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
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, as 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

\[
\begin{aligned}
\frac{dV}{dt} &= \frac{T \cos \alpha \cos \beta - X}{m} - \frac{\gamma r \sin \gamma - \gamma \Phi \cos \gamma \sin \psi + \psi \gamma \tau \cos (\sin \gamma \cos \phi - \cos \gamma \sin \phi \cos \psi)}{V} \\
\frac{d\gamma}{dt} &= \frac{T \sin \alpha + V \cos \gamma}{mV} + \frac{\gamma \tau \cos \gamma}{r} - \frac{\gamma \Phi \sin \gamma \sin \psi}{V} + \frac{\psi \gamma \tau \cos \phi}{V} + \frac{2\omega_e \cos \phi \sin \psi}{V} \\
\frac{d\psi}{dt} &= \frac{T \cos \alpha \cos \beta - Z}{mV \cos \gamma} - \frac{\psi \gamma \tau \cos \phi \sin \psi}{V} - \frac{\gamma \Phi \cos \psi}{V \cos \gamma} - \frac{2\omega_e \cos \phi \sin \psi}{V} \\
\frac{d\phi}{dt} &= \frac{\psi \gamma \tau \cos \phi}{V \cos \gamma} \\
\frac{d\Theta}{dt} &= \frac{V \cos \gamma \cos \psi}{\tau \cos \phi} \\
\frac{d\Phi}{dt} &= \frac{V \cos \gamma \sin \psi}{\tau} \\
\frac{d \mathbf{m}}{dt} &= -\frac{T}{V_e}
\end{aligned}
\]

where \( V \) is the dimensionless velocity, \( \gamma \) is the velocity inclination angle, \( \psi \) is the course angle (the angle between the projection of the velocity vector on the local horizontal plane and the latitude tangent), \( \tau \) is the dimensionless geocentric distance, \( \Theta \) is the longitude, \( \Phi \) is the latitude, \( T \) is the engine thrust, \( V_e \) is the gas jet velocity, \( \mathbf{m} \) 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 gravitational force, \( \alpha \) is the angle of attack, \( \beta \) is the sideslip angle, \( \omega_e \) is the dimensionless earth rotation angular rate, \( \overline{g}_e \) and \( \overline{g}_r \) 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:

\[
\begin{align*}
\overline{V} &= \frac{V}{V_{ref}} \\
\overline{r} &= \frac{r}{r_{ref}} \\
\overline{T} &= \frac{T}{t_{ref}} \\
\overline{m} &= \frac{m}{m_{ref}} \\
\overline{t} &= \frac{t}{t_{ref}} \\
\overline{X} &= \frac{X}{m_{ref}g_{ref}} \\
\overline{Y} &= \frac{Y}{m_{ref}g_{ref}} \\
\overline{Z} &= \frac{Z}{m_{ref}g_{ref}} \\
\overline{\omega}_e &= \omega_{e,ref} \\
\overline{g}_r &= \frac{1}{2\pi^2}(1 - \frac{3J_2}{2}\sin^2 \phi - 1) \\
\overline{g}_\phi &= \frac{3J_2 \sin 2\phi}{2\pi^2} \\
\end{align*}
\]

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. \( r_{ref}, m_{ref}, V_{ref}, t_{ref} \) and \( g_{ref} \) are given by the following equations:

\[
\begin{align*}
\left\{ \begin{array}{l}
V_{ref} = \sqrt{\frac{\mu}{r_{ref}}} \\
\mu = \frac{m_0}{V_{ref}} \\
t_{ref} = \frac{r_{ref}}{V_{ref}} \\
g_{ref} = \frac{\mu}{r_{ref}^2} \\
\end{array} \right. \\
\end{align*}
\]  

where \( \mu \) is the gravitational constant and the value of \( \mu \) is 3.986 005e+14 m³/s². \( m_0 \) is the initial mass of the vehicle and \( R_E \) is the radius of the earth [24]:

\[
0.000 003 549 \cos(4\Phi) + 0.000 000 008 \cos(6\Phi) \].
\]

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

\[
\begin{align*}
V_I &= \sqrt{\frac{\mu(1 + 2e\cos f + e^2)}{a(1 - e^2)}} \\
\gamma_I &= \arctan \frac{e \sin f}{1 + e \cos f} \\
\Psi_I &= \arctan[\tan i \cos(\omega + f)] \\
\Theta_I &= \Omega + \arctan[\tan(\omega + f) \sin i] \\
\phi_I &= \arcsin[\sin(\omega + f) \sin i] \\
r_I &= \frac{a(1 - e^2)}{1 + e \cos f} \\
f &= \arcsin \frac{\sin \phi_I - \omega}{\sin i} \\
\end{align*}
\]

where \( V_I, \gamma_I, \Psi_I, r_I, \Theta_I \) and \( \phi_I \) 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

\[
\begin{align*}
V &= \sqrt{V_I - 2\omega_e r_I \cos \gamma_I \cos \psi_I \cos \phi_I + (\omega_e r_I \cos \phi_I)^2} \\
\tan \gamma &= \frac{V_I \sin \gamma_I}{(V_I \cos \gamma_I)^2 - 2\omega_e r_I V_I \cos \gamma_I \cos \psi_I \cos \phi_I + (\omega_e r_I \cos \phi_I)^2} \\
\tan \psi &= \frac{V_I \cos \gamma_I \sin \psi_I}{V_I \cos \gamma_I \cos \psi_I - \omega_e r_I \cos \phi_I} \\
r &= r_I \\
\Theta &= \Theta_I - \Gamma - \omega_e t \\
\phi &= \phi_I \\
\end{align*}
\]
2.2 Computational steps of the landing footprints

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

\[
\min J_1 = \begin{cases} 
|\Theta_b - \Theta_m|, & 0^\circ \leq i \leq 45^\circ \text{ or } 135^\circ \leq i \leq 180^\circ \\
|\Phi_b - \Phi_m|, & 45^\circ < i < 135^\circ 
\end{cases} 
\quad (7)
\]

\[
\min J_2 = \begin{cases} 
-|\Theta_b - \Theta_m|, & 0^\circ \leq i \leq 45^\circ \text{ or } 135^\circ \leq i \leq 180^\circ \\
-|\Phi_b - \Phi_m|, & 45^\circ < i < 135^\circ 
\end{cases} 
\quad (8)
\]

\[
\min J_3 = \begin{cases} 
\Phi_b, & 0^\circ \leq i \leq 45^\circ \text{ or } 135^\circ \leq i \leq 180^\circ \\
-\Phi_b, & 0^\circ \leq i \leq 45^\circ \text{ or } 135^\circ \leq i \leq 180^\circ \\
-\Theta_b, & 45^\circ < i < 135^\circ \\
\Theta_b, & 45^\circ < i < 135^\circ 
\end{cases} 
\quad (9)
\]

where \(J_1, J_2\) and \(J_3\) are optimization indexes corresponding to research ideas (i), (ii) and (iii), respectively. \(\Theta_m\) and \(\Phi_m\) are the longitude and the latitude of the manoeuvre point. The variables \(\Theta_b\) and \(\Phi_b\) are the longitude and the latitude of the landing point.

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

\[
H = \lambda_r \overline{V} + \lambda_\gamma \dot{\gamma} + \lambda_\psi \dot{\psi} + \lambda_\Theta \dot{\Theta} + \lambda_\Phi \dot{\Phi} + \lambda_m \overline{m}.
\]

\[
(10)
\]

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.

\[
\begin{align*}
\frac{d\lambda_r}{dt} &= \frac{\partial H}{\partial \overline{V}} \\
\frac{d\lambda_\gamma}{dt} &= \frac{\partial H}{\partial \gamma} \\
\frac{d\lambda_\psi}{dt} &= \frac{\partial H}{\partial \psi} \\
\frac{d\lambda_\Theta}{dt} &= \frac{\partial H}{\partial \Theta} \\
\frac{d\lambda_\Phi}{dt} &= \frac{\partial H}{\partial \Phi} \\
\frac{d\lambda_m}{dt} &= \frac{\partial H}{\partial \overline{m}}
\end{align*}
\]

\[
(11)
\]

where the expressions of \(\overline{V}, \overline{V}_\gamma, \overline{V}_\psi\) and \(\overline{V}_\theta\) are given by

\[
\begin{align*}
\overline{V}_r &= -\frac{2r_j}{\overline{V}} + \frac{3J_2}{2}\frac{\sin \Phi}{\overline{V}} \\
\overline{V}_\gamma &= -\frac{6J_2}{\overline{V}} \sin \Phi \\
\overline{V}_\psi &= -\frac{9J_2}{\overline{V}} \sin \Phi \\
\overline{V}_\Theta &= \frac{3J_2}{\overline{V}} \cos \Phi \\
\overline{V}_\Phi &= \frac{3J_2}{\overline{V}} \cos \Phi
\end{align*}
\]

\[
(12)
\]

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

\[
\begin{align*}
\frac{\partial H}{\partial \alpha} &= 0 = -\lambda_r \overline{V} \sin \alpha \cos \beta + \lambda_\gamma \overline{V} \sin \alpha \sin \beta \\
&+ \frac{\lambda_\psi \overline{V} \cos \alpha \sin \beta}{m} \\
&+ \frac{\lambda_\Theta \overline{V} \cos \alpha \cos \beta}{m} \\
&+ \frac{\lambda_\Phi \overline{V} \sin \gamma}{m} \\
&+ \frac{\lambda_m \overline{V} \cos \gamma}{m},
\end{align*}
\]

\[
(13)
\]

and

\[
\begin{align*}
\alpha &= \arctan \frac{\text{sgn}(\overline{\lambda}_\gamma) \cdot \lambda_\gamma \cos \gamma}{\sqrt{(\lambda_\psi \overline{V} \cos \gamma)^2 + \lambda_\psi^2}} \\
\beta &= \arctan \frac{\lambda_\psi \overline{V} \cos \gamma}{\lambda_\psi}
\end{align*}
\]

\[
(14)
\]

where \(\text{sgn}(\cdot)\) is the sign function.

The transition trajectory satisfies the initial boundary conditions:

\[
\begin{align*}
\overline{V}(\overline{t}_0) - \overline{V}_0 &= 0 \\
\gamma(\overline{t}_0) - \gamma_0 &= 0 \\
\overline{r}(\overline{t}_0) - \overline{r}_0 &= 0 \\
\psi(\overline{t}_0) - \psi_0 &= 0 \\
\Theta(\overline{t}_0) - \Theta_0 &= 0 \\
\Phi(\overline{t}_0) - \Phi_0 &= 0 \\
\overline{m}(\overline{t}_0) - \overline{m}_0 &= 0
\end{align*}
\]

\[
(15)
\]

where \(\overline{V}_0, \gamma_0, \psi_0, \overline{r}_0, \Theta_0, \Phi_0\) 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:

\[
\begin{align*}
\overline{V}(\overline{t}_a) - \overline{V}_a &= 0 \\
\gamma(\overline{t}_a) - \gamma_a &= 0 \\
\overline{r}(\overline{t}_a) - \overline{r}_a &= 0
\end{align*}
\]

\[
(16)
\]

where \(\overline{V}_a, \gamma_a\) and \(\overline{r}_a\) are the dimensionless speed, the velocity inclination angle and the geocentric distance at the
terminal of the transition trajectory respectively; \( \bar{t}_a \) is the dimensionless time at the terminal point of the transition trajectory.

Therefore, the cross-section conditions are given by
\[
\begin{align*}
\lambda_{\psi}(\bar{t}_a) &= 0 \\
\lambda_{\theta}(\bar{t}_a) &= 0 \\
\lambda_{\phi}(\bar{t}_a) &= 0 \\
\lambda_{\Omega}(\bar{t}_a) &= 0.
\end{align*}
\] (17)

When the optimization index \( J_3 \) is calculated, the re-entry trajectory satisfies the constraints:
\[
\begin{align*}
\Theta(\bar{t}_b) &= \Theta_b, \quad 0^\circ \leq i \leq 45^\circ \text{or} 135^\circ \leq i \leq 180^\circ \\
\phi(\bar{t}_b) &= \phi_b, \quad 45^\circ < i < 135^\circ
\end{align*}
\] (18)
where \( \Theta_b \) and \( \phi_b \) are the longitude and latitude of the landing point at the dimensionless time \( \bar{t}_b \), respectively.

Because the optimization indexes do not include time,
\[
H(\bar{t}_0) = H(\bar{t}) = H(\bar{t}_a) = 0.
\] (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 \( \lambda_{\Omega}(\bar{t}_0) \), \( \lambda_{\psi}(\bar{t}_0) \), \( \lambda_{\theta}(\bar{t}_0) \), \( \lambda_{\phi}(\bar{t}_0) \), \( \lambda_{\Theta}(\bar{t}_0) \) and \( \lambda_{\Phi}(\bar{t}_0) \), the values of four state variables \( \gamma_a \), \( \gamma_a \), \( \bar{t}_a \) and \( \bar{t}_a \) or the values of five state variables \( \gamma_a \), \( \gamma_a \), \( \bar{t}_a \), \( \bar{t}_a \) and \( \phi_b(\Theta_b) \). Since the initial value of an adjoint variable can be obtained from (10) and (19) and \( \lambda_{\Omega}(\bar{t}_0) \equiv 0 \), 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, \( \lambda_{\Omega}(\bar{t}_0) \), \( \lambda_{\psi}(\bar{t}_0) \), \( \lambda_{\theta}(\bar{t}_0) \), \( \lambda_{\phi}(\bar{t}_0) \), \( \lambda_{\Theta}(\bar{t}_0) \), \( \lambda_{\Phi}(\bar{t}_0) \), \( \gamma_a \), \( \gamma_a \) and \( \bar{t}_a \) are encoded. When optimization index \( J_3 \) is calculated, \( \lambda_{\Omega}(\bar{t}_0) \), \( \lambda_{\psi}(\bar{t}_0) \), \( \lambda_{\theta}(\bar{t}_0) \), \( \lambda_{\phi}(\bar{t}_0) \), \( \lambda_{\Omega}(\bar{t}_0) \), \( \lambda_{\psi}(\bar{t}_0) \), \( \lambda_{\theta}(\bar{t}_0) \), \( \lambda_{\phi}(\bar{t}_0) \), \( \lambda_{\Theta}(\bar{t}_0) \), \( \lambda_{\Phi}(\bar{t}_0) \), \( \gamma_a \), \( \gamma_a \) and \( \bar{t}_a \) 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 \( 0^\circ \leq i \leq 45^\circ \) or \( 135^\circ \leq i \leq 180^\circ \), 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 $4^\circ$ are used in the least squares fitting method.

![Fig. 1 Sketch diagram of the landing footprints when $0^\circ \leq i \leq 45^\circ$ or $135^\circ \leq i \leq 180^\circ$](image1)

![Fig. 2 Sketch diagram of landing footprints when $45^\circ < i < 135^\circ$](image2)

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

$$y = mm_1^1 x^4 + mm_2^1 x^3 + mm_3^1 x^2 + mm_4^1 x + mm_5^1 \tag{20}$$

and

$$y = mm_1^2 x^4 + mm_2^2 x^3 + mm_3^2 x^2 + mm_4^2 x + mm_5^2 \tag{21}$$

where $x$ is the longitude and $y$ is the latitude; the values $mm_1^1, mm_2^1, mm_3^1, mm_4^1$ and $mm_5^1$ are the fitting coefficients of the boundary curve on the high latitude side; and the values $mm_2^1, mm_2^2, mm_3^2, mm_4^2$ and $mm_5^2$ are the fitting coefficients of the boundary curve on the low latitude side.

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

$$x = nn_1^1 y^4 + nn_2^1 y^3 + nn_3^1 y^2 + nn_4^1 y + nn_5^1 \tag{22}$$

and

$$x = nn_1^2 y^4 + nn_2^2 y^3 + nn_3^2 y^2 + nn_4^2 y + nn_5^2 \tag{23}$$

where the values $nn_1^1, nn_2^1, nn_3^1, nn_4^1$ and $nn_5^1$ are the fitting coefficients of the boundary curve on the high longitude side; and the values $nn_2^1, nn_2^2, nn_3^2, nn_4^2$ and $nn_5^2$ 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 $0^\circ \leq i \leq 45^\circ$ or $135^\circ \leq i \leq 180^\circ$. We define $m_1^1, m_2^1, m_3^1, m_4^1$ and $m_5^1$ as the output parameters of the boundary curve on the high latitude side, and $m_1^2, m_2^2, m_3^2, m_4^2$ and $m_5^2$ 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 $0^\circ \leq i \leq 45^\circ$ or $135^\circ \leq i \leq 180^\circ$, we get

$$y = m_1^1 x^4 + m_2^1 x^3 + m_3^1 x^2 + m_4^1 x + m_5^1 \tag{24}$$

and

$$y = m_2^1 x^4 + m_2^2 x^3 + m_3^2 x^2 + m_4^2 x + m_5^2 \tag{25}$$

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

$$x = n_1^1 y^4 + n_2^1 y^3 + n_3^1 y^2 + n_4^1 y + n_5^1 \tag{26}$$

and

$$x = n_2^1 y^4 + n_2^2 y^3 + n_3^2 y^2 + n_2^2 y + n_5^2 \tag{27}$$

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

$$T = 2\pi \sqrt{\frac{a^3}{\mu}}. \tag{28}$$
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 $\Omega$, the perigee angle $\omega$, the true anomaly $f$ and the manoeuvre moment $t_m$.

The input parameters need to be normalized as follows:

$$D = \frac{D - D_{\min}}{D_{\max} - D_{\min}}$$ (29)

where $D = [a, e, i, \Omega, \omega, f, t_m]$, $D$, $D_{\min}$ 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.

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 $\Omega$, the perigee angle $\omega$, 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 $m_1^1$, $m_2^1$, $m_3^1$, $m_4^1$ and $m_5^1$; the output parameters of the No.2 BP neural network model are $m_1^2$, $m_2^2$, $m_3^2$, $m_4^2$ and $m_5^2$; the output parameters of the No.3 BP neural network model are $n_1^1$, $n_1^1$, $n_1^1$, $n_1^1$ and $n_1^1$; the output parameters of the No.4 BP neural network model are $n_1^2$, $n_2^2$, $n_3^2$, $n_4^2$ and $n_5^2$. The number of hidden layer nodes of the four neural network models is

$$l = \sqrt{p + q + s}$$ (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

$$net_i = \sum_{j=1}^{P} w_{ij}x_j + \theta_i$$  \hspace{1cm} (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 $\theta_i$ 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

$$o_i = \phi(net_i) = \phi \left( \sum_{j=1}^{P} w_{ij}x_j + \theta_i \right)$$  \hspace{1cm} (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

$$net_k = \sum_{i=1}^{l} t_{ki}o_i + a_k = \sum_{i=1}^{l} \left[ t_{ki} \phi \left( \sum_{j=1}^{P} w_{ij}x_j + \theta_i \right) \right] + u_k$$  \hspace{1cm} (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

$$o_k = \psi(net_k) = \psi \left( \sum_{i=1}^{l} \left[ t_{ki} \phi \left( \sum_{j=1}^{P} w_{ij}x_j + \theta_i \right) \right] + u_k \right)$$  \hspace{1cm} (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 $\theta_i$ 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 $E_P$ of the BP neural network is

$$E_P = \frac{1}{2} \sum_{p=1}^{P} \sum_{k=1}^{l} \left( T_{k}^{p} - o_{k}^{p} \right)^2$$  \hspace{1cm} (35)

where $T_{k}^{p}$ is the true value of the $p$th sample at the $k$th 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 $\Delta w_{ij}$ in the hidden layer is performed by using

$$\Delta w_{ij} = \eta \sum_{p=1}^{P} \sum_{k=1}^{l} \left( T_{k}^{p} - o_{k}^{p} \right) \cdot \psi^\prime(net_k) \cdot t_{ki} \cdot \phi^\prime(net_i) \cdot \frac{\partial net_i}{\partial w_{ij}}.$$  \hspace{1cm} (36)

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

$$\Delta \theta_i = \eta \sum_{p=1}^{P} \sum_{k=1}^{l} \left( T_{k}^{p} - o_{k}^{p} \right) \cdot \psi^\prime(net_k) \cdot t_{ki} \cdot \phi^\prime(net_i).$$  \hspace{1cm} (37)

The adjustment of the weight value $\Delta t_{ki}$ in the output layer is implemented by using

$$\Delta t_{ki} = \eta \sum_{p=1}^{P} \sum_{k=1}^{l} \left( T_{k}^{p} - o_{k}^{p} \right) \cdot \psi^\prime(net_k) \cdot \frac{\partial net_k}{\partial t_{ki}}.$$  \hspace{1cm} (38)

The adjustment of the threshold value $\Delta u_k$ in the output layer is implemented by using

$$\Delta u_k = \eta \sum_{p=1}^{P} \sum_{k=1}^{l} \left( T_{k}^{p} - o_{k}^{p} \right) \cdot \psi^\prime(net_k).$$  \hspace{1cm} (39)

In (36)–(39), the parameter $\eta$ 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.
The relative error $\Delta S$ is calculated by using the following formula:

$$\Delta S = \frac{S_1 + S_2 + S_2 + S_3}{S}. \quad (40)$$

When $0^\circ \leq i \leq 45^\circ$ or $135^\circ \leq i \leq 180^\circ$, $S$ 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, $S_1$, $S_2$, $S_3$ and $S_4$.

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$^\circ$, 180$^\circ$], the right ascension of ascending node $\Omega$ is in the range [0$^\circ$, 360$^\circ$], the argument of the perigee $\omega$ is in the range [0$^\circ$, 360$^\circ$], the true anomaly $f$ is in the range [0$^\circ$, 360$^\circ$]. Each of the six parameters $a$, $e$, $\Omega$, $\omega$, $f$ and $t_m$ takes three random values within their respective range. The parameter $i$ takes four random values, two in the range of $0^\circ \leq i \leq 45^\circ$ or $135^\circ \leq i \leq 180^\circ$ and the other two in the range of $45^\circ < i < 135^\circ$. The values (2 916 = 3$^6$ × 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.

| No. | $a$/km | $e$ | $i$/($^\circ$) | $\Omega$/($^\circ$) | $\omega$/($^\circ$) | $f$/($^\circ$) | $t_m$/s |
|-----|--------|----|-------------|----------------|----------------|-------------|-------|
| 1   | 9 406  | 0.234450 | 29.321 | 224.038 | 307.947 | 135.862 | 4 605.5 |
| 2   | 7 159  | 0.000325 | 98.084 | 185.381 | 283.332 | 4 486.3 |
| 3   | 9 337  | 0.000265 | 52.006 | 172.282 | 350.025 | 188.701 | 4 509.2 |
| 4   | 7 838  | 0.001419 | 101.618 | 293.441 | 282.681 | 4 466.1 |
| 5   | 7 717  | 0.054058 | 99.013 | 347.918 | 308.882 | 4 486.0 |
| 6   | 7 654  | 0.002862 | 90.045 | 245.644 | 85.205 | 113.968 | 4 524.4 |
| 7   | 8 328  | 0.201675 | 82.382 | 262.455 | 29.931 | 44.341 | 4 438.5 |
| 8   | 9 511  | 0.213250 | 28.126 | 34.617 | 152.421 | 325.392 | 4 459.4 |
| 9   | 9 431  | 0.016526 | 64.422 | 302.102 | 337.357 | 56.304 | 4 483.6 |
| 10  | 10 024 | 0.321619 | 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 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-
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](image)

![Fig. 10 Landing footprint of No.2 experiment](image)

![Fig. 11 Landing footprint of No.3 experiment](image)

![Fig. 12 Landing footprint of No.4 experiment](image)

![Fig. 13 Landing footprint of No.5 experiment](image)
Table 2 and Table 3 give the coefficients of the fitting boundary curves obtained by using the least squares method, including $mm_1^1$, $mm_2^1$, $mm_3^1$, $mm_4^1$ and $mm_5^1$ (boundary curves on the high latitude side); $mm_1^2$, $mm_2^2$, $mm_3^2$, $mm_4^2$ and $mm_5^2$ (boundary curves on the low latitude side); $nn_1^1$, $nn_2^1$, $nn_3^1$, $nn_4^1$ and $nn_5^1$ (boundary curves on the high longitude side); $nn_1^2$, $nn_2^2$, $nn_3^2$, $nn_4^2$ and $nn_5^2$ (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.
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 shows that the relative errors \( \Delta S \) 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 the variables \( m_1 \), \( m_2 \) and \( m_3 \) are the computational coefficients of the landing points of the high and low latitude sides.

Table 2 shows the coefficients of fitting curves (high latitude or longitude side).

Table 3 shows the coefficients of fitting curves (low latitude or longitude side).

Table 4 shows the results (high latitude or longitude side) obtained by the BP neural networks.

Table 5 shows the results (low latitude or longitude side) obtained by the BP neural networks.
The method, the method in [14] and the method in [15]. The computational time in each experiment obtained by methods proposed in [14] and in [15], respectively. As can be seen from Table 9, the computational time of the proposed method in this paper.

Table 7 Comparison of the computational time

| Number | $t_d$/s | $t_e$/s | $t_c$/s | $t_d$/s/$t_e$/s/($^{\circ}/$000) | $t_c$/s/$t_e$/s/($^{\circ}/$000) |
|--------|---------|---------|---------|----------------------------------|----------------------------------|
| 1      | 1 020.442 | 501.781 | 0.009   | 0.008                            | 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                            |

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 maneuver 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, ARMELLINI 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 (Architectural Science & Construction), 2018, 77: 635 – 642.

[20] EDWARDS M R. Deep mantle plumes and an increasing Earth radius. Geodesy and Geodynamics, 2019, 10(3): 173 – 178.

[21] KOPECZ S, MEISTER A. On order conditions for modified Patankar-Runge-Kutta schemes. Applied Numerical Mathematics, 2018, 123: 159 – 179.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] CSATO L. A characterization of the logarithmic least squares method. European Journal of Operational Research, 2019, 270(1): 212 – 216.

[27] MACIEJ K, MARCIN P. The least squares method for option pricing revisited. Applicationes Mathematicae, 2018, 45(1): 5 – 29.

[28] 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: ccaddsp@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