

Research Article

Staring Imaging Real-Time Optimal Control Based on Neural Network

Peiyun Li ,^{1,2} Yunfeng Dong ,^{1,2} and Hongjue 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 10 May 2020; Revised 21 September 2020; Accepted 8 October 2020; Published 17 November 2020

Academic Editor: Giovanni Palmerini

Copyright © 2020 Peiyun 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.

In this paper, a real-time optimal attitude controller is designed for staring imaging, and the output command is based on future prediction. First, the mathematical model of staring imaging is established. Then, the structure of the optimal attitude controller is designed. The controller consists of a preprocessing algorithm and a neural network. Constructing the neural network requires training samples generated by optimization. The objective function in the optimization method takes the future control effect into account. The neural network is trained after sample creation to achieve real-time optimal control. Compared with the PID (proportional-integral-derivative) controller with the best combination of parameters, the neural network controller achieves better attitude pointing accuracy and pointing stability.

1. Introduction

Staring mode of satellites can be used to collect the dynamic information of ground targets. Emergency rescue, airport, and traffic monitoring are all cases where staring imaging is more advantageous than conventional modes.

The optical axis of the camera is required to point fixedly to the ground target in the process of staring imaging. Thus, the attitude controller should run in a three-axis mode. Imaging quality is closely related to attitude control performance which is affected by the control strategy.

Control laws in analytical type are widely applied in attitude controllers for staring imaging. In those analytical control laws, one-step decision is usually considered instead of long-term planning. Lian proposed two different calculation methods of the desired angular velocity for staring imaging [1]. Zhang designed the control strategy under the condition that the target error can be detected, and that the actuator acts with delay [2]. Cui designed a PD adaptive controller for multiobjective staring imaging [3]. Zhang designed a controller to observe spatial cooperative targets [4].

Staring imaging is one application of attitude tracking control. For the more general tracking control problem, one-step decision instead of long-term planning is also the main concern. Wu [5] designed a combined control strategy consisting of a saturation controller for spatial motion targets and proposed a backstepping controller using multiexecuting mechanism control. Leonard designed a control law to monitor space debris [6]. An adaptive fault-tolerant controller with backstepping control is designed by Hu for attitude tracking control problems of rigid spacecraft under unknown but constant inertial parameters, unexpected disturbances, and faults under actuators or saturation [7]. Lu proposed an FNTSM control law with finite time control [8]. Pong designed a control law to achieve spacecraft pointing in a specified direction [9]. Wu implemented attitude tracking control based on iterative learning [10].

Recently, neural networks are mainly applied to the following three aspects in aerospace guidance and control.

First, neural networks are used to accelerate some evolutionary algorithms in aerospace. Cassioli et al. [11] applied SVMs (support vector machines) to improve initial guesses

for some trajectory problems in the GTOP (global trajectory optimization problem) database. Basu et al. [12] deployed a neural network to set initial conditions for the PSO (particle swarm optimization), which reduced the planning time of reorientation for a space telescope.

Another application of neural networks is real-time trajectory planning. Izzo et al. [13] trained a DNN (deep neural network) to design optimal interplanetary trajectories. Furfaro et al. [14] designed neural networks based on LSTM (long short-term memory) to generate optimal trajectories for planetary landing tasks.

Neural networks are also directly used in aerospace controllers. Zou [15] proposed a scheme for limited time attitude tracking control of spacecraft using the terminal sliding mode and Chebyshev neural network. Biggs and Fournier [16] trained MLP to optimize the thruster application scheme to control the attitude of a 12 U CubeSat. CNNs (convolutional neural networks) have shown superior performance in the image classification, and they are also used as control networks. Furfaro et al. [17] designed a CNN to process simulated moon images and generate landing control commands in the simulation of a 2D environment.

Different from traditional control methods, the optimal control method proposed in this paper seeks the optimal control command sequence for a period of time in the future. The beginning of the command sequence is the current optimal control decision. However, the optimal control method requires iterative calculation, which consumes high computational resources and is not a real-time control method. Thus, the optimal control method is used to create training samples where the relationship between input and output is contained. Finally, the neural network is trained to achieve optimal control and real-time control simultaneously.

The main contributions of this paper are stated as follows.

- (i) In the aspect of staring imaging, a controller which considers a future period of time and makes the optimal control decision is proposed, and it performs better than the traditional controller
- (ii) An algorithm for compressing the satellite state information is proposed. The state information consists of current dynamics of the satellite as well as the desired sequence of the future angle and angular velocity. After compression, only 12 scalars are prepared as the input of the neural network

This paper is organized as follows. Section 2 states the staring imaging problem of the spacecraft system. The structure of the real-time optimal controller for staring imaging is designed in Section 3. In Section 4, auxiliary parts of the controller are designed. In Sections 5 and 6, the training sample set is generated, and the neural network in the controller is constructed, respectively. Simulation results are provided in Section 7. Finally, conclusions are given in Section 8.

2. Problem Formulation

2.1. Definition of the Coordinate System. Inertial frame S_i ($Ox_iy_iz_i$): its origin is located at the mass center of the

Earth. The x_i axis points from the Earth towards the vernal equinox. The z_i axis is along the Earth rotation axis. The y_i axis is determined by the right-hand rule.

Satellite body frame S_b ($Ox_by_bz_b$): its origin is located at the mass center of the satellite. x_b , y_b , and z_b axes lie along the corresponding principal axes of the satellite.

Desired satellite body frame S_d ($Ox_dy_dz_d$): it is used for describing the desired attitude of the satellite. Its three axes lie along x_b , y_b , and z_b when the satellite is located at the desired attitude.

Orbital reference frame S_o ($Ox_oy_oz_o$): its origin is located at the mass center of the satellite. The y_o axis is along the anti-symmetric direction of the orbit momentum. The z_o axis points to the Earth center. The x_o axis is determined by the right-hand rule.

2.2. Kinematic and Dynamic Equations. The satellite is considered as a rigid body equipped with 3 reaction wheels installed orthogonally along the three axes of the satellite body frame. The camera is fixed to the satellite, and its optical axis lies along the z_b axis of the satellite body frame.

The mathematical model contains orbit and attitude constraints, both of which are described by kinematic and dynamic equations. The orbital kinematic equation in the inertial frame is given by

$$\dot{\mathbf{R}}_s = \mathbf{V}_s, \quad (1)$$

where $\mathbf{R}_s \in \mathbb{R}^{3 \times 1}$ is the satellite position and $\mathbf{V}_s \in \mathbb{R}^{3 \times 1}$ is the satellite velocity.

The orbital dynamic equation in the inertial frame is

$$\dot{\mathbf{V}}_s = -\frac{\mu}{|\mathbf{R}_s|^3} \mathbf{R}_s, \quad (2)$$

where μ is the product of the gravitational constant and the mass of the Earth.

The attitude kinematic equation ([18], pp. 344) is defined using quaternions

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & \omega_{bz} & -\omega_{by} & \omega_{bx} \\ -\omega_{bz} & 0 & \omega_{bx} & \omega_{by} \\ \omega_{by} & -\omega_{bx} & 0 & \omega_{bz} \\ -\omega_{bx} & -\omega_{by} & -\omega_{bz} & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}, \quad (3)$$

where $\mathbf{q} = [q_1 \ q_2 \ q_3 \ q_4]^T = [\mathbf{q}_v \ q_4]^T$ is the quaternion denoting the rotation from the inertial frame S_i to the satellite body frame S_b . $\mathbf{q}_v = [q_1 \ q_2 \ q_3]^T$ is the vector part, and q_4 is the scalar part. $\boldsymbol{\omega}_b = [\omega_{bx} \ \omega_{by} \ \omega_{bz}]^T$ is the angular velocity vector of the satellite body frame S_b with respect to the inertial frame S_i . It can also be expressed in terms of basis vectors of S_b as $\boldsymbol{\omega}_b = \omega_{bx}\mathbf{x}_b + \omega_{by}\mathbf{y}_b + \omega_{bz}\mathbf{z}_b$.

The attitude dynamics equation is

$$\mathbf{I}\dot{\boldsymbol{\omega}}_b + \boldsymbol{\omega}_b \times (\mathbf{I}\boldsymbol{\omega}_b) + \boldsymbol{\omega}_b \times (\mathbf{I}_w \boldsymbol{\Omega}_w) = \mathbf{M}_c + \mathbf{M}_{ext}, \quad (4)$$

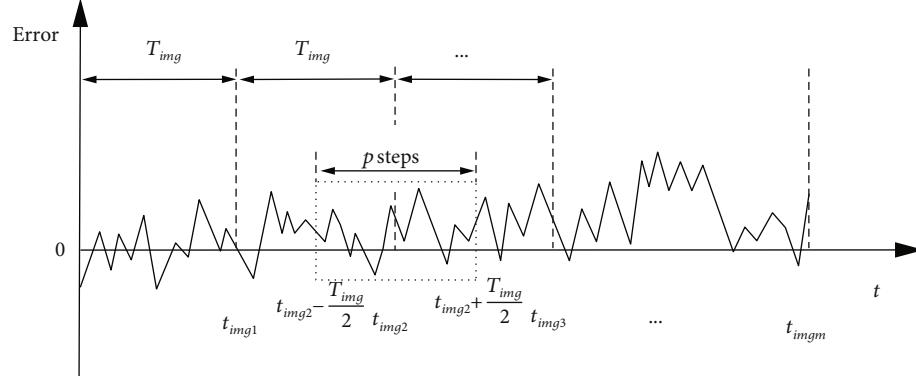


FIGURE 1: Imaging time interval and imaging instants.

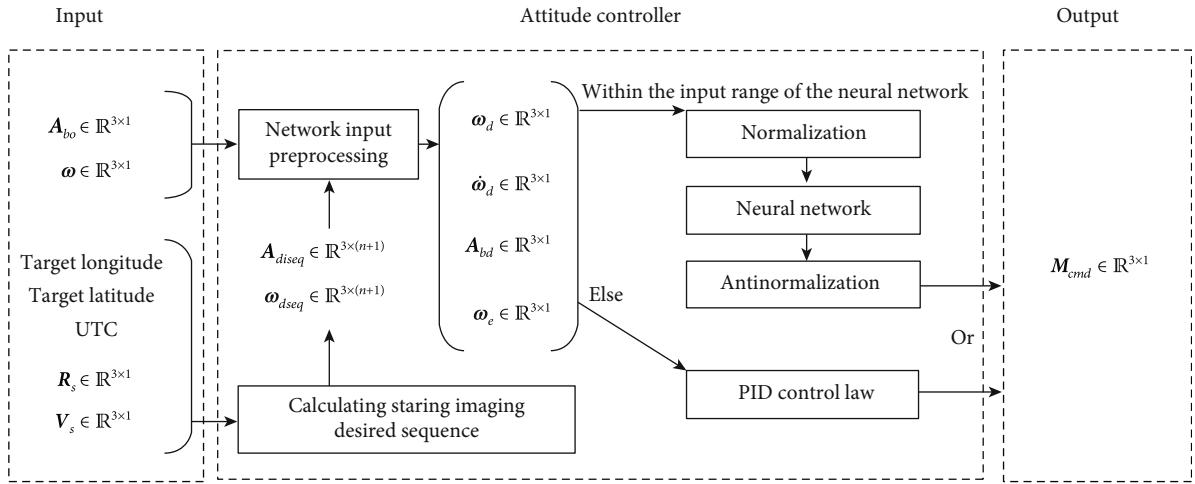


FIGURE 2: The structure of the attitude controller.

where $I \in \mathbb{R}^{3 \times 3}$ is the inertia tensor of the whole satellite.

$$I = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{xy} & I_y & -I_{yz} \\ -I_{xz} & -I_{yz} & I_z \end{bmatrix}. \quad (5)$$

$I_w \in \mathbb{R}^{3 \times 3}$ is the inertia tensor of the 3 reaction wheels. $\Omega_w \in \mathbb{R}^{3 \times 1}$ is the angular velocity of the 3 reaction wheels relative to the satellite body. $M_c \in \mathbb{R}^{3 \times 1}$ is the control torque generated by reaction wheels. $M_{ext} \in \mathbb{R}^{3 \times 1}$ is the external disturbance torque.

2.3. Attitude Control Effect Evaluation. As illustrated in Figure 1, T_{img} is the imaging time interval, and t_{imgi} ($i=1, 2, \dots$) is the instant when image i is taken. Imaging quality has a close relationship with the nearby attitude control effect. For example, the image taken in t_{img2} is affected by the attitude control effect during $[t_{img2} - T_{img}/2, t_{img2} + T_{img}/2]$.

Thus, two steps are designed to properly evaluate the attitude control effect for the whole task of staring imaging. First, the attitude control effects nearby each image are evaluated, respectively, (i.e., the attitude control effect nearby image i is evaluated based on the angle and angular velocity within

$[t_{imgi} - T_{img}/2, t_{imgi} + T_{img}/2]$). Then, the average value of those effects is obtained for evaluating the whole task.

Pointing accuracy and stability are the two major categories that the payload attitude control requirements fall into [19]. Therefore, mean pointing error [20] and rate error [19] are included in the attitude control effect evaluation. Assuming that the number of all images is m , the pointing accuracy J_A is proposed in Eq. (6), the pointing stability J_ω is proposed in Eq. (7), and the control effect of the whole task is obtained by Eq. (8).

$$J_A = \frac{1}{3m(p+1)} \sum_{i=1}^m \sum_{j=1}^3 \sum_{k=1}^{p+1} |A_{eseq(j,k)}|. \quad (6)$$

In Eq. (6), symbol $| \ |$ means getting the absolute value of the variable within. p is the number of simulation steps during $[t_{imgi} - T_{img}/2, t_{imgi} + T_{img}/2]$, and $A_{eseq} \in \mathbb{R}^{3 \times (p+1)}$ is the Euler angle error sequence during that time. $A_{eseq(j,k)}$ is the element in row j and column k of the matrix A_{eseq} . The column k in the matrix A_{eseq} is denoted as $A_{eseq(k)} \in \mathbb{R}^{3 \times 1}$, and the Euler angle error vector $A_{eseq(k)} = [\varphi_e, \theta_e, \psi_e]^T$ represents the rotation from the desired satellite body frame to the

real satellite body frame, with the order 3-1-2 (i.e., $\psi_e - \varphi_e - \theta_e$). Other Euler angles defined in this paper share the same method of rotation ($\psi_e - \varphi_e - \theta_e$) as $A_{\text{eseq}(k)}$.

$$J_\omega = \frac{1}{3m(p+1)} \sum_{i=1}^m \sum_{j=1}^3 \sum_{k=1}^{p+1} |\omega_{\text{eseq}(j,k)}|. \quad (7)$$

In Eq. (7), $\omega_{\text{eseq}} \in \mathbb{R}^{3 \times (p+1)}$ is the angular velocity error sequence during $[t_{\text{img}_i} - T_{\text{img}}/2, t_{\text{img}_i} + T_{\text{img}}/2]$, and $\omega_{\text{eseq}(j,k)}$ is the element in row j and column k of matrix ω_{eseq} . The column k in matrix ω_{eseq} is denoted as $\omega_{\text{eseq}(k)} \in \mathbb{R}^{3 \times 1}$, and the angular velocity error vector $\omega_{\text{eseq}(k)} = \omega_b - \omega_d$ represents the difference between the real angular velocity $\omega_b \in \mathbb{R}^{3 \times 1}$ (in the satellite body frame) and the desired angular velocity $\omega_d \in \mathbb{R}^{3 \times 1}$ (in the desired satellite body frame).

$$J = k_1 \cdot J_A + k_2 \cdot J_\omega. \quad (8)$$

In Eq. (8), k_1 and k_2 are the weights of each term.

3. Structure of the Real-Time Optimal Controller

As is shown in the dashed box named “Attitude Controller” in Figure 2, six modules are outlined. They are established in the following sections of this paper.

The modules named “Calculating Staring Imaging Desired Sequence” and “Network Input Preprocessing” are explained in the section named “Auxiliary Parts of the Real-time Optimal Controller.”

The modules named “Normalization,” “Neural Network,” and “Anti-normalization” are central parts, which will be explained in two sections of this paper. In order to establish the neural network, the method of creating training samples will be firstly shown in the section named “Sample Creation.” Then, the design of those three modules will be shown in the section named “Learning the Optimal Control Strategy.”

The module named the “PID Control Law” is designed in case that the real inputs exceed the designed input range of the neural network. This module only works under big disturbances, and it is tested in “Impulse Response” of “Results and Discussion.” The PID coefficients are shown in Table 1.

The inputs of the controller consist of seven parts, which are current Euler angle $A_{bo} \in \mathbb{R}^{3 \times 1}$ from the star sensor, current angular velocity $\omega_b \in \mathbb{R}^{3 \times 1}$ from the gyro, target longitude and latitude from telecommand, UTC (coordinated universal time) from the on-board computer, and current $R_s \in \mathbb{R}^{3 \times 1}$ as well as $V_s \in \mathbb{R}^{3 \times 1}$ from GPS (Global Positioning System). A_{bo} represents the rotation from the orbital reference frame to the satellite body frame.

The output of the controller is the torque command $M_{\text{cmd}} \in \mathbb{R}^{3 \times 1}$. Reaction wheels receive the torque command and generate the M_c in Eq. (4).

TABLE 1: The optimal PID coefficient combination.

Coefficient	<i>x</i> -axis	<i>y</i> -axis	<i>z</i> -axis
Proportional	17	54	18
Integral	0	0	0
Differential	46	77	50

4. Auxiliary Parts of the Real-Time Optimal Controller

4.1. Calculating Staring Imaging Desired Sequence. The module named “Calculating Staring Imaging Desired Sequence” is used to predict and obtain the future data for the controller. The outputs consist of the desired Euler angle sequence $A_{\text{diseq}} \in \mathbb{R}^{3 \times (n+1)}$ and the desired angular velocity sequence $\omega_{\text{diseq}} \in \mathbb{R}^{3 \times (n+1)}$. n is the number of steps in the simulation to predict the future.

Denote τ as the time taken in a single simulation step. Column j in matrix A_{diseq} is defined as $A_{\text{diseq}(j)} \in \mathbb{R}^{3 \times 1}$, and it represents the desired Euler angle (representing the rotation from the inertial frame to the desired satellite body frame) in time $\tau(j-1)$. Column j in matrix ω_{diseq} is defined as $\omega_{\text{diseq}(j)} \in \mathbb{R}^{3 \times 1}$, and it represents the desired angular velocity (in the desired satellite body frame) in time $\tau(j-1)$. A_{diseq} and ω_{diseq} are calculated according to the states of the satellite, and the algorithm is shown in ref. [1].

4.2. Network Input Preprocessing. The module named “Network Input Preprocessing” is used to calculate the input for neural network. The inputs consist of A_{bo} , ω_b , A_{diseq} , and ω_{diseq} , and the outputs consist of $\omega_d \in \mathbb{R}^{3 \times 1}$, $\dot{\omega}_d \in \mathbb{R}^{3 \times 1}$, $A_{bd} \in \mathbb{R}^{3 \times 1}$, and $\omega_e \in \mathbb{R}^{3 \times 1}$.

The desired angular velocity ω_d is the first column in matrix ω_{diseq} .

$$\omega_d = \omega_{\text{diseq}(1)}. \quad (9)$$

The method of calculating the first derivative of the desired angular velocity $\dot{\omega}_d$ is

$$\dot{\omega}_d = \frac{\omega_{\text{diseq}(n)} - \omega_{\text{diseq}(1)}}{\tau \cdot n}. \quad (10)$$

The Euler angle A_{bd} is obtained by the following steps. First, the satellite current Euler angle A_{bo} is converted [18] to coordinate transformation matrix $L_{bo} \in \mathbb{R}^{3 \times 3}$ (representing the rotation from S_o to S_b). The current desired Euler angle $A_{\text{diseq}(1)}$ is converted to coordinate transformation matrix L_{di} . The coordinate transformation matrix from S_i to S_o is defined as L_{oi} , and it can be obtained [21] according to R_s and V_s . The coordinate transformation matrix from S_d to S_b is obtained by $L_{bd} = L_{bo} \cdot L_{oi} \cdot L_{di}^T$. Finally, L_{bd} can be converted to A_{bd} .

The current angular velocity error vector $\omega_e = \omega_b - \omega_d$ represents the difference between the angular velocity ω_b and the desired angular velocity ω_d .

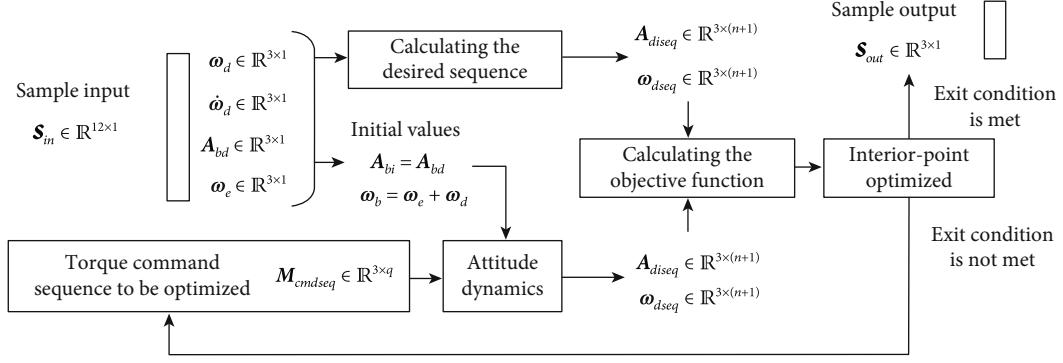


FIGURE 3: The algorithm of sample creation.

5. Sample Creation

5.1. Structure of the Sample Creation Algorithm. A sample set should be created before the neural networks are trained. The algorithm for creating a sample in the sample set is shown in Figure 3.

To start the algorithm, the interior-point optimizer firstly provides a three-axis zero torque command sequence $M_{cmdseq} \in \mathbb{R}^{3 \times q}$, which is the matrix to be optimized in the loop. q is the number of torque commands used in the simulation of the “Attitude Dynamics” module.

The inputs of the “Attitude Dynamics” module are M_{cmdseq} as well as the initial values, and the outputs of that module are the Euler angle sequence $A_{biseq} \in \mathbb{R}^{3 \times (n+1)}$ as well as the angular velocity sequence $\omega_{biseq} \in \mathbb{R}^{3 \times (n+1)}$. n is the number of steps in the simulation.

The sample input $S_{in} \in \mathbb{R}^{12 \times 1}$ is also used for calculating the desired sequence $A_{diseq} \in \mathbb{R}^{3 \times (n+1)}$ and $\omega_{diseq} \in \mathbb{R}^{3 \times (n+1)}$. The more A_{biseq} approaches A_{diseq} and ω_{biseq} approaches ω_{diseq} , the better M_{cmdseq} is. Finally, $S_{out} \in \mathbb{R}^{3 \times 1}$ is obtained when the exit condition is met.

5.2. Definitions of Sample Input and Output. Each sample is divided into a sample input vector S_{in} and a sample output vector S_{out} , and their definitions are

$$S_{in} = [\omega_d, \dot{\omega}_d, A_{bd}, \omega_e]^T \quad (11)$$

$$S_{out} = M_{cmd} \quad (12)$$

The number of samples should cover the input range of the neural network. The input vector S_{in} of each sample is created randomly, with the 12 elements selected within the designated input ranges shown in Table 2. The ranges are selected according to the requirement of staring imaging tasks. If the input is out of the designated range, M_{cmd} will be generated by the PID controller instead of the neural network. This function is tested in the simulation part of this paper. The variables in Table 2 are defined as $\omega_d = [\omega_{dx}, \omega_{dy}, \omega_{dz}]^T$, $\dot{\omega}_d = [\dot{\omega}_{dx}, \dot{\omega}_{dy}, \dot{\omega}_{dz}]^T$, $A_{bd} = [\varphi_{bd}, \theta_{bd}, \psi_{bd}]^T$, and $\omega_e = [\omega_{ex}, \omega_{ey}, \omega_{ez}]^T$.

TABLE 2: Input range of sample and the neural network.

Sample input	Range
ω_{dx}	$[-0.1, 0.1] \text{ } ^\circ/\text{s}$
ω_{dy}	$[-1.2, 0] \text{ } ^\circ/\text{s}$
ω_{dz}	$[-0.05, 0.05] \text{ } ^\circ/\text{s}$
$\dot{\omega}_{dx}$	$[-0.005, 0.005] \text{ } ^\circ/\text{s}^2$
$\dot{\omega}_{dy}$	$[-0.015, 0.015] \text{ } ^\circ/\text{s}^2$
$\dot{\omega}_{dz}$	$[-0.005, 0.005] \text{ } ^\circ/\text{s}^2$
φ_{bd}	$[-0.02, 0.02] \text{ } ^\circ$
θ_{bd}	$[-0.02, 0.02] \text{ } ^\circ$
ψ_{bd}	$[-0.02, 0.02] \text{ } ^\circ$
ω_{ex}	$[-0.004, 0.004] \text{ } ^\circ/\text{s}$
ω_{ey}	$[-0.004, 0.004] \text{ } ^\circ/\text{s}$
ω_{ez}	$[-0.004, 0.004] \text{ } ^\circ/\text{s}$

5.3. Calculating the Desired Sequence. From the desired angular velocity ω_d and its first derivative $\dot{\omega}_d$ at the initial time, the desired angular velocity sequence ω_{diseq} can be generated by

$$\omega_{diseq(j)} = \omega_d + \tau(j-1)\dot{\omega}_d \quad (j = 1, 2, \dots, n+1), \quad (13)$$

where τ is the simulation time step and $\omega_{diseq(j)} \in \mathbb{R}^{3 \times 1}$ is the column j in matrix ω_{diseq} .

The desired Euler angle sequence $A_{diseq} \in \mathbb{R}^{3 \times (n+1)}$ is calculated via quaternions. The desired angular velocity sequence ω_{diseq} can be integrated in time to obtain the quaternion sequence $q_{diseq} \in \mathbb{R}^{4 \times (n+1)}$ (denoting the rotation from the inertial frame S_i to the desired satellite body frame S_d). The column j in matrix $q_{diseq} \in \mathbb{R}^{4 \times (n+1)}$ is $q_{diseq(j)} \in \mathbb{R}^{4 \times 1}$. It is set that the inertial frame coincides with the desired attitude frame at the beginning of the simulation in sample creation. Thus, the quaternion at the initial time is $q_{diseq(1)} = [1, 0, 0, 0]^T$. The following equations are used to calculate the quaternion sequence q_{diseq} .

$$\dot{q}_{diseq(j)} = \frac{1}{2} \boldsymbol{\Omega}(q_{diseq(j)}) q_{diseq(j)} \quad (j = 1, 2, \dots, n+1), \quad (14)$$

$$\mathbf{q}_{\text{diseq}(j+1)} = \mathbf{q}_{\text{diseq}(j)} + \tau \cdot \dot{\mathbf{q}}_{\text{diseq}(j)} \quad (j = 1, 2, \dots, n+1), \quad (15)$$

where the function $\Omega(\omega)$ is defined as

$$\Omega(\omega) = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix}, \quad (16)$$

$$\omega = [\omega_x, \omega_y, \omega_z]^T. \quad (17)$$

The quaternion sequence $\mathbf{q}_{\text{diseq}}$ can be converted to the desired Euler angle sequence $\mathbf{A}_{\text{diseq}}$.

5.4. Calculating the Initial Values of Simulation. The desired satellite body frame coincides with the inertial frame at the initial time. Thus, the initial Euler angle of the satellite is

$$\mathbf{A}_{bi} = \mathbf{A}_{bd}, \quad (18)$$

where \mathbf{A}_{bi} denotes the rotation from the inertial frame S_i to the satellite body frame S_b .

The initial satellite angular velocity in the satellite body frame is

$$\omega_b = \omega_e + \omega_d. \quad (19)$$

5.5. Attitude Dynamics. It has been defined that q is the number of commands in $\mathbf{M}_{\text{cmdseq}} \in \mathbb{R}^{3 \times q}$, and that n is the number of steps in the simulation. Assuming that T is the time interval between two commands, the relationship between q, T, n , and the simulation time step τ is

$$qT = nt \quad (n = k \cdot q, \quad k = 1, 2, 3, \dots), \quad (20)$$

where k is the ratio of T to τ and it is a positive integer.

The standard fourth-order Runge-Kutta method is used to calculate the kinematics and dynamics of the satellite. The Euler angle sequence $\mathbf{A}_{biseq} \in \mathbb{R}^{3 \times (n+1)}$ and the angular velocity sequence $\omega_{bseq} \in \mathbb{R}^{3 \times (n+1)}$ in the whole process are recorded, and they are obtained after the simulation.

5.6. Calculating the Objective Function. The objective function J (defined in Eq. (8)) is used to evaluate the performance of $\mathbf{M}_{\text{cmdseq}}$. The smaller the objective function is, the better the control performance is.

5.7. Interior-Point Optimization. At the beginning of the sample creation for a sample input \mathbf{S}_{in} , the interior-point optimizer creates a three-axis zero torque command sequence $\mathbf{M}_{\text{cmdseq}} \in \mathbb{R}^{3 \times q}$. The objective function J is obtained after the simulation where $\mathbf{M}_{\text{cmdseq}}$ is tested. Then, the optimizer makes the judgment according to J . If the exit condition is satisfied, the loop exits. If it is not satisfied, a new $\mathbf{M}_{\text{cmdseq}}$ is generated and the optimization continues.

TABLE 3: Training parameters.

Parameter	Values	Total number of values
Activation function	Tansig, logsig, sigmoid, relu	4
Nodes in hidden layer	10, 11, 12, ..., 60	51
Training iterations	100, 150, 200, ..., 1000	19

TABLE 4: Optimal values of training parameters.

Parameter	Optimal value
Activation function	Tansig
Nodes in hidden layer	40
Training iterations	700

The terminal value of $\mathbf{M}_{\text{cmdseq}}$ is generated when the loop exits. The sample output \mathbf{S}_{out} is obtained by

$$\mathbf{S}_{\text{out}} = \mathbf{M}_{\text{cmdseq}(1)}, \quad (21)$$

where $\mathbf{M}_{\text{cmdseq}(1)}$ is the first column of matrix $\mathbf{M}_{\text{cmdseq}}$.

6. Learning the Optimal Control Strategy

In this section, the neural network, the normalization module, and the antinormalization module in the attitude controller are established, which are the three central parts shown in Figure 2. The three modules can be modeled as function N . The algorithms of normalization and antinormalization can be found in ref. [22].

$$\mathbf{M}_z = N([\omega_d, \dot{\omega}_d, \mathbf{A}_{bd}, \omega_e]^T). \quad (22)$$

6.1. Division and Normalization of the Sample Set. The whole sample set is divided into two parts, which are the training sample set and the test sample set. 75% of the samples are randomly selected as training samples, and the others are test samples.

The training sample set consists of two matrices. One is the sample input matrix $\mathbf{I}_{\text{train}} \in \mathbb{R}^{12 \times N_{\text{train}}}$ and the other is the corresponding sample output matrix $\mathbf{O}_{\text{train}} \in \mathbb{R}^{3 \times N_{\text{train}}}$. N_{train} is the number of samples in the training sample set. Each row of the two matrices has been normalized to [-1,1] before the training sample set is applied to training neural networks.

The test sample set also consists of two matrices. One is the sample input matrix $\mathbf{I}_{\text{test}} \in \mathbb{R}^{12 \times N_{\text{test}}}$, and the other is the corresponding sample output matrix $\mathbf{O}_{\text{test}} \in \mathbb{R}^{3 \times N_{\text{test}}}$. N_{test} is the number of samples in the test sample set. Each row of the two matrices has been normalized to [-1,1] before the test sample set is applied to testing the neural networks.

6.2. Training the Neural Networks. The structure of the neural network is inputlayer-feedforwardlayer-feedforwardlayer. The stochastic gradient descent (SGD) algorithm is applied

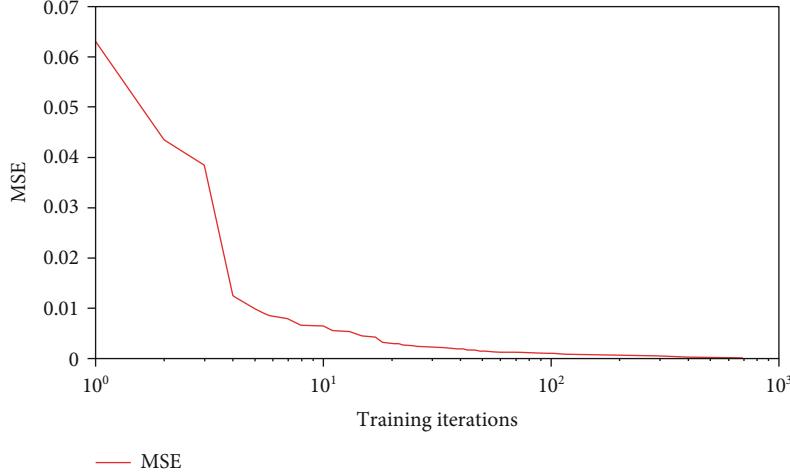


FIGURE 4: Training process of the neural network with the minimum MSE.

to the training process. In order to reduce overfitting, dropout (the fraction of dropout is set to 0.1) and weight decay (the regularization parameter is set to 0.01) are also applied.

Different network structure parameters and training parameters are tested to achieve the best performance. The values of parameters are selected in a linear method and are shown in Table 3. All parameter combinations are used in training, and 3876 neural networks ($3876 = 4 \times 51 \times 19$) are obtained.

6.3. Evaluating the Neural Networks. In the process of evaluating a neural network, the output matrix $\hat{\mathbf{O}}_{\text{test}} \in \mathbb{R}^{3 \times N_{\text{test}}}$ is obtained by

$$\hat{\mathbf{O}}_{\text{test}(j)} = N(\mathbf{I}_{\text{test}(j)}) \quad (j = 1, 2, \dots, N_{\text{test}}), \quad (23)$$

where $\mathbf{I}_{\text{test}(j)} \in \mathbb{R}^{12 \times 1}$ is the column j in matrix $\mathbf{I}_{\text{test}} \in \mathbb{R}^{12 \times N_{\text{test}}}$ and $\hat{\mathbf{O}}_{\text{test}(j)} \in \mathbb{R}^{3 \times 1}$ is the column j in matrix $\hat{\mathbf{O}}_{\text{test}} \in \mathbb{R}^{3 \times N_{\text{test}}}$. Define that $O_{\text{test}(i,j)}$ is the element in row i and column j of matrix \mathbf{O}_{test} , and that $\hat{O}_{\text{test}(i,j)}$ is the element in row i , and column j of matrix $\hat{\mathbf{O}}_{\text{test}}$. MSE (Mean Squared Error) is defined as

$$MSE = \sum_{i=1}^3 \sum_{j=1}^{N_{\text{test}}} (O_{\text{test}(i,j)} - \hat{O}_{\text{test}(i,j)})^2. \quad (24)$$

Finally, the neural network with the smallest MSE (0.00011686) is obtained. Its parameters used in training are shown in Table 4, and its training process is shown in Figure 4.

7. Results and Discussion

7.1. Simulation Condition Design. The satellite orbit parameters at the initial time are shown in Table 5. The initial time is 2017-05-31-04-40-00 (in yyyy-MM-dd-HH-mm-ss format).

As is shown in Figure 5, 100 ground targets are randomly selected in the circle with the center at 120°E 30°N and a

TABLE 5: Orbit parameters.

Parameter	Value
Semimajor axis	6878.14 km
Eccentricity	0
Inclination	97.3°
Ascending node	255°
Argument of perigee	0°
True anomaly	145°

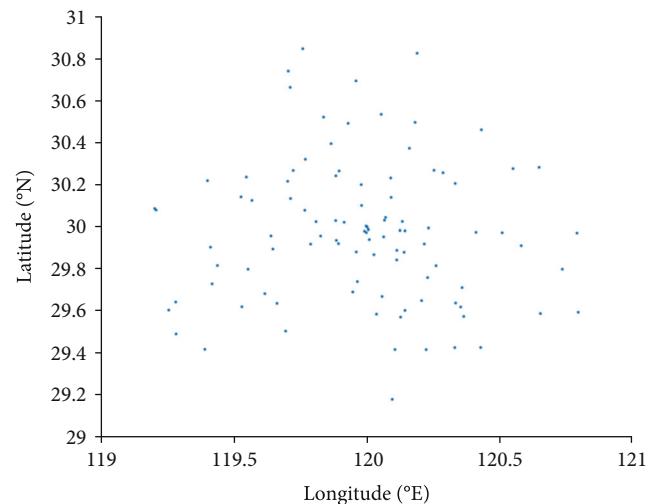


FIGURE 5: Ground targets randomly selected in the circular range.

radius of 100 km. After all the ground targets are tested, the average performance \bar{J} is obtained by

$$J = \frac{1}{N_{\text{target}}} \sum_{i=1}^{N_{\text{target}}} J_i, \quad (25)$$

where $N_{\text{target}} = 100$ and J is the performance defined in Eq. (8) of a single test.

TABLE 6: Other simulation parameters.

Parameter	Value
Satellite mass	1000 kg
Inertia tensor	$I_x = 797.34, I_y = 304.86, I_z = 792.48, I_{xy} = 14.3, I_{xz} = 25.1, I_{yz} = 10.5 \text{ (kgm}^2)$
Star sensor constant error	3"
Star sensor random error (3σ)	3"
Gyro constant error	0.15°/h
Gyro random error (1σ)	0.15°/h
Reaction wheels constant error	0.01 nm
Reaction wheels random error (1σ)	0.01 nm
External disturbance	Gravitational torque, aerodynamic torque, solar radiation torque and magnetic torque. (modeled according to ref. [23])
Initial Euler angle	Equal to the desired initial Euler angle for each ground target
Initial angular velocity	Equal to the desired initial angular velocity for each ground target

TABLE 7: Combinations of coefficient in PID control.

Coefficient	Values	Total number of values
Proportional	1, 2, 3, ..., 60	60
Integral	0, 0.01, 0.02, 0.03	4
Differential	40, 41, 42, ..., 90	51

TABLE 8: Performances of staring at the central ground target.

Controller	J_A	J_ω	\bar{J} ($k_1 = 1$ and $k_2 = 5$)
PID controller	3.96×10^{-4}	4.23×10^{-5} /s	6.07×10^{-4}
Neural network controller	2.13×10^{-4}	3.81×10^{-5} /s	4.03×10^{-4}

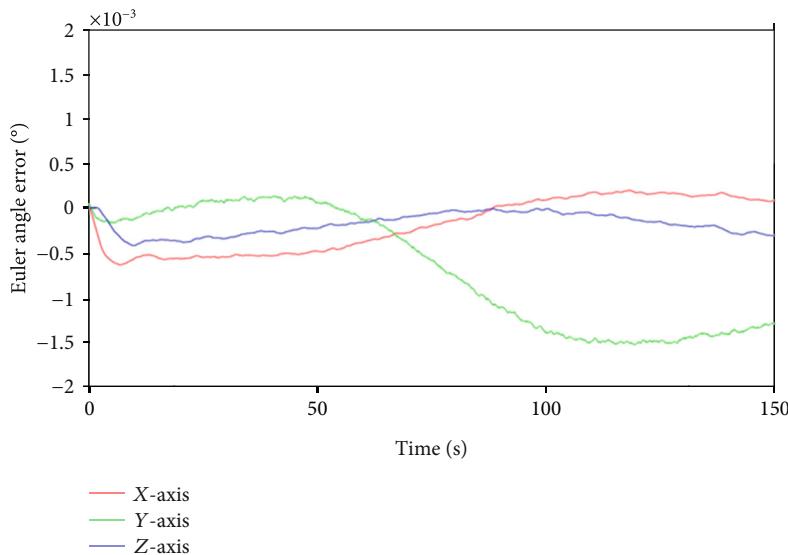


FIGURE 6: Euler angle error of the PID controller with the optimal coefficient combination.

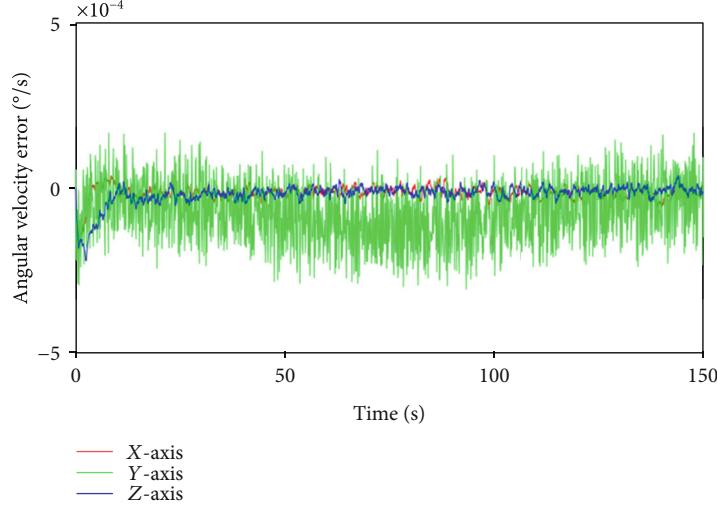


FIGURE 7: Angular velocity error of the PID controller with the optimal coefficient combination.

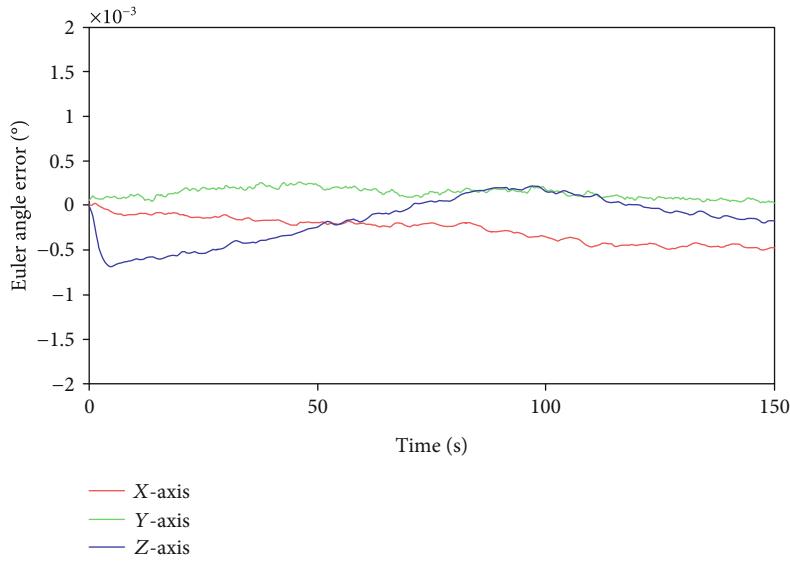


FIGURE 8: Euler angle error under the control of neural network.

The average pointing accuracy \bar{J}_A is obtained by

$$J_A = \frac{1}{N_{\text{target}}} \sum_{i=1}^{N_{\text{target}}} J_A. \quad (26)$$

The average pointing stability \bar{J}_ω is obtained by

$$J_\omega = \frac{1}{N_{\text{target}}} \sum_{i=1}^{N_{\text{target}}} J_\omega. \quad (27)$$

Other simulation parameters are defined in Table 6.

7.2. Searching the Optimal Coefficients for the PID Controller. Different combinations of coefficients in the PID controller are tested to achieve its best performance as the baseline for comparison. The coefficients of each axis are optimized, respectively, using the same combinations ($12240 = 60 \times 4 \times 51$)

defined in Table 7. The Euler angles and angular velocities used in the PID control law are expressed in degrees.

All the ground targets defined in Figure 5 are tested for each coefficient combination, and the average performance \bar{J} is calculated. Then, the optimal PID coefficient combination is obtained according to Eq. (25) ($k_1 = 1$ and $k_2 = 5$, in order to balance the pointing accuracy and the pointing stability).

7.3. Performance Comparison of Steady-State Control. The comparison of steady-state control performance between the PID controller and the neural network controller consists of two parts. The first part is testing the condition in which the satellite stares at the central ground target (at 120°E 30°N). Curves of the Euler angle error and angular velocity error are provided. The second part includes testing all the ground targets shown in Figure 5 and calculating the average performance \bar{J} .

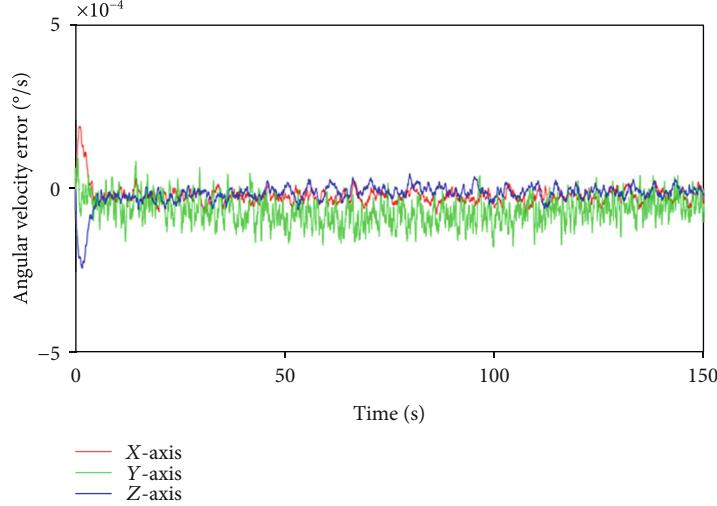


FIGURE 9: Angular velocity error under the control of neural network.

TABLE 9: Performances of staring at all the ground targets.

Controller	\bar{J}_A	\bar{J}_ω	\bar{J} ($k_1 = 1$ and $k_2 = 5$)
PID controller	3.96×10^{-4}	$4.25 \times 10^{-5} / \text{s}$	6.09×10^{-4}
Neural network controller	2.13×10^{-4}	$3.76 \times 10^{-5} / \text{s}$	4.01×10^{-4}

7.4. Testing the Central Ground Target. The performances are tested for the neural network controller and the PID controller with the optimal coefficient combination when the satellite stares at the central ground target. The neural network controller performs better than the PID controller according to the results shown in Table 8 and the error curves.

Error curves of the PID controller are shown in Figure 6 and Figure 7.

Error curves under the control of neural network are shown in Figures 8 and 9.

7.5. Testing all the Ground Targets. All the ground targets defined in Figure 5 are tested for the neural network controller and the PID controller with the best coefficient combination. The neural network controller performs better according to the results shown in Table 9.

Figure 10 is provided to visualize the performances and excludes the influence of k_1 and k_2 . The x -axis and y -axis denote the pointing accuracy J_A and the pointing stability J_ω , respectively. The blue points represent the performances of the PID controller on all the ground targets, and the red points represent the performances of the neural network controller.

7.6. Performance Comparison of the Dynamic Response

7.6.1. Step Response. In order to test the step response, two simulations are executed for each controller and each ground target. One is the normal simulation, and the other is the simulation in which an extra angle measurement error (0.005° to three axes) is added at 30s and later. According to the difference between the two Euler angle sequences, the overshoot,

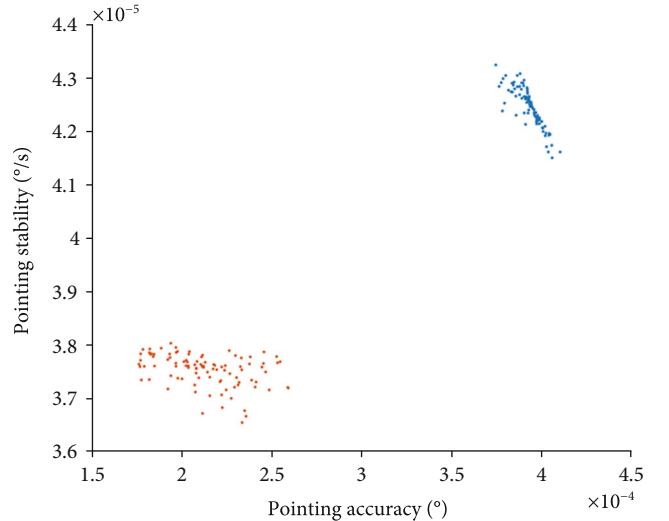


FIGURE 10: Performance comparison between the PID controller and the neural network controller.

settling time (the error band is $\pm 5\%$), and steady-state error are obtained.

The average values of overshoot, settling time, and steady-state error for all the ground targets are shown in Table 10 (PID controller) and Table 11 (neural network controller). The Euler angle difference of the two simulations (the satellite stares at $120^\circ\text{E } 30^\circ\text{N}$) are provided in Figure 11 (PID controller) and Figure 12 (neural network controller). The neural network controller shows an advantage over the PID controller on all the three indices (in column “Average of Three Axes”) according to the results.

TABLE 10: Average step response performance of the PID controller for all the ground targets.

Index	<i>x</i> -axis	<i>y</i> -axis	<i>z</i> -axis	Average of three axes
Overshoot	13.49%	0.97%	1.12%	5.25%
Settling time	10.33 s	3.67 s	8.06 s	7.35 s
Steady-state error	7.34×10^{-6}	4.19×10^{-7}	2.95×10^{-4}	1.24×10^{-5}

TABLE 11: Average step response performance of the neural network controller for all the ground targets.

Performance	<i>x</i> -axis	<i>y</i> -axis	<i>z</i> -axis	Average of three axes
Overshoot	2.80%	0.001%	3.56%	2.12%
Settling time	3.07 s	3.42 s	2.99 s	3.16 s
Steady-state error	7.13×10^{-6}	5.38×10^{-8}	1.33×10^{-5}	6.83×10^{-6}

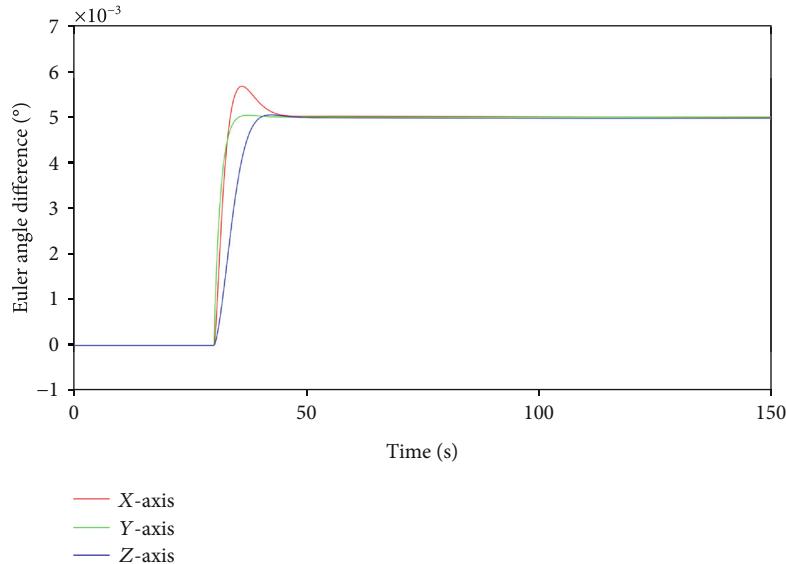


FIGURE 11: Step response of the PID controller when the satellite stares at the central ground target.

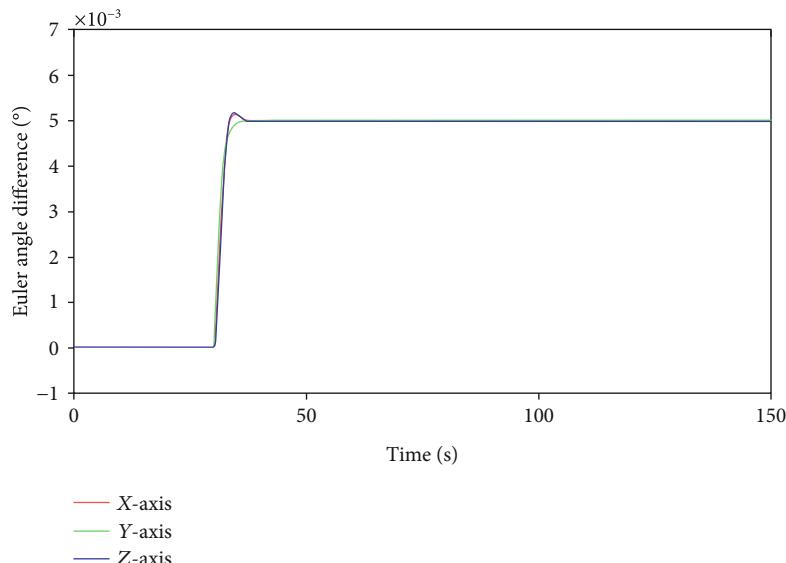


FIGURE 12: Step response of the neural network controller when the satellite stares at the central ground target.

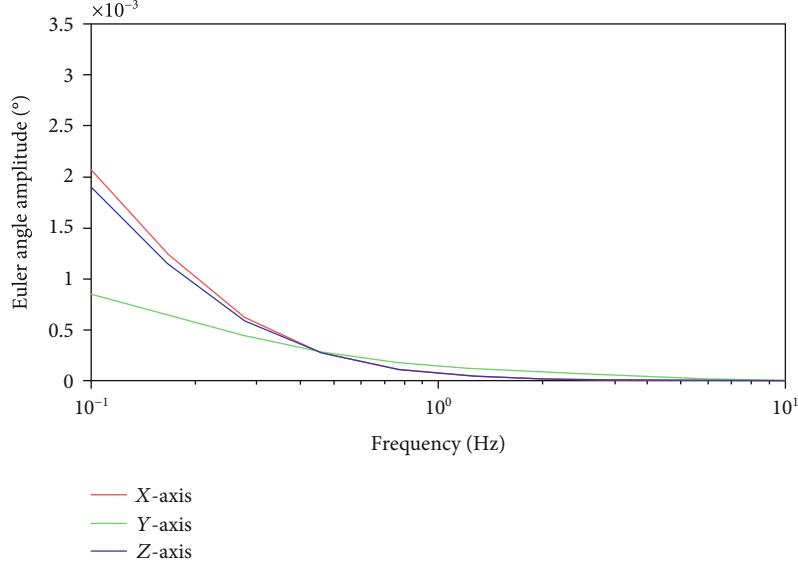


FIGURE 13: Magnitude-frequency characteristic of the PID controller.

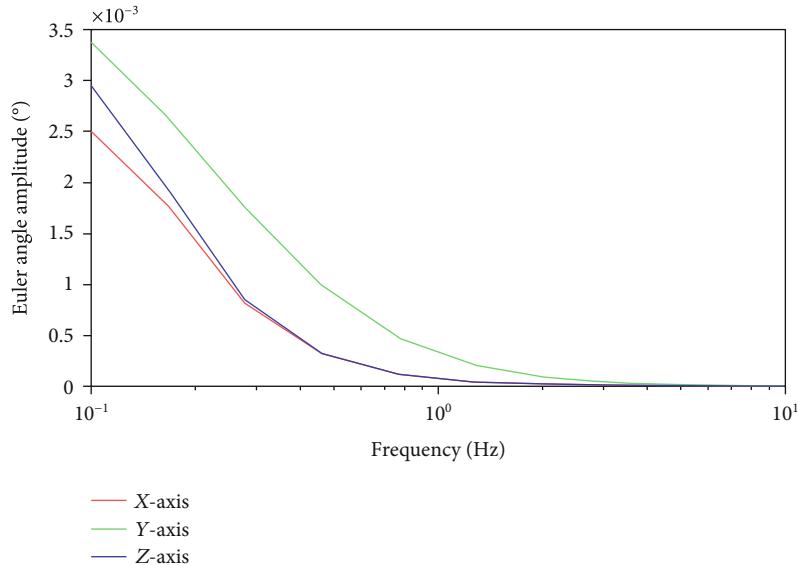


FIGURE 14: Magnitude-frequency characteristic of the neural network controller.

7.7. Frequency Response. In order to test the frequency response, external periodical torque is applied in the simulation in which the satellite stares at the central ground target. The amplitude of the external periodical torque is 0.03 Nm, and the response of three axes is tested, respectively. Magnitude-frequency characteristic curves of the two controllers are shown in Figure 13 (PID controller) and Figure 14 (neural network controller). The amplitude of the neural network controller is higher than that of the PID controller when the frequency is lower than 2 Hz, and the amplitudes are similar when the frequency is higher.

7.8. Impulse Response. A big disturbance is applied to the system under the control of the neural network in order to test the stability of the system (the satellite stares at 120°E

30°N). The disturbance starts at 30s and lasts for 0.5 s, and its amplitudes for the three axes are [1.0, 1.0, 1.0] Nm. Figure 15 shows that though the Euler angle error has exceed the input range (defined in Table 2) of the neural network, and the PID control law in the proposed attitude controller (in Figure 2) manages to converge and gives the control back to the neural network.

7.9. Testing the Real-Time Control Ability. It takes an average of 6.1 seconds to create a sample using CPU i7-8700. As a result, applying the loop of optimization (defined in Figure 3) to the satellite attitude controller does not meet the real-time control requirement (e.g., the control period is usually set to 0.1 s). The calculation time of the whole neural network controller using ARM (STM32F417) is 3.3 ms

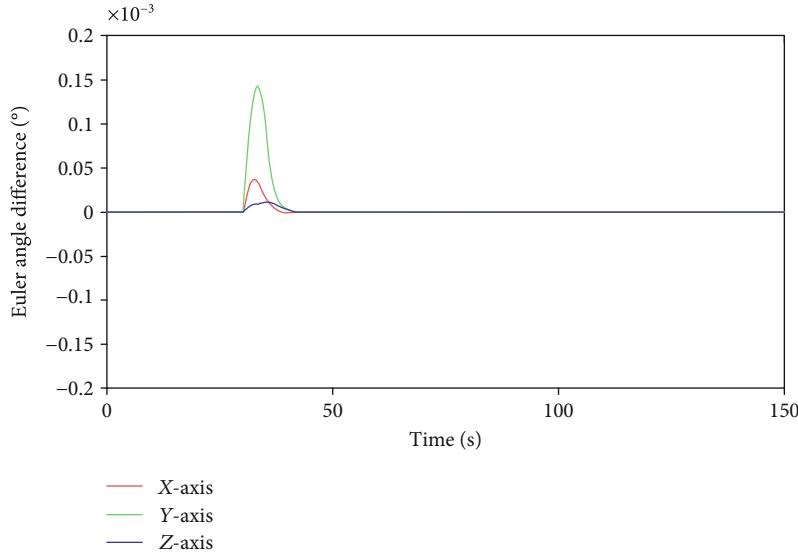


FIGURE 15: Impulse response of the neural network controller.

(3.1 ms in the process of neural network forward propagation, 0.2 ms in other modules), which meets the requirement of real-time control.

8. Conclusions

In this paper, the optimal control and machine learning method are applied unitedly in the attitude control of staring imaging. Compared with the PID controller with the best coefficients, the neural network controller shows advantages in the steady-state performance (including pointing accuracy and pointing stability), which is the most important in staring imaging tasks. As for the dynamic response, the neural network controller performs better in the step response and worse in the frequency response than the PID controller. The proposed controller also shows the ability to converge when a big disturbance is applied.

The calculation time of the proposed attitude controller on ARM (STM32F417) is 3.3 ms, which meets the requirement of real-time control.

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. This research was funded by the National Key Research and Development Program of China (No. 2016YFB0501102) and the Key Labora-

tory of Spacecraft Design Optimization and Dynamic Simulation Technologies, Ministry of Education.

References

- [1] Y. Lian, Y. Gao, and G. Zeng, "Staring imaging attitude control of small satellites," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 5, pp. 1278–1285, 2017.
- [2] X.-y. Zhang and J.-h. Xiang, "Tracking imaging feedback attitude control of video satellite," *ELECTRICAL ENGINEERING AND AUTOMATION: Proceedings of the International Conference on Electrical Engineering and Automation (EEA2016)*, 2017.
- [3] K.-k. Cui and J.-h. Xiang, "Variable coefficients pd adaptive attitude control of video satellite for ground multi-object staring imaging," *ELECTRICAL ENGINEERING AND AUTOMATION: Proceedings of the International Conference on Electrical Engineering and Automation (EEA2016)*, 2017.
- [4] X. Zhang, J. Xiang, and Y. Zhang, "Tracking imaging attitude control of video satellite for cooperative space object," in *2016 IEEE Advanced Information Management, Communications, Electronic and Automation Control Conference (IMCEC)*, IEEE, Xi'an, China, 2016.
- [5] Y.-H. Wu, F. Han, M.-H. Zheng et al., "Attitude tracking control for a space moving target with high dynamic performance using hybrid actuator," *Aerospace Science and Technology*, vol. 78, pp. 102–117, 2018.
- [6] L. Felicetti and M. Reza Emami, "Image-based attitude maneuvers for space debris tracking," *Aerospace Science and Technology*, vol. 76, pp. 58–71, 2018.
- [7] Q. Hu, X. Shao, and L. Guo, "Adaptive fault-tolerant attitude tracking control of spacecraft with prescribed performance," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 1, pp. 331–341, 2017.
- [8] K. Lu and Y. Xia, "Adaptive attitude tracking control for rigid spacecraft with finite-time convergence," *Automatica*, vol. 49, no. 12, pp. 3591–3599, 2013.
- [9] C. M. Pong and D. W. Miller, "Reduced-attitude boresight guidance and control on spacecraft for pointing, tracking,

- and searching,” *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 6, pp. 1027–1035, 2015.
- [10] B. Wu, “High precision satellite attitude tracking control via iterative learning control,” *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 3, pp. 528–534, 2015.
 - [11] A. Cassioli, D. Di Lorenzo, M. Locatelli, F. Schoen, and M. Sciandrone, “Machine learning for global optimization,” *Computational Optimization and Applications*, vol. 51, no. 1, pp. 279–303, 2012.
 - [12] K. Basu, R. G. Melton, and S. Aguasvivas-Manzano, “Time-optimal reorientation using neural network and particle swarm formulation,” *Proceedings of 2017 AAS/AIAA Astrodynamics Specialist Conference*, 2017.
 - [13] D. Izzo, C. Sprague, and D. Tailor, “Machine learning and evolutionary techniques in interplanetary trajectory design,” 2018, <http://arxiv.org/abs/1802.00180>.
 - [14] R. Furfaro, I. Bloise, M. Orlandelli, P. Di Lizia, F. Topputo, and R. Linares, “A Recurrent Deep Architecture for Quasi-Optimal Feedback Guidance in Planetary Landing,” *Proceedings of IAA SciTech Forum on Space Flight Mechanics and Space Structures and Materials*, pp. 1–24, 2018.
 - [15] A.-M. Zou, K. D. Kumar, Z. Hou, and X. Li, “Finite-time attitude tracking control for spacecraft using terminal sliding mode and Chebyshev neural network,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 4, pp. 950–963, 2011.
 - [16] 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.
 - [17] R. Furfaro, I. Bloise, M. Orlandelli, P. Di Lizia, F. Topputo, and R. Linares, “Deep learning for autonomous lunar landing,” *Proceedings of 2018 AAS/AIAA Astrodynamics Specialist Conference*, 2018.
 - [18] B. Wie, “Space Vehicle Dynamics and Control,” *AIAA Education Series*, 1998.
 - [19] ESA, *ESA Pointing Error Handbook*, ESA-NCR-502, 1993.
 - [20] ESA, *ESA pointing error engineering handbook*, ESSB-HB-E-003, 2011.
 - [21] M. J. Sidi, *Spacecraft dynamics and control: a practical engineering approach*. Vol. 7, Cambridge university press, 1997.
 - [22] P. Harrington, *Machine Learning in Action*, Manning Publications Co., 2012.
 - [23] S. A. Ibrahim and E. Yamaguchi, “Comparison of solar radiation torque and power generation of deployable solar panel configurations on nanosatellites,” *Aerospace* 6.5, p. 50, 2019.