DOI: 10.13009/EUCASS2023-527
Aerospace Europe Conference 2023 – 10TH EUCASS – 9TH CEAS

Entry Vehicle Trajectory Optimization using Convex Programming and Post-correction Technique

Cheol-Goo Jung[1], Tae-Hun Kim[2], Hyung-Sik Choi[3], and Chang-Hun Lee[4†]
[1,4] Korea Advanced Institute of Science and Technology
[2]Sunchon National University
[3]Korea Aerospace Research Institute
[1]jjulgoo22@kaist.ac.kr, [2]thkim@scnu.ac.kr, [3]chs@kari.re.kr, [4]lckdgns@kaist.ac.kr
[†] Corresponding Author

Abstract
This paper considers the trajectory optimization of entry vehicles using convex programming. Optimal entry trajectory satisfying path constraints on heating rate, dynamic pressure, and load factor is obtained with sequential second-order cone programming (SOCP). The nonlinear dynamics and path constraints are simplified by changing the independent variable into energy. To mitigate infeasibility and solution divergence problems, the post-correction technique is utilized as well as slack variables and trust-region. The step length of obtained solution at each iteration is modified based on the amount of linearization error. The robustness and effectiveness of the proposed method is demonstrated through numerical simulation.

1. Introduction

The objective of the entry guidance is to steer the direction of the aerodynamic lift so as to guide the space vehicle from its entry interface point (EIP) to the terminal conditions. The entry guidance problem is identified as a challenging problem because of many entry path constraints as well as its highly nonlinear dynamics. These inequality entry path constraints on heating rate, dynamic pressure, and load factor should be taken into account when developing the entry guidance not to incur fatal damage on entry vehicles. In the past, entry guidance relies on approximate, and heuristic methods due to a lack of computational requirements. In [1], the entry guidance is designed based on the analytic solution of the drag-acceleration equation satisfying terminal conditions and constraints. Drag-energy-based guidance for trajectory control is proposed in [2]. In this paper, a linear control law for following the drag reference is derived. An adaptive entry guidance algorithm is presented for mission planning and trajectory updates during the onboard mission in [3]. The trajectory is obtained by maximizing the range of the vehicle. The predictor-corrector method is proposed in [4]. The main notable potential of this method is that the trajectory of the entry vehicle is adaptively modified when the trajectory largely deviates from the nominal trajectory without any path planning. The class of predictor-corrector method has been successfully applied to entry guidance problems in many literatures [5-8]. The main challenges of developing entry guidance law are path constraints on heating rate, dynamic pressure, and load factor which are crucial for the successful entry mission. Since the previous studies rely on analytic or heuristic methods, much effort and time are needed to handle the path constraints when deriving entry guidance law. Furthermore, this heuristic method can adversely affect and reduce the performance of the entry guidance.
A trajectory optimization method which is one of the computational methods can be an effective alternative to handle these path constraints and performance degradation. The trajectory optimization method is a powerful tool for optimal control problems. The dynamics, path constraints, and performance index are discretized, and obtained static problems are solved using a numerical optimization solver. The optimal solution that satisfies nonlinear dynamics and path constraints can be obtained without relying on approximate and empirical relationships. As computational power increases and efficient optimization algorithms are developed, trajectory optimization is gaining popularity in aerospace engineering applications such as guidance and path planning. Convex programming is a promising optimization method for real-time applications because of its polynomial complexity and predictable and bounded calculation time. Convex problems can be solved by well-developed optimization algorithms such as the primal-dual interior point method [9]. Convex programming has been applied to many aerospace applications such as planetary-powered landing [10], multiagent path planning [11], missile trajectory optimization [12], collision avoidance [13], and so on. Also, entry trajectory optimization using convex programming is carried out in some studies [14-17]. To

DOI: 10.13009/EUCASS2023-527

apply convex programming to entry trajectory optimization, the problem should be reformulated into a specific form suitable to convex programming. Since the convex programming requires the equations to be linear, the highly nonlinear dynamics of entry trajectory and path constraints should be successively linearized with respect to the previous solution as in most of the previous entry trajectory optimization studies in [14-17]. Then, the linearized problem is iteratively solved using successive convex programming. The significant challenge of this iterative process is that the linearized problem may become infeasible and the solution is prone to diverge due to large deviations from the original nonlinear dynamics. To tackle these problems, auxiliary methods such as the trust-region method are utilized to assist the convergence of solutions and to mitigate infeasibility problems.

The technique developed in this paper focuses on alleviating the solution divergence and infeasibility problem that can easily occur during the successive convexification process. To handle the problem of inconsistency in linearization, additional slack variables are introduced in the quadratic trust-region and augmented to the performance index for stable convergence. Furthermore, the post-correction technique regulates the step length from the previous solutions and prevents excessive deviation from the original dynamics by estimating the error between linear and nonlinear dynamics. The control input is changed into new variables contained within the convex set. This change of control input helps to circumvent the undesirable jittering control profile caused by successive linearization of dynamics with respect to the control variables. By using energy as an independent variable, the path constraints on heating rate, dynamic pressure, and load factor are transformed into linear constraints of altitude which is implementable in convex programming without additional modification. By aggregating all these methods, the trajectory optimization problem is formulated and the optimal trajectory solution is obtained using successive convex programming. The proposed methods alleviate the oscillation of solution that occurs during the iterative process and show fast and smooth convergence. Numerical simulations are performed to demonstrate the effectiveness and some of the notable aspects of the proposed method.

The remainder of this paper is organized as follows. The problem statement is given in section 2. Then, the trajectory optimization problem is reformulated into a form suitable for convex programming in section 3. Finally, the numerical simulation and conclusions are given.

2. Problem statement

This section presents the details of the dynamics and path constraints of entry vehicles. Then, the trajectory optimization problem is defined with a specific performance index.

2.1 Dynamics and constraints of entry vehicle

The nondimensionalized dynamics of the entry vehicle with respect to energy variables is presented as follows [8] :

$$
\begin{aligned}
\frac{dr}{de} &= \sin\gamma / D \\
\frac{d\theta}{de} &= \cos\gamma \sin\psi / (rD\cos\phi) \\
\frac{d\phi}{de} &= \cos\gamma\cos\psi / (rD) \\
\frac{d\gamma}{de} &= [L\cos\sigma + (V^2 - 1/r)\cos\gamma / r + 2\Omega V\cos\phi\sin\psi \\
&+\Omega^2 r\cos\phi(\cos\gamma\cos\phi + \sin\gamma\cos\psi\sin\phi)] / (V^2 D) \\
\frac{d\psi}{de} &= [L\sin\sigma/\cos\gamma + V^2\cos\gamma\sin\psi\tan\phi / r \\
&-2\Omega V(\tan\gamma\cos\psi\cos\phi - \sin\phi) + \Omega^2 r\sin\psi\sin\phi\cos\phi/\cos\gamma] / (V^2 D)
\end{aligned}
\tag{1}
$$

where r is the radial distance between the center of the Earth and the entry vehicle, θ and ϕ are the longitude and altitude, respectively. V is earth-relative velocity. γ is the flight-path angle of the velocity vector, and ψ is the heading angle of the velocity vector measured clockwise from the north direction. The radial distance is normalized by the Earth's radius R_0 = 6378.14 km. The time is normalized by $\sqrt{R_0/g_0}$, where g_0 = 9.81 m/s^2. From these two

DOI: 10.13009/EUCASS2023-527

normalization factors, the velocity is normalized by $\sqrt{R_0 g_0}$. The Earth's rotation speed Ω is normalized by $\sqrt{g_0/R_0}$. The bank angle σ is the control input and is defined by the rotation angle of the entry vehicle about the velocity vector. The aerodynamic lift and drag are denoted by L and D which are normalized by g_0 as follows :

$$L = \frac{1}{2} R_0 \rho V^2 S_{\mathrm{ref}} C_L / m \tag{2}$$

$$D = \frac{1}{2} R_0 \rho V^2 S_{\mathrm{ref}} C_D / m \tag{3}$$

where S_{ref} and m are the reference area and mass of the entry vehicle, respectively. ρ is the atmospheric density which is a function of r and is modeled as follows :

$$\rho = \rho_0 \exp(-(r - R_0) / h_s) \tag{4}$$

where ρ_0 is the density of the Earth's surface and h_s is constant which can be determined by the atmospheric density data. C_L and C_D are the lift and drag coefficients which are functions of the angle-of-attack and Mach number. To simplify the problem, the following models of lift and drag coefficient in the hypersonic range are chosen [18].

$$C_L = -0.041065 + 0.016292\alpha + 0.0002602\alpha^2 \tag{5}$$

$$C_D = 0.080505 - 0.03026 C_L + 0.86495 C_L^2 \tag{6}$$

Where α denotes the angle-of-attack in degrees. The independent variable of the entry vehicle dynamics is an energy-like variable e that is defined by

$$e = \frac{1}{r} - \frac{V^2}{2} \tag{7}$$

This energy variable monotonically increases when the Earth's rotation speed is neglected [8]. From Eq. (7), the velocity can be calculated at any given e and r. Furthermore, velocity can be approximated by the following equation since the normalized r is almost unity.

$$V = \sqrt{2(1/r - e)} \approx \sqrt{2(1 - e)} \tag{8}$$

This denotes that velocity is determined only by the independent variable. The angle-of-attack profile is typically predesigned with respect to the velocity. The angle-of-attack profile is determined by considering the thermal protection, downrange/crossrange requirements, and trim flight. In this paper, the reference angle-of-attack profile in [14] is used.

$$\alpha = \begin{cases} 40, & \text{if } V > 4570 m/s \\ 40 - 0.20705(V - 4570)^2 / 340^2, & \text{otherwise} \end{cases} \tag{9}$$

The reference angle-of-attack profile is shown in Fig. 1. Where the angle-of-attack gradually decreases below Mach 13.

DOI: 10.13009/EUCASS2023-527

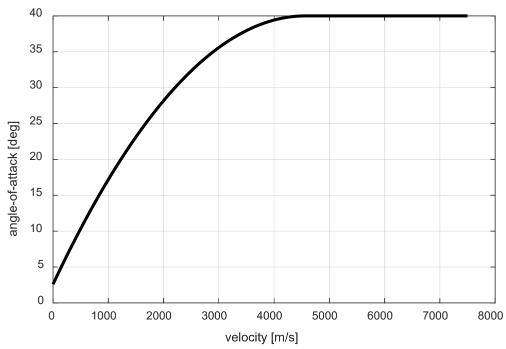

Figure 1: Reference angle-of-attack profile

The initial conditions of the state at a given initial energy e_0 is presented below. The initial conditions are determined from the entry interface point.

$$r(e_0) = r_0^*,\ \theta(e_0) = \theta_0^*,\ \phi(e_0) = \phi_0^*,\ \gamma(e_0) = \gamma_0^*,\ \psi(e_0) = \psi_0^* \tag{10}$$

The terminal conditions at the given terminal energy are shown below. The appropriate terminal conditions can be determined by the terminal area energy management phase.

$$r(e_f) = r_f^*,\ \theta(e_f) = \theta_f^*,\ \phi(e_f) = \phi_f^*,\ \gamma(e_f) = \gamma_f^*,\ \psi(e_f) = \psi_f^* \tag{11}$$

The common entry path constraints on heating rate, dynamic pressure, and load factor should be satisfied during the entry phase for safety issues. The heating rate constraint prevents the ablation caused by extreme aerodynamic heating. Excessive hinge moment of the aerodynamic fin can cause breakage of actuators and be prevented by dynamic pressure constraint. Finally, the load factor constraint is considered to avoid structural damage to the vehicle. The path constraints are presented as follows:

$$\dot{Q} = k_Q \sqrt{g_0 R_0}^{3.15} \sqrt{\rho} V^{3.15} \le \dot{Q}_{\max} \tag{12}$$

$$\overline{q} = 0.5 g_0 R_0 \rho V^2 \le \overline{q}_{\max} \tag{13}$$

$$n = \sqrt{L^2 + D^2} \le n_{\max} \tag{14}$$

The heating rate $\dot{Q}$ is defined at a stagnation point on the surface of the vehicle. The heating rate is in W/m^2 and k_Q = 9.4369×10^{-5}. The unit of dynamic pressure $\bar{q}$ is given as N/m^2, and unit of load factor n is in g_0. Since the velocity V is determined by the independent variable e by Eq. (8) and density is a function of r as Eq. (4), the path constraints of Eqs. (12-14) can be equivalently replaced by simple constraints on r as follows:

$$r(e) \ge c_Q(e),\ r(e) \ge c_q(e),\ r(e) \ge c_n(e) \tag{15}$$

The path constraints of Eq.(15) can be simplified by a single constraint.

$$r(e) \ge \max\{\ c_Q(e), c_q(e), c_n(e)\ \} = c_{\max}(e) \tag{16}$$

DOI: 10.13009/EUCASS2023-527

Finally, the bank angle σ is bounded by maximum and minimum values. The bank angle is bounded by the same magnitude since the entry vehicle typically has a nearly symmetrical body shape.

$$-\sigma_{\max} \le \sigma \le \sigma_{\max} \tag{17}$$

2.2 Entry trajectory optimization problem

The performance index of the trajectory optimization problem can be selected depending on the specific purpose. In this paper, the following performance index on total heat is considered to minimize the heat load.

$$J = \int_{t_0}^{t_f} k_Q \sqrt{g_0 R_0}^{3.15} \sqrt{\rho} V^{3.15} dt \tag{18}$$

This performance index can be replaced with respect to e as follows :

$$J = k_J \int_{e_0}^{e_f} \left[\sqrt{\rho} V^{3.15} / (DV) \right] de \tag{19}$$

where the constant k_J is $k_Q \sqrt{R_0 g_0}^{3.15}$. By aggregating dynamics, initial and terminal conditions, path constraints, and performance index, the entry trajectory optimization problem is defined.

Problem 1 (entry trajectory optimization problem) :

minimize : (19)

subject to : (1), (10), (11), (16), (17)

3. Solution by successive convex programming

In this section, **Problem 1** is reformulated into a form suitable for application to convex programming.

3.1 Control input change

To utilize convex programming, successive linearization of nonlinear dynamics is necessary. However, it is revealed that the linearization with respect to control input σ causes a jittering input solution profile of high frequency. This jitter in solution results in a slow convergence rate during the successive process. The reason for this jittering profile is the coupling between states and control in dynamics. To address this problem, a pair of controls are introduced by following [15].

$$u_1 = \cos\sigma, \ u_2 = \sin\sigma \tag{20}$$

Since new control inputs u_1 and u_2 are not independent, the following additional constraint should be satisfied.

$$u_1^2 + u_2^2 = 1 \tag{21}$$

DOI: 10.13009/EUCASS2023-527

Instead of linearizing Eq. (21), it is relaxed to the second-order cone constraint of Eq. (22). The following constraint satisfies the convex set and can be directly applied to convex programming.

$$u_1^2 + u_2^2 \le 1 \tag{22}$$

It was proved that this relaxation does not change the optimal solution and the solution of control input remains in Eq. (21) with an appropriate regularization term [15]. The proof of the exactness of relaxation is extensively explained in [15]. In this paper, we used the relaxation method without proof and the regularization term is explained in the following subsections. The constraint on the magnitude of the control input of Eq. (17) is converted into the following constraint.

$$u_1 \ge u_{1,\max} = \cos\sigma_{\max} \tag{23}$$

By defining states and control vectors as $\mathbf{x} = [r\ \theta\ \phi\ \gamma\ \phi]^T$ and $\mathbf{u} = [u_1\ u_2]^T$, the dynamics of Eq. (1) is reformulated into the following compact form.

$$\mathbf{x}' = \mathbf{f}(\mathbf{x}) + B(\mathbf{x})\mathbf{u} + \mathbf{f}_\Omega(\mathbf{x}) \tag{24}$$

where

$$\mathbf{f} = \begin{bmatrix} \sin\gamma / D \\ \cos\gamma \sin\psi (rD\cos\phi) \\ \cos\gamma\cos\psi / (rD) \\ (1/D - 1/(rV^2D))\cos\gamma / r \\ \cos\gamma\sin\psi\tan\phi / (rD) \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \dfrac{(C_L / C_D)}{V^2} & 0 \\ 0 & \dfrac{(C_L / C_D)}{V^2\cos\gamma} \end{bmatrix}$$

The derivative $(\,)'$ denotes the differentiation with respect to e. The third term of Eq. (24) is related to the rotation rate of the Earth and is relatively small than the other terms. This term can be easily derived from Eq. (1) and is not presented here.

3.2 Linearization

Since the dynamics of Eq. (24) is highly nonlinear, this cannot be utilized in convex programming. Suppose that the solution of k-th iteration is $\mathbf{x}^{(k)} = \left[r^{(k)}\ \theta^{(k)}\ \phi^{(k)}\ \gamma^{(k)}\ \phi^{(k)}\right]^T$. Then, the linearized dynamics is derived as follows :

$$\frac{d\mathbf{x}}{de} = A(\mathbf{x}^{(k)})\mathbf{x} + B(\mathbf{x}^{(k)})\mathbf{u} + C(\mathbf{x}^{(k)}) \tag{25}$$

where

$$A(\mathbf{x}^{(k)}) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}^{(k)}),\ \ C(\mathbf{x}^{(k)}) = \mathbf{f}(\mathbf{x}^{(k)}) - A(\mathbf{x}^{(k)})\mathbf{x}^{(k)} + \mathbf{f}_\Omega(\mathbf{x}^{(k)})$$

Since the derivation of the Jacobian matrix A is straightforward, the detailed description is omitted. The only state-dependent value in matrix B is $\cos\gamma$. In the entry flight phase, the flight path angle γ is small and slowly varying. From this observation, $\partial B / \partial \mathbf{x}$ is neglected. The third term of Eq. (24) is a very small value and simply lagged using the previous state solutions.

In convex programming, the performance index should be linear in the variables. The performance index of Eq. (19) is linearized as follows :

DOI: 10.13009/EUCASS2023-527

$$J_0 = \bar{k}_J \int_{e_0}^{e_f} \frac{V(e)^{0.15}}{C_D(e)} \left[c_1(r^{(k)}) r + c_2(r^{(k)}) \right] de \tag{26}$$

where

$$\bar{k}_J = \frac{2k_J m}{R_0 S_{\text{ref}}},\ c_1(r^{(k)}) = -\frac{\rho_r^{(k)}}{2(\rho^{(k)})^{3/2}},\ c_2(r^{(k)}) = \frac{1}{(\rho^{(k)})^{1/2}} - \frac{\rho_r^{(k)}}{2(\rho^{(k)})^{3/2}} r^{(k)}$$

Note that the velocity and drag coefficient are determined by the independent variable e from Eq. (6), (8), and (9). The constraints on the terminal position of θ and ϕ are rarely satisfied in early iterations. The problem may become infeasible because the linearized dynamics derived from the initial guess is inconsistent. To address this difficulty, the terminal conditions on θ and ϕ are replaced by the soft constraints as follows :

$$\left|\theta(e_f) - \theta_f^*\right| \le s_\theta,\ \ \left|\phi(e_f) - \phi_f^*\right| \le s_\phi \tag{27}$$

The auxiliary variables s_θ and s_ϕ are augmented to the performance index. The performance index is presented below by adding the regularization term for exact relaxation.

$$J = J_0 + w_\theta s_\theta + w_\phi s_\phi + w_\psi \int_{e_0}^{e_f} \psi de \tag{28}$$

The weights w_θ, w_ϕ, and w_ψ should be selected to be small enough so that the original performance index J_0 is essentially unaffected.

3.3 Discretization and trust-region method

In this section, the dynamics, constraints, and performance index in the continuous-time domain are discretized. The independent variable e is divided into the uniformly distributed N+1 node points with a step size of $\Delta e = (e_f - e_o)/N$. The discretized node points are denoted by $e_i = e_o + i\Delta e\ (i = 0,1,\ldots,N)$. The states corresponding to each node point are simplified by $r_i = r(e_i)$, $\theta_i = \theta(e_i)$, $\phi_i = \phi(e_i)$, $\gamma_i = \gamma(e_i)$, and $\psi_i = \psi(e_i)$. The dynamics of Eq. (25) is integrated using the trapezoidal rule.

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \frac{\Delta e}{2}\left[A_{i-1}^{(k)}\mathbf{x}_{i-1} + B_{i-1}^{(k)}\mathbf{u}_{i-1} + C_{i-1}^{(k)} + A_i^{(k)}\mathbf{x}_i + B_i^{(k)}\mathbf{u}_i + C_i^{(k)} \right],\ i = 1,2,\ldots,N \tag{29}$$

where

$$A_m^{(k)} = A\left(\mathbf{x}^{(k)}(e_i)\right),\ B_m^{(k)} = B\left(\mathbf{x}^{(k)}(e_i)\right),\ \ C_m^{(k)} = C\left(\mathbf{x}^{(k)}(e_i)\right)$$

To address the inconsistency problem of dynamics linearization, the following trust-region constraint is typically utilized in successive linearization processes.

$$\left\|\mathbf{x}_i - \mathbf{x}_i^k\right\| \le \boldsymbol{\delta},\ \ i = 0,1,2,\ldots,N \tag{30}$$

where $\boldsymbol{\delta}$ is the constant vector that confines the change of each state. However, it is difficult to determine the proper magnitude of $\boldsymbol{\delta}$ for each state. The problem may become infeasible if inappropriate $\boldsymbol{\delta}$ is selected. In this paper, the following trust-region is used.

$$\left[\mathbf{x}_i - \mathbf{x}_i^k\right]^T \left[\mathbf{x}_i - \mathbf{x}_i^k\right] \le t_i,\ \ i = 0,1,2,\ldots,N \tag{31}$$

DOI: 10.13009/EUCASS2023-527

The auxiliary variables $t_i(i = 0,1,\dots,N)$ are multiplied by a weight w_{tr} and augmented to the performance index. This modified trust-region method can address the infeasibility problem even though the initial trajectories are roughly guessed.
The initial and terminal conditions are formulated using the discretized node points. Note that the terminal conditions on θ and ϕ are replaced by soft constraints from Eq. (27).

$$r_0 = r_0^*,\ \theta_0 = \theta_0^*,\ \phi_0 = \phi_0^*,\ \gamma_0 = \gamma_0^*,\ \psi_0 = \psi_0^* \tag{32}$$

$$r_N = r_f^*,\ \left|\theta_N - \theta_f^*\right| \le s_\theta,\ \left|\phi_N - \phi_f^*\right| \le s_\phi,\ \gamma_N = \gamma_f^*,\ \psi_N = \psi_f^* \tag{33}$$

The input constraints are discretized in the following form.

$$\left(u_1\right)_i \ge u_{1,\max},\ \ i = 0,1,2,\dots,N \tag{34}$$

$$\left[\left(u_1\right)_i\right]^2 + \left[\left(u_2\right)_i\right]^2 \le 1,\ \ i = 0,1,2,\dots,N \tag{35}$$

The final performance index is presented. The soft constraints on θ and ϕ, the regularization term, and the modified trust-region are considered.

$$J = \bar{k}_J \sum_{i=0}^{N}\left[\frac{V_i^{0.15}}{(C_D)_i}\left((c_1)_i^k r_i + (c_2)_i^k\right)\Delta e\right] + w_\theta s_\theta + w_\phi s_\phi + w_\psi \sum_{i=0}^{N}\psi_i \Delta e + w_{tr}\sum_{i=0}^{N} t_i \tag{36}$$

where

$$(c_1)_i^k = c_1(r_i^{(k)}),\ (c_2)_i^k = c_2(r_i^{(k)})$$

From the discretized dynamics, constraints, and performance index, the convex programming problem is defined by the following.

Problem 2 (convex programming problem) :

minimize : (36)

subject to : (29), (31), (32), (33), (34), (35)

Problem 2 is sequentially solved using convex programming until the following termination condition is satisfied.

$$\max_i \left|\mathbf{x}_i^k - \mathbf{x}_i^{k-1}\right| < \boldsymbol{\varepsilon}_c \tag{37}$$

where $\boldsymbol{\varepsilon}_c$ is the constant vector. If the difference between the previous solution and the current solution is smaller than $\boldsymbol{\varepsilon}_c$, successive convex programming is terminated.

3.4 Post-correction technique

Even though the soft constraints on terminal conditions and the turst-region method are used in successive processes, the solution diverges or very slowly converges to the final solution in some flight scenarios. This is because the entry flight phase has narrow solution space and significant deviation from the true dynamics caused by the linearization of highly nonlinear dynamics. To circumvent this difficulty, the post-correction technique is introduced. First, the error between nonlinear and linearized dynamics for each state and node is as follows :

DOI: 10.13009/EUCASS2023-527

$$\mathbf{E}_i = \left|\mathbf{f}(\mathbf{x}^{(k)}) - \mathbf{f}_L(\mathbf{x}^{(k)})\right|,\ i = 0,1,2,...,N \tag{38}$$

where

$$\mathbf{f}_L(\mathbf{x}^{(k)}) = A(\mathbf{x}^{(k-1)})\mathbf{x}^{(k)} + \left(\mathbf{f}(\mathbf{x}^{(k-1)}) - A(\mathbf{x}^{(k-1)})\mathbf{x}^{(k-1)}\right)$$

The linearization error $\mathbf{E}_i$ is calculated using the solution of k-th iteration $\mathbf{x}^{(k)}$. The $\mathbf{f}_L(\mathbf{x}^{(k)})$ is obtained by substituting the solution $\mathbf{x}^{(k)}$ into (k -1)-th linearized dynamics. $\mathbf{E}_i$ indicates how much the obtained solution deviates from the true nonlinear dynamics. By utilizing this linearization error, the step size of the solution update is modified. The state is updated by following the algorithm.

State update algorithm :

$$\begin{cases} \mathbf{x}^{(k)}_{i,j,updated} \leftarrow \mathbf{x}^{(k-1)}_{i,j} + \beta\left(\mathbf{x}^{(k)}_{i,j} - \mathbf{x}^{(k-1)}_{i,j}\right), & \text{if } \mathbf{E}_{i,j} > \varepsilon_{LE} \\ \mathbf{x}^{(k)}_{i,j,updated} \leftarrow \mathbf{x}^{(k)}_{i,j}, & \text{otherwise} \end{cases}$$

where $\mathbf{E}_{i,j}$ denotes the linearization error at i-th node of j-th state. β is the step size between 0 and 1. The variable $\mathbf{x}_{i,j}$ is the j-th state of i-th node (e.g. $\mathbf{x}_{3,1}$ is range r at the third node). If the linearization error at a specific node and state is sufficiently larger than the threshold $\boldsymbol{\varepsilon}_{LE}$, the state is updated with a small step size to prevent excessive deviation from the true dynamics. The overall iterative process of the proposed method is summarized as follows :

Solution by successive convex programming and post-correction technique :

Step 1) Initial trajectory guess : $\mathbf{x}_i^0 (i = 0,1,\dots,N)$ at every node point is selected. And set $k = 1$.

Step 2) Solving the convex programming problem : For $k \geq 1$, new optimal solutions $\boldsymbol{x}_i^k, \boldsymbol{u}_i^k$ $(i = 0,1,\dots,N)$ is calculated by solving **Problem 2** with the previous solutions $\mathbf{x}_i^{k-1}, \mathbf{u}_i^{k-1}$ $(i = 0,1,\dots,N)$.

Step 3) State update : The state is updated using **State update algorithm**. If $E_{i,j} > \varepsilon_{LE}$, the j-th state of the i-th node point $\mathbf{x}_{i,j}^k$ is updated by $\mathbf{x}_{i,j}^{k-1} + \beta\left(\mathbf{x}_{i,j}^k - \mathbf{x}_{i,j}^{k-1}\right)$.

Step 4) Termination condition check : If the maximum values of the difference between $\mathbf{x}_{i,j}^k$ and $\mathbf{x}_{i,j}^{k-1}$ satisfy the condition $\max_i\left|\mathbf{x}_i^k - \mathbf{x}_i^{k-1}\right| < \boldsymbol{\varepsilon}_c$, the iteration is finished and the optimal solution is set as $\boldsymbol{x}_i^k, \boldsymbol{u}_i^k$ $(i = 0,1,\dots,N)$. Otherwise, go to **Step 2**).

4. Numerical simulation

In this section, the proposed method is applied to the entry trajectory optimization problem. The parameters used in the simulations are summarized in Table 1. The initial and terminal conditions of the first simulation are shown in Table 2. The initial trajectories of states are assumed to be the line connecting the initial and terminal conditions (i.e. $\mathbf{x}_i^0 = \mathbf{x}_0^* + i \cdot \left(\mathbf{x}_f^* - \mathbf{x}_0^*\right)/N$). The initial control inputs $\boldsymbol{u}_i^o$ are set to zero on all nodes. The initial and terminal energies are given as $e_0 = 0.5403$ and $e_f = 0.9888$. The step size β is 0.7 and the termination condition is $\varepsilon_c = [100\text{ m}, 0.6\text{ deg}, 0.6\text{ deg}, 0.6\text{ deg}, 1\text{ deg}]$. The proposed method is compared with successive convex programming (SCP) without the post-correction technique and the general-purpose optimal control software (GPOPS). Figs. 2-3 shows the altitude profile solution at every iteration with velocity as the x-axis for the proposed method and SCP without post-correction technique. As shown in Fig. 2, the altitude profile stably and rapidly converges to solution with 10 iterations. The altitude profile at 4-th iteration is already almost converged to the final solution profile. However, SCP without the post-correction technique results in divergence of solution as shown in Fig. 3. The proposed post-correction technique can alleviate the solution divergence problem.

DOI: 10.13009/EUCASS2023-527

Table 1: Simulation parameters

Parameter	Value	Parameter	Value
Mass, M	105 000 [kg]	Max heating rate, $\dot{Q}_{max}$	1000 [kW/m^2]
Reference area, S_{ref}	300 [m^2]	Max dynamic pressure, $\bar{q}_{max}$	17 [kN/m^2]
Surface density, ρ_0	1.225 [kg/m^3]	Max load factor, n_{max}	3.5 [g]
Density parameter, h_s	8420 [km]	Number of nodes, N	120
Max bank angle, σ_{max}	80 [deg]	Weight of P.I., $(w_\theta, w_\phi, w_\psi, w_{tr})$	(200, 200, 0.1, 0.005)

Table 2: Initial and terminal conditions of Simulation 1

Initial/terminal conditions	$r - R_0$, [km]	θ, [deg]	$\boldsymbol{\phi}$, [deg]	$\boldsymbol{\gamma}$, [deg]	$\boldsymbol{\psi}$, [deg]
$\mathbf{x}_0$	105	0	0	-0.5	0
$\mathbf{x}_f$	30	15	70	-10	15

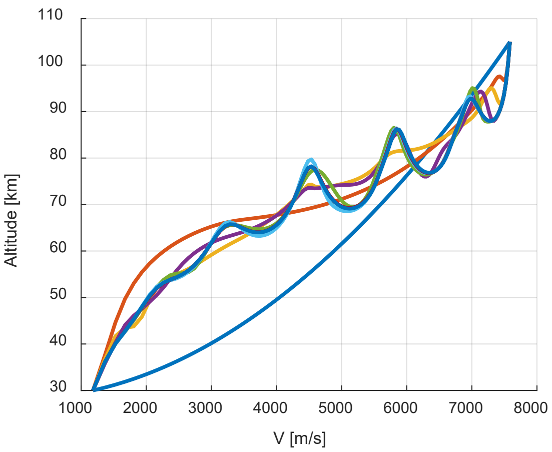

Figure 2: Altitude profile of every iteration (proposed method)

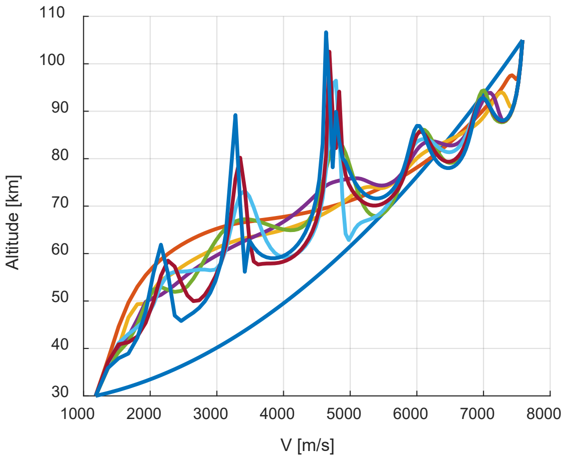

Figure 3: Altitude profile of every iteration (w/o post-correction technique)

Figure 4 presents the convergence of the performance index of the proposed method which shows the stable convergence. The changes of states from the previous solution are shown in Fig. 5. As shown in Fig. 5, every change of state rapidly converges to zero which denotes the stable convergence of iterative processes. Then, the solution of the proposed method is compared to that of GPOPS. The proposed method only takes about 0.08~0.12 sec to solve **Problem 2** at each iteration with Intel Core i5-12500 at 3GHz, and the total calculation time is 1.20 sec. On the other hand, GPOPS takes 16.13 sec. Fig. 6. Shows the very similar altitude profiles between the proposed method and GPOPS. Also, the path constraints on heating rate, dynamic pressure, and load factor are satisfied. The bank angle and position profiles show considerable similarity with little difference as shown in Fig. 7-8. It can be seen that the maximum bank angle command of 80 deg is given at around flight time of 600 sec and 850 sec.

DOI: 10.13009/EUCASS2023-527

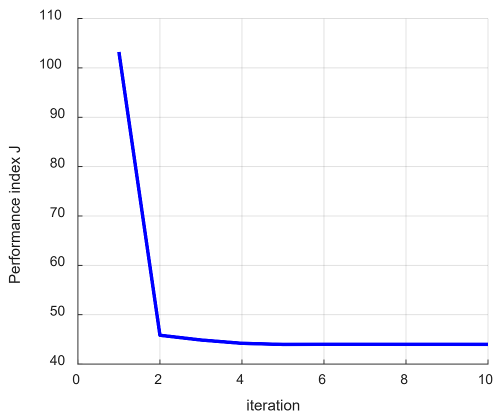

Figure 4: Performance index

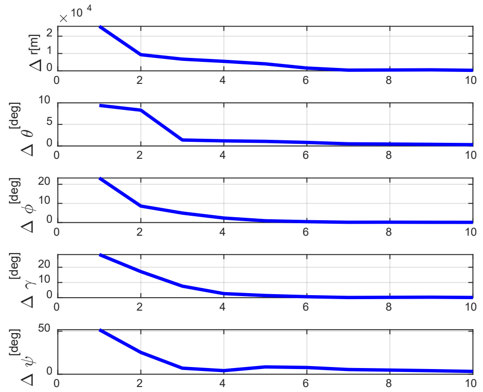

Figure 5: Change of states from the previous solutions

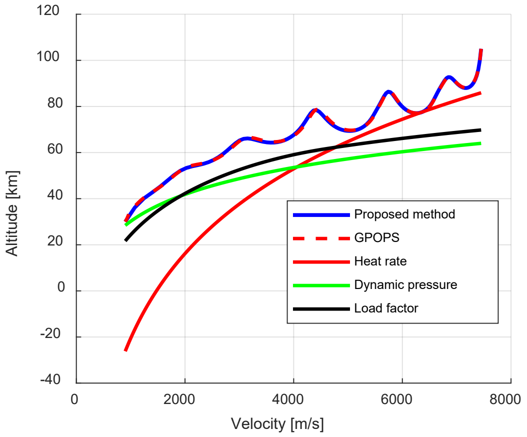

Figure 6: Altitude profile

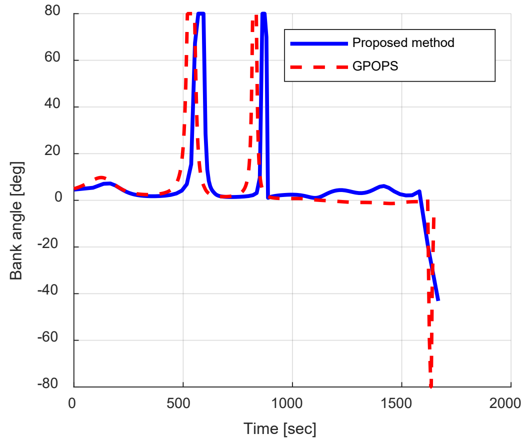

Figure 7: Bank angle profile

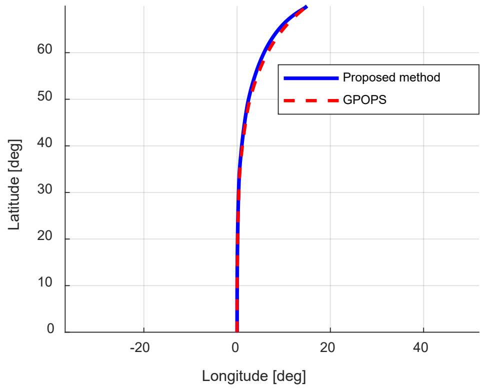

Figure 8: Longitude and Latitude profile

The initial and terminal conditions of the second simulation are shown in Table 3. Although not shown in the figures, it is confirmed that the performance index and change of states stably converge as in the first simulation. The results of the proposed method are compared to that of GPOPS. As shown in Fig. 9, the proposed method results in an altitude solution profile similar to GPOPS. The bank angle profiles are almost indiscernible from that of GPOPS except for the last flight time.

DOI: 10.13009/EUCASS2023-527

Table 3: Initial and terminal conditions of Simulation 2

Initial/terminal conditions	$r - R_0$, [km]	θ, [deg]	$\boldsymbol{\phi}$, [deg]	$\boldsymbol{\gamma}$, [deg]	$\boldsymbol{\psi}$, [deg]
$\mathbf{x}_0$	100	0	0	-0.5	0
$\mathbf{x}_f$	32	20	70	-10	40

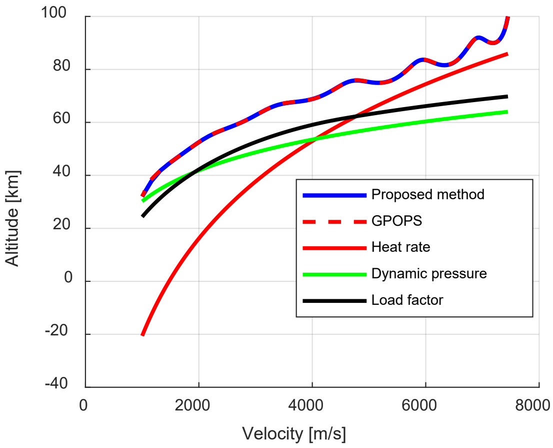

Figure 9: Altitude profile

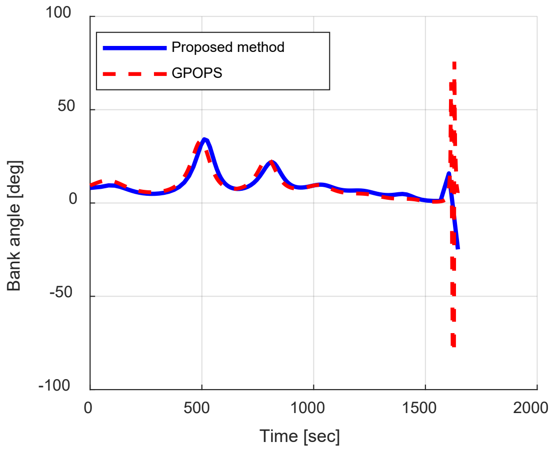

Figure 10: Bank angle profile

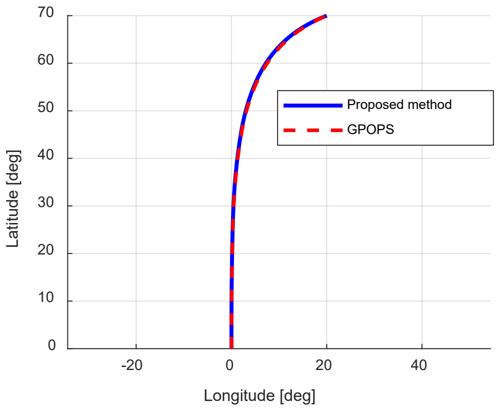

Figure 11: Longitude and latitude profile

5. Conclusions

This paper presents a successive convex programming to solve the entry trajectory optimization problem. The technique developed in this paper focuses on alleviating the solution divergence and infeasibility problem that can easily occur during the successive linearization process. The original dynamics is reformulated using energy as the independent variable. The path constraints on heating rate, dynamic pressure, and load factor are equivalently replaced by simple linear inequality constraints on the radial distance from the Earth's center. Then, the control input constraint is relaxed into a convex set without alternation of the optimal solution. To address the inconsistency caused by linearization, the modified trust-region method and post-correction technique are utilized. The proposed method alleviates the oscillation of the solution during the iterative process and results in fast and stable solution convergence.

DOI: 10.13009/EUCASS2023-527

6. Acknowledgement

This work has been supported by the Korea Aerospace Research Institute (KARI)'s research project entitled "Development of Control Performance Analysis Program for Spaceplanes."

References

[1] Harpold, J. C., and C. A. Graves Jr. 1978. Shuttle entry guidance. American Astronautical Society.
[2] Roenneke, A., and Markl, A. 1994. Re-entry control to a drag-vs-energy profile. *Journal of Guidance, Control, and Dynamics.* 17(5):916-920
[3] Roenneke, A. 2001. Adaptive on-board guidance for entry vehicles. *AIAA guidance, navigation, and control conference and exhibit*. 4048.
[4] Braun, R., and Powell, R.W. 1992. Predictor-corrector guidance algorithm for use in high-energy aerobraking system studies. *Journal of Guidance, Control, and Dynamics*. 15(3):672-678.
[5] Brunner, C., and Lu, P. 2008. Skip entry trajectory planning and guidance. *Journal of Guidance, Control, and Dynamics.* 31(5):1210-1219.
[6] Brunner, C., and Lu, P. 2012. Comparison of fully numerical predictor-corrector and Apollo skip entry guidance algorithms. *The Journal of the Astronautical Sciences.* 59:517-540.
[7] Lu, P. 2008. Predictor-corrector entry guidance for low lifting vehicles. *Journal of Guidance, Control, and Dynamics.* 31(4):1067-1075.
[8] Lu, P. 2014. Entry guidance: a unified method. *Journal of Guidance, Control, and Dynamics*. 37(3):713-728.
[9] Andersen, E.D., Roos, C., and Terlaky. T. 2003. On implementing a primal-dual interior-point method for conic quadratic optimization. *Mathematical Programming*. 95:249-277.
[10] Acikmese, B., Ploen, S.R. 2007. Convex programming approach to powered descent guidance for mars landing. *Journal of Guidance, Control, and Dynamics*. 30(5):1353-1366.
[11] Chen, Y., Cutler, M. and How, J.P. 2015. Decoupled multiagent path planning via incremental sequential convex programming. In: *2015 IEEE International Conference on Robotics and Automation*. 5954-5961.
[12] Rho, H., Oh, Y.J., Tahk, M.J., Kwon, K.J., and Kwon. H.H. 2020. L1 penalized sequential convex programming for fast trajectory optimization: with application to optimal missile guidance. *International Journal of Aeronautical and Space Sciences*. 21:493-503.
[13] Augugliaro, F., Schoelling, A.P., and D'Andrea, R. 2012. Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach. In: *IEEE/RSK international conference on Intelligent Robots and Systems*. 1917-1922
[14] Wang, Z., and Grant. M.J. 2017. Constrained trajectory optimization for planetary entry via sequential convex programming. *Journal of Guidance, Control, and Dynamics*. 40(10):2603-2615.
[15] Liu, X., Shen, Z. and Lu, P. 2016. Entry trajectory optimization by second-order cone programming. *Journal of Guidance, Control and Dynamics*. 39(2):227-241.
[16] Wang, Z., Lu, Y. 2020. Improved sequential convex programming algorithms for entry trajectory optimization. *Journal of Spacecraft and Rockets*. 57(6):1373-1386.
[17] Zhou, X., He, R.Z., Zhang, H.B., Tang, G.J., Bao, W.M. 2021. Sequential convex programming method using adaptive mesh refinement for entry trajectory planning problem. *Aerospace Science and Technology*. 109(2021).
[18] Lu, P. 1997. Entry guidance and trajectory control for reusable launch vehicle. *Journal of Guidance, Control and Dynamics*. 20(1):143-149.