Journal of Systems Engineering and Electronics
Vol. 31, No. 5, October 2020, pp.1062 – 1076

A fast computational method for the landing footprints of space-to-ground vehicles

LIU Qingguo, LIU Xinxue, WU Jian*, and LI Yaxiong

Xi'an High-tech Institute, Xi'an 710025, China

Abstract: Fast computation of the landing footprint of a space-to-ground vehicle is a basic requirement for the deployment of parking orbits, as well as for enabling decision makers to develop real-time programs of transfer trajectories. In order to address the usually slow computational time for the determination of the landing footprint of a space-to-ground vehicle under finite thrust, this work proposes a method that uses polynomial equations to describe the boundaries of the landing footprint and uses back propagation (BP) neural networks to quickly determine the landing footprint of the space-to-ground vehicle. First, given orbital parameters and a manoeuvre moment, the solution model of the landing footprint of a space-to-ground vehicle under finite thrust is established. Second, given arbitrary orbital parameters and an arbitrary manoeuvre moment, a fast computational model for the landing footprint of a space-to-ground vehicle based on BP neural networks is provided. Finally, the simulation results demonstrate that under the premise of ensuring accuracy, the proposed method can quickly determine the landing footprint of a space-to-ground vehicle with arbitrary orbital parameters and arbitrary manoeuvre moments. The proposed fast computational method for determining a landing footprint lays a foundation for the parking-orbit configuration and supports the design of real-time transfer trajectories.

Keywords: space-to-ground vehicle, landing footprint, back propagation (BP) neural network, fast computational method, Pontryagin's minimum principle.

DOI: 10.23919/JSEE.2020.000080

1. Introduction

Spacecraft such as return satellites, manned spacecraft, space shuttles, and space-to-ground kinetic weapons, manoeuvring from their orbits to the earth surface are uniformly referred to as space-to-ground vehicles [1 – 3]. The landing footprint of the vehicle is an important indicator for assessing the ability of the space-to-ground vehicles. The fast computation of the landing footprint can lay a solid foundation for the work of a large number of repeated and real-time computations for the landing footprint, such as the parking-orbit configuration of the space-to-ground vehicle and the real-time programs of transfer trajectories developed by decision-makers. A typical space-to-ground transfer trajectory consists of a transition trajectory segment and a re-entry trajectory segment. Depending on the drag and lift of the spacecraft in the re-entry segment, the re-entry trajectories are classified as ballistic, semi-ballistic, or gliding [4,5].

The landing footprint consists of landing points on the boundaries. Each of the landing points is the result obtained by solving the transfer trajectory. Therefore, the optimization of the transfer trajectory is the basis of determination of the landing footprint. In the existing literature, direct methods [6,7], indirect methods [8,9], and hybrid methods [10 – 12] are used to study the optimization of transfer trajectories of different types of space-to-ground vehicles. The hybrid methods, one of which is adopted in this paper to optimize the transfer trajectory, combine the advantages of the direct and indirect methods.

Some methods have been proposed to study the landing footprint of a space-to-ground vehicle in the existing literature. Saraf et al. [13] proposed an analytical method for calculating the landing footprint based on the guidance mode of the space shuttle. Li et al. [14] proposed a method that uses scheduling of the drag profile to the normalized energy between the upper and lower bounds. This leads to finding the near and far edges of the landing zone. Hu et al. [15] used a genetic algorithm (GA) to compute the landing footprint. The existing literature on the landing footprint consists of solving for the plurality of points on the boundaries of the landing footprint, and then connecting these points on the boundary to determine the landing footprint. However, such methods are too long time consuming to efficiently solve the transfer trajectory of the space-to-ground vehicle under finite thrust, and to determine the

Manuscript received June 17, 2019.
*Corresponding author.
This work was supported by the National Natural Science Foundation of China (61603398).

landing footprint. In order to shorten the computational time, this work departs from the traditional methods of determining the landing footprint by connecting boundary points, and proposes a fast method, which uses polynomial equations to describe the boundaries of the landing footprint. Moreover, back propagation (BP) neural networks are used to quickly achieve a nonlinear mapping of the boundary of the landing footprint with arbitrary orbital parameters and arbitrary manoeuvre moments.

The research content of this paper is as follows: First, a solution model for the landing footprint of a space-to-ground vehicle under finite thrust is established. Second, given arbitrary orbital parameters and an arbitrary manoeuvre moment, a fast computational model for determining the landing footprint of the space-based vehicle based on BP neural networks is provided. Finally, the effectiveness of the method is demonstrated through simulation.

2. Computation of the landing footprints

In this work, the equations of motion are described in the coordinate system fixed to the earth. The determination of the landing footprint is performed with the longitude as the x-coordinate and the latitude as the y-coordinate. Given orbital parameters and a manoeuvre moment, the outlines of the solution model are as follows:

(i) Determine the minimum longitude or latitude of all landing points;

(ii) Determine the maximum longitude or latitude of all landing points;

(iii) Choose many points between the minimum and the maximum longitude or latitude in (i) and (ii), and determine the boundary points corresponding to the chosen points;

(iv) Connect all of the points obtained in (i), (ii) and (iii) to determine the landing footprint of the space-to-ground vehicle.

The transfer trajectories are required to compute the minimum longitude or latitude and the maximum longitude or latitude of all of the landing points, ae well as the boundary points between the minimum and the maximum longitude or latitude. The procedure for using a mixed method to solve the transfer trajectory problem is described by the following: Take the longitudes or latitudes of the landing points of (i), (ii) and (iii) as the optimization index. The transfer trajectory optimization problem is then converted into a two-point boundary value problem using Pontryagin's minimum principle [16 – 18]. The initial values of the adjoint variables and the values of the partial state variables are adjusted by means of a GA [19 – 21], and the transfer trajectory and the landing footprint are then obtained.

2.1 Motion differential equations in the earth-fixed coordinate system

The differential equations [22,23] of motion are

$$
\begin{cases}
\dfrac{\mathrm{d}\overline{V}}{\mathrm{d}\overline{t}} = \dfrac{\overline{T}\cos\alpha\cos\beta - \overline{X}}{\overline{m}} - \overline{g}_r\sin\gamma - \overline{g}_{\Phi}\cos\gamma\sin\psi + \overline{\omega}_e^2\overline{r}\cos\Phi(\sin\gamma\cos\Phi - \cos\gamma\sin\Phi\cos\psi) \\
\dfrac{\mathrm{d}\gamma}{\mathrm{d}\overline{t}} = \dfrac{\overline{T}\sin\alpha + \overline{Y}}{\overline{m}\overline{V}} + \dfrac{\overline{V}\cos\gamma}{\overline{r}} - \dfrac{\overline{g}_r\cos\gamma}{\overline{V}} + \dfrac{\overline{g}_{\Phi}\sin\gamma\sin\psi}{\overline{V}} + 2\overline{\omega}_e\cos\Phi\sin\psi + \\
\qquad \dfrac{\overline{\omega}_e^2\overline{r}\cos\Phi}{\overline{V}}(\cos\gamma\cos\Phi - \sin\gamma\sin\Phi\cos\psi) \\
\dfrac{\mathrm{d}\psi}{\mathrm{d}\overline{t}} = \dfrac{\overline{T}\cos\alpha\sin\beta - \overline{Z}}{\overline{m}\overline{V}\cos\gamma} - \dfrac{\overline{V}\cos\gamma\cos\psi\tan\Phi}{\overline{r}} - \dfrac{\overline{g}_{\Phi}\cos\psi}{\overline{V}\cos\gamma} - 2\overline{\omega}_e(\cos\Phi\cos\psi\tan\gamma - \sin\Phi) + \\
\qquad \dfrac{\overline{\omega}_e^2\overline{r}}{\overline{V}\cos\gamma}\sin\psi\sin\Phi\cos\Phi \\
\dfrac{\mathrm{d}\overline{r}}{\mathrm{d}\overline{t}} = \overline{V}\sin\gamma \\
\dfrac{\mathrm{d}\Theta}{\mathrm{d}\overline{t}} = \dfrac{\overline{V}\cos\gamma\cos\psi}{\overline{r}\cos\Phi} \\
\dfrac{\mathrm{d}\Phi}{\mathrm{d}\overline{t}} = \dfrac{\overline{V}\cos\gamma\sin\psi}{\overline{r}} \\
\dfrac{\mathrm{d}\overline{m}}{\mathrm{d}\overline{t}} = -\dfrac{\overline{T}}{\overline{V}_e}
\end{cases}
\tag{1}
$$

where $\overline{V}$ is the dimensionless velocity, γ is the velocity inclination angle, ψ is the course angle (the angle between the projection of the velocity vector on the local horizontal plane and the latitude tangent), $\overline{r}$ is the dimensionless geocentric distance, Θ is the longitude, Φ is the latitude, $\overline{T}$ is the engine thrust, $\overline{V}_e$ is the gas jet velocity, $\overline{m}$ is the dimen-

sionless spacecraft quality, $\overline{t}$ is the dimensionless time, $\overline{X}$ is the dimensionless drag, $\overline{Y}$ is the dimensionless lift, $\overline{Z}$ is the dimensionless lateral force, α is the angle of attack, β is the sideslip angle, $\overline{\omega}_e$ is the dimensionless earth rotation angular rate, $\overline{g}_r$ and $\overline{g}_{\Phi}$ are the dimensionless gravitational components when only the first three terms of the spherical harmonic expansion are considered. In the transition trajectory segment, the values of $\overline{X}$, $\overline{Y}$ and $\overline{Z}$ are zero. In the re-entry trajectory segment, the value of $\overline{T}$ is zero and the atmospheric model is the US standard atmosphere (1976).

The equations for the dimensionless parameters are as follows:

$$\begin{cases} \overline{V} = \dfrac{V}{V_{\rm ref}} \\ \overline{r} = \dfrac{r}{r_{\rm ref}} \\ \overline{T} = \dfrac{T}{m_{\rm ref} g_{\rm ref}} \\ \overline{V}_e = \dfrac{V_e}{V_{\rm ref}} \\ \overline{m} = \dfrac{m}{m_{\rm ref}} \\ \overline{t} = \dfrac{t}{t_{\rm ref}} \\ \overline{X} = \dfrac{X}{m_{\rm ref} g_{\rm ref}} \\ \overline{Y} = \dfrac{Y}{m_{\rm ref} g_{\rm ref}} \\ \overline{Z} = \dfrac{Z}{m_{\rm ref} g_{\rm ref}} \\ \overline{\omega}_e = \omega_e t_{\rm ref} \\ \overline{g}_r = \dfrac{1}{\overline{r}^2}[1 - \dfrac{3J_2}{2\overline{r}^2}(3\sin^2 \Phi - 1)] \\ \overline{g}_{\Phi} = \dfrac{3J_2 \sin 2\Phi}{2\overline{r}^4} \end{cases} \tag{2}$$

where J_2 is the coefficient of the second order principal spherical harmonic function and the value of J_2 is 1.082 63e–3. $r_{\rm ref}$, $m_{\rm ref}$, $V_{\rm ref}$, $t_{\rm ref}$ and $g_{\rm ref}$ are given by the following equations:

$$\begin{cases} r_{\rm ref} = r \\ m_{\rm ref} = m_0 \\ V_{\rm ref} = \sqrt{\dfrac{\mu}{r_{\rm ref}}} \\ t_{\rm ref} = \dfrac{r_{\rm ref}}{V_{\rm ref}} \\ g_{\rm ref} = \dfrac{\mu}{r_{\rm ref}^2} \end{cases} \tag{3}$$

where μ is the gravitation constant and the value of μ is 3.986 005e+14 $\rm m^3/s^2$. m_0 is the initial mass of the vehicle and R_E is the radius of the earth [24]:

$$R_E = a[0.998\,320\,047 + 0.001\,683\,494\cos(2\Phi) - 0.000\,003\,549\cos(4\Phi) + 0.000\,000\,008\cos(6\Phi)]. \tag{4}$$

The parameters used to describe the orbit of a space-to-ground vehicle in space are the orbital radius a, the flattening e, the orbital inclination i, the ascending node right ascension Ω, the perigee angle ω and the true anomaly f. The relationship between the orbit parameters and the motion parameters in the absolute coordinate system is

$$\begin{cases} V_I = \sqrt{\dfrac{\mu(1 + 2e\cos f + e^2)}{a(1 - e^2)}} \\ \gamma_I = \arctan \dfrac{e \sin f}{1 + e\cos f} \\ \Psi_I = \arctan[\tan i \cos(\omega + f)] \\ \Theta_I = \Omega + \arctan[\tan(\omega + f)\cos i] \\ \Phi_I = \arcsin[\sin(\omega + f)\sin i] \\ r_I = \dfrac{a(1 - e^2)}{1 + e\cos f} \\ f = \arcsin \dfrac{\sin \Phi_I}{\sin i} - \omega \end{cases} \tag{5}$$

where V_I, γ_I, Ψ_I, r_I, Θ_I and Φ_I are the velocity, the velocity inclination angle, the course angle, the geocentric distance, the longitude and the latitude in the absolute coordinate system, respectively.

The relation between the motion parameters in the absolute coordinate system and the motion parameters in the earth-fixed coordinate system are

$$\begin{cases} V = \sqrt{V_I - 2\omega e r_I V_I \cos\gamma_I \cos\psi_I \cos\Phi_I + (\omega_e r_I \cos\Phi_I)^2} \\ \tan\gamma = \dfrac{V_I \sin\gamma_I}{\sqrt{(V_I\cos\gamma_I)^2 - 2\omega_e r_I V_I \cos\gamma_I \cos\psi_I \cos\Phi_I + (\omega_e r_I \cos\Phi_I)^2}} \\ \tan\psi = \dfrac{V_I \cos\gamma_I \sin\psi_I}{V_I \cos\gamma_I \cos\psi_I - \omega_e r_I \cos\Phi_I} \\ r = r_I \\ \Theta = \Theta_I - \Gamma - \omega_e t \\ \Phi = \Phi_I \end{cases} . \tag{6}$$

2.2 Computational steps of the landing footprints

The optimization indexes of research ideas (i), (ii) and (iii) are given by

$$\min J_1 = \begin{cases} |\Theta_b - \Theta_m|, & 0^\circ \leqslant i \leqslant 45^\circ \text{ or} \\ & 135^\circ \leqslant i \leqslant 180^\circ \\ |\Phi_b - \Phi_m|, & 45^\circ < i < 135^\circ \end{cases} \tag{7}$$

$$\min J_2 = \begin{cases} -|\Theta_b - \Theta_m|, & 0^\circ \leqslant i \leqslant 45^\circ \text{ or} \\ & 135^\circ \leqslant i \leqslant 180^\circ \\ -|\Phi_b - \Phi_m|, & 45^\circ < i < 135^\circ \end{cases} \tag{8}$$

$$\min J_3 = \begin{cases} \Phi_b, & 0^\circ \leqslant i \leqslant 45^\circ \text{ or } 135^\circ \leqslant i \leqslant 180^\circ \\ -\Phi_b, & 0^\circ \leqslant i \leqslant 45^\circ \text{ or } 135^\circ \leqslant i \leqslant 180^\circ \\ -\Theta_b, & 45^\circ < i < 135^\circ \\ \Theta_b, & 45^\circ < i < 135^\circ \end{cases} \tag{9}$$

where J_1, J_2 and J_3 are optimization indexes corresponding to research ideas (i), (ii) and (iii), respectively. Θ_m and Φ_m are the longitude and the latitude of the manoeuvre point. The variables Θ_b and Φ_b are the longitude and the latitude of the landing point.

According to the Pontryagin's minimum principle, the Hamiltonian function is given by

$$H = \lambda_{\overline{V}}\dot{\overline{V}} + \lambda_\gamma\dot{\gamma} + \lambda_\psi\dot{\psi} + \lambda_{\overline{r}}\dot{\overline{r}} + \lambda_\Theta\dot{\Theta} + \lambda_\Phi\dot{\Phi} + \lambda_{\overline{m}}\dot{\overline{m}}. \tag{10}$$

From (10), the covariate variables satisfy the differential equations in (11). It should be noted that only the transition trajectory with the thrust control is considered and the thrust is constant in the computational model of the landing footprints.

$$\begin{cases} \dfrac{\mathrm{d}\lambda_{\overline{V}}}{\mathrm{d}\overline{t}} = \dfrac{\partial H}{\partial \overline{V}} \\ \dfrac{\mathrm{d}\lambda_\gamma}{\mathrm{d}\overline{t}} = \dfrac{\partial H}{\partial \gamma} \\ \dfrac{\mathrm{d}\lambda_\psi}{\mathrm{d}\overline{t}} = \dfrac{\partial H}{\partial \psi} \\ \dfrac{\mathrm{d}\lambda_{\overline{r}}}{\mathrm{d}\overline{t}} = \dfrac{\partial H}{\partial \overline{r}} \\ \dfrac{\mathrm{d}\lambda_\Theta}{\mathrm{d}\overline{t}} = \dfrac{\partial H}{\partial \Theta} \\ \dfrac{\mathrm{d}\lambda_\Phi}{\mathrm{d}\overline{t}} = \dfrac{\partial H}{\partial \Phi} \\ \dfrac{\mathrm{d}\lambda_{\overline{m}}}{\mathrm{d}\overline{t}} = \dfrac{\partial H}{\partial \overline{m}} \end{cases} \tag{11}$$

where the expressions of $\overline{g}_r^r, \overline{g}_\Phi^r, \overline{g}_r^\Phi$ and $\overline{g}_\Phi^\Phi$ are given by

$$\begin{cases} \overline{g}_r^r = -\dfrac{2\overline{g}_r}{\overline{r}} + \dfrac{3J_2}{\overline{r}^5}\left(3\sin^2\Phi - 1\right) \\ \overline{g}_\Phi^r = -\dfrac{6J_2\sin 2\Phi}{\overline{r}^5} \\ \overline{g}_r^\Phi = -\dfrac{9J_2\sin 2\Phi}{2\overline{r}^4} \\ \overline{g}_\Phi^\Phi = \dfrac{3J_2\cos 2\Phi}{\overline{r}^4} \end{cases}. \tag{12}$$

From the sufficient conditions of optimality and (8), the optimal thrust directions are obtained by

$$\begin{cases} \dfrac{\partial H}{\partial \alpha} = 0 = -\lambda_{\overline{V}}\dfrac{\overline{T}\sin\alpha\cos\beta}{\overline{m}} + \\ \qquad \lambda_\gamma\dfrac{\overline{T}\cos\alpha}{\overline{m}\overline{V}} - \lambda_\psi\dfrac{\overline{T}\sin\alpha\sin\beta}{\overline{m}\overline{V}\cos\gamma} \\ \dfrac{\partial H}{\partial \beta} = 0 = -\lambda_{\overline{V}}\dfrac{\overline{T}\cos\alpha\sin\beta}{\overline{m}} + \\ \qquad \lambda_\psi\dfrac{\overline{T}\cos\alpha\cos\beta}{\overline{m}\overline{V}\cos\gamma} \end{cases} \tag{13}$$

and

$$\begin{cases} \alpha = \arctan\dfrac{\mathrm{sgn}(\lambda_{\overline{V}})\cdot\lambda_\gamma\cos\gamma}{\sqrt{(\lambda_{\overline{V}}\overline{V}\cos\gamma)^2 + \lambda_\psi^2}} \\ \beta = \arctan\dfrac{\lambda_\psi}{\lambda_{\overline{V}}\overline{V}\cos\gamma} \end{cases} \tag{14}$$

where $\mathrm{sgn}(\ )$ is the sign function.

The transition trajectory satisfies the initial boundary conditions:

$$\begin{cases} \overline{V}(\overline{t}_0) - \overline{V}_0 = 0 \\ \gamma(\overline{t}_0) - \gamma_0 = 0 \\ \overline{r}(\overline{t}_0) - \overline{r}_0 = 0 \\ \psi(\overline{t}_0) - \psi_0 = 0 \\ \Theta(\overline{t}_0) - \Theta_0 = 0 \\ \Phi(\overline{t}_0) - \Phi_0 = 0 \\ \overline{m}(\overline{t}_0) - \overline{m}_0 = 0 \end{cases} \tag{15}$$

where $\overline{V}_0$, γ_0, ψ_0, $\overline{r}_0$, Θ_0, Φ_0 and $\overline{m}_0$ are the dimensionless velocity, the velocity inclination angle, the course angle, the dimensionless geocentric distance, the longitude, the latitude, and the dimensionless mass of space-to-ground vehicles at the dimensionless manoeuvre time $\overline{t}_0$, respectively.

The transition trajectory satisfies the terminal boundary constraints:

$$\begin{cases} \overline{V}(\overline{t}_a) - \overline{V}_a = 0 \\ \gamma(\overline{t}_a) - \gamma_a = 0 \\ \overline{r}(\overline{t}_a) - \overline{r}_a = 0 \end{cases} \tag{16}$$

where $\overline{V}_a$, γ_a and $\overline{r}_a$ are the dimensionless speed, the velocity inclination angle and the geocentric distance at the

terminal of the transition trajectory respectively; $\overline{t}_a$ is the dimensionless time at the terminal point of the transition trajectory.

Therefore, the cross-section conditions are given by

$$\begin{cases} \lambda_{\psi}(\overline{t}_a) = 0 \\ \lambda_{\Theta}(\overline{t}_a) = 0 \\ \lambda_{\Phi}(\overline{t}_a) = 0 \\ \lambda_{\overline{m}}(\overline{t}_a) = 0 \end{cases}. \tag{17}$$

When the optimization index J_3 is calculated, the reentry trajectory satisfies the constraints:

$$\begin{cases} \Theta(\overline{t}_b) = \Theta_b, & 0^\circ \leqslant i \leqslant 45^\circ \text{ or } 135^\circ \leqslant i \leqslant 180^\circ \\ \Phi(\overline{t}_b) = \Phi_b, & 45^\circ < i < 135^\circ \end{cases} \tag{18}$$

where Θ_b and Φ_b are the longitude and latitude of the landing point at the dimensionless time $\overline{t}_b$, respectively.

Because the optimization indexes do not include time,

$$H(\overline{t}_0) = H(\overline{t}) = H(\overline{t}_a) = 0. \tag{19}$$

In order to achieve the optimization indexes under the given constraints, 11 or 12 parameters are taken as optimization variables including the initial values of seven adjoint variables $\lambda_{\overline{V}}(\overline{t}_0)$, $\lambda_{\gamma}(\overline{t}_0)$, $\lambda_{\psi}(\overline{t}_0)$, $\lambda_{\overline{r}}(\overline{t}_0)$, $\lambda_{\Theta}(\overline{t}_0)$, $\lambda_{\Phi}(\overline{t}_0)$ and $\lambda_{\overline{m}}(\overline{t}_0)$, the values of four state variables $\overline{V}_a$, γ_a, $\overline{r}_a$ and $\overline{t}_a$ or the values of five state variables $\overline{V}_a$, γ_a, $\overline{r}_a$, $\overline{t}_a$ and $\Phi_b(\Theta_b)$. Since the initial value of an adjoint variable can be obtained from (10) and (19) and $\lambda_{\Theta}(\overline{t}_0) \equiv 0$, only nine or ten variables need to be optimized. A fourth order Runge-Kutta method [25,26] is used to compute the starts from (1) and (6) and the Adams predictor-corrector method is adopted to compute the remaining integral equations. The implementation steps of the GA are as follows:

Step 1 When optimization indexes J_1 and J_2 are calculated, $\lambda_{\overline{V}}(\overline{t}_0)$, $\lambda_{\gamma}(\overline{t}_0)$, $\lambda_{\psi}(\overline{t}_0)$, $\lambda_{\overline{r}}(\overline{t}_0)$, $\lambda_{\Theta}(\overline{t}_0)$, $\lambda_{\overline{m}}(\overline{t}_0)$, $\overline{V}_a$, γ_a and $\overline{t}_a$ are encoded. When optimization index J_3 is calculated, $\lambda_{\overline{V}}(\overline{t}_0)$, $\lambda_{\gamma}(\overline{t}_0)$, $\lambda_{\psi}(\overline{t}_0)$, $\lambda_{\overline{r}}(\overline{t}_0)$, $\lambda_{\Theta}(\overline{t}_0)$, $\lambda_{\overline{m}}(\overline{t}_0)$, $\overline{V}_a$, γ_a, $\overline{t}_a$ and $\Phi_b(\Theta_b)$ are encoded. Twenty chromosomes are generated randomly as the initial population. The crossover probability P_c, the mutation probability P_m and the maximum number of iterations N are set.

Step 2 The fitness function is shown in (7), (8) and (9). When a chromosome is determined, (11) is integrated to get the optimal thrust directions under the constraints (15) to (19). And (1) is then integrated to obtain the transfer trajectory. The fitness values of all the chromosomes in the current generation are computed.

Step 3 The chromosomes are selected using roulette wheel selection. Some genes on two different chromosomes reciprocally cross according to the crossover probability and others mutate according to the mutation probability. The execution of selection, crossover and mutation leads to the next generation population.

Step 4 Steps 2 and 3 are executed until the GA converges with $\varepsilon = 10^{-6}$ or the maximum number of iterations N is reached.

The hybrid method gives the transfer trajectories and landing footprints at a given manoeuvre point.

3. The proposed fast computational method of landing footprints

BP neural networks have the advantages of strong learning ability, good nonlinear mapping ability and good fault tolerance [27–29]. They consist of two sub-networks: the signal forward propagation network and the error BP network. The signal forward propagation network operates in such a way that the output results are obtained after the input parameters are processed layer by layer in the neural network. The error BP network operates in such a way that the output values are transmitted to the network in the opposite direction to modify the weight and threshold values between the neurons of the entire network, until the requirements of the output results are met. The signal forward propagation and error BP are called the BP neural networks training process.

In this section, the input and output parameters of the BP neural network are determined and then used to provide a model based on BP neural networks for the fast computation of landing footprints. Finally, a computational model of relative errors is given.

3.1 Determination of input and output parameters

The determination of the input and output parameters is the premise of the BP neural networks training. Polynomial equations (fitting curves) are adopted to describe the boundaries of the landing footprint, transforming the problem of determining the landing footprint into a problem of finding the coefficients of the polynomial equations. Given a manoeuvre moment of the space-to-ground vehicle, a polynomial equation can be used to approximately express the landing footprint. However, the landing footprint varies with manoeuvre moments. Thus, the BP neural networks can effectively solve the problem of an irregular change in landing footprints.

Fig. 1 shows the boundary of the landing footprints under the conditions $0^\circ \leqslant i \leqslant 45^\circ$ or $135^\circ \leqslant i \leqslant 180^\circ$, where i is the orbital inclination. Fig. 2 shows the boundary of the landing footprints under the conditions $45^\circ < i < 135^\circ$. In order to achieve a one-to-one correspondence between the horizontal and vertical coordinates by

means of polynomial equations, the coordinates should be interchanged under the conditions $45^\circ < i < 135^\circ$. The boundary is divided into a boundary on the high latitude (longitude) side and a boundary on the low latitude (longitude) side as shown in Fig. 1 and Fig. 2. The least squares method [30,31] is used to obtain the four fitting boundary curves in Fig. 1 and Fig. 2. These curves are viewed as the boundary of the landing footprints. The coefficients of the fitting curves are taken as the output parameters of the BP neural network. Polynomial equations of 4° are used in the least squares fitting method.

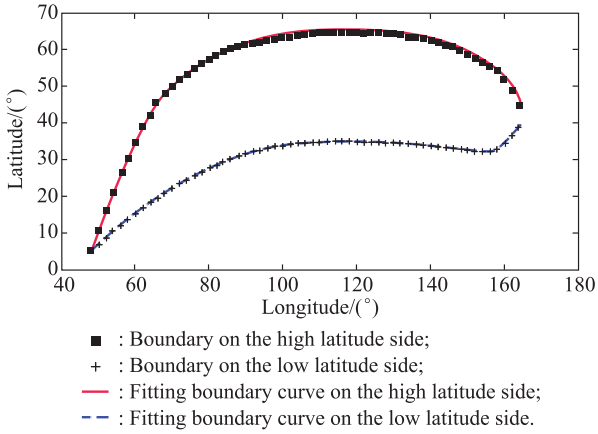


Fig. 1 Sketch diagram of the landing footprints when $0^\circ \leqslant i \leqslant 45^\circ$ or $135^\circ \leqslant i \leqslant 180^\circ$

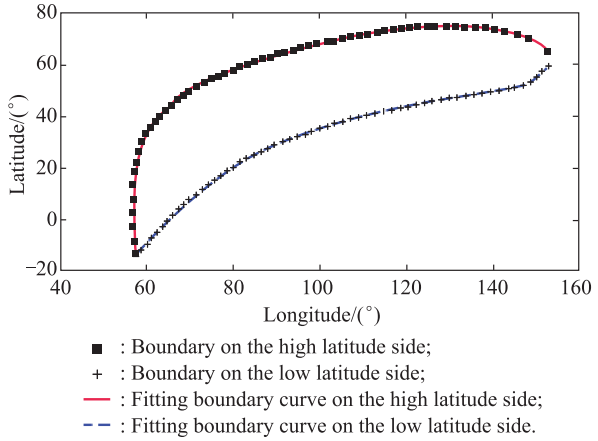


Fig. 2 Sketch diagram of landing footprints when $45^\circ < i < 135^\circ$

When $0^\circ \leqslant i \leqslant 45^\circ$ or $135^\circ \leqslant i \leqslant 180^\circ$, we get

$$y = mm_1^1 x^4 + mm_1^2 x^3 + mm_1^3 x^2 + mm_1^4 x + mm_1^5 \quad (20)$$

and

$$y = mm_2^1 x^4 + mm_2^2 x^3 + mm_2^3 x^2 + mm_2^4 x + mm_2^5 \quad (21)$$

where x is the longitude and y is the latitude; the values mm_1^1, mm_1^2, mm_1^3, mm_1^4 and mm_1^5 are the fitting coefficients of the boundary curve on the high latitude side; and the values mm_2^1, mm_2^2, mm_2^3, mm_2^4 and mm_2^5 are the fitting coefficients of the boundary curve on the low latitude side.

Similarly, when $45^\circ < i < 135^\circ$, we get

$$x = nn_1^1 y^4 + nn_1^2 y^3 + nn_1^3 y^2 + nn_1^4 y + nn_1^5 \quad (22)$$

and

$$x = nn_2^1 y^4 + nn_2^2 y^3 + nn_2^3 y^2 + nn_2^4 y + nn_2^5. \quad (23)$$

where the values nn_1^1, nn_1^2, nn_1^3, nn_1^4 and nn_1^5 are the fitting coefficients of the boundary curve on the high longitude side; and the values nn_2^1, nn_2^2, nn_2^3, nn_2^4 and nn_2^5 are the fitting coefficients of the boundary curve on the low longitude side.

Once (20), (21), (22) and (23) are determined, the coefficients of x^4, x^3, x^2, x and x^0 are taken as the output parameters of the BP neural network under the conditions $0^\circ \leqslant i \leqslant 45^\circ$ or $135^\circ \leqslant i \leqslant 180^\circ$. We define m_1^1, m_1^2, m_1^3, m_1^4 and m_1^5 as the output parameters of the boundary curve on the high latitude side, and m_2^1, m_2^2, m_2^3, m_2^4 and m_2^5 as the output parameters of the boundary curve on the low latitude side. Similarly the coefficients of y^4, y^3, y^2, y and y^0 are taken as the output parameters under the conditions of $45^\circ < i < 135^\circ$. The polynomial equations with the coefficients obtained from the BP neural network are shown in (24), (25), (26) and (27).

When $0^\circ \leqslant i \leqslant 45^\circ$ or $135^\circ \leqslant i \leqslant 180^\circ$,

$$y = m_1^1 x^4 + m_1^2 x^3 + m_1^3 x^2 + m_1^4 x + m_1^5 \quad (24)$$

and

$$y = m_2^1 x^4 + m_2^2 x^3 + m_2^3 x^2 + m_2^4 x + m_2^5. \quad (25)$$

When $45^\circ < i < 135^\circ$,

$$x = n_1^1 y^4 + n_1^2 y^3 + n_1^3 y^2 + n_1^4 y + n_1^5 \quad (26)$$

and

$$x = n_2^1 y^4 + n_2^2 y^3 + n_2^3 y^2 + n_2^4 y + n_2^5. \quad (27)$$

The factors affecting the landing footprint are the orbital parameters and the manoeuvre moment. Viewing the earth as a homogeneous ellipsoid and considering the periodicity characteristic of the space-to-ground vehicle, the determination of the landing footprint in one period can represent the landing footprint at an arbitrary manoeuvre moment. The period T is given by

$$T = 2\pi\sqrt{\frac{a^3}{\mu}}. \quad (28)$$

The manoeuvre moment t_m is in the range $[k \cdot 2\pi\sqrt{a^3/\mu}, (k+1) \cdot 2\pi\sqrt{a^3/\mu}]$, where k is an arbitrary non-negative integer. The input parameters of the BP neural network are the orbital radius a, the flattening e, the orbital inclination i, the ascending node right ascension Ω, the perigee angle ω, the true anomaly f and the manoeuvre moment t_m.

The input parameters need to be normalized as follows:

$$\overline{D} = \frac{D - D_{\min}}{D_{\max} - D_{\min}} \tag{29}$$

where $D = [a, e, i, \Omega, \omega, f, t_m]$, $\overline{D}$, $D_{\min}$ and $D_{\max}$ are the normalized, minimum and maximum values of the input parameter, respectively.

3.2 Construction of the BP neural network model

We provide four BP neural network models, each of which consists of a three-layer network: input layer, output layer and hidden layer as shown in Fig. 3 – Fig. 6.

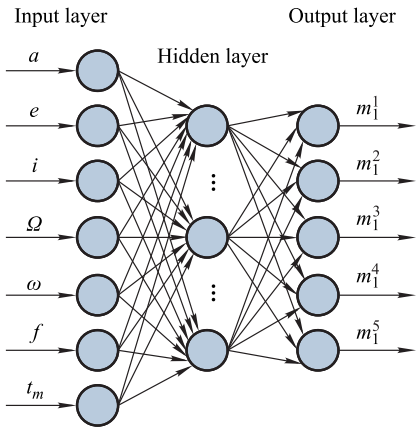


Fig. 3 No.1 BP neural network model

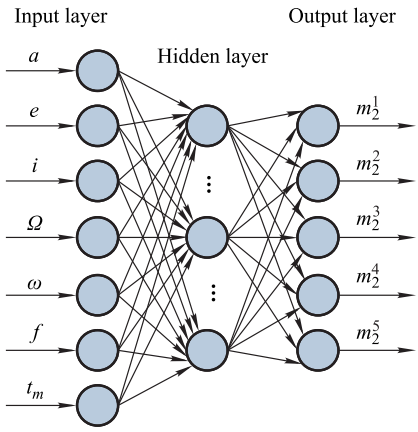


Fig. 4 No.2 BP neural network model

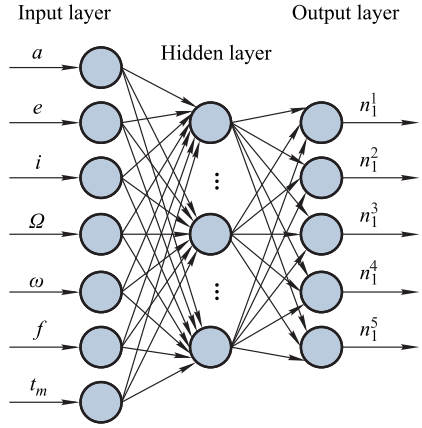


Fig. 5 No.3 BP neural network model

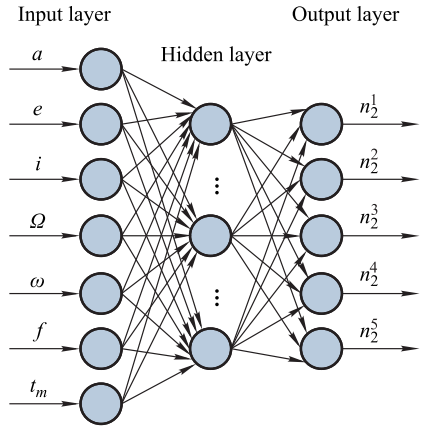


Fig. 6 No.4 BP neural network model

The differences of these four BP neural network models are the output parameters and the training data. The input layer of the four BP neural network models has seven nodes. The input parameters of the BP neural network are the orbital radius a, the flattening e, the orbital inclination i, the ascending node right ascension Ω, the perigee angle ω, the true anomaly f and the manoeuvre moment t_m. The four BP neural network models have five nodes in the output layer. The output parameters of the No.1 BP neural network model are m_1^1, m_1^2, m_1^3, m_1^4 and m_1^5; the output parameters of the No.2 BP neural network model are m_2^1, m_2^2, m_2^3, m_2^4 and m_2^5; the output parameters of the No.3 BP neural network model are n_1^1, n_1^2, n_1^3, n_1^4 and n_1^5; the output parameters of the No.4 BP neural network model are n_2^1, n_2^2, n_2^3, n_2^4 and n_2^5. The number of hidden layer nodes of the four neural network models is

$$l = \sqrt{p+q} + s \tag{30}$$

where p is the number of nodes in the input layer, q is the number of nodes in the output layer, and s is a natural number between 0 and 10. The training data of the four BP

neural network model are the actual footprints successively obtained under conditions of (24), (25), (26) and (27).

The signal forward propagation means that the signal received by the input layer is transmitted layer by layer until the output layer results are generated. The relevant mathematical expressions for signal forward propagation are shown in (31) – (34).

The input signal net_i of the number i node of the hidden layer is given by

$$\mathrm{net}_i = \sum_{j=1}^{p} w_{ij} x_j + \theta_i \tag{31}$$

where w_{ij} is the weight from the number j node of the input layer to the number i node of the hidden layer, x_j is the input parameter of the number j node of the input layer, and θ_i is the threshold of the number i node of the hidden layer.

The output signal o_i of the number i node of the hidden layer is given by

$$o_i = \phi(\mathrm{net}_i) = \phi\left(\sum_{j=1}^{p} w_{ij} x_j + \theta_i\right) \tag{32}$$

where $\phi(\ )$ is the Sigmoid function, an activation function commonly used in BP neural networks.

The input signal net_k of the number k node of the output layer is given by

$$\mathrm{net}_k = \sum_{i=1}^{l} t_{ki} o_i + a_k = \sum_{i=1}^{l}\left[t_{ki}\phi\left(\sum_{j=1}^{p} w_{ij} x_j + \theta_i\right)\right] + u_k \tag{33}$$

where t_{ki} is the weight from the number i node of the hidden layer to the number k node of the output layer, and u_k is the threshold of the number k node of the output layer.

The output signal o_k of the number k node of the output layer is given by

$$o_k = \psi(\mathrm{net}_k) = \psi\left(\sum_{i=1}^{l}\left[t_{ki}\phi\left(\sum_{j=1}^{p} w_{ij} x_j + \theta_i\right)\right] + u_k\right) \tag{34}$$

where $\psi(\ )$ is the Purelin function [32], another activation function commonly used in BP neural networks.

BP of the error means that the error computed from the output layer is back propagated to the hidden layer. The weight values w_{ij} and t_{ki} and threshold values θ_i and u_k are then adjusted using the gradient descent method related to the errors of the nodes in the hidden layer and the output layer. This process is iterated until the output of the modified network is close to the expected value. The mathematical expressions related to error BP are shown in (35) – (39).

For P training samples, the quadratic error criterion function E_P of the BP neural network is

$$E_P = \frac{1}{2}\sum_{p=1}^{P}\sum_{k=1}^{l}(T_k^p - o_k^p) \tag{35}$$

where T_k^p is the true value of the pth sample at the kth node in the output layer; and o_k^P is the corresponding value as computed by the BP neural network model.

The adjustment of the weight value Δw_{ij} in the hidden layer is performed by using

$$\Delta w_{ij} = \eta\sum_{p=1}^{P}\sum_{k=1}^{l}(T_k^p - o_k^p)\cdot\psi'(\mathrm{net}_k)\cdot t_{ki}\cdot\phi'(\mathrm{net}_i)\cdot\frac{\partial \mathrm{net}_i}{\partial w_{ij}}. \tag{36}$$

The adjustment of the threshold value $\Delta\theta_i$ in the hidden layer is performed by using

$$\Delta\theta_i = \eta\sum_{p=1}^{P}\sum_{k=1}^{l}(T_k^p - o_k^p)\cdot\psi'(\mathrm{net}_k)\cdot t_{ki}\cdot\phi'(\mathrm{net}_i). \tag{37}$$

The adjustment of the weight value Δt_{ki} in the output layer is implemented by using

$$\Delta t_{ki} = \eta\sum_{p=1}^{P}\sum_{k=1}^{l}(T_k^p - o_k^p)\cdot\psi'(\mathrm{net}_k)\cdot\frac{\partial \mathrm{net}_k}{\partial t_{ki}}. \tag{38}$$

The adjustment of the threshold value Δu_k in the output layer is implemented by using

$$\Delta u_k = \eta\sum_{p=1}^{P}\sum_{k=1}^{l}(T_k^p - o_k^p)\cdot\psi'(\mathrm{net}_k). \tag{39}$$

In (36) – (39), the parameter η is the learning rate.

The training error of the BP neural network propagates forward through the signal, and the BP of the error becomes iteratively smaller and smaller until stable weights and thresholds are obtained.

3.3 Computation of relative error

The relative error refers to the error between the landing footprint obtained by the fast computational method based on the BP neural network and the landing footprint obtained by the least squares method.

Fig. 7 shows the fitting boundary curves and the boundary curves obtained by the BP neural network. Fig. 8 shows the area of the landing footprints enclosed by the fitted curve. The regions S_1, S_2, S_3 and S_4 are the areas of the non-overlapping landing footprints enclosed by the fitting curves and the BP neural network.

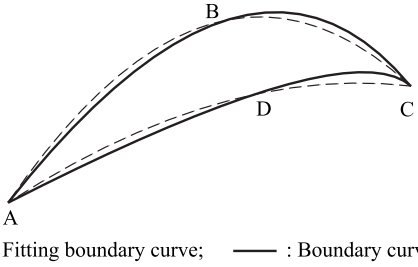


Fig. 7 Fitting boundary curves and the boundary curves obtained by the BP neural network

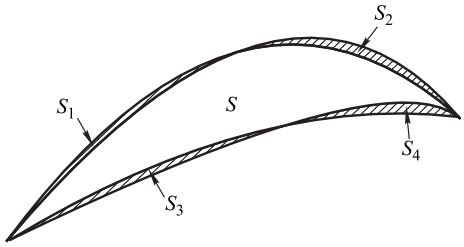


Fig. 8 Area of the landing footprints enclosed by the fitted curve

The relative error ΔS is calculated by using the following formula:

$$\Delta S = \frac{S_1 + S_2 + S_2 + S_4}{S}. \tag{40}$$

When $0^\circ \leqslant i \leqslant 45^\circ$ or $135^\circ \leqslant i \leqslant 180^\circ$, S refers to the area enclosed by the fitting curves of the (20) and (21), and the range of x is obtained by using (7) and (8). After combining (20), (21), (24) and (25) to obtain the intersections of four curves, S_1, S_2, S_3 and S_4 are obtained by computing the area enclosed by four curves with the longitudinal range between any two intersections.

When $45^\circ < i < 135^\circ$, S refers to the area enclosed by the fitting curves of the (22) and (23), and the range of x is obtained by the (7) and (8). After combining (22), (23), (26) and (27) to obtain the intersections of four curves, S_1, S_2, S_3 and S_4 are obtained by computing the area enclosed by four curves with latitudinal range between any two intersections.

In Fig. 7, four regions S_1, S_2, S_3 and S_4 are taken as the numerators of (40), and there are K regions that can be used as molecules, $S_1, S_2, \ldots, S_k$.

4. Simulations and analyses of results

4.1 Simulations

The values of the parameters of the space-to-ground vehicle simulations are as follows: the mass of the vehicle is 500 kg, the engine thrust is 100 N, the gas jet velocity is 3 000 m/s, the shape of the vehicle is an axisymmetric cone and the characteristic area of the vehicle is 0.02 m^2. The orbital radius a is in the range [6 700 km, 11 000 km], the flattening e is in the range [0,1), the value of the orbital inclination i is in the range [0°, 180°], the right ascension of ascending node Ω is in the range [0°, 360°], the argument of the perigee ω is in the range [0°, 360°], the true anomaly f is in the range [0°, 360°]. Each of the six parameters a, e, Ω, ω, f and t_m takes three random values within their respective range. The parameter i takes four random values, two in the range of $0^\circ \leqslant i \leqslant 45^\circ$ or $135^\circ \leqslant i \leqslant 180^\circ$ and the other two in the range of $45^\circ < i < 135^\circ$. The values ($2\,916 = 3^6 \times 4$ groups) are used as training data of the BP neural network. The population size is 20, the maximum number of iterations N is 50, the crossover probability P_c is 0.85 and the mutation probability P_m is 0.1. In order to verify the effectiveness of the proposed method, ten experiments are designed with parameters shown in Table 1. In the ten experiments, the total number of landing points on the boundary of each of the landing footprints is 60.

Table 1 Ten experiments

No.	a/km	e	$i/(^\circ)$	$\omega/(^\circ)$	$\Omega/(^\circ)$	$f/(^\circ)$	t_m/s
1	9 406	0.234 450	29.321	224.038	307.947	135.862	4 605.5
2	7 159	0.000 325	98.084	76.415	185.381	283.332	4 486.3
3	9 337	0.000 265	52.006	171.286	350.025	188.701	4 509.2
4	7 838	0.001 419	101.618	87.162	293.441	282.681	4 466.1
5	7 717	0.054 058	99.013 1	56.362	347.918	308.882	4 486.0
6	7 654	0.002 862	90.045	245.644	85.205	113.968	4 524.4
7	8 328	0.201 675	82.382	262.455	29.931	44.341	4 438.5
8	9 511	0.213 250	28.126	34.617	152.421	325.392	4 459.4
9	9 431	0.016 526	64.422	302.102	337.357	56.304	4 483.6
10	10 024	0.321 619	56.927	231.699	218.468	128.371	4 518.3

The learning rate of the four BP neural network models is 0.01, the maximum number of iterations is 2 000, the training error is 0.001, and the number of hidden layer nodes is ten. If the relative error is smaller than 2%, it is considered that the usage requirements are met.

Twenty-four blade servers are used for the simulations, and each of the blade servers has 10 blades. The program is performed using visual studio 2012. The main steps of the program are as follows:

Step 1 The training data of each space-to-ground vehicle is computed. At first, the minimum longitude or latitude and the maximum longitude or latitude of all of the landing points at one manoeuvre moment are computed by using two blades. Then, 58 blades are used to compute 58 landing points between the minimum longitude or latitude landing point and the maximum longitude or latitude landing point. Then, 60 manoeuvre moments are uniformly selected in one orbit period and the landing footprints at these manoeuvre moments are computed. This step corresponds to the theory in Section 2.

Step 2 The boundary curve and its coefficients are obtained by least squares fitting. This step is described in de-

tail in Section 3.1.

Step 3 BP neural network training. This step is described in Sections 3.2 – 3.3.

Step 4 Test the effectiveness of the proposed method.

Methods proposed by Li et al. [14] and Hu et al. [15] are used to compare to the proposed method. The parameters of the simulation are taken from [14,15].

4.2 Analyses of results

Fig. 9 – Fig. 18 show the boundaries of, the fitting boundary curves of and the boundary curves of each landing footprint as obtained by the BP neural networks in the ten sets of testing experiments. As can be seen from Fig. 9 – Fig. 18, the boundary curves obtained by the BP neural network, the fitting boundary curves and the boundaries of the landing footprints are basically the same.

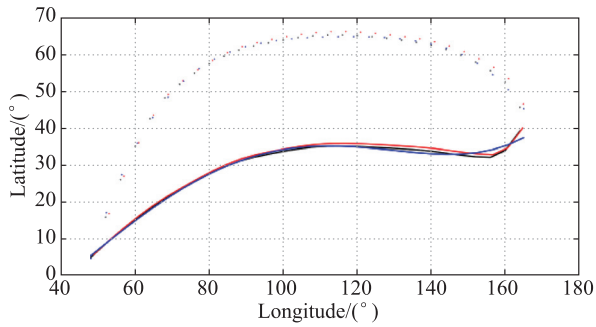


Fig. 9 Landing footprint of No.1 experiment

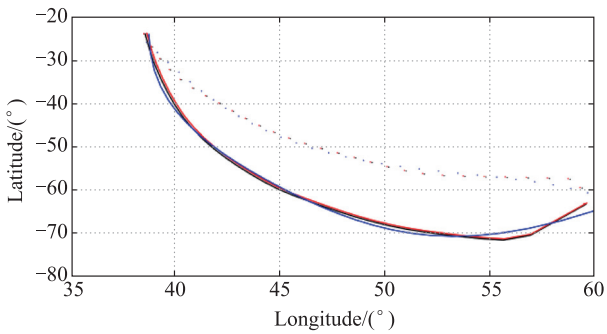


Fig. 10 Landing footprint of No.2 experiment

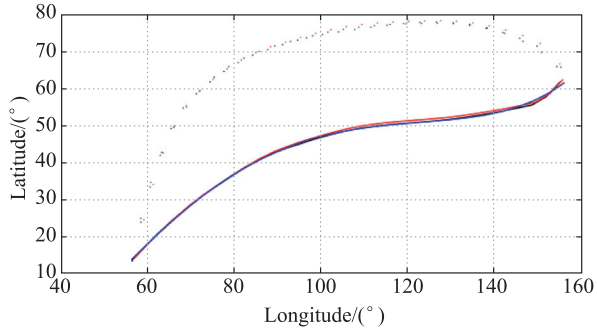


Fig. 11 Landing footprint of No.3 experiment

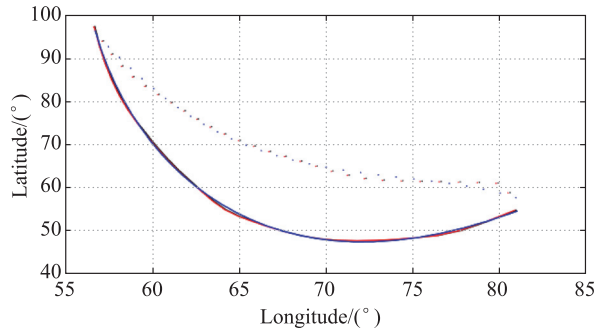


Fig. 12 Landing footprint of No.4 experiment

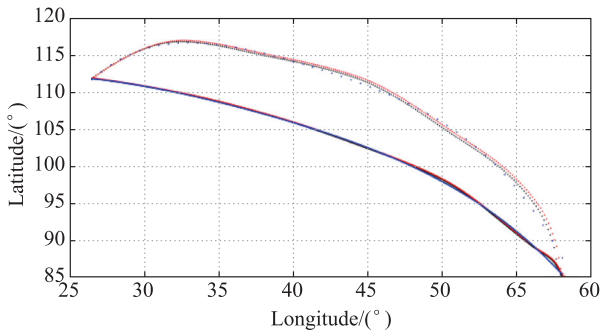
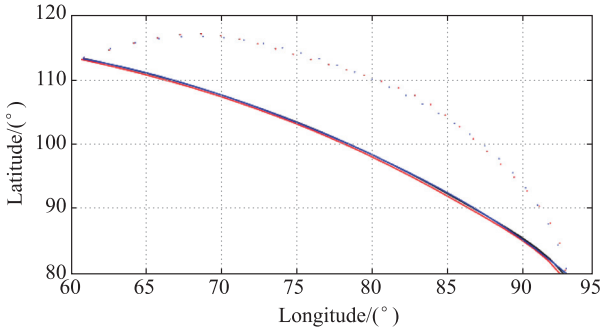
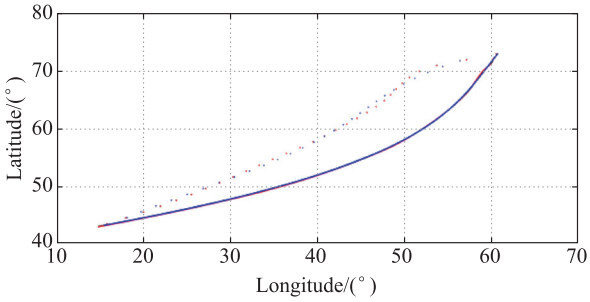


Fig. 13 Landing footprint of No.5 experiment



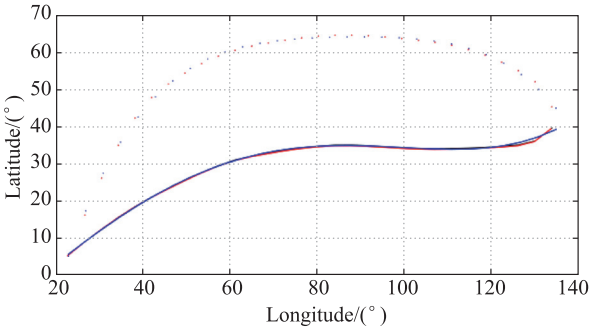
• : Boundary on the high latitude side;
— : Boundary on the low latitude side;
• : Fitting boundary curve on the high latitude side;
— : Fitting boundary curve on the low latitude side;
• : Boundary curve on the high latitude side (BP);
— : Boundary curve on the low latitude side (BP).

Fig. 14 Landing footprint of No.6 experiment



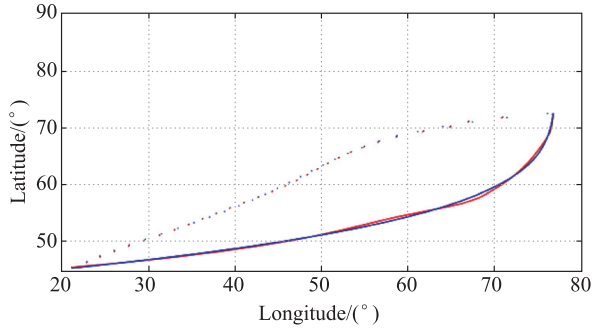
• : Boundary on the high latitude side;
— : Boundary on the low latitude side;
• : Fitting boundary curve on the high latitude side;
— : Fitting boundary curve on the low latitude side;
• : Boundary curve on the high latitude side (BP);
— : Boundary curve on the low latitude side (BP).

Fig. 15 Landing footprint of No.7 experiment



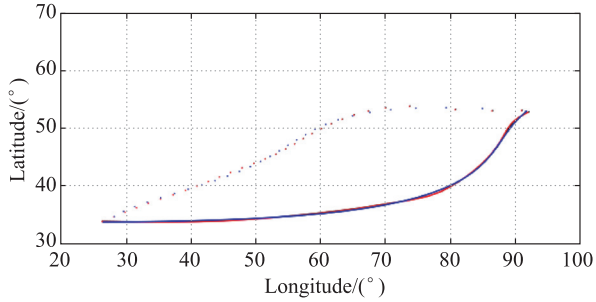
• : Boundary on the high latitude side;
— : Boundary on the low latitude side;
• : Fitting boundary curve on the high latitude side;
— : Fitting boundary curve on the low latitude side;
• : Boundary curve on the high latitude side (BP);
— : Boundary curve on the low latitude side (BP).

Fig. 16 Landing footprint of No.8 experiment



• : Boundary on the high latitude side;
— : Boundary on the low latitude side;
• : Fitting boundary curve on the high latitude side;
— : Fitting boundary curve on the low latitude side;
• : Boundary curve on the high latitude side (BP);
— : Boundary curve on the low latitude side (BP).

Fig. 17 Landing footprint of No.9 experiment



• : Boundary on the high latitude side;
— : Boundary on the low latitude side;
• : Fitting boundary curve on the high latitude side;
— : Fitting boundary curve on the low latitude side;
• : Boundary curve on the high latitude side (BP);
— : Boundary curve on the low latitude side (BP).

Fig. 18 Landing footprint of No.10 experiment

Table 2 and Table 3 give the coefficients of the fitting boundary curves obtained by using the least squares method, including mm_1^1, mm_1^2, mm_1^3, mm_1^4 and mm_1^5 (boundary curves on the high latitude side); mm_2^1, mm_2^2, mm_2^3, mm_2^4 and mm_2^5 (boundary curves on the low latitude side); nn_1^1, nn_1^2, nn_1^3, nn_1^4 and nn_1^5 (boundary curves on the high longitude side); nn_2^1, nn_2^2, nn_2^3, nn_2^4 and nn_2^5 (boundary curves on the low longitude side). Table 4 and Table 5 give the coefficients of the boundary curves obtained by the BP neural network, including m_1^1, m_1^2, m_1^3, m_1^4 and m_1^5 (boundary curves on the high latitude side); m_2^1, m_2^2, m_2^3, m_2^4 and m_2^5 (boundary curves on the low latitude side); n_1^1, n_1^2, n_1^3, n_1^4 and n_1^5 (boundary curves on the high longitude side); n_2^1, n_2^2, n_2^3, n_2^4 and n_2^5 (boundary curves on the low longitude side). From Table 2 to Table 5, we can see that the errors between coefficients of the fitting boundary curves obtained by the least squares method and those obtained by the BP neural network are smaller than the training error of 0.001.

Table 2 Coefficients of fitting curves (high latitude or longitude side)

Number	$m_1^1(n_1^1)$	$m_1^2(n_1^2)$	$m_1^3(n_1^3)$	$m_1^4(n_1^4)$	$m_1^5(n_1^5)$
1	–3.194 444 e–6	1.233 779 e–3	–1.919 443 e–1	1.514 211 e+1	–3.883 817 e+2
2	–8.256 123 e–5	2.024 390 e–2	–1.619 387	5.305 329 e+1	–6.151 001 e+2
3	–4.012 587 e–6	1.495 236 e–3	–2.784 367 e–1	2.249 619 e+1	–6.293 869 e+2
4	–5.772 980 e–5	1.255 224 e–2	–1.005 445	2.842 024 e+1	–1.540 246 e+2
5	–1.519 489 e–4	2.663 004 e–2	1.478 789	4.130 832 e+1	–3.109 007 e+2
6	–4.037 333 e–5	8.784 321 e–3	–7.144 400 e–1	2.549 818 e+1	–2.075 674 e+2
7	–1.397 543 e–4	3.408 221 e–2	–2.787 432	9.969 876 e+1	–1.309 384 e+3
8	–4.214 520 e–6	1.114 532 e–3	–1.340 298 e–1	8.091 355	–1.224 080 e+2
9	–2.331 988 e–4	3.725 921 e–2	–3.201 335	1.244 620 e+2	–1.789 788 e+3
10	–3.090 332 e–4	3.844 785 e–2	3.145 540 9	1.084 894 e+2	–1.384 916 e+3

Table 3 Coefficients of fitting curves (low latitude or longitude side)

Number	$m_2^1(n_2^1)$	$m_2^2(n_2^2)$	$m_2^3(n_2^3)$	$m_2^4(n_2^4)$	$m_2^5(n_2^5)$
1	6.441 445 e–7	–2.563 355 e–4	2.536 902 e–2	–1.817 822 e–1	–2.005 818 e+1
2	3.573 245 e–4	–3.856 786 e–2	3.328 224	–1.309 137 e+2	1.941 689 e+e3
3	8.439 678 e–7	–3.980 074 e–4	3.934 290 e–2	–8.826 228 e–1	–1.826 913 e+1
4	2.699 356 e–5	6.645 901 e–3	5.467 708 e–1	2.391 602 e+1	5.170 087 e+2
5	–3.102 356 e–5	4.248 810 e–3	2.511 124 e–1	6.101 356	5.977 024 e+1
6	–4.884 679 e–6	1.253 366 e–3	–9.820 098 e–2	3.268 925	8.756 833 e+1
7	–7.493 214 e–6	5.948 623 e–4	1.142 389 e–1	–1.444 100 e+1	3.714 782 e+2
8	2.332 990 e–7	–3.002 315 e–4	4.742 596 e–3	9.270 998 e–1	–1.701 552 e+1
9	–5.454 633 e–5	7.603 279 e–3	–6.165 638 e–1	1.584 644 e+1	–5.327 344 e+1
10	–3.295 180 e–5	1.211 239 e–2	–8.845 530 e–1	2.358 660 e+1	–1.759 743 e+2

Table 4 Results (high latitude or longitude side) obtained by the BP neural networks

Number	$m_1^1(n_1^1)$	$m_1^2(n_1^2)$	$m_1^3(n_1^3)$	$m_1^4(n_1^4)$	$m_1^5(n_1^5)$
1	–2.364 762 e–6	1.093 990 e–3	–1.919 619 e–1	1.514 202 e+1	–3.883 818 e+2
2	–9.386 719 e–5	2.053 956 e–2	–1.619 389	5.305 330 e+1	–6.151 000 e+2
3	–3.243 843 e–6	1.545 236 e–3	–2.784 475 e–1	2.249 620 e+1	–6.293 869 e+2
4	–7.052 381 e–5	1.415 389 e–2	–1.005 241	2.842 022 e+1	–1.540 246 e+2
5	–1.409 326 e–4	2.343 824 e–2	1.478 954	4.130 829 e+1	–3.109 008 e+2
6	–4.277 951 e–5	8.904 313 e–3	–7.144 316 e–1	2.549 826 e+1	–2.075 673 e+2
7	–1.492 619 e–4	3.358 251 e–2	–2.787 314	9.969 881 e+1	–1.309 385 e+3
8	–2.874 275 e–6	1.004 532 e–3	–1.340 367 e–1	8.091 336	–1.224 080 e+2
9	–1.491 477 e–4	3.585 920 e–2	–3.201 586	1.244 616 e+2	–1.789 787 e+3
10	–1.840 043 e–4	3.954 786 e–2	3.145 550 4	1.084 895 e+2	–1.384 915 e+3

Table 5 Results (low latitude or longitude side) obtained by the BP neural networks

Number	$m_2^1(n_2^1)$	$m_2^2(n_2^2)$	$m_2^3(n_2^3)$	$m_2^4(n_2^4)$	$m_2^5(n_2^5)$
1	7.261 776 e–7	–2.546 767 e–4	2.542 910 e–2	–1.817 815 e–1	–2.005 819 e+1
2	1.683 400 e–4	–3.843 956 e–2	3.328 331	–1.309 139 e+2	1.941 689 e+e3
3	9.789 098 e–7	–3.551 292 e–4	3.932 091 e–2	–8.826 921 e–1	–1.826 912 e+1
4	3.519 243 e–5	6.694 692 e–3	5.467 192 e–1	2.391 697 e+1	5.170 088 e+2
5	–2.993 518 e–5	4.384 012 e–3	2.510 960 e–1	6.101 730	5.977 056 e+1
6	–5.956 112 e–6	1.193 456 e–3	–9.824 865 e–2	3.268 925	8.756 860 e+1
7	–8.504 437 e–6	6.485 160 e–4	1.147 097 e–1	–1.444 156 e+1	3.714 789 e+2
8	7.390 693 e–7	–1.630 918 e–4	4.747 751 e–3	9.270 948 e–1	–1.701 510 e+1
9	–3.333 035 e–5	7.864 112 e–3	–6.164 476 e–1	1.584 621 e+1	–5.327 374 e+1
10	–5.666 975 e–5	1.208 691 e–2	–8.844 547 e–1	2.358 612 e+1	–1.759 747 e+2

Table 6 shows that the relative errors ΔS are smaller than 2%, which meet the usage requirements.

Table 7 is a comparison of the different computational times. The variables t_a, t_b and t_c are the computational time required for computing all of the landing points of each set of experiments, the average computational time for one landing point and the computational time for the landing footprint based on the BP neural network, respectively. As can be seen from Table 7, t_c is less than $0.1‰$ of t_a, and t_c is less than $0.2‰$ of t_b.

Table 6 Relative errors in ten experiments

Number	$S/(^\circ{}^2)$	K	$\Delta S/(^\circ{}^2)$	$\delta/\%$
1	3 895.231	2	61.365	1.575
2	405.332	4	6.225	1.536
3	3 698.298	3	33.946	0.918
4	824.369	4	5.364	0.651
5	305.251	3	3.001	0.983
6	442.264	5	2.003	0.453
7	399.257	4	5.221	1.308
8	3 591.945	3	10.399	0.290
9	610.239	5	3.987	0.653
10	701.885	4	1.986	0.283

Table 7 Comparison of the computational time

Number	t_a/s	t_b/s	t_c/s	$\frac{t_c}{t_a}/(0/_{000})$	$\frac{t_c}{t_b}/(0/_{000})$
1	1 020.442	501.781	0.009	0.088	0.179
2	1 062.002	522.388	0.009	0.085	0.172
3	1 040.661	495.365	0.009	0.086	0.182
4	998.730	489.668	0.009	0.090	0.184
5	1 082.793	526.398	0.009	0.083	0.171
6	1 081.976	504.412	0.009	0.083	0.178
7	1 040.666	511.162	0.009	0.086	0.176
8	998.471	480.114	0.009	0.090	0.187
9	1 078.563	529.652	0.009	0.083	0.170
10	1 000.211	478.200	0.009	0.089	0.188

Table 8 shows the relative errors of the proposed method, the method in [14] and the method in [15]. ΔS_1 and ΔS_2 are the relative errors obtained by the methods proposed in [14] and in [15], respectively. We can see from Table 8 that the relative errors in each experiment obtained by methods proposed in [14] and in [15] are bigger than the proposed method in this paper.

Table 8 Relative errors of three methods

Number	$\Delta S/(^\circ{}^2)$	$\Delta S_1/(^\circ{}^2)$	$\Delta S_2/(^\circ{}^2)$
1	61.365	90.220	80.365
2	6.225	41.613	25.935
3	33.946	52.902	43.023
4	5.364	60.003	42.594
5	3.001	25.623	28.330
6	2.003	39.565	31.020
7	5.221	28.001	18.674
8	10.399	22.000	24.336
9	3.987	36.297	15.236
10	1.986	15.336	19.902

Table 9 shows the computational time of the proposed method, the method in [14] and the method in [15]. The parameters t_d and t_e are the computational time for the landing footprints based on methods proposed in [14] and [15], respectively. As can be seen from Table 9, the computational time in each experiment obtained by methods proposed in [14] and [15] are longer than the proposed method.

Table 9 Computational time of three methods

Number	t_c/s	t_d/s	t_e/s
1	0.009	11.781	92.365
2	0.009	10.229	98.334
3	0.009	11.005	129.005
4	0.009	11.209	102.650
5	0.009	10.556	109.728
6	0.009	12.001	97.443
7	0.009	11.225	96.005
8	0.009	11.300	103.820
9	0.009	11.023	108.000
10	0.009	11.058	119.326

Above all, the simulation results demonstrate that the proposed method can efficiently determine the landing footprints of space-to-ground vehicles with arbitrary orbital parameters and arbitrary manoeuver moment under the premise of ensuring accuracy.

5. Conclusions and future research

A BP neural network for determining the coefficients of the fitting boundary curves, which realizes fast computation of the landing footprint of a space-to-ground vehicle is proposed. The simulation results demonstrate that the proposed method can determine landing footprints in 0.01 s while ensuring a relative error within 2%. The proposed method lays a foundation for the deployment of parking orbits and for decision makers to develop real-time programs of transfer trajectories, which enriches the relevant theories of space engineering.

There are still some shortcomings in this paper. For example, it is important to try to decrease the number of hardware devices used to compute the training data. Additionally, finding a way to solve the transfer trajectories under finite thrust should be further studied. Moreover, the polynomial equations used to describe the boundaries of landing footprints by BP neural networks can be further generalized by using alternatives such as piecewise polynomial functions (splines) as fitting curves.

References

[1] NAZARENKO A I, USOVIK I V. The effect of parameters of the initial data updating algorithm on the accuracy of spacecraft reentry time prediction. Journal of Space Safety Engineering, 2019, 6(1): 24 – 29.

[2] DESIKAN S L N, SRINIVASAN K. Longitudinal aerodynamics of a winged reentry vehicle at supersonic speed. Journal of Spacecraft and Rockets, 2018, 55(5): 1144 – 1153.

[3] LOBBIA M A. Multidisciplinary design optimization of waverider-derived crew reentry vehicles. Journal of Spacecraft and Rockets, 2017, 54(1): 233 – 245.

[4] SGUBINI S, PALMERINI G B. Evaluation of main parameters in re-entry trajectories. Aerotecnica Missili & Spazio, 2016, 95(1): 42 – 49.

[5] GANGIREDDY S, ASHOK J. Re-entry trajectory optimiza-

tion using pigeon inspired optimization based control profiles. Advances in Space Research, 2018, 62(11): 3170 – 3186.
[6] JIANG X Q, LI S. Mars atmospheric entry trajectory optimization via particle swarm optimization and Gauss pseudospectral method. Proc. of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 2016: 2320 – 2329.
[7] WANG Z B, GRANT M J. Constrained trajectory optimization for planetary entry via sequential convex programming. Journal of Guidance, Control, and Dynamics, 2017, 40(10): 1 – 13.
[8] GUO C M, ZHANG J, LUO Y Z, et al. Phase-matching homotopic method for indirect optimization of long-duration low-thrust trajectories. Advances in Space Research, 2018, 62(3): 568 – 579.
[9] SHEN H X. No-guess indirect optimization of asteroid mission using electric propulsion. Optimal Control Applications and Methods, 2018, 39(2): 1061 – 1070.
[10] RASOTTO M, ARMELLIN R, DI L P. Multi-step optimization strategy for fuel-optimal orbital transfer of low-thrust spacecraft. Engineering Optimization, 2016, 48(3): 519 – 542.
[11] GAO Y, WANG J, WU W, et al. A hybrid method for mobile agent moving trajectory scheduling using ACO and PSO in WSNs. Sensors, 2019, 19(3): 3 – 15.
[12] LEOBARDO C M, DAVID G G, RODRIGO A L, et al. A hybrid method for online trajectory planning of mobile robots in cluttered environments. IEEE Robotics and Automation Letters, 2017, 2(2): 935 – 942.
[13] SARAF A, LEAVITT J A, MEASE K D. Landing footprint computation for entry vehicles. Proc. of the AIAA Guidance, Navigation, and Control Conference and Exhibit, 2004: 1 – 14.
[14] LI D W, JIANG R K, LIU B. Reentry landing boundary prediction. Applied Mechanics and Materials, 2013, 433 – 435: 1123 – 1126.
[15] HU H L, NAN Y, WEN X. Research on dynamic reentry footprint for the spacecraft based on the genetic algorithm. Chinese Space Science and Technology, 2014, 34(1): 35 – 41. (in Chinese)
[16] NGUYEN B H, GERMAN R, TROVAO J, et al. Real-time energy management of battery/supercapacitor electric vehicles based on an adaptation of Pontryagin's minimum principle. IEEE Trans. on Vehicular Technology, 2019, 68(1): 203 – 212.
[17] SARI C, SUBCHAN S. Application of Pontryagin's minimum principle in optimum time of missile manoeuvring. Cauchy, 2016, 4(3): 107 – 111.
[18] NADIR O, LOUNIS A, RUSTEM A. From offline to adaptive online energy management strategy of hybrid vehicle using Pontryagin's minimum principle. International Journal of Automotive Technology, 2018, 19(3): 571 – 584.
[19] GWAK H S, BEA S H, LEE D E. Resource leveling using genetic algorithm. Journal of the Architectural Institute of Korea Structure & Construction, 2018, 34(2): 67 – 74.
[20] KITA H. Genetic algorithms for noisy fitness functions — applications, requirements and algorithms. Proc. of the ISCIE International Symposium on Stochastic Systems Theory and Applications, 2001: 137 – 142.
[21] SHOJAEDINI E, MAJD M, SAFABAKHSH R. Novel adaptive genetic algorithm sample consensus. Applied Soft Computing, 2019, 77: 635 – 642.
[22] NAEENI A A, ROSHANIAN J. Developing a hybrid algorithm to design the optimal trajectory of reentry vehicles. Modares Mechanical Engineering, 2015, 14(13): 143 – 149.
[23] ROBERTO P, GIANFRANCO M, MARCO C. Reentry trajectory optimization for mission analysis. Journal of Spacecraft and Rockets, 2017, 54(1): 331 – 336.
[24] EDWARDS M R. Deep mantle plumes and an increasing Earth radius. Geodesy and Geodynamics, 2019, 10(3): 173 – 178.
[25] KOPECZ S, MEISTER A. On order conditions for modified Patankar-Runge-Kutta schemes. Applied Numerical Mathematics, 2018, 123: 159 – 179.
[26] RUBIO A. Propagators for the time-dependent Kohn-Sham equations: multistep, Runge-Kutta, exponential Runge-Kutta, and commutator free Magnus methods. Journal of Chemical Theory and Computation, 2018, 14(6): 3040 – 3052.
[27] AMINE E, HICHAM A, NOURA A. An intelligent model for enterprise resource planning selection based on BP neural network. Proc. of the 2nd Mediterranean Symposium on Smart City Applications, 2017: 212 – 222.
[28] MA L, LIN X, JIANG L H. Differential-weighted global optimum of BP neural network on image classification. Proc. of the International Conference on Information Science and Applications, 2017: 544 – 552.
[29] CHANG X L, LI W J. Evaluation of creative talents in cultural industry based on BP neural network. International Journal of Performability Engineering, 2018, 14(11): 8 – 20.
[30] CSATO L. A characterization of the logarithmic least squares method. European Journal of Operational Research, 2019, 276(1): 212 – 216.
[31] MACIEJ K, MARCIN P. The least squares method for option pricing revisited. Applicationes Mathematicae, 2018, 45(1): 5 – 29.
[32] KHAN T, YADAV J S. Adaptive learning based improved performance of activation functions in hidden layer using artificial neural network. International Journal of Engineering and Technology, 2018, 7(4): 3223 – 3227.

Biographies

LIU Qingguo was born in 1991. He received his B.S. degree in aeronautical and astronautical science and technology from Xi'an High-tech Institute in 2015. He is pursuing his Ph.D. degree in Xi'an High-tech Institute. His research interests are flight mechanics, structural analysis of space vehicles and decision optimization.
E-mail: teamalpha@163.com

LIU Xinxue was born in 1964. He received his Ph.D. degree in aeronautical and astronautical science and technology from Northwestern Polytechnical University in 2003. Now he is a professor in Xi'an High-tech Institute. His research interests are flight mechanics, structural analysis of space vehicles and operations research.
E-mail: ccadd_sp@163.com

WU Jian was born in 1985. He received his Ph.D. degree in aeronautical and astronautical science and technology from Xi'an High-tech Institute in 2013. His research interests are flight mechanics, structural analysis of space vehicles and decision optimization.
E-mail: wujian6029@163.com

LI Yaxiong was born in 1979. He received his Ph.D. degree in operations research from Xi'an High-tech Institute in 2013. Now he is an associate professor in Xi'an High-tech Institute. His research interest is operations research.
E-mail: 13571996716@139.com