




RESEARCH ARTICLE

# Rapid generation of time-optimal rendezvous trajectory based on convex optimisation and DNN

R.D. Zhang , W.W. Cai  and L.P. Yang 

College of Aerospace Science and Engineering, National University of Defense Technology, Changsha, China

Corresponding author: W.W. Cai; Email: [tsaiweiwei@hotmail.com](mailto:tsaiweiwei@hotmail.com)

Received: 10 July 2023; Revised: 17 October 2023; Accepted: 20 October 2023

**Keywords:** time-optimal trajectory; convex optimisation; deep neural network; natural-motion circumnavigation injection; swarm reconfiguration

## Abstract

The minimum flight time of spacecraft rendezvous is one of the fundamental indexes for mission design. This paper proposes a rapid trajectory planning method based on convex optimisation and deep neural network (DNN). The time-optimal trajectory planning problem is reconstructed into a double-layer optimisation framework, with the inner being a convex optimisation problem and the outer being a root-finding problem. The thrust properties corresponding to time-optimal control are analysed theoretically. A DNN-based rapid planning method (DNN-RPM) is put forward to improve computational efficiency, in which the trained DNN provides a high-quality initial guess for Newton's method. The DNN-RPM is extended to search for the optimal entering angle of natural-motion circumnavigation orbit injection problem and the minimum reconfiguration time of spacecraft swarm. Numerical simulations show that the proposed method can improve the computational efficiency while ensuring the calculation accuracy.

## Nomenclature

DNN	deep neural network
DNN-RPM	DNN-based rapid planning method
ECI	Earth-centred inertial frame
LVLH	local vertical/local horizontal
MFT	minimum flight time
MSE	mean square error
NMC	natural-motion circumnavigation
SCP	sequential convex programming
SDP	semidefinite programmes
SQP	sequential quadratic programming
$b$	semi-minor axis of in-plane ellipse of NMC orbit (m)
$c$	maximum displacement of NMC orbit in $z$ -axis (m)
$g_0$	gravitational acceleration at sea level ( $\text{m/s}^2$ )
$I_{sp}$	vacuum specific impulse of thruster (s)
$J_1$	performance metric of Problem 1
$J_2$	performance metric of Problem 2
$J_3$	performance metric of root-finding problem
$\bar{J}_3$	preset boundary value of $J_3$
$m$	mass of deputy spacecraft (kg)
$m_0$	initial mass deputy spacecraft (kg)
$M$	maximum iteration of hybrid search method
$p, p'$	initial parameter and normalised parameter

$p_{\min}, p_{\max}$	minimum and maximum values of initial parameter
$\mathbf{r}$	relative position of deputy spacecraft (m)
$\mathbf{r}_c$	position vector of chief spacecraft (m)
$\mathbf{r}_d$	position vector of deputy spacecraft (m)
$t_f$	terminal flight time (s)
$t_f^*$	minimum flight time (s)
$t_{LB}, t_{UB}$	upper limit and lower limit of flight time (s)
$\mathbf{T}$	thrust of deputy spacecraft (N)
$\mathbf{T}^*$	optimal thrust of time-optimal control problem
$T_{\max}$	maximum thrust (N)
$u$	adjusting parameter of thrust magnitude
$\mathbf{v}$	relative velocity of deputy spacecraft (m/s)
$\mathbf{x}$	relative state of deputy spacecraft (m, m/s)
$\mathbf{x}_0, \mathbf{x}_f$	initial and terminal relative states of deputy spacecraft (m, m/s)
$y_c$	displacement of in-plane ellipse centre in y-axis (m)
$y_i, \hat{y}_i$	real value and predicted value of DNN
$\varepsilon$	tolerance value of hybrid search method
$\boldsymbol{\varepsilon}$	angular acceleration of chief spacecraft (rad/s <sup>2</sup> )
$\boldsymbol{\omega}$	angular velocity of chief spacecraft (rad/s)
$\mu$	gravitational constant of Earth (m <sup>3</sup> /s <sup>2</sup> )
$\boldsymbol{\theta}$	unit direction vector of thrust
$H$	Hamiltonian function
$\boldsymbol{\lambda}$	costate of states variables
$\boldsymbol{\lambda}_r$	costate of position variables
$\boldsymbol{\lambda}_v$	costate of velocity variables
$\lambda_m$	costate of mass
$\phi$	phase angle of the in-plane motion (rad)
$\Phi$	performance index of time-optimal control problem
$\psi$	phase angle of the out-plane motion (rad)
$\Psi$	terminal boundary conditions
$\mathbf{v}$	Lagrangian multiplier vector
$\rho$	switching function

## 1.0 Introduction

The time-optimal trajectory planning aims at finding the time histories of the spacecraft's position and velocity, while minimising the flight time and observing the various boundary and path constraints [1]. The limited onboard computing resources make trajectory autonomous planning challenging, especially in multi-satellites manoeuvre scenarios, thus it's urgent to develop rapid planning methods for time-optimal rendezvous problem.

Trajectory planning for spacecraft rendezvous is one of the enabling technologies for various space missions, such as rendezvous and docking [2, 3], on-orbit servicing [4], and swarm reconfiguration [5, 6]. Extensive work has been conducted on fuel/energy optimal trajectory planning by different strategies that can be mainly classified into two categories, the direct and indirect methods [7]. Due to the complicated nonlinear dynamics, the construction of first-order necessary conditions of optimality usually becomes costly, limiting the further application of the indirect method [8]. Direct methods parameterise control space (sometimes the state space) and reduce the optimal control problem to nonlinear programming problem [9], mainly including pseudo-spectral method [10, 11], mixed integer linear programming [12], and symplectic algorithms [13, 14]. Although effective results were achieved, most of the previous studies suffered from computational efficiency due to the increasing number of spacecraft. Recently, the convex optimisation algorithm has been found to be a promising technique for online planning since its superior advantages on computational efficiency. Researchers have applied the convex optimisation and its derived sequential convex programming (SCP) in three-body problem [15], hypersonic

vehicle reentry [16] and interplanetary transfer [17], Han's research shows that the SCP has higher computational efficiency in comparison with the pseudo-spectral method [18]. In addition, the convex algorithm can be written using available CVX toolbox, and the resulting semidefinite programmes (SDP) would then be solved by mature solvers, such as SDPT3 and SeDuMi [19]. Unfortunately, the objective function of time-optimal control problem is non-convex, which is intractable for convex optimisation algorithm. Yang innovatively used the minimum landing error problem as a gateway to solving the time-optimal asteroid landing problem, and searched for the minimum flight time (MFT) through a combination of extrapolation and bisection methods [20]. However, the performance of interpolation depends on the quality of initial guess, and iteration steps will increase with poor start points. In addition, the calculation time increases dramatically with the swarm size when it comes to satellite swarm.

With the rapid development of artificial intelligence technology, more and more scholars have combined the emerging neural network technologies with the traditional numerical optimisation methods to solve the trajectory planning problems [21]. Viavattene [22] applied artificial neural networks to quickly predict the time and fuel consumption of interplanetary low-thrust transfers, and the predicted result was used to search the feasible asteroid rendezvous sequence. Considering the reentry trajectory planning of hypersonic vehicles, Wang [23] utilised the Chebyshev pseudo-spectral method to generate optimal trajectory dataset and established a neural network to predict the control variables. In Ref. [24], the sequential second-order cone programming method was applied to generate optimal entry trajectories offline, and a neural network was trained to output the predicted control command online. However, there are inevitable errors between predicted results of neural network and optimal control commands. When it comes to the rendezvous problem, the control errors would invalidate terminal constraints, which may lead to collisions between spacecraft.

Concentrating on the time-optimal rendezvous problem, a rapid trajectory planning method based on convex optimisation and deep neural network (DNN) is proposed, which can not only improve the computational efficiency of time-optimal trajectory, but also perform well in natural-motion circumnavigation (NMC) orbit injection and swarm reconfiguration. The first contribution of this paper lies in transforming the time-optimal rendezvous problem into a double-layer optimisation framework, where the inner layer is a convex optimisation problem, and the outer layer is a root-finding problem. Inspired by the concept of minimum landing error problem in Mars and asteroid landing [20, 25], the inner layer constructs a minimum terminal error problem that takes the terminal error between the actual and desired state as optimisation index. The thrust properties corresponding to time-optimal control are analysed theoretically. The objective function of the outer root-finding problem is constructed based on the thrust magnitude and the terminal error, so that the corresponding control law is time-optimal when the objective equals zero.

The second contribution lies in proposing a hybrid search algorithm (HSM) and a deep neural network based rapid planning method (DNN-RPM) to search for MFT. For the outer layer root-finding problem, the bisection and Newton's methods are two typical and effective iterative-based algorithms. The bisection method is robust but computationally inefficiently. The convergence rate for classical Newton's method is at least quadratic, but it fails to converge when the Hessian matrix is not positive definite [26]. In order to balance computational efficiency and robustness, the HSM that combines the advantages of above two methods is proposed to generate dataset for DNN training. And then, the DNN-RPM is put forward to further improve computational efficiency, in which the trained DNN provides a high-quality initial guess for Newton's method. In addition, the DNN-RPM can also be used to calculate the best entering angle of NMC orbit injection problem and the minimum reconfiguration time of spacecraft swarm.

The remainder of this paper is organised as follows. Firstly, the time-optimal problem is formulated, and the optimal control is derived based on Pontryagin's maximum principle. Then, the non-convex time-optimal problem is reformulated into a double-layer optimisation framework, making it more tractable to solve with convex programming. Thirdly, the HSM and DNN-RPM are put forward to

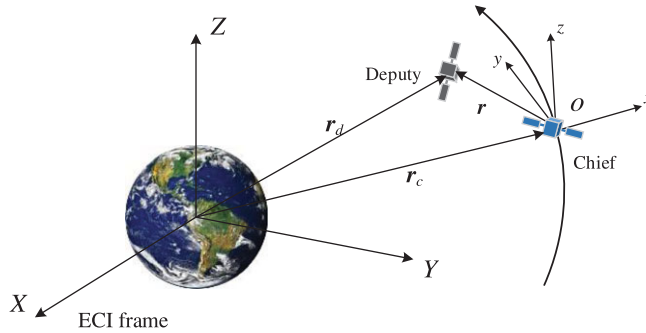


Figure 1. The description of coordinate frames.

search for MFT. Finally, comparative numerical simulations are carried out to validate the performance of the proposed optimisation framework and methods.

### 2.0 Problem formulation

The spacecraft rendezvous contains deputy and chief satellites, generally described in the local vertical/local horizontal (LVLH) coordinate frame attached to the mass centre of chief spacecraft. As shown in Fig. 1,  $r_d$  and  $r_c$  represent the position vector of deputy and chief spacecraft in the Earth centred inertial frame (ECI). The  $x$ -axis coincides with  $r_c$ , the  $z$ -axis is aligned with moment of momentum, and the  $y$ -axis completes the right-hand system [27].

The relative dynamic of deputy satellite in LVLH frame can be described as

$$\dot{v} = -2\omega \times v - \omega \times (\omega \times r) - \epsilon \times r - \frac{\mu r_d}{r_d^3} + \frac{\mu r_c}{r_c^3} + \frac{T}{m} \tag{1}$$

where  $r$  and  $v$  are the relative position and velocity of deputy spacecraft,  $\omega$  and  $\epsilon$  denote the instantaneous angular velocity and acceleration of chief spacecraft,  $T$  and  $m$  mean the thrust and mass of deputy spacecraft,  $\mu$  is the gravitational constant of Earth,  $r_d = \|r_d\|_2$ ,  $r_c = \|r_c\|_2$ . If the target orbit is a near-circular orbit and its radius is much larger than relative distance simultaneously, the equation (1) can be simplified as

$$\dot{v} = -2\omega \times v - \omega \times (\omega \times r) - \frac{\mu}{r_c^3} \left( r - \frac{3r_c \cdot r \cdot r_c}{r_c^2} \right) + \frac{T}{m} \tag{2}$$

Mass consumption obeys

$$\dot{m} = -\frac{\|T\|_2}{I_{sp}g_0} \tag{3}$$

where  $I_{sp}$  is vacuum specific impulse of thruster,  $g_0=9.80665\text{m/s}^2$  denote gravitational acceleration at sea level.

In addition, the manoeuvre procedure for spacecraft rendezvous needs to satisfy boundary conditions in equation (4) and (5), and the limitation on thrust in equation (6).

$$x(t_0) = x_0, \quad m(t_0) = m_0 \tag{4}$$

$$x(t_f) = x_f \tag{5}$$

$$\|T\|_2 \leq T_{\max} \tag{6}$$

where  $\mathbf{x}$  is the relative state of deputy spacecraft,  $\mathbf{x}_0$  and  $\mathbf{x}_f$  are the initial and terminal relative states,  $m_0$  is the initial mass,  $t_f$  means terminal flight time,  $T_{\max}$  is the maximum thrust. The time-optimal control problem can be summarised as Problem 1.

**Problem 1 (time-optimal control problem)**

$$\begin{cases} \min J_1 = t_f \\ s.t. \\ \dot{\mathbf{v}} = -2\boldsymbol{\omega} \times \mathbf{v} - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) - \frac{\mu}{r_c^3} \left( \mathbf{r} - \frac{3\mathbf{r}_c \cdot \mathbf{r} \cdot \mathbf{r}_c}{r_c^2} \right) + \frac{\mathbf{T}}{m} \\ \dot{m} = -\frac{\|\mathbf{T}\|_2}{I_{sp}g_0} \\ \mathbf{x}(t_0) = \mathbf{x}_0, \mathbf{x}(t_f) = \mathbf{x}_f, m(t_0) = m_0 \\ \|\mathbf{T}\|_2 \leq T_{\max} \end{cases} \quad (7)$$

**Assumption 1.** The maximum acceleration is larger than the resultant acceleration of gravitational, centripetal and Coriolis accelerations.

$$\frac{T_{\max}}{m} > \left\| -2\boldsymbol{\omega} \times \mathbf{v} - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) - \frac{\mu}{r_c^3} \left( \mathbf{r} - \frac{3\mathbf{r}_c \cdot \mathbf{r} \cdot \mathbf{r}_c}{r_c^2} \right) \right\|_2 \quad (8)$$

The resultant acceleration varies with orbit height and relative states  $\mathbf{x}$  and it's of the order of  $10^{-5} \sim 10^{-3} \text{m/s}^2$ . Assumption 1 is rational and easy to satisfy since the thruster acceleration is generally greater than the relevant acceleration terms on the right-hand side.

**Proposition 1.** The optimal control of Problem 1 must be  $\|\mathbf{T}^*\|_2 = T_{\max}$  when  $t < t_f$ , and  $\|\mathbf{T}^*\|_2 \leq T_{\max}$  can only happen at  $t=t_f$  and  $\lambda_v(t_f) = 0$ .

**Proof.** Denoting the thrust as

$$\mathbf{T} = T_{\max} u \boldsymbol{\theta} \quad (9)$$

where  $u \in [0, 1]$  is the parameter to adjust thrust magnitude,  $\boldsymbol{\theta}$  denotes the unit direction vector.

Assuming that the  $t_f^*$  and  $\mathbf{T}^*$  are the MFT and optimal control for Problem 1. According to Pontryagin's maximum principle, the Hamiltonian function for Problem 1 is

$$\mathbf{H}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\theta}, u) = \lambda_r \dot{\mathbf{r}} + \lambda_v \left[ -2\boldsymbol{\omega} \times \mathbf{v} - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) - \frac{\mu}{r_2^3} \left( \mathbf{r} - \frac{3\mathbf{r}_2 \cdot \mathbf{r} \cdot \mathbf{r}_2}{r_2^2} \right) + \frac{T_{\max} u \boldsymbol{\theta}}{m} \right] - \lambda_m \frac{T_{\max} u}{I_{sp}g_0} \quad (10)$$

The costates  $\boldsymbol{\lambda} = [\lambda_x, \lambda_y, \lambda_z, \lambda_{v_x}, \lambda_{v_y}, \lambda_{v_z}, \lambda_m]^T$  satisfy the following differential equations and transversality condition,

$$\begin{aligned} \dot{\lambda}_r &= -\frac{\partial \mathbf{H}}{\partial \mathbf{r}} = \frac{\partial \{\lambda_v \cdot [\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r})]\}}{\partial \mathbf{r}} + \frac{\mu}{r_2^3} \frac{\partial [\lambda_v \cdot (\mathbf{r} - 3(\mathbf{r}_2 \cdot \mathbf{r} \cdot \mathbf{r}_2)/r_2^2)]}{\partial \mathbf{r}} \\ &= \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \lambda_v) + \frac{\mu}{r_2^3} (\lambda_v - 3(\mathbf{r}_2 \cdot \mathbf{r}_2^T) \cdot \lambda_v / r_2^2) \end{aligned} \quad (11)$$

$$\dot{\lambda}_v = -\frac{\partial \mathbf{H}}{\partial \mathbf{v}} = -\lambda_r + 2 \frac{\partial [\lambda_v \cdot (\boldsymbol{\omega} \times \mathbf{v})]}{\partial \mathbf{H}} = -\lambda_r + 2\lambda_v \times \boldsymbol{\omega} \quad (12)$$

$$\dot{\lambda}_m = -\frac{\partial \mathbf{H}}{\partial m} = \frac{T_{\max} u}{m^2} \lambda_v \cdot \boldsymbol{\theta} \quad (13)$$

$$\lambda(t_f) = \frac{\partial(\phi + \mathbf{v}^T \boldsymbol{\psi})}{\partial \mathbf{x}(t_f)} = [v_1, v_2, v_3, v_4, v_5, v_6, 0]^T \quad (14)$$

where  $\lambda_r = [\lambda_x, \lambda_y, \lambda_z]^T$ ,  $\lambda_v = [\lambda_{v_x}, \lambda_{v_y}, \lambda_{v_z}]^T$ ,  $\phi$  and  $\psi$  are the performance index and terminal boundary conditions, respectively.  $v$  is the Lagrangian multiplier vector. Since there is no boundary condition for  $m$  at  $t_f$ ,  $\lambda_m(t_f) = 0$ .

The optimal terminal condition is

$$H(t_f) = -\frac{\partial(\phi + v^T\psi)}{\partial t_f} = -1 \tag{15}$$

**Case 1:**  $\|\lambda_v\| \neq 0$  at  $t \in [t_0, t_f]$ .

The optimal control is used to minimise Hamiltonian function,

$$\min H(x^*, \lambda^*, \theta, u) = H(x^*, \lambda^*, \theta^*, u^*) \tag{16}$$

Hence the optimal direction and magnitude of optimal control are

$$\theta^* = -\frac{\lambda_v}{\|\lambda_v\|} \tag{17}$$

$$u^* = \begin{cases} 1 & \text{if } \rho < 0 \\ [0, 1] & \text{if } \rho = 0 \\ 0 & \text{if } \rho > 0 \end{cases} \tag{18}$$

The switching function  $\rho$  is

$$\rho = -T_{\max} \left( \frac{\|\lambda_v\|}{m} + \frac{\lambda_m}{I_{sp}g_0} \right) \tag{19}$$

Since  $\lambda_m(t_f) = 0$  and  $\dot{\lambda}_m < 0$ , thus

$$\begin{cases} \lambda_m > 0 & \text{if } t < t_f \\ \lambda_m = 0 & \text{if } t = t_f \end{cases} \tag{20}$$

Substituting equation (20) into (19), the switching function  $\rho < 0$ . Therefore, if  $\lambda_v \neq 0$  at  $t \in [t_0, t_f]$ ,  $u^* = 1$ , namely  $\|T^*\|_2 = T_{\max}$ .

**Case 2:**  $\lambda_v = 0$  occurs in one or more time intervals.

Assuming that  $\lambda_v = 0$  at  $t \in [t_i, t_j]$ , so  $\dot{\lambda}_v$  is zero,  $\dot{\lambda}_m = \dot{\lambda}_r = \lambda_r = 0$  can then be derived according to equations (11)–(13). Since  $\lambda_r = \dot{\lambda}_r = 0$  at  $t \in [t_i, t_j]$ ,  $\lambda_r$  will stay at zero when  $t \in [t_j, t_f]$ . In the same way,  $\lambda_v$  stay at zero when  $t \in [t_j, t_f]$ .

According to equation (10),  $H(t_f) = 0$  since  $\lambda_r(t_f) = \lambda_v(t_f) = \lambda_m(t_f) = 0$ , which however contradicts to equation (15). Therefore, the assumption is invalid, and case 2 will not happen.

**Case 3:**  $\lambda_v = 0$  occurs at some discrete points when  $t \in [t_0, t_f]$ .

If  $\lambda_v = 0$ , the equation (10) can be reduced as

$$H(x, \lambda, \theta, u) = \lambda_r \dot{r} - \lambda_m \frac{T_{\max} u}{I_{sp}g_0} \tag{21}$$

Accordingly,  $\rho$  is

$$\rho = -\frac{T_{\max} \lambda_m}{I_{sp}g_0} \tag{22}$$

The equations (17)–(20) still hold for all other points where  $\|\lambda_v\| \neq 0$ . Only if  $t = t_f$  and  $\lambda_v(t_f) = 0$ ,  $\rho = 0$ , so  $u^* = [0, 1]$ . Otherwise,  $\rho < 0$  and  $u^* = 1$  still hold.

According to the analysis of above three cases, the magnitude of optimal thrust always remains at maximum when  $t < t_f$ , and  $\|T^*\|_2 = T_{\max}$  can only happen at  $t = t_f$  and  $\lambda(t_f) = 0$ .

### 3.0 The double-layer optimisation framework

In this section, the minimum terminal error problem, which can be described as convex form, is introduced firstly, and then a double-layer optimisation framework is proposed to search for MFT. Taking the terminal error between actual and desired state as objective, the minimum terminal error problem at given flight time  $t_f$  can be described as follows,

**Problem 2 (Minimum-terminal-error problem)**

$$\begin{cases} \min \min J_2 = \|\mathbf{x}(t_f) - \mathbf{x}_f\|_2 \\ \text{s.t.} \\ \dot{\mathbf{v}} = -2\boldsymbol{\omega} \times \mathbf{v} - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) - \frac{\mu}{r^3}(\mathbf{r} - \frac{3\mathbf{r}_c \cdot \mathbf{r} \mathbf{r}_c}{r_c^2}) + \frac{\mathbf{T}}{m} \\ \dot{\mathbf{m}} = -\frac{\|\mathbf{T}\|_2}{I_{sp}g_0} \\ \mathbf{x}(t_0) = \mathbf{x}_0, \mathbf{m}(t_0) = \mathbf{m}_0 \\ \|\mathbf{T}\|_2 \leq T_{\max} \end{cases} \quad (23)$$

It's worth noting that the objective and inequality constraints of Problem 2 are convex functions, and equality constraints are affine form, so it could be transformed into a convex optimisation problem. There are some propositions for Problem 2.

**Proposition 2.** *The optimal objective of Problem 2 is larger than zero when  $t_f < t_f^*$ , and the optimal objective is equal to zero when  $t_f \geq t_f^*$ .*

**Proof:** Assuming that the optimal objective  $J_2(t_f) = 0$  when  $t_f < t_f^*$ , namely  $\mathbf{x}(t_f) - \mathbf{x}_f = 0$ . It means that  $t_f$  is a feasible solution for Problem 1 and  $t_f < t_f^*$ , which however is contrary to the definition of time-optimal control problem. Therefore, the optimal objective of Problem 2 is more than zero when  $t_f < t_f^*$ .

If  $t_f = t_f^*$ , the optimal solution to Problem 1 is such that  $\mathbf{x}(t_f) - \mathbf{x}_f = 0$ , and hence the optimal objective of Problem 2 is zero.

When  $t_f > t_f^*$ , the following control is feasible for Problem 2, and the objective remains zero.

$$\mathbf{T} = \begin{cases} \mathbf{T}^* & \text{if } t \leq t_f^* \\ m(2\boldsymbol{\omega} \times \mathbf{v} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) + \frac{\mu}{r^3}(\mathbf{r} - \frac{3\mathbf{r}_c \cdot \mathbf{r} \mathbf{r}_c}{r_c^2})) & \text{if } t_f^* \leq t \leq t_f \end{cases} \quad (24)$$

**Proposition 3.** *The optimal thrust of Problem 2 must be  $\|\mathbf{T}^*\|_2 = T_{\max}$  when  $t_f < t_f^*$ .*

**Proof:** Since the Hamiltonian function and costates function of Problem 2 are the same as Problem 1, the analysis of cases 1 and 3 still hold for Problem 2. When it comes to case 2, the transversality condition for minimum terminal error problem is

$$\boldsymbol{\lambda}(t_f) = \frac{\partial \phi}{\partial \mathbf{x}(t_f)} = \frac{[\mathbf{x}(t_f) - \mathbf{x}_f; 0]}{\|\mathbf{x}(t_f) - \mathbf{x}_f\|} \quad (25)$$

According to Proposition 2,  $\mathbf{x}(t_f) \neq \mathbf{x}_f$  when  $t_f < t_f^*$ , thus  $\|\boldsymbol{\lambda}(t_f)\| \neq 0$ , which contradicts with the condition that  $\|\boldsymbol{\lambda}(t_f)\| = 0$ .

**Proposition 4.** *If  $t_f \leq t_f^*$ , the optimal terminal error for Problem 2 decreases with  $t_f$ .*

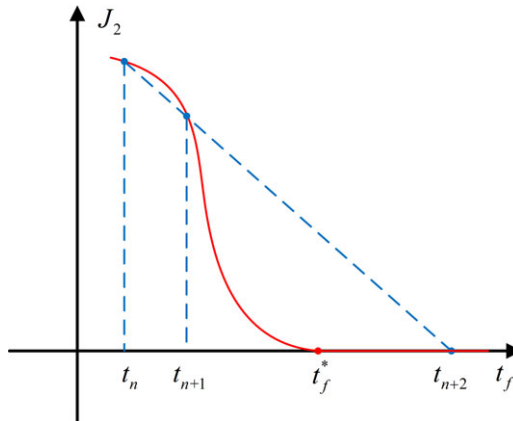
**Proof:** If  $t_a < t_b \leq t_f^*$ ,  $\mathbf{T}_a^*$  and  $\mathbf{T}_b^*$  are the corresponding optimal control, assuming that  $\mathbf{T}_b$  is

$$\mathbf{T}_b = \begin{cases} \mathbf{T}_a^* & \text{if } t \leq t_a \\ m \left( 2\boldsymbol{\omega} \times \mathbf{v} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) + \frac{\mu}{r^3} \left( \mathbf{r} - \frac{3\mathbf{r}_c \cdot \mathbf{r} \mathbf{r}_c}{r_c^2} \right) \right) & \text{if } t_a \leq t \leq t_b \end{cases} \quad (26)$$

It's evident that  $J(\mathbf{T}_b) = J(\mathbf{T}_a^*)$ . In addition,  $J(\mathbf{T}_b) < J(\mathbf{T}_b^*)$  according to Proposition 3. Therefore,  $J(\mathbf{T}_b) = J(\mathbf{T}_a^*) < J(\mathbf{T}_b^*)$  when  $t_a < t_b \leq t_f^*$ .

**Table 1.** The relationship  $J_3$  and  $t_f$

$t_f$	$J_{31} = \ \mathbf{x}(t_f) - \mathbf{x}_f\ _2$	$J_{32} = -\int_{t_0}^{t_f} (\ \mathbf{T}(t)\ _2 - T_{\max}) dt$	$J_3 = J_{31} + J_{32}$
$t_f < t_f^*$	$J_{31} > 0$	$J_{32} = 0$	$J_3 > 0$
$t_f = t_f^*$	$J_{31} = 0$	$J_{32} = 0$	$J_3 = 0$
$t_f > t_f^*$	$J_{31} = 0$	$J_{32} < 0$	$J_3 < 0$



**Figure 2.** The iterative process of Newton's method.

Based on the analysis in Propositions 2 and 4, the minimum terminal error  $J_2$  decreases monotonically as  $t_f$  when  $t_f < t_f^*$  and remains zero when  $t_f \geq t_f^*$ . Therefore, the MFT is equal to the turning point where  $J_2=0$ .

Unfortunately, this kind of root-finding problem is still intractable for typical search algorithms, such as the bisection and Newton's method. As shown in Fig. 2, if the iterative points  $t_n$  and  $t_{n+1}$  are far away  $t_f^*$  or once the iterative points lie on the right of  $t_f^*$ , Newton's method will not converge. The bisection method is also infeasible since it iterates over two values with opposite signs, while  $J_2$  remains zero when  $t_f \geq t_f^*$ .

Inspired by Proposition 3, a new performance metrics is constructed as follows

$$J_3 = \|\mathbf{x}(t_f) - \mathbf{x}_f\|_2 - \int_{t_0}^{t_f} (\|\mathbf{T}(t)\|_2 - T_{\max}) dt \tag{27}$$

**Proposition 5.**  $J_3 > 0$  when  $t_f < t_f^*$ ,  $J_3 = 0$  when  $t_f = t_f^*$ , and  $J_3 < 0$  when  $t_f > t_f^*$ .

**Proof:** As shown in Table 1, denoting the two terms in equation (27) as  $J_{31}$  and  $J_{32}$  respectively. Based on Proposition 2,  $J_{31}$ , namely the terminal error, is larger than zero when  $t_f < t_f^*$ , and remains zero when  $t_f \geq t_f^*$ . According to Proposition 3,  $\|\mathbf{T}^*\|_2 = T_{\max}$  when  $t_f \leq t_f^*$ , so  $J_{32} = 0$ . When  $t_f > t_f^*$ ,  $\|\lambda(t_f)\| \neq 0$  in equation (25) will not hold, and thus  $\|\mathbf{T}^*\|_2 = T_{\max}$  is impossible. In addition, the equation (24) is a feasible solution for Problem 2, and the magnitude of thrust is not equal to  $T_{\max}$  during the whole flight time, so  $J_{32} < 0$  when  $t_f > t_f^*$ . Combining  $J_{31}$  and  $J_{32}$ ,  $J_3 > 0$  when  $t_f < t_f^*$ ,  $J_3 = 0$  when  $t_f = t_f^*$ , and  $J_3 < 0$  when  $t_f > t_f^*$ .

As shown in Fig. 3, the initial time-optimal problem is transformed into a double-layer optimisation framework, the inner layer is a convex programming problem, namely the minimum-terminal-error problem at given flight time, and the optimisation results are used to construct performance metric for outer layer root-finding problem. Furthermore, it's worth distinguishing that  $J_2$  denotes the optimisation objective for convex programming, while  $J_3$  is only a performance metric constructed based on thrust and terminal error to guide the search for MFT.



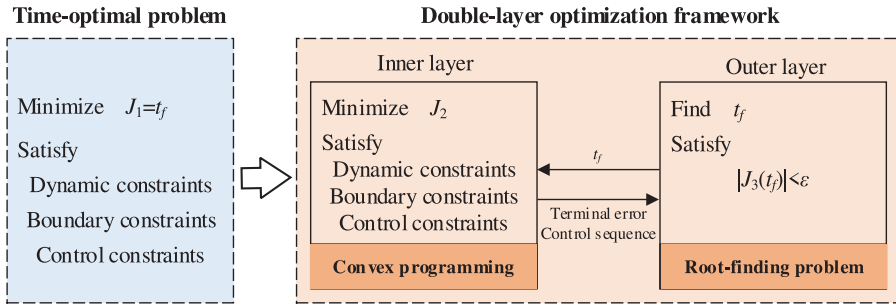


Figure 3. The double-layer optimisation framework.

4.0 The DNN-based rapid search method

In this section, the HSM combining bisection and Newton’s method is proposed to generate dataset for DNN training, and then a rapid search algorithm based on DNN is put forward to search for MFT.

4.1 Dataset generation and preprocessing

Based on Proposition 5, the time-optimal control problem has been transformed into a double-layer optimisation problem, where the outer layer can be solved by the iteration-based numerical method, such as the bisection method and Newton’s method. The bisection method repeatedly bisects the solution space and then selects subintervals in which the function changes signs until the convergence condition is satisfied. It’s robust and easy to implement, but computationally inefficient. On the contrary, the convergence of Newton’s method is quadratic, but it’s sensitive to initial points.

In view of this, we proposed the HSM that combines the advantage of both methods. The main idea is to reduce the solution space by the bisection method until the absolute function value of two boundary points is less than the preset value  $\bar{J}_3$  and then the boundary points are used as starting points for Newton’s method. The bisection method provides a rough interval for Newton’s method to reduce iteration steps and thus improve robustness and calculation efficiency. The flowchart of HSM is concluded in Fig. 4.

A dataset containing  $5 \times 10^4$  scenarios is randomly generated, and the MFT of each scenario is solved by the HSM. Assuming the chief spacecraft moves on 500km high circular orbit, the initial and final positions of deputy spacecraft are uniformly distributed within  $\pm 5\text{km}$ , and the initial and final velocities of deputy spacecraft are uniformly distributed within  $\pm 2\text{m/s}$ . Due to the large magnitude difference among the parameters, each parameter of the initial dataset is transformed into dimensionless values within  $[0, 1]$  by min-max normalisation.

$$p' = \frac{p - p_{\min}}{p_{\max} - p_{\min}} \tag{28}$$

where  $p$  is the initial parameter,  $p_{\min}$  and  $p_{\max}$  are the minimum and maximum values of each parameter,  $p'$  denotes the normalised parameter. The normalised dataset is divided into training datasets, validation datasets and test datasets in a ratio of 8:1:1. The three datasets are used to train the network, prevent overfitting and evaluate network performance, respectively.

4.2 Neural network design and training

The DNN consists of an input layer, an output layer and multiple hidden layers, each hidden layer contains multiple neurons. The dimensions of input and output layers are consistent with input and output data. In this paper, the sequence  $[x_0, y_0, z_0, \dot{x}_0, \dot{y}_0, \dot{z}_0, x_f, y_f, z_f, \dot{x}_f, \dot{y}_f, \dot{z}_f]$  consisting of the initial and final

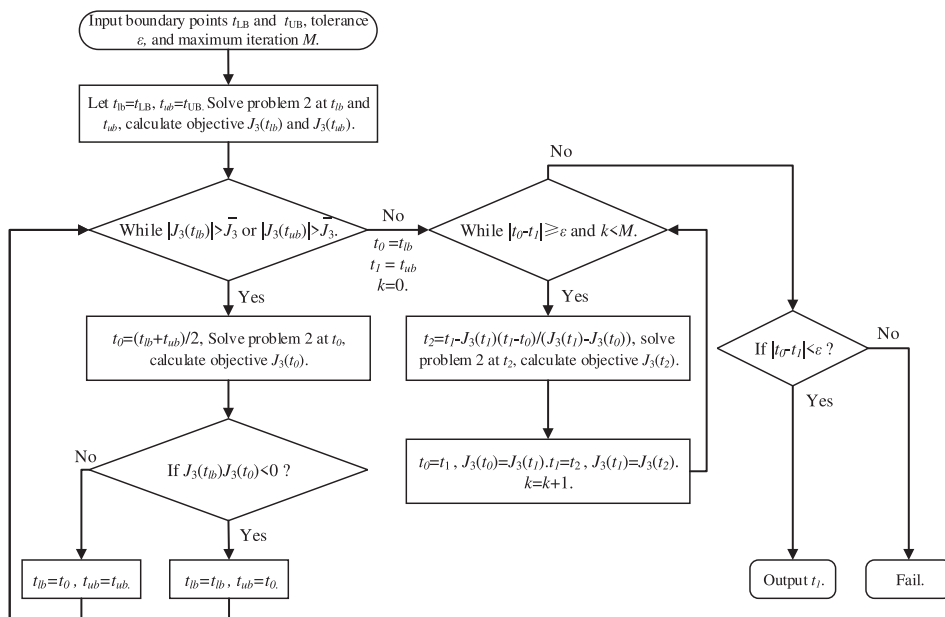


Figure 4. Flowchart of hybrid search method.

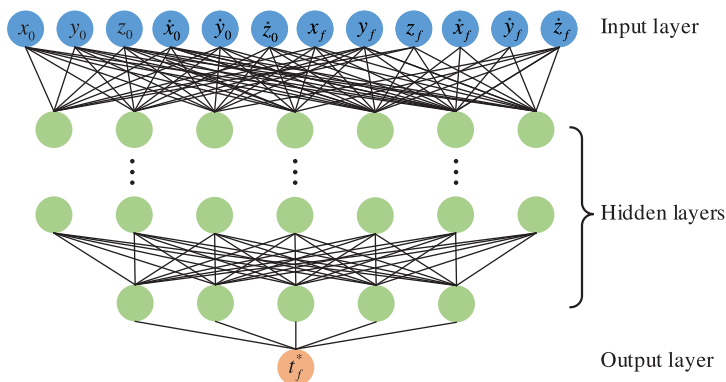


Figure 5. The structure of DNN.

relative states is selected as the input of DNN and the MFT  $t_f^*$  as the output, the structure of DNN is shown in Fig. 5.

The optimisation objective of neural network is the mean square error (MSE) between the training sample and the predicted value,

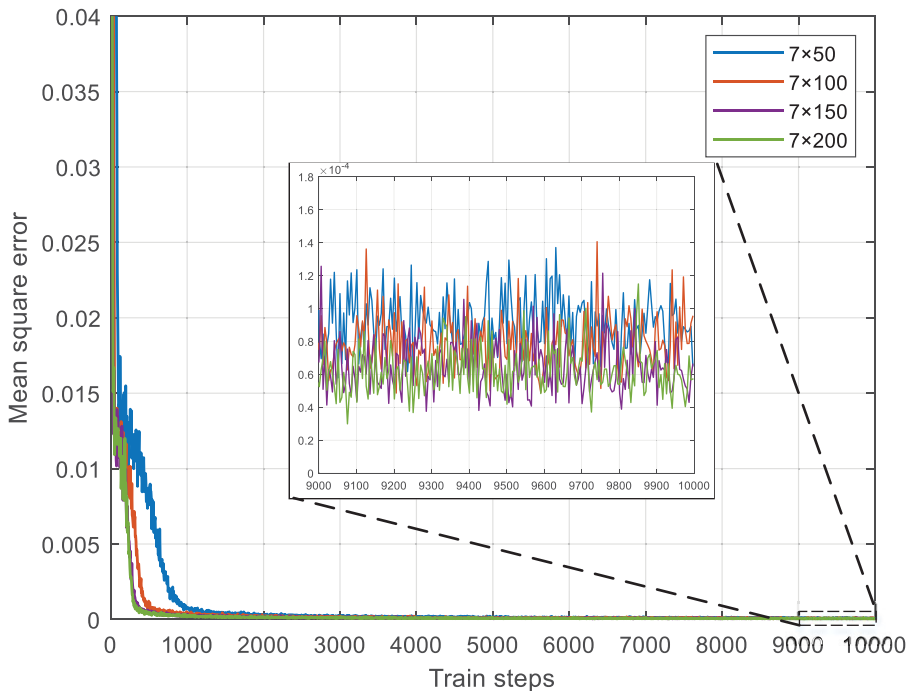
$$MSE = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i) \tag{29}$$

where  $N$  is the size of train dataset,  $y_i$  and  $\hat{y}_i$  respectively represents real value and predicted value of DNN. The DNN training is based on deep learning toolbox built in MATLAB, the related training options are listed in Table 2.

The training process of four DNNs with different size is shown in Fig. 6, where the mark  $Net_{7 \times 50}$  indicates the number of hidden layers and neurons in hidden layer are 7 and 50. As shown in Fig. 6, the mean square error decreases rapidly with the increase of training steps, and finally fluctuates at the

**Table 2.** The training options for DNN

Training options	Value
Optimiser	Adaptive moment estimation (Adam)
Max epochs	500
Mini-batch size	200
Validation frequency	500
Initial learn rate	$1 \times 10^{-4}$
Learn rate schedule	Piecewise
Learn rate drop period	100
Learn rate drop period factor	0.5

**Figure 6.** The training process of DNNs.

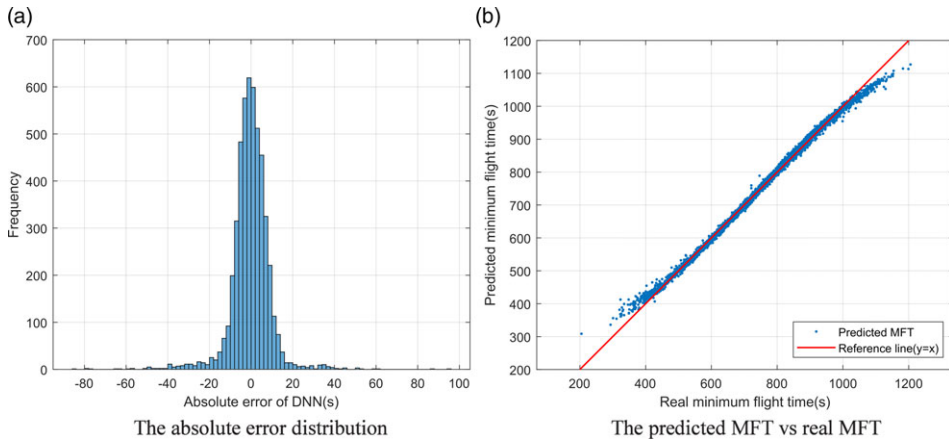
order of  $10^{-4}$ . In addition, the convergence speeds of  $Net_{7 \times 200}$  and  $Net_{7 \times 150}$  with larger network size are faster than that of  $Net_{7 \times 50}$  and  $Net_{7 \times 100}$ .

The performance of DNNs in test dataset is listed in Table 3. The relative error between real and predicted MFT follows the Gaussian distribution, and the mean of the relative error is approximately zero. With the increase of neurons, the standard deviation of relative error first decreases and then increases. The  $Net_{7 \times 150}$  performs well in both mean and standard deviation, and was selected to predicted MFT in the subsequent simulations. The  $3\sigma$  interval of  $Net_{7 \times 150}$  is  $[-5.11\%, 5.21\%]$ , namely the probability of relative error within this range is 99.73%.

The performance of  $Net_{7 \times 150}$  in test dataset are shown in Fig. 7. It can be seen from Fig. 7(a) that the absolute error of predicted MFT obeys Gaussian distribution, most of the predicted errors are within 20s. As shown in Fig. 7(b), the predicted MFT (y axis) can fit real MFT (x axis) well, where the blue point and red line represent predicted MFT and reference line ( $y=x$ ) respectively. Generating dataset and training DNNs took about 15 hours and 1 hour in a computer with Intel i7 processor and 16GB random

**Table 3.** The relative error distribution of different DNNs

DNN size	Relative error mean (%)	Relative error standard deviation (%)	3σ interval
7×50	0.08	1.96	[−5.78%, 5.95%]
7×100	0.07	1.76	[−5.21%, 5.36%]
7×150	0.05	1.72	[−5.11%, 5.21%]
7×200	0.09	1.84	[−5.43%, 5.61%]



**Figure 7.** The performance of  $Net_{7 \times 150}$ .

access memory. Notably, the process of generating dataset and training the network can be implemented offline, and MFT will be predicted using trained DNN parameters online.

### 4.3 DNN-based rapid search method

The Newton’s method is sensitive to the initial point, high-quality initial points can ensure quadratic convergence, while poor initial points may lead to failure to converge. The proposed HSM takes the iterative result of the bisection method as the initial point of Newton’s method, which is robust but time-consuming.

Section 4.2 shows that the trained DNN can directly output the high-precision predicted value of MFT according to the initial and terminal boundary conditions. Although the predicted MFT cannot be directly applied to space tasks with high reliability requirements due to unavoidable errors, it can be taken as the initial points of Newton’s method, thus eliminating the binary inefficient search process and guaranteeing quadratic convergence of Newton’s method. The proposed DNN-RPM is robust and efficient, the diagram of HSM and DNN-RPM is shown in Fig. 8.

## 5.0 Numerical simulations

In this section, the HSM and DNN-RPM are firstly applied to generate time-optimal manoeuvring trajectory for single spacecraft, and the relevant properties of the time-optimal problem proved above are validated by simulation results. And then, the time-optimal NMC orbit injection problem is solved to validate the performance of proposed methods. Finally, three DNN-based strategies corresponding to different reliability are proposed to solve the minimum reconfiguration time of swarm.

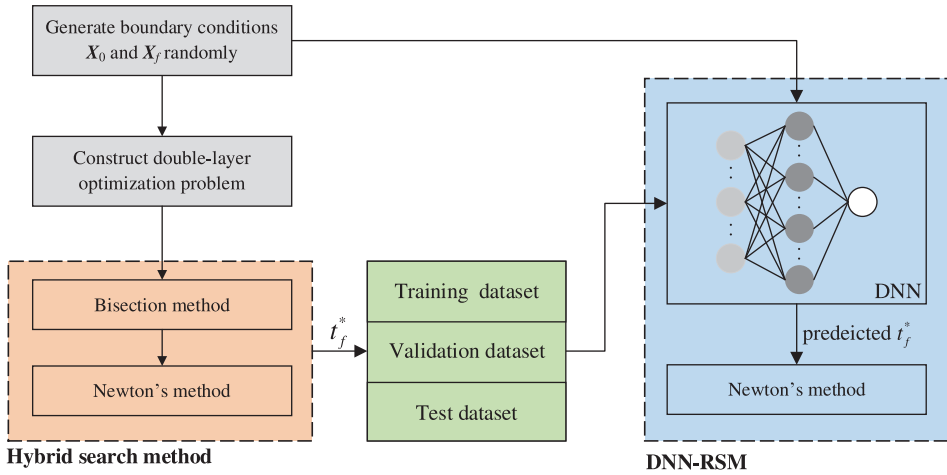


Figure 8. The diagram of HSM and DNN-RPM.

Assuming that the chief spacecraft is in a 500km high near-circular orbit, the initial mass of each satellite in the swarm is 1,000kg. The maximum thrust and  $I_{sp}$  of the engine are 50N and 200s, respectively. The simulation process is discretised into 100 steps. Additionally, the tolerance and max iteration for HSM are  $10^{-3}$  and 50, the preset  $\bar{J}_3 = 50$ , the lower and upper boundary of flight time are 100 and 3,000s. The CVX software is utilised to solve convex programming problem, where the SDPT3 is chosen as solvers, and other parameters remain the default setting. The numerical simulations are implemented on a computer with a 3.0GHz Intel Core i7 processor and 16GB random access memory.

5.1 Time-optimal trajectory planning for spacecraft rendezvous

For time-optimal trajectory planning of single spacecraft, the initial and terminal relative states are set to  $X_0=[10^3 \text{ m}, 10^4 \text{ m}, 0 \text{ m}, 0 \text{ m/s}, -2.21 \text{ m/s}, 2.21 \text{ m/s}]^T$ ,  $X_f=[866.03 \text{ m}, -10^3 \text{ m}, 0 \text{ m}, -0.55 \text{ m/s}, -1.9 \text{ m/s}, 0 \text{ m/s}]^T$ .

To begin with, the minimum terminal error problem (i.e., Problem 2) is solved with a given flight time, sampled every 10s between lower and upper boundary, and the performance index  $J_{31}$ ,  $J_{32}$  and  $J_3$  are listed in Fig. 9. As defined in Section 3, the  $J_{31}$  denotes the terminal error,  $J_{32}$  is used to measure the difference between thrust curve and maximum thrust, and the  $J_3$  is the sum of these two indexes. It can be seen from Fig. 9 that the optimal terminal error  $J_{31}$  is larger than zero and decreases with time when  $t_f < t_f^*$ , and it's equal to zero when  $t_f \geq t_f^*$ , which is consistent with Propositions 2 and 4. As proved in Proposition 3, the thrust magnitude remains  $T_{max}$  when  $t_f \leq t_f^*$ , and it's less than  $T_{max}$  when  $t_f > t_f^*$ , the index  $J_{32}$  is equal to zero in the first part but not in the second part. Therefore, the MFT for satellite manoeuvring is equal to the root of  $J_3$ .

And then, the time-optimal problem is solved with the proposed HSM and DNN-RPM, the relevant state and thrust curves are illustrated in Figs. 10–11. It can be seen from Fig. 10 that the proposed method can successfully generate transfer trajectory with the limitation of various constraints. As shown in Fig. 11, although the thrust components vary with time, the thrust magnitude remains at  $T_{max}$ , which is consistent with Proposition 3.

Furthermore, in order to evaluate the performance of the proposed methods, the time-optimal problem with the same configuration is also solved by the traditional bisection method, Newton's method and sequential quadratic programming (SQP). The SQP method can be implemented by invoking the Fmincon function from Matlab optimisation toolbox directly, which is recognised as a powerful non-linear programming solver. It's worth noting that the first four methods solve the time-optimal problem

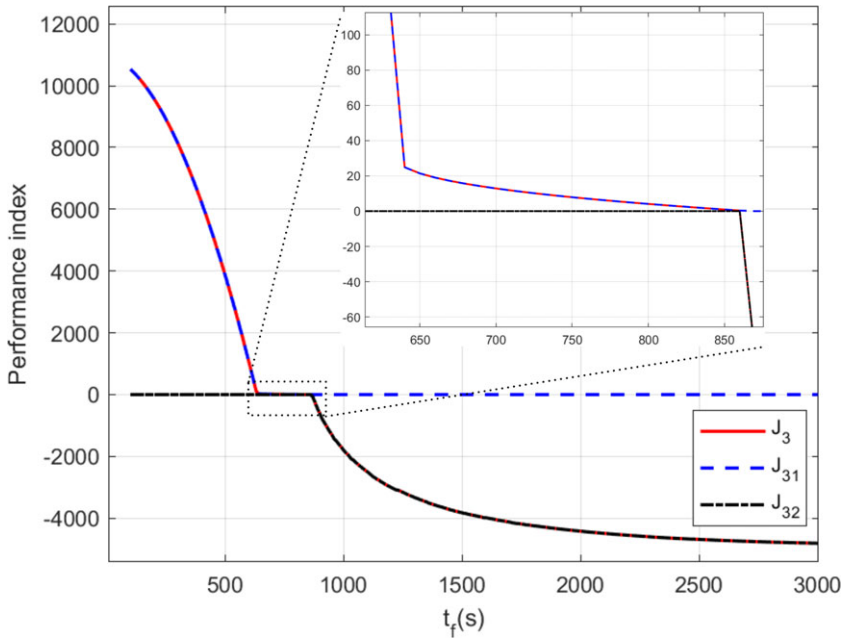


Figure 9. The performance index for different flight time.

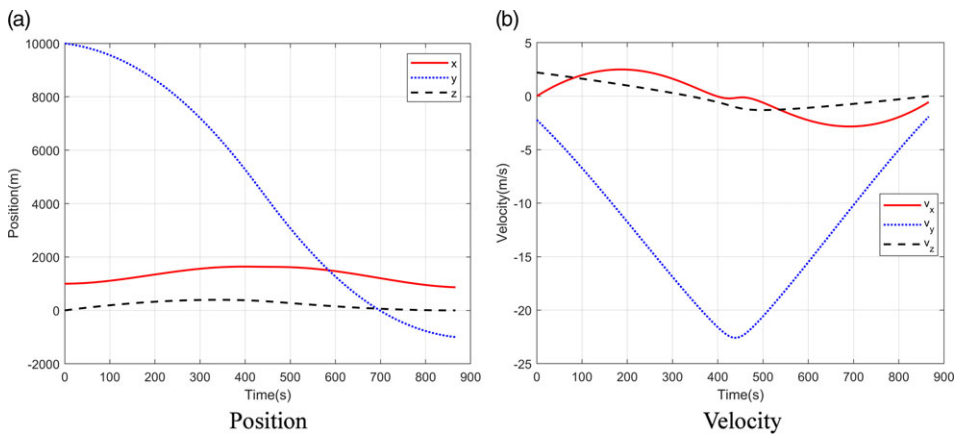


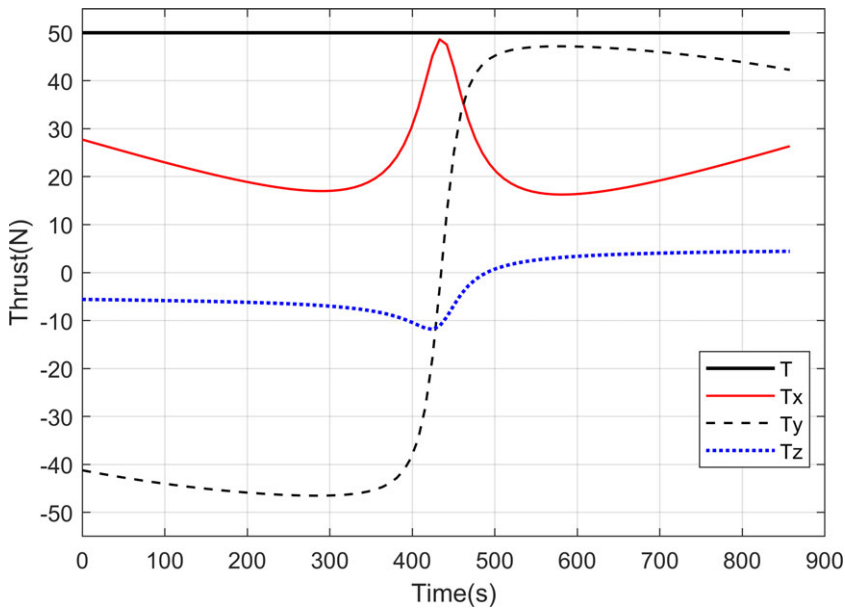
Figure 10. The time histories of state variables.

based on convex programming, and the methods themselves are only used to search the MFT, while the SQP method optimises the control sequence and MFT simultaneously. Since the computational efficiency and convergence speed of SQP and traditional Newton’s method are greatly affected by initial value, these two methods are executed 50 times with random initial values. The performance comparison of different methods is listed in Table 4. Since the computing time of SQP is much larger than that of other methods, Fig. 12 only shows the computing time comparison of DNN-RPM, HSM, bisection method and Newton’s method.

It can be directly observed from Table 4 that the MFT obtained by the above five methods are almost the same. When it comes to the computation time criterion in Table 4, the first four methods take about two orders of magnitude less time than the SQP. Though the computation time could be affected by various factors like computer configuration and programme integration level, it shows the superiority

**Table 4.** The performance metrics comparison

Methods	Minimum flight time (s)	Average computing time (s)	Maximum computing time (s)	Minimum computing time (s)
DNN-RPM	866.15	7.31	7.72	7.27
HSM	866.15	14.54	14.82	14.49
Bisection method	866.15	30.14	32.07	30.02
Newton’s method	866.15	12.10	16.90	7.25
SQP	866.15	1494.96	1844.01	1082.54



**Figure 11.** The time histories of thrust.

of proposed double layer framework over nonlinear optimisation method in computational efficiency. The DNN-RPM takes 7.31s to calculate MFT, HSM takes about twice as long, and the bisection method takes about four times as long. As shown in Fig. 12 and Table 4, the traditional Newton’s method is slightly more computationally efficient than the HSM, but fails to converge in 14 out of 50 calculations. Therefore, the proposed DNN-RPM demonstrates its superiority in both computational efficiency and robustness.

**5.2 Natural-motion circumnavigation injection**

The NMC orbit, also known as passive relative orbit, is a kind of closed relative orbit where the deputy can fly around chief without extra fuel consumption [28]. In order to form an NMC orbit, the secular term in y direction should be omitted,

$$\dot{y} = -2nx \tag{30}$$

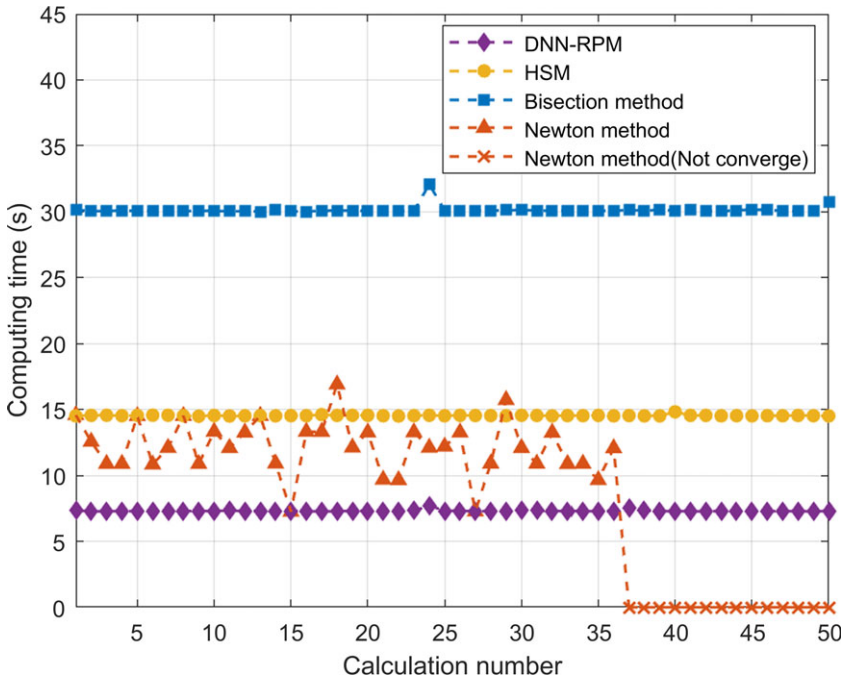


Figure 12. The computing time comparison.

The projection of the NMC orbit into  $x$ - $y$  plane is a  $2 \times 1$  ellipse, and the NMC orbit can be represented by a set of configuration parameters  $[b, c, y_c, \psi, \phi]$ . The conversion equation between relative states and NMC orbit configuration parameters is as follows,

$$\begin{cases} x(t) = b \sin (nt + \phi) \\ y(t) = 2b \cos (nt + \phi) + y_c \\ z(t) = c \sin (nt + \phi + \psi) \\ \dot{x}(t) = nb \cos (nt + \phi) \\ \dot{y}(t) = -2nb \sin (nt + \phi) \\ \dot{z}(t) = nc \cos (nt + \phi + \psi) \end{cases} \quad (31)$$

where  $b$  is the semi-minor axis of in-plane ellipse,  $y_c$  is the displacement of in-plane ellipse centre in  $y$ -axis,  $c$  is the maximum displacement of NMC orbit in  $z$ -axis,  $\phi$  and  $\psi$  are phase angle of the in-plane and out-plane motion.

For the time-optimal NMC injection problem, the target is to find control vector  $T(t)$  and entering phase angle  $\phi$  to minimise flight time and satisfy corresponding constraints. Compared with time-optimal trajectory planning for a single satellite, the NMC injection problem needs to optimise not only how to go (control vector), but also where to go (entering phase angle  $\phi$ ), which greatly increases the difficulty of optimisation. In this case, the SQP algorithm is adopted to search entering phase angle  $\phi$ , the control vectors are generated by HSM and DNN-RPM, respectively.

The NMC orbit configuration parameters are set to  $b=2,000\text{m}$ ,  $c=2,000\text{m}$ ,  $y_c=0\text{m}$ ,  $\psi = \pi/3\text{rad}$ , the initial relative state of deputy satellite is  $X_0 = [5 \times 10^3\text{m}, 5 \times 10^3\text{m}, 5 \times 10^3\text{m}, 0 \text{ m/s}, 0 \text{ m/s}, 0 \text{ m/s}]^T$ . The simulation results are listed in Figs. 13–14 and Table 5.

As shown in Fig. 13, the red solid line and blue dotted line, respectively, represent the time-optimal injection trajectory generated by DNN-RPM and HSM, while the black line represents the NMC orbit. It can be directly observed that the time-optimal injection trajectories of DNN-RPM and HSM are close, which is consistent with the optimised entering phase angle in Table 5.



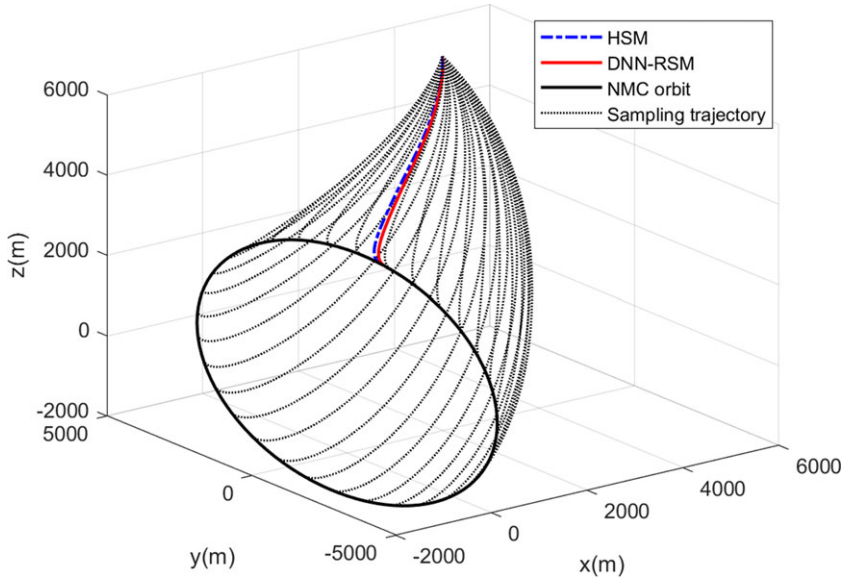


Figure 13. The manoeuvring trajectory of NMC injection.

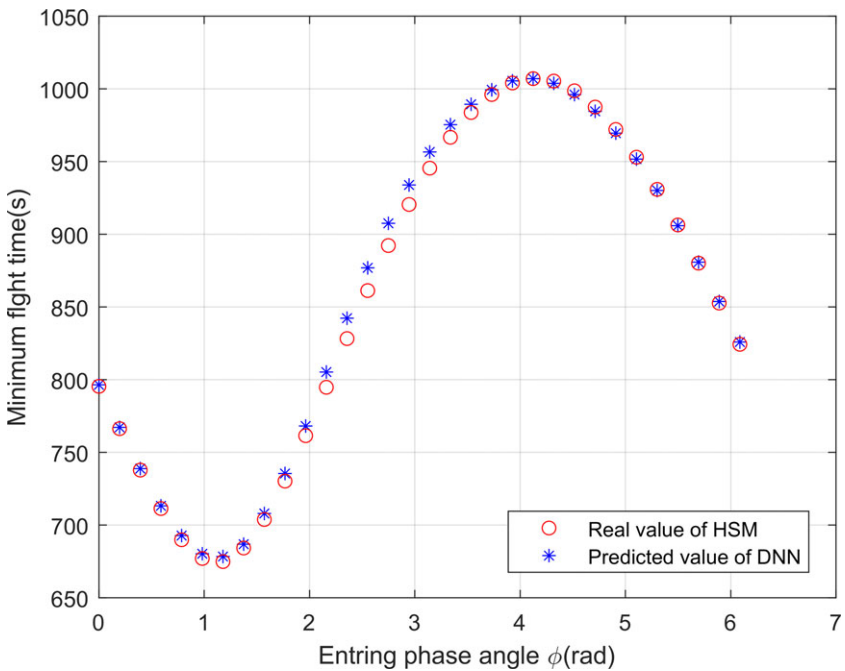


Figure 14. The minimum flight time generated by DNN and HSM.

In addition, Fig. 13 shows the manoeuvring trajectories at 32 sampling points on the NMC orbit with black dot lines, and Fig. 14 demonstrates the MFTs predicted by DNN and calculated by HSM at each sampling point. The predicted MFTs at all sampling points are similar to the real MFTs, which again indicated the high prediction accuracy of DNN.

As shown in Table 5, the difference between the first strategy and the latter two lies in using the calculation result of HSM to guide SQP algorithm to search entering phase angle, rather than DNN's

**Table 5.** The performance metrics of different strategies

Strategy	$\phi$ (rad)	Minimum flight time (s)	Computing time (s)
HSM	1.14	674.30	608.03
DNN	1.13	677.81	0.37
DNN-RPM	1.13	674.32	6.15

predicted value. The entering phase angle calculated by HSM strategy and DNN strategy is very close, and the relative error is less than 1%. The DNN-RPM strategy takes the convergence entering phase angle of DNN strategy as the terminal state, and then calculates the MFT with DNN-RMS method. In other words, the flight time of 677.81 and 674.32s, respectively, represents the MFT predicted by DNN and calculated by DNN-RPM when entering phase angle  $\phi=1.13$ rad. Additionally, the average computing time of HSM is 608.03 s, and that of DNN and DNN-RMS is 0.37 and 6.15s respectively. Due to the high computational efficiency of DNN, the computing time of DNN-RPM is about two orders of magnitude smaller than that of HSM.

### 5.3 The minimum reconfiguration time for swarm

Swarm reconfiguration is one of the enabling technologies for various space missions, which means that all spacecraft manoeuvre from the initial states to the specified terminal states simultaneously. For a cluster containing  $N_s$  spacecraft, each spacecraft corresponds to an MFT, and the swarm reconfiguration time must be larger than  $\max\{MFT_1, MFT_2, \dots, MFT_{N_s}\}$  to ensure that all spacecraft can generate feasible trajectories. Therefore,  $\max\{MFT_1, MFT_2, \dots, MFT_{N_s}\}$  is regarded as the minimum reconfiguration time of spacecraft swarm in this section. To determine the minimum reconfiguration time, the simplest idea is to apply the above HSM or DNN-RPM to calculate the MFT of each spacecraft in turn, and take the maximum MFT as the minimum reconfiguration time of the whole swarm. However, this progress is time-consuming.

Three DNN-based strategies corresponding to different reliability are proposed, which are named high, medium and low reliability, respectively. The main difference among these three strategies is which spacecraft in the swarm should be chosen to calculate its real MFT. The high reliability strategy uses DNN-RPM to calculate the MFTs of all spacecraft in the swarm, and then takes the maximum MFT as the minimum reconfiguration time of the whole swarm. The low reliability strategy first predicts the MFTs of all spacecraft through DNN, and then calculates the real MFT of the slowest spacecraft by DNN-RPM as the minimum reconfiguration time.

For the medium reliability strategy, the MFTs of all spacecraft are firstly predicted by DNN, and then  $n\text{-}\sigma$  intervals of maximum MFT are taken as a filter, thus reducing call number of DNN-RPM. As shown in Fig. 15, the hexagons represent the predicted MFTs, and the I-boxes mean the responding  $n\text{-}\sigma$  intervals. Only one satellite's  $n\text{-}\sigma$  interval intersects with that of the maximum predicted MFT in Fig. 15, so there is no need to calculate the real MFT of other satellites. According to the Gaussian distribution characteristics, when  $n$  is set to 1, 2 and 3, the reliability of this strategy is 68.27%, 95.45% and 99.73%. The difference among these three strategies is shown in Fig. 16.

The initial positions of swarm satellites are located on the eight corners of a  $10\text{km} \times 10\text{km} \times 10\text{km}$  cube centred at the chief in the LVLH frame, and the initial velocities are set to  $0\text{m/s}$ . The terminal positions are uniformly distributed on an NMC orbit with  $b=2,000\text{m}$ ,  $c=0\text{m}$ ,  $y_c=0\text{m}$ ,  $\psi = \pi$ rad. The manoeuvring trajectories generated by DNN-RPM are presented in Fig. 17, and the computing time of four strategies are listed in Table 6.

As shown in Table 6, the minimum reconfiguration times calculated by HSM and three DNN based strategies are similar, while the computing times are quite different. The first two strategies are based on the traversal idea and require calling HSM and DNN-RPM eight times, respectively. The difference in computing time of these two strategies depends on the calculation efficiency of HSM and DNN-RPM.

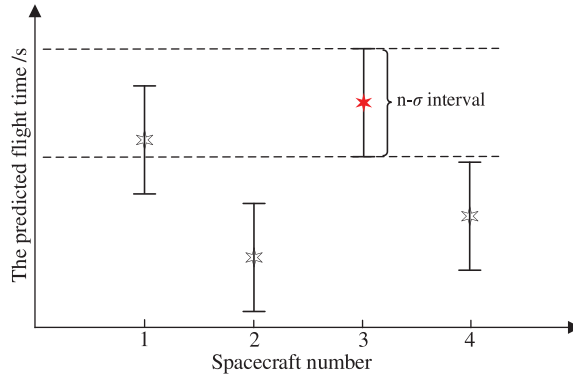


Figure 15. The  $n\text{-}\sigma$  intervals of predicted MFT.

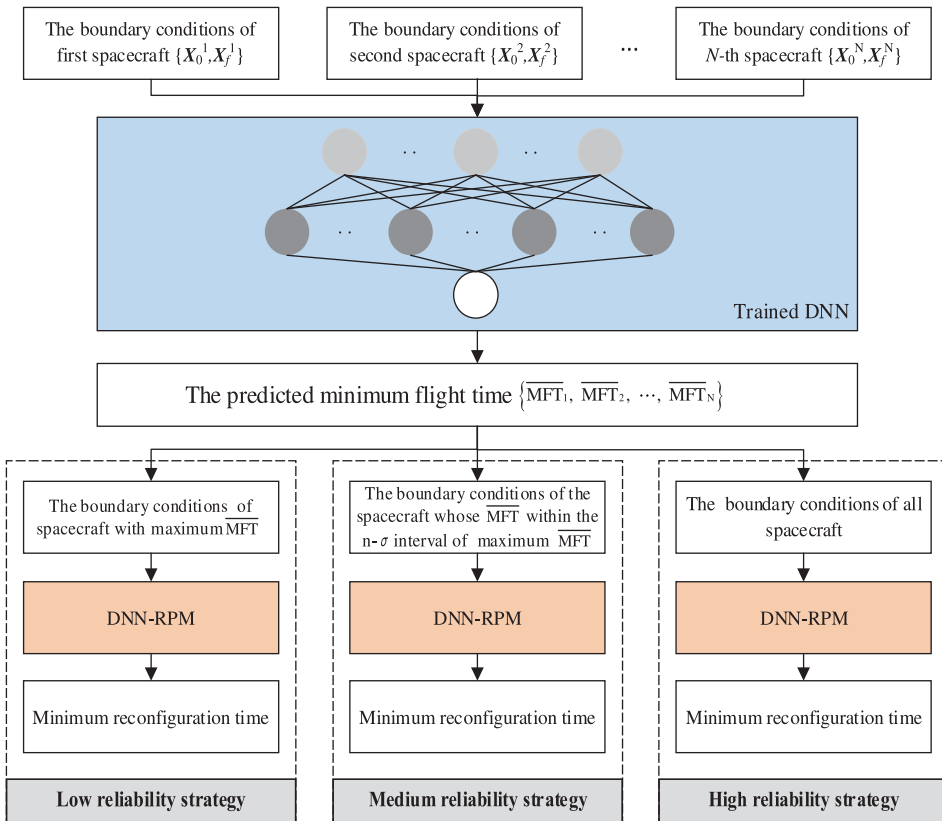
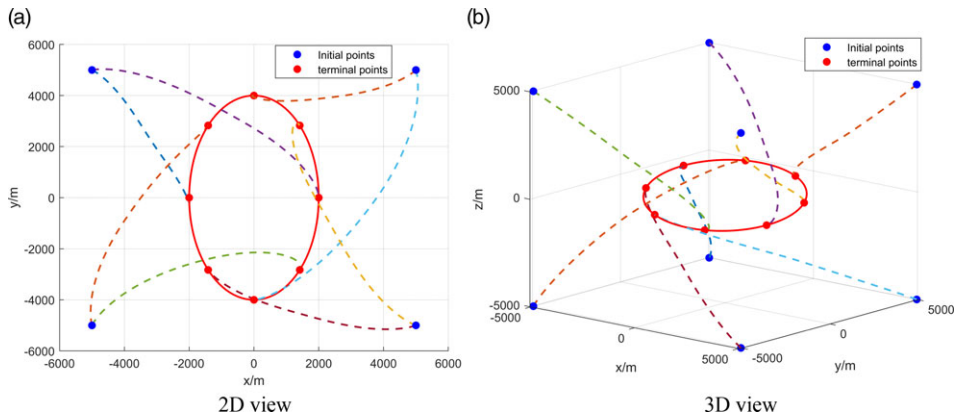


Figure 16. The diagram of low, medium and high reliability strategy.

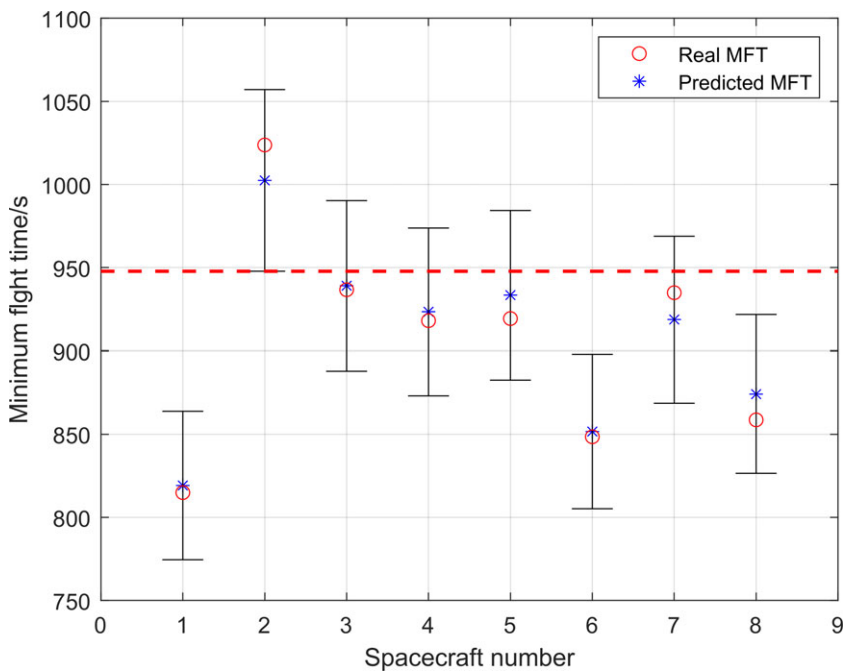
The medium reliability strategy uses the  $n\text{-}\sigma$  interval as a filter, thereby reducing call numbers of the underlying MFT optimisation methods. Taking  $3\sigma$  interval for example, as shown in Fig. 18, red and blue marks represent the real and predicted MFTs of each spacecraft, and the black intervals are the corresponding  $3\sigma$  intervals. The lower boundary of the  $3\sigma$  interval of satellite 2 is larger than the upper boundary of spacecraft 1, 6 and 8. That is to say, the probability that the MFT of satellite 2 is larger than these spacecraft is 99.7%. Therefore, the process of calculating the accurate value of these spacecraft’s MFT can be omitted, and only need to call the DNN-RPM 5 times to calculate real MFTs of

**Table 6.** The performance metrics of different strategies

Strategy	Minimum reconfiguration time (s)	Computing time (s)
HSM	1,023.73	129.96
High reliability	1,023.72	50.44
Medium reliability( $3\sigma$ )	1,023.72	30.34
Medium reliability( $2\sigma$ )	1,023.72	19.53
Medium reliability( $1\sigma$ )	1,023.72	5.69
Low reliability	1,023.72	5.69



**Figure 17.** The manoeuvring trajectories of swarm reconfiguration.



**Figure 18.** The predicted and real MFT of each spacecraft.

other spacecraft. Finally, the low reliability strategy only needs to call DNN-RMS once for calculating the MFT of the slowest spacecraft 2. When  $n$  is 1, 2 and 3, the number of calls of DNN-RPM is 1, 3 and 5, and the corresponding calculation time is 5.69, 19.53 and 30.34s.

In summary, the high reliability strategy can be used for tasks with high precision requirements, and the probability of obtaining optimal solution is 100%. The low reliability strategy is suitable for tasks with high timeliness requirements, so as to determine the minimum reconfiguration time as quickly as possible. As a compromise method, the medium reliability strategy can flexibly adjust the computing efficiency according to the reliability requirements.

It's worth noting that the trajectories presented in Fig. 17 may not be the final feasible trajectories because the collision avoidance constraints are neglected during programming, but the minimum reconfiguration time obtained by those strategies is still reliable. Since the minimum reconfiguration time of is determined by the slowest satellite, the trajectory of the slowest satellite can be regarded as a reference trajectory, and then the collision avoidance constraints can be convexified and solved by sequential convex programming proposed in Ref. [8].

## 6.0 Conclusion

Concentrating on time-optimal trajectory planning for spacecraft rendezvous, this paper proposes a rapid trajectory planning method based on convex optimisation and deep neural network. The time-optimal control problem is transformed into convex form, and the thrust properties are analysed theoretically and validated numerically. The convexification technique can be extended to other time-optimal control problems. Compared with nonlinear optimisation method and traditional root-finding algorithm, the proposed DNN-RPM is capable of improving computational efficiency while ensuring algorithm robustness in generating time-optimal trajectory. For NMC orbit injection problem and swarm reconfiguration, the trained DNN can predict MFT with high precision, thus evidently reducing iteration steps and improving calculation efficiency. Future work will focus on improving the generalisation ability and prediction accuracy of DNN.

**Competing interests.** The author(s) declared no potential competing interests concerning the research, authorship and publication of this paper.

## References

- [1] Zhuang, Y. and Huang, H. Time-optimal trajectory planning for underactuated spacecraft using a hybrid particle swarm optimization algorithm, *Acta Astronaut.*, 2014, **94**, (2), pp 690–698.
- [2] Leomanni, M., Quartullo, R., Bianchini, G., Garulli, A. and Giannitrapani, A. Variable-horizon guidance for autonomous rendezvous and docking to a tumbling target, *J. Guid. Control. Dyn.*, 2022, **45**, (5), pp 846–858. <https://doi.org/10.2514/1.G006340>
- [3] Jewison, C. and Miller, D.W. Probabilistic trajectory optimization under uncertain path constraints for close proximity operations, *J. Guid. Control Dyn.*, 2018, **41**, (9), pp 1843–1858. <https://doi.org/10.2514/1.G003152>
- [4] Boge, T. and Benninghoff, H. Rendezvous simulation for on-orbit servicing missions using advanced robotic technology, *IFAC Proc.*, 2013, **46**, (19), pp 155–160. <https://doi.org/10.3182/20130902-5-DE-2040.00093>
- [5] Bandyopadhyay, S., Foust, R., Subramanian, G.P. and Hadaegh, F.Y. Review of formation flying and constellation missions using nanosatellites, *J. Spacecr. Rockets*, 2016, **53**, (3), pp 567–579. <https://doi.org/10.2514/1.A33291>
- [6] Scala, F., Gaias, G., Colombo, C. and Martín, M. Design of optimal low-thrust manoeuvres for remote sensing multi-satellite formation flying in low Earth orbit, *Adv. Sp. Res.*, 2021, **68**, (11), pp 4359–4378. <https://doi.org/10.1016/j.asr.2021.09.030>
- [7] Betts, J.T. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, Philadelphia, USA, 2010.
- [8] Morgan, D., Chung, S. and Hadaegh, F.Y. Model predictive control of swarms of spacecraft using sequential convex programming, *J. Guid. Control Dyn.*, 2014, **37**, (6), pp 1725–1740. <https://doi.org/10.2514/1.G000218>
- [9] Chai, R. A review of optimization techniques in spacecraft flight trajectory design, *Prog. Aerosp. Sci.*, 2019, **109**, p 100543. <https://doi.org/10.1016/j.paerosci.2019.05.003>
- [10] Wu, B., Wang, D., Poh, E. and Xu, G. Nonlinear optimization of low-thrust trajectory for satellite formation: Legendre pseudospectral approach, *J. Guid. Control. Dyn.*, 2009, **32**, (4), pp 1371–1381. <https://doi.org/10.2514/1.37675>
- [11] Li, J. Fuel-optimal low-thrust formation reconfiguration via Radau pseudospectral method, *Adv. Sp. Res.*, 2016, **58**, (1), pp 1–16. <https://doi.org/10.1016/j.asr.2016.04.005>

- [12] Mauro, G., Spiller, D., Bevilacqua, R. and Curti, F. Optimal continuous maneuvers for satellite formation reconfiguration in J2-perturbed orbits, Space Flight Mechanics Meeting, 2018. <https://doi.org/10.2514/6.2018-0216>
- [13] Cheng, L., Wen, H. and Jin, D. Reconfiguration control of satellite formation using online quasi-linearization iteration and symplectic discretization, *Aerosp. Sci. Technol.*, 2020, **107**, p 106348. <https://doi.org/10.1016/j.ast.2020.106348>
- [14] Peng, H. and Jiang, X. Nonlinear receding horizon guidance for spacecraft formation reconfiguration on libration point orbits using a symplectic numerical method, *ISA Trans.*, 2016, **60**, pp 38–52. <https://doi.org/10.1016/j.isatra.2015.10.015>
- [15] Kayama, Y., Howell, K.C., Bando, M. and Hokamoto, S. Low-thrust trajectory design with successive convex optimization for libration point orbits, *J. Guid. Control. Dyn.*, 2022, **45**, (4), pp 1–15. <https://doi.org/10.2514/1.G005916>
- [16] Xie, L., He, R., Zhang, H. and Tang, G. Oscillation phenomenon in trust-region-based sequential convex programming for the nonlinear trajectory planning problem, *IEEE Trans. Aerosp. Electron. Syst.*, 2022, **58**, (4), pp 3337–3352. <https://doi.org/10.1109/TAES.2022.3153761>
- [17] Morelli, A., Hofmann, C. and Topputo, F. Robust low-thrust trajectory optimization using convex programming and a homotopic approach, *IEEE Trans. Aerosp. Electron. Syst.*, 2022, **58**, (3), pp 2103–2116. <https://doi.org/10.1109/TAES.2021.3128869>
- [18] Han, H., Qiao, D., Chen, H. and Li, X. Rapid planning for aerocapture trajectory via convex optimization, *Aerosp. Sci. Technol.*, 2019, **84**, pp 763–775. <https://doi.org/10.1016/j.ast.2018.11.009>
- [19] Grant, M. and Boyd, S. CVX: Matlab software for disciplined convex programming, 2014. <http://cvxr.com/cvx>
- [20] Yang, H., Bai, X. and Baoyin, H. Rapid generation of time-optimal trajectories for asteroid landing via convex optimization, *J. Guid. Control. Dyn.*, 2017, **40**, (3), pp 628–641. <https://doi.org/10.2514/1.G002170>
- [21] Cheng, L., Li, H., Wang, Z. and Jiang, F. Fast solution continuation of time-optimal asteroid landing trajectories using deep neural networks, *Acta Astronaut.*, 2020, **167**, pp 63–72.
- [22] Viavattene, G. and Ceriotti, M. Artificial neural networks for multiple NEA rendezvous missions with continuous thrust, *J. Spacecraft Rockets*, 2021, **59**, (2), pp 574–586.
- [23] Wang, J., Wu, Y., Liu, M., Yang, M. and Liang, H. A real-time trajectory optimization method for hypersonic vehicles based on a deep neural network, *Aerospace*, 2022, **9**, (4), pp 188–205.
- [24] Dai, P., Feng, D., Feng, W., Cui, J. and Zhang, L. Entry trajectory optimization for hypersonic vehicles based on convex programming and neural network, *Aerospace Sci. Technol.*, 2023, **137**, p 108259.
- [25] Blackmore, L., Açikmeşe, B. and Scharf, D. Minimum-landing-error powered-descent guidance for mars landing using convex optimization, *J. Guid. Control. Dyn.*, 2010, **33**, (4), pp 1161–1171. <https://doi.org/10.2514/1.47202>
- [26] Press, W., Teukolsky, S., Vetterling, M. and Flannery, B. *Numerical Recipes: The Art of Scientific Computing*, Cambridge, 2007, New York, USA.
- [27] Zhang, R., Cai, W., Yang, L. and Si, C. Online collision avoidance trajectory planning for spacecraft proximity operations with uncertain obstacle, *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.*, 2021, **236**, (11), pp 2254–2270. <https://doi.org/10.1177/09544100211056164>
- [28] Lu, P., Lewis, A., Adams, R., DeVore, M.D. and Petersen, C.D. Finite-thrust natural-motion circumnavigation injection by convex optimization, *J. Guid. Control. Dyn.*, 2022, **45**, (3), pp 453–467. <https://doi.org/10.2514/1.G006123>