Journal of Systems Engineering and Electronics
Vol. 28, No. 6, December 2017, pp.1162 – 1173

Optimal midcourse trajectory cluster generation and trajectory modification for hypersonic interceptions

Humin Lei[1], Jin Zhou[1,*], Dailiang Zhai[1], Lei Shao[1], and Dayuan Zhang[2]

1. Air and Missile Defense College, Air Force Engineering University, Xi'an 710051, China;
2. Unit 95948 of the PLA, Jiuquan 732750, China

Abstract: The hypersonic interception in near space is a great challenge because of the target's unpredictable trajectory, which demands the interceptors of trajectory cluster coverage of the predicted area and optimal trajectory modification capability aiming at the consistently updating predicted impact point (PIP) in the midcourse phase. A novel midcourse optimal trajectory cluster generation and trajectory modification algorithm is proposed based on the neighboring optimal control theory. Firstly, the midcourse trajectory optimization problem is introduced; the necessary conditions for the optimal control and the transversality constraints are given. Secondly, with the description of the neighboring optimal trajectory existence theory (NOTET), the neighboring optimal control (NOC) algorithm is derived by taking the second order partial derivations with the necessary conditions and transversality conditions. The revised terminal constraints are reversely integrated to the initial time and the perturbations of the co-states are further expressed with the states deviations and terminal constraints modifications. Thirdly, the simulations of two different scenarios are carried out and the results prove the effectiveness and optimality of the proposed method.

Keywords: neighboring optimal control (NOC), midcourse guidance, trajectory cluster generation, optimal trajectory modification.

DOI: 10.21629/JSEE.2017.06.14

1. Introduction

The rapid developments of the hypersonic vehicles in near space have formed severe threats to the existing air and missile defense systems. A successful hypersonic interception against these targets is a great challenge due to the following two difficulties.

(i) The potential maneuverability and the unique skipping trajectory of the hypersonic vehicles have improved the stealth ability. The location of the real target can only be described by a probability density function (PDF) [1], and the predicted impact point (PIP) may be repeatedly revised with more accurate detections from the interceptors' onboard sensors and ground radars as the interception proceeds.

(ii) The huge velocity over Mach 5 of the hypersonic vehicles has dramatically extended the striking area and shrunk the interception process at the same time. The traditional terminal guidance phase has been compressed to several seconds with little error compensation abilities [2]. If the initial states of the terminal guidance divert far from the desired ones, the hit-to-kill accuracy cannot be ensured. Thus the future interceptors against the hypersonic vehicles should develop corresponding strategies to solve the above difficulties. i) The fire unit should generate trajectory cluster to cooperatively cover the predicted reachable areas of the hypersonic vehicles to maximize the probability of interception. ii) In order to relive the burden of the terminal guidance, the aiming errors should be negated to the maximum extent in the midcourse phase to form a better handover situation such as the zero effort miss (ZEM) sets [3], which demands the online trajectory modification capability to satisfy the revised PIP constraints.

However, the research about the midcourse trajectory cluster generation and trajectory modification is relatively little. Relative literature mostly adopts the traditional trajectory optimization algorithms, such as the polynomial parameterization [4], trajectory interpolation [5], particle swarm optimization [6], and inverse dynamic in virtual domain optimization [7,8], and recalculates the nominal trajectory to satisfy the revised constraints [9 – 14]. These accomplishments achieve the trajectory regeneration at the cost of totally deserting the former nominal one and thus put a crucial burden on the computing efficiency and onboard computation. In fact, the former nominal trajectory could be used to generate a new trajectory with the neighboring optimal control (NOC) theory if the revised terminal constraints or the perturbed states are not far from the nominal ones. Tian et al. [15] used the indirect Legendre

Manuscript received May 24, 2016.
*Corresponding author.
This work was supported by the National Natural Science Foundation of China (61503408; 61573374).

pseudospectral feedback method to generate a robust state feedback guidance law, but his accomplishments only researched on the compensation of the perturbations in the initial states, and the general idea still fell into the category of trajectory following, which was unable to deal with the trajectory cluster generation and modification problem.

This paper extends the idea of [15] to cope for the terminal constraints modifications in order to meet the future advanced interceptor demands. An optimal midcourse trajectory cluster generation and trajectory modification algorithm is derived based on the NOC theory, which has the merits of high precision and fast convergence. It is shown that the time efficiency of the proposed method is about five times faster than that of the Gauss pseudospectral method (GPM) with an equivalent high precision.

The paper is organized as follows. In Section 2, the interceptor dynamic model is given, and the method used to generate the nominal optimal midcourse trajectory that satisfies the necessary optimality conditions and the trasversality conditions is described. Section 3 gives the midcourse trajectory cluster generation and optimal modification algorithm in detail. Numerical simulations of two different scenarios are given in Section 4 and Section 5 gives the conclusion.

2. Midcourse trajectory optimization problems

2.1 Interceptor dynamic model

The interceptor considered in this paper is assumed to be a rigid body. Because the optimal trajectories are designed to directly aim at the targets and avoid horizontal maneuvers to save the energy, the planner point-mass kinematic equations are considered, which can be described [16,17] as follows:

$$\dot{V} = \frac{P\cos\alpha - C_x qS}{m} - g\sin\theta \tag{1}$$

$$\dot{\theta} = \frac{P\sin\alpha + C_y qS}{mV} - \frac{g\cos\theta}{V} + \frac{V\cos\theta}{R_0 + y} \tag{2}$$

$$\dot{x} = \frac{R_0 V\cos\theta}{R_0 + y} \tag{3}$$

$$\dot{y} = V\sin\theta \tag{4}$$

where V is the velocity of the interceptor, P is the thrust value, g is the gravitational acceleration, R_0 is the radius of the earth, m is the total mass of the interceptor, θ is the flight path angle, q is the dynamic pressure, S is the reference area, x and y are the locations in the inertial coordinate frame, and C_x and C_y are the drag coefficient parameter and the lift coefficient parameter respectively, which have the mathematical forms as

$$C_y = C_{y0} + C_y^{\alpha}\alpha \tag{5}$$

$$C_x = C_{x0} + C_x^{\alpha}\alpha + C_x^{\alpha^2}\alpha^2 \tag{6}$$

where α is the angle of attack, C_{x0}, C_x^{α} and $C_x^{\alpha^2}$ are the drag coefficients, and C_{y0} and C_y^{α} are the lift coefficients.

2.2 Nominal optimal midcourse trajectory generation

As the most time-consuming phase in the interception process, the midcourse guidance is critical to a successful engagement because its final states serve as the starting conditions of the terminal guidance. If the final states of the midcourse guidance converge to the ZEM sets, the terminal interception will be ensured. However, if the final states diverge far from the desired sets, the terminal guidance will be unable to guarantee the successful interception. The objective of the midcourse guidance is to deliver the interceptor to a certain point (usually the PIP) with specified conditions to ensure a successful handover between the midcourse guidance and the terminal guidance. Since in the terminal guidance, a kinematic kill is preferred which demands the interceptor of high velocity and accurate handover precisions, the midcourse guidance objective is selected as the maximum velocity at the midcourse terminal time t_f as

$$J = \phi(V(t_f), t_f) = -V_f \tag{7}$$

which is subjected to the interceptors kinematic dynamics

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \tag{8}$$

the terminal attitude and position constraint

$$\boldsymbol{\psi} = [\theta - \theta_f \quad x - x_f \quad y - y_f]^{\mathrm{T}} = \boldsymbol{0} \tag{9}$$

the control limitation

$$\|\alpha\| \leqslant \alpha_{\max} \tag{10}$$

and the path constraints [16,18]

$$q \leqslant q_{\max} \tag{11}$$

$$\begin{cases} q_a = h_0 + h_1\alpha + h_2\alpha^2 + h_3\alpha^3 \\ q_r = C\rho^{0.5}V^{3.07} \\ \dot{Q} = q_a q_r \leqslant \dot{Q}_{\max} \end{cases} \tag{12}$$

$$n_L = \frac{\sqrt{(C_x qS)^2 + (C_y qS)^2}}{mg} \leqslant n_{L\max} \tag{13}$$

where the subscript f denotes the final value of the parameters, $\boldsymbol{0}$ denotes the zeros matrix with appropriate dimensions, J is the index criterion, $\boldsymbol{x} = [V \quad \theta \quad x \quad y]^{\mathrm{T}}$ is the system state, $\boldsymbol{f} = [\dot{V} \quad \dot{\theta} \quad \dot{x} \quad \dot{y}]^{\mathrm{T}}$ is the system dynamic equation, $\boldsymbol{u} = \boldsymbol{\alpha}$ is the control input, $\dot{Q}$ indicates the heating transfer rate at a specified point on the surface, generally the stagnation point on the nose, $\rho = \rho_0 \mathrm{e}^{-\beta h}$ is the atmospheric density with ρ_0 being the atmospheric density at the sea level, β the density coefficient constant, h the altitude from the sea level, $C = 1.783\,45\times10^{-4}\ \mathrm{J\cdot s^{2.07}/(m^{3.57}\cdot kg^{0.5})}$, $h_0 = 1.067$, $h_1 = -1.101$, $h_2 = 0.698\,8$, $h_3 = -0.190\,3$, $\alpha_{\max} = 40°$, $q_{\max} = 50\ \mathrm{kPa}$, $\dot{Q}_{\max} = 7.945\times10^5\ \mathrm{W/m^2}$, $n_{L\max} = 5$.

The problem formulated above falls to the category of the optimization problems. And the optimal control theory is the most appropriate framework for its solutions. By introducing the co-states $\boldsymbol{\lambda} = [\lambda_v \quad \lambda_\theta \quad \lambda_x \quad \lambda_y]^{\mathrm{T}}$ which have the same dimensions with the states $\boldsymbol{x}$, the Hamiltonian equations $\boldsymbol{H}$ can be formulated as

$$\boldsymbol{H} = \lambda_V\dot{V} + \lambda_\theta\dot{\theta} + \lambda_x\dot{x} + \lambda_y\dot{y} = \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{f}. \tag{14}$$

By introducing the constant Lagrange multiplier vector $\boldsymbol{\nu} = [\nu_\theta \quad \nu_x \quad \nu_y]^{\mathrm{T}}$, the augmented index criterion can be reformulated as

$$\bar{J} = \boldsymbol{\Phi}_f + \int_{t_0}^{t_f}(\boldsymbol{H} - \boldsymbol{\lambda}^{\mathrm{T}}\dot{\boldsymbol{x}})\mathrm{d}t \tag{15}$$

where the subscript 0 denotes the initial value of the parameters, and the auxiliary function $\boldsymbol{\Phi}_f$ is denoted as

$$\boldsymbol{\Phi}_f = [\phi + \boldsymbol{\nu}^{\mathrm{T}}\psi]_f. \tag{16}$$

If the final time t_f is not specified, the necessary conditions for the optimal control are given by the states and co-states dynamic equations

$$\dot{\boldsymbol{x}} = \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{\lambda}} \tag{17}$$

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{x}} \tag{18}$$

and the control optimality condition

$$\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{u}} = \boldsymbol{0} \tag{19}$$

which are subjected to the transversality conditions

$$\boldsymbol{\lambda}(t_f) = -\left(\frac{\partial \boldsymbol{\Phi}}{\partial \boldsymbol{x}}\right)_f \tag{20}$$

$$\left(\boldsymbol{H} + \frac{\partial \boldsymbol{\Phi}}{\partial t}\right)_{t=t_f} = \boldsymbol{0} \tag{21}$$

$$\boldsymbol{\psi}(\boldsymbol{x}(t_f), t_f) = \boldsymbol{0}. \tag{22}$$

From the equations above, it is clear that the necessary optimality conditions for the midcourse trajectory optimization problem result in a two point boundary value problem (TPBVP), the search for solution of which is a tedious task and out of the scope of this article. In order to be simple and representative, the nominal optimal midcourse trajectory generated by the GPM is assumed to be preliminarily acquired in the following description.

3. Design of the algorithm

Once the nominal optimal trajectory is designed, the same method could be repeatedly used to calculate the trajectory cluster for cooperative coverage of the hypersonic vehicle predicted area with respective PDF, and it can also be used to re-compute new optimal trajectories that satisfy the revised PIP constraints for the trajectory modification. However, this approach is computationally expensive and intensive because it totally abandons the nominal optimal trajectory information and searches for the optimal control solutions within relatively large appropriate sets to resolve the TPBVP. Actually, the revised constraints both in the trajectory cluster and modification situations may be not very far from the nominal ones, and a series of control modifications may be sufficient to reshape the nominal trajectory to a new optimal trajectory that meets the revised constraints.

3.1 Neighboring optimal trajectory existence theory

In order to derive the theory of neighboring optimal trajectory existence theory (NOTET), the following two reasonable assumptions are made.

Assumption 1 The dynamic system equations $\boldsymbol{f}$ in (8) is continuous in the $(t, \boldsymbol{x}, \boldsymbol{\lambda})$ space. $\boldsymbol{f}$ is locally Lipschitz in $\boldsymbol{x}$ and uniformly Lipschitz in t and $\boldsymbol{\lambda}$. $\boldsymbol{x} \in \mathbf{C} \subset \mathbf{R}^n$, $\boldsymbol{u} \in \boldsymbol{\Omega} \subset \mathbf{R}^m$, $\boldsymbol{\lambda} \in \boldsymbol{P} \subset \mathbf{R}^p$ are convex and connected sets.

Assumption 2 The nominal optimal trajectory $\boldsymbol{x}^*$ exists and the optimal nominal control command $\boldsymbol{u}^*$ is preliminarily acquired.

Theorem 1 For an actual perturbed initial states $\delta\boldsymbol{x}_0 = (\boldsymbol{x} - \boldsymbol{x}^*)_0$ or terminal constraints $\delta\boldsymbol{\psi}_f = (\boldsymbol{\psi} - \boldsymbol{\psi}^*)_f$, there exists a neighboring optimal states trajectory $\boldsymbol{x}(t)$, satisfying $\sup\limits_t \|\boldsymbol{x}(t) - \boldsymbol{x}^*(t)\| \leqslant K$, where K is the upper bound. And the neighboring optimal control $\boldsymbol{u}(t)$ is determined by $\boldsymbol{u}(t) = \boldsymbol{u}^*(t) + \delta\boldsymbol{u}$, where $\delta\boldsymbol{u} = \boldsymbol{Z}(t)[\delta\boldsymbol{x}(t) \quad \delta\boldsymbol{\lambda}(t)]^{\mathrm{T}}$.

Proof According to the Assumption 1 and Assumption 2, the neighboring optimal trajectory $\boldsymbol{x}(t)$ exists and the proof can be found in [19]. Based on the Hamilton optimal condition (19), we assume that the control is of the form $\boldsymbol{u} = \boldsymbol{g}(t, \boldsymbol{x}, \boldsymbol{\lambda})$. In the neighborhood of the nominal states $\boldsymbol{x}^*(t)$, the perturbed states can be defined as $\boldsymbol{x} = (\delta\boldsymbol{x} + \boldsymbol{x}^*)$, and the co-states can be defined as $\boldsymbol{\lambda} = (\delta\boldsymbol{\lambda} + \boldsymbol{\lambda}^*)$. By the Taylor series expansion, the actual control command can be expressed by

$$\boldsymbol{u} = \boldsymbol{u}^* + \begin{bmatrix}\dfrac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} & \dfrac{\partial \boldsymbol{g}}{\partial \boldsymbol{\lambda}}\end{bmatrix}\begin{bmatrix}\delta\boldsymbol{x}(t) \\ \delta\boldsymbol{\lambda}(t)\end{bmatrix} + \mathrm{H.O.T} \tag{23}$$

where H.O.T denotes the higher order terms that can be neglected in this problem. Thus the control command modification has the form of

$$\delta \boldsymbol{u}=\begin{bmatrix}\dfrac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} & \dfrac{\partial \boldsymbol{g}}{\partial \boldsymbol{\lambda}}\end{bmatrix}\begin{bmatrix}\delta \boldsymbol{x}(t)\\ \delta \boldsymbol{\lambda}(t)\end{bmatrix}=\boldsymbol{Z}(t)\begin{bmatrix}\delta \boldsymbol{x}(t)\\ \delta \boldsymbol{\lambda}(t)\end{bmatrix} \tag{24}$$

where $\boldsymbol{Z}(t)=\begin{bmatrix}\dfrac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} & \dfrac{\partial \boldsymbol{g}}{\partial \boldsymbol{\lambda}}\end{bmatrix}$.

In order to get the deviation dynamics, the necessary optimal conditions (17) – (19) are further differentiated to the second order

$$\delta \dot{\boldsymbol{x}}=\frac{\partial^2 \boldsymbol{H}}{\partial \boldsymbol{\lambda}\partial \boldsymbol{x}}\delta \boldsymbol{x}+\frac{\partial^2 \boldsymbol{H}}{\partial \boldsymbol{\lambda}\partial \boldsymbol{u}}\delta \boldsymbol{u} \tag{25}$$

$$\delta \dot{\boldsymbol{\lambda}}=-\frac{\partial^2 \boldsymbol{H}}{\partial \boldsymbol{x}^2}\delta \boldsymbol{x}-\frac{\partial^2 \boldsymbol{H}}{\partial \boldsymbol{x}\partial \boldsymbol{\lambda}}\delta \boldsymbol{\lambda}-\frac{\partial^2 \boldsymbol{H}}{\partial \boldsymbol{x}\partial \boldsymbol{u}}\delta \boldsymbol{u} \tag{26}$$

$$\boldsymbol{0}=\frac{\partial^2 \boldsymbol{H}}{\partial \boldsymbol{u}\partial \boldsymbol{x}}\delta \boldsymbol{x}+\frac{\partial^2 \boldsymbol{H}}{\partial \boldsymbol{u}\partial \boldsymbol{\lambda}}\delta \boldsymbol{\lambda}+\frac{\partial^2 \boldsymbol{H}}{\partial \boldsymbol{u}^2}\delta \boldsymbol{u}. \tag{27}$$

If $\partial^2 \boldsymbol{H}/\partial \boldsymbol{u}^2$ is non-singular over the entire midcourse process, the control modifications $\delta \boldsymbol{u}$ could be solved from (27) as

$$\delta \boldsymbol{u}=-\left(\frac{\partial^2 \boldsymbol{H}}{\partial \boldsymbol{u}^2}\right)^{-1}\left(\frac{\partial^2 \boldsymbol{H}}{\partial \boldsymbol{u}\partial \boldsymbol{x}}\delta \boldsymbol{x}+\frac{\partial^2 \boldsymbol{H}}{\partial \boldsymbol{u}\partial \boldsymbol{\lambda}}\delta \boldsymbol{\lambda}\right). \tag{28}$$

Then $\boldsymbol{Z}(t)$ has the form of the following equation:

$$\boldsymbol{Z}(t)=-\left(\frac{\partial^2 \boldsymbol{H}}{\partial \boldsymbol{u}^2}\right)^{-1}\begin{bmatrix}\dfrac{\partial^2 \boldsymbol{H}}{\partial \boldsymbol{u}\partial \boldsymbol{x}} & \dfrac{\partial^2 \boldsymbol{H}}{\partial \boldsymbol{u}\partial \lambda}\end{bmatrix}. \tag{29}$$

3.2 NOC algorithm

From (28), it is apparent that, the calculation of the control modifications $\delta \boldsymbol{u}$ demands the information of the states perturbations $\delta \boldsymbol{x}$ and the co-states perturbations $\delta \boldsymbol{\lambda}$. $\delta \boldsymbol{x}$ could be acquired with the onboard sensors by comparing the actual states with the nominal ones. However, the co-states perturbations $\delta \boldsymbol{\lambda}$ can hardly be measured or precisely estimated. The revisions of the terminal constraints $\delta \boldsymbol{\psi}_f$ does not appear in (28), so the expression of the co-states perturbations must include the accessible states deviations $\delta \boldsymbol{x}$ and the terminal constraints modifications $\delta \boldsymbol{\psi}_f$. Firstly, the transversality conditions (20) – (22) are further differentiated to get the expressions of $\delta \boldsymbol{\psi}_f$.

$$\mathrm{d}\boldsymbol{\lambda}(t_f)=\left(\frac{\partial^2 \phi}{\partial \boldsymbol{x}^2}+\boldsymbol{\nu}^{\mathrm{T}}\frac{\partial^2 \boldsymbol{\psi}}{\partial \boldsymbol{x}^2}\right)\mathrm{d}\boldsymbol{x}+ \left(\frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}}\right)^{\mathrm{T}}\delta \boldsymbol{\nu}+\left(\frac{\partial^2 \phi}{\partial \boldsymbol{x}\partial t}+\boldsymbol{\nu}^{\mathrm{T}}\frac{\partial^2 \boldsymbol{\psi}}{\partial \boldsymbol{x}\partial t}\right)\mathrm{d}t_f \tag{30}$$

$$\left(\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{x}}+\frac{\partial^2 \phi}{\partial t\partial \boldsymbol{x}}+\boldsymbol{\nu}^{\mathrm{T}}\frac{\partial^2 \boldsymbol{\psi}}{\partial t\partial \boldsymbol{x}}\right)\mathrm{d}\boldsymbol{x}+\left(\frac{\partial \boldsymbol{\psi}}{\partial t}\right)^{\mathrm{T}}\delta \boldsymbol{\nu}+ \left(\frac{\partial \boldsymbol{H}}{\partial t}+\frac{\partial^2 \phi}{\partial t^2}+\boldsymbol{\nu}^{\mathrm{T}}\frac{\partial^2 \boldsymbol{\psi}}{\partial t^2}\right)\mathrm{d}t_f+ \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{\lambda}}\mathrm{d}\boldsymbol{\lambda}+\frac{\partial \boldsymbol{H}}{\partial u}\mathrm{d}u=\boldsymbol{0} \tag{31}$$

$$\delta \boldsymbol{\psi}=\frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}}\mathrm{d}\boldsymbol{x}+\frac{\partial \boldsymbol{\psi}}{\partial t}\mathrm{d}t_f. \tag{32}$$

Taking (28) into (25) and (26), the second order dynamics can be expressed as

$$\delta \dot{\boldsymbol{x}}=\boldsymbol{A}(t)\delta \boldsymbol{x}-\boldsymbol{B}(t)\delta \boldsymbol{\lambda} \tag{33}$$

$$\delta \dot{\boldsymbol{\lambda}}=-\mathbf{C}(t)\delta \boldsymbol{x}-\boldsymbol{A}^{\mathrm{T}}(t)\delta \boldsymbol{\lambda} \tag{34}$$

where

$$\boldsymbol{A}(t)=\frac{\partial^2 \boldsymbol{H}}{\partial \boldsymbol{\lambda}\partial \boldsymbol{x}}-\frac{\partial^2 \boldsymbol{H}}{\partial \boldsymbol{\lambda}\partial \boldsymbol{u}}\left(\frac{\partial^2 \boldsymbol{H}}{\partial \boldsymbol{u}^2}\right)^{-1}\frac{\partial^2 \boldsymbol{H}}{\partial \boldsymbol{u}\partial \boldsymbol{x}} \tag{35}$$

$$\boldsymbol{B}(t)=\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{\lambda}\partial \boldsymbol{u}}\left(\frac{\partial^2 \boldsymbol{H}}{\partial \boldsymbol{u}^2}\right)^{-1}\frac{\partial^2 \boldsymbol{H}}{\partial \boldsymbol{u}\partial \boldsymbol{\lambda}} \tag{36}$$

$$\boldsymbol{C}(t)=\frac{\partial^2 \boldsymbol{H}}{\partial \boldsymbol{x}^2}-\frac{\partial^2 \boldsymbol{H}}{\partial \boldsymbol{x}\partial \boldsymbol{u}}\left(\frac{\partial^2 \boldsymbol{H}}{\partial \boldsymbol{u}^2}\right)^{-1}\frac{\partial^2 \boldsymbol{H}}{\partial \boldsymbol{u}\partial \boldsymbol{x}}. \tag{37}$$

Noticing the difference between $\mathrm{d}\boldsymbol{\lambda}(t_f)$ and $\delta \boldsymbol{\lambda}(t_f)$, $\mathrm{d}\boldsymbol{x}(t_f)$ and $\delta \boldsymbol{x}(t_f)$ when the final time t_f is free, which can be described with the following equations:

$$\mathrm{d}\boldsymbol{\lambda}(t_f)=\delta \boldsymbol{\lambda}(t_f)+\dot{\boldsymbol{\lambda}}(t_f)\mathrm{d}t_f \tag{38}$$

$$\mathrm{d}\boldsymbol{x}(t_f)=\delta \boldsymbol{x}(t_f)+\dot{\boldsymbol{x}}(t_f)\mathrm{d}t_f. \tag{39}$$

With (38) and (39), (30) – (32) could be simplified as

$$\delta \boldsymbol{\lambda}(t_f)=\left(\frac{\partial^2 \phi}{\partial \boldsymbol{x}^2}+\boldsymbol{\nu}^{\mathrm{T}}\frac{\partial^2 \boldsymbol{\psi}}{\partial \boldsymbol{x}^2}\right)\delta \boldsymbol{x}+\left(\frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}}\right)^{\mathrm{T}}\delta \boldsymbol{\nu}+ \left(\frac{\partial^2 \phi}{\partial \boldsymbol{x}\partial t}+\boldsymbol{\nu}^{\mathrm{T}}\frac{\partial^2 \boldsymbol{\psi}}{\partial \boldsymbol{x}\partial t}+\left(\frac{\partial^2 \phi}{\partial \boldsymbol{x}^2}+\boldsymbol{\nu}^{\mathrm{T}}\frac{\partial^2 \boldsymbol{\psi}}{\partial \boldsymbol{x}^2}\right)\dot{\boldsymbol{x}}-\dot{\boldsymbol{\lambda}}\right)\mathrm{d}t_f \tag{40}$$

$$\left(-\dot{\boldsymbol{\lambda}}+\frac{\partial^2 \phi}{\partial t\partial \boldsymbol{x}}+\boldsymbol{\nu}^{\mathrm{T}}\frac{\partial^2 \boldsymbol{\psi}}{\partial t\partial \boldsymbol{x}}\right)(\delta \boldsymbol{x}+\dot{\boldsymbol{x}}\mathrm{d}t_f)+ \left(\frac{\partial \boldsymbol{\psi}}{\partial t}\right)^{\mathrm{T}}\delta \boldsymbol{\nu}+\left(\frac{\partial^2 \phi}{\partial t^2}+\boldsymbol{\nu}^{\mathrm{T}}\frac{\partial^2 \boldsymbol{\psi}}{\partial t^2}\right)\mathrm{d}t_f+ \dot{\boldsymbol{x}}\left(\delta \boldsymbol{\lambda}+\dot{\boldsymbol{\lambda}}\mathrm{d}t_f\right)=\boldsymbol{0} \tag{41}$$

$$\delta \boldsymbol{\psi}=\frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}}\delta \boldsymbol{x}+\left(\frac{\partial \boldsymbol{\psi}}{\partial t}+\frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}}\right)\mathrm{d}t_f. \tag{42}$$

Taking the expression of $\delta \boldsymbol{\lambda}(t_f)$ in (40) into (41), one can get

$$\boldsymbol{0}=\left(-\dot{\boldsymbol{\lambda}}+\frac{\partial^2 \phi}{\partial t\partial \boldsymbol{x}}+\boldsymbol{\nu}^{\mathrm{T}}\frac{\partial^2 \boldsymbol{\psi}}{\partial t\partial \boldsymbol{x}}\right)(\delta \boldsymbol{x}+\dot{\boldsymbol{x}}\mathrm{d}t_f)+ \left(\frac{\partial \boldsymbol{\psi}}{\partial t}\right)^{\mathrm{T}}\delta \boldsymbol{\nu}+\left(\frac{\partial^2 \phi}{\partial t^2}+\boldsymbol{\nu}^{\mathrm{T}}\frac{\partial^2 \boldsymbol{\psi}}{\partial t^2}\right)\mathrm{d}t_f+\dot{\boldsymbol{x}}\dot{\boldsymbol{\lambda}}\mathrm{d}t_f+ \dot{\boldsymbol{x}}\Bigg(\left(\frac{\partial^2 \phi}{\partial \boldsymbol{x}^2}+\boldsymbol{\nu}^{\mathrm{T}}\frac{\partial^2 \boldsymbol{\psi}}{\partial \boldsymbol{x}^2}\right)\delta \boldsymbol{x}+\left(\frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}}\right)^{\mathrm{T}}\delta \boldsymbol{\nu}+ \left(\frac{\partial^2 \phi}{\partial \boldsymbol{x}\partial t}+\boldsymbol{\nu}^{\mathrm{T}}\frac{\partial^2 \boldsymbol{\psi}}{\partial \boldsymbol{x}\partial t}+\left(\frac{\partial^2 \phi}{\partial \boldsymbol{x}^2}+\boldsymbol{\nu}^{\mathrm{T}}\frac{\partial^2 \boldsymbol{\psi}}{\partial \boldsymbol{x}^2}\right)\dot{\boldsymbol{x}}-\dot{\boldsymbol{\lambda}}\right)\mathrm{d}t_f\Bigg). \tag{43}$$

Rewrite (40), (42) and (43) into a matrix form, one can get

$$\begin{bmatrix} \delta\boldsymbol{\lambda}(t_f) \\ \delta\boldsymbol{\psi} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial^2\phi}{\partial\boldsymbol{x}^2}+\boldsymbol{\nu}^{\mathrm{T}}\dfrac{\partial^2\boldsymbol{\psi}}{\partial\boldsymbol{x}^2} & \left(\dfrac{\partial\boldsymbol{\psi}}{\partial\boldsymbol{x}}\right)^{\mathrm{T}} & \dfrac{\partial^2\phi}{\partial\boldsymbol{x}\partial t}+\boldsymbol{\nu}^{\mathrm{T}}\dfrac{\partial^2\boldsymbol{\psi}}{\partial\boldsymbol{x}\partial t}+\left(\dfrac{\partial^2\phi}{\partial\boldsymbol{x}^2}+\boldsymbol{\nu}^{\mathrm{T}}\dfrac{\partial^2\boldsymbol{\psi}}{\partial\boldsymbol{x}^2}\right)\dot{\boldsymbol{x}}-\dot{\boldsymbol{\lambda}} \\ \dfrac{\partial\boldsymbol{\psi}}{\partial\boldsymbol{x}} & \mathbf{0} & \dfrac{\partial\boldsymbol{\psi}}{\partial t}+\left(\dfrac{\partial\boldsymbol{\psi}}{\partial\boldsymbol{x}}\right)^{\mathrm{T}}\dot{\boldsymbol{x}} \\ -\dot{\boldsymbol{\lambda}}+\dfrac{\partial^2\phi}{\partial t\partial\boldsymbol{x}}+\boldsymbol{\nu}^{\mathrm{T}}\dfrac{\partial^2\boldsymbol{\psi}}{\partial t\partial\boldsymbol{x}}+\dot{\boldsymbol{x}}\left(\dfrac{\partial^2\phi}{\partial\boldsymbol{x}^2}+\boldsymbol{\nu}^{\mathrm{T}}\dfrac{\partial^2\boldsymbol{\psi}}{\partial\boldsymbol{x}^2}\right) & \left(\dfrac{\partial\boldsymbol{\psi}}{\partial t}\right)^{\mathrm{T}}+\dot{\boldsymbol{x}}^{\mathrm{T}}\left(\dfrac{\partial\boldsymbol{\psi}}{\partial\boldsymbol{x}}\right) & \dot{\boldsymbol{x}}^{\mathrm{T}}\left(\dfrac{\partial^2\phi}{\partial\boldsymbol{x}^2}+\boldsymbol{\nu}^{\mathrm{T}}\dfrac{\partial^2\boldsymbol{\psi}}{\partial\boldsymbol{x}^2}\right)\dot{\boldsymbol{x}}+2\dot{\boldsymbol{x}}^{\mathrm{T}}\left(\dfrac{\partial^2\phi}{\partial\boldsymbol{x}\partial t}+\boldsymbol{\nu}^{\mathrm{T}}\dfrac{\partial^2\boldsymbol{\psi}}{\partial\boldsymbol{x}\partial t}\right)-\dot{\boldsymbol{x}}^{\mathrm{T}}\dot{\boldsymbol{\lambda}}+\dfrac{\partial^2\phi}{\partial t^2}+\boldsymbol{\nu}^{\mathrm{T}}\dfrac{\partial^2\boldsymbol{\psi}}{\partial t^2} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{x}(t_f) \\ \delta\boldsymbol{\nu} \\ \mathrm{d}t_f \end{bmatrix}. \tag{44}$$

Noticing the symmetry of (44), $[\delta\boldsymbol{\lambda} \quad \delta\boldsymbol{\psi} \quad \mathbf{0}]^{\mathrm{T}}$ could be assumed to have the following expression:

$$\begin{bmatrix} \delta\boldsymbol{\lambda} \\ \delta\boldsymbol{\psi} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{S} & \boldsymbol{R} & \boldsymbol{m} \\ \boldsymbol{R}^{\mathrm{T}} & \boldsymbol{Q} & \boldsymbol{n} \\ \boldsymbol{m}^{\mathrm{T}} & \boldsymbol{n}^{\mathrm{T}} & \boldsymbol{\alpha} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{x} \\ \delta\boldsymbol{\nu} \\ \mathrm{d}t_f \end{bmatrix} \tag{45}$$

where $\boldsymbol{S},\boldsymbol{R},\boldsymbol{Q},\boldsymbol{m},\boldsymbol{n}$, and $\boldsymbol{\alpha}$ are time varying parameters, whose terminal values are given with (44):

$$\boldsymbol{S}_f = \frac{\partial^2\phi}{\partial\boldsymbol{x}^2}+\boldsymbol{\nu}^{\mathrm{T}}\frac{\partial^2\boldsymbol{\psi}}{\partial\boldsymbol{x}^2} \tag{46}$$

$$\boldsymbol{R}_f = \left(\frac{\partial\psi}{\partial\boldsymbol{x}}\right)^{\mathrm{T}} \tag{47}$$

$$\boldsymbol{Q}_f = \mathbf{0} \tag{48}$$

$$\boldsymbol{m}_f = \frac{\partial^2\phi}{\partial\boldsymbol{x}\partial t}+\boldsymbol{\nu}^{\mathrm{T}}\frac{\partial^2\boldsymbol{\psi}}{\partial\boldsymbol{x}\partial t}+ \left(\frac{\partial^2\phi}{\partial\boldsymbol{x}^2}+\boldsymbol{\nu}^{\mathrm{T}}\frac{\partial^2\boldsymbol{\psi}}{\partial\boldsymbol{x}^2}\right)\dot{\boldsymbol{x}}-\dot{\boldsymbol{\lambda}} \tag{49}$$

$$\boldsymbol{n}_f = \frac{\partial\boldsymbol{\psi}}{\partial t}+\left(\frac{\partial\boldsymbol{\psi}}{\partial\boldsymbol{x}}\right)^{\mathrm{T}}\dot{\boldsymbol{x}} \tag{50}$$

$$\boldsymbol{\alpha}_f = \dot{\boldsymbol{x}}^{\mathrm{T}}\left(\frac{\partial^2\phi}{\partial\boldsymbol{x}^2}+\boldsymbol{\nu}^{\mathrm{T}}\frac{\partial^2\boldsymbol{\psi}}{\partial\boldsymbol{x}^2}\right)\dot{\boldsymbol{x}}+ 2\dot{\boldsymbol{x}}^{\mathrm{T}}\left(\frac{\partial^2\phi}{\partial\boldsymbol{x}\partial t}+\boldsymbol{\nu}^{\mathrm{T}}\frac{\partial^2\boldsymbol{\psi}}{\partial\boldsymbol{x}\partial t}\right)-\dot{\boldsymbol{x}}^{\mathrm{T}}\dot{\boldsymbol{\lambda}}+ \frac{\partial^2\phi}{\partial t^2}+\boldsymbol{\nu}^{\mathrm{T}}\frac{\partial^2\boldsymbol{\psi}}{\partial t^2}. \tag{51}$$

Differentiating (45) with time, one can get

$$\delta\dot{\boldsymbol{\lambda}} = \dot{\boldsymbol{S}}\delta\boldsymbol{x}+\dot{\boldsymbol{R}}\delta\boldsymbol{\nu}+\dot{\boldsymbol{m}}\mathrm{d}t_f+\boldsymbol{S}\delta\dot{\boldsymbol{x}} \tag{52}$$

$$\mathbf{0} = \dot{\boldsymbol{R}}^{\mathrm{T}}\delta\boldsymbol{x}+\dot{\boldsymbol{Q}}\delta\boldsymbol{\nu}+\dot{\boldsymbol{n}}\mathrm{d}t_f+\boldsymbol{R}^{\mathrm{T}}\delta\dot{\boldsymbol{x}} \tag{53}$$

$$\mathbf{0} = \dot{\boldsymbol{m}}^{\mathrm{T}}\delta\boldsymbol{x}+\dot{\boldsymbol{n}}^{\mathrm{T}}\delta\boldsymbol{\nu}+\dot{\boldsymbol{\alpha}}\mathrm{d}t_f+\boldsymbol{m}^{\mathrm{T}}\delta\dot{\boldsymbol{x}}. \tag{54}$$

Taking (33) and (45) into (52), one can get

$$\delta\dot{\boldsymbol{\lambda}} = \dot{\boldsymbol{S}}\delta\boldsymbol{x}+\dot{\boldsymbol{R}}\delta\boldsymbol{\nu}+\dot{\boldsymbol{m}}\mathrm{d}t_f+ \boldsymbol{S}(\boldsymbol{A}\delta\boldsymbol{x}-\boldsymbol{B}(\boldsymbol{S}\delta\boldsymbol{x}+\mathbf{R}\delta\boldsymbol{\nu}+\boldsymbol{m}\mathrm{d}t_f)). \tag{55}$$

Taking (45) into (34), $\delta\dot{\boldsymbol{\lambda}}$ can also be denoted as

$$\delta\dot{\boldsymbol{\lambda}} = -\boldsymbol{C}\delta\boldsymbol{x}-\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{S}\delta\boldsymbol{x}+\mathbf{R}\delta\boldsymbol{\nu}+\boldsymbol{m}\mathrm{d}t_f). \tag{56}$$

Comparing (55) and (56), the following dynamic equations can be acquired.

$$\dot{\boldsymbol{S}} = -\boldsymbol{S}\boldsymbol{A}-\boldsymbol{A}^{\mathrm{T}}\boldsymbol{S}+\boldsymbol{S}\boldsymbol{B}\boldsymbol{S}-\boldsymbol{C} \tag{57}$$

$$\dot{\boldsymbol{R}} = -(\boldsymbol{A}^{\mathrm{T}}-\boldsymbol{S}\boldsymbol{B})\mathbf{R} \tag{58}$$

$$\dot{\boldsymbol{m}} = -(\boldsymbol{A}^{\mathrm{T}}-\boldsymbol{S}\boldsymbol{B})\boldsymbol{m}. \tag{59}$$

Taking (45) and (33) into (53), one can get

$$\mathbf{0} = \dot{\boldsymbol{R}}^{\mathrm{T}}\delta\boldsymbol{x}+\dot{\boldsymbol{Q}}\delta\boldsymbol{\nu}+\dot{\boldsymbol{n}}\mathrm{d}t_f+ \boldsymbol{R}^{\mathrm{T}}(\boldsymbol{A}\delta\boldsymbol{x}-\boldsymbol{B}(\boldsymbol{S}\delta\boldsymbol{x}+\mathbf{R}\delta\boldsymbol{\nu}+\boldsymbol{m}\mathrm{d}t_f)). \tag{60}$$

In order to make (60) be held at any condition, the following equations must be satisfied:

$$\dot{\boldsymbol{R}}^{\mathrm{T}} = -\boldsymbol{R}^{\mathrm{T}}(\boldsymbol{A}-\boldsymbol{B}\boldsymbol{S}) \tag{61}$$

$$\dot{\boldsymbol{Q}} = \boldsymbol{R}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{R} \tag{62}$$

$$\dot{\boldsymbol{n}} = \boldsymbol{R}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{m}. \tag{63}$$

Taking (45) and (33) into (54), one can get

$$\mathbf{0} = \dot{\boldsymbol{m}}^{\mathrm{T}}\delta\boldsymbol{x}+\dot{\boldsymbol{n}}^{\mathrm{T}}\delta\boldsymbol{\nu}+\dot{\boldsymbol{\alpha}}\mathrm{d}t_f+ \boldsymbol{m}^{\mathrm{T}}(\boldsymbol{A}\delta\boldsymbol{x}-\boldsymbol{B}(\boldsymbol{S}\delta\boldsymbol{x}+\mathbf{R}\delta\boldsymbol{\nu}+\boldsymbol{m}\mathrm{d}t_f)) \tag{64}$$

and the following dynamic equations

$$\dot{\boldsymbol{m}}^{\mathrm{T}} = -\boldsymbol{m}^{\mathrm{T}}\boldsymbol{A}+\boldsymbol{m}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{S} \tag{65}$$

$$\dot{\boldsymbol{n}}^{\mathrm{T}} = \boldsymbol{m}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{R} \tag{66}$$

$$\dot{\boldsymbol{\alpha}} = \boldsymbol{m}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{m}. \tag{67}$$

With the terminal values denoted as (46)–(51), the parameters $\boldsymbol{S}, \boldsymbol{R}, \boldsymbol{Q}, \boldsymbol{m}, \boldsymbol{n}$, and $\boldsymbol{\alpha}$ can be inversely integrated to the initial time t_0. From (53) and (54), one can get the expression of $\delta\boldsymbol{\nu}$ and $\mathrm{d}t_f$ with the initial parameters

$$\delta\boldsymbol{\nu} = \left(\left(\boldsymbol{Q} - \frac{\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}}{\boldsymbol{\alpha}}\right)^{-1}\left(\delta\psi_f - \left(\boldsymbol{R}^{\mathrm{T}} - \frac{\boldsymbol{n}\boldsymbol{m}^{\mathrm{T}}}{\boldsymbol{\alpha}}\right)\delta\boldsymbol{x}\right)\right)_{t_0} \tag{68}$$

$$\mathrm{d}t_f = \left(-\frac{\boldsymbol{n}^{\mathrm{T}}}{\boldsymbol{\alpha}}\left(\boldsymbol{Q} - \frac{\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}}{\boldsymbol{\alpha}}\right)^{-1}\delta\psi_f - \left(\frac{\boldsymbol{m}^{\mathrm{T}}}{\boldsymbol{\alpha}} - \frac{\boldsymbol{n}^{\mathrm{T}}}{\boldsymbol{\alpha}}\left(\boldsymbol{Q} - \frac{\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}}{\boldsymbol{\alpha}}\right)^{-1}\left(\boldsymbol{R}^{\mathrm{T}} - \frac{\boldsymbol{n}\boldsymbol{m}^{\mathrm{T}}}{\boldsymbol{\alpha}}\right)\right)\delta\boldsymbol{x}\right)_{t_0}. \tag{69}$$

Taking (68) and (69) into (52), the initial value of $\delta\boldsymbol{\lambda}$ could be denoted as

$$\delta\boldsymbol{\lambda}_0 = \left(\boldsymbol{R} - \frac{\boldsymbol{m}\boldsymbol{n}^{\mathrm{T}}}{\boldsymbol{\alpha}}\right)\left(\boldsymbol{Q} - \frac{\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}}{\boldsymbol{\alpha}}\right)^{-1}\delta\psi_f - \left(\boldsymbol{R} - \frac{\boldsymbol{m}\boldsymbol{n}^{\mathrm{T}}}{\boldsymbol{\alpha}}\right)\left(\boldsymbol{Q} - \frac{\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}}{\boldsymbol{\alpha}}\right)^{-1}\left(\boldsymbol{R}^{\mathrm{T}} - \frac{\boldsymbol{n}\boldsymbol{m}^{\mathrm{T}}}{\boldsymbol{\alpha}}\right)\delta\boldsymbol{x}_0 + \left(\boldsymbol{S} - \frac{\boldsymbol{m}\boldsymbol{m}^{\mathrm{T}}}{\boldsymbol{\alpha}}\right)\delta\boldsymbol{x}_0. \tag{70}$$

With the initial value $\delta\boldsymbol{x}_0$, $\delta\boldsymbol{\lambda}_0$ and the dynamic equations (33) and (34), one can integrate forward to get the value of $\delta\boldsymbol{x}$ and $\delta\boldsymbol{\lambda}$ as a function of time, then taking $\delta\boldsymbol{x}$ and $\delta\boldsymbol{\lambda}$ into (28), the control modifications $\delta\boldsymbol{u}$ can be acquired. Noticing that the final time has been modified with $\mathrm{d}t_f$ from (69), the ratio p of the revised final time and the nominal final time could be expressed as

$$p = \frac{t_f^* + \mathrm{d}t_f}{t_f^*}. \tag{71}$$

If the control command is given at a time step of Δt^*, the revised time step should be given as

$$\Delta t = p\Delta t^*. \tag{72}$$

The prosed algorithm could be used with the following steps.

Step 1 Compare the revised terminal constraints with the nominal trajectory, and calculate the modifications $\delta\psi_f$. Compare the initial states perturbations $\delta\boldsymbol{x}_0 = (\boldsymbol{x} - \boldsymbol{x}^*)_0$.

Step 2 Calculate the terminal values of the parameters $\boldsymbol{S}_f, \boldsymbol{R}_f, \boldsymbol{Q}_f, \boldsymbol{m}_f, \boldsymbol{n}_f$, and $\boldsymbol{\alpha}_f$ with (46)–(51), and reversely integrate them to the initial time to get $\boldsymbol{S}_0, \boldsymbol{R}_0, \boldsymbol{Q}_0, \boldsymbol{m}_0, \boldsymbol{n}_0$, and $\boldsymbol{\alpha}_0$.

Step 3 Take the parameters $\delta\boldsymbol{x}_0, \delta\psi_f, \boldsymbol{S}_0, \boldsymbol{R}_0, \boldsymbol{Q}_0, \boldsymbol{m}_0, \boldsymbol{n}_0$, and $\boldsymbol{\alpha}_0$ into (68)–(70) to get the value of $\delta\boldsymbol{\lambda}_0$, $\delta\boldsymbol{\nu}$, and $\mathrm{d}t_f$.

Step 4 Take the initial states perturbations $\delta\boldsymbol{x}_0$ and co-states perturbations $\delta\boldsymbol{\lambda}_0$ into (33)–(34), and integrate forward to get the perturbations $\delta\boldsymbol{x}$ and $\delta\boldsymbol{\lambda}$ expressed with a function of time.

Step 5 Take the perturbations into (28) to get the control modifications $\delta\boldsymbol{u}$.

Step 6 Calculate the modified time and the ratio p with (71), the revised step could be acquired with (72). The new control commands are composed of the control modifications and the nominal control commands, which are taken as inputs with a revised time step Δt.

4. Simulations and analysis

In order to prove the effectiveness of the proposed NOC algorithm, the following two simulations are carried out. The midcourse guidance initiates after the primary guidance. The interceptor should firstly climb to a high altitude to save the energy, and then dive to the engagement area, which is typically the parabolic trajectory. In order to be simple and representative, the simulation begins at the diving phase with the initial altitude being 80 km and the flight path angle being –1°.

4.1 Trajectory cluster generation

The predicted impact area with respective PDF is assumed to be a circle with its center located at (500 km, 35 km) in the inertial frame, and the radius is estimated to be 3 km. The trajectory cluster is designed to effectively cover the area as depicted in Fig. 1.

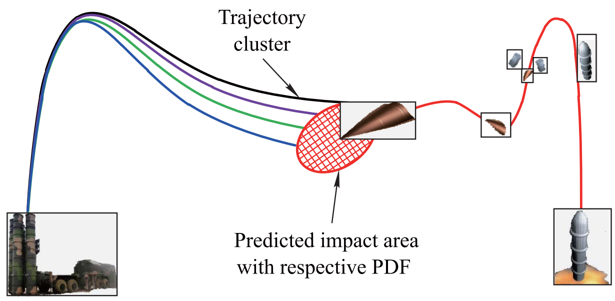

Fig. 1 Trajectory cluster that covers the targets predicted impact area

The nominal trajectory is designed to aim at the center of the area, and the nominal final flight path angle θ_f is designed as $-5°$ to acquire a better detection attitude, with the velocity to be maximized. The other eight trajectories are designed to cooperatively form a circle to cover the area. The terminal locations of the trajectory cluster are listed in Table 1, noticing that the flight path angle and the terminal velocity are preferred to be held with the nominal trajectory.

As stated above, the difference of the design objectives of trajectory cluster lies in the terminal location constraint ψ. The terminal constraints modifications are calculated with $\delta\psi_f = (\psi - \psi^*)_f$, and the initial states perturbations are set as $\delta x_0 = \mathbf{0}$.

The simulation results are given in Fig. 2 – Fig. 8. The position errors compared with the predesigned locations, the velocity and the flight path angle deviations compared with the nominal trajectory are given in Table 2.

Table 1 Terminal locations of the trajectory cluster

Trajectory number	x/m	y/m
Tra 1	500 000.000	38 000.000
Tra 2	497 878.680	37 121.320
Tra 3	497 000.000	35 000.000
Tra 4	497 878.680	32 878.680
Tra 5	500 000.000	32 000.000
Tra 6	502 121.320	32 878.680
Tra 7	503 000.000	35 000.000
Tra 8	502 121.320	37 121.320

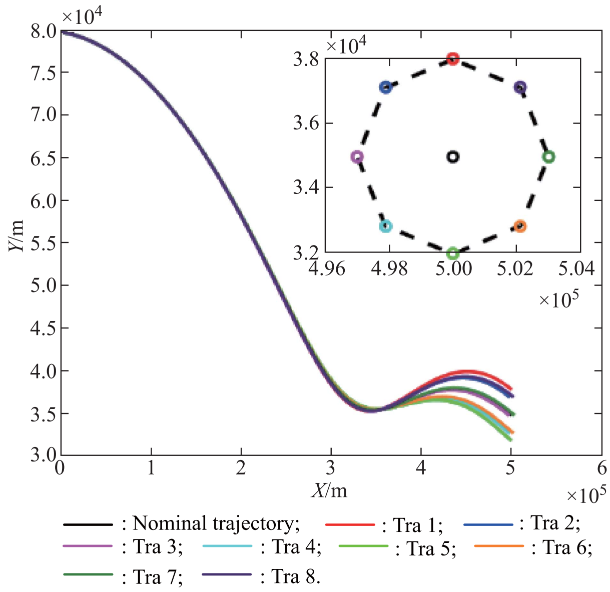

Fig. 2 Trajectory cluster

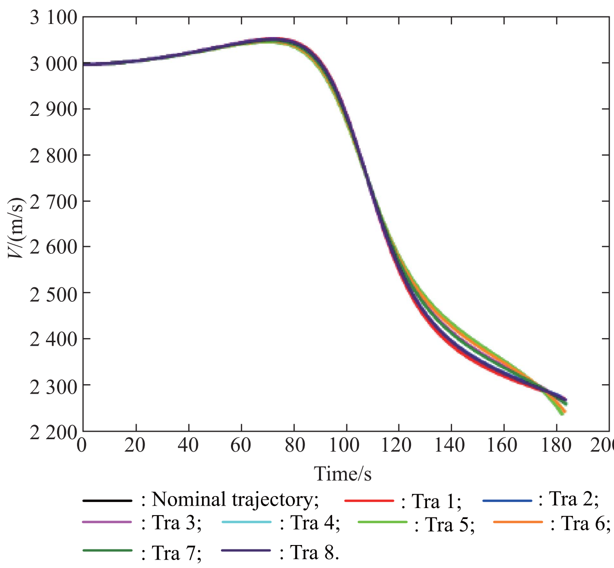

Fig. 3 Curves of the velocity

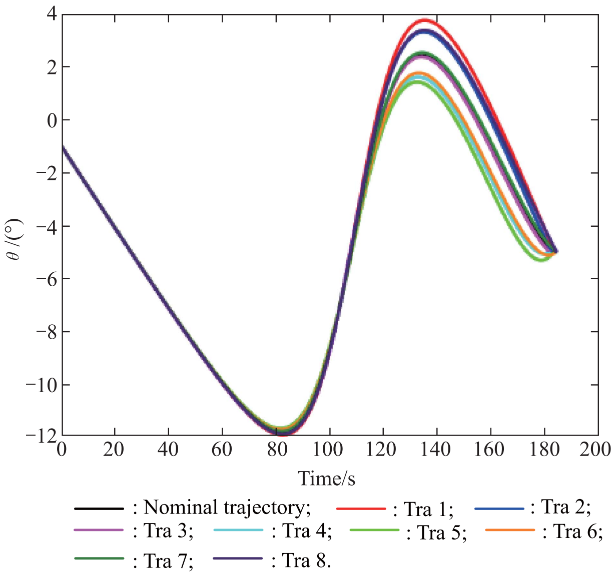

Fig. 4 Curves of the flight path angle

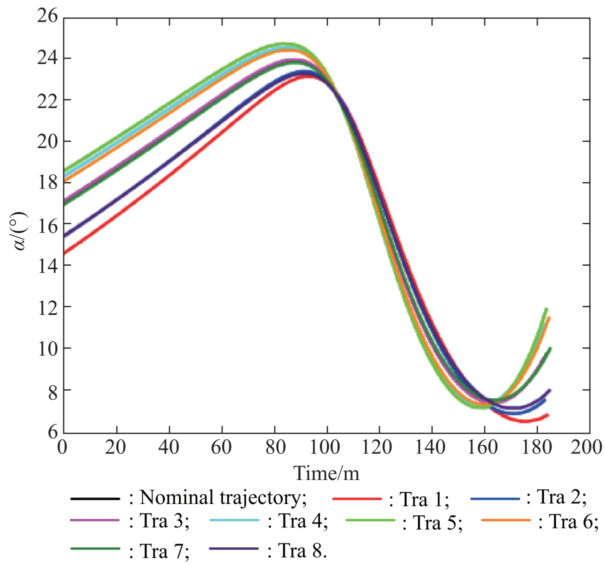

Fig. 5 Curves of the control command

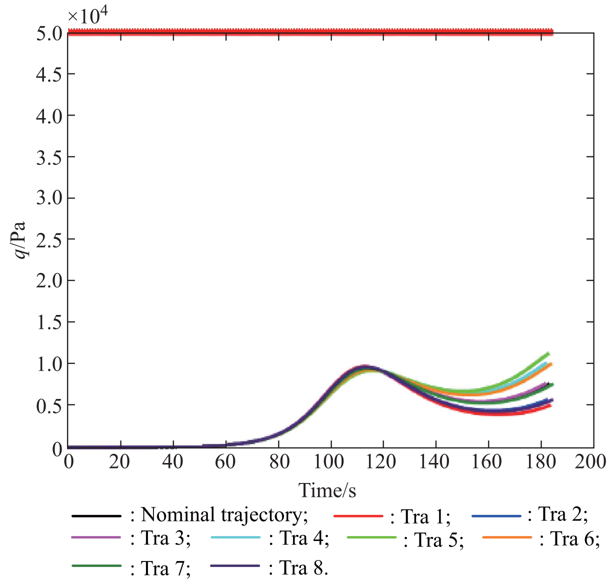

Fig. 6 Curves of the dynamic pressure

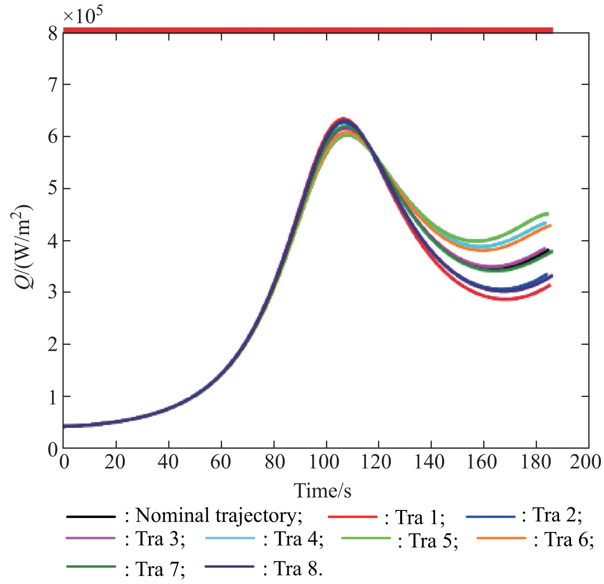

Fig. 7 Curves of the heating transfer rate

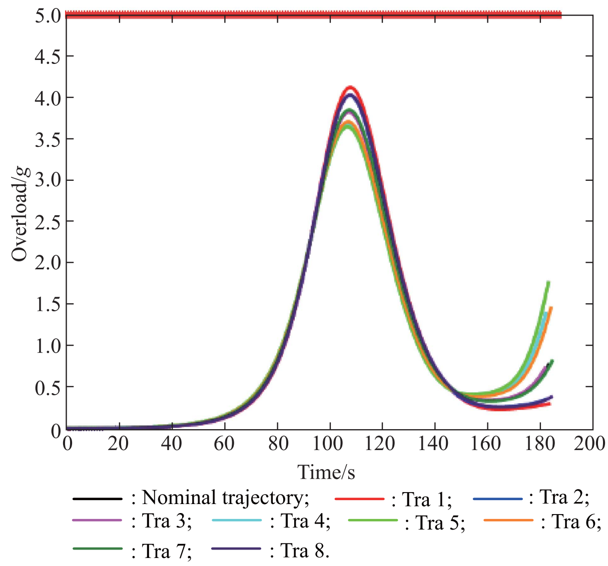

Fig. 8 Curves of the overload

Table 2 Positions errors, velocity and flight path deviations

Trajectory number	Δx/m	Δy/m	ΔV/(m/s)	$\Delta\theta/(°)$
Tra 1	−363.177	118.299	7.930	0.140
Tra 2	−356.804	117.934	10.926	0.123
Tra 3	−348.757	117.955	5.375	0.061
Tra 4	−339.921	118.226	−13.641	−0.029
Tra 5	−338.532	118.687	−28.165	−0.080
Tra 6	−349.112	119.165	−22.020	−0.040
Tra 7	−362.402	119.319	−5.328	0.047
Tra 8	−367.036	118.934	4.558	0.114

Fig. 2 gives the trajectory cluster generated with the proposed algorithm and the figure in the upper right side depicts the terminal positions of the different trajectories. It is apparent that the cooperative coverage situation has been established. In Table 2, the position errors are relatively small, with the maximum absolute value being 367.036 m and the minimum absolute value being 117.934 m, which proves that the revised terminal position constraints are well satisfied. Fig. 3 gives the curves of the velocity of different trajectories. It can be seen that the deviations of the velocities from the nominal one are very small, with its minimum value being 2 245 m/s, with a decrease of 28.165 m/s as in Tra 5. In Table 2, the small deviations of the velocities prove that the index criterion is maximized and the optimality is retained. Fig. 4 gives the curves of the flight path angle θ. It can be seen that the flight path angles of different trajectories converge to the nominal one at the terminal time, and the maximum deviation is less than 1° as given in Table 2, which proves that the proposed algorithm is able to maintain the unrevised states of the nominal trajectory. In Fig. 5, the curves of the control command are given. It can be seen that the control commands are relatively smooth. The deviation of the control command from the nominal one is relatively far at the initial time to compensate for the revised terminal perturbations.

Fig. 6 – Fig. 8 depict the path constraints for the hypersonic vehicle trajectory cluster. It can be seen that the inequality constraints are satisfied.

4.2 Optimal trajectory modification

When the terminal constraints of the midcourse trajectory are modified with the updated PIP, the midcourse guidance law should guide the interceptor to the revised PIP for a better handover situation and a maximum probability of successful interception as depicted in Fig. 9. The unique hypersonic interception forms a critical limit for the time consumption of the modification algorithm, which should be able to converge to a stable solution that meets the revised constraints within an update interval. In order to further test the optimality and the online application capacity of the proposed algorithm, the simulation scenario is designed as follows. During the first 50 s, the interceptor follows the nominal trajectory with the terminal location as (500 km, 35km), and the flight path angle as $-5°$. The PIP is revised at 50 s with (500 km, 37 km), and the interceptor should immediately modify the terminal constraints. At 100 s, the PIP is revised again with (500 km, 38 km) and holds until the end of the simulation. It should be noted that, at the time of the second modification, the trajectory has already diverted from the nominal trajectory, and the initial states perturbations should be considered as $\delta \boldsymbol{x}_0 = (\boldsymbol{x} - \boldsymbol{x}^*)_0$, where 0 should be the time of 100 s. The simulation results given by the proposed method (denoted as NOC) are further compared with those generated by the GPM, in which 50 nodes are selected. The related figures are given in Fig. 10 – Fig. 16. The comparison of time consumption of the different algorithms is given in Table 3, and the terminal states deviations compared with the predesigned locations and nominal ones are given in Table 4.

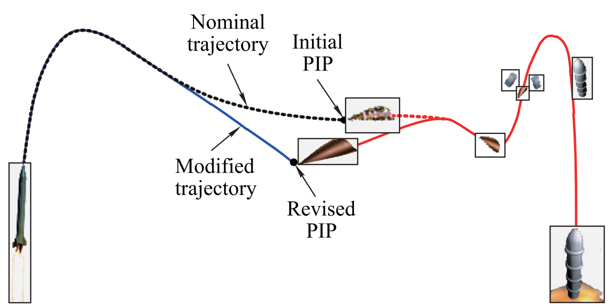

Fig. 9 Trajectory modification satisfying the revised PIP constraints

Table 3 Time consumption of NOC and GPM s

Algorithm	Modification 1	Modification 2
NOC	0.778 184	0.396 658
GPM	4.356 478	4.346 843

Table 4 Terminal states deviations comparison

Algorithm	Δx/km	Δy/km	ΔV/(m/s)	$\Delta\theta$/(°)
NOC	−236.543	106.201	1.074	0.238
GPM	−302.386	110.814	4.889	0.331

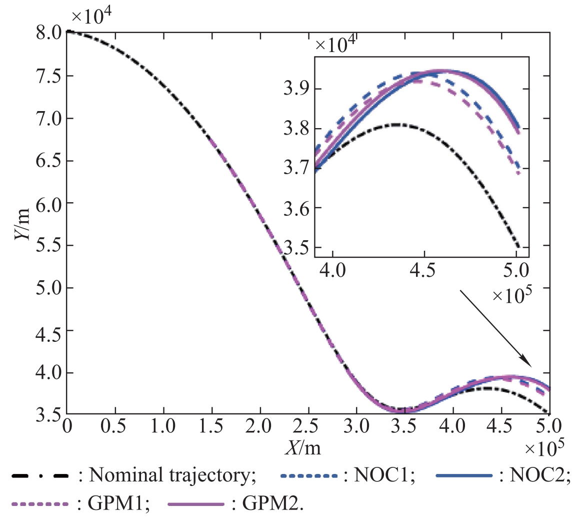

Fig. 10 Trajectory modification

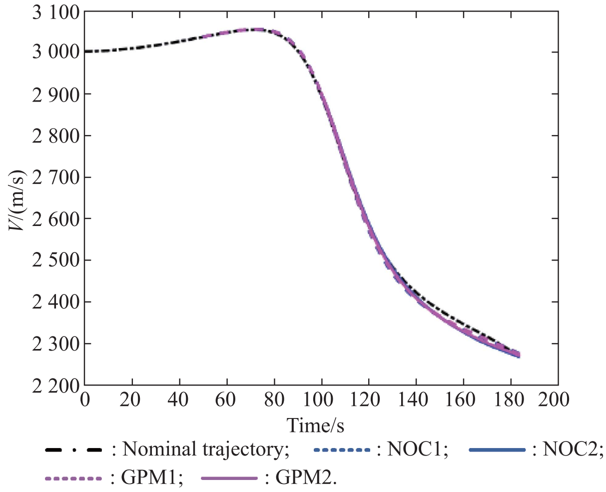

Fig. 11 Curves of the modified velocities

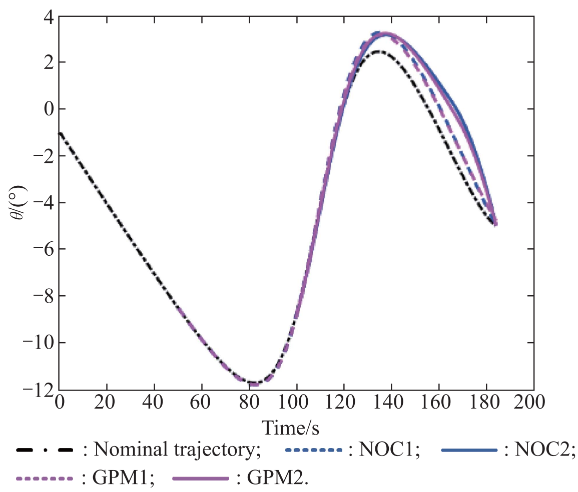

Fig. 12 Curves of the modified flight path angle

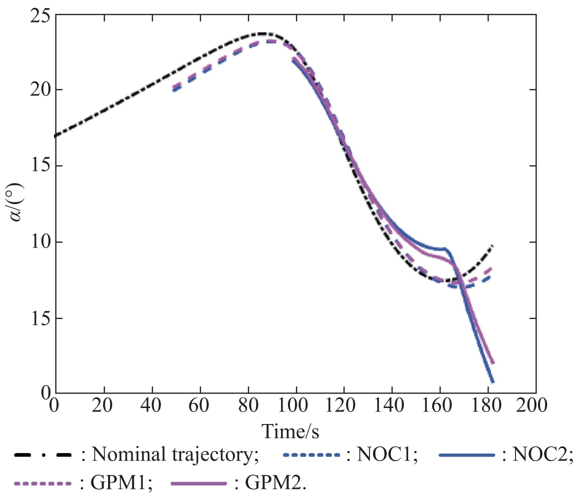

Fig. 13 **Curves of the modified control command**

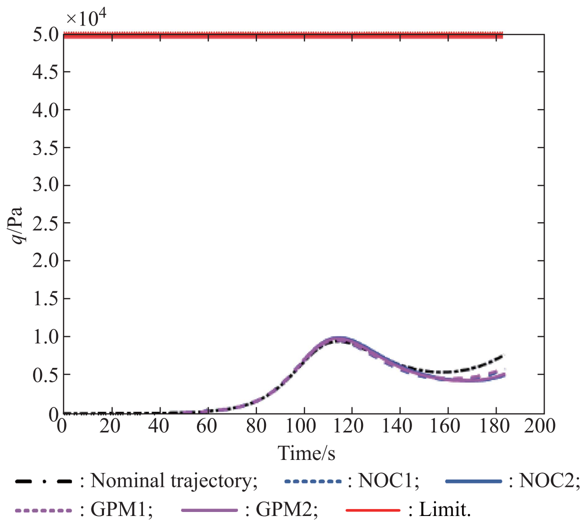

Fig. 14 **Curves of the modified dynamic pressure**

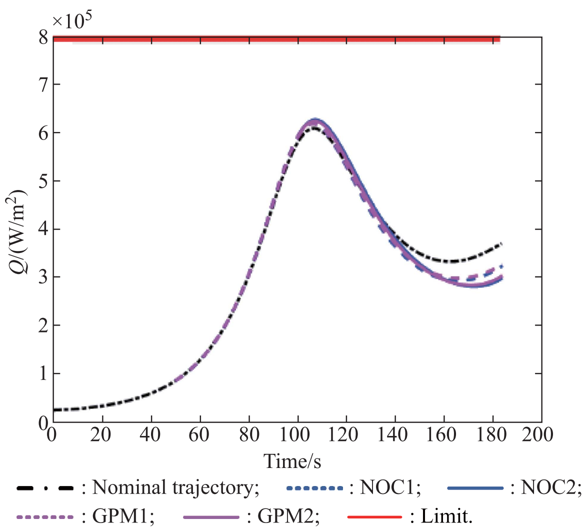

Fig. 15 **Curves of the modified heating transfer rate**

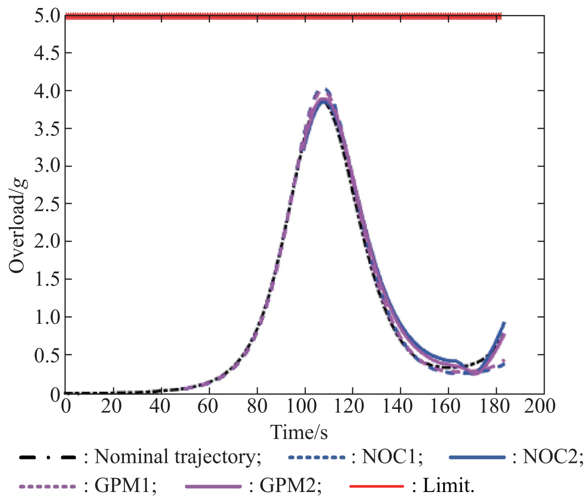

Fig. 16 **Curves of the modified overload**

It can be seen from Fig. 10 that the modified trajectories generated by NOC and GPM are smooth enough. In fact, the trajectories generated by the two different algorithms are relatively the same, for most parts of the trajectories are overlapped, which proves the optimality of the proposed method. The terminal position deviations listed in Table 4 show that both constraints are met and the results generated by NOC have the advantage of a better precision. The time consumed by GPM is longer than that of the NOC. In Modification 1 of the PIP, the time consumption of the GPM is 4.356 478 s, almost 5.59 times longer than that of the NOC. And in the Modification 2, the time consumption of GPM is 4.346 843 s, with little change compared with Modification 1 due to the unchanged nodes of 50. The time consumption of NOC in Modification 2 is 0.396 658 s, which has decreased by 0.381 526 s because of the shortened remaining time. It can be concluded that the proposed method is about five times faster than GPM, which proves its online application capacity.

Fig. 11 gives the curves of the modified velocities. The NOC has well remained the optimality of the index criterion, with the terminal velocity decreased by 1.074 m/s compared with the nominal one, and that of the GPM is 4.889 m/s. In Fig. 12, the constraints of the flight path angle are both satisfied by the NOC and GPM, with the terminal deviation being 0.238° and 0.331° respectively. Fig. 14 – Fig. 16 depict the path constraints for the hypersonic vehicle trajectory modification. It can be seen that the dynamic pressure constraints, the heating transfer rate constraints and the overload constraints are all satisfied.

5. Conclusions

An optimal midcourse trajectory cluster generation and trajectory modification algorithm is derived based on the neighboring optimal control theory, which has the merits of high precision and fast convergence. By reversely integrating the revised terminal constraints to the initial time, the perturbations of the co-states are expressed with the states deviations and terminal constraints modifications. A series of control modifications are calculated to reshape the trajectory to satisfy the modified terminal constraints. Simulation results show that the time efficiency of the proposed method is about five times faster than that of the GPM with an equivalent high precision. The potential application of the algorithm could be the generation of trajectory cluster for the cooperative coverage of the hypersonic vehicle predicted area with respective PDF and the online optimal trajectory modification aiming to the continuously updating PIP in the hypersonic interception.

References

[1] D. Dionne, H. Michalska, C. A. Rabbath. A predictive guidance law with uncertain information about the target state. *Proc. of the American Control Conference*, 2006: 1062 – 1067.

[2] D. Dionne, H. Michalska, C. A. Rabbath. Predictive guidance for pursuit-evasion engagements involving multiple decoys. *Journal of Guidance, Control and Dynamics*, 2007, 30(5): 1277 – 1286.

[3] P. N. Dwivedi, S. N. Tiwari, A. Bhattacharya, et al. A ZEM-based effective integrated estimation and guidance of interceptors in terminal phase. *Proc. of the AIAA Guidance, Navigation, and Control Conference*, 2010: 2010 – 8318.

[4] J. Yang, Z. Qu, J. Wang, et al. Real-time optimized trajectory planning for a fixed-wing vehicle flying in a dynamic environment. *Journal of Aerospace Engineering*, 2009, 22(4): 331 – 341.

[5] Dever, B. Mettler, E. Feron, et al. Nonlinear trajectory generation for autonomous vehicles via parameterized maneuver classes. *Journal of Guidance, Control and Dynamics*, 2006, 29(2): 289 – 302.

[6] J. A. Alejo, G. Cobano, A. Heredia, et al. Collision-free 4D trajectory planning in unmanned aerial vehicles for assembly and structure construction. *Journal of Intelligent Robotic Systems*, 2014, 73(1 – 4): 783 – 795.

[7] Y. Zhang, J. Chen, L. Shen. Real-time trajectory planning for UCAV air-to-surface attack using inverse dynamics optimization method and receding horizon control. *Chinese Journal of Aeronautics*, 2013, 26(4): 1038 – 1056.

[8] X. Gu, Y. Zhang, J. Chen, et al. Real-time decentralized cooperative robust trajectory planning for multiple UCAVs air-to-ground target attack. *Journal of Aerospace Engineering*, 2015, 229(4): 581 – 600.

[9] Verma. On-line adaptive estimation and trajectory reshaping. *Proc. of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005: 6436.

[10] J. Doebbler, P. Gesting, J. Valasek. Real-time path planning and obstacle avoidance for general aviation aircraft. *Proc. of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005: 5825.

[11] T. Tsuchiya, M. Miwa, S. Suzuki. Real-time flight trajectory optimization and its verification in flight. *Journal of Aircraft*, 2009, 46(4): 1468 – 1470.

[12] N. Faiz, S. K. Agrawal. Trajectory planning of differentially flat systems with dynamics and inequalities. *Proc. of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2000: 4259.

[13] J. Lwis, J. Rudolph. Real-time trajectory generation for flat systems with constraints. *Nonlinear and Adaptive Control*, 2003, 281: 385 – 394.

[14] M. A. Hurnil, P. Sekhavat, I. M. Ross. Autonomous trajectory planning using real-time information updates. *Proc. of the AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008: 6305.

[15] B. L. Tian, Q. Zong. Optimal guidance for reentry vehicles based on indirect Legendre pseudospectral method. *Acta Astronautica*, 2011, 68: 1176 – 1184.

[16] J. T. Betts. *Practical methods for optimal control using nonlinear programming*. Philadelphia: Society for Industrial and Applied Mechanics, 2001.

[17] H. M. Lei, J. Zhou, D. L. Zhai, et al. Midcourse optimal trajectory modification using the second variation. *Systems Engineering and Electronics*, 2016, 38(12): 2807 – 2813. (in Chinese).

[18] J. Zhao, R. Zhou. Reentry trajectory optimization for hypersonic vehicle satisfying complex constraints. *Chinese Journal of Aeronautics*, 2013, 26(6): 1544 – 1553.

[19] F. Rampazzo, R. B. Vinter. A theorem on existence of neighboring trajectories satisfying a state constant, with applications to optimal control. *IMA Journal of Mathematical Control Information*, 1999, 16(4): 335 – 351.

Biographies

Humin Lei was born in 1960. He is now a professor and a supervisor for doctors in Air and Missile Defense College, Air Force Engineering University, Xi'an, China. He received his M.S. and Ph.D. degrees from Northwestern Polytechnical University, in 1989 and 1999. His current research interests lie in the field of advanced guidance law design, hypersonic vehicle controller design and hypersonic interception strategy.

E-mail: hmleinet@21cn.com

Jin Zhou was born in 1989. He is a Ph.D. candidate in control theory and engineering at Air and Missile Defense College, Air Force Engineering University, Xi'an, China. He received his B.S. degree in navigation, guidance and control engineering in 2015 and M.S. degree in automatic control theory in 2012, both from Air Force Engineering University. His current research interests focus on the hypersonic interception, advanced guidance law design, and missile automatic controller design.

E-mail: 1019922781@qq.com

Dailiang Zhai was born in 1987. He received his B.S. and M.S. degrees in Air Force Engineering University. He is now a Ph. D. candidate of signal processing and filtering. His research interests include the hypersonic vehicle modeling and hypersonic target tracking.

E-mail: quietzdl@126.com

Lei Shao was born in 1982. He is now a Ph.D. and assistant professor at Air and Missile Defense College, Air Force Engineering University, Xi'an, China. He received his M.S. and B.S. degrees in the same university. His research interests include the hypersonic vehicle controller design and aircraft trajectory optimization.
E-mail: shaoleijin@163.com

Dayuan Zhang was born in 1987. He is now a Ph.D. and assistant engineer at Unit 95948 of the PLA, Jiuquan, China. He received his Ph.D. and B.S. degrees in Air Force Engineering University. His research interests include the hypersonic vehicle trajectory optimization and midcourse guidance law design.
E-mail: dayuanyjs@163.com