AIAA Atmospheric Flight Mechanics Conference and Exhibit
18 - 21 August 2008, Honolulu, Hawaii

AIAA 2008-6216

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

Bradley A. Steinfeldt*, Michael J. Grant*, Daniel M. Matz†, Robert D. Braun‡
Georgia Institute of Technology, Atlanta, GA, 30332

and

Gregg H. Barton§
Charles Stark Draper Laboratory, Houston, TX 77058

Landing site selection is a compromise between safety concerns associated with the site's terrain and scientific interest. Therefore, technologies enabling pinpoint landing (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 various guidance, navigation, and control technologies that could allow pinpoint landing to occur at Mars has shown that negligible propellant mass fraction benefits are seen for reducing the three-sigma position dispersion at parachute deployment below approximately 3 km. Four different propulsive terminal descent guidance algorithms were analyzed with varying applicability to flight. 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 regards to the initiation time of the propulsive terminal descent was shown to exist for various flight conditions. In addition, subsonic guided parachutes are shown to provide marginal performance benefits due to the timeline associated with Martian entries, and a low computational-cost, yet near fuel optimal propulsive terminal descent algorithm is identified. This investigation also demonstrates that navigation is a limiting technology for Mars pinpoint landing, with overall landed performance being largely driven by navigation sensor and map tie accuracy.

Nomenclature

a_i	=	Acceleration along the i^{th} direction
a_j	=	Scalar defining convex state constraints
$\mathbf{a}$	=	Acceleration vector $= \begin{pmatrix} a_1 & a_2 & a_3 \end{pmatrix}^T$
b	=	Scalar weighting parameter
C_{j_i}	=	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

* Graduate Research Assistant, Guggenheim School of Aerospace Engineering, AIAA Student Member
† Undergraduate Research Assistant, Guggenheim School of Aerospace Engineering, AIAA Student Member
‡ David and Andrew Lewis Associate Professor of Space Technology, Guggenheim School of Aerospace Engineering, AIAA Fellow
§ Group Leader, Mission Design and Analysis Group, AIAA Senior Member

I_{JJ} = Partition used in the optimal-control solution

$I_{J\psi}$ = Partition used in the optimal-control solution

$I_{\psi J}$ = Partition used in the optimal-control solution

$I_{\psi\psi}$ = Partition used in the optimal-control solution

J = Performance index

L = Scalar objective function describing path parameters

m_0 = Initial mass of the vehicle

m_{prop} = Mass of propellant

$\mathbf{p}$ = Influence function vector

r_i = Position along the i^{th} direction

$\mathbf{r}$ = Position vector $= \begin{pmatrix} r_1 & r_2 & r_3 \end{pmatrix}^T$

R = Matrix of influence functions

S_j = Matrix defining convex state constraints

t = Time

t_{go} = Time-to-go until touchdown

$\mathbf{u}$ = Control vector

$\delta\mathbf{u}$ = Control vector increment

v_i = Velocity along the i^{th} direction

$\mathbf{v}$ = Velocity vector $= \begin{pmatrix} v_1 & v_2 & v_3 \end{pmatrix}^T$

W = Positive-definite weighting matrix

$\mathbf{x}$ = State vector $= \begin{pmatrix} \mathbf{r}^T & \mathbf{v}^T & m \end{pmatrix}^T$

α = Mass consumption rate

Γ = Weighting on final time to go

ε = Tolerance level

ζ = Slack variable bounding thrust magnitude

ρ_1 = Thrust magnitude lower bound

ρ_2 = Thrust magnitude upper bound

$\boldsymbol{\tau}_\mathbf{c}$ = Commanded thrust vector

$\boldsymbol{\upsilon}_j$ = Vector defining convex state constraints

ϕ = Scalar objective function

$\boldsymbol{\psi}$ = Adjoint constraint equations

CDF = Cumulative Distribution Function

EDL = Entry, Descent, and Landing

EI = Entry Interface

GNC = Guidance, Navigation, and Control

MER = Mars Exploration Rovers

MSL = Mars Science Laboratory

PMF = Propellant Mass Fraction

SOCP = Second-order Cone Problem

TCM = Trajectory Correction Maneuver

TRN = Terrain Relative Navigation

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 is

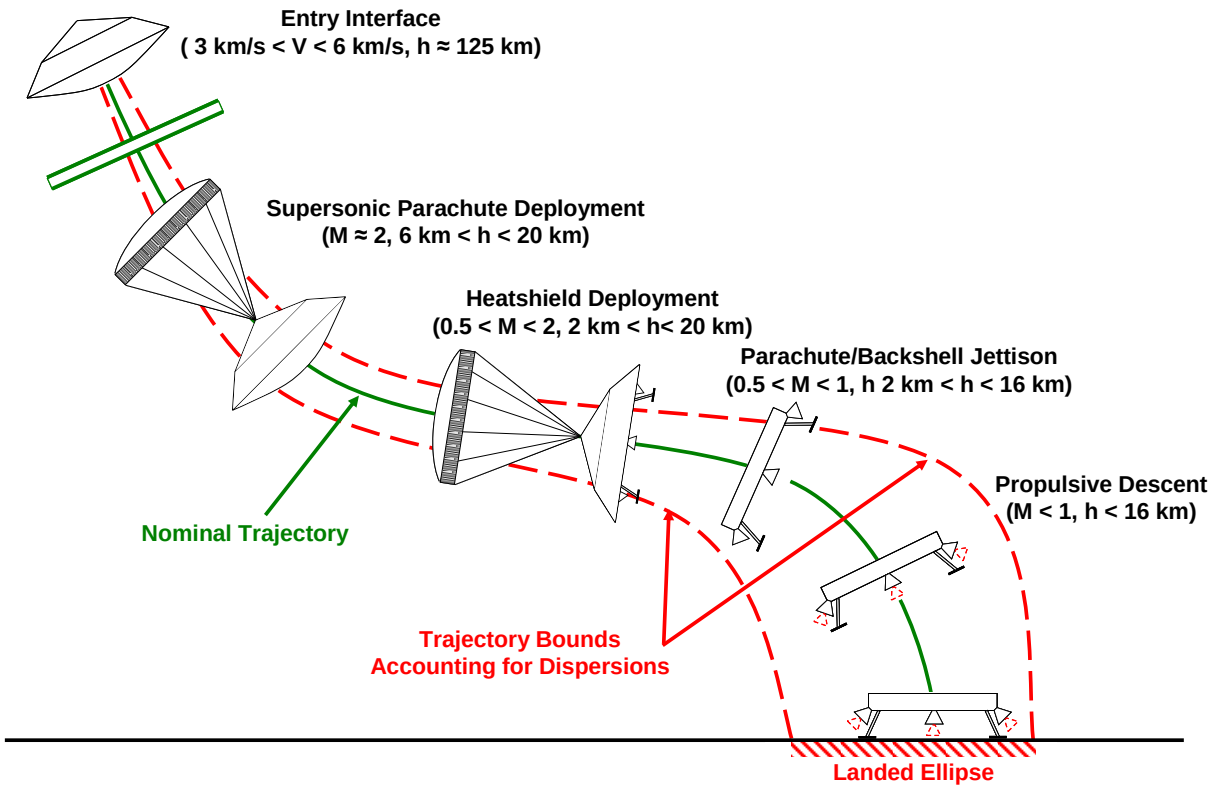

Figure 1. Typical Martian EDL sequence with uncertainty.

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 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 comprised 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 landing ellipse, as shown in Figure 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 2009, 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 MER, this landed ellipse accuracy improvement is largely the result of the inclusion of a modified Apollo hypersonic guidance algorithm which 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 further two-order of magnitude reduction to MSL's 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].

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 the ratio of the propellant's mass to the initial mass

$$PMF = \frac{m_{prop}}{m_0} \tag{1}$$

Terrain relative navigation (TRN) is also investigated by examining the effect of termination altitude and sampling frequency on TRN sensor performance as well as examining the effect of map-tie error on the overall landed vehicle's accuracy.

Table 1. Vehicle and state parameters[1,5].

Parameter	*Nominal*	*Distribution*	*Deviation (3σ or min/max)*
Entry Mass	2616 kg	Gaussian	±3 kg
Entry Flight Path Angle	-14°	Gaussian	±0.6°
Vehicle Diameter	4.5 m	--	--
Trim Angle of Attack	11°	Gaussian	±2°
Parachute Deploy Distance from Nominal	0 km	Uniform	8 km
Parachute Deploy Velocity	488 m/s	Gaussian	1.3 m/s
Parachute Deploy Flight Path Angle	-20°	Uniform	±0.2°
C_A Multiplier (Kn≥0.1)	1	Gaussian	5%
C_N Multiplier (Kn≥0.1)	1	Gaussian	10%
C_A Multiplier (M>10)	1	Gaussian	3%
C_N Multiplier (M>10)	1	Gaussian	5%
C_A Multiplier (0.8<M<5)	1	Gaussian	10%
C_N Multiplier (0.8<M<5)	1	Gaussian	8%
C_A Multiplier (M<0.8)	1	Gaussian	5%
Supersonic Parachute Diameter	19 m	--	--
Supersonic Parchute C_D	0.61	Uniform	±10%
Subsonic Parachute Diameter	19 m	--	--
Subsonic Parchute C_D	0.68	Uniform	±10%
Maximum Terminal Descent Engine Thrust	3047 N	Uniform	±5%
Minimum Terminal Descent Engine Thrust*	1142 N	Uniform	±5%
Terminal Descent Engine I_{SP}	220 s	Uniform	±0.67%

*Only used in second-order cone algorithm

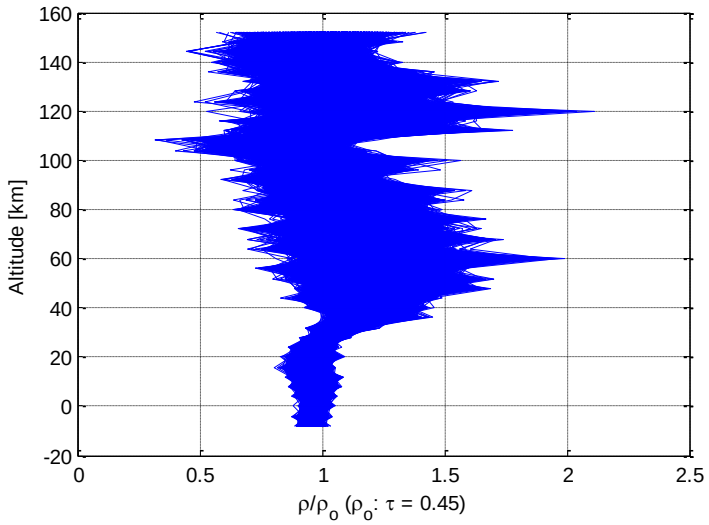

Figure 2. Density variation used in simulations.

A three degree-of-freedom simulator with bank modulation is used for trajectory propagation. The simulator incorporates modular capability allowing easy incorporation of the various guidance algorithms assessed. The nominal trajectory has the vehicle, a 4.5 m, Viking heritage, 70° sphere-cone, starting at parachute deployment at an altitude of 8 km MOLA and Mach 2 with a flight path angle of -20°. It is assumed for all but the hypersonic guidance study, that the vehicle's initial state at parachute deployment has a state dispersion similar to that of MSL, implying that a modified Apollo guidance algorithm is used throughout the hypersonic

phase of flight, except a constant parachute deployment altitude is assumed[5]. MarsGRAM 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, acceleration due to gravity, and density[6]. Figure 2 shows a plot of the nominal density variation used in the 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[5].

III. Propulsive Terminal Descent

Four different propulsive terminal descent algorithms were evaluated in this study. The first of which is a modified Apollo lunar module terminal descent algorithm which assumes linear variation of the vertical acceleration with quadratic variation in the remaining two axes and has no optimality conditions[7]. The second algorithm considered is a constrained gradient-based, indirect optimal control algorithm with iteration required to derive the control history[8]. The third algorithm, originally derived by D'Souza, is a fuel-optimal algorithm which assumes flight over a flat planet neglecting aerodynamic forces. These assumptions allow an analytic solution to be found which D'Souza showed to be optimal[9]. The fourth algorithm examined is a second-order cone formulation where 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 on-board the vehicle[10].

A. Modified Apollo Lunar Module Terminal Descent 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, crossrange, and altitude) relative to the target[7]. In equation form, that is to say that the acceleration in each direction is given by

$$a_i(t) = C_{0_i} + C_{1_i} t + C_{2_i} t^2 \tag{2}$$

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

$$v_i(t) = C_{0_i} t + \frac{C_{1_i}}{2} t^2 + \frac{C_{2_i}}{3} t^3 + v_{0_i} \tag{3}$$

$$r_i(t) = \frac{C_{0_i}}{2} t^2 + \frac{C_{1_i}}{6} t^3 + \frac{C_{2_i}}{12} t^4 + v_{0_i} t + r_{0_i} \tag{4}$$

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

$$\mathbf{r}(t=0) = \mathbf{r_0} \text{ and } \mathbf{v}(t=0) = \mathbf{v_0} \tag{5 a-b}$$

and final conditions

$$\mathbf{r}(t=t_f) = \mathbf{r_f},\ \mathbf{v}(t=t_f) = \mathbf{v_f}, \text{ and } \mathbf{a}(t=t_f) = \mathbf{a_f} \tag{6 a-c}$$

allows the solution for the coefficients in each axis to be solved. The resulting coefficients are given by

$$C_{0_i} = a_{f_i} - \frac{6}{t_{go}}\left(v_{f_i} + v_{0i}\right) + \frac{12}{t_{go}^2}\left(r_{f_i} - r_{0_i}\right) \tag{7}$$

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

$$C_{2_i} = \frac{6}{t_{go}^2} a_{f_i} - \frac{12}{t_{go}^3}\left(2v_{f_i} + v_{0_i}\right) + \frac{36}{t_{go}^4}\left(r_{f_i} - r_{0_i}\right) \tag{9}$$

By assuming a linear acceleration profile in the vertical axis (*i.e.*, setting C_2=0) the time-to-go, t_{go}, to be solved for analytically and is given by the expression

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

or

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

Thus, the commanded thrust vector is given by

$$\boldsymbol{\tau}_{\mathbf{C}} = m\left(\mathbf{a} - \mathbf{g}\right) \tag{12}$$

The primary advantage of this algorithm is that it is computationally non-complex and allows for the acceleration profile to be found for all time. However, the algorithm does not provide for conditions to obtain the fuel optimal solution 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 as a low altitude hover ensures that pinpoint accuracy is achieved.

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

$$J = \phi\left(\mathbf{x}(t_f), t_f\right) + \int_{t_0}^{t_f} L\left(\mathbf{x}(t), \mathbf{u}(t), t\right) dt \tag{13}$$

for a given set of system equations

$$\dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}, \mathbf{u}, t\right) \tag{14}$$

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[8]. 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 later 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 comprised of solely the integrated control vector, $\mathbf{u}(t)$

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

The solution algorithm for the gradient based method implemented for this study is as follows[8]:

1. Obtain the equations describing the motion of the vehicle, $\mathbf{f}(\mathbf{x},\mathbf{u},t)$
2. Determine the constraints for the problem, thrust magnitude and 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, Eq. (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 backwards in time the equations

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

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

   to obtain the influence functions and a matrix of influence functions.
6. Simultaneously with the backward integration of Step 5, compute the quantities

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

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

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

   where the matrix W is an arbitrary, time varying matrix that is positive-definite.
7. Choose values of $d\boldsymbol{\psi}$ that moves 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

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

where b is a weighting constant

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

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

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

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

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

$$t_f^{new} = t_f^{old} + dt_f \tag{25}$$

11. Iterate using steps 4 through 10 until $\boldsymbol{\psi}\left[\mathbf{x}(t_f), t_f\right] = \mathbf{0}$, $\left[\frac{d\phi}{dt} + \mathbf{v}^T\frac{d\boldsymbol{\psi}}{dt} + L\right]_{t=t_f} = 0$, and $I_{JJ} - I_{J\psi}I_{\psi\psi}^{-1}I_{\psi J} \le \varepsilon$, where ε is the acceptable tolerance

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

This iterative solution is advantageous as it finds a local minimum in the fuel consumption robustly and as accurately as the tolerance prescribed. However, it does suffer from being computationally intensive, requiring numerous iteration before convergence occurs, particularly if a poor initial solution is given. Additionally, the algorithm is dependent on numerical derivatives which increases the number of function calls dramatically depending on the scheme used to evaluate the derivatives.

C. Closed-form, Analytic, Fuel Optimal Control Algorithm

By assuming a planar, non-rotating planet with no atmosphere, D'Souza derived an analytic, unconstrained fuel-optimal propulsive terminal descent algorithm that meets the necessary and sufficient conditions for an optimal control law[9]. The problem described by D'Souza minimizes the performance index

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

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

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

where

$$\Delta\mathbf{r} = \begin{pmatrix} r_1 - r_{f_1} & r_2 - r_{f_2} & r_3 - r_{f_3}\end{pmatrix}^T \tag{28}$$

$$\Delta\mathbf{v} = \begin{pmatrix} v_1 - v_{f_1} & v_2 - v_{f_2} & v_3 - v_{f_3}\end{pmatrix}^T \tag{29}$$

$$\mathbf{g} = \begin{pmatrix} 0 & 0 & g \end{pmatrix}^T \tag{30}$$

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

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

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 closed-form, analytic algorithm has a clear computational advantage compared to 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 formation of the algorithm does not have any constraints on either the maximum thrust magnitude or 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. While requiring numerical integration and iteration, obtaining the proper increment on the time-to-go weighting was shown to be significantly less computationally intensive than the gradient method described previously.

D. Second-order Cone Algorithm

For propulsive terminal descent, the control space is, in general, non-convex due to a minimum allowable thrust magnitude, which none of the prior three methods described take into account. Due to this non-convex constraint, determining a control history that results in the global minimum with regards to PMF is not guaranteed. Açikmeşe and Ploen have shown that there exits a convex programming approach to the propulsive terminal descent problem which relaxes this non-convex constraint and, in turn, guarantees arrival at the global minimum in PMF[10]. 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 exits for 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 non-convex propulsive terminal descent guidance law problem can be posed as

$$\text{Minimize: } J = \int_{t_0}^{t_f} \left\| \boldsymbol{\tau}_\mathbf{C} \right\| dt$$

$$\text{Subject to: } \ddot{\mathbf{r}} = \mathbf{g} + \boldsymbol{\tau}_\mathbf{C}/m,\ \dot{m} = -\alpha\left\|\boldsymbol{\tau}_\mathbf{C}\right\|,\ 0 < \rho_1 < \left\|\boldsymbol{\tau}_\mathbf{C}\right\| \le \rho_2,\ r_3(t) \ge 0, \tag{32}$$

$$\left\| S_j\mathbf{x}(t) - \boldsymbol{\upsilon}_\mathbf{j} \right\| + \mathbf{c}_\mathbf{j}^T\mathbf{x}(t) + a_j \le 0,\ j = 1,...,n,\ \mathbf{r}(0) = \mathbf{r_0},\ \dot{\mathbf{r}}(0) = \dot{\mathbf{r}}_\mathbf{0},\ m(0) = m_0,\ \mathbf{r}(t_f) = \dot{\mathbf{r}}(t_f) = \mathbf{0}$$

where convex path constraints are included. The problem can be made convex, by reformulating it in terms of a new variable, ζ. This introduces an additional constraint to ensure that the new variable acts as a slack variable.

$$\text{Minimize: } J = \int_{t_0}^{t_f} \zeta(t)dt$$

$$\text{Subject to: } \ddot{\mathbf{r}} = \mathbf{g} + \boldsymbol{\tau}_\mathbf{C}/m,\ \dot{m} = -\alpha\|\boldsymbol{\tau}_\mathbf{C}\|,\ \|\boldsymbol{\tau}_\mathbf{C}\| \le \zeta(t),\ 0 < \rho_1 < \zeta(t) \le \rho_2,\ r_3(t) \ge 0 \tag{33}$$

$$\left\|S_j\mathbf{x}(t) - \boldsymbol{\upsilon}_\mathbf{j}\right\| + \mathbf{c}_\mathbf{j}^T\mathbf{x}(t) + a_j \le 0,\ j = 1,...,n,\ \mathbf{r}(0) = \mathbf{r_0},\dot{\mathbf{r}}(0) = \dot{\mathbf{r}}_\mathbf{0},m(0) = m_0,\ \mathbf{r}(t_f) = \dot{\mathbf{r}}(t_f) = \mathbf{0}$$

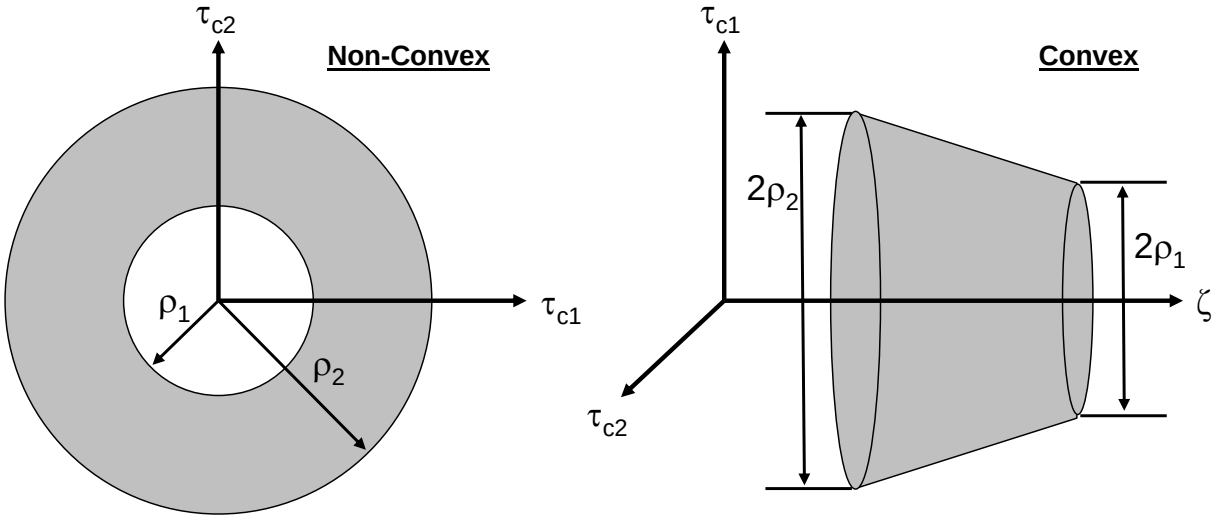

Figure 3. Non-convex thrust control space and convex thrust control space.

Graphically, this transformation of the non-convex control space to convex control space by the introduction of this slack variable is shown in Figure 3 for a two-dimensional case. In general, the solution to the non-convex problem is a feasible solution to the relaxed problem; however, the converse is not guaranteed to be true. However, Açikmeşe and Ploen have shown that the optimal solution found by the relaxed problem is also the optimal solution to the non-convex problem[10].

To implement this algorithm numerically, the continuous time problem needs to discretized. In their work, Açikmeşe and Ploen describe a change of variables transformation for Eq. (33) that leads to a continuous optimization problem, but one which has 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].

Remark: When implementing this algorithm, several existing software packages exist. In particular, SeDuMi was utilized for this study, which is a Matlab oriented software package that solves symmetric cone optimization problems, such as the problem posed here[11]. 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 Xia and Alizadeh[12,13].

E. Comparison of the Four Propulsion 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 comparison, a gravity turn from starting at Mach 0.9 (h = 1.6 km, v = 200 m/s, γ = -34°) 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 closed-form analytic guidance algorithm, the second-order cone algorithm, and the modified Apollo lunar module guidance algorithm. Additionally, two curves are shown in Figure 5 for the second-order cone algorithm—one which imposes a minimum thrust bound (1142±57 N) and one which does not. This figure effectively shows the propellant cost of not pulsing the engines during the terminal descent. Table 2 provides a quantitative and qualitative comparison between each of the four algorithms, where the qualitative metrics were assessed on a relative basis to one another. In this table, the minimum thrust bound was eliminated from the second-order cone algorithm so that a fair comparison on the optimality can be made.

Table 2. Comparison of the four propulsive descent algorithms.

	Modified Apollo Lunar Module Algorithm	*Gradient Based Optimal Control Algorithm*	*Closed-Form, Analytic Algorithm*	*Second-order Cone Algorithm*
Optimality at 99% C.I. [PMFmin/PMF]	0.36	1.00	0.97	0.99
Equations of Motion Evalutations	1	327	8	163
Robustness	Poor	Good	Moderate	Good
Ease of Implementation	Good	Poor	Good	Moderate
Applicability to Flight	Good	Moderate	Moderate	Good
Numerical Stability	Good	Moderate	Moderate	Good

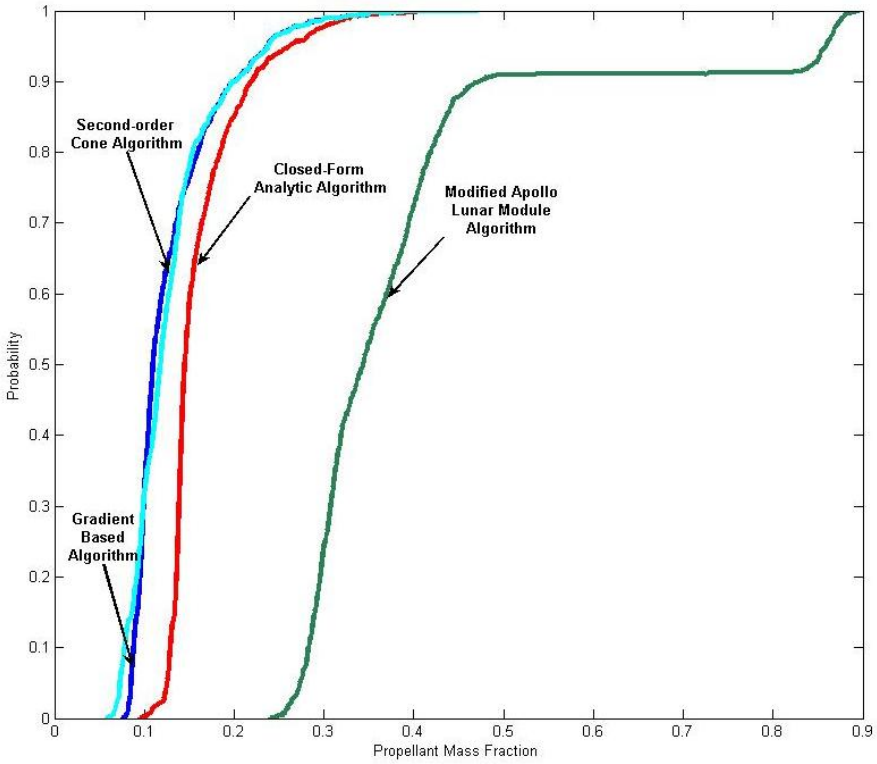

Figure 4. CDFs of the PMFs for each of the algorithms without a minimum thrust constraint.

At the 99% confidence level, there is little discernable difference in the PMF for the gradient based optimal algorithm, second-order cone algorithm, and the closed-form analytical. Even at lower confidence levels, the PMF is only a few percent different. The PMF for the modified Apollo lunar module algorithm is significantly higher (0.87 at the 99% confidence interval), which is unachievable in a realistic design. This significant difference in PMF is due to the lack of robustness to the environmental parameters variations seen since 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).

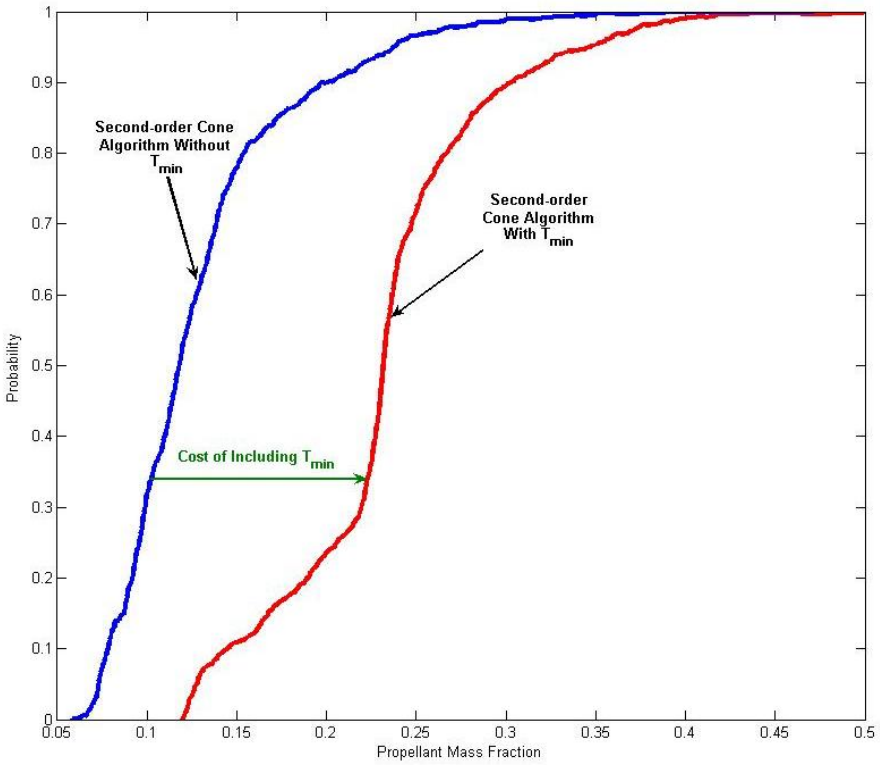

Figure 5. CDF of the PMF for the second-order cone algorithm with the cost of the minimum thrust constraint.

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 to 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 afforded by the closed-form analytic algorithm and makes this algorithm the preferred algorithm for further studies conducted as well as for conceptual design.

For flight applications, the second-order cone algorithm with a lower thrust bound is preferred. For flight systems, the cycling of engines on and off, which can result from not specifying a minimum thrust constraint, is not desired as it increases the probability of loss of mission. It can be seen that both 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. As seen in Table 2, the number of evaluations in the equations of motion for the second-order cone algorithm is half that of the gradient based algorithm. This results in a run-time on the order of one-quarter that of the gradient based algorithm due to the polynomial convergence properties of the SOCP, making the SOCP preferable in conceptual studies as well to the gradient based optimal guidance law.

F. Propulsive Descent Initiation

Using the analytic fuel-optimal guidance law, a design space investigation was performed of the altitude, velocity, downrange, and flight path angle space to determine if a combination of those sensible parameters yields a minimum in the PMF required to achieve pinpoint landing. The minimum PMF observed for a representative case in the downrange-altitude space is shown in Figure 6(a), while Figure 6(b) shows the variation of PMF for changes in velocities for different initial flight path angles at parachute deployment. These two representative figures help to demonstrate 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.

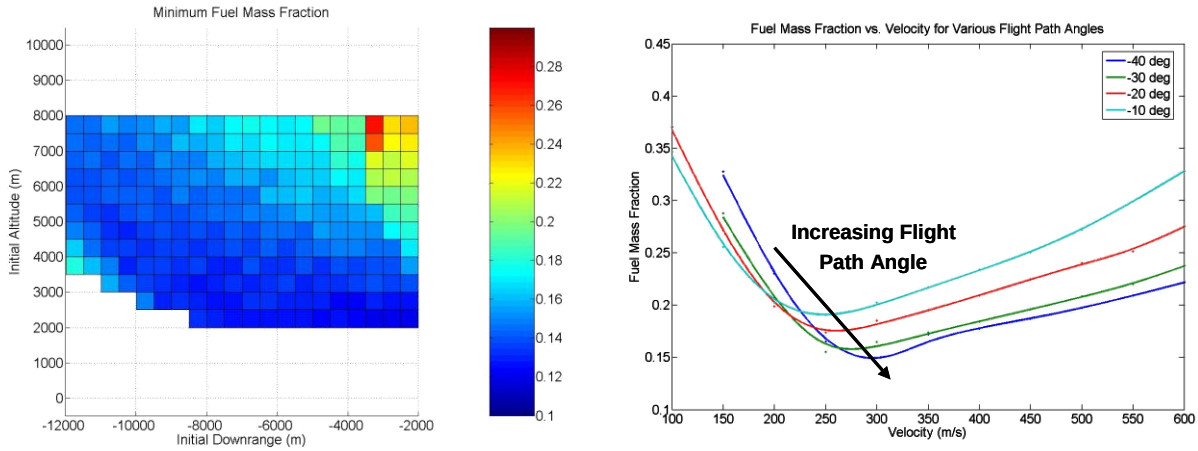

Figure 6. Design space slice for propulsive terminal descent.

IV. Subsonic Guided Parachute Performance

A subsonic steerable parachute was included in the entry vehicle system used for terminal descent. Several other studies have investigated the inclusion of subsonic parachutes in the past, these include one performed by Mitcheltree, *et al*, in which a test program is laid out for Earth qualification of the parachute and one performed by Witkowski, *et al*, where the inclusion of a subsonic chute was investigated[14,15]. For this trade, 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 towards a reference trajectory using a performance index to minimize the amount of time it takes to arrive on the reference[16]. 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.

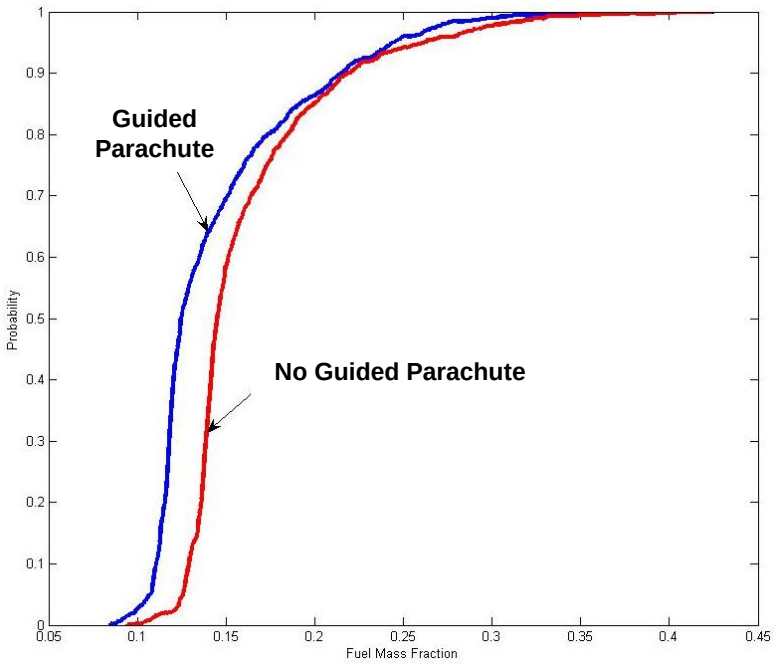

Figure 7. CDFs of the benefit of a subsonic guided parachute.

As shown in Figure 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 with a significant fraction of the Martian entries (*e.g.*, a CDF above 80%) does not allow for the subsonic guided parachute to be inflated for a significant length of time, as would occur on 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 once on the trajectory, which could be a dominant factor as a spiraling trajectory about the reference will increase the energy and increase the propulsive force required to negate it.

V. Hypersonic Guidance Performance

The ramifications of the inclusion of a hypersonic guidance algorithm was evaluated by examining the semi-major 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 Figure 8, these results demonstrate a marginal PMF reduction below a position delivery error at 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.

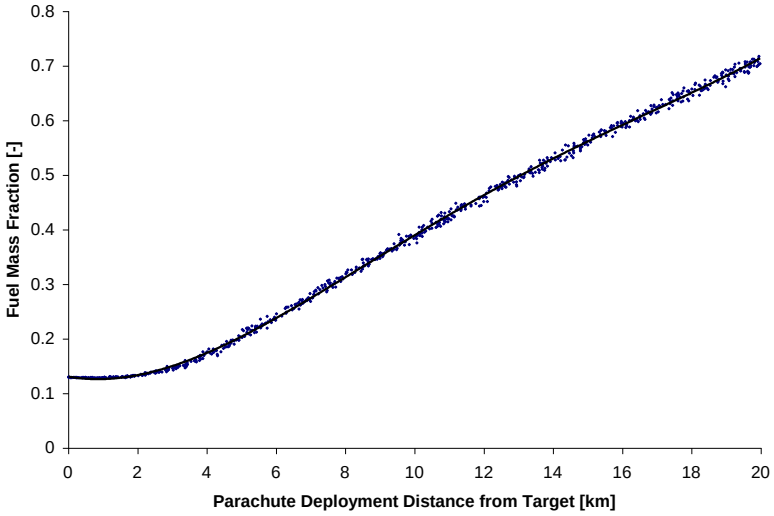

Figure 8. PMF for supersonic parachute deployment semi-major errors.

VI. Terrain Relative Navigation Sensor Performance

The impact of terrain relative navigation 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 3-dimensional TRN sensor. This sensor suite has the specifications shown in Table 3. Additionally, a covariance matrix at EI-10 minutes 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 navigational knowledge error of approximately 200 m was seen at parachute deployment. Thus, pinpoint landing is precluded.

Table 3. Navigation sensor data.

Sensor	*Altitude Active [km]*	*Termination Altitude [km]*	*Bias*	*Sensor Realization Error*	*Noise (1σ)*
Accelerometer	N/A	N/A	30 μg	66 ppm	5 μg/Hz$^{1/2}$
Gyroscope	N/A	N/A	0.02 deg/hr	1.6 ppm	50 μrad/Hz$^{1/2}$
Radar	20	0.01	-0.1 m	0 m	5 m
Velocimeter	2	0.01	(-0.14, 0.07, -0.06) m/s	(5.79×10-5, 5.79×10-5, 5.79×10-5) rad	0.5 m/s
3D Terrain Relative Navigation Sensor	8	2.5	(4.3, 28.7, -15.7) m	(0, 0, 0) m	42 m

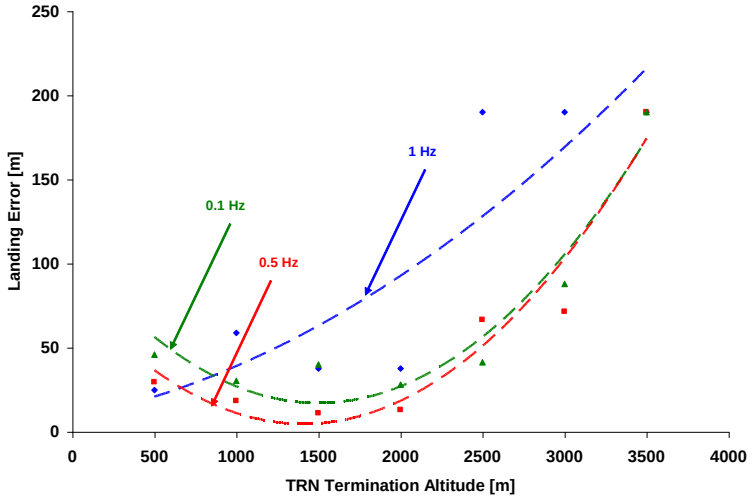

Figure 9. Landed accuracy for various TRN termination altitudes and sampling frequencies.

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 landed error variation for the nominal case is seen in Figure 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 Hz and 0.5 Hz case at approximately 1.5 km. Sensor data below this altitude may be more inaccurate that the known state, which, in turn, leads to the overall knowledge error decreasing. This is not seen in the 1 Hz sampling rate as more accurate data is obtained between 1 and 1.5 km improving the knowledge of the known state.

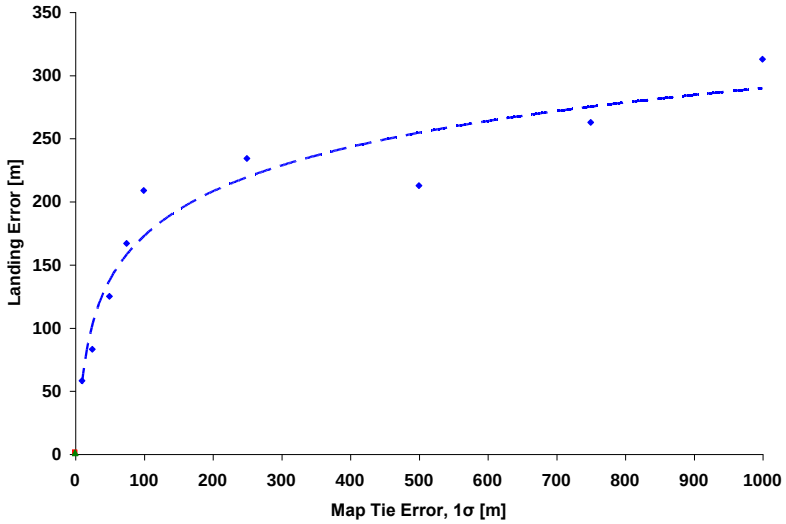

Figure 10. Map-tie error impact on landing uncertainty.

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 Figure 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 compared the performance of four propulsive terminal descent algorithms for a sub-100 m landing accuracy application at Mars. An iterative propellant optimal guidance law was implemented. However, it was seen that a near-optimal guidance law which assumes a flat, atmosphere-free planet was of sufficient accuracy for conceptual design, while being 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-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 performed demonstrated that driving hypersonic guidance requirements to an accuracy below 3 km results in marginal performance gains. In addition, it was shown that the use of a subsonic guided parachute does not significantly decrease the propellant mass fraction required for pinpoint landing, while the added system complexity of a subsonic guided parachute would have to be seriously considered. This investigation also demonstrates that navigational uncertainty is the principal driving factor precluding pinpoint landing at Mars with map-tie error needing to be driven below 25 m and TRN sensor termination altitude needing to be driven below 1 km for sub-100 m landed accuracy.

The results of this study were obtained assuming a parachute deployment ellipse with similar semi-major axis length to MSL at a constant altitude. The altitude variations that will inevitably result from the system and environment dispersions should be accounted for in order to further add fidelity to these results. The initiation of the propulsive descent for this work started at Mach 0.9; however, a trade between the PMF and the initiation state relative to the target exists and should be explored in future work. A third area of potential extension of this work is the influence of various hypersonic phase guidance algorithms (*e.g.*, a predictor-corrector algorithm or the modified Apollo algorithm) and navigation performance on the propulsive terminal descent phase. For instance, does a predictor-corrector algorithm that targets an optimal parachute deployment altitude for the terminal descent guidance algorithm strongly impact the performance of the system?

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, Big Sky, Montana, March 2006.

[2]Braun, R.D. and Manning, R.M., "Mars Exploration Entry, Descent, and Landing Challenges," *Journal of Spacecraft and Rockets*, Vol. 44, No. 2, pp. 310-323, 2007.

[3]Mendeck, G.F., and Carman, G.L., "Guidance Design for Mars Smart Landers Using the Entry Terminal Point Controller," Paper No. AIAA 2002-4502, AIAA Atmospheric Flight Mechanics Conference, 5-8 August 2002, Monterey, California.

[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, Jun. 1998.

[5]Striepe, S., *et al*, "Mars Science Laboratory Simulations for Entry, Descent, and Landing," *Journal of Spacecraft and Rockets*, Vol. 43, No. 2, pp. 311-323, 2006.

[6]Duvall, A., *et al*, "Global Reference Atmospheric Model (GRAM) Series for Aeroassist Applications," Paper No. AIAA 2005-1239, AIAA Aerospace Sciences Meeting and Exhibit, January 2005, Reno, Nevada.

[7]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, August 2002, Monterey, California.

[8]Bryson, A., and Ho, Y., *Applied Optimal Control*, Hemisphere Publishing Company, 1963.

[9]D'Souza, C., "An Optimal Guidance Law for Planetary Landing." Paper No. AIAA 97-3709, 1997.

[10]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, pp. 1353-1366, 2007.

[11]Sturm, J., "Using SeDuMi 1.02, a MATLAB Toolbox for Optimization Over Symmetric Cones," *Optimization Methods and Software*, Vol. 11, No. 1, pp. 625-653, 1999.

[12]Lobo, M., et al, "Applications of Second-Order Cone Programming," *Linear Algebra and its Applications*, Vol. 284, No. 1, pp. 193-228, 1998.

[13]Xia, Y., and Alizadeh, F., "The Q method for second order cone programming," *Computers & Operations Research*, Vol. 35, pp. 1510-1538, 2008.

[14]Mitcheltree, R., *et al*, "High Altitude Test Program for a Mars Subsonic Parachute", *18th AIAA Aerodynamic Decelerator Conference*, AIAA 2005-1659, Munich, Germany, May 2005.

[15]Witkowski, A., Machalick, A., and Taeger, Y., "Mars Subsonic Parachute Technology Task System Overview", *18th AIAA Aerodynamic Decelerator Conference*, AIAA 2005-1657, Munich, Germany, May 2005.

[16]Yakimenko, O., *et al*, "On Control of Autonomous Circular Parachutes," Paper No. 2002-4753, AIAA Atmospheric Flight Mechanics Conference, 5-8 August 2002, Monterey, California.