Enhancements on the Convex Programming Based Powered Descent Guidance Algorithm for Mars Landing

Behçet Açıkmeşe*, Lars Blackmore, Daniel P. Scharf, and Aron Wolf

In this paper, we present enhancements on the powered descent guidance algorithm developed for Mars pinpoint landing. The guidance algorithm solves the powered descent minimum fuel trajectory optimization problem via a direct numerical method. Our main contribution is to formulate the trajectory optimization problem, which has nonconvex control constraints, as a finite dimensional convex optimization problem, specifically as a finite dimensional second order cone programming (SOCP) problem. SOCP is a subclass of convex programming, and there are efficient SOCP solvers with deterministic convergence properties. Hence, the resulting guidance algorithm can potentially be implemented onboard a spacecraft for real-time applications. Particularly, this paper discusses the algorithmic improvements obtained by: (i) Using an efficient approach to choose the optimal time-of-flight; (ii) Using a computationally inexpensive way to detect the feasibility/infeasibility of the problem due to the thrust-to-weight constraint; (iii) Incorporating the rotation rate of the planet into the problem formulation; (iv) Developing additional constraints on the position and velocity to guarantee no-subsurface flight between the time samples of the temporal discretization; (v) Developing a fuel-limited targeting algorithm; (vi) Initial result on developing an onboard table lookup method to obtain almost fuel optimal solutions in real-time.

I. Introduction

Our earlier papers [1,2] presented a convex programming algorithm to solve the powered descent guidance (PDG) problem for Mars pinpoint landing. The PDG algorithm solves the powered descent minimum fuel trajectory optimization problem via a direct numerical method. Our main contribution in [1] is the formulation of the trajectory optimization problem, which has nonconvex control constraints, as a finite dimensional convex optimization problem, specifically as a finite dimensional second order cone programming (SOCP) problem. SOCP is a subclass of convex programming, and there are efficient SOCP solvers with deterministic convergence properties. Consequently, the resulting guidance algorithm can potentially be implemented onboard a spacecraft for real-time applications. It is also shown in Ref. 3, that when compared with more traditional polynomial based guidance algorithms such as the ones used in Refs 4–6, the optimal trajectories obtained from this PDG algorithm can lead to significant fuel savings as well as significant improvements in the lateral divert capability. In this paper, we present enhancements on this algorithm and the status of the algorithmic development towards an onboard PDG algorithm.

In order to ensure a safe landing on Mars, landing sites for first-generation Mars missions such as Viking and Mars Pathfinder were determined by considering the scientific merit of various sites as well as the overall terrain quality (e.g., slope, roughness). As these missions were exploratory in nature, the exact location of the landing site within the error ellipse was not critical. For example, landing accuracy was approximately 150km for Mars Pathfinder and 35km for the Mars Exploration Rovers. The Mars Surface Laboratory mission scheduled for launch in 2009 will further increase landing accuracy resulting in a delivery of the lander to within 10km. In next generation Mars missions the need to perform pinpoint landing will be required. Further improvements in landing accuracy require enhancements to heritage flight systems and are discussed in detail in Ref. 7. For example, it may be required to land next to scientifically interesting targets located within hazardous terrain, or to land in close proximity to other pre-positioned surface assets such as a rover from a previous mission. Further, human missions will require the crew to land near pre-positioned cargo or fuel.

Mars pinpoint landing involves an entry phase through the Mars atmosphere, a phase of descent with a parachute, and then the final phase of *powered descent*, which is initiated with the parachute cutoff. The PDG problem for

*Authors are with Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Dr., M/S 198-326, Pasadena, CA 91109, USA. Corresponding author email: Behcet.Acikmese@jpl.nasa.gov.

pinpoint landing is defined as finding the fuel optimal trajectory that takes a lander with a given initial state (position and velocity) to a prescribed final state in a uniform gravity field, with magnitude constraints on the available net thrust, and various state constraints. A simplified, one-dimensional version of this problem is also known in the optimal control literature as the *soft landing problem* [8], and its solutions have well known characterizations (see pp. 28-33 in Ref. 9). However, to the best of our knowledge, a closed form solution of the thrust profile is not available for the general three dimensional case with realistic state and control constraints. A trajectory based on a quartic polynomial in time was used during the Apollo mission [4]. Various other approaches to obtain both numerical and approximate solutions of the pinpoint landing terminal guidance problem have been described over the last few years. More recently, in Ref. 10 the first order necessary conditions for the problem are developed and it is shown that the optimal thrust profile has a maximum-minimum-maximum structure. Numerical solutions are obtained for various initial conditions by converting the guidance problem into a constrained non-convex parameter optimization problem using both direct collocation and direct multiple shooting methods. In Ref. 5 Legendre pseudospectral methods are used to develop a numerical solution to the pinpoint landing guidance problem. In Ref. 11 an approximate analytical solution is developed by solving a related optimal control problem that does not utilize a minimum fuel cost functional.

Direct numerical methods for trajectory optimization are attractive because explicit consideration of the necessary conditions (adjoint equations, transversality conditions, maximum principle) are not required [12]. The infinite dimensional optimal control problem is directly converted into a finite dimensional parameter optimization problem [13–15], which is then solved via a nonlinear programming method. However, real-time onboard solution of a nonlinear program via a general iterative algorithm may not be desirable without explicit knowledge of the convergence properties of the algorithm. As Mars pinpoint landing requires real-time onboard computation of the optimal trajectory, it is essential to exploit the structure of the problem to design algorithms with guaranteed convergence to the global optimum with a deterministic convergence criterion. This lead us, in our prior work [1, 2], to formulate the problem in a convex optimization [16, 17] framework. Specifically, we formulated the resulting parameter optimization problem as a second order cone programming problem (SOCP) that is a subclass of convex programming [18]. SOCP problems have low complexity, and they can be solved in polynomial time. There exist algorithms, such as interior-point methods [19–21], that compute an optimal solution with a deterministic stopping criteria, and with prescribed level of accuracy. Therefore, numerical SOCP solution algorithms are very well-suited for real-time onboard computations.

We first summarize the formulation of the problem given in [1], then the formulation of the additional constraints to prevent inter-sample subsurface flight and the incorporation of the rotation rate of the planet into the formulation of the problem are presented. The resulting optimization problem for PDG is an SOCP for each time-of-flight (time from the start to the end of the maneuver). Hence an optimal solution can be obtained for each time-of-flight by finding the global optimum solution of this SOCP. An outer iteration, which is a simple line search, must be applied to find the optimal time-of-flight corresponding to the minimum fuel consumption. This can become quite costly since the optimal fuel computation for each time-of-flight requires the solution the SOCP. We devised a method to approximate the optimal time-of-flight and it only requires the solution of the SOCP four times. This approach is motivated from the observations that the time-of-flight versus the optimal fuel (per time-of-flight) relation can be approximated by a quadratic function in a close neighborhood of the minimum feasible time-of-flight and that the optimal time-of-flight is also in this vicinity of the minimum time-of-flight. Therefore, once a quadratic is uniquely fitted by using three time-of-flight values, the time that minimizes the quadratic can be obtained easily, as well as the optimal minimum fuel trajectory corresponding to this time-of-flight in a fourth solution of the SOCP.

However a method to choose optimal time-of-flight requires the knowledge of minimum time-of-flight when a feasible trajectory exists or the detection of the infeasibility of the problem. In Ref. 8, a computationally inexpensive algorithm is given to determine the minimum fuel trajectories for the one-dimensional version of the PDG problem, that is along the vertical channel only. We utilize this solution to develop a method to detect whether or not the PDG problem is feasible. If it is feasible then we solve for the fuel optimal trajectory.

We also provide an approach for fuel-limited targeting, which is necessary when there is insufficient fuel to reach the landing target. In this case, we would like to ensure safe landing, while minimizing the final distance to the target. We show that this problem can also be solved by performing SOCP optimizations, and is hence amenable to onboard implementation.

Finally we present an initial table lookup method concept to approximately solve the SOCP resulting from the PDG problem. This approach is numerically very simple to implement and execute onboard and it does not require significant onboard computational resources. Consequently it is a very promising approach towards developing an onboard PDG algorithm that generates almost fuel optimal solutions on a real-time processor.

II. Problem Formulation

The minimum fuel powered descent trajectory optimization problem associated with the pinpoint landing is described as follows (see [1] for the details):

Problem 1.

$$\max_{t_f,\, T_c(\cdot)} m(t_f) = \min_{t_f,\, T_c(\cdot)} \int_0^{t_f} \|T_c(t)\|\, dt \qquad \text{subject to}$$
$$\ddot{r}(t) = g + T_c(t)/m(t), \qquad \dot{m}(t) = -\alpha\|T_c(t)\|$$
$$0 < \rho_1 \le \|T_c(t)\| \le \rho_2,$$
$$\|r_d(t)\| \le \beta r_v(t)$$
$$m(0) = m_{wet}, \quad m(t_f) \ge 0, \quad r(0) = r_0, \quad \dot{r}(0) = \dot{r}_0, \quad r(t_f) = \dot{r}(t_f) = 0.$$

In the above problem,

$$\alpha = \frac{1}{I_{sp} g_e \cos\phi}, \quad \rho_1 = nT_1 \cos\phi, \quad \rho_2 = nT_2 \cos\phi$$

where ϕ is the cant angle for the thrusters, n is the number of thrusters, I_{sp} is the specific impulse, m_{wet} is the mass of the lander with initial available fuel, g_e is the Earth's gravitational constant, T_1 and T_2 are lower and upper limits of the thrust force that can be provided by each thruster, $r_0 \in \mathbb{R}^3$ and $\dot{r}_0 \in \mathbb{R}^3$ are initial position and velocity vectors for the lander relative to the target landing point at the start of powered descent maneuver, $\beta = \tan\gamma$ where $\gamma \in (0, \pi/2)$ is the glide slope angle, $r_v \in \mathbb{R}$ is the altitude and $r_d \in \mathbb{R}^2$ is downrange position vector of the lander such that $r = [r_v,\, r_d^T]^T$. Note that more general state constraints (constraints on the position and velocity vectors such as bounds on the velocity magnitude) can be imposed in this formulation [1], but we only impose the glide slope constraint in this paper for simplicity.

Remark 1. In Problem 1, the inequality $0 < \rho_1 \le \|T_c(t)\| \le \rho_2$ defines nonconvex constraints on the control input. $\diamond$

In [1], it is shown that the following relaxed version of Problem 1, which is a convex optimization problem, can be solved to obtain the optimal solution of the PDG problem.

Problem 2.

$$\min_{t_f,\, T_c(\cdot),\, \Gamma(\cdot)} \int_0^{t_f} \Gamma(t)\, dt \quad \text{subject to}$$
$$\dot{m}(t) = -\alpha\Gamma(t), \tag{1}$$
$$\|T_c(t)\| \le \Gamma(t), \tag{2}$$
$$0 < \rho_1 \le \Gamma(t) \le \rho_2, \tag{3}$$
$$\ddot{r}(t) = g + T_c(t)/m(t),$$
$$\|r_d(t)\| \le \beta r_v(t)$$
$$m(0) = m_{wet}, \quad m(t_f) \ge 0, \quad r(0) = r_0, \quad \dot{r}(0) = \dot{r}_0, \quad r(t_f) = \dot{r}(t_f) = 0.$$

The following lemma describes the main technical result that lead to the PDG algorithm (see [1]).

Lemma 1. *Consider a solution of Problem 2 given by* $(t_f^*, T_c^*(\cdot), \Gamma^*(\cdot))$. *Then,* $(t_f^*, T_c^*(\cdot))$ *is also a solution of Problem 1 and* $\|T_c^*(t)\| = \rho_1$ *or* $\|T_c^*(t)\| = \rho_2$ *for* $t \in [0,\, t_f^*]$.

Lemma 1 implies that the main nonconvex constraint on the thrust magnitude in Problem 1 is convexified by replacing it with (2) and (3) in Problem 2 via the introduction of a scalar slack variable Γ. Furthermore, Lemma 1 presents a relation between the optimal solutions of Problem 1 and Problem 2. It states that if there exists an optimal solution for Problem 2, then there also exists one for Problem 1 and it can be obtained directly from the optimal solution of Problem 2. Also, note that the set of admissible controls of Problem 2 strictly contains the set of admissible

controls of Problem 1. Consequently, one can solve Problem 2 with convex control constraints to obtain the solution of Problem 1 with non-convex control constraints. See Figure 1 for an illustration of the constraint convexification in a two-dimensional control space: The nonconvex constraint set for the control input, represented by the shaded annulus, is mapped into a convex set, represented by the cone, via the slack variable Γ.

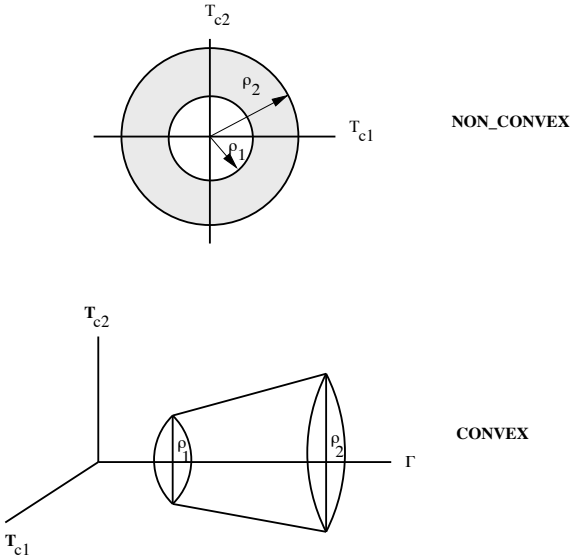

Figure 1. Convexification of Thrust Magnitude Constraint

In Problem 2 the state and constraints define convex feasible regions. However the dynamics are nonlinear due to the time variation of the mass due to thrusting. This will lead to nonlinear equality constraints in the discretized version of the problem, which will be solved numerically. Again in [1], it is shown that this issue can be resolved via a change of variables to obtain the following form of the PDG problem:

Problem 3.

$$\min_{t_f,\, u(\cdot),\, \sigma(\cdot)} \int_0^{t_f} \sigma(t)\, dt \qquad \text{subject to}$$

$$\ddot{r}(t) = u(t) + g \qquad \dot{z}(t) = -\alpha\sigma(t)\,,$$

$$\|u(t)\| \le \sigma(t)$$

$$\mu_1(t)\left[1 - (z(t) - z_0(t)) + \frac{(z(t) - z_0(t))^2}{2}\right] \le \sigma(t) \le \mu_2(t)\left[1 - (z(t) - z_0(t))\right],$$

$$z_0(t) \le z(t) \le \ln(m_{wet} - \alpha\rho_1 t)\,,$$

$$\|r_d(t)\| \le \beta r_v(t)$$

$$z_0(t) = \ln(m_{wet} - \alpha\rho_2 t), \quad \mu_1(t) = \rho_1 e^{-z_0(t)}, \quad \mu_2(t) = \rho_2 e^{-z_0(t)}$$

$$z(0) = \ln m_{wet}, \quad r(0) = r_0, \quad \dot{r}(0) = \dot{r}_0, \quad r(t_f) = \dot{r}(t_f) = 0.$$

In the above problem $z = \ln m$, $u = T_c/m$, and $\sigma = \Gamma/m$. For a given time-of-flight t_f, Problem 3 has all its equality and inequality constraints defining convex feasible regions in the state and control space and it has a convex cost function. Therefore, it can be converted into a finite dimensional convex optimization problem, particularly into an SOCP, after an appropriate discretization. Once the numerical algorithm is devised to solve the optimal fuel trajectory design problem for each t_f, a line search can be applied to find the "best" time-of-flight t_f^* [1, 2].

III. Discretization of Problem 3

This section describes the discretization of the optimal control problem 3. Our discretization is based on piecewise linear control input on a number of time intervals with equal length. More precisely, given $\Delta t > 0$ and the corresponding time instances $t_k = k\Delta t,\ k = 0,...,N$,

$$\begin{aligned} u(t) &= u_k + (u_{k+1} - u_k)\tau \\ \sigma(t) &= \sigma_k + (\sigma_{k+1} - \sigma_k)\tau \end{aligned} \qquad \tau = \frac{t - t_k}{\Delta t}, \quad \text{for } t \in [t_k, t_{k+1}), \qquad k = 0,...,N-1.$$

Note that the above characterization of control input results in piecewise cubic position trajectories. This discretization together with the imposition of the control and state constraints at the prescribed time instances $t_k,\ k = 0,...,N$ lead to the following finite-dimensional SOCP for a given t_f:

Problem 4.

$$\min_{u_0,...,u_N,\ \sigma_0,...,\sigma_N} -z_N \qquad \text{subject to, for } k = 0,\ldots,N,$$

$$r_{k+1} = r_k + \frac{\Delta t}{2}(\dot r_k + \dot r_{k+1}) + \frac{\Delta t^2}{12}(u_{k+1} - u_k)$$

$$\dot r_{k+1} = \dot r_k + \frac{\Delta t}{2}(u_k + u_{k+1}) + g\Delta t$$

$$z_{k+1} = z_k - \frac{\alpha \Delta t}{2}(\sigma_k + \sigma_{k+1}),$$

$$\|u_k\| \le \sigma_k$$

$$\mu_{1,k}\left[1 - (z_k - z_{0,k}) + \frac{(z_k - z_{0,k})^2}{2}\right] \le \sigma(t) \le \mu_{2,k}\left[1 - (z_k - z_{0,k})\right],$$

$$z_{0,k} = \ln(m_{wet} - \alpha\rho_2 k\Delta t), \quad \mu_{1,k} = \rho_1 e^{-z_{0,k}}, \quad \mu_{2,k} = \rho_2 e^{-z_{0,k}}$$

$$z_{0,k} \le z_k \le \ln(m_{wet} - \alpha\rho_1 t),$$

$$\|r_{dk}\| \le \beta r_{vk}$$

$$z_0 = \ln m_{wet}, \quad r(0) = r_0, \quad r_N = \dot r_N = 0, \quad N\Delta t = t_f.$$

Once we have a procedure to compute the fuel optimal trajectory for a given time-of-flight, t_f, we use the Golden search algorithm [22] to determine the time-of-flight with the least fuel need, t_f^*. This line search algorithm is motivated by the observation that the minimum fuel is a unimodal function of time with a global minimum; see Figure 2 for a typical t_f versus fuel plot. The lower and upper bounds on t_f line search are determined by utilizing the vertical channel only solution given by [8]. We also utilize this vertical channel only solution to develop a method to detect whether or not the PDG problem is feasible. If it is feasible then we solve for the fuel optimal trajectory.

IV. Avoiding Inter-Sample Glide Slope Constraint Violation due to Time Discretization and Mars Rotation

One issue that arises due to time discretization is inter-sample subsurface flight. Since the glide slope constraint is imposed only at a finite number of time instances $t_k = k\Delta t,\ k = 0,...,N$, there is no theoretical guarantee that there will be no violation of this constraint between the time sample, that is, when $t \in (t_k, t_{k+1}),\ k = 0,...,N-1$. Particularly this lack of guarantee can lead to subsurface flight close to the end of the maneuver. Here we impose additional constraints to make sure that there is no subsurface flight at any time during the flight.

We first formulate a condition that guarantees the satisfaction of the glide slope over all time when the glide slope is imposed by a single plane. The glide slope constraint can be approximated via a plane in the following generic form

$$n^T r(t) \ge 0, \qquad \forall t \in [0, t_f],$$

where the vector $n \in \mathbb{R}^3$ defines a plane approximating the glide slope. The glide slope constraints at the discrete time instances already guarantees that

$$n^T r_k \ge 0, \qquad k = 0,\ldots,N.$$

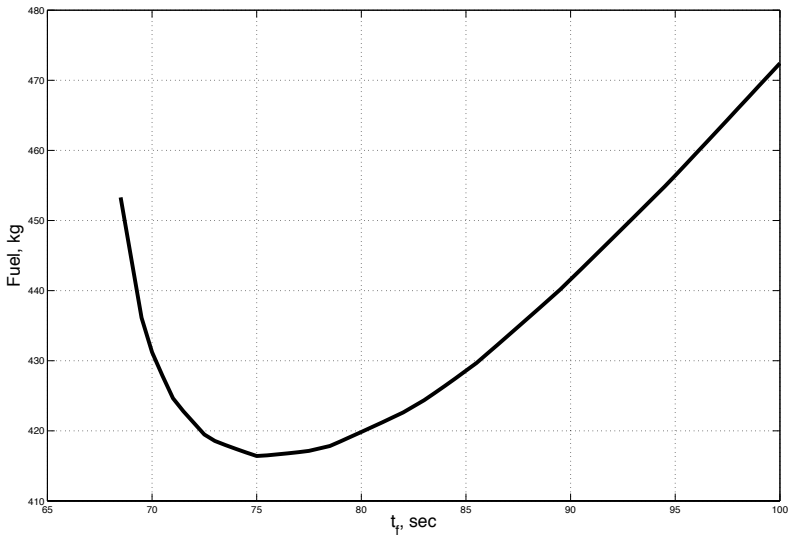

Figure 2. Minimum fuel requirement as a function of the time-of-flight

Now consider the following scalar valued function defined over any time interval $[t_k,\ t_{k+1}]$

$$h(\tau) = n^T r(\tau), \qquad \tau = \frac{(t - t_k)}{\Delta t}.$$

Clearly $h(0) \geq 0$ and $h(1) \geq 0$. Next we derive a condition that guarantees $h(\tau) \geq 0$ for all $t \in (0, 1)$. Since the position trajectory r is a cubic function of time on $t \in [t_k,\ t_{k+1}]$, we have

$$h(\tau) = a\tau^3 + b\tau^2 + c\tau + d$$

where:

$$a = \frac{n^T(u_{k+1} - u_k)\Delta t^2}{6}, \quad b = \frac{n^T(u_k + g)\Delta t^2}{2}, \quad c = n^T \dot{r}_k \Delta t, \quad d = n^T r_k.$$

This implies that:

$$\frac{dh}{d\tau} = 3a\tau^2 + 2b\tau + c,$$

and hence:

$$h(\tau) = \frac{dh}{d\tau}\frac{\tau}{3} + \frac{1}{3}b\tau^2 + \frac{2}{3}c\tau + d.$$

Since h is continuously differentiable on $[0,\ 1]$, it reaches a minimum either at the end points or in the interior of the domain. Since $h(0) \geq 0$ and $h(1) \geq 0$, if the minimum of h is at one of the end points then $h(t) \geq 0$ for all $t \in [0,\ 1]$. Otherwise, if the minimum occurs at $\tau_* \in (0,\ 1)$ then:

$$\frac{dh}{d\tau}(\tau_*) = 0 \quad \Rightarrow \quad h(\tau_*) = \frac{1}{3}b{\tau_*}^2 + \frac{2}{3}c\tau_* + d.$$

Hence, letting:

$$f(\tau) := \frac{1}{3}b\tau^2 + \frac{2}{3}c\tau + d,$$

if $f(\tau) \geq 0$ for all $\tau \in (0,\ 1)$, then $f(\tau_*) = h(\tau_*) \geq 0$. Then, since the minimum value of h is greater than zero, $h(\tau) \geq 0$ for all $\tau \in (0,\ 1)$.

We now consider the cases $b \neq 0$ and $b = 0$ separately. When $b \neq 0$, f assumes a minimum value at $\tau_o = -c/b$ and:

$$f(\tau_o) = -\frac{c^2}{3b} + d = \frac{c\tau_o}{3} + d.$$

Since $d \geq 0$ (from the fact that $g(0) \geq 0$), $f(\tau_o) < 0$ can only happen when $c < 0$. When this is the case, since $\tau_o \in [0, 1]$,

$$f(\tau_o) \geq \frac{c}{3} + d.$$

Hence when $b \neq 0$, having the following constraint will guarantee that the plane defined by the vector n won't be crossed during the time interval:

$$\frac{c}{3} + d \geq 0. \tag{4}$$

Now suppose that $b = 0$. Then $f(\tau)$ is linear in τ, so if both $f(0) \geq 0$ and $f(1) \geq 0$ then $f(\tau) \geq 0$ for all $\tau \in (0, 1)$ and, as before, $h(\tau) \geq 0$ for all $\tau \in (0, 1)$. We know that $f(0) \geq 0$ since $d > 0$. We can ensure that $f(1) \geq 0$ by imposing the following constraint:

$$f(1) = \frac{2c}{3} + d \geq 0. \tag{5}$$

For both $b = 0$ and $b \neq 0$, additional constraints are necessary only when $c < 0$. In this case, $2c/3 + d < c + 3d$ and thus constraint (5) implies (4). In other words, we can guarantee no inter-sample violation of the glideslope for any b by imposing constraint (5), which is rewritten here as:

$$n^T \left(r_k + \frac{2\dot{r}_k \Delta t}{3} \right) \geq 0. \tag{6}$$

The conclusion of the above analysis is that, when the constraint given by the inequality (6) is imposed together with $n^T r_k \geq 0$ and $n^T r_{k+1} \geq 0$, the plane defined by the vector n is not crossed over for $t \in [t_k, t_{k+1}]$. We can clearly approximate the glide slope with three planes and implement these constraints for each plane simultaneously to guarantee the satisfaction of the glide slope constraints for all times. Inspired by the constraint (6) resulted from the above analysis, the following inter-sample constraint is used and it is observed to satisfy the actual (non-approximated) glide slope constraint for all times in all of the simulations we performed:

$$\left\| r_{dk} + \frac{\dot{r}_{dk} \Delta t}{3} \right\| \leq \beta \left(r_v k + \frac{\dot{r}_{vk} \Delta t}{3} \right). \tag{7}$$

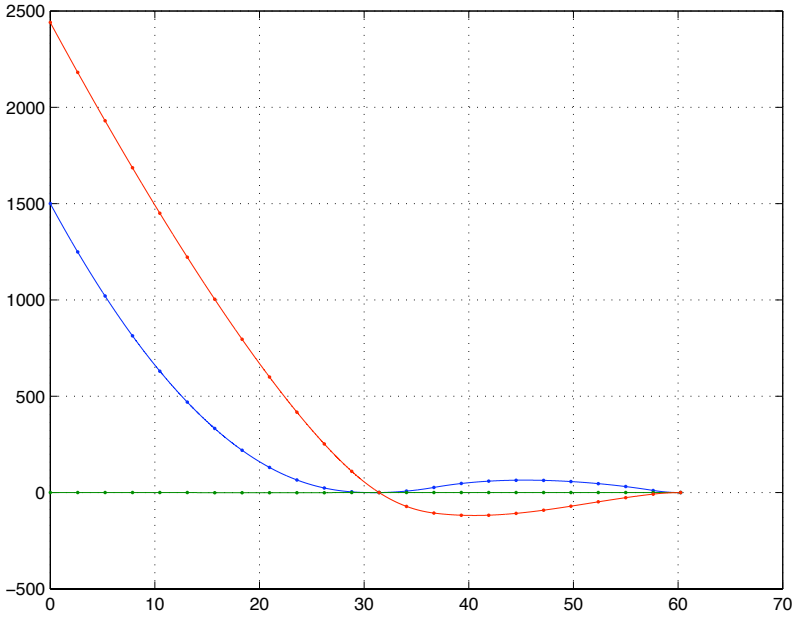

Figure 3. Inter-sample subsurface flight case

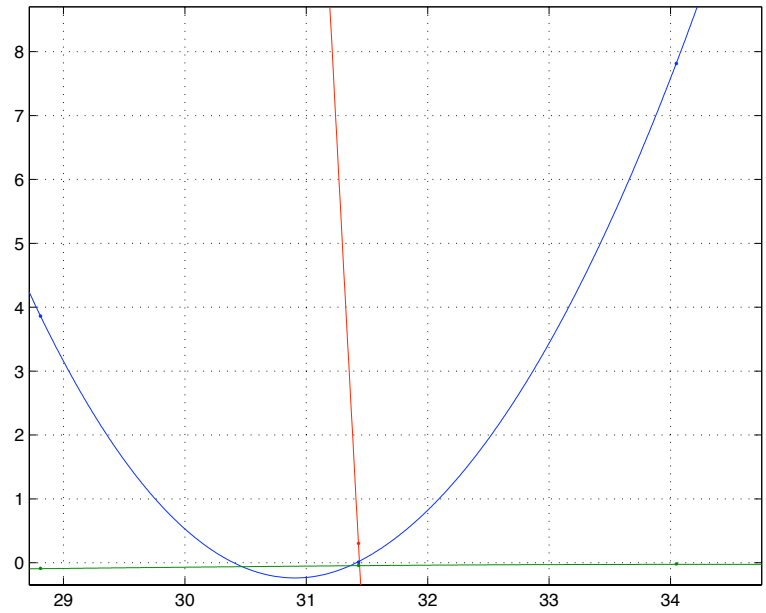

Figure 4. Inter-sample subsurface flight between time samples

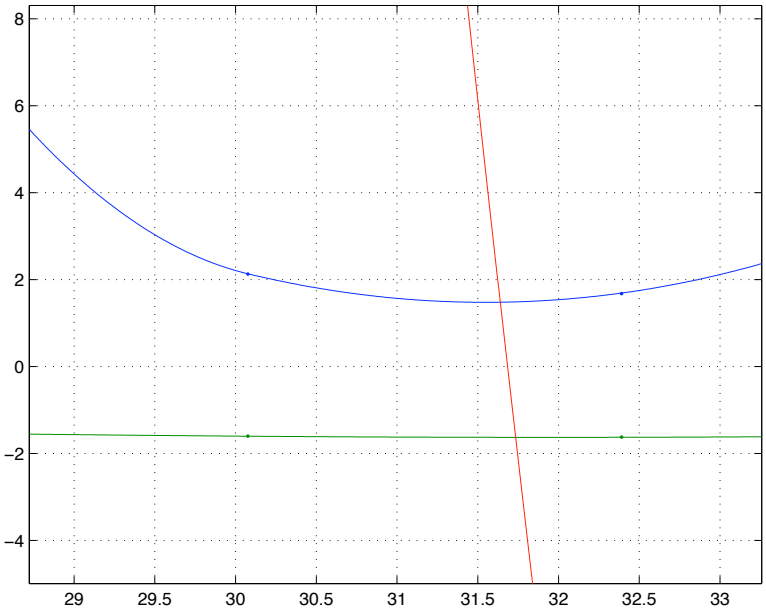

Figure 5. Inter-sample subsurface flight between time samples prevented via additional constraints

Another source of error the rotation rate of the Mars, which is ignored in the dynamics describing the motion of the lander spacecraft. This is not a major effect considering that the powered descent guidance only lasts a few minutes. However it can be significant enough to case subsurface flight in some cases. Figures 6 and 7 show a case where ignoring the Mars' rotation rate causes the lander to miss its final target point by approximately 30-40 meters (the solid line is the actual trajectory with the Mars' rotation whereas the dotted line is the sampled ideal trajectory without Mars' rotation). Indeed we can write discretized equations of motion exactly in terms of the piecewise constant control accelerations and set up the SOCP for PDG accordingly. However since the effects of the Mars rotation is not too pronounced, the following approximation to the equations of motion over the time intervals $[t_k, t_{k+1}),\ k = 0, ..., N-1$ is used successfully

$$\ddot{r} \approx 2\dot{r}_k \times \omega + (\omega \times r_k) \times \omega + u_k + g \qquad \text{for} \quad t \in [t_k,\ t_{k+1}) \tag{8}$$

where ω is the Mars angular velocity vector. Since ω is constant, we lump the acceleration terms due to the Mars rotation into the gravitational acceleration term by defining a piecewise constant g_k over $[t_k, t_{k+1})$ and use the following form of the approximate dynamics in PDG algorithm formulation

$$\begin{aligned} \ddot{r} &= u_k + g_k, \qquad t \in [t_k,\ t_{k+1}) \\ g_k &= g + 2\dot{r}_k \times \omega + (\omega \times r_k) \times \omega. \end{aligned} \tag{9}$$

Figure 8 presents the improved results for the case presented in Figure 7 when the Mars' rotation is incorporated to the PDG algorithm by using the equation 9.

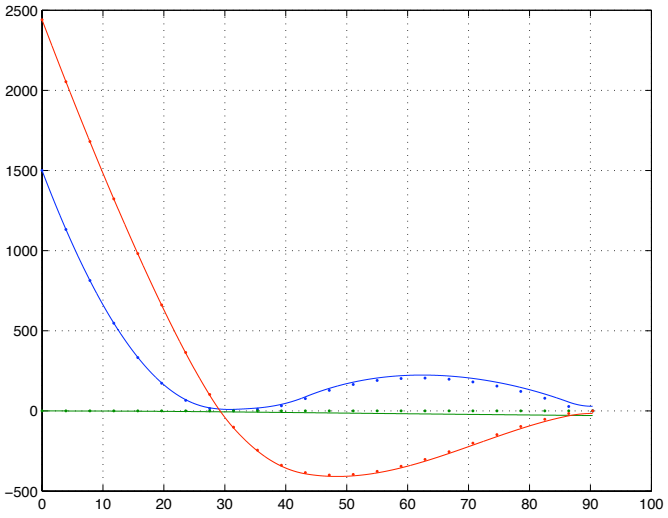

Figure 6. Effects of Ignoring Mars' Rotation

V. Fuel-limited targeting

In certain cases, there may not be sufficient fuel to reach the landing target. In a mission, this could occur because the initial condition is further than expected from the target. In this case, we would like to ensure that the spacecraft lands safely, and minimizes the final distance to the target. This is known as the *fuel-limited targeting* problem and is stated in Problem 5. The key differences between this and Problem 1 are: first, that the cost is the final distance from the target, rather than fuel use; and second, that the downrange position vector is no longer constrained to be at the target at the final time. The final distance from the target $\|r_d(t_f)\|$ is referred to as the *landing error*.

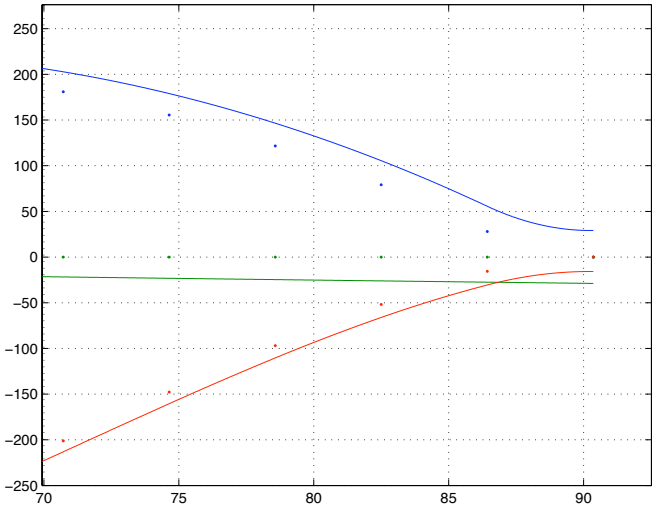

Figure 7. Effects of Ignoring Mars' Rotation

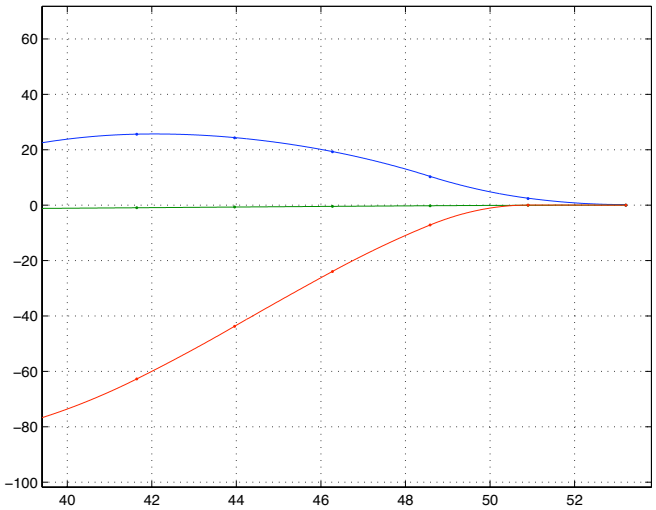

Figure 8. Effects of Ignoring Mars' Rotation

Problem 5.

$$\min_{t_f,\, T_c(\cdot)} \|r_d(t_f)\| \qquad \text{subject to}$$
$$\ddot{r}(t) = g + T_c(t)/m(t), \qquad \dot{m}(t) = -\alpha\|T_c(t)\|$$
$$0 < \rho_1 \le \|T_c(t)\| \le \rho_2,$$
$$\|r_d(t)\| \le \beta r_v(t)$$
$$m(0) = m_{wet}, \quad m(t_f) \ge 0, \quad r(0) = r_0, \quad \dot{r}(0) = \dot{r}_0, \quad r_v(t_f) = 0 \quad \dot{r}(t_f) = 0.$$

Our approach to solving Problem 5 is as follows. First we relax the nonconvex thrust constraints to give Problem 6, which is a convex optimization problem. This relaxation is performed in the same way that the original PDG problem (Problem 1) is relaxed to give Problem 2. We solve this convex optimization problem using existing solvers to get the closest final location that can be reached given the available fuel, which we refer to as the *closest target*. Since we are now minimizing landing error, rather than fuel use, the guarantees of Lemma 1 do not hold. In particular, an optimal solution to Problem 6 is not necessarily a feasible solution to Problem 5. As a second step, we therefore solve Problem 8, which is the relaxed minimum-fuel problem *with the final location constrained to be the closest target*. As we prove in Lemma 2, the solution to this problem is guaranteed to be an optimal, feasible solution to Problem 5; that is, we achieve the closest possible final location while obeying the nonconvex thrust constraints.

Problem 6.

$$\min_{t_f,\, T_c(\cdot),\, \Gamma(\cdot)} \|r_d(t_f)\| \quad \text{subject to}$$
$$\dot{m}(t) = -\alpha\Gamma(t), \tag{10}$$
$$\|T_c(t)\| \le \Gamma(t), \tag{11}$$
$$0 < \rho_1 \le \Gamma(t) \le \rho_2, \tag{12}$$
$$\ddot{r}(t) = g + T_c(t)/m(t),$$
$$\|r_d(t)\| \le \beta r_v(t)$$
$$m(0) = m_{wet}, \quad m(t_f) \ge 0, \quad r(0) = r_0, \quad \dot{r}(0) = \dot{r}_0, \quad r_v(t_f) = 0 \quad \dot{r}(t_f) = 0.$$

Problem 7.

$$\max_{t_f,\, T_c(\cdot)} m(t_f) = \min_{t_f,\, T_c(\cdot)} \int_0^{t_f} \|T_c(t)\|\, dt \qquad \text{subject to}$$
$$\ddot{r}(t) = g + T_c(t)/m(t), \qquad \dot{m}(t) = -\alpha\|T_c(t)\|$$
$$0 < \rho_1 \le \|T_c(t)\| \le \rho_2,$$
$$\|r_d(t)\| \le \beta r_v(t)$$
$$m(0) = m_{wet}, \quad m(t_f) \ge 0, \quad r(0) = r_0, \quad \dot{r}(0) = \dot{r}_0, \quad r_v(t_f) = 0 \quad r_d(t_f) = R \quad \dot{r}(t_f) = 0.$$

Problem 8.

$$\min_{t_f,\, T_c(\cdot),\, \Gamma(\cdot)} \int_0^{t_f} \Gamma(t)\, dt \quad \text{subject to}$$
$$\dot{m}(t) = -\alpha\Gamma(t), \tag{13}$$
$$\|T_c(t)\| \le \Gamma(t), \tag{14}$$
$$0 < \rho_1 \le \Gamma(t) \le \rho_2, \tag{15}$$
$$\ddot{r}(t) = g + T_c(t)/m(t),$$
$$\|r_d(t)\| \le \beta r_v(t)$$
$$m(0) = m_{wet}, \quad m(t_f) \ge 0, \quad r(0) = r_0, \quad \dot{r}(0) = \dot{r}_0, \quad r_v(t_f) = 0 \quad r_d(t_f) = R \quad \dot{r}(t_f) = 0.$$

Lemma 2. *Assume that a solution to Problem 6 exists, which we denote* $\{t_f^6, T_c^6(\cdot), r^6(\cdot)\}$. *Let* $\{t_f^8, T_c^8(\cdot), r^8(\cdot)\}$ *be the solution to Problem 8 with* $R = r_d^6(t_f)$. *Then* $\{t_f^8, T_c^8(\cdot), r^8(\cdot)\}$ *is an optimal solution to Problem 5.*

Proof. First, note that $\{t_f^6, T_c^6(\cdot), r^6(\cdot)\}$ is a feasible solution to Problem 8 with $R = r_d^6(t_f)$, since the only additional constraint in Problem 8 is $r_d(t_f) = R$, and by design we have $r_d^6(t_f) = R$. Hence a feasible solution to Problem 8 exists for $R = r_d^6(t_f)$. Solving this problem we obtain $\{t_f^8, T_c^8(\cdot), r^8(\cdot)\}$, and from Lemma 1, we know that this is a feasible solution to Problem 7 for $R = r_d^6(t_f)$. Any feasible solution to Problem 7 is a feasible solution to Problem 5 since the constraints in these problems differ only by the added $r_d(t_f) = R$ constraint in Problem 7. Hence $\{t_f^8, T_c^8(\cdot), r^8(\cdot)\}$ is a feasible solution to Problem 5. This means that:

$$\|r_d^8(t_f)\| \geq \|r_d^5(t_f)\|. \tag{16}$$

Now note that since Problem 6 is a relaxation of Problem 5, we know that $\|r_d^6(t_f)\| \leq \|r_d^5(t_f)\|$. Since in Problem 8 we have assigned $R = r_d^6(t_f)$, we know that $r_d^8(t_f) = r_d^6(t_f)$ and hence:

$$\|r_d^8(t_f)\| \leq \|r_d^5(t_f)\|. \tag{17}$$

Combining (16) and (17) we have $\|r_d^8(t_f)\| = \|r_d^5(t_f)\|$. Hence the targeting error in the optimal solution to Problem 8 is the same as that in the optimal solution to Problem 5. We have already shown that $\{t_f^8, T_c^8(\cdot), r^8(\cdot)\}$ is a feasible solution to Problem 5. Hence the optimal solution to Problem 8 is an optimal solution to Problem 5. □

In order to solve Problems 6 and 8 in practice, we perform the same change of variables and discretization as described in Sections II and III. For a given t_f, this leads to two SOCP problems that, as before, can be solved to global optimality using existing solvers, with guarantees on convergence and accuracy. As with the original PDG problem (Problem 1) we must perform a line search to find the optimal time-of-flight t_f^*; we use the same line search described in Section III, except that at each iteration of the outer loop, instead of solving Problem 1, we solve Problem 6. Then, in order to ensure that the nonconvex constraints are satisfied, we solve Problem 8 once, using the targeting error from the optimal solution found to problem Problem 6.

Simulation results obtained using the new fuel-limited targeting approach are shown in Figures 9 and 10. In this example, the spacecraft is at an initial altitude of $2km$ and initial lateral position of $(10km, 5km)$. The initial vertical velocity is $-20m/s$ downwards, and the initial lateral velocity is $(100m/s, 100m/s)$ In this situation, the minimum fuel mass required to reach the target is $672kg$. However we restrict the fuel mass to $600kg$, and hence the fuel-limited targeting algorithm finds the closest feasible landing location; in this case the closest landing location is $2.4km$ from the target.

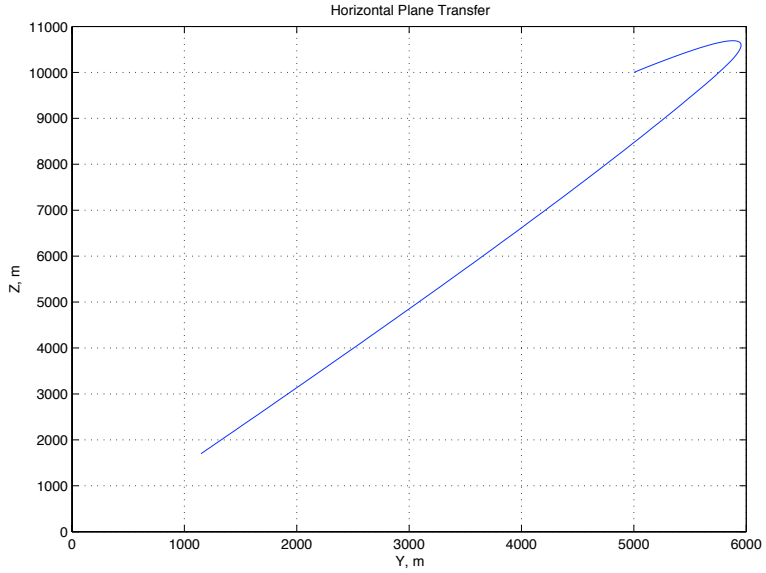

Figure 9. Fuel-limited horizontal transfer. The initial position is at $(10000, 5000)$ **and the target is at** $(0, 0)$**. Despite not having enough fuel to reach the target, the fuel-limited algorithm performs a safe landing as close as possible to the target.**

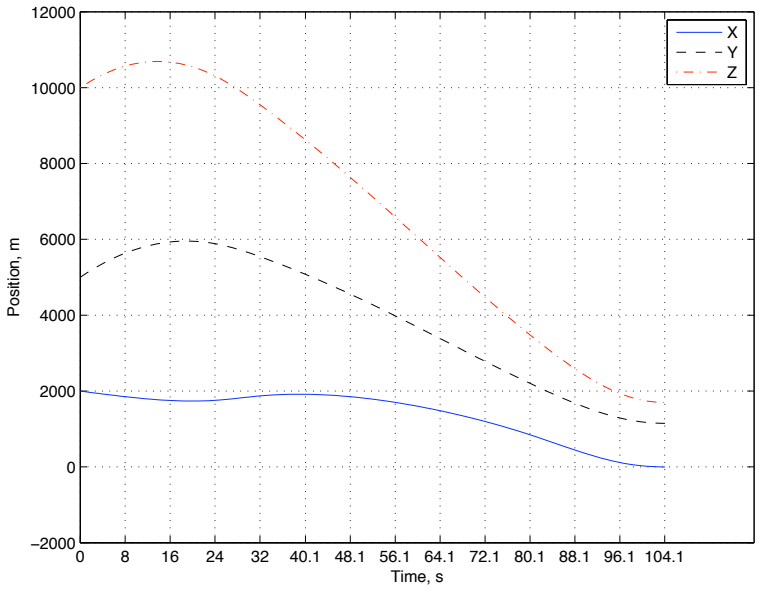

Figure 10. Fuel-limited trajectory. Zero vertical position (X) and velocity is achieved, but the final lateral position (Y,Z) is some distance from the target.

VI. Towards an Onboard Solution

The SOCP formulation presented in earlier sections enables the fuel optimal trajectory generation problem to be solved robustly and very efficiently using numerical SOCP solvers, which have guarantees of convergence to the globally optimal solution. One such algorithm, based on a primal dual interior-point method [**?**,**?**], has been implemented in the Matlab and C programming languages for a general class of SOCPs and tested extensively [**?**, **?**] . A typical fuel optimal solution is computed in 2-3 seconds by using the Matlab version on a 2.16 GHz Intel processor. Note that the numerical solver is a generic one and it is not customized for the powered descent guidance (PDG) problem. These execution times and its robust convergence global fuel optimal solutions make algorithm an excellent ground analysis software element, and illustrate its potential for onboard guidance optimization.

However, due to mission operational constraints and risk, it is desirable to have a non-iterative solution approach for onboard use. This motivated us to investigate alternative methods for onboard solution of the PDG problem. The main idea behind this approach is to generate a table of solutions of the PDG problem for carefully chosen initial states (initial positions and velocities) to generate all other possible solutions onboard. This approach is currently under development and the mathematical details involved in constructing the methodology is beyond the scope of this paper. In this paper, we just present the main idea and the research direction. The main property of this table lookup approach is the following: Given a requirement on the maximum sub-optimality that can be tolerated, for example *"all onboard generated solutions must not spend more than* 1% *fuel than the actual optimal fuel"*, a table of pre-computed solutions is constructed. This table is then used to obtain feasible solutions for *any possible* initial state (not just the tabulated initial states), while guaranteeing no more than 1% suboptimality. The onboard method has the following steps:

- Tabulated way-points (TWPs) are generated on the ground for a set of trajectories with different initial positions and velocities.
- The ISC (initial state cell) that a given initial state belongs to is found via a table lookup.
- TWPs are used by the flight computer to generate interpolated way-points (IWPs) via a smart interpolation, which has a closed form formula.
- IWPs are used to generate the smooth trajectories in-between waypoints. Piecewise 5th order polynomials of time are used, based on closed form formulas.

See Figure 11 for an illustration of the table lookup method. Figure 12 presents some initial results generated using

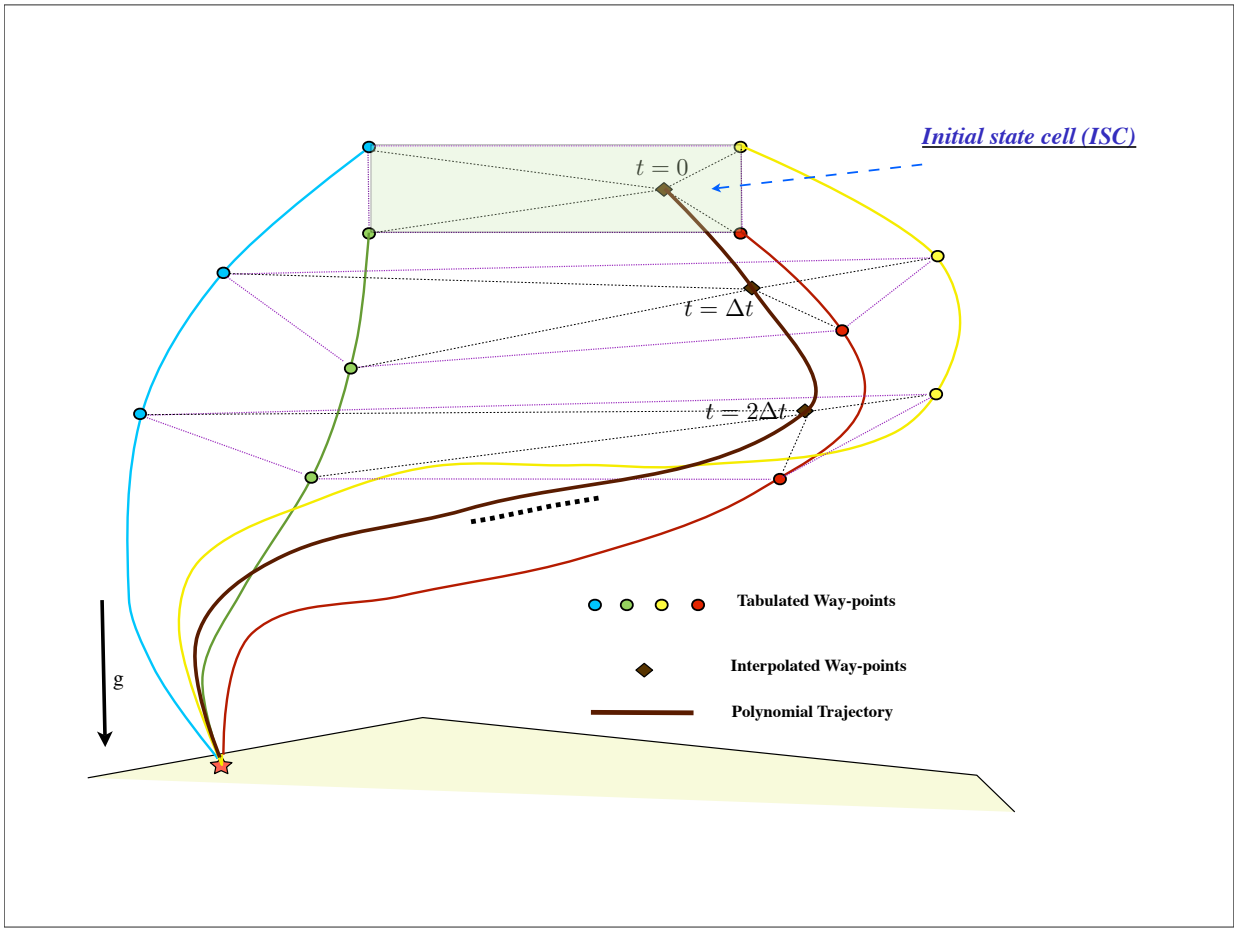

Figure 11. Illustration of the Table Lookup Interpolation Method

the table lookup method summarized above. An initial state cell (ISC) with the following dimensions is chosen

$$\begin{aligned} \text{Altitude} &= 3 \rightarrow 6 \ \ km \\ \text{Downrange} &= 1 \rightarrow 4 \ \ km \\ \text{Descent velocity} &= 40 \rightarrow 100 \ \ m/s \\ \text{Downrange velocity} &= 30 \rightarrow 60 m/s \\ \text{Crossrange velocity} &= 0 \rightarrow 30 m/s \end{aligned}$$

and the time-of-flight is fixed at 65 seconds for simplicity. Note that this is a relatively large initial state region. The theoretical analysis suggested a worst-case sub-optimality of 3%. For comparison, an empirical analysis of the sub-optimality encountered in practice was carried out. The empirical results are obtained by uniformly sampling initial states from the ISC. These show that, in practice, this approach achieves less than 1% worst-case sub-optimality. Since the initial theoretical analysis of the method and the simulation results are extremely promising, this table lookup method will be pursued for the onboard implementation.

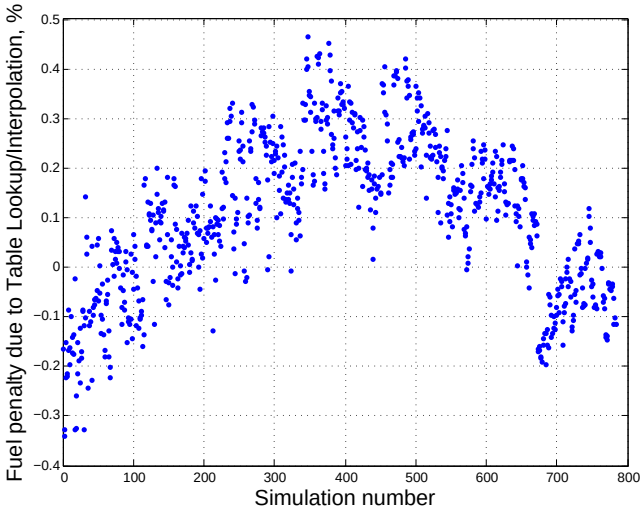

Figure 12. Fuel Penalty in a ISC. The fuel sub-optimality introduced using the table lookup approach is less than 1%. Note that sometimes the interpolated solution uses *less* full than that from the numerical optimization. This is because the numerical optimization stops once certain convergence criteria are met, and so may return a slightly suboptimal solution.

VII. Conclusions

In this paper, we present enhancements for a powered descent guidance algorithm for Mars pinpoint landing. This algorithm is based on the convexification of a nonconvex optimization problem and computes the global optimal fuel trajectories in polynomial time, that is, with guaranteed convergence to the optimal solution with a deterministic upper bound on the number of iterations needed to converge to any prescribed neighborhood of it. Hence the algorithm hold a strong promise for onboard implementation, which is a currently ongoing research effort. We enhance this algorithm by incorporating: (i) An efficient way to compute the optimal time-of-flight; (ii) A method to guarantee no inter-sample subsurface flight occurs due to the discretization; (iii) A formulation to consider Mars' rotation rate; (iv) A very efficient method to check the feasibility of the problem before attempting to find any optimal solution numerically; and (v) a new algorithm for computing the optimal trajectory when there is insufficient fuel to reach the target. Future work will focus on the implementation of a customized table lookup solution onboard to solve powered descent guidance fuel optimization problem.

Acknowledgments

We wish to gratefully acknowledge Jeffrey Tooley and Mark Ivanov of JPL for their very valuable comments and suggestions. We also acknowledge Fred Y. Hadaegh of JPL for his encouragements and support in writing of this paper. This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration, and funded through the internal Research and Technology Development program.

References

[1]Acikmese, B. and Ploen, S. R., "Convex Programming Approach to Powered Descent Guidance for Mars Landing," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, 2007, pp. 1353–1366.

[2]Acikmese, B. and Ploen, S., "A Powered Descent Guidance Algorithm for Mars Pinpoint Landing," *AIAA Guidance, Navigation, and Control Conference, San Francisco, CA*, 2005.

[3]Ploen, S., Acikmese, B., and Wolf, A., "A Comparison of Powered Descent Guidance Laws for Mars Pinpoint Landing," *AIAA Guidance, Navigation, and Control Conference, Keystone, CO*, 2006.

[4]Klumpp, A. R., "Apollo Lunar Descent Guidance," *Automatica*, Vol. 10, 1974, pp. 133–146.

[5]Sostaric, R. and Rea, J., "Powered Descent Guidance Methods for the Moon and Mars," *AIAA Guidance, Navigation, and Control Conference, San Francisco, CA*, 2005.

[6]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*, 2002.

[7]Wolf, A. A., Graves, C., Powell, R., and Johnson, W., "Systems for Pinpoint Landing at Mars," *14th AIAA/AAS Space Flight Mechanics Meeting, Maui, Hawaii*, 2004.

[8]Meditch, J. S., "On the Problem of Optimal Thrust Programming For a Lunar Soft Landing," *IEEE Transactions on Automatic Control*, Vol. AC-9, No. 4, 1964, pp. 477–484.

[9]Flemming, W. H. and Rishel, R. W., *Deterministic and Stochastic Optimal Control*, Springer-Verlag, 1975.

[10]Topcu, U., Casoliva, J., and Mease, K., "Fuel Efficient Powered Descent Guidance for Mars landing," *AIAA Guidance, Navigation, and Control Conference, San Francisco, CA*, 2005.

[11]Najson, F. and Mease, K., "A Computationally Non-expensive Guidance Algorithm for Fuel Efficient Soft Landing," *AIAA Guidance, Navigation, and Control Conference, San Francisco, CA*, 2005.

[12]Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193–207.

[13]Hull, D. G., "Conversion of Optimal Control Problems into Parameter Optimization Problems," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, 1997, pp. 57–60.

[14]Fahroo, F. and Ross, I. M., "Direct Trajectory Optimization by a Chebyshev Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, 2002, pp. 160–166.

[15]Vlassenbroeck, J. and Dooren, R. V., "A Chebyshev Technique for Solving Nonlinear Optimal Control Problems," *IEEE Transactions on Automatic Control*, Vol. 33, No. 4, 1988, pp. 333–340.

[16]Berkovitz, L. D., *Convexity and Optimization in* $\mathbb{R}^n$, John Wiley & Sons, Inc., 2002.

[17]Boyd, S. and Vandenberghe, L., *Convex Optimization*, Cambridge University Press, 2004.

[18]Vandenberghe, L. and Boyd, S., "Semidefinite Programming," *SIAM Review*, Vol. 38, No. 1, 1995, pp. 49–95.

[19]Nesterov, Y. and Nemirovsky, A., *Interior-point Polynomial Methods in Convex Programming*, SIAM, 1994.

[20]Toh, K. C., Todd, M. J., and Tutuncu, R. H., "SDPT3-A Matlab software package for semidefinite programming," *Optimization Methods and Software*, Vol. 11, 1999, pp. 545–581.

[21]Ye, Y., *Interior Point Algorithms*, John Wiley & Sons, Inc., 1997.

[22]Bertsekas, D. P., *Nonlinear Programming, second edition*, Athena Scientific, 2000.