JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 32, No. 3, May–June 2009

Analytical Lunar Descent Guidance Algorithm

Christina T. Chomel*
University of Texas at Austin, Austin, Texas 78713
and
Robert H. Bishop†
University of Texas at Austin, Austin, Texas 78712

DOI: 10.2514/1.37700

The desire to return humans to the moon has motivated the evaluation of existing lunar descent guidance algorithms and the development of new ideas. This paper outlines a design for a targeting algorithm that quickly and reliably generates a two-dimensional reference trajectory and a real-time three-dimensional guidance algorithm that can use this or any reference trajectory as its basis. The targeting algorithm is analytical in nature and allows for updates to the reference in real time. Monte Carlo analysis demonstrates the reliable performance of the algorithms in both a general sense and for retargeting once the lunar descent has been initiated.

Nomenclature

- d = range from targeted landing location to current spacecraft location projected along the lunar surface parallel to the desired bearing β, or downrange
- c = range from targeted landing location to current spacecraft location projected along the lunar surface perpendicular to the desired bearing β, or crossrange
- $\hat{\mathbf{e}}_1^M$ = one-unit vector in a right-handed maneuver reference frame, $\mathbf{v}/v$
- $\hat{\mathbf{e}}_2^M$ = two-unit vector in a right-handed maneuver reference frame, $(\mathbf{r} \times \mathbf{v})/ \parallel \mathbf{r} \times \mathbf{v} \parallel$
- $\hat{\mathbf{e}}_3^M$ = three-unit vector in a right-handed maneuver reference frame, $\hat{\mathbf{e}}_1^M \times \hat{\mathbf{e}}_2^M$
- g = gravity vector magnitude ($\|\mathbf{g}\|$)
- h = spacecraft altitude above the lunar surface
- h_S = reference radius of the spherical lunar model
- m = spacecraft mass
- T = thrust vector magnitude ($\|\mathbf{T}\|$)
- t = time
- v = spacecraft velocity vector magnitude ($\|\mathbf{v}\|$) or speed
- α = angle of the thrust vector with respect to $-\mathbf{v}$ in the local vertical plane
- θ = angle between the local horizontal and velocity vector, or flight-path angle
- ϕ = rotation angle of the thrust vector about $\mathbf{v}$ vector, or thrust roll angle
- ψ = angle between the local vertical plane and velocity vector projection into the local horizontal plane, or crossing angle

I. Introduction

IN RECENT years, NASA has indicated a desire to return humans to the moon. With NASA planning manned missions within the next couple of decades, the concept development for these lunar vehicles has begun. The guidance, navigation, and control computer programs that will perform the function of safely landing a spacecraft on the moon are part of that development. The lunar descent guidance algorithm takes the horizontally oriented spacecraft from orbital speeds at a point hundreds of kilometers from the desired landing point to the landing point at an almost vertical orientation and very low speed. Existing lunar descent guidance, navigation, and control algorithms date back to the Apollo era [1,2]. Though these algorithms met the criteria of the 1960s, they cannot meet the more stringent landing requirements of a precision landing at any time and portion of the mission. Additional work is available from the Apollo era that evaluates a two-dimensional gravity-turn descent algorithm [3–5] with more recent complementary work [6–8]. The targeting solution presented in this paper will have the essence of this gravity-turn descent algorithm but will develop the equations into a readily implementable algorithm.

The design and implementation of a lunar descent is composed of two elements: the targeting, which generates a reference trajectory, and the real-time guidance, which forces the spacecraft to fly that trajectory. The Apollo algorithm used a complex, iterative, numerical optimization scheme for computing the reference trajectory. The reference was generated premission in the form of two quartic polynomials. The solution required a combination of analytical equations, iterative simulation, and user judgement regarding the validity of the solution to develop the trajectory. The real-time guidance used this reference trajectory to force the real-time trajectory errors to converge to zero. The drawback of the real-time guidance algorithm is that it required the reference to be in the form of the two quartic polynomials, such that a generic reference trajectory could not be used.

The proposed algorithm presented in this paper [9] implements an analytical targeting algorithm used in real time to quickly and reliably generate two-dimensional trajectories before descent initiation or retarget to another landing site after descent initiation. It is based on analytical solutions to the equations for speed, downrange, altitude, and time as a function of flight-path angle and assumes two arcs of constant thrust acceleration. The proposed real-time guidance algorithm uses the three-dimensional nonlinear equations of motion and a control law that is proven to converge under certain conditions through Lyapunov analysis to a reference trajectory presented as a function of downrange, altitude, speed, and flight-path angle. The two elements of the guidance algorithm are investigated using Monte Carlo analysis to prove their robustness to initial state dispersions and mass and thrust errors. The robustness of the retargeting algorithm is also demonstrated.

II. Target Trajectory Design

The fundamental three-dimensional equations of motion used to describe the spacecraft motion with respect to a uniformly

Presented as Paper 08-065 at the AAS/AIAA Guidance and Control Conference, Breckenridge, CO, 1–6 February 2008; received 24 March 2008; revision received 15 December 2008; accepted for publication 15 December 2008.

*Engineering Scientist, Advanced Technology Laboratory, Applied Research Laboratories, P. O. Box 8029. Member AIAA.

†Professor and Chairman, Department of Aerospace Engineering and Engineering Mechanics, 1 University Station, Mail Code C0600. Fellow AIAA.

homogenous spherical lunar body are given by

$$\dot{v} = g\sin\theta - \frac{T}{m}\cos\alpha \tag{1}$$

$$v\dot{\theta} = \left(g - \frac{v^2}{h+h_S}\right)\cos\theta - \frac{T}{m}\sin\alpha\cos\phi \tag{2}$$

$$v\dot{\psi}\cos\theta = -\frac{T}{m}\sin\alpha\sin\phi \tag{3}$$

$$\dot{h} = -v\sin\theta \tag{4}$$

$$\dot{d} = v\cos\theta\cos\psi\frac{h_S}{h+h_S} \tag{5}$$

$$\dot{c} = v\cos\theta\sin\psi\frac{h_S}{h+h_S} \tag{6}$$

where Fig. 1 illustrates the states, and the derivations of these fundamental equations is discussed in detail in [9].

A. Solution Assuming Constant Thrust Acceleration

When developing the targeting algorithm, various assumptions are made to allow the equations of motion to be solved analytically. First, the motion is assumed to be constrained to two dimensions, implying that $c(0) = 0$, $\dot{c}(0) = 0$, $\psi(0) = 0$, $\dot{\psi}(0) = 0$, $\phi(t) = 0$, and $\dot{\phi}(t) = 0$, reducing the equations to

$$\dot{v} = g\sin\theta - \frac{T}{m}\cos\alpha \tag{7}$$

$$v\dot{\theta} = \left(g - \frac{v^2}{h+h_S}\right)\cos\theta - \frac{T}{m}\sin\alpha \tag{8}$$

$$\dot{h} = -v\sin\theta \tag{9}$$

$$\dot{d} = v\cos\theta\frac{h_S}{h+h_S} \tag{10}$$

Analytical solutions can be developed in a similar fashion to the classical gravity-turn descent work [3–5] by recognizing

$$\frac{dv}{d\theta} = \frac{dv}{dt}\Big/\frac{d\theta}{dt} \tag{11}$$

$$\frac{dh}{d\theta} = \frac{dh}{dt}\Big/\frac{d\theta}{dt} \tag{12}$$

$$\frac{dd}{d\theta} = \frac{dd}{dt}\Big/\frac{d\theta}{dt} \tag{13}$$

$$\frac{dt}{d\theta} = 1\Big/\frac{d\theta}{dt} \tag{14}$$

such that the two-dimensional equations can be solved as a function of flight-path angle given some assumptions. Starting with Eq. (11), recognize

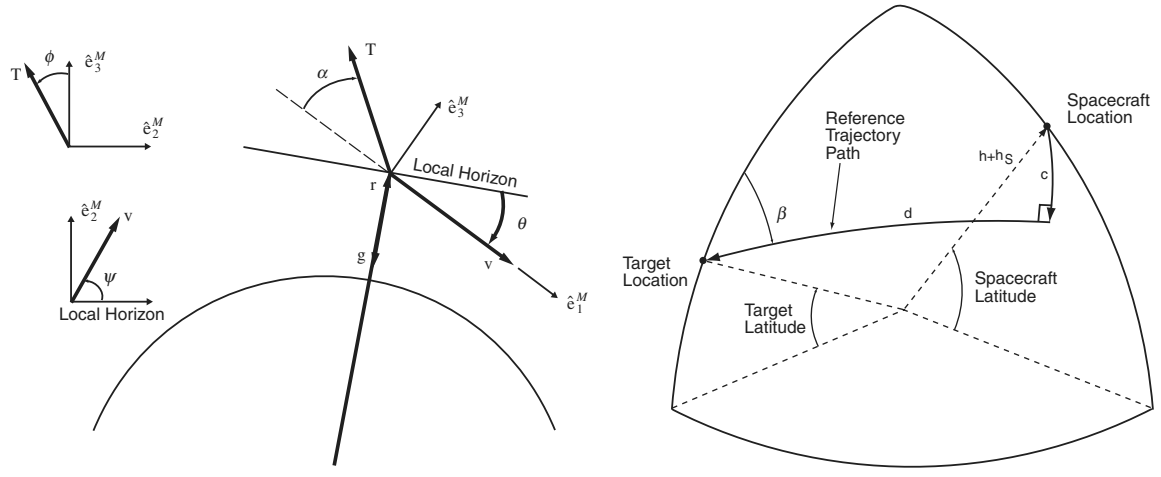
$$\frac{dv}{d\theta}\Big/v = \frac{g\sin\theta - (T/m)\cos\alpha}{(g - (v^2/h + h_S))\cos\theta - (T/m)\sin\alpha} \triangleq \Phi(\theta) \tag{15}$$

which can be solved analytically if the right-hand side of the equation is reduced to a simple function of θ. First, gravity is assumed to be constant. Second, the thrust-to-mass ratio or thrust acceleration T/m is assumed to be constant; though this would appear to be an undesirable assumption, as will be shown later, the real-time guidance readily compensates for the varying mass. Third, the thrust angle α is assumed to be zero; at a minimum, α is required to be constant for an analytical solution and no benefit was found to setting this to a nonzero value. This follows the classical gravity-turn descent concept. Finally, the assumption must be made that the centrifugal acceleration term $v^2/(h + h_S)$ is constant. Though this is obviously a highly varying value, the real-time guidance will compensate for the errors between the model and the environment. The centrifugal acceleration term is set to a percentage of the gravity such that

$$g - \frac{v^2}{h+h_S} \approx (1-k)g \tag{16}$$

With these assumptions, Eq. (15) reduces to

$$\frac{dv}{d\theta}\Big/v = \frac{g\sin\theta - (T/m)}{(1-k)g\cos\theta} \triangleq \Phi(\theta) \tag{17}$$



a) Dynamic states **b) Kinematic states**

Fig. 1 Description of states.

or

$$\frac{dv}{v} = \Phi(\theta)d\theta \tag{18}$$

which can be integrated as

$$\ell_n \frac{v}{v_0} = \int_{\theta_0}^{\theta} \Phi(\xi)d\xi \tag{19}$$

where v_0 is the initial speed and θ_0 is the initial flight-path angle. If the exponential of both sides is taken, the velocity becomes a function of flight-path angle:

$$v(\theta) = v_0 e^{\int_{\theta_0}^{\theta} \Phi(\xi)d\xi} \tag{20}$$

The integration of $\Phi(\theta)$, along with some trigonometric manipulations, yields

$$\begin{aligned} v(\theta) &= v_0 \left(\frac{\cos\theta_0}{\cos\theta}\right)^{\frac{1}{1-k}} \left(\frac{\cos\theta_0}{1-\sin\theta_0} \Big/ \frac{\cos\theta}{1-\sin\theta}\right)^{\frac{T/m}{(1-k)g}} \\ &= v_0 \frac{(\cos\theta_0)^{\frac{g+(T/m)}{(1-k)g}}}{(1-\sin\theta_0)^{\frac{T/m}{(1-k)g}}} \frac{(1-\sin\theta)^{\frac{T/m}{(1-k)g}}}{(\cos\theta)^{\frac{g+(T/m)}{(1-k)g}}} \end{aligned} \tag{21}$$

where $\theta \neq \pm\pi/2$ and $\theta_0 \neq \pi/2$.

At this point, define the constant

$$p \triangleq \frac{T/m}{g} \tag{22}$$

where $p \in \Re$ and $p > 0$, so that

$$v(\theta) = v_0 \frac{(\cos\theta_0)^{2p+2}}{(1-\sin\theta_0)^{2p}} \frac{(1-\sin\theta)^{2p}}{(\cos\theta)^{2p+2}} \tag{23}$$

This equation is similar to that developed in the gravity-turn descent development. The primary difference in that work is that in the initial work [3,5], no attempt was made to compensate for the centrifugal acceleration term, implying $k = 0$, where the latter work compensated for this term [6].

The time, altitude, and downrange differential equations are

$$\frac{dt}{d\theta} = \kappa_t \frac{(1-\sin\theta)^{\frac{p}{(1-k)}}}{(\cos\theta)^{\frac{1+p}{(1-k)}+1}} \tag{24}$$

$$\frac{dh}{d\theta} = -\kappa_h \frac{(1-\sin\theta)^{\frac{2p}{(1-k)}} \sin\theta}{(\cos\theta)^{\frac{2+2p}{(1-k)}+1}} \tag{25}$$

$$\frac{dd}{d\theta} = \kappa_d \frac{(1-\sin\theta)^{\frac{2p}{(1-k)}}}{(\cos\theta)^{\frac{2+2p}{(1-k)}}} \tag{26}$$

where

$$\kappa_t \triangleq \frac{v_0}{(1-k)g} \frac{(\cos\theta_0)^{\frac{p+1}{(1-k)}}}{(1-\sin\theta_0)^{\frac{p}{(1-k)}}} \tag{27}$$

$$\kappa_h \triangleq \frac{v_0^2}{(1-k)g} \frac{(\cos\theta_0)^{2\frac{p+1}{(1-k)}}}{(1-\sin\theta_0)^{2\frac{p}{(1-k)}}} \tag{28}$$

$$\kappa_d \triangleq \frac{v_0^2}{(1-k)g} \frac{(\cos\theta_0)^{2\frac{p+1}{(1-k)}}}{(1-\sin\theta_0)^{2\frac{p}{(1-k)}}} \tag{29}$$

The authors could find no analytical solution to these equations for any $k \in \Re$, but did find that they could be solved if $1/(1-k)$ is an integer or $k = \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}, \ldots$. Two approaches were taken to determine an acceptable value for k. First, if the centrifugal acceleration term $v^2/(h+h_S)$ is integrated by approximating speed as a linear function of time from starting at orbital speed and linearly decaying to zero, a reasonable approximation is $k = \frac{1}{3}$. However, this value is not allowed. Therefore, the differential equations for speed, time, altitude, and downrange were numerically integrated with no approximation for the centrifugal acceleration term. These numerical solutions were compared with the analytical solutions, which had various values for k. The authors felt that a value of $k = \frac{1}{2}$ provided the best comparison [9].

With the definition of k, the time, downrange, and altitude equations can be integrated. The solution for the time equation is

$$\begin{aligned} t(\theta) = -\kappa_t \Bigg(&\frac{(1-\sin\theta)^{2p}}{(2p+2)(\cos\theta)^{2p+2}} + \frac{(1-\sin\theta)^{2p-1}}{p(2p+2)(\cos\theta)^{2p}} \\ &+ \frac{(1-\sin\theta)^{2p-2}}{p(2p+2)(2p-2)(\cos\theta)^{2p-2}} \Bigg) + \eta_t \end{aligned} \tag{30}$$

where

$$\kappa_t \triangleq \frac{2v_0}{g} \frac{(\cos\theta_0)^{2p+2}}{(1-\sin\theta_0)^{2p}} \tag{31}$$

$$\eta_t = t_0 - t(\theta_0) \tag{32}$$

and t_0 is the initial time. The solution for altitude is

$$\begin{aligned} h(\theta) = -\kappa_h \Bigg(&\frac{(1-\sin\theta)^{4p}}{(4p+4)(\cos\theta)^{4p+4}} - \frac{2p(1-\sin\theta)^{4p}}{(4p+4)(4p+2)(\cos\theta)^{4p+2}} \\ &- \frac{(2p+3)(1-\sin\theta)^{4p-1}}{2(4p+4)(4p+2)(\cos\theta)^{4p}} - \frac{2p(1-\sin\theta)^{4p-3}}{(4p+4)(4p-2)(\cos\theta)^{4p-2}} \\ &+ \frac{(4p^2-3)(1-\sin\theta)^{4p-2}}{2(4p+4)(4p+2)(4p-2)(\cos\theta)^{4p-2}} \\ &+ \frac{(p+2)(1-\sin\theta)^{4p-5}}{(4p+4)(4p-4)(\cos\theta)^{4p-4}} - \frac{(p+2)(1-\sin\theta)^{4p-6}}{2(4p+4)(4p-4)(\cos\theta)^{4p-6}} \\ &+ \frac{(2-6p)(1-\sin\theta)^{4p-4}}{(4p+4)(4p-4)(4p-2)(\cos\theta)^{4p-4}} \Bigg) + \eta_h \end{aligned} \tag{33}$$

where

$$\kappa_h \triangleq \frac{2v_0^2}{g} \frac{(\cos\theta_0)^{4p+4}}{(1-\sin\theta_0)^{4p}} \tag{34}$$

$$\eta_h = h_0 - h(\theta_0) \tag{35}$$

and h_0 is the initial altitude. The solution for downrange is

$$\begin{aligned} d(\theta) = \kappa_d \Bigg(&- \frac{(1-\sin\theta)^{4p}}{(4p+3)(\cos\theta)^{4p+3}} \\ &+ \frac{(2\sin\theta-1)(1-\sin\theta)^{4p-1}}{(4p+3)(4p+1)(\cos\theta)^{4p+1}} \\ &+ \frac{(4p\sin\theta+3\sin\theta-4p-1)(1-\sin\theta)^{4p-2}}{(4p+3)(4p+1)(4p-1)(\cos\theta)^{4p-1}} \\ &+ \frac{(3-16p^2)(1-\sin\theta)^{4p-3}}{(4p+3)(4p-3)(4p+1)(4p-1)(\cos\theta)^{4p-3}} \Bigg) + \eta_d \end{aligned} \tag{36}$$

where

$$\kappa_d \triangleq \frac{2v_0^2}{g} \frac{(\cos\theta_0)^{4p+4}}{(1-\sin\theta_0)^{4p}} \tag{37}$$

$$\eta_d = d_0 - d(\theta_0) \tag{38}$$

and d_0 is the initial downrange.

B. Targeting Algorithm Design

A planar target trajectory can be generated as a function of downrange, altitude, speed, and flight-path angle. This implies that the trajectory is not specific to any one point on the lunar surface or even to a bearing from the north. Any single trajectory can be easily relocated or rotated as necessary to meet mission objectives. The target trajectory will be composed of n segments, where T/m is constant over each segment and the combination of the segments results in a cumulative downrange and altitude. These segments will be joined by the continuity conditions that the speed and flight-path angle must be continuous over each junction. If the initial downrange and altitude of the entire trajectory are given by d_0 and h_0 and the final downrange and altitude of the entire trajectory are given by d_f and h_f, then the total change in downrange and altitude is defined by

$$\Delta d = d_f - d_0 = \sum_{i=1}^{n} \Delta d_i \tag{39}$$

$$\Delta h = h_f - h_0 = \sum_{i=1}^{n} \Delta h_i \tag{40}$$

where Δd_i and Δh_i are the downrange and altitude change over each segment of constant thrust acceleration. The final flight-path angle and speed for the entire trajectory are also correlated to the segments by the fact that the final flight-path angle θ_f is the same as the nth flight-path angle θ_n, and the final speed v_f is the same as the nth speed v_n.

From Eqs. (33–36), the individual segment change in downrange and altitude are given by

$$\Delta d_i = \frac{2v_i^2}{g} D(\theta_i, p_i) - \frac{2v_{i-1}^2}{g} D(\theta_{i-1}, p_i) \tag{41}$$

$$\Delta h_i = -\frac{2v_i^2}{g} H(\theta_i, p_i) + \frac{2v_{i-1}^2}{g} H(\theta_{i-1}, p_i) \tag{42}$$

where θ_i is the flight-path angle at the end of the segment and θ_{i-1} is the flight-path angle at the beginning of the segment. Additionally,

$$\begin{aligned} H(\theta, p) = {} & \frac{1}{(4p+4)} - \frac{2p(\cos\theta)^2}{(4p+4)(4p+2)} \\ & - \frac{(2p+3)(\cos\theta)^4}{2(4p+4)(4p+2)(1-\sin\theta)} \\ & - \frac{2p(\cos\theta)^6}{(4p+4)(4p-2)(1-\sin\theta)^3} \\ & + \frac{(4p^2-3)(\cos\theta)^6}{2(4p+4)(4p+2)(4p-2)(1-\sin\theta)^2} \\ & + \frac{(p+2)(\cos\theta)^8}{(4p+4)(4p-4)(1-\sin\theta)^5} \\ & - \frac{(p+2)(\cos\theta)^{10}}{2(4p+4)(4p-4)(1-\sin\theta)^6} \\ & + \frac{(2-6p)(\cos\theta)^8}{(4p+4)(4p-4)(4p-2)(1-\sin\theta)^4} \end{aligned} \tag{43}$$

$$\begin{aligned} D(\theta, p) = {} & -\frac{\cos\theta}{(4p+3)} + \frac{(2\sin\theta - 1)(\cos\theta)^3}{(4p+3)(4p+1)(1-\sin\theta)} \\ & + \frac{(4p\sin\theta + 3\sin\theta - 4p - 1)(\cos\theta)^5}{(4p+3)(4p+1)(4p-1)(1-\sin\theta)^2} \\ & + \frac{(3-16p^2)(\cos\theta)^7}{(4p+3)(4p-3)(4p+1)(4p-1)(1-\sin\theta)^3} \end{aligned} \tag{44}$$

From Eq. (23), the speed at the end of the segment is given by

$$v_i = v_{i-1} \frac{(\cos\theta_{i-1})^{2p_i+2}}{(1-\sin\theta_{i-1})^{2p_i}} \frac{(1-\sin\theta_i)^{2p_i}}{(\cos\theta_i)^{2p_i+2}} \tag{45}$$

where v_{i-1} is the speed at the beginning of the segment and

$$p_i = \frac{T/m_i}{g_i} \tag{46}$$

where p_i is constant over the segment.

When generating a reference trajectory, the goal will typically be to specify the flight-path angle and speed at the beginning and end of the total trajectory (θ_0, θ_f, v_0, and v_f) as well as the total downrange and altitude that will be spanned by the trajectory (Δh and Δd); this constrains six variables. To formulate the trajectory, there exist $n+2$ equations [one each of Eqs. (39) and (40) and $n\times$ Eq. (45)] and $3n+4$ variables (Δd, Δh, θ_0, v_0, $n\times\theta_i$, $n\times v_i$, and $n\times T/m_i$). If a single segment is used to define the total trajectory ($n=1$), there exist three equations [Eqs. (39) and (40), where $n=1$, and Eq. (45), where $i=1$] with seven variables (θ_0, θ_1, v_0, v_1, Δd, Δh, and T/m). Given the desire to specify six of those variables (θ_0, θ_1, v_0, v_1, Δd, and Δh), the problem is quickly overconstrained. This can be remedied by adding more segments to the solution.

If two segments are used with the same goals (define θ_0, θ_f, v_0, v_f, Δd, and Δh), the problem becomes perfectly constrained. There exist four equations [Eqs. (39) and (40), where $n=2$, and two from Eq. (45), where $i=1$ and $i=2$] with 10 variables (θ_0, θ_1, θ_2, v_0, v_1, v_2, Δd, Δh, T/m_1, and T/m_2). The goal is to compute θ_1, v_1, T/m_1, and T/m_2 given the other six specified variables (θ_0, θ_2, v_0, v_2, Δd, and Δh).

If $n=2$, computing v_1 is straightforward if θ_1 is known and either the set of T/m_1, θ_0, and v_0 or T/m_2, θ_2, and v_2 are defined:

$$\begin{aligned} v_1 &= v_0 \frac{(\cos\theta_0)^{2p_1+2}}{(1-\sin\theta_0)^{2p_1}} \frac{(1-\sin\theta_1)^{2p_1}}{(\cos\theta_1)^{2p_1+2}} \\ &= v_2 \frac{(\cos\theta_2)^{2p_2+2}}{(1-\sin\theta_2)^{2p_2}} \frac{(1-\sin\theta_1)^{2p_2}}{(\cos\theta_1)^{2p_2+2}} \end{aligned} \tag{47}$$

The equality in Eq. (47) can be exploited to compute θ_1 as a function of θ_0, θ_2, v_0, v_2, p_1, and p_2:

$$\theta_1 = -2\left(\tan^{-1}\left(\frac{v_2}{v_0}\frac{(\cos\theta_2)^{2p_2+2}}{(1-\sin\theta_2)^{2p_2}}\frac{(1-\sin\theta_0)^{2p_1}}{(\cos\theta_0)^{2p_1+2}}\right)^{\frac{1}{2p_1-2p_2}} - \frac{\pi}{4}\right) \tag{48}$$

Equation (47) is then used to compute v_1. From Eqs. (39) and (40), the two-segment trajectory spans the following downrange and altitude:

$$\begin{aligned} \Delta d = {} & \frac{2v_1^2}{g}(D(\theta_1, p_1) - D(\theta_1, p_2)) - \frac{2v_0^2}{g} D(\theta_0, p_1) \\ & + \frac{2v_2^2}{g} D(\theta_2, p_2) \end{aligned} \tag{49}$$

$$\begin{aligned} \Delta h = {} & -\frac{2v_1^2}{g}(H(\theta_1, p_1) - H(\theta_1, p_2)) + \frac{2v_0^2}{g} H(\theta_0, p_1) \\ & - \frac{2v_2^2}{g} H(\theta_2, p_2) \end{aligned} \tag{50}$$

where θ_1 is given in Eq. (48) and v_1 is given in Eq. (47). This makes Δd and Δh functions of θ_0, θ_2, v_0, v_2, T/m_1, and T/m_2. The ideal situation is that Eqs. (49) and (50) can be inverted to solve for T/m_1 and T/m_2 as functions of Δd and Δh. However, upon examination of the equations, an analytical solution cannot be found for the inversion.

Given the restrictions on the inversion of Eqs. (49) and (50), some options exist for creating the targeting law. The first option is to specify all of the independent variables (θ_0, θ_2, v_0, v_2, T/m_1, and T/m_2) with reasonable values. The actual handover conditions from the orbital phase to the descent phase must then be initiated close to the downrange and altitude values generated by inserting those values for $\theta_0, \theta_2, v_0, v_2, T/m_1$, and T/m_2 into Eqs. (49) and (50). This is by far the simplest solution, but may not be reasonable because of mission constraints on where the handover will occur.

Another solution is to specify θ_0, θ_2, v_0, and v_2 and search through reasonable values for T/m_1 and T/m_2 to generate a trajectory space. A desirable trajectory can then be selected from the options available. This is not automatic, as there is no guarantee that the desired trajectory will be in the solution space. It does, however, offer the best option for merging the handover conditions between the orbital-phase termination and the descent-phase initiation in an acceptable fashion.

A sample trajectory space is given in Fig. 2. The initial and final flight-path angles were set at 0.1 and 89 deg and the initial and final speeds were set at 1688 and 8 m/s.[‡] The gravity used in this analysis is that at the lunar surface. The diamonds represent altitudes versus downranges that can be reached with Eqs. (49) and (50) by varying the values for T/m that span 0.1 to 10.1 N/kg in 0.25 N/kg increments for both T/m_1 and T/m_2. Note that the number of points generated is actually much larger than those shown in Fig. 2; the downranges and altitudes generated span values that are thousands of kilometers from the planet surface and thereby well out of a reasonable range of values. Figure 2 appears to have sets of curves. This is the impact of varying the thrust acceleration. Each curve is created by varying T/m_1 and those curves are transformed by changing T/m_2.

Depending on the trajectory design requirements, the trajectory space can be rather limited. Therefore, the initial and final flight-path angles, the initial and final speeds, and the gravity can also be varied to increase the trajectory space [9]. This can be automated into an algorithm that computes a matrix of available trajectory spaces and then selects the most desirable trajectory based on some user-defined criteria. Because this targeting algorithm is not iterative in nature, no risk of divergence exists in creating this trajectory space. However, the spacecraft may be at a distance that is too far from or too close to the targeted location for a safe landing, meaning that a desirable trajectory is not available. If the spacecraft is too far from the targeted landing site, the real-time guidance algorithm would wait. If the spacecraft is too close, the decision should be made to wait another orbit for the descent initiation.

A comparison against existing trajectory designs is always useful. A current design being evaluated by NASA Johnson Space Center engineers uses the existing Apollo-era trajectory design algorithm with some modifications (see footnote [‡]). The proposed design used the following parameters: $\theta_0 = 0.1$ deg, $\theta_f = 77.4$ deg, $v_0 = 1688$ m/s, $v_f = 8$ m/s, $T/m_1 = 4.8$ N/kg, $T/m_2 = 1.4$ N/kg, and $g = 1.136$ m/s$^2 = 0.7\mu/h_0^2$. Figure 3a shows that the comparison of the flight-path angle versus downrange for the two designs is quite close. Figure 3b illustrates that the proposed trajectory runs lower in altitude as compared with the existing trajectory. The slightly lower speed in Fig. 3c is likely the cause of the extended time in Fig. 3d (approximately 25 s increase in the timeline for the proposed trajectory). Figure 3e shows that the acceleration for the proposed design is close to the average of the first segment and drops off in the second segment, as does the existing design.

A final note on the target trajectory design should be addressed. The authors chose to develop a two-segment trajectory because it perfectly constrained the problem being addressed herein. However, more than two segments can be used to add flexibility in adding other boundary conditions and optimizing the trajectory based on requirements for elements such as mass-consumption limits or time requirements. One element not addressed in this trajectory solution is the requirement during Apollo missions of viewing angles to the lunar surface; this could be addressed by adding more segments.

III. Guidance Algorithm

The methodology chosen to design the real-time three-dimensional guidance algorithm that will control the spacecraft to the reference trajectory was to work with the nonlinear equations rather than a linear model. One method by which to develop the guidance algorithm and to prove that the states converge to the target trajectory given that algorithm is to develop a Lyapunov candidate function that can be manipulated to prove stability. Through the formulations in this section, the goal is to prove that the altitude h, downrange d, crossrange c, speed v, flight-path angle θ, and crossing angle ψ all converge to their respective target states denoted by h_{ref}, d_{ref}, c_{ref}, v_{ref}, θ_{ref}, and ψ_{ref}, regardless of how those reference states were generated.

The Lyapunov candidate function selected for this application is

$$V = \frac{1}{2}\left[\dot{h}_{\text{err}}^2 + \dot{d}_{\text{err}}^2 + \dot{c}_{\text{err}}^2 + \frac{1}{\gamma_h}k_h^2 + \frac{1}{\gamma_d}k_d^2 + \frac{1}{\gamma_c}k_c^2\right] + \frac{\beta_h A}{\lambda_{h_1}}\ell n\ \cosh\lambda_{h_1} h_{\text{err}} + \frac{\beta_d A}{\lambda_{d_1}}\ell n\ \cosh\lambda_{d_1} d_{\text{err}} + \frac{\beta_c A}{\lambda_{c_1}}\ell n\ \cosh\lambda_{c_1} c_{\text{err}} \tag{51}$$

where

$$h_{\text{err}} = h - h_{\text{ref}} \qquad d_{\text{err}} = d - d_{\text{ref}} \qquad c_{\text{err}} = c - c_{\text{ref}}$$

$$\dot{h}_{\text{err}} = \dot{h} - \dot{h}_{\text{ref}} \qquad \dot{d}_{\text{err}} = \dot{d} - \dot{d}_{\text{ref}} \qquad \dot{c}_{\text{err}} = \dot{c} - \dot{c}_{\text{ref}}$$

and the variables k_h, k_d, and k_c are time-varying tuning parameters. The value A is a positive value for the maximum expected acceleration. The gains $\lambda_{h_1}, \lambda_{d_1}, \lambda_{c_1}, \gamma_h, \gamma_d$, and γ_c are positive and $0 \le \beta_h, \beta_d, \beta_c \le 1$. This formulation of the Lyapunov candidate function was chosen because the $\ell n \cosh x$ terms account for the fact that acceleration limiting will be present. In the formulation of the controller, these terms will yield $\tanh x$ terms that are fundamentally limited to ± 1. The tuning functions k_h, k_d, and k_c add some complexity to the controller, but yield an algorithm that performs very well by balancing the proportional control with the derivative control in a manner that is straightforward to tune.

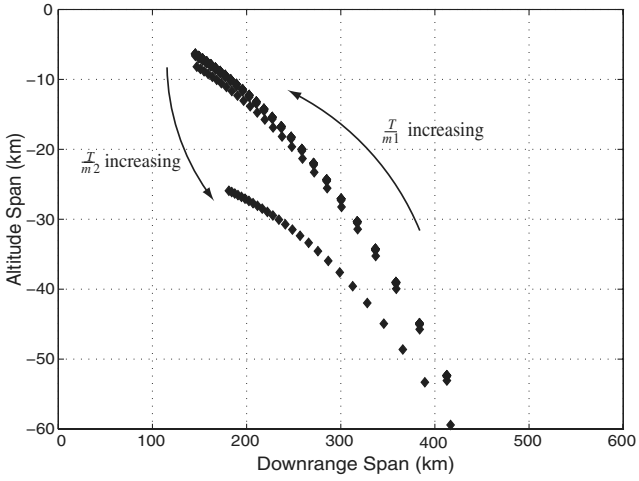
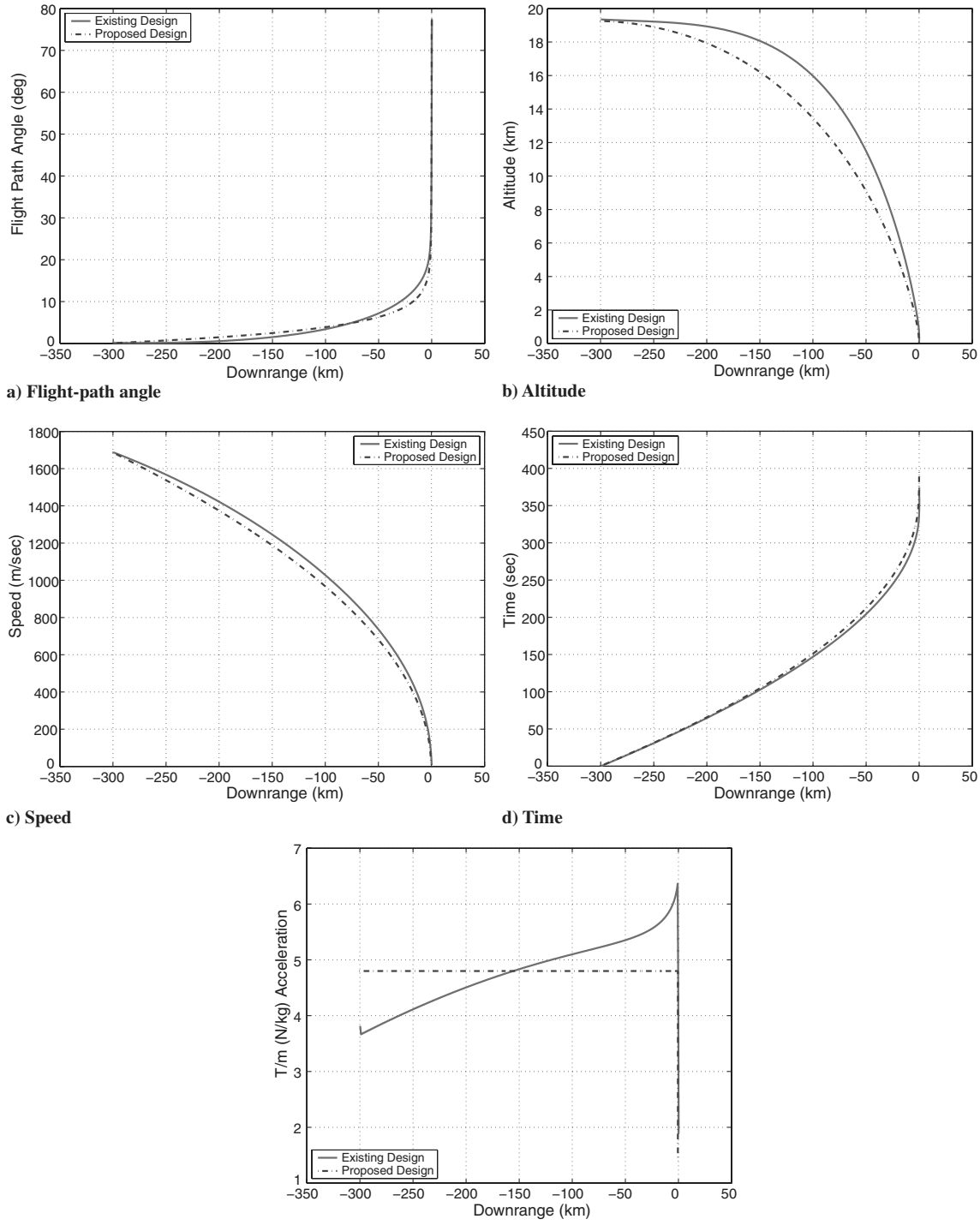


Fig. 2 Sample trajectory space with varying T/m values.

[‡]Private communication with R. Sostaric, March 2006.



a) Flight-path angle **b) Altitude**

c) Speed **d) Time**

e) Thrust acceleration

Fig. 3 Comparison between proposed and existing trajectory.

The real-time guidance operates as a function of time. However, the timeline of the reference trajectory and the real-time timeline can vary significantly. Therefore, the reference trajectory can be thought of as a function of downrange, and the goal of the real-time guidance is to force the states to the reference trajectory regardless of time so that the terminal speed and flight-path angle constraints can be met. The reference states h_{ref}, c_{ref}, v_{ref}, θ_{ref}, and ψ_{ref} are computed by interpolating on each reference trajectory state as a function of downrange; note that if the methodology in Sec. II is used, c_{ref} and ψ_{ref} are zero, although any reference trajectory can be used. If the true states are beyond the initial reference downrange, the initial reference states are used. If the true states are beyond the final reference downrange, the final reference states are used.

Taking the derivative of the Lyapunov candidate function yields

$$\begin{aligned}\dot{V} = {} & \dot{h}_{\text{err}}(\ddot{h}_{\text{err}} + \beta_h A \tanh \lambda_{h_1} h_{\text{err}}) + \frac{1}{\gamma_h} k_h \dot{k}_h \\ & + \dot{d}_{\text{err}}(\ddot{d}_{\text{err}} + \beta_d A \tanh \lambda_{d_1} d_{\text{err}}) + \frac{1}{\gamma_d} k_d \dot{k}_d \\ & + \dot{c}_{\text{err}}(\ddot{c}_{\text{err}} + \beta_c A \tanh \lambda_{c_1} c_{\text{err}}) + \frac{1}{\gamma_c} k_c \dot{k}_c \end{aligned} \tag{52}$$

Assume that the control will make the following true:

$$\ddot{h}_{\text{err}} + \beta_h A \tanh \lambda_{h_1} h_{\text{err}} = -(1-\beta_h) A \tanh \lambda_{h_2} (\dot{h}_{\text{err}} + k_h h_{\text{err}}) \tag{53}$$

$$\ddot{d}_{\text{err}} + \beta_d A \tanh \lambda_{d_1} d_{\text{err}} = -(1-\beta_d) A \tanh \lambda_{d_2} (\dot{d}_{\text{err}} + k_d d_{\text{err}}) \tag{54}$$

$$\ddot{c}_{\text{err}} + \beta_c A \tanh \lambda_{c_1} c_{\text{err}} = -(1-\beta_c) A \tanh \lambda_{c_2} (\dot{c}_{\text{err}} + k_c c_{\text{err}}) \tag{55}$$

where λ_{h_2}, λ_{d_2}, and λ_{c_2} are additional positive control gains and the tuning functions are implemented such that

$$\dot{k}_h = -\gamma_h h_{\text{err}} (1-\beta_h) A \tanh \lambda_{h_2} (\dot{h}_{\text{err}} + k_h h_{\text{err}}) - \gamma_h h_{\text{err}} \tanh k_h h_{\text{err}} \tag{56}$$

$$\dot{k}_d = -\gamma_d d_{\text{err}} (1-\beta_d) A \tanh \lambda_{d_2} (\dot{d}_{\text{err}} + k_d d_{\text{err}}) - \gamma_d d_{\text{err}} \tanh k_d d_{\text{err}} \tag{57}$$

$$\dot{k}_c = -\gamma_c c_{\text{err}} (1-\beta_c) A \tanh \lambda_{c_2} (\dot{c}_{\text{err}} + k_c c_{\text{err}}) - \gamma_c c_{\text{err}} \tanh k_c c_{\text{err}} \tag{58}$$

It follows that Eq. (52) is a negative definite derivative of the Lyapunov candidate function ($\dot{V} \le 0$):

$$\begin{aligned}\dot{V} = &-(\dot{h}_{\text{err}} + k_h h_{\text{err}})(1-\beta_h) A \tanh \lambda_{h_2} (\dot{h}_{\text{err}} + k_h h_{\text{err}}) \\ &- k_h h_{\text{err}} \tanh k_h h_{\text{err}} - (\dot{d}_{\text{err}} + k_d d_{\text{err}})(1-\beta_d) A \tanh \lambda_{d_2} (\dot{d}_{\text{err}} \\ &+ k_d d_{\text{err}}) - k_d d_{\text{err}} \tanh k_d d_{\text{err}} - (\dot{c}_{\text{err}} + k_c c_{\text{err}})(1 \\ &- \beta_c) A \tanh \lambda_{c_2} (\dot{c}_{\text{err}} + k_c c_{\text{err}}) - k_c c_{\text{err}} \tanh k_c c_{\text{err}}\end{aligned} \tag{59}$$

The Lyapunov candidate function V is positive and nonincreasing if the gains λ_{h_1}, λ_{d_1}, λ_{c_1}, λ_{h_2}, λ_{d_2}, λ_{c_2}, γ_h, γ_d, and γ_c are positive and $0 \le \beta_h, \beta_d, \beta_c \le 1$. Therefore, it is upper-bounded and its limit as $t \to \infty$ exists and is finite:

$$\lim_{t\to\infty} V(t) \triangleq V_\infty \tag{60}$$

$$V(t) \in L_\infty \tag{61}$$

If V is bounded, then its components must also be bounded. Because V contains the squared norms of $\dot{h}_{\text{err}}$, $\dot{d}_{\text{err}}$, $\dot{c}_{\text{err}}$, k_h, k_d, and k_d and the function $\ell_n \cosh x$ being bounded implies that x must be bounded,

$$h_{\text{err}}, d_{\text{err}}, c_{\text{err}}, \dot{h}_{\text{err}}, \dot{d}_{\text{err}}, \dot{c}_{\text{err}}, k_h, k_d, k_d \in L_\infty \tag{62}$$

Assuming again that Eqs. (53–55) are true, two of the three terms in each equation are bounded (note $-1 \le \tanh x \le 1$), and so the third term in each must also be bounded, implying

$$\ddot{h}_{\text{err}}, \ddot{d}_{\text{err}}, \ddot{c}_{\text{err}} \in L_\infty \tag{63}$$

Also note that in Eqs. (56–58), the signals on the right-hand side of the equations are bounded, and so

$$\dot{k}_h, \dot{k}_d, \dot{k}_c \in L_\infty \tag{64}$$

Now that all of the signals in the system have been proven to be bounded, the next step is to prove that the error states asymptotically approach zero as a function of time. Because the Lyapunov function is bounded and its limit exists and is finite,

$$\lim_{t\to\infty} \int_0^t \dot{V} d\tau = V_\infty - V_0 \tag{65}$$

Taking the derivative of $\dot{V}$ yields a bounded function. This means that $\dot{V}$ is uniformly continuous [10]. Because $\dot{V}$ is uniformly continuous and its integral exists and is finite, by Barbalet's lemma [10],

$$\lim_{t\to\infty} \dot{V}(t) = 0 \tag{66}$$

With further analysis, this implies

$$\lim_{t\to\infty} \dot{h}_{\text{err}}(t) = 0 \tag{67}$$

$$\lim_{t\to\infty} \dot{d}_{\text{err}}(t) = 0 \tag{68}$$

$$\lim_{t\to\infty} \dot{c}_{\text{err}}(t) = 0 \tag{69}$$

The derivatives of Eqs. (53–55) are bounded, implying

$$\dddot{h}_{\text{err}}, \dddot{d}_{\text{err}}, \dddot{c}_{\text{err}} \in L_\infty \tag{70}$$

The fact that the derivatives of $\ddot{h}_{\text{err}}$, $\ddot{d}_{\text{err}}$, and $\ddot{c}_{\text{err}}$ are bounded means that $\ddot{h}_{\text{err}}$, $\ddot{d}_{\text{err}}$, and $\ddot{c}_{\text{err}}$ are uniformly continuous [10]. The fact that

$$\lim_{t\to\infty} \int_0^t \ddot{h}_{\text{err}}(\tau) d\tau + \dot{h}_{\text{err}}(0) = 0 \tag{71}$$

$$\lim_{t\to\infty} \int_0^t \ddot{d}_{\text{err}}(\tau) d\tau + \dot{d}_{\text{err}}(0) = 0 \tag{72}$$

$$\lim_{t\to\infty} \int_0^t \ddot{c}_{\text{err}}(\tau) d\tau + \dot{c}_{\text{err}}(0) = 0 \tag{73}$$

Table 1 Lunar planetary constants

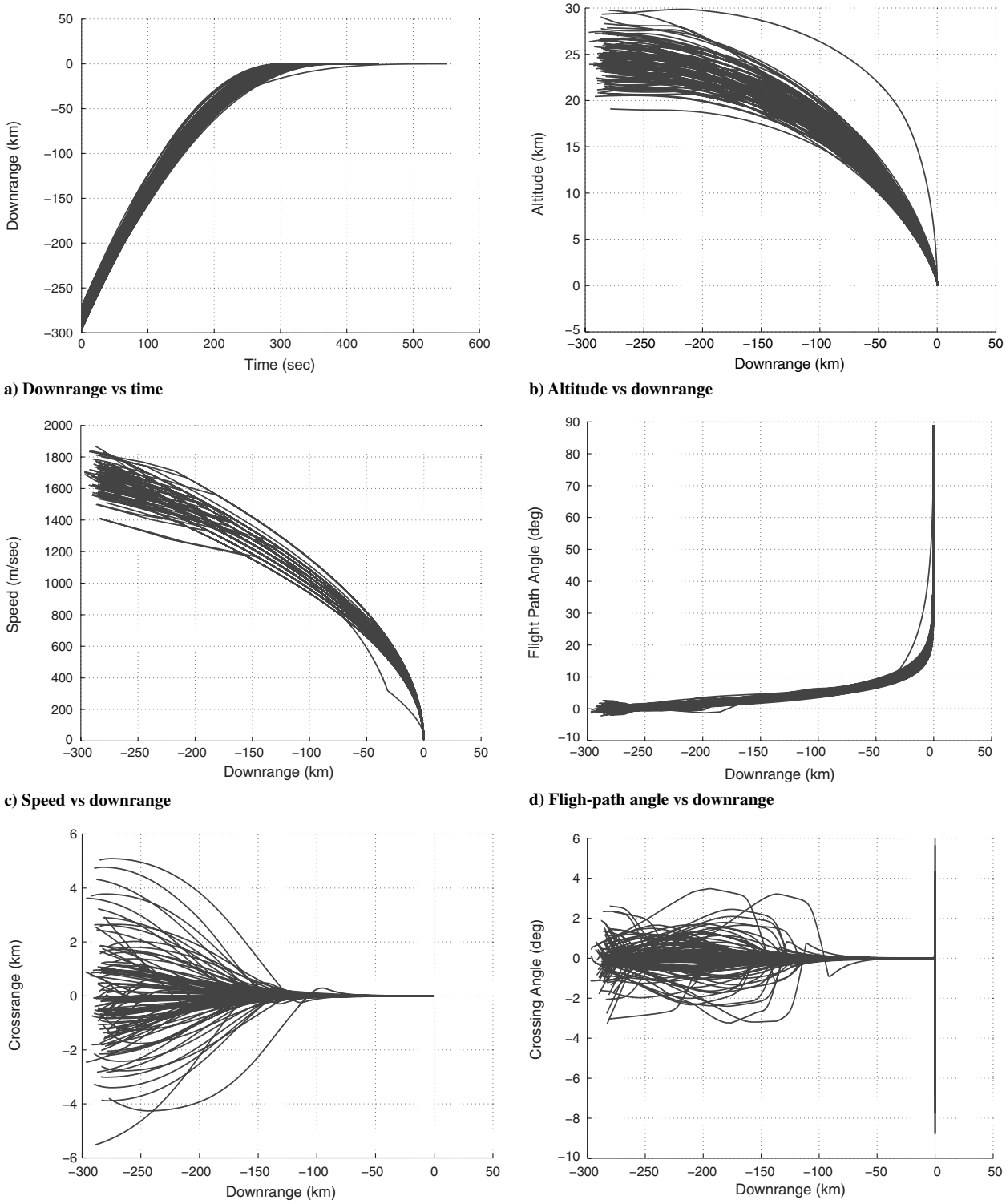
Equatorial radius	1.738×10^6 m
Polar radius	1.735×10^6 m
J_2	202.7×10^{-6}
μ	4902.799×10^9 m^3/s^2
Rotational rate	4.2365×10^{-7} rad/s

Table 2 Real-time guidance gains

Gain	Value	Gain	Value
λ_{h_1}	0.05	β_h	0.45
λ_{d_1}	0.05	β_d	0.45
λ_{c_1}	0.005	β_c	0.45
λ_{h_2}	1.0	γ_h	0.00001
λ_{d_2}	1.0	γ_d	0.00001
λ_{c_2}	1.0	γ_c	0.000001
$k_h(0)$	0.01	$k_c(0)$	0.01
$k_d(0)$	0.01	A	154.1 kN

Table 3 Monte Carlo simulation perturbation values

Perturbation source	Mean value	Standard deviation
Initial downrange	0 km	5 km
Initial altitude	0 km	2 km
Initial crossrange	0 km	2 km
Initial speed	0 m/s	100 m/s
Initial flight-path angle	0 rad	20 mrad
Initial crossing angle	0 rad	20 mrad
Initial mass	0 kg	2000 kg
Mass flow rate	0%	5%
Thrust	0%	5%
Thrust angle α bias	0 rad	30 mrad
Roll angle ϕ bias	0 rad	50 mrad



a) Downrange vs time

b) Altitude vs downrange

c) Speed vs downrange

d) Fligh-path angle vs downrange

e) Crossrange vs downrange

f) Crossing angle vs downrange

Fig. 4 Monte Carlo state history results with run-specific target trajectory.

means that the integrals of $\ddot{h}_{\text{err}}(t)$, $\ddot{d}_{\text{err}}(t)$, and $\ddot{c}_{\text{err}}(t)$ exist and are finite, with the valid assumption that $\dot{h}_{\text{err}}(0)$, $\dot{d}_{\text{err}}(0)$, and $\dot{c}_{\text{err}}(0)$ are finite. This in combination with the fact that the signals $\ddot{h}_{\text{err}}(t)$, $\ddot{d}_{\text{err}}(t)$, and $\ddot{c}_{\text{err}}(t)$ are uniformly continuous means, by Barbalet's lemma [10],

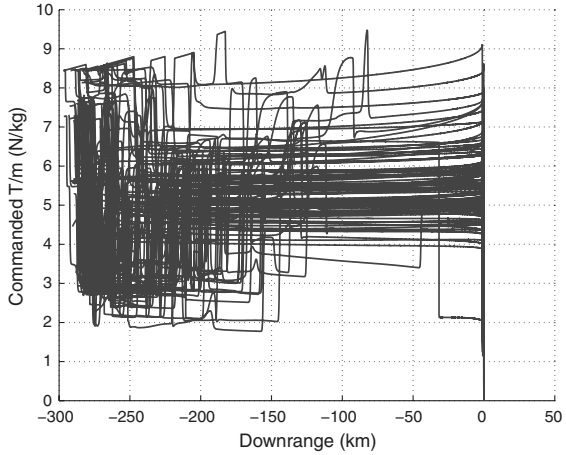
$$\lim_{t\to\infty} \ddot{h}_{\text{err}}(t) = 0 \tag{74}$$

$$\lim_{t\to\infty} \ddot{d}_{\text{err}}(t) = 0 \tag{75}$$

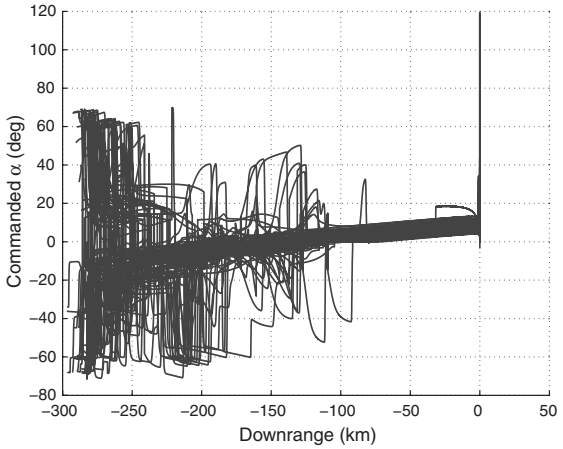
$$\lim_{t\to\infty} \ddot{c}_{\text{err}}(t) = 0 \tag{76}$$

This finally implies

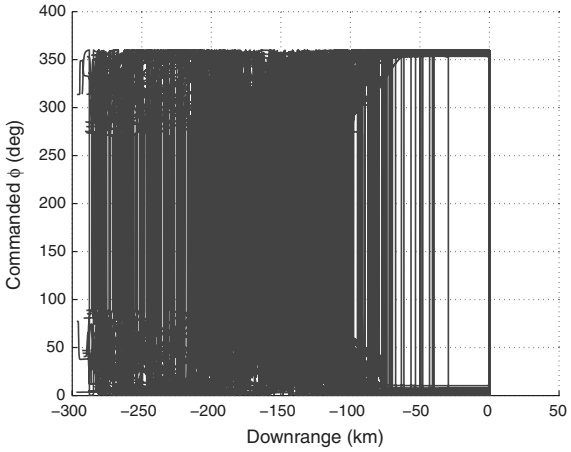
$$\lim_{t\to\infty} h_{\text{err}}(t) = 0 \tag{77}$$



a) Commanded thrust acceleration vs time



b) Commanded thrust angle vs downrange



c) Commanded roll angle vs downrange

Fig. 5 Monte Carlo control history results with run-specific target trajectory.

$$\lim_{t\to\infty} d_{\text{err}}(t) = 0 \tag{78}$$

$$\lim_{t\to\infty} c_{\text{err}}(t) = 0 \tag{79}$$

Because all of the error states and their derivatives converge,

$$h \to h_{\text{ref}} \qquad d \to d_{\text{ref}} \qquad c \to c_{\text{ref}} \tag{80}$$

and

$$v\sin\theta \to v_{\text{ref}}\sin\theta_{\text{ref}} \tag{81}$$

$$\frac{h_S}{h+h_S}v\cos\theta\cos\psi \to \frac{h_S}{h_{\text{ref}}+h_S}v_{\text{ref}}\cos\theta_{\text{ref}}\cos\psi_{\text{ref}} \tag{82}$$

$$\frac{h_S}{h+h_S}v\cos\theta\sin\psi \to \frac{h_S}{h_{\text{ref}}+h_S}v_{\text{ref}}\cos\theta_{\text{ref}}\sin\psi_{\text{ref}} \tag{83}$$

which implies

$$v \to v_{\text{ref}} \qquad \theta \to \theta_{\text{ref}} \qquad \psi \to \psi_{\text{ref}} \tag{84}$$

Now that all states have been proven to converge to their references, Eqs. (53–55) must be used to define the control law. The time derivatives of the fundamental equations of motion from Eqs. (4–6) will be used in Eqs. (53–55) to account for the expected dynamics of the spacecraft:

$$\ddot{h} = -\dot{v}\sin\theta - v\dot{\theta}\cos\theta \tag{85}$$

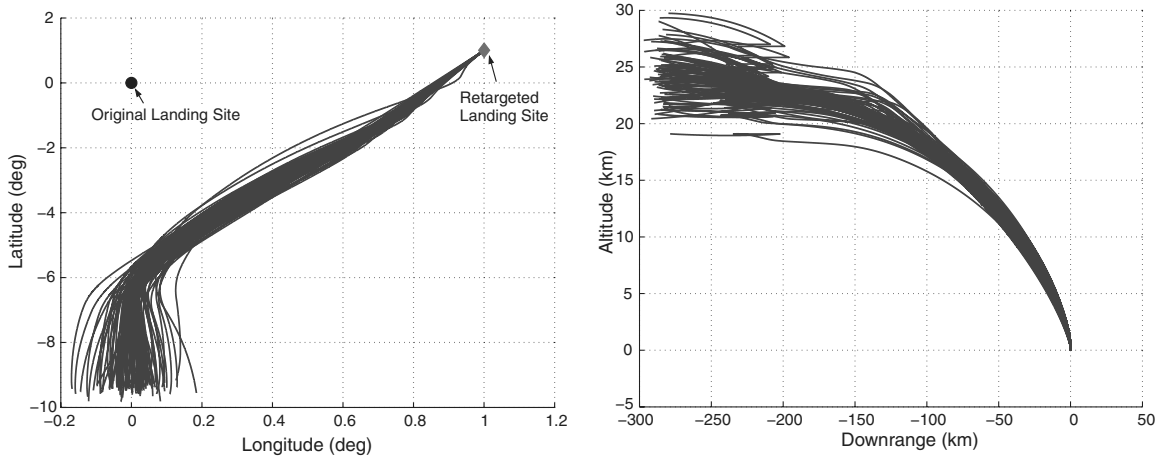
$$\ddot{d} = (\dot{v}\cos\theta\cos\psi - v\dot{\theta}\sin\theta\cos\psi - v\dot{\psi}\cos\theta\sin\psi)\frac{h_S}{h+h_S} - v\cos\theta\cos\psi\frac{h_S\dot{h}}{(h+h_S)^2} \tag{86}$$

$$\ddot{c} = (\dot{v}\cos\theta\sin\psi - v\dot{\theta}\sin\theta\sin\psi + v\dot{\psi}\cos\theta\cos\psi)\frac{h_S}{h+h_S} - v\cos\theta\sin\psi\frac{h_S\dot{h}}{(h+h_S)^2} \tag{87}$$

Rearranging terms in Eqs. (53–55) and combining these with Eqs. (85–87) yields

Table 4 Monte Carlo simulation final state error statistics for run-specific targeting

Error	Target value	Mean value	Standard deviation
Time	Varies	400.59 s	29.48 s
Downrange	0 m	−0.02 m	0.05 m
Crossrange	0 m	0.01 m	0.32 m
Speed	8 m/s	8.04 m/s	0.38 m/s
Fligh- path angle	89 deg	89.01 deg	0.02 deg
Crossing angle	0 deg	−0.16 deg	2.23 deg



a) Latitude vs longitude **b) Altitude vs downrange**

Fig. 6 Retargeting Monte Carlo results (northeast example).

$$-\dot{v}\sin\theta - v\dot{\theta}\cos\theta = -\beta_h A\tanh\lambda_{h_1}h_{\text{err}} - (1-\beta_h)A\tanh\lambda_{h_2}(\dot{h}_{\text{err}} + k_h h_{\text{err}}) + \ddot{h}_{\text{ref}} \tag{88}$$

$$\left(\dot{v}\cos\theta\cos\psi - v\dot{\theta}\sin\theta\cos\psi - v\dot{\psi}\cos\theta\sin\psi - \frac{v\cos\theta\cos\psi\dot{h}}{h+h_S}\right)\times\frac{h_S}{h+h_S} = -\beta_d A\tanh\lambda_{d_1}d_{\text{err}} - (1-\beta_d)A\tanh\lambda_{d_2}\times(\dot{d}_{\text{err}} + k_d d_{\text{err}}) + \ddot{d}_{\text{ref}} \tag{89}$$

$$\left(\dot{v}\cos\theta\sin\psi - v\dot{\theta}\sin\theta\sin\psi + v\dot{\psi}\cos\theta\cos\psi - \frac{v\cos\theta\sin\psi\dot{h}}{h+h_S}\right)\times\frac{h_S}{h+h_S} = -\beta_c A\tanh\lambda_{c_1}c_{\text{err}} - (1-\beta_c)A\tanh\lambda_{c_2}\times(\dot{c}_{\text{err}} + k_c c_{\text{err}}) + \ddot{c}_{\text{ref}} \tag{90}$$

If the fundamental equations of motion for the dynamic states from Eqs. (1–3) are substituted into the previous equations, the resulting equation that must be solved to compute the control variables is

$$\begin{bmatrix} \sin\theta & \cos\theta & 0 \\ -\cos\theta\cos\psi & \sin\theta\cos\psi & \sin\psi \\ -\cos\theta\sin\psi & \sin\theta\sin\psi & -\cos\psi \end{bmatrix}\begin{bmatrix} \frac{T}{m}\cos\alpha \\ \frac{T}{m}\sin\alpha\cos\phi \\ \frac{T}{m}\sin\alpha\sin\phi \end{bmatrix} = \begin{bmatrix} H \\ D \\ C \end{bmatrix} \tag{91}$$

where

$$H = \ddot{h}_{\text{ref}} + g - \frac{v^2}{h+h_S}\cos^2\theta - \beta_h A\tanh\lambda_{h_1}h_{\text{err}} - (1-\beta_h)A\tanh\lambda_{h_2}(\dot{h}_{\text{err}} + k_h h_{\text{err}}) \tag{92}$$

$$D = \ddot{d}_{\text{ref}}\frac{h+h_S}{h_S} - \frac{2v^2}{h+h_S}\cos\theta\sin\theta\cos\psi - (\beta_d A\tanh\lambda_{d_1}d_{\text{err}} + (1-\beta_d)A\tanh\lambda_{d_2}(\dot{d}_{\text{err}} + k_d d_{\text{err}}))\frac{h+h_S}{h_S} \tag{93}$$

$$C = \ddot{c}_{\text{ref}}\frac{h+h_S}{h_S} - \frac{2v^2}{h+h_S}\cos\theta\sin\theta\sin\psi - (\beta_c A\tanh\lambda_{c_1}c_{\text{err}} + (1-\beta_c)A\tanh\lambda_{c_2}(\dot{c}_{\text{err}} + k_c c_{\text{err}}))\frac{h+h_S}{h_S} \tag{94}$$

This can be inverted so that if the control variables T/m, α, and ϕ are given by

$$\begin{bmatrix} \frac{T}{m}\cos\alpha \\ \frac{T}{m}\sin\alpha\cos\phi \\ \frac{T}{m}\sin\alpha\sin\phi \end{bmatrix} = \begin{bmatrix} \sin\theta & -\cos\theta\cos\psi & -\cos\theta\sin\psi \\ \cos\theta & \sin\theta\cos\psi & \sin\theta\sin\psi \\ 0 & \sin\psi & -\cos\psi \end{bmatrix}\begin{bmatrix} H \\ D \\ C \end{bmatrix} \tag{95}$$

all states are guaranteed to converge to their reference trajectories in an infinite amount of time.

Equation (95) is the basis for the guidance algorithm, along with the numerical integration of Eqs. (56–58). Equation (95) can be rewritten in the form

$$\begin{bmatrix} \frac{T}{m}\cos\alpha \\ \frac{T}{m}\sin\alpha\cos\phi \\ \frac{T}{m}\sin\alpha\sin\phi \end{bmatrix} = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} \tag{96}$$

The solution for the control variables is

$$\frac{T}{m} = \sqrt{n_1^2 + n_2^2 + n_3^2} \tag{97}$$

$$\alpha = \tan^{-1}\left(\frac{\sqrt{n_2^2 + n_3^2}}{n_1}\right) \tag{98}$$

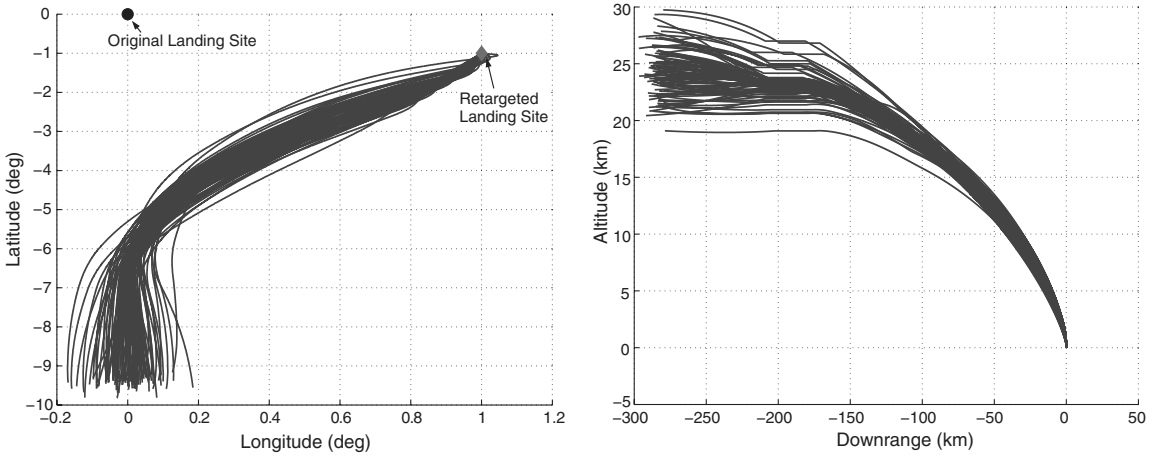
$$\phi = \tan^{-1}\left(\frac{n_3}{n_2}\right) \tag{99}$$

IV. Algorithm Simulation

Any valid guidance law should work under ideal conditions. However, it must be tested given a realistic setting to verify its usability in a real application. A three-degree-of-freedom simulation was developed to simulate the translational dynamics of the vehicle.

Table 5 Monte Carlo simulation final state error statistics for northeast retargeting

Error	Target value	Mean value	Standard deviation
Time	Varies	431.94 s	23.61 s
Downrange	0 m	−0.02 m	0.04 m
Crossrange	0 m	0.00 m	0.03 m
Speed	8 m/s	8.01 m/s	0.38 m/s
Flight-path angle	89 deg	89.01 deg	0.02 deg
Crossing angle	0 deg	−0.05 deg	0.68 deg



a) Latitude vs longitude **b) Altitude vs downrange**

Fig. 7 Retargeting Monte Carlo results (southeast example).

The full translational equations of motion are used as the basis, with no approximations for a flat surface. The gravity model is that for an oblate spheroid and is nonconstant. Planetary constants used for the moon are given in Table 1. Mass is modeled as linearly time-varying as a function of some computed mass flow rate; nominal mass is 31,624.0 kg and nominal specific impulse is 459.7 s (see footnote [‡]). The real-time guidance algorithm resides in this framework with the gains in Table 2.

Monte Carlo analysis can be performed in two different manners. The first is to generate a reference trajectory at the beginning of a set of sample runs based on expected states and use that for every perturbation of the states. The second, and arguably more desirable, manner is to generate a reference trajectory at the beginning of any simulated run based on the initial states for that sample. The results of the second methodology are presented here; additionally, the results of the first methodology are presented in [9]. The algorithm is that described in Sec. II.B. The two-segment target trajectory is used with Eqs. (47–50), creating the altitude-vs-downrange solution space. For this analysis, the target trajectory that matched the altitude span closest to the current altitude was selected; this means that the vehicle would not have to accelerate upward or dive downward to catch the reference, whereas the downrange would eventually catch up with the reference, with little impact on the response in the interim. Constraints were added that $\theta_0 \leq \theta_1 \leq \theta_2$ and that $v_2 \leq v_1 \leq v_0$ to keep the trajectories in a more desirable space. An additional requirement was levied that the initial flight-path angle θ_0 used to define the target trajectory was positive to avoid creating target trajectories that initially increase in altitude.

The perturbations of Table 3 were introduced and 100 individual runs were performed to illustrate the integrated performance of the targeting algorithm and the real-time guidance algorithm. The state histories are shown in Figs. 4a–4f, and the control histories are shown in Figs. 5a–5c. The terminal states are summarized in Table 4 and illustrate that the real-time guidance performs well.

Note the single case in all of the plots that appears to be an outlier. This is simply a situation in which the target trajectory selected is different in character from all of the other runs; the targeting and real-time guidance algorithms still perform well. This run does illustrate the need for the user of this algorithm to think clearly about all states that should be constrained, because this run does have a long final trajectory time compared with the other runs (about 100 s longer). If time is a major constraint (which it could easily be, due to fuel constraints), then balancing the different trajectory options is important and would have to be considered in a specific targeting algorithm.

Assessing the control variables in Figs. 5a–5c, the commanded thrust angle stays nicely constrained between ± 70 deg. Though this might appear large in magnitude, note that the angles are relatively smooth, meaning that the larger thrust angles are required to force the vehicle onto the reference trajectory. Once the vehicle is safely on the reference trajectory, the commanded thrust angle approaches zero. Some runs exhibit large commanded thrust angles at the end; these are cases in which the vehicle has passed the targeted landing point and the guidance is attempting to turn the vehicle back toward the target. The nonconstant nature of both the commanded thrust acceleration and the thrust angle when the guidance errors have been largely eliminated illustrates how the real-time guidance absorbs the errors induced by modeling the thrust acceleration, gravity, and centrifugal acceleration as constants.

At some point in a trajectory, the decision may be made that the current landing site is no longer desirable. This would require the landing site to be relocated and all of the guidance states translated to be with respect to that new landing site; a resultant change in kinematic states would thereby have to be handled by the real-time guidance (the downrange and crossrange to the new target location will likely be considerably different from that to the old location). Additionally, a new target trajectory may be necessary, which will again require a response by the real-time guidance.

The retargeting methodology chosen here was to specify a new landing-site location that was converted to latitude and longitude. From there, a new target trajectory bearing was computed assuming that the shortest distance between the current vehicle location and the new desired landing location defines the new downrange. With this new bearing, the new downrange, altitude, and crossrange to the desired landing site is computed as before. With the vehicle states now defined with respect to the new landing site, the new target trajectory is defined by using those states as the basis. With the new bearing defined, the vehicle states adjusted, and the new target trajectory selected, the real-time guidance algorithm continues as before with no changes.

Two examples of retargeting are presented. The first example moves the target location 1 deg north and 1 deg east and the second moves the target location 1 deg south and 1 deg east. These illustrate the impact of moving the target location both further from and closer to the current vehicle location. In both cases, the retargeting occurs 50 s into the event.

Monte Carlo runs were performed against the two retargeting cases. Figures 6a and 6b illustrate the performance against the northeast case, with the results summarized in Table 5. Figures 7a

Table 6 Monte Carlo simulation final state error statistics for southeast retargeting

Error	Target value	Mean value	Standard deviation
Time	Varies	384.14 s	23.10 s
Downrange	0 m	−0.02 m	0.04 m
Crossrange	0 m	−0.40 m	1.37 m
Speed	8 m/s	8.02 m/s	0.36 m/s
Flight-path angle	89 deg	88.92 deg	0.47 deg
Crossing angle	0 deg	6.32 deg	14.56 deg

and 7b illustrate the performance against the southeast case, with the results summarized in Table 6. As these figures illustrate, the retargeting algorithm and the real-time guidance algorithm work well in concert.

V. Conclusions

The Apollo lunar descent guidance worked well for the goal of taking men safely to the moon. However, as the goal of lunar exploration has changed to encompass a desire to easily and cheaply explore many locations on the moon, new algorithms that allow for ease of change in landing sites is necessary. This document details a viable option for that algorithm.

The Apollo lunar descent guidance algorithm consists of two elements: the targeting algorithm and the real-time guidance algorithm. The proposed algorithm developed herein followed that format to allow for the separate use of the targeting and real-time guidance algorithms. The targeting algorithm can be used to develop an analytical reference trajectory that can then be used as the basis for the real-time guidance. Optionally, a reference trajectory from another development model can be formatted in terms of downrange, altitude, speed, and flight-path angle and used in the real-time guidance algorithm with some guarantees of convergence as long as the thrust magnitude stays below the maximum allowable thrust.

References

[1] "Guidance System Operations Plan for Manned LM Earth Orbital and Lunar Missions Using Program Luminary 1D," Massachusetts Inst. of Technology, Charles Stark Draper Lab., TR R-567, Rev. 8, Sec. 5.3.4, Cambridge, MA, Mar. 1970, pp. 5.3-47–5.3-121a.

[2] 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.

[3] Cheng, R. K.,"Lunar Terminal Guidance," *Lunar Missions and Exploration*, edited by C. T. Leondes, and R. W. Vance, University of California Engineering and Physical Sciences Extension Series, Wiley, New York, 1964, pp. 308–355.

[4] Citron, S. J., Dunin, S. E., and Messinger, H. F., "A Terminal Guidance Technique for Lunar Landing," *AIAA Journal*, Vol. 2, No. 3, 1964, pp. 503–509.
doi:10.2514/3.2362

[5] Cheng, R. K., Meredith, C. M., and Conrad, D. A., "Design Considerations for Surveyor Guidance," *Journal of Spacecraft and Rockets*, Vol. 3, No. 11, 1966, pp. 1569–1576.
doi:10.2514/3.28709

[6] McInnes, C. R., "Gravity-Turn Descent from Low Circular Orbit Conditions," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 1, Jan.–Feb. 2003, pp. 183–185.
doi:10.2514/2.5033

[7] McInnes, C. R., "Direct Adaptive Control for Gravity-Turn Descent," *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 2, Mar.–Apr. 1999, pp. 373–375.
doi:10.2514/2.4392

[8] Uchiyama, K., Shimada, Y., and Nakajima, S., "Tracking Control to Near-Optimal Trajectory for a Lunar Lander," *Proceedings of the 23rd International Symposium on Space Technology and Science*, Vol. 1, Japan Society for Aeronautical and Space Sciences, Tokyo, May 2002, pp. 977–982.

[9] Chomel, C., "Development of an Analytical Guidance Algorithm for Lunar Descent," Ph.D. Dissertation, Dept. of Aerospace Engineering and Engineering Mechanics, Univ. of Texas at Austin, Austin, TX, 2007.

[10] Sastry, S. and Bodson, M., *Adaptive Control: Stability, Convergence, and Robustness*, Prentice–Hall, New York, 1989, p. 19.