

Research Article

Variable-Time-Domain Online Neighboring Optimal Trajectory Modification for Hypersonic Interceptors

Ningbo Li,^{1,2} Humin Lei,¹ Jin Zhou,¹ Lei Shao,¹ and Bin Wang¹

¹Air and Missile Defence College, Air Force Engineering University, Xi'an 710051, China

²School of Electronic and Information Engineering, Xian Jiao Tong University, Xi'an 710049, China

Correspondence should be addressed to Ningbo Li; lnb_las@163.com

Received 19 December 2016; Revised 24 April 2017; Accepted 8 June 2017; Published 12 July 2017

Academic Editor: Paul Williams

Copyright © 2017 Ningbo Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The predicted impact point (PIP) of hypersonic interception changes continually; therefore the midcourse guidance law must have the ability of online trajectory optimization. In this paper, an online trajectory generation algorithm is designed based on neighboring optimal control (NOC) theory and improved indirect Radau pseudospectral method (IRPM). A trajectory optimization model is designed according to the features of operations in near space. Two-point boundary value problems (TPBVPs) are obtained based on NOC theory. The second-order linear form of transversality conditions is deduced backward to express the modifications of terminal states, costates, and flight time in terms of current state errors and terminal constraints modifications. By treating the current states and the optimal costates modifications as initial constraints and perturbations, the feedback control variables are obtained based on improved IRPM and nominal trajectory information. The simulation results show that when the changes of terminal constraints are not relatively large, this method can generate a modified trajectory effectively with high precision of terminal modifications. The design concept can provide a reference for the design of the online trajectory generation system of hypersonic vehicles.

1. Introduction

After decades of development, hypersonic technology has made a lot of progress [1–3]. Hypersonic vehicles will be used in military as precision strike weapons or platforms in the near future. The demand on the research of advanced intercept and defense technology is urgent.

In allusion to the high speed and large maneuverability of hypersonic weapons, the interceptor should adopt the compound guidance strategy to improve the success rate of interception. From the end of program control flight to the target acquisition of terminal guidance, the interceptor spends most of the time flying in the midcourse guidance. So the flight performance of midcourse guidance determines the overall performance.

If we adopt the traditional guidance method to intercept the high dynamic target directly, it will cause the frequent changes of the trajectory, which is not conducive to the

implementation of the interception and leads to unnecessary loss of energy. So it is a reasonable strategy to design the trajectory of midcourse guidance. The main objective of the midcourse guidance is to minimize the energy consumption in the flight process of interceptor and enter the terminal guidance with the best relative geometric relationship [4]. At present, the challenges which the midcourse guidance law design of near space interceptors faces are mainly in the following two aspects: on the one hand, during the flight process of interceptors, problems such as structural strength, thermal protection, normal work of the engine, and control stability give rise to strict requirements on the heat flux, dynamic pressure, overload, control, and other characteristics of the process. Additionally, the predicted impact point (PIP) provides terminal position constraints, and the terminal guidance acquisition state-space becomes the strong constraints conditions of interceptors [5]. On the other hand, in actual flight process, the variation ranges of altitude and

velocity are relatively large. Due to the initial conditions errors, atmospheric environment changes, and great uncertainty of aerodynamic model and navigation equipment, it will cause trajectory tracking errors. Furthermore, the nominal trajectory generated before launch is based on remote detection results. In this condition, tracking and prediction of the target have great error. With the approaching of the interceptor and the target, tracking and prediction accuracy will be improved by means of approaching detection with on-board equipment. And the target is also likely to maneuver actively. All those will lead to the change of PIP, and terminal constraints also need updates accordingly. Therefore, the designed midcourse guidance law must have the ability of online trajectory optimization.

Taking all kinds of terminal constraints and a series of process characteristics into account, the online optimization problem is actually a complex nonlinear optimization problem [6], and many scholars have studied it. Lin and Tsai [7, 8] obtained a trajectory shaping guidance law by simplifying the model and realized online trajectory generation. But the trajectory was not true optimal due to too much approximation. Indig et al. [6, 9, 10] first linearized the dynamic model of interceptors. According to the location of PIP, the optimal guidance law was designed based on the trajectory shaping guidance. Then, in allusion to the actual nonlinear dynamic model, the terminal angle constraints of interceptor were added according to the midcourse and terminal guidance handover. Finally, the optimal trajectory model was deduced based on Pontryagin minimum principle and solved by Gauss pseudospectral method. Dwivedi et al. [11, 12] studied the problem of midcourse guidance trajectory planning using model predictive static planning method. Based on the state equations of discretized system, the control variables were updated by solving the costate vectors of the whole time period, whose coefficient matrix could be obtained by recursive method. The simulation results show that this method has high accuracy and efficiency. Yakimenko et al. [13–15] proposed a trajectory shaping guidance law based on the inverse dynamic of virtual domain. In this method, the coordinates of aircraft were expressed by higher order polynomials with virtual arc, and the dynamic equations were transformed from the time domain to the virtual domain. Introducing the virtual arc as an argument made it possible to optimize the speed history along the trajectory independently. Additionally, the number of the optimized variables was reduced and the integration process was avoided. By this method, the trajectory can be generated with good convergence robustness. In addition, some scholars have also studied this problem by pseudospectral method [16, 17], intelligent algorithm [18, 19], and so on.

The aforementioned scholars have made many accomplishments in the domain of trajectory optimization; however, they mostly focused on the trajectory regeneration in regard to the research of online trajectory optimization problems. They completely abandoned the original trajectory and reoptimized again, which caused large amount of calculation and more time consumption, and put forward higher requirements on the storage capacity and computing ability of on-board computer. So it is not easy to realize in

engineering. In allusion to the characteristics of near space targets, such as small overload, long flight time, and large flight range, prediction of impact point will be continually updated and the terminal constraints change correspondingly. When the perturbations of current states and changes of terminal constraints are not very large, according to the neighboring optimal control theory, the modified trajectory can be generated quickly within the neighborhood of the nominal trajectory using nominal trajectory data to realize online trajectory generation.

We still need to solve the TPBVPs [20] when adopting above modification algorithm. The time-intensive backward integration has brought difficulties to the online implementation. Recently, Fahroo and Ross [21] and Yan et al. [22] proposed the indirect Legendre pseudospectral method (ILPM) for solving the TPBVPs by matrix inversion which improved solving efficiency a lot. In this method, the modifications of states and costates were solved by initial perturbations of states and the nominal trajectory information, and then the feedback control modifications were obtained to form the closed loop control. In this paper, the idea of ILPM is used for reference and the IRPM is proposed which is improved in two aspects. On the one hand, Fahroo and Hui only solved the trajectory tracking problems with initial perturbations, while the improved IRPM considers not only trajectory tracking errors but also modifications of terminal constraints. If computation speed is fast enough, the trajectory can be generated in real time to realize closed loop guidance without designing the feedback control law. On the other hand, Radau pseudospectral method is superior to Legendre pseudospectral method in approximating precision of control, state, and costate variables. In terms of computational efficiency, the two methods have little difference when solving the same scale problem.

The main purpose of this paper is to design an online optimal modification algorithm and realize online trajectory optimization under the condition of changing terminal constraints. The rest of paper is organized as follows. In Section 2, the nominal trajectory satisfying terminal and path constraints is generated offline. Next, the trajectory modified model with unspecified terminal time is deduced by NOC in Section 3 and is solved by improved IRPM in Section 4. In Section 5, numerical simulations are presented. Finally, some conclusions are drawn in Section 6.

2. Nominal Optimal Trajectory

2.1. Dynamic Model of Interceptor. When the nominal optimal trajectory is generated offline, the interceptor should be directly guided towards the PIP, so the dynamic model is considered in the longitudinal plane. The states of interceptor are represented by four-dimensional vector (v, θ, x, y) consisting of speed, flight path angle, and location in the inertial coordinate and given as follows [23]:

$$\begin{aligned}\dot{v} &= \frac{P \cos \alpha - C_x q S}{m} - g \sin \theta, \\ \dot{\theta} &= \frac{P \sin \alpha + C_y q S}{mv} - \frac{g \cos \theta}{v},\end{aligned}$$

TABLE 1: The related parameters of booster engine.

Parameter name	Value
I_{sp} (stage 1/2)	280/200 (s)
V_e (stage 1/2)	30/8.9 (kg/s)
Burning time (stage 1/2)	20/50 (s)
Mass of booster shell	250 kg
Mass of interceptor	60 kg

$$\begin{aligned}\dot{x} &= v \cos \theta, \\ \dot{y} &= v \sin \theta,\end{aligned}\quad (1)$$

where m is interceptor mass; q is the dynamic pressure; S is the reference area; g is gravitational acceleration; P is the engine thrust. In order to achieve a higher flight speed, the two-stage booster program is designed. The thrust calculation formulas are shown in

$$P = V_e \frac{dm}{dt}, \quad V_e = g I_{sp}, \quad (2)$$

where V_e is the fuel gas velocity and I_{sp} is the specific impulse, whose value is generally 200~300 s for the solid rocket engine [24]. This paper designs a two-stage solid rocket booster. The related parameters of the engine are shown in Table 1.

C_x, C_y denote the drag and lift coefficients, respectively, which are expressed as a function of Ma, the Mach number, and α , the angle of attack:

$$\begin{aligned}C_y &= C_y^\alpha (\text{Ma}) \alpha, \\ C_x &= C_{x0} (\text{Ma}) + K (\text{Ma}) \cdot C_y^2,\end{aligned}\quad (3)$$

where C_{x0}, K denote zero-lift drag coefficient and induced drag coefficient, respectively; C_y^α denotes the partial derivative of C_y with respect to α . q denotes the dynamic pressure:

$$q = 0.5 \rho v^2. \quad (4)$$

And ρ is the atmosphere density expressed as

$$\rho = \rho_0 \exp\left(\frac{-y}{y_0}\right), \quad (5)$$

where $\rho_0 = 1.2250 \text{ kg/m}^3$, $y_0 = 7254.3 \text{ m}$.

2.2. Trajectory Optimization Problem. In order to ensure kinematic kill effect of interception, terminal speed is usually selected as cost function:

$$J = \phi(v(t_f), t_f) = -v_f. \quad (6)$$

In this paper, subscript 0 denotes initial conditions and subscript f denotes terminal conditions. As is discussed above, the main objective of the midcourse guidance is to deliver the interceptor to a certain position (usually the PIP) with specified conditions to ensure a successful handover

between midcourse and terminal guidance. So the position constraints (x_f, y_f) are given by PIP and the flight path angle constraint θ_f at terminal time is given by the terminal guidance acquisition conditions

$$\psi = [\theta - \theta_f \ x - x_f \ y - y_f]^T = \mathbf{0}, \quad (7)$$

where $\mathbf{0}$ denotes zero matrix with the corresponding dimension.

The above trajectory optimization problem is solved with the help of the software package GPOPS [25]. The whole trajectory was composed of three phases: boost stages 1 and 2 and nonpropulsive phase. Compared with other direct optimization methods, pseudospectral method can obtain higher accuracy with fewer nodes, which improves efficiency of the algorithm. The nominal optimal trajectory is shown in Figure 1.

3. Trajectory Modification Based on Neighborhood Optimization

3.1. Two-Point Boundary Value Problem. In the process of actual flight, when tracking errors of current states and changes of terminal constraints are not very large, according to the neighboring optimal control theory, we can further differentiate the first-order necessary conditions of the optimal trajectory to the second order to obtain perturbation equations, and the modified trajectory satisfying the changed terminal conditions can be obtained using the nominal trajectory data. At the same time, the cost function can still maintain certain optimality.

First of all, the first-order optimal necessary conditions are derived based on optimal control. By introducing the costate variable $\lambda = [\lambda_v \ \lambda_\theta \ \lambda_x \ \lambda_y]^T$ with the same dimension as the state variable $\mathbf{x} = [v \ \theta \ x \ y]^T$, Hamilton equation can be expressed as

$$\mathbf{H} = \lambda^T \mathbf{f} = \lambda_v \dot{v} + \lambda_\theta \dot{\theta} + \lambda_x \dot{x} + \lambda_y \dot{y}, \quad (8)$$

where $\mathbf{f} = [\dot{v} \ \dot{\theta} \ \dot{x} \ \dot{y}]^T$ denotes system equations.

Then the augmented performance index with terminal constraints \bar{J} can be expressed as

$$\bar{J} = [\phi + \boldsymbol{\nu}^T \psi]_f + \int_{t_0}^{t_f} (\mathbf{H} - \lambda^T \dot{\mathbf{x}}) dt, \quad (9)$$

where $\boldsymbol{\nu} = [\nu_\theta \ \nu_x \ \nu_y]^T$. Canonical equation and coupled equation can be expressed as

$$\dot{\mathbf{x}} = \frac{\partial \mathbf{H}}{\partial \lambda}, \quad (10)$$

$$\dot{\lambda} = -\frac{\partial \mathbf{H}}{\partial \mathbf{x}}, \quad (11)$$

$$\frac{\partial \mathbf{H}}{\partial \mathbf{u}} = \mathbf{0}. \quad (12)$$

Considering the change of terminal conditions, the terminal time of modified trajectory will change accordingly. We

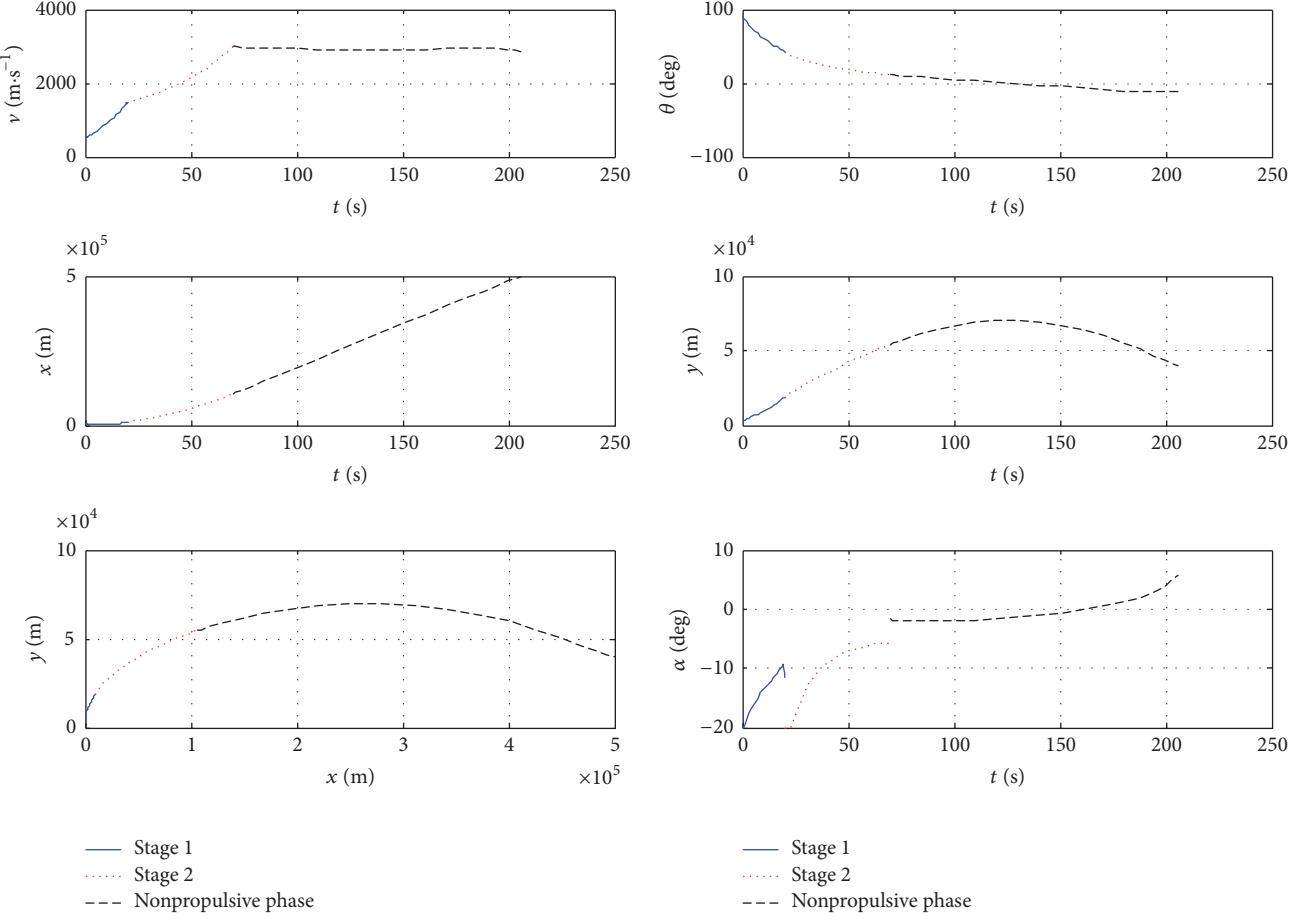


FIGURE 1: Nominal trajectory.

can derive the equations of boundary conditions specified at an unspecified terminal time and the terminal constraints of Hamiltonian function along the optimal trajectory based on NOC [26, 27]

$$\psi(\mathbf{x}(t_f)) = \mathbf{0}, \quad (13)$$

$$\lambda(t_f) = -\left(\frac{\partial \phi}{\partial \mathbf{x}} + \boldsymbol{\nu}^T \frac{\partial \psi}{\partial \mathbf{x}}\right)_{t=t_f} = -\frac{\partial \Phi}{\partial \mathbf{x}} \Big|_{t=t_f}, \quad (14)$$

$$\begin{aligned} \mathbf{0} &= \left(\mathbf{H} + \frac{\partial \phi}{\partial t_f} + \boldsymbol{\nu}^T \frac{\partial \psi}{\partial t_f} \right)_{t=t_f} \\ &= \left(\mathbf{H} + \frac{\partial \Phi}{\partial t_f} \right)_{t=t_f}. \end{aligned} \quad (15)$$

Equations (10)~(12) can be further differentiated to second order as follows:

$$\delta \dot{\mathbf{x}} = \frac{\partial^2 \mathbf{H}}{\partial \lambda \partial \mathbf{x}} \delta \mathbf{x} + \frac{\partial^2 \mathbf{H}}{\partial \lambda \partial \mathbf{u}} \delta \mathbf{u}, \quad (16)$$

$$\delta \dot{\lambda} = -\frac{\partial^2 \mathbf{H}}{\partial \mathbf{x}^2} \delta \mathbf{x} - \frac{\partial^2 \mathbf{H}}{\partial \mathbf{x} \partial \lambda} \delta \lambda - \frac{\partial^2 \mathbf{H}}{\partial \mathbf{x} \partial \mathbf{u}} \delta \mathbf{u}, \quad (17)$$

$$\mathbf{0} = \frac{\partial^2 \mathbf{H}}{\partial \mathbf{u} \partial \mathbf{x}} \delta \mathbf{x} + \frac{\partial^2 \mathbf{H}}{\partial \mathbf{u} \partial \lambda} \delta \lambda + \frac{\partial^2 \mathbf{H}}{\partial \mathbf{u}^2} \delta \mathbf{u}. \quad (18)$$

If $\partial^2 \mathbf{H}/\partial \mathbf{u}^2$ is nonsingular in the whole flight process, we can get the expression of control modifications $\delta \mathbf{u}$ by (18):

$$\delta \mathbf{u} = -\left(\frac{\partial^2 \mathbf{H}}{\partial \mathbf{u}^2}\right)^{-1} \left(\frac{\partial^2 \mathbf{H}}{\partial \mathbf{u} \partial \mathbf{x}} \delta \mathbf{x} + \frac{\partial^2 \mathbf{H}}{\partial \mathbf{u} \partial \lambda} \delta \lambda \right). \quad (19)$$

The dynamic equations of state modifications $\delta \mathbf{x}$ and costate modifications $\delta \lambda$ can be obtained by substituting (19) into (16)~(17):

$$\delta \dot{\mathbf{x}} = \mathbf{A}(t) \delta \mathbf{x} - \mathbf{B}(t) \delta \lambda, \quad (20)$$

$$\delta \dot{\lambda} = \mathbf{C}(t) \delta \mathbf{x} - \mathbf{A}^T(t) \delta \lambda, \quad (21)$$

where

$$\begin{aligned}\mathbf{A}(t) &= \frac{\partial^2 \mathbf{H}}{\partial \lambda \partial \mathbf{x}} - \frac{\partial^2 \mathbf{H}}{\partial \lambda \partial \mathbf{u}} \left(\frac{\partial^2 \mathbf{H}}{\partial \mathbf{u}^2} \right)^{-1} \frac{\partial^2 \mathbf{H}}{\partial \mathbf{u} \partial \mathbf{x}}, \\ \mathbf{B}(t) &= \frac{\partial \mathbf{H}}{\partial \lambda \partial \mathbf{u}} \left(\frac{\partial^2 \mathbf{H}}{\partial \mathbf{u}^2} \right)^{-1} \frac{\partial^2 \mathbf{H}}{\partial \mathbf{u} \partial \lambda}, \\ \mathbf{C}(t) &= \frac{\partial^2 \mathbf{H}}{\partial \mathbf{x}^2} - \frac{\partial^2 \mathbf{H}}{\partial \mathbf{x} \partial \mathbf{u}} \left(\frac{\partial^2 \mathbf{H}}{\partial \mathbf{u}^2} \right)^{-1} \frac{\partial^2 \mathbf{H}}{\partial \mathbf{u} \partial \mathbf{x}}.\end{aligned}\quad (22)$$

Equations (20)~(21) satisfy initial conditions $\delta \mathbf{x}(t_0) = \delta \mathbf{x}_0$, $\delta \lambda(t_0) = \delta \lambda_0$ and terminal conditions (13)~(15) form the TPBVPs of $\delta \mathbf{x}$, $\delta \lambda$.

3.2. Terminal Linearization and Backward Recursion. Equations (6), (15), and (16) were differentiated to second order with unspecified terminal time. Since $\Phi = \phi + \mathbf{v}^T \psi$, we can get

$$d\psi = \left[\frac{\partial \psi}{\partial \mathbf{x}} \delta \mathbf{x} + \left(\frac{\partial \psi}{\partial t} + \frac{\partial \psi}{\partial \mathbf{x}} \frac{d\mathbf{x}}{dt} \right) dt_f \right]_{t=t_f}, \quad (23)$$

$$\begin{aligned}d\lambda(t_f) &= \delta \lambda(t_f) + \frac{d\lambda}{dt} \Big|_{t=t_f} dt_f = \left[\frac{\partial^2 \Phi}{\partial \mathbf{x}^2} \delta \mathbf{x} \right. \\ &\quad \left. + \left(\frac{\partial \psi}{\partial \mathbf{x}} \right)^T d\mathbf{v} + \frac{\partial^2 \Phi}{\partial \mathbf{x} \partial t} dt_f \right]_{t=t_f},\end{aligned}\quad (24)$$

$$\begin{aligned}\mathbf{0} &= \left[\left(\frac{\partial \mathbf{H}}{\partial \mathbf{x}} + \frac{\partial^2 \Phi}{\partial t \partial \mathbf{x}} \right) \delta \mathbf{x} + \frac{\partial \psi^T}{\partial t} d\mathbf{v} + \frac{\partial \mathbf{H}}{\partial \lambda} \delta \lambda \right. \\ &\quad \left. + \left(\frac{\partial \mathbf{H}}{\partial t} + \frac{\partial^2 \Phi}{\partial t^2} \right) dt_f \right]_{t=t_f}.\end{aligned}\quad (25)$$

Equations (23)~(26) consist of $n + q + 1$ linear equations with $2n+q+1$ unknown variables. They are state modifications $\delta \mathbf{x}$ of dimension n , costate modifications $\delta \lambda$ of dimension n , variables $d\mathbf{v}$ of dimension q , and variable dt_f of dimension one. Variables $d\mathbf{v}$ and state modifications of dimension $n - q$ unspecified at terminal time are defined as free variables

$$d\boldsymbol{\mu} = [d\mathbf{v}_1, d\mathbf{v}_2, \dots, d\mathbf{v}_q, \delta \mathbf{x}_{q+1}, \delta \mathbf{x}_{q+2}, \dots, \delta \mathbf{x}_n]_{t=t_f}. \quad (26)$$

Since the changes of terminal constraints are not very large, (23)~(25) can be linearized based on small perturbation hypothesis, so that we get

$$\begin{pmatrix} \delta \mathbf{x} \\ \delta \lambda \\ dt_f \end{pmatrix}_{t=t_f} = \mathbf{P} \begin{pmatrix} d\boldsymbol{\mu} \\ d\psi_f \end{pmatrix} = \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \\ \mathbf{P}_{31} & \mathbf{P}_{32} \end{pmatrix} \begin{pmatrix} d\boldsymbol{\mu} \\ d\psi_f \end{pmatrix}, \quad (27)$$

where

$$\begin{aligned}\mathbf{P}_{11} &= \frac{1}{\dot{V} \lambda_V} \begin{bmatrix} 0 & 0 & 0 & \dot{V} \lambda_V \\ \dot{\theta}^2 & \dot{\theta} \dot{x} & \dot{\theta} \dot{y} & \dot{\theta} \lambda_V \\ \dot{\theta} \dot{x} & \dot{x}^2 & \dot{x} \dot{y} & \dot{x} \lambda_V \\ \dot{\theta} \dot{y} & \dot{x} \dot{y} & \dot{y}^2 & \dot{y} \lambda_V \end{bmatrix}, \\ \mathbf{P}_{12} &= \frac{1}{\dot{V} \lambda_V} \begin{bmatrix} 0 & 0 & 0 \\ \dot{V} \lambda_V + \dot{\theta} \lambda_\theta & \dot{\theta} \lambda_x & \dot{\theta} \lambda_y \\ \dot{x} \lambda_\theta & \dot{V} \lambda_V + \dot{x} \lambda_x & \dot{x} \lambda_y \\ \dot{y} \lambda_\theta & \dot{y} \lambda_x & \dot{V} \lambda_V + \dot{y} \lambda_y \end{bmatrix}, \\ \mathbf{P}_{21} &= \frac{1}{\dot{V} \lambda_V} \cdot \begin{bmatrix} \dot{\theta} \lambda_V & \dot{x} \lambda_V & \dot{y} \lambda_V & \dot{\lambda}_V^2 \\ -\dot{V} \lambda_V + \dot{\theta} \lambda_\theta & \dot{x} \lambda_\theta & \dot{y} \lambda_\theta & \dot{\lambda}_V \lambda_\theta \\ \dot{\theta} \lambda_x & -\dot{V} \lambda_V + \dot{x} \lambda_x & \dot{y} \lambda_x & \dot{\lambda}_V \lambda_x \\ \dot{\theta} \lambda_y & \dot{x} \lambda_y & -\dot{V} \lambda_V + \dot{y} \lambda_y & \dot{\lambda}_V \lambda_y \end{bmatrix}, \\ \mathbf{P}_{22} &= \frac{1}{\dot{V} \lambda_V} \begin{bmatrix} \dot{\lambda}_V \lambda_\theta & \dot{\lambda}_V \lambda_x & \dot{\lambda}_V \lambda_y \\ \dot{\lambda}_\theta^2 & \dot{\lambda}_\theta \lambda_x & \dot{\lambda}_\theta \lambda_y \\ \dot{\lambda}_\theta \lambda_x & \dot{\lambda}_x^2 & \dot{\lambda}_x \lambda_y \\ \dot{\lambda}_\theta \lambda_y & \dot{\lambda}_x \lambda_y & \dot{\lambda}_y^2 \end{bmatrix},\end{aligned}\quad (28)$$

$$\mathbf{P}_{31} = \frac{1}{\dot{V} \lambda_V} [-\dot{\theta} \ -\dot{x} \ -\dot{y} \ -\dot{\lambda}_V],$$

$$\mathbf{P}_{32} = \frac{1}{\dot{V} \lambda_V} [-\dot{\lambda}_\theta \ -\dot{\lambda}_x \ -\dot{\lambda}_y].$$

Equations (20)~(21) can be expressed as the following form:

$$\delta \dot{\Theta} = \begin{pmatrix} \mathbf{A}(t) & \mathbf{B}(t) \\ \mathbf{C}(t) & -\mathbf{A}(t) \end{pmatrix} \delta \Theta = \mathbf{G} \delta \Theta, \quad (29)$$

where $\delta \Theta = (\delta \mathbf{x} \ \delta \lambda)^T$ and the variables of above differential equation are discretized as $M + 1$ points by Δt . We can get by backward recursion

$$\begin{aligned}\delta \Theta_{M-1} &= \delta \Theta_M + \mathbf{G}_M \delta \Theta_M (-\Delta t) \\ &= (\mathbf{I} - \mathbf{G}_M \Delta t) \delta \Theta_M, \\ \delta \Theta_k &= \delta \Theta_{k+1} + \mathbf{G}_{k+1} \delta \Theta_{k+1} (-\Delta t) \\ &= (\mathbf{I} - \mathbf{G}_{k+1} \Delta t) \delta \Theta_{k+1} \\ &= (\mathbf{I} - \mathbf{G}_{k+1} \Delta t) (\mathbf{I} - \mathbf{G}_{k+2} \Delta t) \delta \Theta_{k+2}.\end{aligned}\quad (30)$$

Passing to the first point in turn we can get

$$\delta \Theta_0 = \mathbf{G} \delta \Theta_M, \quad (31)$$

where $\mathbf{G} = (\mathbf{I} - \mathbf{G}_1 \Delta t)(\mathbf{I} - \mathbf{G}_2 \Delta t) \cdots (\mathbf{I} - \mathbf{G}_M \Delta t)$, and according to (27) we have

$$\begin{pmatrix} \delta \mathbf{x} \\ \delta \lambda \end{pmatrix}_{t=t_f} = \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{pmatrix} \begin{pmatrix} d\boldsymbol{\mu} \\ d\psi_f \end{pmatrix}. \quad (32)$$

Substituting (32) into (31)

$$\begin{aligned} \begin{pmatrix} \delta\mathbf{x}(t_0) \\ \delta\lambda(t_0) \end{pmatrix} &= \begin{pmatrix} \mathbf{G}_1 & \mathbf{G}_2 \\ \mathbf{G}_3 & \mathbf{G}_4 \end{pmatrix} \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{pmatrix} \begin{pmatrix} d\boldsymbol{\mu} \\ d\psi_f \end{pmatrix} \\ &= \begin{pmatrix} \Pi_1 & \Pi_2 \\ \Pi_3 & \Pi_4 \end{pmatrix} \begin{pmatrix} d\boldsymbol{\mu} \\ d\psi_f \end{pmatrix}, \end{aligned} \quad (33)$$

where

$$\begin{aligned} \Pi_1 &= \mathbf{G}_1 \mathbf{P}_{11} + \mathbf{G}_2 \mathbf{P}_{21}, \\ \Pi_2 &= \mathbf{G}_1 \mathbf{P}_{12} + \mathbf{G}_2 \mathbf{P}_{22}, \\ \Pi_3 &= \mathbf{G}_3 \mathbf{P}_{11} + \mathbf{G}_4 \mathbf{P}_{21}, \\ \Pi_4 &= \mathbf{G}_3 \mathbf{P}_{12} + \mathbf{G}_4 \mathbf{P}_{22}, \end{aligned} \quad (34)$$

and if Π_1 is reversible, $d\boldsymbol{\mu}$ can be expressed as

$$d\boldsymbol{\mu} = \Pi_1^{-1} (\delta\mathbf{x}_0 - \Pi_2 d\psi_f). \quad (35)$$

Substituting (35) into (27) we can get

$$\begin{aligned} \delta\mathbf{x}_f &= \mathbf{P}_{11} \Pi_1^{-1} (\delta\mathbf{x}_0 - \Pi_2 d\psi_f) + \mathbf{P}_{12} d\psi_f, \\ \delta\lambda_f &= \mathbf{P}_{21} \Pi_1^{-1} (\delta\mathbf{x}_0 - \Pi_2 d\psi_f) + \mathbf{P}_{22} d\psi_f, \\ dt_f &= \mathbf{P}_{31} \Pi_1^{-1} (\delta\mathbf{x}_0 - \Pi_2 d\psi_f) + \mathbf{P}_{32} d\psi_f. \end{aligned} \quad (36)$$

3.3. The Optimality Theory

Theorem 1. *The modified trajectories obtained in the nominal trajectory neighborhood based on the neighborhood optimal control theory meet the changed terminal constraints, and the performance index has second-order optimality.*

Proof. According to the division integral theorem, the performance index can be expressed as

$$\bar{J} = [\phi + \boldsymbol{\nu}^T \boldsymbol{\psi}]_f - \lambda^T \mathbf{x}|_{t_0}^{t_f} + \int_{t_0}^{t_f} (\mathbf{H} + \dot{\lambda}^T \mathbf{x}) dt. \quad (37)$$

Considering the first-order variation of the performance index and taking canonical equations (10)-(11) and coupled equation (12) into it, we can get by ignoring the high order terms:

$$\begin{aligned} \delta\bar{J} &= [\lambda^T \delta\mathbf{x}]_0 + [\boldsymbol{\nu}^T \delta\boldsymbol{\psi}]_f + \left[\frac{1}{2} \delta\mathbf{x}^T \frac{\partial^2 \Phi}{\partial \mathbf{x}^2} \delta\mathbf{x} \right]_f \\ &\quad + \int_{t_0}^{t_f} \frac{1}{2} [\delta\mathbf{x}^T \ \delta\mathbf{u}^T] \begin{bmatrix} \frac{\partial^2 \mathbf{H}}{\partial \mathbf{x}^2} & \frac{\partial^2 \mathbf{H}}{\partial \mathbf{x} \partial \mathbf{u}} \\ \frac{\partial^2 \mathbf{H}}{\partial \mathbf{u} \partial \mathbf{x}} & \frac{\partial^2 \mathbf{H}}{\partial \mathbf{u}^2} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x} \\ \delta\mathbf{u} \end{bmatrix} dt. \end{aligned} \quad (38)$$

Multiplying (16) and (23) by the $\delta\lambda$ and $\delta\boldsymbol{\nu}$, respectively, we can get by taking the resulting equations into (38):

$$\begin{aligned} \delta\bar{J} &= [\lambda^T \delta\mathbf{x}]_0 + [\boldsymbol{\nu}^T \delta\boldsymbol{\psi}]_f + \left[\frac{1}{2} \delta\mathbf{x}^T \frac{\partial^2 \Phi}{\partial \mathbf{x}^2} \delta\mathbf{x} \right]_f \\ &\quad + \left[d\boldsymbol{\nu}^T \frac{\partial \boldsymbol{\psi}}{\partial \mathbf{x}} \delta\mathbf{x} \right]_f \\ &\quad + \int_{t_0}^{t_f} \delta\lambda^T \left(\delta\dot{\mathbf{x}} - \frac{\partial^2 \mathbf{H}}{\partial \lambda \partial \mathbf{x}} \delta\mathbf{x} + \frac{\partial^2 \mathbf{H}}{\partial \lambda \partial \mathbf{u}} \delta\mathbf{u} \right) dt \\ &\quad + \int_{t_0}^{t_f} \frac{1}{2} [\delta\mathbf{x}^T \ \delta\mathbf{u}^T] \begin{bmatrix} \frac{\partial^2 \mathbf{H}}{\partial \mathbf{x}^2} & \frac{\partial^2 \mathbf{H}}{\partial \mathbf{x} \partial \mathbf{u}} \\ \frac{\partial^2 \mathbf{H}}{\partial \mathbf{u} \partial \mathbf{x}} & \frac{\partial^2 \mathbf{H}}{\partial \mathbf{u}^2} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x} \\ \delta\mathbf{u} \end{bmatrix} dt. \end{aligned} \quad (39)$$

Considering the second-order variation of (39), we can get according to the division integral theorem:

$$\begin{aligned} \delta^2 \bar{J} &= \left[\left(\delta\mathbf{x}^T \frac{\partial^2 \Phi}{\partial \mathbf{x}^2} + d\boldsymbol{\nu}^T \frac{\partial \boldsymbol{\psi}}{\partial \mathbf{x}} - \delta\lambda^T \right) \delta^2 \mathbf{x} \right]_f \\ &\quad + \int_{t_0}^{t_f} \left(\delta\mathbf{x}^T \frac{\partial^2 \mathbf{H}}{\partial \mathbf{x} \partial \mathbf{u}} + \delta\lambda^T \frac{\partial^2 \mathbf{H}}{\partial \lambda \partial \mathbf{u}} + \delta\mathbf{u}^T \frac{\partial^2 \mathbf{H}}{\partial \mathbf{u}^2} \right) \delta^2 \mathbf{u} dt \\ &\quad + \int_{t_0}^{t_f} \left(\frac{d}{dt} (\delta\lambda^T) + \delta\mathbf{x}^T \frac{\partial^2 \mathbf{H}}{\partial \mathbf{x}^2} + \delta\lambda^T \frac{\partial^2 \mathbf{H}}{\partial \lambda \partial \mathbf{x}} \right. \\ &\quad \left. + \delta\mathbf{u}^T \frac{\partial^2 \mathbf{H}}{\partial \mathbf{u} \partial \mathbf{x}} \right) \delta^2 \mathbf{x} dt. \end{aligned} \quad (40)$$

Taking (17)-(18) and (24) into (40), we can get $\delta^2 \bar{J} = 0$ for any $\delta^2 \mathbf{x}$ and $\delta^2 \mathbf{u}$. Therefore, the second-order optimality of the performance index can be guaranteed with satisfying the changed terminal constraints. \square

4. Solving Algorithm Based on Improved IRPM

In the last section, considering the changes of terminal constraints of some state variables, the optimal terminal modification of costates variables $\delta\lambda_f$ and optimal modification of terminal time dt_f are derived at an unspecified terminal time based on the NOC. Improved IRPM will be used to solve the modified optimal trajectory according to $\delta\lambda_f$ and dt_f in this section.

Since dt_f has been calculated in the last section, the nominal trajectory information which will be used to solve the modified trajectory changes correspondingly. The new nominal trajectory information can be obtained by interpolating the original nominal trajectory. Because the trajectory tracking errors have been taken into account when we solved $\delta\lambda_f$ in the last section, we can regard current states of the interceptor as initial constraints and $\delta\lambda_f$ as the perturbations when solving the modified trajectory. Then we can use the new nominal trajectory information to design the solving algorithm to generate a trajectory from the current states to

the modified terminal states, whose performance index can still maintain certain optimality.

The boundary conditions in last section change to the following form:

$$\psi' = [\theta - \theta_0 \ x - x_0 \ y - y_0]^T = \mathbf{0}, \quad (41)$$

$$\lambda(t_0) = -\left(\mathbf{v}^T \frac{\partial \psi'}{\partial \mathbf{x}}\right)_{t=t_0}. \quad (42)$$

Equations (41)~(42) can be expressed as follows:

$$\Omega(\mathbf{x}(t_0), \lambda(t_0)) = \mathbf{0}. \quad (43)$$

Variation of (43) is solved as

$$\left[\frac{\partial \Omega}{\partial \mathbf{x}} \delta \mathbf{x} + \frac{\partial \Omega}{\partial \lambda} \delta \lambda \right]_{t=t_0} = \mathbf{0}. \quad (44)$$

Now we use improved IRPM to solve the optimal trajectory modified model. Firstly, the variables are discretized at Legendre-Gauss-Radau (LGR) points. Since the time domain for the above problem is $[t_0, t_f]$, while Radau pseudospectral method is used in the time domain $[-1, 1]$, it is necessary to introduce a new time variable τ :

$$\tau = \frac{2t}{t_f - t_0} - \frac{t_f + t_0}{t_f - t_0}. \quad (45)$$

States and costates variables are approximated as follows:

$$\delta \mathbf{x} \approx \delta \mathbf{x}^N(\tau) = \sum_{l=0}^{N-1} \delta \mathbf{x}(\tau_l) L_l(\tau), \quad (46)$$

$$\delta \lambda \approx \delta \lambda^N(\tau) = \sum_{l=0}^{N-1} \delta \lambda(\tau_l) L_l(\tau),$$

where $\tau_0 = -1$ and $\tau_l (l = 1, 2, \dots, N)$ are N LGR points distributed at the interval $(-1, 1]$, which are defined as zero points of $\dot{P}_N(\tau) - \dot{P}_{N-1}(\tau)$. \dot{P}_N is Legendre polynomial of order N . And L_i is Lagrange interpolation polynomial basis function:

$$L_i = \prod_{j=0, j \neq i}^K \frac{\tau - \tau_j}{\tau_i - \tau_j}. \quad (47)$$

The derivatives of $\delta \mathbf{x}^N(\tau)$ and $\delta \lambda^N(\tau)$ at the point of τ_l are obtained by differentiating (46) as follows:

$$\begin{aligned} \delta \dot{\mathbf{x}}(\tau_k) &\approx \delta \dot{\mathbf{x}}^N(\tau_k) = \sum_{l=0}^{N-1} \delta \mathbf{x}(\tau_l) \dot{L}_l(\tau) \\ &= \sum_{l=0}^{N-1} D_{kl} \delta \mathbf{x}(\tau_l), \\ \delta \lambda(\tau_k) &\approx \delta \lambda^N(\tau_k) = \sum_{l=0}^{N-1} \delta \lambda(\tau_l) \dot{L}_l(\tau) \\ &= \sum_{l=0}^{N-1} D_{kl} \delta \lambda(\tau_l), \end{aligned} \quad (48)$$

where D_{kl} is the difference matrix of dimension $(N+1) \times N$:

$$D_{kl} = \begin{cases} \frac{\dot{q}(\tau_k)}{(\tau_k - \tau_l) \dot{q}(\tau_l)}, & l \neq k, \\ \frac{\ddot{q}(\tau_l)}{2\dot{q}(\tau_l)}, & l = k, \end{cases} \quad (49)$$

$$q(\tau_l) = (1 + \tau_l) [P_N(\tau_l) - P_{N-1}(\tau_l)].$$

The TPBVPs of $\delta \mathbf{x}, \delta \lambda$ are transformed into the following algebraic equations:

$$\begin{aligned} \sum_{j=1}^N D_{ij} \delta \mathbf{x}_j(\tau) \\ - \frac{t_f - t_0}{2} (\mathbf{A}_i(\tau) \delta \mathbf{x}_i(\tau) - \mathbf{B}_i(\tau) \delta \lambda_i(\tau)) = \mathbf{0}, \end{aligned} \quad (50)$$

$$\begin{aligned} \sum_{j=1}^N D_{ij} \delta \lambda_j(\tau) \\ + \frac{t_f - t_0}{2} (\mathbf{C}_i(\tau) \delta \mathbf{x}_i(\tau) + \mathbf{A}_i^T(\tau) \delta \lambda_i(\tau)) = \mathbf{0}, \end{aligned} \quad (51)$$

$$[\partial \Omega_{\mathbf{x}} \delta \mathbf{x} + \partial \Omega_{\lambda} \delta \lambda]_{\tau=\tau_0} = \mathbf{0}. \quad (52)$$

$\mathbf{0}$ represents the zero matrix with the corresponding dimension. For simplicity, substituting $\mathbf{X} = [\delta \mathbf{x}_0^T, \delta \mathbf{x}_1^T, \dots, \delta \mathbf{x}_{N-1}^T]^T$, $\Lambda = [\delta \lambda_0^T, \delta \lambda_1^T, \dots, \delta \lambda_{N-1}^T]^T$ into (50)~(51) we can get

$$\begin{aligned} \mathbf{E}\mathbf{X} - \frac{t_f - t_0}{2} \mathbf{F}\Lambda &= \mathbf{0}, \\ \frac{t_f - t_0}{2} \mathbf{G}\mathbf{X} + \mathbf{H}\Lambda &= \mathbf{0}, \end{aligned} \quad (53)$$

where $\mathbf{E}, \mathbf{F}, \mathbf{G}, \mathbf{H}$ are matrices of dimension $[n(N+1) \times nN]$:

$$\begin{aligned} [E]_{ij} &= \begin{cases} D_{ij} \mathbf{I}_n & i \neq j \\ D_{ij} \mathbf{I}_n - \frac{t_f - t_0}{2} \mathbf{A}_i & i = j, \end{cases} \\ [F]_{ij} &= \begin{cases} \mathbf{0}_n & i \neq j \\ \mathbf{B}_i & i = j, \end{cases} \\ [G]_{ij} &= \begin{cases} \mathbf{0}_n & i \neq j \\ \mathbf{C}_i & i = j, \end{cases} \\ [H]_{ij} &= \begin{cases} D_{ij} \mathbf{I}_n & i \neq j \\ D_{ij} \mathbf{I}_n + \frac{t_f - t_0}{2} \mathbf{A}_i^T & i = j, \end{cases} \end{aligned} \quad (54)$$

where $\mathbf{0}_n$ and \mathbf{I}_n are zero and identity matrices of dimension $n \times n$. The purpose of this algorithm is to solve (53) subjected

to the boundary conditions (52). These equations can be summarized in matrix form as

$$\begin{bmatrix} \Gamma_1 & \Gamma_2 \\ \frac{t_f - t_0}{2} \mathbf{G} & \mathbf{H} \\ \mathbf{E} & -\frac{t_f - t_0}{2} \mathbf{F} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \Lambda \end{bmatrix} = \mathbf{VZ} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad (55)$$

where $\mathbf{Z}^T = [\mathbf{X}^T, \Lambda^T]$, Γ_1 and Γ_2 are matrices of dimension $n \times n(N+1)$. $\Gamma_1 = [\mathbf{0}_n, \dots, \mathbf{0}_n, \partial\Omega/\partial\mathbf{x}]$, $\Gamma_2 = [\mathbf{0}_n, \dots, \mathbf{0}_n, \partial\Omega/\partial\lambda]$. Dividing \mathbf{V} and \mathbf{Z} as $\mathbf{V} = [\mathbf{V}_e \ \mathbf{V}_f]$ and $\mathbf{Z} = [\mathbf{Z}_e \ \delta\lambda_f]$ we get

$$\mathbf{V}_e \mathbf{Z}_e + \mathbf{V}_f \delta\lambda_N = \mathbf{0}, \quad (56)$$

where \mathbf{V}_e of dimension $[n(2N+1) \times n(2N+1)]$ and \mathbf{V}_f of dimension $n(2N+1) \times n$ are block matrices of \mathbf{V} . $\mathbf{Z}_e = [\delta\mathbf{x}_0^T, \delta\mathbf{x}_1^T, \dots, \delta\mathbf{x}_N^T, \delta\lambda_0^T, \dots, \delta\lambda_{N-1}^T]^T$ is a vector of dimension $n(2N+1)$, which can be obtained by solving

$$\mathbf{Z}_e = -\mathbf{V}_e \setminus \mathbf{V}_f \delta\lambda_N = \mathbf{W} \delta\lambda_N. \quad (57)$$

Since $\mathbf{Z} = [\delta\mathbf{x}_0^T, \dots, \delta\mathbf{x}_N^T, \delta\lambda_0^T, \dots, \delta\lambda_N^T]^T$, we can get

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_e \\ \delta\lambda_N \end{bmatrix} = \begin{bmatrix} \mathbf{W} \\ \mathbf{I}_n \end{bmatrix} \delta\lambda_N = \begin{bmatrix} \mathbf{W}_x \\ \mathbf{W}_\lambda \end{bmatrix} \delta\lambda_N, \quad (58)$$

where \mathbf{W}_x and \mathbf{W}_λ of dimension $n(N+1) \times n$ are the block matrices of $[\mathbf{W} \ \mathbf{I}_n]^T$. Then we have

$$\begin{aligned} \delta\mathbf{x}_k &= \mathbf{W}_{xk} \delta\lambda_N, \\ \delta\lambda_k &= \mathbf{W}_{\lambda k} \delta\lambda_N, \end{aligned} \quad (59)$$

where \mathbf{W}_{xk} of dimension $n \times n$ is block matrix of \mathbf{W}_x and $\mathbf{W}_{\lambda k}$ of dimension $n \times n$ is block matrix of \mathbf{W}_λ . Subscript k represents the k th LGR point.

Substituting (59) into (19), we can get the optimal control modifications:

$$\delta\mathbf{u}(\tau_i) = -\mathbf{H}_{uu}^{-1} (\mathbf{H}_{ux} \mathbf{W}_{xi} + \mathbf{f}_u^T \mathbf{W}_{\lambda i}) \delta\lambda_N. \quad (60)$$

Since $\delta\lambda_N = \delta\lambda_f$ has been solved in the last section, the modifications of states, costates, and controls at the LGR points can be obtained by (59)-(60) and the values at instants of time between the nodes can be obtained by interpolation. The above solutions are obtained without a wide range of iterative optimization process, so it can ensure the high computational efficiency.

The modified algorithm can be concluded as Figure 2 and specific procedures are as follows:

- (1) Considering various process constraints and terminal constraints and selecting the maximum velocity as the performance index, we can establish the optimal model according to the dynamic equations of interceptor.
- (2) After further differentiating the canonical equations and coupled equations to second order, we can get the expression of feedback control $\delta\mathbf{u}$ and the TPBVP equations of $\delta\mathbf{x}, \delta\lambda$.

(3) After further differentiating the boundary conditions at unspecified terminal time to second order, we can get the expressions of $\delta\mathbf{x}, \delta\lambda$ in terms of free variables $d\mu$ and terminal constraints $d\psi_f$; the relationship between $\delta\mathbf{x}_0, \delta\lambda_0$ and $\delta\mathbf{x}_f, \delta\lambda_f$ is established by backward recursion, which can be used to deduce the expressions of $\delta\mathbf{x}_0, \delta\lambda_0$ in terms of $d\mu, d\psi_f$, and then we get the expressions of $\delta\lambda_f, dt_f$ in terms of $\delta\mathbf{x}_0, d\psi_f$.

- (4) The flight time of original nominal trajectory t_f plus dt_f can be used as new flight time of the new nominal trajectory which can be obtained by interpolating the original nominal trajectory.
- (5) Regarding current states of the interceptor as initial constraint and $\delta\lambda_f$ as perturbation, we solve the feedback control $\delta\mathbf{u}$ using the improved IRPM and finally realize online trajectory optimization of midcourse guidance.

5. Simulations and Discussion

In order to verify the validity of the proposed algorithm, the following assumption of intercept operations is designed. Because the nonpropulsive phase is more difficult for trajectory adjustment, the terminal PIP is modified at the initial time of the nonpropulsive phase with altitude increasing 2 km and flight path angle increasing 5 deg. And the current tracking errors of flight path angle and altitude are 0.1 deg and 10 m.

The GPM has high precision in solving optimization problems. So when the tracking error generates or the terminal constraints change, the reoptimized trajectory by GPM, which is solved by MATLAB programs GPOPS [25], will be used as a standard optimal modified trajectory for comparison. The comparison of modified trajectory is shown in Figure 3. NOC represents the modified trajectory generated based on NOC theory and the improved IRPM solving algorithm designed in this paper. Nominal represents the nominal trajectory with the unchanged terminal constraints; GPOPS represents the modified trajectory reoptimized by GPM.

All simulation results of this paper are obtained on a Lenovo laptop with an Intel 2.50 GHz quad-core processor, using windows 7 operating system. All the codes are run under the MATLAB® R2014a environment.

From Figure 3 we know that the modified trajectory (NOC), which is substantially overlapping the reoptimized trajectory (GPOPS), can satisfy the changed terminal constraints caused by the changed PIP. And the flight time of modified trajectory is approximatively equal to that of the reoptimized trajectory by GPM. These results prove the high accuracy of the algorithm designed in this paper. In the initial time after modification, the error of angle of attack between the modified trajectory and the nominal trajectory is not large, no more than 0.5 deg, which represents that the angle of attack does not have a large jump and therefore ensures the stability of the control.

In order to verify the higher precision and efficiency of improved IRPM, the contrast simulation is designed as follows: in the fourth section, according to the tracking error

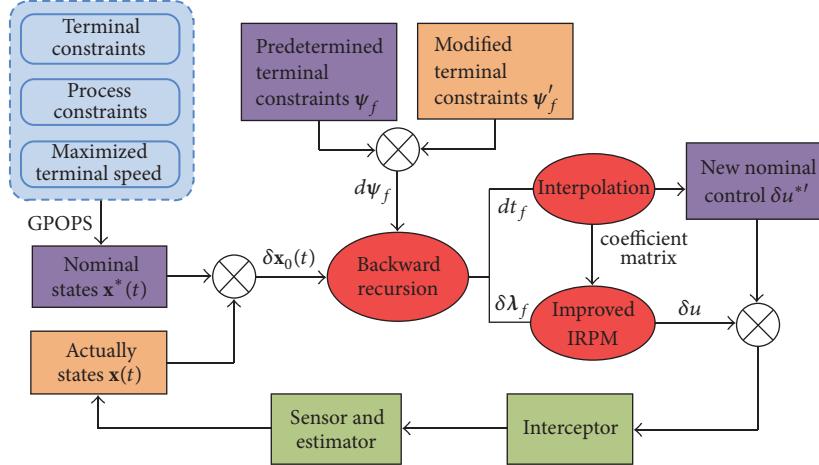


FIGURE 2: Online neighboring optimal modified algorithm.

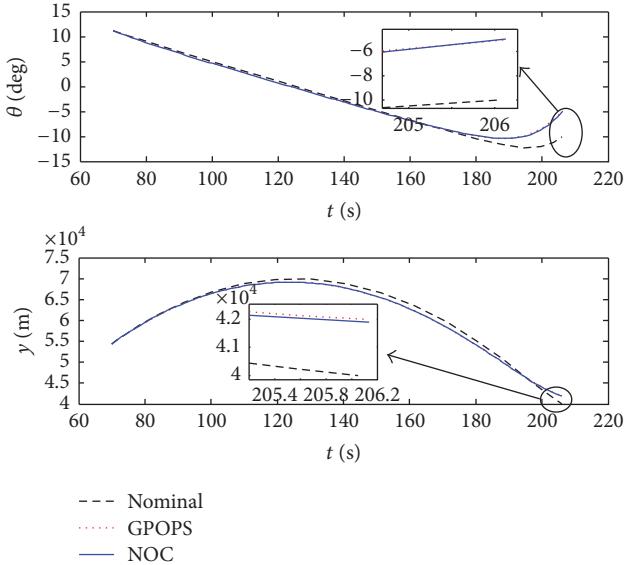


FIGURE 3: Curves of altitude and flight path angle.

and terminal constraints modifications, $\delta\mathbf{x}_f$ and $\delta\lambda_f$ are obtained by backward recursive algorithm. And then we solve feedback control variables by inversely integrating differential equations (19)–(21) of trajectory modification model rather than improved IRPM. In the integration process, 50, 500, and 1000 nodes are, respectively, selected at the same time interval. The curves of flight path angle modification and altitude modification are shown in Figures 4–6; the comparison of time consumption and terminal constraints modifications is shown in Table 2.

From Figures 4 and 5 and Table 2 we know that compared with the modified trajectory solved by improved IGPM with that reoptimized by GPM, the most part of curves is close except for initial several seconds after modification with flight path angle error of no more than 0.15 degrees and altitude error of no more than 120 m. Furthermore, the two methods have similar precision in terminal modifications, but the time

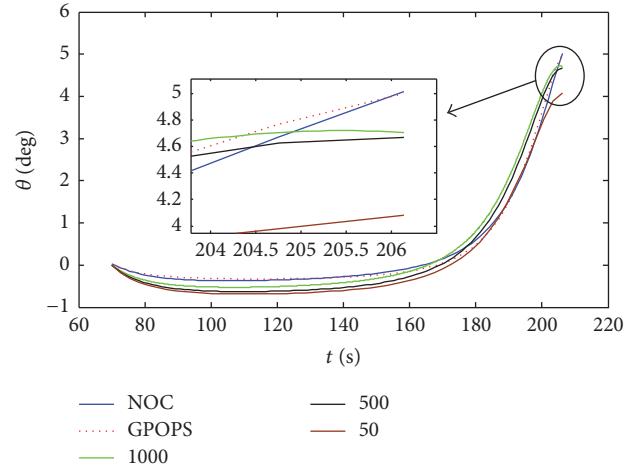


FIGURE 4: Comparison of flight path angle modification.

consumption of the former is much shorter than the latter. The method of solving differential equations directly has relatively large errors compared to reoptimization by GPM. With the increase of the number of nodes, the errors decrease gradually and the precision of terminal modifications accordingly increases. But the time consumption increases as well. And even if 1000 nodes are selected, the precision of terminal modifications still has a certain gap compared with improved IGPM while time consumption increases a lot.

In summary, the concept of modified trajectory design using nominal trajectory information can guarantee the modified precision with improving efficiency a lot; and the solving method based on improved IGPM method is superior to the method by solving differential equations directly in accuracy and efficiency.

In order to verify the adaptivity of modified algorithm, the simulation is designed as follows: the value of δy is set randomly within $[-3000, 3000]$, and the value of $\delta\theta$ is set from the -5 deg to 5 deg at the interval of 1 deg, which adds up to 11 sets of different changed terminal constraints. The

TABLE 2: Comparison of time consumption terminal modification.

Method	GPOPS	NOC	50	500	1000
t/s	7.116	0.049	0.144	1.248	2.532
$\delta\theta_f/\text{deg}$	5.00	4.994	4.081	4.667	4.707
$\delta y_f/\text{m}$	2000	1924	1391	2376	2144

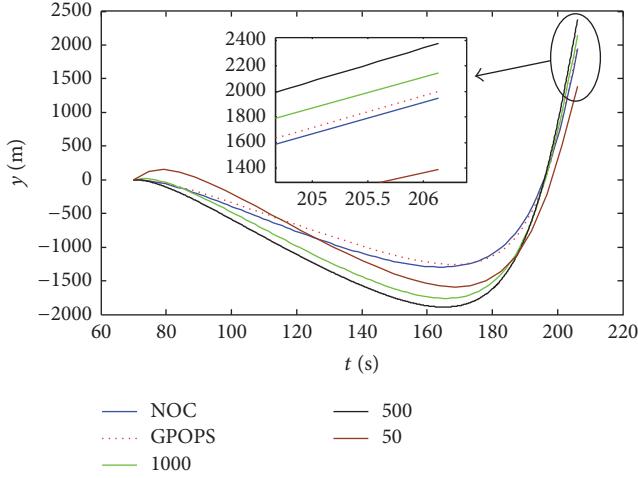


FIGURE 5: Comparison of altitude modification.

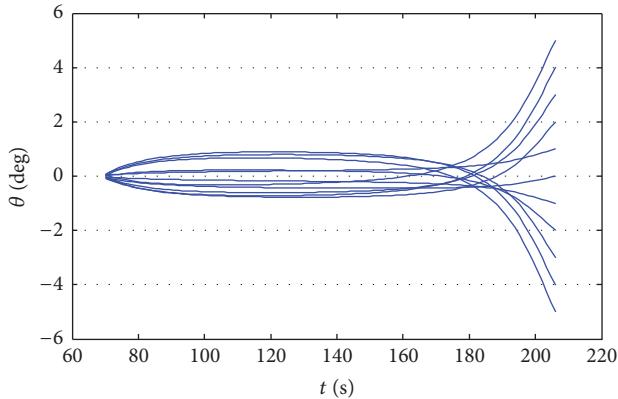


FIGURE 6: Flight path angle modifications with different terminals.

results are shown in Figure 7. Similarly, the value of $\delta\theta$ is set randomly within $[-5, 5]$, the value of δy is set from the -3000 m to 3000 m at the interval of 600 m which adds up to 11 sets of different changed terminal constraints. The results are shown in Figure 7. We can know from Figures 6 and 7 that the algorithm has the modified ability for different changes of terminal constraints. All initial errors of altitude are close to 0 and the maximal initial error of flight path angle is no more than 0.1 deg , which can be adjusted by the trajectory tracking control process.

$\delta\theta$ is set as 10 deg and δy is set to 0; the curve of modifications is shown in Figure 8. Similarly, δy is set to 8000 m and $\delta\theta$ is set to 0; the curve of modifications is shown

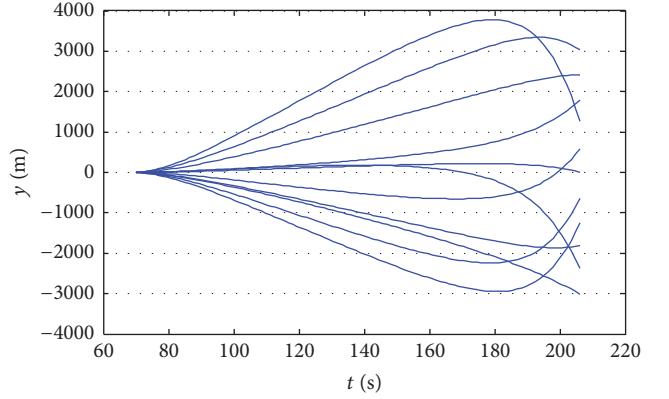


FIGURE 7: Altitude modifications with different terminals.

in Figure 9. According to Figures 8 and 9 we can know that when terminal constraints change largely, the error between the modified trajectory generated by the algorithm of this paper and the reoptimized trajectory by GPM is relatively large, which means the optimality of modified trajectory is difficult to guarantee. Because the modified algorithm of this paper is based on the NOC theory, when the changes of terminal constraints are beyond the neighborhood of nominal trajectory, the algorithm is no longer applicable.

6. Conclusion

An online trajectory optimization algorithm of midcourse guidance for hypersonic interception is designed based on the NOC theory and the improved IGPM. The main conclusions are as follows:

- (1) The algorithm considers not only the tracking error but also the change of terminal constraints. Therefore, it has the disturbance rejection ability and the ability to adjust terminal constraints online.
- (2) By using the original nominal trajectory information, a modified trajectory with higher accuracy can be rapidly generated based on neighboring optimal control theory. The iterative optimization process is avoided, and the trajectory generation is realized online.
- (3) When the changes of terminal constraints are relatively large, beyond the neighborhood of the original nominal trajectory, the online modified algorithm is no longer applicable. It needs to reoptimize online to get the modified trajectory.

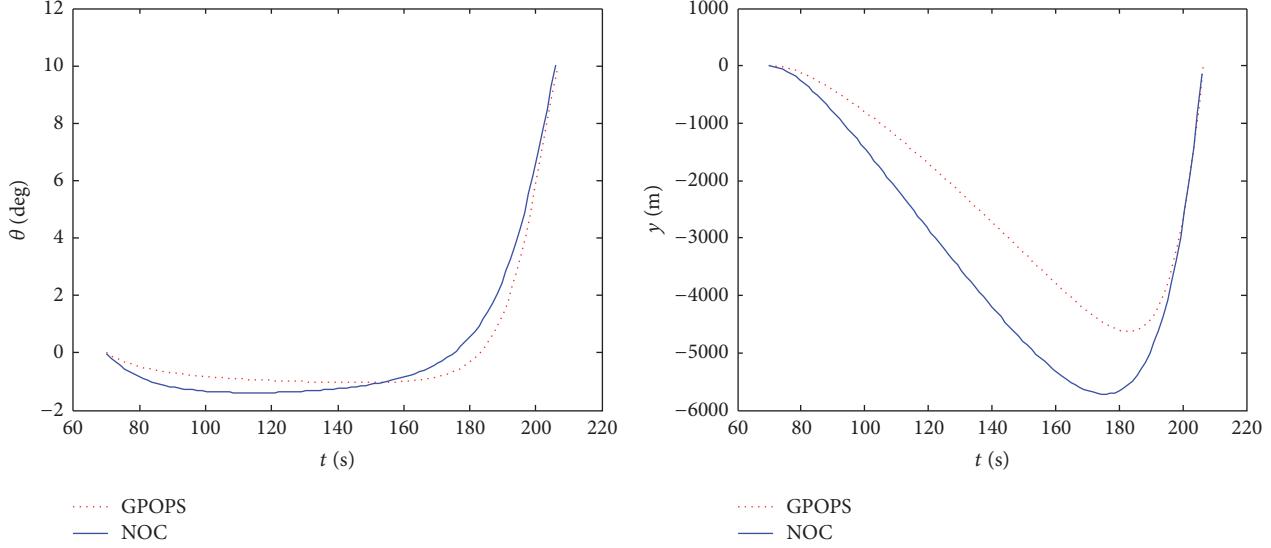


FIGURE 8: Curves of modifications with flight path angle increasing 10 deg.

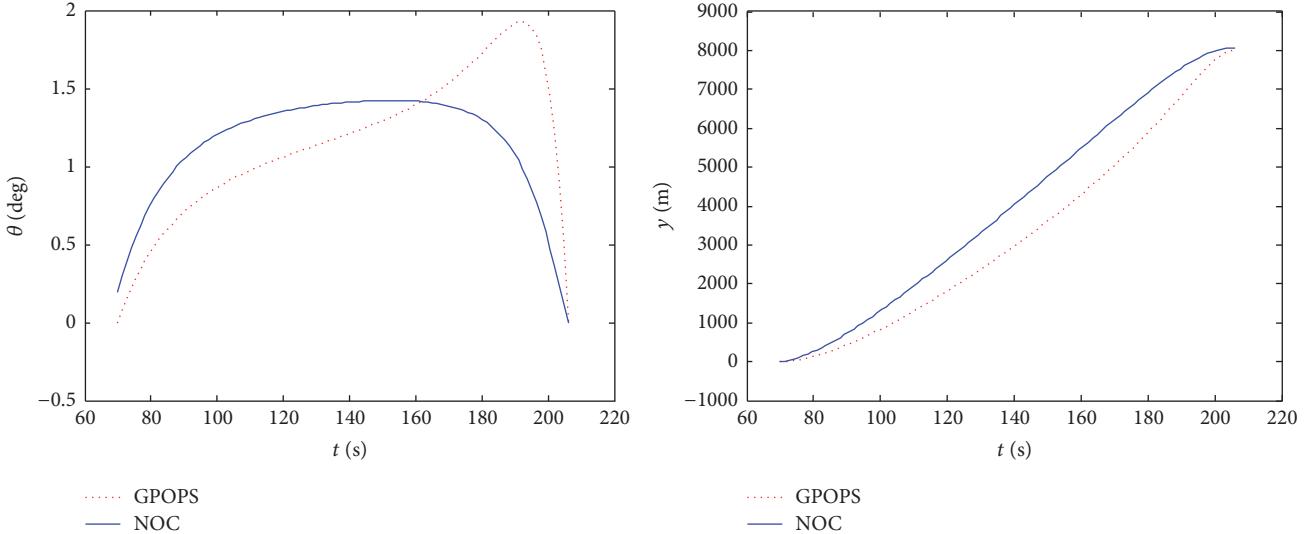


FIGURE 9: Curves of modifications with altitude increasing 8000 m.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was cosupported by the National Natural Science Foundation of China (Grant no. 61573374 and no. 61503408) and Aeronautical Science Foundation of China (Grant no. 20140196004 and no. 20150196006).

References

- [1] J. J. Bertin and R. M. Cummings, "Fifty years of hypersonics: where we've been, where we're going," *Progress in Aerospace Sciences*, vol. 39, no. 6-7, pp. 511–536, 2003.
- [2] S. M. Richman, A. J. Kenyon, and R. M. Sega, "High speed and hypersonic science and technology," in *Proceedings of the 41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference Exhibit*, pp. 1–9, Tucson, Ariz, USA, 2005.
- [3] G. Z. Feng, Q. C. Guan, and H. W. Zhang, *An Introduction to Hypersonic Aircraft*, Beijing Institute of Technology Press, Beijing, China, 2016.
- [4] R. W. Morgan and A. P. Noseck, "Generalized optimal mid-course guidance," in *Proceedings of the 2014 IEEE 53rd Annual Conference on Decision and Control (CDC)*, pp. 260–264, Los Angeles, CA, USA, December 2014.
- [5] S. Ghosh, D. Ghose, and S. Raha, "Capturability analysis of a 3-D retro-PN guidance law for higher speed nonmaneuvering targets," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 5, pp. 1864–1874, 2014.

- [6] N. Indig, J. Z. Ben-Asher, and N. Farber, "Near-optimal spatial midcourse guidance law with an angular constraint," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 1, pp. 214–223, 2014.
- [7] C. F. Lin and L. L. Tsai, "Analytical solution of optimal trajectory-shaping guidance," *Journal of Guidance, Control, and Dynamics*, vol. 10, no. 1, pp. 60–66, 1987.
- [8] C. F. Lin, *Modern Navigation Guidance, and Control Processing*, Prentice Hall, Saddle River, NJ, USA, 1991.
- [9] N. Indig, J. Z. Ben-Asher, and N. Farber, "Near-optimal spatial mid-course guidance law with angular constraint," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, August 2012.
- [10] N. Indig, J. Z. Ben-Asher, and E. Sigal, "Near-optimal minimum time guidance under a spatial angular constraint in atmospheric flight," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, January 2015.
- [11] P. N. Dwivedi, A. Bhattacharya, and R. Padhi, "Suboptimal midcourse guidance of interceptors for high-speed targets with alignment angle constraint," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 3, pp. 860–877, 2011.
- [12] O. Halbe, R. G. Raja, and R. Padhi, "Robust reentry guidance of a reusable launch vehicle using model predictive static programming," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 1, pp. 134–148, 2014.
- [13] O. A. Yakimenko, "Direct method for rapid prototyping of near-optimal aircraft trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 5, pp. 865–875, 2000.
- [14] J. A. Lukacs and O. A. Yakimenko, "Trajectory-shape-varying missile guidance for interception of ballistic missiles during the boost phase," in *Proceedings of AIAA Guidance, Navigation and Control Conference and Exhibit, 2007*, AIAA 2007-6538, pp. 2181–2201, August 2007.
- [15] G. A. Boyarko, R. Marcello, and O. A. Yakimenko, "Time-optimal reorientation of a spacecraft using an inverse dynamics optimization method," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 4, pp. 1197–1207, 2011.
- [16] I. M. Ross and F. Fahroo, "A perspective on methods for trajectory optimization," in *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference and Exhibit*. Monterey: AIAA Press, AIAA Paper No. 2002-4727, pp. 1–7, USA, August 2002.
- [17] H. Peng, X. Wang, and M. Li, "An hp symplectic pseudospectral method for nonlinear optimal control," *Communications in Nonlinear Science & Numerical Simulation*, vol. 42, pp. 623–644, 2017.
- [18] F. B. Wang and C. H. Dong, "Fast intercept trajectory optimization for multi-stage air defense missile using hybrid algorithm," in *Proceedings of the 7th Asian-Pacific Conference on Aerospace Technology and Science, APCATS 2013*, pp. 447–456, May 2013.
- [19] J. Zhao and R. Zhou, "Particle swarm optimization applied to hypersonic reentry trajectories," *Chinese Journal of Aeronautics*, vol. 28, no. 3, pp. 823–831, 2015.
- [20] S. M. Roberts and J. S. Shipman, *Two Point Boundary Value Problems: Shooting Methods*, Elsevier, New York, NY, USA, 1972.
- [21] F. Fahroo and I. Ross, "Trajectory optimization by indirect spectral collocation methods," in *Proceedings of the AIAA Paper 2000-4028, in: Proceedings of the AIAA/AAS Astrodynamics Specialist Conference*, Denver, Colo, USA.
- [22] H. Yan, F. Fahroo, and I. M. Ross, "Real-time computation of neighboring optimal control laws," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Monterey, Calif, USA.
- [23] E.-J. Song and M.-J. Tahk, "Real-time neural-network mid-course guidance," *Control Engineering Practice*, vol. 9, no. 10, pp. 1145–1154, 2001.
- [24] P. J. Mantle, *The Missile Defense Equation*, American Institute of Aeronautics and Astronautics, Inc., Washington, DC, USA, 2004.
- [25] V. A. Rao, D. Benson, T. G. Huntingdon et al., "Algorithm 902: GPOPS, a MATLAB software for solving multiple-phase optimal control problems using gauss pse- udospectral methods," *ACM Transactions on Mathematical Software*, vol. 37, no. 2, pp. 22–39, 2010.
- [26] A. E. Bryson and Y. C. Ho, *Applied Optimal Control*, Halsted Press, Australia, 1975.
- [27] A. E. Bryson, *Dynamic Optimization*, Addison Wesley, Menlo Park, Calif, USA, 1999.

