

Guidance, Navigation, and Control System Performance Trades for Mars Pinpoint Landing

Bradley A. Steinfeldt,* Michael J. Grant,* Daniel A. Matz,† and Robert D. Braun‡

Georgia Institute of Technology, Atlanta, Georgia 30332

and

Gregg H. Barton§

Charles Stark Draper Laboratory, Inc., Houston, Texas 77058

DOI: 10.2514/1.45779

Landing site selection is a compromise between safety concerns associated with the site's terrain and scientific interest. Therefore, technologies enabling pinpoint landing performance (sub-100-m accuracies) on the surface of Mars are of interest to increase the number of accessible sites for in situ research, as well as allow placement of vehicles nearby prepositioned assets. A survey of the performance of guidance, navigation, and control technologies that could allow pinpoint landing to occur at Mars was performed. This assessment has shown that negligible propellant mass fraction benefits are seen for reducing the three-sigma position dispersion at the end of the hypersonic guidance phase (parachute deployment) below approximately 3 km. Four different propulsive terminal descent guidance algorithms were examined. Of these four, a near propellant-optimal analytic guidance law showed promise for the conceptual design of pinpoint landing vehicles. The existence of a propellant optimum with regard to the initiation time of the propulsive terminal descent was shown to exist for various flight conditions. Subsonic guided parachutes were shown to provide marginal performance benefits, due to the timeline associated with descent through the thin Mars atmosphere. This investigation also demonstrates that navigation is a limiting technology for Mars pinpoint landing, with landed performance being largely driven by navigation sensor and map tie accuracy.

Nomenclature

\mathbf{a}	= acceleration vector, $[a_1 \ a_2 \ a_3]^T$
a_i	= acceleration along the i th direction
b	= scalar weighting parameter on the terminal constraint sensitivity
C_{ji}	= j th constant coefficient used in the modified Apollo lunar module guidance algorithm
dt_f	= terminal time increment
\mathbf{f}	= set of first-order differential equations of motion
g	= local acceleration due to gravity
\mathbf{g}	= acceleration vector due to gravity
i	= index
I_{JJ}	= partition used in the optimal control solution
$I_{J\psi}$	= partition used in the optimal control solution
$I_{\psi\psi}$	= partition used in the optimal control solution
$I_{\psi\psi}$	= partition used in the optimal control solution
J	= performance index
Kn	= Knudsen number
L	= scalar objective function describing path parameters
M	= Mach number
m_{prop}	= mass of propellant
m_0	= initial mass of the vehicle

\mathbf{p}	= influence function vector
R	= matrix of influence functions
\mathbf{r}	= position vector, $[r_1 \ r_2 \ r_3]^T$
r_i	= position along the i th direction
S_j	= matrix defining convex state constraints
t	= time
t_{go}	= time to go until touchdown
\mathbf{u}	= control vector
\mathbf{v}	= velocity vector, $[v_1 \ v_2 \ v_3]^T$
v_i	= velocity along the i th direction
W	= positive definite weighting matrix
\mathbf{x}	= state vector, $[\mathbf{r}^T \ \mathbf{v}^T \ m]^T$
α	= mass consumption rate
β_j	= scalar defining convex state constraints
Γ	= weighting on final time to go
$\delta\mathbf{u}$	= control vector increment
ε	= tolerance level
ζ	= slack variable bounding thrust magnitude
ρ_1	= thrust magnitude lower bound
ρ_2	= thrust magnitude upper bound
τ	= dust tau (opacity measure of the atmosphere)
$\boldsymbol{\tau}_c$	= commanded thrust vector
\mathbf{v}_j	= vector defining convex state constraints
ϕ	= scalar objective function for terminal state constraints
ψ	= adjoint constraint equations

Presented as Paper 6216 at the AIAA Atmospheric Flight Mechanics Conference and Exhibit, Honolulu, HI, 18–21 August 2008; received 2 June 2009; revision received 19 October 2009; accepted for publication 2 November 2009. Copyright © 2009 by Bradley A. Steinfeldt and the Charles Stark Draper Laboratory. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, on condition that the copier pay the \$10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0022-4650/10 and \$10.00 in correspondence with the CCC.

*Graduate Research Assistant, Guggenheim School of Aerospace Engineering. Student Member AIAA.

†Undergraduate Research Assistant, Guggenheim School of Aerospace Engineering. Student Member AIAA.

‡David and Andrew Lewis Associate Professor of Space Technology, Guggenheim School of Aerospace Engineering. Fellow AIAA.

§Group Leader, Mission Design and Analysis Group. Senior Member AIAA.

I. Introduction

AT PRESENT, the choice of landing sites for Mars exploration vehicles is a trade between scientific interest and landing safety, in which the safety element may preclude many interesting regions of the planet. The landed accuracy of an entry system is a function of four major items: delivery error at entry interface (EI), knowledge uncertainty at EI, environmental uncertainty, and vehicle performance [1]. Delivery error at EI refers to how closely the vehicle's actual position and velocity vector at EI match the desired EI position and velocity vectors, and it is driven primarily by interplanetary navigation and how accurately trajectory correction maneuvers (TCMs) are performed. Knowledge uncertainty at EI is a result of accumulated sensor error from the last navigational update, as well as the

propagated accuracy of that navigation update. Environmental uncertainty consists primarily of atmospheric deviations from the nominal density and wind profiles through the atmosphere, although other sources, such as gravitational field modeling, impact this uncertainty as well. The dispersions associated with the performance of the vehicle are composed of uncertainties in the physical model of the entry system (mass properties, aerodynamic characteristics, etc.) and how its systems perform (deployment events associated with the parachute; performance of the guidance, navigation, and control (GNC) systems; thrust and duration of burns, etc.). The landed ellipse, as shown in Fig. 1, is the cumulative effect of these uncertainties, propagated throughout the vehicle's entire trajectory, mapped to a physical location on the surface of the destination planet. The Mars Science Laboratory (MSL), planned to launch in 2011, is anticipating a landing ellipse major axis of approximately 20 km, which is a four-time reduction from the Mars Exploration Rovers (MER) and over an order-of-magnitude improvement from the Mars Pathfinder mission [2]. Relative to the MER, this landed ellipse accuracy improvement is largely the result of the inclusion of a modified Apollo hypersonic guidance algorithm that modulates the direction of the vehicle's lift vector to accommodate uncertainties in the atmospheric flight path [3]. Pinpoint landing accuracy is defined as a reduction in the landing ellipse major axis to sub-100-m levels. By achieving this level of accuracy, a number of benefits can be realized, such as minimizing rover traverse times to scientifically-rich locations and enabling entry systems to land near prepositioned assets on the surface, as was outlined in the Mars Design Reference Mission for human exploration or suggested for robotic sample return missions [4].

Whereas previous investigations, such as those in [1,5], have examined technologies required to achieve pinpoint landing independent of the propulsive terminal descent guidance algorithm, this investigation couples the terminal descent guidance algorithm choice with a common baseline vehicle and dispersion set to comprehensively examine the GNC technology performance necessary to achieve pinpoint landing on Mars. Section II discusses the simulation and dispersion environment assumed for this investigation. A discussion on terminal descent guidance algorithms is located in Sec. III, in which various propulsive terminal descent algorithms and their performances are examined. Sections IV and V discuss the impact of hypersonic guidance and subsonic guidance in the form of a steerable parachute, respectively. Finally, Sec. VI investigates the impact of navigation [particularly, terrain relative

navigation (TRN)] on the landed accuracy of the entry, descent, and landing (EDL) system at Mars.

II. Simulation and Vehicle Parameters

Various GNC technologies are studied on a large-scale robotic entry vehicle in order to understand their implications on the capability to achieve sub-100-m-level landed accuracies. The technologies investigated span the entire EDL sequence: from the hypersonic phase through terminal descent and touchdown. Hypersonic, subsonic parachute, and propulsive terminal descent guidance is investigated using ideal navigation and identifying the propellant mass fraction (PMF) required to achieve pinpoint level accuracy. The PMF is defined as the ratio of the propellant's mass to the initial mass:

$$\text{PMF} = \frac{m_{\text{prop}}}{m_0} \quad (1)$$

This is a surrogate for the entry system's mass at EI (and hence the system's mass at launch). TRN was also investigated by examining the effect of sensor termination altitude and sampling frequency on TRN sensor performance, as well as examining the effect of map tie error (i.e., the error associated with relating physical features on the surface of the planet to inertial space) on the overall landed vehicle's accuracy.

A three-degree-of-freedom simulator with bank modulation was used for trajectory propagation. The simulator incorporates modular capability, allowing easy incorporation of the various guidance algorithms assessed. For the entirety of this study, a Viking heritage entry vehicle geometry was assumed, consisting of a 4.5 m, 70 deg sphere-cone aeroshell. For the terminal descent studies, the nominal trajectory starts at a parachute deployment point with an altitude of 8 km above the Mars Orbiter Laser Altimeter reference ellipsoid and Mach 2 with a flight-path angle of -20 deg. The dispersion on this initial condition was assumed to be consistent with that of the MSL, implying that a modified Apollo guidance algorithm is used throughout the hypersonic phase of flight, except a constant parachute deployment altitude is assumed [6]. The Mars Global Reference Atmosphere Model was interrogated at a single latitude and longitude corresponding to the nominal landing site, with dust tau varying between 0.1 and 0.9 to provide the mean and variation for the various environmental parameters used throughout the trajectory, including the wind, the acceleration due to gravity, and the density [7]. Figure 2 shows a plot of the nominal density variation used in the

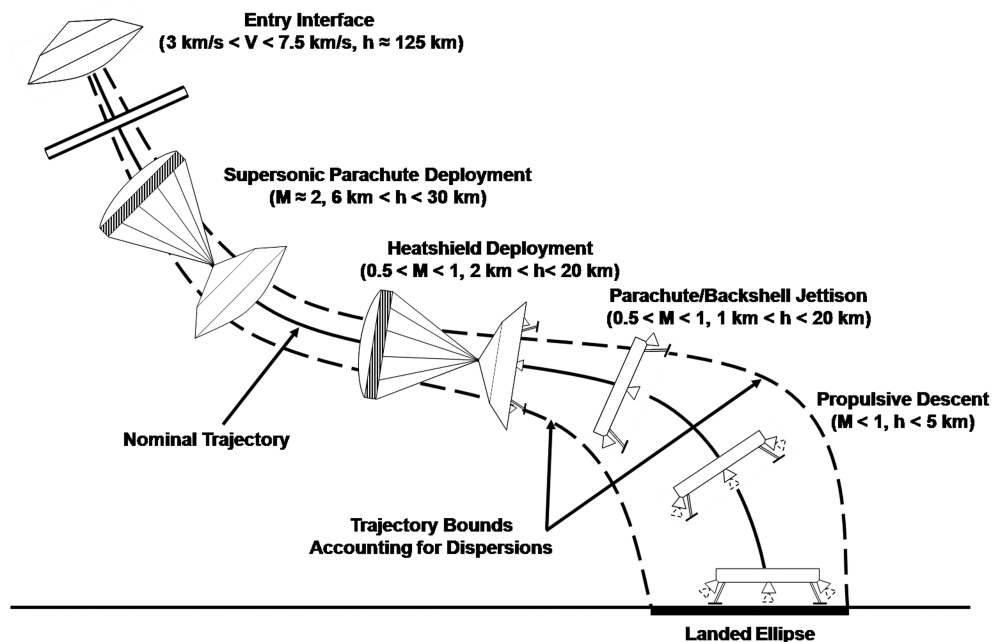


Fig. 1 Typical Martian EDL sequence with uncertainty.

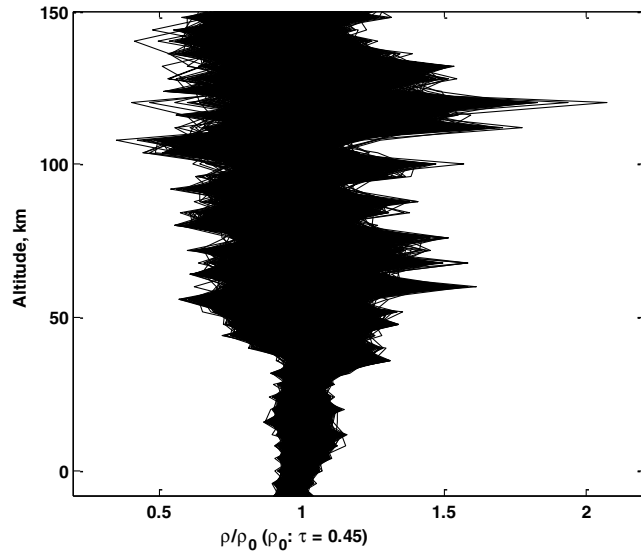


Fig. 2 Density variation used in simulations.

simulations. Nominal vehicle, state, and environmental parameters with their dispersions are shown in Table 1. The entry state and hypersonic parameters were used by Striepe et al. to derive the parachute deployment dispersion used for the initial conditions for the principal trades conducted [6]. Note that the parachute deploy conditions (e.g., the altitude, velocity, and flight-path angle) are used as the initial state for the terminal descent study, whereas the navigation studies used a propagated covariance matrix for MSL, obtained from the Jet Propulsion Laboratory 10 min before EI, assuming the fifth TCM was performed.

III. Propulsive Terminal Descent

Four different propulsive terminal descent algorithms were evaluated in this investigation. The first of which is a modified Apollo lunar module terminal descent algorithm that assumes linear variation of the vertical acceleration, with quadratic variation in the remaining two axes and no optimality conditions [8]. The second algorithm considered is a constrained gradient-based, indirect optimal control algorithm, with iteration required to derive the control history [9]. The third algorithm, originally derived by D'Souza, is an energy-optimal algorithm that assumes flight over a flat planet,

neglecting aerodynamic forces. These assumptions allow an analytic solution to be found, which has been shown to be optimal for the simplified problem [10]. The fourth algorithm examined is a second-order cone formulation, in which convexity ensures that a global optimum is reached in a finite number of iterations with a feasible result obtained at each iteration, which is desirable should the algorithm be implemented onboard the vehicle [11].

A. Modified Apollo Lunar Module Terminal Guidance Algorithm

The modified Apollo lunar module guidance algorithm begins by assuming that the acceleration profile is quadratic in each of the three directions (downrange, cross range, and altitude) relative to the target [8]. In equation form, that is to say, acceleration in each direction is given by

$$a_i(t) = C_{0i} + C_{1i}t + C_{2i}t^2; \quad i = 1, 2, 3 \quad (2)$$

This can be integrated to give the velocity and distance variation with time in each axis:

$$v_i(t) = C_{0i}t + \frac{C_{1i}}{2}t^2 + \frac{C_{2i}}{3}t^3 + v_{0i}; \quad i = 1, 2, 3 \quad (3)$$

$$r_i(t) = \frac{C_{0i}}{2}t^2 + \frac{C_{1i}}{6}t^3 + \frac{C_{2i}}{12}t^4 + v_{0i}t + r_{0i}; \quad i = 1, 2, 3 \quad (4)$$

Evaluating Eqs. (2–4) at the initial conditions,

$$\mathbf{r}(t=0) = \mathbf{r}_0 \quad (5a)$$

and

$$\mathbf{v}(t=0) = \mathbf{v}_0 \quad (5b)$$

and final conditions

$$\mathbf{r}(t=t_f) = \mathbf{r}_f \quad (6a)$$

$$\mathbf{v}(t=t_f) = \mathbf{v}_f \quad (6b)$$

and

$$\mathbf{a}(t=t_f) = \mathbf{a}_f \quad (6c)$$

allow the solutions for the coefficients in each axis to be solved. The resulting coefficients are given by

Table 1 Vehicle and state parameters [1,6]

Parameter	Nominal	Distribution	Deviation, 3- σ or min / max
Entry mass	2616 kg	Gaussian	± 3 kg
Vehicle diameter	4.5 m	—	—
Trim angle of attack	11 deg	Gaussian	± 2.0 deg
Parachute deploy distance from nominal	0 km	Uniform	8 km
Parachute deploy altitude	8 km	—	—
Parachute deploy velocity	488 m/s	Gaussian	± 1.3 m/s
Parachute deploy flight-path angle	-20.0 deg	Uniform	$-19.8 / -20.2$ deg
C_A multiplier, $Kn \leq 0.1$	1.0	Gaussian	5%
C_N multiplier, $Kn \leq 0.1$	1.0	Gaussian	10%
C_A Multiplier ($M > 10$)	1.0	Gaussian	3%
C_N multiplier, $M > 10$	1.0	Gaussian	5%
C_A multiplier, $0.8 < M \leq 10$	1.0	Gaussian	10%
C_N multiplier, $0.8 < M \leq 10$	1.0	Gaussian	8%
C_A multiplier, $0.8 \leq M$	1.0	Gaussian	5%
Supersonic parachute diameter	19.0 m	—	—
Supersonic parachute C_D	0.61	Uniform	0.55/0.67
Subsonic parachute diameter ^a	19.0 m	—	—
Subsonic parachute C_D ^a	0.68	Uniform	0.61/0.75
Maximum terminal descent engine thrust	3047 N	Uniform	2895/3199 N
Minimum terminal descent engine thrust ^b	1142 N	Uniform	1085/1194 N
Terminal descent engine I_{sp}	220 s	Uniform	218/222 s

^aFor subsonic guided parachute only

^bFor second-order cone algorithm only

$$C_{0_i} = a_{f_i} - \frac{6}{t_{go}}(v_{f_i} + v_{0_i}) + \frac{12}{t_{go}^2}(r_{f_i} - r_{0_i}); \quad i = 1, 2, 3 \quad (7)$$

$$C_{1_i} = -\frac{6}{t_{go}}a_{f_i} + \frac{6}{t_{go}^2}(5v_{f_i} + 3v_{0_i}) - \frac{48}{t_{go}^3}(r_{f_i} - r_{0_i}); \quad i = 1, 2, 3 \quad (8)$$

$$C_{2_i} = \frac{6}{t_{go}^2}a_{f_i} - \frac{12}{t_{go}^3}(2v_{f_i} + v_{0_i}) + \frac{36}{t_{go}^4}(r_{f_i} - r_{0_i}); \quad i = 1, 2, 3 \quad (9)$$

By assuming a linear acceleration profile in the vertical axis (i.e., setting $C_{2_3} = 0$), the time to go t_{go} can be solved for analytically and is given by the expression

$$t_{go} = \frac{2v_{f_3} + v_{0_3}}{a_{f_3}} + \left[\left(\frac{2v_{f_3} + v_{0_3}}{a_{f_3}} \right)^2 + \frac{6}{a_{f_3}}(r_{0_3} - r_{f_3}) \right]^{1/2} \quad (10)$$

$a_{f_3} \neq 0$

or

$$t_{go} = 3 \frac{r_{f_3} - r_{0_3}}{2v_{f_3} + v_{0_3}}; \quad a_{f_3} = 0 \quad (11)$$

Thus, the commanded thrust vector is given by

$$\boldsymbol{\tau}_c = m(\mathbf{a} - \mathbf{g}) \quad (12)$$

The primary advantage of this algorithm is that it is computationally simple and allows for the acceleration profile to be found for all times. However, the algorithm does not provide for conditions to obtain the fuel-optimal solutions or constraints on the maximum commanded thrust. For some trajectories, these limitations can result in a very large relative PMF when the loop is closed around the guidance algorithm in order to achieve pinpoint accuracy in the presence of dispersions. In the case of this algorithm, should pinpoint landing accuracy not be observed to be achievable at 200 m above ground level (AGL), a hover is undertaken by the vehicle in order to ensure pinpoint accuracy at touchdown (i.e., a landed accuracy of less than 100 m).

B. Gradient-Based Optimal Control Algorithm

The general optimal control problem is the process of finding the control history $\mathbf{u}(t)$ and final time t_f that minimizes the performance index:

$$J = \phi[\mathbf{x}(t_f), t_f] + \int_{t_0}^{t_f} L[\mathbf{x}(t), \mathbf{u}(t), t] dt \quad (13)$$

for a given set of system equations,

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \quad (14)$$

that describe the physical system. For the terminal descent problem, the state variables of interest (namely, the position and velocity vectors) are known at an unknown terminal time. The main difficulty associated with this type of problem is the free terminal time, which increases the dimensions of the optimization problem to be solved. Often, the terminal time is thought of as an additional control parameter. Classical optimal control theory presents several solution methods for the class of problem, with the terminal conditions being specified at a free terminal time, including neighboring extremal methods, gradient methods, and quasi-linearization methods [9]. All three methods are iterative and rely on an initial solution that is modified through successive linearization. A gradient-based approach allows for less stringent conditions to be imposed on the initial solution than other classical methods, making it preferable for conceptual design for propulsive terminal descent. However, near the optimum, the number of iterations increases dramatically. The constraints associated with the terminal descent problem, namely the surface constraint and the maximum available thrust, can either be handled through penalty methods that penalize deviations from the constraints or by adjoining them to the objective function, with the

latter being implemented in this analysis. For the propulsive terminal descent problem, the states $\mathbf{x}(t)$ are the position and velocity of the vehicle relative to the target, and the control vector $\mathbf{u}(t)$ is the magnitude and direction of the thrust, or equivalently, the acceleration of the vehicle. A maximum thrust magnitude and an altitude restriction to prevent subterranean trajectories provide the constraints for the problem. With no weighting on the final time, a quadratic performance index can be formulated in the form of Eq. (13), which is solely composed of the integrated control vector $\mathbf{u}(t)$:

$$J = \frac{1}{2} \int_{t_0}^{t_f} \mathbf{u}(t)^T \mathbf{u}(t) dt \quad (15)$$

The solution algorithm for the gradient-based method implemented for this study is as follows [9]:

1. Obtain the equations describing the motion of the vehicle $\mathbf{f}(\mathbf{x}, \mathbf{u}, t)$.
2. Determine the constraints for the problem, the thrust magnitude, and the radius of the planet's surface and form the adjoint constraint equations $\boldsymbol{\psi}[\mathbf{x}(t), t]$.
3. Estimate the control history $\mathbf{u}(t)$ for the thrust vector and the terminal time t_f .
4. Integrate the equations of motion (14) forward, using the initial conditions $\mathbf{x}(t_0)$ and estimated control history from step 3 from t_0 to t_f . Record $\mathbf{x}(t)$, $\mathbf{u}(t)$, $\boldsymbol{\psi}[\mathbf{x}(t_f), t_f]$,

$$\left[\frac{d\phi}{dt} + L \right]_{t=t_f}$$

and

$$\left[\frac{d\boldsymbol{\psi}}{dt} \right]_{t=t_f}$$

5. Integrate backward in time the equations:

$$\dot{\mathbf{p}} = - \left(\frac{d\mathbf{f}}{d\mathbf{x}} \right)^T \mathbf{p} - \left(\frac{dL}{d\mathbf{x}} \right)^T; \quad \mathbf{p}(t_f) = \left(\frac{d\phi}{d\mathbf{x}} \right)_{t=t_f} \quad (16)$$

$$\dot{R} = - \left(\frac{d\mathbf{f}}{d\mathbf{x}} \right)^T R; \quad R(t_f) = \left(\frac{d\boldsymbol{\psi}}{d\mathbf{x}} \right)_{t=t_f} \quad (17)$$

to obtain the influence functions and the matrix of influence functions.

6. Simultaneously, with the backward integration of step 5, compute the quantities:

$$I_{\psi\psi} = \int_{t_0}^{t_f} R^T \frac{\partial \mathbf{f}}{\partial \mathbf{u}} W^{-1} \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right)^T R dt \quad (18)$$

$$I_{J\psi} = I_{\psi J}^T = \int_{t_0}^{t_f} \left(\mathbf{p}^T \frac{\partial \mathbf{f}}{\partial \mathbf{u}} + \frac{\partial L}{\partial \mathbf{u}} \right) W^{-1} \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right)^T R dt \quad (19)$$

$$I_{JJ} = \int_{t_0}^{t_f} \left(\mathbf{p}^T \frac{\partial \mathbf{f}}{\partial \mathbf{u}} + \frac{\partial L}{\partial \mathbf{u}} \right) W^{-1} \left[\left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right)^T \mathbf{p} + \left(\frac{\partial L}{\partial \mathbf{u}} \right)^T \right] dt \quad (20)$$

where the matrix W is an arbitrary, time-varying matrix that is positive definite.

7. Choose values of $d\boldsymbol{\psi}$ that move the terminal condition $\boldsymbol{\psi}[\mathbf{x}(t_f), t_f]$ closer to the desired value of $\boldsymbol{\psi}[\mathbf{x}(t_f), t_f] = 0$.

8. Determine the vector:

$$\mathbf{v} = - \left[I_{\psi\psi} + \frac{1}{b} \frac{d\boldsymbol{\psi}}{dt} \left(\frac{d\boldsymbol{\psi}}{dt} \right)^T \right]^{-1} \left[d\boldsymbol{\psi} + I_{J\psi} + \frac{1}{b} \left(\frac{d\phi}{dt} + L \right) \left(\frac{d\boldsymbol{\psi}}{dt} \right) \right] \quad (21)$$

where b is a weighting constant on the sensitivity terminal constraint sensitivity and is used to limit the step size.

9. Determine increments to the control vector $\delta \mathbf{u}(t)$ and the terminal time dt_f :

$$\delta \mathbf{u}(t) = -W^{-1} \left[\frac{\partial L}{\partial \mathbf{u}} + (\mathbf{p} + R\mathbf{v})^T \left(\frac{d\boldsymbol{\psi}}{dt} \right) \right] \quad (22)$$

$$dt_f = -\frac{1}{b} \left(\frac{d\phi}{dt} + \mathbf{v}^T \frac{d\boldsymbol{\psi}}{dt} + L \right)_{t=t_f} \quad (23)$$

10. Increment the estimates for the control vector $\mathbf{u}(t)$ and the terminal time t_f :

$$\mathbf{u}^{\text{new}}(t) = \mathbf{u}^{\text{old}}(t) + \delta \mathbf{u}(t) \quad (24)$$

$$t_f^{\text{new}} = t_f^{\text{old}} + dt_f \quad (25)$$

11. Iterate using steps 4 through 10 until $\boldsymbol{\psi}[\mathbf{x}(t_f), t_f] = \mathbf{0}$:

$$\left[\frac{d\phi}{dt} + \mathbf{v}^T \frac{d\boldsymbol{\psi}}{dt} + L \right]_{t=t_f} = 0$$

and $I_{JJ} - I_{J\boldsymbol{\psi}} I_{\boldsymbol{\psi}\boldsymbol{\psi}}^{-1} I_{\boldsymbol{\psi}J} \leq \varepsilon$, where ε is the acceptable tolerance.

12. Record the solution for the control history $\mathbf{u}(t) = \mathbf{u}^{\text{new}}(t)$.

This iterative solution is advantageous, as it finds a local minimum in the control effort relatively robustly and as accurately as the tolerance prescribed (or until a maximum number of iterations is reached). However, this approach is computationally intensive and may require numerous iterations before convergence occurs (if it occurs), particularly if a poor initial solution is given. Because convergence on the optimum is not guaranteed for a gradient-based algorithm, it is necessary to incorporate a limit on the number of iterations. Further adding to the computational complexity of the algorithm, the gradient is computed using numerical derivatives that increase the number of function calls dramatically, depending on the scheme used to evaluate the derivatives.

C. Analytic Energy-Optimal Control Algorithm

By assuming a planar nonrotating planet with no atmosphere, D'Souza derived an analytic, unconstrained energy-optimal propulsive terminal descent algorithm that met the necessary and sufficient conditions for an optimal control law [10]. The problem described by D'Souza minimizes the performance index:

$$J = \Gamma t_f + \frac{1}{2} \int_{t_0}^{t_f} \mathbf{a}^T \mathbf{a} dt \quad (26)$$

which includes a weighting Γ on the final time. It is shown that the control law that minimizes this index, under the assumptions mentioned previously, is given by

$$\mathbf{a} = -4 \frac{\Delta \mathbf{v}}{t_{\text{go}}} - 6 \frac{\Delta \mathbf{r}}{t_{\text{go}}^2} - \mathbf{g} \quad (27)$$

where

$$\Delta \mathbf{r} = (r_1 - r_{f_1} \quad r_2 - r_{f_2} \quad r_3 - r_{f_3})^T \quad (28)$$

$$\Delta \mathbf{v} = (v_1 - v_{f_1} \quad v_2 - v_{f_2} \quad v_3 - v_{f_3})^T \quad (29)$$

$$\mathbf{g} = (0 \quad 0 \quad g)^T \quad (30)$$

The time to go t_{go} is shown from the transversality condition from the Euler-Lagrange equations to be the real positive root of the equation:

$$t_{\text{go}}^4 - 2 \frac{\Delta \mathbf{v}^T \Delta \mathbf{v}}{\Gamma + (g^2/2)} t_{\text{go}}^2 - 12 \frac{\Delta \mathbf{v}^T \Delta \mathbf{r}}{\Gamma + (g^2/2)} t_{\text{go}} - 18 \frac{\Delta \mathbf{r}^T \Delta \mathbf{r}}{\Gamma + (g^2/2)} = 0 \quad (31)$$

Equation (31) can be solved either numerically or analytically and substituted into Eq. (27) to obtain the desired acceleration vector for all time. The commanded thrust is then this acceleration vector multiplied by the mass of the vehicle at the given instant in time.

This analytic algorithm has a clear computational advantage when compared with the iterative optimal control solution, as it requires a single computation for the free time to go, which is, in turn, substituted into an equation of known state parameters (relative position and velocity to the target) to obtain the commanded thrust. However, the formulation does not have constraints on either the maximum thrust magnitude or the minimum altitude. Without these constraints, a physically impossible solution could be obtained. However, by propagating ahead in time, violations in these constraints can be foreseen, and an appropriate adjustment in the weighting on time to go Γ can be prescribed using Newton iteration. Although requiring numerical integration and iteration, obtaining the proper increment on the time to go weighting was shown to be significantly less computationally intensive (a reduction in the number of iterations by approximately an order of magnitude) than the gradient method described previously.

D. Second-Order Cone Algorithm

For propulsive terminal descent, the control space is, in general, nonconvex due to a minimum allowable thrust magnitude, which none of the prior three methods described take into account. Because of this nonconvex constraint, determining a control history that results in the global minimum with regard to PMF is not guaranteed. Açıkmüşe and Ploen have shown that there exists a convex programming approach to the propulsive terminal descent problem that relaxes this nonconvex constraint and, in turn, guarantees arrival at the global minimum in PMF [11]. Their work also reformulates the convex propulsive terminal descent guidance problem in the discretized case as a second-order cone programming problem (SOCP), which can be solved using interior point solution methods. Interior point methods are well studied and are known to converge to within a given tolerance of the optimum in a known finite number of iterations, which cannot be said for any solution method of the general propulsive terminal descent guidance problem. Furthermore, the solution obtained by the interior point method to any desired accuracy is feasible. Therefore, potential application to flight exists for derivatives of this algorithm. The derivation of this algorithm assumes constant gravitational acceleration and negligible atmospheric forces; however, variations in these quantities from that modeled can be treated as disturbances when the guidance algorithm is implemented in a closed-loop sense.

The nonconvex propulsive terminal descent guidance law problem can be posed as

Minimize:

$$J = \int_{t_0}^{t_f} \|\boldsymbol{\tau}_C\| dt \quad (32)$$

Subject to:

$$\ddot{\mathbf{r}} = \mathbf{g} + \boldsymbol{\tau}_C/m; \quad \dot{m} = -\alpha \|\boldsymbol{\tau}_C\|; \quad 0 < \rho_1 \leq \|\boldsymbol{\tau}_C\| \leq \rho_2$$

$$r_3(t) \geq 0; \quad \|S_j \mathbf{x}(t) - \mathbf{v}_j\| + \mathbf{c}_j^T \mathbf{x}(t) + \beta_j \leq 0$$

$$j = 1, \dots, n; \quad \mathbf{r}(0) = \mathbf{r}_0; \quad \dot{\mathbf{r}}(0) = \dot{\mathbf{r}}_0$$

$$m(0) = m_0; \quad \mathbf{r}(t_f) = \dot{\mathbf{r}}(t_f) = \mathbf{0}$$

where path constraints are included (i.e., S_j , \mathbf{v}_j , \mathbf{c}_j , and β_j). Observing that the path constraints are already convex, the entire problem can be made convex by reformulating the thrust in terms of a new variable ζ . This introduces an additional constraint to ensure that the new variable acts as a slack variable.

Minimize:

$$J = \int_{t_0}^{t_f} \zeta(t) dt \quad (33)$$

Subject to:

$$\begin{aligned} \ddot{\mathbf{r}} &= \mathbf{g} + \boldsymbol{\tau}_c/m; & \dot{m} &= -\alpha\|\boldsymbol{\tau}_c\|; & \|\boldsymbol{\tau}_c\| &\leq \zeta(t) \\ 0 < \rho_1 < \zeta(t) &\leq \rho_2; & r_3(t) &\geq 0\|S_j\mathbf{x}(t) - \mathbf{v}_j\| + \mathbf{c}_j^T\mathbf{x}(t) + \beta_j \leq 0 \\ j &= 1, \dots, n; & \mathbf{r}(0) &= \mathbf{r}_0; & \dot{\mathbf{r}}(0) &= \dot{\mathbf{r}}_0 \\ m(0) &= m_0; & \mathbf{r}(t_f) &= \dot{\mathbf{r}}(t_f) = \mathbf{0} \end{aligned}$$

Graphically, this transformation of the nonconvex control space to convex control space by the introduction of this slack variable is shown in Fig. 3 for a two-dimensional case. In general, the solution to the nonconvex problem is a feasible solution to the relaxed problem; however, the converse is not guaranteed to be true. Although, Açıkmese and Ploen have shown that the optimal solution found by the relaxed problem is also the optimal solution to the nonconvex problem [11].

To implement this algorithm numerically, the continuous time problem needs to be discretized. In their work, Açıkmese and Ploen [11] describe a transformation for Eq. (33) that leads to a continuous optimization problem, but one which has a convex performance index as well as convex state and control constraints. In particular, the change of variables they introduce lead to constraints that are either linear or of the form of a second-order cone. Following this change of variables, the problem is discretized in time while enforcing the constraints at the nodes of the resulting mesh, which results in a finite-dimensional SOCP problem. The resulting SOCP can be solved using an interior point method algorithm, which will arrive at the global minima in polynomial time [10].

When implementing this algorithm, several existing software packages can be used. In particular, SeDuMi was used for this study, which is a MATLAB® oriented software package that solves symmetric cone optimization problems, such as the problem posed here [12]. Should one not want to rely on a preexisting package, several other SOCP solution methods exist, such as the interior point method described by Lobo et al. and the Q method for second-order cones described by Lobo et al. [13] and Xia and Alizadeh [14].

E. Comparison of the Four Propulsive Guidance Algorithms

The four propulsive terminal descent algorithms were evaluated in a dispersed environment, shown in Table 1, for the case of a large robotic entry vehicle from the parachute deployment point through the soft touchdown, assuming ideal navigation knowledge. One thousand cases for each algorithm were sampled, and the PMF required to achieve pinpoint accuracy was computed. For this investigation, the propulsive descent is initiated at Mach 0.9 in all cases. For the dispersions considered in Table 1, the Mach 0.9 initiation condition correlates to altitudes between 0.8 and 3.6 km and distances that range from 7.8 km behind the landing site to 8.9 km ahead of the landing site (behind the landing site refers to cases when $\mathbf{v}^T \mathbf{r} > 0$ and ahead of the target refers to cases when $\mathbf{v}^T \mathbf{r} < 0$). For comparison, a gravity turn starting at Mach 0.9 ($h = 1.6$ km, $v = 200$ m/s, $\gamma = -34$ deg) to target a soft touchdown of the nominal vehicle results requires 231 kg of propellant (a PMF of

0.088). Figure 4 shows the cumulative distribution functions (CDFs) for the PMF of the gradient-based optimal guidance algorithm, the energy-optimal analytic guidance algorithm, the second-order cone algorithm, and the modified Apollo lunar module guidance algorithm. The results shown in Fig. 4, as would be expected, are dependent on the initiation conditions (altitude and downrange distance from the target). Furthermore, the inclusion of other operational constraints, such as the maximum off-nadir angle allowable for operation of navigational instruments that are not considered in this investigation, would also impact the distribution of the PMF shown in Fig. 4 for each of the algorithms.

At the 99% confidence level, there is little discernible difference in the PMF for the gradient-based optimal algorithm, second-order cone algorithm, and the analytical algorithm. Even at lower confidence levels, the PMF is only a few percent different. For example, the closed-form analytic algorithm has a PMF that is 7% worse at 50% confidence and 0.5% worse at 90% confidence. On the other hand, the PMF for the modified Apollo lunar module algorithm is significantly higher at all confidence levels, reaching 0.87 at the 99% confidence interval (CI), an unachievable value in a realistic design. This significant difference in PMF is due to the lack of robustness to the environmental parameters variations seen because the algorithm was developed for use on a planetary body lacking an atmosphere, and the prediction of the acceleration profiles does not agree with that experienced. Therefore, in order to achieve a soft landing with pinpoint accuracy, a constant altitude translational maneuver is undertaken once the vehicle has crossed an altitude threshold (200 m AGL).

For application in conceptual design studies, the PMF performance of the closed-form analytic algorithm is sufficiently close to the gradient-based fuel-optimal PMF. As an example of the computational efficiency of the closed-form algorithm, as compared with the iterative gradient algorithm, for a single control history determination (i.e., a call of the closed-loop guidance algorithm at one instant in time during the descent), the closed-form algorithm required eight functional evaluations of the equations of motion, whereas the iterative algorithm required 327 functional evaluations of the equations of motion. This clear computational advantage is afforded by the closed-form analytic algorithm and makes this the preferred algorithm for further studies conducted, as well as for conceptual design.

The gradient-based algorithm and the second-order cone algorithm arrive at nearly the same PMF for the propulsive pinpoint landing problem with no minimum thrust constraint, as both problems (as posed) are convex. However, for flight applications, the second-order cone algorithm with a lower thrust bound is preferred, unless the flight system's engines have been specifically designed for multiple restarts. Figure 5 shows a CDF of the PMF for the second-order cone algorithm, which includes a minimum thrust constraint of $\sim 35\%$ of the maximum thrust value (1142 ± 57 N). Comparison between Figs. 4 and 5 shows the propellant cost of including this constraint into the problem formulation. This emphasizes the problem's dependence on the constraints incorporated into the algorithm's formulation.

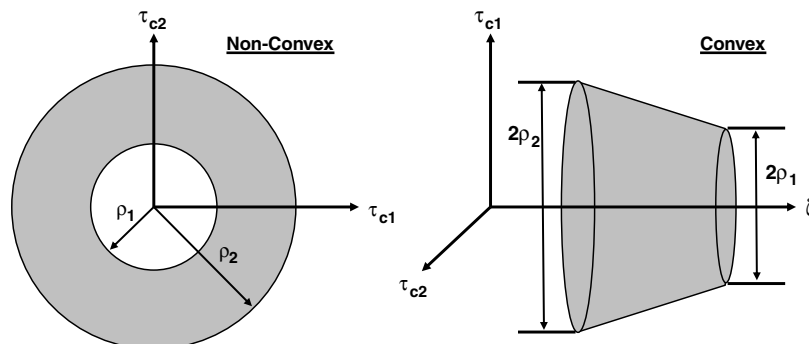


Fig. 3 Nonconvex thrust control space and convex thrust control space [11].

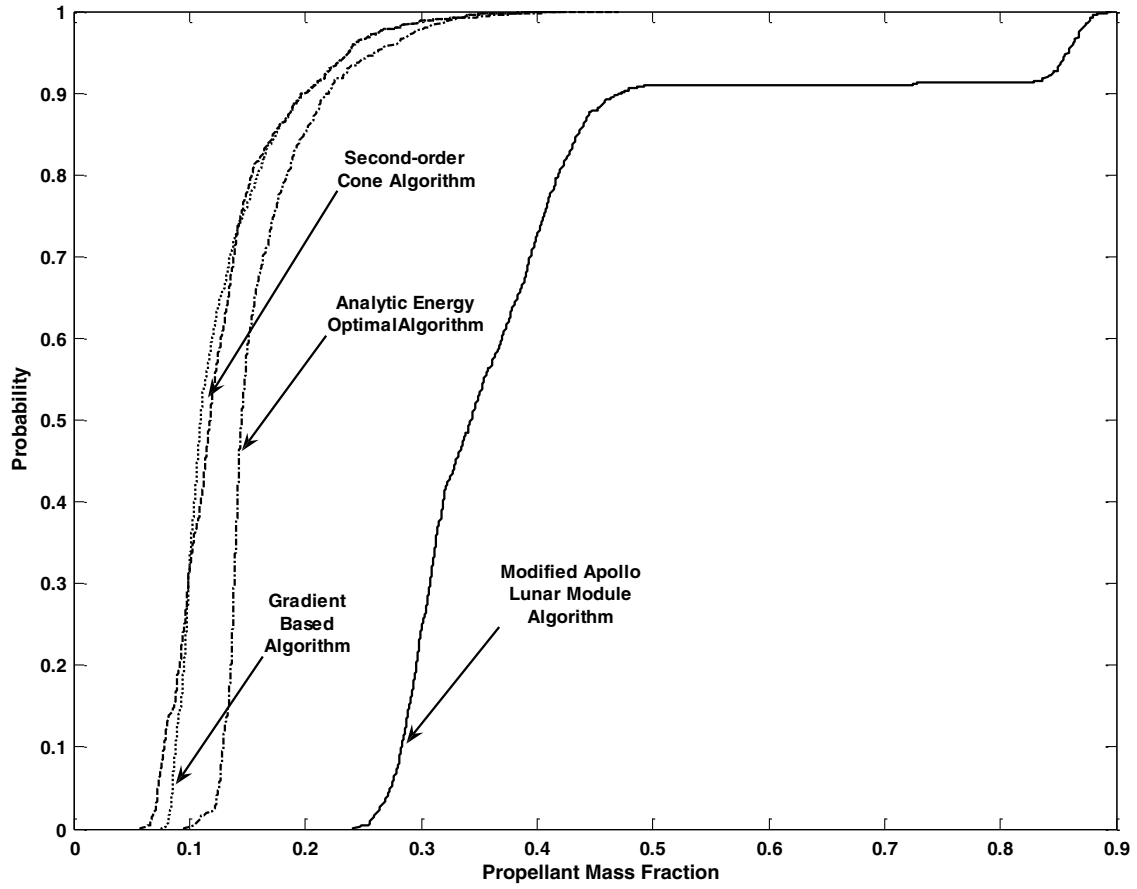


Fig. 4 CDFs of the PMFs for each of the algorithms without a minimum thrust constraint.

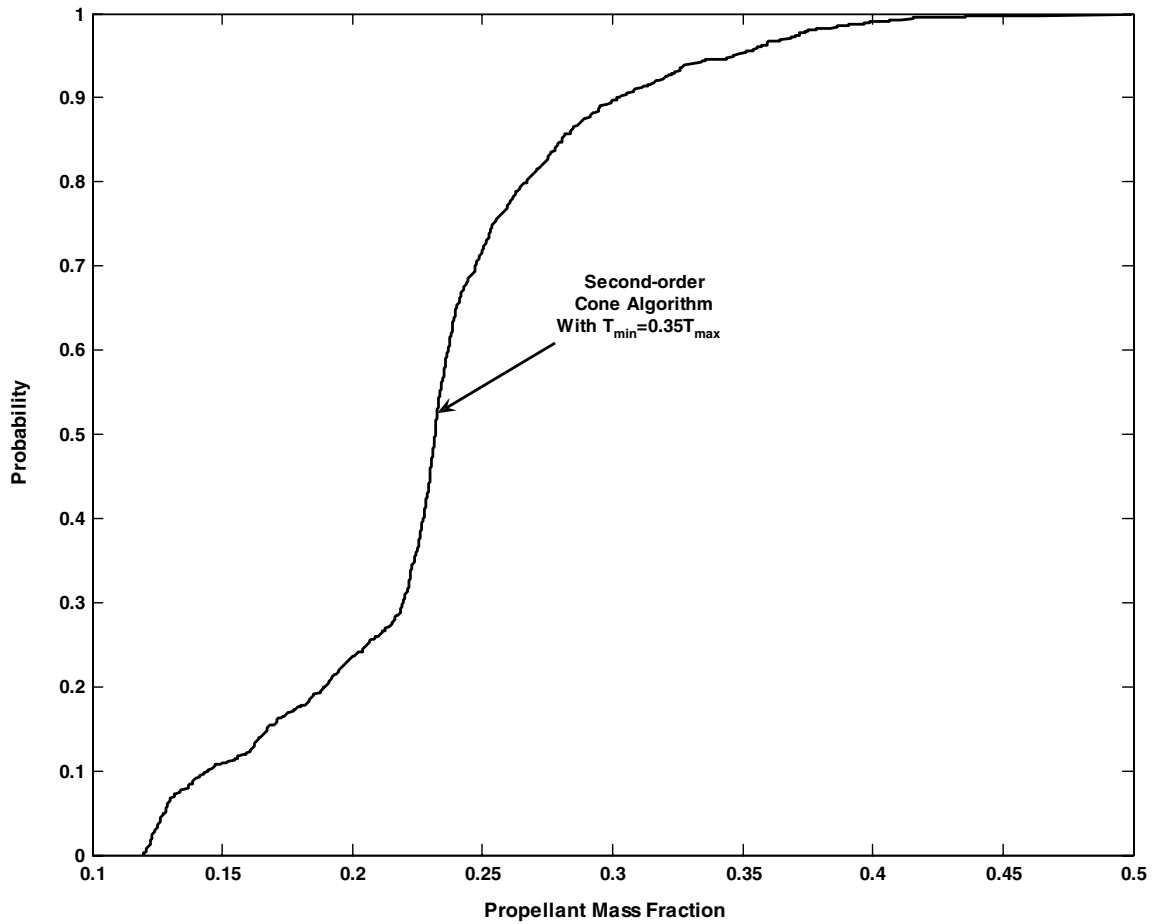


Fig. 5 CDF of the PMF for the second-order cone algorithm with a minimum thrust constraint.

Table 2 Comparison of the four propulsive terminal descent algorithms

Metrics	Modified Apollo lunar module algorithm	Gradient-based optimal control algorithm	Analytic energy-optimal algorithm	Second-order cone algorithm
PMF _{min} /PMF 99% CI	0.36	1.00	0.97	0.78
Equations of motion evaluations	1	327	8	163
Robustness	Poor	Good	Moderate	Good
Ease of implementation	Good	Poor	Moderate	Poor
Potential extensibility to flight	Good	Poor	Poor	Moderate
Numerical stability	Good	Moderate	Moderate	Good

Table 2 provides a quantitative and qualitative comparison between each of the four algorithms, in which the qualitative metrics were assessed on a relative basis to one another. In this table, the minimum thrust was included for the second-order cone algorithm, so that a comparison on the optimality (including all included constraints) can be made. In Table 2, the quantitative metrics include the following:

1) The ratio of the minimum PMF at the 99% confidence level is observed across all algorithms (PMF_{min}) to the algorithm's PMF at the 99% confidence level (PMF), which is a surrogate for the optimality of the algorithm.

2) The mean number of equations of motion evaluations are at one time step in order to compute a converged solution (a solution for which the percent change in PMF is less than 0.01%), which is a measure of the run-time complexity of the algorithm.

The qualitative aspects considered in Table 2 include the following:

1) The robustness of the algorithm is observed by stressing the algorithm with low altitude (less than 1 km AGL) and large downrange (greater than 4 km) diverts ahead and behind the designated landing site, observing how reliably a flyable solution is achieved (PMF < 0.5 and $T < T_{max}$).

2) The ease of implementation is assessed from a conceptual design perspective, which is a function of the length of the algorithm as implemented and the time it took to code.

3) The potential extensibility to flight is a measure of the feasibility of the guidance solutions obtained through the iterations, how easily realistic mission constraints can be incorporated, as well as the overall optimality of the solution in terms of PMF.

4) The numerical stability of the algorithm is a qualitative assessment of the stability of the algorithm (e.g., the tolerance required to be maintained in the integration in order to achieve a solution, how many infeasible solutions are commanded, etc.).

As shown in Table 2, at the 99% confidence level, the gradient-based algorithm provided the solution with the minimum PMF with the analytic algorithm close to this value. By incorporating the minimum thrust bound, the second-order cone algorithm's PMF is 78% of the best, and the modified Apollo Lunar Module algorithm's performance is significantly poorer. The table also shows that the number of equations of motion evaluations required for the second-order cone algorithm are half that of the gradient-based algorithm. This results in a run time on the order of one-quarter of the gradient-based algorithm, due to the polynomial convergence properties of the SOCP, making the second-order cone algorithm preferable in conceptual studies to the gradient-based optimal guidance law. Large low altitude diverts were handled the best by the gradient-based algorithm and the second-order cone algorithm, whereas the analytic energy-optimal algorithm obtained some solutions that were impractical to actually command, due to the maximum thrust constraint not being handled directly. The modified Apollo Lunar Module algorithm was the shortest algorithm when implemented and took the least time to do so, whereas the more complex algorithms (the gradient-based algorithm and the second-order cone algorithm) both took longer to implement and required more lines of code. As the modified Apollo Lunar Module algorithm has flown before, it is extremely flight applicable, with the primary concern being incorporating the thrust constraint and other mission constraints into the algorithm (e.g., the maximum allowable offnadir angle). The thrust constraint is handled by changing the initiation criteria to

altitude instead of Mach number, whereas the nadir angle can be tuned by changing the desired terminal vertical velocity and acceleration vectors. The second-order cone algorithm has more extensibility to flight due to the fact that it approaches solutions during iteration from the feasible direction (i.e., $T_{min} \leq T \leq T_{max}$, which allows the vehicle to reach the targeted site), so that, should the iteration be terminated early, a solution reaching the target will be achieved that does not violate thrust constraints. This is in contrast to the gradient-based algorithm and analytic fuel-optimal algorithm, which have no guarantee that a feasible solution will be achieved during the iteration. Finally, the numerical stability of all four algorithms was observed to be moderate to good, with the gradient-based algorithm only requiring tuning of the initial solution guess to speed convergence and the analytic algorithm requiring special handling of complex roots if they arose in the solution of t_{go} (for very large or very short diverts at large velocities).

F. Propulsive Descent Initiation

Using the analytic energy-optimal guidance law, a design space investigation was performed of the altitude, the velocity, the downrange, and the flight-path angle space to determine if a combination of those sensible parameters yields a minimum in the PMF required to achieve pinpoint landing. Figure 6 shows contours of the PMF required to achieve a pinpoint landing in the altitude downrange space, as well as contours of the thrust-to-weight ratio required to achieve that PMF. The contours of the PMF represent the minimum possible propellant usage for a fixed velocity (i.e., allowing the flight-path angle to be optimized between representative bounds). This figure demonstrates the existence of a single PMF minimum in this design space. The existence of such a minimum persisted for all examined cases. This minimum could be leveraged in the further development of guidance algorithms in order to identify the time to begin the propulsive terminal descent and further minimize the amount of propellant required for the propulsive terminal descent phase of flight.

IV. Subsonic Guided Parachute Performance

Several studies have investigated the inclusion of subsonic parachutes on Mars EDL systems in the past; these include one performed by Mitcheltree et al. [15], in which a test program is laid out for Earth qualification of the parachute, and one performed by Witkowski et al. [16]. Guided subsonic steerable parachutes were also considered in the present investigation, in which a circular parachute with a nominal drag coefficient of 0.68 was assumed. The guidance algorithm is based on a model described by Yakimenko et al., which modulates the drag vector in order to steer toward a reference trajectory, using a performance index to minimize the amount of time it takes to arrive on the reference [17]. For evaluation purposes, the parachute is deployed at Mach 0.9 in the descent trajectory subsequent to release of the supersonic parachute and is used until 1 km AGL. At this altitude, the analytic propulsive guidance algorithm is activated. One thousand Monte Carlo cases were conducted using the parameters in Table 1.

As shown in Fig. 7, the results of this Monte Carlo analysis show no major improvement at appreciable confidence levels (e.g., greater than 90%) in the PMF required to achieve pinpoint accuracy. The principal reason for this lack of benefit is that the timeline associated

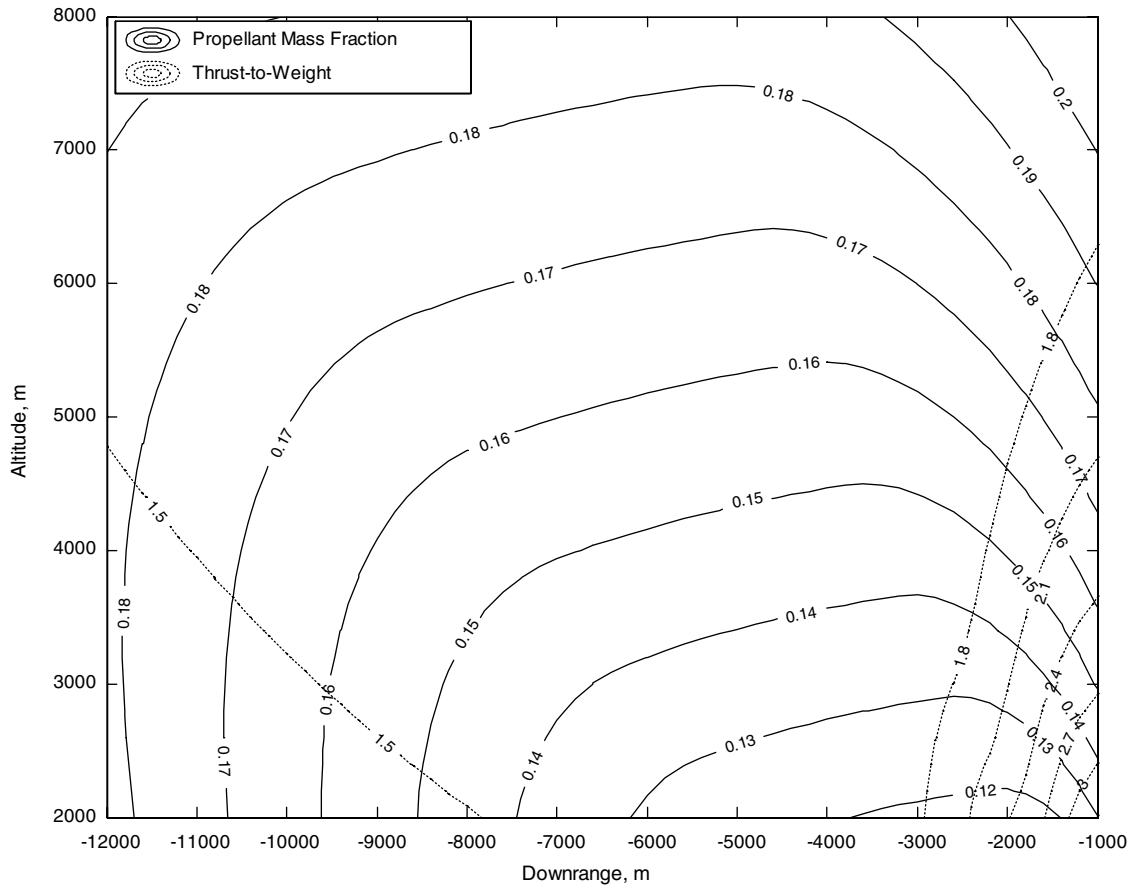


Fig. 6 Design space slice for propulsive terminal descent.

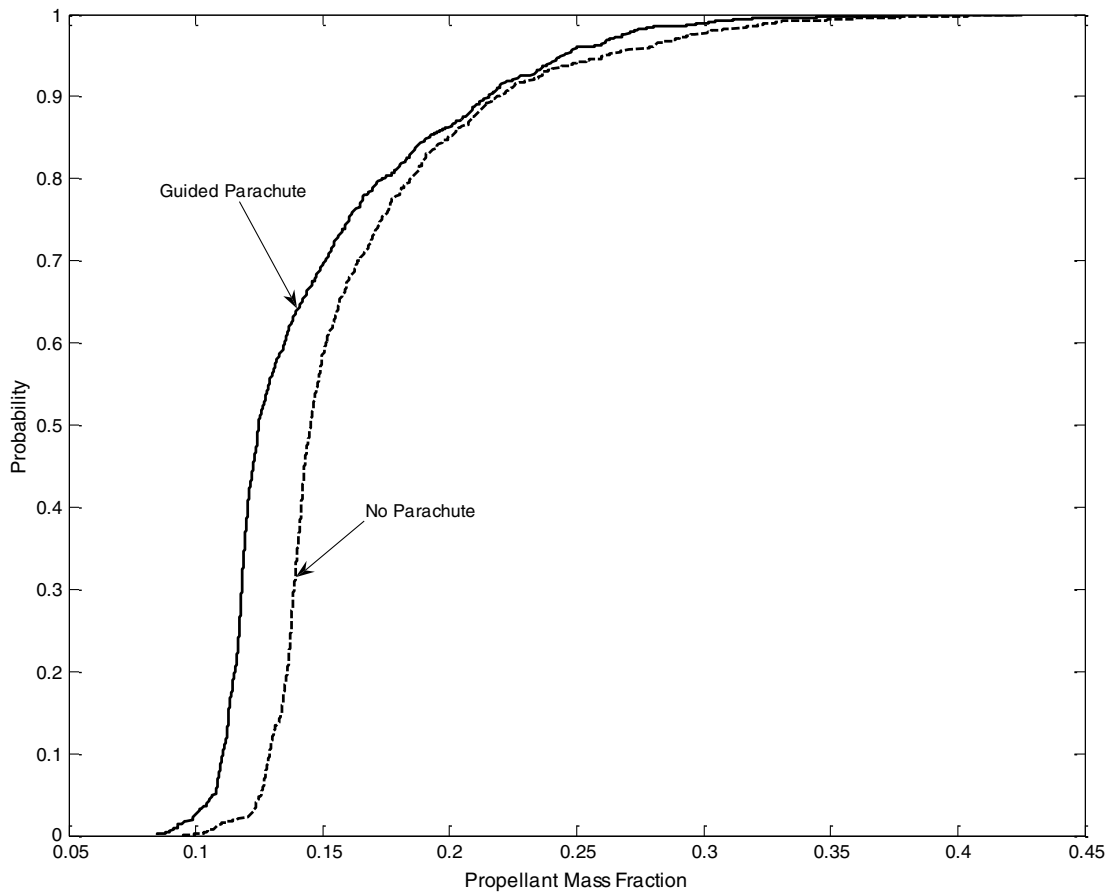


Fig. 7 CDFs showing the impact of a subsonic guided parachute.

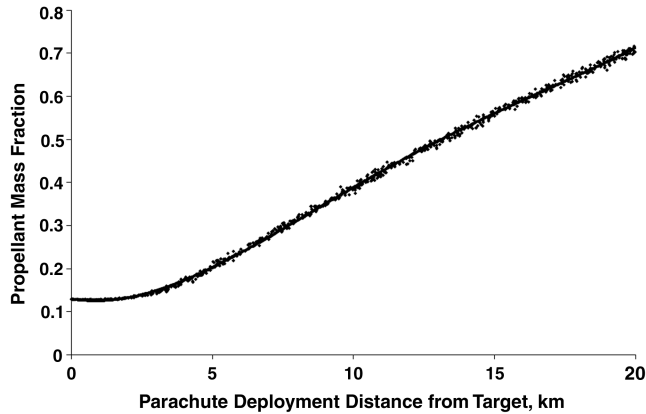


Fig. 8 PMF for supersonic parachute deployment semimajor errors.

with a significant fraction of the Martian entries (e.g., a CDF above 80%) does not allow for the subsonic guided parachute to perform over a significant length of time, as would occur in Earth applications. For all of the cases investigated, the deployment of the subsonic parachute occurred below 5 km, allowing a maximum 4 km guided descent segment. Additionally, there is no consideration for the energy state near the reference trajectory, which is defined as altitude as a sole function of distance to the target. Hence, provided the vehicle is near the reference with proper altitude and distance to the target, the velocity could be exceedingly high, resulting in spiraling trajectory about the reference. These spiraling trajectories, in turn (because they have higher kinetic energy), will increase the propulsive force required to negate it, as compared with a trajectory that follows the reference without spiraling.

V. Hypersonic Guidance Performance

The ramifications of the inclusion of a hypersonic guidance algorithm was evaluated by examining the semimajor axis of the supersonic parachute deployment ellipse and examining the PMF required to achieve a pinpoint landing using the closed-form analytic guidance law and ideal navigation. As shown in Fig. 8, these results demonstrate a marginal PMF reduction below a position delivery error at a parachute deployment of 3 km. Because a supersonic parachute deployment ellipse greater than 3 km leads to a dramatic rise in the PMF required to achieve pinpoint landing, a 3 km dispersion footprint is suggested as a target for Mars hypersonic guidance algorithm technology development. Note that the wind speeds could alter the PMF required for pinpoint accuracy during the parachute drift phase, and that strong variable winds (greater than 50 m/s) could cause a PMF increase of as much as 0.3.

VI. Terrain Relative Navigation Sensor Performance

The impact of TRN on the ability to perform pinpoint landing was studied via linear covariance by assuming a sensor suite consisting of an inertial measurement unit (accelerometer and gyroscope), a radar altimeter, a velocimeter, and a three-dimensional TRN sensor. This sensor suite has the specifications shown in Table 3. Additionally, a

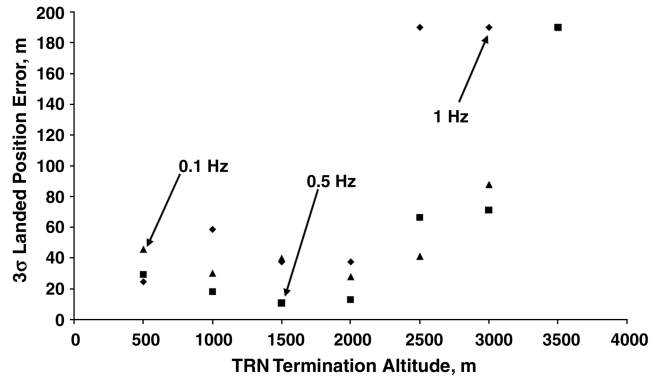


Fig. 9 Three-sigma landed accuracy for various TRN termination altitudes and sampling frequencies.

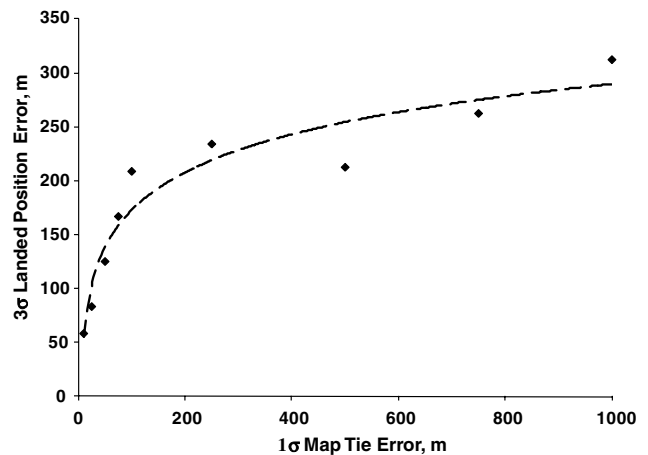


Fig. 10 Map tie error impact on landing uncertainty.

covariance matrix at an EI of 10 min was obtained from the Jet Propulsion Laboratory for MSL, assuming TCM-5 was performed. This entry uncertainty was propagated to the parachute deployment ellipse with sensor error accumulating throughout the descent. Using this sensor suite, a three-sigma navigational knowledge error of approximately 200 m was seen at parachute deployment. Because the knowledge error exceeds the physical delivery error required for pinpoint landing (for this case), pinpoint landing is precluded even in the presence of a perfect delivery state.

By varying the termination altitude of the TRN sensor, a surrogate of the performance achievable by the sensor is available. Additionally, the frequency of navigation data updates performed in-flight is investigated. The three-sigma landed error variation for the nominal case is seen in Fig. 9 as a function of these two variables. As expected, the general trend shows improving landed accuracy with decreased termination altitude, as the TRN sensor's accuracy is altitude dependent. A minimum is seen as well for the 0.1 and 0.5 Hz cases at approximately 1.5 km. The TRN navigation sensor data below this altitude can be more inaccurate than the known state as a

Table 3 Navigation sensor data

Sensor	Altitude active, km	Termination altitude, km	Mean bias	Sensor realization error	Noise, 1σ
Accelerometer	N/A	N/A	30 μg	66 ppm	5 $\mu\text{g}/\text{Hz}^{1/2}$
Gyroscope	N/A	N/A	0.02 deg/hr	1.6 ppm	50 $\mu\text{rad}/\text{Hz}^{1/2}$
Radar	20	0.01	-0.1 m	0 m	5 m
Velocimeter	2	0.01	$\begin{pmatrix} -0.14 \\ 0.07 \\ -0.06 \end{pmatrix}$ m/s	$\begin{pmatrix} 5.79 \times 10^{-5} \\ 5.79 \times 10^{-5} \\ 5.79 \times 10^{-5} \end{pmatrix}$ m/s	$\begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \end{pmatrix}$ m/s
3-D terrain relative sensor	8	2.5	$\begin{pmatrix} 4.3 \\ 28.7 \\ -15.7 \end{pmatrix}$ m	$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ m	42 m

result of sensor bias (~ 40 m) and noise (~ 40 m, 1σ). These error sources, when integrated in the filter, can lead to less accurate actual state knowledge due to the corruption of the navigated state data with newly sampled inaccurate data. This phenomenon is not seen in the 1 Hz sampling rate, as the sampled data continually improve the estimated state, because the filter has less information for its synthesis.

Additionally, the impact of map tie error was investigated by considering values from 0 to 1000 m and examining the total landed accuracy of the vehicle. As shown in Fig. 10, for a sub-100-m landed accuracy, the maximum allowable one standard deviation map tie error is approximately 25 m. The variation in landed uncertainty with map tie error is approximately logarithmic. This can be attributed to the TRN sensor negating the majority of the map tie error above 200 m and is largely a function of the capabilities of the sensor.

VII. Conclusions

This investigation conducted an integrated assessment on a common vehicle of various propulsive descent guidance algorithms, as well as other GNC technologies that could be considered for pinpoint landing (sub-100-m landed accuracy) at Mars, including hypersonic guidance accuracy, subsonic steerable parachutes, and terrain relative navigational accuracy requirements. An iterative gradient-based optimal guidance law, which minimized control effort (acceleration) was implemented. However, it was seen that a near propellant-optimal guidance law, which assumes a flat atmosphere-free planet, was of sufficient accuracy for conceptual design, although it was significantly less computationally intensive. Additionally, a convex algorithm, which arrives at the constrained global minimum in PMF, was investigated for its strong applicability to flight. Using the near propellant-optimal analytic guidance law, a minimum was shown to exist in the altitude, velocity, downrange, flight-path angle space, which could be leveraged during the implementation of the algorithm.

Pinpoint landing technology system trade studies were performed and demonstrated that driving hypersonic guidance terminal state conditions (parachute deployment conditions) to an accuracy below 3 km resulted in marginal performance gains. In addition, it was shown that the use of a subsonic guided parachute did not significantly decrease the PMF required for pinpoint landing, whereas the added system complexity of a subsonic guided parachute would have to be seriously considered. This investigation also demonstrated that map tie error needed to be driven below 25 m, and TRN sensor termination altitude needed to be driven below 1 km for sub-100-m landed accuracy.

Several important caveats on the implications of this study's results should be mentioned. The results of this study were obtained assuming a parachute deployment ellipse with similar semimajor axis length to the MSL mission at a constant altitude. The altitude variations that will inevitably result from the system and environment dispersions should be accounted for to increase the fidelity of these conceptual results. Additionally, the initiation of the propulsive descent for this work started at Mach 0.9, not at a specific altitude; the results are only representative of the algorithm's performance under the given set of dispersions. However, for a different set of boundary conditions (initiation state and terminal state), the results demonstrated could vary. The results could further vary when other mission applicable constraints (e.g., nadir angle constraints) are considered.

References

- [1] Wolf, A., Tooley, J., Ploen, S., Gromov, K., Ivanov, M., and Acikmese, B., "Performance Trades for Mars Pinpoint Landing," *2006 IEEE Aerospace Conference*, Paper 1661, IEEE Publications, Piscataway, NJ, March 2006.
doi:10.1109/AERO.2006.1655793
- [2] Braun, R. D., and Manning, R. M., "Mars Exploration Entry, Descent, and Landing Challenges," *Journal of Spacecraft and Rockets*, Vol. 44, No. 2, 2007, pp. 310–323.
doi:10.2514/1.25116
- [3] Mendek, G. F., and Carman, G. L., "Guidance Design for Mars Smart Landers Using the Entry Terminal Point Controller," *AIAA Atmospheric Flight Mechanics Conference*, AIAA Paper 2002-4502, Aug. 2002.
- [4] Drake, B. G. (ed.), "Reference Mission Version 3.0 Addendum to the Human Exploration of Mars: The Reference Mission of the NASA Mars Exploration Study Team," NASA SP-6107-ADD, June 1998.
- [5] Brand, T., Fuhrman, L., Geller, D., Hattis, P., Paschall, S., and Tao, Y., "GN&C Technology Needed to Achieve Pinpoint Landing Accuracy at Mars," *AIAA/AAS Astrodynamics Specialists Conference and Exhibit*, AIAA Paper 2004-4748, Aug. 2004.
- [6] Stripe, S., Way, D. W., Dwyer, A. M., and Balaran, J., "Mars Science Laboratory Simulations for Entry, Descent, and Landing," *Journal of Spacecraft and Rockets*, Vol. 43, No. 2, 2006, pp. 311–323.
doi:10.2514/1.19649
- [7] Duvall, A., Justus, C., and Keller, V., "Global Reference Atmospheric Model (GRAM) Series for Aeroassist Applications," *43rd AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2005-1239, Jan. 2005.
- [8] Wong, E., Singh, G., and Masciarelli, J., "Autonomous Guidance and Control Design for Hazard Avoidance and Safe Landing on Mars," *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, AIAA Paper 2002-4619, Aug. 2002.
- [9] Bryson, A., and Ho, Y., *Applied Optimal Control*, Hemisphere, New York, 1963.
- [10] D'Souza, C., "An Optimal Guidance Law for Planetary Landing," *AIAA Paper 1997-3709*, 1997.
- [11] Açikmeşe, B., and Ploen, S., "Convex Programming Approach to Powered Descent Guidance for Mars Landing," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, 2007, pp. 1353–1366.
doi:10.2514/1.27553
- [12] Sturm, J., "Using SeDuMi 1.02: a MATLAB Toolbox for Optimization Over Symmetric Cones," *Optimization Methods and Software*, Vol. 11, No. 1, 1999, pp. 625–653.
doi:10.1080/10556789908805766
- [13] Lobo, M., Vandenberghe, S., Boyd, S., and Lebret, H., "Applications of Second Order Cone Programming," *Linear Algebra and Its Applications*, Vol. 284, No. 1, 1998, pp. 193–228.
doi:10.1016/S0024-3795(98)10032-0
- [14] Xia, Y., and Alizadeh, F., "The Q Method for Second Order Cone Programming," *Computers and Operations Research*, Vol. 35, No. 5, 2008, pp. 1510–1538.
doi:10.1016/j.cor.2006.08.009
- [15] Mitcheltree, R., Bruno, R., Slimko, E., Baffes, C., and Konefat, E., "High Altitude Test Program for a Mars Subsonic Parachute," *18th AIAA Aerodynamic Decelerator Conference*, AIAA Paper 2005-1659, May 2005.
- [16] Witkowski, A., Machalick, A., and Taeger, Y., "Mars Subsonic Parachute Technology Task System Overview," *18th AIAA Aerodynamic Decelerator Conference*, AIAA Paper 2005-1657, May 2005.
- [17] Yakimenko, O., Dobrokhodov, V., Johnson, J., Kaminer, I., Dellicker, S., and Benney, R., "On Control of Autonomous Circular Parachutes," *AIAA Atmospheric Flight Mechanics Conference*, AIAA Paper 2002-4753, Aug. 2002.

B. Marchand
Associate Editor