

OPTIMAL GUIDANCE ALGORITHM FOR BOOST-BACK OF REUSABLE LAUNCH VEHICLES

Jaewook Shim[1], Jae-Il Jang[1] & Chang-Hun Lee[1]

[1]Korea Advanced Institute of Science and Technology, Daejeon 34141, Republic of Korea

Abstract

This paper introduces a fuel-optimal boost-back guidance algorithm for reusable launch vehicles (RLV). The guidance problem is formulated as an optimal control problem with the predicted impact point as a terminal constraint. It is demonstrated that the optimal attitude during boost-back is constant under simplifying assumptions, and a simplified static optimization problem is proposed whose decision variables are remaining burn time and attitude. Based on the similarity of guidance problems between consecutive guidance cycles, a single-step correction algorithm is used to issue the guidance command. The proposed method shows fuel-optimal performance in both Downrange Landing (DRL) and Return-to-Launch-Site (RTLS) missions. Monte Carlo analysis is performed to verify the robustness of the proposed algorithm under model uncertainties.

Keywords: Boost-back Guidance, Reusable Launch Vehicle, Fuel-optimal Guidance, Parameter Optimization

1. Introduction

Since the success of SpaceX's Falcon 9, the global space launch landscape has been transitioning toward the development and operation of reusable launch vehicles (RLVs). A major challenge in this shift is the safe recovery of the vehicle. Following upper-stage separation, the booster undergoes a sequence of recovery flight phases until the soft pin-point touchdown at the landing site. These consist of the boost-back burn, ballistic free-fall, reentry burn, aerodynamic flight, and finally the landing burn. The booster's trajectory needs to be adjusted by a boost-back burn to ensure the vehicle flies toward the landing site. The boost-back burn plays an important role as it resets the dispersion accumulated over the aerodynamic ascent phase and provides a new initial condition for the remaining phases. This phase is particularly pivotal in Return-to-Launch-Site (RTLS) missions as it requires a large change in the velocity vector. To maximize payload capacity and secure sufficient margins for subsequent phases, it is crucial to minimize fuel consumption during the boost-back phase.

Various guidance laws have been proposed and studied to address the boost-back guidance problem. For instance, Jo et al. derived a non-iterative closed-form acceleration command using Instantaneous Impact Point (IIP) dynamics and its derivatives, demonstrating near-optimal performance for Downrange Landing (DRL) missions [1]. However, this command suffers from significant performance degradation in RTLS scenarios. To mitigate this, flight path angle rate control logic was applied, though it requires a predetermined optimal rate [2]. Other approaches include solving the Lambert problem iteratively to calculate the minimum velocity increment direction as a guidance command [3], or employing Iterative Guidance Mode (IGM) to satisfy orbit insertion conditions at boost-back termination, utilizing geometry and time updaters to account for landing site position change in the inertial reference frame [4].

https://doi.org/10.71945/icas2026_0505

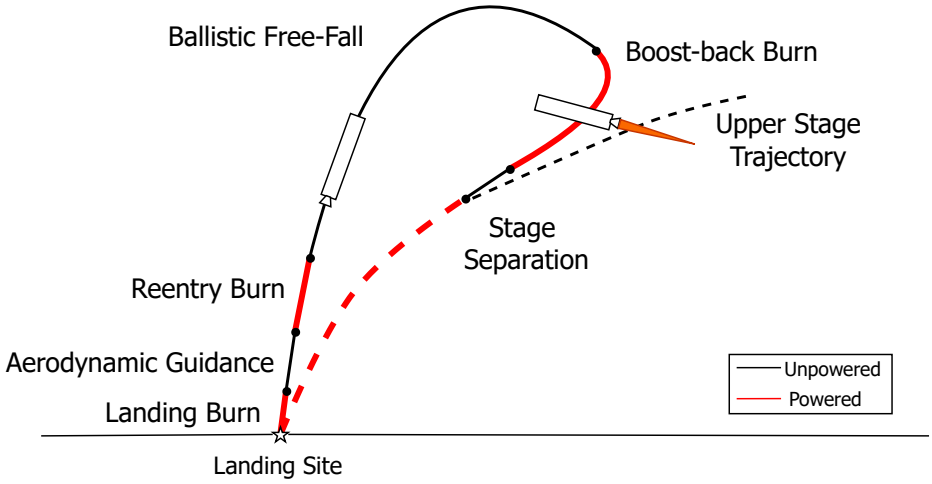


Figure 1 – Recovery flight phases of RLVs.

Despite these previous studies, only a few works have analyzed the fundamental characteristics of optimal guidance laws for the boost-back phase or utilized such analysis as the cornerstone of guidance algorithm design. In this research, the optimal control problem for boost-back burn guidance is mathematically formulated. We demonstrate that the optimal attitude during the boost-back burn remains constant under mild assumptions, and we identify the optimal attitude angles for various desired landing locations. Leveraging the solution to this optimal control problem, the original problem is converted into a simplified parameter-finding problem. This paper proposes a single-step correction algorithm integrating concepts of sensitivity dynamics and null-space optimization. The proposed algorithm is verified through numerical simulations of a realistic mission on a 3D Earth model. Comparative analysis with the optimal solution reveals that the algorithm achieves optimal performance in both DRL and RTLS cases without requiring any structural modifications. Furthermore, the algorithm requires only a single prediction and correction per guidance step, facilitating its application in real-time guidance command generation.

The rest of this paper is organized as follows. Section 2. describes the vehicle dynamics and formulates the boost-back guidance as a fuel-optimal control problem. In Section 3., we propose a detailed procedure to calculate the guidance command based on the single-step correction method. The proposed algorithm is tested and analyzed by numerical simulation in Section 4.. Finally, the paper is concluded in Section 5..

2. Problem Formulation

2.1 Dynamics Model

In this study, the motion of the RLV is described by a 3-DOF point-mass model as follows:

$$\begin{aligned} \dot{\boldsymbol{r}} &= \boldsymbol{v} \\ \dot{\boldsymbol{v}} &= \frac{T}{m}\boldsymbol{u} - \frac{\mu}{\|\boldsymbol{r}\|^3}\boldsymbol{r} \\ \dot{m} &= -\frac{n_{eng}T_{vac}}{v_{ex}} \end{aligned} \tag{1}$$

where $\boldsymbol{r} \in \mathbb{R}^3$ is the position vector relative to the Earth's center in the inertial frame, $\boldsymbol{v} \in \mathbb{R}^3$ is the velocity vector, and m is the mass of the vehicle. The control input $\boldsymbol{u} \in \mathbb{R}^3$ is a unit vector that indicates the direction of the thrust force. μ denotes the gravitational parameter of the Earth. The thrust force is obtained as (2).

$$T = \begin{cases} n_{eng}(T_{vac} - p(r)A_e) & \text{(powered phase)} \\ 0 & \text{(otherwise)} \end{cases} \tag{2}$$

The parameters n_{eng}, T_{vac}, and v_{ex} represent the number of engines, throttleable vacuum thrust magnitude, and exhaust velocity, respectively. The ambient pressure $p(r)$ can be neglected during the boost-back. For notational brevity, the vehicle state vector is defined as $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{r}^T & \boldsymbol{v}^T & m \end{bmatrix}^T$, and the dynamics (1) are expressed as $\dot{\boldsymbol{x}} = \boldsymbol{f}_{dyn}(\boldsymbol{x}, \boldsymbol{u})$. The variables t_0 and t_f^{BB} denote the current time and the boost-back engine cut-off time. The boost-back burn is performed after upper-stage separation at an altitude higher than 50 km, and the aerodynamic forces are negligible during this phase.

2.2 Fuel-Optimal Guidance Problem Formulation

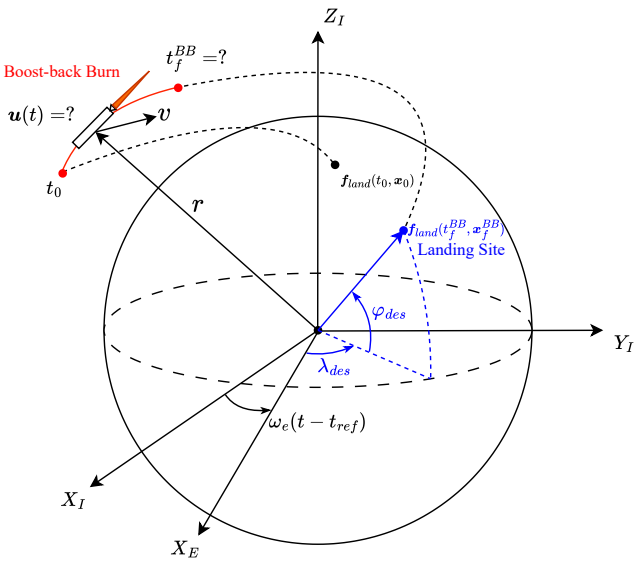


Figure 2 – Concept of the boost-back burn.

In this section, the fuel-optimal boost-back guidance problem is formulated as a form of optimal control problem. The concept of the boost-back burn is visualized in Fig. 2. Initially, the predicted landing position is denoted as $\boldsymbol{f}_{land}(t_0, \boldsymbol{x}_0)$. The guidance goal is to satisfy predicted landing site constraints while minimizing fuel consumption during the boost-back by applying proper thrust direction $\boldsymbol{u}(t)$ and thrust magnitude $T(t)$. The fuel-optimal boost-back guidance problem is formulated as Problem P_{BB}:

Problem P_{BB}

$$\begin{aligned}
\underset{T,\boldsymbol{u},t_f^{BB}}{\text{minimize}} \quad & J = -\int_{t_0}^{t_f^{BB}} \dot{m}\, dt = \int_{t_0}^{t_f^{BB}} \frac{T}{v_{ex}}\, dt \\
\text{subject to} \quad & \dot{\boldsymbol{r}} = \boldsymbol{v} \\
& \dot{\boldsymbol{v}} = \frac{T}{m}\boldsymbol{u} - \frac{\mu}{\|\boldsymbol{r}\|^3}\boldsymbol{r} \\
& \dot{m} = -\frac{T}{v_{ex}} \\
& \boldsymbol{u}^T\boldsymbol{u} = 1 \\
& 0 \le T \le T_{\max} \\
& \boldsymbol{e}_{land}(t_f^{BB}, \boldsymbol{x}(t_f^{BB})) = 0
\end{aligned} \tag{3}$$

The dynamics during the boost-back burn neglect the aerodynamic forces and ambient pressure from (1), and the maximum engine thrust is $T_{max} = n_{eng}^{BB} T_{vac,max}$. The boost-back termination time is expressed by the symbol t_f^{BB}, which is a free variable. The guidance goal is described by $\boldsymbol{e}_{land}$, which is an explicit function of the final time and final state, whose definition is as follows:

$$\begin{aligned}
\boldsymbol{f}_{land}(t,\boldsymbol{x}) &= [\varphi_p, \lambda_p]^T \in \mathbb{R}^2 \\
\boldsymbol{e}_{land}(t,\boldsymbol{x}) &= \boldsymbol{f}_{land}(t,\boldsymbol{x}) - [\varphi_{des}, \lambda_{des}]^T
\end{aligned} \tag{4}$$

A function $\boldsymbol{f}_{land}$ is any landing site predictor whose output consists of the latitude and longitude of the predicted landing location in the ECEF (Earth-Centered Earth-Fixed) frame, and the desired landing site location is described by $[\varphi_{des}, \lambda_{des}]^T$. $\boldsymbol{e}_{land}$ is the error between predicted and desired landing location to be nullified at the boost-back burnout.

A typical procedure to solve the problem is discussed. The Hamiltonian of P_{BB} is obtained as follows:

$$\begin{aligned}
H &= \frac{T}{v_{ex}} + \boldsymbol{\lambda}_r^T\boldsymbol{v} + \boldsymbol{\lambda}_v^T\left(\frac{T}{m}\boldsymbol{u} - \frac{\mu}{\|\boldsymbol{r}\|^3}\boldsymbol{r}\right) + \lambda_m\left(-\frac{T}{v_{ex}}\right) + \mu_u(1-\boldsymbol{u}^T\boldsymbol{u}) \\
&= \boldsymbol{\lambda}_r^T\boldsymbol{v} - \boldsymbol{\lambda}_v^T\frac{\mu}{\|\boldsymbol{r}\|^3}\boldsymbol{r} + T\left(\frac{1-\lambda_m}{v_{ex}} + \frac{\boldsymbol{\lambda}_v^T\boldsymbol{u}}{m}\right) + \mu_u(1-\boldsymbol{u}^T\boldsymbol{u})
\end{aligned} \tag{5}$$

where $\boldsymbol{\lambda} = \begin{bmatrix}\boldsymbol{\lambda}_r^T & \boldsymbol{\lambda}_v^T & \lambda_m\end{bmatrix}^T$ are costates corresponding to position, velocity, and mass. μ_u is the Lagrange multiplier for the path equality constraint. Applying Pontryagin's minimum principle, the optimal thrust direction is parallel to the velocity costate vector, and the optimal thrust magnitude is determined by the sign of the switching function as follows:

$$\boldsymbol{u}^* = -\frac{\boldsymbol{\lambda}_v}{\|\boldsymbol{\lambda}_v\|}, \quad T^* = \begin{cases} T_{\max} & (S<0) \\ 0 & (S>0) \end{cases} \qquad \text{where} \quad S = \frac{1-\lambda_m}{v_{ex}} + \frac{\boldsymbol{\lambda}_v^T\boldsymbol{u}}{m} \tag{6}$$

The costate equations are derived by applying the necessary conditions:

$$\dot{\boldsymbol{\lambda}}_r = -\frac{\mu}{\|\boldsymbol{r}\|^5}\left(3\boldsymbol{r}\boldsymbol{r}^T - \|\boldsymbol{r}\|^2 I_{3\times3}\right)\boldsymbol{\lambda}_v, \qquad \dot{\boldsymbol{\lambda}}_v = -\boldsymbol{\lambda}_r, \qquad \dot{\lambda}_m = \frac{T}{m^2}\boldsymbol{\lambda}_v^T\boldsymbol{u} \tag{7}$$

The transversality conditions are as follows:

$$\boldsymbol{\lambda}(t_f^{BB}) = \left(\frac{\partial \boldsymbol{e}_{land}}{\partial \boldsymbol{x}(t_f^{BB})}\right)^T \boldsymbol{\nu}, \qquad H(t_f^{BB}) = -\left(\frac{\partial \boldsymbol{e}_{land}}{\partial t_f^{BB}}\right)^T \boldsymbol{\nu} \tag{8}$$

Therefore, it is concluded that the boost-back guidance problem reduces to a classical fuel-optimal

rocket steering problem [5] with special transversality constraints described by $\boldsymbol{e}_{land}$. The problem P_{BB} can be solved via shooting method with 10 unknowns $(\boldsymbol{\lambda}(t_0), \boldsymbol{\nu}, t_f^{BB})$ with 10 constraints (transversality conditions (8) and $\boldsymbol{e}_{land} = 0$). Although it is possible to solve this root-finding problem, providing initial guesses for 10 unknowns is not straightforward as these variables do not have explicit physical meaning. Also, providing wrong initial guesses may prevent the problem from converging. Hence, the alternative approach to approximately solve the above problem will be proposed in Section 3..

2.2.1 Prediction of Landing Position

Before introducing the detailed algorithm, the terminal condition of the boost-back burn should be discussed thoroughly in this subsection. The function $\boldsymbol{f}_{land} : \mathbb{R} \times \mathbb{R}^7 \to \mathbb{R}^2$ is any appropriate mapping from the current absolute time relative to the epoch and the vehicle states to the expected ECEF landing latitude and longitude. This mapping is indeed the prediction of the subsequent flight phases such as ballistic free-fall, reentry burn, aerodynamic guidance, and landing burn. The most straightforward way to evaluate $\boldsymbol{f}_{land}$ is to know all control actions in the subsequent phases in advance and to integrate the trajectory until the altitude reaches zero. However, it is difficult to know in advance which control will be applied in the future phases as it depends on the current action. If no future control action is assumed and aerodynamic forces are neglected, the ground impact point after the boost-back burnout can be obtained by Keplerian IIP. This method models the subsequent flight phases as a ballistic free-fall trajectory, neglecting both thrust and aerodynamic forces. It represents one of the simplest methods for predicting the terminal impact point, inherently decoupling the prediction from future control dependencies. Based on the methodology presented in [6], the IIP in the ECEF frame can be determined without requiring iterative numerical procedures. The entire computation is structured as an analytic closed-form formulation, enabling the direct and analytic evaluation of both the landing point and its sensitivities. In the remainder of the paper, the landing position predictor $\boldsymbol{f}_{land}$ by Keplerian IIP is denoted as $\boldsymbol{f}_{kep}(t, \boldsymbol{x})$.

2.2.2 Guidance Parameters for Boost-back Burn

In this subsection, the original boost-back burn problem P_{BB} is simplified by proposing guidance parameters for the boost-back burn. During the recovery of the booster, the following assumptions are valid, since the temporal and spatial ranges of the booster motion are limited.

- **Assumption 1.** The Earth's gravitational field is uniform and constant.
- **Assumption 2.** The Earth's rotational effect is negligible.
- **Assumption 3.** Subsequent flight phases are ignored and assumed to be a gravity free-fall.

Applying the assumptions to P_{BB} with $\boldsymbol{f}_{kep}$ as a terminal constraint is equivalent to imposing terminal position constraint at ground impact $(\boldsymbol{r}(t_f) = \boldsymbol{r}_{des})$. By extending the time horizon up to the ground impact (t_f), the solution is obtained as follows:

$$\boldsymbol{u}^* \text{ is constant and } T^* = \begin{cases} T_{\max} & t \in [t_0, t_f^{BB*}] \\ 0 & t \in [t_f^{BB*}, t_f] \end{cases} \quad \text{where} \quad S(t_f^{BB*}) = 0 \tag{9}$$

The above solution follows from the simplification of the costate equations (7) under a vanishing gravity gradient, together with the transversality condition $\boldsymbol{\lambda}_v(t_f) = \boldsymbol{0}$ implied by the position-only terminal constraint. The optimal thrust direction $\boldsymbol{u}^*$ and burn time t_f^{BB*} could be obtained by solving the boundary value problem defined by (6), (7), and (8). However, to do so, a separate outer-loop to compensate for the Earth's rotation is required and the optimality of the solution degrades as the free-fall ballistic trajectory also relies on the assumptions. Hence, in this research, only the optimal form (9) of the problem with assumptions is taken, and an alternative approach is proposed. Inspired by (9), guidance parameters to be found during the boost-back burn phase at the k^{th} time step $(t = t_k)$

are defined as follows.

$$\boldsymbol{p}_k \equiv \begin{bmatrix} t_{f,k}^{BB} & \theta_k^{BB} & \psi_k^{BB} \end{bmatrix}^T \tag{10}$$

$t_{f,k}^{BB}$ is the remaining time to burnout, θ_k^{BB} and ψ_k^{BB} are pitch and yaw angles with respect to a convenient inertial guidance frame to be maintained until burnout. The unit thrust direction in the guidance frame is expressed as follows.

$$\boldsymbol{u}^G = \begin{bmatrix} \cos\psi\cos\theta & \sin\psi & -\cos\psi\sin\theta \end{bmatrix}^T \tag{11}$$

Figures 3 and 4 illustrate the definitions of the guidance frame and the attitude angles. The X_G-axis is aligned with the launch direction, while the Z_G-axis is normal to the Earth's surface. The origin is fixed at the inertial position of the launch site at the instant of launch.

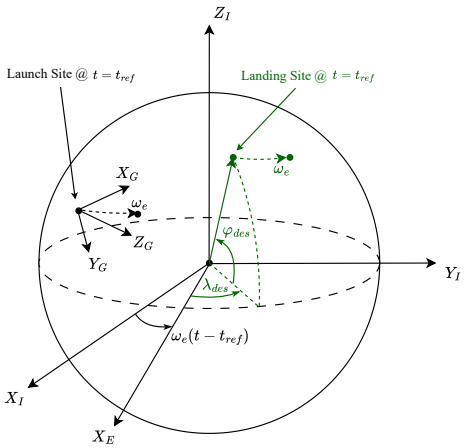


Figure 3 – Definition of guidance frame.

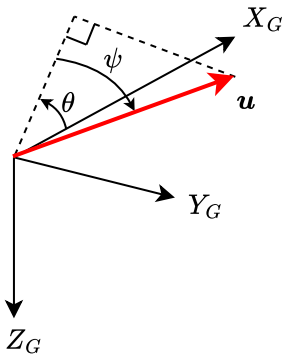


Figure 4 – Definition of pitch and yaw angles.

3. Proposed Guidance Algorithm

3.1 Single-step Correction Parameter Optimization

The problem to be solved at each guidance step k is described by P_k:

Problem P_k

$$\begin{aligned} &\underset{\boldsymbol{p}_k}{\text{minimize}} \quad J_k = t_{f,k}^{BB} \\ &\text{subject to} \quad \boldsymbol{g}_k(\boldsymbol{p}_k) \equiv \boldsymbol{e}_{land}\left(t_k + t_{f,k}^{BB}, \quad \boldsymbol{x}_k + \int_{t_k}^{t_k + t_{f,k}^{BB}} \boldsymbol{f}_{dyn}(\boldsymbol{x}, \boldsymbol{u}_k)dt\right) = 0 \\ &\qquad\qquad \text{where } \boldsymbol{u}_k \text{ is defined as in (11)} \end{aligned} \tag{12}$$

To maximize the closed-loop feedback performance, the optimization problem P_k is iteratively solved at each guidance step $t = t_k$ using the most recent state information $\boldsymbol{x}_k$. Although P_k is a Nonlinear Programming (NLP) problem with only three parameters—and thus possesses a relatively small dimension—evaluating the equality constraint $\boldsymbol{g}_k(\boldsymbol{p}_k)$ and its gradient $\nabla_{\boldsymbol{p}_k}\boldsymbol{g}_k$ inherently requires the numerical integration of the flight dynamics $\boldsymbol{f}_{dyn}$. Consequently, solving P_k to strict convergence at every step demands multiple evaluations of $\boldsymbol{g}_k$, which degrades real-time applicability. To resolve this computational bottleneck, a single-step correction algorithm inspired by the Real-Time Iteration

(RTI) scheme [7] is proposed. Let $\boldsymbol{p}_k^* = \begin{bmatrix} t_{f,k}^{BB*} & \theta_k^{BB*} & \psi_k^{BB*} \end{bmatrix}^T$ denote the optimal solution for P_k. At the subsequent guidance step $k+1$, which occurs after a time interval Δt_{gid}, the corresponding problem advances to P_{k+1}. Assuming the absence of model uncertainties and external disturbances, it can be readily demonstrated that $\hat{\boldsymbol{p}}_{k+1} = \begin{bmatrix} t_{f,k}^{BB*} - \Delta t_{gid} & \theta_k^{BB*} & \psi_k^{BB*} \end{bmatrix}^T$ constitutes a *feasible* solution for P_{k+1}. Furthermore, for a sufficiently small Δt_{gid}, it is reasonable to expect that the optimal solutions of consecutive steps remain close to each other; thus, $\theta_{k+1}^{BB*} \approx \theta_k^{BB*}$ and $\psi_{k+1}^{BB*} \approx \psi_k^{BB*}$. Consequently, $\hat{\boldsymbol{p}}_{k+1}$ serves as a highly accurate initial guess for P_{k+1}. Leveraging this proximity, a single correction of the guidance parameter $\boldsymbol{p}$ is sufficient to approximate the true optimal solution of P_{k+1}, thereby ensuring both near-optimality and deterministic real-time execution.
The correction step at $t = t_k$ is constructed from the following two terms:

$$\begin{aligned} \Delta \boldsymbol{p}_{k,err} &= -\boldsymbol{J}^+(\boldsymbol{p}_k)\boldsymbol{g}_k(\boldsymbol{p}_k) \\ \Delta \boldsymbol{p}_{k,opt} &= -(I_{3\times 3} - \boldsymbol{J}^+(\boldsymbol{p}_k)\boldsymbol{J}(\boldsymbol{p}_k))\nabla_{\boldsymbol{p}_k} J_k \end{aligned} \tag{13}$$

where $\boldsymbol{J} = \frac{\partial \boldsymbol{e}_{land}}{\partial \boldsymbol{p}} \in \mathbb{R}^{2\times 3}$. At $\boldsymbol{p} = \boldsymbol{p}_k$ the landing error constraint $\boldsymbol{g}_k(\boldsymbol{p})$ can be approximated to first order as:

$$\boldsymbol{g}_k(\boldsymbol{p}) \approx \boldsymbol{g}_k(\boldsymbol{p}_k) + \boldsymbol{J}(\boldsymbol{p}_k)(\boldsymbol{p} - \boldsymbol{p}_k) \tag{14}$$

Equating the linearized constraint in (14) to zero yields $\Delta \boldsymbol{p}_{k,err}$, which represents the minimum-norm solution. This functions as a conventional Newton-Raphson correction to drive the parameter vector $\boldsymbol{p}$ onto the constraint manifold. Because a degree of freedom remains within the null space of the Jacobian matrix $\boldsymbol{J}$, the objective function gradient $\nabla_{\boldsymbol{p}_k} J_k = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$ is projected onto this null space. This projection strategically exploits the redundant degree of freedom to simultaneously minimize the cost function.
Finally, the correction at the k^{th} step is a combination of the two terms in (13):

$$\boldsymbol{p}_{k+1} = \boldsymbol{p}_k + \Delta \boldsymbol{p}_k \quad \text{where} \quad \Delta \boldsymbol{p}_k = \alpha \Delta \boldsymbol{p}_{k,err} + \beta \Delta \boldsymbol{p}_{k,opt} \tag{15}$$

α and β are step sizes to be tuned appropriately considering the model uncertainties and Δt_{gid}. After the correction, the thrust attitude angles, θ_k and ψ_k, are issued as guidance commands.

3.2 Calculation of Terminal Error

The landing site prediction is achieved by Keplerian IIP ($\boldsymbol{f}_{kep}$) when the boost-back guidance does not need to consider subsequent phases in advance. It is advantageous that the evaluation of $\boldsymbol{f}_{kep}$ and $\frac{\partial \boldsymbol{f}_{kep}}{\partial t}, \frac{\partial \boldsymbol{f}_{kep}}{\partial \boldsymbol{x}}$ only involves a sequence of analytic closed-form formulas, thanks to the work in [6]. The evaluation of Keplerian IIP at the boost-back burnout requires $t_f^{BB}, \boldsymbol{x}(t_f^{BB})$. Hence, the dynamics $\boldsymbol{f}_{dyn}$ are integrated from t_0 to t_f^{BB} to obtain the terminal vehicle state $\boldsymbol{x}(t_f^{BB})$, from which the terminal constraint $\boldsymbol{e}_{land}$ is calculated subsequently. The sensitivity of the vehicle state with respect to the parameters θ^{BB} and ψ^{BB} is defined as follows:

$$S(t) \equiv \frac{\partial \boldsymbol{x}(t)}{\partial(\theta, \psi)} \tag{16}$$

It can be obtained by forward integration of the sensitivity dynamics of $\boldsymbol{f}_{dyn}$ [8]. The advantage of using the sensitivity dynamics is that it only requires a single forward integration and provides an accurate gradient without the truncation error of finite-difference approximations. During the boost-back burn phase, which requires real-time command generation, reducing the computation time for evaluating $\boldsymbol{e}$ and $\boldsymbol{J}$ is beneficial. The sensitivity dynamics for (1) are as follows:

$$\dot{S} = AS + B \tag{17}$$

where

$$A = \begin{bmatrix} 0_{3\times3} & I_3 & 0_{3\times1} \\ \frac{\mu}{\|\boldsymbol{r}\|^5}\left(3\boldsymbol{r}\boldsymbol{r}^T - \|\boldsymbol{r}\|^2 I_3\right) & 0_{3\times3} & -\frac{T_{\max}}{m^2}\boldsymbol{u}(\theta,\psi) \\ 0_{1\times3} & 0_{1\times3} & 0 \end{bmatrix}$$

$$B = \frac{\partial \boldsymbol{f}_{dyn}}{\partial(\theta,\psi)} = \begin{bmatrix} & 0_{3\times2} & \\ \frac{T_{\max}}{m} C_G^I & \begin{bmatrix} -\cos\psi\sin\theta & -\sin\psi\cos\theta \\ 0 & \cos\psi \\ -\cos\psi\cos\theta & \sin\psi\sin\theta \end{bmatrix} & \\ & 0_{1\times2} & \end{bmatrix}$$

The sensitivity with respect to the burn time is obtained as $\frac{\partial \boldsymbol{x}(t_f^{BB})}{\partial t_f^{BB}} = \boldsymbol{f}_{dyn}(\boldsymbol{x}(t_f^{BB}), \boldsymbol{u}(\boldsymbol{p}))$ from the Leibniz integral rule. Finally, from the chain rule,

$$\boldsymbol{J} = \frac{\partial \boldsymbol{e}_{land}}{\partial \boldsymbol{p}} = \left[\frac{\partial \boldsymbol{f}_{kep}}{\partial t_f^{BB}} + \frac{\partial \boldsymbol{f}_{kep}}{\partial \boldsymbol{x}(t_f^{BB})}\frac{\partial \boldsymbol{x}(t_f^{BB})}{\partial t_f^{BB}} \quad \frac{\partial \boldsymbol{f}_{kep}}{\partial \boldsymbol{x}(t_f^{BB})}\frac{\partial \boldsymbol{x}(t_f^{BB})}{\partial \theta^{BB}} \quad \frac{\partial \boldsymbol{f}_{kep}}{\partial \boldsymbol{x}(t_f^{BB})}\frac{\partial \boldsymbol{x}(t_f^{BB})}{\partial \psi^{BB}} \right] \tag{18}$$

The above procedure is summarized in the flowchart shown in Fig. 5. At the first call right before the burn starts (pre-thrust call), the correction is repeated until the correction amount becomes less than ε_{th}.

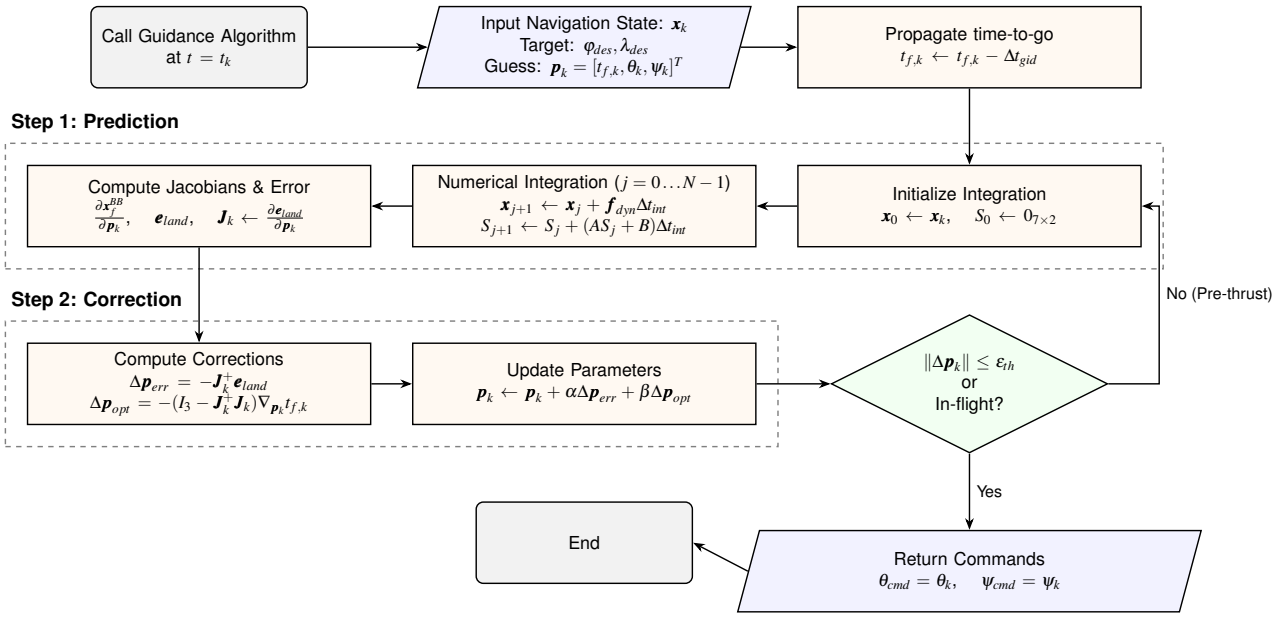


Figure 5 – Flowchart of the proposed fuel-optimal boost-back guidance algorithm.

4. Numerical Simulation

The proposed algorithm is verified through numerical simulations. The launch vehicle specification is based on the Falcon 9 vehicle as shown in Table 1. The initial conditions and the landing site are selected based on RTLS and DRL missions from [3] and are shown in Table 2. The desired landing sites for RTLS and DRL missions are shown in Fig. 6.

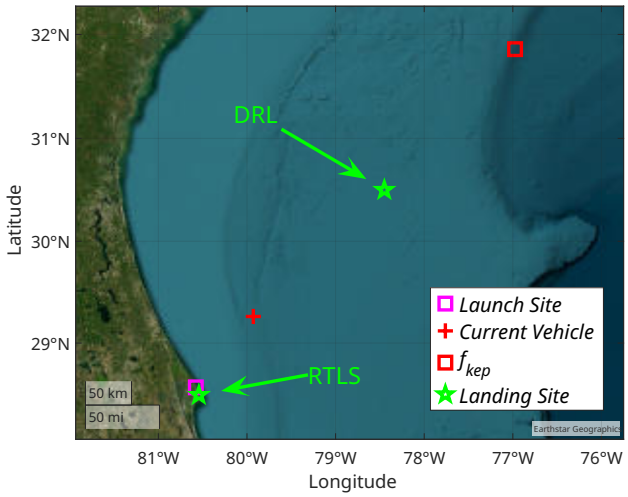


Figure 6 – Simulation scenarios. [3]

Table 1 – Launch vehicle specification

Specification	Value	Unit
Maximum thrust, $T_{\max}$	2736.9	kN
Engine exhaust velocity, v_{ex}	3049.98	m/s

Table 2 – Boost-back burn initial conditions [3]

Initial Condition	Value	Unit
Reference time, t_{ref}	0.0	sec
Time, t_0	160.0	sec
Position, $\boldsymbol{r}_0$	$[1051, -5541, 3160]^T$	km
Velocity, $\boldsymbol{v}_0$	$[1475.98, 1395]^T$	m/s
Mass, m_0	79200	kg

Table 3 shows the values of the algorithm parameters used for the simulation. The same parameters and initial guesses are used for both RTLS and DRL scenarios. The boost-back burn is terminated when the predicted landing position error falls below 50 m ($R_e\|\boldsymbol{e}_{land}\| \leq 50$ m, R_e: Earth's radius).

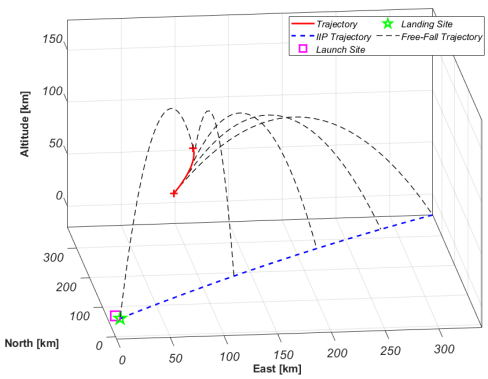
Table 3 – Algorithm setup

Parameters	Value	Unit
α	0.8	-
β	0.01	-
ε_{th}	1×10^{-6}	-
t_f^{BB} guess	30.0	sec
θ guess	180.0	deg
ψ guess	0.0	deg
Integration Method	Euler's Method, $\Delta t_{int} = 0.1s$	-
Guidance frequency	50	Hz

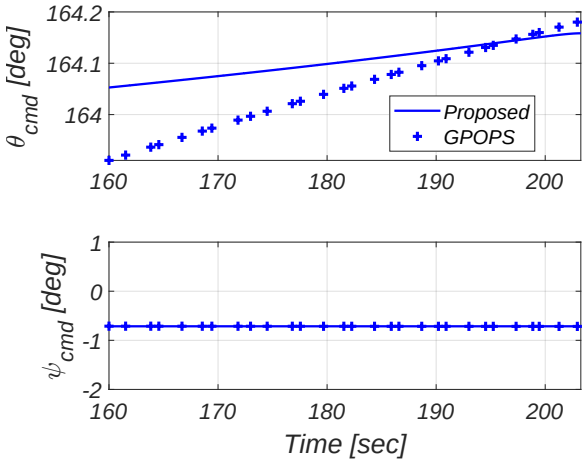
4.1 Simulation for RTLS and DRL Missions

Figure 7a illustrates the simulation results for the RTLS scenario. The red line denotes the boost-back trajectory, the black dashed line shows the instantaneous free-fall trajectories during the burn, and the blue dashed line is the ground track of the predicted landing site $\boldsymbol{f}_{kep}$. The proposed algorithm successfully achieved the guidance objective, generating a nearly constant attitude command as shown in Fig. 7b. The pitch angle command is maintained at 164.1 degrees, while the yaw angle command is about -0.71 degrees. As the yaw angle generates acceleration in the cross-track direction, it corrects the heading error. On the other hand, the pitch maneuver is the main control action to bring the initial predicted impact point to the desired landing site location in the downrange direction. The attitude command is nearly constant but slowly varying during the burn. This is because the solution of the problem P_k varies between guidance steps. The proposed single-step correction algorithm successfully maintained the guidance parameters close to the optimal solution throughout the burn. Also, the algorithm converged to the solution of P_0 during the pre-thrust calls despite the poor initial guess shown in Table 3. The velocity profile first decreases and then increases. For the RTLS mission, additional velocity should be gained before burnout in order to redirect the vehicle toward the launch site.

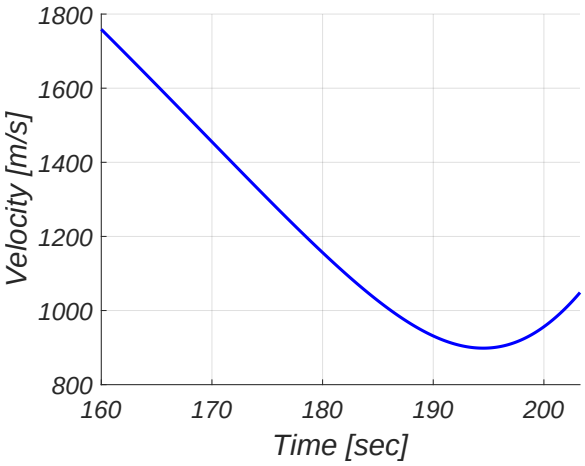
The boost-back burn is maintained for 43.265 seconds until $R_e\|\boldsymbol{e}_{land}\| \leq 50$m is satisfied. At the last guidance call before the termination, the remaining time-to-go $t_{f,N}^{BB}$ is 0.007 seconds. These two values add up to 43.272 seconds, which represents the expected total burn time for the error to become zero. This compensated burn time can be used as the nominal boost-back burn time of the proposed method.



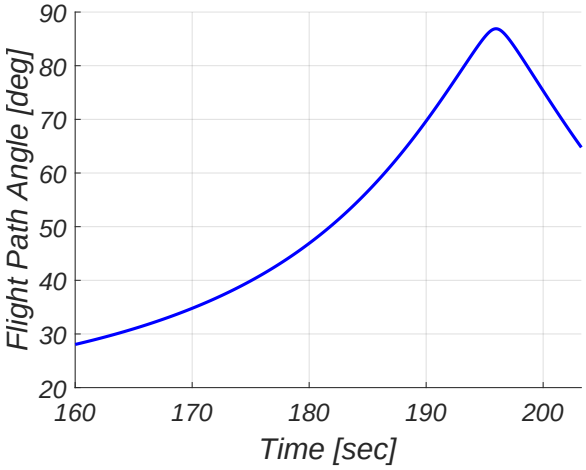
(a) Trajectory



(b) Attitude command



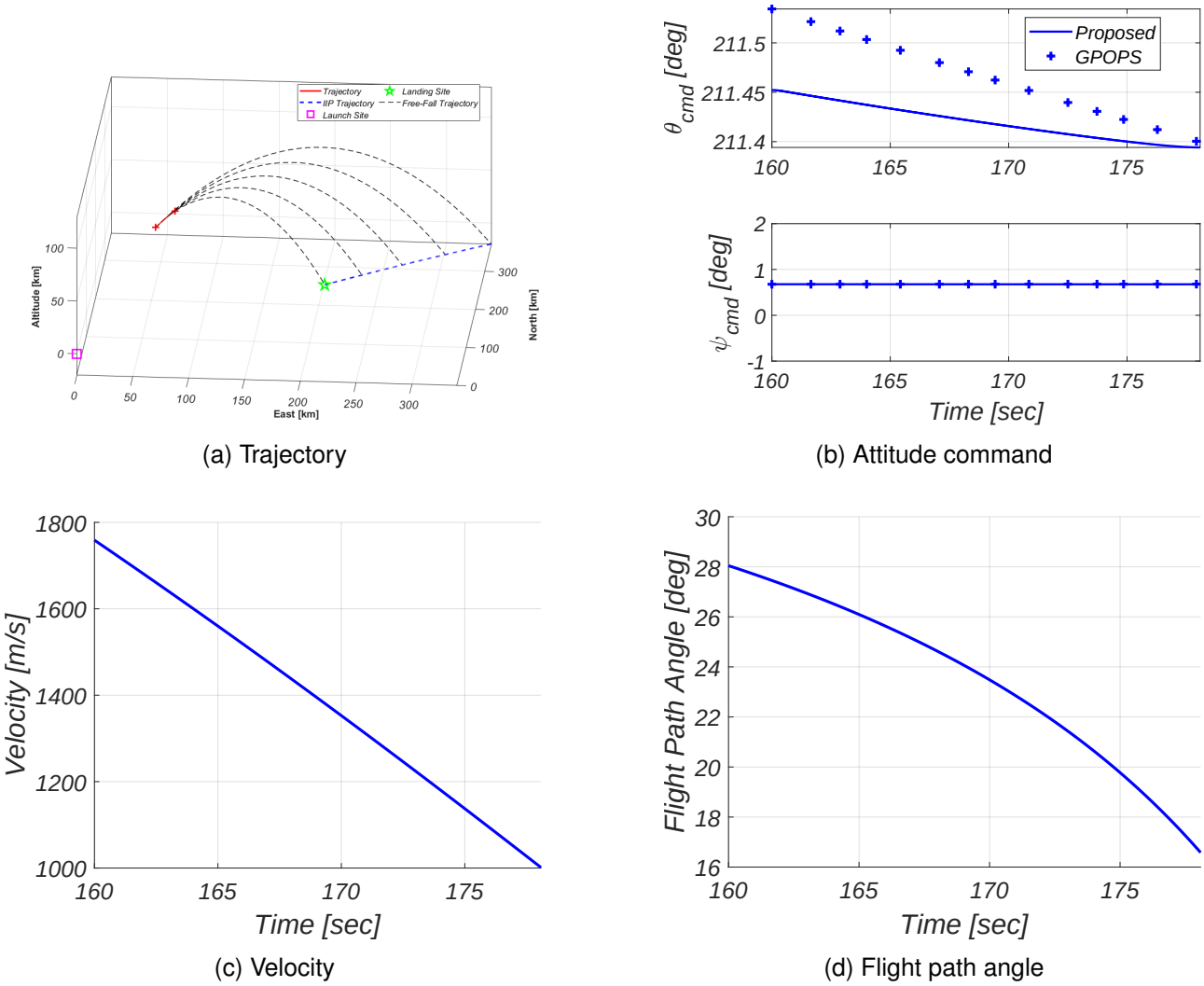
(c) Velocity



(d) Flight path angle

Figure 7 – RTLS scenario.

Figure 8 shows the simulation results for the DRL mission. The commanded yaw angle is nearly constant at 0.68 degrees, and the pitch angle is around 211.4 degrees. In both cases, the same initial guess was used, and the algorithm converged to the solution of P_0. Both DRL and RTLS missions correct the guidance error mainly by a pitch maneuver in the downrange direction. However, the optimal pitch angle for minimum fuel consumption is significantly different. This is the main reason that a greedy strategy that does not consider the future horizon until burnout, such as maximizing the rate of IIP toward the landing site [1], sometimes loses optimality. The proposed method inherently prevents this issue through its problem formulation.
The DRL mission does not require as much trajectory correction as the RTLS case. The total burn time until the error falls below 50 m is 18.084 seconds, and the time-to-go compensated total burn time is 18.092 seconds.



(a) Trajectory

(b) Attitude command

(c) Velocity

(d) Flight path angle

Figure 8 – DRL scenario.

4.2 Fuel-optimality Analysis

The constant attitude profile results from the simplified optimal control problem, which is not an exact solution of the original problem P_{BB}. The fuel optimality of the proposed method is compared with that of the offline-optimized trajectory of P_{BB}. The problem is solved using the GPOPS-II solver [9]. Figures 7b and 8b show the attitude command from the proposed algorithm and the offline optimization results. The proposed method issued attitude commands very similar to the optimization results. The difference comes from the formulation of P_k, which forces the remaining attitude profile to be constant. Table 4 shows a comparison of the burn times between the proposed method and the optimization results. As the thrust magnitude is fixed at the maximum value, the burn time is directly proportional to

the fuel consumption. The optimality degradation of the proposed method is practically non-existent. This result confirms that the optimal solution of (9) is practically valid under a spatially varying gravitational field and a rotating Earth for the boost-back scenario.

Table 4 – Total burn time comparison

	Burn Time (t_f^{BB})		
Case	Proposed (sec)	GPOPS (sec)	Difference (%)
RTLS	43.2719	43.2668	**0.012**
DRL	18.0920	18.0880	**0.022**

4.3 Monte Carlo Test

The proposed boost-back algorithm fundamentally relies on the nominal vehicle dynamics model to predict the burnout state. This is an inherent characteristic of predictor-corrector type guidance algorithms. Hence, model uncertainty or unknown disturbance induces a mismatch between the true vehicle state and predicted vehicle states. Since the guidance algorithm is repeatedly called with the latest vehicle state from sensor measurements, a feedback mechanism is effectively embedded in the algorithm design. The robustness of the proposed algorithm is tested for the RTLS mission with model uncertainty imposed. Unlike the nominal simulation in the previous section, the autopilot delay is modeled as a first-order response with a time constant of 1.0 second for the pitch and yaw axes. The initial conditions for the attitude delay model are summarized in Table 5. As the proposed algorithm is the most sensitive to thrust acceleration uncertainty ($\Delta T/m$), the model uncertainties are imposed on the thrust magnitude, initial mass, and specific impulse. The distributions of the uncertainties are shown in Table 6.

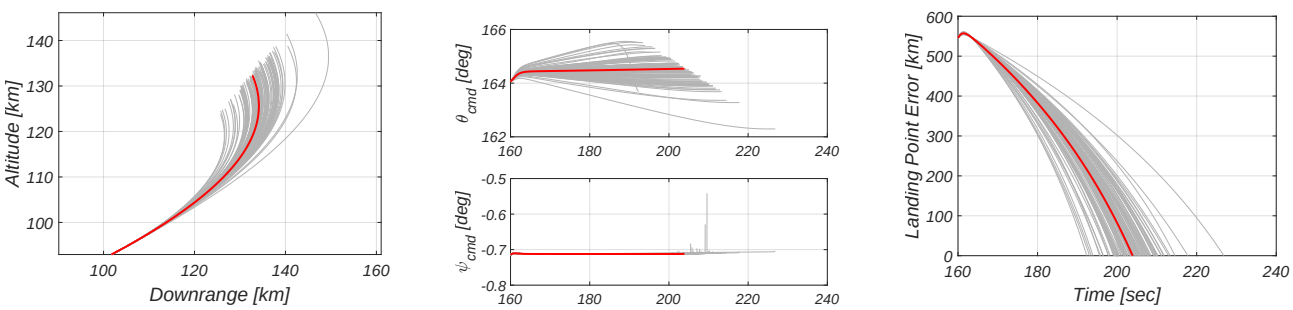
A total of 500 samples were used for the Monte Carlo simulation. The same algorithm setup as shown in Table 3 was used. Figure 9 shows the Monte Carlo results. All test cases succeeded in correcting the landing point error to within 50 m at the boost-back burnout despite model uncertainties and autopilot delay. Thrust acceleration uncertainty alters the resulting closed-loop trajectory significantly compared to the nominal trajectory. However, the issued attitude command remains nearly constant.

Table 5 – Attitude autopilot delay model

Variable	Value	Unit
Pitch Angle, θ_0	30.0	deg
Yaw Angle, ψ_0	0.0	deg
Time Constant, τ	1.0	sec

Table 6 – Thrust uncertainty distributions

Variable	Distribution
Thrust, ΔT	$\mathcal{N}(0, 10^2)\%$
Specific Impulse, ΔI_{sp}	$\mathcal{N}(0, 10^2)\%$
Initial Mass, Δm_0	$\mathcal{N}(0, 5^2)\%$



(a) Trajectory (b) Attitude command (c) Landing position error

Figure 9 – Monte Carlo results.

The pitch angle commands lie within $164 \pm 2^\circ$, and the yaw angle remains around -0.7°. In some cases, an abrupt yaw-angle command change occurs right before burnout. This phenomenon is characteristic of predictive guidance algorithms as time-to-go approaches zero. Despite this singularity, the proposed algorithm did not fail. Also, as the guidance algorithm provides the time-to-go information, it may no longer be called after a specified time-to-go limit in order to avoid singularity. It is concluded from the Monte Carlo analysis that the proposed algorithm is able to effectively bring the guidance parameter $\boldsymbol{p}_k$ to the constraint manifold by a single-step correction at each guidance cycle, even under relatively severe model uncertainty in the thrust acceleration and mass profiles.
The average computation time per guidance call, measured over 500 test cases with a MATLAB implementation, was 1.152 ms on an i7-14700K 3.4 GHz desktop CPU. Considering the slower computation speed of onboard flight computers, the proposed algorithm is still computationally lightweight and suitable for real-time use, owing to the deterministic number of corrections.

5. Conclusion

This research proposes an optimal guidance algorithm for the boost-back burn phase of reusable launch vehicles. The optimal control problem is mathematically formulated, and the optimal attitude command is demonstrated to be constant. Based on the insight of constant attitude commands, a parameter-optimization-based algorithm incorporating a single-step correction mechanism is developed. Numerical simulations demonstrate that the proposed algorithm successfully nullifies the predicted landing position error while minimizing fuel consumption, showing virtually optimal performance for both RTLS and DRL mission types. Monte Carlo analysis verified that the algorithm successfully corrected the guidance parameter onto the terminal constraint manifold even under model uncertainty. Furthermore, the algorithm is computationally tractable and suitable for real-time implementation.

6. Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2025-02213804).

7. Contact Author Email Address

Jaewook Shim: jaewooki1@kaist.ac.kr
Jae-Il Jang: jji249@kaist.ac.kr
Chang-Hun Lee: lckdgns@kaist.ac.kr

8. Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.

References

[1] B. U. Jo and J. Ahn. Near time-optimal feedback instantaneous impact point (iip) guidance law for rocket. *Aerospace Science and Technology*, 76:523–529, 2018.

[2] B. U. Jo and J. Ahn. Midcourse guidance algorithm for return to launch site of reusable rockets. *ASCEND 2020*, 2020.

[3] B. U. Jo, S. Y. Han, and J. Ahn. Fuel-optimal boost-back guidance algorithms for reusable launch vehicles. *Advances in Space Research*, 76(9):5294–5308, 2025.

[4] Changzhu Wei, Xiaozhe Ju, R. Wu, Y. He, and Y. Diao. Geometry and time updaters-based arbitrary-yaw iterative explicit guidance for fixed-thrust boost back of vertical take-off/vertical landing reusable launch vehicles. *Aerospace Science and Technology*, 95:105433, 2019.

[5] Binfeng Pan, Zheng Chen, Ping Lu, and Bo Gao. Reduced Transversality Conditions in Optimal Space Trajectories. *Journal of Guidance, Control, and Dynamics*, 36(5):1289–1300, September 2013.

[6] J. Ahn and W. R. Roh. Noniterative instantaneous impact point prediction algorithm for launch operations. *Journal of Guidance, Control, and Dynamics*, 35(2):645–648, 2012.

[7] Moritz Diehl, Hans Georg Bock, and Johannes P. Schlöder. A Real-Time Iteration Scheme for Nonlinear Optimization in Optimal Feedback Control. *SIAM Journal on Control and Optimization*, 43(5):1714–1736, January 2005.

[8] Robert P. Dickinson and Robert J. Gelinas. Sensitivity analysis of ordinary differential equation systems, A direct method. *Journal of Computational Physics*, 21(2):123–143, June 1976.

[9] Michael A. Patterson and Anil V. Rao. Gpops-ii: A matlab software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming. *ACM Trans. Math. Softw.*, 41(1), 2014.