Journal of Mathematical Modeling
Vol. 12, No. 4, 2024, pp. 733–752. Research Article

JMM

Using a modified Sinc neural network to identify the chaotic systems with an application in wind speed forecasting

Ghasem Ahmadi*

Department of Mathematics, Payame Noor University, Tehran, Iran
Email(s): g.ahmadi@pnu.ac.ir

Abstract. Continuous-time models for dynamic nonlinear systems offer greater reliability than their discrete-time counterparts. Discrete-time models can suffer from information loss and increased noise susceptibility. Chaotic and hyperchaotic systems pose significant challenges due to their unpredictable nature. These systems are prevalent in various fields, including weather, climate, finance, and biology. Artificial neural networks, inspired by the human nervous system, are effective in approximating complex nonlinear systems. A recent innovation, the Sinc neural network (SNN), leverages the properties of the Sinc function, which is smooth and oscillatory, making it suitable for approximation tasks. Despite limited research, SNNs have shown promising results in applications like speech recognition, human motion recognition, and fractional optimal control problems. This study introduces a modified Sinc neural network (MSNN) to enhance the performance of SNN in identifying continuous-time nonlinear systems. The MSNN employs a stable online training algorithm based on Lyapunov stability theory. It is utilized to identify several chaotic systems, including the Duffing-Van der Pol oscillator, the Lorenz system, and a financial hyperchaotic system. Additionally, the MSNN is used for forecasting wind speed, an important factor in renewable energy generation. Data from Khorramabad, Iran, is utilized for this purpose. The MSNN's simple structure and strong performance in identifying nonlinear systems and forecasting wind speed demonstrate its potential.

Keywords: Chaotic system, hyperchaotic system, modified Sinc neural network, system identification, wind speed forecasting.
AMS Subject Classification 2010: 93B30, 68T07, 34C28.

1 Introduction

Continuous-time models for dynamic nonlinear systems tend to be more reliable than discrete-time models. This reliability arises since most physical laws and classical theories, such as Newton's laws, Fara-

*Corresponding author
Received: 21 June 2024 / Revised: 7 August 2024 / Accepted: 16 August 2024
DOI: 10.22124/jmm.2024.27752.2442

day's laws, Laplace transforms, and PID controllers, are originally formulated in continuous time. Converting a continuous-time system description to a discrete-time description can result in information loss. Additionally, discrete-time models are more susceptible to the effects of noise compared to the continuous-time models. However, the identification of continuous-time systems is less studied in the literature [1, 12, 15, 17, 24, 25, 31, 33].

Chaotic and hyperchaotic dynamic systems are extremely sensitive to initial conditions, a property often referred to as the butterfly effect [30]. These systems are unpredictable since a small error in the initial conditions can lead to a significant error in predicting their future behavior, despite being governed by deterministic equations. The theory of chaotic systems began in 1963 with Lorenz's work on weather prediction [19]. Many real-world systems exhibit chaotic behavior, including weather, climate, finance, and the human heart. Chaotic systems have applications across several scientific fields, such as electronics, physics, chemistry, economics, and biology. Significant research has been conducted on hyperchaotic systems in recent years [7,34]. These are complex chaotic systems with at least two positive Lyapunov exponents, meaning their trajectories diverge exponentially in multiple directions. This field is active and attractive for research, with applications in secure communication, cryptography, and the control of complex systems.

Artificial neural networks (ANNs) draw inspiration from the human nervous system, comprising computational neurons interconnected through weighted edges. By adjusting these weights with learning rules, ANNs can approximate complex nonlinear functions. Recently, a Sinc neural network (SNN) has been designed based on Sinc numerical methods, which are powerful tools in numerical analysis and approximation theory. The Sinc function is smooth and oscillates between positive and negative values, with its output approaching zero as the input tends to infinity.

There are few studies on ANNs using the Sinc activation function. Notably, Elwasif and Fausett have used SNN with a single input and a single output to approximate functions with one variable [11]. Borra et al. have utilized Sinc-based convolutional neural structures for electroencephalogram (EEG) motor execution decoding [9]. Bria et al. have applied Sinc-based convolutional neural networks for EEG-based Brain-Computer Interface motor imagery classification [10]. Other researchers have applied SNNs to various tasks, including speech recognition [21], automatic speaker and age identification [23], EEG motor imagery classification [18], and human motion recognition [8]. Recently, Heydari and Ahmadi have developed SNNs to solve fractional optimal control problems [14].

In the applications of ANNs, the training methodology is crucial to their performance. Essentially, there are two approaches: online and offline learning. In online learning, parameters are updated with one observation at a time, whereas in offline learning, parameters are updated using all observations. In real-time applications, online learning is often unavoidable. The stability of learning algorithm is essential, and recent years have seen the development of online learning algorithms based on Lyapunov stability theory for training ANNs [2, 4, 5].

In the recent decades, ANNs have proven their usefulness in modeling nonlinear systems. Several works have been done in identifying continuous-time and chaotic nonlinear systems by neural networks. Ren et al. have used the dynamic neural networks to identify continuous-time nonlinear systems [25]. Pozniak et al. used dynamic neural networks to identify chaotic systems [22]. Rubio applied ANNs and Kalman filters to identify chaotic systems [26]. Garcia et al. employed recurrent wavelet first-order neural networks to identify chaotic dynamics in jerky-based systems [20]. Ahmadi utilized rough extreme learning machines for the identification of uncertain continuous-time nonlinear systems [1]. Ahmadi and Dehghandar used rough-neural networks for the prediction of chaotic time series [5]. Forgione and Piga

employed tailor-made neural model structures for the identification of continuous-time systems [12].

To enhance the performance of SNNs, this study designs and applies a modified SNN (MSNN) to identify nonlinear systems. The MSNN is interesting and novel for several reasons:

- Traditional neural networks often rely on large numbers of parameters and layers to capture complex features. MSNN, by leveraging the properties of the Sinc function, can achieve comparable or even superior performance with fewer parameters. This reduces computational load and memory requirements, making it more efficient.
- Due to its reduced parameter space and more structured representation of data, MSNN can be less prone to overfitting compared to traditional networks. This leads to better generalization on unseen data, which is critical for building reliable machine learning models.
- MSNN can be integrated with existing neural network architectures, enhancing performance without requiring a complete overhaul of the model. This flexibility makes it a practical choice for improving existing systems incrementally.
- MSNN incorporates frequency-based information, beneficial for applications such as time-series forecasting, signal processing, and other tasks where frequency patterns are important.

These advantages make MSNN a compelling choice for a range of applications and positions it as a valuable tool in the evolution of neural network design and application.

The proposed model, MSNN, is trained with a stable online algorithm developed based on the Lyapunov stability theory. To demonstrate the efficiency of proposed approach, it is used to identify chaotic systems such as the Duffing-Van der Pol oscillator, the Lorenz system, and a financial hyperchaotic dynamic system. Additionally, MSNN is used to forecast wind speed chaotic time series. Wind, as a green renewable energy source, is of global interest, and its power forecasting depends directly on accurate wind speed forecasting. The data used for wind speed forecasting is gathered from the city of Khorramabad in Iran. MSNN, with a simple structure, demonstrates strong capability in identifying nonlinear systems and wind speed forecasting. The innovations of this work can be summarized as follows:

- A modification of SNN, called MSNN, is designed to improve performance.
- MSNN is applied to identify continuous-time systems.
- MSNN is used to identify crucial chaotic and hyperchaotic systems.
- MSNN is implemented to forecast wind speed chaotic time series.

The rest of this paper is organized as follows: Section 2 describes the Sinc function and its applications. Section 3 explains the structure of MSNN. Section 4 discusses the identification of continuous-time nonlinear systems with MSNN. Section 5 presents a Lyapunov-based learning algorithm for MSNN. Section 6 covers the identification of chaotic and hyperchaotic systems with MSNN. Section 7 demonstrates the use of MSNN to forecast wind speed time series. Finally, Section 8 draws conclusions.

2 Sinc function and its properties

This section is devoted to the definition and properties of Sinc function. This mathematical function arises frequently in signal processing and the theory of Fourier transforms. There are two standard definitions of the Sinc function:

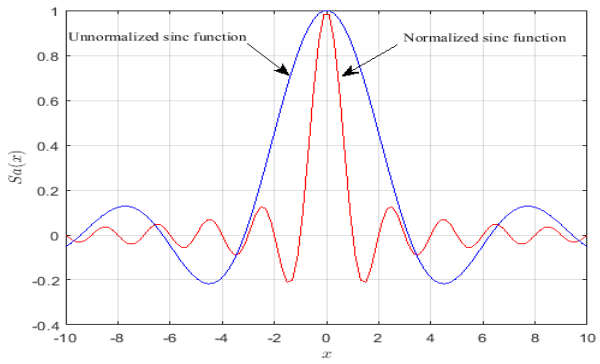

Figure 1: The structure of MSNN.

- Unnormalized Sinc function:

$$Sa(x) = \begin{cases} \frac{\sin(x)}{x}, & x \neq 0, \\ 1, & x = 0. \end{cases} \tag{1}$$

- Normalized Sinc function:

$$Sa(x) = \begin{cases} \frac{\sin(\pi x)}{\pi x}, & x \neq 0, \\ 1, & x = 0. \end{cases} \tag{2}$$

Remark 1. *Sometimes, the Sinc function is called the sampling function. Therefore, we show the Sinc function with $Sa(x)$. In this work, we use the normalized form of Sinc function.*

Sinc function is a beneficial function in different fields of applied sciences such as Sinc interpolation and Sinc numerical methods with valuable results in solving nonlinear problems [29]. It is the inverse Fourier transform of the rectangular function, making it crucial in signal processing for understanding the relations between signals in time and frequency domains [28]. The Sinc function is smooth, oscillating between positive and negative values (Figure 1). Its output approaches zero as the input tends to infinity. This function has been used to form the Whittaker cardinal function as an important approach in approximation theory. Suppose that

$$Sa_i(x) = Sa\left(\frac{1}{h}(x - ih)\right), \quad i = -\infty, \ldots, \infty \tag{3}$$

is the i-th function in the set of Sinc basis functions where h is a positive real number. For the real function f, the Whittaker cardinal function is defined as follows:

$$C(f,h)(x) = \sum_{i=-\infty}^{\infty} f(ih) Sa_i(x). \tag{4}$$

In fact, the relation (4) interpolates function f at the points ih $(i = -\infty \cdots \infty)$.

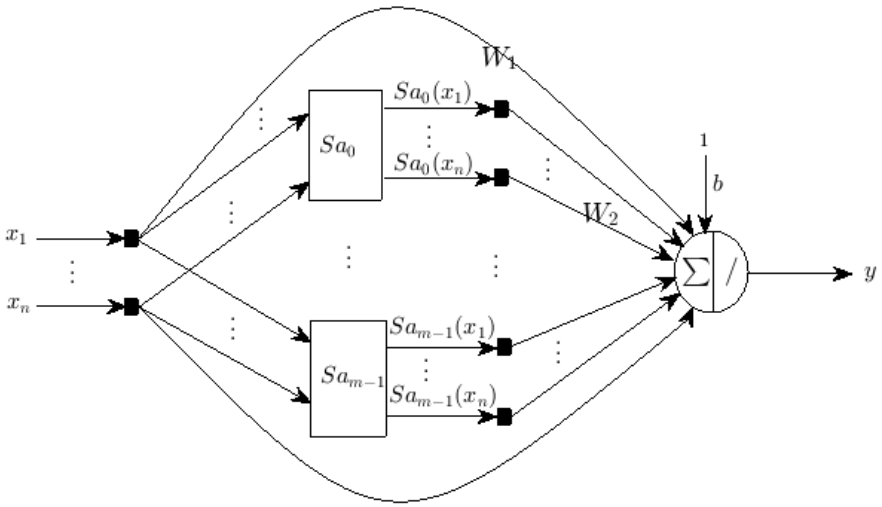

Figure 2: The structure of MSNN.

Suppose that in the complex ω-plane, D_d is defined as follows [27]:

$$D_d = \{\omega = t + is : |s| < d\}. \tag{5}$$

When a problem is defined on a subinterval I of $\mathbb{R}$, we utilize a conformal map ϕ such that $\phi(I) = \mathbb{R}$. Suppose that ϕ be a conformal map, and ζ be its inverse, from the simply-connected domain D containing $(0,1)$, onto D_d. Then, on a subinterval $I = (0,1) = \zeta(R)$ where $\phi(0) = -\infty$ and $\phi(1) = +\infty$, we have

$$f(x) \approx \sum_{i=-N}^{N} f(x_i) Sa_i(\phi(x)), \tag{6}$$

where $x_i = \zeta(ih)$.

3 Modified Sinc neural network (MSNN)

There are few studies on neural networks using the Sinc activation function. One notable example is the work by Elwasif and Fausett in 1996, where SNN with a single input and a single output was used for approximating functions with one variable [11]. Recently, Heydari and Ahmadi have improved the SNN to solve the fractional optimal control problems [14].

Based on the presented structure for SNN in [11, 14], and using the Sinc basis functions described in (3), (4), and (6), this work proposes the MSNN. It has three layers where its hidden layer contains the Sinc basis functions, the inputs are connected to the outputs, and the parameters between inputs and hidden layer are supposed to be one. Let $X = [x_1, x_2, \ldots, x_n]^T$ and $Y = [y_1, y_2, \ldots, y_q]^T$ show the inputs and outputs of MSNN, respectively. Suppose that $S_i(t)$, $i = 0, 1, 2, \ldots, m-1$ be the i-th function in the

set of Sinc basis functions, and

$$\begin{aligned} Sa_i(X) &= [Sa_i(x_1), Sa_i(x_2), \ldots, Sa_i(x_n)]^T, \quad i = 0,1,2,\ldots,m-1, \\ \mathbf{Sa}(X) &= \begin{bmatrix} Sa_0(X) \\ Sa_1(X) \\ \vdots \\ Sa_{m-1}(X) \end{bmatrix}, \end{aligned} \tag{7}$$

W_1 be the matrix parameters between inputs and outputs, W_2 be the matrix of parameters between the hidden nodes and outputs, and $\mathbf{b}$ be the biases of output neurons. Then, we have

$$\begin{aligned} Y &= W_1 X + W_2 \mathbf{Sa}(X) + \mathbf{b} \\ &= \begin{bmatrix} W_1 & W_2 & \mathbf{b} \end{bmatrix} \begin{bmatrix} X \\ \mathbf{Sa}(X) \\ 1 \end{bmatrix} \\ &= W\mathbf{S}(X), \end{aligned} \tag{8}$$

where

$$W = \begin{bmatrix} W_1 & W_2 & \mathbf{b} \end{bmatrix}, \quad \mathbf{S}(X) = \begin{bmatrix} X \\ \mathbf{Sa}(X) \\ 1 \end{bmatrix}. \tag{9}$$

4 System identification by MSNN

Consider the continuous-time nonlinear dynamic system:

$$\dot{Z}(t) = f(Z(t), U(t)), \tag{10}$$

where U and Z denote the vectors of system inputs and system outputs, respectively. Suppose that (10) is completely controllable, and the function f is Lipschitz continuous. Let A be a Hurwitz matrix (i.e. all eigenvalues of A have negative real parts). By adding and subtracting $AZ(t)$ to the right-hand side of (10), it can be rewritten as:

$$\dot{Z}(t) = AZ(t) + g(Z(t), U(t)), \tag{11}$$

where the function $g(Z(t), U(t)) = f(Z(t), U(t)) - AZ(t)$ represents the nonlinear part of (10). Assume that MSNN can approximate the $g(Z(t), U(t))$ with an accuracy of $\varepsilon(t)$, using the optimal parameters $W_\star$. For this purpose, we use the MSNN in nonlinear autoregressive exogenous (NARX) configuration. In this configuration, the inputs vector $U(t)$ and the outputs vectors $Z(t)$ are fed to the model. Therefore, the inputs vector of MSNN can be shown as $X(t) = [U(t), Z(t)]^T$. Then, according to (8), we have

$$\dot{Z}(t) = AZ(t) + W_\star \mathbf{S}(X(t)) + \varepsilon(t), \tag{12}$$

where $\mathbf{S}(X(t))$ is described in the equations (7) and (9). Based on (12), we can write a parametric model of the system (10) as follows:

$$\dot{\widehat{Z}}(t) = A\widehat{Z}(t) + \widehat{W}(t)\mathbf{S}(X(t)), \tag{13}$$

where $\widehat{Z}(t)$ and $\widehat{W}(t)$ are the approximations of $Z(t)$ and $W_\star$, respectively.

Let $E(t) = Z(t) - \widehat{Z}(t)$ indicates the identification error. Using the equations (12) and (13), we have

$$\begin{aligned}\dot{E}(t) &= \dot{Z}(t) - \dot{\widehat{Z}}(t)\\ &= AZ(t) + W_\star \mathbf{S}(X(t)) + \varepsilon(t) - A\widehat{Z}(t) - \widehat{W}(t)\mathbf{S}(X(t))\\ &= A(Z(t) - \widehat{Z}(t)) + (W_\star - \widehat{W}(t))\mathbf{S}(X(t)) + \varepsilon(t)\\ &= AE(t) + \widetilde{W}(t)\mathbf{S}(X(t)) + \varepsilon(t), \end{aligned} \tag{14}$$

where $\widetilde{W}(t) = W_\star - \widehat{W}(t)$.

5 Stable Learning Algorithm for MSNN

Based on Lyapunov stability theory, this section presents an online learning algorithm for MSNN. Recently, this approach has been used for training the rough extreme learning machines [3]. In the proof of Theorem 1, we need a special case of Barbalat's Lemma [16] that follows.

Lemma 1. *If $f,\ \dot{f} \in \boldsymbol{L}_\infty$, and $f \in \boldsymbol{L}_p$ for some $p \in [1,\infty)$, then $\lim_{t\to\infty} f(t) = 0$.*

Theorem 1. *Suppose that MSNN is used to identify the system (10) where its parameters are adjusted as follows:*

$$\dot{\widehat{W}}(t) \quad = E(t)\boldsymbol{S}(X(t))\Gamma^{-1}, \tag{15}$$

where the positive definite matrix Γ contains the learning rates. If

$$\|E(t)\| > \frac{\|\varepsilon(t)\|}{|\lambda_{min}(A)|}, \tag{16}$$

where $|\lambda_{min}(A)|$ shows the eigenvalue of A with minimum absolute value, then, the error $E(t)$ approaches zero, and the parameters $\widehat{W}(t)$ are bounded.

Proof. Let

$$v(t) \quad = \quad \frac{1}{2}E(t)^T E(t) + \frac{1}{2}\mathrm{tr}\left(\widetilde{W}(t)\Gamma\widetilde{W}(t)^T\right). \tag{17}$$

Then, according to (14), we have

$$\begin{aligned}\dot{v}(t) &= E(t)^T\dot{E}(t) + \mathrm{tr}\left(\dot{\widetilde{W}}(t)\Gamma\widetilde{W}(t)^T\right)\\ &= E(t)^T AE(t) + E(t)^T\widetilde{W}(t)\mathbf{S}(X(t)) + E(t)^T\varepsilon(t) + \mathrm{tr}\left(\dot{\widetilde{W}}(t)\Gamma\widetilde{W}(t)^T\right)\\ &= E(t)^T AE(t) + E(t)^T\varepsilon + \mathrm{tr}\left(E(t)\left[\mathbf{S}(X(t))\right]^T\widetilde{W}(t)^T\right) + \mathrm{tr}\left(\dot{\widetilde{W}}(t)\Gamma\widetilde{W}(t)^T\right)\\ &= E(t)^T AE(t) + E(t)^T\varepsilon + \mathrm{tr}\left(E\left[\mathbf{S}(X(t))\right]^T\widetilde{W}(t)^T + \dot{\widetilde{W}}(t)\Gamma\widetilde{W}(t)^T\right). \end{aligned} \tag{18}$$

From the fact that $\widetilde{W}(t) = W_\star - \widehat{W}(t)$, we have $\dot{\widetilde{W}}(t) = -\dot{\widehat{W}}(t)$. Therefore

$$\begin{aligned}\dot{v}(t) &= E(t)^T AE(t) + E(t)^T \varepsilon(t) + \operatorname{tr}\left(E(t)\left[\mathbf{S}(X(t))\right]^T \widetilde{W}(t)^T - \dot{\widehat{W}}(t)\Gamma\widetilde{W}(t)^T\right) \\ &= E(t)^T AE(t) + E(t)^T \varepsilon(t). \end{aligned} \tag{19}$$

In (19), the last equality is the result of assumption (15). From the fact that A is a Hurwitz matrix, we have

$$\begin{aligned}\dot{v}(t) &= E(t)^T AE(t) + E(t)^T \varepsilon(t) \\ &\leq -\|E(t)\|^2 |\lambda_{\min}(A)| + \|E(t)\|\|\varepsilon(t)\|. \end{aligned} \tag{20}$$

The last expression is a polynomial of degree two in $\|E(t)\|$, and

$$\|E(t)\| = \frac{\|\varepsilon(t)\|}{|\lambda_{min}(A)|} \tag{21}$$

is a zero for this polynomial. According to the fact that the coefficient of $\|E(t)\|^2$ is negative, for

$$\|E(t)\| > \frac{\|\varepsilon(t)\|}{|\lambda_{min}(A)|} \tag{22}$$

we have

$$\dot{v}(t) \leq -\|E(t)\|^2 |\lambda_{\min}(A)| + \|E(t)\|\|\varepsilon(t)\| < 0. \tag{23}$$

Therefore, $\dot{v}(t) < 0$, $v(t)$ is a decreasing function, and $v(t) < v(0)$. According to the positive definiteness of $v(t)$, we have $0 < v(t) < v(0)$, $v(t) \in \mathbf{L}_\infty$, and hence, $E(t) \in \mathbf{L}_\infty$. Using these results, we conclude that

$$\begin{aligned} 0 &< \int_0^\infty \|E(t)\|^2 \lambda_{\min}(A) dt - \int_0^\infty \|E(t)\|\|\varepsilon(t)\| dt \\ &\leq -\int_0^\infty \dot{v}(t) dt = v(0) - v(\infty) < \infty. \end{aligned} \tag{24}$$

Thus, $E(t) \in \mathbf{L}_2$, and $E(t) \in \mathbf{L}_\infty \cap \mathbf{L}_2$. In addition, since the function f in system (10) is Lipschitz continuous, $\dot{E}(t) \in \mathbf{L}_\infty$. Then, using Lemma 1, we have $E(t) \to 0$ as $t \to \infty$, and according to (15), we conclude that the parameters in the matrix $\widehat{W}(t)$ are bounded. □

6 Identification of chaotic and hyperchaotic systems by MSNN

This section is devoted to identifying some well-known chaotic and hyperchaotic dynamic nonlinear systems using the proposed approach. The first chaotic system is the Van der Pol-Duffing oscillator. It has been used to model many physical systems in electrical mechanics and electronics. Then, the Lorenz chaotic system is identified that arises in the models of many natural systems, such as the lasers, DC motors, electric circuits, and chemical reactions. Finally, we apply the MSNN to identify a financial hyperchaotic system with four dimensions.

6.1 Van der Pol-Duffing oscillator

The Van der Pol oscillator is a nonlinear oscillator with self-excited oscillations, and the Duffing oscillator is a nonlinear oscillator with different dynamical behaviors, such as hardening and softening spring effects. The Van der Pol-Duffing oscillator combines these oscillators, and therefore, it can exhibit a wide range of dynamic behaviors. Van der Pol-Duffing oscillator has many applications in different contexts, such as physics, biology, signal processing, electronics, and control engineering [32]. This chaotic system is described with the equation

$$\frac{d^2x}{dt^2} - \alpha(1-x^2)\frac{dx}{dt} + \beta x + \gamma x^3 = \phi(\zeta,\omega,t),$$
$$x(0) = \kappa, \;\; x^{'}(0) = \tau.$$

We can write this oscillator as a system of two differential equations of order one:

$$\left\{ \begin{array}{lcl} \dot{z}_1 & = & z_2 \\ \dot{z}_2 & = & \alpha(1-z_1^2)z_2 - \beta z_1 - \gamma z_1^3 + \phi(\zeta,\omega,t) \end{array} \right. , \tag{25}$$

Here, we suppose that $\alpha = 0,\ \beta = 0.3,\ \gamma = 0.012,\ \kappa = 0.1,\ \tau = 0.1,\ \zeta = 0.2,\ \omega = 2$, and $\phi(\zeta,\omega,t) = \zeta\sin(\omega t)$.

Table 1: MSEs of neural models in the identification of Van der pol-Duffing oscillator (25).

Model	n_h	h	Para.	epochs	Γ_i	Testing MSE
MLP	45	-	315	1	$2000I_{45}$	0.0019
MLP	65	-	455	1	$2000I_{65}$	0.0016
MLP	45	-	315	50	$2000I_{45}$	4.12e-5
SNN	-	0.2	42	100	$200I_{21}$	0.1427
SNN	-	0.2	42	1000	$200I_{21}$	0.0630
MSNN	-	0.2	52	1	$50I_{26}$	2.75e-5
MSNN	-	0.1	92	1	$50I_{46}$	1.17e-5
MSNN	-	0.05	172	1	$50I_{86}$	1.85e-5
MSNN	-	0.1	92	50	$50I_{46}$	8.11e-6

Van der Pol-Duffing oscillator is identified by MSNN, SNN, and multilayer perceptron (MLP). In MLP, the trainable parameters are initiated with random numbers in the interval $[-0.5, 0.5]$. In SSN and MSNN, the trainable parameters are initiated with random numbers in the interval $[-0.05, 0.05]$. The input vector of neural identifiers is

$$X = [z_1(k-1), z_2(k-1), z_1(k-2), z_2(k-2)]^T.$$

The matrix A in the linear part of the models (Equation (13)) is empirically chosen as

$$A = \left[\begin{array}{cc} -6 & 0 \\ 0 & -42 \end{array} \right], \tag{26}$$

and the discretization step dt for data generation is chosen as $dt = 0.01$. The other hyper-parameters and the MSEs of neural models are presented in Table 1. Figure 3 shows the simulation results in identifying the Van der pol-Duffing oscillator with MSNN. Figure 4 shows the phase plane in identifying the Van der pol-Duffing oscillator with MSNN. From the simulation results, we conclude that in identifying the Van der pol-Duffing oscillator, the performance of MSNN is better than the SNN and MLP.

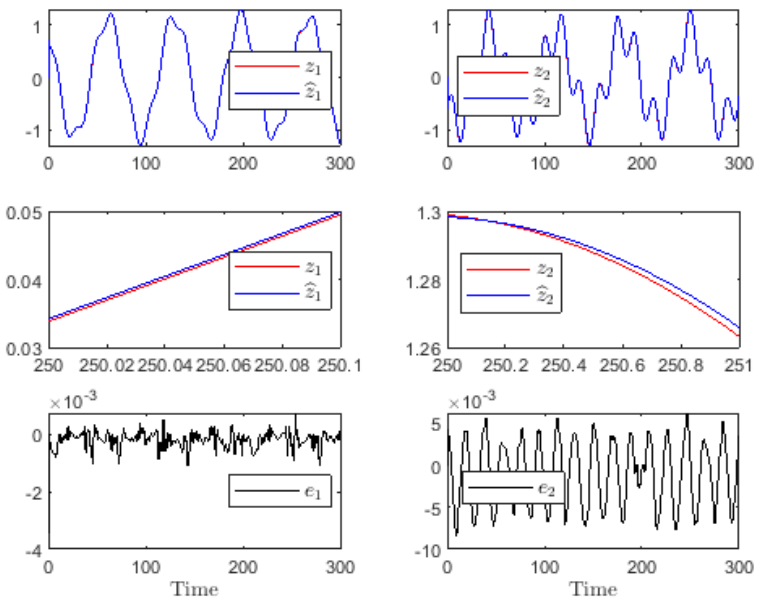

Figure 3: Simulation results in identifying the Van der pol-Duffing oscillator (25) with MSNN ($h = 0.1$ and 50 epochs).

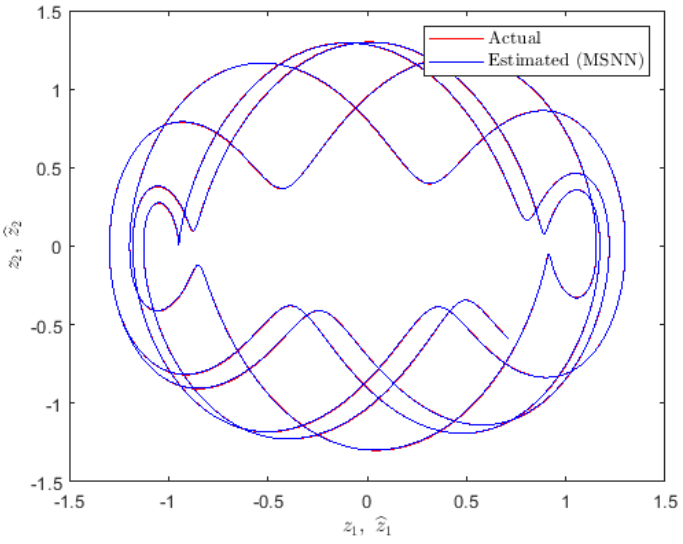

Figure 4: Phase plane of Van der pol-Duffing oscillator (25) in the identification with MSNN ($h = 0.1$ and 50 epochs).

6.2 Lorenz System

Table 2: MSEs of neural models in the identification of Lorenz system (27).

Model	n_h	h	Para.	Γ_i	Testing MSE
MLP	60	-	600	$100I_{60}$	1.55e-4
MLP	100	-	1000	$100I_{100}$	1.41e-4
SNN	-	0.2	93	I_{31}	6.71e-4
SNN	-	0.1	183	I_{61}	3.05e-4
SNN	-	0.05	363	I_{121}	9.27e-5
SNN	-	0.025	723	I_{241}	8.91e-5
MSNN	-	0.2	111	I_{37}	5.00e-5
MSNN	-	0.1	201	I_{67}	4.78e-5
MSNN	-	0.05	381	I_{127}	2.10e-5
MSNN	-	0.025	741	I_{247}	7.71e-6

The Lorenz system is an attractive and important example of a chaotic system. It was first studied by Lorenz in 1963. In the literature, the Lorenz system has been used to model weather and climate, lasers, chemical reactions, etc. The governing equations of the Lorentz system are

$$\begin{cases} \dot{z}_1 &= \sigma(z_2 - z_1), \\ \dot{z}_2 &= rz_1 - z_2 - z_1 z_3, \\ \dot{z}_3 &= z_1 z_2 - bz_3, \end{cases} \tag{27}$$

where $\sigma,\ r,\ b > 0$ are system parameters. Here, we suppose that $z_1(0) = 7.5,\ z_2(0) = -9,\ z_3(0) = 28$, and $\sigma = 10,\ r = 28,\ b = 8/3$.

The Lorenz system is identified by MSNN, SNN, and MLP. In MLP, the trainable parameters are initiated with random numbers in the interval $[-2,2]$. In SNN and MSNN, the trainable parameters are initiated with random numbers in the interval $[-0.05, 0.05]$. The input vector of neural identifiers is

$$X \quad = \quad [z_1(k-1), z_2(k-1), z_3(k-1), z_1(k-2), z_2(k-2), z_3(k-2)]^T.$$

The matrix A in the linear part of the models (Equation (13)) is empirically chosen as

$$A = \begin{bmatrix} -100 & 0 & 0 \\ 0 & -100 & 0 \\ 0 & 0 & -100 \end{bmatrix}, \tag{28}$$

and the discretization step dt for data generation is chosen as $dt = 0.01$. The other hyper-parameters and the MSEs of neural models in identifying the Lorenz chaotic system are presented in Table 2. Figure 5 shows the simulation results in identifying the Lorenz chaotic system with MSNN. Figure 6 shows the phase plane in identifying the Lorenz chaotic system with MSNN. From the simulation results, we conclude that in identifying the Lorenz chaotic system, the performance of MSNN is better than the SNN and MLP.

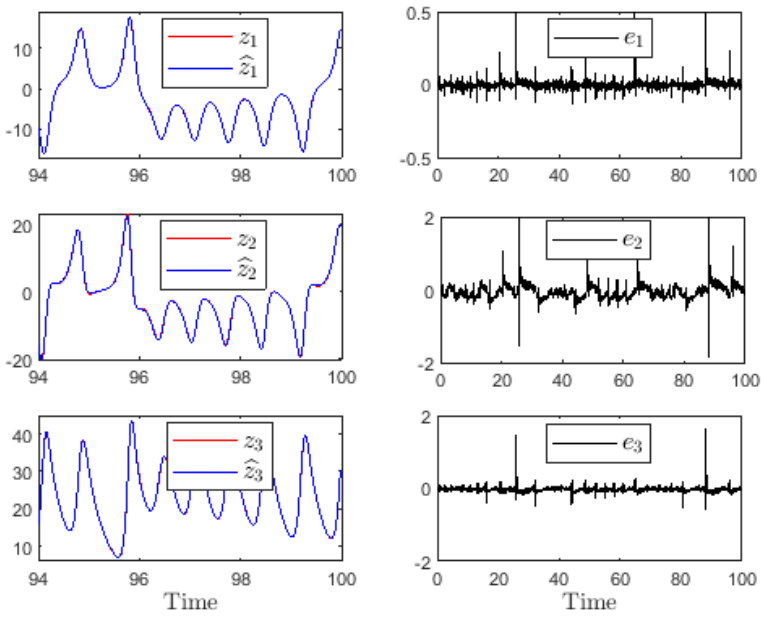

Figure 5: Simulation results in identifying the Lorenz system (27) with MSNN.

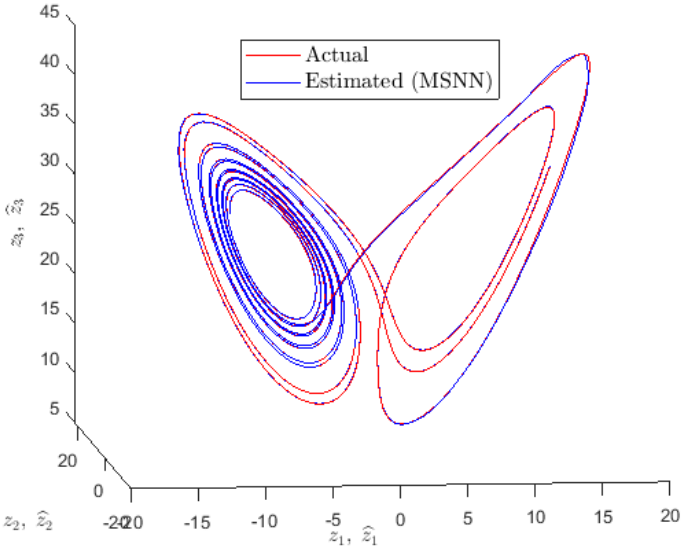

Figure 6: Phase plane of Lorenz system (27) in the identification with MSNN.

Table 3: MSEs of neural models in the identifying of financial hyperchaotic system (29).

Model	n_h	h	Para.	Γ_i	Testing MSE
MLP	30	-	390	$100I_{30}$	7.55e-5
MLP	50	-	650	$100I_{50}$	2.55e-5
MLP	90	-	1170	$100I_{90}$	2.21e-5
SNN	-	0.2	41	$2I_{41}$	1.00e-5
SNN	-	0.1	81	$2I_{81}$	1.66e-6
SNN	-	0.05	161	$2I_{161}$	1.27e-6
SNN	-	0.025	321	$2I_{321}$	2.57e-7
MSNN	-	0.2	49	$2I_{49}$	4.28e-6
MSNN	-	0.1	89	$2I_{89}$	8.78e-7
MSNN	-	0.05	169	$2I_{169}$	3.50e-7
MSNN	-	0.025	329	$2I_{329}$	6.41e-8

6.3 Hyperchaotic financial dynamic system

The complex and unpredictable behavior of financial markets can be modeled by financial hyperchaotic systems. The field of hyperchaotic systems is an active area of research, and few works have been done in this context. An example of financial hyperchaotic systems is described as follows [7]:

$$\begin{cases} z_1(k+1) &= z_3(k)+(z_2(k)-\alpha)z1(k)+z_4(k), \\ z_2(k+1) &= 1-\beta z_2(k)-z_1(k)^2, \\ z_3(k+1) &= -z_1(k)-\gamma z_3(k), \\ z_4(k+1) &= -\zeta z_1(k)z_2(k)-\kappa z_4(k), \end{cases} \tag{29}$$

where α, β, γ, ζ, κ are system parameters. Here, we suppose that $\alpha = 0.8$, $\beta = 0.3$, $\gamma = 1.1$, $\zeta = 0.3$, $\kappa = 0.16$ and $z_1(0) = 1$, $z_2(0) = 2.5$, $z_3(0) = 0.5$, $z_4(0) = 0.5$.

This hyperchaotic financial dynamical system is identified by MSNN, SNN, and MLP. In MLP, the trainable parameters are initiated with random numbers in the interval $[-1,1]$. In SNN and MSNN, the trainable parameters are initiated with random numbers in the interval $[-0.05,0.05]$. The input vector of neural identifiers is

$$X = [z_1(k-1), z_2(k-1), z_3(k-1), z_4(k-1), z_1(k-2), z_2(k-2), z_3(k-2), z_4(k-2)]^T.$$

The matrix A in the linear part of the models (Equation (13)) is empirically chosen as

$$A = \begin{bmatrix} -100 & 0 & 0 & 0 \\ 0 & -100 & 0 & 0 \\ 0 & 0 & -150 & 0 \\ 0 & 0 & 0 & -150 \end{bmatrix}, \tag{30}$$

and the discretization step dt for data generation is chosen as $dt = 0.01$. The other hyper-parameters and the MSEs of neural models in identifying the financial system are presented in Table 3.

Figure 7 shows the simulation results in identifying the hyperchaotic financial system with MSNN. Figure 8 shows the phase plane in identifying the hyperchaotic financial system MSNN. From these figures, we conclude that in identifying the hyperchaotic financial system, the performance of MSNN is better than the other models.

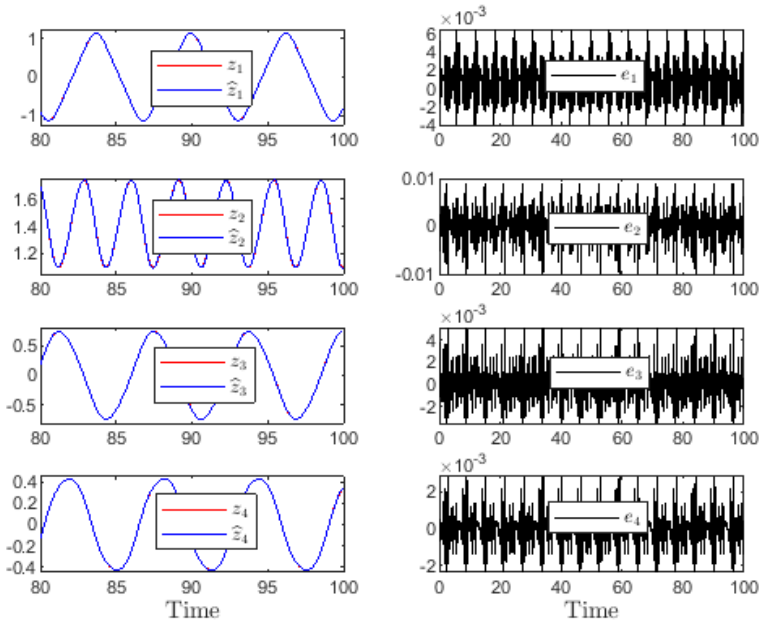

Figure 7: Simulation results in identifying the hyperchaotic financial dynamic system (29) with MSNN.

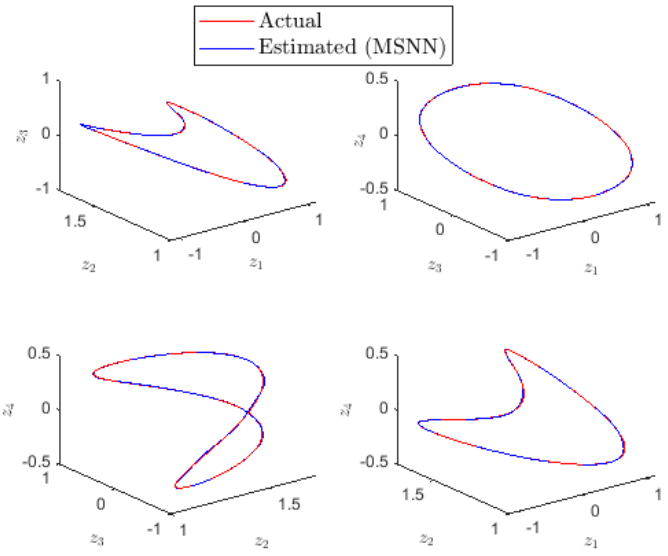

Figure 8: Phase plane of the hyperchaotic financial dynamic system (29) in the identification with MSNN.

7 Wind speed time series forecasting

Time series forecasting involves predicting future values based on the previously observed values. It is widely used in various fields such as finance, weather forecasting, supply chain management, and more. In recent years, many attempts have been made to use the capabilities of ANNs to forecast future values of time series. The temporal evolution of time series can be described using a state space model in dynamical systems [5]. The time series observations $\{x(t)\}_{t=0}^{\infty}$ can be transformed into the state vectors $\{\mathbf{Z}(t)\}_{t=0}^{\infty}$ where $\mathbf{Z}(t) \in \mathbf{R}^n$ shows the system states. The dynamics of these states can be defined as follows:

$$\dot{\mathbf{Z}}(t) = f(\mathbf{Z}(t)), \tag{31}$$

where f is an unknown function. The approximation of this system with MSNN is similar to the identification of (10) that is described in Section 4. Wind power as a renewable energy has the advantage

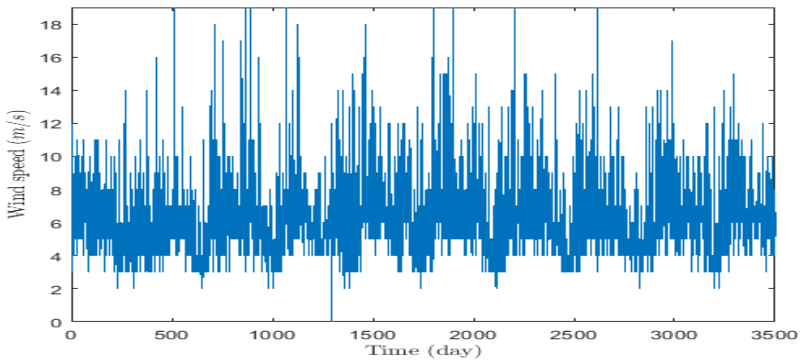

Figure 9: Original data of wind speed time series in Khorram Abad.

of less pollution in comparison with the other energy sources [6]. Therefore, increasing the usage of wind energy would decreases the environmental pollution arising from fossil fuels. Prediction of wind power is directly related to the accuracy of wind speed prediction. This problem has been in the scope of researchers in recent years. It has been shown that wind speed is a chaotic time series [13]. In the present work, we use the proposed model MSNN for wind speed prediction.

The process of wind speed forecasting using the MSNNs can be described through the following steps:

- **Data preparation:** First, we gather historical data relevant to our prediction. The utilized data for this work has been gathered from the meteorological stations in Khorram Abad city, in the west of Iran (Figure 9). To improve the network performance, we normalize the data to the range -1 to 1 and split the data into training and test sets. Additionally, we use a Butterworth filter ($\omega = 0.05$) to smooth the data (Figure 10).

- **Model selection:** In this work, MSNN is utilized as the neural model for wind speed forecasting. To demonstrate the efficiency of MSNN, we compare the results with MLP and SNN.

- **Model training:** The neural models are trained using an online learning algorithm based on the Lyapunov stability theory.
- **Evaluation and tuning:** To evaluate the model performance, MSE is used, and the hyperparameters are adjusted empirically to improve model performance.
- **Prediction:** Using the trained model, the future values of the wind speed time series are forecast.
- **Post-processing:** For normalized data, the model predictions need to be inverse transformed and interpreted, which is done in this work.

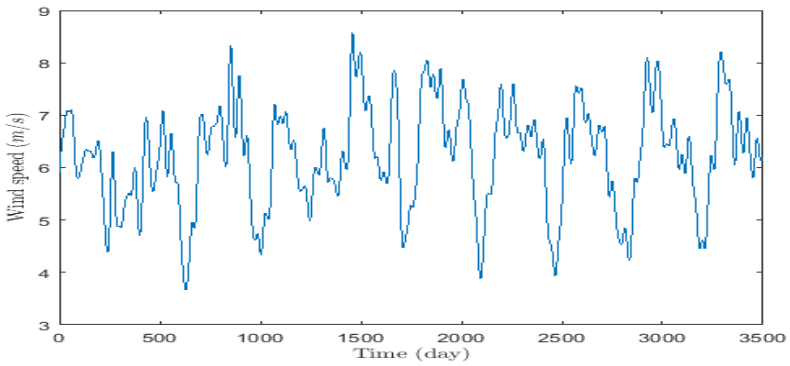

Figure 10: Wind speed time series in Khorram Abad after filtering.

Wind speed time series is forecast by MLP, SNN, and MSNN. In MLP, the trainable parameters are initiated with random numbers in the interval $[-0.3, 0.3]$. In SNN and MSNN, the trainable parameters are initiated with random numbers in the intervals $[-0.05, 0.05]$ and $[-0.005, 0.005]$, respectively. The input vector of neural identifiers is

$$X = [x(k-1), x(k-4), x(k-7), x(k-10), 1]^T.$$

The matrix A in the linear part of the models (Equation (13)) is empirically chosen as $A = [-40]$, and the discretization step dt in the implementation of models, is chosen as $dt = 0.04$. The other hyperparameters and the results in the wind speed forecasting with the neural models, is presented in Table 4. Figure 11 shows the results in wind speed forecasting with MSNN. Therefore, we can conclude that in wind speed forecasting, the performance of MSNN is better than the other models.

8 Conclusion

This work introduces a novel structure for Sinc neural networks (SNNs), called MSNN, designed for identifying nonlinear systems. The parameters of MSNN are adjusted using a Lyapunov-based online algorithm. The proposed model is applied to identify various chaotic and hyperchaotic dynamic systems. Additionally, the model is used to forecast chaotic wind speed time series, and the results are compared with other models. MSNN, designed based on the Sinc interpolation method, boasts a simple yet effective

Table 4: MSEs of neural models in the wind speed forecasting.

Model	n_h	h	Para.	Γ_i	Testing MSE
MLP	10	-	70	$\frac{100}{3}I_{10}$	0.0010
MLP	20	-	140	$\frac{100}{3}I_{20}$	0.0008
MLP	30	-	210	$\frac{100}{3}I_{30}$	0.0012
SNN	-	0.2	21	I_{21}	0.0031
SNN	-	0.1	41	I_{41}	0.0013
SNN	-	0.05	81	I_{81}	0.0041
SNN	-	0.025	161	I_{161}	0.0027
MSNN	-	0.2	26	$5I_{26}$	0.0005
MSNN	-	0.1	46	$5I_{46}$	0.0006
MSNN	-	0.05	86	$5I_{86}$	0.0005
MSNN	-	0.025	166	$5I_{166}$	0.0003

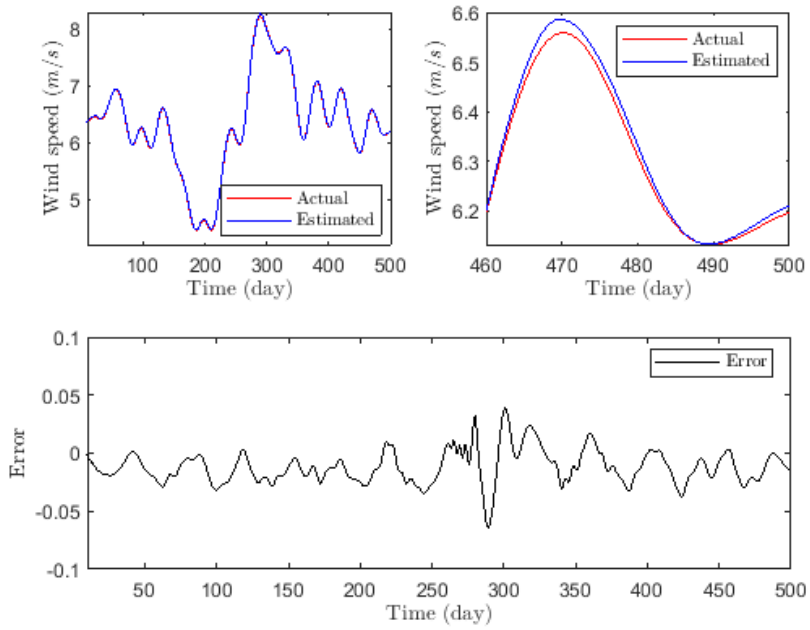

Figure 11: Implementation of MSNN with $h = 0.05$ in the forecasting of wind speed time series.

structure and demonstrates a strong capability in identifying nonlinear systems. Future research will focus on using the proposed model for the identification, prediction, and control of real-world chaotic and hyperchaotic systems. Studying these systems is valuable and exciting due to their presence in many critical natural systems.

References

[1] G. Ahmadi, *Stable rough extreme learning machines for the identification of uncertain continuous-time nonlinear systems*, Control Optim. Appl. Math. **4** (2019) 83–101.

[2] G. Ahmadi, *Stochastic gradient-based hyperbolic orthogonal neural networks for nonlinear dynamic systems identification*, J. Math. Model. **10** (2022) 529–547.

[3] G. Ahmadi, *Stable rough extreme learning machines for the identification of uncertain continuous-time nonlinear systems*, Control Optim. Appl. Math. **4** (2019) 83–101.

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

[5] G. Ahmadi, M. Dehghandar, *Chaotic time series prediction using rough-neural networks*, Math. Interdisc. Res. **8** (2023) 71–92.

[6] C. Ai, S. He, X. Fan, W. Wang, *Chaotic time series wind power prediction method based on OVMD-PE andimproved multi-objective state transition algorithm*, Energy **278** (2023) 127695.

[7] H.P.H. Anh, C.V. Kien, *Robust extreme learning machine neural approach for uncertain nonlinear hyper-chaotic system identification*, Int. J. Robust Nonlinear Control **31** (2021) 9127–9148.

[8] S. Biswas, C.O. Ayna, S.Z. Gurbuz, A.C. Gurbuz, *CV-SincNet: Learning complex Sinc filters from raw radar data for computationally efficient human motion recognition*, IEEE Trans. Radar Syst. **1** (2023) 493-504.

[9] D. Borra, S. Fantozzi, E. Magosso, *Interpretable and lightweight convolutional neural network for EEG decoding: Application to movement execution and imagination*, Neural Netw. **129** (2020) 55–74.

[10] A. Bria, C. Marrocco, F. Tortorella, *Sinc-based convolutional neural networks for EEG-BCI-based motor imagery classification*, In: A.D. Bimbo, et al. *Pattern Recognition. ICPR International Workshops and Challenges, ICPR Lecture Notes in Computer Science*, vol 12661, Springer, Cham, 2021.

[11] W. Elwasif, L.V. Fausett, *Function approximation using a Sinc neural network*, in Proceedings of the SPIE, 1996, pp. 690–701.

[12] M. Forgione, D. Piga, *Continuous-time system identification with neural networks: Model structures and fitting criteria*, Eur. J. Control **59** (2021) 69–81.

[13] Z. Guo, D. Chi, J. Wu, W. Zhang, A new wind speed forecasting strategy based on the chaotic time series modelling technique and the Apriori algorithm, Energy Convers. Manag. **84** (2014) 140–151.

[14] R. Heydari Dastjerdi, G. Ahmadi, *Designing the sinc neural networks to solve the fractional optimal control problem* Iran. J. Numer. Anal. Optim., doi: 10.22067/ijnao.2024.86494.1380, In press.

[15] C. Huang, *Identification of continuous-time systems with irregular samplings by invariant-subspace based method*, IEEE Trans. Autom. Control. **69** (2024) 2406–2413.

[16] P. Ioannou, J. Sun, *Robust Adaptive Control*, Prentice Hall, New Jersey, 1996.

[17] J.J. Jui, M.A. Ahmad, *A hybrid metaheuristic algorithm for identification of continuous-time Hammerstein systems*, Appl. Math. Model. **95** (2021) 339–360.

[18] K. Liu, M. Yang, X. Xing, Z. Yu, W. Wu, SincMSNet: A Sinc filter convolutional neural network for EEG motor imagery classification, J. Neural Eng. **20** (2023) 056024.

[19] E.N. Lorenz, *Deterministic nonperiodic flow*, J. Atmos. Sci. **20** (1963) 130–141.

[20] D.A. Magalln-Garca, L.J. Ontanon-Garcia, J.H. Garca-Lpez, G. Huerta-Cullar, C. Soubervielle-Montalvo, *Identification of chaotic dynamics in Jerky-based systems by recurrent wavelet first-order neural networks with a Morlet wavelet activation function*, Axioms **12** (2023) 200.

[21] T. Parcollet, M. Morchid, G. Linares, E2E-SINCNET: *Toward fully end-to-end speech recognition*, ICASSP, May 2020, Barcelone, Spain, hal-02484600.

[22] A.S. Poznyak, W. Yu, E.N. Sanchez, *Identification and control of unknown chaotic systems via dynamic neural networks*, IEEE Trans. Circuits Syst. I. Fundam. Theory Appl. **46** (1999) 1491–1495.

[23] K. Radha, M. Bansal, R.B. Pachori, *Automatic speaker and age identification of children from raw speech using sincNet over ERB scale*, Speech Commun. **159** (2024) 103069.

[24] G. Rao, H. Unbehauen, *Identification of continuous-time systems*, IEE Proc. Contr. Theor. Appl. **153** (2006) 185–220.

[25] X. Ren, A. Rad, P. Chan, W. Lo, *Identification and control of continuous-time nonlinear systems via dynamic neural networks*, IEEE Trans. Ind. Electron. **50** (2003) 478–486.

[26] J.J. Rubio, *Stable Kalman filter and neural network for the chaotic systems identification*, J. Frank. Inst. **354** (2017) 7444.

[27] A. Eftekhari, A. Saadatmandi, *DE Sinc-Collocation Method for Solving a Class of Second-Order Nonlinear BVPs*, Math. Interdiscip. Res. **6** (2021) 11–22.

[28] T. Schanze, Sinc interpolation of discrete periodic signals, IEEE Trans. Signal Process. **43** (1995) 1502–1503.

[29] F. Stenger, *Handbook of Sinc Numerical Methods*, CRC Press, Inc., USA, 2010.

[30] S. Vaidyanathan, C. Volos, *Advances and Applications in Chaotic Systems*, Springer International Publishing, Switzerland, 2016.

[31] J. Wang, X. Xing, *Identification of continuous-time dynamic systems with uncertainties measured by fuzzy sets subject to model structure errors*, IEEE Trans. Fuzzy Syst. **32** (2024) 3293–3300.

[32] K.-L. Yin, Y.-F. Pu, L. Lu, *Combination of fractional FLANN filters for solving the Van der Pol-Duffing oscillator*, Neurocomputing **399** (2020) 183–192.

[33] B. Zhang, S. Billings, *Identification of continuous-time nonlinear systems: The nonlinear difference equation with moving average noise (ndema) framework*, Mech. Syst. Signal Process. **60** (2015) 810–835.

[34] M. Zhong, Z. Yan, *Data-driven forward and inverse problems for chaotic and hyperchaotic dynamic systems based on two machine learning architectures*, Phys. D: Nonlinear Phenom. **446** (2023) 133656.