Received: 4 December 2017 | Revised: 8 June 2018 | Accepted: 20 June 2018

DOI: 10.1002/acs.2916

RESEARCH ARTICLE

WILEY

Self-recurrent wavelet neural network–based identification and adaptive predictive control of nonlinear dynamical systems

Rajesh Kumar[1] | Smriti Srivastava[2] | J.R.P Gupta[2] | Amit Mohindru[3]

[1]Department of Instrumentation and Control Engineering, Bharati Vidyapeeth's College of Engineering, A-4, Paschim Vihar, New Delhi-110 063, India

[2]Division of Instrumentation and Control Engineering, Netaji Subhas Institute of Technology, New Delhi-110 078, India

[3]Department of Electronics and Communication Engineering, Indraprastha Institute of Information Technology, New Delhi-110 020, India

Correspondence
Rajesh Kumar, Department of Instrumentation and Control Engineering, Bharati Vidyapeeth's College of Engineering, A-4, Paschim Vihar, New Delhi-110 063, India.
Email: rajeshmahindru23@gmail.com

Summary

In this paper, the problem of simultaneous identification and predictive control of nonlinear dynamical systems using self-recurrent wavelet neural network (SRWNN) is addressed. The structure of the SRWNN is a modification of the wavelet neural network (WNN). Unlike WNN, the neurons present in the hidden layer of SRWNN contain the weighted self-feedback loops. Dynamic back-propagation algorithm is employed to derive the necessary parameter update equations. To further improve the convergence speed of the parameters, a time-varying (adaptive) learning rate is used. Four simulation examples are considered for testing the effectiveness of the proposed method. Furthermore, some disturbance rejection tests are also performed on the proposed method. The results obtained through the simulation study confirm the effectiveness of the proposed method.

KEYWORDS

adaptive predictive control, identification, self-recurrent wavelet neural network, time-varying learning rate

1 | INTRODUCTION

In most of the technological processes, we will find the dynamic systems that are characterized with uncertainties in their structure and in the values of their parameters. Deterministic models are not the viable choice for describing these uncertainties, and hence, the conventional control methods that are based on these models are likely to fail in providing the desired performance. Furthermore, the mathematical model of the plant is usually required in case of classical control methods for designing the controller. However, the plant's dynamics are usually complex and not fully understood. The inaccuracy of mathematical modeling of the plants usually degrades the performance of the controller, especially for the nonlinear and complex control problems.[1] For these reasons, various nonlinear control techniques have been developed to tackle the complexity of the nonlinear systems and to improve their performance. Nonlinear methods using soft computing techniques emerged out as the viable option. These bio-inspired techniques mimic the intelligent behavior of the living beings and are very useful in solving the nonlinear problems. Intelligent techniques such as fuzzy systems, neural networks, and genetic algorithms have been widely used in dealing with the control and identification of the nonlinear systems. In particular, there has been a significant development in the field of artificial neural networks (ANNs).[2-4] Features such as adaptive learning, parallelism in computation, and input-output mapping have made ANN the most important tool for system identification and control.[5-7] They are an integral building block of various identification and control strategies for a wide class of nonlinear systems.[8] Structurally, ANN can be classified into two categories: feed-forward neural network (FNN) and recurrent neural network (RNN). The major disadvantages of these ANNs are their requirement for heavy and complex computations, the weight update of ANNs does not utilize internal network information, and the function approximation is sensitive to the training data.[9] To solve these problems, wavelet neural network (WNN) were introduced. A WNN has a nonlinear regression structure that uses the localized basis functions in its hidden layer for achieving the desired input-output mapping. This makes WNN a superior system model than ANN and has been used by many researchers for solving various approximation, control, prediction, and classification type of problems.[10-13] The activation function of the hidden layer neurons of WNN is known as mother wavelet and it comes in a variety of forms. This hidden layer is then connected to the output layer neuron(s) with the help of the connection weights. The activation function used for the output layer neuron(s) is a linear function (as in the ANN case). The mother wavelet (wavelet function) consists of two parameters: translational and dilation factors. Depending upon which parameters of the WNN are to be adjusted, they have been classified into two categories.

1. **Adaptive WNN**: In this type, the unknown parameters include the output layer weights (also known as outer parameters of the network) and the translation and dilation factors (also known as inner parameters). These parameters are then adjusted using some parameter adjustment method.
2. **Fixed grid-type WNN**: In this type, the values of the inner parameters are set to some random values and remain fixed during the learning procedure. Only the outer layer weights are adjusted during the learning process. Gradient-type algorithms are not generally required to train such type of WNN.

The WNN includes the advantages of learning of ANN and multiresolution wavelets. They have been widely used in a nonlinear static function mapping and classification[14] and modeling and control of nonlinear dynamical systems.[15-17] Despite of these several uses, WNN has some shortcomings. First, because of the feed-forward structure, it can only be used for static type of mapping and will give poor performance in case of the dynamical systems. To use it for dynamical systems, a large number of neurons and time delay lines (TDLs) are to be used. This will increase the complexity of the network and, hence, increases the number of dimensions and parameters in the system. Furthermore, to find the number of neurons to be used in WNN is also a very difficult task. Second, they are not suitable for solving temporal type of problems (and hence, they require the use of TDLs). Third, the WNN parameters are usually trained using back-propagation (BP) algorithm that is based on the gradient descent method. It involves a learning rate whose value has a great influence on the convergence time and overall stability of the system. If it is set to a small value, then learning will be slow but system will remain stable. On the other hand if it is set to a higher value, then learning time will be less but system may become unstable. The first two shortcomings of WNN can be solved by using the self-recurrent wavelet neural networks (SRWNNs). It combines the properties of RNN such as attractor dynamics and the fast convergence of WNN. SRWNN hidden neurons contain local self-feedback loops, which provides it with the memory feature and the necessary information of past values of the signals, which allows it to handle the time-varying inputs, changes occurring in the control environment,[18] etc. The presence of temporal feature makes SRWNN structure simple since less number of neurons is required as compared to the WNN. The third problem can be solved by using an adaptive learning rate instead of the fixed one. It has been developed in our present paper. It offers two advantages: training time of the parameters tuning is reduced and the overall stability of the system is also ensured.

2 | RELATED WORK

Zhang and Benveniste[19] have demonstrated the use of WNN for identification of nonlinear system and compare its performance with that of multilayer FNN. The WNN was found to perform better than that of the multilayer FNN. Chen and Cheng[20] have used WNN to provide an adaptive control to the two-link robotic manipulator. The control law was designed so as to achieve the H_∞ tracking performance. Sureshbabu and Farrell[17] have used WNN for nonparametric nonlinear system identification problem. Lekutai[21] has proposed a self-tuning design method and has used WNN for controlling the output of the nonlinear systems. Xu and Ho[22] have applied WNN for system identification problem and used the Lyapunov stability method to provide the proof for the stability of the system. Tan et al[23] have used WNN for the system identification problem. They have proposed a PID-based BP algorithm and compared its performance with the standard BP algorithm. Ho et al[24] have developed a fuzzy-based WNN and used it for the system identification. de Sousa et al[25] have developed a WNN-based control scheme for controlling the robotic system and proved the stability of the system using the Lyapunov stability method. Lin et al[26] have presented the application of WNN for precision position control of linear ultrasonic motor drive system using the adaptive sliding-mode method. Srivastava et al[27] have proposed fuzzy-based WNN for dealing with the system identification and control problem. They have demonstrated that wavelets are able to suppress the noise and disturbance signal that might be present in the reference input signal and shown how the variation in the slope parameter of activation function can improve the quality of the output. Adeli and Jiang[28] have combined the WNN and fuzzy system and then used nonlinear autoregressive moving average with exogenous inputs approach for nonparametric identification. Abiyev and Kaynak[29] have combined fuzzy logic with WNN to design an adaptive controller. They have used gradient descent method for updating the various parameters of the proposed controller. Chen et al[30] have modified the structure of conventional WNN by replacing the output weight vector with the weighted linear combination of the input vector. They used this proposed network for solving the time series forecasting problem. Aadaleesan et al[31] have successfully coupled WNN with Laguerre basis filters in a Wiener-type model structure for the nonlinear system identification. Tzeng[32] has formulated a fuzzy WNN-based identification model and used genetic algorithm for tuning its parameters. The method is applied to identify the unknown dynamics of the nonlinear systems. Yousef et al[33] developed a WNN-based control scheme to control the DC motor. They have applied recursive least square method for online tuning of the controller parameters. Jahangiri et al[34] have used differential WNN and used it as an identification tool for approximating the unknown dynamics of the nonlinear systems. Ko[35] has used annealing dynamical learning algorithm for tuning the WNN parameters and used it for the system identification problem. Hsu[36] has developed a self-constructing WNN and used it to control the chaotic system and a DC motor. Okkan[37] has used WNN for predicting monthly reservoir inflow of dam and found its performance to be better than that of multilayer FNN. Xu and Liu[38] have used WNN as an identification model to predict the water quality of an intensive freshwater pearl breeding ponds in Duchang county, Jiangxi province, China, and compared its performance with that of Elman and multilayer feed-forward network. Zhang et al[39] have developed a recurrent WNN in which the current output and derivative of the current output along with the externally applied input signal are used as an input to the network. They have used it for the identification and control of nonlinear system. Lin et al[40] have proposed a WNN-based robust adaptive back-stepping control algorithm for the nonlinear system control. They have used Lyapunov stability method to derive the necessary parameter adjustment equations.

2.1 | Motivation

After doing the literature survey, it has been found that WNN is suitable for system identification and control. However, at the same time, it offers more number of parameters to be tuned and requires large number of TDLs, which makes it a computationally inefficient tool. Furthermore, it is better suited for the static type of systems. On the other hand, recurrent types of neural networks are proven to be more suitable for handling the complex dynamic systems. This motivated us to use a self-recurrent wavelet type of neural network for the system identification and control problem. Some of the major advantages of SRWNN are as follows: it requires lesser number of parameters to be tuned as compared to the WNN, it has the memory feature in the form of self-feedback weighted loops, and it offers a simpler structure by requiring lesser number of neurons.

2.2 | Contributions of the paper

1. Self-recurrent–type WNN structure is used as an identification and control tool and its detailed parameter update equations that are derived using the dynamic BP method.
2. A time-varying (adaptive) learning rate is included in the scheme for accelerating the convergence of the parameters.
3. An identification and predictive control scheme based on SRWNN is proposed.

2.3 | Main outline of the paper

Following the introduction section, we come to Section 3 in which the problem statement is defined. In Section 4, the mathematical foundation of the SRWNN is given. Section 5 gives the structural comparison between SRWNN and WNN. In Section 6, learning algorithm for identification process is given. In Section 7, the development of the time-varying learning rate is given. Section 8 involves the experimental simulation study that involves the identification of nonlinear systems. Two simulation examples are used for this purpose. In Section 9, introduction to predictive control-based on SRWNN is given. Section 10 involves the simulation study in which both control and identification steps are performed simultaneously. Simulation experiments are performed by considering an one-link robotic manipulator and an inverted pendulum system. Section 11 contains the discussion on the obtained results. In Section 12, conclusion is given.

3 | PROBLEM STATEMENT

Let the given plant be described by following dynamical difference equation:

$$Y_p(k+1) = F[Y_p(k), Y_p(k-1), \ \dots, Y_p(k-n+1)] + G[u_c(k), u_c(k-1), \ \dots, u_c(k-m+1)], \tag{1}$$

where $Y_P(k+1)$ is the one-step ahead output of the plant and $u_c(k)$ represents the control input to the plant. Furthermore, $F \rightarrow \mathfrak{R}$ and $G \rightarrow \mathfrak{R}$ are the linear function or the nonlinear function that are assumed to be differentiable. The term n represents the order of the plant. Let the stable reference model dynamics be described by the following difference equation:

$$Y_d(k+1) = M[Y_d(k), Y_d(k-1), \ \dots, Y_d(k-a+1)] + N[r(k), r(k-1), \ \dots, r(k-b+1)], \tag{2}$$

where $Y_d(k+1)$ denotes the one-step ahead output signal of the reference model, $r(k)$ denotes the externally applied reference input signal, and $a > 0$, $b > 0$. Furthermore, $M \rightarrow \mathfrak{R}$ and $N \rightarrow \mathfrak{R}$ are known and they could be linear or nonlinear differentiable functions. The dynamics of the proposed SRWNN-based identification model (identifier) is given by

$$Y_{\mathrm{SRWNN}}(k+1) = \hat{f}[r(k), Y_p(k)]. \tag{3}$$

Then, the objectives of this study are as follows.

1. To model the system using a SRWNN and identify its dynamics and network parameters (the weights and the translation and dilation factors), such that as the training progresses $Y_{\mathrm{SRWNN}}(k+1) \rightarrow Y_p(k+1)$.
2. To design an output feedback control law such that the plant output tracks a given reference model signal. In other words, it is desired to obtain the controller's parameter update equations in such a way so that the output of the controller $u_c(k)$ can force the plant output to track the reference model output. Mathematically, it can be stated as follows: The plant with an input-output pair $Y_p(k), u_c(k)$ is given as in Equation 1 and the stable reference model specified with its input-output pair $Y_d(k), r(k)$ is given as in Equation 2 with reference input signal of the system $r(k)$. Then, the requirement is to design an iterative parameter update equations for the SRWNN-based controller so that the controller can generate the desired control signal sequence $u_c(k)$ that causes the control error to decrease, that is,

$$\lim_{k\rightarrow\infty} |Y_d(k+1) - Y_p(k+1)| \leq \epsilon, \tag{4}$$

   where $\epsilon \rightarrow 0$. The error between the plant output and the reference model output will be used in the iterative update equations for adjusting the parameters of the SRWNN-based controller. The mode of training will be online. The time-varying learning rate will be used in these update equations.

4 | THEORETICAL FOUNDATION OF SRWNN

There has been a great advancement in the development of new techniques for analyzing the nonstationary signals, particularly development of the short-term Fourier transform, a variation of which known as Gabor transform was first published in 1946. The main contribution in the wavelet theory was the discovery of a smooth mother wavelets that form the orthonormal basis for $L^2(\mathfrak{R})$ using its discrete dilation and translation set, where $\mathfrak{R}$ denotes a set of real numbers and L^2 represents the set of all functions, f, which contain a bounded energy and, hence, satisfy the following condition:

$$\int_{-\infty}^{\infty} |f(t)|^2 \mathrm{d}t < \infty. \tag{5}$$

In the case of Gabor transform, no orthonormal basis can be generated using the smooth wavelets. The mother wavelets are required to satisfy the following conditions:

$$\int_{-\infty}^{\infty} |\phi(t)|^2 \mathrm{d}t < \infty \tag{6}$$

and

$$c_\phi = 2\pi \int_{-\infty}^{\infty} \frac{|\boldsymbol{\phi}(w)|^2}{|w|} \mathrm{d}w < \infty, \tag{7}$$

where $\boldsymbol{\phi}(w)$ represents the Fourier transform of mother wavelet $\phi(t)$. Equation 6 implies that mother wavelet must have the finite energy and Equation 7, which is an admissibility condition, implies that, if $\boldsymbol{\phi}(w)$ is a smooth function, then $\boldsymbol{\phi}(w)|_{w=0} = 0$. Symbol c_ϕ represents the admissibility constant for the function $\phi(t)$. Now, we will be finding suitable functions $\phi : \mathfrak{R}^n \to \mathfrak{R}$ having the following property. A denumerable family of the following form exist:

$$\Phi = \left\{ \det\left(D_k^{\frac{1}{2}}\right) \phi\left[D_k(\mathbf{x} - \mathbf{t}_k)\right] : \mathbf{t}_k \in \mathfrak{R}^n,\ D_k = \mathrm{diag}(\lambda_k),\ \ \lambda_k \in \mathfrak{R}_+^n \right\}. \tag{8}$$

Here, $\mathbf{t}_k$'s denotes the translation vectors and λ_k's are the dilation vectors that constitute the diagonal dilation matrices, D_k. Equation 8 satisfies the following frame property. Let there exist two constants: $c_{\min} > 0$ and $c_{\max} < \infty$, so that, for all f in $L^2(\mathfrak{R}^n)$, the following inequality holds:

$$c_{\min}||f||^2 \leq \sum_{\psi \in \Phi} |< \psi, f >|^2 \leq c_{\max}||f||^2. \tag{9}$$

In this in-equation, $< .,. >$ denotes an inner product in $L^2(\mathfrak{R}^n)$ and this sum has a range that covers all the elements of the family Φ. Furthermore, Equation 9 suggests that the family of Φ is dense in $L^2(\mathfrak{R}^n)$ and the quality of approximation ability depends upon how far the values of $c_{\min}$ and $c_{\max}$ are from 1. Now, if we sum all the linear combinations of the elements of the frame Φ, we will get

$$G(\mathbf{x}) = \sum_{k=1}^{N} w_k(k)\psi_k(\mathbf{x}),\ \ \psi_k \in \Phi, \tag{10}$$

which is dense in $L^2(\mathfrak{R}^n)$. Equation 10 is referred to as wavelet decomposition network. The corresponding wavelet network can be obtained by the collection of all finite sums of the following form:

$$G(\mathbf{x}) = \sum_{i=1}^{N} w_i(k) \left\{ \det\left(D_k^{\frac{1}{2}}\right) \phi\left[D_k(\mathbf{x} - \mathbf{t}_k)\right] \right\}. \tag{11}$$

Wavelet decomposition network is a fixed grid type of network, whereas wavelet network is an adaptive type. Now, from WNN, we can construct the SRWNN by introducing the memory in its structure. Furthermore, it is to be noted that there exist a number of wavelet functions, which satisfy all the above stated assumptions and conditions. In our present paper, we have used the first derivative of Gaussian function $\phi(x) = -x \exp\left(\frac{-x^2}{2}\right)$ as the wavelet activation function.

Figure 1 shows the structure of SRWNN. It consists of four layers. The first layer accepts the externally applied M number of input signals and transmits them directly to the second layer. The second layer is a hidden layer. Each node (represented by dashed brown box) in this layer is known as wavelon and consists of neurons that are equal to the number of inputs, that is, for every element in the input vector, there is a corresponding neuron present in the wavelon. Each neuron in the wavelon is characterized by a mother wavelet function (also called as analyzing wavelet) and self-feedback loop.

The wavelet function is given by

$$\phi(x) = -x \exp\left(\frac{-x^2}{2}\right). \tag{12}$$

The output of any neuron $\phi_{ij}(k)$ present in any of the wavelon is obtained from its mother wavelet as follows:

$$\phi_j(z_{ij}(k)) = \phi\left(\frac{u_{ij}(k) - t_{ij}(k)}{\lambda_{ij}(k)}\right) \tag{13}$$

with

$$z_{ij}(k) = \frac{u_{ij}(k) - t_{ij}(k)}{\lambda_{ij}(k)}, \tag{14}$$

where the first subscript i denote the wavelon position and the second subscript j denote the neuron position in that wavelon. Notice that the subscript for the input signal $u_j(k)$ position is same as the subscript used for specifying the neuron position. Terms $t_{ij}(k)$ and $\lambda_{ij}(k)$ denote the translational and dilation factors values, respectively. Furthermore, the input signal to any neuron (of any wavelon) consists of the signal coming from the input layer plus its own unit delayed weighted output, that is,

$$u_j(k) = x_j(k) + w_{ij}^D(k)\phi_{ij}(k-1). \tag{15}$$

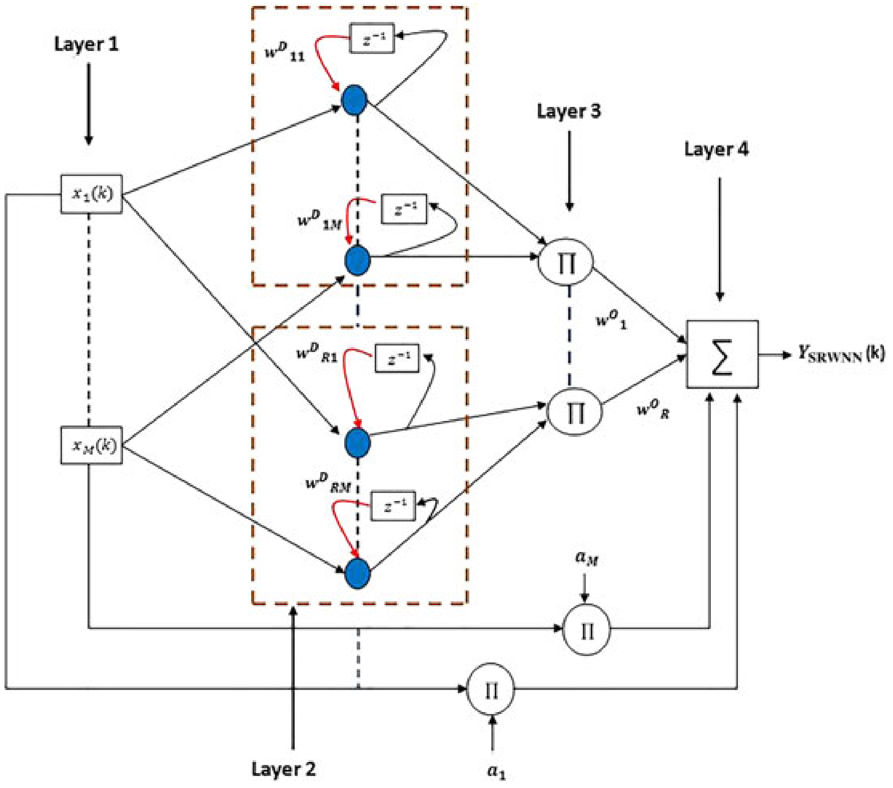

FIGURE 1 Structure of self-recurrent wavelet neural network (SRWNN) [Colour figure can be viewed at wileyonlinelibrary.com]

Here, $w_{ij}^D(k)$ represents the weighted self-feedback loop (denoted by a red arrow in hidden layer as shown in Figure 1). To explain the notation (subscripts) used in Equations 13, 14, and 15, consider an example. Let there be three inputs $u_1(k), u_2(k)$, and $u_3(k)$ applied to the hidden layer and let there be two wavelons present in the hidden layer. Since there are three inputs, each wavelon will consist of three neurons (one neuron corresponding to each input). Let n_1, n_2, and n_3 denote neurons present in the first wavelon and n_4, n_5, and n_6 denote neurons present in the second wavelon. Then, the outputs of the first wavelon neurons n_1, n_2, and n_3 are given by the following.

The output of neuron n_1 is

$$\phi_{11}(z_{11}(k)) = \phi\left(\frac{u_1(k) - t_{11}(k)}{\lambda_{11}(k)}\right) \tag{16}$$

with

$$z_{11}(k) = \frac{u_1(k) - t_{11}(k)}{\lambda_{11}(k)}. \tag{17}$$

The output of neuron n_2 is

$$\phi_{12}(z_{12}(k)) = \phi\left(\frac{u_2(k) - t_{12}(k)}{\lambda_{12}(k)}\right) \tag{18}$$

with

$$z_{12}(k) = \frac{u_2(k) - t_{12}(k)}{\lambda_{12}(k)}. \tag{19}$$

The output of neuron n_3 is

$$\phi_{13}\,(z_{13}(k)) = \phi\left(\frac{u_3(k) - t_{13}(k)}{\lambda_{13}(k)}\right) \tag{20}$$

with

$$z_{13}(k) = \frac{u_3(k) - t_{13}(k)}{\lambda_{13}(k)}. \tag{21}$$

Similarly, the output for the second wavelon neurons n_4, n_5, and n_6 are given by the following.

The output of neuron n_4 is

$$\phi_{21}\left(z_{21}(k)\right) = \phi\left(\frac{u_1(k) - t_{21}(k)}{\lambda_{21}(k)}\right) \tag{22}$$

with

$$z_{21}(k) = \frac{u_1(k) - t_{21}(k)}{\lambda_{21}(k)}. \tag{23}$$

The output of neuron n_5 is

$$\phi_{22}\left(z_{22}(k)\right) = \phi\left(\frac{u_2(k) - t_{22}(k)}{\lambda_{22}(k)}\right) \tag{24}$$

with

$$z_{22}(k) = \frac{u_2(k) - t_{22}(k)}{\lambda_{22}(k)}. \tag{25}$$

The output of neuron n_6 is

$$\phi_{23}\left(z_{23}(k)\right) = \phi\left(\frac{u_3(k) - t_{23}(k)}{\lambda_{23}(k)}\right) \tag{26}$$

with

$$z_{23}(k) = \frac{u_3(k) - t_{23}(k)}{\lambda_{23}(k)}. \tag{27}$$

Equation 16 to Equation 27 satisfy the general equations given by Equations 13 and 14. Hence, the range for subscript i will be from 1 to R (where R represents the total number of wavelons) and the range for subscript j per wavelon will be from 1 to M (where M is equal to number of neurons present in any wavelon. Note that the number of neurons present in each wavelon is equal and their count is equal to the number of input signals present in the input vector (the input signal count is denoted by M)). The dimension of translation factors, dilation factors, and self-feedback weighted loops will be equal to the product $i \times j$, the dimension of output weight vector will be equal to i where, as mentioned earlier, the symbol i represents the total number of wavelons and the symbol j represents the total number of neurons in any wavelon. In matrix form, they are shown as follows:

$$\textit{Translation factor matrix} = \begin{bmatrix} t_{11}(k) & t_{12}(k) & \cdots & t_{1M}(k) \\ t_{21}(k) & t_{22}(k) & \cdots & t_{2M}(k) \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ t_{R1}(k) & t_{R2}(k) & \cdots & t_{RM}(k) \end{bmatrix}. \tag{28}$$

Similarly, for the dilation factors, we have

$$\textit{Dilation factor matrix} = \begin{bmatrix} \lambda_{11}(k) & \lambda_{12}(k) & \cdots & \lambda_{1M}(k) \\ \lambda_{21}(k) & \lambda_{22}(k) & \cdots & \lambda_{2M}(k) \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \lambda_{R1}(k) & \lambda_{R2}(k) & \cdots & \lambda_{RM}(k) \end{bmatrix}. \tag{29}$$

For weighted self-feedback loops,

$$\textit{Dilation factor matrix} = \begin{bmatrix} w^D_{11}(k) & w^D_{12}(k) & \cdots & w^D_{1M}(k) \\ w^D_{21}(k) & w^D_{22}(k) & \cdots & w^D_{2M}(k) \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ w^D_{R1}(k) & w^D_{R2}(k) & \cdots & w^D_{RM}(k) \end{bmatrix}. \tag{30}$$

Notice that, in Equations 28, 29, and 30, the (horizontal) rows represent the wavelons (thus, each row represents the parameters associated with the neurons of any particular wavelon) and the number of columns is equal to the number of neurons present in any wavelon. The output layer weights and the coefficients connecting directly the inputs from the first layer to the output layer neuron are given by the following matrices:

$$\textit{Output layer weights} = \begin{bmatrix} w_1^o(k) \\ w_2^o(k) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ w_R^o(k) \end{bmatrix} \tag{31}$$

and

$$\textit{Coefficients} = \begin{bmatrix} a_1(k) \\ a_2(k) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ a_M(k) \end{bmatrix}. \tag{32}$$

In the case of multidimensional SRWNN ($M > 1$), the overall output of each wavelon is equal to the product of the outputs of each of its neuron. This product operation takes place in the third layer (in the case of one-dimensional SRWNN ($M = 1$), we do not require this product operation as there will be single neuron present in every wavelon, and hence, the wavelon output is actually equal to its neuron output) and for any ith hidden wavelon it is computed as follows:

$$\psi_i(k) = \prod_{j=1}^{M} \left\{ - \left[\frac{u_j(k) - t_{ij}(k)}{\lambda_{ij}(k)} \right] \right\} \times \left\{ \exp \left[-\frac{1}{2} \left(\frac{u_j(k) - t_{ij}(k)}{\lambda_{ij}(k)} \right)^2 \right] \right\}. \tag{33}$$

Further note from Figure 1 that we have a weighted connection between the input signals and the output neuron(s). These weights are represented by $a_j(k)$ and the sum of these weighted input signals, $\sum_{j=1}^{M} x_j(k)a_j(k)$, is called as direct term. By using this direct term, the SRWNN has the advantages of a direct linear feed-through network such as the initialization of parameters of the network based on available plant's knowledge and enhanced the extrapolation outside of examples of the learning data sets.[41] If all the weights ($a_j(k)$) and the self-feedback loop weights are all set equal to zero, then SRWNN reduces to WNN. The final layer, which is the fourth layer, is known as the output layer. Its induced field is equal to the weighted sum of third layer outputs plus the weighted sum of the inputs values received from the first layer. This total sum constitutes the output of SRWNN. Thus,

$$Y_{\text{SRWNN}}(k) = \sum_{i=1}^{R} w_i^o(k)\psi_i(k) + \sum_{j=1}^{M} x_j(k)a_j(k), \tag{34}$$

where $w_i^o(k)$ represents the connection weights between third layer and forth layer. The adjustable parameters of SRWNN are given by

$$P_I = \left\{ a_j(k), w_i^o(k), w_{ij}^D(k), t_{ij}(k), \lambda_{ij}(k) \right\}. \tag{35}$$

5 | STRUCTURAL ANALYSIS OF SRWNN AND WNN

In this section, a mathematical formulation of SRWNN and WNN models is presented, which will aid in computing the number of parameters associated with these structures.

Lemma 1. *Let M, R, and T represents the total number of input signals applied to the first layer, total number of hidden nodes (wavelons) present in the second layer, and the total number of neurons present in the fourth layer, respectively, then the number of adjustable parameters in SRWNN can be computed from the following expression:*

$$N^{\mathrm{SRWNN}} = 2MR + MR + TR + M = R(3M + T) + M. \tag{36}$$

The development of Equation 36 is as follows. We know that each wavelon consists of neurons that are equal to the number of elements in the input signal vector. Each neuron is associated with adjustable dilation and translational factor, which means each neuron present in the wavelon contributes two adjustable parameters, and hence, from each wavelon, we will have 2M number of adjustable parameters. Thus, if R are number of wavelons, then the total number of adjustable translation and dilation factors is equal to 2MR. Each neuron present in the wavelon also has the adjustable feedback weight (denoted by red arrow in the hidden layer in Figure 1) so the total number of feedback weights from each wavelon will be M, and hence, the total feedback weights from all the wavelons will be MR. The output weight vector consists of output weights, which connects third layer output to the neurons of output layer. Thus, if single neuron is present in the output layer, then there will be R number of output weights, and for T number of output neurons, the total number of adjustable output layer weights will be TR. Furthermore, the direct term consists of the weighted sum of input signal that is the part of the output of SRWNN. Hence, we will have M number of adjustable coefficients in the direct term.

For the case of WNN, the number of adjustable parameters can be found from the following expression:

$$N^{\mathrm{WNN}} = R(2M + T). \tag{37}$$

The development of Equation 37 is as follows. In WNN, there will not be any adjustable feedback weights in the second layer and there will not be any direct term. Thus, the only adjustable parameters in WNN case are dilation and translation factors that are equal to 2MR plus the output layer weights, which are equal to TR.

Lemma 2. *The output of the WNN can be evaluated as follows:*

$$Y_{\mathrm{WNN}}(k) = \sum_{i=1}^{R} w_i^o(k)\psi_i(k) \tag{38}$$

$$u_j(k) = x_j(k). \tag{39}$$

If we compare Equation 39 with Equation 15, we can see that SRWNN is a dynamic mapping as the hidden nodes use their own past outputs for evaluating their current outputs, whereas WNN is a static mapping.

6 | IDENTIFICATION OF NONLINEAR SYSTEMS USING SRWNN

We have used series-parallel type of identification structure. Its details can be found in the works of Kumar et al.[6,7] The structure of the SRWNN-based identifier is shown in Figure 2. The identification model consists of SRWNN and tapped delayed lines (modeled as z^{-1}). Only two inputs are considered for the SRWNN identifier. These includes current externally applied input $r(k)$ and the most recent plant's output $Y_P(k)$. The rationale of choosing only two inputs for SRWNN is to test its memory feature and to have lesser number of parameters to be tuned. This also results in a simpler structure. Thus, one-step ahead output of SRWNN identifier is given by

$$Y_{\mathrm{SRWNN}}(k+1) = \hat{f}\{r(k), Y_p(k)\}. \tag{40}$$

The function $\hat{f}$ approximates the unknown dynamical system's mathematical model. In the case of WNN-based identification model, its input vector consists of those signals that form the argument of the unknown function given in the plant's mathematical model.

6.1 | Learning algorithm for SRWNN-based identification model

To update the various parameters of SRWNN, we have used dynamic BP method based on gradient descent approach. For this, we first define a cost function. This cost function is defined as follows:

$$E_i(k) = \frac{1}{2}e_i^2(k), \tag{41}$$

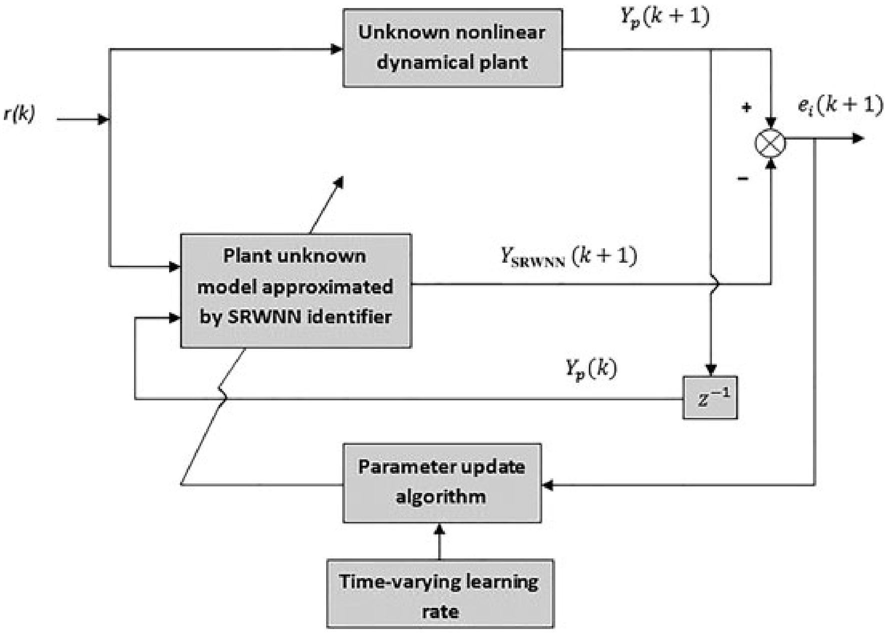
Unknown nonlinear dynamical plant
$Y_p(k+1)$
$r(k)$
+
$e_i(k+1)$
-
Plant unknown model approximated by SRWNN identifier
$Y_{\text{SRWNN}}(k+1)$
$Y_p(k)$
z^{-1}
Parameter update algorithm
Time-varying learning rate

FIGURE 2 Structure of self-recurrent wavelet neural network (SRWNN)–based identifier (proposed)

where $e_i(k) = Y_P(k) - Y_{\text{SRWNN}}(k)$ is the identification error and subscript i in $E_i(k)$ and $e_i(k)$ denotes the identification process. Now, by applying the chain rule in a recursive manner, each parameter will be adjusted so as to minimize the cost function value in every iteration.

6.1.1 | Update equation for output weight vector

The adjustment equation for any jth output weight of SRWNN is obtained as follows:

$$\frac{\partial E_i(k)}{\partial w_j^o(k)} = \frac{\partial E_i(k)}{\partial Y_{\text{SRWNN}}(k)} \times \frac{\partial Y_{\text{SRWNN}}(k)}{\partial w_j^o(k)} \tag{42}$$

or

$$\frac{\partial E_i(k)}{\partial w_j^o(k)} = -e_i(k)\psi_j(k). \tag{43}$$

Thus, the update equation for any jth output weight is given by

$$w_j^o(k+1) = w_j^o(k) - \eta_i(k)\Delta w_j^o(k), \tag{44}$$

where $\Delta w_j^o(k) = \frac{\partial E_i(k)}{\partial w_j^o(k)}$. The term $\eta_i(k)$ represents the learning rate and subscript i refers to the identification process.

6.1.2 | Update equation for the self-feedback weight vector

The adjustment equation for any (ij)th self-feedback weight of SRWNN is obtained as follows:

$$\frac{\partial E_i(k)}{\partial w_{ij}^D(k)} = \frac{\partial E_i(k)}{\partial Y_{\text{SRWNN}}(k)} \times \frac{\partial Y_{\text{SRWNN}}(k)}{\partial w_{ij}^D(k)}. \tag{45}$$

Thus, the update equation for any (ij)th feedback weight in the hidden layer is given by

$$\frac{\partial E_i(k)}{\partial w_{ij}^D(k)} = -e_i(k)\phi_{ij}(k-1)\frac{w_j^o(k)}{\lambda_{ij}(k)}\left[z_{ij}^2(k) - 1\right] \times \exp\left(-\frac{z_{ij}^2(k)}{2}\right). \tag{46}$$

6.1.3 | Update equation for the connection weights present between the input and output layer

Any $a_j(k)$ connection weight that links the $x_j(k)$ input to the output layer is updated as follows:

$$\frac{\partial E_i(k)}{\partial a_j(k)} = \frac{\partial E_i(k)}{\partial Y_{\mathrm{SRWNN}}(k)} \times \frac{\partial Y_{\mathrm{SRWNN}}(k)}{\partial a_j(k)} \tag{47}$$

or

$$\frac{\partial E_i(k)}{\partial a_j(k)} = -e_i(k)x_j(k). \tag{48}$$

Thus, we have the following update equation for $a_j(k)$ connection weight:

$$a_j(k+1) = a_j(k) - \eta_i(k)\Delta a_j(k), \tag{49}$$

where $\Delta a_j(k) = \frac{\partial E_i(k)}{\partial a_j(k)}$.

6.1.4 | Update equations for translation and dilation factors

The update equation of any (ij)th translation factor associated with neuron of any jth wavelon can be obtained as follows:

$$\frac{\partial E_i(k)}{\partial t_{ij}(k)} = \frac{\partial E_i(k)}{\partial Y_{\mathrm{SRWNN}}(k)} \times \frac{\partial Y_{\mathrm{SRWNN}}(k)}{\partial t_{ij}(k)} \tag{50}$$

or

$$\frac{\partial E_i(k)}{\partial t_{ij}(k)} = e_i(k) \times \frac{w_j^o(k)}{\lambda_{ij}(k)} \times Q \times \left[z_{ij}^2(k) - 1\right] \times \exp\left(-\frac{z_{ij}^2(k)}{2}\right), \tag{51}$$

where Q refers to the product of the outputs of all the neurons present in that jth wavelon excluding the output of that neuron (of jth wavelon) whose translational factor is being updated. Thus, we have the following update equation for $t_j(k)$ translation factor:

$$t_{ij}(k+1) = t_{ij}(k) - \eta_i(k)\Delta t_{ij}(k), \tag{52}$$

where $\Delta t_{ij}(k) = \frac{\partial E_i(k)}{\partial t_{ij}(k)}$.

Similarly, update equation of any (ij)th dilation factor associated with neuron of any jth wavelon can be written as

$$\frac{\partial E_i(k)}{\partial \lambda_{ij}(k)} = \frac{\partial E_i(k)}{\partial Y_{\mathrm{SRWNN}}(k)} \times \frac{\partial Y_{\mathrm{SRWNN}}(k)}{\partial \lambda_{ij}(k)} \tag{53}$$

or

$$\frac{\partial E_i(k)}{\partial \lambda_{ij}(k)} = e_i(k)z_{ij}(k)\frac{w_j^o(k)}{\lambda_{ij}(k)} \times Q \times \left[z_{ij}^2(k) - 1\right] \times \exp\left(-\frac{z_{ij}^2(k)}{2}\right). \tag{54}$$

Thus, we have the following update equation for any $\lambda_{ij}(k)$ dilation factor:

$$\lambda_{ij}(k+1) = \lambda_{ij}(k) - \eta_i(k)\Delta\lambda_{ij}(k), \tag{55}$$

where $\Delta\lambda_{ij}(k) = \frac{\partial E_i(k)}{\partial \lambda_{ij}(k)}$.

7 | DEVELOPMENT OF TIME-VARYING LEARNING RATE

In dynamic BP method, the learning rate value plays an important role. If its value is fixed, then the convergence of parameters to their desired values can be slow and it may sometimes produce suboptimal solutions. Thus, in order to improve the performance of BP method, we need to make learning rate adaptive in nature. Thus, we will change its value in each iteration so as to achieve the faster convergence. For implementing adaptive learning rate, we need to develop a mathematical model that will adjust its value in each iteration. The procedure is as follows: If present value of cost function is smaller than its previous instant value, then we can slightly increase the learning rate value. On the other hand, if the current cost function value is greater than its previous instant value, then it indicates that the system may become unstable, and hence, we need to decrease the value of learning rate. Mathematically, these two cases can be implemented as follows.

Lemma 3. *Let $e_r(k)$ be a relative error that is defined as*

$$e_r(k) = \frac{E(k) - E(k-1)}{E(k)}, \tag{56}$$

where $E(k)$ is the cost function (mean square error). Then, the proposed adjustment to be made in the learning rate is given as follows.

Case 1. For $e_r(k) < 0$,

$$\eta_i(k+1) = \eta_i(k)\left[1 + \mu \times \exp\left(\frac{-e_r(k)}{2}\right)\right], \mu \in (0,1]. \tag{57}$$

Case 2. For $e_r(k) \geq 0$,

$$\eta_i(k+1) = \eta_i(k)\left[1 - \mu \times \exp\left(\frac{-e_r(k)}{2}\right)\right], \mu \in (0,1]. \tag{58}$$

Note that, in Equations 57 and 58, we have included a parameter μ. Its role is to maintain the stability of the system even when the exponential term goes a drastic change in the value. This will eventually cause a large change in the learning rate value and the system may become unstable. The presence of μ avoids such situations to happen. The value of μ is kept fixed and is considered equal to 0.0021 in all the simulation examples.

8 | SIMULATION STUDY—IDENTIFICATION

In this section, we have considered two dynamical nonlinear systems whose dynamics are to be approximated by SRWNN- and WNN-based identifiers. The performances of these identifiers are studied and compared in terms of their ability in approximating the dynamics of given system, the number of parameters required to be tuned, and the minimum value of **A**verage **M**ean **S**quare **E**rror (AMSE) obtained during the training.

8.1 | Example 1

Consider a nonlinear dynamical system that is defined by the following difference equation[5]:

$$Y_P(k+1) = f\left[Y_P(k), Y_P(k-1), Y_P(k-2), r(k), r(k-1)\right]. \tag{59}$$

The unknown nonlinear function *f* has the following form:

$$f\left[x_1, x_2, x_3, x_4, x_5\right] = \frac{x_1 x_2 x_3 x_5 \left(x_3 - 1\right) + x_4}{1 + x_2^2 + x_3^2}, \tag{60}$$

where $x_1 = Y_P(k), x_2 = Y_P(k-1), x_3 = Y_P(k-2), x_4 = r(k)$, and $x_5 = r(k-1)$. Let the initial value of $\eta_i(k)$ be set equal to 0.002. The input vector of SRWNN consists of two inputs $\{r(k), Y_P(k)\}$. Thus, the mathematical model of SRWNN-based identification model is given by

$$Y_{\text{SRWNN}}(k+1) = \hat{f}\left[x_1, x_4\right] \tag{61}$$

and the dynamics of the WNN-based identification model is given by

$$Y_{\text{WNN}}(k+1) = \hat{g}[x_1, x_2, x_3, x_4, x_5]. \tag{62}$$

The objective is $\hat{f} \approx \hat{g} \approx f$. Total of 800 input-output training samples are used during the training, which is continued for 800 epochs. The external input considered in this example is $r(k) = \sin\left(\frac{2\pi k}{25}\right)$. The responses of SRWNN- and WNN-based identifiers at the end of the training are shown in Figure 3.

It can be easily seen from the Figures that the response obtained from the SRWNN identifier is very close to the plant's response as compared with the response obtained with the WNN identifier. The corresponding AMSE obtained during the training of SRWNN and WNN identifiers are shown in Figure 4. It can be seen that AMSE of SRWNN obtained with an adaptive learning rate is much smaller as compared with its value obtained with that of the fixed learning rate. Furthermore, the AMSE obtained with WNN took more time to attain the zero value. The comparative analysis between SRWNN and WNN identifiers is also shown in Table 1. The important thing to be noted from the Table is that the numbers

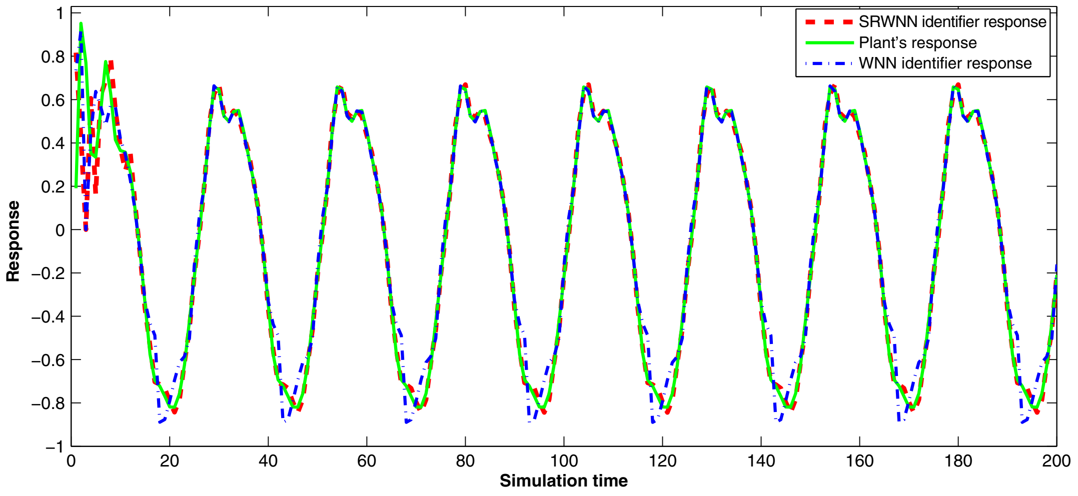

FIGURE 3 Response of self-recurrent wavelet neural network (SRWNN) and wavelet neural network (WNN) identifiers at the end of training (Example 1) [Colour figure can be viewed at wileyonlinelibrary.com]

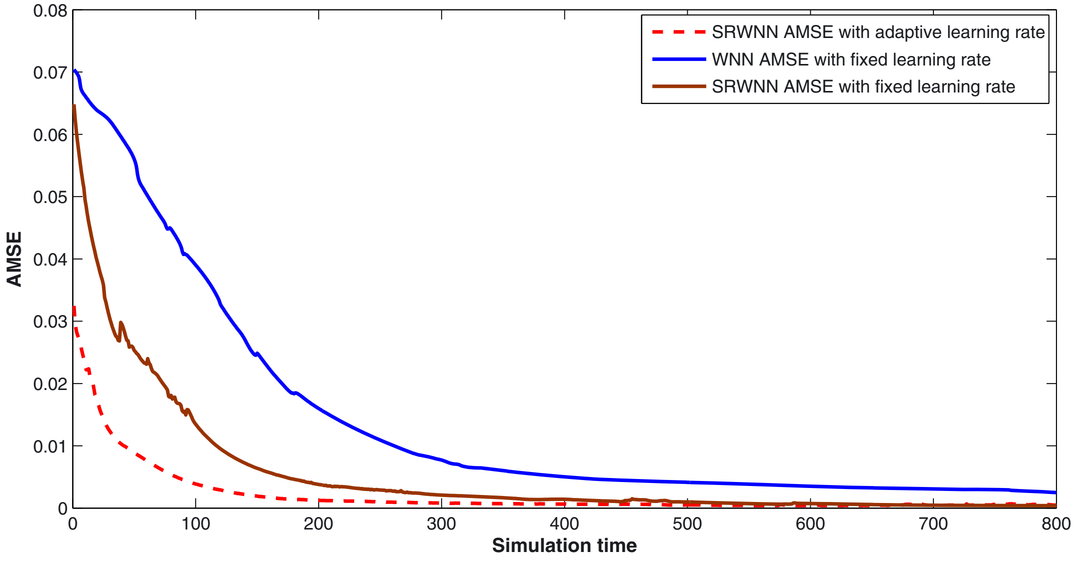

FIGURE 4 Average mean square error (AMSE) of self-recurrent wavelet neural network (SRWNN) and wavelet neural network (WNN) identifiers during the training (Example 1) [Colour figure can be viewed at wileyonlinelibrary.com]

of parameters required to be tuned in the case SRWNN identifier are much lesser as compared with WNN-based identifier. Furthermore, the AMSE value obtained with the SRWNN identifier is much smaller as compared with its value obtained with the WNN identifier. This makes the SRWNN a more viable choice for the system identification problem.

TABLE 1 Performance comparison of self-recurrent wavelet neural network (SRWNN) and wavelet neural network (WNN) identifiers (Example 1)

	SRWNN Identifier	SRWNN Identifier	WNN Identifier
Number of inputs	02	02	05
Number of nodes in hidden layer	10	10	10
Number of parameters to be adjusted	72	72	110
Learning rate	Adaptive	0.002	0.002
Training samples in each epoch	800	800	800
Total number of epochs	1000	1000	1000
Minimum average mean square error	0.000596	0.00325	0.0051

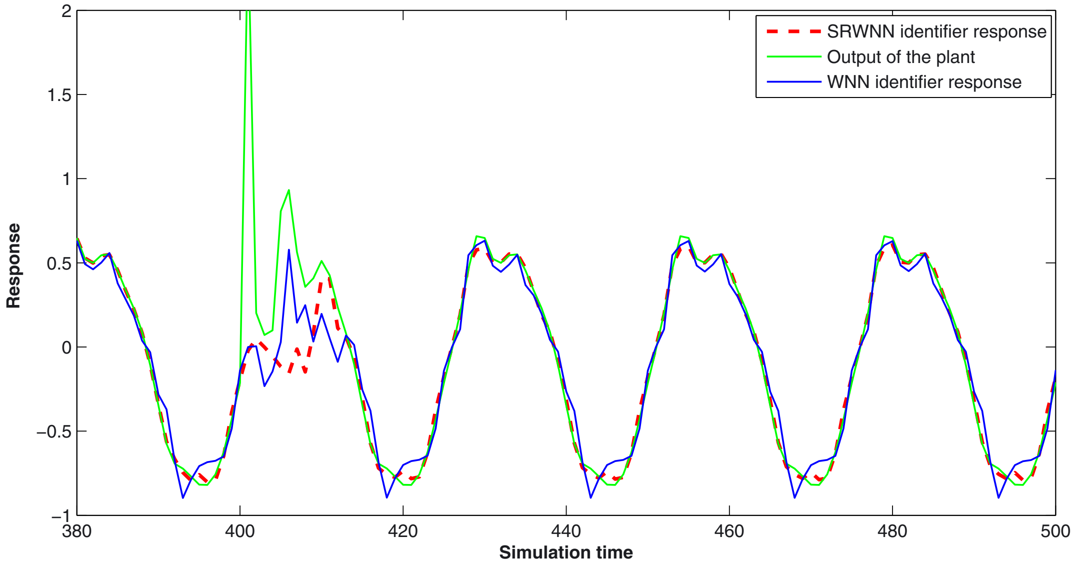

FIGURE 5 Response of self-recurrent wavelet neural network (SRWNN) and wavelet neural network (WNN) identifiers when disturbance signal is added into the system (Example 1) [Colour figure can be viewed at wileyonlinelibrary.com]

8.1.1 | Disturbance rejection tests

Now, we will test whether the proposed SRWNN identifier is able to recover from the external disturbance signal or not. For doing this experiment, we have considered a disturbance signal of magnitude 2.5 units and added it in the output of the plant at 400th time instant during the last 800th epoch. The corresponding responses of SRWNN and WNN identifiers are shown in Figure 5. It can be seen from the Figure that the response of the SRWNN recovered slightly better than the improvement obtained in the WNN identifier output. This shows that SRWNN is more capable in quickly recovering from the effects of the disturbance signal as compared to the WNN.

8.1.2 | Testing of the performance of SRWNN-based identifier with an input containing sinusoid of different frequencies

Considering the same system, we will now be testing the performance of the proposed scheme by considering a different input signal that consists of sinusoidal signals having different frequencies. It is defined as follows:

$$r(k) = 0.54 \sin\left(\frac{5\pi k}{250}\right) + 0.4 \cos\left(\frac{\pi k}{10}\right) \quad \text{for} \quad 0 < k \leq 400 \tag{63}$$

and

$$r(k) = 0.62 \cos\left(\frac{2\pi k}{140}\right) + 0.31 \sin\left(\frac{\pi k}{60}\right) \quad \text{for} \quad 400 < k \leq 800. \tag{64}$$

A total of 800 input-output samples are used during the training. The training is continued for 1000 epochs. The initial learning rate value is set equal to $\eta_i(k) = 0.0027$. The responses obtained from the SRWNN- and WNN-based identification models are shown in Figure 6 and Figure 7, respectively, and the corresponding AMSE obtained during the identifiers training is shown in Figure 8. From the Figures, it can be seen that the proposed SRWNN-based identification scheme have again given better results as compared with the results obtained with the WNN-based identification scheme and is able to approximate the plant's dynamics even when the externally applied input signal, $r(k)$, consists of sinusoids of different frequencies.

8.2 | Example 2

Consider a nonlinear plant whose dynamics are given as[42]

$$Y_P(k+1) = f\left[Y_P(k), Y_P(k-1), r(k), r(k-1), r(k-2)\right], \tag{65}$$

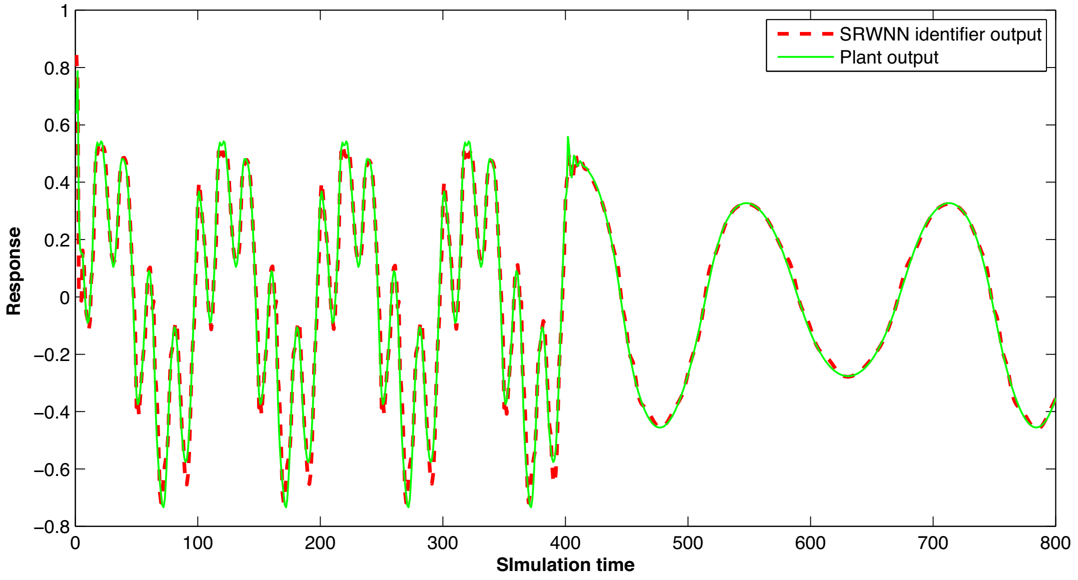

FIGURE 6 Response of the self-recurrent wavelet neural network (SRWNN) identifier obtained at the end of training with the new input (Example 1) [Colour figure can be viewed at wileyonlinelibrary.com]

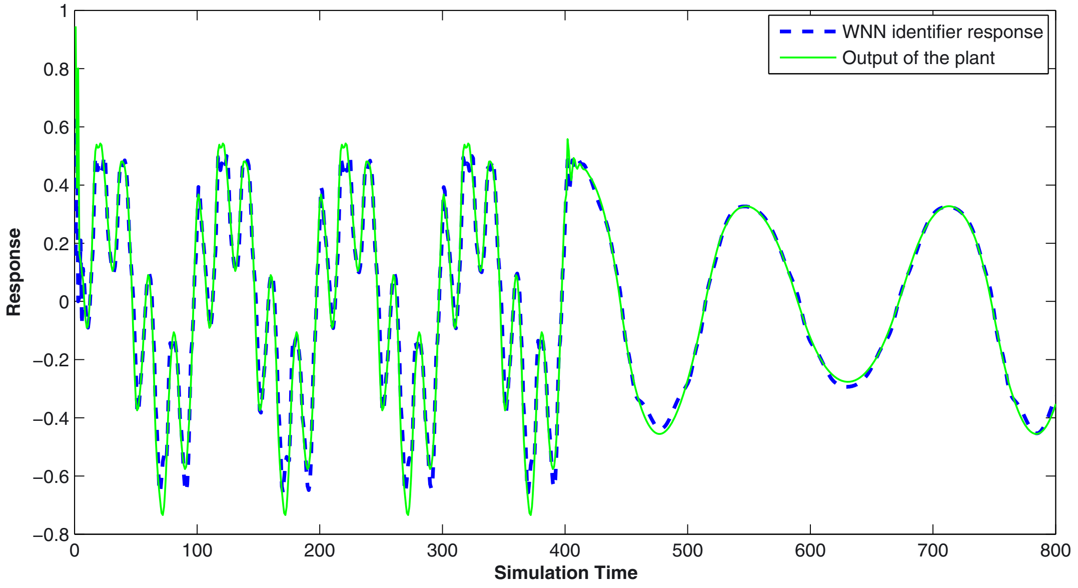

FIGURE 7 Response of the wavelet neural network (WNN) identifier obtained at the end of the training with the new input (Example 1) [Colour figure can be viewed at wileyonlinelibrary.com]

where the unknown function f, to be determined, has the following form:

$$f\,[x_1, x_2, x_3, x_4, x_5] = 0.72x_1 + 0.025x_2x_3 + 0.001x_4^2 + 0.2x_5, \tag{66}$$

where $x_1 = Y_P(k), x_2 = Y_P(k-1), x_3 = r(k), x_4 = r(k-1)$, and $x_5 = r(k-2)$.

Let the initial value of $\eta_i = 0.0053$. The mathematical model of SRWNN identifier is given by

$$Y_{\text{SRWNN}}(k+1) = \hat{f}\,[x_1, x_3]\,, \tag{67}$$

and the WNN-based identification model has the following form:

$$Y_{\text{WNN}}(k+1) = \hat{g}[x_1, x_2, x_3, x_4, x_5]\,. \tag{68}$$

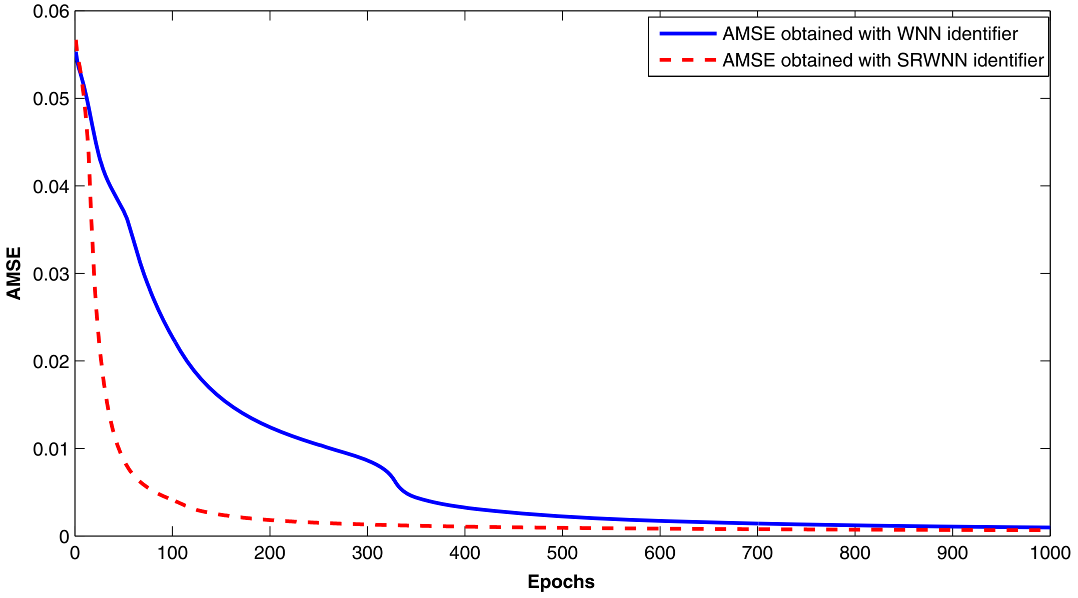

FIGURE 8 Average mean square error (AMSE) obtained of self-recurrent wavelet neural network (SRWNN) and wavelet neural network (WNN) identifiers during the training with new input (Example 1) [Colour figure can be viewed at wileyonlinelibrary.com]

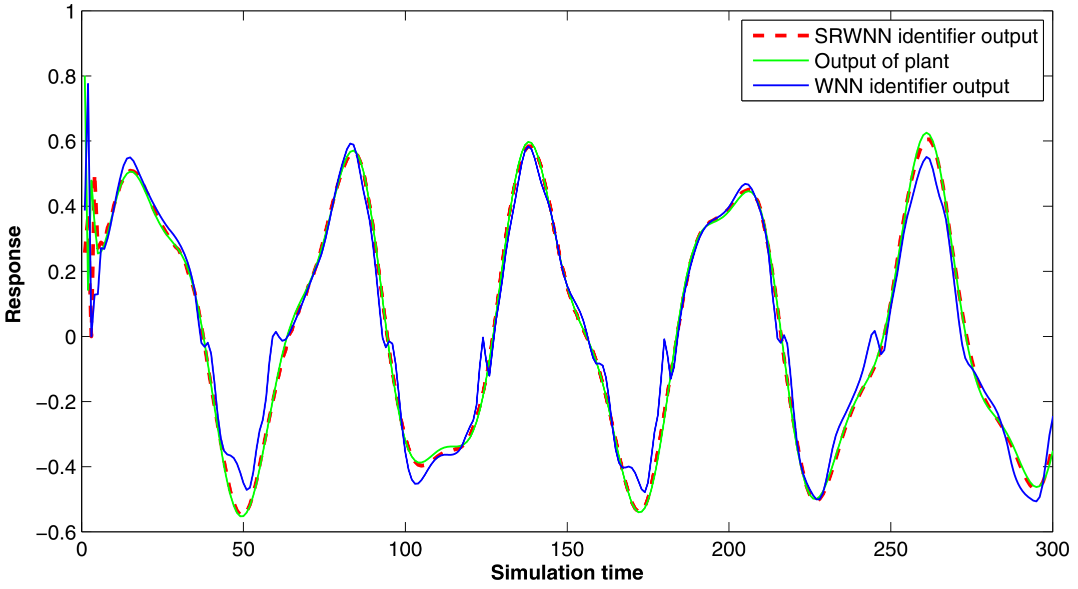

FIGURE 9 Response of self-recurrent wavelet neural network (SRWNN) identifier at the end of training (Example 2). WNN, wavelet neural network [Colour figure can be viewed at wileyonlinelibrary.com]

A total of 1000 epochs are used in the training and the externally applied input signal considered in this example is $r(k) = 0.2\sin\left(\frac{2\pi k}{25}\right) + 0.7\sin\left(\frac{\pi k}{30}\right)$. In each epoch, 800 input-output training samples are presented to the identifier networks. Figure 9 shows the responses of SRWNN and WNN identifiers, respectively, after the last epoch of training. From the Figure, it can be concluded that SRWNN identifier is giving better response that is very close to the plant's response (both with fixed and adaptive learning rate) as compared with the response obtained from the WNN identifier. The AMSE plots obtained during the learning is shown in Figure 10. It can be seen from the plots that the AMSE obtained with SRWNN identifier (both with fixed and adaptive learning rate) is decreasing rapidly toward zero as compared with AMSE obtained with WNN. This shows that SRWNN is approximating the dynamics of the plant at a much faster rate as compared with the speed of convergence obtained with WNN identifier. Furthermore, from Table 2, it can be seen that SRWNN identifier is giving very small AMSE values as compared with AMSE values obtained with WNN. In addition, the

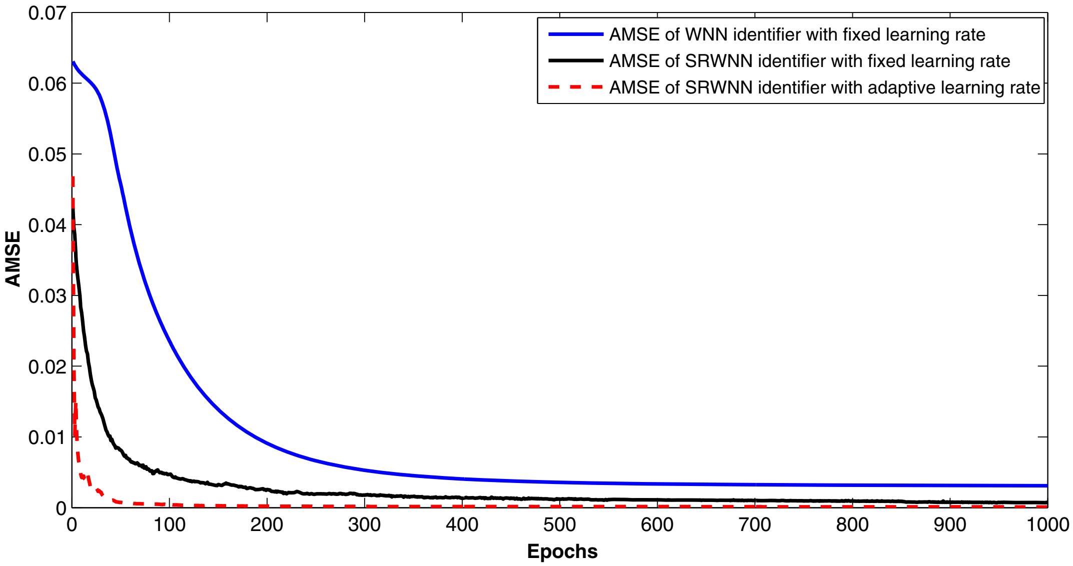

FIGURE 10 Average mean square error (AMSE) of self-recurrent wavelet neural network (SRWNN) and wavelet neural network (WNN) identifiers during the training (Example 2) [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 2 Performance analysis of self-recurrent wavelet neural network (SRWNN) and wavelet neural network (WNN) identifiers (Example 2)

	SRWNN Identifier	SRWNN Identifier	WNN Identifier
Number of inputs	02	02	05
Number of nodes in hidden layer	10	10	10
Number of parameters to be adjusted	72	72	110
Learning rate	Adaptive	0.0053	0.0053
Training samples in each epoch	800	800	800
Total number of epochs	1000	1000	1000
Minimum average mean square error	0.000102	0.000697	0.00321

number of parameters to be tuned in case of SRWNN is found to be much lesser as compared with number of parameters to be tuned in the case of the WNN identifier.

9 | PREDICTIVE CONTROL OF NONLINEAR DYNAMICAL SYSTEM USING SRWNN-BASED CONTROLLER

In this section, the aim is to develop a configuration in which both control and identification of nonlinear plant can be simultaneously done. The proposed predictive control and identification configuration is shown in Figure 11. The output of the SRWNN controller is denoted by $u_c(k)$. For both SRWNN- and WNN-based controllers, only two signals are considered to be their inputs. These include $r(k)$ and $Y_P(k)$. This makes the controller structure simple and reduces the number of parameters to be adjusted. During the online training, the controller parameters are modified so as to generate a suitable control output that forces the plant to generate the output $((Y_p(k+1))$ close to or equal to that of the reference model output $(Y_D(k+1))$. It is to be noted that the process of identification of the plant's dynamics by SRWNN-based identifier is also in operation along with the control. Hence, the performance of the controller depends upon the ability of the identifier to predict the future values of the plant. Furthermore, the identifier gives the necessary Jacobian value to the controller. The procedure of updating parameters of the controller is very much similar to the method developed for identifiers. For updating the various parameters of SRWNN controller, the following cost function is defined first:

$$E_c(k) = \frac{1}{2}e_c^2(k), \tag{69}$$

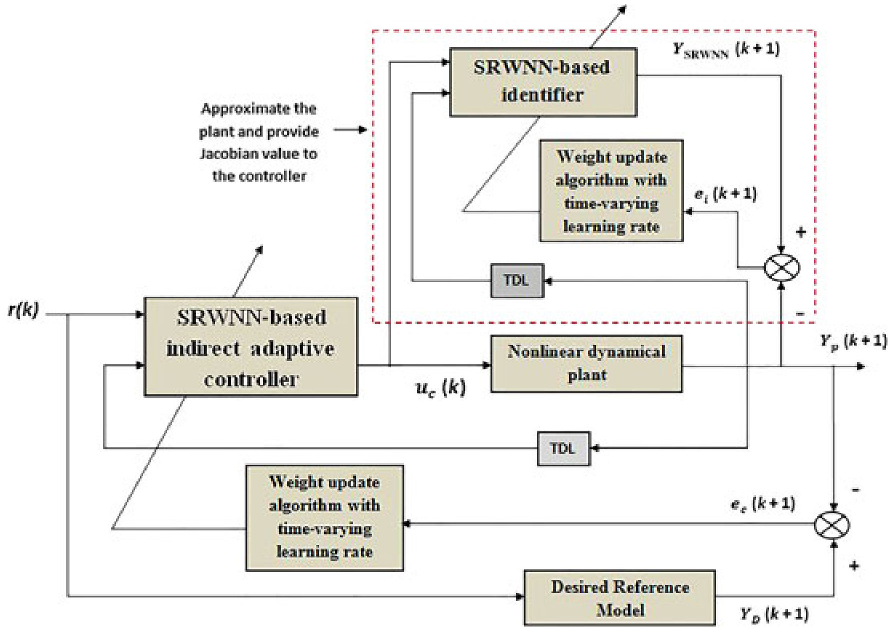
$Y_{SRWNN}(k+1)$
SRWNN-based identifier
Approximate the plant and provide Jacobian value to the controller
Weight update algorithm with time-varying learning rate
$e_i(k+1)$
+
TDL
-
$r(k)$
SRWNN-based indirect adaptive controller
Nonlinear dynamical plant
$Y_p(k+1)$
$u_c(k)$
TDL
-
$e_c(k+1)$
Weight update algorithm with time-varying learning rate
+
Desired Reference Model
$Y_D(k+1)$

FIGURE 11 Predictive control and identification configuration based on self-recurrent wavelet neural network (SRWNN). TDL, time delay line [Colour figure can be viewed at wileyonlinelibrary.com]

where $e_c(k) = Y_D(k) - Y_P(k)$ is the control error and subscript c in $e_c(k)$ and $E_c(k)$ refers to the control. The parameter matrix P_C of SRWNN-based controller includes $\{a_j, w_j^o, t_j, \lambda_j\}$. The update equations of the parameters of SRWNN controller are similar to that of SRWNN identifier. The only additional term that comes in the update equations is the Jacobian term (also known as sensitivity) and is equal to $\frac{\partial Y_P(k)}{\partial u_c(k)}$. For instance,

$$\frac{\partial E_c(k)}{\partial P_C(k)} = \frac{\partial E_c(k)}{\partial Y_{\mathrm{SRWNN}}(k)} \times \frac{\partial Y_{\mathrm{SRWNN}}(k)}{\partial P_C(k)} \tag{70}$$

or

$$\frac{\partial E_c(k)}{\partial P_C(k)} = -e_c(k) \times \frac{\partial Y_P(k)}{\partial u_c(k)} \times \frac{\partial u_c(k)}{\partial P_C(k)}. \tag{71}$$

After sufficient training, we can write

$$\frac{\partial Y_{\mathrm{SRWNN}}(k)}{\partial u_c(k)} \approx \frac{\partial Y_P(k)}{\partial u_c(k)}. \tag{72}$$

Furthermore, the computation of $\frac{\partial u_c(k)}{\partial P_C(k)}$ is same as performed in the case of SRWNN identifier. In addition, the time-varying learning rate, $\eta_c(k)$, is used in the update equations of SRWNN controller's parameters where subscript c in $\eta_c(k)$ refers to the control.

10 | SIMULATION STUDY—SIMULTANEOUS PREDICTIVE CONTROL AND IDENTIFICATION

Two complex practical systems are used to test and compare the performances of SRWNN- and WNN-based control and identification scheme.

10.1 | Example 3: one-link robotic manipulator system

The general mathematical model that represents an n serial links robotic manipulator is given by[43]

$$M(Y_P)\ddot{Y}_p + C(Y_P, \dot{Y}_P)\dot{Y}_P + D\dot{Y}_P + G(Y_P) = \tau. \tag{73}$$

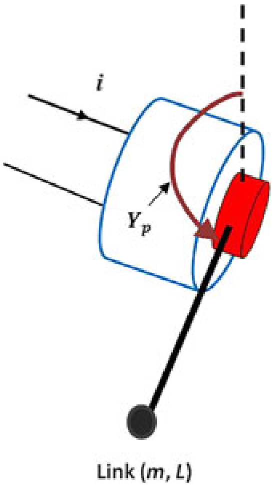

FIGURE 12 One-link robotic manipulator (Example 3) [Colour figure can be viewed at wileyonlinelibrary.com]

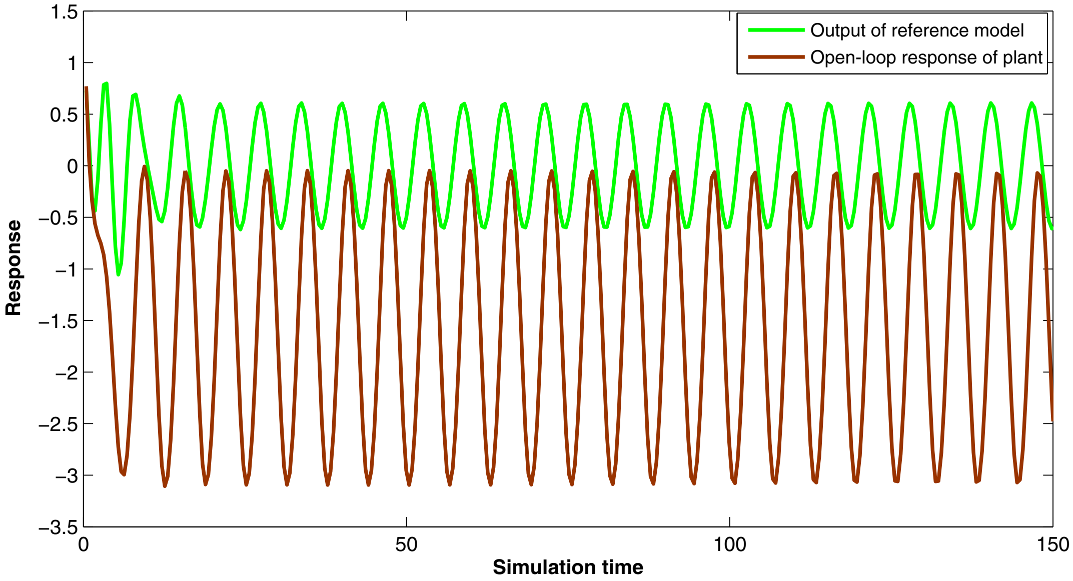

FIGURE 13 Open-loop response of plant (Example 3) [Colour figure can be viewed at wileyonlinelibrary.com]

Here, Y_P, $\dot{Y}_P$, and $\ddot{Y}_P \in R^n$ denotes the joint position, velocity, and accelerations, respectively. Furthermore, $n \times n$ real matrices M, C, and D are symmetric positive definite inertia matrix, centrifugal coriolis matrix, and positive definite diagonal matrix for damping friction coefficients of each joint, respectively. The one-link robotic manipulator is shown in Figure 12.

The real $n \times n$ matrix $G(Y_P)$ represent a gravity term, and $\tau(k) \in R^n$ is an input vector torques applied on the joints. For the case of one-link robotic manipulator, the value of n is 1. The second-order difference equation for the one-link robotic manipulator is given by

$$Y_P(k+1) = f\left[Y_P(k), Y_P(k-1), u_c(k)\right], \tag{74}$$

where the unknown function f is given by

$$f\left[x_1, x_2, x_3\right] = (2-T)x_1 + x_2(T-1) - 9.8T^2\cos(x_2) + T^2x_3. \tag{75}$$

Furthermore, $x_1 = Y_P(k), x_2 = Y_P(k-1)$, and $x_3 = u_c(k)$ and the sampling period is $T = 0.45$ seconds. The externally applied input signal considered in this example is equal to $r(k) = \sin(kT)$. Figure 13 shows the reference model and plant's response when no controller is connected in the loop (that is, open-loop response). It can be seen that plant's output is not following the reference model output. Thus, the control and identification configuration, as shown in Figure 11, is set up.

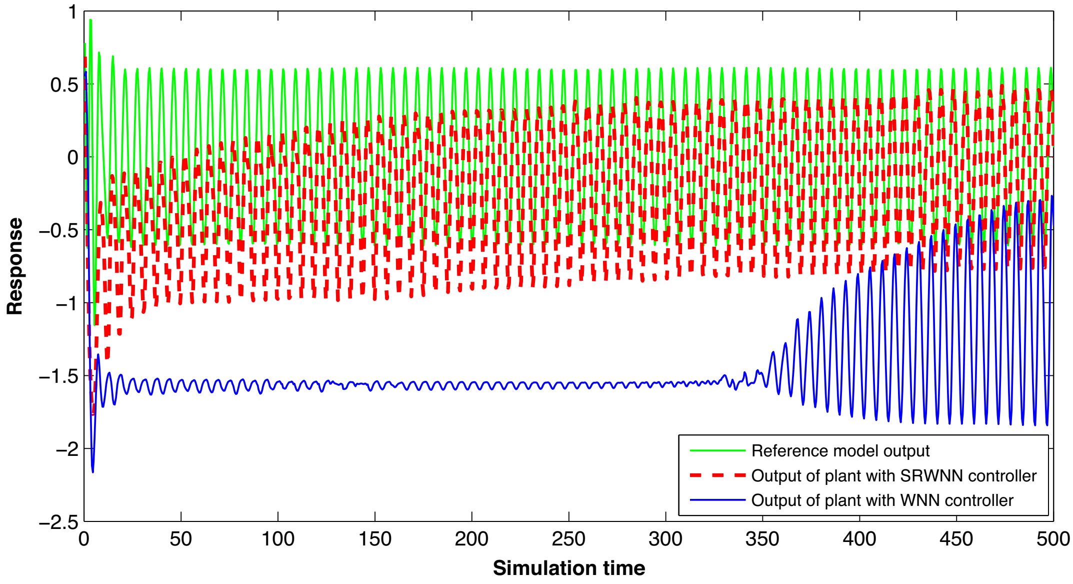

FIGURE 14 Response of the plant under self-recurrent wavelet neural network (SRWNN) and wavelet neural network (WNN) controllers during an initial phase of training (Example 3) [Colour figure can be viewed at wileyonlinelibrary.com]

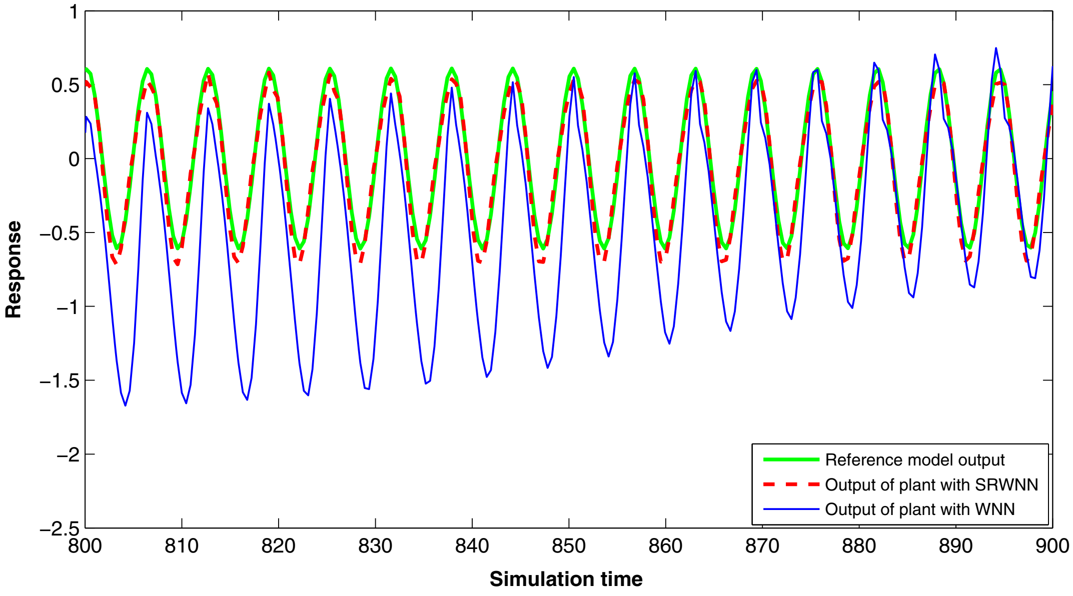

FIGURE 15 Response of the plant under self-recurrent wavelet neural network (SRWNN) and wavelet neural network (WNN) controllers after a certain amount of the training (Example 3) [Colour figure can be viewed at wileyonlinelibrary.com]

The initial values of $\eta_i(k)$ and $\eta_c(k)$ are set the same and is equal to 0.0359. Figure 14 shows the response of the plant under SRWNN- and WNN-based controllers, respectively, during the initial phase of training, and Figure 15 shows the response of the plant as the training progressed. Furthermore, Figure 16 shows the output of the plant when it becomes closer to that of reference model output. It can be easily seen from these Figures that the response of the plant under SRWNN controller action improved faster (by seeing the time axis in these Figures) as compared with response obtained with WNN-based controller. This suggests that the SRWNN-based controller has performed better than the WNN controller. Furthermore, **I**nstantaneous **M**ean **S**quare **E**rror (IMSE) plots of the plant obtained under the SRWNN and WNN control and identification schemes are shown in Figure 17. It can be seen that the IMSE value drops much quickly to zero when SRWNN is used as a controller as compared with WNN.

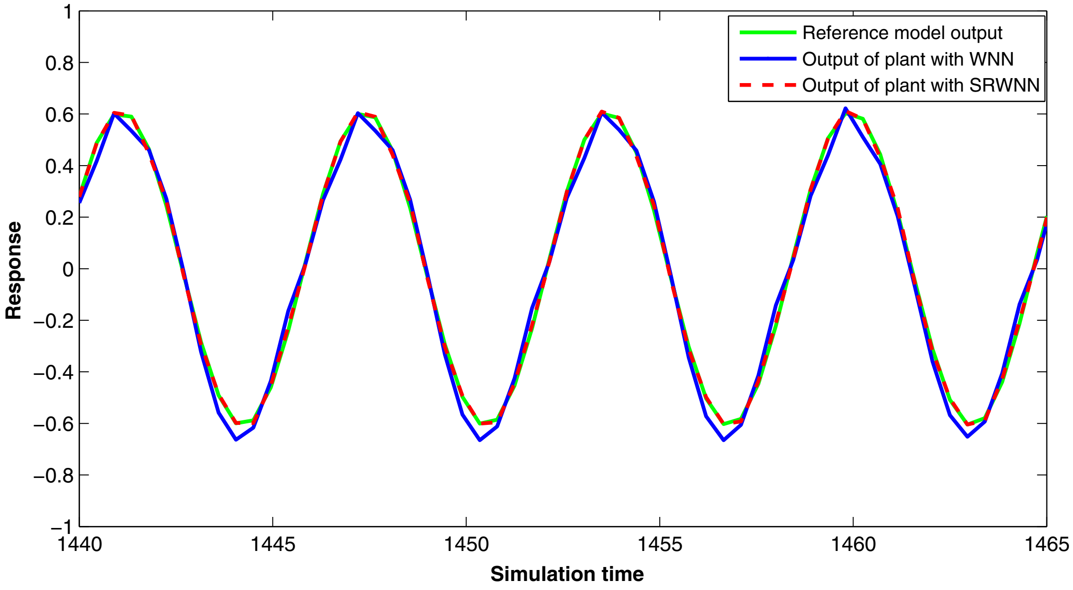

FIGURE 16 Response of the plant under self-recurrent wavelet neural network (SRWNN) and wavelet neural network (WNN) controllers during the final stage of the training (Example 3) [Colour figure can be viewed at wileyonlinelibrary.com]

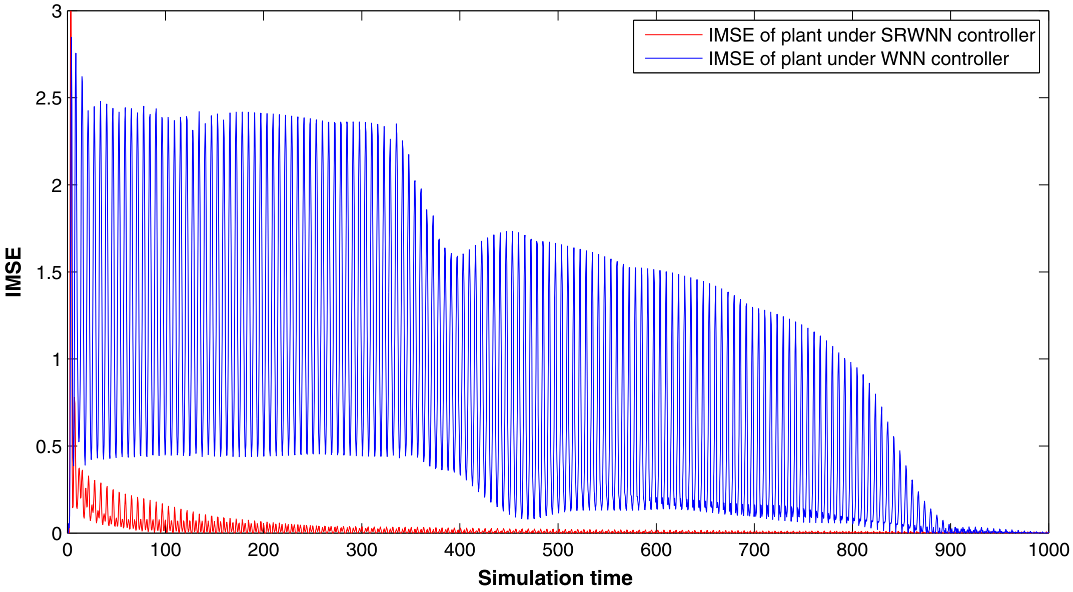

FIGURE 17 Instantaneous mean square error (IMSE) of the plant obtained under self-recurrent wavelet neural network (SRWNN) and wavelet neural network (WNN) controllers action (Example 3) [Colour figure can be viewed at wileyonlinelibrary.com]

Furthermore, Figure 18 and Figure 19 shows the responses of SRWNN- and WNN-based identifiers, respectively, during the initial stage of training. Figure 20 shows the responses of the identifiers as the training progressed, and finally, Figure 21 shows the responses of SRWNN and WNN identifiers when their outputs converged to that of the output of the plant. From the Figures, it can be easily concluded that SRWNN-based identifier has outperformed the WNN identifier as SRWNN identifier took lesser time in approximating the dynamics of the plant. The corresponding IMSE plots obtained during the online training of SRWNN and WNN identifiers are shown in Figure 22. It can be seen from these plots that SRWNN identifier has performed better than that of the WNN identifier. Furthermore, Table 3 shows the performance comparison of SRWNN- and WNN-based identifiers and controllers. It can be easily seen that the minimum value of Average IMSE (AIMSE) is obtained with SRWNN. This shows that SRWNN has performed better than WNN. Figure 23 shows the flowchart that depicts the various steps involved in the training of SRWNN identifier and controller.

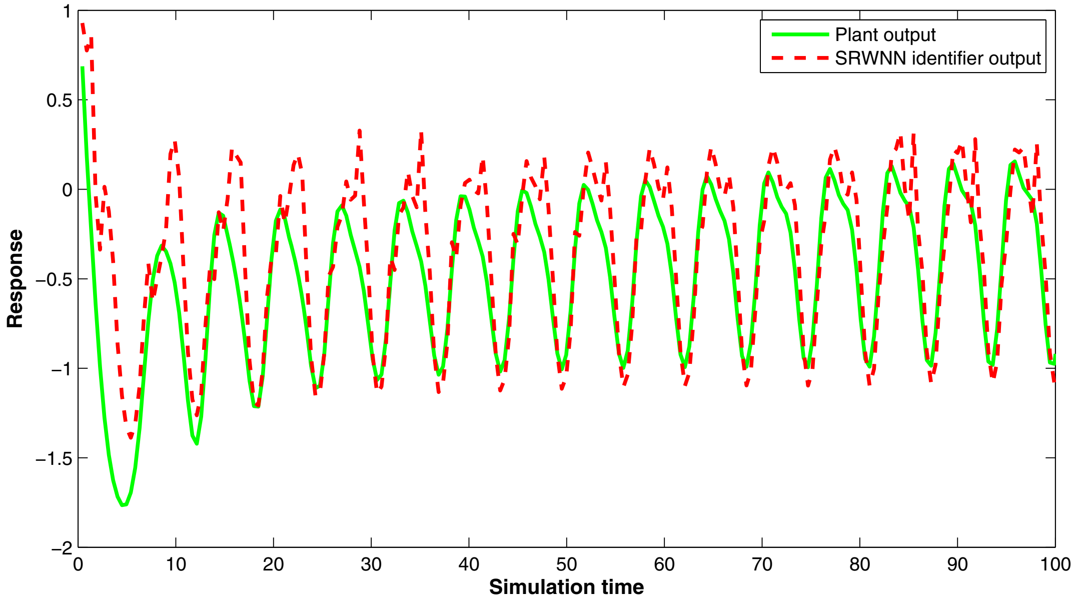

FIGURE 18 Self-recurrent wavelet neural network (SRWNN) identifier response at the initial phase of training (Example 3) [Colour figure can be viewed at wileyonlinelibrary.com]

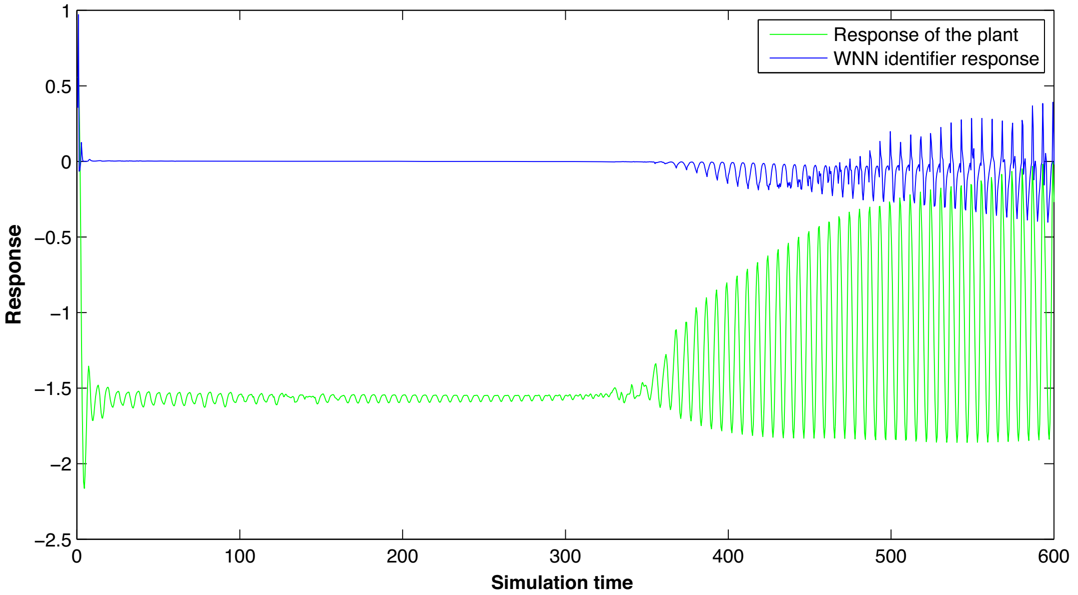

FIGURE 19 Wavelet neural network (WNN) identifier response at the initial phase of training (Example 3) [Colour figure can be viewed at wileyonlinelibrary.com]

10.1.1 | Disturbance rejection test

Now, both SRWNN- and WNN-based controllers will be tested to check whether they will be able to recover from the effects of the disturbance signal or not. For that, a signal of magnitude 2.8 units is added in the output of SRWNN and WNN controllers at 7650th time instant. The disturbance signal causes a rise in the IMSE value. This is shown in Figure 24. It can be easily seen from the IMSE plot that error get reduced quickly to zero under SRWNN action as compared with WNN controller. This shows that SRWNN controller is more suitable than WNN controller in recovering from the effects of the external disturbance signal. Furthermore, the corresponding variation in the output of the plant due to the disturbance signal and the subsequent recovery obtained is shown in Figure 25. It can be seen from the Figure that

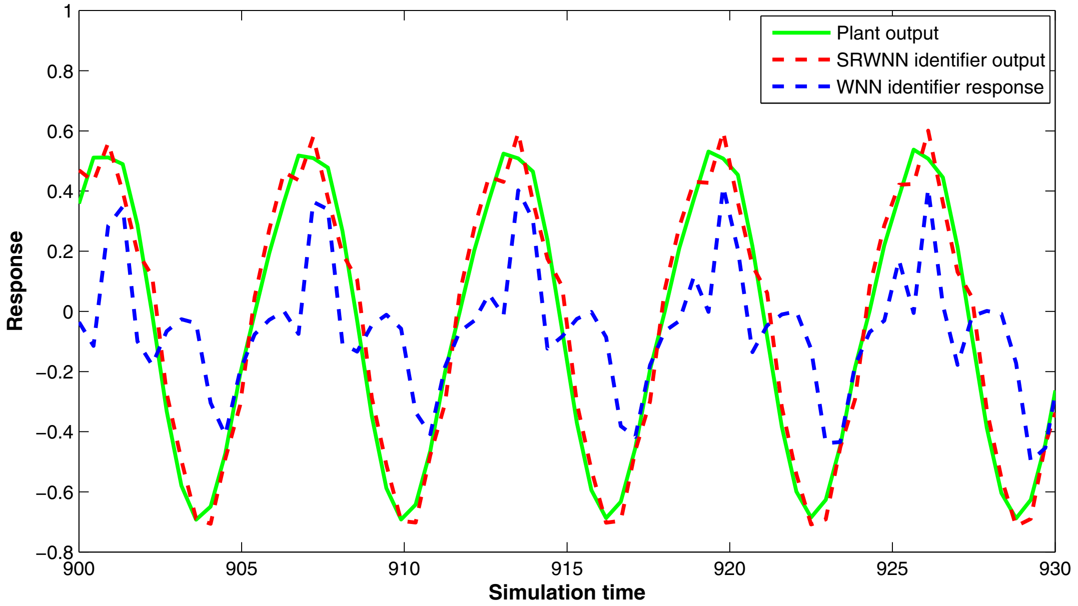

FIGURE 20 Self-recurrent wavelet neural network (SRWNN) and wavelet neural network (WNN) identifiers response as the training progressed (Example 3) [Colour figure can be viewed at wileyonlinelibrary.com]

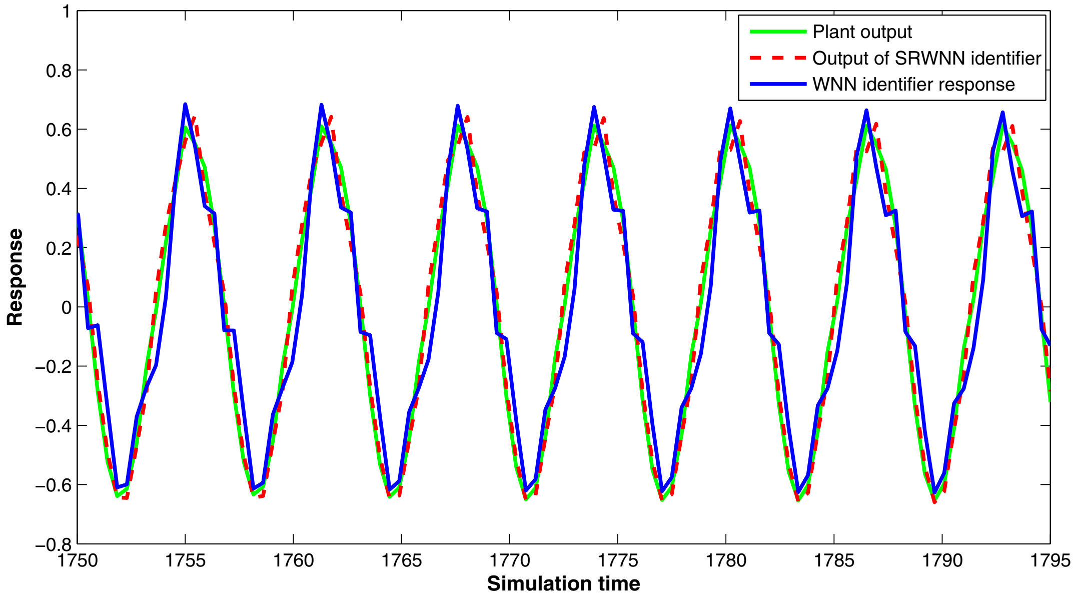

FIGURE 21 Self-recurrent wavelet neural network (SRWNN) and wavelet neural network (WNN) identifiers response when their outputs converged to plant's output (Example 3) [Colour figure can be viewed at wileyonlinelibrary.com]

output of plant recovered much quickly from the impact of disturbance signal under SRWNN controller as compared with WNN-based controller.

10.2 | Example 4: control and identification of inverted pendulum

In this last example, control and identification of an inverted pendulum is performed. Figure 26 depicts the schematic of an inverted pendulum. It consists of a rigid rod that is mounted on a motorized cart. The friction is assumed to be absent; hence, the rod is free to rotate. Furthermore, the motion of the cart is assumed to be one-dimensional along the track. A control force F is required for maintaining the rod in the vertical position; otherwise, it will fall down and will

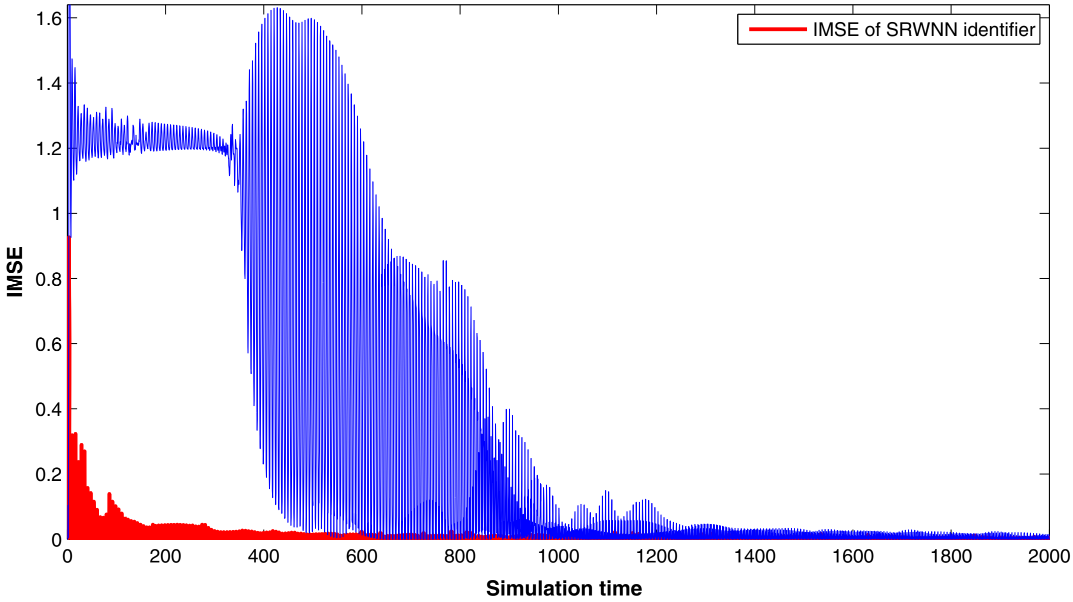

FIGURE 22 Instantaneous mean square error (IMSE) of self-recurrent wavelet neural network (SRWNN) and wavelet neural network (WNN) identifiers during the training (Example 3) [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 3 Performance of the self-recurrent wavelet neural network (SRWNN)– and wavelet neural network (WNN)–based controllers and identifiers (Example 3)

	SRWNN-Based Controller and Identifier	**WNN-Based Controller and Identifier**
Total number of inputs	02 + 02 = 04	02 + 03 = 05
Number of nodes in hidden layers	10 + 10 = 20	10 + 10 = 20
Learning rate	Adaptive	Adaptive
Simulation time	9000	9000
Minimum AIMSE of plant under controller action	0.0036	0.0489
Minimum AIMSE of identifier	0.0017	0.0568

Abbreviation: AIMSE, Average IMSE.

never return back to the vertical position. This suggests that we require some feedback closed-loop control for generating the desired control force. The differential equation that describes the dynamics of an inverted pendulum can be found in the work of Lilly.[44] The corresponding equivalent difference equation that can be obtained from this differential equation with sampling period of $T = 0.01$ is given by

$$Y_P(k+1) = 2Y_P(k) - Y_P(k-1) + T^2\left[\frac{a+bc}{d}\right], \tag{76}$$

where the various terms present in Equation 76 are given by

$$\begin{aligned} a &= 9.81\sin\left(Y_P(k-1)\right) \\ b &= 0.66\cos\left(Y_P(k-1)\right) \\ c &= \left\{-F(k) - 0.25\left[\frac{Y_P(k) - Y_P(k-1)}{T}\right]^2 \times \sin\left(Y_P(k-1)\right)\right\} \\ d &= 0.5\left\{\frac{4}{3} - \frac{1}{3}\cos^2\left(Y_P(k-1)\right)\right\}. \end{aligned} \tag{77}$$

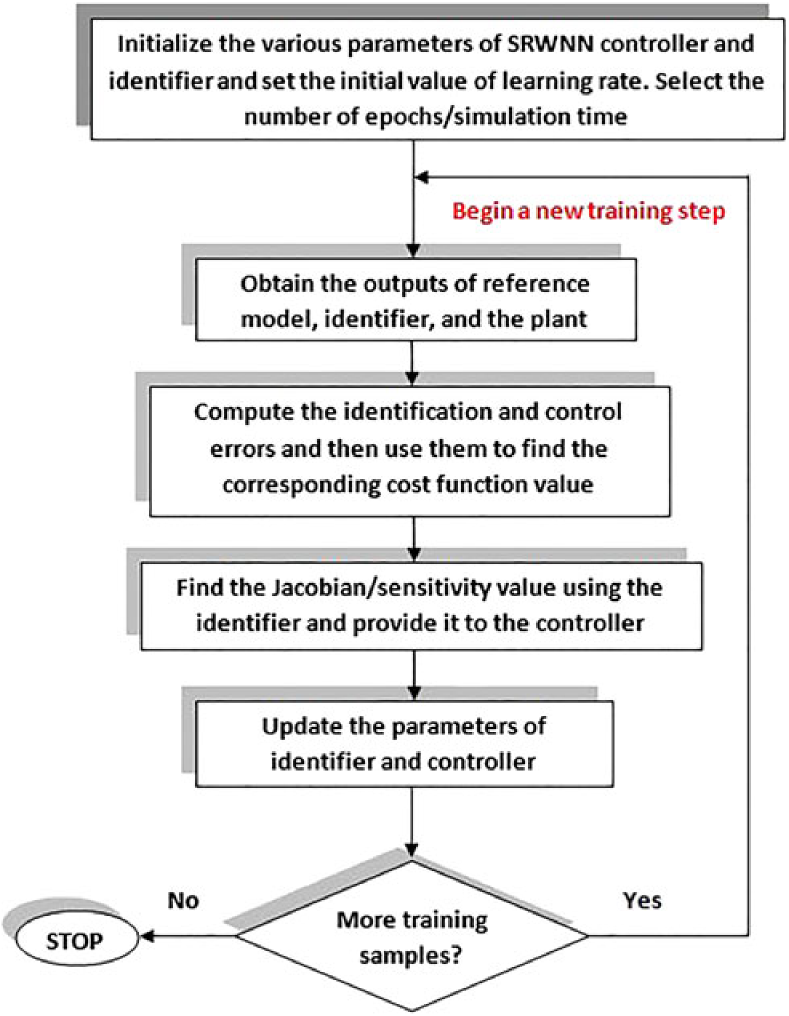
Initialize the various parameters of SRWNN controller and identifier and set the initial value of learning rate. Select the number of epochs/simulation time
Begin a new training step
Obtain the outputs of reference model, identifier, and the plant
Compute the identification and control errors and then use them to find the corresponding cost function value
Find the Jacobian/sensitivity value using the identifier and provide it to the controller
Update the parameters of identifier and controller
More training samples?
No
Yes
STOP

FIGURE 23 Steps involved in the online training of self-recurrent wavelet neural network (SRWNN) identifier and controller [Colour figure can be viewed at wileyonlinelibrary.com]

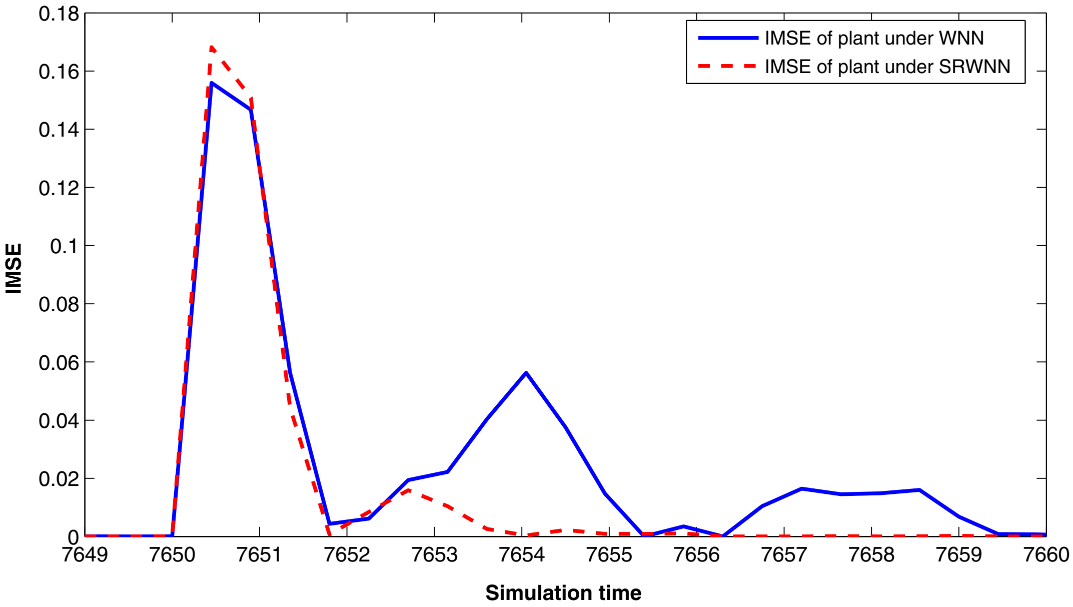

FIGURE 24 Instantaneous mean square error (IMSE) of the plant obtained during the instant of the disturbance signal (Example 3). SRWNN, self-recurrent wavelet neural network; WNN, wavelet neural network [Colour figure can be viewed at wileyonlinelibrary.com]

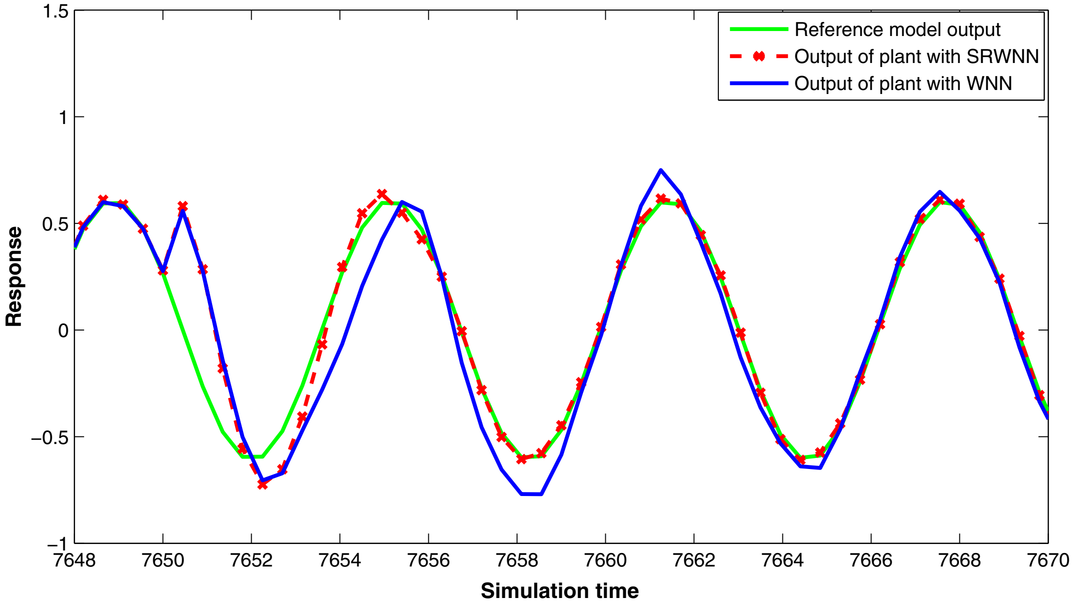

FIGURE 25 Output of the plant obtained during the instant of the disturbance signal (Example 3). SRWNN, self-recurrent wavelet neural network; WNN, wavelet neural network [Colour figure can be viewed at wileyonlinelibrary.com]

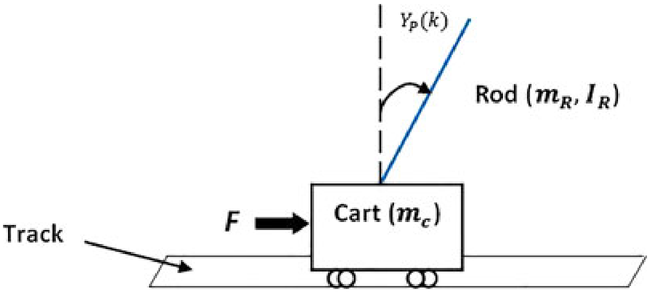

FIGURE 26 Inverted pendulum (Example 4) [Colour figure can be viewed at wileyonlinelibrary.com]

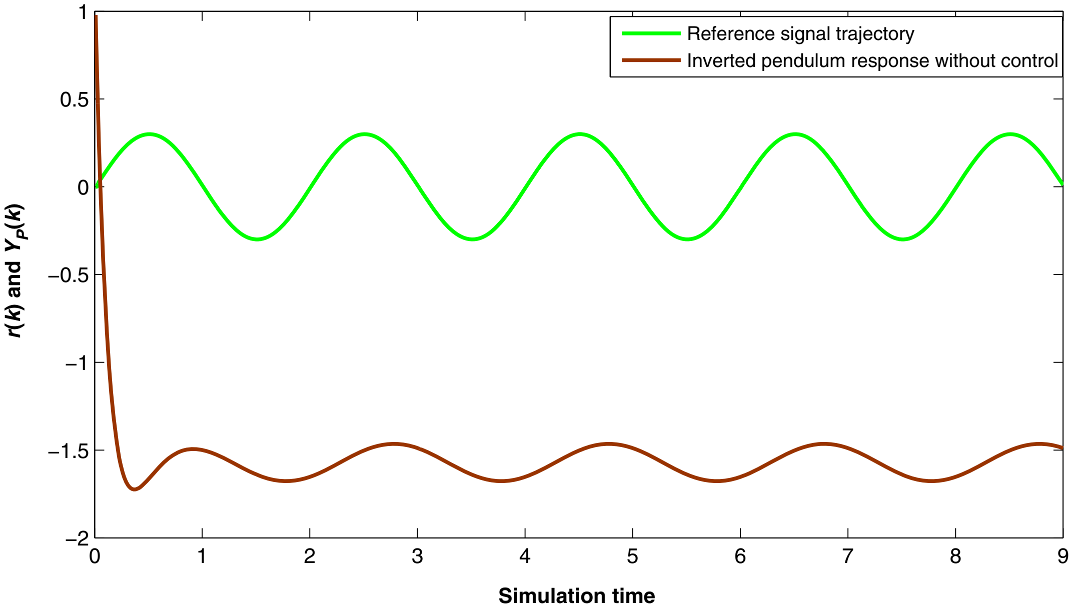

FIGURE 27 Open-loop response of inverted pendulum (Example 4) [Colour figure can be viewed at wileyonlinelibrary.com]

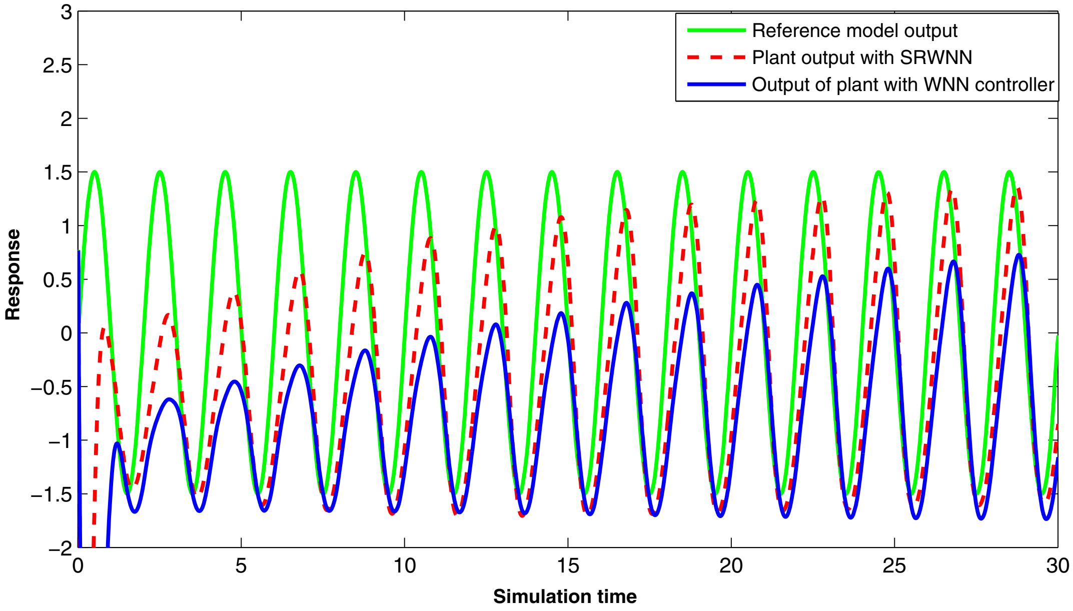

FIGURE 28 Response of the plant obtained during an initial phase of the training of self-recurrent wavelet neural network (SRWNN) controller (Example 4). WNN, wavelet neural network [Colour figure can be viewed at wileyonlinelibrary.com]

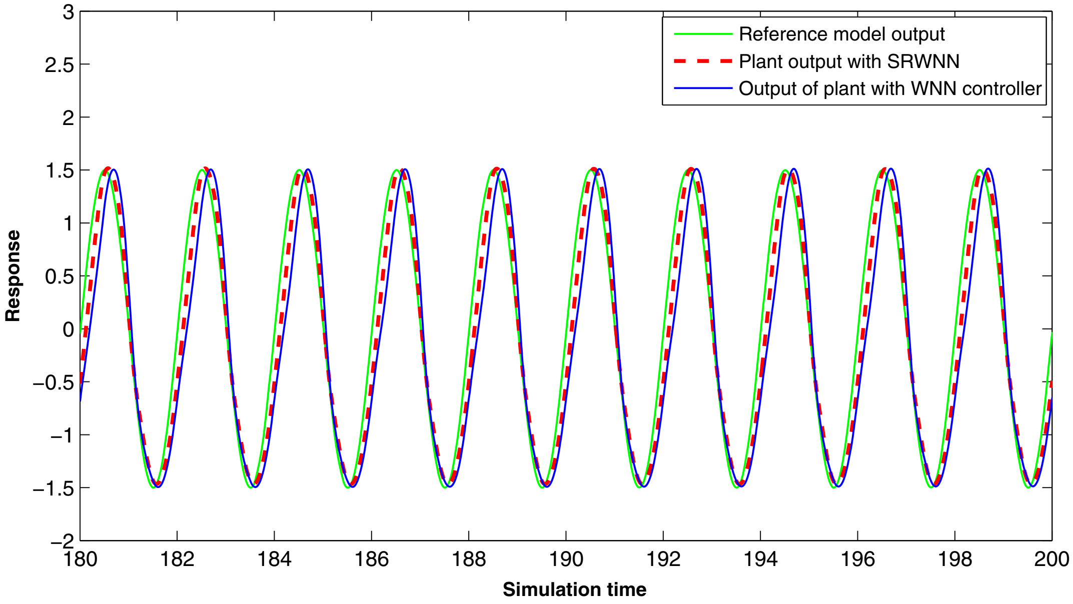

FIGURE 29 Response of the plant after a sufficient training of the self-recurrent wavelet neural network (SRWNN) and wavelet neural network (WNN) controllers (Example 4) [Colour figure can be viewed at wileyonlinelibrary.com]

Furthermore, $Y_P(k)$ is the vertical angle of the rod. In addition, $F(k) = u_c(k)$ is an external control force (control signal) applied to the cart so as to make the rod to follow the desired input trajectory signal, $r(k) = 1.5\sin(\pi kT)$. The response of the pendulum without a control action is shown in Figure 27.

It can be easily seen that output of the pendulum is not following the reference input signal, and hence, control action is required. The initial value of $\eta_i(k)$ and $\eta_c(k)$ is set equal to 0.0037. The responses of the plant during the initial and final stage of the training are recorded and are shown in Figures 28 to 34. It can be seen that the performance obtained with the SRWNN-based controller and identifier is much better as obtained with the WNN-based controller and identifier (in terms of time taken in learning the dynamics of the plant and in forcing the plant's output to follow the desired signal). Furthermore, the details regarding the number of inputs provided to the controller and identifier, simulation time, and minimum AIMSE achieved are shown in Table 4. From the Table, it can be concluded that the SRWNN-based predictive

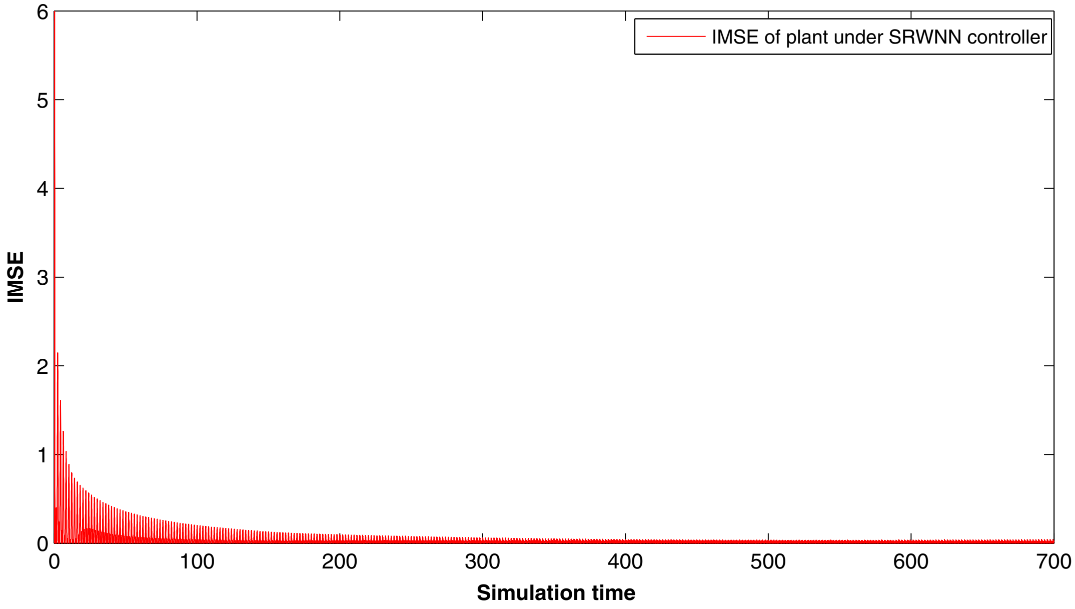

FIGURE 30 Instantaneous mean square error (IMSE) of the plant under self-recurrent wavelet neural network (SRWNN) controller (Example 4) [Colour figure can be viewed at wileyonlinelibrary.com]

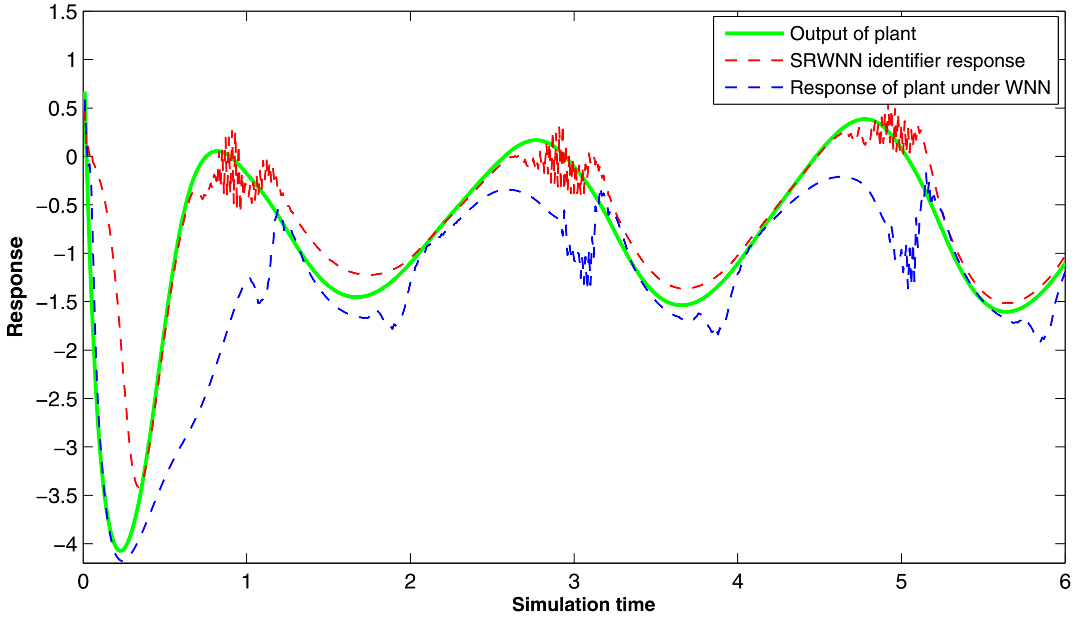

FIGURE 31 Response of the self-recurrent wavelet neural network (SRWNN) identifier during an initial phase of the training (Example 4). WNN, wavelet neural network [Colour figure can be viewed at wileyonlinelibrary.com]

control and identification scheme have performed better than the WNN both in identifying the unknown dynamics of the plant and in forcing the plant to follow the desired trajectory.

10.2.1 | Disturbance rejection test

A signal of magnitude 1 units is added in the output of SRWNN and WNN controllers at 600th time instant. The disturbance signal causes a rise in the IMSE value. This is shown in Figure 35. It can be easily seen from the IMSE plot that error gets reduced quickly to zero with SRWNN controller as compared with the decrease in the error obtained with WNN controller. Furthermore, the corresponding variation in the output of the plant due to the disturbance signal and the subsequent recovery from it is shown in Figure 36. It can be seen from the Figure that output of plant recovered quickly from

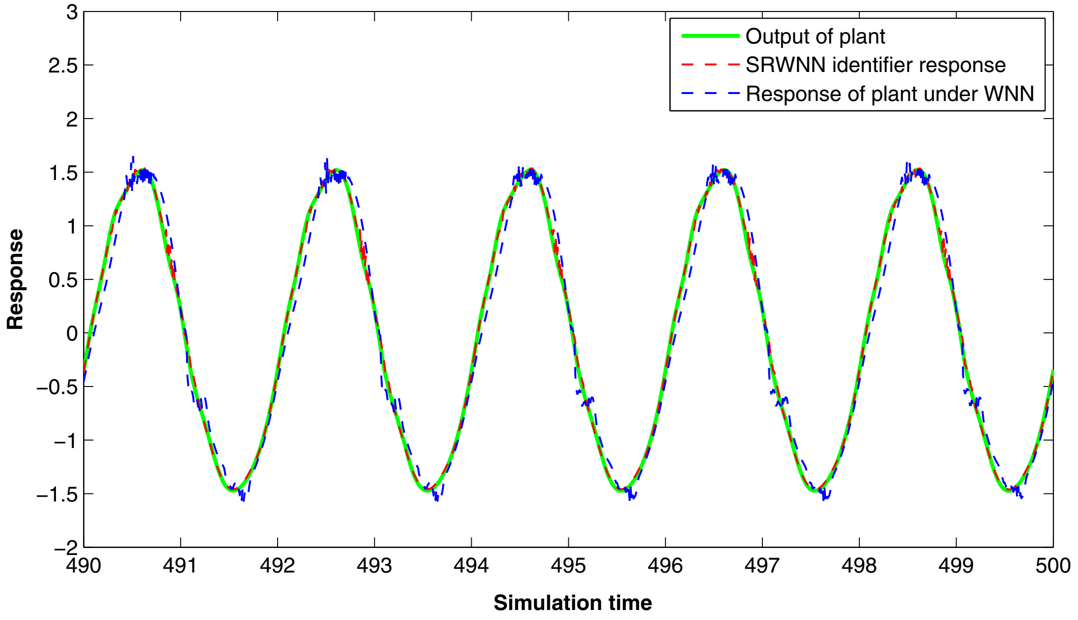

FIGURE 32 Response of the self-recurrent wavelet neural network (SRWNN) identifier after a sufficient amount of the training (Example 4). WNN, wavelet neural network [Colour figure can be viewed at wileyonlinelibrary.com]

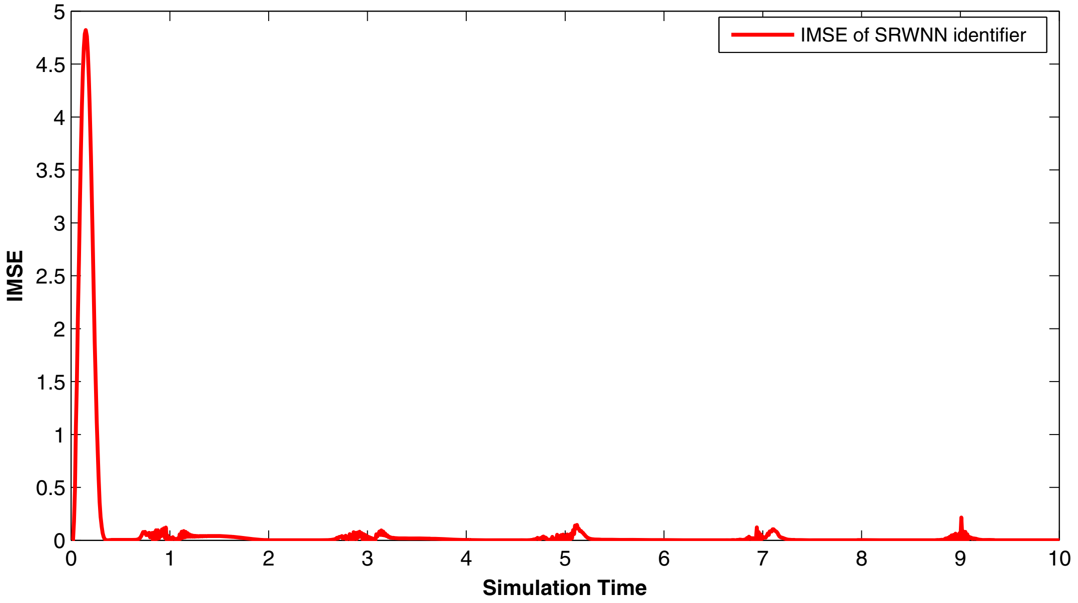

FIGURE 33 Instantaneous mean square error (IMSE) of the self-recurrent wavelet neural network (SRWNN) identifier obtained during the online training (Example 4) [Colour figure can be viewed at wileyonlinelibrary.com]

the impact of disturbance signal under SRWNN controller as compared with the recovery obtained when WNN-based controller is used.

11 | DISCUSSION

From the simulation study, it can be concluded that the proposed SRWNN-based identification and predictive control scheme is able to successfully identify and control the complex nonlinear plants. Furthermore, SRWNN-based scheme has given minimum value of IMSE and AMSE as compared with those obtained with the WNN. The number of parameters required by SRWNN is also lesser in number as compared with WNN. This shows that SRWNN is computationally more

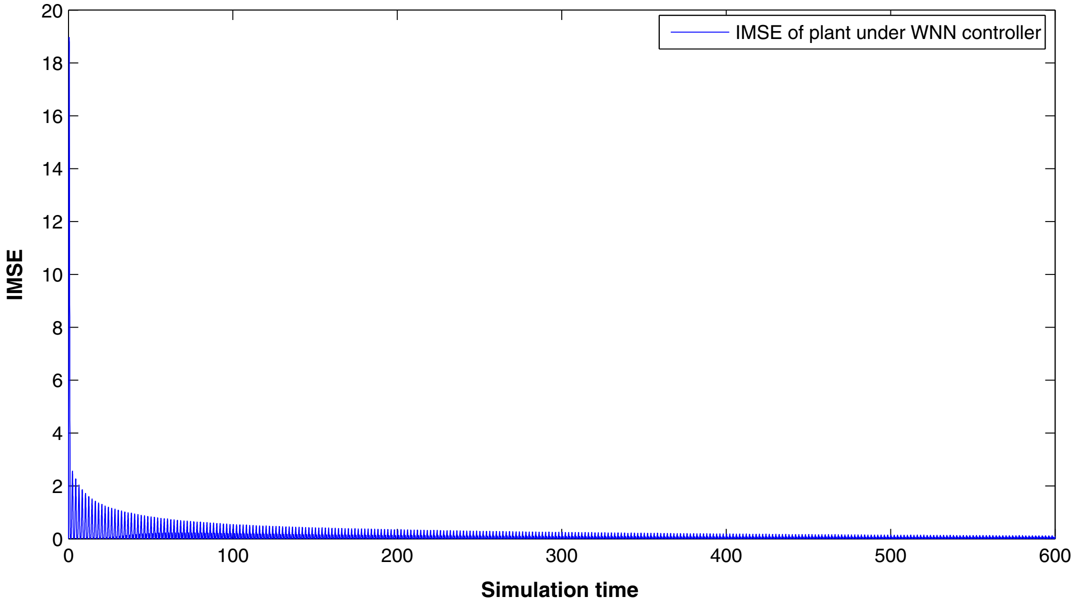

FIGURE 34 Instantaneous mean square error (IMSE) of the plant obtained under wavelet neural network (WNN) controller action (Example 4) [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 4 Performance comparison of self-recurrent wavelet neural network (SRWNN)– and wavelet neural network (WNN)–based controllers and identifiers

	SRWNN-Based Controller and Identifier	WNN-Based Controller and Identifier
Total number of inputs	02 + 02 = 04	02 + 03 = 05
Number of nodes in hidden layers	10 + 10 = 20	10 + 10 = 20
Learning rate	Adaptive	Adaptive
Simulation time	90 000	90 000
AIMSE of controller	0.0011	0.0069
AIMSE of identifier	0.0034	0.0086

Abbreviation: AIMSE, Average IMSE.

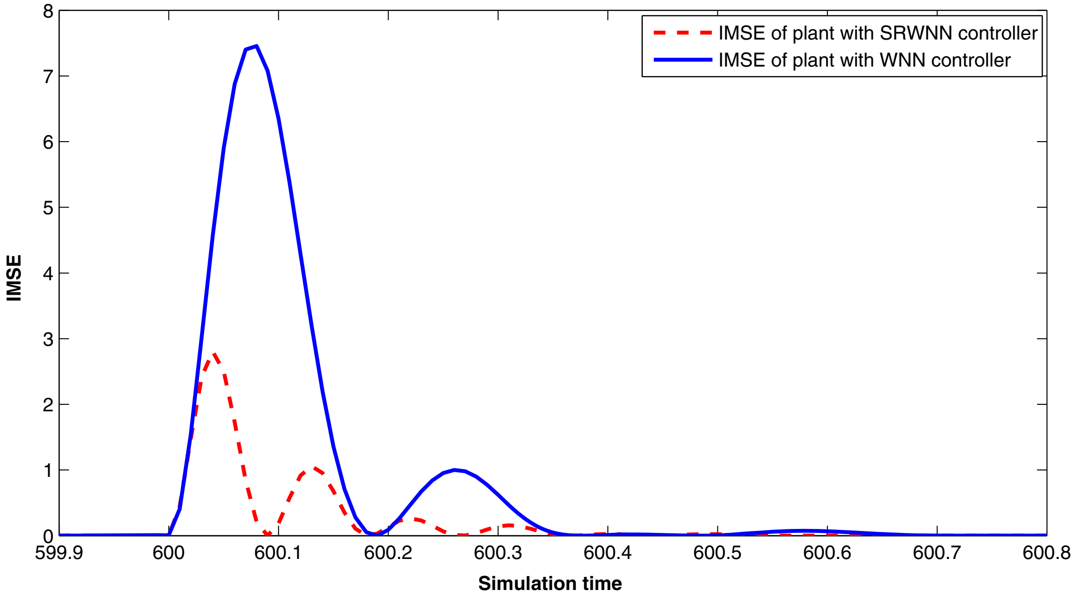

FIGURE 35 Instantaneous mean square error (IMSE) of the plant obtained during the instant of the disturbance signal (Example 4). SRWNN, self-recurrent wavelet neural network; WNN, wavelet neural network [Colour figure can be viewed at wileyonlinelibrary.com]

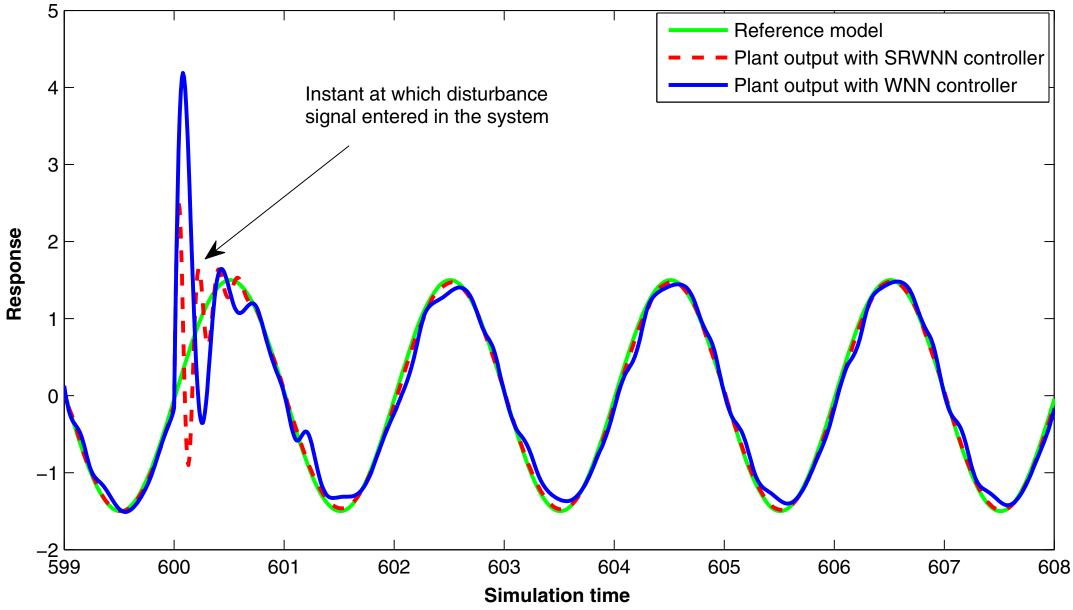

FIGURE 36 Response of the plant obtained during the instant of the disturbance signal (Example 4). SRWNN, self-recurrent wavelet neural network; WNN, wavelet neural network [Colour figure can be viewed at wileyonlinelibrary.com]

efficient than the WNN model. The disturbance rejection tests also show that the performance of SRWNN is found to be better than WNN in recovering from the impact of the disturbance signal. Another thing that can be concluded from the simulation results is the effectiveness of the time-varying learning rate. Not only it maintained the system stability but also it has provided the faster convergence of the parameters.

12 | CONCLUSION

In this paper, SRWNN-based identification and predictive control scheme is proposed. The SRWNN is a modification of WNN structure as the former includes self-feedback weights and the direct feed through of inputs to the output layer. These factors provide the SRWNN with a memory and allow it to better approximate and control the nonlinear systems. The dynamic BP method is used to obtain the parameter update equations. To improve its convergence speed and to ensure the stability of the system, we have used a time-varying learning rate. The proposed scheme is applied on four nonlinear plants that include robotic manipulator and inverted pendulum systems. The simulation results reveals that the proposed scheme based on SRWNN has given better response when disturbance signal entered in the system as compared to the response obtained with the WNN, it also offered lesser number of parameters to be tuned and provided better accuracy than WNN in approximating and controlling the given nonlinear systems.

ORCID

Rajesh Kumar http://orcid.org/0000-0001-7172-1081

REFERENCES

1. Åström KJ, Wittenmark B. *Adaptive Control*. Boston, MA: Addison-Wesley; 1989.
2. Banakar A, Azeem MF. Local recurrent sigmoidal–wavelet neurons in feed-forward neural network for forecasting of dynamic systems: theory. *Appl Soft Comput*. 2012;12(3):1187-1200.
3. Johnson JD. Neural networks for control. *Neurocomputing*. 1997;3(14):301-302.
4. Srivastava S, Singh M, Hanmandlu M, Jha AN. Modeling of non-linear systems by FWNNs and their intelligent control. *Int J Adapt Control Signal Process*. 2005;19(7):505-530.
5. Narendra KS, Parthasarathy K. Identification and control of dynamical systems using neural networks. *IEEE Trans Neural Netw*. 1990;1(1):4-27.

6. Kumar R, Srivastava S, Gupta JRP. Modeling and adaptive control of nonlinear dynamical systems using radial basis function network. *Soft Comput*. 2017;21(15):4447-4463.
7. Kumar R, Srivastava S, Gupta JRP. Online modeling and adaptive control of robotic manipulators using Gaussian radial basis function networks. *Neural Comput Applic*. 2016;30(1):223-239.
8. Ganjefar S, Tofighi M. Single-hidden-layer fuzzy recurrent wavelet neural network: applications to function approximation and system identification. *Inform Sci*. 2015;294:269-285.
9. Cordova JJ, Yu W. Recurrent wavelets neural networks learning via dead zone Kalman filter. Paper presented at: The 2010 International Joint Conference on Neural Networks (IJCNN); 2010; Barcelona, Spain.
10. Szu H, Telfer B, Garcia J. Wavelet transforms and neural networks for compression and recognition. *Neural Netw*. 1996;9(4):695-708.
11. Zhang J, Walter GG, Miao Y, Lee W. Wavelet neural networks for function learning. *IEEE Trans Signal Process*. 1995;43(6):1485-1497.
12. Khoa TQ, Phuong LM, Binh PT, Lien NT. Application of wavelet and neural network to long-term load forecasting. Paper presented at: 2004 International Conference on Power System Technology, 2004. (PowerCon); 2004; Singapore, Singapore.
13. Wai RJ, Duan RY, Lee JD, Chang HH. Wavelet neural network control for induction motor drive using sliding-mode design technique. *IEEE Trans Ind Electron*. 2003;50(4):733-748.
14. Rying EA, Bilbro GL, Lu J-C. Focused local learning with wavelet neural networks. *IEEE Trans Neural Netw*. 2002;13(2):304-319.
15. Cao L, Hong Y, Fang H, He G. Predicting chaotic time series with wavelet networks. *Phys D Nonlinear Phenom*. 1995;85(1-2):225-238.
16. Allingham D, West M, Mees AI. Wavelet reconstruction of nonlinear dynamics. *Int J Bifurcation Chaos*. 1998;8(11):2191-2201.
17. Sureshbabu N, Farrell JA. Wavelet-based system identification for nonlinear control. *IEEE Trans Autom Control*. 1999;44(2):412-417.
18. Lu C-H. Design and application of stable predictive controller using recurrent wavelet neural networks. *IEEE Trans Ind Electron*. 2009;56(9):3733-3742.
19. Zhang Q, Benveniste A. Wavelet networks. *IEEE Trans Neural Netw*. 1992;3(6):889-898.
20. Chen B-S, Cheng Y-M. Adaptive wavelet network control design for nonlinear systems. In: Proceedings of the 35th Conference on Decision and Control; 1996; Kobe, Japan.
21. Lekutai G. Self-tuning control of nonlinear systems using neural network adaptive frame wavelets. Paper presented at: 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation; 1997; Orlando, FL.
22. Xu J, Ho DWC. Adaptive wavelet networks for nonlinear system identification. In: Proceedings of the 1999 American Control Conference (99ACC); 1999; San Diego, CA.
23. Tan Y, Dang X, Liang F, Su C-Y. Dynamic wavelet neural network for nonlinear dynamic system identification. In: Proceedings of the 2000 IEEE International Conference on Control Applications; 2000; Anchorage, AK.
24. Ho DWC, Zhang P-A, Xu J. Fuzzy wavelet networks for function learning. *IEEE Trans Fuzzy Syst*. 2001;9(1):200-211.
25. de Sousa CR, Hemerly EM, Galvão RKH. Adaptive control for mobile robot using wavelet networks. *IEEE Trans Syst Man Cybern B Cybern*. 2002;32(4):493-504.
26. Lin FJ, Wai RJ, Chen MP. Wavelet neural network control for linear ultrasonic motor drive via adaptive sliding-mode technique. *IEEE Trans Ultrason Ferroelectr Freq Control*. 2003;50(6):686-698.
27. Srivastava S, Singh M, Hanmandlu M, Jha AN. New fuzzy wavelet neural networks for system identification and control. *Appl Soft Comput*. 2005;6(1):1-17.
28. Adeli H, Jiang X. Dynamic fuzzy wavelet neural network model for structural system identification. *J Struct Eng*. 2006;132(1):102-111.
29. Abiyev RH, Kaynak O. Fuzzy wavelet neural networks for identification and control of dynamic plants–a novel structure and a comparative study. *IEEE Trans Ind Electron*. 2008;55(8):3133-3140.
30. Chen Y, Yang B, Dong J. Time-series prediction using a local linear wavelet neural network. *Neurocomputing*. 2006;69(4-6):449-465.
31. Aadaleesan P, Miglan N, Sharma R, Saha P. Nonlinear system identification using Wiener type Laguerre-wavelet network model. *Chem Eng Sci*. 2008;63(15):3932-3941.
32. Tzeng S-T. Design of fuzzy wavelet neural networks using the GA approach for function approximation and system identification. *Fuzzy Set Syst*. 2010;161(19):2585-2596.
33. Yousef HA, Elkhatib ME, Sebakhy OA. Wavelet network-based motion control of DC motors. *Expert Syst Appl*. 2010;37(2):1522-1527.
34. Jahangiri F, Doustmohammadi A, Menhaj MB. An adaptive wavelet differential neural networks based identifier and its stability analysis. *Neurocomputing*. 2012;77(1):12-19.
35. Ko C-N. Identification of nonlinear systems with outliers using wavelet neural networks based on annealing dynamical learning algorithm. *Eng Appl Artif Intel*. 2012;25(3):533-543.
36. Hsu C-F. A self-evolving functional-linked wavelet neural network for control applications. *Appl Soft Comput*. 2013;13(11):4392-4402.
37. Okkan U. Wavelet neural network model for reservoir inflow prediction. *Sci Iran*. 2012;19(6):1445-1455.
38. Xu L, Liu S. Study of short-term water quality prediction model based on wavelet neural network. *Math Comput Model*. 2013;58(3-4):807-813.
39. Zhang W, Liu Z, Zhu J, Xu X. Identification and control of time-delay system with recurrent wavelet neural networks. Paper presented at: 2012 Third International Conference on Intelligent Control and Information Processing; 2012; Dalian, China.
40. Lin C-M, Hsueh C-S, Chen C-H. Robust adaptive backstepping control for a class of nonlinear systems using recurrent wavelet neural network. *Neurocomputing*. 2014;142:372-382.
41. Haesloop D, Holt BR. A neural network structure for system identification. In: Proceedings of American Control Conference; 1990; San Diego, CA.

42. Han H-G, Lin Z-L, Qiao J-F. Modeling of nonlinear systems using the self-organizing fuzzy neural network with adaptive gradient algorithm. *Neurocomputing*. 2017;266:566-578.

43. Fu LC, Liao TL. Globally stable robust tracking of nonlinear systems using variable structure control and with an application to a robotic manipulator. *IEEE Trans Autom Control*. 1990;35(12):1345-1350.

44. Lilly JH. *Fuzzy Control and Identification*. Hoboken, NJ: John Wiley & Sons; 2011.

How to cite this article: Kumar R, Srivastava S, Gupta JRP, Mohindru A. Self-recurrent wavelet neural network–based identification and adaptive predictive control of nonlinear dynamical systems. *Int J Adapt Control Signal Process*. 2018;1–33. https://doi.org/10.1002/acs.2916