International Journal of Innovative
Computing, Information and Control
Volume **17**, Number **4**, August **2021**

ICIC International ©2021 ISSN 1349-4198
pp. 1447–1464

ONLINE-OFFLINE INTEGRATED LEARNING METHOD OF NEURAL NETWORK CONTROL FOR A SERVO SYSTEM

Masakazu Morita, Qingjiu Huang* and Mimpei Morishita

Electrical Engineering and Electronics Program
Graduate School of Engineering
Kogakuin University
1-24-2 Nishi-shinjuku, Shinjuku-ku, Tokyo 163-8677, Japan
cd18002@ns.kogakuin.ac.jp; morisita@cc.kogakuin.ac.jp
*Corresponding author: huang@cc.kogakuin.ac.jp

Received March 2021; revised May 2021

Abstract. *Neural network control is attracting attention as the high-precision position control for servo systems such as robots and numerical control machines. In this study, in order to achieve real-time control and deal with load fluctuations and disturbances for position control of a servo system, we proposed an online-offline integrated learning method for neural network control, which combines the advantages of the offline learning method with excellent responsiveness and the advantages of the online learning method with excellent robustness. First, the weights of the neural network obtained by enough online learning are loaded as standard weights, and the neural network control is performed by the offline learning method. Next, if the feedback error does not exceed the threshold value, the offline learning method continues. If the feedback error exceeds the threshold value, the online learning method with few repeating loops is performed within the sampling time. Moreover, we simulated the position control before and after changing the load for a servo motor, and clarified the effectiveness of the proposed method from the simulation result.*

Keywords: Neural network control, Online-offline integrated learning, Adaptive control, Real-time control, Servo system

1. **Introduction.** Neural network control is one kind of adaptive control and nonlinear control [1-5]. In addition, since neural network control can obtain the dynamic characteristics of the controlled object through learning, it is expected to be high-precision position control of servo systems such as robots and numerical control machines [6-12]. Until now, there are two learning methods for neural network control, which are the online learning and the offline learning. According to the results and considerations of the simulation for the position control of the servo motor in this paper, the online learning method constantly learns based on feedback error, so it can deal with load fluctuations and disturbances. However, since its settling time is long, it is not suitable for real-time control. On the other hand, the offline learning method uses a neural network obtained by enough online learning as a feedforward compensator, and the enormous amount of calculation is eliminated, so that the responsiveness of the control system is fast. However, since it does not constantly learn based on the feedback error, it cannot deal with load fluctuations and disturbances.

For the above background, in this study, in order to deal with load fluctuations and disturbances, and to realize real-time control, we proposed an online-offline integrated learning method of neural network control for the position control of a servo system, in

DOI: 10.24507/ijicic.17.04.1447

which the offline learning is performed if the feedback error does not exceed the threshold value, and the online learning with few repeating loops within the sampling time is performed if the feedback error exceeds the threshold value.

Specifically, first, the weights of the neural network obtained by enough online learning are loaded as standard weights, and the neural network control is performed by the offline learning method. Next, if the feedback error does not exceed the threshold value, the offline learning method continues. If the feedback error exceeds the threshold value, the online learning method with few repeating loops is performed within the sampling time. At that time, the learning efficiency is higher and the settling time is shorter than the online learning method performed only once within the sampling time. Further, when the feedback error converges to the threshold value, the method returns to the offline learning method, and the excellent responsiveness of the offline learning can be utilized. Therefore, when viewed from the overall control response, the response becomes faster and real-time control becomes possible. Moreover, when the feedback error exceeds the threshold value, the online learning method with few repeating loops works within the sampling time, so that excellent robustness can be obtained, and even if there is load fluctuation and disturbance, the control system is less likely to affect it. Therefore, for the neural network control, the proposed method can integrate the excellent responsiveness of the offline learning method with the excellent robustness of the online learning method, so it can deal with the load fluctuations and disturbances, and realize the real-time control.

Furthermore, we use the structure of neural network control based on the feedback error learning [13-15] and simulate a position control of a Direct Current (DC) servo motor. In this simulation, before and after changing the load inertia of the servomotor, the online learning method, the offline learning method, and the proposed method respectively are applied to the neural network control of a servo system, and the effectiveness of the proposed method is determined by the results and considerations of the simulation.

The structure of this paper is as follows. Chapter 2 describes the configuration of the servo system using neural network control, which is dealt with in this paper. Chapter 3 explains the algorithms of online learning and the offline learning of the conventional method, and explains the algorithm of the online-offline integrated learning method proposed in this study. Chapter 4 presents an analysis environment using a dynamic model of position control for servo motor. Chapter 5 shows the results of the simulation and summarizes the considerations. Chapter 6 presents the conclusions of this study.

2. **Structure of the Servo System Using Neural Network Control.** In this study, Figure 1 shows the structure of the servo system with the neural network control based on the feedback error learning method [14].

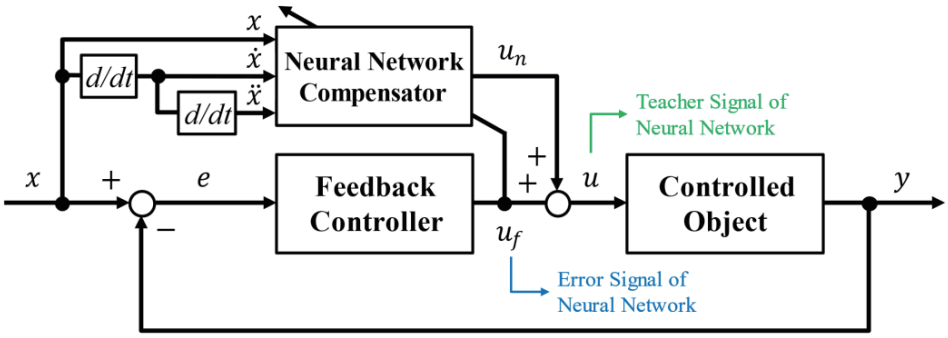


FIGURE 1. Structure of the servo system using neural network control

The explanation of the variables in Figure 1 is as follows:
x: Reference value, also the input to the feedback control system.
e: Tracking error, also the feedback error.
u_f: Output of the feedback controller, also the error signal of the neural network.
u_n: Output of the neural network.
u: Control input for the controlled object, also the teacher signal of the neural network.
y: Controlled variable, also the output to the feedback control system.

This controller in the structure is a feedback controller connected in parallel with a feedforward controller composed of a neural network. In this neural network, in order to obtain the inverse dynamic model of the controlled object with the learning, the input u for the controlled object is treated as a teacher signal. Therefore, the difference between the teacher signal u and the output signal u_n is the error of the neural network and is equal to the output u_f of the feedback controller. Thus, the neural network works to match the output u_n with the teacher signal u through the learning, obtains the inverse dynamic model of the controlled object, and compensates the feedback controller as a feedforward controller.

3. **Online-Offline Integrated Learning Method of the Neural Network Compensator.** In this section, first, we will describe the algorithm for feedback error learning using the backpropagation method in Section 3.1. Next, we will explain the algorithm of the conventional method of online learning and the offline learning in Sections 3.2 and 3.3. Also, we will explain the algorithm of the proposed online-offline integrated learning method by this study in Section 3.4. Finally, we will give a summary of the comparison between the conventional method and the proposed method in Section 3.5.

3.1. **Feedback error learning using the backpropagation method.** Figure 2 shows the structure of the neural network used in this study. $x(i)$, $y(j)$, $z(k)$ are functions indicating neurons in an input layer, a middle layer, and an output layer. $v(j,k)$ is weight from the middle layer to the output layer, and $w(i,j)$ is weight from the input layer to the middle layer. a, b, c are respectively the total number of neurons in the input layer, the middle layer, and the output layer, and i, j, k are respectively the numbers of neurons in the input layer, the middle layer, and the output layer.

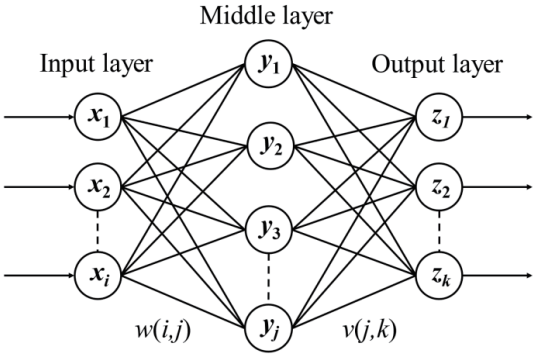


FIGURE 2. Structure of the neural network

In this study, the backpropagation method is used to update the weights between the input layer and the middle layer and the weights between the middle layer and output layer so as to minimize the error of the neural network. Specifically, by minimizing the error E_n between the output u_n of the neural network and the control input u of the controlled object which is used as the teacher signal of the neural network, using the

steepest descent method as shown in Equations (2)-(9), the weights $v(j,k)$ and $w(i,j)$ are updated by the order from the output layer to the middle layer and from the middle layer to the input layer, so that the correct output u_n is finally obtained. Then, in the feedback error learning, control is performed based on this backpropagation method.

Firstly, Equation (1) shows the error signal E_n. In Equation (2), E_n is the partial derivative with respect to the output $u_n(k)$ of the neural network. Equation (3) is the partial derivative of $u_n(k)$ with respect to $v(j,k)$, and Equation (4) shows that the coefficient of $y(j)$ on the right side of Equation (3) is represented by $u_n(k)$. Next, in Equation (5), $u_n(k)$ is the partial derivative with respect to $y(j)$. Equation (6) is the partial derivative of $y(j)$ with respect to $w(i,j)$, and Equation (7) shows that the coefficient of $x(i)$ on the right side of Equation (6) is represented by $y(j)$. In addition, Equations (8) and (9) are derived from Equations (2)-(7).

$$E_n = \frac{1}{2}\sum_{k=1}^{a}(u_n(k) - u(k))^2 = \frac{1}{2}\sum_{k=1}^{a}(-u_f(k))^2 \tag{1}$$

$$\frac{\partial E_n}{\partial u_n(k)} = u_n(k) - u(k) = -u_f(k) \tag{2}$$

$$\frac{\partial u_n(k)}{\partial v(j,k)} = \frac{-2\cdot(-y(j))\cdot e^{-\sum_{j=1}^{b} v(j,k)y(j)}}{\left(1+e^{-\sum_{j=1}^{b} v(j,k)y(j)}\right)^2} = \frac{2e^{-\sum_{j=1}^{b} v(j,k)y(j)}}{\left(1+e^{-\sum_{j=1}^{b} v(j,k)y(j)}\right)^2}\cdot y(j) \tag{3}$$

$$\frac{2e^{-\sum_{j=1}^{b} v(j,k)y(j)}}{\left(1+e^{-\sum_{j=1}^{b} v(j,k)y(j)}\right)^2} = \frac{1}{2}\left(1-\left(\frac{1-e^{-\sum_{j=1}^{b} v(j,k)y(j)}}{1+e^{-\sum_{j=1}^{b} v(j,k)y(j)}}\right)^2\right) = \frac{1}{2}\left(1-(u_n(k))^2\right) \tag{4}$$

$$\frac{\partial u_n(k)}{\partial y(j)} = \frac{-2\cdot(-v(j,k))\cdot e^{-\sum_{j=1}^{b} v(j,k)y(j)}}{\left(1+e^{-\sum_{j=1}^{b} v(j,k)y(j)}\right)^2} = \frac{2e^{-\sum_{j=1}^{b} v(j,k)y(j)}}{\left(1+e^{-\sum_{j=1}^{b} v(j,k)y(j)}\right)^2}\cdot v(j,k) \tag{5}$$

$$\frac{\partial y(j)}{\partial w(i,j)} = \frac{-2\cdot(-x(i))\cdot e^{-\sum_{i=1}^{c} w(i,j)x(i)}}{\left(1+e^{-\sum_{i=1}^{c} w(i,j)x(i)}\right)^2} = \frac{2e^{-\sum_{i=1}^{c} w(i,j)x(i)}}{\left(1+e^{-\sum_{i=1}^{c} w(i,j)x(i)}\right)^2}\cdot x(i) \tag{6}$$

$$\frac{2e^{-\sum_{i=1}^{c} w(i,j)x(i)}}{\left(1+e^{-\sum_{i=1}^{c} w(i,j)x(i)}\right)^2} = \frac{1}{2}\left(1-\left(\frac{1-e^{-\sum_{i=1}^{c} w(i,j)x(i)}}{1+e^{-\sum_{i=1}^{c} w(i,j)x(i)}}\right)^2\right) = \frac{1}{2}\left(1-(y(j))^2\right) \tag{7}$$

$$\frac{\partial E_n}{\partial v(j,k)} = \frac{\partial E_n}{\partial u_n(k)}\cdot\frac{\partial u_n(k)}{\partial v(j,k)} = -\frac{1}{2}(-u_f(k))\cdot\left(1-(u_n(k))^2\right)\cdot y(j) = \delta_1\cdot y(j) \tag{8}$$

$$\begin{aligned}\frac{\partial E_n}{\partial w(i,j)} &= \frac{\partial E_n}{\partial u_n(k)}\cdot\frac{\partial u_n(k)}{\partial y(j)}\cdot\frac{\partial y(j)}{\partial w(i,j)} \\ &= -(-u_f(k))\cdot\frac{1}{2}\left(1-(y(j))^2\right)\cdot v(j,k)\cdot\frac{1}{2}\left(1-(y(j))^2\right)\cdot x(i) \\ &= \delta_1\cdot v(j,k)\cdot\frac{1}{2}\left(1-(y(j))^2\right)\cdot x(i) = \delta_2\cdot x(i)\end{aligned} \tag{9}$$

3.2. **Online learning method.** Figure 3 shows the flowchart of the online learning method. $\Delta v(j,k)(N)$ is the differential of $v(j,k)(N)$, and $\Delta w(i,j)(N)$ is the differential of $w(i,j)(N)$. N is the number of times to calculate the weights $v(j,k)(N)$, $w(i,j)(N)$, η is the learning rate, and α is the inertia coefficient for adjusting the correction amount of η. Firstly, $\Delta v(j,k)(N)$, $v(j,k)(N)$, $w(i,j)(N)$, $y(j)$, $z(k)$ are initialized by a value of zero. Also, $\Delta w(i,j)(N)$ is initialized by values of random. Next, let the position x, the velocity $\dot{x}$, and the acceleration $\ddot{x}$ of reference value be inputs of the neural network. The

error signal E_n is obtained from the output $u_f(k)$ of the feedback controller. Moreover, $\Delta v(j,k)(N)$, $v(j,k)(N)$, $\Delta w(i,j)(N)$, $w(i,j)(N)$, $y(j)$, $z(k)$ are calculated by Equations (10)-(15). Finally, let $z(k)$ be the output $u_n(k)$ of the neural network.

$$\Delta v(j,k)(N) = -\eta \frac{\partial E}{\partial v(j,k)} + \alpha \Delta v(j,k)(N-1) \tag{10}$$

$$v(j,k)(N) = v(j,k)(N-1) + \Delta v(j,k)(N) \tag{11}$$

$$\Delta w(i,j)(N) = -\eta \frac{\partial E}{\partial w(i,j)} + \alpha \Delta w(i,j)(N-1) \tag{12}$$

$$w(i,j)(N) = w(i,j)(N-1) + \Delta w(i,j)(N) \tag{13}$$

$$y(j) = f_y(w(i,j)(N), x(i)) = \frac{2}{1+e^{-\sum_{i=1}^{c}(w(i,j)(N))x(i)}} - 1 \tag{14}$$

$$u_n(k) = z(k) = f_z(v(j,k)(N), y(j)) = \frac{2}{1+e^{-\sum_{j=1}^{b}(v(j,k)(N))y(j)}} - 1 \tag{15}$$

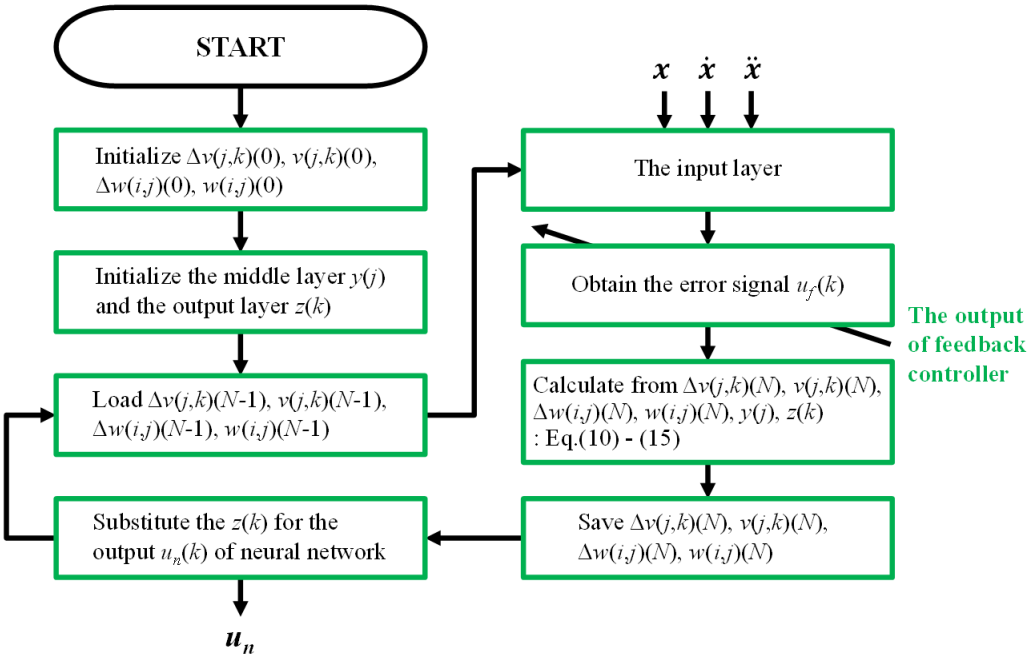


FIGURE 3. Flowchart of the online learning method

3.3. **Offline learning method.** Figure 4 shows the flowchart of the offline learning method. Firstly, the weights $v(j,k)$, $w(i,j)$ obtained by enough online learning are loaded into the neural network. Next, let the position x, the velocity $\dot{x}$, and the acceleration $\ddot{x}$ of reference value be inputs of the neural network. Moreover, $y(j)$ and $z(k)$ are calculated by Equations (16) and (17). Finally, let $z(k)$ be the output $u_n(k)$ of the neural network. In this learning method, the weights $v(j,k)$, $w(i,j)$ are not updated.

$$y(j) = f_y(w(i,j), x(i)) = \frac{2}{1+e^{-\sum_{i=1}^{c} w(i,j)x(i)}} - 1 \tag{16}$$

$$u_n(k) = z(k) = f_z(v(j,k), y(j)) = \frac{2}{1+e^{-\sum_{j=1}^{b} v(j,k)y(j)}} - 1 \tag{17}$$

3.4. **Proposed online-offline integrated learning method.** The flowchart of the proposed method is shown in Figure 5 and calculations are shown in Equations (18)-(26).

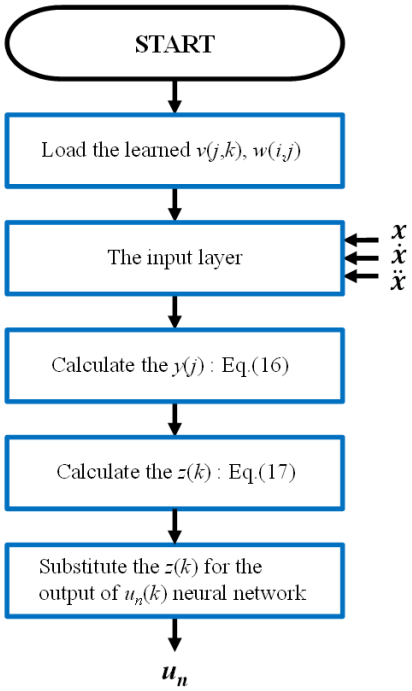


FIGURE 4. Flowchart of the offline learning method

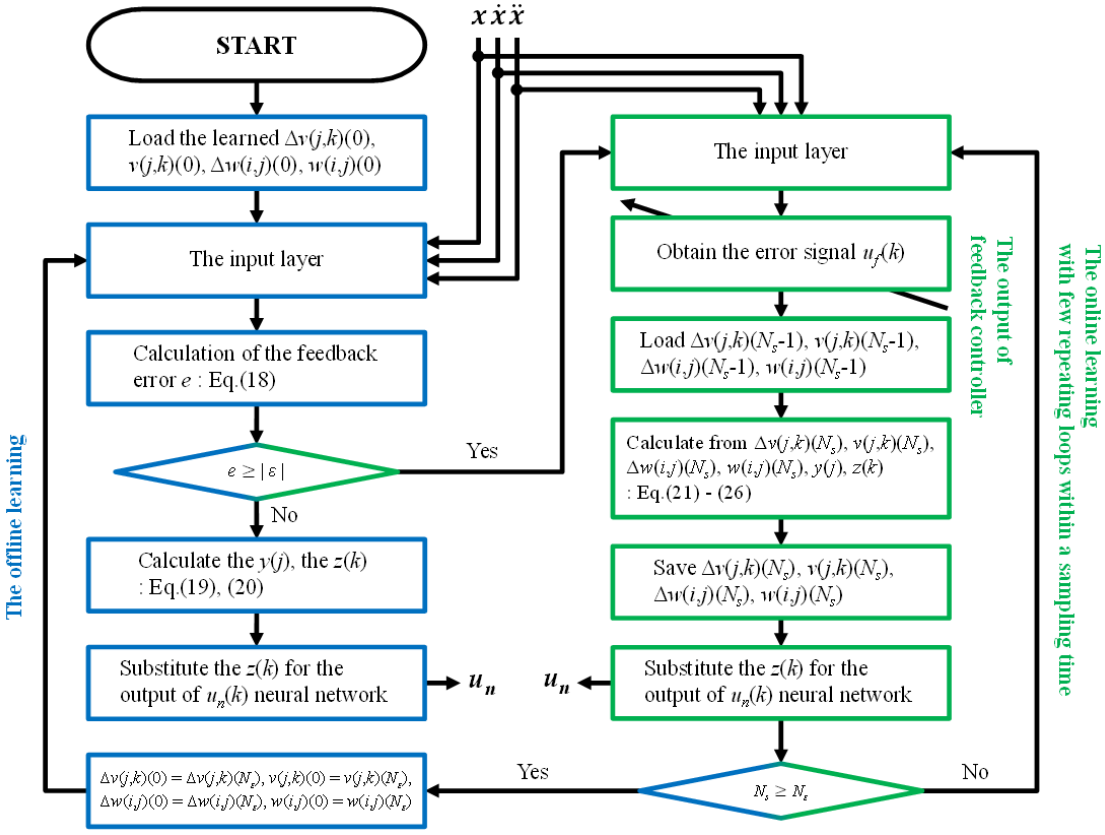


FIGURE 5. Flowchart of the proposed online-offline integrated learning method

First, the weights obtained by enough online learning is loaded into the neural network and set as $\Delta v(j,k)(0)$, $v(j,k)(0)$, $\Delta w(i,j)(0)$, $w(i,j)(0)$. However, according to the results of the simulation for the position control of the servo motor in this paper, if the weights $v(j,k)$ are initialized with a value of zero, better results will be obtained. Therefore, the weights $v(j,k)$ are initialized with a value of zero in this study. Moreover, let the position x, the velocity $\dot{x}$, and the acceleration $\ddot{x}$ of the reference value be the inputs of neural network. In addition, the error signal E_n is obtained from the output $u_f(k)$ of the feedback controller.

Next, the feedback error e is calculated by Equation (18). If the calculated error e does not exceed the threshold $|\varepsilon|$, the offline learning method is performed. And, $y(j)$ and $z(k)$ are calculated by Equations (19) and (20). And then, let $z(k)$ be the output $u_n(k)$ of the neural network. If the calculated error e exceeds the threshold $|\varepsilon|$, the online learning method is performed with few repeating loops within one sampling time. In each loop, the inputs of the neural network are given by the position x, the velocity $\dot{x}$, and the acceleration $\ddot{x}$ of reference value. In addition, the error signal E_n is obtained from the output $u_f(k)$ of the feedback controller. $\Delta v(j,k)(N_s-1)$, $v(j,k)(N_s-1)$, $\Delta w(i,j)(N_s-1)$, $w(i,j)(N_s-1)$ are loaded into the neural network, and $\Delta v(j,k)(N_s)$, $v(j,k)(N_s)$, $\Delta w(i,j)(N_s)$, $w(i,j)(N_s)$, $y(j)$, $z(k)$ are calculated by Equations (21)-(26). The value of $z(k)$ is given to $u_n(k)$. Next, if the number of repeating loops N_s within the sampling time does not reach the threshold N_ε, these calculations are repeated. If the number of repeating loops N_s within the sampling time reaches the threshold N_ε, the online learning method is terminated and the calculations return to the offline learning method. Then, let the obtained weights $\Delta v(j,k)(N_\varepsilon)$, $v(j,k)(N_\varepsilon)$, $\Delta w(i,j)(N_\varepsilon)$, $w(i,j)(N_\varepsilon)$ be the learned weights $\Delta v(j,k)(0)$, $v(j,k)(0)$, $\Delta w(i,j)(0)$, $w(i,j)(0)$ of the offline learning method.

For the neural network control, the proposed learning method as described above can integrate the excellent responsiveness of the offline learning method with the excellent robustness of the online learning method, so it can deal with the load fluctuations and disturbances, and realize the real-time control.

$$e = x - y \tag{18}$$

$$y(j) = f_y(w(i,j)(N_s), x(i)) = \frac{2}{1 + e^{-\sum_{i=1}^{c}(w(i,j)(N_s))x(i)}} - 1 \tag{19}$$

$$u_n(k) = z(k) = f_z(v(j,k)(N_s), y(j)) = \frac{2}{1 + e^{-\sum_{j=1}^{b}(v(j,k)(N_s))y(j)}} - 1 \tag{20}$$

$$\Delta v(j,k)(N_s) = -\eta \frac{\partial E_n}{\partial v(j,k)} + \alpha \Delta v(j,k)(N_s - 1) \tag{21}$$

$$v(j,k)(N_s) = v(j,k)(N_s - 1) + \Delta v(j,k)(N_s) \tag{22}$$

$$\Delta w(i,j)(N_s) = -\eta \frac{\partial E_n}{\partial w(i,j)} + \alpha \Delta w(i,j)(N_s - 1) \tag{23}$$

$$w(i,j)(N_s) = w(i,j)(N_s - 1) + \Delta w(i,j)(N_s) \tag{24}$$

$$y(j) = f_y(w(i,j)(N_s), x(i)) = \frac{2}{1 + e^{-\sum_{i=1}^{c}(w(i,j)(N_s))x(i)}} - 1 \tag{25}$$

$$u_n(k) = z(k) = f_z(v(j,k)(N_s), y(j)) = \frac{2}{1 + e^{-\sum_{j=1}^{b}(v(j,k)(N_s))y(j)}} - 1 \tag{26}$$

3.5. **Comparison of online learning, offline learning and proposed online-offline integrated learning.** Table 1 shows the comparison of three kinds of learning method.

In the online learning method, when load fluctuations and disturbances occur, those changes are reflected in the error signal E_n of neural network shown by Equation (1),

TABLE 1. Comparison of three kinds of learning method

	Online learning	Offline learning	Proposed online-offline integrated learning
Responsiveness	poor	excellent	excellent
Robustness	excellent	poor	excellent

so that the load fluctuations and disturbances can be dealt with and the robustness is excellent. However, since Equations (10)-(15) are calculated only once within the sampling time, the settling time is long and the responsiveness is poor. On the other hand, in the offline learning method, since only Equations (16) and (17) are calculated, the settling time is short and the responsiveness is excellent. However, since load fluctuations and disturbances are not reflected in learning, the robustness is poor.

In our proposed method, if the feedback error does not exceed the threshold value, the calculation with the less calculations of Equations (19) and (20) is performed as in the offline learning method, so that the responsiveness is excellent. In addition, if the feedback error exceeds the threshold value, an online learning method with few repeating loops is performed within the sampling time shown as in Equations (21)-(26), so that load fluctuations and disturbances can be dealt with and robustness is also excellent.

In Sections 4 and 5, we will prove this comparison of the three kinds of learning method using a simulation of the position control of the DC servo motor.

4. Simulation of the Position Control of the DC Servo Motor.

In this section, firstly, Section 4.1 explains the structure of the position control of the servo motor that is the controlled object, and shows the dynamic model of the control system. Next, Section 4.2 shows the analysis model of the control system built by MATLAB/Simulink in order to simulate the position control of the servo motor.

4.1. The structure and dynamic model of the servo system.

As shown in Figure 6, the structure of the position control of the servo motor as controlled object is connected in series from the DC servo motor, the coupling, and the DC motor as a load.

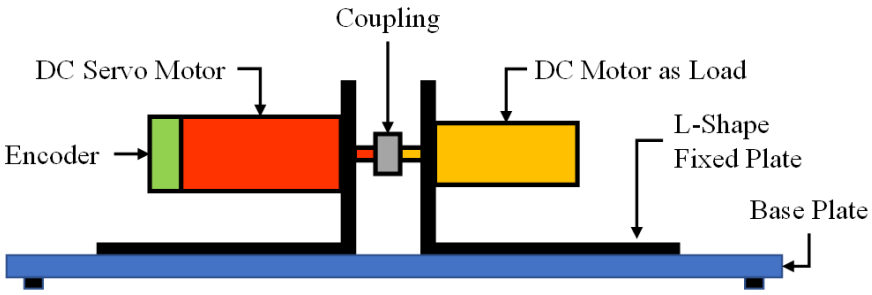


FIGURE 6. Structure of position control of DC servo motor

The structure of the servo control with neural network control is shown in Figure 7. Here, x, $\dot{x}$, $\ddot{x}$ shown in Figure 1 are treated as θ_r, $\dot{\theta}_r$, $\ddot{\theta}_r$. The dynamic model of motor shown in Figure 7 is a mathematical relationship between the armature voltage V and the rotation position θ based on the conversion of the electrical energy and the mechanical energy as shown in Equation (27). And then, the controller is the feedback controller connected in parallel with the feedforward controller composed of the neural network. In addition, the relationship between the armature voltage V and the rotational angular velocity $\dot{\theta}$ is expressed by Equation (28) using the relationship that the torque constant K_t

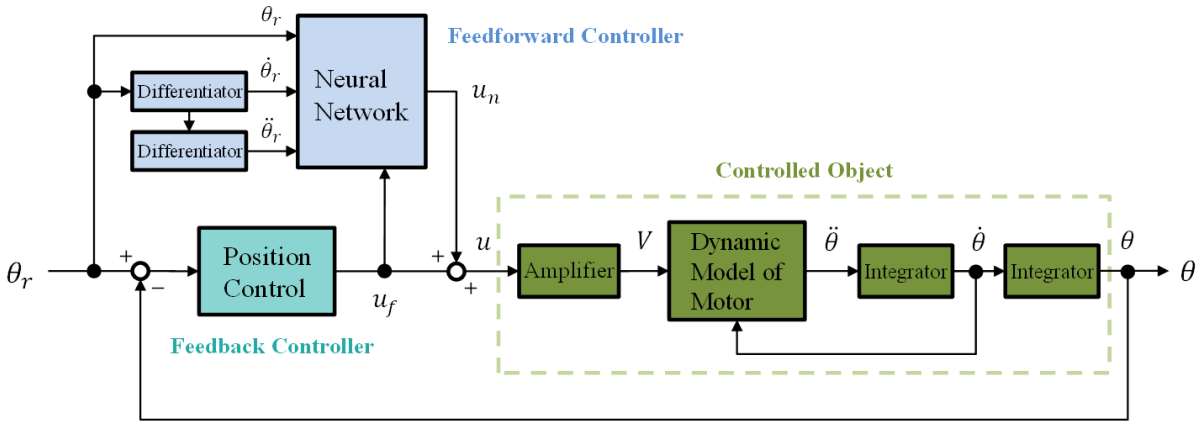


FIGURE 7. Structure of the servo system with neural network control

is equal to the counter electromotive force constant K_e. Moreover, the rotational angular acceleration $\ddot{\theta}$ is obtained by a combination of Equations (27) and (28) and is shown in Equation (29), where R is the terminal resistance phase to phase, I is the armature current, J is the inertia that combines the inertia inertia J_M of the motor and the load inertia J_L.

$$\tau = K_t I = J\frac{d^2\theta}{dt^2} = J\ddot{\theta} = (J_m + J_L)\ddot{\theta} \tag{27}$$

$$V = RI + K_e\frac{d\theta}{dt} = RI + K_e\dot{\theta} \tag{28}$$

$$\ddot{\theta} = \frac{d^2\theta}{dt^2} = \frac{K_t}{JR}\left(V - K_t\frac{d\theta}{dt}\right) = \frac{K_t}{JR}\left(V - K_t\dot{\theta}\right) \tag{29}$$

In the simulation of this study, the DC servo motor deals with Maxon's EC-max 22, and its parameters are shown in Table 2 below. The setting of the load inertia J_L is shown in Section 5.

TABLE 2. Parameters of brushless DC motor (EC-max 22)

Symbol	Name	Value
V_n	Nominal voltage	24 V
I_n	Nominal current	0.657 A
N_n	Nominal speed	8250 rpm
T_n	Nominal torque	10.8 mNm
R	Terminal resistance phase to phase	12.4 Ω
K_t	Torque constant	18.1 mNm/A
J_M	Rotor inertia	2.25×10^{-7} kgm^2

4.2. **Simulation model.** Figure 8 shows the simulation model structured in MATLAB/Simulink. In this model, the sampling time for analysis is 1 ms, and the analysis method for calculus is the 4th-order Runge-Kutta method. The block θ_r is the generator of the reference position, and generates the angular trajectory that is a sine wave with an amplitude of ±90 degrees, a frequency of 1 Hz, and a phase of 0 degrees. The feedback controller uses Proportional-Integral-Differential (PID) control. The gain of PID control is set with the value calculated by the pole placement method. The amplifier K is connected in front

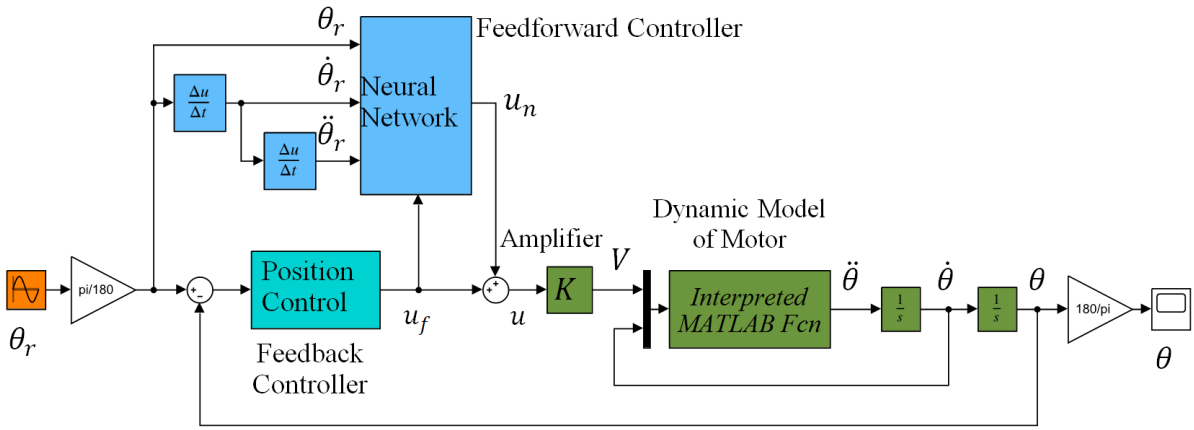


FIGURE 8. Simulation model of the servo system with neural network control

of the dynamic model of motor. And, θ_r, $\dot{\theta}_r$, $\ddot{\theta}_r$ are connected to the block of the neural network compensator. In addition, the angle θ and the angular velocity $\dot{\theta}$, which are the outputs of the motor, are obtained by integrating the angular acceleration $\ddot{\theta}$.

5. **Results of the Simulation.** In this section, the results of position control are shown in Sections 5.1 and 5.2. In Section 5.1, the online learning method is used to obtain the learned weights by enough online learning. In Section 5.2, it shows simulation results on the responsiveness and robustness of the position control using the online learning method, the offline learning method, and the proposed method before and after changing the load inertia. In Section 5.3, it shows the simulation results when the frequency of the reference trajectory is changed. In Section 5.4, it summarizes the results and considerations found from the simulation results.

5.1. **Obtaining the learned weights using the online learning method.** Figures 9 and 10 show the results of the rotation angle θ and the feedback error e when using PID control and the online learning method ($\eta = 0.004$, $\alpha = 0.001$), respectively. The load inertia value was set as $J_L = J_M = 2.25 \times 10^{-7}$ kgm^2. And, since the learning effect did not increase any more after 40 seconds, the weights obtained at 40 seconds were determined to be the learned weights.

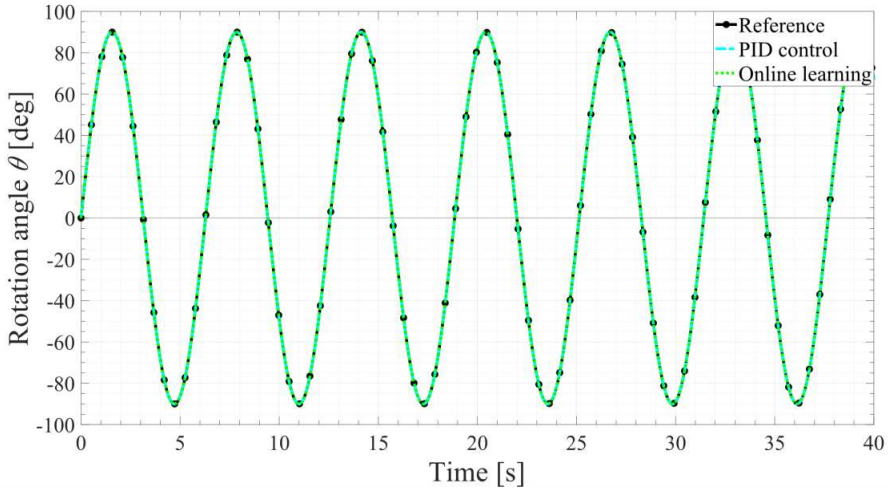


FIGURE 9. Responses of the PID control and the online learning method

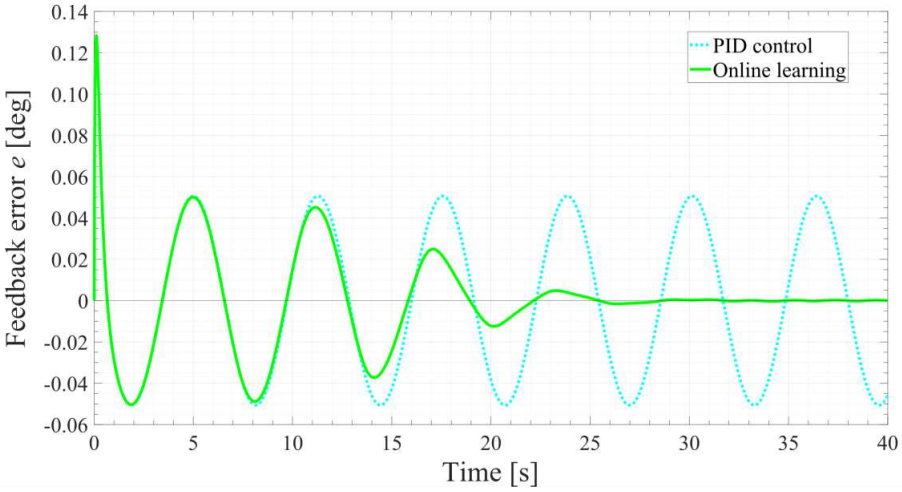


FIGURE 10. Feedback error of the PID control and the online learning method

5.2. **Responsiveness and robustness before and after changing load inertia.** Figure 11 shows the reference and responses θ of the motor before and after changing the load inertia. The parameters of the online learning method are $\eta = 0.004$, $\alpha = 0.001$, and the parameters of the proposed method are $\eta = 0.004$, $\alpha = 0.001$, $N_\varepsilon = 10$. At 15 seconds of the simulation, the magnitude of the load inertia is increased from $J_L = J_M = 2.25 \times 10^{-7}$ kgm^2 to $J_L = 10 \times J_M = 2.25 \times 10^{-6}$ kgm^2.

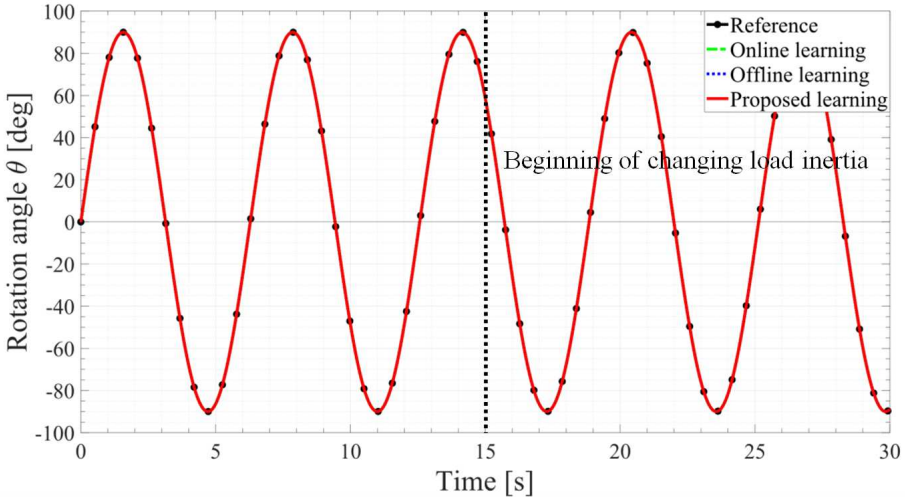
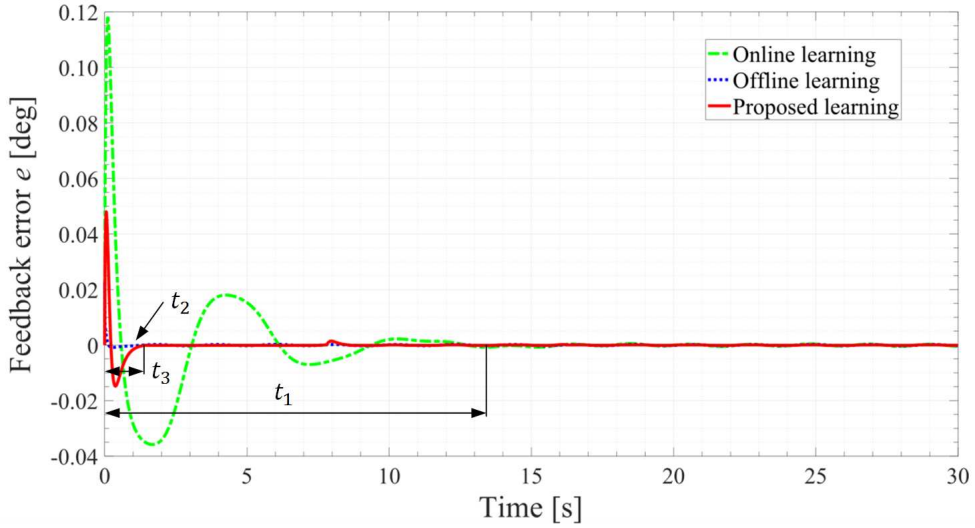
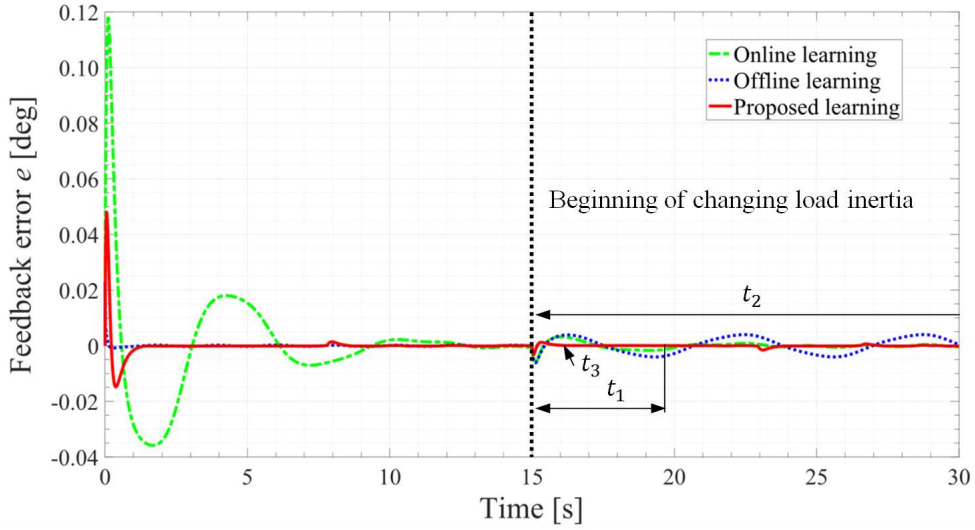


FIGURE 11. Reference and responses with the online learning method, the offline learning method and the proposed method

Figures 12(a) and 12(b) show the results of the feedback error e before and after changing the load inertia, respectively. Figures 13(a) and 13(b) show the results of the output u_n of the neural network compensator before and after changing the load inertia, respectively. Figure 14(a) is an enlarged view of Figure 13(a) from 0 to 14 seconds, and Figure 14(b) is an enlarged view of Figure 13(b) from 14 to 20 seconds. And, Figures 15(a) and 15(b) show the output u_f of the feedback controller before and after changing the load inertia, respectively. From Equation (1), Figure 15 can be regarded as the results of the error signal E_n of the neural network compensator.



(a) Before changing load inertia



(b) After changing load inertia

FIGURE 12. Feedback error with the online learning method, the offline learning method and the proposed method (before and after changing the load inertia)

5.3. **Responsiveness and robustness before and after changing frequency of reference trajectory.** The results of the feedback error e with the frequency of reference trajectory θ_r changed from 1.0 Hz to 0.5 Hz and 3.0 Hz are shown in Figures 16 and 17, respectively. In that case, the parameters related to the learning of the proposed method are $\eta = 0.005$, $\alpha = 0.001$, $N_\varepsilon = 3$.

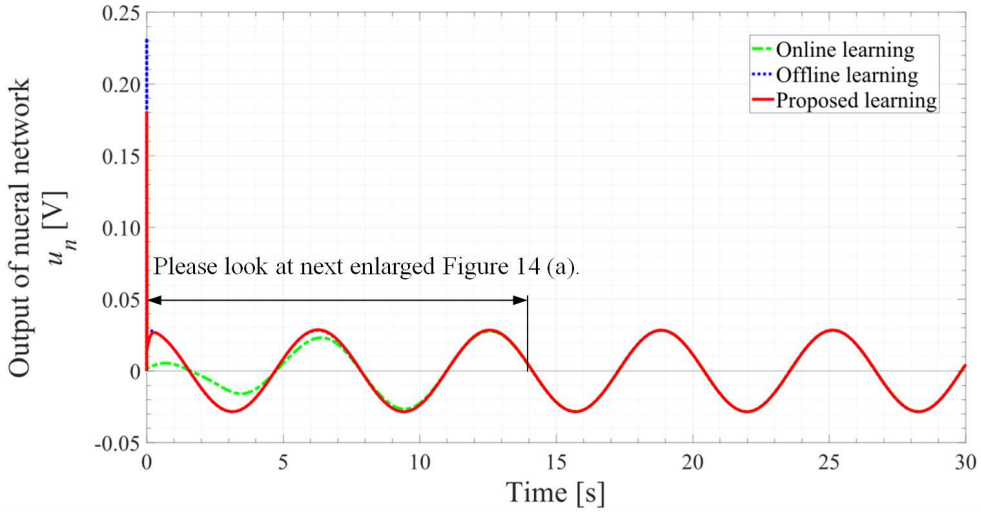
5.4. **Results and considerations.**

(A) The results and considerations before changing the load inertia are as follows.

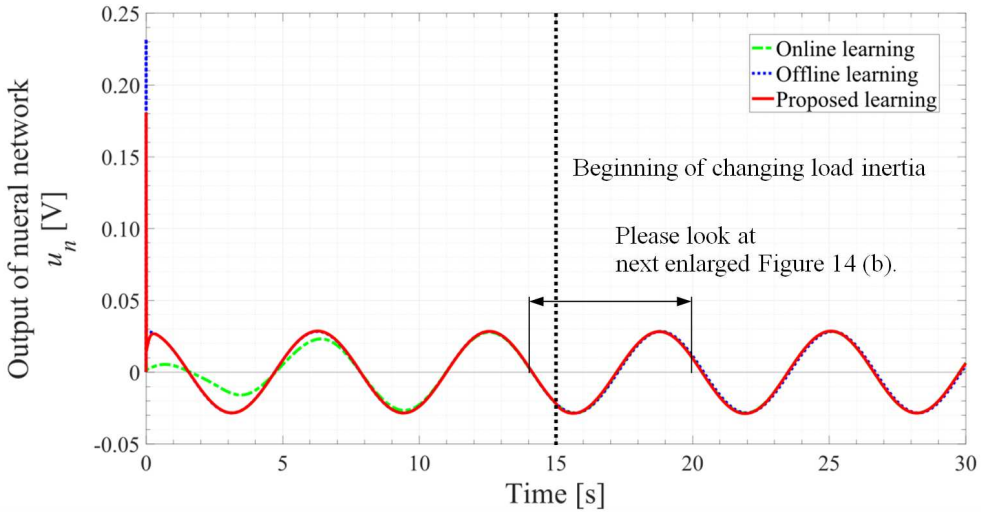
1) The settling time and the steady-state error

The settling time of the online learning method was t_1 in Figure 12(a), which was 13.38 seconds. Then, the maximum value of the steady-state error was ± 0.0008 degrees.

The settling time of the offline learning method was t_2 in Figure 12(a), which was 1.05 seconds. Then, the maximum value of the steady-state error was ± 0.0002 degrees.



(a) Before changing load inertia



(b) After changing load inertia

FIGURE 13. Output of neural network with the online learning method, the offline learning method and the proposed method (before and after changing the load inertia)

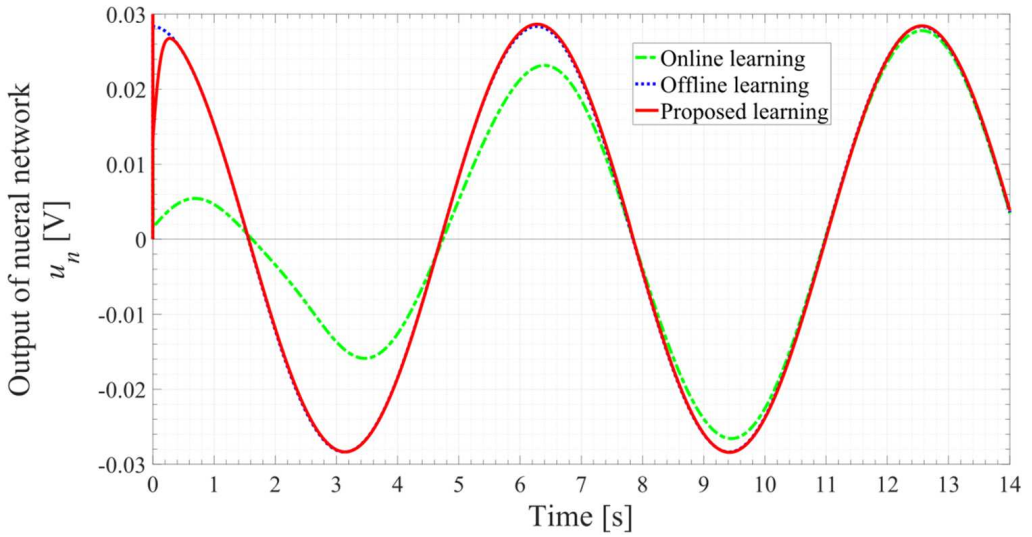
The settling time of the proposed method was t_3 in Figure 12(a), which was 1.41 seconds. Then, the maximum value of the steady-state error was ± 0.0001 degrees.

Regarding the settling time, the proposed method was about 1.3 times longer than the offline learning method, but about 1/9.5 times shorter than the online learning method. In other words, regarding the responsiveness, the proposed method was about 9.5 times faster than the online learning method. Therefore, it has been shown that the proposed method can realize real-time control of neural network control.

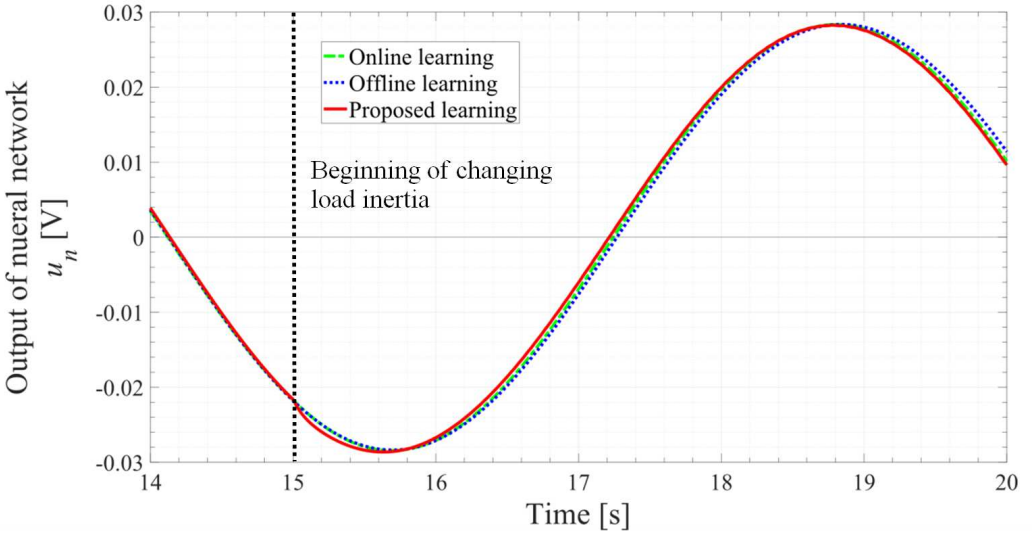
Regarding the steady-state error, it was also confirmed that the proposed method is smaller than the online learning method and the offline learning method.

2) The output of neural network

Figure 13(a) shows the output u_n of the neural network, and Figure 14(a) is an enlarged view of Figure 13(a). The output value of the proposed method was larger than the online learning method and almost the same magnitude as the offline learning method. This means that the learning effect of the proposed method appears in the transient state.



(a) Before changing load inertia



(b) After changing load inertia

FIGURE 14. Enlarged figure view of Figure 13

3) The error signal of neural network

Figure 15(a) shows the error signal E_n of the neural network, that is, the output u_f of the feedback controller. The error signal of the proposed method was smaller than the online learning method and the offline learning method at all times. This means that the learning effect of the proposed method appears in both the transient state and the steady state.

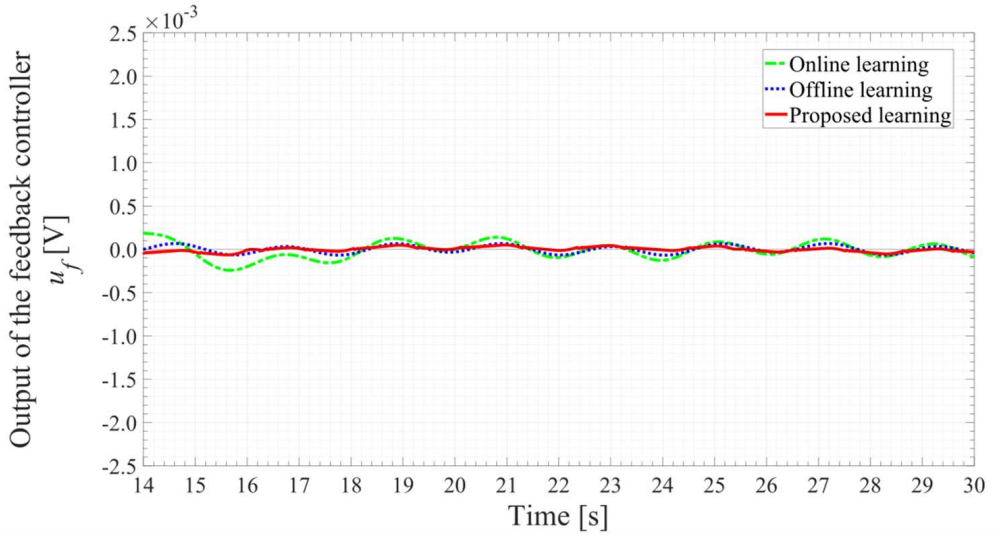
(B) The results and considerations after changing the load inertia are as follows.

1) The settling time and the steady-state error

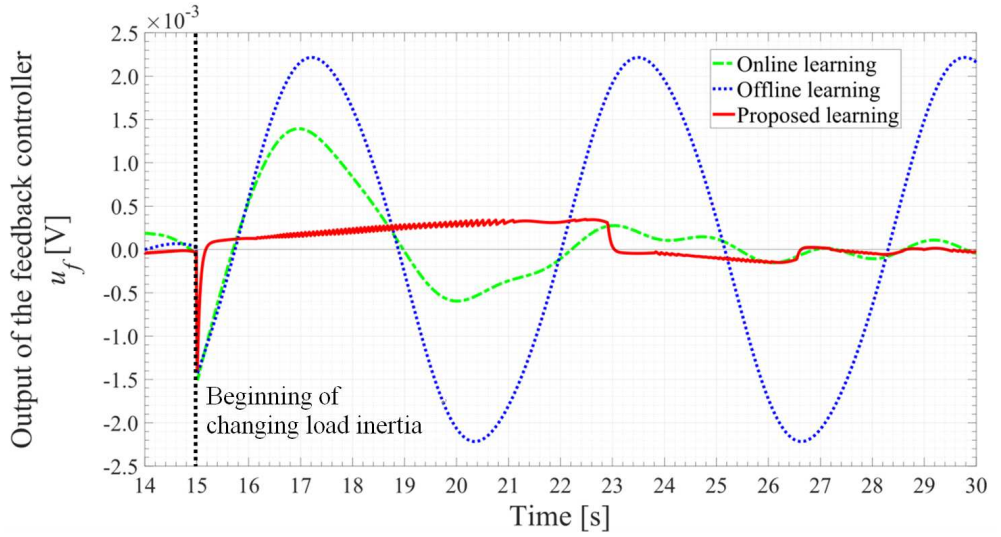
The settling time of the online learning method was t_1 in Figure 12(b), which was 4.84 seconds. Then, the maximum value of the steady-state error was ±0.0010 degrees.

The settling time of the offline learning method was more than 15 seconds as shown by the mark of t_2 in Figure 12(b), because the waveform of the steady-state error oscillated continuously. Then, the maximum value of the steady-state error was ±0.0040 degrees.

The settling times of the proposed method was t_3 in Figure 12(b), which was 1.16 seconds. Then, the maximum value of the steady-state error was ±0.0001 degrees.



(a) Before changing load inertia



(b) After changing load inertia

FIGURE 15. Output of the feedback controller with the online learning method, the offline learning method and the proposed method (before and after changing the load inertia)

Regarding the settling time, the proposed method was about 1/13 shorter than the offline learning method and about 1/4.2 shorter than the online learning method. In other words, regarding the responsiveness, the proposed method was about 13 times faster than the offline learning method and about 4.2 times faster than the online learning method.

Regarding the steady-state error, the maximum value of the steady-state error of the offline learning method was about 40 times larger than that of the proposed method. This means that the offline learning method cannot deal with changes of load inertia and lacks robustness. The steady-state error of the online learning method was about 1/4 that of the offline learning method, but the settling time was about 4.2 times longer than that of the proposed method. This means that the online learning method is not suitable for the real-time control.

Regarding the responsiveness after a sudden change in load, the proposed method was much faster than the offline learning method and about 4.2 times faster than the online

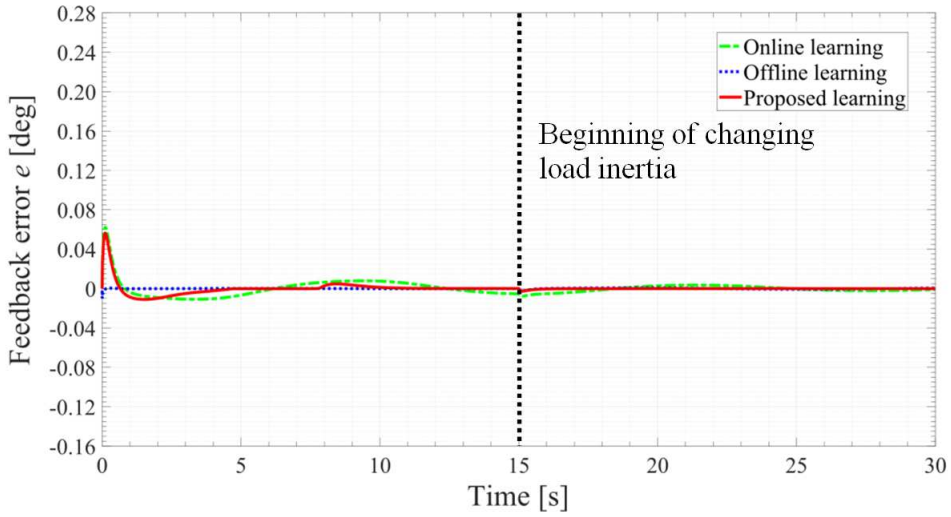


FIGURE 16. Feedback error when the frequency of the reference trajectory is 0.5 Hz

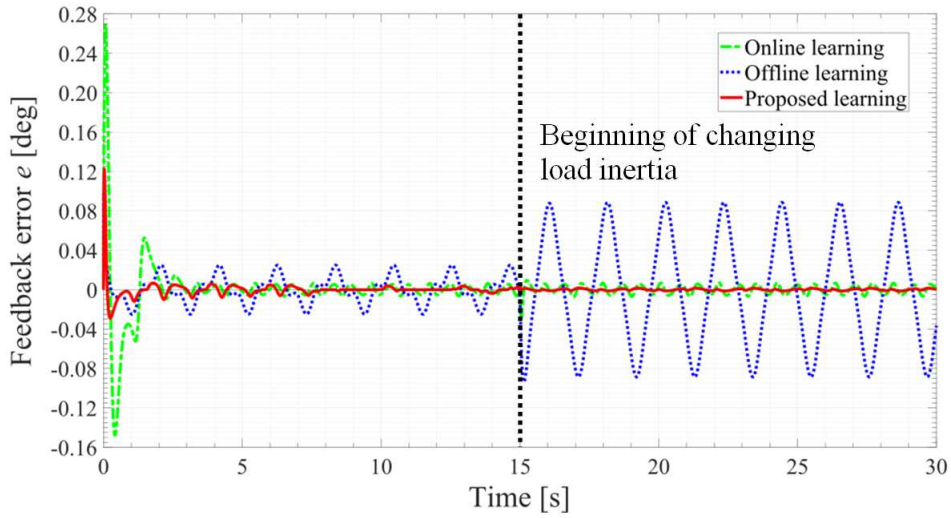


FIGURE 17. Feedback error when the frequency of the reference trajectory is 3.0 Hz

learning method. The maximum steady-state error of the proposed method was reduced to 1/40 of that of the offline learning method and 1/10 of that of the online learning method. Therefore, it is found that the proposed method has high robustness and is suitable for real-time control.

2) The output of neural network

Figure 13(b) shows the output u_n of neural network, and Figure 14(b) is an enlarged view of Figure 13(b). The proposed method obtained larger output values than the online learning method and the offline learning method between 15.0 and 15.6 seconds. This means that the learning effect of the proposed method appears in the transient state.

3) The error signal of neural network

Figure 15(b) shows the error signal E_n of the neural network, that is, the output u_f of the feedback controller. For the error signal E_n shown in Figure 15(b), the proposed method is smaller than the online learning method and the offline learning method. This means that the learning effect of the proposed method appears in both transient state and steady state.

(C) The results and considerations after changing the frequency of reference trajectory are as follows.

1) In case of the frequency of reference trajectory being changed to half

Figure 16 shows the error of the motor rotation angle when the frequency of reference trajectory is changed to half. For the responsiveness and steady-state error before and after the changes in load inertia, Figure 16 shows that the proposed method was inferior to the offline learning method, but much better than the online learning method. It is found that the offline learning method has excellent robustness when the frequency of the reference trajectory is changed to a low value. It is also found that the proposed method has excellent robustness when the frequency of the reference trajectory is changed to a low value.

2) In case of the frequency of reference trajectory being changed to 3 times

Figure 17 shows the error of the motor rotation angle when the frequency of reference trajectory is changed to 3 times. For the responsiveness and steady-state error before and after the changes in load inertia, Figure 17 shows that the proposed method was much better than the online learning method and offline learning method. In addition, the responses of the offline learning method oscillated persistently when the load inertia was changed, indicating that the offline learning method lacks robustness when the frequency of the reference trajectory is changed to a higher value.

From the above results and considerations, the comparison between the online learning method, the offline learning method and the proposed method shown in Table 1 of Section 3.5 was proved. The proposed method solved the disadvantages that the online learning method is not suitable for real-time control and the offline learning method lacks robustness. In other words, the proposed method is able to combine the advantages of the offline learning method with excellent responsiveness and the advantages of the online learning method with excellent robustness.

6. **Conclusions.** Neural network control is expected for high-precision position control of servo systems such as robots and numerical control machines, but the conventional online learning method and the offline learning method of neural network control are not suitable for real-time control due to the long settling time and cannot deal with load fluctuations and disturbances. Therefore, in this study, in order to deal with load fluctuations and disturbances, and to realize real-time control, we proposed the online-offline integrated learning method which is a new learning method of neural network control for the position control of a servo system.

In this paper, firstly, we explained the structure of the servo system using neural network control. Next, after explaining algorithms of the online learning method and the offline learning method, we proposed the online-offline integrated learning method and explained its algorithm. In the proposed method, the offline learning is performed if the feedback error does not exceed the threshold value, and the online learning with few repeating loops is performed within the sampling time if the feedback error exceeds the threshold value. Therefore, the proposed method improved learning efficiency, and obtained excellent responsiveness and excellent robustness, so it can deal with load fluctuations and disturbances and realize real-time control.

Furthermore, we built a simulation model of the position control of the DC servo motor in MATLAB/Simulink, and applied the proposed online-offline integrated learning method in this study to the position control, and performed a simulation of tracking control using a reference value that is a sine wave. In this simulation, we applied the online learning method, the offline learning method, and the proposed method respectively to the neural network control before and after changing the load inertia of the servomotor. Based on

the results and the considerations of simulation, the proposed method could achieve to integrate the excellent responsiveness of the offline learning method with the excellent robustness of the online learning method, and realized the neural network control of the servo system which can deal with the load fluctuations and disturbances and attain the real-time control.

In the future, we would like to clarify the relationship between the learning rate η and the responsiveness and robustness of the servo system by simulation and analysis of the position control of a servo system, and derive a renewal rule for the learning rate η so that self-tuning can be performed.

REFERENCES

[1] S. Omatsu, Neuro control and adaptive control, systems control and information, *Journal of the Institute of Electrical Engineers of Japan*, vol.112, no.7, pp.503-506, 1992.

[2] J. Tsuji, H. Ohmori and A. Sano, Adaptive control incorporating neural network, *Transactions of the Society of Instrument and Control Engineers*, vol.30, no.3, pp.295-302, 1994.

[3] H. J. Guo, S. Sagawa and O. Ichinokura, Position sensorless driving of BLDCM using neural networks, *Electronics Information and System Society, IEE-Part C*, vol.124, no.11, pp.2329-2335, 2003.

[4] Y. Saitoh, Z.-W. Luo, K. Watanabe, E. Muramatu and S. Fujio, 2 D.O.F adaptive force tracking control of a robot manipulator interacting with unknown dynamic environment, *Transactions of the Institute of Systems Control and Information Engineers*, vol.18, no.6, pp.203-212, 2005.

[5] L. Sheng, W. Yan, L. Geng and D. Xu, Adaptive composite neural network control for variable speed wind turbines, *ICIC Express Letters, Part B: Applications*, vol.8, no.9, pp.1339-1346, 2017.

[6] L. Chen and M. Tsutsumi, Compensation for positioning error of CNC machine tools using neural network, *Transactions of the Japan Society of Mechanical Engineers, Part C*, vol.61, no.583, pp.1224-1229, 1995.

[7] B. Li, H. Zhang, P. Ye and J. Wang, Trajectory smoothing method using reinforcement learning for computer numerical control machine tools, *Robotics and Computer-Integrated Manufacturing*, vol.61, 2020.

[8] S. Urushihara, T. Kamano, T. Suzuki and H. Harada, Tracking performance of a positioning system with a linear DC servo motor using neural network, *Industry Applications Society, IEE-Part D*, vol.115, no.3, pp.204-210, 1995.

[9] O. Veligorskyi, R. Chakirov, M. Khomenko and Y. Vagapov, Artificial neural network motor control for full-electric injection moulding machine, *IEEE International Conference on Industrial Technology (ICIT)*, pp.60-65, 2019.

[10] J. L. Calvo-Rolle, O. Fontenla-Romero, B. Pérez-Sánchez and B. Guijarro-Berdiñas, Adaptive inverse control using an online learning algorithm for neural networks, *Informatica*, vol.25, no.3, pp.401-414, 2014.

[11] Q. Huang and K. Nonami, Humanitarian mine detecting six-legged walking robot and hybrid neuro walking control with position/force control, *Mechatronics*, vol.13, nos.8-9, pp.773-790, 2003.

[12] A. L. Hashmia and S. H. Dakheel, Speed control of separately exited DC motor using artificial neural network, *Journal of Engineering and Development*, vol.16, no.4, pp.349-362, 2012.

[13] M. Kawato, K. Furukawa and R. Suzuki, A hierarchical neural-network model for control and learning of voluntary movement, *Biological Cybernetics*, vol.57, no.3, pp.169-185, 1987.

[14] H. Gomi and M. Kawato, Learning control of a closed loop system using feedback-error-learning, *Transactions of the Institute of Systems Control and Information Engineers*, vol.4, no.1, pp.37-47, 1991.

[15] M. Eguchi, M. Qiao, H. Ohmori, Y. Yamasaki and S. Kaneko, Diesel engine combustion control using feedback error learning with artificial intelligence feedforward controller, *Transactions of Society of Automotive Engineers of Japan*, vol.49, no.2, pp.230-234, 2018.