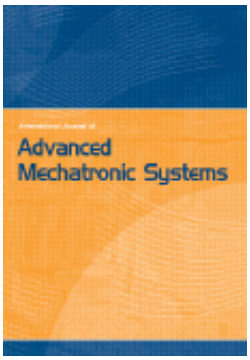


International Journal of Advanced Mechatronic Systems

ISSN online: 1756-8420 - ISSN print: 1756-8412
https://www.inderscience.com/ijamechs

An echo state network-based feedforward feedback controller for application in dynamic systems control

Kazuhiko Takahashi, Naoyuki Kita, Miku Sasaki, Reika Kimura, Masafumi Hashimoto

DOI: 10.1504/IJAMECHS.2024.10063117

Article History:

Received:	19 March 2023
Last revised:	23 October 2023
Accepted:	07 December 2023
Published online:	25 March 2024

An echo state network-based feedforward feedback controller for application in dynamic systems control

Kazuhiko Takahashi*, Naoyuki Kita, Miku Sasaki, Reika Kimura and Masafumi Hashimoto

Faculty of Science and Engineering,
Doshisha University,
Kyoto, Japan
Email: katakaha@mail.doshisha.ac.jp
Email: cgud1025@mail4.doshisha.ac.jp
Email: cgud1053@mail4.doshisha.ac.jp
Email: cgud1024@mail4.doshisha.ac.jp
Email: mhashimo@mail.doshisha.ac.jp
*Corresponding author

Abstract: This study explored the application reservoir computing, particularly echo state networks (ESNs), to control dynamic systems. The design method of a servo-level controller was proposed, where the ESN matches the objective plant output with the reference output. The ESN was combined with a feedback controller to obtain the control input of the plant. The ESN-based controller was first trained using a linear-regression approach with fixed datasets gathered from the objective plant. Thereafter, feedback error learning was performed during the control process in real-time to compensate for the control error due to the identification error of the plant's inverse transfer function. Computational experiments involving the control of a discrete-time nonlinear plant were conducted. The simulation results clarified the feasibility of the proposal and validated the performance of the ESN-based controller.

Keywords: reservoir computing; echo state network; ESN; control system; feedforward feedback controller.

Reference to this paper should be made as follows: Takahashi, K., Kita, N., Sasaki, M., Kimura, R. and Hashimoto, M. (2024) 'An echo state network-based feedforward feedback controller for application in dynamic systems control', *Int. J. Advanced Mechatronic Systems*, Vol. 11, No. 1, pp.40–49.

Biographical notes: Kazuhiko Takahashi received his ME and DE in Mechanical Engineering from the Tohoku University, Sendai, Japan, in 1990 and 1997, respectively. He is currently a Professor of Doshisha University. His research interests include intelligent control systems.

Naoyuki Kita received his BE in Information Engineering from Doshisha University, Kyoto, Japan, in 2023. He has engaged in research on neural networks at Doshisha University.

Miku Sasaki received his BE in Information Engineering from Doshisha University, Kyoto, Japan, in 2023. He has engaged in research on neural networks at Doshisha University.

Reika Kimura received his BE in Information Engineering from Doshisha University, Kyoto, Japan, in 2023. He has engaged in research on neural networks at Doshisha University.

Masafumi Hashimoto received his ME and DE in Aeronautical Engineering from the Osaka Prefecture University, Osaka, Japan, in 1981 and 1988, respectively. He is currently a Professor of Doshisha University. His research interests include vehicle automation.

This paper is a revised and expanded version of a paper entitled 'Remarks on a feedforward feedback controller using an echo state network for controlling dynamic systems' presented at The 2022 International Conference on Advanced Mechatronic System (ICAMechS2022), Toyama, Japan, 17–20 December 2022.

1 Introduction

Machine learning holds significant promise for providing applicable technological solutions to a wide variety of problems. Accordingly, recent advances in machine learning have eased the handling of previously intractable tasks in many fields, including science, engineering, social science, economics, finance and business (Lu, 2019; Sarker, 2021). For instance, although neural networks (NNs) have been employed to solve several engineering difficulties over the last half-century (Prieto et al., 2016), deep learning-integrated NNs are superior and have found application in real-world computing, particularly in image, speech and text processing and recognition (Schmidhuber, 2015; Alzubaidi et al., 2021). However, to train deep learning-integrated NNs to create solutions that mimic complicated real-world problems sufficiently, large volumes of data are required. Additionally, the parameter complexity, memory requirements and computational costs increase as the training data grow.

This problem notwithstanding, many studies have attempted to apply NNs in controlling dynamic systems. These studies often encompass the identification and control of uncertain or unknown nonlinear and non-stationary systems, as well as the exploitation of the nonlinear mapping, learning, generalising and optimising capabilities of NNs. In this regard, many successful control systems using NNs have emerged (Hagan et al., 2002; Meireles et al., 2003; Denaï et al., 2007; Ruano et al., 2014; Khargonekar and Dahleh, 2018). However, NNs must also be able to realise system dynamics if they are to be used as controllers in control systems. Although feedforward NNs (FNNs) can realise system dynamics using tapped delay line inputs, recurrent NNs (RNNs) have been shown to exhibit computational advantages over FNNs in handling temporal patterns structured in space and time. This is the case even though the training process of RNNs has a higher computational complexity, due to backpropagation through time (BPTT) and real-time recurrent learning (RTRL), than that of the FNNs (Salehinejad et al., 2017). In addition, reservoir computing, encompassing echo state networks (ESNs) (Jaeger and Haas, 2004) and liquid state machines (Maass et al., 2002), has attracted substantial attention as a training paradigm for RNNs over the past two decades (Schrauwen et al., 2007; Lukoševičius and Jaeger, 2009; Zhang and Vargas, 2023). To this end, studies have shown that ESNs using several sparsely and randomly generated connections (described as the reservoir) can achieve global optimisation with high convergence speed. They have, therefore, been applied successfully to handle temporal data during various tasks, such as dynamic system approximation, signal processing, time series classification, time series generation and time series prediction (Bala et al., 2018; Tanaka et al., 2019; Sun et al., 2020). Moreover, the utilisation of ESNs in the control engineering field for their computational advantages has been explored. For instance, an ESN was employed as an observer and predictive model to estimate controlled objective plant dynamics in control systems, including state feedback and model predictive control (Pan and Wang, 2012; Schwedersky et al., 2018; Jordanou et al., 2018; Ogawa and Takahashi, 2021). Furthermore, a controller in which an ESN afforded the control input of the objective plant directly has been investigated, with some empirical studies presented (Xu et al., 2005; Salmen and Ploger, 2005; Waegeman et al., 2012; Løvlid et al., 2013; Yao et al., 2020; Perrusquía and Yu, 2021). Therefore, the design of a suitable controller based on an ESN for nonlinear dynamic systems is interesting.

Based on this background, this study explores a possible approach for controlling nonlinear dynamic systems using an ESN and investigates the performance of the ESN-based controller. In the control system, the ESN was designed to comprise the objective plant's control input that acts as a servo-level controller, thereby ensuring that the plant output matches the reference output. The ESN-based controller was first trained using a linear-regression approach with fixed datasets gathered from the objective plant. Subsequently, feedback error learning in real time was conducted during the control process, followed by computational experiments to establish the effectiveness of the ESN in control applications and evaluate the characteristics of the ESN-based controller. The main contributions of this study are summarised as follows:

1 An ESN-based approach applicable to the servo control problem of nonlinear systems is proposed.

2 A practical design method for a feedback feedforward controller based on ESN integration and feedback error learning is provided.

3 The control performance and characteristics of the proposed controller for a range of different nonlinear systems are clarified.

2 ESN-based controller

2.1 The ESN

A standard model of an ESN consists of a reservoir representing a fixed RNN and a readout element that linearly connects the reservoir states to the output layer (a guideline for designing ESN parameters can be found in Lukoševičius (2012). In this study, by incorporating connections from the input layer directly to the output layer and feedback connections from the output layer to the reservoir states, the input-output relationship of the ESN can be obtained, as follows:

$$\begin{cases} \boldsymbol{x}(t+1) = (1-\alpha)\boldsymbol{x}(t) + \alpha f(\boldsymbol{W}^{in}\boldsymbol{s}(t+1) \\ \qquad\qquad +\boldsymbol{W}\boldsymbol{x}(t) + \boldsymbol{W}^{fb}\boldsymbol{\sigma}(t)) \\ \boldsymbol{\sigma}(t+1) = \boldsymbol{W}^{out}\tilde{\boldsymbol{x}}(t+1) \end{cases}, \tag{1}$$

where $\boldsymbol{s}(t) \in \mathbb{R}^L$, $\boldsymbol{x}(t) \in \mathbb{R}^M$ and $\boldsymbol{\sigma}(t) \in \mathbb{R}^N$ represent the external input vector, the reservoir's state vector and the network output vector at the sampling time t, respectively. The extended vector $\tilde{\boldsymbol{x}}(t)$ consists of the external input

and the reservoir's state vectors: $\tilde{\boldsymbol{x}}(t) = [\boldsymbol{s}(t) \,.\, \boldsymbol{x}(t)]^{\mathrm{T}}$. Additionally, $\boldsymbol{W}^{in} \in \mathbb{R}^{M \times L}$ is the input weight matrix representing the connection weight from the input layer to the reservoir, $\boldsymbol{W} \in \mathbb{R}^{M \times M}$ is the internal connection weight matrix between reservoirs, $\boldsymbol{W}^{fb} \in \mathbb{R}^{M \times N}$ is the feedback connection weight matrix from the output layer to the reservoir and $\boldsymbol{W}^{out} \in \mathbb{R}^{N \times (M+L)}$ is the readout matrix representing the output connection weight from the extended vector to the output layer, where the connection weights include the threshold weight based on a constant value. Furthermore, this study notably employs a leaky integrator neuron in the reservoir to tune the network dynamics with a leaking rate of $\alpha \in (0, 1]$, where the function $f(\cdot)$ is an element-wise activation function of the reservoir, which is typically given by a hyperbolic tangent function to attain the nonlinearity of the network.

Subsequently, the connection weight matrices $\boldsymbol{W}^{in}$ and $\boldsymbol{W}^{fb}$ are respectively initialised randomly using uniform distributions of ranges $[-a_{\mathtt{in}}, a_{\mathtt{in}}]$ and $[-a_{\mathtt{fb}}, a_{\mathtt{fb}}]$. Contrarily, the internal connection weight matrix between reservoirs $\boldsymbol{W}$ is first initialised randomly using a uniform distribution of range $[-1, 1]$. The connection fraction or sparseness of this connection weight matrix is controlled using the interconnectivity rate P. Thereafter, it is significantly scaled to satisfy the echo state property with a spectral radius of $\rho(\boldsymbol{W}) < 1$, where the function $\rho(\cdot)$ indicates the maximum eigenvalue. These connection weight matrices are kept constant. The readout matrix $\boldsymbol{W}^{out}$ is trained offline using the linear-regression approach on a fixed dataset of desired input-output sets to minimise the following real-valued cost function J:

$$J = \sum_{t=1}^{T} \|\boldsymbol{r}(t) - \boldsymbol{\sigma}(t)\|^2 + \gamma \|\boldsymbol{W}^{out}\|^2, \tag{2}$$

where $\boldsymbol{r}(t)$ is the reference output vector, $\gamma \in \mathbb{R}^+$ is a regularisation parameter and T indicates the time series length. Notably, the ridge regression could also be employed to derive the following expression:

$$\boldsymbol{W}^{out} = \boldsymbol{R}\tilde{\boldsymbol{X}}^{\mathrm{T}}(\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^{\mathrm{T}} + \gamma \boldsymbol{I})^{-1}, \tag{3}$$

where the matrices, $\tilde{\boldsymbol{X}} = [\tilde{\boldsymbol{x}}(1) \,.\, \tilde{\boldsymbol{x}}(2) \,.\, \ldots \,.\, \tilde{\boldsymbol{x}}(T)] \in \mathbb{R}^{(M+L) \times T}$ and $\boldsymbol{R} = [\boldsymbol{r}(1) \,.\, \boldsymbol{r}(2) \,.\, \ldots \,.\, \boldsymbol{r}(T)] \in \mathbb{R}^{N \times T}$, are the column-wise concatenation of the extended vectors and the reference output vectors in the column direction, respectively, and $\boldsymbol{I}$ is the identity matrix.

2.2 Feedforward feedback controller

In this study, a servo controller based on an ESN, which ensures that the objective plant's output matches the reference output, is considered. To simplify the controller design, we assume that the objective plant is a predominantly linear discrete-time single-input and single-output system, represented as follows:

$$\begin{aligned} y(t) = F\,(&y(t-d), ..., y(t-n+1-d), \\ &u(t-d), ..., u(t-\nu-d))\,, \end{aligned} \tag{4}$$

where $y(t)$ and $u(t)$ are the plant output and the control input, respectively; n and m are the plant orders; d is the plant dead time; $\nu = m + d - 1$ and function $F(\cdot)$ expresses the input-output characteristics of the plant. Here, we also assume that the upper limit order of the plant and the dead time are known. Therefore, assuming the existence of an inverse function of the plant characteristic function in relation to the control input $u(t-d)$, the condition under which the control error, defined by the difference between the reference and objective plant's outputs, is zero can be expressed as follows:

$$\begin{aligned} u(t) = F^{-1}\,(&y_r(t+d), y(t), ..., y(t-n+1), \\ &u(t-1), ..., u(t-\nu))\,, \end{aligned} \tag{5}$$

where the sampling time is shifted by the dead time, and $y_r(t)$ indicates the reference output of the plant. A multilayer NN can approximate a static arbitrary nonlinear function with arbitrary accuracy in general. Therefore, such a network using an input vector whose components are variables on the right side of equation (5) acquires the inverse function $F^{-1}(\cdot)$ after the convergence of its learning and thus yields the control input $u(t)$ for conducting the control task. Similarly, when the ESN is applied to synthesise the control input $u(t)$, the input of the ESN $\boldsymbol{s}(t)$ could be defined using the reference and objective plant's outputs by considering equation (5) and the dynamics of the recurrent network based on the inner feedback loop in the network: $\boldsymbol{s}(t) = [y_r(t+d) \,.\, y(t)]^{\mathrm{T}}$.

In the offline training process of the readout matrix $\boldsymbol{W}^{out}$ based on equation (3), the control input $u(t-d)$ is selected as the teaching signal against the network output $\sigma(t)$, where the input of the ESN is defined as $\boldsymbol{s}(t) = [y(t) \,.\, y(t-d)]^{\mathrm{T}}$. Consequently, the ESN identifies the inverse transfer function of the plant, as shown in Figure 1(a), and acts as an inverse model-based controller in the control process. Additionally, the readout matrix is trained in real time by the least mean square approach during the control task to compensate for the identification error. However, when the cost function E, defined straightforwardly using the squared control error, is minimised during the real-time training, the Jacobian information of the plant $\frac{\partial y(t)}{\partial u(t-d)}$ is required to calculate the gradient of E in relation to the components of the readout matrix. To address this problem, a feedback error learning scheme is introduced. Therefore, the cost function E is defined using a feedback loop output synthesised by a conventional feedback controller, and the updated rule of the readout matrix is finally derived, as follows:

$$\boldsymbol{W}^{out}(t+1) = (1-\beta)\boldsymbol{W}^{out}(t) + \eta v(t)\tilde{\boldsymbol{x}}^{\mathrm{T}}(t), \tag{6}$$

where $v(t)$ is the feedback loop output, $\eta \in \mathbb{R}^+$ is the learning rate and $\beta \in \mathbb{R}$ is the decay rate. Consequently, the control input comprises the ESN output $\sigma(t)$ and the feedback loop output $v(t)$, as shown in Figure 1(b). Although the feedback loop is also expected to prevent system instability due to the ESN's identification error, the ESN becomes a feedforward controller if the ESN

training by the feedback error learning proceeds ideally. Such a controller is called a feedforward feedback controller (Yamada, 2009).

2.3 Controller stability

The stability of the controller is vital for its performance; however, it is occasionally difficult to achieve because of the nonlinearity of the ESN and the plant, even if the feedback controller is linear. Therefore, the local stability condition of the feedforward feedback controller is addressed under the following assumptions:

1 The plant expressed by equaiton (4) is a linear time-invariant system where the upper limit orders of the plant and the dead time are known.

2 The input of the ESN is given by $\boldsymbol{s}(t) = [y_r(t+d)\,.\,y(t)]^{\mathrm{T}}$, and the ESN output $\sigma(t)$ is used to synthesise the control input: $L = 2$ and $N = 1$.

3 The feedback controller uses a proportional (P) control law, i.e., $v(t) = K_p\{y_r(t) - y(t)\}$, where K_p is the gain factor, and there exist the ESN's connection weight matrices that can make the output of the feedback controller sufficiently small.

According to Assumption 1, the plant can be represented by the state-space equation, as follows:

$$\begin{cases} \boldsymbol{\zeta}(t+1) = \boldsymbol{A}\boldsymbol{\zeta}(t) + \boldsymbol{b}u(t) \\ y(t) = \boldsymbol{c}^{\mathrm{T}}\boldsymbol{\zeta}(t) \end{cases}, \tag{7}$$

where $\boldsymbol{\zeta}(t) \in \mathbb{R}^n$ is the state vector. Applying Assumption 3 to equation (6), the readout matrix can be assumed to be a constant matrix when the decay factor is adequately small. By substituting the control input $u(t) = \sigma(t) + v(t)$ given by Assumption 2 into equation (7) and combining with equation (1), we obtain the following expression:

$$\boldsymbol{q}(t+1) = \boldsymbol{\Psi}\boldsymbol{q}(t) + \boldsymbol{\Phi}y_r(t), \tag{8}$$

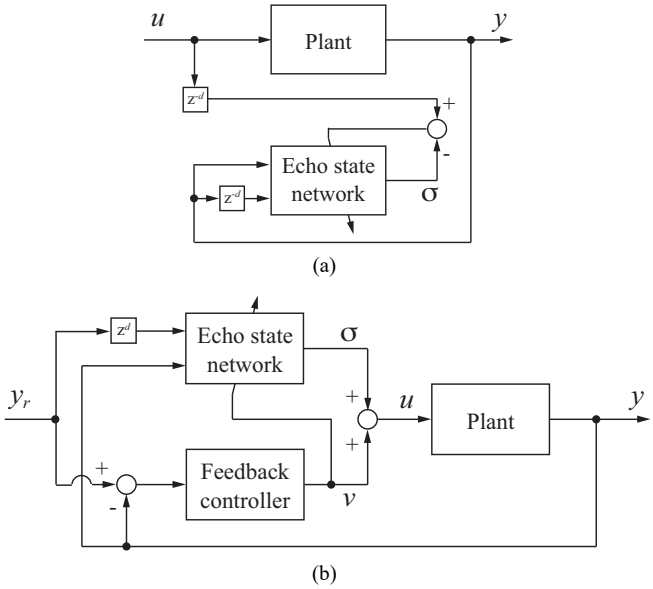
where $\boldsymbol{q}(t) = [\boldsymbol{\zeta}(t)\,.\,\boldsymbol{x}(t)]^{\mathrm{T}}$ and the coefficient matrices are given by

$$\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{A} + \boldsymbol{b}\boldsymbol{W}_{12}^{out}\boldsymbol{c}^{\mathrm{T}} - K_p\boldsymbol{b}\boldsymbol{c}^{\mathrm{T}} & \boldsymbol{b}\boldsymbol{W}_2^{out} \\ \alpha\boldsymbol{H}\boldsymbol{\psi}_{21} & (1-\alpha)\boldsymbol{I} + \alpha\boldsymbol{H}\boldsymbol{\psi}_{22} \end{bmatrix}, \tag{9}$$

$$\boldsymbol{\Phi} = \begin{bmatrix} K_p\boldsymbol{b} \\ \alpha\boldsymbol{H}\boldsymbol{\phi}_2 \end{bmatrix}, \tag{10}$$

where $\boldsymbol{\psi}_{21} = \{\boldsymbol{W}_2^{in}\boldsymbol{c}^{\mathrm{T}}(\boldsymbol{A} + \boldsymbol{b}\boldsymbol{W}_{12}^{out}\boldsymbol{c}^{\mathrm{T}} - K_p\boldsymbol{b}\boldsymbol{c}^{\mathrm{T}}) + \boldsymbol{W}^{fb}\boldsymbol{W}_{12}^{out}\boldsymbol{c}^{\mathrm{T}}\}$, $\boldsymbol{\psi}_{22} = (\boldsymbol{W}^{fb} + \boldsymbol{W}_2^{in}\boldsymbol{c}^{\mathrm{T}}\boldsymbol{b})\boldsymbol{W}_2^{out} + \boldsymbol{W}$ and $\boldsymbol{\phi}_2 = \{\boldsymbol{W}_1^{in}\boldsymbol{z}^{d+1} + (\boldsymbol{W}^{fb} + \boldsymbol{W}_2^{in}\boldsymbol{c}^{\mathrm{T}}\boldsymbol{b})\boldsymbol{W}_{11}^{out}\boldsymbol{z}^d + K_p\boldsymbol{W}_2^{in}\boldsymbol{c}^{\mathrm{T}}\boldsymbol{b}\}$. The connection weight matrices are represented using block matrices: $\boldsymbol{W}^{in} = [\boldsymbol{W}_1^{in}\,.\,\boldsymbol{W}_2^{in}]$ $(\boldsymbol{W}_1^{in}, \boldsymbol{W}_2^{in} \in \mathbb{R}^M)$, $\boldsymbol{W}^{out} = [\boldsymbol{W}_{11}^{out}\,.\,\boldsymbol{W}_{12}^{out}\,.\,\boldsymbol{W}_2^{out}]$ $(\boldsymbol{W}_{11}^{out}, \boldsymbol{W}_{12}^{out} \in \mathbb{R},\ \boldsymbol{W}_2^{out} \in \mathbb{R}^M)$. Moreover, matrix $\boldsymbol{H}$ represents the nonlinear property of the ESN: $\boldsymbol{H} = \mathtt{diag}[\frac{f(\theta_i)}{\theta_i}] \in \mathbb{R}^{M\times M}$, where $\theta_i = \sum_{j=1}^{L} W_{ij}^{in}s_j(t+1) + \sum_{j=1}^{M} W_{ij}x_j(t) + W_i^{fb}\sigma(t)$. Consequently, the local stability can be guaranteed if the spectral radius of matrix $\boldsymbol{\Psi}$ in equation (8) is less than one. This result is derived under very limited conditions; however, it may provide information on the stability condition of the controller.

Figure 1 Block diagram showing the control system using the ESN-based controller, where $\mathtt{z}$ is the shift operator, (a) identification system during the training process (b) feedback error learning during the control process



3 Computational experiments

To investigate the capability and characteristics of the ESN-based feedforward feedback controller, computational experiments were conducted using a second-order dominant autoregressive-moving-average (ARMA) model with nonlinear and parasitic terms as the objective plant:

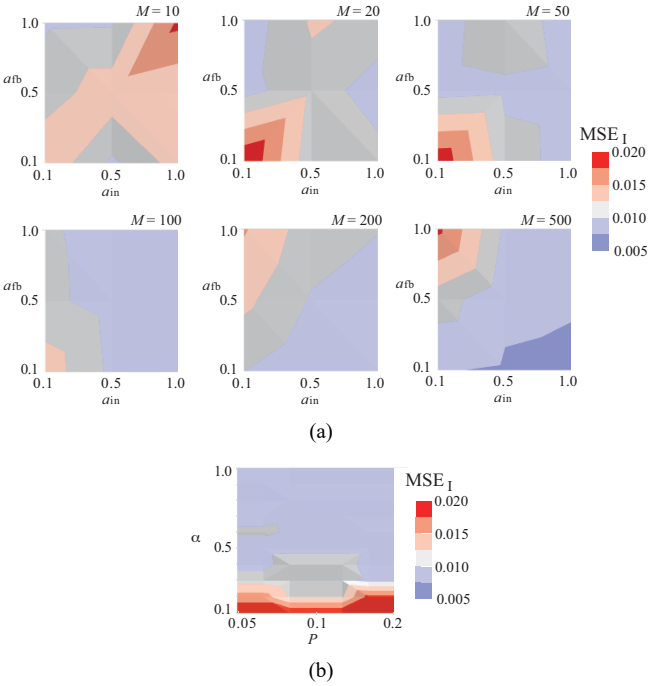
$$y(t) = -\sum_{i=1}^{2} a_i y(t-i) + \sum_{i=1}^{2} b_i u(t-i) + a_n y^2(t-1) + a_p y(t-p) + \sum_{i=1}^{2} c_i \xi(t-i), \tag{11}$$

where $\xi(t)$ is the random noise given by a uniform distribution of range [–0.05, 0.05]. Thus, the coefficients were set as follows: a_1 = –1.3, a_2 = 0.3, b_1 = 1, b_2 = 0.7, a_n = 0.2, a_p = –0.03, p = 3, c_1 = 1 and c_2 = –0.3.

First, the influence of the ESN model parameters on the learning performance was investigated because the performance of the controller depends on the identification accuracy. Using the identification system shown in Figure 1(a), we investigated the scaling model parameters systematically, including the uniform distribution scales $a_{\mathtt{in}}$ and $a_{\mathtt{fb}}$ for the input and feedback connection weight

matrices, the number of reservoirs M, the interconnectivity rate P for the internal connection weight matrix and the leaking rate α. Here, the spectral radius was scaled to ρ = 0.9 after the maximum eigen value of the internal connection weight matrix was evaluated. Training samples consisting of the input-output sets for the plant were assembled in advance. The assembly was achieved through a preliminary experiment where the plant was driven by using the plant input generated randomly based on a uniform distribution of range [–0.5, 0.5]. Two hundred trials were conducted, and the length of a single trial was 50 sampling numbers: $\{(y(1),u(1)),...,(y(50),u(50))\}$. Therefore, the total number of training patterns was 10,000 (hereinafter, this dataset is called 'D10K'). After training the readout matrix using equation (3) with the regularisation parameter of 0.0001, the learning performance was evaluated using the mean of the squared error (MSE) approach during the identification experiment. Here, the control input $u(t)$ was synthesised using the proportional-integral-derivative (PID) controller with P, I and D gains of 0.5, 0.1 and 0.01, respectively, against the rectangular wave as the reference output of the plant. Further, the MSE in the identification experiment was defined using the control input $u(t)$ and the ESN output $\sigma(t)$: $MSE_{\mathrm{I}} = \frac{1}{T_{\mathrm{I}}}\sum_{t=1}^{T_{\mathrm{I}}}|u(t)-\sigma(t)|^2$. In the identification experiment, the length of a single trial T_{I} was 200 sampling numbers. Figure 2(a) shows the resulting MSE_{I} surface obtained using the result of the ten runs for each uniform distribution scale $(a_{\mathtt{in}}, a_{\mathtt{fb}})$ combination for each number of reservoirs M, with the leaking rate of the neuron set to 0.9. The amplitudes of MSE_{I} are indicated by the colour bar on the right side of the figure. The Shapiro-Wilk test with a 5% significance level did not show normal distributions in several cases. Therefore, the evaluation was conducted using the median and interquartile ranges of MSE_{I}. Investigations revealed that increasing the scale $a_{\mathtt{in}}$ decreased the identification error, whereas increasing the scale $a_{\mathtt{fb}}$ increased the identification error. Moreover, the Kruskal-Wallis test showed statistical differences, with a significance level of 5%. Increasing the number of reservoirs also decreased the identification error; however, no statistical difference was observed between the parameter combinations showing MSE_{I} less than 0.01. Moreover, using a considerably large number of reservoirs against the number of datasets might lead to increased computational costs, network overfitting and the occurrence of the ill-posed problem during the readout matrix training. For these investigations, the number of reservoirs was set to 200, and the scales $a_{\mathtt{in}}$ and $a_{\mathtt{fb}}$ were set to 1.0 and 0.1, respectively, in the subsequent experiments. Figure 2(b) shows the resulting MSE_{I} surface obtained using the result of the 10 runs for each interconnectivity rate and leaking rate (P, α) combination. The evaluation was also conducted using the median and interquartile ranges of MSE_{I}. Although the interconnectivity rate did not affect the learning performance, increasing the leaking rate decreased the identification error. Considering this result, the interconnectivity rate and the leaking rate were set to 0.2 and 0.9, respectively. Figure 3 shows an example of the system response during the identification experiment, where the ESN model parameters use the values determined through the investigation, as shown in Figure 2. As shown at the bottom of Figure 3, the ESN output aligned with the control input synthesised using the PID controller. This confirmed that the ESN could sufficiently approximate the control input with adequate accuracy and that the MSE_I was less then 0.01, thereby affording the inverse transfer function of the plant.

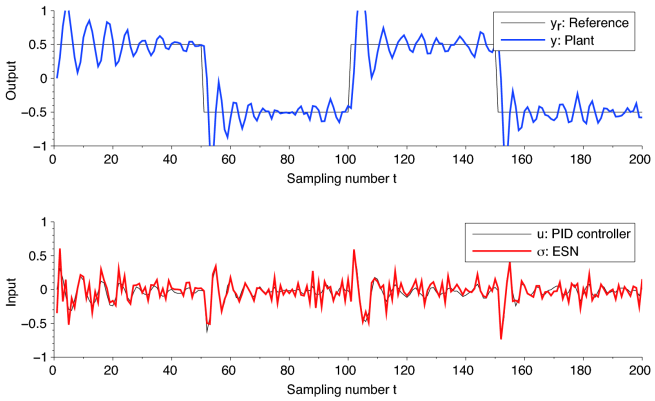
Figure 2 Error (MSE_{I}) surface of the ESN-based controller in the identification experiment, (a) effect of the scales for the input and feedback connection weights and the number of reservoirs (b) effect of the interconnectivity rate for the reservoir and the leaky integrator neuron's leaking rate (see online version for colours)



Next, the control performance of the ESN-based controller was assessed using the control system shown in Figure 1(b), where the P control law was used in the feedback controller to apply the feedback error learning to the ESN. In the control experiment, the ESN was first trained offline with 50,000 training patterns (hereinafter, this dataset is called 'D50K') assembled from 1,000 trials with a length of 50 sampling numbers per single trial, following the same procedure employed for the learning performance experiment. For the feedback error learning, the P gain was 0.1 and the learning and decay rates were set to 0.05 and 0.0001, respectively. Here, the P gain was determined by trial and error to ensure that the feedback error learning proceeds without the divergence of the plant output. The reference output of the plant $y_r(t)$ appeared as a rectangular wave. Figure 4 shows examples of the system response with and without the implementation of the feedback error learning. Control errors, e.g., the offset, can be observed

in Figure 4(a), caused by the identification error of the ESN. Fortunately, subjecting the ESN-based controller to feedback error learning can reduce the control error, as shown in Figure 4(b). Figure 5 compares the control performances obtained using the MSE during the control experiment: $MSE_C = \frac{1}{T_C}\sum_{t=1}^{T_C}|y_r(t) - y(t)|^2$, where the length of a single trial T_C was 200. The evaluation was conducted using the median and interquartile ranges of the MSE_C obtained from the results of the 100 runs. This was done because normal distributions were lacking in these indices, according to the Shapiro-Wilk test with a significance level of 5%. However, the Kruskal-Wallis test showed a statistical difference, with a significance level of 5%. This result confirms the feasibility of the proposed controller and the effectiveness of real-time feedback error learning in reducing the control error. Although the control error could be reduced using feedback error learning, the feedback loop increased the overshoot of the plant output at a transient point where the reference output varied rapidly. This occasionally made the plant output unstable since the sign of the feedback controller output was the same as that of the ESN output. Consequently, the control parameters of the feedback controller should be tuned carefully to guarantee the stability of the control system.

Figure 3 Examples of the system response in identifying the inverse transfer function of the ARMA model plant (see online version for colours)



Notes: The top figure shows the system's output, where the reference and plant outputs are denoted by the thin black and blue lines, respectively. The bottom figure shows the control input, where the PID controller and ESN outputs are denoted by thin black and red lines, respectively.

Next, the control capability of the ESN-based controller against other plants was tested. Figure 6(a) shows the system's response when controlling the Hammerstein model, represented by the following expression:

$$y(t) = -\sum_{i=1}^{2} a_i y(t-i) + \sum_{i=1}^{2}\{b_i u(t-i) + m_i u^2(t-i) + n_i u^3(t-i)\} + \sum_{i=1}^{2} c_i \xi(t-i), \quad (12)$$

where the coefficients were set as follows: a_1 = –0.6, a_2 = 0.1, b_1 = 1.8, b_2 = –0.15, m_1 = –1.8, n_1 = 0.15, m_2 = 0.6 and n_2 = –0.05. Figure 6(b) shows the system's response when controlling the bilinear model, represented by the following expression:

$$y(t) = -\sum_{i=1}^{2} a_i y(t-i) + \sum_{i=1}^{2} b_i u(t-i) + \sum_{i=1}^{2} m_i y(t-i)u(t-i) + \sum_{i=1}^{2} c_i \xi(t-i), \quad (13)$$

where the coefficients were set as follows: a_1 = –0.4, a_2 = 0.09, b_1 = 0.3, b_2 = –0.1, m_1 = 0.1 and m_2 = 0.05. Notably, the ESN-based controller had to be retrained offline using the new datasets obtained from the new plant, and the feedback error learning was performed during the control process. Here, the ESN model parameters were the same as those employed for controlling the ARMA model plant. Investigations revealed that the ESN-based controller ensured that each plant output matched the reference output appropriately even though the ESN was not exactly optimised for each plant. By so doing, the usefulness of the proposed controller for controlling nonlinear plants was established. However, as shown in Figures 4(b) and 6, the ESN-based controller could not compensate for the effect of the plant noise completely. Further, even though the output of the feedback controller appeared to be close to zero, the control error remained because of the plant noise.

The control performance obtained with the ESN-based controller was compared with that obtained using a conventional control method. Here, the generalised predictive control (GPC), a model predictive control approach that outperforms the PID controller (Schwenzer et al., 2021), was considered. Although the GPC was designed using a linear model of the objective plant, i.e., a controlled autoregressive-integrated-moving-average model, it ensured that the plant output matched the desired output, where the prediction and control horizons were set to one. As shown in Figure 7, the ESN-based controller exhibited better performance than the GPC in controlling each plant. The Kruskal-Wallis test demonstrated their statistical differences, revealing a significance level of 5%. This result also demonstrates the effectiveness of the proposed controller in controlling nonlinear plants.

Finally, the characteristics of the ESN in control applications were investigated by comparing it with the other networks. A RNN and a nonlinear autoregressive exogenous network (NARXN) were tested as replacements of the ESN in the feedforward feedback controller. Here, an Elman network was considered as the RNN, and a FNN and a radial basis function network (RBFN) were adopted as the NARXN. Although these networks are popular and their methodologies are well known, there are several variations depending on the intended application. The experimental conditions of each network for this task are as follows:

- *RNN:* The input-output relationship of the Elman network consisting of the input, hidden, context and

output layers can be expressed as follows:

$$\begin{cases} \boldsymbol{x}(t+1) = f(\boldsymbol{W}^{in}\boldsymbol{s}(t+1) + \boldsymbol{W}\boldsymbol{x}(t)) \\ \boldsymbol{\sigma}(t+1) = \boldsymbol{W}^{out}\boldsymbol{x}(t+1) \end{cases}, \tag{14}$$

where $\boldsymbol{s}(t) \in \mathbb{R}^L$, $\boldsymbol{x}(t) \in \mathbb{R}^M$ and $\boldsymbol{\sigma}(t) \in \mathbb{R}^N$ are the external input vector, the output vector from the hidden layer and the network output vector, respectively. Moreover, $\boldsymbol{W}^{in} \in \mathbb{R}^{M\times L}$ is the input weight matrix representing the connection weight from the input layer to the hidden layer, $\boldsymbol{W} \in \mathbb{R}^{M\times M}$ is the connection weight matrix from the context layer to the hidden layer, $\boldsymbol{W}^{out} \in \mathbb{R}^{N\times M}$ is the connection weight matrix from the hidden layer to the output layer and $f(\cdot)$ is an element-wise activation function of the neuron in the hidden layer. In the controller, the network output $\sigma(t)$ was employed to synthesise the control input $u(t)$ and the input of the RNN was the same vector used for the ESN. The activation function $f(\cdot)$ was the same as that for the ESN. Using the truncated BPTT algorithm, all connection weight matrices were trained offline with the training samples to minimise the cost function, defined by summing the squared error between the desired control input and the network output over the length of the single trial.

- *FNN:* The input-output relationship of the FNN consisting of the input, hidden and output layers can be expressed as follows:

$$\boldsymbol{\sigma}(t) = \boldsymbol{W}^{out} f(\boldsymbol{W}^{in}\boldsymbol{s}(t)), \tag{15}$$

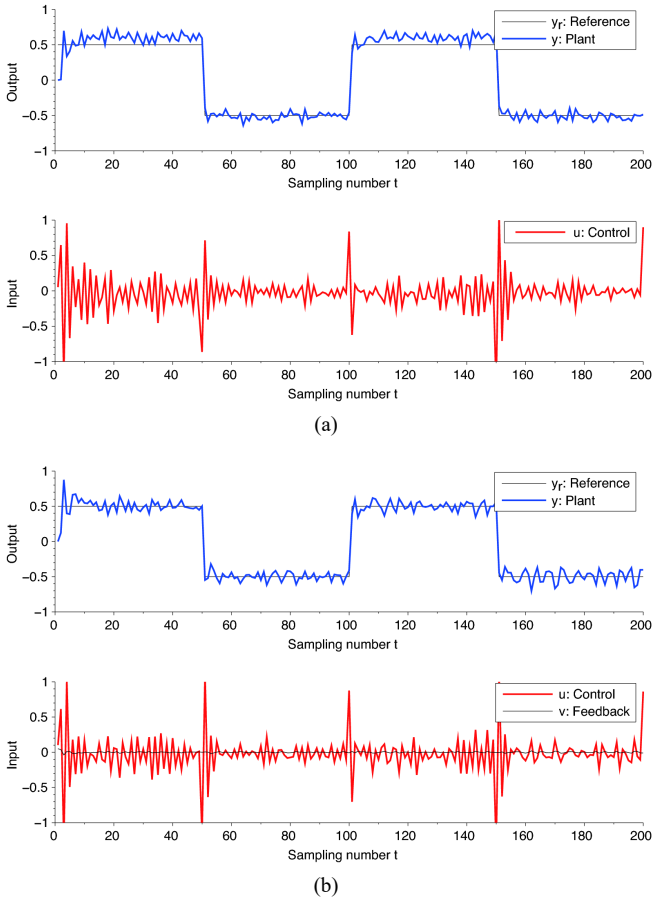
where $\boldsymbol{s}(t) \in \mathbb{R}^L$ and $\boldsymbol{\sigma}(t) \in \mathbb{R}^N$ are the external input vector and the network output vector, respectively. $\boldsymbol{W}^{in} \in \mathbb{R}^{M\times L}$ is the input weight matrix representing the connection weight from the input layer to the hidden layer, $\boldsymbol{W}^{out} \in \mathbb{R}^{N\times M}$ is the connection weight matrix from the hidden layer to the output layer and $f(\cdot)$ is an element-wise activation function of the neuron in the hidden layer. In the controller, the network output $\sigma(t)$ was employed to synthesise the control input $u(t)$ and the input of the FNN was defined based on the tapped delay line of the plant input-output sets: $\boldsymbol{s}(t) = [y_r(t+d)\,.\,y(t)\,.\,y(t-1)\,.\,u(t-1)]^{\mathrm{T}}$. The activation function $f(\cdot)$ was the same as that for the ESN. Applying the stochastic gradient descent method, all connection weight matrices were trained offline with the training samples to minimise the cost function to be the same as that for the RNN.

- *RBFN:* The input-output relationship of the RBFN consisting of the input, RBF and output layers can be expressed as follows:

$$\boldsymbol{\sigma}(t) = \boldsymbol{W}^{out}\boldsymbol{g}(\boldsymbol{s}(t)), \tag{16}$$

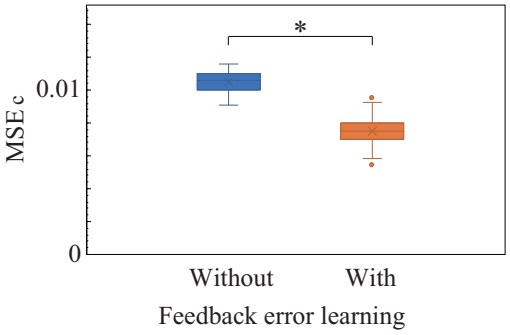
where $\boldsymbol{s}(t) \in \mathbb{R}^L$ and $\boldsymbol{\sigma}(t) \in \mathbb{R}^N$ are the external input vector and the network output vector, respectively; $\boldsymbol{W}^{out} \in \mathbb{R}^{N\times M}$ is the connection weight matrix from the RBF layer to the output layer. The i^{th} component of the vector $\boldsymbol{g}$ is defined by the RBF with centre $\boldsymbol{\mu}_i \in \mathbb{R}^L$ and distribution $\kappa_i \in \mathbb{R}^+$. In the controller, the network output $\sigma(t)$ was employed to synthesise the control input $u(t)$ and the input of the RBFN was the same vector used for the FNN. The RBF was given by a Gaussian function: $g_i(\boldsymbol{x}) = \exp(-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}_i\|^2}{\kappa_i^2})$. The RBF parameters were defined by k-means clustering using the training samples, after which the connection weight matrix was trained offline following the same procedure employed for the ESN.

Figure 4 Examples of the system response in controlling the ARMA model plant using the ESN-based controller, (a) control without feedback error learning (b) control with feedback error learning (see online version for colours)



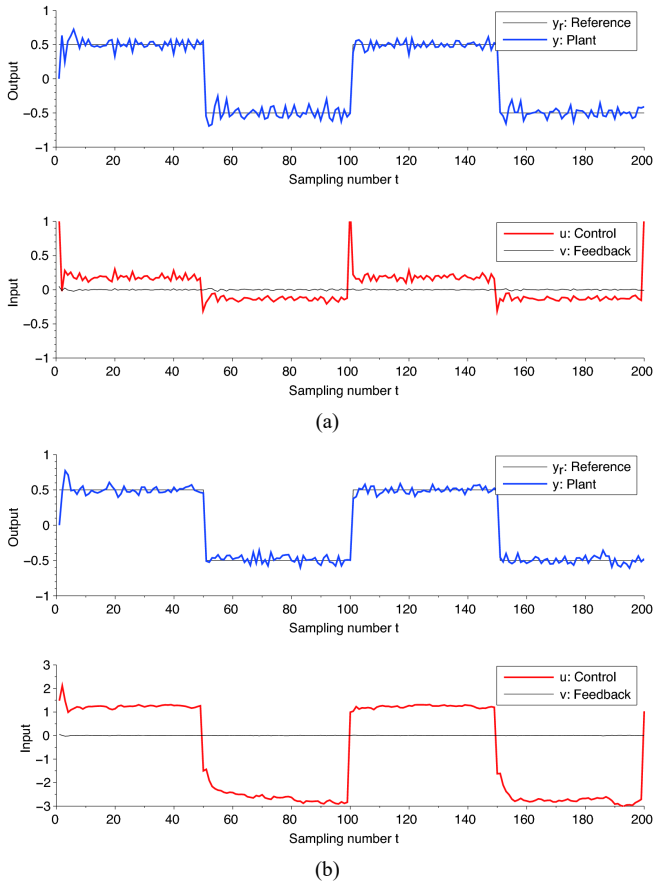
Notes: The top figure shows the system's output, where the reference and plant outputs are denoted by thin black and blue lines, respectively. The bottom figure shows the control input, where the control input and the feedback controller's output are denoted by red and thin black lines, respectively.

Figure 5 Control performance of the ESN-based controller with feedback error learning (see online version for colours)



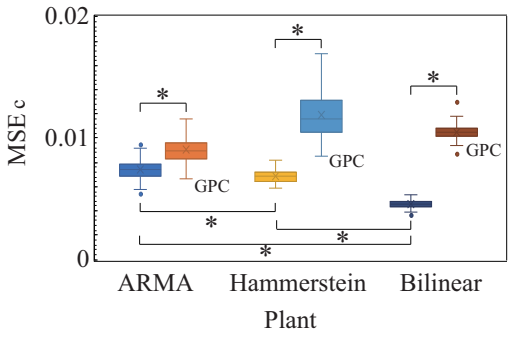
Note: *p < 0.05.

Figure 6 Examples of the system response in controlling the nonlinear plant using the ESN-based controller, (a) controlling the Hammerstein model plant (b) controlling the bilinear model plant (see online version for colours)



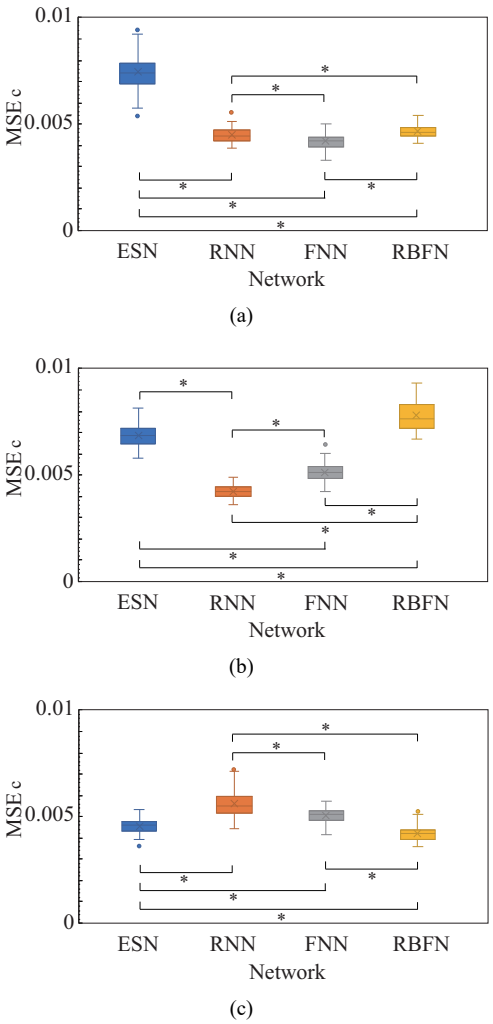
Notes: The top figure shows the system's output, where the reference and plant outputs are denoted by thin black and blue lines, respectively. The bottom figure shows the control input, where the control input and the feedback controller's output are denoted by red and thin black lines, respectively.

Figure 7 Comparison of the control performances of the ESN-based controller for different plants (see online version for colours)



Note: *p < 0.05.

Figure 8 Comparison of the controller performances among the networks, (a) controlling the ARAM model plant (b) controlling the Hammerstein model plant (c) controlling the bilinear model plant (see online version for colours)



Note: *p < 0.05.

In the identification experiment, which was conducted following the same procedure employed for the ESN, each network parameter was first tuned in advance using each network's MSE_{I} based on the result of the ten runs with the D10K, where $5 \leq M \leq 200$ in the RNN (the truncation length τ in the BPTT was $10 \leq \tau \leq 50$), $5 \leq M \leq 100$ in the FNN and $10 \leq M \leq 500$ in the RBFN. The results of the statistical test and the consideration of the computational cost indicated that the RNN used ten units in the hidden layer and ten units in the context layer, truncating the training sample to a length of 25; the FNN used ten units in the hidden layer; and the RBFN used 100 RBFs. The identification and control experiments were conducted in the same manner as those for the ESN; that is, after each network was trained with the D50K corresponding to each plant, the 100 runs using each network were conducted. Thereafter, each network-based controller's performance represented by MSE_{C} was evaluated. Here, the feedback error learning of the connection weight matrix $\boldsymbol{W}^{out}$ was implemented for each network-based controller to make the control experimental condition the same as that for the ESN. In Figures 8(a)–8(c), the results of the control experiment using each network-based controller are summarised. Statistical differences were observed between all networks for all plants according to the Steel-Dwass test, with a significance level of 5%. Investigations revealed that the control performance with the ESN does not always outperform those with the other networks even if the ESN could achieve accurate control. The RNN, FNN and RBFN approximate the mapping function of the target by preserving the information contained in the training data. This includes the intercorrelations and pattern structures distributed in the connection weight matrices and the RBF parameters by training the network. In contrast, the ESN approximates the computational task of the target using dynamic patterns, which are excited by the external input for a constant period within the reservoir, as the reference data. Accordingly, the readout matrix learns the task itself by adjusting the combination of such dynamic patterns rather than merely memorising the training patterns. Therefore, these results indicate that a good reservoir is necessary and important for attaining outstanding performance with the ESN even though the ESN has the advantage in training cost over the other networks. However, the design of such a reservoir in the control application would depend highly on the type of objective plant and the property of the training data.

4 Conclusions

This study explored the application of an ESN to a servo-level controller. A feedforward feedback controller, which ensured that the plant output matched the reference output, was designed using the ESN. Further, the ESN was trained offline in advance of using the ridge regression against the predetermined input-output datasets of the objective plant. The ESN was additionally trained in real time using the feedback error learning during the control task. Subsequently, computational experiments using nonlinear discrete-time systems, such as the ARMA model using a nonlinear term, the Hammerstein model and the bilinear model as the objective plant, were conducted to assess the characteristics of the ESN-based feedforward feedback controller. Simulation results confirmed the feasibility of applying the proposed controller and clarified the effectiveness of the feedback error learning in compensating for the control error associated with the ESN in the controller. Although the control performance of the ESN was not always better than those of the other networks, such as RNN, FNN and RBFN, the experimental result showed that the ESN-based controller exhibited good control accuracy.

Acknowledgements

This work was supported by JSPS KAKENHI Grant No. JP20K11980.

References

Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M. and Farhan, L. (2021) 'Review of deep learning: concepts, CNN architectures, challenges, applications, future directions', *Journal of Big Data*, Vol. 8, Article ID 53, 74pp.

Bala, A., Ismail, I., Ibrahim, R. and Sait, S. M. (2018) 'Applications of metaheuristics in reservoir computing techniques: a review', *IEEE Access*, Vol. 6, pp.58012–58029.

Denaï, M.A., Palis, F. and Zeghbib, A. (2007) 'Modeling and control of non-linear systems using soft computing techniques', *Applied Soft Computing*, Vol. 7, No. 3, pp.728–738.

Hagan, M.T., Demuth, H.B. and Jesús, O.D. (2002) 'An introduction to the use of neural networks in control systems', *International Journal of Robust and Nonlinear Control*, Vol. 12, No. 11, pp.959–985.

Jaeger, H. and Haas, H. (2004) 'Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication', *Science*, Vol. 304, No. 5667, pp.78–80.

Jordanou, J.P., Camponogara, E., Antonelo, E.A. and Aguiar, M.A.S. (2018) 'Nonlinear model predictive control of an oil well with echo state networks', in *Proceedings of the 3rd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production*, pp.13–18.

Khargonekar, P.P. and Dahleh, M.A. (2018) 'Advancing systems and control research in the era of ML and AI', *Annual Reviews in Control*, Vol. 45, pp.1–4.

Løvlid, R.A. (2013) 'A novel method for training an echo state network with feedback-error learning', *Advances in Artificial Intelligence*, Vol. 2013, Article ID 891501, 9pp.

Lu, Y. (2019) 'Artificial intelligence: a survey on evolution, models, applications and future trends', *Journal of Management Analytics*, Vol. 6, No. 1, pp.1–29.

Lukoševičius, M. (2012) 'A practical guide to applying echo state networks', *Neural Networks: Tricks of the Trade*, pp.659–686, Springer, Berlin, Heidelberg.

Lukoševičius, M. and Jaeger, H. (2009) 'Reservoir computing approaches to recurrent neural network training', *Computer Science Review*, Vol. 3, No. 3, pp.127–149.

Maass, W., Natschläeger, T. and Markram, H. (2002) 'Real-time computing without stable states: a new framework for neural computation based on perturbations', *Neural Computation*, Vol. 14, No. 11, pp.2531–2560.

Meireles, M.R.G., Almeida, P.E.M. and Simoes, M.G. (2003) 'A comprehensive review for industrial applicability of artificial neural networks', *IEEE Transactions on Industrial Electronics*, Vol. 50, No. 3, pp.585–601.

Ogawa, H. and Takahashi, Y. (2021) 'Echo state network based model predictive control for active vibration control of hybrid electric vehicle powertrains', *Applied Science*, Vol. 11, No. 14, Article ID 6621, 19pp.

Pan, Y. and Wang, J. (2012) 'Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks', *IEEE Transactions on Industrial Electronics*, Vol. 59, No. 8, pp.3089–3101.

Perrusquía A. and Yu, W. (2021) 'Identification and optimal control of nonlinear systems using recurrent neural networks and reinforcement learning: an overview', *Neurocomputing*, Vol. 438, pp.145–154.

Prieto, A., Prieto, B., Ortigosa, E.M., Ros, E., Pelayo, F., Ortega, J. and Rojas, I. (2016) 'Neural networks: an overview of early research, current frameworks and new challenges', *Neurocomputing*, Vol. 214, pp.242–268.

Ruano, A.E., Ge, S.S., Guerra, T.M., Lewis, F.L., Principe, J.C. and Colnarič, M. (2014) 'Computational intelligence in control', *Annual Reviews in Control*, Vol. 38, No. 2, pp.233–242.

Salehinejad, H., Sankar, S., Barfett, J., Colak, E. and Valaee, S. (2017) *Recent Advances in Recurrent Neural Networks*, arXiv:1801.01078, 21pp.

Salmen, M. and Ploger, P.G. (2005) 'Echo state networks used for motor control', in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp.1953–1958.

Sarker, I.H. (2021) 'Machine learning: algorithms, real-world applications and research directions', *SN Computer Science*, Vol. 2, Article ID 160, 21pp.

Schmidhuber, J. (2015) 'Deep learning in neural networks: an overview', *Neural Networks*, Vol. 61, pp.85–117.

Schrauwen, B., Verstraeten, D. and Campenhout, J.V. (2007) 'An overview of reservoir computing: theory, applications and implementations', in *Proceedings of the 15th European Symposium on Artificial Neural Networks*, pp.471–482.

Schwedersky, B.B., Flesch, R.C.C., Dangui, H.A.S. and Iervolino, L.A. (2018) 'Practical nonlinear model predictive control using an echo state network model', in *Proceedings of the 2018 International Joint Conference on Neural Networks*, pp.1–8.

Schwenzer, M., Ay, M., Bergs, T. and Abel, D. (2021) 'Review on model predictive control: an engineering perspective', *The International Journal of Advanced Manufacturing Technology*, Vol. 117, pp.1327–1349.

Sun, C., Song, M., Qiao, L. and Li, H. (2020) *A Review of Designs and Applications of Echo State Networks*, arXiv: 2012.02974.

Tanaka, G., Yamane, T., Hèroux, J.B., Nakane, R., Kanazawa, N., Takeda, S., Numata, H., Nakano D. and Hirose, A. (2019) 'Recent advances in physical reservoir computing: a review', *Neural Networks*, Vol. 115, pp.100–123.

Waegeman, T., Wyffels, F. and Schrauwen, B. (2012) 'Feedback control by online learning an inverse model', *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 23, No. 10, pp.1637–1648.

Xu, D., Lan, J. and Principe, J.C. (2005) 'Direct adaptive control: an echo state network and genetic algorithm approach', in *Proceedings of the 2005 IEEE International Joint Conference on Neural Networks*, Vol. 3, pp.1483–1486.

Yamada, T. (2009) 'Remarks on the tracking method of neural network weight change for a learning-type neural network feed-forward feed-back controller', *Artificial Life and Robotics*, Vol. 14, pp.384–387.

Yao, X., Fan, S., Zhao, B. and Cao, S. (2020) 'Controller design based on echo state network with delay output for nonlinear system', *Complexity*, Vol. 2020, Article ID 8643029, 6pp.

Zhang, H. and Vargas, D.V. (2023) 'A survey on reservoir computing and its interdisciplinary applications beyond traditional machine learning', *IEEE Access*, Vol. 11, pp.81033–81070.