New Stability Condition for Discrete-Time Fully Coupled Neural Networks with Multivalued Neurons ☆

Wei Zhou[a], Jacek M. Zurada[b,c]

[a] *College of Computer Science and Technology, Southwest University for Nationalities, Chengdu 610041, P. R. China.*
[b] *Computational Intelligence Laboratory, Electrical and Computer Engineering Department, University of Louisville, Louisville, KY, 40292, USA.*
[c] *Information Technology Institute, University of Social Sciences, 90-113 Łódz, Poland.*

Abstract

This paper discusses the stability condition for discrete-time multi-valued recurrent neural networks (MVNRNNs) in asynchronous update mode. In the existing research literature, an MVNRNN in asynchronous update mode has been found convergent if its weight matrix is Hermitian with nonnegative diagonal entries. However, our finding has been that the weight matrix with zero diagonal entries can't guarantee the network stability. Furthermore, the new stability condition and proof is offered to allow diagonal entries to be complex-valued, which extends previous theoretical result. Simulation results are used to illustrate the theory.

Key words: Discrete-time recurrent neural networks, Multivalued neurons, Convergence analysis, Complex-valued neural networks

---

1. Introduction

The multi-valued neuron (MVN) was first introduced in [7], and its theory was further extended in [5, 8]. Neural networks with MVN (MVNNNs))

☆The research reported was supported by National Science Foundation of China under Grant 61105061.

*Corresponding author.

Email addresses: wei.zhou.swun@gmail.com (Wei Zhou), jmzura02@louisville.edu (Jacek M. Zurada)

Preprint submitted to Neurocomputing *January 28, 2015*

adopt a complex-valued activation function, which maps complex-valued inputs into outputs on the unit circle in the complex domain.

An MVN activation function is different to other neural networks' function. Firstly, function output is sensitive to input's argument, which lies on $[0, 2\pi)$. Thus we should consider the imaginary part and real part all together. Such situation never happens in real-valued neural networks (NNs), or those complex-valued NNs which deal with imaginary part and real part separately. Secondly, many activation functions are monotonously increasing or decreasing, and changing the output from its maximum to minimum involves time. However, MVN can switch its output state quickly, by simply multiplying a complex number to change its argument. It has been shown that the functionality of an MVN is higher than the functionality of a sigmoidal neuron [6]. For example, a multilayer neural network based on MVN outperforms a classical multilayer feed-forward network and several kernel-based NNs with faster learning speed and fewer neurons [4]. Some successful recent applications of MVNNNs have been reported in [1, 2, 3, 4].

Dynamical analysis is of primary importance for emulation of stability and of fixed points of recurrent neural networks (RNNs). Comparing dynamical analysis work with real-valued analysis, the work in complex-valued domain is more difficult because we only get partial theoretical support from current mathematical theories. For example, it seems impossible to study the dynamics of continuous-time complex-valued MVNRNNs directly, because MVN activation function is not holomorphic. Fortunately, the requirement for holomorphism can sometimes be eliminated when we deal with discrete-time RNNs. On the other hand, due to the strong component correlation between real and imaginary parts, the decrease or increase in either imaginary or real part usually means nothing. In real-valued domain, it is common to observe, calculate, or analyze the trajectory of RNNs simply based on the network outputs' decrease or increase. However, in complex-valued domain, we need to investigate the trajectory movement on the whole complex plane.

Stability and complete stability are different concepts. If a network is stable, stable fixed point(s) or periodic solution(s) may exist. However, if a network is completely stable, all trajectories will convergent to a fixed point, which means that no periodic solutions can be found. In most cases, we hope RNNs to be completely stable.

RNNs dynamics greatly depends on the update model used. Like other RNNs, MVNRNNs use two update models: synchronous update mode, and asynchronous update mode. The stability condition in synchronous update mode can been found in [14]. Here, we focus on MVNRNNs' dynamics in asynchronous update mode. In the seminal paper [9], the stability condition has established that a discrete-time MVNRNN is convergent if its weight matrix is Hermitian with nonnegative diagonal entries. Applications related to this property focus on associative memory design [5, 6, 9, 11]. In [12], through analyzing the same energy function used in [9], the authors prove that the energy function for each of the stored patterns will also take the minimum values. Although much successful work has been based on [9], an obscure flaw exists in its original proof. According to our most recent research, an MVNRNN may not be completely stable if its weight matrix is Hermitian with zero diagonal entries [15].

Another interesting topic is to study MVN networks with non-Hermitian matrices. In [13], a threshold complex-valued neural networks associative memory is proposed for information retrieval. The test results show that MVN networks with small asymmetry in weight matrix can be stable and function as well as Hermitian one. However, the analytical proof of stability of such MVN networks is still missing.

In this paper, we deal with MVN networks with non-Hermitian weights, and present a revised stability condition based on [9, 15], which extends previous results by allowing MVNRNNs to be completely stable with complex-valued diagonal entries. Regarding asymmetric MVN networks in synchronous update mode, to the best of our knowledge, no theoretical stability result has been reported. Therefore, our work also presents a novel research approach to study the stability of MVN networks with non-Hermitian matrices.

The rest of this paper is organized as follows. The architecture of MVN-RNNs is described in Section 2. Section 3 is the theoretical analysis. Simulations are presented in Section 4. Conclusions are given in Section 5.

2. Multi-valued Recurrent Neural Networks

The MVN model is based on the activation function defined as complex-signum operation (see Fig.1). For a specified number of values K, called the resolution factor, and an arbitrary complex number u, the complex-signum function is defined as follows:

$$CSIGN(u) \triangleq \begin{cases} z^0, & 0 \leq \arg(u) < \varphi_0 \\ z^1, & \varphi_0 \leq \arg(u) < 2\varphi_0 \\ \vdots & \\ z^{\mathrm{K}-1}, & (\mathrm{K}-1)\varphi_0 \leq \arg(u) < \mathrm{K}\varphi_0 \end{cases},$$

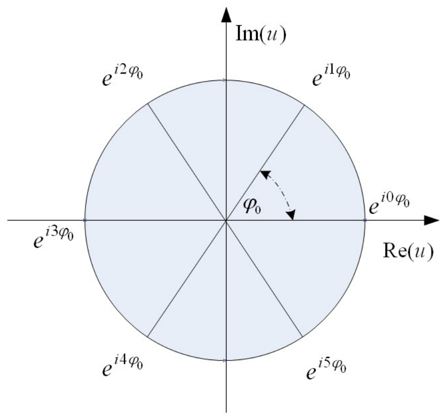

Figure 1: Complex signum function CSIGN (case shown for K = 6).

where φ_0 is a phase quantum delimited by K: $\varphi_0 = 2\pi/\mathrm{K}$, and z is the corresponding Kth root of unity: $z = e^{i\varphi_0}$. Then, the output state of each neuron is represented by a complex number from the set $\{z^0, z^1, \cdots, z^{\mathrm{K}-1}\}$. Thus the network state $s(k)$ at k-th iteration number, is a complex-valued vector of n components $s(k) = [s_1(k), s_2(k), \cdots, s_n(k)]^T$. For simplicity, in this article, we use $\sigma(\cdot)$ instead of $CSIGN(\cdot)$.

Each input $I_m(k+1)$ of the mth neuron is dependent upon the network state $s(k)$ through synaptic weights w_{ij}:

$$I_m(k+1) = \sum_{j=1}^{n} w_{mj} s_j(k) + h_m,$$

where h_m is a bias, $W = (w_{ij})_{n\times n}$ is a complex-valued matrix, each of its elements w_{mj} denotes the synaptic weights and represents the strength of the synaptic connection from neuron m to neuron j. Here, we set $h_m = 0$ for simplicity. The output of the m-th neuron is centered within respective K sectors shown in Fig.1, and we have

$$s_m(k+1) = \sigma(I_m(k+1) \cdot z^{1/2}) \tag{1}$$

where $z^{1/2} = e^{i(\varphi_0/2)}$.

3. New Stability Condition for MVNRNNs

First, we provide preliminaries used in the following to establish the theory.

For any $c \in \mathbb{C}$, we denote

$$c^* = (\bar{c})^T,$$

where $\bar{c}$ is the conjugate of c.

Definition 1. *A vector $s^\dagger$ is called an equilibrium point (fixed point) of network (1), if each element $s^\dagger_m$ in $s^\dagger$ satisfies*

$$s^\dagger_m = \sigma(z^{\frac{1}{2}} \cdot \sum_{j=1}^{n} w_{mj} s^\dagger_j).$$

Clearly, for $s^\dagger$, it holds that

$$s^\dagger = \sigma(WQs^\dagger),$$

where $Q = diag(z^{1/2}, z^{1/2}, \cdots, z^{1/2})$.

Denote by Ω the set of equilibrium points of the network (1).

Definition 2. *The network (1) is said to be completely convergent (completely stable), if each trajectory $s(k)$ satisfies*

$$dist(s(k), \Omega) \triangleq \min_{x^* \in \Omega} \left\| s(k) - s^\dagger \right\| \to 0$$

as $k \to +\infty$.

Theorem 1. *For a complex-valued matrix W, if W can be presented as $W = W^{'} + D$, where $W^{'}$ is a Hermitian matrix with zero diagonal entries ($w^{'}_{ii} = 0$ and $w^{'}_{ij} = w_{ij} = w^*_{ji}$,), D is a diagonal matrix with diagonal elements $d_{ii} = w_{ii} \neq 0$ and $arg(d_{ii}) \in [0, \varphi_0/2) \cup (2\pi - \varphi_0/2, 2\pi)$ for all $i, j \in \{1, 2, \cdots, n\}$, then the network (1) is completely convergent.*

Proof.: Here, we construct the following energy function

$$\begin{aligned} E(s(k)) &= -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} s^*_i(k) s_j(k) \\ &\quad + \left(-\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} s^*_i(k) s_j(k) \right)^* . \end{aligned} \tag{2}$$

Clearly, the energy function is real-valued.

Next, we want to prove that the energy function is monotonously decreasing. The change of the energy function for two consecutive states $s(k+1)$ and $s(k)$ can be written as

$$\Delta E = E(s(k+1)) - E(s(k)). \tag{3}$$

Suppose we update the m-th neuron at step $k+1$. Clearly, at the equilibrium point $s^{\dagger}$, it must hold that

$$s_m^{\dagger} = \sigma(z^{\frac{1}{2}} \cdot \sum_{j=1}^{N} w_{mj} s_j^{\dagger}) \Rightarrow \Delta E = 0 \ for \ all \ m.$$

Because the network is asynchronously updated, it holds that

$$\begin{cases} s_j(k+1) = s_j(k), j \neq m, \\ s_m(k+1) = \sigma(z^{\frac{1}{2}} I_m(k+1)) \end{cases}. \tag{4}$$

Furthermore, $I_m(k+1)$ and $s_m(k+1)$ can be written as

$$\begin{cases} I_m(k+1) = |I_m(k+1)| \, z^r e^{i\Delta\varphi} s_m(k) \\ s_m(k+1) = z^r s_m(k) \end{cases}, \tag{5}$$

where z^r is the bias between $s_m(k)$ and $s_m(k+1)$ on the complex-valued unit circle,

$$r \in \{-(\mathrm{K}-1), -(\mathrm{K}-2), \cdots, \mathrm{K}-1\},$$

and

$$\Delta\phi \in [-\frac{\phi_0}{2}, \ \frac{\phi_0}{2}).$$

Now the energy function (2) can be written as

$$E(s(k)) = M_1 + (M_1)^*, \tag{6}$$

where

$$\begin{aligned} M_1 \quad = \quad & -\frac{1}{2}(\sum_{i=1, i\neq m}^{n} \sum_{j=1, j\neq m}^{n} w_{ij} s_i^*(k) s_j(k) \\ & + \sum_{j=1, j\neq m}^{n} w_{mj} s_m^*(k) s_j(k) \\ & + \sum_{i=1, i\neq m}^{n} w_{im} s_i^*(k) s_m(k) \\ & + w_{mm} s_m^*(k) s_m(k)). \end{aligned} \tag{7}$$

Using (3), (4), (6) , (7) and the fact $s_m^*(k+1)s_m(k+1) = s_m^*(k)s_m(k) = 1$, the energy change ΔE can be expressed as

$$\Delta E = M_2 + (M_2)^*, \tag{8}$$

where

$$\begin{aligned}
M_2 &= -\frac{1}{2}[\sum_{j=1,j\neq m}^{n} w_{mj}s_m^*(k+1)s_j(k) \\
&\quad + \sum_{i=1,i\neq m}^{n} w_{im}s_i^*(k)s_m(k+1) \\
&\quad + w_{mm}s_m^*(k+1)s_m(k+1)] \\
&\quad + \frac{1}{2}[\sum_{j=1,j\neq m}^{n} w_{mj}s_m^*(k)s_j(k) \\
&\quad + \sum_{i=1,i\neq m}^{n} w_{im}s_i^*(k)s_m(k) + w_{mm}s_m^*(k)s_m(k)] \\
&= -\mathrm{Re}(\sum_{j=1,j\neq m}^{n} w_{mj}(s_m^*(k+1) - s_m^*(k))s_j(k)) \\
&= -\mathrm{Re}((s_m^*(k+1) - s_m^*(k))\sum_{j=1}^{n} w_{mj}s_j(k)) \\
&\quad + \mathrm{Re}(w_{mm}(s_m^*(k+1) - s_m^*(k))s_m(k)).
\end{aligned} \tag{9}$$

In order to make the change of energy function $\Delta E \leq 0$, the term M_2 needs to be located in the left half-plane of complex plane.

By (4) and (5), it holds that

$$\begin{aligned}
&-\mathrm{Re}((s_m^*(k+1) - s_m^*(k))\sum_{j=1}^{n} w_{mj}s_j(k)) \\
= &-\mathrm{Re}(((z^r s_m(k))^* - s_m^*(k))\,|I_m(k+1)|\,z^r e^{i\Delta\varphi}s_m(k)) \\
= &-\mathrm{Re}(|I_m(k+1)|\,(e^{i\Delta\varphi} - z^r e^{i\Delta\varphi}))
\end{aligned} \tag{10}$$

Using (9) and (10), M_2 can be written as

$$\begin{aligned}
M_2 &= -\mathrm{Re}(|I_m(k+1)|\,(e^{i\Delta\varphi} - z^r e^{i\Delta\varphi})) \\
&\quad + \mathrm{Re}(w_{mm}((z^r)^* - 1)).
\end{aligned} \tag{11}$$

Further, by (8), (11), we have

$$\begin{aligned}\Delta E &= -2\mathrm{Re}(|I_m(k+1)|\,(e^{i\Delta\varphi} - z^r e^{i\Delta\varphi})) \\ &\quad +2\mathrm{Re}(w_{mm}((z^r)^* - 1)). \end{aligned} \tag{12}$$

At first, we try to analyze the first term of (12). Obviously, for all $r \in \{-(\mathrm{K}-1), -(\mathrm{K}-2), \cdots, k-1\}$, it always holds that

$$-\mathrm{Re}(|I_m(k+1)|\,(e^{i\Delta\varphi} - z^r e^{i\Delta\varphi})) \leq 0. \tag{13}$$

Especially, when $\mathrm{Re}(|I_m(k+1)|\,(e^{i\Delta\varphi} - z^r e^{i\Delta\varphi})) = 0$, three cases should be considered:

(i) If $\Delta\varphi \neq -\phi_0/2$ and $|I_m(k+1)| \neq 0$, we have

$$\Delta\varphi \neq -\frac{\phi_0}{2} \Rightarrow r = 0 \Rightarrow s_m(k+1) = s_m(k). \tag{14}$$

(ii) If $\Delta\varphi = -\phi_0/2$ and $|I_m(k+1)| \neq 0$, we have

$$\begin{aligned}\Delta\varphi = -\frac{\phi_0}{2} \quad &\Rightarrow \quad r = 0 \textit{ or } r = 1 \\ &\Rightarrow \quad s_m(k+1) = s_m(k) \textit{ or } z \cdot s_m(k). \end{aligned} \tag{15}$$

(iii) If $|I_m(k+1)| = 0$, we have

$$\begin{aligned} &|I_m(k+1)| = 0 \Rightarrow r \in \{-(\mathrm{K}-1), \cdots, \mathrm{K}-1\} \\ \Rightarrow \quad & s_m(k+1) = z^r \cdot s_m(k). \end{aligned} \tag{16}$$

Next, we come to investigate the second term of (12). Because $r \in \{-\mathrm{K}-1, \cdots \mathrm{K}-1\}$, we have

$$(z^r)^* - 1 \in \{z^0 - 1, z^1 - 1, \cdots, z^{\mathrm{K}-1} - 1\}. \tag{17}$$

And it also holds that

$$\arg(z^0 - 1) < \arg(z^1 - 1) < \cdots < \arg(z^{\mathrm{K}-1} - 1). \tag{18}$$

Especially, by simply calculation, we have

$$\arg(z^1 - 1) = \frac{\pi}{2} + \frac{\phi_0}{2}, \tag{19}$$

and

$$\arg(z^{\mathrm{K}-1} - 1) = \frac{3\pi}{2} - \frac{\phi_0}{2}. \tag{20}$$

Because $w_{mm} \neq 0$ and $\arg(w_{mm}) \in (2\pi - \phi_0/2, 2\pi) \cup [0, \phi_0/2)$, by (17), (18), (19), and (20), it also holds that

$$\mathrm{Re}(w_{mm}((z^r)^* - 1)) \leq 0. \tag{21}$$

and

$$\begin{aligned} & \mathrm{Re}(w_{mm}((z^r)^* - 1)) = 0 \\ \Rightarrow \quad & r = 0 \Rightarrow s_m(k+1) = s_m(k). \end{aligned} \tag{22}$$

By (12), (13) and (21), we can find $\Delta E \leq 0$. Therefore, $E(k)$ is monotonic and decreasing. Especially, if $\Delta E(k) = 0$, it must hold that

$$\mathrm{Re}(|I_m(k+1)|\,(e^{i\Delta\varphi} - z^r e^{i\Delta\varphi})) = 0$$

and

$$\mathrm{Re}(w_{mm}((z^r)^* - 1)) = 0.$$

By (14), (15), (16), and (22), it is easy to see $r = 0$, which means $s_m(k+1) = s_m(k)$. Furthermore, because $\Delta E(k) = 0$ holds for all $m = \{1, \cdots, n\}$, it must hold that $s(k+1) = s(k)$. Therefore, the network is completely convergent.

This completes the proof.

Theorem 1 can be looked at as the extension of stability condition in [9, 15]. If w_{ii} are real-valued, in order to make the MVN network completely stable, we must set $w_{ii} > 0$. In [9], the complete stability condition is $w_{ii} \geq 0$, which includes $w_{ii} = 0$. However, by Theorem 1, $w_{ii} = 0$ means the possibility that $s_m(k+1) = z^r \cdot s_m(k)(r \neq 0)$ are at the minimum of the energy function, and the trajectory will not be convergent to a fixed point, thus the network is not completely stable at this time.

The condition $w_{ii} = 0$ is often used in MVN research and applications. For example, the generalized projection rule [10] requires $w_{ii} = 0$. Some associative memory design methods using MVN also set $w_{ii} = 0$ [11, 12]. Although there are many successful applications based on zero diagonal values $w_{ii} = 0$, it doesn't mean that the condition $\Delta\varphi = -\phi_0/2$ or $|I_m(k+1)| = 0$ can't happen. In [15], we also provide a counter-example to support above viewpoint.

Unlike in [9, 15], the stability condition in Theorem 1 allows us to adjust the diagonal elements in an interval of the complex plane. Theorem 1 also shows the fact that even if the weight W is not an Hermitian matrix, the network still can be completely stable. In [13], an associative memory using MVN networks with small asymmetry matrices has been verified through simulations. Although Theorem 1 can't be used to prove the correctness of such associative memory design method directly, it offers an alternate approach to explain the dynamics of MVN networks with non-Hermitian matrices.

4. Simulations

In this section, we provide simulation results to illustrate and verify the theory developed. For asynchronous update mode, there are several strategies to pick an updating neuron. Here, we choose to update neurons randomly per round. Each round includes n iterations. In each round, each neuron will be picked at random and updated only once.

Example 1. *Consider a MVN neural network with 3 neurons, $K = 8$*

$$s_m(k+1) = \sigma(z^{1/2} \cdot \sum_{j=1}^{n} w_{mj} s_j(k)). \tag{23}$$

where W is

$$\begin{bmatrix} 1.1564 + 0.9887\mathrm{i} & 0 & 0 \\ 0 & 1.4877 + 0.1564\mathrm{i} & 0.3536 + 0.3536\mathrm{i} \\ 0 & 0.3536 - 0.3536\mathrm{i} & 1.3090 - 0.5878\mathrm{i} \end{bmatrix}.$$

Here, the initial value $s(0) = [0.8022+0.6420i,\ 0.1034-0.2178i,\ 0.0421-0.4559i]^*$. By calculation, $\arg(w_{11}) \approx 0.7069$ and $\arg(w_{33}) \approx 5.8611$. Therefore, the network does not satisfy Theorem 1 and is not completely stable (see Fig.2).

Example 2. *Modify the MVN network as in (23), and assume W is*

$$\begin{bmatrix} 1.8090 + 0.1878\mathrm{i} & 0 & 0 \\ 0 & 1.4877 + 0.1564\mathrm{i} & 0.3536 + 0.3536\mathrm{i} \\ 0 & 0.3536 - 0.3536\mathrm{i} & 1.3090 - 0.1878\mathrm{i} \end{bmatrix}.$$

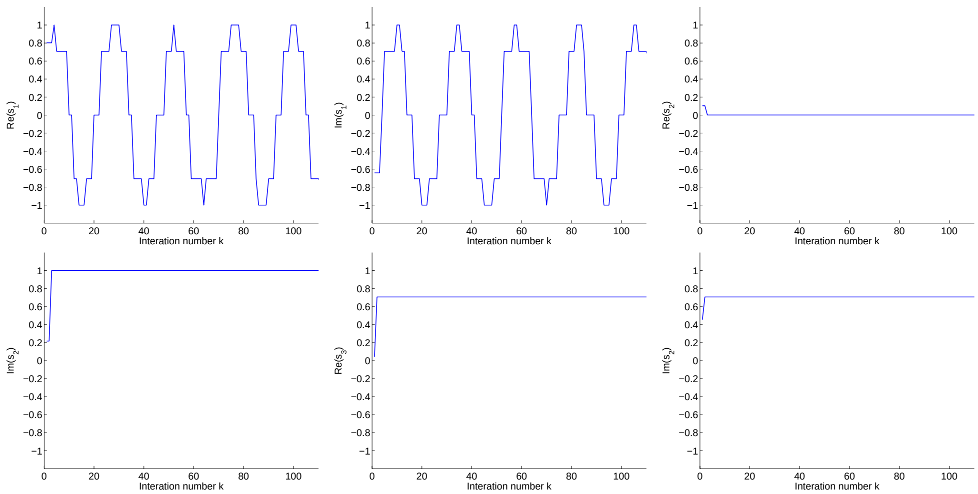

Figure 2: Trajectory of network (Ex.1) showing activities s_1, s_2 and s_3.

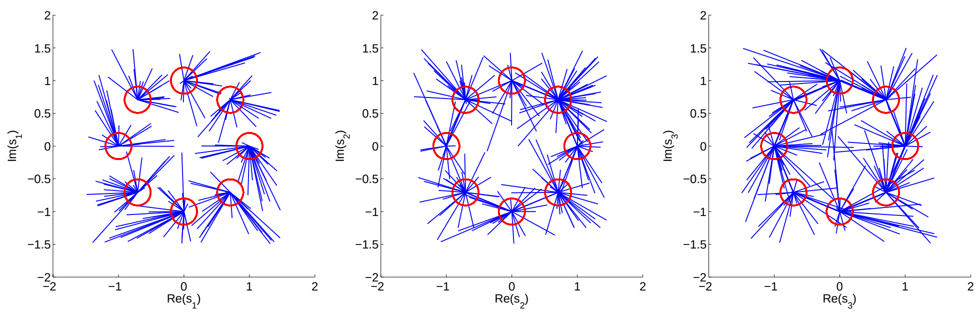

Figure 3: Global attractivity of network (Ex.2), the attractive regions are marked with a circle of s_1, s_2, and s_3.

The MVN network satisfies Theorem 1, but doesn't satisfy the stability condition of [9]. With the same initial value $s(0)$ as in Ex.1, the network will quickly converge to a fixed point $s^{\dagger} \approx [1, \ -i, \ 0.7071 - 0.7071i]^{*}$. Fig. 4 shows the complete convergence of network for 200 trajectories originating from randomly selected initial points in

$$Re(s(0)), Im(s(0)) \in [-1.5, 1.5].$$

Example 3. *In order to illustrate our theory, we also analyze example of MVN networks with different* K*,* N*. The factor* K *is chosen as* 8*,* 20*, or*

80. *The layer N is chosen as* 20 , 100*, or* 300*. We construct the following weight matrices for comparison: (1) We create a Hermitian matrix W_0 with all weight elements satisfying $Re(w_{ij}) \in (-100, 100)$ and $Im(w_{ij}) \in (-100, 100)$. (2) By removing the diagonal elements from a chosen W_0, we build two matrices W_1 and W_2 with different diagonal-element choice strategies respectively. The one is to set W_1 with all $w_{mm} \neq 0$ and $\arg(w_{mm}) \in (2\pi - \phi_0/2, 2\pi) \cup [0, \phi_0/2)$ (Thus, it satisfies Theorem 1). The second one is to set W_2 with at least one $\arg(w_{mm}) \in [\phi_0/2, 2\pi - \phi_0/2]$. (3) For two MVN networks built with W_1 and W_2 respectively, we choose the same initial point x_0 and observe whether or not the trajectory is convergent in fixed iteration rounds τ. Among 200 MVN networks with the fixed K and N, 100 networks are built by using the first strategy, and the second strategy is used for other 100 networks. Therefore, there are totally $3 \times 3 \times 100 \times 2 = 1800$ MVN networks to compare. Because all convergent samples observed cost no more than* 200 *rounds, and most of them will be convergent in less than* 50 *rounds, we choose to set τ as* 600 *here.*

As shown in Table 1, the trajectories observed in those MVN networks satisfying Theorem 1, are all completely stable. However, the test data also shows that even if an MVN network doesn't comply with Theorem 1, it may also be completely stable. Such phenomena may due to the strong interactivity among diagonal and no-diagonal elements, and could be a valuable topic in further research.

Example 4. *Here, we investigate the effect of diagonal elements on network performance. Here, we use a MVN neural network with 50 neurons, $K = 8$, and w_{mj} of W is defined as:*

$$w_{mj} = \begin{cases} c * (1.5 + 0.5i), & m = j; \\ m + j * i, & m > j; \\ m - j * i, & m < j. \end{cases}$$

where $c \in \{0\ 10\ 20\ 30\ 40\ 50\ 60\ 70\ 80\ 90\ 100\}$ is a parameter, thus 11 *networks are created. If $c = 0$, all diagonal elements of W will be zero. If $c \neq 0$, it is easy to identify that other* 10 *networks satisfy Theorem 1. Thus, we can compare our method with the one done in [9] by changing c. We concern two factors: average iteration number, and average energy function value based on off-diagonal weight. In some cases, such as associative memory design [11], more lower average energy function value based on off-diagonal*

Table 1: **Comparison for two diagonal-element choice strategies** *Here, "diagonal-element choice = 1"means that all* $w_{mm} \neq 0$ *and* $\arg(w_{mm}) \in (2\pi - \phi_0/2, 2\pi) \cup [0, \phi_0/2)$*, while "diagonal-element choice = 0"means that at least one* $\arg(w_{mm}) \in [\phi_0/2, 2\pi - \phi_0/2]$*. Stability Ratio = stable sample number observed / sample number tested.*

K	N	Diagonal-element Choice	Stability Ratio
8	20	0	63%
8	100	0	59 %
8	300	0	62 %
20	20	0	57%
20	100	0	64%
20	300	0	57%
80	20	0	12 %
80	100	0	61%
80	300	0	63%
8,20,or 80	20, 100, or 300	1	all equal to 100%

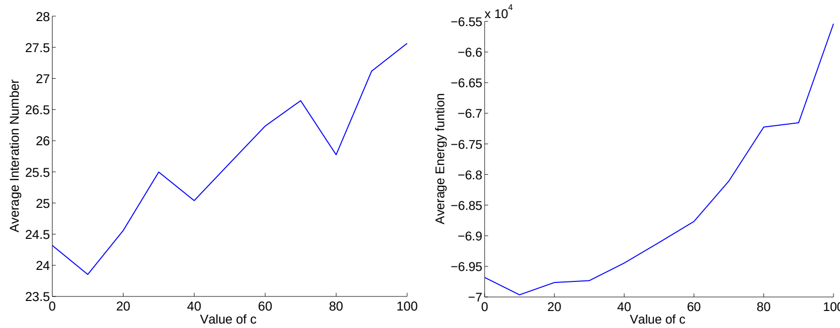

Figure 4: Changing of iteration numbers and energy function values based on off-diagonal weight according to different c.

weight means better performance. we randomly choose 500 *initial points* x_0 *satisfying* $Re(w_{ij}) \in (-10, 10)$ *and* $Im(w_{ij}) \in (-10, 10)$*. For each initial point, we record the iteration number and the energy function value based on off-diagonal weight. Therefore, there are totally* $11 \times 500 = 5500$ *tests to compare.* τ *is fixed as* 200*.*

As shown in Fig.4, the network with $c = 10$ has the lowest average iteration number and energy function value among all 11 networks, thus it outperforms the network with $w_{mm} = 0$. However, the test data also shows that as c increases, the performance tends to decrease. The reason is due to the interrelationship between two terms of (12). If we set $c = 0$, only the first term (13) takes effect. In proportion as c increases, the contribution coming from the second term (21) will rise. Consequently, by choosing appropriate diagonal elements of W, it is possible for us to increase network performance.

5. Conclusions

This paper investigates a class of discrete-time recurrent neural networks with multivalued neurons in asynchronous update mode. A new stability condition is presented here that extends previous results and allows weight matrix diagonal entries to be complex-valued. Simulations carried out have validated our theoretical findings.

Currently, the stability of MVN networks with non-Hermitian matrices presents a challenge. Our work can be looked as a beneficial and meaningful shaping to this direction. Another research topic is to investigate interactivity among diagonal and no-diagonal elements, which helps to find new stability property. On the other hand, because so many application based on MVNRNNs still use $w_{ii} = 0$, the modifications for the learning algorithm could be valuable.

Acknowledgment The authors wish to thank those anonymous reviewers and their helpful suggestions.

References

[1] Aizenberg, I., 2008. Solving the XOR and parity N problems using a single universal binary neuron. Soft Computing, 12(3), 215–222.

[2] Aizenberg, I., 2013. Hebbian and error-correction learning for complex-valued neurons. Soft Computing 17(2), 265–273.

[3] Aizenberg, I., Moraga, C., 2007. Multilayer feedforward neural network based on multi-valued neurons (MLMVN) and a backpropagation learning algorithm. Soft Computing 11, 169–183.

[4] Aizenberg, I., Paliy, D. V., Zurada, J. M., Astola, J. T., 2008. Blur identification by multilayer neural network based on multivalued neurons. IEEE Transactions on Neural Networks 19, 883–898.

[5] Aizenberg, I. N., Aizenberg, N. N., Vandewalle, J. P. L., 2000. Multi-Valued and Universal Binary Neurons: Theory, Learning, Applications. Kluwer Academic Publishers.

[6] Aizenberg, N. N., Aizenberg, I. N., 1992. CNN based on multi-valued neuron as a model of associative memory for gray-scale images. Proc. 2nd IEEE Int.Workshop Cellular Neural Netw. Their Appl., 36–41.

[7] Aizenberg, N. N., Ivaskiv, Y. L., 1977. Multiple-Valued Threshold Logic (in Russian). Kiev, Ukraine: Naukova Dumka Publisher House.

[8] Hirose, A. (Ed.), 2003. Complex-valued Neural Networks: Theories and Applications. World Scientific.

[9] Jankowski, S., Lozowski, A., Zurada, J. M., 1996. Complex-valued multistate neural associative memory. IEEE Transactions on Neural Networks 7(6), 1491–1496.

[10] Lee, D.-L., 2006. Improvements of complex-valued Hopfield associative memory by using generalized projection rules. IEEE Transactions on Neural Networks 17(5), 1341–1347.

[11] M*ü*ezzino*ğ*lu, M. K., G*ü*zelis, C., Zurada, J. M., 2003. A new design method for the complex-valued multistate Hopfield associative memory. IEEE Transactions on Neural Networks 14(4), 891–899.

[12] Tanaka, G., Aihara, K., 2009. Complex-valued multistate associative memory with nonliear multilevel functions for gray-level image reconstruction. IEEE Transactions on Neural Networks 20(9), 1463–1473.

[13] Zheng, P., 2014. Threshold complex-valued neural associative memory. IEEE Transactions on Neural Networks and Learning Systems 25(9), 1714–1718.

[14] Zhou, W., Zurada, J. M., 2009. A class of discrete time recurrent neural networks with multivalued neurons. Neurocomputing 72(16-18), 3782–3788.

[15] Zhou, W., Zurada, J. M., 2014. Stability condition for discrete time multi-valued recurrent neural networks in asynchronous update mode. 2014 International Joint Conference on Neural Networks (IJCNN), 3402–3405.