

Research Article

Real-Time Optimal Approach and Capture of ENVISAT Based on Neural Networks

Hongjue Li ^{1,2} Yunfeng Dong ^{1,2} and Peiyun Li^{1,2}

¹School of Astronautics, Beihang University, Beijing 100191, China

²Key Laboratory of Spacecraft Design Optimization and Dynamic Simulation Technologies, Ministry of Education, Beijing 100191, China

Correspondence should be addressed to Yunfeng Dong; sinosat@buaa.edu.cn

Received 16 October 2019; Revised 8 December 2019; Accepted 11 January 2020; Published 30 January 2020

Academic Editor: James D. Biggs

Copyright © 2020 Hongjue Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A neural network-based controller is developed to enable a chaser spacecraft to approach and capture a disabled Environmental Satellite (ENVISAT). This task is conventionally tackled by framing it as an optimal control problem. However, the optimization of such a problem is computationally expensive and not suitable for onboard implementation. In this work, a learning-based approach is used to rapidly generate the control outputs of the controller based on a series of training samples. These training samples are generated by solving multiple optimal control problems with successive iterations. Then, Radial Basis Function (RBF) neural networks are designed to mimic this optimal control strategy from the generated data. Compared with a traditional controller, the neural network controller is able to generate real-time high-quality control policies by simply passing the input through the feedforward neural network.

1. Introduction

ENVISAT is one of the largest satellites launched by the European Space Agency (ESA) which lost communication with the ground station in April 2012 [1]. A collision between ENVISAT and an object of mass 10 kg could produce an extremely large cloud of debris, initiating a self-sustaining chain reaction of collisions and fragmentation with the production of new debris, a phenomenon known as the Kessler syndrome [2].

The capture of ENVISAT is a complex space mission complicated by its uncertain rotation and the coupling motion between the chaser spacecraft and its manipulators. Many efforts were made by researchers to determine the attitude of ENVISAT. Sommer et al. [3] estimated the rotation vector and rotation speed of ENVISAT through ISAR images from 2011 to 2017, and he indicated that the rotational axis is not fixed and ENVISAT performs a nutation. Koshkin et al. [4] determined the spin motion of ENVISAT using photometry and provided an estimation formula to calculate the rotation period. Pittet et al. [5] suggested that ENVISAT per-

formed no precession and a possible but limited nutation. Theoretically, these ground observations pave the way of estimating the rotation of ENVISAT by multiple equations. Unfortunately, in practice, there are still some inconsistencies within these observations that may suffer from different measurement information, measurement equipment, and target motion data. Such inaccuracy affects the performance of the capture mission.

The capture mission is usually divided into two stages, i.e., the approach stage and the capture stage. In the approach stage, only 6-Degree-Of-Freedom (DOF) motion planning and control are considered since the manipulators are locked. In general, motion planning can be executed before the implementation of the tracking control. Boyarko et al. [6] employed Gauss Pseudospectral Method (GPM) to plan 6-DOF time and fuel optimal motion trajectories. Chu et al. [7] further considered an obstacle avoidance constraint during GPM planning. However, [6, 7] require accurate measurement of three-dimensional orbit position and relative distance information. In contrast, Zhang et al. [8] introduced an angles-only offline trajectory planning method which is

tracked by a nonlinear model predictive control (NMPC). Offline planning has the advantage to optimize trajectories under various performance indices and constraints. However, onboard implementation is not efficient since these algorithms require accurate measurements and high computational expense. Lee and Vukovich [9] established a robust adaptive terminal sliding mode control scheme on the Lie group $SE(3)$ that ensures the finite-time convergence of trajectory tracking errors. Sun and Zheng [10] extended the adaptive control to consider thrust misalignment in spacecraft rendezvous and proximity mission. Zhang et al. [11] further applied an adaptive finite-time fast terminal sliding mode controller to achieve spacecraft formation flying control. These approaches can obtain finite-time convergence control in the presence of model uncertainties and external disturbances.

New approaches have been developed that can yield rapid or even real-time, near-optimal, trajectory generation by taking simplifications [12, 13] or approximations [14]. Other methods include planar optimal energy planning [15], inverse methods [16], modified Rapidly exploring Random Tree (RRT*) [17], modified Fast Marching Tree (FMT*) [18], convex planning [19], artificial potential field-based guidance [20], and polynomial planning [21]. Convex planning is suitable for problems subject to convex constraints such as descent and landing [19], unmanned path planning [22], and rendezvous and proximity operations [23, 24]. On the contrary, time-varying nonconvex motion constraint can be considered by combining artificial potential field with backstepping methodology [20]. Inverse dynamics based polynomial planning methods consisted of parameterizing nominal trajectories via high-order polynomials, converting the original optimal control problem into a nonlinear programming problem [21, 25]. These studies mainly focused on the rapid guidance of the chaser body, ending with final conditions suitable for docking or capture, while neglecting the dynamic coupling between the manipulator and the base in the capture stage. The orbit [26] and/or attitude [27] of the space robots are generally assumed to be uncontrolled so that the focus is placed on the motion planning and control of manipulators [26–28]; real-time near-optimal [29] or optimal [30] manipulator motion trajectories can be obtained.

Significant advancements have been made with regard to both relative motion control and manipulator motion control; however, the reconciliation between the two fields remains to be solved. The division of the mission into approach stage and capture stage can reduce the complexity of the problem [15], but the sum of the two optimal results of the two stages is not necessarily the optimal solution of the whole mission. Considering the highly dynamic nature of the capture mission, the narrow capture window, and the limited energy, the objective is to find optimal trajectories and controls in real-time, during the whole approach and capture procedure. Benefiting from machine learning technology [31, 32], coordination between the optimal control and real-time control becomes possible. Izzo et al. [33] performed a survey on the artificial intelligence trends in spacecraft guidance dynamics and control, which involved a review of the intersection of the guidance and control with evolutionary optimization, tree searches, and machine learn-

ing. Sánchez-Sánchez and Izzo [34] applied deep learning methods to represent the optimal control action during a pinpoint landing. Furthermore, Izzo et al. [35] investigated the use of machine learning and evolutionary techniques in interplanetary trajectory design. Biggs and Fournier [36] trained a multilayer perceptron (MLP) to achieve real-time optimal attitude control of a 12U CubeSat. All these inspiring works attempted to train deep neural networks to learn from traditional optimal control solutions and later directly provide control commands by inputting the measured variables or even images.

Offline solutions of optimal control problems require heavy computational costs to solve an iterative optimization that restricts their applications in real time. Meanwhile, online guidance and control only consider designing control policy in a single scenario, e.g., either 6-DOF guidance control or manipulator motion control. In this work, a more comprehensive implementation of the dynamic system that covers the spacecraft, the manipulators, and their inherent interactions is considered. While the system is more complicated, they can be efficiently solved via our proposed neural network-based approach. Three neural networks are trained to mimic the optimal control policy in a data-driven manner. Then, the optimal relative motion guidance and control for 6-DOF relative motion spacecraft can be implemented in real-time through neural network inference. The main contributions are stated as follows:

- (i) Machine learning technology has been successfully applied to solving complicated optimal control problems. This problem involves considerably more complicated dynamics, DOFs of state variables and control variables compared with landing problems or interplanetary transfer problems
- (ii) Multiple neural networks are jointly designed to collaboratively learn optimal solutions in a synchronized framework. It is difficult to train a single network with a small number of samples, while the generation of huge optimal control solutions as training samples is costly. Simulation results indicate that multiple networks perform well through carefully designed inputs and outputs even when the number of samples are small
- (iii) The proposed control strategy exhibits comparable capture performance as tracking of optimal solutions in ideal cases and can be implemented in real time

The remainder of this article is organized as follows. Section 2 is devoted to the problem formulation, where the motion dynamic model of the approach and capture mission is presented. In Section 3, control strategies applied to generate optimal relative motion control and manipulator motion control solutions are presented. In Section 4, a series of Radial Basis Function (RBF) neural networks are designed and their inputs and outputs are discussed. As an illustration, numerical simulations are carried out in Section 5. Finally, brief concluding remarks in Section 6 end this study.

2. Problem Formulation

According to the measurement of Kucharski et al. [1], ENVISAT is presently flying in an approximate sun-synchronous orbit with a side rotating attitude. According to the ESA, the main body of ENVISAT can be regarded as a $5\text{ m} \times 10\text{ m} \times 5\text{ m}$ cube rotating approximately around its z axis at an angular velocity of $0.8631^\circ/\text{s}$, as obtained by the approximate calculation of the formula proposed by Koshkin et al. [4]. The $+z$ surface of this body is full of sensors and devices, the $-z$ surface is obscured by antennas, the $\pm x$ surfaces are obscured by the Advanced Synthetic Aperture Radar (ASAR), and the $-y$ surface is obscured by the solar array of the satellite. Consequently, only the $+y$ surface is suitable for capture. A chaser spacecraft (hereinafter referred to as the chaser) installed with two 7-DOF manipulators is designed to approach and capture the ENVISAT. The chaser is a small spacecraft launched from a mother spacecraft, which reaches up to a dozen meters from the ENVISAT after a Lambert transfer. Subsequently, the spacecraft starts to approach and capture the ENVISAT, which is the focus of this paper.

The overall mission is depicted in Figure 1. The inertial coordinate system is termed S_i ($Ox_i y_i z_i$). The centroid orbital coordinate system is termed S_o ($Ox_o y_o z_o$). The x axis of this system is along the flight velocity direction, z axis points to the earth, y axis can be determined by right-hand rule, and origin is at the centroid of the ENVISAT. The body coordinate system of the ENVISAT is termed S_{tb} ($Ox_{tb} y_{tb} z_{tb}$), and the body coordinate system of the chaser is termed S_{cb} ($Ox_{cb} y_{cb} z_{cb}$). The x axes of these systems are along the spacecraft origin forward directions, and their z axes point to the earth when both the spacecraft are in the normal earth pointing flight mode.

2.1. Dynamical Equations. A series of equations are required to describe the motion regulations of the ENVISAT, chaser, and manipulators, namely, the orbit kinematics equations, orbit dynamics equations [37], attitude kinematics equations, attitude dynamics equations [38], multibody kinetics equations, and multibody dynamics equations [39].

The orbit and attitude motion regulation of the target can be expressed as

$$\begin{aligned} \dot{\mathbf{r}}_{\text{tar}} &= \mathbf{v}_{\text{tar}}, \\ \dot{\mathbf{v}}_{\text{tar}} &= -\frac{\mu}{r^3} \mathbf{r}_{\text{tar}}, \\ \dot{\mathbf{q}}_{\text{tar}} &= \frac{1}{2} \boldsymbol{\Omega}(\boldsymbol{\omega}_{\text{tar}}) \mathbf{q}_{\text{tar}}, \\ \mathbf{I}_{\text{tar}} \dot{\boldsymbol{\omega}}_{\text{tar}} + \boldsymbol{\omega}_{\text{tar}} \times \mathbf{I}_{\text{tar}} \boldsymbol{\omega}_{\text{tar}} &= 0, \end{aligned} \quad (1)$$

where the subscription ‘‘tar’’ refers to ‘‘target,’’ $\mathbf{r}_{\text{tar}} = [r_{\text{tar}x}, r_{\text{tar}y}, r_{\text{tar}z}]$ is the 3-axis position vector of the ENVISAT, $\mathbf{v}_{\text{tar}} = [v_{\text{tar}x}, v_{\text{tar}y}, v_{\text{tar}z}]$ is the 3-axis velocity vector of the ENVISAT, \mathbf{I}_{tar} is the moment of inertia of the target, $\mathbf{q}_{\text{tar}} = [q_{\text{tar}0}, q_{\text{tar}1}, q_{\text{tar}2}, q_{\text{tar}3}]$ is the quaternion of the ENVISAT,

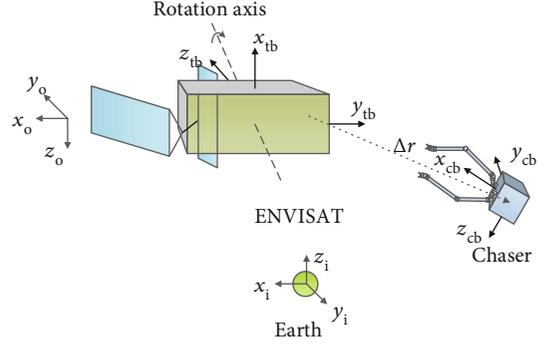


FIGURE 1: Schematic diagram of the approach and capture mission. The chaser spacecraft is equipped with two 7-DOF manipulators.

$\boldsymbol{\omega}_{\text{tar}} = [\omega_{\text{tar}x}, \omega_{\text{tar}y}, \omega_{\text{tar}z}]$ is the 3-axis angular velocity of the ENVISAT, and $\boldsymbol{\Omega}(\boldsymbol{\omega}_{\text{tar}})$ can be expressed as

$$\boldsymbol{\Omega}(\boldsymbol{\omega}_{\text{tar}}) = \begin{bmatrix} 0 & \omega_{\text{tar}z} & -\omega_{\text{tar}y} & \omega_{\text{tar}x} \\ -\omega_{\text{tar}z} & 0 & \omega_{\text{tar}x} & \omega_{\text{tar}y} \\ \omega_{\text{tar}y} & -\omega_{\text{tar}x} & 0 & \omega_{\text{tar}z} \\ -\omega_{\text{tar}x} & -\omega_{\text{tar}y} & -\omega_{\text{tar}z} & 0 \end{bmatrix}. \quad (2)$$

The motion regulation of the chaser with its manipulators can be expressed as

$$\mathbf{H}(\boldsymbol{\Theta}) \ddot{\boldsymbol{\Theta}} + \mathbf{C}(\dot{\boldsymbol{\Theta}}, \boldsymbol{\Theta}) = \mathbf{U}, \quad (3)$$

where $\boldsymbol{\Theta} = [\mathbf{r}_{\text{cha}}, \mathbf{e}_{\text{cha}}, \boldsymbol{\theta}]$ is the generalized coordinate of the chaser including its 3-axis position vector $\mathbf{r}_{\text{cha}} = [r_{\text{cha}x}, r_{\text{cha}y}, r_{\text{cha}z}]$, 3-axis Euler angle $\mathbf{e}_{\text{cha}} = [e_{\text{cha}x}, e_{\text{cha}y}, e_{\text{cha}z}]$ (which is replaced by chaser’s quaternion $\mathbf{q}_{\text{cha}} = [q_{\text{cha}0}, q_{\text{cha}1}, q_{\text{cha}2}, q_{\text{cha}3}]$ to avoid ambiguity in Equation (5)), and the joint angles of the two 7-DOF manipulators $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_{14}]$; $\mathbf{H}(\boldsymbol{\Theta})$ is a symmetrically positive definite inertia matrix of the chaser with manipulators; $\mathbf{C}(\dot{\boldsymbol{\Theta}}, \boldsymbol{\Theta})$ is a nonlinear matrix including Coriolis and centrifugal forces; and $\mathbf{U} = [\mathbf{F}, \mathbf{M}, \mathbf{T}]^T$ is the control variables for the chaser including its 3-axis control forces $\mathbf{F} = [F_x, F_y, F_z]$, 3-axis control moment $\mathbf{M} = [M_x, M_y, M_z]$, and the 14 joint torques $\mathbf{T} = [T_1, T_2, \dots, T_{14}]$. The subscription ‘‘cha’’ refers to ‘‘chaser.’’ Detailed definition of $\mathbf{H}(\boldsymbol{\Theta})$ and $\mathbf{C}(\dot{\boldsymbol{\Theta}}, \boldsymbol{\Theta})$ can be seen in [40] and is not discussed here. The dynamic coupling of the chaser spacecraft and the manipulators can be more obviously seen if Equation (3) is written in matrix form as follows:

$$\begin{bmatrix} ME & M\tilde{\mathbf{r}}_{0g}^T & \mathbf{J}_{T\omega} \\ M\tilde{\mathbf{r}}_{0g} & \mathbf{H}_\omega & \mathbf{H}_{\omega\phi} \\ \mathbf{J}_{T\omega}^T & \mathbf{H}_{\omega\phi} & \mathbf{H}_\phi \end{bmatrix} \begin{bmatrix} \mathbf{r}_{\text{cha}} \\ \mathbf{e}_{\text{cha}} \\ \boldsymbol{\theta} \end{bmatrix} + \mathbf{C}(\dot{\boldsymbol{\Theta}}, \boldsymbol{\Theta}) = \begin{bmatrix} \mathbf{F} \\ \mathbf{M} \\ \mathbf{T} \end{bmatrix}. \quad (4)$$

The control variables \mathbf{F} , \mathbf{M} , and \mathbf{T} are related to all the components of the generalized coordinate. For example, \mathbf{F}

is not only related to the position (\mathbf{r}_{cha}), velocity ($\dot{\mathbf{r}}_{\text{cha}}$ in $\hat{\mathbf{O}}$), and acceleration ($\ddot{\mathbf{r}}_{\text{cha}}$ in $\hat{\mathbf{O}}$) of the chaser spacecraft but also related to \mathbf{e}_{cha} and $\boldsymbol{\theta}$. The coupled nature can be more clearly reflected by applying the Newton-Euler method [41] due to its recursive form; i.e., a forward recursion is performed to propagate the link velocities and accelerations, followed by a backward recursion to propagate forces.

Equations (1) and (3) define a 54-state system of differential equations governing the approach and capture dynamics. Combined into the state vector \mathbf{X} , these states are

$$\mathbf{X} = \left[\mathbf{r}_{\text{cha}}, \mathbf{r}_{\text{tar}}, \mathbf{v}_{\text{cha}}, \mathbf{v}_{\text{tar}}, \mathbf{q}_{\text{cha}}, \mathbf{q}_{\text{tar}}, \boldsymbol{\omega}_{\text{cha}}, \boldsymbol{\omega}_{\text{tar}}, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}} \right]^T, \quad (5)$$

where $\mathbf{v}_{\text{cha}} = [v_{\text{chax}}, v_{\text{chay}}, v_{\text{chaz}}]$ is the 3-axis velocity vector of the chaser, $\boldsymbol{\omega}_{\text{cha}} = [\omega_{\text{chax}}, \omega_{\text{chay}}, \omega_{\text{chaz}}]$ is the 3-axis angular velocity of the chaser, and $\dot{\boldsymbol{\theta}} = [\dot{\theta}_1, \dot{\theta}_2, \dots, \dot{\theta}_{14}]$ denotes the 14 angular velocities of the joints.

The governing dynamics assume 20 controls that can be used by the chaser and the manipulators to achieve the approach and capture mission:

$$\mathbf{U} = [\mathbf{F}, \mathbf{M}, \mathbf{T}]^T, \quad (6)$$

which has been fully defined after Equation (3).

2.2. Performance Indices. Three performance indices are considered to evaluate the control optimality of the chaser, namely, the fuel consumption J_f , time consumption J_t , and impact of the manipulator end effector J_v .

Considering I_{sp} as the specific impulse of the thrusters, J_f can be expressed as

$$J_f = \int_{t_0}^{t_f} \left(\frac{|\mathbf{F}|}{I_{\text{sp}}} \right) dt. \quad (7)$$

Considering t_f as the final capture time, J_t can be expressed as

$$J_t = t_f. \quad (8)$$

J_v can be measured using the x component of the velocity of the manipulator end effector relative to the desired capture point $\Delta \mathbf{v}_{\text{endx}}$ as follows:

$$J_v = \Delta \mathbf{v}_{\text{endx}}, \quad (9)$$

where $\Delta \mathbf{v}_{\text{endx}}$ is the x axis projection of $\Delta \mathbf{v}_{\text{end}}$ in S_{ib} .

By defining the fuel consumption coefficient as k_f , time consumption coefficient as k_t , and impact coefficient as k_v , the overall performance index J can be obtained as follows:

$$J = k_f \cdot J_f + k_t \cdot J_t + k_v \cdot J_v. \quad (10)$$

2.3. Constraints. Two main constraints are considered during the approach and capture procedure, namely, the collision avoidance constraint C_1 and state variable constraint C_2 . The collision avoidance can be taken into consideration by

imposing the following path constraint: the mass center of the chaser spacecraft must remain at a distance larger than the minimum distance (a “keep out” sphere with a radius Δr_{min}) from the mass center of the ENVISAT, which can be defined as

$$C_1 : \|\mathbf{r}_{\text{tar}} - \mathbf{r}_{\text{cha}}\| \geq \Delta r_{\text{min}}. \quad (11)$$

The state variable constraints state that all the state variables and control variables cannot exceed their range of values. By defining \mathbf{X}_{min} and \mathbf{X}_{max} as the minimum and maximum values of state vector \mathbf{X} , respectively, and \mathbf{U}_{min} and \mathbf{U}_{max} as the minimum and maximum values of controls \mathbf{U} , respectively, the state variable constraints can be expressed as

$$C_2 : \mathbf{X}_{\text{min}} \leq \mathbf{X} \leq \mathbf{X}_{\text{max}}, \mathbf{U}_{\text{min}} \leq \mathbf{U} \leq \mathbf{U}_{\text{max}}. \quad (12)$$

3. Real-Time Optimal Control Strategy

It is difficult to directly optimize the entire problem due to the complex coupling motion of the orbit, attitude, and manipulator. Thus, it is necessary to reduce the complexity of the problem step by step. The coupling motion of the orbit and attitude is first considered and optimized using the GPM. The manipulator deployment, point, and capture control are then considered, and a feedforward compensation control algorithm is designed to eliminate the coupling among the manipulators, orbit, and attitude. Finally, an iterative optimization is established to obtain the optimal control strategy of the manipulators, orbit, and attitude.

Let \mathbf{W} be the range of initial values of the state variables. A set of initial values \mathbf{X}_0 can be randomly selected in \mathbf{W} . Let Φ be a neighborhood sufficiently close to \mathbf{X}_0 . A series of value sets $\{\mathbf{X}_i\}$ can be randomly selected in Φ , and each value set is known as a neighborhood member. For each neighborhood member, the optimal control sequence $\mathbf{U}(t)$ and state sequence $\mathbf{X}(t)$ can be obtained by using the methods described in the following sections.

3.1. Relative Motion Control. When the manipulators are locked, only the orbit and attitude motions of the chaser are needed to be considered. The state variables and control variables can be written as

$$\begin{aligned} \mathbf{x} &= [\Delta \mathbf{r}, \Delta \mathbf{v}, \mathbf{q}_{\text{cha}}, \mathbf{q}_{\text{tar}}, \boldsymbol{\omega}_{\text{cha}}, \boldsymbol{\omega}_{\text{tar}}]^T, \\ \mathbf{u} &= [\mathbf{F}, \mathbf{M}]^T, \end{aligned} \quad (13)$$

where $\Delta \mathbf{r}$ and $\Delta \mathbf{v}$ denote the relative position and velocity vectors between the chaser and the ENVISAT, respectively.

The relative orbit dynamics of the chaser and ENVISAT in the orbit frame centered at ENVISAT can be described using the Hill–Clohessy–Wiltshire (C–W) equations [42]. By replacing the orbit dynamics equations with the C–W equations and removing the multibody kinetic and dynamic

TABLE 1: Neighborhood.

Variable	Value range
Δr_x (m)	[-0.5, 0.5]
Δr_y (m)	[14.0, 16.0]
Δr_z (m)	[-0.5, 0.5]
$\Delta v_x \Delta v_y \Delta v_z$ (m·s ⁻¹)	0.0
ω_{tarx} (rad·s ⁻¹)	[0.0, 0.0349]
ω_{tary} (rad·s ⁻¹)	0.0
ω_{tarz} (rad·s ⁻¹)	0.1484
$\omega_{chax} \omega_{chay} \omega_{chaz}$ (rad·s ⁻¹)	0.0
$e_{tarx} e_{tary} e_{tarz}$ (rad)	0.0
$e_{chax} e_{chay} e_{chaz}$ (rad)	0.0

equations, a series of explicit differential equations to govern the law of motion of the approaching stage can be obtained as follows:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}). \quad (14)$$

Since the manipulators are not considered, the value of k_v in the overall performance index J is zero. Thus, the approach mission can be transferred into an optimal control problem. That is, for a system described by Equation (14), the control \mathbf{u} is calculated to transfer the state vector \mathbf{x} from the initial state \mathbf{x}_i to the terminal state \mathbf{x}_f under constraints C_1 and C_2 while optimizing the performance index J .

In the optimal control problem, besides path constraints C_1 and C_2 , certain extra constraints at the final time t_f (often referred to as the terminal constraints) are needed to define \mathbf{x}_f and make the optimization problem solvable. The terminal constraints include the position constraint T_1 , velocity constraint T_2 , and attitude constraint T_3 as follows:

$$\begin{aligned} T_1 : \|\Delta \mathbf{r}_d\| &= \|\Delta \mathbf{r}(t_f)\| \geq r_{\min}, \\ T_2 : \Delta \mathbf{v}(t_f) &= (\mathbf{L}_{oi} \boldsymbol{\omega}_{cha}(t_f)) \times \Delta \mathbf{r}(t_f) + V, \\ T_3 : \mathbf{L}_{ob} \cdot [1, 0, 0]^T &= \frac{\Delta \mathbf{r}(t_f)}{\|\Delta \mathbf{r}(t_f)\|}, \end{aligned} \quad (15)$$

where \mathbf{L}_{oi} is the rotation matrix between S_i and S_o and V is the extra relative velocity. If $V = 0$, the chaser remains relatively static with the ENVISAT; otherwise, the chaser flies over the ENVISAT at $\Delta \mathbf{r}_d$ with a relative speed of V . \mathbf{L}_{ob} is the rotation matrix between S_{cb} and S_o and is determined by \mathbf{q}_{cha} .

The above optimal control problem can be solved using the well-developed numerical toolkit Gauss Pseudospectral Optimization Software (GPOPS) [43]. The output is the optimal state variable sequence $\mathbf{x}(t)$ and the optimal control variable sequence $\mathbf{u}(t)$, $t = t_0, \dots, t_f$. Boyarko et al. [6] proved that a pseudospectral solver provided a solution remarkably close to the truly optimal one by using the results obtained by applying the Minimum Principle (MP). In their research, the chaser started to approach at a distance of 5 m, and the MP solution returned a value of a minimum control perfor-

mance index $J = 0.11341185$ compared with $J = 0.1133$ obtained by the pseudospectral solver. The GPOPS has also been employed to compute the optimal trajectories and generate the appropriate training set in a lunar landing mission [44]. In this case, the training set was used to train a deep Convolutional Neural Network (CNN) followed by a Recurrent Neural Network connecting with a Long Short-Term Memory (RNN-LSTM) and is effective in the planar landing problem.

Given the size of the ENVISAT, a similar approach scenario is considered, in which the chaser starts at a distance of approximately 15 m from the mass center of the ENVISAT. The ENVISAT has an initial small angular velocity around its x body axis and an angular velocity of 0.1484 rad/s around its z body axis. The minimum safe distance Δr_{\min} is set to be 6.8 m, which means that the chaser approaches the ENVISAT to a distance of 1.8 m since the distance of the mass center of the ENVISAT to its $+y$ surface is 5 m. The detailed ranges of all the initial values are presented in Table 1 to form a neighborhood, from which 388 numbers of neighborhood members (initial conditions) are randomly selected.

Researchers determine the coefficients in the performance index according to the mission requirements. If the chaser is required to approach the ENVISAT as fast as possible, k_f is set as zero to ensure that the optimal control problem becomes a minimum time control problem. On the contrary, if the chaser is required to approach the ENVISAT with the least fuel consumption, the problem becomes a minimum control problem, and k_t is set as zero. Here, k_f is set to be 2/3 and k_t is set to be 1, which means that both the fuel consumption and approach time are considered.

For each neighborhood member, an optimal control sequence and the corresponding optimal trajectory are determined using the GPM. Each trajectory consists of 40 discrete points. The 388 optimal relative orbit trajectories are shown in Figure 2. A sample relative orbit trajectory ($m = 12$) along with the attitude of the chaser is shown in Figure 3.

Note that not all the optimal trajectories and corresponding control sequences are used to train the networks. 26 trajectories of bad quality are eliminated so that only 362 trajectories are used to train the networks.

3.2. Manipulator Control. The manipulator installed on the chaser spacecraft is shown in Figure 4. The base coordinate system is termed S_m (more specifically termed S_{lm}/S_{rm} for the left and right manipulators). The modified Denavit–Hartenberg (D–H) method is used to establish the multibody kinetics of the manipulator, and the D–H parameters are listed in Table 2. The initial joint angles of the folded manipulators are listed in the last column of Table 2.

A common deployment strategy is designed by using the following defined desired joint angular accelerations:

$$\ddot{\theta}_i(t) = \begin{cases} \frac{-\Delta \theta_i}{(\Delta t/2)^2}, & t < \frac{\Delta t}{2}, \\ \frac{\Delta \theta_i}{(\Delta t/2)^2}, & t \geq \frac{\Delta t}{2}, \end{cases} \quad (16)$$

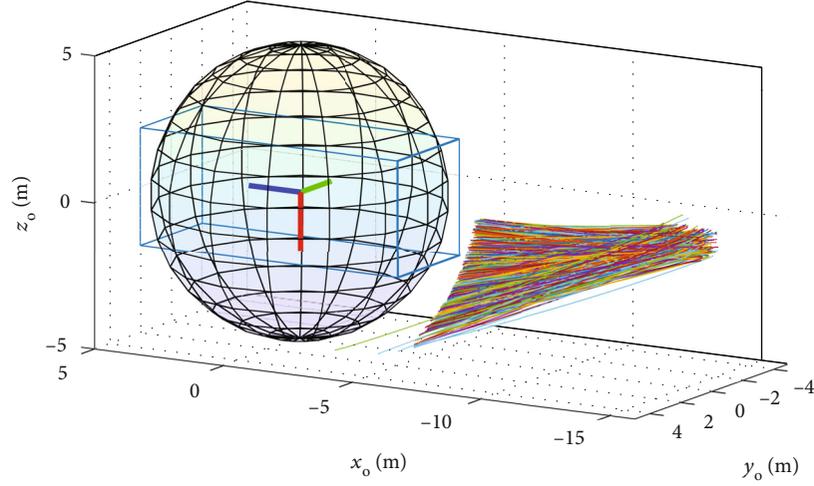


FIGURE 2: Optimal trajectories of all 388 neighborhood members. The radius of the sphere on the left is Δr_{\min} . The blue cuboid on the left is the ENVISAT with an attitude of t_0 . The red, blue, and green lines perpendicular to one another denote the $+x$, $+y$, and $+z$ axes, respectively, of the orbital coordinate system S_0 .

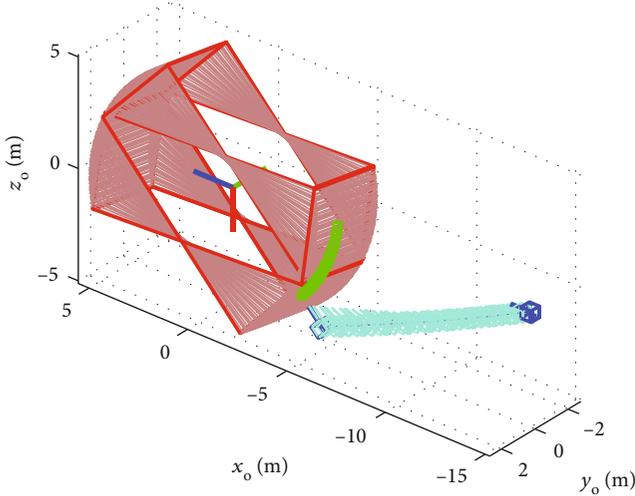


FIGURE 3: Optimal trajectory of the 12th neighborhood member. The light green trajectory is the track of the “subsatellite point” of the chaser on the ENVISAT. The dark green cuboids denote the chaser body.

$i = 1, 2, \dots, 14$, where $\Delta t = t_i - t_e$ is the total deployment time.

By integrating Equation (16), the joint angular velocities can be obtained, and by integrating it again, the joint angles can be obtained. By determining $\theta_i, \dot{\theta}_i, \ddot{\theta}_i, \mathbf{v}_c, \dot{\mathbf{v}}_c, \mathbf{v}_i, \dot{\mathbf{v}}_i, \boldsymbol{\omega}_c, \dot{\boldsymbol{\omega}}_c$, the joint torques \mathbf{T} at any time during the deployment can be calculated by using Equation (3).

After deployment, a Proportional-Differential (PD) control algorithm is designed to control the joints of the manipulators. The calculation of the desired joint angles can be divided into the following 4 steps:

Step 1. Let $\theta_3 = \theta_{3\text{const}}$, and calculate the angles of the other joints as ${}^1\theta_i$.

The desired angle of joints 4, 2, and 1 can be calculated as follows:

$$\theta_4 = \pm \left[\frac{\pi - \cos^{-1} \left(\frac{2l^2 - p_{dx}^2 - p_{dy}^2 - p_{dz}^2}{2l^2} \right)}{2l^2} \right],$$

$$\theta_2 = \begin{cases} \sin^{-1} \left(\frac{p_{dz}}{\sqrt{A^2 + B^2}} \right) - C, \\ \pi - \sin^{-1} \left(\frac{p_{dz}}{\sqrt{A^2 + B^2}} \right) + C, \end{cases} \quad (17)$$

$$\theta_1 = \tan^{-1} \left(\frac{(p_{dx}n_y - p_{dy}n_x) / (p_{dx}m_y - p_{py}m_x)}{(m_xn_y - m_yn_x) / (m_yn_x - m_xn_y)} \right),$$

where A, B, C, m_x, n_x, m_y , and n_y are

$$\begin{cases} A = -l_2 \sin \theta_4 \cos \theta_3, \\ B = l_1 + l_2 \cos \theta_4, \\ C = -\tan^{-1} \left(\frac{B}{A} \right), \\ m_x = -l \sin \theta_4 \sin \theta_3, \\ n_x = l \sin \theta_4 \cos \theta_2 \cos \theta_3 + (l + l \cos \theta_4) \sin \theta_2, \\ m_y = n_x, \\ n_y = l \sin \theta_4 \sin \theta_3. \end{cases} \quad (18)$$

The rotation matrix \mathbf{M}_{04} can be derived from the above-mentioned desired angles of joints 1, 2, 3, and 4. \mathbf{M}_{07} can be obtained according to the desired position and attitude of the manipulator end effector. Thus, \mathbf{M}_{47} (transformation matrix between the coordinate systems of links 4 and 7) can be

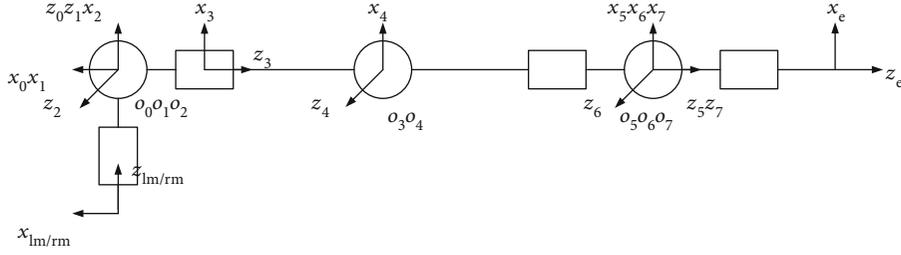


FIGURE 4: Definition of joints and links of the 7-DOF manipulator.

TABLE 2: D-H parameters of the 7-DOF manipulator.

Joint (link) i	α_i ($^\circ$)	a_i (m)	d_i (m)	θ_i ($^\circ$)
1	0	0	0	15.57335
2	-90	0	0	120.40453
3	90	0	$l_2 + l_3$	28.84101
4	-90	0	0	99.4784
5	90	0	$l_4 + l_5$	0
6	-90	0	0	46.70365
7	90	0	0	0

calculated by performing the inverse product of M_{04} and M_{07} as follows:

$$M_{47} = M_{04}^{-1} M_{07} = \begin{bmatrix} g_{11} & g_{12} & g_{13} & d_x \\ g_{21} & g_{22} & g_{23} & d_y \\ g_{31} & g_{32} & g_{33} & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (19)$$

Subsequently, the desired angles of joints 5, 6, and 7 can be calculated using the components of M_{47} as follows:

$$\begin{cases} \theta_5 = \tan^{-1} \left(\frac{g_{33}}{g_{13}} \right), \\ \theta_6 = \cos^{-1} (-g_{23}), \\ \theta_7 = \tan^{-1} \left(\frac{g_{22}}{g_{21}} \right), \end{cases} \quad (20)$$

or

$$\begin{cases} \theta_5 = \tan^{-1} \left(\frac{-g_{33}}{-g_{13}} \right), \\ \theta_6 = \cos^{-1} (-g_{23}), \\ \theta_7 = \tan^{-1} \left(\frac{-g_{22}}{-g_{21}} \right). \end{cases} \quad (21)$$

For the attitude of the end effector of a certain manipulator, 2 groups of values of θ_5 , θ_6 , and θ_7 are calculated using Equations (20) and (21). For the position of the end effector of a certain manipulator, 4 groups of values of θ_1 , θ_2 , θ_3 , and θ_4 are calculated using Equation (17). Together,

8 groups of desired joint angles are available. One group of joint angles is chosen as the desired joint angles to minimize the sum of the rotation angles from the current angles to the desired angles.

Step 2. Let $\theta_3 = \theta_{3\text{const}} + \Delta\theta_3$, and calculate the angles of the other joints again as ${}^2\theta_i$ using Equations (17), (20), and (21).

Step 3. Linearize the equation for the manipulator in intervals $[{}^1\theta_i, {}^2\theta_i]$. The variation of the angle of joint 3 $\Delta\theta_{3\text{best}}$ can be optimized by using linear programming.

Step 4. Let $\theta_3 = \theta_3 + \Delta\theta_{3\text{best}}$, and obtain the other joint angles by solving Equations (17), (20), and (21).

The desired angular velocity of each joint can be calculated by using the Moore-Penrose inverse of the Jacobian matrix $J(\theta)$ as follows:

$$\begin{aligned} \dot{\theta} &= J(\theta)^+ \cdot [v_e, \omega_e]^T, \\ J(\theta)^+ &= J(\theta)^T (J(\theta)J(\theta)^T)^{-1}, \end{aligned} \quad (22)$$

where v_e is the desired velocity of the manipulator end effector and ω_e is the angular velocity of the manipulator end effector.

Since the desired angle and angular velocity of each joint have been calculated, the PD control law is employed to solve for the joint torque as follows:

$$T_i = k_p \cdot \theta_i + k_d \cdot \dot{\theta}_i. \quad (23)$$

Twelve neighborhood members are randomly selected from the 388 neighborhood members. Each member of the manipulator neighborhood is calculated, and all the trajectories of the joint angle varying with time are obtained. Taking joint 2 of the left manipulator as an example, the optimal joint angle trajectories of all the 12 neighborhood members are determined, as shown in Figure 5.

3.3. Feedforward Compensation and Iterative Optimization. A recursive form of the multirigid body dynamic model can be established using the Newton-Euler method as follows:

$$\Gamma_i = h(\theta_i, \dot{\theta}_i, \ddot{\theta}_i, \dot{v}_{\text{cha}}, \dot{v}_{\text{tar}}), \quad i = 1, 2, \dots, 14, \quad (24)$$

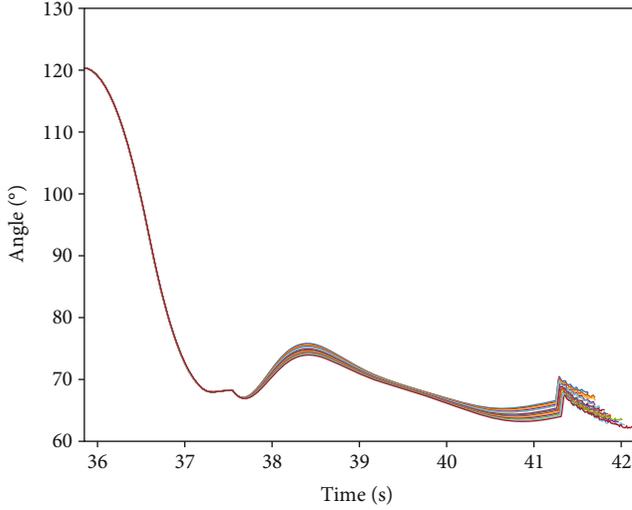


FIGURE 5: Twelve optimal joint angle trajectories for joint 2 of the left manipulator.

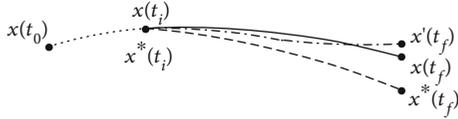


FIGURE 6: Iterative optimization process.

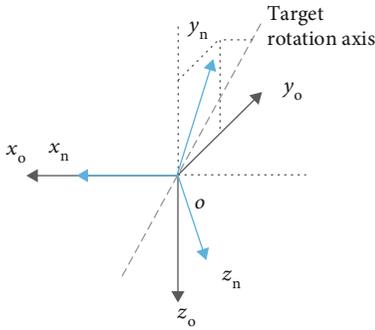


FIGURE 7: Training coordinate system. The subscript “o” denotes the orbital coordinate system S_o , and the subscript “n” denotes the training coordinate system S_n .

where $\Gamma_i = [\mathbf{f}_i, \mathbf{m}_i]^T$ denotes the force and moment applied to each joint, \mathbf{m}_i denotes the transferring torque of each joint, and \mathbf{f}_i denotes the transferring force of each joint.

When $i = 0$, $\Gamma_0 = [\mathbf{f}_0, \mathbf{m}_0]^T$ denotes the transferring force and moment between the manipulator base and the chaser body. By calculating and applying these values inversely as the feedforward to the chaser orbit and attitude control, the control variables of the chaser can be updated as follows:

$$\mathbf{U}(t) = \mathbf{u}(t) + \Gamma_0(t). \quad (25)$$

Since the disturbance is estimated, it cannot be precisely eliminated by using the feedforward compensation using $\mathbf{u}(t)$. The possible influences may include the deviation of

the optimal trajectory, which makes $\mathbf{u}(t)$ not optimal, as shown in Figure 6. The orbit and attitude motion trajectory is assumed as $\mathbf{x}(t_0), \dots, \mathbf{x}(t_i), \dots, \mathbf{x}(t_f)$ (solid line) by assigning the terminal state $\mathbf{x}(t_f)$ when the manipulators do not move. When the manipulators move and finish the capturing, the orbit and attitude motion trajectory is modified to $\mathbf{x}^*(t_0), \dots, \mathbf{x}^*(t_i), \dots, \mathbf{x}^*(t_f)$ (dash line) with the terminal state $\mathbf{x}^*(t_f)$. One or more iterations are performed to update $\mathbf{u}(t)$ to make $\mathbf{x}^*(t_f)$ closer to $\mathbf{x}(t_f)$. Considering that $\Delta \mathbf{x}^*(t_f) = \mathbf{x}^*(t_f) - \mathbf{x}(t_f)$, the iteration stops when $\Delta \mathbf{x}^*(t_f)$ is less than a preset constant value, and the optimal orbit and attitude trajectory $\mathbf{x}'(t_f)$ (dot-dash line) is obtained. The corresponding optimal control sequence is $\mathbf{U}(t) = \mathbf{u}'(t) + \Gamma_0(t)$.

4. Learning the Optimal Control

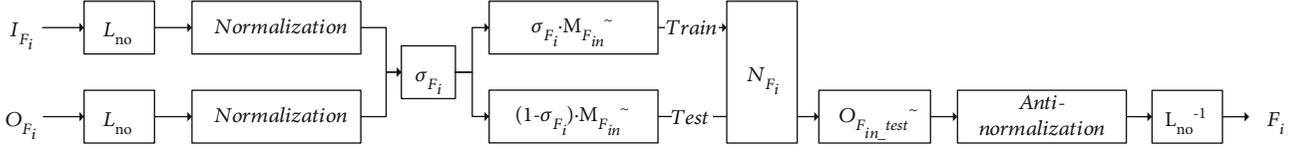
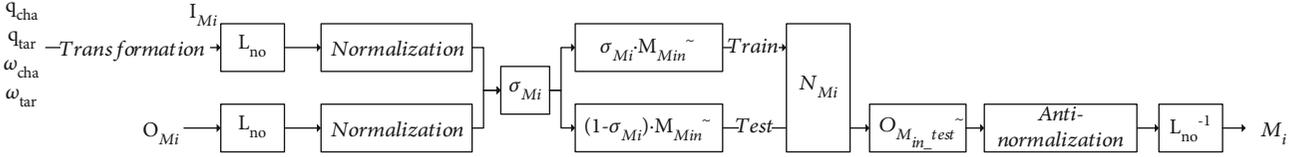
A series of RBF networks are chosen to reduce the optimization time consumption while maintaining the solution to be optimal or at least suboptimal. The network input and output are obtained from the optimal solutions and their initial values of the state variables. By assuming the number of neighborhood members as m and the line number of state sequence \mathbf{X} of each neighborhood member as n , the number of lines of all the neighborhood members can be defined as $m \cdot n$. Considering $\mathbf{x}(t)$ and $\mathbf{u}(t)$ as the t^{th} lines of $\mathbf{X}(t)$ and $\mathbf{U}(t)$, respectively, $\mathbf{x}(t)$ and $\mathbf{u}(t)$ together form a sample, where $\mathbf{x}(t)$ is the input and $\mathbf{u}(t)$ is the output.

It is unrealistic to directly provide all the $\mathbf{x}(t)$ and $\mathbf{u}(t)$ as samples to train a single large neural network, and decoupling the input and output according to the motion characteristics of the chaser is necessary. In this case, the motion of the chaser can be divided into 3 parts: the orbit motion, attitude motion, and manipulator motion. Each part can be further divided according to the dimensions.

4.1. Coordinate Transformation. The optimal trajectories and the corresponding optimal controls form the so called “state-action pairs” and are supposed to be used to train networks. However, the RBF networks cannot be directly trained by using these state-action pairs because the state variables are anisotropic, which means that the same relative trajectory has different relative state variables in different directions. To eliminate the anisotropy, it is necessary to transform the state variables and control variables to a new body-based coordinate system.

Here, a training coordinate system S_n is defined as shown by the blue line in Figure 7. The x axis of S_n coincides with the x axis of S_o . The y axis of S_n is the projection of the rotation axis of the ENVISAT on the yo z plane of S_o . The z axis can be determined using the right-hand rule. The rotation matrix between the training coordinate system and the orbital coordinate system is \mathbf{L}_{no} . By transferring the inputs and outputs to S_n , the anisotropy can be eliminated, and the regulations between the inputs and outputs can be more easily learned.

4.2. Orbit Control Networks. Three RBF neural networks N_{F_x} , N_{F_y} , and N_{F_z} are designed according to the three-axis motion of the chaser orbit. These networks are supposed to provide the three-axis control forces by providing certain inputs.

FIGURE 8: Training process of the orbit control network N_{F_i} ($i = x, y, z$).FIGURE 9: Training process of the attitude control network N_{M_i} ($i = x, y, z$).

Therefore, it is vital to determine the factors that affect the decision to implement the control forces. The instant relative position $\Delta \mathbf{r}(t)$ and relative velocity $\Delta \mathbf{v}(t)$ play a dominant role in the decision, while the terminal relative position $\Delta \mathbf{r}(t_f)$ and relative velocity $\Delta \mathbf{v}(t_f)$ may influence the decision, especially when the chaser is close to the terminal position. The maximum relative velocity Δv_{\max} affects the decision of the acceleration or deceleration. Hence, $\mathbf{r}(t)$, $\mathbf{v}(t)$, $\mathbf{r}(t_f)$, $\mathbf{v}(t_f)$, and Δv_{\max} are selected to form $n \times 5$ matrices \mathbf{I}_{F_i} ($i = x, y, z$) as the input, and \mathbf{F} from $\mathbf{u}(t)$ is selected to form $n \times 1$ matrices \mathbf{O}_{F_i} ($i = x, y, z$) as the output, which can be formulated as

$$\begin{aligned} \mathbf{I}_{F_i} &= [\Delta r_i(t), \Delta v_i(t), \Delta r_i(t_f), \Delta v_i(t_f), \Delta v_{\max}]^T, \quad i = x, y, z, \\ \mathbf{O}_{F_i} &= [F_i(t)]^T, \quad i = x, y, z. \end{aligned} \quad (26)$$

The training procedure for the orbit control networks is shown in Figure 8. First, \mathbf{I}_{F_i} and \mathbf{O}_{F_i} are transferred to the training coordinate system while neglecting the orbital angular velocity. Next, both \mathbf{I}_{F_i} and \mathbf{O}_{F_i} are divided into a training dataset and a testing dataset after normalization. The networks are trained using the training dataset and tested using the testing dataset. The trained networks onboard provide the three-axis forces by inputting the appropriate values from the sensors.

During the normalization procedure, the inputs and outputs are both normalized to the $[-1, 1]$ interval. $\mathbf{M}_{F_{in}}^{~}$ is defined as the overall dataset of the input and output data after normalization, $\sigma_{F_i} \in [0, 1]$ as the test ratio, $j = \sigma_{F_i} \cdot n \cdot m$ as the line number of the training data, and $k = 1 - j$ as the line number of the test data. j lines are randomly selected from $\mathbf{M}_{F_{in}}^{~}$ to form the training dataset $\mathbf{M}_{F_{in_train}}^{~}$. The remaining k lines are used to form the test dataset $\mathbf{M}_{F_{in_test}}^{~}$. The N_{F_i} network can be trained using $\mathbf{M}_{F_{in_train}}^{~}$ and tested using $\mathbf{M}_{F_{in_test}}^{~}$ after setting an appropriate spread speed and training goal.

The output of the trained network when the input is $\mathbf{I}_{F_{in_test}}^{~}$ is $\mathbf{O}_{F_{in}}^{~}$. The deviation between $\mathbf{O}_{F_{in}}^{~}$ and the desired output $\mathbf{O}_{F_{in_test}}^{~}$ is $\Delta \mathbf{O}_{F_{in}}$. For every moment t_k , the

deviation can be written as $\Delta F_{F_{in}}(t_k)$. Mean square error (MSE) is used to evaluate the training error of the network, which can be calculated as

$$\varepsilon_{sdF_i} = \sum_{k=1}^j \Delta F_{F_{in}}^{~}(t_k)^2. \quad (27)$$

By antinormalizing $\mathbf{O}_{F_{in}}^{~}$, the actual output $\mathbf{O}_{F_i} = F_i$ of the network N_{F_i} ($i = x, y, z$) can be obtained.

4.3. Attitude Control Networks. Similarly, the attitude motion can be controlled by three networks N_{M_x} , N_{M_y} , and N_{M_z} . The training process of N_{M_i} ($i = x, y, z$) is shown in Figure 9. The only difference between the orbit control network and the attitude control network is in terms of the input segment, which contains a process to transform \mathbf{q}_{cha} and \mathbf{q}_{tar} to relative quaternions $\Delta \mathbf{q}$ and relative angular velocities $\Delta \boldsymbol{\omega}$.

Furthermore, $\Delta \mathbf{q}$ is transferred to the relative Euler angle $\Delta \mathbf{e}$ so that the dimension of $\mathbf{x}(t)$ can be reduced as follows:

$$\begin{aligned} \Delta \mathbf{e} &= f(\mathbf{q}_{tar}, \mathbf{q}_{cha}) \\ \mathbf{x}(t) &= \frac{\Delta \boldsymbol{\omega} = \boldsymbol{\omega}_{tar} - \boldsymbol{\omega}_{cha}}{\Delta \mathbf{e}} \rightarrow \mathbf{x}'(t) \\ &= [\mathbf{r}(t), \mathbf{v}(t), \Delta \mathbf{e}(t), \Delta \boldsymbol{\omega}(t)]^T. \end{aligned} \quad (28)$$

Similar to that for the orbit control networks, the selection of the input and output elements for the attitude control networks is determined by analyzing the factors affecting the output moments. Here, the elements selected from $\mathbf{x}'(t)$ to form the inputs \mathbf{I}_{M_x} , \mathbf{I}_{M_y} , and \mathbf{I}_{M_z} and the outputs \mathbf{O}_{M_x} , \mathbf{O}_{M_y} , and \mathbf{O}_{M_z} are as follows:

$$\begin{aligned} \mathbf{I}_{M_x} &= [\Delta e_x(t), \Delta \omega_x(t)]^T, \\ \mathbf{I}_{M_y} &= [\Delta r_z(t), \Delta v_z(t), \Delta e_y(t), \Delta \omega_y(t)]^T, \\ \mathbf{I}_{M_z} &= [\Delta r_y(t), \Delta v_y(t), \Delta e_z(t), \Delta \omega_z(t)]^T, \\ \mathbf{O}_{M_i} &= [M_i(t)]^T \quad i = x, y, z. \end{aligned} \quad (29)$$

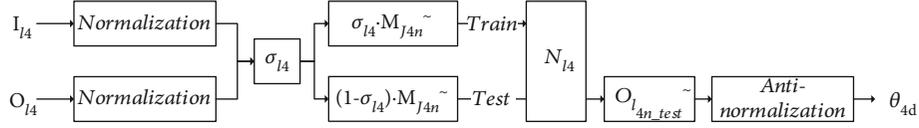
FIGURE 10: Training process of the manipulator control network N_{l_4} .

TABLE 3: Training parameters of the control networks.

Network	Spread speed	Training goal
N_{F_x}	0.1	0.005
N_{F_y}	0.1	0.008
N_{F_z}	0.1	0.012
N_{M_x}	0.1	0.003
N_{M_y}	0.1	0.005
N_{M_z}	0.1	0.003
N_{l_2}	1	0.001
N_{l_4}	1	0.001
N_{r_2}	1	0.001
N_{r_4}	1	0.001

Note that the attitude control is related not only to the relative attitude states, i.e., $\Delta e_i(t)$ and $\Delta \omega_i(t)$, but also to the relative position and velocity states, i.e., $\Delta r_i(t)$ and $\Delta v_i(t)$ in Equation (29), because the orbit and attitude are coupled in relative motion.

The other procedures are similar to those of the orbit control networks and are not repeated here.

4.4. Manipulator Joint Control Networks. Joints 2 and 4 are selected to be controlled by the networks because they exert a considerable influence on the manipulator shape. For a 7-DOF manipulator, once a redundant joint (joint 2 or joint 4) is determined, the other joint angles can be calculated. The left and right manipulator control networks can be designed as N_{l_4} and N_{r_4} , respectively. The training process of N_{l_4} is shown in Figure 10 as an example.

The inputs \mathbf{I}_{l_4} and \mathbf{I}_{r_4} and outputs \mathbf{O}_{l_4} and \mathbf{O}_{r_4} no longer need to be transferred to a certain coordinate system, and they are designed as follows:

$$\mathbf{I}_{l_2} = \mathbf{I}_{l_4} = \mathbf{I}_{r_2} = \mathbf{I}_{r_4} = [r_x(t), r_z(t), v_x(t), v_z(t), \Delta e_x(t), \Delta e_y(t), \Delta e_z(t), \Delta \omega_x(t), \Delta \omega_y(t), \Delta \omega_z(t)]^T,$$

$$\mathbf{O}_{l_2} = [\theta_2(t)]^T,$$

$$\mathbf{O}_{l_4} = [\theta_4(t)]^T,$$

$$\mathbf{O}_{r_2} = [\theta_8(t)]^T,$$

$$\mathbf{O}_{r_4} = [\theta_{10}(t)]^T.$$

(30)

Note that the outputs of the manipulators are the joint angles θ_j , and the joint torques \mathbf{T} can be obtained using Equation (23) after determining $\hat{\theta}_j$.

The other procedures are similar to those of the orbit control networks and are not repeated here.

4.5. Training Process. The training parameters of the orbit, attitude, and manipulator control networks are listed in Table 3. In our investigation, the training performances using the parameters different from those listed in Table 3 either match the training performance using the parameters in the table or exhibit a degradation. The training process of N_{F_x} with 50 iterations is shown in the left part of Figure 8, and the training process of the left manipulator joint 4 control network N_{l_4} with 22 iterations is shown in the right part of Figure 11.

After training, the MSEs of the orbit control networks N_{F_x} , N_{F_y} , and N_{F_z} are 0.499%, 0.788%, and 0.119%, respectively. The MSEs of the attitude control networks N_{M_x} , N_{M_y} , and N_{M_z} are 0.299%, 0.499%, and 0.299%, respectively. The MSEs of the manipulator control networks N_{l_2} , N_{l_4} , N_{r_2} , and N_{r_4} are 0.084%, 0.099%, 0.085%, and 0.096%, respectively.

The trained networks can be completely described by a series of parameters containing the input dimension n_i , output dimension n_o , number of neurons n_n , input layer weight \mathbf{I}_w , output layer weight \mathbf{L}_w , hidden layer weights \mathbf{b}_1 and \mathbf{b}_2 , maximum value $\mathbf{O}_{o \max}$, and minimum value $\mathbf{O}_{o \min}$ of the normalization range. Once the networks are well trained, the network parameters are loaded into the control computer of the chaser to control its orbit, attitude, and manipulators.

5. Results and Discussion

5.1. Simulation Case Design. As discussed above, the ENVISAT can be regarded as a $5 \text{ m} \times 10 \text{ m} \times 5 \text{ m}$ cube by neglecting its solar array and other deployed devices. The geometric center of the ENVISAT, mass center of the ENVISAT, and origin of S_{tb} are considered to coincide for simplicity. According to Gómez and Walker [45], the mass of the ENVISAT is 8211 kg, the moment of inertia is (124801.21, 16979.74, 129180.25) $\text{kg} \cdot \text{m}^2$, and the product of inertia is $0 \text{ kg} \cdot \text{m}^2$. The z axis is the maximum inertia principal axis of the satellite, and ENVISAT is assumed to rotate at an angular velocity of $0.8631^\circ/\text{s}$ [4]. At some point, the instantaneous angular velocity of ENVISAT is assumed to be $(0.15, 0, 0.85)^\circ/\text{s}$ and the corresponding nutation angle is 9.6752° , which satisfies the conclusion of Pittet et al. [5]. The instantaneous attitude of ENVISAT is assumed to be $(0, 78, 86)^\circ$ as if it is “lying” on its side in the orbit. According to Section 2, only the $+y$ surface of the ENVISAT is suitable for capture. Therefore, $(0, 5, -0.2) \text{ m}$ and $(0, 5, 0.2) \text{ m}$ expressed in S_{tb} are

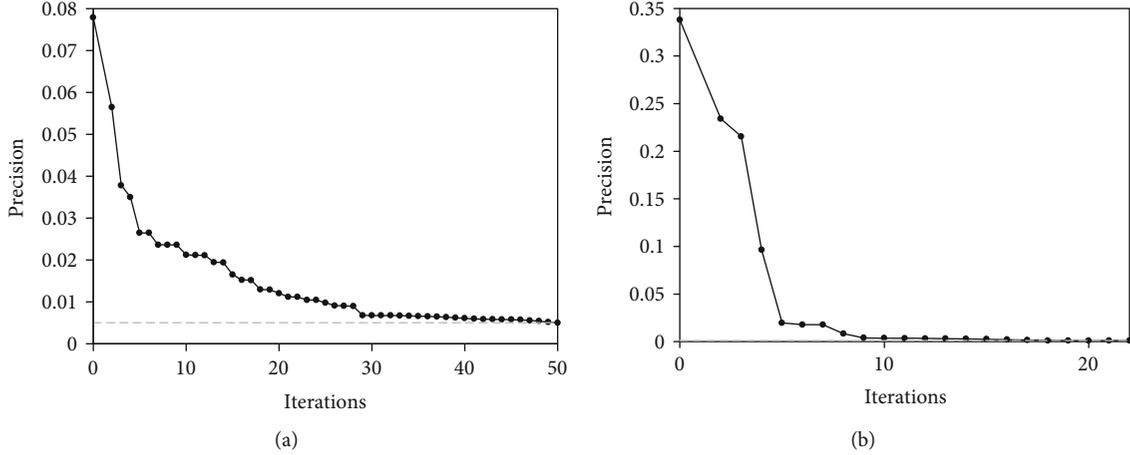


FIGURE 11: Training process for the RBF networks. The gray dashed line denotes the preset training goal. (a) Training process for N_{F_x} . (b) Training process for N_{I_4} .

TABLE 4: Mass and inertia parameters of the manipulator.

Link i	Length (m)	Mass (kg)	Moment of inertia ($\text{kg}\cdot\text{m}^2$)
0	0	0	(0, 0, 0)
1	0.045	0.08	$(6.35, 6.35, 10) \times 10^{-5}$
2	0.075	0.3	$(3.28, 3.28, 3.75) \times 10^{-4}$
3	0.58	2.32	$(6.65, 6.65, 2.90) \times 10^{-2}$
4	0.58	2.32	$(6.65, 6.65, 2.90) \times 10^{-2}$
5	0.075	0.3	$(3.28, 3.28, 3.75) \times 10^{-4}$
6	0.045	0.08	$(6.35, 6.35, 10) \times 10^{-5}$
7	0.1	0.4	$(5.83, 5.83, 5) \times 10^{-4}$

TABLE 5: The 6 designed simulation cases.

Cases	Real ω_t ($^\circ/\text{s}$)	Initial measured ω_t ($^\circ/\text{s}$)	Initial relative \mathbf{r} (m)
1			(0.124, 15.765, -0.417)
2	(0.17, 0.0, 0.85)	(0.17, 0.0, 0.85)	(0.347, 14.871, -0.308)
3			(-0.423, 15.364, -0.124)
4		(0.0, 0.0, 0.85)	
5	(0.15, 0.0, 0.85)	(0.13, 0.02, 0.85)	(-0.218, 14.964, -1.024)
6		(0.17, -0.01, 0.85)	

selected as the two desired capture points on the $+y$ surface of the ENVISAT.

A small chaser is designed as a $0.88 \text{ m} \times 1 \text{ m} \times 1 \text{ m}$ cuboid with a length of 1.5 m after installing the two unfolded manipulators. The mass of the chaser is 100 kg . The moment of inertia of the chaser is $(100, 100, 100) \text{ kg}\cdot\text{m}^2$, and the product of inertia is $0 \text{ kg}\cdot\text{m}^2$. The geometric center of the chaser is fixed with the mass center of the chaser, and it coincides with the origin of S_{cb} . The two manipulators are installed at $(0.49, -0.37, 0) \text{ m}$ and $(0.49, 0.37, 0) \text{ m}$ on the $+x$ surface (front surface) of the chaser. The mass and inertia parameters of the manipulator are listed in Table 4 by referring to Papadopoulos and Dubowsky [46] who considered the manipulator as a cylinder for simplicity.

The maximum thrust force of the chaser is 15 N , the maximum control moment is $15 \text{ N}\cdot\text{m}$, and the specific impulse is $300 \text{ m}\cdot\text{s}^{-1}$. The maximum joint torque is 40 Nm , and the maximum rotation angular velocity of each joint is 20 rpm .

An overall of six cases listed in Table 5 are designed to evaluate the performance and verify the effectiveness of the network controller. The first three cases are designed to evaluate the performance of the network controller compared to a tracking control strategy of the optimal trajectories without considering measurement uncertainty. The chaser is locating

at three different positions expressed in S_{tb} relative to the ENVISAT near $(0, 15, 0) \text{ m}$ at the initial time. The latter three cases are designed to verify the effectiveness of the network controller under different nutation situations with initial measurement uncertainty. The chaser is locating at the same position expressed in S_{tb} relative to the ENVISAT. In the latter three cases, the ENVISAT has a slightly smaller absolute rotation angular velocity ($0.8631^\circ/\text{s}$ compared to $0.8668^\circ/\text{s}$ in the previous three cases). Case 4 simulates the situation in which the ENVISAT has nutation, although it is not considered and measured. Cases 5 and 6 simulate the situations in which the ENVISAT has nutation, but it is not measured accurately at the initial time. The real value of ω_t in the table is the instantaneous angular velocity of the ENVISAT relative to the inertial coordinate system expressed in S_{tb} at the initial time.

In each case, two different control strategies are applied on the chaser: the traditional control strategy that plans at first and then tracks the optimal control result and the network control strategy proposed in this paper. The following performance indices are used to evaluate the mission performance, they are

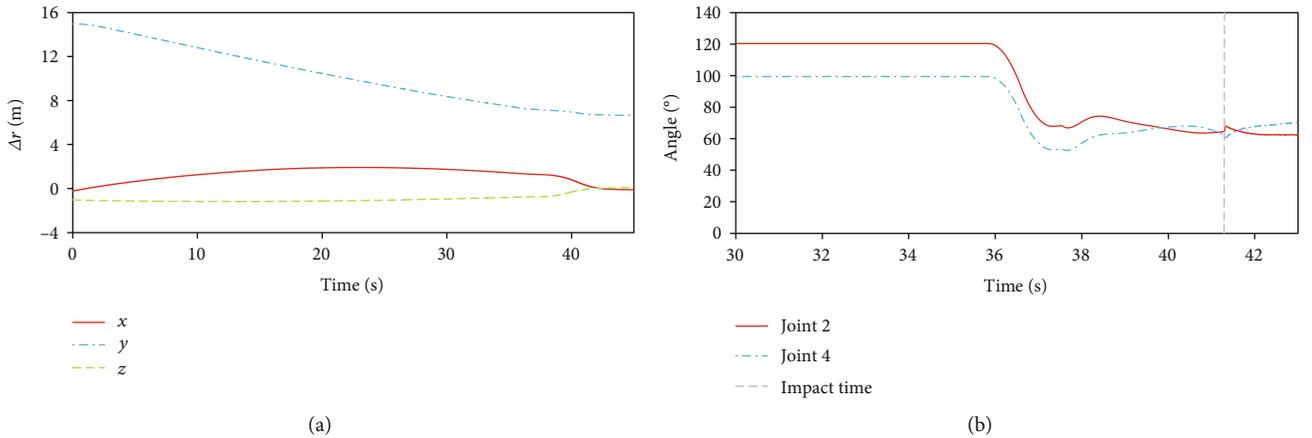
- (i) J_t : the overall mission duration, defined by Equation (8)
- (ii) $\Delta r_{\text{end}} = |\Delta \mathbf{r}_{\text{end}}|$: the relative position between the manipulator end effector and the desired capture

TABLE 6: Simulation results for the first three verification cases.

Index	Case 1		Case 2		Case 3	
	Tracking	Network	Tracking	Network	Tracking	Network
J_t (s)	33.361	38.833	35.593	40.129	35.953	37.521
Left Δr_{end} (m)	0.030	0.100	0.051	0.053	0.056	0.053
Right Δr_{end} (m)	0.086	0.064	0.065	0.047	0.066	0.055
Left J_v (m/s)	0.070	0.008	0.043	0.029	0.010	0.027
Right J_v (m/s)	0.015	0.001	0.019	0.056	0.057	0.043
J_f (kg)	0.466	0.594	0.534	0.602	0.552	0.562

TABLE 7: Simulation results for the latter three verification cases.

Index	Case 4		Case 5		Case 6	
	Tracking	Network	Tracking	Network	Tracking	Network
J_t (s)	31.857	41.281	40.273	41.809	37.377	41.969
Left Δr_{end} (m)	0.088	0.061	0.055	0.060	0.043	0.025
Right Δr_{end} (m)	0.057	0.053	0.128	0.056	0.054	0.053
Left J_v (m/s)	0.086	0.041	0.077	0.077	0.145	0.074
Right J_v (m/s)	0.170	0.023	0.114	0.018	0.154	0.039
J_f (kg)	0.577	0.897	0.485	0.808	0.490	0.797

FIGURE 12: Relative orbit motion and manipulator joint motions in case 5. (a) Relative position expressed in S_b . (b) Angles of joint 2 and joint 4 of the left manipulator.

point within the capture plane at the moment of impact

- (iii) J_v : the relative velocity between the manipulator end effector and the desired capture point at the moment of impact, defined by Equation (9)
- (iv) J_f : fuel consumption defined by Equation (7)

5.2. Capture without Measurement Uncertainty. The simulation results of the first three cases are presented in Table 6. The average J_t of the traditional control strategy is 34.849 s, and that of the network method is 38.828 s. The average J_v of the traditional control strategy is 0.036 m/s, and that of the network method is 0.027 m/s. The average J_f of the traditional control strategy is 0.517 kg, and that of the network method is 0.586 kg. Considering that k_f is 2/3, k_t is 1, and

k_v is 100 (considerably larger to show the importance of the manipulator end impact and to balance the small magnitude of J_v), the overall J of the tracking control strategy is 27.350, and that of the network controller is 29.171. Compared to the tracking control strategy, the performance loss of the network controller is approximately 6.6%.

Since there is no measurement uncertainty in the first three cases, the optimal trajectories planned offline are the true optimal solutions. The tracking result of those optimal trajectories can be regarded as the standard to evaluate the performance of the network controller proposed in this paper. The first three cases indicate that there exist approximately 6.6% performance loss of the network controller. That is, the network controller provides a suboptimal control close to the true optimal control solution. The losses in performance occur due to the following three aspects:

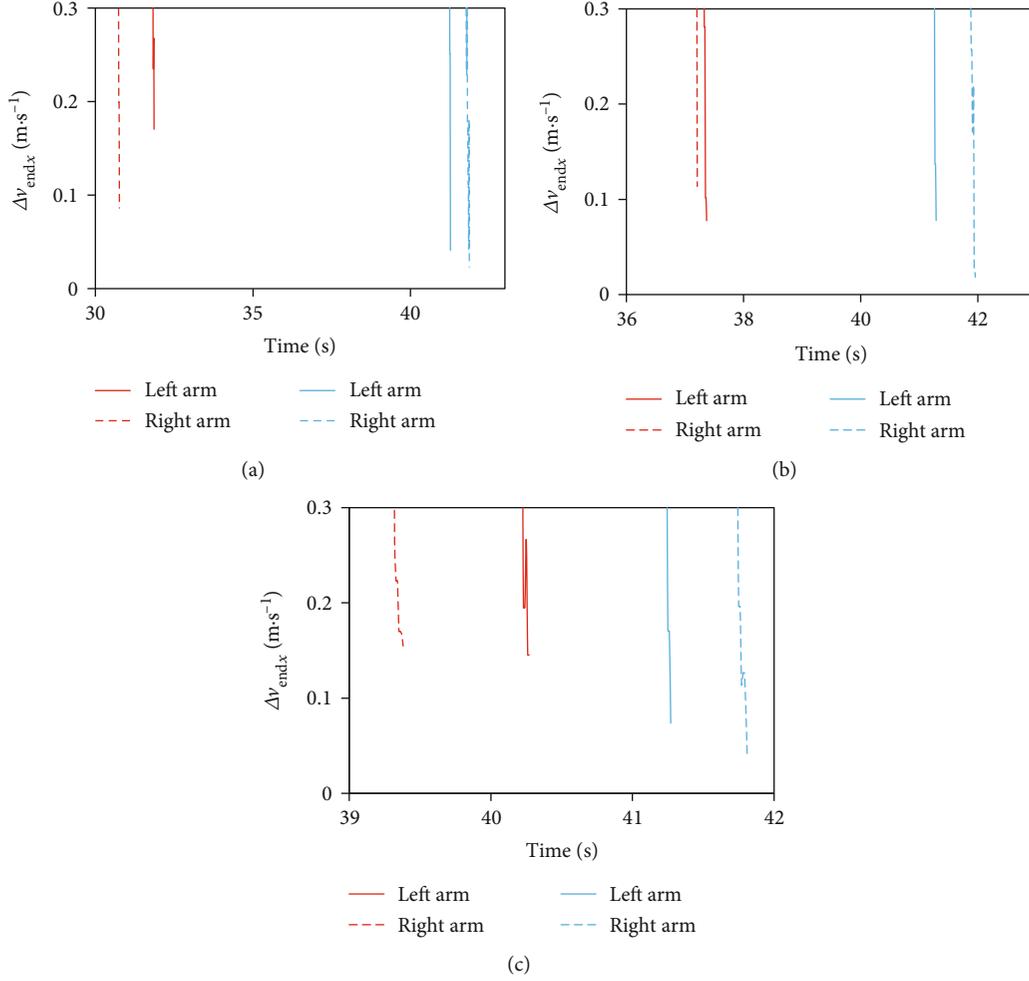


FIGURE 13: Vertical velocities of both the left and right manipulator end effectors relative to the desired capture points. The solid lines represent the left manipulator, and the dotted lines denote the right manipulator. The red lines denote the chaser controlled by the traditional control strategy, and blue lines denote the chaser controlled by the networks. All the lines are terminated when the manipulators hit the target. (a) Case 4. (b) Case 5. (c) Case 6.

- (1) *Samples*. Samples of inferior quality lead to poor learned networks. Unlike other machine learning problems in which the samples are commonly selected from open source datasets, the samples of the approach and capture problem are generated using the GPM and designed algorithms. Thus, the appropriate initial conditions of the samples contribute to the GPM planning stage, and necessary elimination of samples of inferior quality can improve the training accuracy.
- (2) *Training errors (commonly evaluated considering the MSE) that reflect the gaps between the output of the networks and the test datasets*. The training error can be reduced by selecting the appropriate network parameters, which can be realized with experience by using the Monte Carlo method.
- (3) *Generalization errors that reflect the prediction precision to the unknown flight conditions*. The generalization errors are generally related to the network types and structures. The RBF network is a shallow net-

work, and the number of neurons is determined automatically in the training process. The division of the networks according to the dimensions is based on human intelligence. Further research on different kinds of networks and different division methods is in progress.

5.3. Capture with Measurement Uncertainty. The simulation results of the latter three cases are presented in Table 7. The average J_t of the traditional control strategy is 36.502 s, and that for the network methods is 41.686 s. The average J_v of the traditional control strategy is 0.124 m/s, and that for the network controller is 0.045 m/s. The average J_f of the traditional control strategy is 0.517 kg, and that for the network controller is 0.834 kg. Considering that k_f is 2/3, k_t is 1, and k_v is 100, the overall J of the traditional control strategy is 37.252, and that of the network controller is 33.125.

Since there exist measurement errors of ω_t at the initial time, when the nutation is not measured accurately, the tracking of the offline planned trajectories results in larger Δr_{end} and J_v (bold numbers in Table 7). By using the network

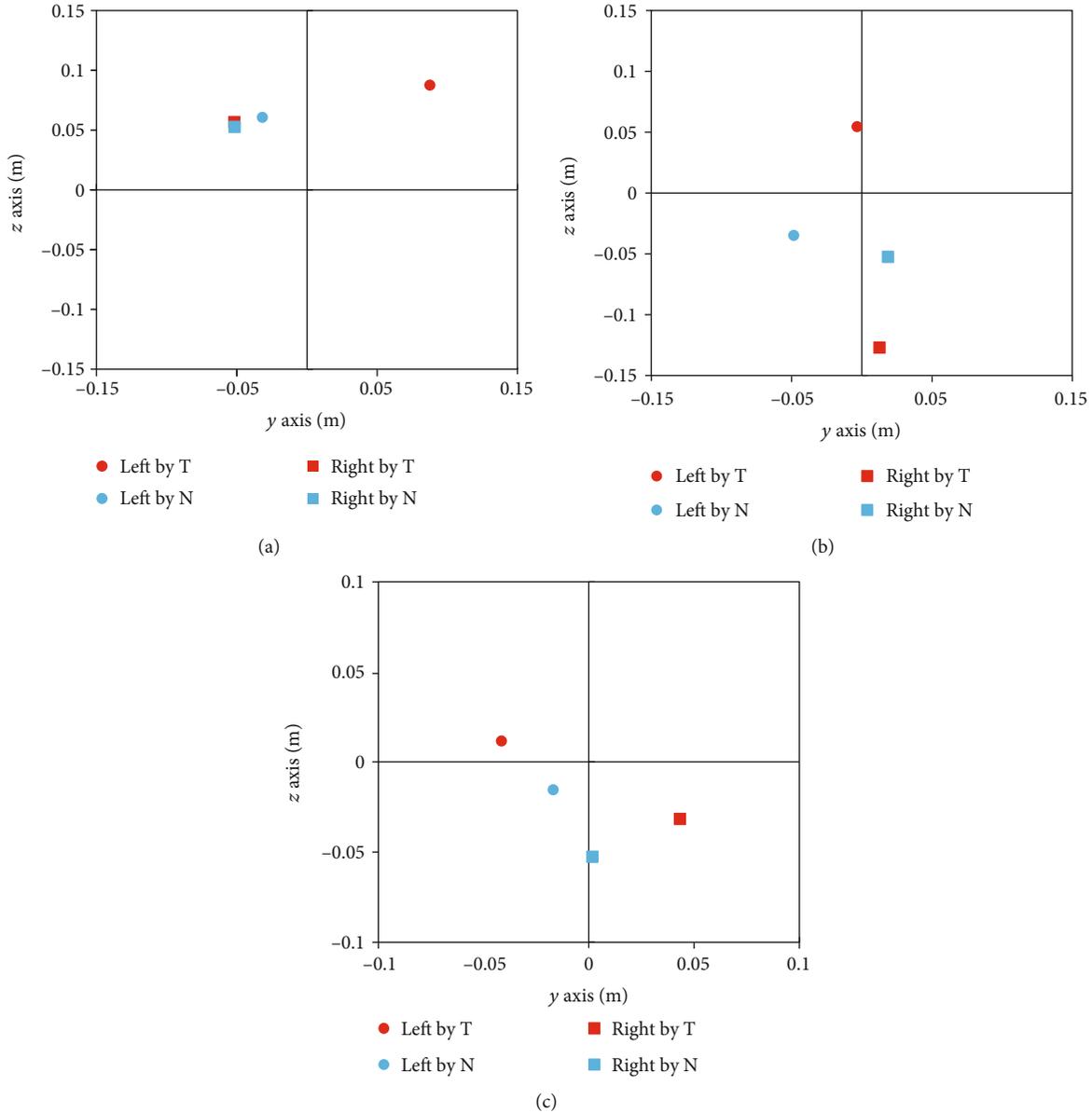


FIGURE 14: Relative positions between both the manipulator end effectors and the desired capture points within the capture plane at the moment of impact. The red dots denote the chasers controlled using the traditional control strategy, and the blue dots denote the chaser controlled by the networks. The circular dots denote the left manipulator end effectors, and the square dots denote the right manipulator end effectors. (a) Case 4. (b) Case 5. (c) Case 6.

controller, the average J_v is 63.71% lower than that of the tracking control strategy, which means that the capture mission has a lower risk to produce debris. On the contrary, the network controller consumes more time and fuel than those controlled by the traditional control strategy, which can be seen from the average values of J_t and J_f .

Taking the simulation results of case 5 as an example, the relative orbit varying with time expressed in S_{ib} is shown in Figure 12(a). The joint 2 and joint 4 angles of the left manipulator varying with time are shown in Figure 12(b).

The vertical velocities of both the left and right manipulator end effectors relative to the desired capture points in all the latter three cases under different control strategies

are shown in Figure 13. When the nutation was neglected, the chaser's manipulator end effectors hit the ENVISAT at a relatively high speed when controlled by the traditional control strategy, as depicted using the red lines in case 4. When the nutation is taken into account but the measurement is not accurate (cases 5 and 6), the chaser's manipulator end effectors still heavily impact the ENVISAT when controlled by the traditional tracking control strategy (red lines), compared to when the chaser is controlled by the networks (blue lines). This indicates that the network controller can conduct a softer capture with less impact.

The relative positions between both the manipulator end effectors and the desired capture points within the capture

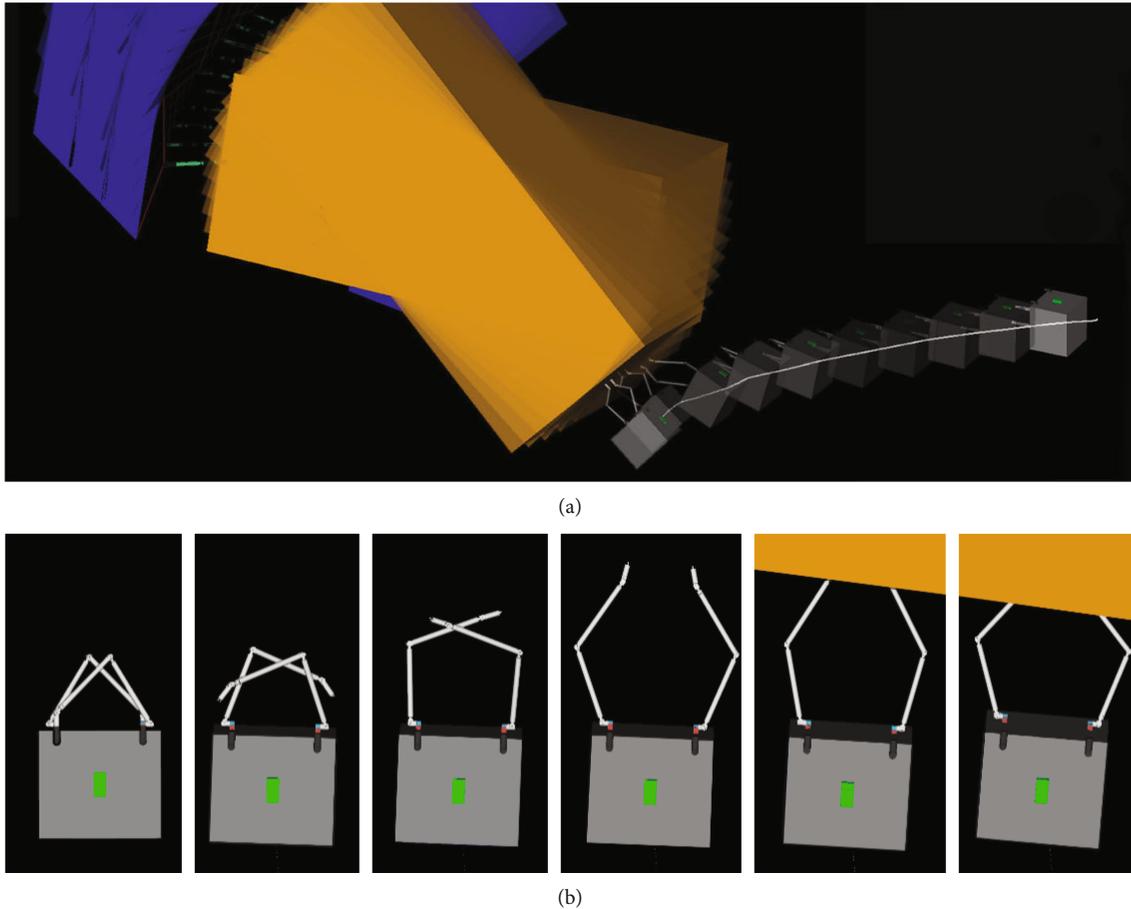


FIGURE 15: Flight history of the chaser in case 5. (a) The approach and capture process in the orbital coordinate system. (b) The attitude motion of the chaser and the deployment of the two manipulators from the negative z direction of the orbital coordinate system.

plane at the moment of impact are shown in Figure 14. The dots closer to the origin indicated a smaller capture position deviation. It can be seen from the image that in all the latter three cases, the chaser controlled by the networks exhibits a smaller capture position deviation (blue dots) compared to that of the chaser controlled using the traditional control strategy (red dots). This indicates that the network controller can conduct a more accurate capture action.

The total flight history of the chaser in case 5 is shown in Figure 15. Multiple flight moments from the initial time to the final time are shown in Figure 15(a). At the initial time, the chaser is at the right of the ENVISAT, and the ENVISAT rotates approximately around its z axis (not exactly around the z axis because of nutation). The relative flight trajectory between the chaser and the ENVISAT is plotted in the white line. Figure 15(b) shows the deployment, pointing, and capturing procedure of the two manipulators.

6. Conclusions

This paper proposes a real-time optimal control strategy to capture the ENVISAT in the presence of nutation and measurement uncertainty using artificial intelligence. From the simulation results, the following observations have been obtained. Firstly, in ideal cases, the capture performance

of our method is comparable with the tracking control approach. Secondly, the average J_v (the relative velocity between the manipulator end effector) is 63.71% lower than that of the tracking control approach, implying the capture mission has a lower risk to produce debris. Thirdly, the solution of the tracking control problem requires an average computational cost of 1128 seconds to generate the policies while our approach is accomplishable in real time. It is noted that the inference time of the neural network is 12.97 milliseconds on ARM (STM32F417) that can be omitted in this case. As a result, our proposed control strategy is particularly suitable to solve a variety of other control tasks where performance and efficiency are both required.

Data Availability

The training sample data and trained networks used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the Key Laboratory of Spacecraft Design Optimization and Dynamic Simulation Technologies, Ministry of Education.

References

- [1] D. Kucharski, G. Kirchner, F. Koidl et al., "Attitude and spin period of space debris Envisat measured by satellite laser ranging," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 12, pp. 7651–7657, 2014.
- [2] G. Andrea, "Don Kessler on Envisat and the Kessler syndrome," *Space Safety Magazine*, 2012, December 2019, <http://www.spacesafetymagazine.com/space-debris/kessler-syndrome/don-kessler-envisat-kessler-syndrome/>.
- [3] S. Sommer, J. Rosebrock, D. Cerutti-Maori, and L. Leushacke, "Temporal analysis of Envisat's rotational motion," in *Proceedings of the 7th European Conference on Space Debris*, Darmstadt, Germany, April 2017.
- [4] N. Koshkin, E. Korobeynikova, L. Shakun, S. Strakhova, and Z. H. Tang, "Remote sensing of the Envisat and Cbers-2B satellites rotation around the centre of mass by photometry," *Advances in Space Research*, vol. 58, no. 3, pp. 358–371, 2016.
- [5] J. N. Pittet, J. Šilha, and T. Schildknecht, "Spin motion determination of the Envisat satellite through laser ranging measurements from a single pass measured by a single station," *Advance in Space Research*, vol. 61, no. 4, pp. 1121–1131, 2018.
- [6] G. Boyarko, O. Yakimenko, and M. Romano, "Optimal rendezvous trajectories of a controlled spacecraft and a tumbling object," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 4, pp. 1239–1252, 2011.
- [7] X. Y. Chu, J. R. Zhang, S. Lu, Y. Zhang, and Y. Sun, "Optimised collision avoidance for an ultra-close rendezvous with a failed satellite based on the Gauss pseudospectral method," *Acta Astronautica*, vol. 128, pp. 363–376, 2016.
- [8] Y. Z. Zhang, P. F. Huang, K. H. Song, and Z. J. Meng, "An angles-only navigation and control scheme for noncooperative rendezvous operations," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 11, pp. 8618–8627, 2019.
- [9] D. Lee and G. Vukovich, "Robust adaptive terminal sliding mode control on SE(3) for autonomous spacecraft rendezvous and docking," *Nonlinear Dynamics*, vol. 83, pp. 2263–2279, 2016.
- [10] L. Sun and Z. Zheng, "Adaptive relative pose control for autonomous spacecraft rendezvous and proximity operations with thrust misalignment and model uncertainties," *Advances in Space Research*, vol. 59, no. 7, pp. 1861–1871, 2017.
- [11] J. Q. Zhang, D. Ye, J. D. Biggs, and Z. W. Sun, "Finite-time relative orbit-attitude tracking control for multi-spacecraft with collision avoidance and changing network topologies," *Advances in Space Research*, vol. 63, no. 3, pp. 1161–1175, 2019.
- [12] R. Zappulla, J. Virgili-Llop, and M. Romano, "Near-optimal real-time spacecraft guidance and control using harmonic potential functions and a modified RRT," in *27th AAS/AIAA Space Flight Mechanics Meeting*, pp. 3813–3831, San Antonio, USA, February 2017.
- [13] A. Reiter, A. Müller, and H. Gattringer, "Rapid nearly-optimal rendezvous trajectory planning using parameter sensitivities," in *ROMANSY 22 - Robot Design, Dynamics and Control*, V. Arakelian and P. Wenger, Eds., vol. 584 of CISM International Centre for Mechanical Sciences (Courses and Lectures), pp. 515–522, Springer, Cham, 2019.
- [14] D. Zhou, Y. Q. Zhang, and S. L. Li, "Receding horizon guidance and control using sequential convex programming for spacecraft 6-DOF close proximity," *Aerospace Science and Technology*, vol. 87, pp. 459–477, 2019.
- [15] Z. Y. Xu, Y. K. Chen, and Z. X. Xu, "Optimal guidance and collision avoidance for docking with the rotating target spacecraft," *Advance in Space Research*, vol. 63, no. 10, pp. 3223–3234, 2019.
- [16] R. Volpe and C. Circi, "Optical-aided, autonomous and optimal space rendezvous with a non-cooperative target," *Acta Astronautica*, vol. 157, pp. 528–540, 2019.
- [17] Y. Chen, Z. He, D. Zhou, Z. H. Yu, and S. L. Li, "Integrated guidance and control for microsatellite real-time automated proximity operations," *Acta Astronautica*, vol. 148, pp. 175–185, 2018.
- [18] J. A. Starek, E. Schmerling, G. D. Maher, B. W. Barbee, and M. Pavone, "Fast, safe, propellant-efficient spacecraft motion planning under Clohessy-Wiltshire-Hill dynamics," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 418–438, 2017.
- [19] M. Sagliano, "Generalized hp pseudospectral-convex programming for powered descent and landing," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 7, pp. 1562–1570, 2019.
- [20] Q. Li, J. P. Yuan, B. Zhang, and H. Wang, "Artificial potential field based robust adaptive control for spacecraft rendezvous and docking under motion constraint," *ISA Transactions*, vol. 95, pp. 173–184, 2019.
- [21] J. Ventura, M. Ciarcia, M. Romano, and U. Walter, "An inverse dynamics-based trajectory planner for autonomous docking to a tumbling target," in *AIAA Guidance, Navigation, and Control Conference*, San Diego, CA, USA, 2016.
- [22] Z. Zhang, J. Wang, and J. X. Li, "Lossless convexification of nonconvex MINLP on the UAV path-planning problem," *Optimal Control Applications and Methods*, vol. 39, no. 2, pp. 845–859, 2018.
- [23] X. F. Liu and P. Lu, "Robust trajectory optimization for highly constrained rendezvous and proximity operations," in *AIAA Guidance, Navigation, and Control (GNC) Conference*, pp. 2013–4720, Boston, MA, USA, 2013.
- [24] P. Lu and X. F. Liu, "Autonomous trajectory planning for rendezvous and proximity operations by conic optimization," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 2, pp. 375–389, 2013.
- [25] A. Caubet and J. D. Biggs, "An inverse dynamics approach to the guidance of spacecraft in close proximity of tumbling debris," in *66th International Astronautical Congress*, Israel, 2015.
- [26] G. Misra and X. L. Bai, "Optimal path planning for free-flying space manipulators via sequential convex programming," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 11, pp. 1–8, 2017.
- [27] J. J. Zhang, S. D. Yao, B. Peng, Q. L. Zhang, and T. Zhang, "Path optimization of free-floating space robot based on adaptive genetic algorithm," in *Proceedings of the 3rd International Conference on Material, Mechanical and Manufacturing Engineering*, pp. 1070–1076, Guangzhou, June 2015.

- [28] J. C. Hu and T. S. Wang, "Minimum base attitude disturbance planning for a space robot during target capture," *Journal of Mechanisms and Robotics*, vol. 10, no. 5, pp. 1–13, 2018.
- [29] T. Zahroof, A. Bylard, H. Shageer, and M. Pavone, "Perception-constrained robot manipulator planning for satellite servicing," in *2019 IEEE Aerospace Conference*, Big Sky, MT, USA, USA, 2019.
- [30] G. Misra and X. L. Bai, "Task-constrained trajectory planning of free-floating space-robotic systems using convex optimization," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 11, pp. 2857–2870, 2017.
- [31] Y. Deng, F. Bao, X. Deng, R. Wang, Y. Kong, and Q. Dai, "Deep and structured robust information theoretic learning for image analysis," *IEEE Transactions on Image Processing*, vol. 25, no. 9, pp. 4209–4221, 2016.
- [32] Y. Deng, F. Bao, Y. Yang et al., "Information transduction capacity reduces the uncertainties in annotation-free isoform discovery and quantification," *Nucleic Acids Research*, vol. 45, no. 15, article e143, 2017.
- [33] D. Izzo, M. Märtens, and B. F. Pan, "A survey on artificial intelligence trends in spacecraft guidance dynamics and control," 2018, <https://arxiv.org/abs/1812.02948>.
- [34] C. Sánchez-Sánchez and D. Izzo, "Real-time optimal control via deep neural networks: study on landing problems," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 5, pp. 1122–1135, 2018.
- [35] D. Izzo, C. Sprague, and D. Taylor, "Machine learning and evolutionary techniques in interplanetary trajectory design," 2018, <https://arxiv.org/abs/1802.00180v2>.
- [36] J. D. Biggs and H. Fournier, "Neural-network-based optimal attitude control using four impulsive thrusters," *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 2, pp. 299–309, 2020.
- [37] R. Deutsch, *Orbital Dynamics of Space Vehicles*, Literary Licensing, Whitefish, MT, USA, 1964.
- [38] P. C. Hughes, *Spacecraft Attitude Dynamics*, Wiley, Hoboken, NJ, USA, 1986.
- [39] Y. K. Zhu, J. Z. Qiao, and L. Guo, "Adaptive sliding mode disturbance observer-based composite control with prescribed performance of space manipulators for target capturing," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 3, pp. 1973–1983, 2018.
- [40] B. Liang and W. F. Xu, *Space Robotics: Modeling, Planning and Control*, Tsinghua University Press, Beijing, PRC, 2017.
- [41] J. He, H. C. Zheng, F. Gao, and H. B. Zhang, "Dynamics and control of a 7-DOF hybrid manipulator for capturing a non-cooperative target in space," *Mechanism and Machine Theory*, vol. 140, pp. 83–103, 2019.
- [42] D. A. Vallado, *Fundamentals of Astrodynamics and Applications*, Microcosm Press, New York, 2007.
- [43] M. A. Patterson and A. V. Rao, "GPOPS-II: a MATLAB software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming," *ACM Transactions on Mathematical Software*, vol. 41, no. 1, pp. 1–37, 2014.
- [44] R. Furfaro, I. Bloise, M. Orlandelli, P. D. Lizia, F. Topputo, and R. Linares, "Deep learning for autonomous lunar landing," in *AAS/AIAA Astrodynamics Specialist Conference*, pp. 1–22, Snowbird, UT, USA, August 2018.
- [45] N. O. Gómez and S. J. I. Walker, "Earth's gravity gradient and eddy currents effects on the rotational dynamics of space debris objects: Envisat case study," *Advances in Space Research*, vol. 56, no. 3, pp. 494–508, 2015.
- [46] E. Papadopoulos and S. Dubowsky, "Coordinated manipulator/spacecraft motion control for space robotic systems," in *Proceedings of 1991 IEEE International Conference on Robotics and Automation*, pp. 1696–1701, Sacramento, California, USA, 1991.