CEAS-GNC-2022-040

CEAS EuroGNC 2022
"Conference on Guidance, Navigation and Control"
3-5 May 2022 @ Technische Universität Berlin, Germany

Mid-course Guidance for Dual-pulse Rocket Air-to-Air Missiles using Pseudospectral Sequential Convex Programming

Boseok Kim M.S. Student, KAIST, Dept. of Aerospace Engineering, 34141, Dajeon, Republic of Korea. kbobo6@kaist.ac.kr

Chang-Hun Lee Assistant Professor, KAIST, Dept. of Aerospace Engineering, 34141, Dajeon, Republic of Korea. lckdgns@kaist.ac.kr

Min-Jea Tahk Emeritus Professor, KAIST, Dept. of Aerospace Engineering, 34141, Dajeon, Republic of Korea. mjtahk@kaist.ac.kr

ABSTRACT

This paper proposes a mid-course guidance algorithm for dual-pulse rocket air-to-air missiles based on the trajectory optimization framework. Pseudospectral Sequential Convex Programming (PSCP) is utilized to solve the trajectory optimization problem for dual-pulse rocket missiles with control inputs of the angle-of-attack and ignition time of the second pulse. Multi-phase pseudospectral discretization enables us to treat the ignition time control problem by introducing an optimization parameter corresponding to the time interval between two pulses. Convex sub-problems are composed to solve the non-convex discretized trajectory optimization problem by linearizing the nonlinear dynamic constraints with a variable trust-region method and a line search method. Numerical simulations show promising results that the proposed algorithm can provide an accurate optimal solution within a few seconds in MATLAB.

Keywords: Mid-course Guidance, Dual-pulse Rocket Missiles, Trajectory Optimization, Pseudospectral Sequential Convex Programming (PSCP)

1 Introduction

Air-to-air missiles(AAMs) are one of the most significant combat systems in modern warfare. AAMs can destroy enemy flight machines such as fighter aircraft and missiles before they complete the missions. In order for AAMs to intercept the assigned targets successfully, a proper guidance algorithm is necessary. Entire guidance algorithms can be distinguished into three phases: the initial guidance phase, the mid-course guidance phase, and the terminal guidance phase. In the initial guidance phase, missiles are stabilized after separation from the fighter and align their flight path angle for the mid-course guidance phase. Then, a mid-course guidance algorithm brings the missiles around the targets to achieve the seeker's lock-on condition. Once the seeker acquires the target during the mid-course guidance phase, the terminal guidance phase that missiles try to intercept the targets based on the seeker's information begins. To increase the chance of interception in the terminal guidance phase, missiles need to have optimal operating conditions at handover time from the mid-course guidance phase to the terminal guidance phase. In this respect, there have been many studies to develop an optimal mid-course guidance algorithm to improve the terminal performance of missiles. A minimum time mid-course guidance

law was proposed in [1] based on the singular perturbation technique. The singular perturbation-based optimal mid-course guidance algorithm can also be found in [2–4] from the merit of the real-time applicability in the singular perturbation technique. In [5], a maximum final velocity mid-course guidance problem and a minimum time mid-course guidance problem were solved by the steepest ascent method. A near-optimal mid-course guidance algorithm that facilitates the on-board real-time calculation was proposed in [6] based on the neighboring optimal control theory. In [7], an analytic mid-course guidance law maximizing the final velocity was proposed. In [8], a combined optimal guidance law was derived analytically, which can be applied to both the mid-course guidance and the terminal guidance phases. In addition, neural network-based optimal mid-course guidance algorithms were proposed in [9–11].

As seen above, various optimal mid-course guidance algorithms have been studied so far. Many existing mid-course guidance algorithms, including the previously reviewed works, were developed for solid rocket missiles since conventional AAMs employed a solid rocket motor as a propulsion system. This feature restricted their maximum range and terminal performance from high drag after the boosting phase. On the other hand, the development of dual-pulse rocket motors enabled igniting two pulses separately with a variable delay. It gave missiles some freedom to adjust the thrusting pattern to relieve the energy loss from high drag by controlling the ignition time of the second pulse. From this inspection, it can be noted that the ignition time of the second pulse should also be optimized with the load factor to derive the maximal performance of dual-pulse rocket missiles. In this regard, some related works can be found. Optimal mid-course guidance algorithms for dual-pulse rocket missiles were developed based on the singular perturbation technique in [12, 13], the steepest ascent method in [14], the calculus of variations in [15, 16], and the quasi-Newton parameter optimization in [17]. Although several optimal mid-course guidance algorithms for dual-pulse rocket missiles were developed in previous studies, there is a limitation that the exact optimality is not guaranteed since they include certain approximations (ex. time scale separation, constant altitude) for real-time implementation of the guidance algorithms.

From this observation, this paper proposes an optimal mid-course guidance algorithm for dual-pulse rocket air-to-air missiles, which can provide the exact solution with the real-time applicability based on the trajectory optimization framework using the concept of computational guidance and control (CG&C) [18]. There are some previous works to solve missile guidance problems based on the trajectory optimization framework [19, 20], but none of them has ever considered optimizing the thrusting pattern of dual-pulse rocket missiles. In this paper, the mid-course guidance problem for dual-pulse rocket air-to-air missiles is defined as a trajectory optimization problem, and Pseudospectral Sequential Convex Programming (PSCP) is employed to solve the trajectory optimization problem. PSCP, which refers to a combination of Pseudospectral methods and Sequential Convex Programming, is a numerical optimization method successfully adapted to various problems and recognized for its high accuracy and real-time applicability [21–24]. In addition, a variable trust-region method and a line search method are applied in this study to attain the stable convergence property of the algorithm.

The remainder of this paper is constructed as follows: In Section 2, the mid-course guidance problem for dual-pulse rocket air-to-air missiles is formulated as a trajectory optimization problem. In Section 3, the PSCP formulation of the trajectory optimization problem is presented. In Section 4, simulation results are given to demonstrate the performance of the proposed method. Finally, the conclusion of this work is provided in Section 5.

2 Problem Formulation

2.1 Dynamics and Constraints

This subsection presents the equations of motion, boundary conditions, and flight constraints for dual-pulse rocket air-to-air missiles. In this study, a two-dimensional engagement in a longitudinal plane is considered. The missile is regarded as a point mass, then the equations of motion are given as

$$\dot{x} = V\cos\gamma \tag{1}$$

$$\dot{y} = V\sin\gamma \tag{2}$$

$$\dot{V} = \frac{-D + T\cos\alpha}{m} - g_0\sin\gamma \tag{3}$$

$$\dot{\gamma} = \frac{L + T\sin\alpha}{mV} - \frac{g_0\cos\gamma}{V} \tag{4}$$

where the variables x, y, V, γ, α denote the downrange, altitude, velocity, flight path angle, and angle-of-attack, respectively. The parameter g_0 represents the constant gravitational acceleration. To ensure the physical limitation of missiles that the angle-of-attack cannot vary too fast, the following state equation and constraint are additionally imposed.

$$\dot{\alpha} = \xi \tag{5}$$

$$|\xi| \leq \dot{\alpha}_{\max} \tag{6}$$

where ξ represents the angle-of-attack rate, and $\dot{\alpha}_{max}$ denotes the maximum angle-of-attack rate. In addition, the angle-of-attack is also bounded as follows.

$$\alpha_{\min} \leq \alpha \leq \alpha_{\max} \tag{7}$$

The variables L and D in Eqs. (3) and (4) represent the aerodynamic lift and drag force which can be computed by the following equations.

$$L = \frac{1}{2}\rho V^2 S_{ref} C_L \tag{8}$$

$$D = \frac{1}{2}\rho V^2 S_{ref} C_D \tag{9}$$

where the variables C_L and C_D are the aerodynamic lift and drag coefficients which are functions of Mach number and angle-of-attack, the variable ρ represents the atmospheric density governed by altitude. The parameter S_{ref} is the reference area. In this study, the following models are utilized to compute the atmospheric density ρ and the aerodynamic coefficients C_L and C_D.

$$\rho = \rho_0\, e^{-y/y_s} \tag{10}$$

$$C_L = C_{L_\alpha}\alpha \tag{11}$$

$$C_D = C_{D_0} + KC_L^2 \tag{12}$$

The variables T and m in Eqs. (3) and (4) represent the thrust and mass. For dual-pulse rocket missiles, they are governed by the predesignated profiles described in Eqs. (13) and (14). Here, η_1 corresponds to the time interval between two pulses.

$$T = \begin{cases} T_1, & t \le \Delta t_{b1} \\ 0, & \Delta t_{b1} < t \le \Delta t_{b1} + \eta_1 \\ T_2, & \Delta t_{b1} + \eta_1 < t \le \Delta t_{b1} + \eta_1 + \Delta t_{b2} \\ 0, & \Delta t_{b1} + \eta_1 + \Delta t_{b2} < t \le t_f \end{cases} \tag{13}$$

$$m = \begin{cases} m_0 - \dot{m}_1 t, & t \le \Delta t_{b1} \\ m_0 - \dot{m}_1 \Delta t_{b1}, & \Delta t_{b1} < t \le \Delta t_{b1} + \eta_1 \\ m_0 - \dot{m}_1 t - \dot{m}_2 (t - \Delta t_{b1} - \eta_1), & \Delta t_{b1} + \eta_1 < t \le \Delta t_{b1} + \eta_1 + \Delta t_{b2} \\ m_0 - \dot{m}_1 \Delta t_{b1} - \dot{m}_2 \Delta t_{b2}, & \Delta t_{b1} + \eta_1 + \Delta t_{b2} < t \le t_f \end{cases} \tag{14}$$

As seen in Eqs. (13) and (14), the entire time domain can be divided into four phases: two boosting phases and two gliding phases. It should be noted that the boosting phases have the fixed time intervals: Δt_{b1} and Δt_{b2}, but the gliding phases have the variable time intervals: η_1 and η_2 as depicted in Fig. 1.

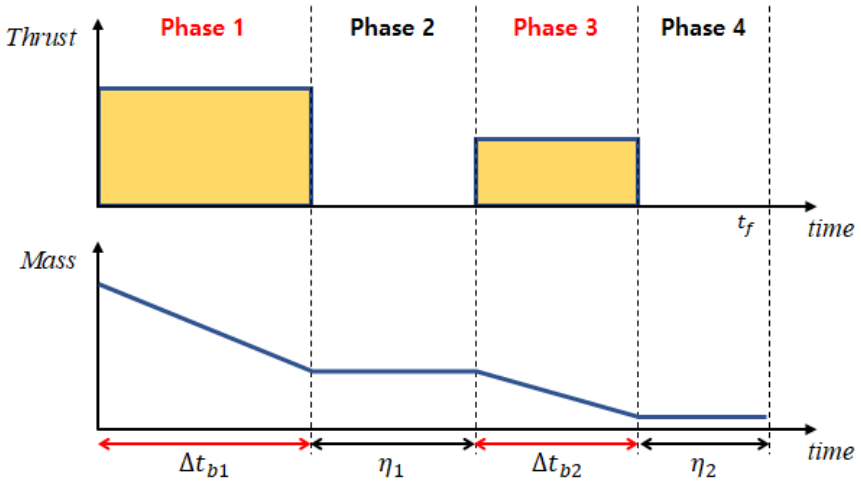


Fig. 1 Thrust and Mass Profile

Note that the total flight time can be represented as the sum of flight time in each phase. In addition, the variable time intervals η_1 and η_2 should be greater than or equal to zero.

$$t_f = \Delta t_{b1} + \eta_1 + \Delta t_{b2} + \eta_2 \tag{15}$$

$$\eta_1 \ge 0, \quad \eta_2 \ge 0 \tag{16}$$

Next, boundary conditions are defined for the mid-course guidance problem. In this study, it is assumed that the initial states are constrained by the following conditions.

$$x(t_0) = x_0 \qquad y(t_0) = y_0 \qquad V(t_0) = V_0 \qquad \gamma(t_0) = \gamma_0 \tag{17}$$

In many cases, terminal constraints are directly connected to the purpose of problems. The major goal of the mid-course guidance is to lead missiles to a Predicted Intercept Point(PIP) with attaining the high chance of the interception of targets in the terminal guidance phase. To this end, the following final conditions are considered in this study.

$$x(t_f) = x_f \qquad y(t_f) = y_f \qquad \gamma(t_f) = \gamma_f \qquad \alpha(t_f) = 0 \tag{18}$$

The first two things represent the PIP constraint, and the last two things are for the field-of-view(FOV) constraint to ensure the seeker captures the target at the lock-on moment. In addition, the last constraint can provide the operational margin of the acceleration at the beginning of the terminal guidance phase. The terminal velocity, instead of being constrained, is set to be an optimization parameter to be maximized, as explained in the following subsection.

2.2 Maximum Terminal Velocity Problem

This study focuses on the mid-course guidance problem maximizing the terminal velocity of missiles. It is advantageous to maximize the terminal velocity of missiles during the mid-course guidance phase for attaining high maneuverability in the terminal guidance phase. The performance index to be minimized can be represented as follows.

$$J = -V\left(t_f\right) \tag{19}$$

Then, the maximum terminal velocity problem for dual-pulse rocket air-to-air missiles can be formulated to the following trajectory optimization problem.

$$\begin{aligned} \mathbf{P0}: \quad & \text{minimize} \quad J = -V\left(t_f\right) \\ & \text{subject to} \quad \text{Eqs.}\,(1)-(18) \end{aligned} \tag{20}$$

The above problem **P0** is a continuous optimal control problem with a non-convex structure and free-final time. The problem **P0** is firstly discretized using pseudospectral methods, and then, the discretized problem is solved by Sequential Convex Programming(SCP) in the following section.

3 Pseudospectral Sequential Convex Programming

3.1 System Equation

Based on the equations of motion and the state equation described in Eqs. (1)-(5), let the state vector and the temporary control input be defined as follows.

$$z \triangleq \begin{bmatrix} x & y & V & \gamma & \alpha \end{bmatrix}, \quad \tilde{u} \triangleq \dot{\alpha} \tag{21}$$

Then, the system equation can be represented as

$$\dot{z} = f(z) + B\tilde{u} \tag{22}$$

where

$$f(z) = \begin{bmatrix} V\cos\gamma \\ V\sin\gamma \\ \dfrac{-D+T\cos\alpha}{m} - g\sin\gamma \\ \dfrac{L+T\sin\alpha}{mV} - \dfrac{g\cos\gamma}{V} \\ 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{23}$$

Note that the system equation is given in the form of the control-affine system. This feature makes the convexification process simpler when applying Sequential Convex Programming (SCP) to the discretized trajectory optimization problem.

3.2 Multi-phase Pseudospectral Discretization

In this study, multi-phase pseudospectral discretization is employed to deal with the trajectory optimization problem **P0**, which has a free-final time and the decision variable η_1 corresponding to the time interval between two pulses in the time domain. The entire time domain is divided into four phases according to the thrust profile in Fig. 1, and Legendre-Gauss-Radau pseudospectral method is applied to each phase. The first step is to normalize the time domain of each phase by its time interval. Then, the system equation splits up into the following four system equations according to the phases.

$$z'_{p_4} = \frac{\Delta t_{b1}}{2} f(z_{p_1}) + B u_{p_1} \tag{24}$$

$$z'_{p_2} = \frac{\eta_1}{2} f(z_{p_2}) + B u_{p_2} \tag{25}$$

$$z'_{p_3} = \frac{\Delta t_{b2}}{2} f(z_{p_3}) + B u_{p_3} \tag{26}$$

$$z'_{p_4} = \frac{\eta_2}{2} f(z_{p_4}) + B u_{p_4} \tag{27}$$

where $z_{p_1}, z_{p_2}, z_{p_3}, z_{p_4}$ denote the state vectors, $u_{p_1}, u_{p_2}, u_{p_3}, u_{p_4}$ denote the control inputs for each phase, f and B are defined in Eq. (23), and z' represents the differentiation of the state vector with respect to the normalized time. The control inputs are defined as the product of the temporary control input in Eq. (21) and half of the time interval for each phase as follows.

$$u_{p_1} \triangleq \frac{\Delta t_{b1}}{2}\tilde{u}, \quad u_{p_2} \triangleq \frac{\eta_1}{2}\tilde{u}, \quad u_{p_3} \triangleq \frac{\Delta t_{b2}}{2}\tilde{u}, \quad u_{p_4} \triangleq \frac{\eta_2}{2}\tilde{u} \tag{28}$$

Suppose that the four phases have $N_1, N_2, N_3,$ and N_4 numbers of LGR points, respectively, then the discretized state vectors and control inputs can be expressed as

$$\mathbf{Z}_{p_i} \triangleq \left[Z_{p_i,1}, Z_{p_i,2}, ..., Z_{p_i,N_i+1}\right]^T, \quad \mathbf{U}_{p_i} \triangleq \left[U_{p_i,1}, U_{p_i,2}, ..., U_{p_i,N_i+1}\right]^T \tag{29}$$

where

$$Z_{p_i,n} \triangleq [x_{p_i}(\tau_n), y_{p_i}(\tau_n), V_{p_i}(\tau_n), \gamma_{p_i}(\tau_n), \alpha_{p_i}(\tau_n)]^T, \quad U_{p_i,n} \triangleq u_{p_i}(\tau_n) \tag{30}$$

where $i = 1,2,3,4$. In applying the multi-phase pseudospectral discretization, linkage conditions between two adjacent phases should be imposed for continuity of state vectors as

$$Z_{p_1,N_1+1} = Z_{p_2,1}, \quad Z_{p_2,N_2+1} = Z_{p_3,1}, \quad Z_{p_3,N_3+1} = Z_{p_4,1} \tag{31}$$

The boundary conditions in Eqs. (17), (18) can be represented using the discretized state vectors as follows. Here, $*$ means that the corresponding state is not constrained.

$$Z_{p_1,1} = \begin{bmatrix} x_0 & y_0 & V_0 & \gamma_0 & * \end{bmatrix}^T, \qquad Z_{p_4,N_4+1} = \begin{bmatrix} x_f & y_f & * & \gamma_f & 0 \end{bmatrix}^T \tag{32}$$

where $*$ means that the corresponding states are not constrained. The state and control constraints in Eqs. (6), (7) can also be represented using the discretized states and control inputs as follows.

$$\alpha_{\min} \leq \alpha_{p_i,n} \leq \alpha_{\max} \tag{33}$$

$$\left|u_{p_1,n}\right| \leq \frac{\Delta t_{b1}}{2}\dot{\alpha}_{\max}, \quad \left|u_{p_2,n}\right| \leq \frac{\eta_1}{2}\dot{\alpha}_{\max}, \quad \left|u_{p_3,n}\right| \leq \frac{\Delta t_{b2}}{2}\dot{\alpha}_{\max}, \quad \left|u_{p_4,n}\right| \leq \frac{\eta_2}{2}\dot{\alpha}_{\max} \tag{34}$$

where $n = 1,2,...,N_{i+1}$ and $i = 1,2,3,4$. The remaining things to be discretized are the dynamic constraints and the objective function. Denoting the Radau differentiation matrix of each phase to $\mathbf{D}_{p_i}$, the

system equations in Eqs. (24)-(27) can be replaced by the following algebraic equations [25].

$$\mathbf{D}_{p_1}\mathbf{Z}_{p_1} = \frac{\Delta t_{b1}}{2}\mathbf{F}\left(\mathbf{Z}_{p_1}\right) + B\mathbf{U}_{p_1} \tag{35}$$

$$\mathbf{D}_{p_2}\mathbf{Z}_{p_2} = \frac{\eta_1}{2}\mathbf{F}\left(\mathbf{Z}_{p_2}\right) + B\mathbf{U}_{p_2} \tag{36}$$

$$\mathbf{D}_{p_3}\mathbf{Z}_{p_3} = \frac{\Delta t_{b2}}{2}\mathbf{F}\left(\mathbf{Z}_{p_3}\right) + B\mathbf{U}_{p_3} \tag{37}$$

$$\mathbf{D}_{p_4}\mathbf{Z}_{p_4} = \frac{\eta_2}{2}\mathbf{F}\left(\mathbf{Z}_{p_4}\right) + B\mathbf{U}_{p_4} \tag{38}$$

where

$$\mathbf{F}\left(\mathbf{Z}_{\mathbf{p_i}}\right) \triangleq \begin{bmatrix} f\left(Z_{p_i,1}\right) & f\left(Z_{p_i,2}\right) & \cdots & f\left(Z_{p_i,N_i}\right) \end{bmatrix}^T \tag{39}$$

where $n = 1,2,...,N_{i+1}$ and $i = 1,2,3,4$. The discretized objective function is simply expressed as

$$J = -V_{p_4,N_4+1} \tag{40}$$

All constraints and the objective function of the problem **P0** are discretized, and the discretized trajectory optimization problem can be written as follows.

$$\begin{aligned} \mathbf{P1}: \quad & \text{minimize} \quad J = -V_{p_4,N_4+1} \\ & \text{subject to} \quad \text{Eqs.}\,(31)-(38) \end{aligned} \tag{41}$$

Note that the constraints in Eqs. (31)-(34) are convex, but the constraints in Eqs. (35)-(38) are non-convex from the nonlinear dynamics represented in Eq. (39) and the presence of optimization variables η_1, η_2 in Eqs. (36) and (38). In this study, the non-convexity is handled by the successive convexification method. The entire process to solve the problem **P1**, called Sequential Convex Programming, is examined in the following subsection.

3.3 Sequential Convex Programming

Sequential Convex Programming refers to an algorithm to solve general optimization problems by solving a sequence of convex sub-problems. There are various convexification methods such as change of variables, relaxation, and successive linearization/approximation [26]. The successive linearization method is mainly used in this study to convexify the nonlinear dynamic constraints in Eqs. (35)-(38). They are given in the form of matrix equations, so for convenience, let us construct the matrix equations in the form of column vector equations as

$$\begin{bmatrix} D_{p_1}^{(n,1)} \times \mathbf{I}_5 & \cdots & D_{p_1}^{(n,N_1+1)} \times \mathbf{I}_5 \end{bmatrix} \mathbf{Z}_{r_1} = \frac{\Delta t_{b1}}{2} f\left(Z_{p_1,n}\right) + BU_{p_1,n}, \quad n = 1,...,N_1 \tag{42}$$

$$\begin{bmatrix} D_{p_2}^{(n,1)} \times \mathbf{I}_5 & \cdots & D_{p_2}^{(n,N_2+1)} \times \mathbf{I}_5 \end{bmatrix} \mathbf{Z}_{r_2} = \frac{\eta_1}{2} f\left(Z_{p_2,n}\right) + BU_{p_2,n}, \quad n = 1,...,N_2 \tag{43}$$

$$\begin{bmatrix} D_{p_3}^{(n,1)} \times \mathbf{I}_5 & \cdots & D_{p_3}^{(n,N_3+1)} \times \mathbf{I}_5 \end{bmatrix} \mathbf{Z}_{r_3} = \frac{\Delta t_{b2}}{2} f\left(Z_{p_3,n}\right) + BU_{p_3,n}, \quad n = 1,...,N_3 \tag{44}$$

$$\begin{bmatrix} D_{p_4}^{(n,1)} \times \mathbf{I}_5 & \cdots & D_{p_4}^{(n,N_4+1)} \times \mathbf{I}_5 \end{bmatrix} \mathbf{Z}_{r_4} = \frac{\eta_2}{2} f\left(Z_{p_4,n}\right) + BU_{p_4,n}, \quad n = 1,...,N_4 \tag{45}$$

where

$$\mathbf{Z}_{r_i} \triangleq \left[Z_{p_i,1}^T, Z_{p_i,2}^T, ..., Z_{p_i,N_i+1}^T\right]^T, \qquad i = 1,2,3,4 \tag{46}$$

where $D_{p_i}^{(j,k)}$ represents the (j,k)-th element of $\mathbf{D}_{p_i}$, and $\mathbf{I}_5$ denotes the identity matrix of dimension 5. Hereafter, the coefficient matrices of the left-hand sides are written as $\mathbf{D}_{r_i}$. Then, successive linearization is applied to the right-hand side of the equations. It should be noted that the linearization schemes are slightly different for the boosting phases in Eqs. (42) and (44) and the gliding phases in Eqs. (43) and (45) from the existence of optimization variables η_1 and η_2. Using the notation of $(\cdot)^k$ to represent the solution obtained in the previous k-th iteration, the dynamic constraints for the boosting phases can be linearized as

$$\mathbf{D}_{r_1}\mathbf{Z}_{r_1} = \frac{\Delta t_{b1}}{2} A\left(Z^k{}_{p_1,n}\right)\left(Z_{p_1,n} - Z^k_{p_1,n}\right) + BU_{p1,n} \quad n = 1,...,N_1 \tag{47}$$

$$\mathbf{D}_{r_3}\mathbf{Z}_{r_3} = \frac{\Delta t_{b2}}{2} A\left(Z^k{}_{p_3,n}\right)\left(Z_{p_3,n} - Z^k_{p_3,n}\right) + BU_{p3,n} \quad n = 1,...,N_3 \tag{48}$$

where

$$A\left(Z^k_{p_i,n}\right) \triangleq \left.\frac{\partial f}{\partial Z}\right|_{Z=Z^k_{p_i,n}} \tag{49}$$

The linearized dynamic constraints for the gliding phases are given as follows.

$$\mathbf{D}_{r_2}\mathbf{Z}_{r_2} = \frac{\eta_1^k}{2} A\left(Z^k_{p_2,n}\right) Z_{p_2,n} + \frac{1}{2} f\left(Z^k_{p_2,n}\right)\eta_1 - \frac{\eta_1^k}{2} f\left(Z^k_{p_2,n}\right) + BU_{p_2,n}, \quad i = 1,...,N_2 \tag{50}$$

$$\mathbf{D}_{r_4}\mathbf{Z}_{r_4} = \frac{\eta_2^k}{2} A\left(Z^k_{p_4,n}\right) Z_{p_4,n} + \frac{1}{2} f\left(Z^k_{p_4,n}\right)\eta_1 - \frac{\eta_1^k}{2} f\left(Z^k_{p_4,n}\right) + BU_{p_4,n}, \quad i = 1,...,N_4 \tag{51}$$

The nonlinear dynamic constraints in Eqs. (35)-(38) are transformed to the linearized dynamic constraints in Eqs. (47)-(51) which are convex. For the convex sub-problems to be composed, the non-convex constraints of Eqs. (35)-(38) in the problem **P1** are replaced by the convex constraints of Eqs. (47)-(51). When solving non-convex optimization problems using the successive linearization method, trust-region constraints are required for the linearization process to be valid in constructing convex sub-problems. This study utilizes a variable quadratic trust-region method with a line-search method introduced in [27]. The variable quadratic trust-region method can be realized by the state constraints and the augmentation of a penalty term to the objective function. The state constraints are given as

$$\left(Z_{p_i,n} - Z^k_{p_i,n}\right)^T \left(Z_{p_i,n} - Z^k_{p_i,n}\right) \leq s_{p_i,n} \tag{52}$$

where $n = 1,2,...,N_i$ and $i = 1,2,3,4$. The augmented objective function is given in Eq. (53). The augmented objective function converges to the original objective function in Eq. (40) if the iterative solution converges sufficiently, which is the equivalent to $\|\mathbf{s}\|_2 \to 0$.

$$J = -V_{p_4,N_4+1} + w_s\|\mathbf{s}\|_2 \tag{53}$$

where

$$\mathbf{s} \triangleq \left[s_{p_1,1},\ \ldots,s_{p_1,N_1},s_{p_2,1},\ \ldots,s_{p_2,N_2},\ \ldots,\ s_{p_4,1},\ \ldots,s_{p_4,N_4}\right] \tag{54}$$

Then, the non-convex optimization problem **P1** can be formulated to the following convex sub-problem by including the augmented objective function, the linearized dynamic constraints and the quadratic trust-region constraints.

$$\begin{aligned} \mathbf{P2}: \quad & \text{minimize} \quad J = -V_{p_4,N_4+1} + w_s\|\mathbf{s}\|_2 \\ & \text{subject to} \quad \text{Eqs.}\,(31)-(34),\,(47)-(52) \end{aligned} \tag{55}$$

Note that the convex optimization problem **P2** can be solved efficiently and reliably by commercial convex optimization solvers such as MOSEOK, SeDuMi, and SDPT3. The solution of the problem **P1** can be obtained by solving the problem **P2** repeatedly until the iterative solution converges. In addition, this study applies a line search method after each iteration if the convergence condition is not satisfied. Line search methods are known that can provide robustness for the SCP algorithm [27]. To this end, let us define the following merit function that measures both the cost and the violation of original dynamic constraints.

$$\phi\left(\mathbf{X}^k;\mu\right) = J^k + \mu\sum_{i=1}^{4}\sum_{n=1}^{N_i}\left\|h_{p_i,n}\left(\mathbf{X}^k\right)\right\|_1 \tag{56}$$

where $\mathbf{X}^k = \left\{\mathbf{Z}^k, \eta^k, \mathbf{U}^k, \mathbf{s}^k\right\}$ is the solution pair, J^k is the cost, and $h_{p_i,n}\left(\mathbf{X}^k\right)$ is the residual error of the original dynamic constraints of Eqs. (35)-(38) at each node in the k-th iteration. Denoting the $(k+1)$-th solution pair of the convex sub-problem **P2** as $\mathbf{X}^{k+1}$, the search direction is computed as $p^k = \mathbf{X}^{k+1} - \mathbf{X}^k$. The $(k+1)$-th solution pair is updated after the line search along the search direction with the step length ζ^k as

$$\hat{\mathbf{X}}^{k+1} = \mathbf{X}^k + \zeta^k p^k \tag{57}$$

This study employs a backtracking line search with a sufficient decrease condition to determine the proper step length ζ^k. The step length is initiated with the value of 1 and decreased at a constant rate as $\zeta \leftarrow \kappa\zeta,\ \ \kappa \in (0,1)$ until the sufficient decrease condition is satisfied.

$$\phi\left(\mathbf{X}^k + \zeta^k p^k;\mu\right) \le \phi\left(\mathbf{X}^k;\mu\right) + \lambda\,\zeta^k \nabla_{p^k}\phi\left(\mathbf{X}^k;\mu\right) \tag{58}$$

where $\lambda \in (0,1)$ is a design parameter related to the desired reduction rate of the merit function, and $\nabla_{p^k}\phi\left(\mathbf{X}^k;\mu\right)$ is the directional derivative of $\phi\left(\mathbf{X}^k;\mu\right)$ along the search direction p^k given by

$$\nabla_{p^k}\phi\left(\mathbf{X}^k;\mu\right) = \lim_{\varepsilon\to 0}\frac{\phi\left(\mathbf{X}^k + \varepsilon p^k;\mu\right) - \phi\left(\mathbf{X}^k;\mu\right)}{\varepsilon} \tag{59}$$

The SCP algorithm to solve the problem **P1** can be described by Eqs. (55)-(59). Since the problem **P1** comes from the pseudospectral discretization of the problem **P0**, we call the entire process to solve the original problem **P0** as Pseudospectral Sequential Convex Programming (PSCP) algorithm. Then, the PSCP algorithm can be summarized as follows.

PSCP Algorithm

1) Set $k = 0$ and provide an initial guess for the solution pair $\mathbf{X}^0$
2) Using the previous solution pair $\mathbf{X}^k$, construct and solve the convex sub-problem **P2**
3) Check the convergence condition given in Eq. (60). If the following condition is satisfied, go to Step 4. If not, go to Step 5.

$$\max_{i,n}\left|Z_{p_i,n}^{k+1} - \hat{Z}_{p_i,n}^{k}\right| \preceq \varepsilon_{tol}, \quad k \ge 1 \tag{60}$$

4) Return the newly obtained solution pair $\mathbf{X}^{k+1}$ as the optimal solution, and stop the algorithm
5) Apply the line search described in Eqs. (56)-(59), update the iteration number as $k \leftarrow k+1$, and go back to Step 2.

As shown in Step 1, the initial guess is required for the first iteration of the algorithm. In this study, the initial guess of states is set to linear interpolation of both endpoints if specified. The downrange, altitude, and flight path angle are those cases, but the velocity and angle-of-attack are not. The angle-of-attack is just set to zero at each node, and for the velocity, the maximum value is roughly estimated at first by the following equation.

$$V_{max} = V_0 + \frac{T_1 \Delta t_{b1} + T_2 \Delta t_{b2}}{m_0 - \dot{m}_1 \Delta t_{b1}} \tag{61}$$

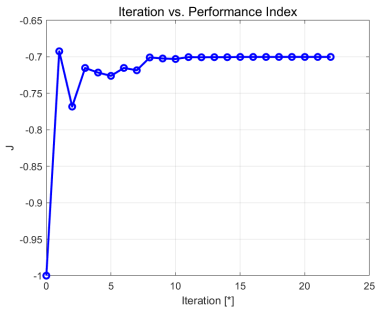
Then, linear interpolation between the initial and maximum velocity is utilized for the initial guess. The variable time intervals are estimated by subtracting the burning times from half of the straight distance to PIP divided by average speed.

$$\eta_1^0 = \frac{\sqrt{\left(x_f - x_0\right)^2 + \left(y_f - y_0\right)^2}}{V_0 + V_{\max}} - \Delta t_{b1}, \quad \eta_2^0 = \frac{\sqrt{\left(x_f - x_0\right)^2 + \left(y_f - y_0\right)^2}}{V_0 + V_{max}} - \Delta t_{b2} \tag{62}$$

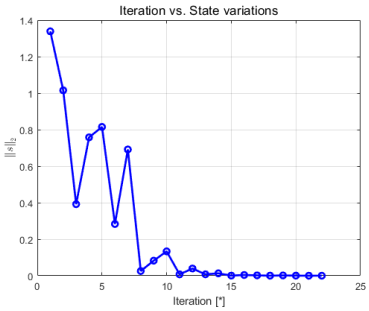
4 Simulation Results

In this section, the performance of the proposed algorithm is demonstrated by numerical simulations. For trajectory optimization of dual-pulse rocket air-to-air missiles, simulation parameters are given as follows. The parameters for the exponential atmosphere model are set to $\rho_0 = 1.225 kg/m^3$ and $h_s = 7254.24m$. Missile operating parameters are selected as $\alpha_{min} = -20\,\text{deg}$, $\alpha_{max} = 20\,\text{deg}$, $\dot{\alpha}_{max} = 4\text{deg}/s$, $T_1 = T_2 = 8000N$, $m_0 = 135kg$, $\dot{m}_1 = \dot{m}_2 = 3.5kg/s$, and $\Delta t_{b1} = \Delta t_{b2} = 5.0s$, respectively. The parameters related to missile configurations are chosen as $C_{L_\alpha} = 20.0$, $C_{D_0} = 0.3$, $K = 0.2$, and $S_{ref} = 0.0249m^2$. The number of LGR points is set to 15 for each phase (i.e., $N_1 = N_2 = N_3 = N_4 = 15$), and the weight for the variable quadratic trust-region method is set to $w_s = 0.05$. The design parameters for the line search are selected as $\kappa = 0.7$, $\mu = 100$, and $\lambda = 0.3$. The stopping criteria for the SCP algorithm in Eq. (60) is selected as $\varepsilon_{tol} = [50m, 50m, 1.0m/s, 0.01\,\text{deg}, 0.025\,\text{deg}]$. The considered engagement scenario is to reach the PIP at $(x_f, y_f) = (10km, 7km)$ with the terminal flight path angle $\gamma_f = -10\,\text{deg}$ and angle-of-attack $\alpha_f = 0\,\text{deg}$ from the initial states of $x_0 = 0km$, $y_0 = 10km$, $V_0 = 350m/s$, and $\gamma_0 = 10\,\text{deg}$.

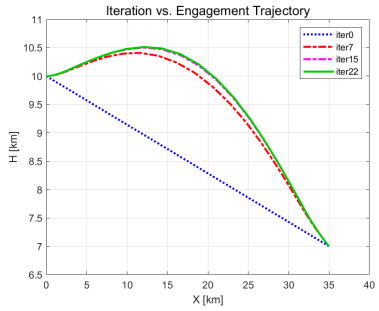
Then, MATLAB with MOSEK [28] is utilized to solve the convex sub-problem **P2**. The trajectory optimization is performed on a desktop with an Intel(R) i7-9700 3.00GHz processor, and each iteration takes around 0.05 seconds to solve the problem. It takes 22 iterations for the SCP algorithm to solve the discretized trajectory optimization problem **P1**. The convergence patterns for the performance index J, the magnitude of state variations $\|s\|_2$, the variable time intervals η_1 and η_2, and the engagement trajectory are depicted in Fig. 2. The results show that the performance index gradually approaches its final value as the engagement trajectory and variable time intervals converge along with the iterations. In addition, Fig. 2(b) shows that the magnitude of state variations converges to zero. Note that the magnitude of state variations $\|s\|_2$ is augmented to the objective function as described in Eq. (53). It means that even though we solve the problem **P2** having the augmented objective function, the converged optimal solution has the same value with the original objective function since $\|\mathbf{s}\|_2 = 0$ at the last moment. To check the validity of the optimal solution, this study compares the optimization results obtained by the proposed algorithm (PSCP) and the general-purpose MATLAB software, GPOPS-II [29]. The results of engagement trajectory, velocity, flight path angle, angle-of-attack, and thrust profile are compared in Fig. 3, and it can be seen that the optimal solutions obtained by the proposed algorithm and GPOPS-II are almost identical. It verifies that the proposed algorithm can provide an accurate optimal solution in trajectory optimization for dual-pulse rocket air-to-air missiles.



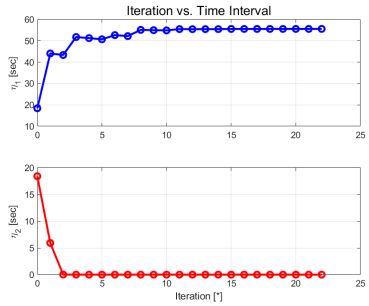
(a) Performance index



(b) Magnitude of state variations

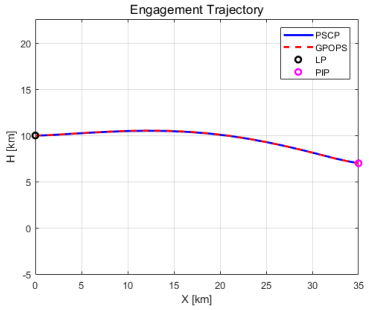


(c) Engagement trajectory

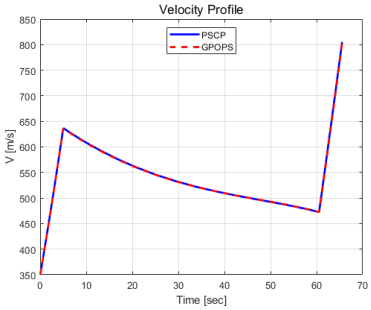


(d) Variable time intervals

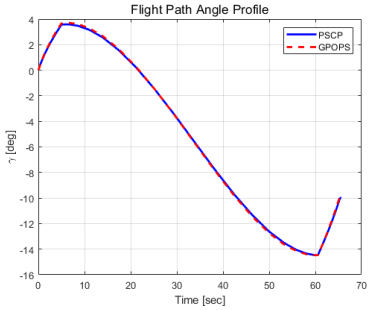
Fig. 2 Convergence patterns for the performance index, magnitude of state variations, engagement trajectory, and variable time intervals



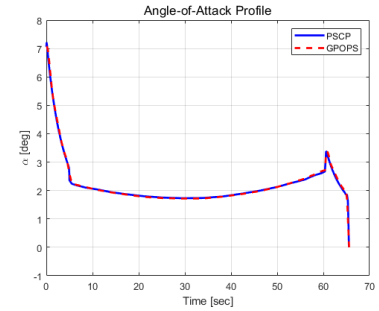
(a) Engagement trajectory



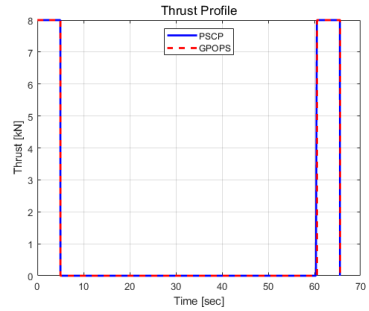
(b) Velocity profile



(c) Flight path angle profile



(d) Angle-of-attack profile



(e) Thrust profile

Fig. 3 Comparison results for the engagement trajectory, velocity, flight path angle, angle-of-attack, thrust profiles between the proposed method (PSCP) and GPOPS-II

5 Conclusion

This paper presents a way to solve the mid-course guidance problem for dual-pulse rocket air-to-air missiles based on the trajectory optimization framework. The optimal trajectory of dual-pulse rocket missiles is established by optimizing the angle-of-attack and thrust profiles. Multi-phase pseudospectral discretization that considers the entire time domain as two fixed-time domains and two variable-time domains is applied to deal with the trajectory optimization problem, which is a free-final time problem and has the decision variable in the time domain. Then, convex sub-problems are composed by linearizing the nonlinear dynamic constraints of the discretized trajectory optimization problem. The solution to the original problem is obtained by solving the convex sub-problems iteratively with the line search method. In addition, numerical simulations are conducted to examine the performance of the proposed algorithm. It is shown that the proposed algorithm obtains the solution after 22 iterations which take about 0.05s in each iteration using a commercial solver MOSEK with MATLAB. Comparison with the results obtained by GPOPS-II reveals that the proposed algorithm can provide an exact solution. From these observations, it can be concluded that the proposed algorithm has potential for trajectory optimization-based closed-loop mid-course guidance for dual-pulse rocket air-to-air missiles.

Acknowledgments

This work was supported by Theater Defense Research Center funded by Defense Acquisition Program Administration under Grant UD200043CD.

References

[1] V.H.L Cheng and N.K Gupta. Advanced midcourse guidance for air-to-air missiles. *Journal of Guidance, Control, and Dynamics*, 9(2):135–142, 1986.

[2] P.K.A. Menon and M.M. Briggs. A midcourse guidance law for air-to-air missiles. In *Proceedings of the Guidance, Navigation and Control Conference*, Monterey, CA, August 1987.

[3] P.K.A. Menon and M.M. Briggs. Near-optimal midcourse guidance for air-to-air missiles. *Journal of Guidance, Control, and Dynamics*, 13(4):596–602, 1990.

[4] M. Manickavasagam, A.K. Sarkar, and V. Vaithiyanathan. A singular perturbation based midcourse guidance law for realistic air-to-air engagement. *Defence Science Journal*, 67(1), 2017.

[5] F. Imado, T. Kuroda, and S. Miwa. Optimal midcourse guidance for medium-range air-to-air missiles. *Journal of Guidance, Control, and Dynamics*, 13(4):603–608, 1990.

[6] S. Katzir, E.M. Cliff, and F.H. Lutze. An approach to near optimal guidance on-board calculations. In *Proceedings of ICCON IEEE International Conference on Control and Applications*, April 1989.

[7] Shih-Ming Yang. Analysis of optimal midcourse guidance law. *IEEE Transactions on Aerospace and Electronic Systems*, 32(1):419–425, 1996.

[8] C.F. Lin and L.L. Tsai. Analytical solution of optimal trajectory-shaping guidance. *Journal of Guidance, Control, and Dynamics*, 10(1):60–66, 1987.

[9] Eun-Jung Song and Min-Jea Tahk. Real-time midcourse guidance with intercept point prediction. *Control Engineering Practice*, 6(8):957–967, 1998.

[10] Eun-Jung Song and Min-Jea Tahk. Real-time midcourse missile guidance robust against launch conditions. *Control Engineering Practice*, 7(4):507–515, 1999.

[11] Eun-Jung Song and Min-Jea Tahk. Real-time neural-network midcourse guidance. *Control Engineering Practice*, 9(10):1145–1154, 2001.

[12] V.H.L. Cheng, P.K.A. Menon, N.K. Gupta, and M.M. Briggs. Reduced-order pulse-motor ignition control logic. *Journal of Guidance, Control, and Dynamics*, 10(4):343–350, 1987.

[13] C. Annam, A. Ratnoo, and D. Ghose. Singular-perturbation-based guidance of pulse motor interceptors with look angle constraints. *Journal of Guidance, Control, and Dynamics*, pages 1–15, 2021.

[14] F. Imado, T. Kuroda, and S. Miwa. Optimal thrust control of a missile with a pulse motor. *Journal of guidance, control, and dynamics*, 14(2):377–382, 1991.

[15] A.J. Calise and J. Nagy. Necessary conditions for optimal pulse control. *Journal of Guidance, Control, and Dynamics*, 9(1):53–57, 1986.

[16] A.J. Calise and J.V.R. Prasad. Pulse motor control for maximizing average velocity. *Journal of Guidance, Control, and Dynamics*, 12(2):169–174, 1989.

[17] P.K.A. Menon, V.H.L. Cheng, Ching-An Lin, and M.M. Briggs. High-performance missile synthesis with trajectory and propulsion system optimization. *Journal of Spacecraft and Rockets*, 24(6):552–557, 1987.

[18] P. Lu. Introducing computational guidance and control. *Journal of Guidance, Control, and Dynamics*, 40(2):193, 2017.

[19] H.H. Kwon and H.L. Choi. A convex programming approach to mid-course trajectory optimization for air-to-ground missiles. *International Journal of Aeronautical and Space Sciences*, pages 1–14, 2019.

[20] Xinfu Liu, Zuojun Shen, and Ping Lu. Exact convex relaxation for optimal flight of aerodynamically controlled missiles. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1881–1892, 2016.

[21] M. Sagliano. Pseudospectral convex optimization for powered descent and landing. *Journal of guidance, control, and dynamics*, 41(2):320–334, 2018.

[22] J. Wang and N. Cui. A pseudospectral-convex optimization algorithm for rocket landing guidance. In *AIAA Guidance, Navigation, and Control Conference*, January 2018.

[23] M. Sagliano and E. Mooij. Optimal drag-energy entry guidance via pseudospectral convex optimization. *Aerospace Science and Technology*, 117:106946, 2021.

[24] J. Wang, N. Cui, and C. Wei. Rapid trajectory optimization for hypersonic entry using convex optimization and pseudospectral method. *Aircraft Engineering and Aerospace Technology*, 2019.

[25] D. Garg, M. Patterson, W. W. Hager, A. V Rao, D. A Benson, and G. T. Huntington. A unified framework for the numerical solution of optimal control problems using pseudospectral methods. *Automatica*, 46(11):1843–1851, 2010.

[26] X. Liu, P. Lu, and B. Pan. Survey of convex optimization for aerospace applications. *Astrodynamics*, 1(1):23–40, 2017.

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

[28] M. ApS. Mosek optimization toolbox for matlab. *User's Guide and Reference Manual, Version*, 4, 2019.

[29] 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 (TOMS)*, 41(1):1–37, 2014.