doi: 10.82124/CEAS-GNC-2026-032 CEAS-GNC-2026-032

EuroGNC Navigation Guidance Control CEAS Madrid, Spain May 5th-7th uc3m Universidad Carlos III de Madrid AIAA 2026

Onboard Trajectory Optimization for Rocket Landing Considering Aerodynamic Lift and Drag

Jae-Il Jang Ph.D. Student, Korea Advanced Institute of Science and Technology, Department of Aerospace Engineering, 34141, Daejeon, South Korea. jji249@kaist.ac.kr

Dain Jeong M.S. Candidate, Korea Advanced Institute of Science and Technology, Department of Aerospace Engineering, 34141, Daejeon, South Korea. dainj0124@kaist.ac.kr

Jaewook Shim M.S. Candidate, Korea Advanced Institute of Science and Technology, Department of Aerospace Engineering, 34141, Daejeon, South Korea. jaewooki1@kaist.ac.kr

Ki-Wook Jung Ph.D. Candidate, Korea Advanced Institute of Science and Technology, Department of Aerospace Engineering, 34141, Daejeon, South Korea. jkw1031@kaist.ac.kr

Min-Jea Tahk Professor Emeritus, Korea Advanced Institute of Science and Technology, Department of Aerospace Engineering, 34141, Daejeon, South Korea. mj-tahk317@gmail.com

Chang-Hun Lee Associate Professor, Korea Advanced Institute of Science and Technology, Department of Aerospace Engineering, 34141, Daejeon, South Korea. lckdgns@kaist.ac.kr

ABSTRACT

This paper presents a trajectory optimization problem for rocket landing that accounts for aerodynamic lift and drag, along with its onboard implementation. Considering a rocket landing on Earth, where aerodynamic effects cannot be neglected, the problem formulation incorporates the coupled drag and lift forces parameterized by the angle of attack. In addition, the formulated problem constrains the thrust direction to be aligned with the vehicle attitude, thereby reflecting the typical characteristic of rocket systems with small thrust deflection angle. This formulation preserves additional control authority for rejection of unexpected disturbances. The optimal solution is computed using sequential convex programming, and its algorithm is implemented on an onboard computing platform using an in-house developed custom convex optimization solver to demonstrate its practical feasibility.

Keywords: Reusable Launch Vehicle, Powered Descent Guidance, Onboard Trajectory Optimization, Custom Solver

Nomenclature

X, Y	=	Position of the vehicle represented in the reference frame
V	=	Speed of the vehicle
γ	=	Flight path angle of the vehicle
m	=	Mass

α = Angle of attack of the vehicle
t = Time
ρ = Air Density
S_{ref} = Reference area for aerodynamics coefficient
M = Mach number
κ = $\frac{1}{2}\rho S_{ref}$
P_{atm} = Atmospheric pressure at current altitude
A_e = Exit area of the engine nozzle
m_{dry} = Dry mass of the vehicle
t_f = Final time in the optimal control problem

1 Introduction

Reusable launch vehicles have become an essential option for economically viable launch operations. Among the various flight phases required for recovery such as the boost-back burn and reentry burn, the landing burn is particularly critical, as it should eliminate any remaining dispersion and ensure a soft touchdown at the designated landing site [1]. To accomplish this goal, powered descent guidance is employed to generate the commands required to drive the vehicle to the target states at landing. A representative example of such a guidance scheme is the Apollo powered-descent guidance [2]. It analytically computes the acceleration command based on the current position and velocity to reach the desired position, velocity and thrust acceleration at a specified time of flight. However, such guidance laws derive their commands by assuming the acceleration profile to be a polynomial function of time, which may lead to suboptimal solutions. Moreover, these approaches have difficulty in explicitly incorporating path constraints, which limits the feasible region [3], and also face challenges in handling nonlinear dynamics. To overcome these limitations of analytical guidance laws, approaches that solve an trajectory optimization problem in real time on the onboard computer have been proposed to generate landing trajectories and to use them for powered descent guidance. In particular, convex optimization has attracted significant attention for such applications, as it offers guaranteed global optimality and algorithms with polynomial-time convergence, making it well suited for safe and real-time solution [4]. A notable example is the study of powered descent guidance for Mars landing that employs a convexification technique known as lossless convexification [5]. In that work, the optimal control problem for Mars landing is transformed into a convex optimization problem by exactly relaxing the nonconvex constraint arising from the lower bound on the thrust magnitude. However, only linear dynamics are considered by neglecting the effects of aerodynamic forces, taking into account the thin Martian atmosphere. In contrast, on Earth, where aerodynamic effects cannot be neglected, such an approach may not be suitable for practical operations [1]. To address the resulting nonconvexity caused by the nonlinear dynamics, a method to utilize sequential convex programming (SCP) has been proposed to solve the nonconvex problem using convex optimization techniques. Using this framework, several studies [6–8] have optimized landing trajectories while accounting for the aerodynamic effects. Nevertheless, most of these studies consider only aerodynamic drag, and relatively few have employed coupled drag-and-lift models parameterized by angle of attack. Although a drag and lift model coupled through the angle of attack is considered in [9], the problem is formulated such that the thrust direction could be set independently of the vehicle attitude and angle of attack. Such a formulation may not be practically applicable to rocket applications, where thrust vectoring deflection angles are typically limited to small values.

Another major challenge of trajectory optimization-based powered descent guidance is the need to solve optimization problems in real time on resource-constrained onboard computers. Therefore, a convex optimization solver capable of providing reliable and fast solutions is essential for such computationally limited environments. Among the available methods, the interior-point method (IPM) [10] is widely recognized for its robustness and efficiency in solving small to medium scale problems such as trajectory

optimization applications. Moreover, a custom solver implementation can further exploit the specified problem structure by eliminating redundant mathematical operations and computational branches during execution. In addition, it allows all data to be statically allocated in memory, making it well suited for onboard applications [11].

Motivated by these considerations, this paper presents a trajectory optimization-based powered descent guidance that accounts for aerodynamic lift and drag. Given that the thrust vectoring deflection angles are typically small, the problem is formulated such that the thrust direction is constrained by the current attitude of the vehicle rather than being treated independently. To avoid singularities in the flight path dynamics at the terminal time and to ensure a safer landing, a two-phase structure is employed: the first phase aims to reduce trajectory dispersion, while the second phase aims to achieve a safe and soft touchdown. Furthermore, numerical experiments are performed on an embedded onboard computer using an in-house developed custom SOCP solver [12], thereby demonstrating the practical applicability of the proposed approach in real-world scenarios.

2 Problem Formulation

This section presents the formulation of an optimal control problem for obtaining a landing trajectory and guidance commands, and describes the convexification process of the formulated problem through a SCP technique.

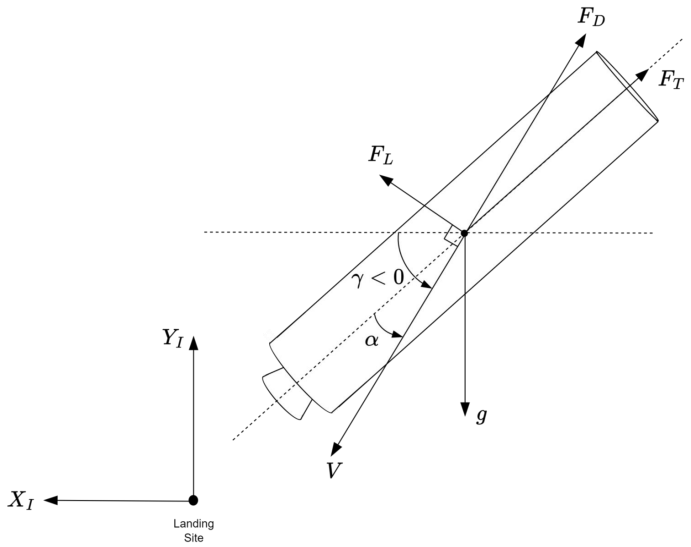


Fig. 1 Landing Frame

2.1 Dynamics

Prior to presenting the dynamics model used in the guidance problem, we first introduce the assumptions underlying the problem formulation.

Assumptions

1) Non-rotating Earth and constant gravitational acceleration are assumed, as the flight distance and duration during the landing-burn phase are relatively short.
2) A point mass dynamics model is employed in a 2-dimensional planar reference frame.
3) The reference frame, represented in Fig. 1, is defined as a planar coordinate system with its origin located at the landing site, where the x-axis points in the downrange direction and the y-axis represents altitude. This frame is treated as an inertial frame.

4) The thrust is assumed to be aligned with the longitudinal axis of the vehicle's body frame, since the deflection angle introduced by thrust vector control is small.

The translational dynamics of the vehicle expressed in the landing frame can be formulated as follow:

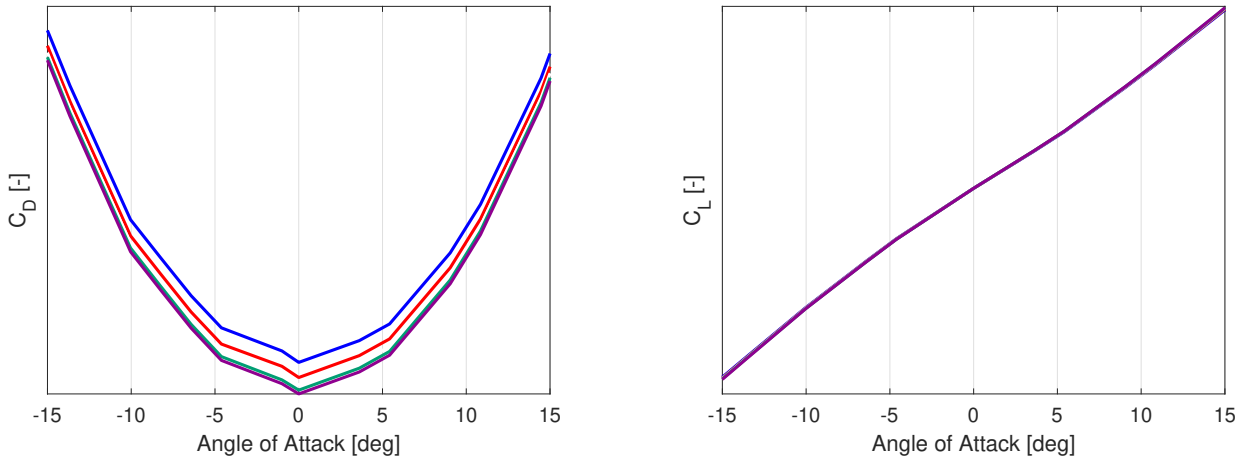
$$\dot{X} = V\cos\gamma \tag{1a}$$
$$\dot{Y} = V\sin\gamma \tag{1b}$$
$$\dot{V} = \frac{-F_D - F_T\cos\alpha}{m} - g\sin\gamma \tag{1c}$$
$$\dot{\gamma} = \frac{F_L - F_T\sin\alpha}{mV} - \frac{g\cos\gamma}{V} \tag{1d}$$
$$\dot{m} = -\frac{F_T + P_{atm}A_e}{g_0 I_{sp}} \tag{1e}$$

The aerodynamic drag F_D and lift force F_T are modeled using a aerodynamic coefficients $C_D(M,\alpha)$, and lift curve slope $C_{L,\alpha}(M) := \frac{\partial C_L}{\partial \alpha}$, as follow.

$$F_D = \kappa C_D(M,\alpha)V^2 \tag{2}$$

$$F_L = \kappa C_{L,\alpha}(M)V^2\alpha \tag{3}$$

As shown in Fig. 2, the induced drag effect due to nonzero α is captured in the drag coefficient C_D, and the near-linear variation of C_L with respect to α indicates that the lift force can be reasonably modeled using the lift-curve slope $C_{L,\alpha}$.



(a) $C_D - \alpha$ **Curve** **(b)** $C_L - \alpha$ **Curve**

Fig. 2 Aerodynamic Coefficient by Angle of Attack

The thrust is modeled by incorporating the effect of thrust loss caused by ambient pressure, as follows.

$$F_T = T_{vac} - P_{atm}A_e \tag{4}$$

Note that, according to Assumption 4), the thrust direction in Eqs. (1c) and (1d) is expressed as a function of the angle of attack. In this formulation, the required accelerations to satisfy the guidance objectives are generated by attaining the corresponding trim angle of attack, rather than by directly commanding TVC deflections. As a result, the available TVC deflections are conserved in the guidance block, providing additional control authority for guidance error compensation as well as for attitude stabilization under external disturbances and model uncertainties.

2.2 Optimal Control Problem

Along with the dynamics model presented in the previous section, several constraints are considered in the guidance problem to ensure a physically feasible and safe landing solution. First, to ensure that the propellant consumption does not exceed the available amount, a constraint on the final mass is imposed as follows.

$$m(t_f) \geq m_{dry} \tag{5}$$

Considering the engine's operable throttle range and the thrust loss induced by ambient pressure, the thrust magnitude is constrained as follows.

$$T_{min,vac} - P_{atm}A_e \leq F_T \leq T_{max,vac} - P_{atm}A_e \tag{6}$$

Furthermore, a thrust magnitude rate constraint is imposed to avoid abrupt throttle command.

$$-\dot{T}_{max} \leq \dot{F}_T \leq \dot{T}_{max} \tag{7}$$

To maintain the aerodynamic load within its allowable range, the following constraints are imposed on the use of the angle of attack.

$$-\alpha_{max} \leq \alpha \leq \alpha_{max} \tag{8}$$

Finally, a rate constraint on the AoA is incorporated to ensure continuous and smooth AoA command profile.

$$-\dot{\alpha}_{max} \leq \dot{\alpha} \leq \dot{\alpha}_{max} \tag{9}$$

The initial condition is set to the current values of the vehicle states as,

$$X(t_0) = X_0, \quad Y(t_0) = Y_0, \quad V(t_0) = V_0, \quad \gamma(t_0) = \gamma_0, \quad m(t_0) = m_0 \tag{10}$$

while terminal boundary conditions are enforced to satisfy the desired final position, speed, and flight path angle.

$$X(t_f) = X_f, \quad Y(t_f) = Y_f, \quad V(t_f) = V_f, \quad \gamma(t_f) = \gamma_f \tag{11}$$

Finally, an objective function is considered to minimize the fuel consumption during the landing flight, and the corresponding optimal control problem is formulated in $\mathcal{P}1$.

In most conventional landing problems, it is common to set $V_f = 0$ along with $Y_f = 0$. However, this setting leads to a singularity in the flight path angle dynamics Eq. (1d) as $t \to t_f$. Furthermore, due to the nature of the fuel-minimization problem [13], the optimal solution typically commands full throttle until just before touchdown, a strategy often referred to as a "suicide burn". Such a touchdown can impart a significant impact on the vehicle in the presence of model uncertainties and guidance error, posing a high risk of a failed soft landing. To address these issues, this study introduces a two-phase structure for the landing-burn segment: a *landing-flight phase* aimed at reducing trajectory dispersion and a *touchdown phase* dedicated to achieving the final soft landing. In the landing-flight phase, the vehicle flies toward the vicinity of the landing site and reduces its speed based on the solution of the optimal control problem defined by $\mathcal{P}1$. In the touchdown phase, the vehicle regulates its acceleration to track a reference profile designed to achieve a safe and soft landing at the designated site. Based on this profile, the terminal conditions on position and velocity in Eq. (11), i.e. the values of X_f, Y_f, V_f and γ_f, can be determined. For example, if the touchdown phase is designed to be initiated at a certain altitude Y_{td} and to use a constant vertical acceleration a_{td} as the reference profile, the terminal conditions are given by

$$\begin{aligned} X_f = 0, \quad Y_f = Y_{td} > 0 \\ V_f = \sqrt{2a_{td}Y_{td}}, \quad \gamma_f = 90^\circ \end{aligned} \tag{12}$$

This kind of two-phase strategy both eliminates the singularity in the dynamics at the terminal time of the optimal control problem and improves the safety of the landing by mitigating the pre-touchdown acceleration.

$\mathcal{P}1$: Optimal Control Problem for Rocket Landing

$$
\begin{aligned}
\underset{F_T,\alpha}{\text{minimize}} \quad & -m(t_f) \\
\text{subject to} \quad & \dot{X} = V\cos\gamma \\
& \dot{Y} = V\sin\gamma \\
& \dot{V} = \frac{-F_D - F_T\cos\alpha}{m} - g\sin\gamma \\
& \dot{\gamma} = \frac{F_L - F_T\sin\alpha}{mV} - \frac{g\cos\gamma}{V} \\
& \dot{m} = -\frac{F_T + P_{atm}A_e}{g_0 I_{sp}} \\
& X(t_0) = X_0,\ Y(t_0) = Y_0,\ V(t_0) = V_0,\ \gamma(t_0) = \gamma_0,\ m(t_0) = m_0 \\
& X(t_f) = X_f,\ Y(t_f) = Y_f,\ V(t_f) = V_f,\ \gamma(t_f) = \gamma_f \\
& m(t_f) \geq m_{dry} \\
& T_{min,vac} - P_{atm}A_e \leq F_T \leq T_{max,vac} - P_{atm}A_e \\
& -\dot{T}_{max} \leq \dot{F}_T \leq \dot{T}_{max} \\
& -\alpha_{max} \leq \alpha \leq \alpha_{max},\ -\dot{\alpha}_{max} \leq \dot{\alpha} \leq \dot{\alpha}_{max}
\end{aligned}
\tag{13}
$$

2.3 Convexification through Sequential Convex Programming

The resulting optimal control problem $\mathcal{P}1$ is nonconvex owing to the nonlinear dynamics, which makes it unsuitable for real-time onboard solution. To address this challenge, this study applies a SCP approach [7] to convexify the problem. For notational convenience, let the state and control input be denoted by the vector $x = [X,\ Y,\ V,\ \gamma,\ m] \in \mathbb{R}^5$ and $u := [F_T,\ \alpha] \in \mathbb{R}^2$, respectively, and express the dynamics in Eq. (1) as follows.

$$\dot{x} = f(x,u) \tag{14}$$

where $\dot{x}$ denotes the derivative of x with respect to time t (i.e., dx/dt). To efficiently handle the free-final time problem, we introduce the normalized time variable $\tau \in [0,1]$, and apply time-dilation [7] to Eq. (14).

$$x' = t_f \cdot f(x,u) := F(t_f,x,u) \tag{15}$$

where x' denotes the derivative of x with respect to the normalized time τ (i.e., $dx/d\tau$). The actual time t can be recovered from the τ through the following mapping.

$$t = t_f\tau \tag{16}$$

To construct the convexified subproblem for SCP, the time-dilated dynamics are linearized as

$$x' \simeq A(\bar{t}_f,\bar{z})x + B(\bar{t}_f,\bar{z})u + D(\bar{t}_f,\bar{z})t_f + w(\bar{t}_f,\bar{z}) \tag{17}$$

where the overbar denotes the reference point for the linearization, $\bar{z} := [\bar{x},\ \bar{u}] \in \mathbb{R}^7$, and the coefficient matrices are given as

$$A(\bar{t}_f,\bar{z}) := \left.\frac{\partial F}{\partial x}\right|_{(\bar{t}_f,\bar{z})}, \quad B(\bar{t}_f,\bar{z}) := \left.\frac{\partial F}{\partial u}\right|_{(\bar{t}_f,\bar{z})}, \quad D(\bar{t}_f,\bar{z}) := \left.\frac{\partial F}{\partial t_f}\right|_{(\bar{t}_f,\bar{z})} \tag{18}$$

$$w(\bar{t}_f, \bar{z}) := F(\bar{t}_f, \bar{z}) - A\bar{x} - B\bar{u} - D\bar{t}_f \tag{19}$$

To transcribe the optimal control problem into a parameter optimization problem, the linearized dynamics are discretized with N nodes using the trapezoidal method.

$$H_k^- x_k + G_k^- u_k + H_k^+ x_{k+1} + G_k^+ u_{k+1} + M_k = E_k \qquad (k = 0, 1, \cdots, N-1) \tag{20}$$

where the coefficient matrices are defined as

$$A_k := A(\bar{t}_f, \bar{z}_k), \quad B_k := B(\bar{t}_f, \bar{z}_k), \quad D_k := D(\bar{t}_f, \bar{z}_k), \quad w_k := w(\bar{t}_f, \bar{z}_k) \tag{21}$$

$$\begin{aligned} H_k^- &:= -I - \frac{\Delta\tau}{2} A_k, \quad G_k^- := -\frac{\Delta\tau}{2} B_k \\ H_k^+ &:= I - \frac{\Delta\tau}{2} A_{k+1}, \quad G_k^+ := -\frac{\Delta\tau}{2} B_{k+1} \\ M_k &:= -\frac{\Delta\tau}{2} \left(D_k + D_{k+1}\right), \quad E_k := \frac{\Delta\tau}{2} \left(w_k + w_{k+1}\right) \end{aligned} \tag{22}$$

with $\Delta\tau := 1/N$, $\tau_k = k/N$, and $\bar{z}_k := \bar{z}(\tau_k)$.

To prevent artificial unboundedness [4] introduced by linearization, a soft trust region term [7] with weight parameter W_{t_f} and $W_{tr,k}$

$$J_{tr} = W_{t_f}(t_f - \bar{t}_f)^2 + \sum_{k=0}^{N} (z_k - \bar{z}_k)^T W_{tr,k} (z_k - \bar{z}_k) \tag{23}$$

is added to the objective function.

Finally, the convexified subproblem for applying SCP is formulated in $\mathcal{P}2$. Its optimal solution at each iteration serves as the linearization reference for the next iteration, and this process is repeated until convergence is achieved. For the initial reference, a naive straight line trajectory connecting the initial and terminal conditions are used in this study.

$\mathcal{P}2$: Convex Subproblem for Rocket Landing

$$\begin{aligned} \underset{z_0, \cdots z_N, t_f}{\text{minimize}} \quad & -m_N + J_{tr} \\ \text{subject to} \quad & H_k^- x_k + G_k^- u_k + H_k^+ x_{k+1} + G_k^+ u_{k+1} + M_k = E_k \quad (k = 0, 1, \cdots, N-1) \\ & x_0 = [X_0,\ Y_0,\ V_0,\ \gamma_0,\ m_0]^T \\ & x_{N,1:4} = [X_f,\ Y_f,\ V_f,\ \gamma_f]^T \\ & m_N \geq m_{dry} \\ & T_{min,vac} - P_{atm} A_e \leq F_{T,k} \leq T_{max,vac} - P_{atm} A_e \quad (k = 0, 1, \cdots, N) \\ & -\dot{T}_{max} t_f \Delta\tau \leq F_{T,k+1} - F_{T,k} \leq \dot{T}_{max} t_f \Delta\tau \quad (k = 0, 1, \cdots, N-1) \\ & -\alpha_{max} \leq \alpha_k \leq \alpha_{max} \quad (k = 0, 1, \cdots, N) \\ & -\dot{\alpha}_{max} t_f \Delta\tau \leq \alpha_{k+1} - \alpha_k \leq \dot{\alpha}_{max} t_f \Delta\tau \quad (k = 0, 1, \cdots, N-1) \end{aligned} \tag{24}$$

3 Onboard Implementation

This section presents the onboard implementation framework for the landing trajectory optimization problem formulated in the previous section. Since onboard computers typically operate under limited computational power and memory, this study employs an in-house developed custom convex optimization

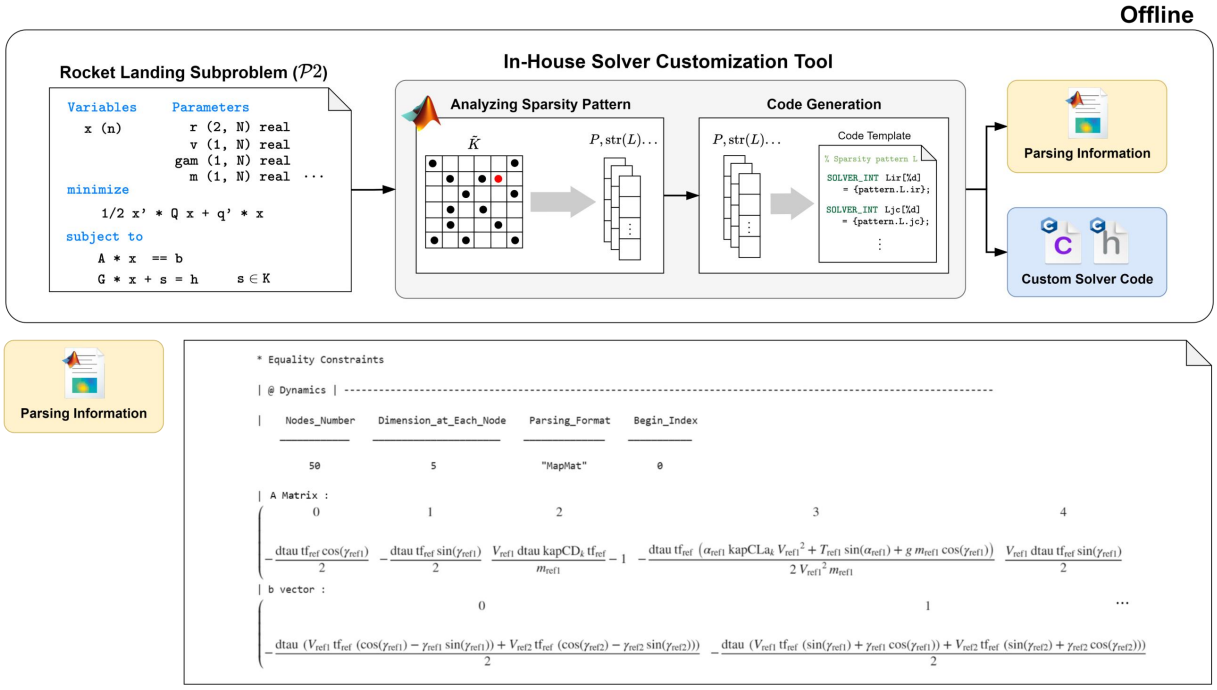


Fig. 3 Solver Customization Process in Offline

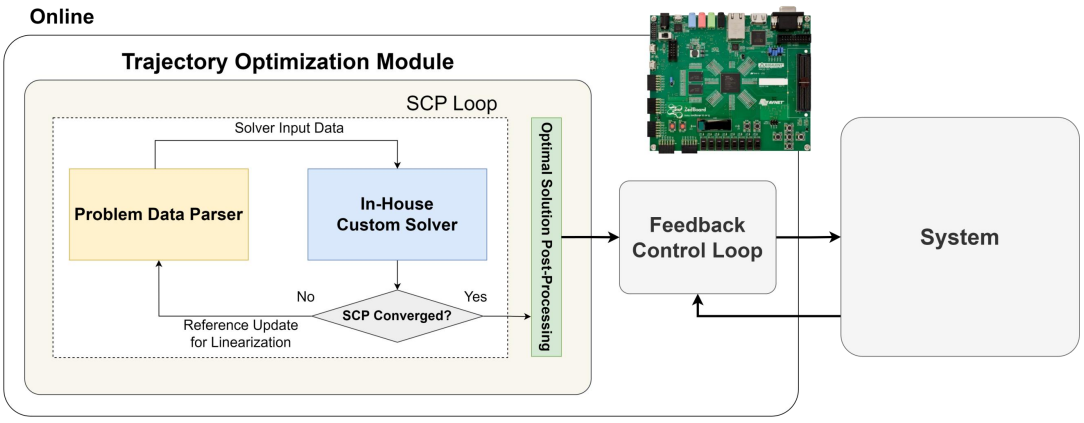


Fig. 4 Trajectory Optimization in Online Operation

solver [12]. The solver is based on a primal-dual IPM algorithm, which has fast and robust performance for small to medium scale problems. Unlike most existing solvers that cannot directly detect infeasibility in the presence of quadratic objective functions, the in-house solver employs a homogeneous embedding formulation that enables infeasibility detection without reformulating the problem. This infeasibility detection capability ensures that the algorithm converges within a finite number of iterations even for infeasible problem instances and explicitly provides a certificate of infeasibility, facilitating the handling of abnormal situation.

In addition to its efficient algorithmic structure, the custom solver framework also offers advantages for operation in such resource-constrained environments. The custom solver exploits the fixed problem structure by performing an offline analysis of the sparsity pattern of the problem data (i.e., the positions of nonzero elements in the problem data matrices) and uses this information to remove redundant operations at runtime, thereby reducing computational overhead. Furthermore, unlike general-purpose solvers that require dynamic allocation to accommodate arbitrary problem structures, custom solver can statically allocate all data in memory. This feature simplifies the verification and validation process of flight software and makes the solver highly suitable for onboard applications.

To facilitate customization and data handling for the use of the solver, an in-house developed solver customization tool is utilized. Implemented in MATLAB, the tool analyzes the problem structure specified by the user and automatically generates customized solver code in C. It also provides parsing information to allow users to easily handle the problem input data. Since sparse matrices are stored in the compressed column storage (CCS) format in the C implementation, whose indexing scheme for accessing each element is not intuitive, the provided parsing information facilitates the use of solver and improves productivity of users in solver use. The overall customization process is illustrated in Fig. 3.

Based on the custom solver and the provided parsing information, a trajectory optimization module for onboard execution is implemented in this study to perform trajectory optimization online, as illustrated in Fig. 4.

4 Numerical Experiment

To validate the feasibility of the proposed formulation and implementation, numerical experiments are performed on an embedded computing platform. In this study, AMD Zynq-7000 system on chip featuring an ARM Cortex-A9 architecture running at a maximum clock frequency of 650MHz and 512MB of RAM is used. The solver and the implemented SCP algorithm are written in C and complied with the `-O3` optimization level to maximize execution efficiency. The parameters for the SCP subproblem $\mathcal{P}2$ are listed in Table 2.

Monte Carlo simulations with 1000 samples are conducted by perturbing the nominal initial conditions in Table 3 with dispersions sampled from uniform distributions defined over the corresponding intervals. The number of discretization nodes were set to $N = 50$, and the maximum number of SCP iteration is limited to 10. A convergence tolerance of 10^{-4} is used to determine the SCP convergence.

Parameters	Values	Unit	Parameters	Values	Unit
Min. Throttle	52.0	%	α_{max}	5.0	deg
Max. Throttle	100.0	%	$\dot{\alpha}_{max}$	2.5	deg/s
$\dot{T}_{max}$	10.0	%/s	Y_{td}	20.0	m

Table 2 SCP Parameters

Parameters	Values	Unit	Parameters	Values	Unit
Nominal X_0	-800.0	m	Dispersion Interval of X_0	[-10.0, 10.0]	m
Nominal Y_0	2500.0	m	Dispersion Interval of Y_0	[-10.0, 10.0]	m
Nominal V_0	140.0	m/s	Dispersion Interval of V_0	[-5.0, 5.0]	m/s
Nominal γ_0	-50.0	deg	Dispersion Interval of γ_0	[-5.0, 5.0]	deg
			Dispersion Interval of m_0	[-1.5, 1.5]	%

Table 3 Numerical Experiment Conditions

The Monte Carlo results show that the proposed algorithm successfully converged to an optimal solution for 97.1% of the samples. For the remaining 2.9% of cases, the SCP did not converge. As shown in Fig. 5, the failed cases are characterized by more demanding initial conditions, including higher initial speeds, larger flight path angles in the positive direction, and, on average, approximately 50kg lower mass, compared to the nominal case. These conditions require larger maneuvering accelerations to satisfy the terminal boundary condition, which may render the problem infeasible under the given path constraints and limited control authority.

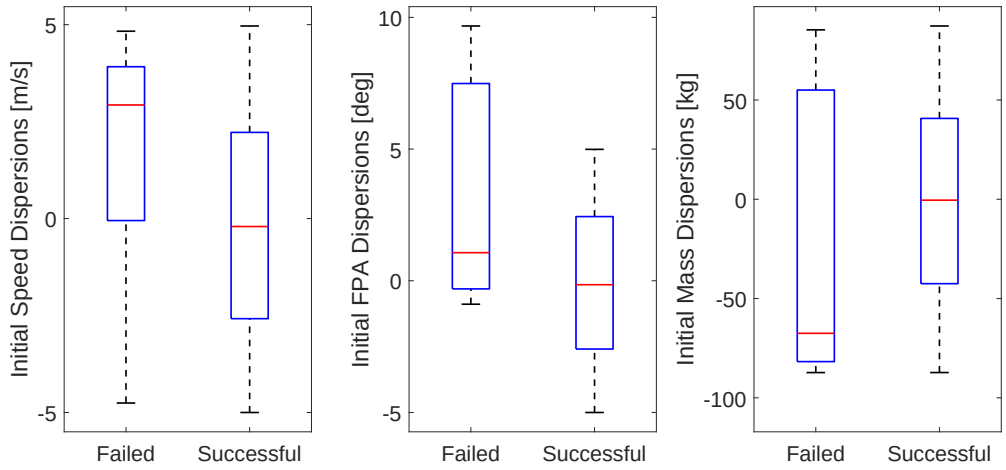
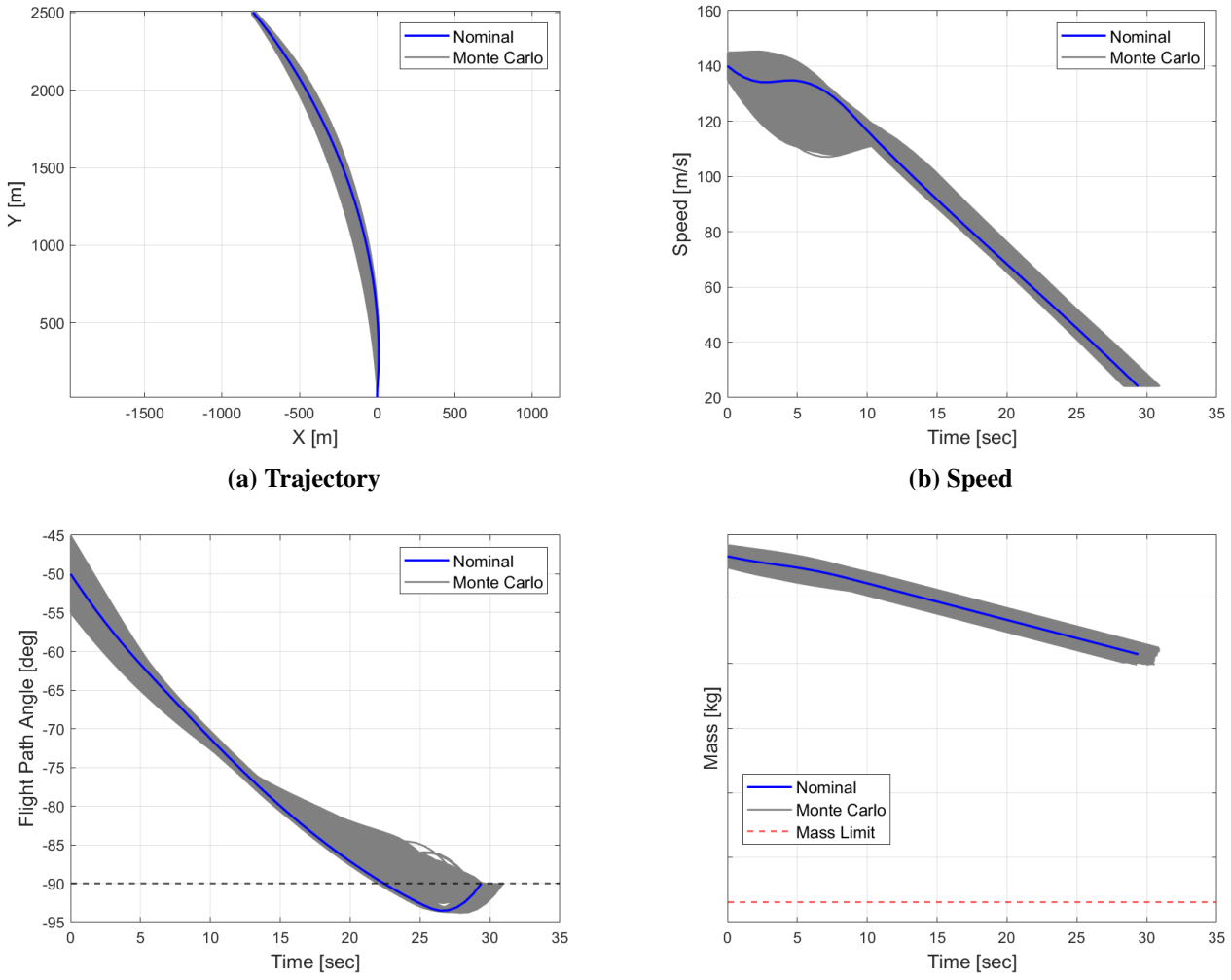


Fig. 5 Comparison of Initial Condition Dispersion for Successful and Failed Cases

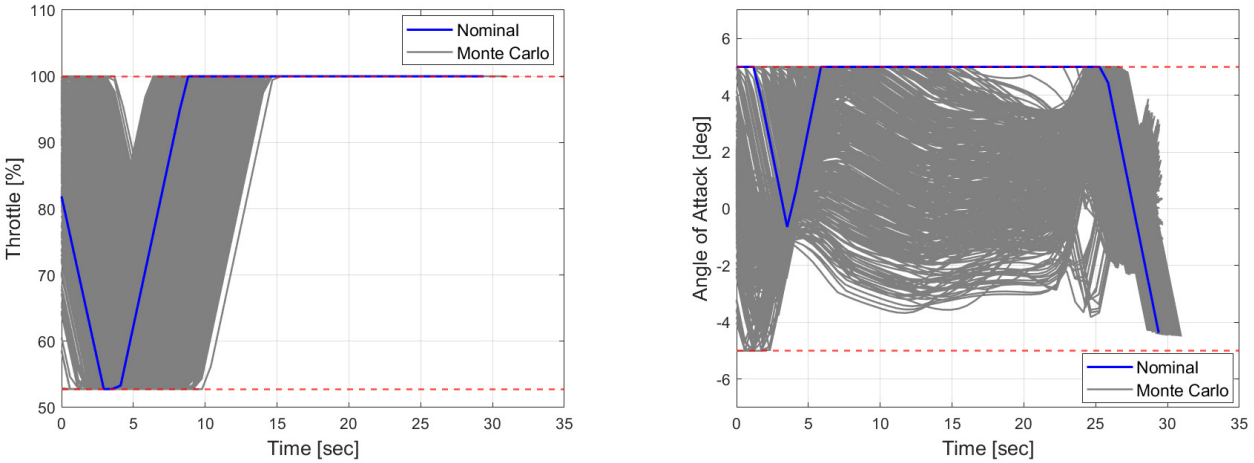


(a) Trajectory **(b) Speed**

(c) Flight Path Angle **(d) Mass**

Fig. 6 Monte-Carlo Results (States)

Figure 6 shows the vehicle states in the optimal landing trajectory from the Monte Carlo simulations. The blue line indicates the trajectory corresponding to the nominal initial condition, whereas the gray lines represent trajectories resulting from uniformly sampled dispersed initial conditions. Figures 6a and 6b show the resulting optimal trajectory and speed profiles. It can be observed that the remaining dispersions are effectively reduced during the landing-flight phase, and guiding the vehicle toward the



(a) Throttle **(b) Angle of Attack**

Fig. 7 Monte-Carlo Results (Control Inputs)

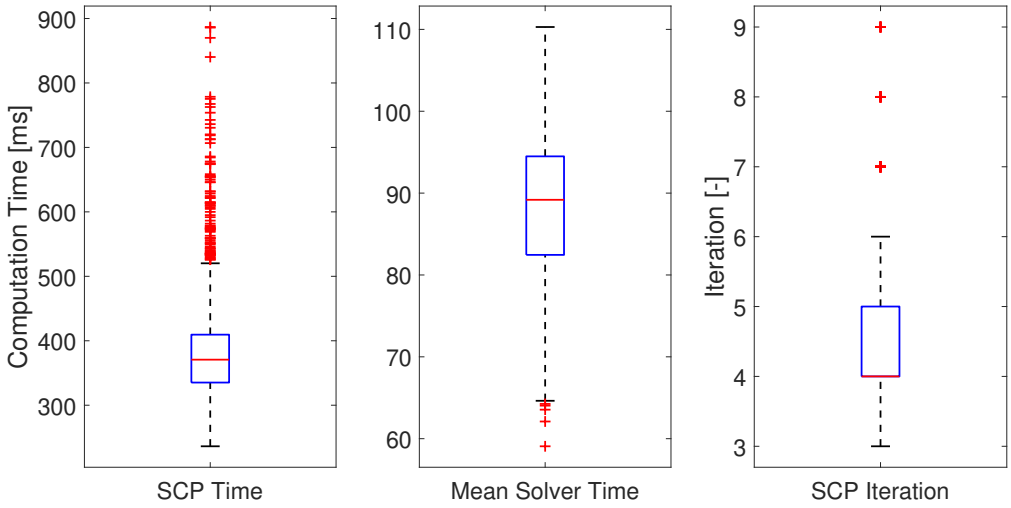


Fig. 8 SCP Performance Statistics in a Onboard Computer

desired terminal state as Eq. 11. This allows the remaining touchdown phase to be constructed such that the vehicle can be smoothly landed on the landing site using an analytical guidance method or a similar approach. Figure 6c shows the flight path angle profile. The results indicate that the vehicle can complete the remaining touchdown phase with minimal attitude maneuvering by setting the terminal flight path angle to be vertical to the ground. Finally, Fig. 6d shows the mass consumption results, confirming that the mass consumption during the landing-flight phase does not exceed the available propellant.

Figure 7 shows the control input commands corresponding to the optimal trajectory. It can be observed that the control inputs for thrust and angle of attack satisfy the prescribed constraints and that the thrust profile exhibits the well-known min–max shape characteristic of fuel-minimization problems. Moreover, in most cases, the resulting angle of attack profile tends to utilize nonzero values in the initial high dynamic pressure region, where aerodynamic effects are significant. This is because exploiting the induced drag generated by nonzero angles of attack as shown in Fig. 2b helps decelerate the vehicle and thereby reduces propellant consumption.

Figure 8 shows the statistics of the performance of the implemented SCP algorithm on the onboard computer. The results show that the SCP algorithm converged in an average of 4 iterations. The optimal

trajectory is generated within 750 ms under the 3-sigma criterion, even on a processor running at a limited clock speed of 650MHz.

5 Conclusion

In conclusion, this paper presents a trajectory optimization-based powered descent guidance for rocket landing considering aerodynamic drag and lift. The proposed problem formulation accounts for coupled drag and lift force parameterized by angle of attack, which is essential for rocket landing on Earth where aerodynamic effects cannot be ignored. Considering that thrust deflection angles are typically small in rockets, the problem is formulated such that the thrust direction is constrained by the vehicle attitude, thereby preserving additional control authority for disturbance rejection in practical applications. Furthermore, to prevent the singularity that arises in the flight path angle dynamics as the terminal velocity approaches zero, this study employs a two phase structure consisting of a landing-flight phase and a touchdown phase. This structure not only avoids the singularity in the dynamics but also provides additional degrees of freedom to design the touchdown phase with a desired acceleration profile, thereby enabling a safer landing. The resulting landing problem is solved using a SCP approach to enable efficient real-time computation, and the SCP algorithm is implemented with the in-house custom convex optimization solver for efficient onboard operation. Finally, numerical experiments are conducted on the onboard computing platform to verify the practical applicability of the algorithm. The results demonstrate successful convergence for the vast majority of Monte Carlo samples and show that the computation times remain within a practically acceptable range, even in a resource-constrained computing environment.

6 Acknowledgement

This study is a result of “Development of on-board guidance&control techniques and performance verification research for powered descent of reusable launch launchers” (RS-2022-00164702), which is hosted by KARI with the funding of KASA.

Declaration of Use of Artificial Intelligence

The authors acknowledge the use of ChatGPT 5 to improve the syntax and grammar of several paragraphs in the manuscript. The manuscript was originally drafted by the authors, with the AI system employed to enhance clarity and readability without modifying the technical content. No AI tool was used in the research work, analysis, or generation of results.

References

[1] Lars Blackmore. Autonomous precision landing of space rockets. In *Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2016 Symposium*, volume 46, pages 15–20. The Bridge Washington, DC, USA.

[2] Allan R. Klumpp. Apollo lunar descent guidance. *Automatica*, 10(2):133–146, 1974. doi: https://doi.org/10.1016/0005-1098(74)90019-3.

[3] John M. Carson, Behçet Acikmeşe, Lars Blackmore, and Aron A. Wolf. Capabilities of convex powered-descent guidance algorithms for pinpoint and precision landing. In *2011 Aerospace Conference*, pages 1–8, 2011. doi: 10.1109/AERO.2011.5747244.

[4] Danylo Malyuta, Taylor P. Reynolds, Michael Szmuk, Thomas Lew, Riccardo Bonalli, Marco Pavone, and Behçet Açıkmeşe. Convex optimization for trajectory generation: A tutorial on generating dynami-

cally feasible trajectories reliably and efficiently. *IEEE Control Systems Magazine*, 42(5):40–113, 2022. doi: 10.1109/MCS.2022.3187542.

[5] Behcet Acikmese and Scott R. Ploen. Convex programming approach to powered descent guidance for mars landing. *Journal of Guidance, Control, and Dynamics*, 30(5):1353–1366, 2007. doi: 10.2514/1.27553.

[6] Michael Szmuk, Behcet Acikmese, and Andrew W Berning. Successive convexification for fuel-optimal powered landing with aerodynamic drag and non-convex constraints. In *AIAA Guidance, Navigation, and Control Conference*, page 0378, 2016. doi: 10.2514/6.2016-0378.

[7] Michael Szmuk, Taylor P. Reynolds, and Behçet Açıkmeşe. Successive convexification for real-time six-degree-of-freedom powered descent guidance with state-triggered constraints. *Journal of Guidance, Control, and Dynamics*, 43(8):1399–1413, 2020. doi: 10.2514/1.G004549.

[8] Marco Sagliano, David Seelbinder, Stephan Theil, and Ping Lu. Six-degree-of-freedom rocket landing optimization via augmented convex–concave decomposition. *Journal of Guidance, Control, and Dynamics*, 47(1):20–35, 2024. doi: 10.2514/1.G007570.

[9] Xinfu Liu. Fuel-optimal rocket landing with aerodynamic controls. *Journal of Guidance, Control, and Dynamics*, 42(1):65–77, 2019. doi: 10.2514/1.G003537.

[10] Yurii Nesterov and Arkadii Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics, 1994. doi: 10.1137/1.9781611970791.

[11] Daniel Dueri, Behçet Açıkmeşe, Daniel P. Scharf, and Matthew W. Harris. Customized real-time interior-point methods for onboard powered-descent guidance. *Journal of Guidance, Control, and Dynamics*, 40(2):197–212, 2017. doi: 10.2514/1.G001480.

[12] Jae-Il Jang and Chang-Hun Lee. Customized interior-point methods solver for embedded real-time convex optimization, 2025. arXiv:2505.14973 [math.OC]. `https://arxiv.org/abs/2505.14973`.

[13] Ping Lu and Ryan Callan. Propellant-optimal powered descent guidance revisited. *Journal of Guidance, Control, and Dynamics*, 46(2):215–230, 2023. doi: 10.2514/1.G007214.