

TRAJECTORY OPTIMIZATION OF HIT-TO-KILL INTERCEPTORS VIA SEQUENTIAL CONVEX PROGRAMMING APPROACH

Suk-jae Im[1], Chang-hun Lee[1], Jae-hyun Jin[2] & Hyo-choong Bang[1]

[1]Korea Advanced Institute of Science and Technology, Daejeon 34141, Republic of Korea
[2]Sunchon National University, Suncheon 57922, Jeonnam, Korea

Abstract

A sequential convex programming-based trajectory optimization scheme is proposed for hit-to-kill interceptors against ballistic missiles. The interceptor trajectory is modeled using boost, glide/coast, kill-vehicle divert, and post-handover coast segments that capture phase-dependent dynamics and handover requirements. A minimum-time problem is formulated in which the durations of three post-boost phases are optimized. The nonlinear dynamics are reformulated as a control-affine representation via state promotion and epigraph lifting, enabling the construction of convex subproblems within the SCP framework. To accommodate handover conditions at the onset of seeker acquisition, predicted ballistic target motion is considered to determine the timing of the event. Numerical simulations show that the proposed method achieves flight times within 1% of a direct NLP benchmark while solving the representative cases in less than 0.2 s on a desktop CPU.

Keywords: Trajectory Optimization, Sequential convex programming

1. Introduction

Ballistic missile defense (BMD) has received sustained attention owing to the short reaction time and severe consequences associated with high-speed missile threats. To effectively neutralize incoming threats, hit-to-kill (HTK) interception, in which the interceptor directly collides with the target, is widely adopted to achieve high lethality through kinetic impact rather than explosive fragmentation. Achieving a direct hit requires guidance strategies that account for vehicle dynamics and operational constraints throughout the entire engagement sequence. Long-range HTK interceptors typically operate across multiple flight phases, including boost, midcourse (glide/coast), and kill-vehicle divert phases. Each phase is governed by distinct dynamics and control mechanisms, such as angle-of-attack control through aerodynamic surfaces or thrust vector control (TVC) during the boost phase and divert and attitude control systems (DACS) in the latter phases. Consequently, interceptor guidance is naturally formulated as a multi-phase problem.

Moreover, BMD requires the interceptor to reach a feasible engagement condition within a limited reaction window. Since a failed intercept may leave only a limited opportunity for re-engagement, early arrival at a predicted intercept or handover condition can increase engagement margin and improve the possibility of layered defense. In this sense, the minimum arrival time can serve as a reachability-oriented metric for assessing whether a candidate interception/handover scenario can be achieved within the available engagement window; a shorter minimum arrival time is also closely related to maximizing the reachable engagement region within a prescribed time window. Because the predicted intercept condition is affected by target estimation errors and may require updates during flight, the trajectory generation problem may need to be repeatedly solved as the engagement geometry changes. These characteristics of the BMD motivate a trajectory generation framework that can minimize the arrival time for a given interception point and can be repeatedly solved along the flight.

Analytical and semi-analytical guidance laws, including proportional navigation (PN) and its variants, can be readily implemented onboard and have been successfully extended to handle practical constraints such as seeker field-of-view (FOV) limits, impact angle, and impact time requirements [1, 2, 3]. Recent midcourse guidance studies for BMD interceptors have further incorporated handover-related requirements and exo-atmospheric ballistic trajectory prediction, often combining analytical guidance laws with low-dimensional numerical optimization [4, 5, 6, 7]. In HTK interception scenarios, related studies have further considered collision-geometry constraints with pulsed guidance [8], optimal guidance utilizing gravity rather than compensating it [9], and attitude-hold guidance law suitable for off-axis seekers [10]. However, these methods are usually tailored to a prescribed guidance structure, trajectory parameterization, or phase configuration, and extending them to the unified multi-phase optimization problem generally requires problem-specific reformulation. On the other hand, direct nonlinear programming (NLP) methods provide a general framework for high-fidelity trajectory optimization [11]. Nevertheless, they are often sensitive to initial guesses [12], affecting convergence behavior and computational time, which could be undesirable for repeated trajectory generation in time-sensitive guidance applications.

Motivated by these considerations, sequential convex programming (SCP) has emerged as a practical approach in which a nonconvex trajectory optimization problem is successively approximated by a sequence of convex subproblems that can be solved reliably using mature convex optimization algorithms [13]. Over the past decade, successive convexification techniques have been successfully applied to a wide range of aerospace guidance problems, including powered planetary landing [14, 15], launch vehicle ascent [16, 17], trajectory optimizations for various types of missiles [18, 19], atmospheric entry [20, 21], and spacecraft rendezvous and proximity operations [22, 23]. These studies collectively demonstrate the potential of SCP to generate feasible high-quality trajectories with favorable computational performance for guidance applications.

To this end, this paper investigates a sequential convex programming approach as a structured direct-method framework for multi-phase hit-to-kill interceptor missile trajectory generation. The multi-phase formulation allows the four flight phases–boost, glide/coast, kill-vehicle divert, and unforced post-handover segments to be integrated into a single minimum-time optimization problem. This structure lets the handover event be represented explicitly along the trajectory, thereby providing a reference timing for seeker activation and related events, such as shroud separation. Interval-averaged duty ratio is implemented considering the pulsed nature of DCS thrust, and a magnitude-lifting formulation is introduced to obtain a control-affine convex subproblem. Numerical simulations demonstrate that the proposed approach generates feasible multi-phase trajectories with flight times comparable to those obtained by a direct NLP benchmark, while showing potential in terms of computational time for guidance-level trajectory generation.

2. Optimal Control Problem Formulation

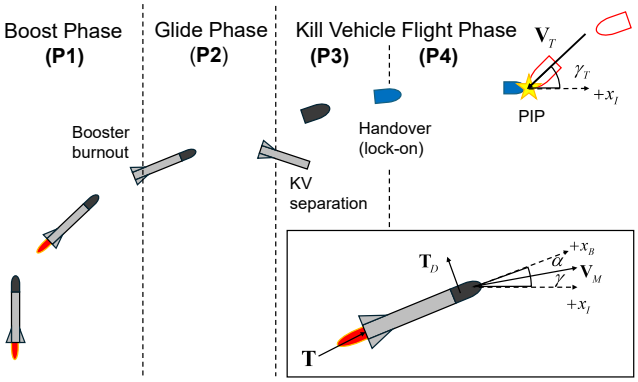


Figure 1 – Hit-to-kill interception problem depiction.

In this section, the trajectory optimization problem is formulated using the original nonlinear dynamics and path constraints as a reference for the convex optimization approach.

2.1 Vehicle Dynamics and Constraints

Assuming planar engagement, the interceptor missile goes through four flight phases: boost, glide/coast, kill-vehicle divert phase before handover, and post-handover coast phase to the PIP. These segments are denoted as phases 1–4, respectively. The last two phases are collectively termed kill vehicle flight phase. State and control inputs are defined for each phase with the corresponding governing equations. Fig. 1 depicts the missile dynamic models for the entire flight phases with the events that trigger transitions between phases. x and h denote downrange and altitude of the interceptor, respectively. V stands for the speed of the missile, γ the flight path angle, M the mach number, and α the body angle-of-attack. For representing aerodynamic forces, D and L indicate drag and lift, respectively. Denoting the dynamic pressure $Q = 0.5\rho V^2$, reference area S_{ref}, C_D drag coefficient and C_L lift coefficient, $D = QS_{\text{ref}}C_D$ and $L = QS_{\text{ref}}C_L$. The lift and drag coefficients are modeled as a drag polar form

$$\begin{aligned} C_L &= C_{L,\alpha}(M)\alpha \\ C_D &= C_{D,0}(M) + K(M)C_L^2 \end{aligned} \tag{1}$$

Regarding missile mass and thrust parameters, m denotes the total mass of the interceptor and m_{KV} is used to denote specifically the KV mass. T is the thrust of the missile in the boost phase, while T_D denotes the DCS thrust of the KV. Then, the equations of motion of the interceptor throughout the engagement can be written as

$$\begin{aligned} \dot{x} &= V\cos\gamma \\ \dot{h} &= V\sin\gamma \\ \dot{V} &= \frac{T\cos\alpha - T_D\sin\alpha - D}{m} - g\sin\gamma \\ \dot{\gamma} &= \frac{T\sin\alpha + T_D\cos\alpha + L}{mV} - \frac{g\cos\gamma}{V} \\ \dot{m}_{\text{KV}} &= -\frac{|T_D|}{I_{\text{sp,KV}}g_0}. \end{aligned} \tag{2}$$

2.1.1 Boost Phase Dynamics and Constraints

In the boost phase (phase 1), the interceptor maneuvers via controlling the body angle-of-attack α, while the DCS thruster is not used in the boost phase, so $T_D = 0$ in Eq.(2). States are defined as $\mathbf{x}_{\text{P1}} = [x,\ h,\ V,\ \gamma; \alpha]^\top$ and control input $\mathbf{u}_{\text{P1}} = \dot{\alpha}$. The thrust magnitude is assumed constant as T_{boost} during a predefined burnout time t_{bo}. During phase 1, the thrust and mass are given by

$$\begin{aligned} T(t) &= T_{\text{boost}}, \\ m(t) &= m_0 - \dot{m}t, \end{aligned} \qquad 0 \le t \le t_{\text{bo}}. \tag{3}$$

where $\dot{m} = \frac{T_{\text{boost}}}{I_{\text{sp}}g_0}$ is the constant booster propellant mass-flow rate. After burnout, $T = 0$, and the phase-dependent mass definitions are used. The constraints in the boost phase are initial conditions, angle-of-attack limit, and its rate limit. The initial conditions are imposed on initial position and velocity of the interceptor, while the angle of attack is treated as a free variable.

$$[x,\ h,\ V,\ \gamma]|_{t=0} = \mathbf{x}_0 \tag{4}$$

where $\mathbf{x}_0 = [x_0,\ h_0,\ V_0,\ \gamma_0]^\top$. To ensure a trajectory respecting the maneuverability and the structural load limit of the interceptor, the magnitude of the angle of attack α is limited to be less or equal to the maximum allowed limit, $\alpha_{\max}$. Also, its rate is constrained under $\dot{\alpha}_{\max}$.

$$\begin{aligned} -\alpha_{\max} \le \alpha \le \alpha_{\max} \\ -\dot{\alpha}_{\max} \le \dot{\alpha} \le \dot{\alpha}_{\max}. \end{aligned} \tag{5}$$

2.1.2 Coast/Glide Phase Dynamics and Constraints

The glide phase (phase 2) starts after the booster burnout until the separation of the KV. Due to the aerodynamic instability of the KV, the empty booster must be attached until a specific separation

altitude h_{sep}. The DCS thruster is assumed to generate lateral acceleration to the combined body, while the aerodynamic force induced by attitude change is neglected due to large static stability. Then, the state and control input can be defined as $\mathbf{x}_{\text{P2}} = [x,\, h,\, V,\, \gamma,\, m_{\text{KV}}]^\top$ and $\mathbf{u}_{\text{P2}} = T_D$, respectively. Note that the missile's total mass m is related to the KV mass m_{KV} as $m = m_{\text{KV}} + m_{\text{BS}} + m_{\text{shroud}}$, where m_{BS} is the retained booster assembly mass after burnout and m_{shroud} is the shroud mass. It is assumed that the angle of attack α is kept zero during the glide phase. In the gliding phase, the empty booster is kept attached to the KV for airframe stabilization, i.e. being statically stable. With large dynamic pressure shortly after burnout, the restoring moment is so large that DCS thrust alone is not sufficient to rotate the body large enough to generate meaningful aerodynamic lift. Therefore, for guidance purposes, maneuver via angle of attack in this phase is neglected. The path constraints for the second phase are for maximum DCS thrust limit and KV fuel limit

$$-T_{D,\max} \le T_D \le T_{D,\max} \tag{6}$$

$$m_{\text{KV,dry}} + m_{f,\text{hom}} \le m_{\text{KV}} \le m_{\text{KV},0} \tag{7}$$

where $m_{\text{KV,dry}}$ is the KV dry mass, $m_{f,\text{hom}}$ is the reserved KV fuel for homing phase, and $m_{\text{KV},0}$ is the fully loaded KV weight. Thus the KV propellant allowed for the entire midcourse guidance is $m_{\text{KV},0} - (m_{\text{KV,dry}} + m_{f,\text{hom}})$.

2.1.3 Kill Vehicle Flight Phase Dynamics and Constraints

On reaching the booster-KV separation altitude h_{sep}, the kill-vehicle flight phase begins by jettisoning the empty booster so that KV alone can reach PIP. Before the KV-target distance reaches the seeker acquisition range R_{seeker}, the KV should eject the shroud protecting the seeker from aero-thermal effects. For guidance-level robustness, two assumptions are made. First, the shroud ejection event is modeled to happen at the onset of the handover event so that the effective KV acceleration is not over-estimated. Second, after the handover, the KV is assumed to coast without any maneuver in the trajectory generation scheme, so as not to interfere with the terminal/homing guidance acceleration, i.e. $T_D = 0$. With Eq.(6), constraints for the last phase are about acquisition range of the seeker and the terminal constraint.

Let $\eta_2 > 0$, $\eta_3 > 0$, and $\eta_4 > 0$ denote the decision variable corresponding to the duration of the second, third, and final phase, respectively. Then, $t_f = t_{\text{bo}} + \eta_2 + \eta_3 + \eta_4$. As depicted in Fig.1, let the speed and flight path angle of the target missile be V_T and $\pi + \gamma_T$, so γ_T denotes the terminal incidence angle magnitude. Denoting time-to-go before impact $t_{\text{go}} = t_f - t$ and applying the ballistic flight assumption, the estimated position of the ballistic target can be approximated as

$$\begin{aligned} x_T(t) &= x_{\text{PIP}} + V_T \cos(\gamma_T) t_{\text{go}} \\ h_T(t) &= h_{\text{PIP}} + V_T \sin(\gamma_T) t_{\text{go}} - \frac{g}{2} t_{\text{go}}^2. \end{aligned} \tag{8}$$

For simplicity, the engagement is assumed to occur on a common vertical plane; for a non-maneuvering ballistic target, extension to a three-dimensional engagement would require the target motion in its estimated ballistic plane. Moreover, it is assumed that the attitude control loop of DACS aligns seeker towards the target at the terminal phase. Consequently, the acquisition condition is reduced to a range-based handover model in this study.

As $\eta_4 = t_{\text{go}}$ at the handover event, the lock-on constraint can be modeled as

$$\| [x(t_f - \eta_4) - x_T(t_f - \eta_4),\, h(t_f - \eta_4) - h_T(t_f - \eta_4)] \|_2 = R_{\text{seeker}}. \tag{9}$$

The interceptor must collide with the target with zero miss distance. Denoting the downrange and the altitude at the terminus x_f and h_f, respectively,

$$[x_f,\, h_f]^\top = [x_{\text{PIP}},\, h_{\text{PIP}}]^\top. \tag{10}$$

2.1.4 Inter-phase Connectivity Constraints

At the transition between boost-glide phase and glide-homing phase, the state variables must match to ensure physically feasible trajectory generation. Therefore, denoting subscripts '0' and 'f' as the initial and final condition of a phase, the following conditions must hold:

$$\begin{aligned}
[x,\, h,\, V,\, \gamma]^\top_{f,\mathrm{P1}} &= [x,\, h,\, V,\, \gamma]^\top_{0,\mathrm{P2}} \\
\alpha_{f,\mathrm{P1}} &= 0 \\
m_{\mathrm{KV},0,\mathrm{P2}} &= m_{\mathrm{KV},0} \\
h_{f,\mathrm{P2}} &= h_{\mathrm{sep}} \\
[x,\, h,\, V,\, \gamma\,, m_{\mathrm{KV}}]^\top_{f,\mathrm{P2}} &= [x,\, h,\, V,\, \gamma,\, m_{\mathrm{KV}}]^\top_{0,\mathrm{P3}} \\
[x,\, h,\, V,\, \gamma\,, m_{\mathrm{KV}}]^\top_{f,\mathrm{P3}} &= [x,\, h,\, V,\, \gamma,\, m_{\mathrm{KV}}]^\top_{0,\mathrm{P4}}.
\end{aligned} \tag{11}$$

Notice that m_{KV} excludes the shroud mass; however, the mass m used in Eq.(2) for translational dynamics is phase-dependent: $m = m_{\mathrm{KV}} + m_{\mathrm{shroud}}$ in phase 3, but $m = m_{\mathrm{KV}}$ in phase 4. Collecting together the dynamics, constraints, and the minimum-time objective, the original optimal control problem **OCP1** can be formulated as follows:

$$\begin{aligned}
\textbf{(OCP1)} \quad & \min_{\mathbf{X},\mathbf{u},\eta} \quad t_f \\
& \text{subject to} \quad \text{Eqs.} \quad (2)-(11).
\end{aligned} \tag{12}$$

3. Convex Programming Reformulation

OCP1 is a non-convex problem due to non-convex dynamics and constraints. In this section, nonconvexities in the original problem **OCP1** are analyzed, and proper convex reformulation methodologies are discussed.

3.1 Convexification of Dynamics and Constraints

One basic method to convert dynamics expressed in differential equation is to linearize and discretize to a convex, i.e., linear equality constraint. In such procedure, it is well known that linearizing with respect to both state and control is prone to control input oscillation behavior, and partial linearization about state for control-affine dynamics can relieve the issue [24]. Moreover, mass consumption model of the KV incorporates $|T_D|$, which requires appropriate transformations. To this end, reformulation of the dynamics in Eq.(2) suitable for convex programming is introduced for each phase.

3.1.1 Boost Phase Dynamics

To formulate a control-affine dynamics for partial linearization, a new control input is defined as the angle-of-attack rate, $\dot{\alpha}$. Then, the modified dynamics for the first phase can be written as

$$\frac{d}{dt}\begin{bmatrix} x \\ h \\ V \\ \gamma \\ \alpha \end{bmatrix} \triangleq \mathbf{f}_1(\mathbf{x}_{\mathrm{P1}}) + \mathbf{B}_{\mathrm{P1}}(\mathbf{x}_{\mathrm{P1}})\mathbf{u}_{\mathrm{P1}} = \begin{bmatrix} V\cos\gamma \\ V\sin\gamma \\ \dfrac{T\cos\alpha - D}{m} - g\sin\gamma \\ \dfrac{T\sin\alpha + L}{mV} - \dfrac{g\cos\gamma}{V} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \dot{\alpha}. \tag{13}$$

The second process for convexification is linearization and discretization. Linearization is performed only with respect to the state $\mathbf{x}_{\mathrm{P1}}$. In the successive convexification framework, the reference point for linearization is the solution obtained by the previous iteration, which is denoted as $\bar{\mathbf{x}}_{\mathrm{P1}}$. Denoting $\mathbf{A}_{\mathrm{P1}} \triangleq \frac{\partial \mathbf{f}_1(\mathbf{x}_{\mathrm{P1}})}{\partial \mathbf{x}_{\mathrm{P1}}}\big|_{\mathbf{x}_{\mathrm{P1}}=\bar{\mathbf{x}}_{\mathrm{P1}}}$, the linearized boost phase dynamics is

$$\frac{d}{dt}\begin{bmatrix} x \\ h \\ V \\ \gamma \\ \alpha \end{bmatrix} = \mathbf{f}_1(\bar{\mathbf{x}}_{\mathrm{P1}}) + \mathbf{A}_{\mathrm{P1}}(\bar{\mathbf{x}}_{\mathrm{P1}})(\mathbf{x}_{\mathrm{P1}} - \bar{\mathbf{x}}_{\mathrm{P1}}) + \mathbf{B}_{\mathrm{P1}}(\bar{\mathbf{x}}_{\mathrm{P1}})\mathbf{u}_{\mathrm{P1}} \tag{14}$$

where

$$\mathbf{A}_{\mathrm{P1}}(\bar{\mathbf{x}}_{\mathrm{P1}}) = \begin{bmatrix} 0 & 0 & \cos\gamma & -V\sin\gamma & 0 \\ 0 & 0 & \sin\gamma & V\cos\gamma & 0 \\ 0 & -\dfrac{D_h}{m} & -\dfrac{D_V}{m} & -g\cos\gamma & -\dfrac{T\sin\alpha + D_\alpha}{m} \\ 0 & \dfrac{L_h}{mV} & -\dfrac{T\sin\alpha + L}{mV^2} + \dfrac{L_V}{mV} + \dfrac{g\cos\gamma}{V^2} & \dfrac{g\sin\gamma}{V} & \dfrac{T\cos\alpha + L_\alpha}{mV} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{15}$$

Here, $D_h = \partial D/\partial h$, $D_V = \partial D/\partial V$, $D_\alpha = \partial D/\partial \alpha$, which are evaluated at the reference trajectory. The lift derivatives are defined likewise, while $g_h = \partial g/\partial h$ is neglected. To discretize Eq.(14), trapezoidal integration is used. Moreover, in order to cope with the artificial infeasibility possibly triggered by the linearization, virtual control term $\nu = \nu_1 - \nu_2$, where $\nu_1, \nu_2 \geq \mathbf{0}$, is introduced. Given k-th and $k+1$-th node with time step Δt, the discretized equations of motion expressed as a linear equality constraint is derived as

$$\left(\mathbf{I} - \frac{\Delta t}{2}\mathbf{A}_{\mathrm{P1},k+1}\right)\mathbf{x}_{\mathrm{P1,k+1}} = \left(\mathbf{I} + \frac{\Delta t}{2}\mathbf{A}_{\mathrm{P1},k}\right)\mathbf{x}_{\mathrm{P1,k}} + \frac{\Delta t}{2}\left(\mathbf{B}_{\mathrm{P1,k}}\mathbf{u}_k + \mathbf{B}_{\mathrm{P1,k+1}}\mathbf{u}_{k+1} + \mathbf{C}_k + \mathbf{C}_{k+1}\right) + (\nu_{1,k} - \nu_{2,k}) \tag{16}$$

where $\mathbf{I}$ is an identity matrix of appropriate size, and $\mathbf{C} = \mathbf{f}_1(\bar{\mathbf{x}}_{\mathrm{P1}}) - \mathbf{A}_{\mathrm{P1}}\bar{\mathbf{x}}_{\mathrm{P1}}$.

3.1.2 Coast/Glide and Kill Vehicle Flight Phase Dynamics

The KV DCS generates pulsed thrust approximately constant in magnitude. Therefore, pulsed guidance scheme that respects DCS characteristics should be implemented. Considering that the dynamics is discretized, the concept of duty ratio is adopted for each discretization interval. The interval-averaged DCS thrust over some time interval $[\tau_k, \tau_{k+1}]$ is expressed as $T_{D,k} = T_{D,\max}u_{1,k}$, where $-1 \leq u_1 \leq 1$ denotes duty ratio. Subsequently, the absolute value of the duty ratio is represented using an epigraph variable. Denoting the epigraph variable u_2, the control input of phase 2–4 can be augmented as $\mathbf{u}_{\mathrm{Pi}} \triangleq [u_1,\ u_2]^\top$, where $0 \leq u_2 \leq 1$ satisfies

$$|u_1| \leq u_2 \equiv -u_2 \leq u_1 \leq u_2. \tag{17}$$

Notice that $u_2 = |u_1|$ is required for physical consistency. To promote the tightness, small KV terminal mass reward is added to the objective:

$$-w_{\mathrm{KV}}m_{\mathrm{KV}}(t_f) \tag{18}$$

where $w_{\mathrm{KV}} \geq 0$. This term is equivalent to penalizing integration of the DCS fuel used over time, which is again proportional to the integration of u_2. For compactness, phases 2–4 are written using single dynamics, while DCS control is available only in phases 2 and 3. The index set for phase 2–4 is $i \in I_D = \{2,3,4\}$, and $\mathbf{P_i}$ corresponds to the $i \in I_D$-th phase. The post-handover coast segment indicated by $i = 4$ does not use DCS as control thus $u_{1,\mathrm{P4}} = u_{2,\mathrm{P4}} = 0$ and $\dot{m}_{\mathrm{KV}} = 0$. Then, the control-affine formulation of glide and terminal phase is

$$\frac{d}{dt}\begin{bmatrix} x \\ h \\ V \\ \gamma \\ m_{\mathrm{KV}} \end{bmatrix} \triangleq \mathbf{f}_i(\mathbf{x}_{\mathrm{P_i}}) + \mathbf{B}_{\mathrm{P_i}}(\mathbf{x}_{\mathrm{P_i}})\mathbf{u}_{\mathrm{P_i}} = \begin{bmatrix} V\cos\gamma \\ V\sin\gamma \\ -\dfrac{D}{m} - g\sin\gamma \\ \dfrac{L}{mV} - \dfrac{g\cos\gamma}{V} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \dfrac{T_{\mathrm{D,max}}}{mV} & 0 \\ 0 & -\dfrac{T_{\mathrm{D,max}}}{I_{\mathrm{sp,KV}}g_0} \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \tag{19}$$

To linearize Eq.(19), denote $\mathbf{A}_{\mathrm{P_i}} \triangleq \frac{\partial \mathbf{f}(\mathbf{x}_{\mathrm{P_i}})}{\partial \mathbf{x}_{\mathrm{P_i}}}|_{\mathbf{x}_{\mathrm{P_i}} = \bar{\mathbf{x}}_{\mathrm{P_i}}}$. The linearized dynamics for the glide and terminal

phases is

$$\begin{aligned}\frac{d}{dt}\begin{bmatrix} x \\ h \\ V \\ \gamma \\ m_{\mathrm{KV}} \end{bmatrix} &= \mathbf{f}_i(\bar{\mathbf{x}}_{\mathrm{P_i}}) + \mathbf{A}_{\mathrm{P_i}}(\mathbf{x}_{\mathrm{P_i}} - \bar{\mathbf{x}}_{\mathrm{P_i}}) + \mathbf{B}_{\mathrm{P_i}}(\bar{\mathbf{x}}_{\mathrm{P_i}})\mathbf{u}_{\mathrm{P_i}} \\ &= \mathbf{f}_i(\bar{\mathbf{x}}_{\mathrm{P_i}}) + \begin{bmatrix} 0 & 0 & \cos\gamma & -V\sin\gamma & 0 \\ 0 & 0 & \sin\gamma & V\cos\gamma & 0 \\ 0 & -\dfrac{D_h}{m} & -\dfrac{D_V}{m} & -g\cos\gamma & \dfrac{D}{m^2} \\ 0 & \dfrac{L_h}{mV} & \dfrac{L_V}{mV} - \dfrac{L}{mV^2} + \dfrac{g\cos\gamma}{V^2} & \dfrac{g\sin\gamma}{V} & -\dfrac{L}{m^2V} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} (\mathbf{x}_{\mathrm{P_i}} - \bar{\mathbf{x}}_{\mathrm{P_i}}) \\ &\quad + \mathbf{B}_{\mathrm{P_i}}(\bar{\mathbf{x}}_{\mathrm{P_i}})\mathbf{u}_{\mathrm{P_i}}. \end{aligned} \tag{20}$$

The characteristics of DCS thrust and the effect of free final time are considered on the discretization of Eq.(20). First, the physical time for each phase t is normalized to $t = \eta_i \tau$, where $\tau \in [0,1]$. Then, $\dfrac{d}{dt} = \dfrac{1}{\eta_i}\dfrac{d}{d\tau}$. Second, as the DCS emits pulsed thrusts for maneuver, the control input is better defined when interpreted as an interval-averaged duty ratio of the DCS. For these, the discretized dynamics is derived as

$$\begin{aligned}\left(\mathbf{I} - \frac{\Delta\tau}{2}\bar{\eta}_i\mathbf{A}_{\mathrm{P_i},k+1}\right)\mathbf{x}_{\mathrm{P_i},k+1} &= \left(\mathbf{I} + \frac{\Delta\tau}{2}\bar{\eta}_i\mathbf{A}_{\mathrm{P_i},k}\right)\mathbf{x}_{\mathrm{P_i},k} \\ &\quad + \frac{\Delta\tau}{2}\Big(\bar{\eta}_i\mathbf{B}_{\mathrm{P_i},k}\mathbf{u}_{\mathrm{P_i},k} + \bar{\eta}_i\mathbf{B}_{\mathrm{P_i},k+1}\mathbf{u}_{\mathrm{P_i},k} \\ &\qquad + \eta_i(\mathbf{f}_i(\bar{\mathbf{x}}_{\mathrm{P_i},k}) + \mathbf{B}_{\mathrm{P_i},k}\bar{\mathbf{u}}_{\mathrm{P_i},k}) + \eta_i(\mathbf{f}_i(\bar{\mathbf{x}}_{\mathrm{P_i},k+1}) + \mathbf{B}_{\mathrm{P_i},k+1}\bar{\mathbf{u}}_{\mathrm{P_i},k}) + \mathbf{E}_{i,k} + \mathbf{E}_{i,k+1}\Big) \\ &\quad + (\nu_{1,k} - \nu_{2,k}). \end{aligned} \tag{21}$$

where $\mathbf{A}_{\mathrm{P}_i,k} = \left.\frac{\partial \mathbf{f}_i(\mathbf{x}_{\mathrm{P_i}})}{\partial \mathbf{x}_{\mathrm{P_i}}}\right|_{\mathbf{x}_{\mathrm{P_i}} = \bar{\mathbf{x}}_{\mathrm{P}_i,k}}$, $\mathbf{E}_{i,k} = -\bar{\eta}_i\,(\mathbf{A}_{\mathrm{P}_i,k}\bar{\mathbf{x}}_{\mathrm{P}_i,k} + \mathbf{B}_{\mathrm{P}_i,k}\bar{\mathbf{u}}_{\mathrm{P}_i,k})$, $\mathbf{E}_{i,k+1} = -\bar{\eta}_i\,(\mathbf{A}_{\mathrm{P}_i,k+1}\bar{\mathbf{x}}_{\mathrm{P}_i,k+1} + \mathbf{B}_{\mathrm{P}_i,k+1}\bar{\mathbf{u}}_{\mathrm{P}_i,k})$, and $\mathbf{I}$ is an identity matrix of appropriate size. The subscript (i,k) or $(\mathrm{P_i},k)$ indicates the k-th node at i-th phase, but for control $\mathbf{u}_{\mathrm{P}_i,k}$, k denotes the k-th interval, i.e. the interval between k and $k+1$-th node.

3.1.3 Handover/Lock-on Constraint

The seeker lock-on range constraint expressed in Eq.(9) is non-convex as range expression is quadratic in the positions of the interceptor and the target missiles and the target missile position is again dependent on the duration of the phase 4. To accurately represent the lock-on distance while allowing good convergence property, virtual variable $0 \le \hat{R}_{\mathrm{lock}} \le R_{\mathrm{seeker}}$ is introduced to represent the linearized distance equality

$$\hat{R}_{\mathrm{lock}} = \bar{R} + \frac{\partial R}{\partial x}(x - \bar{x}) + \frac{\partial R}{\partial h}(h - \bar{h}) + \frac{\partial R}{\partial \eta}(\eta - \bar{\eta}) + \nu_1 - \nu_2 \tag{22}$$

where $\dfrac{\partial R}{\partial \eta} = \dfrac{\partial R}{\partial x_T}\dfrac{\partial x_T}{\partial \eta} + \dfrac{\partial R}{\partial h_T}\dfrac{\partial h_T}{\partial \eta}$. Then, relaxed equality constraint is imposed as

$$\hat{R}_{\mathrm{lock}} + s_{\mathrm{lock}} = R_{\mathrm{seeker}} \tag{23}$$

where penalization on $s_{\mathrm{lock}} \ge 0$ lets $\hat{R}_{\mathrm{lock}}$ close to R_{seeker} without necessitating strict equality. s_{lock} is linearly penalized as

$$w_{\mathrm{lock}} s_{\mathrm{lock}}. \tag{24}$$

3.2 Sequential Convex Programming Approach

3.2.1 Trust Region

As discussed, the artificial infeasibility is present as the linearized constraints are sufficiently accurate only near the reference point, which is usually the solution of the previous iteration. For the same reason, linearization induces artificial unboundedness [15], where the linearization lets the solution decrease (in the minimization problem) infinitely. To handle this, decision variables that were included in the linearization are regulated not to change its value compared to the reference. By virtue of control-affine formulation of the dynamics, only the variations on states can be regulated, which is formulated as

$$\|\mathbf{x} - \bar{\mathbf{x}}_k\|_2 \leq \delta_k \tag{25}$$

where δ_k is the trust region corresponding to k-th node that controls the allowed amount of deviation of the decision variables from each reference point. On convergence, trust region is redundant, as if the optimum is the reference point, no further progress needs to be made. In this spirit, adaptive hard trust region approach [25], wherein the trust region is controlled by the ratio between expected and true merit function improvement, and soft trust region approach [26] where the magnitude of the trust region is penalized, were devised. In this paper, the latter is chosen, as it can adaptively control the trust region without any additional evaluations. Thus, an additional trust region penalization term is added to the cost function:

$$w_{\mathrm{TR}} \sum_{k=1}^{N} \delta_k. \tag{26}$$

3.2.2 Sub-problem Formulation

Based on convexification and corresponding penalization schemes, the modified convex problem **OCP2** is solved iteratively until convergence. At each subproblem, **OCP2** is formulated as a second-order cone program (SOCP) and can be efficiently solved using an interior-point solver. The previous solution becomes the reference for the next iteration in terms of convexification.

$$\begin{aligned} \textbf{(OCP2)} \quad & \min_{\mathbf{X},\mathbf{u},\eta,\nu,\delta} \quad \eta_2 + \eta_3 + \eta_4 - w_{\mathrm{KV}} m_{\mathrm{KV}}(t_f) + w_{\mathrm{lock}} s_{\mathrm{lock}} + \sum_k w_\nu \|\nu_k\|_1 + w_{\mathrm{TR}} \sum_k \delta_k \\ & \text{subject to} \quad \text{Eqs. (4), (5), (7),} \\ & \qquad\qquad\quad \text{(10)–(11), (16)–(17),} \\ & \qquad\qquad\quad \text{(21)–(23), (25).} \end{aligned} \tag{27}$$

For practicality, a simplified stopping criterion

$$\max_{k\in 1,\ldots,N} |\mathbf{x}_k - \bar{\mathbf{x}}_k| \leq \varepsilon_X \tag{28}$$

is used to detect subproblem convergence, where ε_X is a predefined stopping criterion for each state variable. The sequential convex programming algorithm for solving **OCP2** repeatedly is described in Algorithm 1.

Algorithm 1: Soft Trust-Region Sequential Convex Programming

Input: Initial reference trajectory $\mathbf{Z}_{\mathrm{ref}}^{(0)}$; tolerances ε_X; penalty weights $w_{\mathrm{KV}}, w_{\mathrm{lock}}, w_\nu, w_{\mathrm{TR}}$;
Output: Optimized trajectory $\mathbf{Z}^\star$
Initialize $\mathbf{Z}_{\mathrm{ref}} \leftarrow \mathbf{Z}_{\mathrm{ref}}^{(0)}$, $j \leftarrow 0$ $\Delta_X \leftarrow \infty$;
while $\Delta_X > \varepsilon_X$ **do**
  Linearize the nonlinear dynamics and nonconvex constraints about $\mathbf{Z}_{\mathrm{ref}}$;
  Formulate and solve the convex subproblem in Eq. (27), yielding $\mathbf{Z}^\star$;
  Compute the SCP convergence error $\Delta_X = \|X(\mathbf{Z}^\star) - X(\mathbf{Z}_{\mathrm{ref}})\|_\infty$;
  Update the reference trajectory $\mathbf{Z}_{\mathrm{ref}} \leftarrow \mathbf{Z}^\star$;
  $j \leftarrow j + 1$;

4. Numerical Simulation

4.1 Simulation Setup and Case Definition

This section provides numerical simulation results to verify the feasibility of the proposed methodology. For solving convex optimization problem **OCP2** and direct nonlinear problem, MOSEK[27] and GPOPS-II[28] are used respectively. For benchmarking, GPOPS-II solves the direct NLP counterpart using the angle-of-attack promotion as a state with Eq.(5), lifted DCS control as in Eq.(17), and terminal mass-regularization. The remaining dynamics and handover constraints in **OCP1** are imposed in their original nonlinear forms.

The simulation parameters for the simulations are described in Table 1. For phases 3 and 4, the effective drag area $S_{\text{ref}}C_D$ is taken as $1/3$ of the full interceptor (KV+booster stack). All cases were tested using Ryzen R9-7900 CPU at 3.7GHz and computation times were reported by averaging 20 times of solving the same problem after 5 times of warm-up.

The initial state and angle of attack of the interceptor are $\mathbf{x}_0 = [0\text{km},\ 0\text{km},\ 10\text{m/s},\ 90\text{deg}]^\top$ and α_0 is left free. The KV separation altitude is set as $h_{\text{sep}} = 20$ km, and the tolerance ε for decision variables is $\varepsilon = [\varepsilon_x,\ \varepsilon_h,\ \varepsilon_V,\ \varepsilon_\gamma,\ \varepsilon_{m_{\text{KV}}},\ \varepsilon_\alpha] = [5\text{ m},\ 5\text{ m},\ 1\text{ m/s},\ 0.1^\circ,\ 0.01\text{ kg},\ 0.1^\circ]$. The P1 initial guess assumes the speed to linearly increase from V_0 to $V_{\text{bo}} = 2000$ m/s, with position linearly interpolated from 0 to $0.5(V_0 + V_{\text{bo}})t_{\text{bo}}$. The flight path angle is kept constant as γ_0, and the angle-of-attack is assumed constant as 0. For glide and terminal phase, linear interpolation from the booster burnout point to the PIP is assumed for position. Speed is assumed constant as V_{bo}, and the flight path angle is assumed to linearly vary from γ_0 to $\arctan\frac{h_{\text{PIP}}-h_{\text{bo}}}{x_{\text{PIP}}-x_{\text{bo}}}$, respectively. DCS thrust is assumed 0 and $m_{\text{KV}} = m_{\text{KV},0}$.

All variables are scaled to lie approximately within $[0,1]$ or $[-1,1]$ for better convergence behavior. For state variables, downrange and altitude are normalized by the downrange of the PIP, speed by $V_{\text{bo}} = 2000\text{m/s}$, flight path angle by $\pi/2$, and KV mass by $m_{\text{KV},0} + m_{\text{shroud}}$. For control variables, angle of attack is scaled by $\alpha_{\max}$, angle of attack rate by $\dot{\alpha}_{\max}$, and DCS thrust by $T_{D,\max}$. The time variables are scaled by 50s.

To assess the proposed approach, two PIPs are selected: $[113.137,\ 40]$ km and $[60,\ 80]$ km. The former is the case with long range(≈ 120km) and low elevation($\approx 20^\circ$), requiring heavy divert in midcourse guidance; the latter case has higher elevation, requiring less midcourse maneuver.

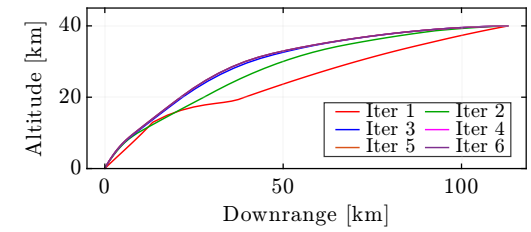
Table 1 – Simulation parameters (rounded off to the second decimal place)

Parameter	Value	Parameter	Value
Booster thrust T_{boost}	79.39 kN	Booster specific impulse I_{sp}	230 s
Booster burnout time t_{bo}	12.47 s	Interceptor initial mass m_0	590.58 kg
Booster assembly mass m_{BS}	96.78 kg	Shroud mass m_{shroud}	9.73 kg
AoA limit $\alpha_{\max}$	10°	AoA rate limit $\dot{\alpha}_{\max}$	$4^\circ/\text{s}$
DCS thrust limit $T_{D,\max}$	2.33 kN	KV specific impulse $I_{\text{sp,KV}}$	250 s
KV initial mass $m_{\text{KV},0}$	45.44 kg	Minimum KV mass $m_{\text{KV,dry}} + m_{f,\text{hom}}$	31.44 kg
Seeker acquisition range R_{seeker}	50 km	Number of nodes (N_1, N_2, N_3, N_4)	$(20, 10, 20, 10)$
Constraint weight w_ν	10^2	Trust-region weight w_{TR}	10^{-2}
Kill-vehicle mass weight w_{KV}	10^{-4}	Seeker-range weight w_{lock}	10

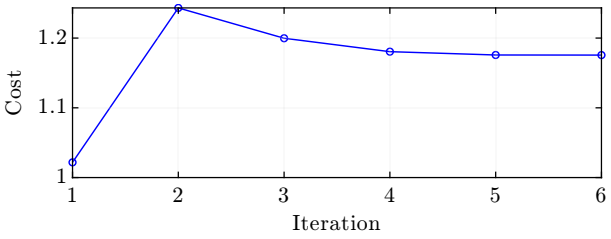
4.2 Case 1: Convergence Characteristics and NLP Benchmark

For the first case (PIP=$[113.137,\ 40]$ km), we validate the convex reformulations of constraints as well as the numerical optimality of the proposed SCP approach. For this case, it is assumed that the target missile is intercepted in PIP with $V_T = 3\text{km/s}$ and $\gamma_T = 50^\circ$. First, the convergence and computational characteristics of the proposed approach are shown. Figs. 2a–2b show that the case 1 is solved at 6 iterations after initial trajectory guess. Fig. 2c shows that the trust region and slack variables have converged to near-zero, indicating sufficient reduction of errors due to convexifications. Moreover, Fig. 2d shows that the handover condition holds at the seeker detection range.

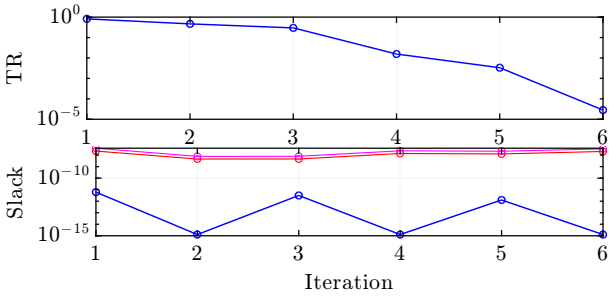
Fig. 3 shows the interceptor trajectories and the corresponding states and controls. The terminal times of the direct NLP and SCP solutions are 71.05 and 71.26 s, respectively, corresponding to a difference of 0.29%. Less than 0.2 seconds were taken to solve **OCP2** via SCP approach; these



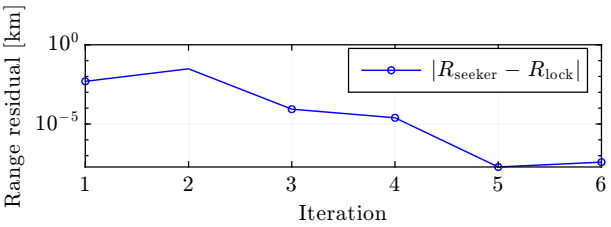
(a) Trajectory convergence.



(b) Cost function.

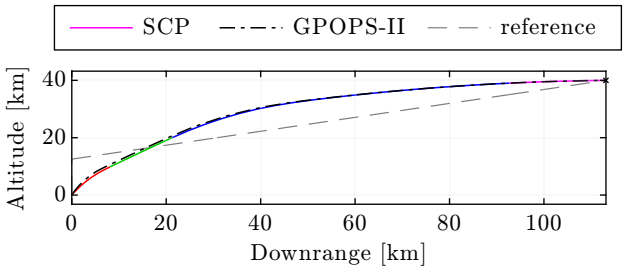


(c) Trust region and slack variables.

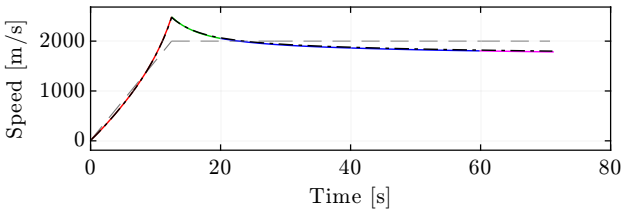


(d) Handover condition.

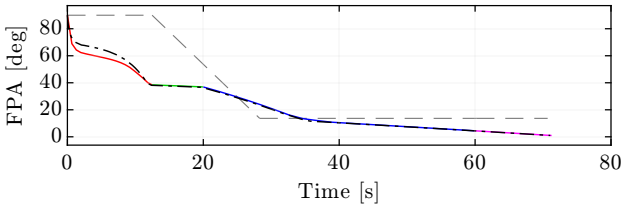
Figure 2 – Case 1: SCP solution convergence history. For slack variables, red, magenta, and blue represent ν_{lock}, s_{lock}, and virtual controls, respectively.



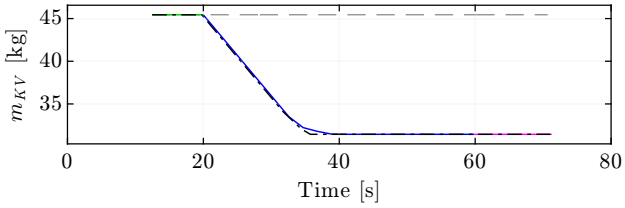
(a) Interceptor trajectory.



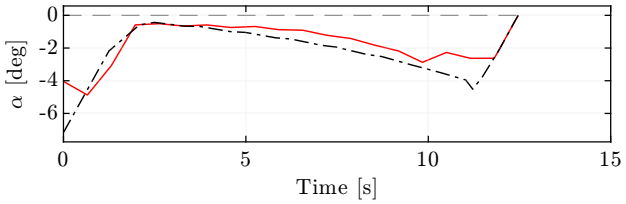
(b) Speed profile.



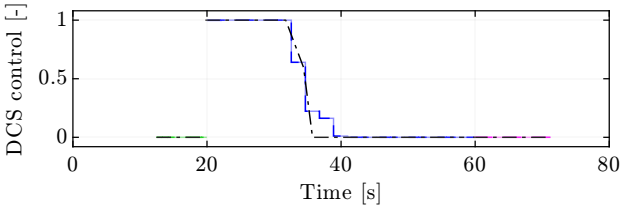
(c) Flight-path angle profile.



(d) KV mass profile.



(e) Angle-of-attack command.



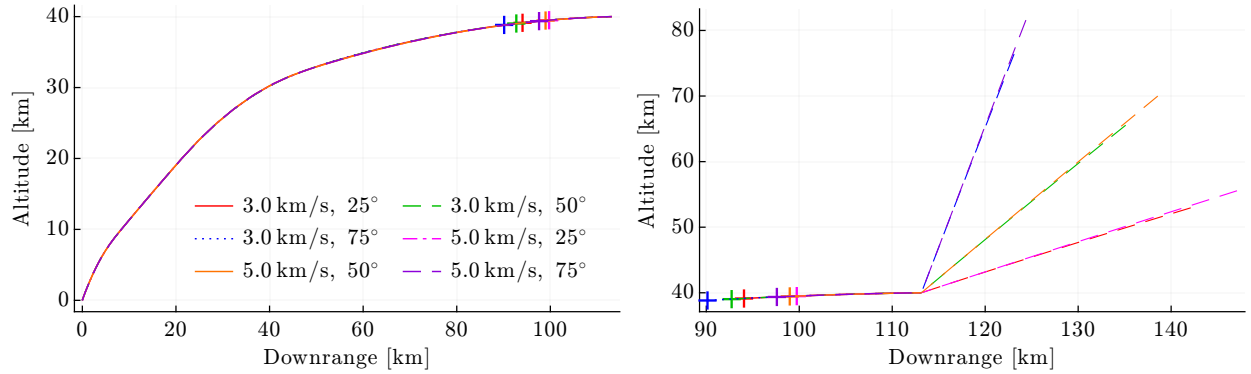
(f) DCS control input.

Figure 3 – Case 1 results: Red, green, blue, and magenta solid lines represent phases 1–4, respectively.

results show that the proposed approach preserves the minimum-time performance for this scenario, and that it has potential for fast trajectory generation. In Fig. 3f, light-colored solid lines and darker-colored dashed lines represent u_2 and $|u_1|$, respectively. It can be seen that the epigraph tightness $|u_1| = u_2$ is achieved via small KV mass regularization term in Eq.(18).

Fig.4 shows the effect of considering target motion. The speed and the terminal flight path angle of

the ballistic missiles are tested for $V_T \in \{3,5\}\mathrm{km/s}$ and $\gamma_T \in \{25,50,75\}^\circ$, each of which represents depressed, nominal, and lofted trajectories. The points at which the handover/phase-transition event occur for each case is marked using '+' marker. The interceptor trajectories provided are nearly identical as the PIP and objective function are the same. However, the handover points vary with the target terminal speed and flight path angle. This result indicates that the proposed formulation can adjust the handover/seeker activation timing according to the predicted target motion.

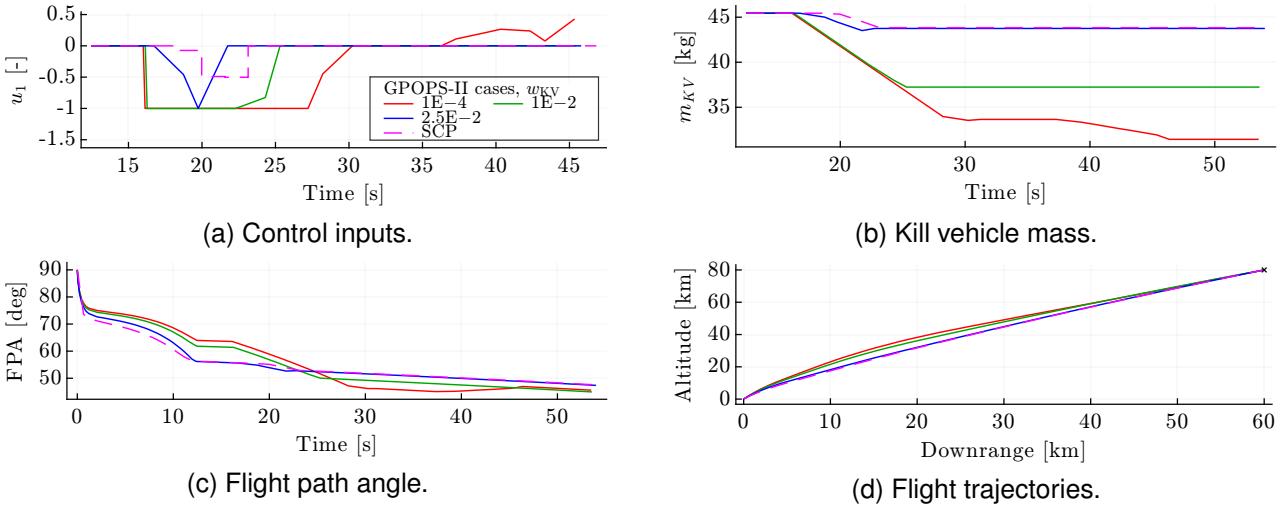


(a) Trajectories. (b) Handover geometries.

Figure 4 – Case 1: Handover geometries for various target missile conditions.

4.3 Case 2: Sensitivity to DCS Regularization

For the second case (PIP=$[60,\ 80]$ km), the target terminal conditions are set to $V_T = 5\mathrm{km/s}$ and $\gamma_T = 50^\circ$. This case required less trajectory correction compared to the case 1 due to larger PIP elevation. It took 4 iterations after initial guess to converge with less than 0.15 seconds. As can be seen from Fig. 5, the NLP solution varies depending on the magnitude of w_{KV}. While $w_{\mathrm{KV}} = 10^{-4}$ produces abrupt KV maneuver using up all available fuel, increasing w_{KV} to 10^{-2} and $2.5 \cdot 10^{-2}$ results in less abrupt midcourse maneuver while preserving substantial amount of fuel. Nevertheless, the arrival time differs from 53.51 seconds to 54.01 seconds, showing little change in objective function. It can be also noticed that the NLP with $w_{\mathrm{KV}} = 2.5 \cdot 10^{-2}$ yields similar results to the SCP, while the terminal arrival time for each case lies between 53.51 and 54.04 seconds, which is 0.99% difference. Within the two cases considered, the results suggest that for the case requiring less divert effort, propellant consumption is sensitive to the penalization coefficient, while the final time is relatively consistent from the penalization.



(a) Control inputs. (b) Kill vehicle mass.

(c) Flight path angle. (d) Flight trajectories.

Figure 5 – Case 2: Direct NLP solutions for varying w_{KV}, compared with the SCP solution.

5. Conclusions

This paper proposed a trajectory optimization scheme for hit-to-kill interceptors against ballistic missiles. Multi-phase formulation was utilized to model four distinct flight phases of the interceptor. Reformulation of dynamics and path constraints is derived so that the original nonlinear optimization problem is converted to the second-order cone programming for each subproblem, enabling sequential convex programming approach. The numerical simulations verified that the convexified problem yields comparable solutions to the original nonconvex solution in terms of minimum-time performance with computation times below one second, showing potential for repeated guidance-level trajectory generation.

6. Contact Author Email Address

Suk-jae Im: sjlmm1@kaist.ac.kr
Hyo-choong Bang (corresponding author): hcbang@kaist.ac.kr

7. Acknowledgment

This work was supported by the Theater Defense Research Center funded by the Defense Acquisition Program Administration under Grant UD240002SD. ChatGPT and OpenAI Codex were used to assist with code implementation and to check algebraic and mathematical steps. The manuscript was drafted and finalized by the authors, with LLM support for manuscript preparation limited to improving clarity and text presentation. All AI-assisted outputs were reviewed, validated, and approved by the authors. The authors retain full responsibility for all methodological choices, mathematical formulations, numerical results, interpretations, and final conclusions.

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] C.-H. Lee, T.-H. Kim, M.-J. Tahk, and I.-H. Whang. Polynomial guidance laws considering terminal impact angle and acceleration constraints. *IEEE Transactions on Aerospace and Electronic Systems*, 49(1):74–92, January 2013.

[2] S.-W. Shim, S.-M. Hong, G.-H. Moon, and M.-J. Tahk. Impact angle and time control guidance under field-of-view constraints and maneuver limits. *International Journal of Aeronautical and Space Sciences*, 19(1):217–226, March 2018.

[3] T. Han, Q. Hu, and M. Xin. Analytical solution of field-of-view limited guidance with constrained impact and capturability analysis. *Aerospace Science and Technology*, 97:105586, February 2020.

[4] Y. Jung, H. Lee, Y. Lee, Y. Kim, and J. Ahn. Composite zero-effort-miss guidance with predictor–corrector steps for ballistic interception. *Journal of Guidance, Control, and Dynamics*, 47(2):311–325, 2024.

[5] S. Zhao, W. Chen, L. Yang, and M. Wang. Optimal midcourse guidance law for the exo-atmospheric interceptor with solid-propellant booster. *Aerospace Science and Technology*, 127:107670, 2022.

[6] N. Indig, J. Z. Ben-Asher, and N. Farber. Near-optimal spatial midcourse guidance law with an angular constraint. *Journal of Guidance, Control, and Dynamics*, 37(1):214–223, 2014.

[7] S. Ann, S. Lee, Y. Kim, and J. Ahn. Midcourse guidance for exoatmospheric interception using response surface based trajectory shaping. *IEEE Transactions on Aerospace and Electronic Systems*, 56(5):3655–3673, 2020.

[8] A. Ratnoo and D. Ghose. Collision-geometry-based pulsed guidance law for exoatmospheric interception. *Journal of Guidance, Control, and Dynamics*, 32(2):669–675, 2009.

[9] S. He and C.-H. Lee. Gravity-turn-assisted optimal guidance law. *Journal of Guidance, Control, and Dynamics*, 41(1):171–183, 2018.

[10] M.-J. Tahk, E.-T. Jeong, C.-H. Lee, and S.-J. Yoon. Attitude hold proportional navigation guidance for interceptors with off-axis seeker. *Journal of Guidance, Control, and Dynamics*, 47(5):1015–1025, 2024.

[11] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1997.

[12] J. T. Betts, S. L. Campbell, and C. Digirolamo. Initial guess sensitivity in computational optimal control problems. *Numerical Algebra, Control and Optimization*, 10(1):39–43, 2020.

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

[14] M. Szmuk, T. P. Reynolds, and B. Acikmese. 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.

[15] D. Malyuta, T. P. Reynolds, M. Szmuk, T. Lew, R. Bonalli, M. Pavone, and B. Acikmese. Convex optimization for trajectory generation: a tutorial on generating dynamically feasible trajectories reliably and efficiently. *IEEE Control Systems Magazine*, 42(5):40–113, 2022.

[16] B. Benedikter, A. Zavoli, G. Colasurdo, S. Pizzurro, and E. Cavallini. Convex approach to three-dimensional launch vehicle ascent trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 44(6):1116–1131, 2021.

[17] K. Zhang, S. Yang, and F. Xiong. Rapid ascent trajectory optimization for guided rockets via sequential convex programming. *Proceedings of the Institution of Mechanical Engineers, Part G*, 233(13):4800–4809, 2019.

[18] B. Kim and C.-H. Lee. Optimal midcourse guidance for dual-pulse rocket using pseudospectral sequential convex programming. *Journal of Guidance, Control, and Dynamics*, 46(7):1425–1436, 2023.

[19] C.-G. Jung, B. Kim, K.-W. Jung, and C.-H. Lee. Thrust integrated trajectory optimization for multipulse rocket missiles using convex programming. *Journal of Spacecraft and Rockets*, 60(3):957–971, 2023.

[20] Z. Wang and Y. Lu. Improved sequential convex programming algorithms for entry trajectory optimization. *Journal of Spacecraft and Rockets*, 57(6):1373–1386, 2020.

[21] Y. Luo, J. Wang, J. Jiang, and H. Liang. Reentry trajectory planning for hypersonic vehicles via an improved sequential convex programming method. *Aerospace Science and Technology*, 149:109130, 2024.

[22] D. Zhou, Y. Zhang, and S. Li. Receding horizon guidance and control using sequential convex programming for spacecraft 6-DOF close proximity. *Aerospace Science and Technology*, 87:459–477, 2019.

[23] Y. Zhang, B. Zhu, M. Cheng, and S. Li. Trajectory optimization for spacecraft autonomous rendezvous and docking with compound state-triggered constraints. *Aerospace Science and Technology*, 127:107733, 2022.

[24] X. Liu, Z. Shen, and P. Lu. Entry trajectory optimization by second-order cone programming. *Journal of Guidance, Control, and Dynamics*, 39(2):227–241, 2016.

[25] Y. Mao, M. Szmuk, X. Xu, and B. Acikmese. Successive convexification: a superlinearly convergent algorithm for non-convex optimal control problems. arXiv preprint arXiv:1804.06539v2, 2019.

[26] M. Szmuk, B. Acikmese, and A. W. Berning. Successive convexification for fuel-optimal powered landing with aerodynamic drag and non-convex constraints. In *AIAA Guidance, Navigation, and Control Conference*, 2016.

[27] MOSEK ApS. *MOSEK Optimization Toolbox for MATLAB 11.0.30*, 2025.

[28] M. A. Patterson and A. 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 Transactions on Mathematical Software*, 41(1):1:1–1:37, October 2014.