




Zeroing Neural Network Based on Neutrosophic Logic for Calculating Minimal-Norm Least-Squares Solutions to Time-Varying Linear Systems

Vasilios N. Katsikis^{1,2,3}  · Predrag S. Stanimirović^{4,5} · Spyridon D. Mourtas^{1,5} · Lin Xiao⁶ · Dragiša Stanujkić⁷ · Darjan Karabašević⁸

Accepted: 1 February 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

This paper presents a dynamic model based on neutrosophic numbers and a neutrosophic logic engine. The introduced neutrosophic logic/fuzzy adaptive Zeroing Neural Network dynamic is termed NSFZNN and represents an improvement over the traditional Zeroing Neural Network (ZNN) design. The model aims to calculate the matrix pseudo-inverse and the minimum-norm least-squares solutions of time-varying linear systems. The improvement of the proposed model emerges from the advantages of neutrosophic logic over fuzzy and intuitionistic fuzzy logic in solving complex problems associated with predictions, vagueness, uncertainty, and imprecision. We use neutrosophication, de-fuzzification, and de-neutrosophication instead of fuzzification and de-fuzzification exploited so far. The basic idea is based on the known advantages of neutrosophic systems compared to fuzzy systems. Simulation examples and engineering applications on localization problems and electrical networks are presented to test the efficiency and accuracy of the proposed dynamical system.

Keywords Time-varying problems · Moore–Penrose inverse · Zeroing neural network (ZNN) · Neutrosophic logic · Localization problem · Electrical network

1 Introduction and Preliminary Motivation

The problem addressed in this paper is to calculate the following time-varying (TV) matrix expression:

$$Y(t) := M^\dagger(t)L(t), \quad (1)$$

in which $M(t) \in \mathbb{R}^{m \times n}$ is an arbitrary TV matrix, $L(t) \in \mathbb{R}^{m \times k}$, $k \geq 1$ and $M^\dagger(t) \in \mathbb{R}^{n \times m}$ is the Moore–Penrose (MP) inverse (or pseudo-inverse) of $M(t)$.

The main problem in defining an appropriate ZNN design is the permanent imperfection, which is fundamentally caused by the absence of the best or unique value of the varying-gain

✉ Vasilios N. Katsikis
vaskatsikis@econ.uoa.gr

Extended author information available on the last page of the article

parameter involved in the ZNN design. The starting point of our discussion is an approach based on fuzzy logic in developing ZNN dynamics. This paper presents a neutrosophic logic/fuzzy adaptive ZNN model (termed as NSFZNN model) for calculating the best approximate solutions to TV problems (1). The basic principle in defining the scaling parameter was the fact that fuzzy logic (FL), intuitionistic fuzzy logic (IFL) and Neutrosophic logic (NL) appear as efficient tools for handling mathematical models with uncertainty, fuzziness, ambiguity, inaccuracy, incomplete certainty, inconsistency, and redundancy.

The current research is important because of two main principles: the generality and applicability of the problem (1), as well as known advantages of neutrosophic systems, compared to the fuzzy systems used so far. A more detailed description of the basic principles follows in the rest of this section.

1.1 Importance of Least Squares Solutions

The expressions $Y(t)$ of the general form (1) are intensely applicable in science and in numerous and various engineering applications. The most important cases of (1) are listed below.

- (i) In the case $L(t) = I$, the expression (1) becomes the MP inverse $M^\dagger(t)$, which coincides with the usual inverse $M^{-1}(t)$ in the case of a nonsingular $M(t)$. The MP inverse is widely used in various fields of science but also in practical applications, mainly in biology [7], power forecasting [23], robotics [20] as well as in image restoration [34]. For more information on the history and properties of the MP inverse, the reader is referred to [3, 4]. The MP inverse has been used in many ways to solve linear systems of vector and matrix equations. The expression $Y = M^\dagger L$ represents the unique best-approximate solution of the linear matrix equations (LME) $MY = L$, i.e., the unique solution of the minimal norm $\|M^\dagger L\|$ between least-squares solutions $\|MY - L\| \geq \|MM^\dagger L - L\|$ [25]. It is worth mentioning that the best-approximate solution $M^\dagger L$ has been used to reconstruct blurry digital images [34].
- (ii) For $M \in \mathbb{R}^{m \times n}$ and $L = b \in \mathbb{R}^m$, the MP inverse solution $M^\dagger b$ is the unique solution of minimum-norm among the least-squares solutions to $My = b$, i.e., $\|My - b\| \geq \|MM^\dagger b - b\|$ and $\|M^\dagger b\| < \|y\|$ for all $y \neq M^\dagger b$ satisfying $My = b$ [25].
- (iii) Consider the case $M = LA$, for $A \in \mathbb{R}^{m \times n}$ and $L \in \mathbb{R}^{s \times m}$. Then (1) produces the generalized inverse $Y = (LA)^\dagger L \in A\{2, 4\}$ which satisfies the matrix equations $YAY = Y$ and $(YA)^* = YA$. It is worth noting that $YAY = Y$ and $(YA)^* = YA$, respectively, are the MP equations 2 and 4, while $A\{2, 4\}$ is the set of generalized inverses that satisfy these two equations [3, 4].
- (iv) Many efficient algorithms have been developed for calculating the MP inverse, mainly based on iterations [12], singular value decomposition [38] or Greville's recursive method [49]. It is well recognized that the efficiency and applicability of standard numerical algorithms applied to input matrices constant throughout time are limited in the TV case. The zeroing neural network (ZNN) dynamical approach is an efficient tool for solving TV problems [35]. The ZNN algorithm was initially proposed for solving the TV matrix inversion problem [43]. In [44], the authors proposed and investigated various ZNN models for solving online the time-varying reciprocal problem. By generalizing the field of application, a more general ZNN design for calculating the TV MP inverse of a full-column or full-row rank matrix was proposed in [45]. Five complex ZFs and accordingly developed ZNNs for calculating the TV complex MP inverse were proposed in [19].

This fact is a clear motivation for using ZNN dynamical systems to calculate the expression (1).

- (v) The time-varying linear matrix equation (LME) $M(t)Y(t) = L(t) \in \mathbb{R}^{m \times m}$, where $M(t) \in \mathbb{R}^{n \times n}$ is nonsingular, $L(t) \in \mathbb{R}^{n \times m}$, and $Y(t) \in \mathbb{R}^{n \times m}$ is the unknown matrix of interest, was considered in [40]. In addition, the LME $M(t)Y(t) = L(t) \in \mathbb{R}^{m \times m}$, where $M(t) \in \mathbb{R}^{m \times n}$, $L(t) \in \mathbb{R}^{m \times k}$, and $Y(t) \in \mathbb{R}^{n \times k}$ is the unknown matrix of interest, was investigated in [16] in terms of the QR decomposition. It is important to mention that (1) provides the best approximate solution to $M(t)Y(t) = L(t)$ in the case of arbitrary matrix $M(t)$ and an arbitrary vector or matrix $L(t)$.
- (vi) Let $A(t) \in \mathbb{R}^{m \times n}$ be of constant rank r , $G(t) \in \mathbb{R}_s^{n \times m}$ be of a constant rank s such that $0 < s \leq r$, and let $F(t) \in \mathbb{R}^{n \times m}$ be arbitrary. In the case

$$M(t) = G(t)A(t) + \mu I, \quad \mu > 0, \quad L(t) = F(t), \tag{2}$$

the general problem (1) produces the following result:

$$Y(t) = (G(t)A(t) + \mu I)^{-1} F(t), \tag{3}$$

where the most popular choices of A , F , G and initiated limit values

$$\lim_{\mu \rightarrow 0} (G(t)A(t) + \mu I)^{-1} F(t) \tag{4}$$

were surveyed in [32, 33].

- (vii) Various ZNN models for solving an arbitrary system of linear equations $M(t)y(t) = b(t)$ were presented in [31].
- (viii) Numerous applications have employed ZNN dynamics to solve TV problems with global convergence [16–18, 39, 41, 47].

1.2 Neutrosophy’s Justification and Expectations in the ZNN Dynamics

The development of ZNN models enhanced by the advantages of fuzzy logic systems is a current trend in ZNN development. The term fuzzy logic system (FLS) is derived from the fuzzy set theory [42], and it can handle uncertainties, ambiguities, vagueness, and imprecision. An arbitrary FLS is based on fuzzy rules and linguistic rules. Due to such ability, FLSs have been studied in many research articles [36, 37] and have been applied to various scientific fields [10, 26]. As a consequence, recent research has focused on incorporating fuzzy control into ZNN design [6, 9, 14, 15, 17].

The basis for establishing fuzzy numbers and neutrosophic numbers is the universe of discourse \mathcal{U} and a subset $\mathcal{N} \subseteq \mathcal{U}$. The fuzzy set theory enables the use of the membership function $T_{\mathcal{N}}(x) \in [0, 1]$ [42]. Intuitionistic fuzzy set theory is based on the additional non-membership function $F_{\mathcal{N}}(x) \in [0, 1]$ which satisfies the restriction $0 \leq T_{\mathcal{N}}(x) + F_{\mathcal{N}}(x) \leq 1$ [2]. Smarandache in [28] extended the intuitionistic fuzzy set theory by introducing the indeterminacy-membership function. Consequently, in the neutrosophic set theory [28, 29], each element of \mathcal{N} is defined by three independent membership functions: the truth-membership function $T_{\mathcal{N}}(x)$, the indeterminacy-membership function $I_{\mathcal{N}}(x)$, and the falsity-membership function $F_{\mathcal{N}}(x)$ function. Entries of a neutrosophic set (NS) \mathcal{N} are neutrosophic numbers of the form $\mathcal{N} = \{ \langle x : T_{\mathcal{N}}(x), I_{\mathcal{N}}(x), F_{\mathcal{N}}(x) \rangle \mid x \in \mathcal{U} \}$, where $T_{\mathcal{N}}(x)$, $I_{\mathcal{N}}(x)$, and $F_{\mathcal{N}}(x)$ are the truth-membership function, indeterminacy-membership function, and falsity-membership function, respectively. Values of these functions are independent of each other and within $[0, 1]$. It is important to note that NSs have essential uses in

denoising, clustering, segmentation, classification, and numerous medical image-processing applications [5].

The main contributions of this research include the following highlights.

- This paper offers an improvement in the Zeroing Neural Network (ZNN) design based on the application of neutrosophic numbers (NSFZNN model).
- The application of the NSFZNN dynamics in computing the MP inverse $M^\dagger(t)$ and the best approximate solution $Y(t) = M^\dagger(t)L(t)$ of a large class of TV matrix or vector equations $M(t)Y(t) = L(t)$ is presented.
- The convergence characteristics of the NSFZNN dynamics are investigated.
- Various numerical experiments present a comparison of numerical results obtained using neutrosophy versus numerical results obtained using known fuzzy approaches.
- The NSFZNN model's application in the domain of engineering is explored.

Sections within this paper are scheduled in the following manner. Existing Fuzzy Neural Network models are surveyed in Sect. 2 to highlight the position of ongoing research. In line with the examination of well-known methods, the motivation of our work is described in the same section. The general FZNN model based on the neutrosophic set theory is described in Sect. 3. Application of the general model in solving a class of TV problems is presented in Sect. 4. Simulation experiments in solving matrix equations and applications in solving practical linear systems are presented in Sect. 5. Some concluding remarks are given in Sect. 6.

2 Survey of Existing Fuzzy Neural Network Models and Motivation

Following the description in [43], three global steps are recognizable during the definition of the traditional ZNN (TZNN) model. These three steps are described as follows:

Step ZNN1. Define the error function (or Zhang function, ZF), denoted by $F(t)$.

Step ZNN2. Calculate the time derivative $\dot{F}(t) = \frac{dF(t)}{dt}$.

Step ZNN3. Utilize $F(t)$ to force each element to converge to zero in accordance with the dynamic rule:

$$\dot{F}(t) = -\lambda(t)F(t), \quad (5)$$

where $\lambda(t) > 0$ is an appropriately defined varying scaling parameter which can adjust the convergence rate.

Nevertheless, many heterogeneous rules for determining gain parameters have been proposed and applied in developing ZNN dynamics. Initial research in this direction was done in [46] in which the time-variant parameter $\lambda(t) := \lambda + \lambda^t$ was proposed and used as an efficient tool for solving a TV linear matrix equation. With the aim to solve the time-variant overdetermined system of linear equations, Zhang et al. in [48] proposed a varying-parameter $\lambda(t) := \lambda e^t$, $\lambda > 0$, inside the ZNN.

The approach that determines proper values of the varying scaling parameter $\lambda(t)$ by utilizing appropriate fuzzy control systems has become an upward trend in the ZNN development. The intention in [47] was the usage of a proper fuzzy value ν with the aim to replace the standard acceleration value λ with the fuzzy parameter $\lambda(t) := \lambda + \nu$, wherein ν is defined by utilizing an appropriate fuzzy logic control system. Furthermore, the adaptive fuzzy recurrent neural network (AFRNN) proposed in [47] was successfully employed in the motion control problem.

Using a combination of ZNN and an appropriate fuzzy logic system (FLS), the authors in [14] originated the fuzzy-power ZNN (FPZNN) for solving the TV pseudo-inversion

problem of full-row or full-column rank input matrices. To be more precise, the acceleration parameter in [14] is set to $\lambda(t) := \lambda_1 + \lambda_2^\nu$, such that $\lambda_1 > 0$, $\lambda_2 > 1$, and the value ν is the output of a properly defined FLS. In [15], the authors included a fuzzy-gain acceleration in the ZNN dynamical system in the AFT-ZNN dynamics solving quadratic problems with the acceleration gain parameter $\lambda(t) := \lambda + \nu$, where $\lambda > 0$ is the classical acceleration, and $\nu > 0$ is an appropriate fuzzy control value. A recent result regarding the inclusion of fuzzy-gain acceleration in the ZNN dynamical system was proposed in [17] based on the dynamical flow

$$\dot{F}(t) = -\lambda^\nu F(t), \quad (6)$$

where $\lambda > 1$ is the classical acceleration, and ν is a proper exponent generated as a fuzzy number within an appropriate FLC.

Our main goal is to initiate a new trend in defining the fuzzy parameter ν . The approach is based on the usage of *neutrosophic set theory* and *neutrosophic logic* (NL). The leading idea for incorporating NL fuzziness into the ZNN evolution is to make decisions about values ν within an appropriate NL control (NLC), based on the Frobenius norm $\|F(t)\|_F$ error function.

Further predictions regarding the decision-making problems on gain parameters depend on $\|F(t)\|_F$ and are associated with uncertainties and predictions. Thus, it is reasonable to allow the use of fuzzy numbers and neutrosophic triples in making decisions on the fuzzy value ν and, accordingly, on the composite gain acceleration λ^ν . Our goal is to apply NL principles which are more general and closer to human thinking than FL. NLC is a better choice than FL and IFL in representing real-world data and their implementation for several reasons.

- (a) FL and IFL systems neglect the importance of indeterminacy. A fuzzy logic controller (FLC) is based on the membership and non-membership of a particular element to a particular set and does not consider the indeterminate nature of generated data. The importance of uncertainty is fully addressed in NLC-controlled systems. We expect that the experience in FL, IFL, and NLC, gained in many areas of science and applications, such as neutrosophic set/logic/probability/statistics/measure/integral, will stay valid in dynamic systems.
- (b) FL or IFL systems are further constrained by the fact that the sum of membership and non-membership values is limited to 1. Trustiness and falsity in NL systems are independent, whereas in IFL are dependent. More details are available in [30].
- (c) NL reasoning clearly distinguishes between the concepts of absolute truth and relative truth by assuming the existence of absolute truth with the assigned value 1^+ .
- (d) NL is applicable in situations of overlapping regions of fuzzy systems [1].
- (e) NL is known as a generalization of the IFL [28]. Consequently, using NL allows greater and more varied choices in numerical experiments and decision-making.
- (f) A hesitancy in an IFL specifies indeterminacy so that IFL-based systems are not efficient in handling inconsistent information. On the other hand, an NL system can handle both indeterminacy and inconsistency [22].

3 NSFZNN Model Based on Neutrosophic Set Theory

The proposed fuzzy ZNN model for solving (1) is based on the application of an appropriate NLC in defining $\lambda(t) := \lambda^\nu > 0$ and will be termed as NSFZNN. Neutrosophication aims to

solve the indeterminacy and selectivity of parameters involved in dynamical systems. Its goal is a transformation of crisp values into appropriate neutrosophic ordered triples. On the other hand, a unique approach to de-fuzzification and de-neutrosophication, i.e., transforming a neutrosophic number into a corresponding crisp number, is proposed. The basic stages in defining the NLC can be highlighted as follows.

- (1) *Neutrosophication.* Using three membership functions, NL-based fuzzification maps the input $\vartheta := \|F(t)\|_F$, where $F(t)$ is appropriately defined ZF and $\|F(t)\|_F$ is the Frobenius norm of $F(t)$. The values of the gain parameter λ^v are defined based on estimates and predictions based on the truth, the indeterminacy, and the falsity percentage. Since the final goal is $F(t) = \mathbf{0}$, it is suitable to use $\vartheta := \|F(t)\|_F$ as a measure in developed NLC. The neutrosophication exploited in this paper transforms the input set in the fuzzy input set $\tilde{\mathcal{J}}$ and the output set into the fuzzy output set in the neutrosophic format $\mathfrak{D} = \{T_{\mathcal{N}}, I_{\mathcal{N}}, F_{\mathcal{N}}\}$. According to the general NL, it will be appropriate to use three membership functions for generating an output set, corresponding to $T_{\mathcal{N}}(\vartheta)$, $I_{\mathcal{N}}(\vartheta)$, $F_{\mathcal{N}}(\vartheta)$. The truth-membership function is defined as the sigmoid membership function

$$T(\vartheta) = 1/(1 + e^{-c_1(\vartheta-c_2)}), \tag{7}$$

in which the parameter c_1 is responsible for its slope at the crossover point $\vartheta = c_2$. The falsity-membership function is defined as the Z-shaped membership function

$$F(\vartheta) = \begin{cases} 1, & \vartheta \leq c_1 \\ 1 - 2 \left(\frac{\vartheta-c_1}{c_2-c_1} \right)^2, & c_1 \leq \vartheta \leq \frac{c_1+c_2}{2} \\ 2 \left(\frac{\vartheta-c_2}{c_2-c_1} \right)^2, & \frac{c_1+c_2}{2} \leq \vartheta \leq c_2 \\ 0, & \vartheta \geq c_2 \end{cases}, \tag{8}$$

where the parameter c_1 signifies the shoulder of the function, and c_2 signifies its foot. The indeterminacy-membership function is the Gaussian membership function

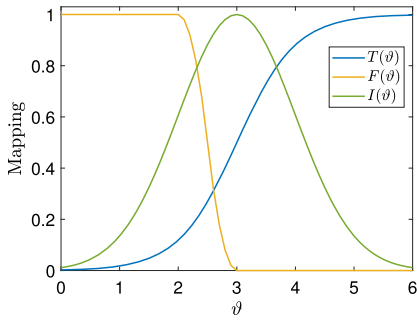
$$I(\vartheta) = e^{-\frac{(\vartheta-c_2)^2}{2\sigma^2}}, \tag{9}$$

where the parameter σ signifies the standard deviation, and the parameter c_2 signifies the mean. The neutrosophication of the crisp value $\vartheta \in \mathbb{R}$ is its transformation into $\langle \vartheta : T(\vartheta), I(\vartheta), F(\vartheta) \rangle$, where the membership functions are defined in (7), (8) and (9). Graphs of $T(\vartheta)$, $I(\vartheta)$, $F(\vartheta)$ with $c_1 = 2, c_2 = 3, \sigma = 1$ in 7, 8, 9 are presented in Fig. 1 a for $\vartheta \in [0, 6]$. Since $\theta \geq 0$, we will continue to use $c_2 = 0$ in further. Graphs of $T(\vartheta)$, $I(\vartheta)$, $F(\vartheta)$ for $c_1 = 1, c_2 = 0, \sigma = 0.9$ in 7, 8, 9 are presented in Fig. 1b for $\vartheta \in [0, 6]$. Graphs of $T(\vartheta) + I(\vartheta) + F(\vartheta)$ for $c_1 = 2, c_2 = 3, \sigma = 1$ are presented in Fig. 2 for $\vartheta \in [0, 6]$. Fulfillment of the condition $0 \leq T(\vartheta) + I(\vartheta) + F(\vartheta) \leq 3$ is evident and observable in this figure.

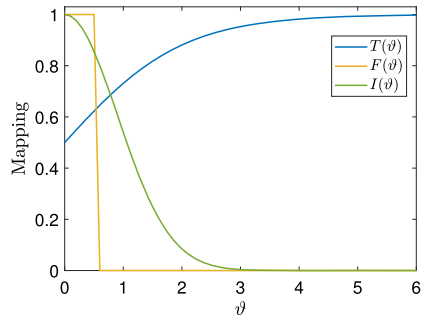
- (2) *Neutrosophic inference engine:* The neutrosophic rule between the fuzzy input set $\tilde{\mathcal{J}}$ and the fuzzy output set under the neutrosophic format $\mathfrak{D} = \{T_{\mathcal{N}}, I_{\mathcal{N}}, F_{\mathcal{N}}\}$ is described as the following ‘‘IF-THEN’’ rule:

$$R : \text{If } \tilde{\mathcal{J}} = HE \text{ then } \mathfrak{D} = \{T_{\mathcal{N}}, I_{\mathcal{N}}, F_{\mathcal{N}}\},$$

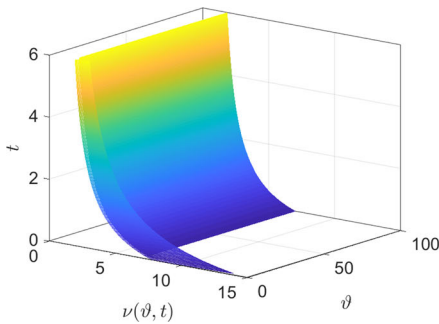
where HE represents a fuzzy set that indicates a significant error. The rule R is obtained as $\mathfrak{D} = \tilde{\mathcal{J}} \circ R$, where \circ means the fuzzy transformation symbol. Furthermore, it follows that $\kappa_{\tilde{\mathcal{J}} \circ R}(\zeta) = \kappa_{\tilde{\mathcal{J}} \circ R}$ and $\kappa_{\tilde{\mathcal{J}} \circ R} = \kappa_{\tilde{\mathcal{J}}} \wedge \kappa_{\mathfrak{D}}$, where \vee and \wedge are the maximum and



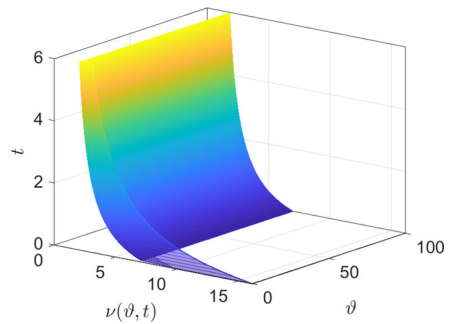
(a) $T(\vartheta), I(\vartheta), F(\vartheta)$ with $c_1 = 2, c_2 = 3, \sigma = 1$.



(b) $T(\vartheta), I(\vartheta), F(\vartheta)$ with $c_1 = 1, c_2 = 0, \sigma = 0.9$.



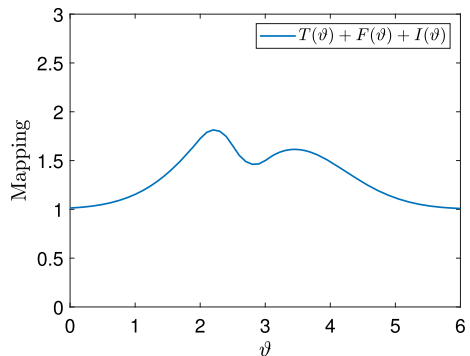
(c) $\nu(\vartheta, t)$ values using $T(\vartheta), I(\vartheta), F(\vartheta)$ of Fig. 1a.



(d) $\nu(\vartheta, t)$ values using $T(\vartheta), I(\vartheta), F(\vartheta)$ of Fig. 1b.

Fig. 1 Graphs of $T(\vartheta), I(\vartheta), F(\vartheta)$ under different parameters settings in 7, 8, 9 and their $\nu(\vartheta, t)$ values for $k_1 = 1, k_2 = 6$ and $\vartheta \in [0, 100], t \in [0, 6]$

Fig. 2 Graphs of $T(\vartheta) + I(\vartheta) + F(\vartheta)$ for $c_1 = 2, c_2 = 3, \sigma = 1$



minimum values operators, respectively. The following defuzzification method, called centroid, is employed to obtain the fuzzy vector $\zeta = [T_N, I_N, F_N]$:

$$\zeta = \frac{\int_{\mathcal{D}} \zeta \kappa_{\mathcal{J} \circ R}(\zeta) d\zeta}{\int_{\mathcal{D}} \kappa_{\mathcal{J} \circ R}(\zeta) d\zeta}. \tag{10}$$

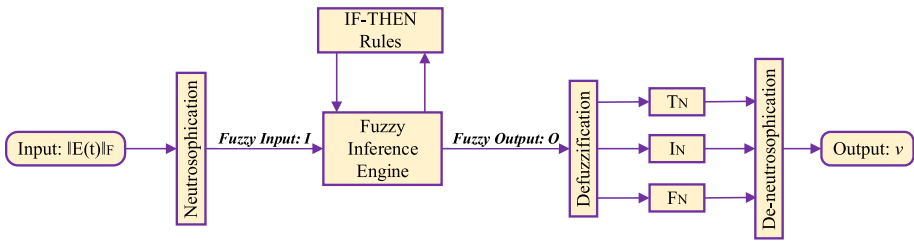


Fig. 3 The NLC structure

So, we will consider the dynamic neutrosophic set (DNS) of ordered triples $\mathcal{D} := \{(T(\vartheta), I(\vartheta), F(\vartheta)) | \vartheta \in \mathbb{R}_0^+\}$, where \mathbb{R}_0^+ denotes nonnegative real numbers. The definition of the set \mathcal{D} fulfils formal requirements of the DNS from [27], because $\vartheta := \|F(t)\|_F$ is the function of the time t .

- (3) *De-neutrosophication*. This step assumes the conversion $\langle \vartheta : T(\vartheta), I(\vartheta), F(\vartheta) \rangle \rightarrow v(\vartheta, t) \in \mathbb{R}$, resulting to a single (crisp) value $v(\vartheta, t)$. The following *de-neutrosophication rule* is proposed to obtain the parameter $v(\vartheta, t)$:

$$v(\vartheta, t) = k_1 + k_2 \frac{T(\vartheta) + I(\vartheta) + F(\vartheta)}{t + 1}. \tag{11}$$

The parameter $k_1 \geq 1$ in (11) indicates the lower limit of (11), the parameter $k_2 \geq 0$ is used to adjust the upper limit of (11), and $t \geq 0$ is a real number which satisfies $t \in \mathbb{R}_0^+$. The process of obtaining initiated output v is illustrated in Fig. 3.

Based on the general ZNN strategy, our imperative requirement is $v(\vartheta, t) \geq 1$. The results of Lemma 1 define criteria for fulfilling that requirement.

Lemma 1 *Under the constraints*

$$k_1 \geq 1, \quad k_2 \geq 0, \quad t \in \mathbb{R}_0^+, \tag{12}$$

the de-neutrosophy rule (11) satisfies

$$k_1 + \frac{3k_2}{t + 1} \geq v(\vartheta, t) \geq k_1 \geq 1. \tag{13}$$

Proof Since $T(\vartheta), I(\vartheta), F(\vartheta) \in [0, 1]$ in (7), (8), (9), it follows that $3 \geq T(\vartheta) + I(\vartheta) + F(\vartheta) \geq 0$. Furthermore, based on the assumptions in (12), we conclude for arbitrary $t \geq 0$

$$v(\vartheta, t) = \begin{cases} k_1 + k_2 (T(\vartheta) + I(\vartheta) + F(\vartheta)) \in [k_1, k_1 + 3k_2], & t = 0 \\ k_1 + k_2 \frac{T(\vartheta) + I(\vartheta) + F(\vartheta)}{t + 1} \in [k_1, k_1 + \frac{3k_2}{t + 1}], & t > 0. \end{cases} \tag{14}$$

In addition, as $t \rightarrow \infty$ it follows

$$\lim_{t \rightarrow \infty} v(\vartheta, t) = \lim_{t \rightarrow \infty} \left(k_1 + k_2 \frac{T(\vartheta) + I(\vartheta) + F(\vartheta)}{t + 1} \right) = k_1, \tag{15}$$

which completes the proof. □

In each time instant t defined by the NSFZNN evolution, $v(\vartheta, t)$ takes one unique value, depending on the value ϑ . Values $v(\vartheta, t)$ for $T(\vartheta), I(\vartheta), F(\vartheta)$ defined with $c_1 = 2, c_2 = 3, \sigma = 1$ in (7), (8), (9) and $k_1 = 1, k_2 = 6, \vartheta \in [0, 100], t \in [0, 6]$ are presented in Fig. 1c,

while values $\nu(\vartheta, t)$ for $T(\vartheta)$, $I(\vartheta)$, $F(\vartheta)$ with $c_1 = 1, c_2 = 0, \sigma = 0.9$ in (7), (8), (9) and $k_1 = 1, k_2 = 6, \vartheta \in [0, 100], t \in [0, 6]$ are presented in Fig. 1d.

The following generic dynamics will represent both the ZNN and NSFZNN dynamic systems:

$$\dot{F}(t) = -\lambda^{\nu(\vartheta, t)} F(t), \tag{16}$$

where $\nu(\vartheta, t)$ is defined in (11). Notice that the dynamics (16) will be denoted by NSFZNN. Moreover, the general NSFZNN dynamics is described by Algorithm 1.

Algorithm 1 General NSFZNN dynamics.

- 1: **Input:** Appropriate input matrices, the time interval $[0, T]$, and the parameters $c_1, c_2, \sigma, k_1, k_2$.
 - 2: Define the error function $F(t)$.
 - 3: Set $t := 0$.
 - 4: **while** $t \leq T$ **do**
 - 5: Compute $\vartheta := \|F(t)\|_F$.
 - 6: Compute $T(\vartheta), I(\vartheta), F(\vartheta)$ based on the neutrosophic inference engine.
 - 7: Compute $\nu(\vartheta, t)$ using (11).
 - 8: Solve (16) in expanded form and find new time instant t .
 - 9: **end while**
 - 10: **return** The variable state matrix $X(t)$ included in $F(t)$.
-

4 Problem and Model Description

This section describes the NSFZNN model. To calculate $Y(t) = M^\dagger(t)L(t)$, we follow the process described in Sect. 2 using NSFZNN of dynamics (16) or TZNN of design (5). As a result, the evolution of NSFZNN dynamics is developed in the following three steps.

Step NSZNN1. Define the ZF denoted as $F_1(t)$, which forces the state variables matrix $X(t)$ on the convergence $X(t) \mapsto M^\dagger(t)$, and the ZF $F_2(t)$, which forces the convergence $Y(t) - X(t)L(t) \mapsto \mathbf{0}$.

Given TV arbitrary matrix $M(t) \in \mathbb{R}^{m \times n}$; the ZF $F_1(t)$ which forces the convergence $X(t) \mapsto M^\dagger(t) \in \mathbb{R}^{n \times m}$ in the case $\text{rank}(M(t)) = \min\{m, n\}$ is defined by [11]

$$F_1(t) = \begin{cases} F_\alpha(t) \equiv M^T(t)M(t)X(t) - M^T(t), & m \geq n \\ F_\beta(t) \equiv X(t)M(t)M^T(t) - M^T(t), & m < n. \end{cases} \tag{17}$$

Then, for an arbitrary TV matrix $Y(t) \in \mathbb{R}^{m \times k}$, the ZF for finding $Y(t) := M^\dagger(t)L(t) \in \mathbb{R}^{n \times k}$ is defined as

$$F_2(t) = Y(t) - X(t)L(t) \in \mathbb{R}^{m \times k}. \tag{18}$$

Zeroing the ZFs in (17) and (18) at the same time, we are able to find both $X(t) = M^\dagger(t)$ and $Y(t) = M^\dagger(t)L(t)$.

Step NSZNN2. Compute the time derivatives $\dot{F}_1(t)$ and $\dot{F}_2(t)$. The time-derivative of $F_1(t)$ is

$$\dot{F}_1(t) = \begin{cases} \dot{F}_\alpha(t), & m \geq n \\ \dot{F}_\beta(t), & m < n \end{cases} \tag{19}$$

where

$$\dot{F}_\alpha(t) = M^T(t)M(t)\dot{X}(t) + H_\alpha(t), \tag{20}$$

with

$$H_\alpha(t) = \dot{M}^T(t)M(t)X(t) + M^T(t)\dot{M}(t)X(t) - \dot{M}^T(t), \tag{21}$$

and

$$\dot{F}_\beta(t) = \dot{X}(t)M(t)M^T(t) + H_\beta(t) \tag{22}$$

with

$$H_\beta(t) = X(t)\dot{M}(t)M^T(t) + X(t)M(t)\dot{M}^T(t) - \dot{M}^T(t). \tag{23}$$

The time-derivative of $F_2(t)$ is equal to

$$\dot{F}_2(t) = \dot{Y}(t) - \dot{X}(t)L(t) - X(t)\dot{L}(t). \tag{24}$$

Step NSZNN3. According to steps NSZNN1 and NSZNN2, define NSFZNN dynamic system for computing $M^\dagger(t)L(t)$, with the dynamic flow which is based on $Y(t) - X(t)L(t) \mapsto \mathbf{0}$, where $X(t) \mapsto M^\dagger(t)$.

Replacement of $F_1(t)$ and $\dot{F}_1(t)$ into the NSFZNN of (16) implies

$$\begin{cases} \dot{F}_\alpha(t) = -\lambda^{v(\vartheta,t)}F_\alpha(t), & m \geq n \\ \dot{F}_\beta(t) = -\lambda^{v(\vartheta,t)}F_\beta(t), & m < n \end{cases} \tag{25}$$

or equivalently

$$\begin{cases} M^T(t)M(t)\dot{X}(t) = -\lambda^{v(\vartheta,t)}F_\alpha(t) - H_\alpha(t), & m \geq n \\ \dot{X}(t)M(t)M^T(t) = -\lambda^{v(\vartheta,t)}F_\beta(t) - H_\beta(t), & m < n. \end{cases} \tag{26}$$

Applying the Kronecker product \otimes in conjunction with the vectorization operator $\text{vec}()$, the dynamic flow (26) is transformed into the equivalent vector form

$$\begin{cases} (I_m \otimes M^T(t)M(t)) \dot{\mathbf{x}}(t) = \text{vec}(-\lambda^{v(\vartheta,t)}F_\alpha(t) - H_\alpha(t)), & m \geq n \\ (M(t)M^T(t) \otimes I_n) \dot{\mathbf{x}}(t) = \text{vec}(-\lambda^{v(\vartheta,t)}F_\beta(t) - H_\beta(t)), & m < n, \end{cases} \tag{27}$$

where I_p signifies the $p \times p$ identity matrix and $\dot{\mathbf{x}}(t) = \text{vec}(\dot{X}(t))$.

It is known that the invertibility of the coefficient matrices of $\dot{\mathbf{x}}(t)$ in the left hand sides of (27) is sufficient to find $\dot{\mathbf{x}}(t)$. Assuming that $M(t)$ fulfills $\text{rank}(M(t)) < \min\{m, n\}$, the Tikhonov regularization should be applied on the results obtained in [21]. As a result, using the mass matrix

$$K(t) = \begin{cases} I_m \otimes M^T(t)M(t), & \text{rank}(M(t)) = n \leq m \\ M(t)M^T(t) \otimes I_n, & \text{rank}(M(t)) = m < n \\ I_m \otimes M^T(t)M(t) + \delta I_{mn}, & \text{rank}(M(t)) < n \leq m \\ M(t)M^T(t) \otimes I_n + \delta I_{mn}, & \text{rank}(M(t)) < m < n \end{cases} \tag{28}$$

with $\delta > 0$ signifies the ridge parameter, and gives

$$\mathbf{h}(t) = \begin{cases} \text{vec}(-\lambda^{v(\vartheta,t)}F_\alpha(t) - H_\alpha(t)), & m \geq n \\ \text{vec}(-\lambda^{v(\vartheta,t)}F_\beta(t) - H_\beta(t)), & m < n. \end{cases} \tag{29}$$

In this way, (27) can be modified to satisfy both of its cases for an arbitrary TV matrix $M(t)$, as follows:

$$K(t)\dot{\mathbf{x}}(t) = \mathbf{h}(t). \tag{30}$$

Further, replacing $F_2(t)$ and $\dot{F}_2(t)$ in NSFZNN (16), it is concluded

$$\dot{F}_2(t) = -\lambda^{v(\vartheta,t)}F_2(t) \tag{31}$$

or equivalently

$$\dot{Y}(t) - \dot{X}(t)L(t) = -\lambda^{v(\theta,t)}F_2(t) + X(t)\dot{L}(t). \tag{32}$$

Applying the Kronecker product in common with the vectorization, the evolution (32) becomes the flow

$$\dot{\mathbf{y}}(t) - (L^T(t) \otimes I_n) \dot{\mathbf{x}}(t) = \mathbf{z}(t), \tag{33}$$

where $\dot{\mathbf{y}}(t) = \text{vec}(\dot{Y}(t))$ and

$$\mathbf{z}(t) = \text{vec}\left(-\lambda^{v(\theta,t)}F_2(t) + X(t)\dot{L}(t)\right). \tag{34}$$

A combination of (30) and (33) leads to the dynamics

$$\begin{matrix} K(t) & \mathbf{0}_{mn \times nk} \\ -(L^T(t) \otimes I_n) & I_{nk} \end{matrix} \cdot \begin{matrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{y}}(t) \end{matrix} = \begin{matrix} \mathbf{h}(t) \\ \mathbf{z}(t) \end{matrix}, \tag{35}$$

where $\mathbf{0}_{mn \times nk}$ denotes the $mn \times nk$ zero matrix. It is worth noting that (35) is the proposed NSFZNN model for calculating the expression (1). Also keep in mind that (35) is solved using the ordinary differential equation (ode) *Matlab* solver.

Note that the invertibility of $K(t)$ is a prerequisite for the applicability of (30). The invertibility of $K(t)$ is verified in Theorem 1.

Theorem 1 *Let $M(t) \in \mathbb{R}^{m \times n}$ is an arbitrary TV matrix, $L(t) \in \mathbb{R}^{m \times k}$, $k \geq 1$ be given.*

(a) *The mass matrix $K(t)$ required in (30) and defined in (28), is nonsingular for $\delta > 0$ with the inverse*

$$K^{-1}(t) = \begin{cases} I_m \otimes (M^T(t)M(t))^{-1}, & \text{rank}(M(t))=n \leq m \\ (M(t)M^T(t))^{-1} \otimes I_n, & \text{rank}(M(t))=m < n \\ I_m \otimes (M^T(t)M(t) + \delta I_n)^{-1}, & \text{rank}(M(t)) < n \leq m \\ (M(t)M^T(t) + \delta I_m)^{-1} \otimes I_n, & \text{rank}(M(t)) < m < n. \end{cases} \tag{36}$$

(b) *Furthermore, the following limits hold:*

$$\begin{aligned} \lim_{\delta \rightarrow 0} \mathbf{x}(t) &= \text{vec}(M^\dagger(t)) \\ \lim_{\delta \rightarrow 0} \mathbf{y}(t) &= \text{vec}(M^\dagger(t)L(t)). \end{aligned} \tag{37}$$

Proof (a) The inverse $K^{-1}(t)$ as in (36) is obtained by applying the main properties of the Kronecker product. The matrix $M^T(t)M(t)$ is invertible in the case $\text{rank}(M(t)) = n$, which implies the invertibility of $K(t) = I_m \otimes M^T(t)M(t)$ in this case. On the other hand, $M(t)M^T(t)$ is invertible if $\text{rank}(M(t)) = m$, which implies that $M(t)M^T(t) \otimes I_n$ is invertible in this case. In the rank-deficient environment $\text{rank}(M(t)) < \min\{m, n\}$, the inverse $K^{-1}(t)$ exists on the basis of the assumption $\delta > 0$.

(b) On the basis of results derived in [11, 45] it is concluded $X(t) \rightarrow M^\dagger(t)$, which in common with $\dot{\mathbf{x}}(t) = \text{vec}(\dot{X}(t))$ implies

$$\lim_{\delta \rightarrow 0} \mathbf{x}(t) = \lim_{\delta \rightarrow 0} \text{vec}(X(t)) = \text{vec}(M^\dagger(t)), \tag{38}$$

which coincides with the first statement in (37). Finally, the ZNN dynamics (31) implies $F_2(t) \rightarrow 0$, which implies $Y(t) = X(t)L(t) \rightarrow M^\dagger(t)L(t)$, which in view of $\dot{\mathbf{y}}(t) = \text{vec}(\dot{Y}(t))$ leads to

$$\lim_{\delta \rightarrow 0} \mathbf{y}(t) = \lim_{\delta \rightarrow 0} \text{vec}(Y(t)) = \text{vec}(M^\dagger(t)L(t)). \quad (39)$$

So, the second statement in (37) is confirmed.

The NSFZNN model, compared to the FZNN models for calculating the MP inverse, has a unique design with the following novelties:

- NSFZNN simultaneously calculates the MP inverse $M^\dagger(t)$ and the minimal-norm least-squares solution $Y(t) = M^\dagger(t)L(t)$ for the TV linear systems $M(t)Y(t) = L(t)$.
- Based on vectorization and regularization, NSFZNN accepts an arbitrary real TV matrix $M(t)$, which makes it universally applicable. In this way, we overcome one of the ZNN approaches' drawbacks in computing the MP inverse, which assumes the invertibility of the mass matrix and causes some restrictions on the input matrix.
- NSFZNN uses an adaptive NLC based on three membership functions to increase the convergence speed of the FZNN design. The NLC, based on three membership functions, is capable of handling problems in which indeterminacy is present, allowing it to calculate the theoretical solution faster.
- Moreover, numerical comparison of NSFZNN dynamics against FZNN dynamics and experience gained in two applications have proven that NLC is credible in creating dynamic systems.

5 Simulative Experiments and Applications

This section presents two examples and two engineering applications designed to test the efficiency and accuracy of the NSFZNN model (35). The applications include solving the Angle of Arrival (AoA) localization and computing the dynamic alternating current (AC) of an electrical network. Inhere, the TZNN in (5) and the FZNN dynamic from [14] are the designs compared against the proposed NSFZNN model (35). Note that all the designs are employed under the linear activation function. In all examples and the applications, the TZNN, FZNN, and NSFZNN designs are evaluated with the gain value $\lambda = 10$, inside the time interval $[0, 10]$, which assumes the start time $t_0 = 0$ and the end time $t_f = 10$. Initial states that are used in all the designs for producing their solutions are zero matrices for $X(0)$ and identity matrices for $Y(0)$, of appropriate dimensions, while the `ode15s` MATLAB solver is being used. The FZNN parameters are set as proposed in [14], while the NSFZNN parameter values are set as presented in Table 1. Notice that the notations " X_{TZNN} ", " X_{FZNN} ", " X_{NSFZNN} " and " Y_{TZNN} ", " Y_{FZNN} ", " Y_{NSFZNN} " involved in the figures' legends correspond to the solutions to X and Y generated by TZNN, FZNN, and NSFZNN, respectively.

Table 1 NLC’s parameters

Module	Function	Range	Neutrosophic set	Parameters	Weight	
Neutrosophic	Input	Sigmoid	[0,10]	HE	$c_1 = 0.6, c_2 = 0$	1
Inference		Sigmoid	[0,1]	$T_{\mathcal{N}}$	$c_1 = 5.4, c_2 = 0.5$	1
Engine	Output	Gaussian	[0,1]	$I_{\mathcal{N}}$	$c_2 = 0.2, \sigma = 0.5$	1
		Z-shaped	[0,1]	$F_{\mathcal{N}}$	$c_1 = -0.5, c_2 = 1.5$	1
De-neutrosophication	(9)	–	–		$k_1 = 1, k_2 = 6$	–

5.1 Example 1

Let us consider the following matrices

$$M(t) = \begin{bmatrix} 1 + \cos(t) & 1 + \cos(t) & 6 + \sin(2t) \\ 5 + \sin(2t) & 5 + \sin(2t) & -4 - \sin(t) \\ 3 + \cos(t) & 3 + \cos(t) & -6 + \cos(t) \\ 3 + \cos(t) & 3 + \cos(t) & -6 + \cos(t) \end{bmatrix},$$

$$L(t) = \begin{bmatrix} 4 + \cos(t) & 2 + \sin(t) \\ -2 + \sin(t) & 3 - \sin(t) \\ -2 + \cos(2t) & 7 + \cos(t) \\ 1 + \cos(t) & -1 + \cos(2t) \end{bmatrix},$$

where $M(t)$ is a rank-deficient matrix of rank 2. The results generated during the calculation of the MP inverse $M^\dagger(t)$ and the expression $M^\dagger(t)L(t)$ with the regularization parameter $\delta = 1e-6$ are presented in Fig. 4.

5.2 Example 2

This example deals with the following matrices:

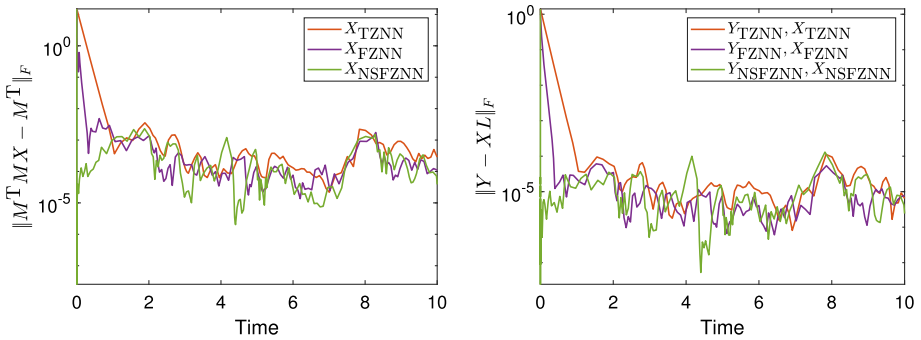
$$M(t) = \left[4 + \cos(t) \ 4 + \frac{\cos(t)}{2} \ \dots \ 4 + \frac{\cos(t)}{n} \right] \odot \mathbf{1}_{m \times n},$$

$$L(t) = \begin{bmatrix} 1 + \sin(t) \\ 2 + \frac{\sin(t)}{2} \\ \vdots \\ m + \frac{\sin(t)}{m} \end{bmatrix} \odot \mathbf{1}_{m \times k} \odot \begin{bmatrix} 1 \\ 2 \\ \vdots \\ m \end{bmatrix},$$

where \odot denotes the Hadamard product and $\mathbf{1}_{m \times n} \in \mathbb{R}^{m \times n}$, $\mathbf{1}_{m \times k} \in \mathbb{R}^{m \times k}$ are matrices of ones. It is worth noting that $M(t)$ is rank-deficient and of rank 1. Setting $m = 5$, $n = 10$, $k = 6$, the results generated during the calculation of $M^\dagger(t)L(t)$ and during solving the LME $M(t)Y(t) = L(t)$ with $\delta = 1e-6$ are presented in Fig. 5.

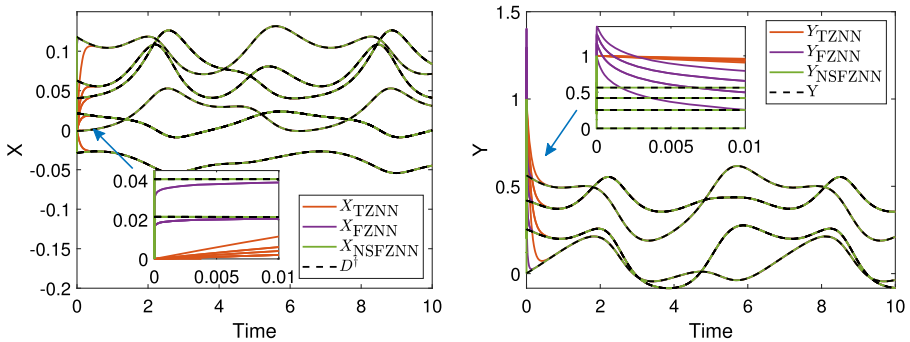
5.3 Discussion of Examples 1 and 2

The results of the TZNN, FZNN and NSFZNN models for finding the MP inverse of $M(t)$ are depicted in Fig. 4a, c in the case of Ex. 5.1 and in Fig. 5a, c in the case of Ex. 5.2, while



(a) $\|F_1(t)\|_F$ value in Ex. 5.1.

(b) $\|F_2(t)\|_F$ value in Ex. 5.1.



(c) $X(t) \approx M^\dagger(t)$ trajectories in Ex. 5.1. (d) $Y(t) \approx M^\dagger(t)L(t)$ trajectories in Ex. 5.1.

Fig. 4 The Frobenius norms values of ZFs, MP inverse trajectories and the LME solutions' trajectories in Ex. 5.1

the results in finding the expression $M^\dagger(t)L(t)$ are depicted in Fig. 4b, d in the case of Ex. 5.1 and in Fig. 5b, d in the case of Ex. 5.2.

It is noticeable that the values of Frobenius norms $\|F_1(t)\|_F$ in Figs. 4a and 5a, and $\|F_2(t)\|_F$ in Figs. 4b and 5b are minimal and, at the same time, almost similar across the time interval $[0, 10]$ in the cases of Ex. 5.1 and 5.2, respectively. However, it is also observable that the NSFZNN model converges faster to zero than FZNN, while the FZNN model converges faster to zero than TZNN. Another observation is that the MP inverse solutions' state trajectories, arranged in Figs. 4c and 5c, as well $M^\dagger(t)L(t)$ solutions' state trajectories, arranged in Figs. 4d and 5d, follow the same conclusion in the cases of Ex. 5.1 and 5.2, respectively. The conclusion is that NSFZNN converges faster to the solutions' state trajectories than TZNN, while the FZNN model converges faster to the solutions' state trajectories than TZNN.

General conclusions from Ex. 5.1 and 5.2 are that the NSFZNN model (35) generates the MP inverse of an arbitrary matrix $M(t)$ and, at the same time, computes $M^\dagger(t)L(t)$. Moreover, the NSFZNN model performs better than FZNN, while the FZNN model performs better than TZNN. An important observation from Figs. 4 and 5 is that the NSFZNN design achieves the fastest convergence in the initial part of the considered time interval. This observation confirms the fastest convergence of NSFZNN compared to FZNN and TZNN. In order to

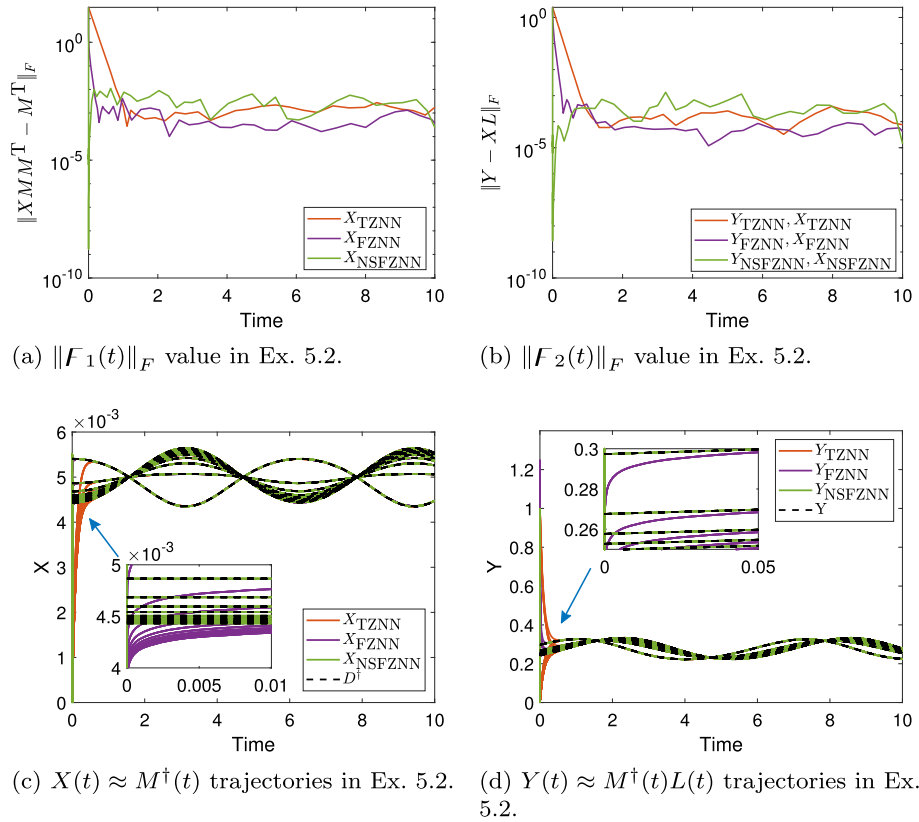


Fig. 5 The Frobenius norms values of ZFs, MP inverse trajectories and the LME solutions' trajectories in Ex. 5.2

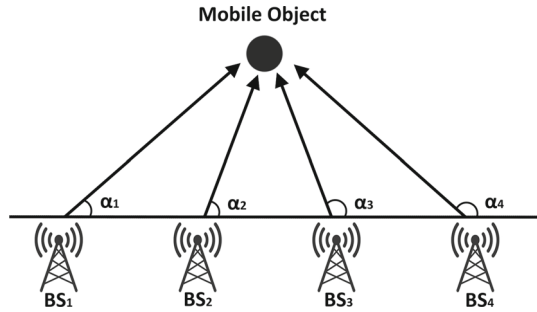
highlight such a situation, we have singled out the initial parts of some figures as involved sub-images. According to figures 4b, 5b, 7b, and 8b, the suggested NSFZNN model doubles the improved performance of FZNN compared to TZNN. The general conclusion is that the NSFZNN model performs better than FZNN, while the FZNN model performs better than TZNN.

5.4 Application on Solving Localization Problem

A specific solution to the AoA method in estimating a mobile object localization based on solving TV linear matrix equations and the QR decomposition was proposed in [16]. Inhere, we demonstrate applications of the TZNN, FZNN, and NSFZNN models in mobile object localization, based on the AoA algorithm. It is worth noting that AoA is commonly employed in the navigation [8], localization system [13], and wireless communication [24].

For convenience, a two-dimensional localization is examined in-depth, although this strategy can be extended to three-dimensional localization. The angle of incidence of the signal emitted by the mobile item determines the passive sensor array of each base station (BS), as shown in Fig. 6. At the time t , assume that the incident angles of four BSs are $\alpha_1(t), \dots, \alpha_d(t)$ and the mobile object position is the target of straight lines each passing through a BS. The

Fig. 6 Illustration of the AoA technique for the estimation of mobile object location



coordinates of i th BS are $(x_i, y_i)^T, i = 1, 2, \dots, d$, and the mobile object's TV coordinates are $(u(t), v(t))^T$. According to the geometrical treatment, the AoA localization is reduced to the linear system $M(t)Y(t) = L(t)$, where

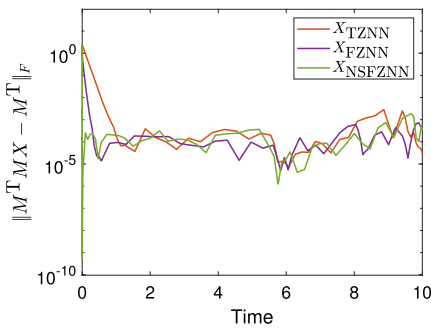
$$M(t) = \begin{bmatrix} -\tan \alpha_1(t) & 1 \\ -\tan \alpha_2(t) & 1 \\ \vdots & \vdots \\ -\tan \alpha_d(t) & 1 \end{bmatrix}, \quad L(t) = \begin{bmatrix} y_1 - x_1 \tan \alpha_1(t) \\ y_2 - x_2 \tan \alpha_2(t) \\ \vdots \\ y_d - x_d \tan \alpha_d(t) \end{bmatrix}. \tag{40}$$

The best approximate solution to (40) is given by $Y(t) = M^\dagger(t)L(t)$. We will use the TZNN, FZNN, and NSFZNN models to $M^\dagger(t)$ and, at the same time, to calculate $M^\dagger(t)L(t)$. In this way, the movable object's coordinates $(u(t), v(t))^T$ can be tracked by $M^\dagger(t)L(t)$.

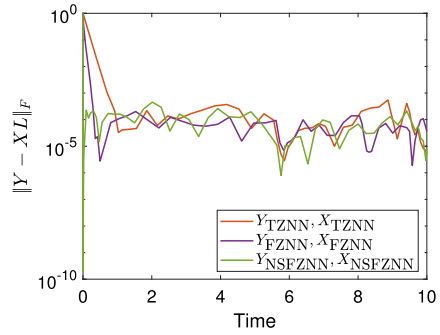
Consider the case $d = 4$ and the actual trajectory (AT) of the mobile object location defined by $u(t) = t/2 - 3.6$ and $v(t) = t/5 + \cos(t)/2 - 1.5$, while the four BSs ($BS_i, i = 1, \dots, 4$) have coordinates $(2, 3)^T, (5, -4)^T, (2, 5)^T$ and $(3, -3)^T$, respectively. That is, $M(t) \in \mathbb{R}^{4 \times 2}$ and $L(t) \in \mathbb{R}^4$. The input matrix is of rank 2 and the regularization parameter is $\delta = 0$. The results generated during the LME solution are depicted in Fig. 7 a–d, while a 3D representation of the actual trajectories (AT) along with the estimated trajectories (ET) of the mobile object is depicted in Fig. 10. A successful solution to the path-tracking task of the mobile object is observable.

The results of the TZNN, FZNN and NSFZNN models for finding the MP inverse of $M(t)$ are depicted in Fig. 7 a, c. It is observable that the values of the Frobenius norms $\|F_1(t)\|_F$ in Fig. 7 a and $\|F_2(t)\|_F$ in Fig. 7 b are minimal and, at the same time, almost similar across the time interval $[0, 10]$. However, it is also observable that the NSFZNN model converges faster to zero than FZNN, while the FZNN model converges faster to zero than TZNN. Moreover, the state trajectories of $M^\dagger(t)$ arranged in Fig. 7 c as well as the state trajectories of $M^\dagger(t)L(t)$ arranged in Fig. 7 d follow the same trajectory. The ET of the NSFZNN model converge faster to the AT than the ET of the FZNN, while the ET of the FZNN model converges faster to the AT than the ET of the TZNN. In conclusion, the NSFZNN model performs better than FZNN, while the FZNN model performs better than TZNN.

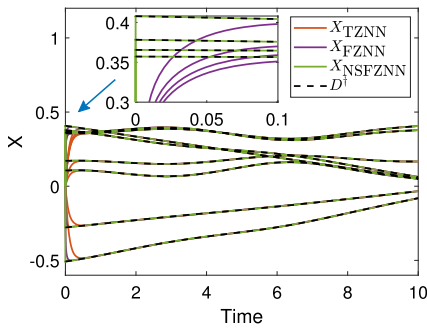
The general conclusion from Sect. 5.4 is that the NSFZNN model (35) generates the MP inverse of an arbitrary matrix $M(t)$ and, at the same time, computes $M^\dagger(t)L(t)$. In addition, the NSFZNN model performs better than FZNN, while the FZNN model performs better than TZNN.



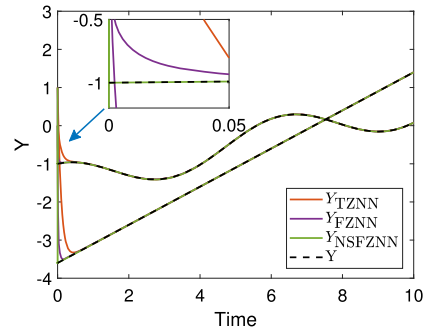
(a) $\|F_1(t)\|_F$ in App. 5.4.



(b) $\|F_2(t)\|_F$ in App. 5.4.



(c) $X(t) \approx M^\dagger(t)$ trajectories in App. 5.4.



(d) $Y(t) \approx M^\dagger(t)L(t)$ trajectories in App. 5.4.

Fig. 7 The Frobenius norms values of ZFs, MP inverse trajectories and the LME solutions' trajectories in Sect. 5.4

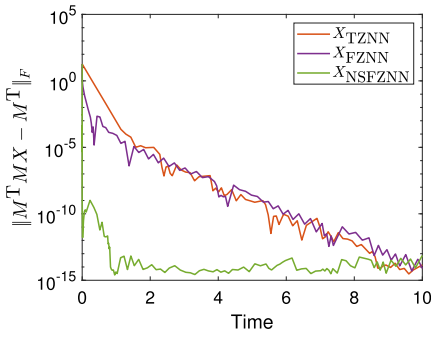
5.5 Application on Computing the Dynamic AC of an Electrical Network

The electronic circuit is presented in Fig. 9, and $V(t)$ is a dynamic AC voltage source. The TZNN, FZNN, and NSFZNN models are applied to calculate the dynamic AC currents of the circuit. According to the loop-current method and Kirchoff's voltage law, the linear circuit equations are obtained in the form

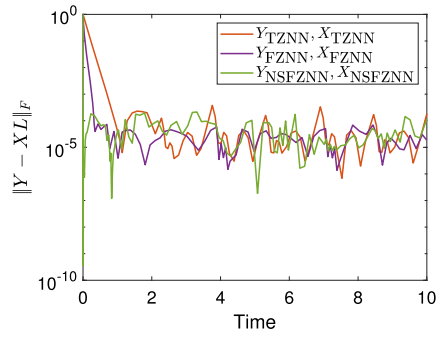
$$\begin{cases} (R_1 + R_2)I_1 - R_2I_3 = V(t) \\ (R_2 + 2R_3)I_2 - R_3I_3 = 0 \\ (R_2 + 2R_3)I_3 - R_2I_1 - R_3I_2 - R_3I_4 = 0 \\ (R_2 + R_3 + R_4)I_4 - R_3I_3 = 0 \end{cases}$$

where R_1, R_2, R_3, R_4 are the electrical resistance, $V(t)$ is the dynamic AC voltage source and I_1, I_2, I_3, I_4 are the electric currents of each loop to be solved.

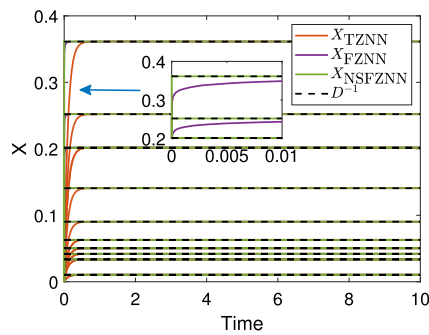
The electronic circuit equation above can be represented as the linear system $M(t)Y(t) = L(t)$ with



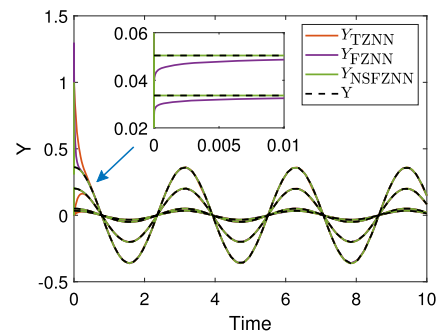
(a) $\|F_1(t)\|_F$ in App. 5.5.



(b) $\|F_2(t)\|_F$ in App. 5.5.



(c) $X(t) \approx M^\dagger(t)$ trajectories in App. 5.5.



(d) $Y(t) \approx M^\dagger(t)L(t)$ trajectories in App. 5.5.

Fig. 8 The Frobenius norms values of ZFs, MP inverse trajectories and the LME solutions' trajectories in Sect. 5.5

Fig. 9 The actual and the estimated trajectories of the mobile object in Sect. 5.4

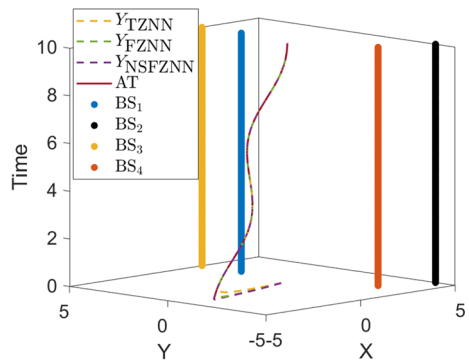
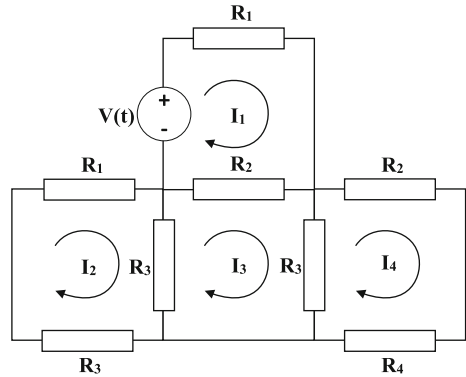


Fig. 10 Electrical network



$$M(t) = \begin{bmatrix} R_1 + R_2 & 0 & -R_2 & 0 \\ 0 & R_2 + 2R_3 & -R_3 & 0 \\ -R_2 & -R_3 & R_2 + 2R_3 & -R_3 \\ 0 & 0 & -R_3 & R_2 + R_3 + R_4 \end{bmatrix}$$

$$L(t) = \begin{bmatrix} V(t) \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad Y(t) = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix},$$

where the unknown matrix to be obtained is the current matrix $Y(t)$. Setting $R_1 = 1\Omega$, $R_2 = 4\Omega$, $R_3 = 2\Omega$, $R_4 = 6\Omega$ and $V(t) = \cos(2t)$, the input matrix $M(t)$ is of rank 4 and, hence, the regularization parameter is set to $\delta = 0$. The results generated during the LME solving are depicted in Fig. 8a–d.

The findings of the TZNN, FZNN, and NSFZNN models for calculating the MP inverse of $M(t)$ are depicted in Fig. 8a, c. The values of the Frobenius norms $\|F_1(t)\|_F$ and $\|F_2(t)\|_F$ are minimal with the NSFZNN model producing lower overall error than FZNN and TZNN across the time interval $[0, 10]$ in Fig. 8a, while all the errors are almost similar in Fig. 8b. Note that, in Fig. 8a and b, the NSFZNN model converges faster to the zero vector than FZNN, which converges faster to zero than the TZNN model. The state trajectories of $M^\dagger(t)$ are arranged in Fig. 8c and the state trajectories of $M^\dagger(t)L(t)$ are arranged in Fig. 8d. The NSFZNN model performs better than FZNN, and the FZNN model performs better than TZNN.

The general conclusion that emerges from Sect. 5.5 is that the NSFZNN model (35) generates the MP inverse of an arbitrary matrix $M(t)$ and simultaneously calculates $M^\dagger(t)L(t)$. Furthermore, the NSFZNN model outperforms the FZNN model, whereas FZNN outperforms the TZNN model. Graphs included in Figs. 7 and 8 reaffirm that NSFZNN design achieves the fastest convergence in initial time instants. This fact confirms the fastest convergence of NSFZNN compared to FZNN and TZNN in both considered applications.

6 Conclusion and Vision of Further Research

The best approximate solution $Y(t) = M^\dagger(t)L(t)$ of an LME $M(t)Y(t) = L(t)$, based on the MP inverse of an arbitrary TV matrix $M(t)$, is approached by the fuzzy ZNN (FZNN)

method enhanced by the neutrosophic system. A novel FZNN model, termed NSFZNN, is introduced based on the neutrosophic set theory. The solution $Y(t)$ includes various useful cases in linear algebra. The most important cases are the inverse of a nonsingular TV matrix, the MP inverse of a singular square, full-row or full-column rank, or rank-deficient real TV matrix. Based on the inverse or the MP inverse, the proposed dynamic system can solve an arbitrary LME or system of linear equations. Overall experience is that dynamical systems based on the NSFZNN reach a more rapid convergence than the FZNN and TZNN dynamics. Convergence characteristics on arbitrary order models are explored in performed simulation experiments. Applications are used to verify the obtained theoretical statements.

Some areas of future study can be featured as follows.

1. Various nonlinear activation functions, which may cause terminal convergence, can be considered as an approach to accelerate the NSFZNN model's flow.
2. Further research may involve the application of the NSFZNN evolution in approximating diverse matrix functions.
3. One of the advantages arising from the use of fuzzy and neutrosophic logic is the possibility of analyzing different scenarios, such as: pessimistic, realistic, and optimistic.

Acknowledgements Predrag Stanimirović is supported by the Ministry of Education, Science and Technological Development, Republic of Serbia, Contract No. 451-03-68/2020-14/200124. Predrag Stanimirović is supported by the Science Fund of the Republic of Serbia, #GRANT No 7750185, Quantitative Automata Models: Fundamental Problems and Applications—QUAM. This work was supported by the Ministry of Science and Higher Education of the Russian Federation (Grant No. 075-15-2022-1121).

Declarations

Conflicts of interest The authors Vasilios N. Katsikis, Predrag S. Stanimirović, Spyridon D. Mourtas, Lin Xiao, Dragiša Stanujkić and Darjan Karabašević of the paper entitled “Zeroing Neural Network based on Neutrosophic Logic for Calculating Minimal-norm Least-squares Solutions to Time-varying Linear Systems,” declare that there is no conflict of interest.

References

1. Ansari AQ (2017). Keynote speakers: From fuzzy logic to neutrosophic logic: a paradigm shift and logics. pp 11–15, Jaipur, India. IEEE
2. Atanassov KT (1986) Intuitionistic fuzzy sets. *Fuzzy Sets Syst* 20(1):87–96
3. Ben-Israel A (1986) Generalized inverses of matrices: a perspective of the work of Penrose. *Math Proc Camb Philos Soc* 100(3):407
4. Ben-Israel A (2002) The Moore of the Moore–Penrose inverse. *Electron J Linear Algebra* 9:150–157
5. Christianto V, Smarandache F (2019) A review of seven applications of neutrosophic logic: in cultural psychology, economics theorizing, conflict resolution, philosophy of science, etc. *Multidiscip Sci J* 2:128–137
6. Dai J, Chen Y, Xiao L, Jia L, He Y (2022) Design and analysis of a hybrid GNN–ZNN model with a fuzzy adaptive factor for matrix inversion. *IEEE Trans Ind Inform* 18(4):2434–2442
7. Dean P, Porrill J (1998) Pseudo-inverse control in biological systems: a learning mechanism for fixation stability. *Neural Netw* 7–8:1205–1218
8. Dempster AG, Cetin E (2016) Interference localization for satellite navigation systems. *Proc IEEE* 104(6):1318–1326
9. Deng Y, Ren Z, Kong Y, Bao F, Dai Q (2017) A hierarchical fused fuzzy deep neural network for data classification. *IEEE Trans Fuzzy Syst* 25(4):1006–1012
10. Feng S, Wu H (2017) Hybrid robust boundary and fuzzy control for disturbance attenuation of nonlinear coupled ode-beam systems with application to a flexible spacecraft. *IEEE Trans Fuzzy Syst* 25(5):1293–1305

11. Hu Z, Xiao L, Li K, Li K, Li J (2021) Performance analysis of nonlinear activated zeroing neural networks for time-varying matrix pseudoinversion with application. *Appl Soft Comput* 98:106735
12. Huang F, Zhang X (2006) An improved Newton iteration for the weighted Moore-Penrose inverse. *Appl Math Comput* 174(2):1460–1486
13. Huang H, Fu D, Xiao X, Ning Y, Wang H, Jin L, Liao S (2020) Modified Newton integration neural algorithm for dynamic complex-valued matrix pseudoinversion applied to mobile object localization. *IEEE Trans Ind Inf* 17(4):2432–2442
14. Jia L, Xiao L, Dai J, Cao Y (2021) A novel fuzzy-power zeroing neural network model for time-variant matrix Moore-Penrose inversion with guaranteed performance. *IEEE Trans Fuzzy Syst* 29(9):2603–2611
15. Jia L, Xiao L, Dai J, Qi Z, Zhang Z, Zhang Y (2021) Design and application of an adaptive fuzzy control strategy to zeroing neural network for solving time-variant QP problem. *IEEE Trans Fuzzy Syst* 29(6):1544–1555
16. Katsikis VN, Mourtas SD, Stanimirović PS, Zhang Y (2022) Solving complex-valued time-varying linear matrix equations via QR decomposition with applications to robotic motion tracking and on angle-of-arrival localization. *IEEE Trans Neural Netw Learn Syst* 33(8):3415–3424
17. Katsikis VN, Stanimirović PS, Mourtas SD, Xiao L, Karabasević D, Stanujkić D (2022) Zeroing neural network with fuzzy parameter for computing pseudoinverse of arbitrary matrix. *IEEE Trans Fuzzy Syst* 30(9):3426–3435
18. Li W, Ma X, Luo J, Jin L (2021) A strictly predefined-time convergent neural solution to equality- and inequality-constrained time-variant quadratic programming. *IEEE Trans Syst Man Cybern Syst* 51(7):4028–4039
19. Liao B, Zhang Y (2014) Different complex ZFs leading to different complex ZNN models for time-varying complex generalized inverse matrices. *IEEE Trans Neural Netw Learn Syst* 25(9):1621–1631
20. Lin J, Lin CC, Lo HS (2009) Pseudo-inverse Jacobian control with grey relational analysis for robot manipulators mounted on oscillatory bases. *J Sound Vib* 326(3–5):421–437
21. Liu X, Yu Y, Zhong J, Wei Y (2012) Integral and limit representations of the outer inverse in Banach space. *Linear Multilinear Algebra* 60:333–347
22. Mallik S, Mohanty S, Mishra BS (2022) Biologically Inspired Techniques in Many Criteria Decision Making, volume 271 of *Smart Innovation, Systems and Technologies*, chapter Neutrosophic Logic and its Scientific Applications. Springer, Singapore
23. Naik J, Dash S, Dash P, Bisoi R (2018) Short term wind power forecasting using hybrid variational mode decomposition and multi-kernel regularized pseudo inverse neural network. *Renew Energy* 118:180–212
24. Noroozi A, Oveis AH, Hosseini SM, Sebt MA (2018) Improved algebraic solution for source localization from TDOA and FDOA measurements. *IEEE Wirel Commun Lett* 7(3):352–355
25. Penrose R (1956) On a best approximate solutions to linear matrix equations. *Proc Camb Phil Soc* 52:17–19
26. Precup R-E, Tomescu M-L, Dragos C-A (2014) Stabilization of Rössler chaotic dynamical system using fuzzy logic control algorithm. *Int J Gen Syst* 43(5):413–433
27. Salama AA, Alhasan KF, Elagamy HA, Smarandache F (2021) Neutrosophic dynamic set. *Neutrosophic Knowl* 3
28. Smarandache F (1998) *Neutrosophy: Neutrosophy probability, set and logic*. American Research Press, Analytic Synthesis and Synthetic Analysis
29. Smarandache F (2001) A unifying field in logics: Neutrosophic logic, neutrosophic set, neutrosophic probability and statistics (fourth edition). [arXiv:math/0101228](https://arxiv.org/abs/math/0101228)
30. Smarandache F (2003) Neutrosophic logic—generalization of the intuitionistic fuzzy logic. [arXiv:math/0303009](https://arxiv.org/abs/math/0303009)
31. Stanimirović PS, Katsikis VN, Jin L, Mosaic D (2021) Properties and computation of continuous-time solutions to linear systems. *Appl Math Comput* 405:126242
32. Stanimirović PS, Katsikis VN, Li S (2019) Integration enhanced and noise tolerant ZNN for computing various expressions involving outer inverses. *Neurocomputing* 329:129–143
33. Stanimirović PS, Katsikis VN, Zhang Z, Li S, Chen J, Zhou M (2020) Varying-parameter Zhang neural network for approximating some expressions involving outer inverses. *Optim Methods Softw* 35(6):1304–1330
34. Stanimirović PS, Stojanović I, Katsikis VN, Pappas D, Zdravec Z (2015) Application of the least squares solutions in image deblurring. *Math Probl Eng*, 2015. Article ID 298689
35. Tan Z, Hu Y, Xiao L, Chen K (2019) Robustness analysis and robotic application of combined function activated RNN for time-varying matrix pseudo inversion. *IEEE Access* 7:33434–33440
36. Wang A, Liu L, Qiu J, Feng G (2019) Event-triggered robust adaptive fuzzy control for a class of nonlinear systems. *IEEE Trans Fuzzy Syst* 27(8):1648–1658

37. Wang F, Chen B, Liu X, Lin C (2018) Finite-time adaptive fuzzy tracking control design for nonlinear systems. *IEEE Trans Fuzzy Syst* 26(3):1207–1216
38. Wang H, Li J, Liu H (2006) Practical limitations of an algorithm for the singular value decomposition as applied to redundant manipulators. In: 2006 IEEE conference on robotics, automation and mechatronics, pp 1–6
39. Xiao L, Liao B, Li S, Zhang Z, Ding L, Jin L (2018) Design and analysis of FTZNN applied to the real-time solution of a nonstationary Lyapunov equation and tracking control of a wheeled mobile manipulator. *IEEE Trans Ind Inform* 14(1):98–105
40. Xiao L, Zhang Y (2014) From different Zhang functions to various ZNN models accelerated to finite-time convergence for time-varying linear matrix equation. *Neural Process Lett* 39(3):309–326
41. Yu F, Liu L, Xiao L, Li K, Cai S (2019) A robust and fixed-time zeroing neural dynamics for computing time-variant nonlinear equation using a novel nonlinear activation function. *Neurocomputing* 350:108–116
42. Zadeh LA (1965) Fuzzy sets. *Inf Control* 8(3):338–353
43. Zhang Y, Ge SS (2005) Design and analysis of a general recurrent neural network model for time-varying matrix inversion. *IEEE Trans Neural Netw* 16(6):1477–1490
44. Zhang Y, Li F, Yang Y, Li Z (2012) Different Zhang functions leading to different Zhang-dynamics models illustrated via time-varying reciprocal solving. *Appl Math Model* 36(9):4502–4511
45. Zhang Y, Yang Y, Tan N, Cai B (2011) Zhang neural network solving for time-varying full-rank matrix Moore–Penrose inverse. *Computing* 92(2):97–121
46. Zhang Z, Fu Z, Zheng L, Gan M (2018) Convergence and robustness analysis of the exponential-type varying gain recurrent neural network for solving matrix-type linear time-varying equation. *IEEE Access* 6:57160–57171
47. Zhang Z, Yan Z (2020) An adaptive fuzzy recurrent neural network for solving non-repetitive motion problem of redundant robot manipulators. *IEEE Trans Fuzzy Syst* 28(4):684–691
48. Zhang Z, Zheng L, Qiu T, Deng F (2020) Varying-parameter convergent-differential neural solution to time-varying overdetermined system of linear equations. *IEEE Trans Autom Control* 65(2):874–881
49. Zhou J, Zhu Y, Li XR, You Z (2002) Variants of the Greville formula with applications to exact recursive least squares. *SIAM J Matrix Anal Appl* 24(1):150–164

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Vasilios N. Katsikis^{1,2,3}  · Predrag S. Stanimirović^{4,5} · Spyridon D. Mourtas^{1,5} · Lin Xiao⁶ · Dragiša Stanujkić⁷ · Darjan Karabašević⁸

Predrag S. Stanimirović
pecko@pmf.ni.ac.rs

Spyridon D. Mourtas
spirosmourtas@gmail.com

Lin Xiao
xiaolin860728@163.com

Dragiša Stanujkić
dstanujkic@tfbor.bg.ac.rs

Darjan Karabašević
darjan.karabasevic@mef.edu.rs

- ¹ Department of Economics, Division of Mathematics-Informatics and Statistics-Econometrics, National and Kapodistrian University of Athens, Sofokleous 1 Street, 10559 Athens, Greece
- ² University of Athens Master of Business Administration, National and Kapodistrian University of Athens, Athens, Greece
- ³ Administration of Businesses and Organizations Department, Hellenic Open University, 26335 Patras, Greece
- ⁴ Faculty of Sciences and Mathematics, University of Niš, Višegradska 33, Niš 18000, Serbia
- ⁵ Laboratory “Hybrid Methods of Modelling and Optimization in Complex Systems”, Siberian Federal University, Prosp. Svobodny 79, Krasnoyarsk, Russia 660041
- ⁶ Hunan Provincial Key Laboratory of Intelligent Computing and Language Information Processing, Changsha 410081, China
- ⁷ Technical Faculty at Bor, University of Belgrade, Vojske Jugoslavije 12, Bor 19210, Serbia
- ⁸ Faculty of Applied Management, Economics and Finance, University Business Academy in Novi Sad, Jevrejska 24, Belgrade 11000, Serbia

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).

Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval, sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com