Research Paper
Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. *, No. *, *, pp. 1-13
DOI:*

Flexible fractional wavelet neural network for non-linear system identification

Hasan Dadashzadeh[1], **Ghasem Ahmadi**[1,*], **Yousef Edrisi Tabriz**[1], **and Behrouz Rezaei**[2]
[1]Department of Mathematics, Payame Noor University, Tehran, Iran.
[2]Faculty of Electrical and Computer Engineering, Babol Noshirvani University of Technology, Babol, Iran.

Abstract

This paper presents a novel neural network architecture called the flexible fractional wavelet neural network (FFrWNN), which enhances traditional wavelet networks by introducing two additional fractional wavelet parameters. These fractional parameters, along with the translation and scale parameters, are dynamically adjusted during the learning process, offering greater flexibility and improved approximation power. The network is trained using a stochastic gradient descent algorithm, and iterative online training formulas are developed for optimizing both the wavelet parameters and network weights. The stability of the network is proven through the Lyapunov stability approach, ensuring reliable convergence. The proposed FFrWNN is evaluated in the context of both one-dimensional and multi-dimensional dynamic system identification. Results demonstrate that the fractional wavelet parameters significantly improve the network's accuracy and efficiency. Compared to conventional neural networks, the FFrWNN shows superior performance in terms of precision and learning capability, making it a powerful tool for complex system modeling and signal processing applications.

Keywords. Fractional wavelet neural network, Stochastic gradient descent, Fractional order wavelets theory, System identification.
2010 Mathematics Subject Classification. 65T60, 92B20, 93B30.

1. Introduction

1.1. **Wavelet Theory.** Wavelet theory emerged in the late 20th century as a tool for signal processing, with its roots in Fourier analysis and multiresolution analysis. In 1989, Yves Meyer and Stphane Mallat developed key concepts that led to the mathematical formalization of wavelets. The introduction of discrete wavelet transforms (DWT) further expanded their applications in image compression, denoising, and data analysis. Wavelet transforms provide a powerful method for representing functions, particularly when dealing with non-stationary signals or data that exhibit localized changes in time or space. Unlike traditional Fourier transforms, which represent functions as a sum of sinusoidal components, wavelet transforms use a set of basis functions known as "wavelets", which are localized in both time (or space) and frequency[30].

1.2. **Neural networks and wavelet networks.** System identification is the science and art of involving the development of mathematical models for dynamic systems using input-output data, along with prior knowledge of the system. Neural networks (NNs) have emerged as pivotal tools for function approximation and black-box system identification. Their versatility spans numerous fields including engineering [31], medicine [9], finance, commerce [21], and security [27]. Despite being inspired by the human brain, NNs are relatively simple, yet their capacity for learning and adaptability has made them indispensable in artificial intelligence. Neural networks learn through a process called training, where they adjust their internal parameters (weights) to minimize the error between predicted and actual outputs. The most common method is supervised learning, where labeled data is used to guide the model. Unsupervised learning involves finding patterns in unlabeled data, while reinforcement learning is based on feedback from actions and rewards. Training often uses gradient descent, an optimization technique that updates weights by

Received: *; Accepted:*.
* Corresponding author. Email: g.ahmadi@pnu.ac.ir.

calculating the gradient of the loss function. Backpropagation is used to efficiently compute these gradients, starting from the output layer and working backward through the network. Learning can also involve techniques like dropout to prevent over fitting, and regularization to reduce model complexity. Hyperparameters like learning rate, batch size, and architecture design are fine-tuned for better performance[13]. The inception of neural network theory dates back to the 1940s and 1950s, with foundational contributions from McCulloch and Pitts, Donald Hebb, and Frank Rosenblatt. Since then, NNs have undergone numerous innovations and refinements, leading to specialized architectures such as multilayer perceptrons, radial basis function networks, self-organizing maps, and recurrent NNs. These variants have demonstrated wide applicability in various domains, from predictive modeling to decision-making tasks.

However, traditional feedforward neural networks exhibit several limitations, particularly in system identification applications, as noted by Alexandridis et al. [5]. These challenges include:

- Inefficient use of local data structure information to update network weights.
- Time-consuming training processes that are prone to convergence at local extrema.
- Difficulty in determining the optimal number of neurons and defining network topology.

In light of these limitations, researchers have sought to combine NNs with other artificial intelligence techniques (e.g., genetic algorithms, particle swarm optimization, ant colony optimization) as well as with advanced mathematical tools such as wavelet analysis. One such hybrid approach, wavelet neural networks (WNN), integrates the powerful function approximation properties of wavelets with the learning capabilities of NNs.

The pioneering work of Zhang and Benveniste [34] introduced the concept of combining wavelet theory with neural networks by utilizing wavelet functions as activation functions in feedforward networks. This novel approach introduced two new parameters, scale and translation, based on the wavelet transform, expanding the flexibility and effectiveness of the neural network model.

There are two primary strategies for constructing wavelet networks:

- Non-orthogonal wavelets: These utilize wavelet frame theory, with continuous variation of wavelet parameters (scale and translation) over the real line.
- Orthogonal wavelets: In these networks, discrete wavelets are used as activation functions, with scale and translation parameters fixed to predefined values throughout the training process.

In networks utilizing non-orthogonal wavelets, wavelet parameters are continuously varied, allowing the use of continuous wavelet transformations and wavelet frame theory. These networks are characterized by their redundancy and the non-uniqueness of function expansions [5]. On the other hand, networks employing orthogonal wavelets discretize the internal wavelet parameters, maintaining them fixed during training. This discretization, though more rigid, often leads to more stable and computationally efficient models.

Recent advancements have positioned WNNs as promising alternatives to conventional NNs, particularly for complex nonlinear process modeling. Researches have shown that WNNs excel in both optimizing network structure and accelerating convergence, owing to the strong theoretical foundation of wavelet transforms. Their ability to intelligently initialize internal parameters and ensure rapid convergence has made WNNs attractive for practical applications, surpassing traditional NNs in terms of training efficiency and accuracy [15].

Given the sensitivity of wavelet networks to their internal structure and parameters, significant research has focused on optimizing these elements. By enhancing accuracy and reducing training time, the refinement of WNNs has made them highly effective for a wide range of system identification tasks. Continued exploration into the optimal design of WNN architectures promises to further expand their application in various fields, ensuring faster and more accurate models for increasingly complex systems.

1.3. **Fractional wavelets.** In recent years, the fractional Fourier transform (FrFT) and its extension, the fractional wavelet transform (FrWT), have gained prominence in various theoretical and applied fields such as physics, applied mathematics, and engineering. These transforms have proven particularly valuable in addressing challenges related to fractional derivatives, image fusion, denoising, and linear frequency modulation (LFM) signal separation.

The fractional wavelet, like many modern mathematical concepts, do not have a universally accepted definition. The concept of the fractional wavelet transform was first introduced by Mendlovic et al. in 1997 [24], where they demonstrated that the FrWT could improve the reconstruction efficiency of the conventional wavelet transform, offering potential applications in image compression. Their method involved applying the fractional Fourier transform to the entire signal, followed by the wavelet transform on each fractional spectrum. However, this approach had notable limitations, particularly its inability to capture the local characteristics of the input signal, thus restricting its effectiveness in certain practical applications.

Subsequent efforts to refine the fractional wavelet transform led to a series of new definitions proposed by various researchers, including Shi et al. [28], Yang et al. [33], Anoh et al. [6], Dai et al. [10], and Srivastava et al.[29].

Srivastava et al.'s work on fractional wavelet transformations (FWT) has provided a valuable extension to traditional wavelet analysis. The authors have provided a rigorous mathematical foundation for FWT, that it is well-defined and its properties are thoroughly explored. Although their FWT has solid theoretical foundations, further research is needed to explore practical implementations, comparative advantages, and extensions to higher-dimensional data. This work lays the groundwork for future studies aiming to apply fractional wavelet transforms in various scientific and engineering domains. In the last five years studies have focused on various aspects of fractional wavelet theory, including its mathematical formulation, applications, and computational techniques. For example, Wang and Xu [32] survey various fractional wavelet transforms, discuss their theoretical foundation, and provide new methods for signal processing. Bazhlekov and Grozdev [8] introduce the application of fractional wavelet transforms for multi-scale image analysis and compression, highlighting their advantage in preserving image features over traditional methods. Agarwal and Gupta [2] present fractional wavelet transforms for the analysis of non-stationary signals, particularly in engineering applications like communications and bio-signals. Gonzalez and Villarreal [12] explore the use of fractional wavelet transforms in the analysis of chaotic systems and provide examples in physics and engineering where traditional methods fail. Gao and Li [11] discuss the application of fractional wavelet transforms for time-frequency analysis of non-stationary signals, such as seismic data and bio-signal analysis.

A significant advancement came in 2021 when Guo et al.[14] introduced a new definition for the fractional wavelet transform (FRWT) with an additional degree of freedom and fractional multiresolution analysis (FrMRA). This new formulation is widely regarded as the most complete and versatile to date, addressing many of the limitations present in earlier models. FRWT marked a noteworthy contribution to the field of signal processing.

In Guo's definition, two fractional parameters, denoted as α and β, are introduced, allowing the fractional wavelet transform to encompass a range of previous definitions as special cases. By selecting specific values for α and β, the classical wavelet transform, as well as the formulations by Dai et al. and Shi et al., can be derived. This generalization makes Guo's definition highly flexible, adaptable to various signal processing needs. The continuous fractional wavelet transform (FrWT), as defined by Guo, shares the same computational complexity as the continuous wavelet transform (CWT), with an order of $O(N)$. It is characterized by the presence of chirp signals, offering a more detailed analysis of the signal's time-frequency representation.

Guo et al.'s definitions of the fractional wavelet and fractional wavelet transform are as follows [14]:

Definition 1.1. A fractional wavelet with two fractional parameters $\alpha, \beta \neq k\Pi$ $(k \in \mathbb{Z})$ is defined as:

$$\psi_{a,b}^{\alpha,\beta}(t) = \psi_{a,b}(t)\ e^{-i\mathbf{A}_{a,b}^{\alpha,\beta}(t)}, \tag{1.1}$$

where

$$\mathrm{A}_{a,b}^{\alpha,\beta}(t) = \frac{1}{2}\left[(t^2 - b^2)\cot\alpha - (\frac{t-b}{a})^2\cot\beta\right], \tag{1.2}$$

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right), \tag{1.3}$$

and $\psi(\cdot)$ is the continuous wavelet function, with $a \in \mathbb{R}^+$ as the scale (dilation) parameter and $b \in \mathbb{R}$ as the translation (shift) parameter.

Definition 1.2. The classic wavelet transform and Guo et al. fractional wavelet transform of a function $f(t)$ are defined as:

$$W_f(a,b) = \langle f, \psi_{a,b}\rangle = \int_{\mathbb{R}} f(t)\overline{\psi_{a,b}(t)}\ \mathrm{d}t, \tag{1.4}$$

$$W_f^{\alpha,\beta}(a,b) = \langle f, \psi_{a,b}^{\alpha,\beta} \rangle = \int_{\mathbb{R}} f(t)\overline{\psi_{a,b}^{\alpha,\beta}(t)}\,\mathrm{d}t, \tag{1.5}$$

where $\{\psi_{a,b}^{\alpha,\beta}\}$ represents the family of fractional continuous wavelet functions, and $\{\psi_{a,b}\}$ denotes the family of classic continuous wavelet functions. The admissibility condition is defined to ensure the invertibility of the fractional wavelet transform [14].

Wavelet transforms naturally provide a multiresolution analysis (MRA) of a function. MRA decomposes a function into a series of approximation and detail components. This is particularly advantageous in signal compression and denoising, where different frequency components can be handled independently:

$$f(t) = \sum_{m\in\mathbb{Z}}\sum_{k\in\mathbb{Z}} d_{m,k}\psi_{m,k}(t), \qquad d_{m,k} = \langle f, \psi_{m,k}\rangle. \tag{1.6}$$

The decomposition allows for a compact representation of function f(.) in terms of a small number of significant coefficients, with the remainder being negligible.

In addition to this, Guo et al. introduced a novel concept of fractional multiresolution analysis (FrMRA), which extends classical MRA to the fractional domain. FrMRA decomposes the function space $L^2(\mathbb{R})$ into a nested sequence of closed subspaces $\{V_m^{\alpha,\beta}\}_{m\in\mathbb{Z}}$, called approximation subspaces, where each subspace is spanned by a family of fractional scaling functions $\varphi_{m,n}^{\alpha,\beta}$. Similarly, a series of wavelet spaces, $\{W_m^{\alpha,\beta}\}$, is spanned by fractional wavelet functions $\psi_{m,n}^{\alpha,\beta}$, leading to a decomposition of $L^2(\mathbb{R})$ as follows:

$$L^2(\mathbb{R}) = V_{m_0}^{\alpha,\beta} \bigoplus_{m\geq m_0} W_m^{\alpha,\beta}, \tag{1.7}$$

or equivalently,

$$L^2(\mathbb{R}) = \bigoplus_{m\in\mathbb{Z}} W_m^{\alpha,\beta}, \tag{1.8}$$

where $\bigoplus$ denotes the direct sum.

The fractional wavelet transform enables the representation of a function $f(t)$ in $L^2(\mathbb{R})$ as:

$$f(t) = \sum_{k\in\mathbb{Z}} a_k\varphi_{m_0,k}^{\alpha,\beta}(t) + \sum_{m\geq m_0}\sum_{k\in\mathbb{Z}} d_{m,k}\psi_{m,k}^{\alpha,\beta}(t), \tag{1.9}$$

or equivalently,

$$f(t) = \sum_{m\in\mathbb{Z}}\sum_{k\in\mathbb{Z}} d_{m,k}\psi_{m,k}^{\alpha,\beta}(t), \tag{1.10}$$

where $a_k = \langle f, \varphi_{m_0,k}^{\alpha,\beta}\rangle$ and $d_{m,k} = \langle f, \psi_{m,k}^{\alpha,\beta}\rangle$, with m_0 representing the lowest scale in the decomposition.

This innovative framework not only enhances the adaptability of wavelet transforms but also provides a more flexible and efficient means of analyzing complex signals across different scales and fractional orders.

1.4. **Flexible fractional wavelet neural networks (FFrWNNs).** Fractional wavelet neural networks (FrWNN) presented in this paper are advanced models that integrate the principles of NNs with fractional wavelet theory. Fractional wavelets enhance the process of feature extraction, and their integration with NNs enables FrWNNs to efficiently capture both temporal and frequency-domain information. The use of wavelet transforms in these models enables data compression and denoising, making the input data more manageable and facilitating the learning process. Additionally, the memory effects inherent in fractional wavelets allow FrWNNs to model complex dynamics with greater accuracy than conventional networks, making them suitable for a variety of tasks, including regression, prediction, and classification.

FrWNNs are particularly well-suited for solving nonlinear problems. The wavelet transforms ability to extract meaningful features at multiple scales ensures that the network can capture both global trends and fine details from the input data. The fractional wavelet transform (FrWT) can be viewed as a band-pass filter operating in the fractional domain, breaking the limitations of traditional wavelet transforms by offering additional flexibility in feature extraction. This flexibility has led to successful applications in a variety of domains, including ECG signal denoising [17], visual sensors and wearable devices [25], hyperspectral image sensors [7], and power harmonic and inter-harmonic detection [18].

FrWNNs are particularly advantageous for processing and analyzing signal data, such as audio, images, and time-series data. Their ability to handle complex temporal dependencies makes them highly suitable for tasks like stock price forecasting, market trend analysis, and other financial applications. Moreover, they can be used for analyzing medical signals and images, such as ECG or MRI data, providing valuable insights for diagnosis and monitoring in healthcare.

In this work, the relations from Equations (1.8) and (1.10) serve as the theoretical foundation for the proposed FFrWNN model. In this architecture, fractional wavelets act as differentially scaled bandpass filters in the fractional domain, utilizing a fractional window that enhances the network's ability to model a broader range of phenomena compared to conventional NNs. The wavelet transform also plays a crucial role in reducing noise in the input data, which contributes to the model's overall robustness and reliability.

The key characteristics of the proposed FFrWNN are summarized as follows:

- Enhanced flexibility: The fractional parameters of the wavelet (α and β) add an additional layer of flexibility to the network. These parameters can be adjusted during training to improve the network's approximation power, making it adaptable to a wider range of problems.
- Universal approximation: As a generalization of feed-forward neural networks and wavelet networks, the FFr-WNN inherits their ability to serve as a universal approximator. This property, combined with the fractional wavelet transform, allows the network to avoid local minima, enhancing the stability and performance of the model.
- Initialization using fractional wavelets: Similar to conventional wavelet networks, fractional wavelet theory is employed to initialize the network parameters and weights. This approach ensures that the network can be efficiently optimized, with the number of hidden layer nodes adjustable to meet the requirements of specific applications.
- Stochastic gradient-based learning: A stochastic gradient descent (SGD) algorithm is proposed for training the FFrWNN, and the stability of this algorithm is formally proven. This ensures that the network converges to a stable solution, even in complex, high-dimensional spaces.
- Customizable wavelet types: By selecting different types of fractional wavelets (e.g., orthogonal or non-orthogonal continuous wavelets), the network architecture can be customized to create various types of wavelet-based networks. This flexibility allows the FFrWNN to be tailored to specific applications.

The remainder of this paper is organized as follows: First, we provide a detailed illustration of the proposed FFrWNN topology and the stochastic gradient descent-based training algorithm. In section 3, we analyze and prove the stability conditions of the network using the Lyapunov stability approach. In section 4, we present several benchmark examples to evaluate the performance of the proposed FFrWNN and compare its outcomes to those of other similar networks. Finally, the paper concludes with a summary of the findings and recommendations for future research directions.

2. Structure of the proposed fractional wavelet network

The structure of the general multidimensional FFrWNN is designed by incorporating fractional wavelet functions as the activation functions within a single hidden layer feed-forward neural network (see Figure 1). This architecture enables the FFrWNN to perform non-linear input-output mapping, denoted as $\mathbb{R}^n \longrightarrow \mathbb{R}^m$, allowing it to process multidimensional input and output signals.

For a given input signal vector $\mathbf{X}^p = [x_{1p}, x_{2p}, \ldots, x_{np}]^T$, the output of the general multidimensional FFrWNN, $\widehat{\mathbf{Y}}^p = [\widehat{y}_{1p}, \widehat{y}_{2p}, \ldots, \widehat{y}_{mp}]^T$, with M wavelons (neurons with fractional wavelet activation functions) in the hidden layer,

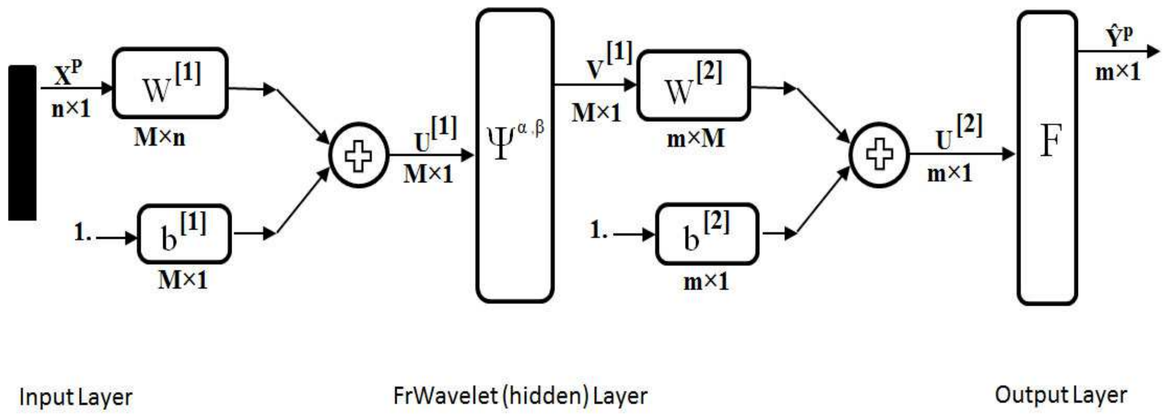

Figure 1. Structure of the proposed FFrWNN.

Table 1. Notations used in the structure of FFrWNN and input-output relations.

Term	Meaning
$N \in \mathbb{N}$	Number of training samples
$NT \in \mathbb{N}$	Number of test data
$p = 1, 2, \cdots, N$	Counter for training samples
$M \in \mathbb{N}$	Number of wavelons (nodes) in hidden layer
$n \in \mathbb{N}$	Input dimension
$m \in \mathbb{N}$	Output dimension
$\mathbf{X}^p \in \mathbb{R}^n$	Input vectors of FFrWNN
$\widehat{\mathbf{Y}}^p \in \mathbb{R}^m$	Output vectors of FFrWNN
$\psi(.)$	Continuous wavelet function
$\psi^{\alpha,\beta}(.)$	fractional continuous wavelet function
$W^{[1]} \in \mathbb{C}^{M\times n}$	Input layer weight matrix
$W^{[2]} \in \mathbb{C}^{m\times M}$	Output layer weight matrix
$\mathbf{b}^{[1]} \in \mathbb{C}^{M\times 1}$	Input layer bias vector
$\mathbf{b}^{[2]} \in \mathbb{C}^{m\times 1}$	Output layer bias vector
$a \in \mathbb{R}^{M\times 1}$	Scaling parameter of wavelets
$b \in \mathbb{R}^{M\times 1}$	Translation parameter of wavelets
$\alpha, \beta \in \mathbb{R} - \{k\pi\}$	Fractional parameters of wavelets
$V^{[1]} \in \mathbb{C}^{M\times 1}$	Hidden layer output
$U^{[1]} \in \mathbb{R}^{M\times 1}$	First layer output

can be mathematically expressed as:

$$\widehat{y}_{kp} = F_k\left(\sum_{j=1}^{M} W_{kj}^{[2]} \psi_j^{\alpha,\beta}\left(U_{jp}^{[1]}\right)\right) + b_k^{[2]}, \quad \text{for all } k = 1, 2, \ldots, m \text{ and } p = 1, 2, \ldots, N$$

The notations used in this expression, as well as the structure of the FFrWNN, are detailed in Table 1. The function $\psi_j^{\alpha,\beta}$ is the activation function of the jth wavelon, defined with two key parameters, a_j and b_j, representing the scaling

Table 2. Parameters type in FFrWNN.

Hyperparameters (Pre set)	Weights and parameters (adjusted with learning algorithm)
M (The number of wavelons)	$W^{[1]}, W^{[2]}$ (Network weights)
Γ_i (Learning rate matrix)	$b^{[1]}, b^{[2]}$ (Network biases)
B_s (Batch size)	$\mathbf{a}_j, \mathbf{b}_j$ (Scaling and translation parameters of jth wavelon)
.	α, β (Fractional wavelet parameters)

and translation factors, respectively:

$$\psi_j^{\alpha,\beta}(t) = \psi_{a_j,b_j}^{\alpha,\beta}(t) = \psi_{a_j,b_j}(t) e^{-i\mathrm{A}_j^{\alpha,\beta}(t)}, \quad (j = 1, 2, \ldots, M)$$

where $\mathrm{A}_j^{\alpha,\beta}(t)$ is given by:

$$\mathrm{A}_j^{\alpha,\beta}(t) = \frac{1}{2}\left[(t^2 - b_j^2)\cot\alpha - \left(\frac{t - b_j}{a_j}\right)^2 \cot\beta\right],$$

and

$$\psi_{a_j,b_j}(t) = \psi_j(t) = \psi\left(\frac{t - b_j}{a_j}\right).$$

The matrix representation of the proposed FFrWNN is as follows:

$$\mathbf{U}^{[1]} = \mathbf{W}^{[1]}\,\mathbf{X}^p + \mathbf{b}^{[1]}$$

$$\widehat{\mathbf{Y}} = F\left(\mathbf{W}^{[2]}\,\mathbf{V}^{[1]} + \mathbf{b}^{[2]}\right)$$

Here, $\mathbf{U}^{[1]} = [u_1, u_2, \ldots, u_M]^T$, and

$$u_j = \sum_{i=1}^{n} W_{ji}^{[1]} x_{ip} + b_j^{[1]}, \quad (j = 1, 2, \ldots, M),$$

$$\mathbf{V}^{[1]} = \left[\psi_1^{\alpha,\beta}(u_1), \psi_2^{\alpha,\beta}(u_2), \ldots, \psi_M^{\alpha,\beta}(u_M)\right]^T.$$

The activation function of the output layer, denoted by F, can be selected from a variety of well-known functions such as linear, sigmoid, or others. In this work, we assume that F is a linear transfer function, meaning $F(\mathbf{U}^{[2]}) = \mathbf{U}^{[2]}$.

2.1. **Training the proposed FFrWNN.** In continuation of the previous section, the parameters of the proposed FFrWNN are summarized in Table 2. Depending on the type of wavelet network, the scaling and translation parameters may either be adjusted during network training (for continuous wavelet networks) or determined using wavelet analysis theory (for orthogonal wavelet networks).

In this paper, the wavelet components are fully integrated into the learning process of the perceptron architecture. Therefore, in the following section, we derive the iterative formulas for adjusting the scaling and translation parameters of each fractional wavelet neuron.

While it is possible to assign constant values to the fractional wavelet parameters (α, β) and train the FFrWNN while keeping these values fixed, in the general case, to increase the network's efficiency and flexibility, these parameters are also updated during the learning process. The iterative formula for updating (α, β) is provided, allowing for adaptive tuning during training.

In this section, we employ the stochastic gradient descent (SGD) learning algorithm, one of the most widely used and powerful methods for training NNs. SGD is based on gradient descent and enables efficient optimization of the FFrWNN's parameters by iteratively adjusting them to minimize the error between the network's predictions and the actual outputs.

Consider the input-output data set $\{(X^p, Y^p) \,|\, X^p \in \mathbb{R}^n, Y^p \in \mathbb{R}^m, p = 1, 2, \dots, N\}$, which is presented to the FFrWNN in a random order, one pattern at a time. For each pattern p ($p = 1, 2, \dots, N$), let the network output be denoted as $\widehat{\mathbf{Y}}^p$. The error between the actual output and the predicted output is given by:

$$\mathbf{E}(p) = \mathbf{Y}^p - \widehat{\mathbf{Y}}^p.$$

The online and offline loss (cost) functions are defined as:

$$L(\theta, p) = \frac{1}{2}(\mathbf{Y}^p - \widehat{\mathbf{Y}}^p)^T(\mathbf{Y}^p - \widehat{\mathbf{Y}}^p) = \frac{1}{2}\|\mathbf{Y}^p - \widehat{\mathbf{Y}}^p\|^2 = \frac{1}{2}\sum_{k=1}^{m}(y_{kp} - \widehat{y}_{kp})^2,$$

$$L(\theta) = \frac{1}{2N}\sum_{p=1}^{N}\|\mathbf{Y}^p - \widehat{\mathbf{Y}}^p\|^2 = \frac{1}{N}\sum_{p=1}^{N}L(\theta, p),$$

where θ represents the set of FFrWNN synaptic weights and parameters. In the full-term case, θ is defined as:

$$\theta = [\mathbf{W}^{[1]}, \mathbf{W}^{[2]}, \mathbf{b}^{[1]}, \mathbf{b}^{[2]}, \mathbf{a}, \mathbf{b}, \alpha, \beta]^T$$

To compute the optimal values of the network's weights and parameters, the cost function $L(\theta, p)$ must be minimized. The optimal parameters are found by solving:

$$\widehat{\theta} = \arg\min L(\theta, p),$$

In order to reduce the number of parameters and eliminate less important ones, a regularization term $\mu\sum_{i=1}^{Np}\|\theta_i\|^2$ can be added to the cost function, where Np is the number of parameters and μ is a regularization coefficient.

An iterative approach is commonly used to update the network parameters. At each iteration k, the update rule is:

$$\Delta\widehat{\theta}(p) = -\Gamma\nabla L(\widehat{\theta}(p)).$$

For $i = 1, 2, \dots, Np$, the parameter update rule becomes:

$$\widehat{\theta}_i(p+1) = \widehat{\theta}_i(p) - \gamma_i\frac{\partial L(\widehat{\theta}(p))}{\partial\theta_i(p)},$$

where $\Gamma = \mathrm{diag}(\gamma_1, \gamma_2, \dots, \gamma_{Np})$ is a diagonal matrix of learning rates.

Since the gradient of the offline cost function is the mean gradient of the online cost function, the proposed network can be trained using either online (sequential) or offline (batch) learning:

$$\nabla L(\widehat{\theta}) = \frac{1}{N}\sum_{p=1}^{N}\nabla L(\widehat{\theta}, p).$$

The gradient of the cost function with respect to each parameter is given by:

$$\frac{\partial L(\theta, p)}{\partial\theta_i} = -E(p)\,\frac{\partial\widehat{\mathbf{Y}}^p}{\partial\theta_i},$$

which can be expanded as:

$$\frac{\partial L(\theta, p)}{\partial\theta_i} = \sum_{k=1}^{m} -(y_{kp} - \widehat{y}_{kp}) \times \frac{\partial\widehat{y}_{kp}}{\partial\theta_i}.$$

This iterative method allows the fractional wavelet neural network to adjust its parameters in response to the observed error, optimizing both the traditional and fractional wavelet components.

2.1.1. *Updating method for network weights.* Next, the gradient of the cost function for each of the network's weights is calculated as:

$$\frac{\partial L(\theta,p)}{\partial \mathbf{W}^{[2]}} = -E(p)\,\frac{\partial \widehat{\mathbf{Y}}(X^p,\theta)}{\partial \mathbf{W}^{[2]}} = -E(p)\,[\mathbf{V}^{[1]}]^T, \tag{2.1}$$

$$\frac{\partial L(\theta,p)}{\partial \mathbf{b}^{[2]}} = -E(p)\,\frac{\partial \widehat{\mathbf{Y}}(X^p,\theta)}{\partial \mathbf{b}^{[2]}} = -E(p), \tag{2.2}$$

$$\frac{\partial L(\theta,p)}{\partial \mathbf{W}^{[1]}} = -E(p)\,\frac{\partial \widehat{\mathbf{Y}}(X^p,\theta)}{\partial \mathbf{W}^{[1]}} = -\left[\dot{\mathbf{V}}^{[1]}\,[\mathbf{W}^{[2]}]^T\,E(p)\right](X^p)^T, \tag{2.3}$$

$$\frac{\partial L(\theta,p)}{\partial \mathbf{b}^{[1]}} = -E(p)\,\frac{\partial \widehat{\mathbf{Y}}(X^p,\theta)}{\partial \mathbf{b}^{[1]}} = -\left[\dot{\mathbf{V}}^{[1]}\,[\mathbf{W}^{[2]}]^T\,E(p)\right], \tag{2.4}$$

where $\dot{\mathbf{V}}^{[1]}$ is a diagonal matrix defined as follows:

$$\dot{\mathbf{V}}^{[1]} = \begin{pmatrix} \frac{\partial \psi_1^{\alpha,\beta}}{\partial u_{1p}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\partial \psi_M^{\alpha,\beta}}{\partial u_{Mp}} \end{pmatrix}.$$

In general, the matrix expression of updating scheme of network weights can be summarized as follows:

$$\mathbf{W}^{[2]}(p+1) = \mathbf{W}^{[2]}(p) - \Gamma_{\mathbf{W}^{[2]}}\left[-E(p)\,(\mathbf{V}^{[1]})^T\right], \tag{2.5}$$

$$\mathbf{b}^{[2]}(p+1) = \mathbf{b}^{[2]}(p) - \Gamma_{\mathbf{b}^{[2]}}\left[-E(p)\right], \tag{2.6}$$

$$\mathbf{W}^{[1]}(p+1) = \mathbf{W}^{[1]}(p) - \Gamma_{\mathbf{W}^{[1]}}\left[S^{[1]}(p)\,(X^p)^T\right], \tag{2.7}$$

$$\mathbf{b}^{[1]}(p+1) = \mathbf{b}^{[1]}(p) - \Gamma_{\mathbf{b}^{[1]}}\left[S^{[1]}(p)\right], \tag{2.8}$$

where

$$S^{[1]}(p) = -\left[\dot{\mathbf{V}}^{[1]}\,[\mathbf{W}^{[2]}]^T\,E(p)\right].$$

Remark 2.1. Given that the weights in the first layer ($\mathbf{W}^{[1]}$) have complex values, the imaginary component of $\mathbf{b}^{[1]}$ is used to ensure that the argument of the fractional wavelet function ($\{u_j^{[1]} \mid j = 1, 2, \ldots, M\}$) remains real during the network's learning process. Moreover, since the weights in the last layer ($\mathbf{W}^{[2]}$) are also complex, and the fractional wavelet functions are inherently complex, the imaginary part of $\mathbf{b}^{[2]}$ is employed to convert the network's output back to a real value, facilitating meaningful interpretations and accurate learning throughout the network's training phase.

2.1.2. *Updating method for wavelet parameters.* To update the wavelet parametersspecifically the scale a, translation b, and fractional parameters (α,β)we apply the following method for the parameter vector θ in the full-term mode.

Let $s = 1, 2, \ldots, M$, where M is the total number of wavelets. The gradient of the loss function $L(\theta,p)$ with respect to the wavelet scale parameters a_s is calculated as:

$$\frac{\partial L(\theta,p)}{\partial a_s} = \sum_{k=1}^{m} -(y_{kp} - \widehat{y}_{kp})\left(\frac{\partial \widehat{y}_{kp}}{\partial a_s}\right) \tag{2.9}$$

$$= \sum_{k=1}^{m} -E_{kp} \sum_{j=1}^{M} W_{kj}^{[2]}\left(\frac{\partial \psi_{a_j,b_j}^{\alpha,\beta}(u_{jp})}{\partial a_s}\right) \tag{2.10}$$

$$= \sum_{k=1}^{m} -E_{kp}\, W_{ks}^{[2]}\left(\frac{\partial \psi_{a_s,b_s}^{\alpha,\beta}(u_{sp})}{\partial a_s}\right) \tag{2.11}$$

$$= \sum_{k=1}^{m} -E_{kp}\, W_{ks}^{[2]} \left(\frac{\partial V_{sp}^{[1]}}{\partial a_s} \right), \tag{2.12}$$

where E_{kp} is the k'th element of vector $E(p) = [E_{1p}, E_{2p}, \dots, E_{mp}]^T$.

In matrix form, this can be expressed as:

$$\frac{\partial L(\theta,p)}{\partial \boldsymbol{a}} = -\left[\mathbf{W}^{[2]}\ \frac{\partial \mathbf{V}^{[1]}}{\partial \boldsymbol{a}} \right]^T\ E(p), \tag{2.13}$$

where $\frac{\partial \mathbf{V}^{[1]}}{\partial \boldsymbol{a}}$ is a diagonal matrix defined as:

$$\frac{\partial \mathbf{V}^{[1]}}{\partial \boldsymbol{a}} = \begin{pmatrix} \frac{\partial \psi_1^{\alpha,\beta}}{\partial a_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\partial \psi_M^{\alpha,\beta}}{\partial a_M} \end{pmatrix}.$$

Each individual element is computed as:

$$\frac{\partial V_{jp}^{[1]}}{\partial a_j} = \frac{\partial \psi_j^{\alpha,\beta}(u_j)}{\partial a_j} \tag{2.14}$$

$$= \frac{\partial \psi_j(u_j)}{\partial a_j}\, e^{A_j^{\alpha,\beta}(u_j)} - i \left(\frac{\partial\ A_j^{\alpha,\beta}(u_j)}{\partial a_j} \right) \psi_j^{\alpha,\beta}(u_j). \tag{2.15}$$

Similarly, the gradient with respect to the translation parameters b_s is given by:

$$\frac{\partial L(\theta,p)}{\partial \boldsymbol{b}} = -\left[\mathbf{W}^{[2]}\ \frac{\partial \mathbf{V}^{[1]}}{\partial \boldsymbol{b}} \right]^T\ E(p). \tag{2.16}$$

The matrix $\frac{\partial \mathbf{V}^{[1]}}{\partial b}$ is also diagonal, with components similar to those in (2.14) and (2.15).

Next, we update the fractional wavelet parameters α and β using the following gradients. For α:

$$\frac{\partial L(\theta,p)}{\partial \alpha} = \sum_{k=1}^{m} -(y_{kp} - \widehat{y}_{kp}) \left(\frac{\partial \widehat{y}_{kp}}{\partial \alpha} \right)$$

$$= \sum_{k=1}^{m} -E_{kp} \sum_{j=1}^{M} W_{kj}^{[2]} \left(\frac{\partial \psi_{a_j,b_j}^{\alpha,\beta}(u_{jp})}{\partial \alpha} \right). \tag{2.17}$$

For β:

$$\frac{\partial L(\theta,p)}{\partial \beta} = \sum_{k=1}^{m} -(y_{kp} - \widehat{y}_{kp}) \left(\frac{\partial \widehat{y}_{kp}}{\partial \beta} \right)$$

$$= \sum_{k=1}^{m} -E_{kp} \sum_{j=1}^{M} W_{kj}^{[2]} \left(\frac{\partial \psi_{a_j,b_j}^{\alpha,\beta}(u_{jp})}{\partial \beta} \right), \tag{2.18}$$

The partial derivatives of the wavelet function with respect to α and β are:

$$\frac{\partial \psi_{a_j,b_j}^{\alpha,\beta}(u_{jp})}{\partial \alpha} = -i \left(\frac{\partial\ A_j^{\alpha,\beta}(u_j)}{\partial \alpha} \right) \psi_{a_j,b_j}^{\alpha,\beta}(u_{jp}), \tag{2.19}$$

$$\frac{\partial \psi_{a_j,b_j}^{\alpha,\beta}(u_{jp})}{\partial \beta} = -i \left(\frac{\partial\ A_j^{\alpha,\beta}(u_j)}{\partial \beta} \right) \psi_{a_j,b_j}^{\alpha,\beta}(u_{jp}), \tag{2.20}$$

where the terms $\frac{\partial A_j^{\alpha,\beta}(u_j)}{\partial\alpha}$ and $\frac{\partial A_j^{\alpha,\beta}(u_j)}{\partial\beta}$ are calculated as:

$$\frac{\partial\ A_j^{\alpha,\beta}(u_j)}{\partial\alpha} = \frac{1}{2}\left(u_{jp}^2 - b_j^2\right)\left(1+\cot^2\alpha\right), \tag{2.21}$$

$$\frac{\partial\ A_j^{\alpha,\beta}(u_j)}{\partial\beta} = \frac{1}{2}\left(\frac{u_{jp}-b_j}{a_j}\right)^2\left(1+\cot^2\beta\right). \tag{2.22}$$

In matrix form, this can be expressed as:

$$\frac{\partial L(\theta,p)}{\partial\alpha} = -E^T(p)\ \mathbf{W}^{[2]}\ \left(\frac{\partial\mathbf{V}^{[1]}}{\partial\alpha}\right), \qquad \frac{\partial L(\theta,p)}{\partial\beta} = -E^T(p)\ \mathbf{W}^{[2]}\ \left(\frac{\partial\mathbf{V}^{[1]}}{\partial\beta}\right), \tag{2.23}$$

where $\frac{\partial\mathbf{V}^{[1]}}{\partial\alpha}$ and $\frac{\partial\mathbf{V}^{[1]}}{\partial\beta}$ are $M\times 1$ vectors defined as :

$$\frac{\partial\mathbf{V}^{[1]}}{\partial\alpha} = \left[\frac{\partial\psi_{a_1,b_1}^{\alpha,\beta}(u_{1p})}{\partial\alpha}, \frac{\partial\psi_{a_2,b_2}^{\alpha,\beta}(u_{2p})}{\partial\alpha}, \cdots, \frac{\partial\psi_{a_M,b_M}^{\alpha,\beta}(u_{Mp})}{\partial\alpha}\right]^T, \tag{2.24}$$

$$\frac{\partial\mathbf{V}^{[1]}}{\partial\beta} = \left[\frac{\partial\psi_{a_1,b_1}^{\alpha,\beta}(u_{1p})}{\partial\beta}, \frac{\partial\psi_{a_2,b_2}^{\alpha,\beta}(u_{2p})}{\partial\beta}, \cdots, \frac{\partial\psi_{a_M,b_M}^{\alpha,\beta}(u_{Mp})}{\partial\beta}\right]^T. \tag{2.25}$$

Finally, the matrix form of the update rules for wavelet parameters can be summarized as follows:

$$\boldsymbol{a}(p+1) = \ \boldsymbol{a}(p) - \Gamma_{\mathbf{a}}\ \left[\mathbf{W}^{[2]}\ \frac{\partial\mathbf{V}^{[1]}}{\partial\boldsymbol{a}}\right]^T\ - E(p), \tag{2.26}$$

$$\boldsymbol{b}(p+1) = \ \boldsymbol{b}(p) - \Gamma_{\mathbf{b}}\ \left[\mathbf{W}^{[2]}\ \frac{\partial\mathbf{V}^{[1]}}{\partial\boldsymbol{b}}\right]^T\ - E(p), \tag{2.27}$$

$$\alpha(p+1) = \alpha(p) - \Gamma_\alpha\left(-E^T(p)\ \mathbf{W}^{[2]}\right)\ \left(\frac{\partial\mathbf{V}^{[1]}}{\partial\alpha}\right), \tag{2.28}$$

$$\beta(p+1) = \beta(p) - \Gamma_\beta\left(-E^T(p)\ \mathbf{W}^{[2]}\right)\left(\frac{\partial\mathbf{V}^{[1]}}{\partial\beta}\right). \tag{2.29}$$

Remark 2.2. We initialized the hyperparameters using a trial-and-error approach. However, it is important to note that several advanced methods exist for optimizing network performance, including regularization techniques (e.g., L1/L2 regularization, wavelet pruning, and dropout), hyperparameter tuning methods (e.g., Adaptive Gradient Algorithms, Genetic Algorithms, and metaheuristic algorithms), and network initialization strategies (e.g., the Xavier Initialization Method).

2.2. **The stability analysis.** The stability of the proposed algorithm for FFrWNN is established through the Lyapunov stability theory. Let fractional parameters $\alpha, \beta \neq k\Pi$ $(k\in\mathbb{Z})$ are arbitrary values and $\theta_\star = [\mathbf{W}_\star^{[1]}, \mathbf{W}_\star^{[2]}, \mathbf{b}_\star^{[2]}, \mathbf{b}_\star^{[1]}]^T$ represent the true or ideal weights of simplified model. Assume that:

$$\widetilde{W}^{[1]} = W_\star^{[1]} - W^{[1]}, \quad \widetilde{b}^{[1]} = b_\star^{[1]} - b^{[1]},$$
$$\widetilde{W}^{[2]} = W_\star^{[2]} - W^{[2]}, \quad \widetilde{b}^{[2]} = b_\star^{[2]} - b^{[2]}.$$

Furthermore,

$$\begin{aligned}\widetilde{U}(p) = U_\star^{[1]} - U^{[1]}(p) &= \left(W_\star^{[1]}X + b_\star^{[1]}\right) - \left(W^{[1]}(p)X + b^{[1]}(p)\right)\\ &= \left(W_\star^{[1]} - W^{[1](p)}\right)X - \left(b_\star^{[1]} + b^{[1](p)}\right)\\ &= \widetilde{W}^{[1]}(p)\ X + \widetilde{b}^{[1]}(p).\end{aligned} \tag{2.30}$$

Remark 2.3. Due to the arbitrary fixed values of fractional parameters α, β, in subsequent relations, we avoid writing the superscripts of Ψ function, for simplify notations

$$
\begin{aligned}
V^{[1]}(p) &= \Psi^{\alpha,\beta}(W^{[1]}(p)X(p) + b^{[1]}(p)) = \Psi(W^{[1]}(p)X(p) + b^{[1]}(p)),\\
V_*^{[1]} &= \Psi^{\alpha,\beta}(W_\star^{[1]}X(p) + b_\star^{[1]}) = \Psi(W_\star^{[1]}X(p) + b_\star^{[1]}),
\end{aligned}
$$

Using the ideal weights, the actual output $Y(p)$ can be expressed as:

$$
Y(p) = W_\star^{[2]}V_\star^{[1]} + b_\star^{[2]} + \varepsilon(p). \tag{2.31}
$$

where $\varepsilon(p)$ represents the unmodeled dynamics. Additionally, the parametric error dynamics are given by:

$$
\begin{aligned}
E(p) &= Y(p) - \widehat{Y}(p)\\
&= W_\star^{[2]}V_\star^{[1]} + b_\star^{[2]} + \varepsilon(p) - \left[W^{[2]}(p)V^{[1]}(p) + b^{[2]}(p)\right]\\
&= W_\star^{[2]}\Psi(W_\star^{[1]}X(p) + b_\star^{[1]}) + b_\star^{[2]} + \varepsilon(p) - \left[W^{[2]}(p)\Psi(W^{[1]}(p)X(p) + b^{[1]}(p)) + b^{[2]}(p)\right]\\
&= W^{[2]}(p)\Psi(W^{[1]}(p)X(p) + b^{[1]}(p)) + \widetilde{W}^{[2]}(p)\Psi(W^{[1]}(p)X(p) + b^{[1]}(p))\\
&+ W^{[2]}(p)\dot{\Psi}(W^{[1]}(p)X(p) + b^{[1]}(p))\widetilde{W}^{[1]}(p)X(p) + W^{[2]}(p)\dot{\Psi}(W^{[1]}(p)X(p) + b^{[1]}(p))\widetilde{b}^{[1]}(p)\\
&+ R(p) + b_\star^{[2]} + \varepsilon(p) - \left[W^{[2]}(p)\Psi(W^{[1]}(p)X(p) + b^{[1]}(p)) + b^{[2]}(p)\right]\\
&= \widetilde{W}^{[2]}(p)\Psi(W^{[1]}(p)X(p) + b^{[1]}(p)) + W^{[2]}(p)\dot{\Psi}(W^{[1]}(p)X(p) + b^{[1]}(p))\widetilde{W}^{[1]}(p)X(p)\\
&+ W^{[2]}(p)\dot{\Psi}(W^{[1]}(p)X(p) + b^{[1]}(p))\widetilde{b}^{[1]}(p) + R(p) + \widetilde{b}^{[2]}(p) + \varepsilon(p)\\
&= \widetilde{W}^{[2]}(p)V^{[1]}(p) + W^{[2]}(p)\dot{V}^{[1]}(p)\widetilde{W}^{[1]}(p)X(p) + W^{[2]}(p)\dot{V}^{[1]}(p)\widetilde{b}^{[1]}(p) + \widetilde{b}^{[2]}(p) + \zeta(p)\\
&= \widetilde{W}^{[2]}(p)V^{[1]}(p) + W^{[2]}(p)\dot{V}^{[1]}(p)\left(\widetilde{W}^{[1]}(p)X(p) + \widetilde{b}^{[1]}(p)\right) + \widetilde{b}^{[2]}(p) + \zeta(p)\\
&= \widetilde{W}^{[2]}(p)V^{[1]}(p) + W^{[2]}(p)\dot{V}^{[1]}(p)\widetilde{U}(p) + \widetilde{b}^{[2]}(p) + \zeta(p),
\end{aligned} \tag{2.32}
$$

where $\zeta(p) = R(p) + \varepsilon(p)$. Here, $R(p)$ is the Taylor series remainder derived from Taylor's theorem.

Theorem 2.4. *Consider the FFrWNN with network weights $W^{[2]}$, $b^{[2]}$, $W^{[1]}$, $b^{[1]}$, and weight update laws as expressed in (2.26-2.29). By arbitrary fractional parameters $\alpha, \beta \neq k\Pi$ $(k \in \mathbb{Z})$ and defining the following notations:*

$$
\begin{aligned}
&\eta_1 = \max\left(\|V^{[1]}\|^2\right),\ \eta_2 = \max\left(\|X\|^2\right),\ \eta_3 = \max\left(\|\dot{V}^{[1]}\ W^{[2]T}\|^2\right),\\
&\xi = 2 + \eta_1\ \lambda_{\max}(\Gamma_3) + \lambda_{\max}(\Gamma_4) + \eta_3\ \left[\eta_2\ \lambda_{\max}(\Gamma_1) + \lambda_{\max}(\Gamma_2)\right].
\end{aligned}
$$

If $\xi < 2\ E(p)\ \zeta(p)$ then, the tracking error $E(p)$ asymptotically converges to zero.

Proof. Lyapunov functions are employed to analyze stability by constructing a function that decreases over time, ensuring convergence to a stable solution, therefore the following Lyapunov function will be a candidate:

$$
\begin{aligned}
\mathbf{v}(p) &= tr\left([\widetilde{W}^{[1]}(p)]^T\ \Gamma_1^{-1}\ \widetilde{W}^{[1]}(p)\right) + tr\left([\widetilde{b}^{[1]}(p)]^T\ \Gamma_2^{-1}\ \widetilde{b}^{[1]}(p)\right)\\
&+ tr\left([\widetilde{W}^{[2]}(p)]^T\ \Gamma_3^{-1}\ \widetilde{W}^{[2]}(p)\right) + tr\left([\widetilde{b}^{[2]}(p)]^T\ \Gamma_4^{-1}\ \widetilde{b}^{[2]}(p)\right),
\end{aligned} \tag{2.33}
$$

where the notation tr refers to the trace of a matrix. Thus,

$$
\begin{aligned}
\Delta\mathbf{v}(p) &= v(p+1) - v(p)\\
&= 2\,tr\left([\widetilde{W}^{[1]}(p)]^T\quad \Gamma_1^{-1}\quad \Delta\widetilde{W}^{[1]}(p)\right) + tr\left([\Delta\widetilde{W}^{[1]}(p)]^T\quad \Gamma_1^{-1}\quad [\Delta\widetilde{W}^{[1]}(p)]\right)\\
&+ 2\,tr\left([\widetilde{b}^{[1]}(p)]^T\quad \Gamma_2^{-1}\quad \Delta\widetilde{b}^{[1]}(p)\right) + tr\left([\Delta\widetilde{b}^{[1]}(p)]^T\quad \Gamma_2^{-1}\quad [\Delta\widetilde{b}^{[1]}(p)]\right)
\end{aligned}
$$

$$
\begin{aligned}
&+ 2\, tr \left([\widetilde{W}^{[2]}(p)]^T \quad \Gamma_3^{-1} \quad \Delta\widetilde{W}^{[2]}(p) \right) + tr \left([\Delta\widetilde{W}^{[2]}(p)]^T \quad \Gamma_3^{-1} \quad [\Delta\widetilde{W}^{[2]}(p)] \right) \\
&+ 2\, tr \left([\widetilde{b}^{[2]}(p)]^T \quad \Gamma_4^{-1} \quad \Delta\widetilde{b}^{[1](p)} \right) + tr \left([\Delta\widetilde{b}^{[2]}(p)]^T \quad \Gamma_4^{-1} \quad [\Delta\widetilde{b}^{[2]}(p)] \right). \qquad (2.34)
\end{aligned}
$$

The first line of (2.34) results from the following calculations:

$$
\begin{aligned}
& tr \left([\widetilde{W}^{[1]}(p+1)]^T \, \Gamma_1^{-1} \, \widetilde{W}^{[1]}(p+1) \right) - tr \left([\widetilde{W}^{[1]}(p)]^T \, \Gamma_1^{-1} \, \widetilde{W}^{[1]}(p) \right) \\
&= tr \left(\left[\widetilde{W}^{[1]}(p) + \Delta\widetilde{W}^{[1]}(p) \right]^T \, \Gamma_1^{-1} \, \left[\widetilde{W}^{[1]}(p) + \Delta\widetilde{W}^{[1]}(p) \right] \right) - tr \left([\widetilde{W}^{[1]}(p)]^T \, \Gamma_1^{-1} \, \widetilde{W}^{[1]}(p) \right) \\
&= tr \left([\widetilde{W}^{[1]}(p)]^T \, \Gamma_1^{-1} \, \widetilde{W}^{[1]}(p) \right) + tr \left([\Delta\widetilde{W}^{[1]}(p)]^T \, \Gamma_1^{-1} \, \widetilde{W}^{[1]}(p) \right) + tr \left([\widetilde{W}^{[1]}(p)]^T \, \Gamma_1^{-1} \, \Delta\widetilde{W}^{[1]}(p) \right) \\
&+ tr \left([\Delta\widetilde{W}^{[1]}(p)]^T \, \Gamma_1^{-1} \, \Delta\widetilde{W}^{[1]}(p) \right) - tr \left([\widetilde{W}^{[1]}(p)]^T \, \Gamma_1^{-1} \, \widetilde{W}^{[1]}(p) \right) \\
&= 2\, tr \left([\widetilde{W}^{[1]}(p)]^T \quad \Gamma_1^{-1} \quad \widetilde{W}^{[1]}(p) \right) + tr \left([\Delta\widetilde{W}^{[1]}(p)]^T \quad \Gamma_1^{-1} \quad [\Delta\widetilde{W}^{[1]}(p)] \right). \qquad (2.35)
\end{aligned}
$$

Similar computations are used to derive the remaining terms in (2.34). It is important to note that:

$$
\begin{aligned}
&\Delta\widetilde{\mathbf{W}}^{[2]}(p) = -\Delta\mathbf{W}^{[2]}(p), \quad \Delta\widetilde{\mathbf{b}}^{[2]}(p) = -\Delta\mathbf{b}^{[2]}(p), \\
&\Delta\widetilde{\mathbf{b}}^{[1]}(p) = -\Delta\mathbf{b}^{[1]}(p), \quad \Delta\widetilde{\mathbf{W}}^{[1]}(p) = -\Delta\mathbf{W}^{[1]}(p),
\end{aligned}
$$

since:

$$
\begin{aligned}
\Delta\widetilde{W}^{[1]}(p) &= \widetilde{W}^{[1]}(p+1) - \widetilde{W}^{[1]}(p) \\
&= \left[W_\star^{[1]} - W^{[1]}(p+1) \right] - \left[W_\star^{[1]} - W^{[1]}(p) \right] \\
&= - \left[W^{[1]}(p+1) - W^{[1]}(p) \right] = -\Delta W^{[1]}(p). \qquad (2.36)
\end{aligned}
$$

Using relations (2.5–2.8), we can express $\Delta v(p)$ as follows:

$$
\begin{aligned}
\Delta\mathbf{v}(p) &= 2\, tr \left([\widetilde{W}^{[1]}(p)]^T \, \dot{V}^{[1]}(p) \, [\widetilde{W}^{[2]}(p)]^T \, E \, X^T(p) \right) \\
&+ tr \left(X(p) \, E^T(p) \, W^{[2]}(p) \, \dot{V}^{[1]}(p) \Gamma_1 \dot{V}^{[1]}(p) \, [W^{[2]}(p)]^T \, E(p) \, X^T(p) \right) \\
&+ 2\, tr \left([\widetilde{b}^{[1]}(p)]^T \, \dot{V}^{[1]}(p) \, [\widetilde{W}^{[2]}(p)]^T \, E(p) \right) \\
&+ tr \left(E^T(p) \, W^{[2]}(p) \, \dot{V}^{[1]}(p) \Gamma_2 \dot{V}^{[1]}(p) \, [W^{[2]}(p)]^T \, E(p) \right) \\
&+ 2\, tr \left([\widetilde{W}^{[2]}(p)]^T \, E(p) \, [V^{[1]}(p)]^T \right) + tr \left(V^{[1]}(p) \, E^T(p) \, \Gamma_3 \, E(p) \, [V^{[1]}(p)]^T \right) \\
&+ 2\, tr \left([\widetilde{b}^{[2]}(p)]^T \, E \right) + tr \left(E^T(p) \, \Gamma_4 \, E(p) \right) \\
&= 2\, tr \left(E^T(p) \, W^{[2]}(p) \, \dot{V}^{[1]}(p) \left[\widetilde{W}^{[1]}(p) \, X + \widetilde{b}^{[1]}(p) \right] + E^T(p) \, \widetilde{W}^{[2]}(p) \, V^{[1]}(p) \, + E^T(p) \, \widetilde{b}^{[2]}(p) \right) \\
&+ tr \left(X^T(p) \, X \, E^T(p) \, W^{[2]}(p) \dot{V}^{[1]}(p) \Gamma_1 \dot{V}^{[1]}(p) \, [W^{[2]}(p)]^T \, E(p) \right) \\
&+ tr \left(E^T(p) \, W^{[2]}(p) \dot{V}^{[1]}(p) \Gamma_2 \dot{V}^{[1]}(p) \, [W^{[2]}(p)]^T \, E \right) \\
&+ tr \left([V^{[1]}(p)]^T \, V^{[1]}(p) \, E^T(p) \, \Gamma_3 \, E(p) \right) + tr \left(E^T(p) \, \Gamma_4 \, E(p) \right) \\
&= 2\, tr \left(E^T(p) \left[W^{[2]}(p) \, \dot{V}^{[1]}(p) \widetilde{U}(p) + \widetilde{W}^{[2]}(p) \, V^{[1]}(p) + \widetilde{b}^{[2]}(p) \right] \right) \\
&+ \|X(p)\|^2 \, tr \left(S^T(p) \, \Gamma_1 \, S(p) \right) + tr \left(S^T(p) \, \Gamma_2 \, S(p) \right) \\
&+ \|V^{[1]}(p)\|^2 \, tr \left(E^T(p) \, \Gamma_3 \, E(p) \right) + tr \left(E^T(p) \, \Gamma_4 \, E(p) \right). \qquad (2.37)
\end{aligned}
$$

From Equation (2.32), we have

$$\widetilde{W}^{[2]}(p)V^{[1]}(p) + W^{[2]}(p)\dot{V}^{[1]}(p))\widetilde{U}(p) + \widetilde{b}^{[2]}(p) = E(p) - \zeta(p). \tag{2.38}$$

Therefore,

$$\begin{aligned}
\Delta \mathbf{v}(p) &= 2\, tr\left[E(p)^T\left[E(p) - \zeta(p)\right]\right] + \|X(p)\|^2\, \|S(p)^{[1]}\|^2\, \lambda_{Max}(\Gamma_1) + \|S(p)^{[1]}\|^2\, \lambda_{Max}(\Gamma_2)\\
&+ \|V^{[1]}(p)\|^2\, \|E(p)\|^2\, \lambda_{Max}(\Gamma_3) + \|E(p)\|^2\, \lambda_{Max}(\Gamma_4)\\
&\le \|E(p)\|^2\left[2 + \|V(p)\|^2\, \lambda_{Max}(\Gamma_3) + \lambda_{Max}(\Gamma_4)\right]\\
&+ \|S^{[1]}(p)\|^2\left[\|X\|^2\, \lambda_{Max}(\Gamma_1) + \lambda_{Max}(\Gamma_2)\right] - 2E(p)^T\, \zeta(p)\\
&\le \|E(p)\|^2\,[2 + \eta_1\, \lambda_{Max}(\Gamma_3) + \lambda_{Max}(\Gamma_4)]\\
&+ \|\dot{V}^{[1]}\, W^{[2]T}\, E(p)\|^2\,[\eta_2\, \lambda_{Max}(\Gamma_1) + \lambda_{Max}(\Gamma_2)] - 2E^T(p)\, \zeta(p)\\
&\le \xi\, \|E(p)\|^2 - 2E^T(p)\, \zeta(p) \;<\; 0, \qquad (p > 0).
\end{aligned} \tag{2.39}$$

This implies $0 < \mathbf{v}(p) < \mathbf{v}(0)$ for all $p > 0$. Consequently, $\lim_{p\to\infty} \mathbf{v}(p) < \infty$. From Equation (2.39), we obtain:

$$0 < \xi \sum_{p=0}^{\infty} \|E(p)\|^2 - 2\sum_{p=0}^{\infty} E(p)^T\, \zeta(p) < -\sum_{p=0}^{\infty} \mathbf{v}(p) = \mathbf{v}(0) - \mathbf{v}_\infty < \infty. \tag{2.40}$$

Thus, $\lim_{p\to\infty} E(p) = 0$. □

Remark 2.5. The assumptions of Theorem 2.4 incorporate unmodeled dynamics, identification error and learning rate matrices. These assumptions ensure that the learning process and output dynamics of the FFrWNN converge to a desired state over time, without exhibiting unbounded or oscillatory behavior. During training, the weights of the FFrWNN are updated iteratively to minimize a predefined cost function. By ensuring that the Lyapunov function decreases over time, the convergence of the FFrWNN to the desired equilibrium point is guaranteed. This property establishes the FFrWNN as a reliable tool for applications such as black-box identification, control systems, and dynamic modeling tasks.

3. Experimental results

To demonstrate the effectiveness of the proposed FFrWNN, simulation and identification experiments are conducted on several benchmark systems considered. All the algorithms are implemented using MATLAB R2018b. The FFrWNN employs Gaussian wavelet, Morelet wavelet, and Mexican hat (Maar) wavelet, similar to traditional wavelet networks commonly used in practical applications. By incorporating the term in Equation (1.2), general fractional wavelets with two fractional variables (α and β) are introduced and serve as the activation functions within the network.

Example 3.1. Narendra et al. proposed a second-order difference equation for identifying a planetary system in their paper [26]. The system is described as follows:

$$y(k+1) = f\left(y(k), y(k-1)\right) + u(k), \qquad y(0) = 1,\; y(1) = 0.5,$$

where

$$f\left(y(k), y(k-1)\right) = \frac{y(k)y(k-1)\left[y(k) + 2.5\right]}{1 + y^2(k) + y^2(k-1)},$$

and $u(k) = \sin(0.08k\pi)$ is the selected input signal.

For system identification, the input vector is defined as $X_p = [u(k-2), u(k-1), y(k-3), y(k-2), y(k-1)]$. The number of training samples is set to $N = 9000$ and test samples to $NT = 1000$. The hyperparameters of the FFrWNN are configured as follows: $M = 4$ or 5, $B_s = 1$, and learning rates $\Gamma_{W^{[1]}} = \Gamma_{b^{[1]}} = \Gamma_{W^{[2]}} = \Gamma_{b^{[2]}} = 0.12$, and $\Gamma_\alpha = \Gamma_\beta = 0.03$. The weights and parameters are initialized using uniformly distributed pseudorandom numbers in the range $[-1, 1]$.

The network's actual output and the estimated output for test data (with $M = 5$ and number of parameters (NP) $= 48$) are shown in Figures 2 and 3. The mean square error (MSE) is used to evaluate the networks performance.

TABLE 3. FFrWNN performance compared to SNN and HONN.

Model	NP	Train MSE	Test MSE	Adjusted α, β
SNN	70	1.0399e-02	1.7896e-03	—
SNN	140	4.4874e-02	7.3730e-04	—
HONN	41	9.3136e-04	1.2567e-04	—
HONN	51	3.3819e-04	1.9484e-05	—
FFrWNN	39	3.4324e-04	1.7775e-04	$\alpha = 1.04898,\ \beta = 1.74403$
FFrWNN	48	9.9267e-06	1.4306e-06	$\alpha = 1.5501,\ \beta = 1.1891$

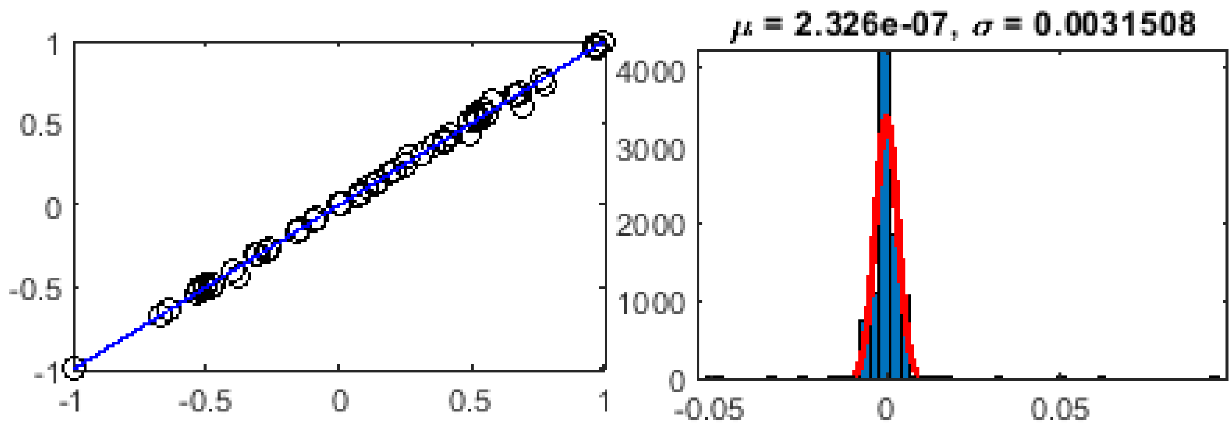

FIGURE 2. MSE, correlation coefficient, error mean and standard deviation for identifying test data in Example 3.1 by proposed FFrWNN (NP=39).

Table 3 compares the results of the FFrWNN with those of the hyperbolic orthogonal neural network (HONN) and the stable sinusoidal neural network (SNN) from [3] and [4].

Comment: *Initially, the fractional wavelet parameters α and β are randomly assigned values ($\alpha = 0.3745$, $\beta = 1.8012$) and optimized during training, yielding final values of $\alpha = 1.5501$ and $\beta = 1.1891$. Training these fractional parameters significantly improves the network's accuracy and reduces errors, as evidenced by the network's performance metrics.*

Example 3.2. Next, we consider a non-linear plant governed by the following difference equation [1, 16, 22]:

$$y(k) = 0.72\, y(k-1) + 0.025\, y(k-2)u(k-1) + 0.01\, u^2(k-2) + 0.02\, u(k-3), \quad y(0) = 1,\ y(1) = 0.5. \tag{3.1}$$

This plant has long input delays, and its identification is driven by the following excitation signal:

$$u(k) = \begin{cases} \sin(k\pi/25), & 0 \le k < 250, \\ +1.0, & 250 \le k < 500, \\ -1.0, & 500 \le k < 750, \\ 0.3\sin(k\pi/25) + 0.1\sin(k\pi/32) + 0.6\sin(k\pi/10), & 750 \le k \le 1000. \end{cases} \tag{3.2}$$

The input vector for the identification task is set as $X = [u(k-3), u(k-2), u(k-1), y(k-2), y(k-1)]$. The model is trained over 200 epochs with 1000 time steps per epoch. Hyperparameters of the FFrWNN are set as $M = 3$, $B_s = 1$, and learning rates $\Gamma_{W^{[1]}} = \Gamma_{b^{[1]}} = \Gamma_{W^{[2]}} = \Gamma_{b^{[2]}} = 0.08$, $\Gamma_\alpha = \Gamma_\beta = 0.015$.

The root mean square error (RMSE) is used as the performance criterion. Figure 4 shows the network's actual and estimated output, along with the RMSE and correlation coefficient for the training data (with NP = 30). Table 4

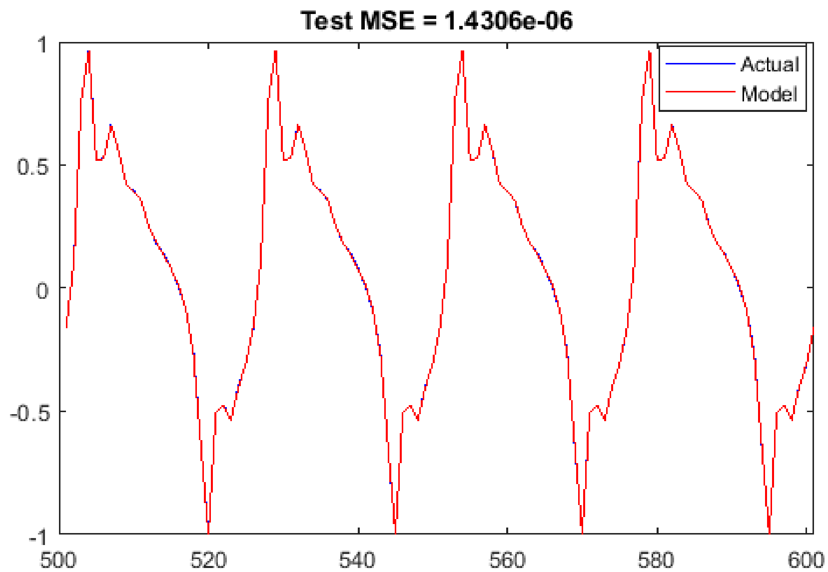

FIGURE 3. Test MSE for identifying test data in Example 3.2 by proposed FFrWNN(NP=39).

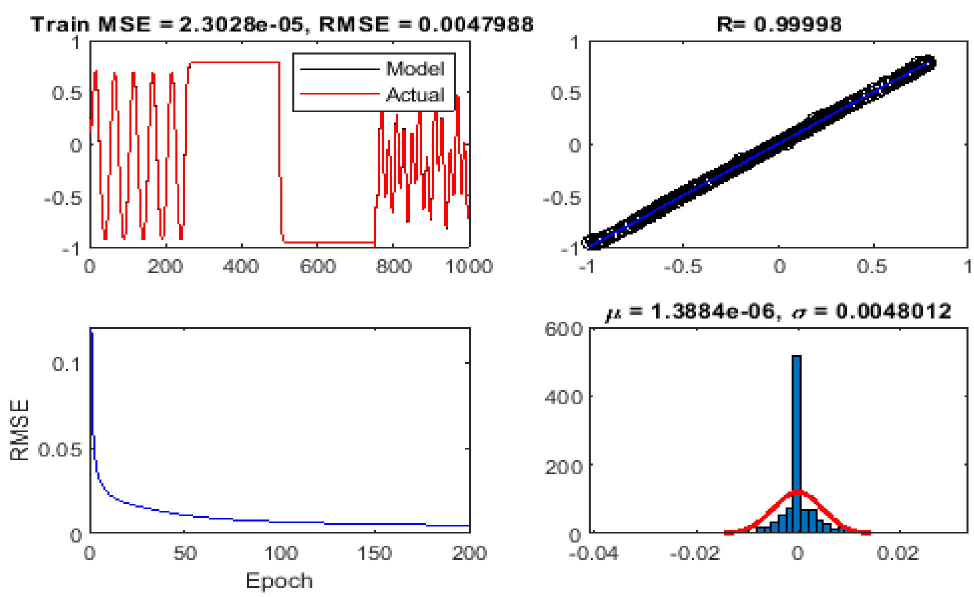

FIGURE 4. The actual state, the estimated state and the errors of identifying test data in Example 3.3 by FFrWNN with 4 waveleons (NP=30).

compares the FFrWNN's performance with other models such as Elmans recurrent neural network (ERNN), recurrent self-organizing neural fuzzy inference network (RSONFIN), and fuzzy wavelet neural networks (FWNN).

Table 4. Proposed FFrWNN performance(RMSE) compared to some models for Example 3.1.

Models	NP	Train RMSE	Adjusted α, β
RSONFIN[20]	49	0.03	
TRFN[19]	33	0.0067	
FWNN[1]	43	0.018713	
ADLAWNNs[22]	–	0.0032	
FrWNN	30	0.004799	$\alpha = 0.980032,\ \beta = 1.648344$
FrWNN	39	0.002107	$\alpha = -0.56532,\ \beta = 1.705115$

Comment: *The FFrWNN, with 39 parameters, outperforms other models while requiring fewer network parameters and offering faster processing due to its lightweight structure. The fractional wavelet parameter tuning further enhances the network's performance.*

Example 3.3. In this example, we aim to identify a non-linear dynamical system described by the following difference equation, as discussed in [26]:

$$y(k+1) = f\left[y(k), y(k-1), y(k-2), u(k), u(k-1)\right], \qquad y(0) = 0.8,\ y(1) = 0.5,\ y(2) = 0. \tag{3.3}$$

The unknown non-linear function f, which we seek to determine, is given by:

$$f\left[x_1, x_2, x_3, x_4, x_5\right] = \frac{x_1 x_2, x_3, x_4, x_5(x_3 - 1) + x_4}{1 + x_2^2 + x_3^2}, \tag{3.4}$$

where $x_1 = y(k)$, $x_2 = y(k-1)$, $x_3 = y(k-2)$, $x_4 = u(k)$, and $x_5 = u(k-1)$.
For this system, as in [23], the external input signal is defined as:

$$u(k) = \sin\left(\frac{2k\pi}{25}\right). \tag{3.5}$$

During training, we use 800 input-output samples and train the model for 800 epochs.
To identify the plant in Equation (3.3) using the proposed FFrWNN, we consider two input configurations for the model:
(1) $X = \left[u(k-1), u(k), y(k-2), y(k-1), y(k)\right]^T$,
(2) $X = \left[u(k), y(k)\right]^T$.
The responses of the FFrWNN identifiers at the end of training are shown in Figure 5, while the root mean square error (RMSE) obtained during training is displayed in Figure 6.
Table 5 compares the performance of the proposed FFrWNN with other methods under similar conditions. Despite using fewer parameters and requiring fewer epochs, the FFrWNN outperforms alternative approaches, proving its effectiveness for this system identification task.

Example 3.4. In this example, we consider a multi-input, multi-output (MIMO) non-linear dynamic system as studied in [26] and [3], represented by the following system of equations:

$$\mathbf{Y(k+1)} = \begin{bmatrix} y_1(k+1) \\ y_2(k+1) \end{bmatrix} = \begin{bmatrix} \frac{y_1(k)}{1+(y_2(k))^2} \\ \frac{y_1(k)y_2(k)}{1+(y_2(k))^2} \end{bmatrix} + \mathbf{U(k)}, \tag{3.6}$$

where the initial conditions are $y_1(0) = y_2(0) = 0$, and the input signals are defined as:

$$\mathbf{U(k)} = \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} = \begin{bmatrix} \sin\left(\frac{2k\pi}{25}\right) \\ \cos\left(\frac{2k\pi}{25}\right) \end{bmatrix}. \tag{3.7}$$

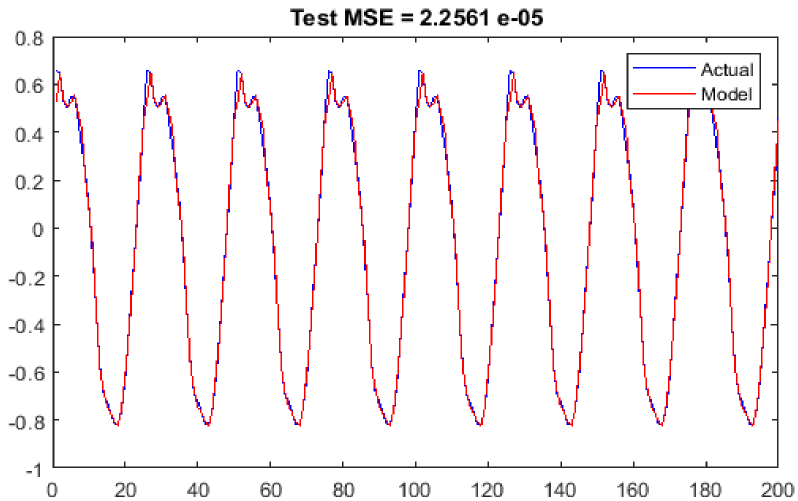

FIGURE 5. Response of FFrWNN identifier at the end of training (the number of inputs is n=5 and NP=39) in Example 3.3.

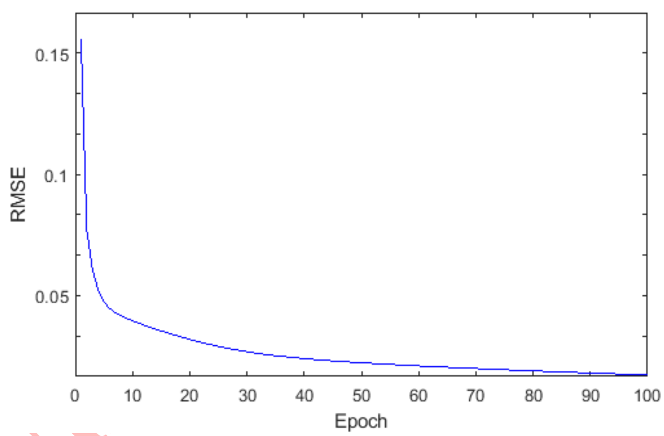

FIGURE 6. Root mean square error (RMSE) of FFrWNN identifier during the training (the number of inputs is n=5 and NP=39) in Example 3.3.

To identify this plant using FFrWNN, we define the input vector of the model as:

$$X = [u_1(k-1), u_2(k-1), y_1(k-1), y_2(k-1)]^T . \tag{3.8}$$

Table 6 compares the performance of the FFrWNN with other existing methods. The results demonstrate that the proposed FFrWNN requires fewer network parameters and epochs to achieve superior performance, highlighting its computational efficiency and effectiveness in solving system identification problems.

Figures 7 show the real system states, the estimated states, and the associated errors during the testing phase for the FFrWNN model with 12 hidden neurons. The results show a high degree of accuracy in the models estimation.

TABLE 5. FFrWNN performance in 800 training data per epoch compared to some models for Example 3.3.

Models	NP	No. of inputs	Total No. of epochs	Learning rate	Min. AMSE	Adjusted α, β
SRWNN [23]	72	02	1000	Adaptive	0.000596	—-
SRWNN [23]	72	02	1000	Fixed	0.00325	—-
WNN	110	05	1000	Fixed	0.0051	—-
FFrWNN (M=4)	39	05	100	Fixed	0.000673	$\alpha = 1.77859,\ \beta = 1.83847$
FFrWNN (M=5)	48	05	100	Fixed	0.000479	$\alpha = 1.77310,\ \beta = 1.90179$
FFrWNN (M=6)	39	02	100	Fixed	0.004309	$\alpha = 0.84932,\ \beta = 1.90567$
FFrWNN (M=9)	57	02	100	Fixed	0.003357	$\alpha = 0.89453,\ \beta = 1.41059$

TABLE 6. FFrWNN performance compared to some models for Example 3.4.

Models	NP	Train MSE	Adjusted α, β
SNN	70	8.0354e-03	——-
SNN	140	1.4402e-02	——-
HONN	41	2.7960e-03	——-
HONN	57	8.2662e-04	——-
HONN	105	5.9180e-04	——-
FFrWNN (M=5)	49	2.585e-3	$\alpha = 1.450954,\ \beta = 1.503939$
FFrWNN (M=6)	58	6.1530e-04	$\alpha = 1.122610,\ \beta = 1.786311$
FFrWNN (M=12)	112	2.870e-04	$\alpha = 1.426055,\ \beta = 1.733292$

Comment: *Initially, the fractional parameters α and β were randomly initialized to $\alpha = 0.5739$ and $\beta = 1.0012$. During the training process, these values converged to their optimal values of $\alpha = 1.4261$ and $\beta = 1.7333$, leading to significant improvements in the network's performance.*

4. CONCLUSION

In this article, we introduced the flexible fractional wavelet neural network (FFrWNN) as an extension of traditional wavelet networks, offering greater flexibility through the inclusion of fractional parameters. To optimize the network, we developed a training algorithm based on the stochastic gradient descent (SGD) method, which allows not only for the adjustment of the network weights but also for the fine-tuning of wavelet translation, dilation, and fractional parameters. The simulation results on non-linear models illustrate that tuning the wavelet's fractional parameters significantly enhances the network's ability to accurately identify complex dynamic systems. In particular, the FFrWNN demonstrated superior performance compared to other NNs, achieving lower identification errors under similar conditions. This highlights the potential of fractional parameters in refining system identification tasks. Future research directions could explore the broader applicability of the FFrWNN, especially in areas such as system control, classification, and image processing, which were beyond the scope of this study. Investigating the network's performance in these domains could reveal further advantages and open up new possibilities for its use in real-world applications.

REFERENCES

[1] R. H. Abiyev and O. Kaynak, *Fuzzy wavelet neural networks for identification and control of dynamic plantsA novel structure and a comparative study*, IEEE Transactions on Industrial Electronics, *55*(8) (2008), 3133–3140.

[2] S. Agarwal and S. Gupta, *Applications of Fractional Wavelets in Non-Stationary Signal Analysis*, Mathematical Methods in the Applied Sciences, *44*(14) (2021), 9532–9549.

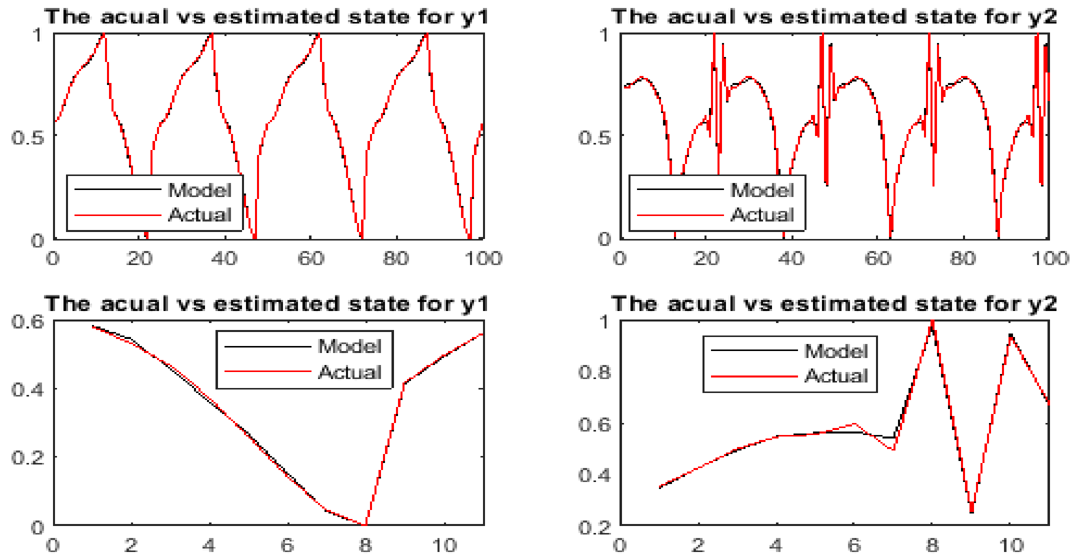

FIGURE 7. Response of FFrWNN identifiers at the end of training in the Example 3.4 (NP=112).

[3] G. Ahmadi, *Stochastic gradient-based hyperbolic orthogonal neural networks for nonlinear dynamic systems identification*, Journal of Mathematical Modeling, *10*(3) (2022), 529–547.

[4] G. Ahmadi and M. Teshnehlab, *Designing and implementation of stable sinusoidal rough-neural identifier*, IEEE Trans. Neural Netw. Learn. Syst., *28*(8) (2017), 1774–1786.

[5] A. K. Alexandridis and A. D. Zapranis, *Wavelet neural networks: A practical guide*, Neural Networks, *42*, (2013), 1–27.

[6] K. Anoh, S. Jones, O. Ochonogor, and Y. Dama, *Novel fractional wavelet transform with closed-form expression*, Int J Adv Comput Sci Appl., (2014), 182–187.

[7] S. Bajpai and N. R. Kidwai, *Fractional wavelet filter based low memory coding for hyperspectral image sensors*, Springer Multimedia Tools and Applications, *83* (2024), 26281–26306.

[8] I. V. Bazhlekov and A. K. Grozdev, *Fractional Wavelets for Multi-Scale Image Analysis and Compression*, Journal of Mathematical Imaging and Vision, *62*(4) (2020), 445–461.

[9] A. Coviello, F. Linsalata, U. Spagnolini, and M. Magarini, *Artificial Neural Networks-Based Real-Time Classification of ENG Signals for Implanted Nerve Interfaces*, IEEE Selected Areas in Communications, *42* (2024), 2080–2095.

[10] H. Dai, Z. Zheng, and W. Wang, *A new fractional wavelet transform*, Commun. Non-linear Sci. Numer. Simul., *44* (2017), 19–36.

[11] X. Gao and L. Li, *Time-Frequency Analysis of Non-Stationary Signals Using Fractional Wavelets*, IEEE Transactions on Signal Processing, *70* (2022), 1554–1565.

[12] J. Gonzalez and J. Villarreal, *Fractional Wavelet Transform in Chaotic Systems*, Chaos, Solitons and Fractals, *144*(110689) (2021).

[13] I. Goodfellow, Y. Bengio, and A.Courville, *Deep Learning (Adaptive Computation and Machine Learning series)*, MIT Press, (2016).

[14] Y. Guo, B. Z. Li, and L. D. Yang, *Novel fractional wavelet transform: Principles, MRA and application*, Digit. Signal Process., *110* (2021).

[15] T. Guo, T. Zhang, E. Lim, M. Lopez-Benitez, F. Ma, and L. Yu, *A review of wavelet analysis and its applications: challenges and opportunities*, IEEE Access., *10* (2022), 58869–58903.

[16] H. G. Han, Z. L. Lin, and J. F. Qiao, *Modeling of nonlinear systems using the self-organizing fuzzy neural network with adaptive gradient algorithm*, Neurocomputing, *226* (2017), 566–578.
[17] I. Houamed, L. Saidi, and F. Srairi, *ECG signal denoising by fractional wavelet transform thresholding*, Research on Biomedical Engineering, (2020), 349–360.
[18] D. Jiawei, Y. Xiaoyu, and Y. Xiaozhong, *Power harmonic and inter-harmonic detection based on fractional wavelet transform combined with variational modal decomposition algorithm*, Review of scientific instruments, *94*(12) (2023).
[19] C. F. Juang, *A TSK-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms*, IEEE Trans. Fuzzy Syst., *10*(2) (2002), 155–170.
[20] C. F. Juang and C. T. Lin, *A recurrent self-organizing neural fuzzy inference network*, IEEE Trans. Neural Netw., *10*(4) (1999), 828–845.
[21] K. P. Kakarla, D. C. S.Kiran, and M. Kanchana, *Stock Price Prediction Using LSTM, CNN and ANN* , Springer, Soft Computing and Signal Processing. ICSCSP., (2023), 141–153.
[22] C. Ko , *Identification of nonlinear systems with outliers using wavelet neural networks based on annealing dynamical learning algorithm*, Engineering Applications of Artificial Intelligence, *25* (2012), 533–543.
[23] R. Kumar, S. Srivastava, J. R. P. Gupta, and A. Mohindru, *Self-recurrent wavelet neural networkbased identification and adaptive predictive control of nonlinear dynamical systems*, Int J Adapt Control Signal Process, (2018), 1–33.
[24] D. Mendlovic, Z. Zalevsky, D. Mas, J. Garca, and C. Ferreira, *Fractional wavelet transform*, Appl. Opt., *36* (1997), 4801–4806.
[25] T. Mohd , J. Abhinandan, K. Ekram, and H. Mohd, *Low Memory Architectures of Fractional Wavelet Filter for Low-Cost Visual Sensors and Wearable Devices*, IEEE sensors journal, *20*(13) (2020), 6863–6871.
[26] K. S. Narendra and K. Parthasarathy, *Identification and control of dynamical systems using neural networks*, IEEE Trans. Neural Netw., *1* (1990), 4–27.
[27] P. Podder, S. Bharati, M. R. H. Mondal, P. K. Paul, and U. Kose, *Artificial Neural Network for Cybersecurity: A Comprehensive Review*, Journal of Information Assurance and Security., *16* (2021), 10–23.
[28] J. Shi, N. T. Zhang, and X. P. Liu, *A novel fractional wavelet transform and its applications*, Sci. China Inf. Sci., *55*(6) (2012), 1270–1279.
[29] H. M. Srivastava, K. Khatterwani, and S. K. Upadhyay, *A certain family of fractional wavelet transformations*, Math Meth Appl Sci., (2019), 1–20.
[30] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press., (1996).
[31] N. V. Sukhanova and S. A. Sheptunov *The Neural Network Models in Mechanical Engineering,*, International Conference on Quality Management, Transport and Information Security, Information Technologies., (2021), 551–553.
[32] Z. Wang and J. Xu, *Fractional Wavelet Transforms for Signal Processing: A Survey and New Results*, Signal Processing., *158* (2019), 1–16.
[33] X. Y. Yang, D. Baleanu, H. M. Srivastava, and J. A. T. Machado, *On local fractional continuous wavelet transform*, Abstr Appl Anal., *2013*(1) (2013), 1–5.
[34] Q. Zhang, and A. Benveniste, *Wavelet networks*, IEEE Trans Neural Netw., (1992), 889–898.