}}{t_{go}}\varepsilon_v + \frac{k_1}{t_{go}^2}\varepsilon_r - \frac{\Phi}{t_{go}}sign(\boldsymbol{s}) + \boldsymbol{p}(\tau)\right) = \\ &= \boldsymbol{s}^T\left(\left(\frac{k_2 t_{go}\tilde{\lambda}+t_{go}^2\tilde{\lambda}^2}{\tilde{\lambda}t_{go}^2}\boldsymbol{\varepsilon_v} + \frac{k_1}{t_{go}^2}\boldsymbol{\varepsilon_r}\right) - \frac{\Phi}{t_{go}}sign(\boldsymbol{s}) + \boldsymbol{p}(\tau)\right) = \\ &= s^T\left(\left(\frac{k_1}{\lambda t_{go}^2}\varepsilon_v + \frac{k_1}{t_{go}^2}\varepsilon_r\right) - \frac{\Phi}{t_{go}}sign(s) + \boldsymbol{p}(\tau)\right) \\ &= \frac{k_1}{t_{go}^2}\boldsymbol{s}^T\boldsymbol{s} + \left(-\frac{\Phi}{t_{go}}sign(\boldsymbol{s}) + \boldsymbol{p}(\tau)\right)\end{aligned} \tag{41}$$

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:

$$\frac{d}{d\tau}V = -\frac{6}{t_{go}^2}\|\boldsymbol{s}\|^2 - \boldsymbol{s}^T\left(\frac{\Phi}{t_{go}}sign(\boldsymbol{s}) + \boldsymbol{p}(\tau)\right) \leq 0 \tag{42}$$

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

C. Local Guidance Law: Linear Quadratic Regulator (LQR)

In order to develop a LQR based guidance law, the system must first be linearized.[10] This is done by taking a Taylor expansion of the dynamics about the reference trajectory, as defined in Eq. (19):

$$\Delta\dot{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{y},\boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{y}_n,\boldsymbol{u}_n) + \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}}\right|_{\mathbf{y_n},\mathbf{u_n}}\Delta\boldsymbol{y} + \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right|_{\mathbf{y_n},\mathbf{u_n}}\Delta\boldsymbol{u} + \mathcal{O}(\|\Delta\boldsymbol{y}\|^2, \|\Delta\boldsymbol{u}\|^2) \tag{43}$$

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. (43) can be re-written as

$$\Delta\dot{\boldsymbol{y}} = A\Delta\boldsymbol{y} + B\Delta\boldsymbol{u} \tag{44}$$

where A and B are defined as the derivatives of the equations of motion with respect to $\boldsymbol{x}$ and $\boldsymbol{u}$ evaluated on the reference trajectory, respectively, as seen in Eq. (43). Notice that Eq. (44) 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:

$$\Delta\boldsymbol{u}(\boldsymbol{y}) = \mathbf{a_c} = -k(\boldsymbol{y} - \boldsymbol{y}_n) \rightarrow \Delta\dot{\boldsymbol{y}} = (A - Bk)\Delta\boldsymbol{y} = A_c\Delta\boldsymbol{y} \tag{45}$$

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:

$$J = \int_0^\infty (\Delta\boldsymbol{y}^T Q \Delta\boldsymbol{y} + \Delta\boldsymbol{u}^T R \Delta\boldsymbol{u}) d\tau = \int_0^\infty \Delta\boldsymbol{y}^T (Q + k^T R k) \Delta\boldsymbol{y}\, d\tau \tag{46}$$

Subject to Eq. (44) as a physical constraint.

In Eq. (46), 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. (46), is set to be true:

$$\boldsymbol{y}^T (Q + k^T R k)\boldsymbol{y} = -\frac{d}{d\tau}(\boldsymbol{y}^T P \boldsymbol{y}) \tag{47}$$

then the following is also true:

$$-(Q + k^T R k) = (A - Bk)^T P + P(A - Bk) \tag{48}$$

Eq. (48) 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. (48) is true. In other words, P satisfies what is known as the Reduced Riccati Equation, i.e.:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{49}$$

If Eq. (49) is held true, the optimal problem shown in Eq. (46) can be solved analytically, with the solution for k being such that

$$k = R^{-1}B^T P \tag{50}$$

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.[11]

V. Guidance Algorithms Implementation and Performance

A. Guidance Law Implantation

The proposed guidance algorithm's flow diagrams are illustrated in Figure 3. The figure shows details of the guidance algorithms as they would be implemented on the lander's onboard avionic system. The classical GNC configuration of a lunar lander (e.g. Apollo GNC subsystem[1,2,12]) requires that the onboard navigation sensors have measurements in real-time. These measurements are inputs for the navigation filters which estimate the current state vector (i.e. position and velocity). The outputs from the filters are used by the guidance algorithm, which computes the acceleration command required to drive the lander to the targeted position with the desired terminal velocity. The acceleration command is used as an input to the attitude maneuver routine coupled with a digital autopilot which controls the on-board attitude thrusters and finds the proper firing sequence that rotates the lander to match the actual thrust direction with the desired acceleration direction. This in turn drives the lander's dynamics which cause the onboard navigation sensors to sense body acceleration and rates completing the control loop.

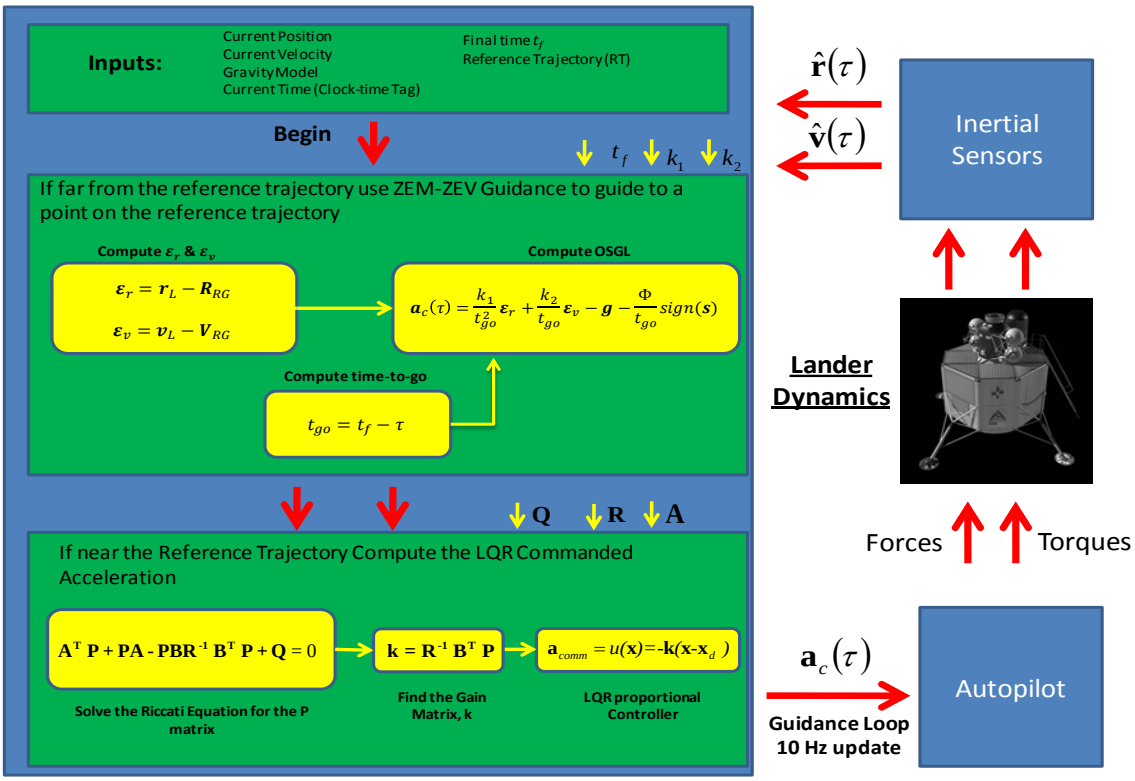

Figure 3. Schematics of the Hybrid Guidance algorithm as implemented in the on-board guidance system

The Hybrid Guidance algorithm seen in Figure 3 is embedded in a guidance loop that provides the required acceleration command to the lander autopilot within a specified time interval called the guidance frame. The Hybrid Guidance Law has been designed to target a specific point on a reference trajectory that has been designed according to the Apollo method[1,2]. Once the Lander is close enough to the reference trajectory the guidance law switches to follow it. The algorithm does not track a specific trajectory prior to the switch, but rather the instantaneous acceleration command is computed such that a) the control effort is minimized, b) the lander will be brought to the neighborhood of the reference trajectory and c) the control law is robust against unmodeled dynamics and uncertain parameters. This guidance is used as the global controller in the hybrid frame work. When the lander is close to the reference trajectory, it switches to the local control and follows the reference trajectory to the desired landing point. At any time during the lunar descent, the Hybrid Guidance law accepts a sequence of inputs which includes an update of the state vector (current lander position and velocity), the desired reference trajectory to guide to, the final time at which the lander will arrive at the desired target point, the current time (clock-time of a current state) and an appropriate gravity model as well as the time-to-go to the guide point. To determine the proper acceleration command, the current $\boldsymbol{\varepsilon}_r$ and $\boldsymbol{\varepsilon}_v$ must be computed to guide to point and its corresponding velocity on the reference

trajectory. The on-line integration is executed assuming that gravity is the sole force on acting on the lander. After the simulation is executed, both $\boldsymbol{\varepsilon}_r$ and $\boldsymbol{\varepsilon}_v$ are computed. The guidance gains (k_1 and k_2) are evaluated a-prioi to ensure optimality and robustness of the guidance algorithms. When the lander approaches the guide to point on the reference trajectory, the Hybrid Guidance Law checks if the lander is within the specified constraints of the jump and flow sets, based on the current time-to-go estimate. If the lander is within the constraints, the hybrid guidance law switches to the LQR guidance control law and follows the reference trajectory to the desired landing point. Equivalently, if the state of the lander strays from the defined neighborhood of the reference trajectory for any reason (i.e. large perturbation, thruster mis-firing, etc.), the hybrid guidance law will switch back to the global OSG law. The reference trajectory is built using the P64 targeting algorithm[1,2] from the Apollo Guidance law design. The guidance command is updated every 1/10 second implying a guidance loop that operates at 10 Hz (this is approximately one order of magnitude higher than the Apollo guidance cycle). The Hybrid Guidance algorithm performance must be tested under realistic conditions to verify their practical implementation for real-time guidance. The next sections describe the testing approach and the results of the Monte Carlo simulations executed to determine the performances of the proposed algorithms.

B. Hybrid Guidance Law for Precision Lunar Landing: Performance

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. (1-2) 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 Perturbations Values

Initial Perturbations	Mean Value	Standard Dev.
Downrange	**-7500 m**	**500 m**
Altitude	**2500 m**	**200 m**
Crossrange	**0 m**	**100 m**
Velocity	**225.6 m/s**	**30 m/s**
Flight Path Angle	**1.77 rad (101 deg)**	**20 mrad**
Crossing Angle	**1.57 rad (90 deg)**	**20 mrad**
Mass	**31624 kg**	**1000 kg**
Mass Flow Rate	**0 %**	**20%**
Thrust angle bias	**0 rad**	**30 mrad**
Roll angle bias	**0 rad**	**50 mrad**
Disturbing Acceleration	**0 m/sec2**	**Uniform Distribution**

Following Chomel and Bishop[6], the mass of the lander is set to be 31,624 Kg with a nominal specific impulse of 459.7 sec. The nominal lander entry point, which is the simulation initial condition, is set to be at an altitude of 2,500 meters, 7,500 meters downrange from the target point and zero crossrange. These conditions are very close to the ideal entry point of the trajectory designed for the Apollo approach phase. Indeed, both nominal entry position and velocity for the lander descent have been computed by implementing the P64 targeting algorithm[1,2] and computing a quartic polynomial trajectory from a family of trajectories suitable for the LQR guidance law. The same polynomial is also used as the reference trajectory that the lander is guided to, and then follows until it lands. 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 zero mean and prescribed 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 verify the robustness of the proposed algorithm.

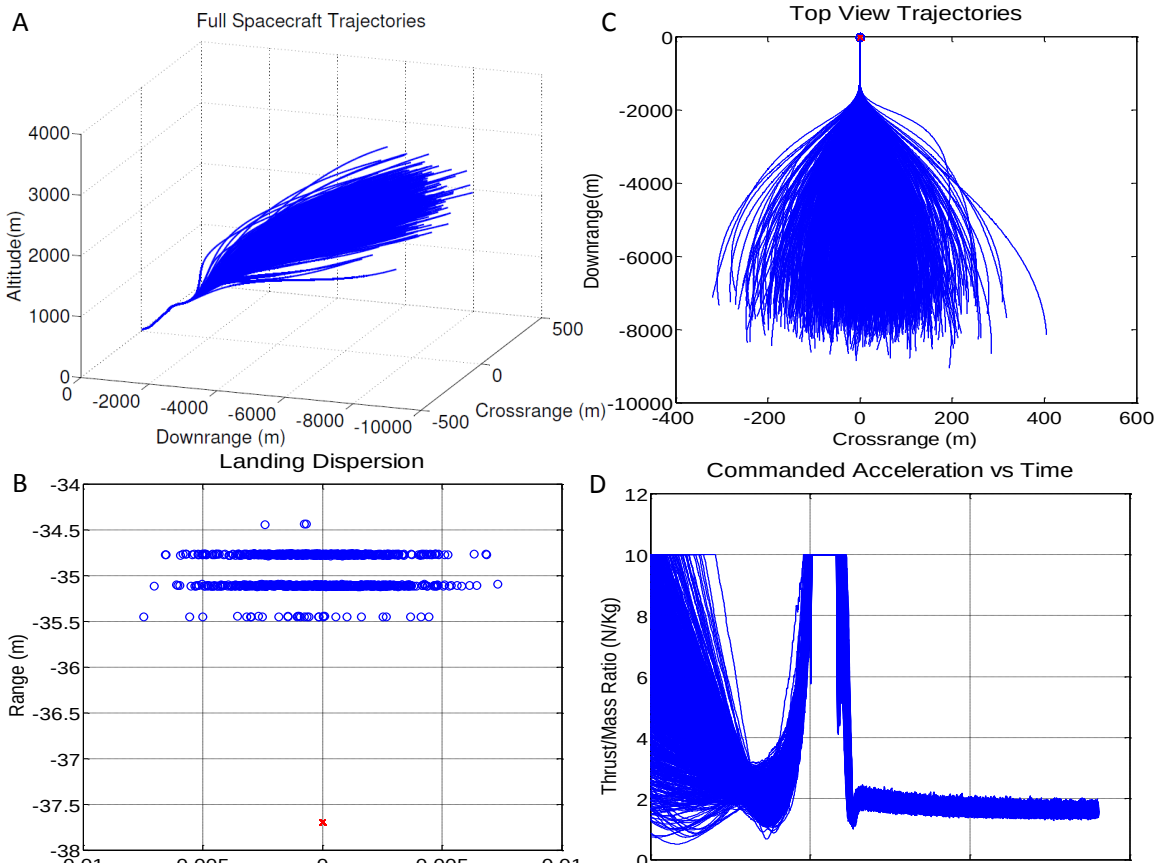

Figure 4. Monte Carlo histories for the Hybrid Guidance algorithm simulations. a) 3-D trajectories of the descending lander. b) Downrange-Crossrange projection of the trajectory histories. c) Histories of the landing accuracy. d) Acceleration command histories (Thrust-to-Mass ratio).

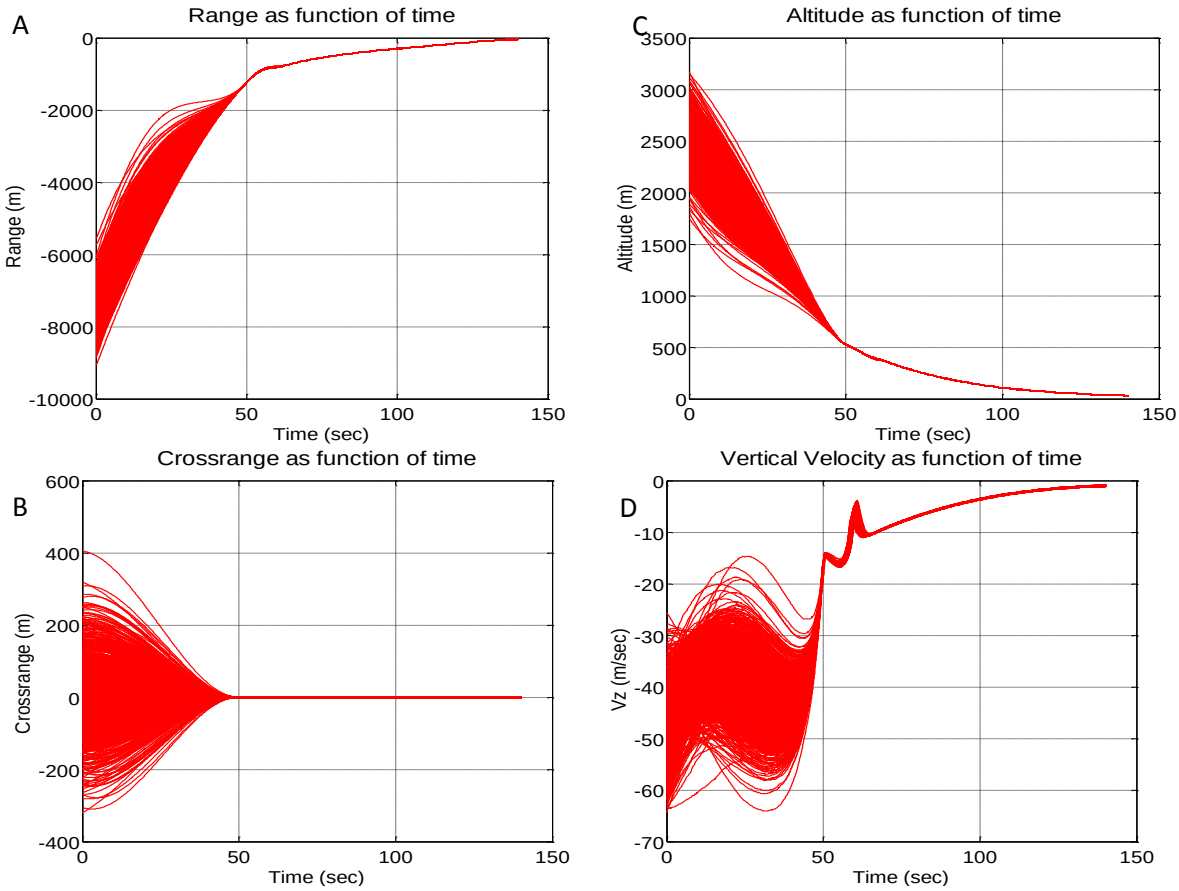

Figure 5. Monte Carlo state history results for the Hybrid Guidance Monte Carlo simulations. a) Range Projection of the Trajectory History b)Crossrange Projection of the Trajectory History c)Altitude Projection of the Trajectory History d)Vertical Velocity Histories

For the Hybrid Guidance algorithm 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 does need 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. The algorithm was asked to target a reference trajectory that ended in a point that is located at an altitude of 30 m above the desired landing point and -37.7 meters downrange (assumed to be at the origin of the guidance reference frame). The Hybrid Guidance law also doesn't target zero velocity; in this case the targeted velocity was 0 m/s crossrange, 3.769 m/s downrange and -1 m/s in altitude. The Hybrid Guidance algorithm does not target a point directly on the surface with zero velocity 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 while the parameters for the reference trajectory used by the trajectory following part of the hybrid guidance is done following the P64 targeting algorithm[1,2] and computing a quartic polynomial trajectory from a the desired landing point and nominal entry point. The terminal state statistics are reported in Table 3.

Table 2 Hybrid Guidance Parameters

Guidance Parameter	Value
Position gain, k_1	6
Velocity gain, k_2	-2
Sliding parameter, Φ	0.4
Q	$5 \cdot \mathrm{I}_{6x6}$
R	$0 \cdot \mathrm{I}_{3x3}$
Λ_{r_1}	50 m
Λ_{v_1}	5 m/s
Λ_{r_2}	55 m
Λ_{v_2}	6 m/s

Table 3 Monte Carlo final states error statistics for the Hybrid Guidance algorithm

	Nominal	Mean	Abs Error	Standard Dev
Altitude (m)	**30**	**30**	**0**	**0**
Downrange (m)	**-37.7**	**-34.9773**	**-34.9773**	**0.1891**
Crossrange (m)	**0**	**-0.000172**	**0.000172**	**0.0024**
Velocity X (m/sec)	**0**	**-1.1323e-4**	**1.1323e-4**	**0.0025**
Velocity Y (m/sec)	**3.7697**	**3.5293**	**3.5293**	**0.0224**
Velocity Z (m/sec)	**-1.00**	**-1.0037**	**-1.0037**	**0.0477**

Generally, the algorithm performs very well. Figure 4b shows the landing disruption that highlights the precision capabilities of the Hybrid Guidance algorithm. On the average, the desired 30 m altitude point was achieved with a precision within a few centimeters. Importantly, all the final landing points resulting from the 1000 simulated guided trajectories fall within dispersion characterized by mean downrange error of 2.7 m (absolute value) with a standard deviation of 189 centimeters assuming a normal distribution. The distribution of crossrange error is nearly zero (absolute value) with a standard deviation of 0.0024 m also assuming a normal distribution. The acceleration command generated by the Hybrid Guidance algorithm is generally higher at the beginning of the landing descent, generally reaching its maximum close to when the guidance law switch occurs. On average, as time increases (and time-to-go decreases), the acceleration command decreases monotonically, reaches a minimum and then increases to a relative peak right before reaching the switch point, at which is decreases until the target point is reached. The acceleration also 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 switching 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. The maximum commanded acceleration value was limited to 10 N/kg, in order to provide a realistic maximum thrust value. This limiter come into play mostly around the switch, as the acceleration command generated by the global guidance law reaches a relative peak just before it reaches the target (switch) point. In general, even with a limit applied to the commanded acceleration, the guidance law still performs well, which low residual errors in position and velocity.

VI. Conclusion

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 (trajectory generation from Apollo) and real-time guidance (trajectory following). The global controller break that paradigm, using a formalism borrowed from recent advancements in non-linear, higher-order sliding mode control theory, generating 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 study and implementation of the hybrid guidance is desired to examine the potential combination of other guidance laws to improve performance or meet other mission criteria, e.g. limiting the contamination of the landing site and to analyze the performance under extreme perturbations that cause the guidance algorithm to switch back the global controller after reaching the neighborhood of the reference trajectory.

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]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.

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

[11]Antsaklis, Panos J., Michel, Anthony N., *Linear Systems*, McGraw-Hill, Engelwood Cliffs, NJ, 1997, pp. 326-350.

[12]Klumpp, A. R., "A Manually Retargeted Automatic Landing System for the Lunar Module (LM)," *Journal of Spacecraft and Rockets*, Volume 5, Issue 2, 1968, pp 129-138