Computational Guidance for Planetary Powered Descent using Collaborative Optimization

Xiuqiang Jiang*, Shuang Li† and Ting Tao‡

1. Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
2. Hubei Sanjiang Space Hongfeng Control Co., Ltd., Xiaogan 432000, China

Abstract: An innovative computational guidance framework is proposed for planetary powered descent using collaborative optimization approach. First, the dynamical model and constraints for planetary powered descent are presented. Then, using collaborative optimization strategy, the computational guidance framework for powered descent is formulated as a multi-discipline optimization problem including trajectory optimization, optimal guidance, and system-level optimization. Finally, the computational guidance approach employs three algorithms for respectively solving the three optimization modules to implement numerical simulations. The optimality and robustness of the computational guidance approach are verified with all constraints satisfied even in the presence of initial state uncertainty.

Keywords: planetary powered descent, computational guidance, multi-discipline optimization, collaborative optimization

I. Introduction

Powered descent is typically the method used to land a spacecraft on a planetary surface. Pinpoint landing missions require the development of more advanced guidance approaches. The guidance algorithms should be able to: a) drive the lander from a given initial state to the desired landing site with approximately zero velocity and a position error of less than 100 m; b) autonomously determine an optimal landing site and retarget to reach it; and c) flexibly coordinate the landing accuracy and fuel consumption to achieve comprehensive optimality under uncertain conditions, un-modeled items, and landing site retargeting [1-3]. Most of the powered descent guidance algorithms developed during the Apollo era are based on linear control theories, which cannot meet the

* Ph.D. Candidate, Department of Astronautics Engineering; also Visiting Research Scholar, Space Systems Engineering Lab, College of Engineering, University of Arizona, Tucson, Arizona 85721, U.S.; jiangxq@nuaa.edu.cn.
† Professor, Department of Astronautics Engineering; lishuang@nuaa.edu.cn.
‡ Spacecraft Control System Engineer.
requirements of current and future planetary landing missions [4].

After the 2012 Mars Science Laboratory (MSL) and the 2013 Chang’e-3 (CE-3) lunar lander missions, planetary powered descent guidance technologies have been further improved. More attention has been paid to the fuel optimality, robustness against uncertainty, and autonomous retargeting to avoid hazards [5, 6]. An improved Apollo suboptimal-fuel guidance and some supplementary linear guidance laws were piecewise employed during MSL powered descent to ensure safe landing under uncertainty. Furthermore, autonomous hazard avoidance technologies were developed in CE-3 lunar landing mission, where proportion-integration-differentiation (PID) guidance is designed to adapt landing site retargeting for hazard avoidance. Different from those segmented guidance laws, Wibben and Furfaro proposed a guidance law for lunar landing and retargeting using a hybrid control strategy. Though it has good flexibility for retargeting during landing, the fuel-consumption optimality is not taken into account [7]. The zero-effort-miss/zero-effort-velocity (ZEM/ZEV) algorithm usually focuses on minimizing the terminal state errors, which has excellent adaptability and robustness against retargeting maneuvers and uncertainties. However, the optimal fuel-consumption is also not taken into account in the ZEM/ZEV guidance [8]. The main concern of the optimal feedback guidance algorithms is always the guidance performance optimization under uncertain perturbations, but this kind of approaches usually doesn’t consider retargeting [9]. To combine hazard avoidance with fuel sub-optimality, Zhang proposed a hybrid guidance algorithm for Mars powered descent using ZEM/ZEV and optimal feedback [10]. In summary, these existing methods for planetary powered descent still have design limitations when combining fuel optimality with retargeting flexibility and robustness against uncertainty.

As a promising solution to provide more optimality and flexibility, the Computational Guidance and Control (CG&C) concept has recently emerged [11]. The distinguishing features of the CG&C concept are summarized as follows [11]: a) guidance laws and controllers of fixed structures are replaced by numerical algorithms; b) the process of determining guidance and control commands may be model-based or data-based and does not require significant pre-mission planning, gain tuning, or extensive offline design of nominal references; c) the generation of guidance and control commands relies extensively on online computation, often involving iterations; d) the output of the CG&C algorithm is typically the optimal solution based on the current actual state and the control effect; and e) the performance of the CG&C algorithm depends upon the optimization problem formulation and corresponding optimization algorithms. For online computational guidance, Lu [12] developed an entry guidance method combining a fully numerical predictor-corrector algorithm for trajectory planning with a linear quadratic regulator (LQR) for tracking control. However, the LQR gains must be designed offline. Pinson [13] applied the
convex optimization method to design an optimal propellant powered descent trajectory that can be quickly computed onboard. This algorithm can run autonomously once the dynamical model coefficients are determined. Dueri and Aci̇kmeş proposed a new onboard-implementable, real-time, convex, optimization-based powered descent guidance algorithm for planetary pinpoint landing [14]. These studies show that the capability to generate optimal powered descent guidance trajectories onboard and in real time can significantly enhance the landing accuracy of a vehicle. At the same time, the robustness and flexibility are not good enough due to the lack of onboard collaborative optimal control.

The multi-disciplinary optimization (MDO) provides the benefits of combining multiple optimization algorithms to solve multi-module, multi-objective and multi-constraint optimization problems [15]. D'Souza [16] proposed a multi-disciplinary design, analysis, and optimization (MDAO) approach that can be used to generate the trajectory and guidance for planetary atmospheric entry. Bonetti [17] conceptually introduced the MDO strategy in the Mars entry descent and landing (EDL) guidance system design. As an effective approach of MDO, collaborative optimization (CO) is a decomposition algorithm which uses projection to transform the MDO problem into a bi-level programming problem consisting of a master problem (i.e. system-level problem) and multiple sub-problems (i.e. subsystem-level problems or disciplines) [15]. Since each of the design disciplines is enclosed into one of the independent sub-problems, CO allows a high-level of modularity in the solution process.

To date, there are still no related reports that using MDO approaches to design a computational guidance for planetary powered descent comprehensively considering fuel optimality, retargeting flexibility, and robustness against uncertainty [16,17]. In this work, we introduce a novel CO-based computational guidance approach for planetary powered descent. To enhance the flexibility, optimality and accuracy, the CO approach is used for the powered descent guidance problem. The proposed computational guidance framework has the inherent capability to incorporate multiple optimization algorithms under a hierarchical mechanism. Thus, the guidance command autonomously generates and flexibly coordinates the trajectory planning and tracking control to adapt to the real state and minimize the state errors caused by uncertainties, un-modeled items, and landing site retargeting. The comprehensive performance of the guidance system is optimized accordingly.

II. Planetary Powered Descent Dynamics Model and Constraints

In formulating the lander guidance problem for planetary powered descent, we model the lander dynamics near the planetary surface using the following three-degree-of-freedom dynamic equations with respect to a coordinate system with the origin on the surface of the planet. This study only considers spherical central planet that
excluding the asteroid cases with non-central gravitational field [18]. The major forces acting on the lander are the gravitational force from the planet and the thrust forces generated by the propulsion system of the lander, while other forces (e.g., aerodynamic disturbances, gravitational perturbation, and un-modeled items) acting on the lander are typically minimal or absent. Under these conditions, the dynamical equations for a planetary lander can be expressed as follows [7, 19, 20]:

$$\dot{r} = \mathbf{v}$$  \hspace{1cm} (1)

$$\dot{\mathbf{v}} = -\frac{\mu}{\|\mathbf{R} + r\|}(\mathbf{R} + r) + \frac{T}{m} + \mathbf{p}$$  \hspace{1cm} (2)

$$m \ddot{\mathbf{r}} = -\frac{\|\mathbf{v}\|}{\mathbf{I}_\infty \mathbf{g}_0}$$  \hspace{1cm} (3)

where $\mathbf{r}$ and $\mathbf{v}$ are the position vector and velocity vector of the lander, respectively, $\mathbf{T}$ denotes the commanded thrust vector with the thrust magnitude $\|\mathbf{T}\| = \sqrt{T_x^2 + T_y^2 + T_z^2}$, $\mathbf{R}$ represents the radius of the planet, $m$ is the mass of the lander, $\mu$ denotes the gravitational constant of the planet, $\mathbf{I}_\infty$ is the specific impulse of the retrorocket, $g_0 = 9.80665 \text{m/s}^2$ is the Earth’s sea-level gravitational acceleration, and $\mathbf{p}$ represents any perturbing or un-modeled accelerations. This model is employed to simulate the real lander descent dynamics driven by the commanded thrust (i.e., guidance command).

Considering the limit of the retrorocket engine, the magnitude of the thrust should satisfy the constraint

$$\Gamma_{\text{min}} \leq \|\mathbf{T}\| \leq \Gamma_{\text{max}}$$  \hspace{1cm} (4)

To plan a trajectory for a planetary pinpoint soft landing, the terminal position should be at the selected landing site $\mathbf{r}_f$, and the terminal velocity should be zero. Thus, the state constraints at the terminal time $t_f$ are

$$\mathbf{r}(t_f) = \mathbf{r}_f, \quad \mathbf{v}(t_f) = [0, 0, 0]^T$$  \hspace{1cm} (5)

The initial states of the lander are assumed to be known before the trajectory planning and tracking control at the initial time $t_0$ are

$$\mathbf{r}(t_0) = \mathbf{r}_0, \quad \mathbf{v}(t_0) = \mathbf{v}_0, \quad m(t_0) = m_0$$  \hspace{1cm} (6)

Without loss of generality, two important assumptions are made:

**Assumption 1**: The available onboard fuel should be sufficient for the entire powered descent and landing process and satisfies
\[ m_\text{dry} \geq m(t) \geq m_\text{dry} \] (7)

where \( m_\text{dry} \) denotes the dry mass of the lander.

**Assumption 2:** The maximum magnitude of the thrust is significantly larger than the magnitude of the resultant force of gravity and any perturbative or un-modeled forces. That is,

\[
\frac{T_{\text{max}}}{m} > \frac{\mu}{\|R + r\|}(R + r) + p
\] (8)

during the entire flight. It is to indicate that the total system is under control with the thrust.

### III. Guidance Strategy Design

The planetary powered descent guidance problem can be generally formulated as follows: given the current state of the lander, determine a real-time acceleration command program that brings the lander to the target landing site on the planetary surface with zero velocity [7, 19, 20]. We further consider the optimality and flexibility of the guidance so that the lander can safely and precisely arrive at the specified landing site with minimal fuel cost in the cases of retargeting, perturbing or un-modeled accelerations, and uncertain initial errors and thrust deviation.

#### A. Guidance Framework

As mentioned above, there are no public reports that formulating the planetary powered descent guidance problem as three interrelated optimization sub-problems from the MDO viewpoint [16, 17]. Accordingly, there is no literature that generating the planetary powered descent guidance by using three online optimization modules from the CO viewpoint. The aim of this work is to develop a computational guidance approach based on the MDO theory in which the entire system can take advantage of multiple optimization algorithms integrated within a unified guidance framework. This approach can provide optimality and flexibility that are impossible with only one optimal guidance algorithm. Here, we construct and integrate a system-level coordinated optimization, a trajectory optimization and an optimal control, which are employed by the CO strategy (a hierarchical optimization mechanism) [15] based on the optimization objective. Figure 1 shows the guidance framework and its function. Note that the estimation errors of the autonomous navigation system would not be taken into consideration in the guidance strategy design.
Normally, Problem 1 is primarily used to generate a minimum-fuel reference trajectory from the current state to the target landing state. Problem 2 is mainly adopted to realize a minimum-state-error control so that the lander accurately follows the reference profile to achieve a pinpoint soft landing. To enhance the flexibility of the guidance system, the total cost function is minimized in Problem 3 via coordinating the weights of the minimum-fuel and the minimum-state-error constraints. Clearly, if the lander always follows the reference trajectory accurately without additional corrections, then the powered descent process also satisfies the minimum-fuel constraint.

In particular, the state error index will be activated and added as a secondary objective into the reference trajectory optimization (Problem 1) for online re-planning of a reference profile when the state errors cannot converge to an acceptably small value in time. Correspondingly, the control cost index will be activated and added as a secondary objective into the optimal control problem (Problem 2). The weights of these indexes will be coordinated by solving Problem 3. The overlarge initial errors and state errors caused by the landing site retargeting will require trajectory re-planning and consider a higher tracking control efficiency (i.e., as little control as possible). In addition, larger tracking control results in more fuel cost. Thus, the optimality and flexibility of the guidance can be realized through coordinating the two optimization processes. From the point of view of the optimization algorithm, the local optimal solutions are sufficient for Problems 1 and 2 while the global optimization is anticipated for Problem 3.

These three optimization problems are formulated as follows.

B. Trajectory Optimization
Given the constraints defined in Eqs. (4) through (8), the powered descent trajectory planning can be formulated and relaxed as the following optimal control problem:

**Problem 1 (trajectory optimization).**

\[
\begin{align*}
\max_{m(t_f) - m(t_0)} \quad & J_i = \int_0^1 \| T(t) \| dt \\
\text{subject to} \quad & \mathcal{G} = v \\
& \mathcal{G} = -\frac{\mu}{\| R + r \|} (R + r) + \frac{T}{m} \\
& \mathcal{G} = -\frac{[T]}{I_p g_0} \\
& \Gamma_{\min} \leq \| T \| \leq \Gamma_{\max} \\
& r(t_0) = r_0, \quad v(t_0) = v_0, \quad m(t_0) = m_0 \\
& r(t_f) = r_f, \quad v(t_f) = [0, 0, 0]^T, \quad m(t_f) \geq m_{\text{dry}}
\end{align*}
\]

Since the disturbance acceleration and the un-modeled acceleration and the thrust deviation (the execution error) in the dynamics are generally on a smaller order of magnitude and are uncertain and unknown in advance, the dynamic constraints (10-12) in the off-line design do not include these uncertain and unknown items.

The fuel-optimal guidance problem is equivalent to minimizing the control costs during the powered descent. Denote \( x = [r, v]^T \). In order to facilitate the subsequent calculation, let \( u = \mathcal{G} = -\frac{\mu}{\| R + r \|} (R + r) + \frac{T}{m} \). As the decrease of height is much less than the radius of the planet, the gravity acceleration \(-\frac{\mu}{\| R + r \|} (R + r)\) is assumed to be nearly a constant here. Thus, minimizing \( \int_0^1 \| T(t) \| dt \) means minimizing \( \int_0^1 u^T(t) R_i u(t) dt \), where \( R_i \) is a positive definite matrix. The objective function defined in Eq. (9) can be re-written as

\[
\min_{J_i} J_i = \frac{1}{2} \int_0^1 u^T(t) R u(t) dt
\]

To match the CO framework, this performance index can be further re-written as a relaxed objective as follows:

\[
\min_{J_i} J_i = \frac{1}{2} \| \delta x(t_f) \| S_i \| \delta x(t_f) \| + \frac{1}{2} \int_0^1 u^T(t) R u(t) dt
\]

where \( \delta x(t_f) = x(t_f) - x_f = [r(t_f) - r_f, v(t_f) - v_f]^T \), \( S_i \) are defined as a positive semi-definite matrix that
determine the relative weight of the effort (i.e., the commanded acceleration) and the terminal state accuracy, respectively. In the CO framework, the major objective of the trajectory optimization is minimum fuel usage, and we initialize \( S_i = 0 \) and stipulate the control cost weight always larger than the terminal state error weight. Thus, the index defined in the Eq. (17) is equivalent to minimizing the control cost depicted in the Eq. (16) when the reference trajectory is tracked accurately (i.e., the state errors \( \delta x(t) = 0 \) and \( \delta x(t_f) = 0 \)).

Therefore, this problem is to find the reference state \( x^* \) and the nominal control \( u^* \) that minimizes the performance index defined in the Eq. (17) to reach the current target landing state \( (r_f, v_f) \).

C. Optimal Control

By solving Problem 1, the lander can realize minimum-fuel guidance once the minimum-state-error constraint is satisfied. Since the offline constructed dynamics model does not include uncertain and un-modeled items, state errors are likely to occur. The tracking control is generated based on the state errors with respect to the reference trajectory. In this fashion, the tracking control problem is cast into a trajectory state regulation problem. The state errors should be minimized with optimal tracking control (Problem 2).

**Problem 2 (optimal control).**

Find the optimal control \( \delta u \) while minimizing the performance index

\[
\min \| r(t) - r^*(t) \|, \| v(t) - v^*(t) \| \quad \Rightarrow \quad J_2 = \frac{1}{2} \delta x^T(t) S_i \delta x(t)
\]

subject to

\[
\delta x(t) = A(t) \delta x(t) + B(t) \delta u(t)
\]

\[
A(t) = \begin{bmatrix} 0_{3x3} & I_{3x3} \\ 0_{3x3} & 0_{3x3} \end{bmatrix}, \quad B(t) = \begin{bmatrix} 0_{3x3} \\ I_{3x3} \end{bmatrix}
\]

\[
\delta x(t_g) = x(t_g) - x^*(t_g), \quad \delta x(t_f) = x(t_f) - x^*(t_f)
\]

\[
0 \leq \| \delta u \| \leq -\frac{\mu}{\| R + r \|} + \frac{\Gamma_{m}}{m} - \| u^* \|
\]

\[
m(t_g) = m_{0g}, \quad m(t_f) \geq m_{0y}
\]

Here, \( \delta x(t) = x(t) - x^*(t) = [r(t) - r^*(t), v(t) - v^*(t)]^T \) represents the differences between the actual and nominal values in the current guidance cycle, \( \delta u(t) = u(t) - u^*(t) \), and \( t_g \) and \( t_f \) denote the initial time and terminal time of the current guidance cycle respectively. The linearized dynamics of equations (10)-(12) can be considered...
the linear time varying system described in the Eq. (19). The matrices \( A(t) \) and \( B(t) \) are obtained analytically from Eqs. (10) through (12).

Similarly, to match the CO framework, this performance index can be further re-written as a relaxed objective

\[
J_2 = \frac{1}{2} \int_{t_f}^{t_0} \left[ \delta x^T(t) S_2 \delta x(t) + \delta u^T(t) R_2 \delta u(t) \right] dt
\]  

where \( S_2 \) and \( R_2 \) are the symmetric positive semi-definite matrices that determine the relative weight of the state errors and tracking control cost. In the CO framework, the major objective of the optimal control is the minimum-state-error constraint; thus, we initialize \( R_2 = 0 \) and stipulate the real-time state error weight always larger than the real-time control cost weight. Therefore, the index defined in the Eq. (24) is the same as the minimizing state errors described in the Eq. (18) when the tracking control is not necessary (i.e., tracking control \( \delta u(t) = 0 \)).

Therefore, the problem is to find the state regulation control \( \delta u \) that minimizes the performance index depicted in the Eq. (24).

**D. Coordinated Optimization**

In order to enhance the flexibility of the optimal guidance to accommodate the initial errors, perturbing and un-modeled items, the fuel consumption and state errors (including the landing error) for the powered descent guidance should be coordinated as optimally as possible to find the optimal guidance, which takes into account a variety of realistic factors.

**Problem 3 (system-level optimization).**

Find the optimal weight matrices \( S_1, R_1, S_2 \), and \( R_2 \) while minimizing the performance index

\[
\begin{align*}
\min & \quad J = \sum_{t_f}^{t_0} \left[ S_1 \delta x(t_f) + S_2 \delta x(t) \right]^T \left[ S_1 \delta x(t_f) + S_2 \delta x(t) \right] + \left[ R_1 u(t) + R_2 \delta u(t) \right]^T \left[ R_1 u(t) + R_2 \delta u(t) \right]
\end{align*}
\]

subject to

\[
\begin{align*}
R_1 \geq S_1, & \quad S_1 = S_1 \cdot I, \quad R_1 = R_1 \cdot I \\
R_2 \geq S_2, & \quad S_2 = S_2 \cdot I, \quad R_2 = R_2 \cdot I
\end{align*}
\]

where \( \delta x(t_f) \) represents the terminal state errors between the desired state and actual landing state, \( \delta x(t) \) denotes the real-time state errors between the reference state and the actual landing state, \( u(t) \) is the real-time nominal control, and \( \delta u(t) \) is the real-time regulation control. Adjusting \( S_1 \) ensures the landing accuracy and safety. Accommodating \( S_2 \) brings the process state errors within a practical control scope. Regulating \( R_2 \)
minimizes the major fuel consumption. Coordinating $R_3$ minimizes the additional control cost. This problem also subject to those constraints formulated in Problems 1 and 2.

To reduce the amount of optimization variables and calculation burden, weight matrices $S_i, R_i, S_2$, and $R_2$ are simplified into $S_1 = S_1 \cdot I_{6 \times 6}$, $R_1 = R_1 \cdot I_{3 \times 3}$, $S_2 = S_2 \cdot I_{6 \times 6}$, and $R_2 = R_2 \cdot I_{3 \times 3}$, respectively. Thus, this problem is to find the weight coefficients $S_i, R_i, S_2$, and $R_2$.

By solving this problem, the comprehensive performance (including the flexibility and robustness, and comprehensive optimality) of the guidance system will be further optimized according to the tracking control ability, the nominal landing error and the actual state errors. The lander will be guided along an optimal fuel and convergent state error profile to arrive at the selected landing site accurately and safely.

E. Optimization Procedure

The computational guidance is obtained through online solving these three optimization problems. As shown in Fig.1, these three problems are organized and integrated according to the MDO concept, and they have their own independent functions and required data exchanges. Problem 1 and Problem 2 are implemented according to their index weights given by Problems 3 and the actual states. The outputs of the whole guidance framework consist of the final results of Problem 1 and Problem 2. The calculation procedure is described as follows (see Fig. 2).

Step 1: Initialization calculation.

a) Substitute initial state $x(t_0)$ and target landing site $r_f$, and then solve Problem 1 with $S_i = 0$. Return optimal solution $x^*(t_0), u^*(t_0)$ to Problem 2, and return the index $J_{1}(R_i, S_i)$ to Problem 3.

b) Substitute initial state $x(t_0)$, and then solve Problem 2 with $R_2 = 0$. Return the index $J_{2}(R_2, S_2)$ to Problem 3. The optimal solution of Problem 2 will not be utilized in initialization calculation.

c) Substitute initial state $x(t_0)$, target landing site $r_f$, the indexes $J_{1}(R_i, S_i)$ and $J_{2}(R_2, S_2)$, solve Problem 3 and return optimal solution (i.e. initial coordinated weights) $S_i, R_i, S_2, R_2$.

Step 2: Subsequent iterative calculation.

a) If it is the first iteration, the calculation will be directly enabled once getting the initial coordinated weights. Otherwise, substitute the actual states $x(t)$, target landing site $r_f$, the indexes $J_{1}(R_i, S_i)$ and $J_{2}(R_2, S_2)$, and optimal control $u^*(t)$ and $\delta u(t)$, solve Problem 3 and return optimal solution (i.e. subsequent coordinated
weights) \( S_i, R_i, S_r, R_r \).

b) Substitute actual state \( x(t) \) and target landing site \( r_f \), and then solve Problem 1 with optimized weights \( S_i, R_i \). Return optimal solution \( x^*(t_0), u^*(t_0) \) to Problem 2, and return the index \( J_i(R_i, S_i) \) and optimal reference control \( u^*(t) \) to Problem 3.

c) Substitute the actual states \( x(t) \) and optimal reference profile \( x^*(t), u^*(t) \), and then solve Problem 2 with optimized weights \( S_r, R_r \). Return the index \( J_2(R_r, S_r) \) and optimal regulation control \( \delta u(t) \) to Problem 3.

The output of the computational guidance can thus be constructed \( u(t) = u^*(t) + \delta u(t) \). Then, return to Step 2(a) and continue the subsequent iterative calculation. The guidance procedure will stop once the lander reaches the planetary surface.

![Diagram of computational guidance](image)

**Fig.2. Calculation procedure of the computational guidance**

**Remark 1**: In Refs.[20][21][22], authors proved that the Problems 1 and 2 defined above are solvable and their
feasible solutions are existing. Thus, the Problem 3 is also solvable and its feasible solution also exists. Here, the three Problems are solved using the existing solvers listed in Table 1 to find their optimal solutions. Sequential quadratic programming (SQP) [23] is adopted to solve Problems 1 and 2, and the pointer automatic optimizer (PAO) [24] is employed to solve Problem 3.

<table>
<thead>
<tr>
<th>Optimization problem</th>
<th>Problem feature</th>
<th>Selected optimizer</th>
<th>Algorithm feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem 1</td>
<td>Quadratic index</td>
<td>SQP</td>
<td>Quadratic programming</td>
</tr>
<tr>
<td>Problem 2</td>
<td>LQR</td>
<td>SQP</td>
<td>Quadratic programming</td>
</tr>
<tr>
<td>Problem 3</td>
<td>Adaptive coordination</td>
<td>PAO</td>
<td>Flexibly global optimization</td>
</tr>
</tbody>
</table>

Some of the frequently used optimizers are encapsulated in iSIGHT® environment and hidden from the user [25], the corresponding optimizers are directly selected according to the features of the three optimization problems and algorithm functions. As a direct numerical optimization technique, the SQP method builds a quadratic approximation to the Lagrange function and linear approximations to all output constraints at each iteration, and a quadratic programming problem is solved to find an improved solution, until the final convergence to the optimum solution [23, 25]. PAO is an automatic optimization engine that controls a set of standard optimization techniques including SQP, simplex algorithm, genetic algorithm, and a linear solver to find the global optimum solution [24, 25].

**Remark 2:** These existing optimizers listed in Table 1 approximate an optimal control problem by a parameter optimization problem (i.e., a nonlinear programming (NLP) problem), and then solve the NLP problem. The resulting approximation for the optimal control problem is an NLP problem of the form [25]:

$$\min \ J(q)$$

subject to

$$C_i(q) = 0, \quad i = 1, L, N_{ec}$$ (28)

$$C_i(q) \leq 0, \quad i = N_{ec} + 1, L, N_e$$ (29)

where $q$ denotes the vector of parameters, $C_i$ are the constraint functions, $N_e$ represents the total number of the constraints, $N_{ec}$ stands for the number of equality constraints, and $(N_e-N_{ec})$ is the number of inequality constraints.

**Remark 3:** In order to confirm that the converged solution is a feasible solution or the extreme point of the problem, an indirect method is implemented for comparison. Referring to Ref.[26], a modification of Powell’s
hybrid algorithm [27] that is a combination of Newton’s method and the method of the gradient is implemented to solve nonlinear optimization problems (nonlinear equations). In fact, the algorithm is similar in nature to the algorithm implemented by the default solver of MATLAB’s fsolve function.

IV. Simulations and Results

A. Simulation Platforms

To validate the effectiveness of the CO-based computational guidance for the planetary powered descent, the iSIGHT® and MATLAB® environments are adopted to implement the numerical simulations and analyses. All three optimization problems are defined in MATLAB. These MATLAB modules are integrated into the iSIGHT optimization platform to implement the optimization computations and output the optimal results. All simulation computations are performed on a laptop PC with an Intel Core (TM) i7-3610QM CPU at 2.3 GHz and 4 GB of memory.

Apollo lunar descent guidance and convex optimization based planetary powered descent guidance are adopted here for comparative analysis in the simulations. As a classical approach in engineering practice, Apollo guidance law [4-6] is derived from polynomial approximation of fuel optimal profile, which has excellent real-time, good terminal state accuracy and near optimal fuel-consumption. The commanded thrust acceleration and thrust angle can be online determined based on the actual states and target states. As a newly emerging technology, convex optimization based planetary powered descent guidance [13, 19, 20-22] is generated through formulating the optimal guidance problem as a finite-dimensional second-order cone programming problem, and solving it in polynomial time via currently available direct numerical algorithms. A global optimum can be efficiently obtained with deterministic stopping criteria and prescribed level of accuracy [20, 22]. In convex optimization, minimum-fuel and minimum-landing-error powered descent guidance algorithms [20, 22] will be respectively employed by Scenario 1 and 2 for comparison.

B. Simulation Results

1. Case I

The fuel optimality of the proposed CO-based computational guidance approach is verified in this case. The MATLAB convex programming (CVX) software is adopted for the convex optimization steps [28]. The default configuration of the CVX is used, and the solver SDPT3 4.0 within CVX is called. The Apollo guidance
algorithm calculates the polynomial coefficients according to the initial and target states, and the perturbation is set as zero here. The parameters used for Scenario 1 are listed as follows: \( r_f = [0, 0, 0]^T \, \text{m} \), \( v_f = [0, 0, 0]^T \, \text{m/s} \), \( m_0 = 1900 \, \text{kg} \), \( m_{dry} = 1500 \, \text{kg} \), \( I_{sp} = 225 \, \text{s} \), \( 0.3 < \kappa < 0.8 \), \( \| r \|_{max} = 16423 \times \cos(27^\circ) = 14633 \, \text{N} \), and \( \mu = 4.282829 \times 10^{13} \, \text{m}^3/\text{s}^2 \). 1000-run Monte Carlo simulations are driven by the random initial state value listed in Table 2.

### Table 2 Initial state value in Monte Carlo simulation.

<table>
<thead>
<tr>
<th>State variable</th>
<th>Mean value</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r_0 )</td>
<td>[1900, 1200, 3100] , m</td>
<td>[100, 100, 100] , m</td>
</tr>
<tr>
<td>( v_0 )</td>
<td>[30, 20, -40] , m/s</td>
<td>[2, 2, 2] , m/s</td>
</tr>
</tbody>
</table>

The state variables during the guidance process of three algorithms are shown in Fig. 3. All guidance algorithms satisfy the prescribed constraints. The results of our approach and convex programming are similar but superior to that of the Apollo guidance. The time costs are 5.189 s ~ 9.658 s for our approach, 41.160 s ~ 49.971 s for the convex programming, and less than 0.001 s for the Apollo guidance. Figs. 3(a), 3(b), 3(c), and 3(d) show that the spacecraft lands on the target site gently. From Fig. 3(e), the mass cost of the two optimal guidance programs are less than that of the Apollo guidance. Figure 3(f) depicts that the thrust magnitude of the two optimal guidance algorithms remains at a higher level during the most parts of the entire flight while a lower magnitude of the thrust curve occurs between ~16 s and ~33 s. The reason behind this phenomenon lies in that the inertia and gravity help the lander adjust the descent state to the optimal descent state during this process.
The CO-based computational guidance provides much more real-time data than the convex programming, although three optimization algorithms are encapsulated and used in the proposed CO framework. In addition, the optimization results of the CO-based computational guidance are better than those of the other two approaches. As noted previously, the performance of the proposed CO-based computational guidance depends on the use of simple and effective optimization algorithms and coordinated optimization to realize rapid convergence and high-efficient optimal guidance.

The corresponding results of optimized parameters ($S_1$, $R_1$, $S_2$, $R_2$) are depicted in Fig. 4. It can be seen from Fig. 4 that the changing trends of the weight coefficients are complementary to each other. The reason is that these weight coefficients are optimized to balance the state errors and control cost. Figures 4(a) and (b) show that the
control cost and initial state errors are the major considerations for optimal trajectory generation. In contrast, Figures 4(c) and (d) illustrate the state errors are the main consideration for optimal control, and the control cost considered here is to compensate the overdraft during trajectory generation.

The corresponding total fuel consumption of each approach is illustrated in Fig.5. To confirm the converged solution is the extreme point or a feasible solution of the problem, an indirect method is implemented for comparison [26, 27]. As shown in the figure, an overwhelming majority of the results for the proposed optimal computational guidance and convex programming algorithm are consistent with the extreme point of the problem obtained via indirect method. However, the Apollo guidance is only a sub-optimal approach from the view point of fuel consumption.
2. Case II

The comprehensive performance of the proposed CO-based guidance approach is tested in this case by implementing 1000-run Monte Carlo simulations. In the past missions, the distance between the initial target site and the final target site is usually no more than 50 m during planetary powered descent [5-7]. Just like the CE-3 lunar landing mission [6], this distance is less than 10 m. To demonstrate the retargeting flexibility of the proposed guidance algorithm, we intentionally assume that the initial target site ([70, -80, 0] m) is more than 100 m away to the final target site ([0, 0, 0] m) here. It means the value of \( r_f \) at the retargeting point would changes from [70, -80, 0] m to [0, 0, 0] m. In order to reflect retargeting and hazard avoidance maneuver during planetary powered descent, the retargeting trigger is enabled once the lander reaches the horizontal distance of 400 m away from the original target landing site. The discussed powered descent involves several kilometers’ flight scope near the surface of a spherical central planet, and the J\(_2\) perturbing force and the irregular gravity change are very small and usually ignored [2]. The aerodynamic perturbing force and thrust deviation would be the main perturbations [2, 4]. In the MSL and CE-3 missions, the maximum thrust deviation is about 5% of the maximum thrust [4, 5]. The initial velocity of planetary powered descent is usually less than 100 m/s, the total wind speed on Mars is usually less than 50 m/s, and the atmospheric density of the Martian surface is about 1.25\( \times 10^{-2} \) kg/m\(^3\) [2]. When the reference area of the lander is assumed to be \( \pi \) m\(^2\), the total aerodynamic perturbing force would be no more than 441 N, i.e. about 3% of the maximum thrust \( \|T\|_{\text{max}} \). Thus, the magnitude of the main perturbation could reach about 8% of the maximum thrust \( \|T\|_{\text{max}} \). Therefore, the total perturbation is.
conservatively set to be a random acceleration within 10% of the maximum thrust acceleration \[ \frac{\nabla F_{\text{max}}}{m} \] to further verify the flexibility and robustness of the proposed guidance approach. The initial state parameters used for Case 2 are listed in Table 3. The other simulation parameters are the same as that of Case 1. The state changes during the guidance process are shown in Fig. 6.

Table 3 Initial state value in Monte Carlo simulation.

<table>
<thead>
<tr>
<th>State variable</th>
<th>Mean value</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathbf{r}_0 )</td>
<td>[150, -850, 2500] m</td>
<td>[100, 100, 100] m</td>
</tr>
<tr>
<td>( \mathbf{v}_0 )</td>
<td>[0, 145, -60] m/s</td>
<td>[5, 5, 5] m/s</td>
</tr>
</tbody>
</table>

(a) ![Graph](image1.png) (b) ![Graph](image2.png) (c) ![Graph](image3.png) (d) ![Graph](image4.png)
Fig. 6 Monte Carlo simulation results for Scenario 2: (a) two-dimensional trajectory, (b) altitude history, (c) landing site dispersion, (d) velocity history, (e) lander mass, and (f) thrust level.

The position and velocity histories for the set of 1000 Monte Carlo simulations are shown in Figs. 6 (a), (b) and (d). The time costs of the three guidance algorithms are similar with that mentioned in Case 1. The terminal landing footprint dispersion is depicted in Fig. 6(c). Additionally, Figs. 6(e) and 6(f), respectively, display the mass fuel consumption and the thrust command histories resulting from the Monte Carlo simulations. Figs. 6(c) and 6(d) illustrate that the terminal position and velocity are close to zero. These results show that the chosen gains are near optimal from a fuel-consumption perspective, and that they are robust to perturbations and off-nominal conditions. It can be concluded that the flexibility and robustness are not provided by traditional reference-tracking guidance and optimization algorithms (e.g., SQP and GPOPS) [29] but require real-time optimization computation.

As these weight coefficients ($S_1$, $R_1$, $S_2$, $R_2$) are optimized to balance the state errors and control cost, the changing trends of weight coefficients are nearly complementary to each other. Figs. 7(a) and 7(b) depict the control cost, initial and retargeting state errors are the major considerations for optimal trajectory generation. On the contrary, Figs. 7 (c) and 7(d) illustrate the state errors are the main consideration for optimal control, and the control cost considered here is to compensate the overdraft during trajectory generation. The overdraft is to minimize the terminal state error (landing error). Meanwhile, it can be also noticed from Fig. 7 that there are few cases the weight coefficients are not complementary. This phenomenon reflects the uncertainties still have adverse effect on comprehensive performance of the proposed computational guidance algorithm, which means there is a need to consume more fuel to ensure safe and precise landing in some extreme cases.
Fig. 7 Results of optimized parameters ($S_1, R_1, S_2, R_2$) in Scenario 1: (a) $S_1$, (b) $R_1$, (c) $S_2$, (d) $R_2$.

The corresponding total fuel consumption of each approach is illustrated in Fig. 8. To confirm the converged solution is the extreme point or a feasible solution of the problem, an indirect method is implemented for comparison [26, 27]. The dynamics with known uncertainty item is used for indirect method (i.e. the value of uncertainty item is known for indirect method), while the nominal dynamics adopted by other three approaches still don’t consider the uncertainty item. As shown in the figure, most results of the proposed computational guidance still agree with well with the extreme point of the problem obtained via indirect method. Meanwhile, most results of the convex programming algorithm fail to converge to the actual global optimums. As the nominal dynamics doesn’t include uncertainty item and the convex programming guidance algorithm cannot adaptively adjust its parameters to online proximate the actual dynamics, the converged optimums of the convex
programming algorithm are not the real optimums, but close to those of the Apollo guidance.

![Fig. 8 Total fuel-consumption.](image)

C. Discussion

The essential reasons for the good performance of the proposed computational guidance are discussed in this section. According to the simulation results illustrated above, the new findings regarding the performance the CO-based computational guidance can be summarized and discussed as follows.

Like the convex optimization and Apollo guidance algorithms, the proposed computational guidance algorithm is also insensitive for the initial state errors due to online calculating and generating the guidance command according to the actual flight state, constraints, and index. However, the proposed guidance algorithm has a superior comprehensive performance compared with convex optimization and Apollo guidance algorithms.

For the traditional fuel-optimal guidance problem from a given initial state to a target landing site (Case I), the CO-based computational guidance can reach global optimal solution for minimum-fuel powered descent. In Case II, under the conditions of uncertainties, the proposed computational guidance approach can reach higher landing accuracy and lower fuel-consumption than that of the other two guidance algorithms. Meanwhile, the time cost of the proposed computational guidance algorithm is lower than that of the convex optimization algorithm.

The main reasons for the superior performance can be found from the design point view. The powered descent guidance problem simultaneously considers fuel-consumption, landing accuracy, robustness against uncertainties, and flexibility for retargeting. First of all, the complex guidance problem is broke into three simple optimization problems. It has been proved by previous work that the feasible solutions and corresponding optimums of these
three problems can be easily found using existing algorithms [3-5, 20-22]. Thus, the proposed computational guidance approach can rapidly generate an optimal guidance command. Secondly, the CO approach, as a MDO strategy, has the inherent properties to ensure high-efficiently converge to the global optimum of the system. Thus, the proposed guidance procedure can find the global optimum in most cases. Thirdly, the uncertainties (e.g. the initial error, dynamics uncertainty, retargeting) are suppressed and compensated through adjusting the absolute magnitude of the weight coefficients of the state errors and control cost. The relative proportion of the weight coefficients is balanced to realize the optimum of the system. Essentially, the current and terminal state errors are introduced to make the guidance algorithm consider the control error and landing error accordingly. Therefore, the proposed guidance method simultaneously has the functions of predictor-corrector guidance and reference-tracking guidance.

V. Conclusions

A novel computational guidance approach based on CO is presented for planetary powered descent in this paper. Comprehensively considering the landing error, fuel cost, and flexibility, the computational guidance for powered descent is transformed to the online solution of three optimization problems. Three optimization algorithms are hierarchically encapsulated into the computational guidance framework. Using this guidance framework, we have tested and verified that the combination of the three problems and optimization algorithms is effective. The results of Monte Carlo simulations show that the CO-based computational guidance is superior to convex optimization. Meanwhile, the proposed computational guidance is robust against the uncertainties and retargeting to a new landing site when the original target site is unacceptable for safe landing. This paper only uses one method in the field of MDO to design a computational guidance procedure for the planetary powered descent problem. Other methods in the region of MDO are also expected to be used to deal with optimal guidance problems.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 11672126), the Opening Grant from the Key Laboratory of Space Utilization, Chinese Academy of Sciences (Grant No. LSU-2016-07-01), Funding of Jiangsu Innovation Program for Graduate Education (Grant No. KYZZ16_0170), Funding for Outstanding Doctoral Dissertation in NUAA (Grant No. BCXJ16-10), and the Fundamental Research Funds for the Central Universities. The authors fully appreciate their financial supports. The first author also would like to acknowledge the financial support provided by the China Scholarship Council for his study at the
University of Arizona (Grand No. 201706830055).

References


