Stability Analysis of Autonomous Ratio-Memory Cellular Nonlinear Networks for Pattern Recognition

Su-Yung Tsai, *Student Member, IEEE*, Chi-Hsu Wang, *Fellow, IEEE*, and Chung-Yu Wu, *Fellow, IEEE*

***Abstract*—The stability analysis via the Lyapunov theorem for Autonomous Ratio-Memory Cellular Nonlinear Networks (ARMCNNs) is proposed. A conservative domain of attraction (DOA) is found from the stability analysis through a graphical method without complicated numerical analysis. The stability analysis shows that ARMCNNs can tolerate large ratio weight variations. This paper also presents the ARMCNN with self-feedback (SARMCNN) to overcome the problem of isolated neurons due to low correlation between neighboring neurons. The SARMCNN recognition rate (RR) is compared with other CNN constructed via the singular value decomposition technique (SVD-CNN).**

***Index Terms*—Cellular nonlinear network (CNN), domain of attraction (DOA), Lyapunov stability, ratio memory (RM), Hebbian learning rule.**

I. Introduction

INFORMATION storage is called associative memory if it permits the recall of information on the basis of a partial knowledge of its contents [1]. Some researchers believe that our memories correspond to attractors in the brain's huge phase space, since the human brain has more than 10^{12} neurons [2]. From this perspective, convergence to an appropriate attractor is called recognition. Hopfield proposed a model of a large fully connected network with symmetrical weights [3] that functions as an associative memory with the capability to recognize patterns. One of Hopfield's most important contributions [3] was to introduce the idea of an energy function into neural network theory [4]. However, the limitation of Hopfield's approach is that it requires fully connected symmetrical weights, which makes it difficult to connect wires in integrated circuits. Another limitation is the difficulty of finding the domain of attraction (DOA) in such a fully connected network. This problem can be addressed if such a decision boundary line for the DOA can be approximately located. Therefore, this paper aims to solve the fully connected problem and provide a conservative DOA which can be found easily through a graphical method without complicated numerical analysis.

Because of its connectivity, researchers consider the Cellular Nonlinear Network (CNN) [5] to be a potential architecture in future nano-electronic systems. To implement the associative memory by CNN, two key aspects of Hebb's postulate, locality and cooperativity [6], are fully exploited in the Ratio Memory CNN (RMCNN) or Autonomous-RMCNN (ARMCNN) [7]–[16]. Locality means that the change of synaptic efficacy only depends on local variables, which is the main point of CNN. Cooperativiy [6] means that the presynaptic and postsynaptic neurons must be active simultaneously for a synaptic weight to change, which is the learning rule in RMCNNs or ARMCNNs. The RM weight (or ratio weight) can be determined either through the elapsed time, as in [9]–[11], or can be determined without the elapsed time, as in [8], [17].

For pattern recognition, stability analysis is important to understand steady state values. However, previous studies regarding RMCNNs and ARMCNNs [7]–[16] do not guarantee that neuron states will converge to the equilibrium states. In [18], [19], the analysis of equilibrium points is mainly based on both the standard activation function [5] and the space-invariant templates whose center element is larger than 1.

While previous studies have proposed several different CNNs on associative memory [20]–[24], few have actually been implemented in analog VLSI [25], mainly due to complex mathematics required to make this possible. In literature, [20], [21] proposed a design method for the realization of associative memories through singular value decomposition technique (SVD). Another study [22], [23] showed the weights through solving linear matrix inequality (LMI) and generalized eigenvalue (GE) problems, whereas the work in [24] computed the CNN parameters by solving a set of linear equations via pseudo-inversion techniques. In particular, the CNNs in [20]–[24] require mathematical operations such as SVD, LMI, and pseudo-inversion. In [26], eigenvector methods are applied to extract pattern features of medical information. These operations are more complex than the simple methods proposed in this paper, and more difficult to implement in analog VLSI [25].

Section II describes the learning algorithm of ARMCNNs. Section III defines the dynamical state space equation for ARMCNN and discusses the ranges of all possible equilibrium points. Section IV provides the stability analysis. Section V discusses the effects of ratio weight variations. Section VI compares ARMCNNs with other CNN [20] constructed via the SVD technique (SVD-CNN). Future research efforts are summarized in Section VII.

II. ARMCNN

Fig. 1 depicts the basic neuron model of ARMCNN [7] at the ith row and jth column with the piecewise-linear (PWL) activation function shown in Fig. 2 where g is the neuron gain, x_{ij} is the neuron state voltage, y_{ij} is the neuron output current, $I_{\rm sat}$ is the output saturation current, and R_{ij}, C_{ij} are the resistor and capacitor associated with the neuron $C(i,j)$ respectively. w_{ijkl} is the ratio weight from neuron $C(k,l)$ to neuron $C(i,j)$.

Manuscript received March 10, 2009; revised June 23, 2009; accepted October 17, 2009. Date of publication February 05, 2010; date of current version August 11, 2010. This paper was recommended by Associate Editor M. Di Marco.

S.-Y. Tsai and C.-Y. Wu are with the Nanoelectronics and Gigascale Systems Laboratory, Department of Electronics Engineering & Institute of Electronics, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: suyung.ee93g@nctu.edu.tw).

C.-H. Wang is with the Department of Electrical Engineering, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: cwang@cn.nctu.edu.tw).

Digital Object Identifier 10.1109/TCSI.2009.2037450

1549-8328/$26.00 © 2010 IEEE

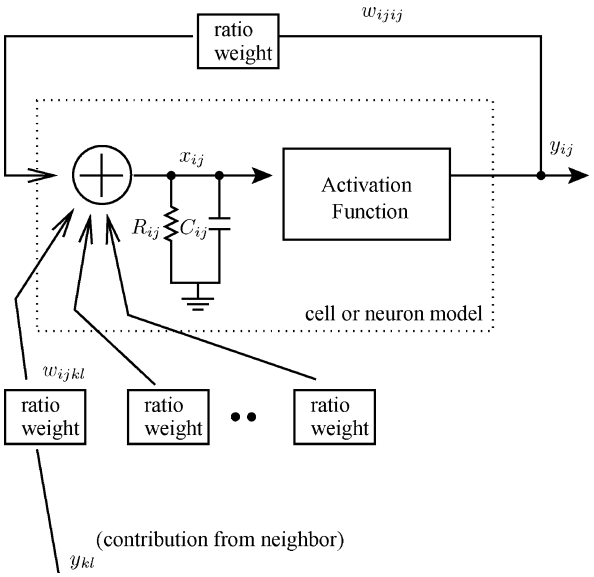

Fig. 1. The neuron model of ARMCNN located at the ith row and jth column.

The activation function is actually a voltage to current converter (V/I) with a transconductance value G_m of g/R_{ij} for the PWL case.

The operation procedures of a ARMCNN without self-feedback [7] can be divided into three periods: learning, elapsed, and recognition. The elapsed period is not required in [8], [17]. In the learning period, assume that the ARMCNN must learn m patterns. The learned weight z_{ijkl} from neuron $C(k,l)$ to neuron $C(i,j)$ can be determined by the Hebbian learning rule [6]

$$z_{ijkl} = \sum_{p=1}^{m} u_{ij}^{p} u_{kl}^{p} \tag{1}$$

where u_{ij}^{p} is the learning pixel image at the ith row and jth column of the pth pattern out of m input patterns, and u_{kl}^{p} is the learning pixel image at the kth row and lth column in the set of $N_{\mathrm{Ra}}^{0}(i,j)$. $N_{\mathrm{Ra}}^{0}(i,j)$ is the set of a Ra-neighborhood system without the neuron $C(i,j)$. The Ra-neighborhood system $N_{\mathrm{Ra}}(i,j)$ of the neuron $C(i,j)$ is defined as the set of all neurons including $C(i,j)$ and its neighboring neurons. $N_{\mathrm{Ra}}(i,j) = \{C(k,l) | 1 \le k \le M, 1 \le l \le N, \max(|k-i|, |l-j|) \le \mathrm{Ra}\}$ where Ra is an integer called the radius of the sphere of influence. Besides, the correlation between two neighboring neurons at $C(i,j)$ and $C(k,l)$ is positively correlative if

$$u_{ij}^{p} = u_{kl}^{p}, \quad \text{for } p = 1, \ldots, m \tag{2}$$

or negatively correlative if

$$u_{ij}^{p} = -u_{kl}^{p}, \quad \text{for } p = 1, \ldots, m. \tag{3}$$

For example, for a ARMCNN without self-feedback, assume four learning patterns with $+1$ for a black pixel image and -1 for a white pixel image. All possible learned weights z_{ijkl} are $\{+4, +2, 0, -2, 0, -4\}$. $+4$ occurs when the input image pair at $C(k,l)$ and $C(i,j)$ are positively correlative for all four input patterns. -4 occurs when the input image pair at $C(k,l)$ and $C(i,j)$ are negatively correlative for all four input patterns.

After learning all input patterns, the learned weight z_{ijkl} is transformed into the ratio weight w_{ijkl} [7]–[16]

$$w_{ijkl} = \frac{z_{ijkl}}{\sum_{C(k,l) \in N_R^0(i,j)} |z_{ijkl}|}.$$

For a ARMCNN with self-feedback (SARMCNN), simply replace $N_{\mathrm{Ra}}^{0}(i,j)$ with $N_{\mathrm{Ra}}(i,j)$. If $k = i, l = j$, then z_{ijij} is the self-feedback weight and w_{ijij} is the self-feedback ratio weight. The sum of the absolute values of the ratio weights for a SARMCNN is also equal to one.

In the elapsed period, the inevitable leakage current [11], [27] associated with the stored weight reduces the absolute value of z_{ijkl}. However, this leakage enhances the ratio weights whose absolute values are larger than the average of the absolute values of w_{ijkl} connected to $C(i,j)$ [12]. This feature enhancement effect was shown in [12]–[15] as a result of storing weights as ratios.

In the ARMCNN recognition period, the neuron or cell dynamic equation is

$$\begin{aligned} \frac{dx_{ij}(t)}{dt} &= \frac{-x_{ij}(t)}{R_{ij}C_{ij}} + \sum_{C(k,l) \in N_R(i,j)} \frac{w_{ijkl} y_{kl}(t)}{C_{ij}} \\ y_{kl}(t) &= \sigma(x_{kl}(t)) \end{aligned} \tag{4}$$

as shown in Fig. 1, where ARMCNN variables are represented as electrical signals such as voltages and currents [17], [27] for hardware implementation. The input initial states (input images) are defined as

$$-I_{\mathrm{sat}} R_{ij} \le x_{ij}(0) \le I_{\mathrm{sat}} R_{ij}. \tag{5}$$

If x_{ij} is normalized with respect to $I_{\mathrm{sat}} R_{ij}$, y_{ij} is normalized with respect to I_{sat}, and C_{ij} is unity, then (4) is expressed as follows,

$$R_{ij} \frac{d\tilde{x}_{ij}(t)}{dt} = -\tilde{x}_{ij}(t) + \sum_{C(k,l) \in N_R(i,j)} w_{ijkl} \tilde{\sigma}(\tilde{x}_{kl}) \tag{6}$$

where

$$\begin{aligned} \tilde{x}_{ij}(t) &= \frac{x_{ij}(t)}{I_{\mathrm{sat}} R_{ij}} \\ \tilde{\sigma}(\tilde{x}_{kl}) &= \frac{g}{2} \left(\left| \tilde{x}_{kl} + \frac{1}{g} \right| - \left| \tilde{x}_{kl} - \frac{1}{g} \right| \right) \\ -1 &\le \tilde{x}_{ij}(0) \le 1. \end{aligned}$$

Equation (6) is the standard CNN dynamic equation if $g = 1$ [5]. Compared with other works in [5], [20]–[24], the neuron gain g of a ARMCNN must be greater than unity for the PWL type activation function. And the sum of the absolute values of the ratio weights is equal to unity. Therefore, one difference between ARMCNNs and [5], [20]–[24] is that for ARMCNNs, the absolute value of the neuron state x_{ij} must be greater than or equal to $I_{\mathrm{sat}} R_{ij}/g$ with $g > 1$ to produce the binary outputs.

Furthermore, since the neuron $C(i,j)$ is a first order RC parallel circuit, and the maximum charging or discharging current to the neuron $C(i,j)$ is I_{sat} by the requirement of the ratio weights, the normalized neuron state is confined between $+1$ and -1. This implies that both the output and the state variables

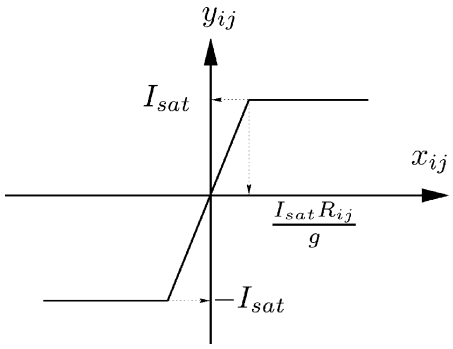

Fig. 2. The transfer characteristic of the PWL activation function, $\sigma(\cdot)$.

are confined for ARMCNN, which favors the analog circuit design [17], [28].

In [20], any equilibrium point in the total saturation region is asymptotically stable if g is equal to unity. However, [20] does not indicate that it is to have binary outputs given any activation function. Also, the resulting weights in [20] could be any real number, which may not be feasible to implement in hardware circuits. On the other hand, the ratio weights found in ARMCNNs are fixed ratio numbers that can be easily realized in hardware circuits using replicas of current mirrors [17], [28].

This paper assumes that R_{ij} and C_{ij} are constant for all neurons, $R_{ij} = R, C_{ij} = C$. This is because C_{ij} only affects the convergent speed, and the mismatch of R_{ij} can be combined with the mismatch of g to form an effective g as g_{eff}, which is

$$I_{\text{sat}} \times (R_{ij} + \Delta R_{ij})/g = I_{\text{sat}} \times R_{ij}/g_{\text{eff}}.$$

The algorithm for determining the ratio weights in ARMCNNs (or SARMCNNs), which requires no elapsed time, is summarized as follows: [17]

1) Find the weights z_{ijkl} of (1) after m patterns are learned.
2) Set
$$G_{\max} = \sum_{p=1}^{m} \left|u_{ij}^{p}\right| \left|u_{kl}^{p}\right|,$$
where u_{ij}^{p} and u_{kl}^{p} are defined in (1).
3) Compare $|z_{ijkl}|$ with $G_{\max}$. The weight z_{ijkl} with its absolute value equal to $G_{\max}$ will be kept. On the contrary, the weight with its absolute value less than $G_{\max}$ will be set to zero.
4) Transform the connecting weights of neuron $C(i,j)$ as
$$w_{ijkl} = \frac{\mathbf{Sgn}(z_{ijkl})}{PN_{N_{\text{Ra}}^{0}(i,j)}}, \qquad \text{if } |z_{ijkl}| = G_{\max},$$
where $PN_{N_{\text{Ra}}^{0}(ij)}$ is the number of preserved weights in $N_{\text{Ra}}^{0}(i,j)$ with respect to neuron $C(i,j)$. $w_{ijkl} = 0$, if $|z_{ijkl}| < G_{\max}$.
For SARMCNNs, simply replace $N_{\text{Ra}}^{0}(i,j)$ with $N_{\text{Ra}}(i,j)$ and if $k = i, l = j$, then z_{ijij} is the self-feedback weight.

To simplify wire connections, only neighboring neurons located on horizontal and vertical positions are connected, that is, up, down, left, right. Neurons located on diagonal sites are not connected. The algorithm without the elapsed time is used in this paper. Therefore, the possible ratio weights are $\pm 1/4, \pm 1/3, \pm 1/2, \pm 1$ for a ARMCNN without self-feedback, whereas for a SARMCNN, the possible ratio weights are $\pm 1/5, \pm 1/4, \pm 1/3, \pm 1/2, \pm 1$. The resultant ratio weights generate many small subsystems. Each subsystem has two binary equilibrium points in the form of $(\pm I_{\text{sat}}R, \ldots, \pm I_{\text{sat}}R)$. The choice of the sign is determined as follows. Set the equilibrium state of the first neuron $C(i,j)$ to $I_{\text{sat}}R(-I_{\text{sat}}R)$. If a negatively correlative input image is found between $C(k,l)$ and $C(i,j)$, then set the equilibrium state of the connected neuron $C(k,l)$ to $-I_{\text{sat}}R(I_{\text{sat}}R)$. Otherwise, if positively correlative, then set the equilibrium state of the connected neuron $C(k,l)$ to $I_{\text{sat}}R(-I_{\text{sat}}R)$. This is true for any number of neurons [7]. For example, Fig. 3 shows that two equilibrium points are $(I_{\text{sat}}R, I_{\text{sat}}R, -I_{\text{sat}}R)$ and $(-I_{\text{sat}}R, -I_{\text{sat}}R, I_{\text{sat}}R)$. The thick bond in Fig. 3 is due to a negatively correlative input image at the two connected neurons and the thin bond is due to a positively correlative input image at the two connected neurons. The dynamic equations of Fig. 3 are

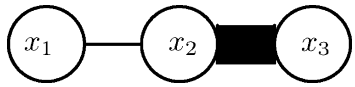

Fig. 3. A 1-D three-neuron (3N) subsystem with two binary equilibrium points of black-black-white and white-white-black, or $(x_1(\infty), x_2(\infty), x_3(\infty)) = \pm(I_{\text{sat}}R, I_{\text{sat}}R, -I_{\text{sat}}R)$.

$$\begin{aligned}
\dot{x_1} &= -\frac{1}{R_1C_1}x_1 + \frac{1}{C_1}\sigma(x_2) \\
\dot{x_2} &= -\frac{1}{R_2C_2}x_2 + \frac{0.5}{C_2}\sigma(x_1) - \frac{0.5}{C_2}\sigma(x_3) \\
\dot{x_3} &= -\frac{1}{R_3C_3}x_3 - \frac{1}{C_3}\sigma(x_2).
\end{aligned} \tag{7}$$

Assume $R_1 = R_2 = R_3 = R$. If we let

$$\begin{aligned}
x_{\text{eq}} &= [\, x_1(\infty) \quad x_2(\infty) \quad x_3(\infty)\,]^T \\
&= [\, I_{\text{sat}}R \quad I_{\text{sat}}R \quad -I_{\text{sat}}R\,]^T
\end{aligned}$$

then the right hand side of (7) becomes

$$\begin{aligned}
-\frac{1}{R_1C_1}I_{\text{sat}}R_1 + \frac{1}{C_1}I_{\text{sat}} &= 0 \\
-\frac{1}{R_2C_2}I_{\text{sat}}R_2 + \frac{0.5}{C_2}I_{\text{sat}} - \frac{0.5}{C_2}(-I_{\text{sat}}) &= 0 \\
-\frac{1}{R_3C_3}(-I_{\text{sat}}R_3) - \frac{1}{C_3}I_{\text{sat}} &= 0.
\end{aligned}$$

Therefore $(I_{\text{sat}}R, I_{\text{sat}}R, -I_{\text{sat}}R)$ is indeed an equilibrium point.

Fig. 4 shows an example of survived ratio weights [7] after learning the three Chinese characters in Fig. 5. There are one 2-D 18N subsystem (top two rows), one 2-D 25N subsystem (indicated with an arrow), six 1-D 2N subsystems, one 2-D 8N subsystem and two 1-D 9N systems (bottom two rows). Two binary equilibrium points for the one 2-D 18N subsystem (top two rows) are $(I_{\text{sat}}R, \ldots, I_{\text{sat}}R)$ and $-(I_{\text{sat}}R, \ldots, I_{\text{sat}}R)$, since the connecting bonds are all positively correlative. The thick bond in Fig. 4 is due to a negatively correlative input image at

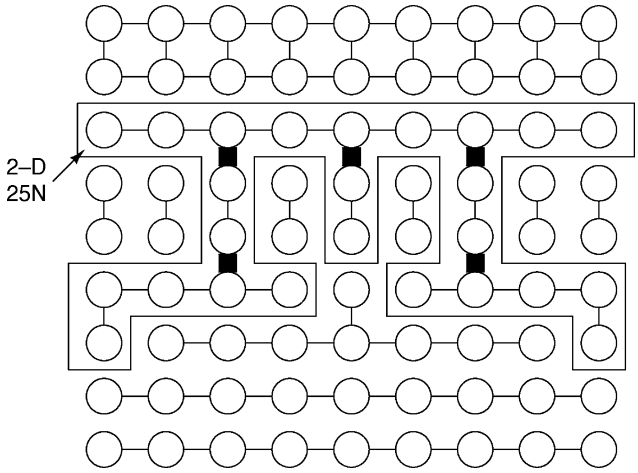

Fig. 4. The survived ratio weights in the 9×9 ARMCNN without self-feedback after learning three Chinese characters ONE, TWO, and FOUR. These three Chinese characters are shown in Fig. 5.

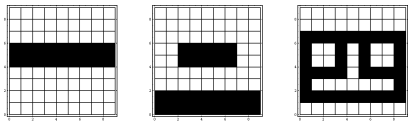

Fig. 5. (a) Chinese character ONE. (b) Chinese character TWO. (c) Chinese character FOUR.

the two connected neurons and the thin bond is due to a positively correlative input image at the two connected neurons.

III. Dynamic State Space Equations for ARMCNN

To perform the ARMCNN stability analysis, it is necessary to represent the ARMCNN with the following dynamic state space equations:

$$\dot{x} = -Cx + T\sigma(x) \quad \text{and} \quad y = \sigma(x) \tag{8}$$

where $x \in R^n$ is the state vector, y is the output vector, and $T = [T_{ij}] \in R^{n\times n}$ is the connection matrix. $C = \mathbf{diag}[(R_1C_1)^{-1}, \ldots, (R_nC_n)^{-1}]$, with $R_i, C_i > 0$ for $i = 1, \ldots, n$, and $\sigma(x) = [\sigma(x_1), \ldots, \sigma(x_n)]^T$ represents the activation function. The input image, that is, the initial state, satisfies $|x_i(0)| \le I_{\text{sat}}R_i$ for $i = 1, \ldots, n$. $n = M \times N$ for a $M \times N$ pattern. Note that this study represents the two-dimensional indices of C_{ab} and R_{ab} by one dimensional C_i and R_i using the following rule:

$$i = (a-1) \times N + b.$$

For the ratio memory requirement,

$$\sum_{k=1}^{n} |T_{ik}| = \frac{1}{C_i}, \quad i = 1, \ldots, n \tag{9}$$

where T_{ik} is the ratio weight from neuron k to neuron i multiplied by $1/C_i$. If PWL is used for the activation function, $g > 1$ is required for the ARMCNN to have binary equilibrium points. Theorem 2 in Section IV explains this requirement.

Fig. 6. A 1-D five-neuron (5N) subsystem having two binary equilibrium points $(x_1(\infty), x_2(\infty), x_3(\infty), x_4(\infty), x_5(\infty))$ as $\pm(I_{\text{sat}}R, I_{\text{sat}}R, I_{\text{sat}}R, I_{\text{sat}}R, I_{\text{sat}}R)$. The definitions of the thin bonds between neurons are the same as in Fig. 3.

The equilibrium points of (8) can be obtained by solving

$$Cx = T\sigma(x). \tag{10}$$

In general, we can have q equilibrium points, $x^1_{\text{eq}}, \ldots, x^q_{\text{eq}}$. Further, the ratio memory requirement shown in (9) confines the ranges of all possible equilibrium states as:

$$\left|x^{mul}_{\text{eq},i}\right| \le I_{\text{sat}}R_i \quad \text{for } i = 1, \ldots, n, \quad mul = 1, \ldots, q$$

where $x^{mul}_{\text{eq},i}$ is the ith equilibrium neuron state for the multh equilibrium point. This is because, for the ith neuron $(i = 1, \ldots, n)$

$$\frac{1}{R_iC_i}x^{mul}_{\text{eq},i} = \sum_{k=1}^{n} T_{ik}\sigma\left(x^{mul}_{eq,k}\right) \le I_{\text{sat}} \sum_{k=1}^{n} |T_{ik}| = \frac{I_{\text{sat}}}{C_i}$$
$$\Rightarrow |x^{mul}_{\text{eq},i}| \le I_{\text{sat}}R_i \quad \text{for } mul = 1, \ldots, q.$$

Because q is possible to be greater than 2, some undesired equilibrium points, that is, spurious memory points, may exist besides two binary equilibrium points. To show spurious memory points, consider the 1-D 5N subsystem example shown in Fig. 6, with the corresponding circuit implementation in Fig. 7. The neuron resistor (R), capacitor (C) and voltage to current converter (V/I) are as defined in Figs. 1 and 2. For each synapse connection between two neighboring neurons, two voltage to current converters and two current mirrors are required. If $g = 2$, then one possible spurious memory point is at $x^1_{\text{eq}} = [0.877, 0.4389, -0.0611, -0.5611, -1]^T I_{\text{sat}}R$. Another possible spurious memory point is at $x^2_{\text{eq}} = [1, 0.5, 0, -0.5, -1]^T I_{\text{sat}}R$. These spurious memory points are obtained by solving (10).

IV. ARMCNN Stability Analysis

Whether or not the initial state can successfully converge to the desired equilibrium point determines recognition performance. Therefore it is important to find the domain of attraction (DOA), or at least a conservative DOA. Without loss of generality, R_i, C_i and I_{sat} in (8) are normalized to unities. This is because these parameters only affect the convergent time which is proportional to $1/(R_iC_i)$, and the state values and output currents can be normalized as $x_i/(I_{\text{sat}}R_i)$ and y_i/I_{sat} as shown in (6). The normalized binary equilibrium point is expressed as:

$$\text{Eqp} = (\ \text{eq}_1 \quad \text{eq}_2 \quad \cdots \quad \text{eq}_n\ )$$

where eq_i is in the form of ± 1. Consider the behavior around one normalized binary equilibrium point using new state variable z_i. The new state variable z_i with respect to the normalized binary equilibrium point is defined as

$$z_i = x_i - \text{eq}_i \quad \text{for } i = 1, \ldots, n. \tag{11}$$

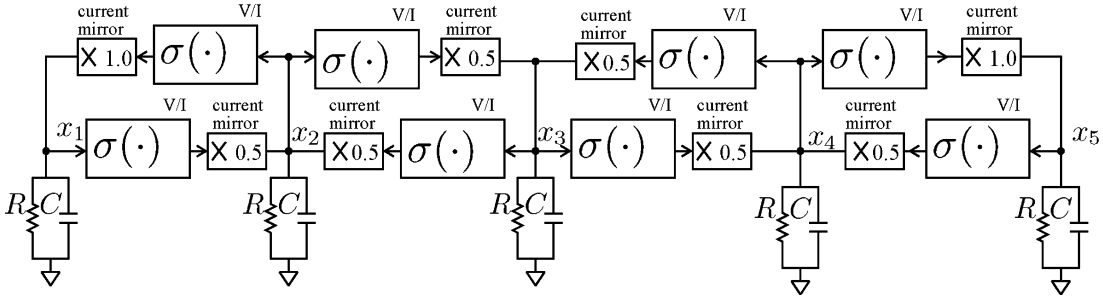

Fig. 7. Circuit implementation of a 1-D 5N ARMCNN subsystem in Fig. 6. For each synapse connection between two neightboring neurons, two voltage to current converters (V/I) and two current mirrors are required.

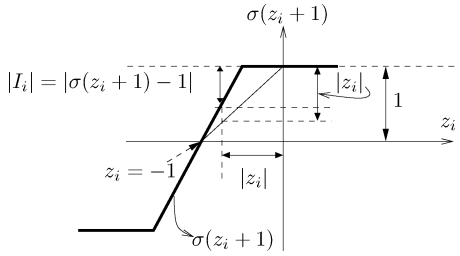

Fig. 8. The shifted PWL activation function, $\sigma(z_i + 1)$.

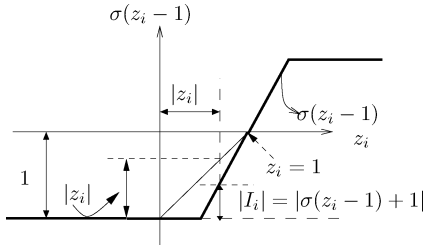

Fig. 9. The shifted PWL activation function, $\sigma(z_i - 1)$.

For instance, if the normalized equilibrium state in the ith neuron is ± 1, then

$$z_i = x_i \mp 1.$$

After this transformation from x to z, the new system has a equilibrium point at the origin and is represented as

$$\dot{z} = -z + g(z) \tag{12}$$

where

$$\begin{aligned} g(z) &= [g_1(z) \cdots g_n(z)]^T \\ g_i(z) &= \left[\sum_{k=1}^{n} T_{ik}\sigma(z_k + \mathrm{eq}_k)\right] - \mathrm{eq}_i. \end{aligned} \tag{13}$$

Figs. 8 and 9 show the shifted PWL activation functions. For these shifted activation functions in Figs. 8 and 9, (14) and (15), must be satisfied in the following stability analysis.

$$|\sigma(z_i+1)-1| \le |z_i|, \quad \text{for } |z_i| < 1, \sigma(1) = 1$$

and

$$\lim_{z_i \to 0} \frac{|\sigma(z_i+1)-1|}{|z_i|} = 0, \quad \text{for } i = 1, \ldots, n \tag{14}$$

$$|\sigma(z_i-1)+1| \le |z_i|, \quad \text{for} |z_i| < 1, \sigma(-1) = -1$$

and

$$\lim_{z_i \to 0} \frac{|\sigma(z_i-1)+1|}{|z_i|} = 0, \quad \text{for } i = 1, \ldots, n. \tag{15}$$

Theorem 1: For the PWL activation function defined in Fig. 2, (14) and (15) are satisfied if and only if the slope g is greater than 1.

Proof: From Fig. 2, it is obvious that $\sigma(1/g) = 1$ for the normalized ARMCNN. The fact that g must be greater than 1 can be argued from the following analysis.

1) If $g < 1$, then $1/g > 1$. The PWL activation function in Fig. 2 is monotonically increasing from $x = 0$ to $x = 1/g > 1$. Therefore,

$$\sigma(0) = 0 < \sigma(1) < \sigma(1/g) = 1$$

which does not meet (14), that is, $\sigma(1) = 1$.

2) If $g = 1$, then

$$|\sigma(z_i+1)-1| = |z_i+1-1| = |z_i| \quad \text{for} \quad -1 \le z_i \le 0$$

therefore,

$$\lim_{z_i \to 0-} \frac{|\sigma(z_i+1)-1|}{|z_i|} = \lim_{z_i \to 0-} \frac{|z_i|}{|z_i|} = 1, \quad \text{for } i = 1, \ldots, n,$$

which does not meet (14), that is,

$$\lim_{z_i \to 0} \frac{|\sigma(z_i+1)-1|}{|z_i|} = 0, \quad \text{for } i = 1, \ldots, n.$$

3) If $g > 1$, then $|z_i| < 1$ is divided into the following two ranges to show the PWL activation function defined in Fig. 2 satisfies (14).

Range 1: If $-1 < z_i < -1 + 1/g < 0$, then

$$\begin{aligned} -g < gz_i &< -g + 1 \\ |\sigma(z_i+1)-1| &= |g \times (z_i+1) - 1| \\ &= 1 - gz_i - g > 0 \\ |z_i| &= -z_i \\ |z_i| - |\sigma(z_i+1)-1| &= -z_i - 1 + gz_i + g \\ &= (g-1) + z_i(g-1) \\ &= (g-1)(z_i+1) > 0 \end{aligned}$$

$$\Rightarrow |\sigma(z_i+1)-1| < |z_i| \text{ for } -1 < z_i < -1 + 1/g < 0.$$

Range 2: If $-1+1/g \le z_i < 1$, then

$$\begin{aligned}
&\Rightarrow 1/g \le z_i + 1 < 2\\
&\Rightarrow \sigma(z_i+1) = 1\\
&\Rightarrow |\sigma(z_i+1)-1| = 0 \le |z_i|\\
&\Rightarrow \lim_{z_i\to 0}\frac{|\sigma(z_i+1)-1|}{|z_i|} = \lim_{z_i\to 0}\frac{0}{|z_i|} = 0.
\end{aligned}$$

In a similar way, $|z_i| < 1$ is divided into the following two ranges to show that the PWL activation function defined in Fig. 2 satisfies (15).

Range 3: If $0 < 1 - 1/g < z_i < 1$, then

$$\begin{aligned}
g - 1 < gz_i &< g\\
|\sigma(z_i-1)+1| &= |g\times(z_i-1)+1|\\
&= gz_i - g + 1 > 0\\
|z_i| &= z_i\\
|z_i| - |\sigma(z_i-1)+1| &= z_i - 1 - gz_i + g\\
&= z_i(1-g) + (g-1)\\
&= (g-1)(1-z_i) > 0
\end{aligned}$$

$\Rightarrow |\sigma(z_i-1)+1| < |z_i|$ for $0 < 1 - 1/g < z_i < 1$.

Range 4: If $-1 < z_i \le 1 - 1/g$, then

$$\begin{aligned}
&\Rightarrow -2 < z_i - 1 \le -1/g\\
&\Rightarrow \sigma(z_i-1) = -1\\
&\Rightarrow |\sigma(z_i-1)+1| = 0 \le |z_i|\\
&\Rightarrow \lim_{z_i\to 0}\frac{|\sigma(z_i-1)+1|}{|z_i|} = \lim_{z_i\to 0}\frac{0}{|z_i|} = 0
\end{aligned}$$

For the PWL activation function, the above case (3) shows that if $g > 1$ then (14) and (15) are satisfied. And the above case (1) and case (2) show that for the PWL activation function, if (14) and (15) are satisfied, then $g < 1$ and $g = 1$ are not possible. Therefore, for the PWL activation function defined in Fig. 2, (14) and (15) are satisfied if and only if the slope g is greater than 1. ■

Let $\phi(t;z)$ be the solution of (12) that starts at initial state z at time $t = 0$. And $\phi(t;z)$ is defined for $t \ge 0$. The DOA is defined by

$$R_A = \{z | \phi(t;z) \to z_{\rm eq} \text{ as } t \to \infty\}.$$

The domain of attraction (DOA) required to guarantee the asymptotic stability of a dynamical system in (12) around the origin can be found in [29]. First, determine a domain D about the origin where $\dot{V}(z)$ is negative definite and a constant $c > 0$ such that

$$\Omega_c = \{z \in R^n | V(z) \le c\}$$

is a subset of D. In other words, if there is a closed and bounded region

$$\Omega_c = \{z \in R^n | V(z) \le c\}$$

such that

1) $\dot{V}(z) < 0$ for all $z \in \Omega_c$ except z_e
2) $\dot{V}(z_e) = 0, V(z_e) = 0, z_e \in \Omega_c$
3) $V(z)$ is positive definite for all $z \in \Omega_c$

then $z = z_e$ is asymptotically stable and Ω_c is a conservative Domain of Attraction (DOA).

The ARMCNN defined in (8) is generated after performing the algorithm in Section II without elapsed time. The ratio weights of this ARMCNN satisfy (9), and the activation functions of each neuron are required to satisfy (14) and (15). Theorem 2 shows that the resultant ARMCNN converges to one of the binary equilibrium points.

Theorem 2: ARMCNN stability analysis with a conservative Domain of Attraction (DOA). For the normalized ARMCNN $(R_i = R = 1, C_i = C = 1, I_{\rm sat} = 1)$ defined in (12) with the activation function satisfying (14) and (15), there exists a conservative domain of attraction (DOA), i.e., $\|z\| < r$, so that the ARMCNN will converge to one of the normalized binary equilibrium points for $\|z\| < r$.

Proof: We need to first prove the asymptotic stability of (12), that is,

$$\dot{z} = -z + g(z).$$

Define a positive quadratic Lyapunov function

$$V(z) = z^T P z.$$

Choosing $P = 0.5\mathbf{1}_n$, where $\mathbf{1}_n$ is the $n \times n$ identity matrix, we have

$$\begin{aligned}
\dot{V}(z) &= -2z^T P z + 2z^T P g(z) = -z^T z + z^T g(z)\\
&= -\|z\|^2 + z^T g(z) \le -\|z\|^2 + \|z\|\|g(z)\|.
\end{aligned}$$

It is obvious that $\|g(z)\|$ determines the sign of the derivative of $V(z)$. To find $\|g(z)\|$, the range of $g_i(z)$ is analyzed first from (13) as

$$\begin{aligned}
g_i(z) &= \left[\sum_{k=1}^{n} T_{ik}\sigma(z_k + {\rm eq}_k)\right] - {\rm eq}_i\\
&= \sum_{k=1}^{n} T_{ik}\left[\sigma(z_k + {\rm eq}_k) - {\rm eq}_k\right] = \sum_{k=1}^{n} T_{ik} I_k \quad (16)
\end{aligned}$$

where

$$I_k \equiv \sigma(z_k + {\rm eq}_k) - {\rm eq}_k.$$

The absolute values of various I_k are shown in Figs. 8 and 9.

The reason for the validity of (16) is as follows. Denoting these two opposite equilibrium states as S^1 and S^{-1}: the index set of all neurons with the normalized equilibrium states equal to 1 is S^1 and the index set of all neurons with the normalized equilibrium states equal to -1 is S^{-1}, then

$$\begin{aligned}
&\sum_{k=1}^{n} T_{ik}[\sigma(z_k + {\rm eq}_k) - {\rm eq}_k]\\
&\quad = \sum_{k\in S^1} T_{ik}[\sigma(z_k+1) - 1] + \sum_{k\in S^{-1}} T_{ik}[\sigma(z_k-1)+1]\\
&\quad = \sum_{k\in S^1} T_{ik}\sigma(z_k+1) - \sum_{k\in S^1} T_{ik}\\
&\qquad + \sum_{k\in S^{-1}} T_{ik}\sigma(z_k-1) + \sum_{k\in S^{-1}} T_{ik}. \quad (17)
\end{aligned}$$

If the equilibrium state in the ith neuron is 1, then

$$\begin{aligned} T_{ik} &> 0, \quad \text{for } k \in S^1 \text{ (positively correlative)} \\ T_{ik} &< 0, \quad \text{for } k \in S^{-1} \text{ (negatively correlative)} \end{aligned}$$

and from (9), we have

$$\sum_{k\in S^1} -T_{ik} + \sum_{k\in S^{-1}} T_{ik} = -1,$$

so (17) is expressed as

$$\begin{aligned} &\sum_{k=1}^{n} T_{ik}[\sigma(z_k + \mathrm{eq}_k) - \mathrm{eq}_k] \\ &\qquad = \sum_{k\in S^1} T_{ik}\sigma(z_k+1) + \sum_{k\in S^{-1}} T_{ik}\sigma(z_k-1) - 1 \end{aligned}$$

which is (13).

If the equilibrium state in the ith neuron is -1, then

$$\begin{aligned} T_{ik} &< 0, \quad \text{for } k \in S^1 \text{ (positively correlative)} \\ T_{ik} &> 0, \quad \text{for } k \in S^{-1} \text{ (negatively correlative)} \end{aligned}$$

and from (9), we have

$$\sum_{k\in S^1} -T_{ik} + \sum_{k\in S^{-1}} T_{ik} = 1,$$

so (17) is expressed as

$$\begin{aligned} &\sum_{k=1}^{n} T_{ik}[\sigma(z_k + \mathrm{eq}_k) - \mathrm{eq}_k] \\ &\qquad = \sum_{k\in S^1} T_{ik}\sigma(z_k+1) + \sum_{k\in S^{-1}} T_{ik}\sigma(z_k-1) + 1 \end{aligned}$$

which is (13).

Therefore, (16) is proved. From (16) and equations of (14) and (15), we have

$$g_i(z) = \sum_{k=1}^{n} T_{ik} I_k, \text{ with } \lim_{z_k\to 0} \frac{|I_k|}{|z_k|} = 0, \quad \text{for } k = 1, \ldots, n. \tag{18}$$

This implies that for any ε $(0 < \varepsilon < 1)$, there exists an r_k, such that

$$|I_k| < \varepsilon |z_k| \quad \text{for } |z_k| < r_k, \quad k = 1, \ldots, n.$$

In fact r_k can be found from Figs. 8 and 9 given the ε value. Let

$$r = \min(r_1, r_2, \ldots, r_n) \tag{19}$$

then

$$\text{if } |z_k| < r \qquad \Rightarrow \qquad |I_k| < \varepsilon |z_k|, \quad \text{for } k = 1, \ldots, n.$$

or

$$\text{if } \|z\| < r \Rightarrow |z_k| < r \Rightarrow |I_k| < \varepsilon |z_k|, \quad \text{for } k = 1, \ldots, n.$$

From (9), we can see that

$$\sum_{k=1}^{n} |T_{ik} z_k| \le \sqrt{|z_1|^2 + |z_2|^2 + \ldots + |z_n|^2} = \|z\|. \tag{20}$$

From (18), (19) and (20) and let

$$I_i^* \equiv \sum_{k=1}^{n} |T_{ik} I_k|,$$

the upper bound of the norm of $g(z)$ can be found as

$$\begin{aligned} \|g(z)\| &= \sqrt{g_1(z)^2 + \cdots + g_n(z)^2} \\ &\le \sqrt{(I_1^*)^2 + \cdots + (I_n^*)^2} \\ &= \sqrt{\left(\sum_{k=1}^{n} |T_{1k} I_k|\right)^2 + \cdots + \left(\sum_{k=1}^{n} |T_{nk} I_k|\right)^2} \\ &< \sqrt{\left(\sum_{k=1}^{n} |T_{1k} \varepsilon z_k|\right)^2 + \cdots + \left(\sum_{k=1}^{n} |T_{nk} \varepsilon z_k|\right)^2} \\ &= \sqrt{\varepsilon^2 \left(\sum_{k=1}^{n} |T_{1k} z_k|\right)^2 + \cdots + \varepsilon^2 \left(\sum_{k=1}^{n} |T_{nk} z_k|\right)^2} \\ &\le \sqrt{\varepsilon^2 \|z\|^2 n} = \varepsilon\sqrt{n}\|z\| = \gamma\|z\|, \quad \text{for } \|z\| < r \end{aligned} \tag{21}$$

where $\gamma \equiv \varepsilon\sqrt{n}$. Using (21), we have

$$\begin{aligned} \dot{V}(z) &\le -\|z\|^2 + \|z\|\|g(z)\| \\ &\le (\varepsilon\sqrt{n} - 1)\|z\|^2 = (\gamma - 1)\|z\|^2 \quad \text{for} \|z\| < r. \end{aligned}$$

If $\gamma < 1$, then $\dot{V}(z) < 0$, which will yield an asymptotic stable ARMCNN defined in (12). Since

$$\gamma \equiv \varepsilon\sqrt{n}$$

we have

$$0 < \gamma = \varepsilon\sqrt{n} < 1 \Rightarrow \varepsilon < \frac{1}{\sqrt{n}}. \tag{22}$$

Therefore, by choosing a proper ε from the above (22), we can find r from (19) to yield a conservative DOA so that

$$V(z) = z^T P z = 0.5\|z\|^2 < 0.5 r^2,$$

that is, $\|z\| < r$ is the conservative DOA. ■

The above derivation is applicable to any activation function satisfying (14) and (15) that includes the PWL function of gain greater than unity, as Theorem 1 shows. Furthermore, only the unity sum of the absolute values of the ratio weights is required. The ratio weights can vary a lot as long as the signs of the corresponding ratio weights in the above proof are kept. Therefore, the ARMCNN can converge to the correct equilibrium states even when there are deviations in the ratio weights due to different VLSI processes.

Example 1: Ratio weight variations do not change the equilibrium points. For the 1-D 3N system in (7) of Fig. 3 under $R_i = 1, C_i = 1, i = 1, 2, 3$, we see that

$$\begin{aligned}\dot{x}_1 &= -x_1 + \sigma(x_2)\\ \dot{x}_2 &= -x_2 + 0.5\sigma(x_1) - 0.5\sigma(x_3)\\ \dot{x}_3 &= -x_3 - \sigma(x_2).\end{aligned} \tag{23}$$

It has been shown in Section II that the $(1, 1, -1)$ is an equilibrium point. If the process variation perturbs the system into

$$\begin{aligned}\dot{x}_1 &= -x_1 + \sigma(x_2)\\ \dot{x}_2 &= -x_2 + 0.7\sigma(x_1) - 0.3\sigma(x_3)\\ \dot{x}_3 &= -x_3 - \sigma(x_2)\end{aligned} \tag{24}$$

then after substitution of the original equilibrium point of $(1, 1, -1)$ into the right hand side of (24), we have

$$\begin{aligned}-1 + 1 &= 0\\ -1 + 0.7 - 0.3(-1) &= 0\\ -(-1) - 1 &= 0.\end{aligned}$$

Therefore $(1, 1, -1)$ is still an equilibrium point.

Example 2: Finding a DOA for a three-neuron system in Fig. 3. The new state variable z_i with respect to the normalized equilibrium point $(1, 1, -1)$ is defined as

$$\begin{aligned}z_1 &= x_1 - 1\\ z_2 &= x_2 - 1\\ z_3 &= x_3 + 1.\end{aligned}$$

After this transformation from x to z, the new system exhibits equilibrium at the origin and is described as

$$\dot{z} = -z + g(z) \tag{25}$$

where

$$\begin{aligned}g(z) &= [g_1(z)g_2(z)g_3(z)]^T\\ g_1(z) &= \sigma(z_2 + 1) - 1 = I_2\\ g_2(z) &= 0.5\sigma(z_1 + 1) - 0.5\sigma(z_3 - 1) - 1\\ &= 0.5(\sigma(z_1 + 1) - 1) - 0.5(\sigma(z_3 - 1) + 1)\\ &= 0.5I_1 - 0.5I_3\\ g_3(z) &= -(\sigma(z_2 + 1) - 1) = -I_2.\end{aligned}$$

Assume that the activation function is the sinusoidal type defined as

$$\begin{aligned}\sigma(z_i) &= \sin\left(\frac{\pi}{2}z_i\right), \quad \text{if } |z_i| \le 1\\ \sigma(z_i) &= 1, \quad \text{if } z_i \ge 1\\ \sigma(z_i) &= -1, \quad \text{if } z_i \le -1.\end{aligned} \tag{26}$$

Also assume that each neuron has the same activation function of (26). Fig. 10 shows the shifted sinusoidal activation function of $\sigma(z_1 + 1)$. Equation (26) satisfies the activation function requirement in (14) and (15). To show this, consider first $0 < z_1 < 1$. It follows that

$$\begin{aligned}&\Rightarrow z_1 + 1 > 1\\ &\Rightarrow \sigma(z_1 + 1) = 1\\ &\Rightarrow |\sigma(z_1 + 1) - 1| = |I_1| = 0 \le |z_1| \qquad (27)\\ &\Rightarrow \lim_{z_1 \to 0^+} \frac{|\sigma(z_1 + 1) - 1|}{|z_1|} = \lim_{z_1 \to 0^+} \frac{0}{|z_1|} = 0. \qquad (28)\end{aligned}$$

Next consider $-1 < z_1 \le 0$. It follows that

$$\begin{aligned}&\Rightarrow \sigma(z_1) = \sin\left(\frac{\pi}{2}z_1\right) \text{ and } 0 < \frac{z_1 + 1}{2} \le \frac{1}{2}\\ &\Rightarrow 0 < \sigma(z_1 + 1) = \sin\left(\pi\frac{z_1 + 1}{2}\right) \le 1.\end{aligned}$$

The sinc function of $\mathrm{sinc}(z) \equiv (\sin \pi z)/(\pi z)$ [30] is monotonically decreasing from $z = 0$ to $z = 1$. So we have the following:

$$\begin{aligned}&\Rightarrow \mathrm{sinc}\left(\frac{1}{2}\right) \le \frac{\sin\left(\pi\frac{z_1+1}{2}\right)}{\pi\frac{z_1+1}{2}} = \mathrm{sinc}\left(\frac{z_1 + 1}{2}\right) < \mathrm{sinc}(0)\\ &\Rightarrow \frac{2}{\pi} \le \frac{\sin\left(\pi\frac{z_1+1}{2}\right)}{\pi\frac{z_1+1}{2}} < 1\\ &\Rightarrow z_1 + 1 \le \sin\left(\pi\frac{z_1 + 1}{2}\right) = \sigma(z_1 + 1)\\ &\Rightarrow z_1 \le \sigma(z_1 + 1) - 1 = I_1\\ &\Rightarrow |z_1| = -z_1 \ge 1 - \sigma(z_1 + 1) = |\sigma(z_1 + 1) - 1| \ge 0 \qquad (29)\end{aligned}$$

Hence the limit as z_1 approaches from 0^- is

$$\begin{aligned}\lim_{z_1 \to 0^-} \frac{|\sigma(z_1 + 1) - 1|}{|z_1|} &= \lim_{z_1 \to 0^-} \frac{1 - \sigma(z_1 + 1)}{-z_1}\\ &= \lim_{z_1 \to 0^-} \frac{1 - \sin\left(\pi\frac{z_1+1}{2}\right)}{-z_1}\\ &= \frac{-(\pi/2)\cos(\pi\frac{z_1+1}{2})}{-1}|_{z_1=0} = 0.\end{aligned} \tag{30}$$

Equation (30) is from L'Hospital's rule [31]. From (27), (28), (29) and (30), we have shown that (26) satisfies the activation function requirement in (14). In a similar way, (26) can be shown to satisfy (15).

From Theorem 2, the finding of the conservative DOA is to find the r such that

$$\text{if } |z_k| < r \Rightarrow |I_k| < \varepsilon|z_k|, \quad \text{for } k = 1, 2, 3 \tag{31}$$

where

$$\varepsilon < 1/\sqrt{3} = 0.577$$

for a three-neuron system. Because of the odd symmetry property of (26), it suffices to use z_1 and I_1 to find the r. Fig. 10 illustrates this process in finding the r such that if $|z_1| < r$, then $|I_1| \le 0.5|z_1|$, which can be solved iteratively in the following:

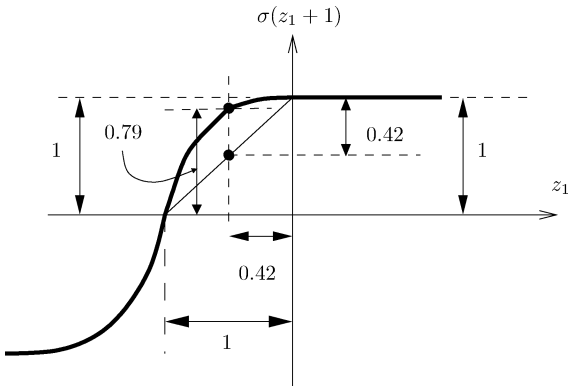

Fig. 10. Shifted sinusoidal activation function of $\sigma(z_1+1)$. The sinusoidal activation function is defined in (26).

1) Try $r = 0.5$, we have

$$\sin\left(\frac{\pi}{2}(-0.5+1)\right) = 0.707, \quad z_1 = -0.5$$
$$|I_1/z_1| = (1-0.707)/(0.5) = 0.586 > 0.5.$$

2) Then select $r = 0.45$, we have

$$\sin\left(\frac{\pi}{2}(-0.45+1)\right) = 0.76, z_1 = -0.45$$
$$|I_1/z_1| = (1-0.76)/(0.45) = 0.53 > 0.5.$$

3) Finally, the following converged solution is

$$\sin(\frac{\pi}{2}(-0.42+1)) = 0.79, \quad z_1 = -0.42$$
$$|I_1/z_1| = (1-0.79)/(0.42) = 0.5.$$

Equation (26) has the property that

$$\left|\frac{I_1}{z_1}\right|_{z_1=z_{1a}} < \left|\frac{I_1}{z_1}\right|_{z_1=z_{1b}}, \quad \text{for} -1 < z_{1b} < z_{1a} < 0,$$

or

$$\frac{d}{dz_1}\left|\frac{I_1}{z_1}\right| = \frac{d}{dz_1}\left(\frac{\sin(\pi\frac{z_1+1}{2})-1}{z_1}\right) < 0, \quad \text{for} -1 < z_1 < 0.$$

So it follows that $||z|| < r = 0.42$ is a conservative DOA for this three-neuron system. This same conservative DOA also holds for the case in Example 1 due to ratio weight variations.

V. Effects of Ratio Weight Variations

Consider the nonideal effect of ratio weight variations. Theorem 2 shows that if there are variations in ratio weights but the sum of the absolute values of ratio weights is kept at unity, then the ARMCNN is stable with a DOA. The other case in which the sum of the absolute values of ratio weights is not kept at unity is as follows. To have a correct recognition at $C(i,j)$, assume that all neighboring neurons produce the correct output $I_{\rm sat}$ or $-I_{\rm sat}$. After multiplied by the corresponding ratio weights, a necessary condition for a correct recognition of the ARMCNN at $C(i,j)$ is

$$g_{ij} \times \sum_{C(k,l)\in N_R(i,j)} |w_{ijkl}| > 1. \tag{32}$$

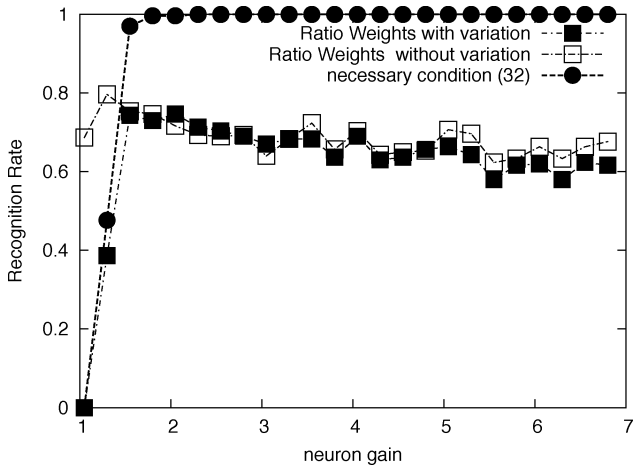

Fig. 11. Equation (32) (solid circle) predicts the RR (solid square) well at smaller neuron gains, while at larger gains, the RR (solid square) is approximately the same as the RR (empty square) with ideal ratio weights.

The neuron gain must be adjusted to overcome the effects of ratio weight variations to satisfy the necessary condition in (32). Fig. 11 shows the simulation results with the three learning patterns of 9×9 Chinese characters ONE, TWO, and FOUR in Fig. 5. $I_{\rm sat}$ is set to 5.5 μA, C_{ij} is 500 fF, and R_{ij} is 130 kΩ in Fig. 11.

Each input pattern is added with a Gaussian noise $I_{n,1}$ or $I_{n,2}$ with a standard deviation (STD) of $0.8\times I_{\rm sat}$ defined in (33) as following,

$$\begin{aligned} x_{ij}(0) &= (I_{\rm sat} + I_{n,1}) \times R_{ij} \quad \text{for a black pixel} \\ x_{ij}(0) &= (-I_{\rm sat} + I_{n,2}) \times R_{ij}, \quad \text{for a white pixel} \end{aligned} \tag{33}$$

where

$$\begin{aligned} &\text{if } (I_{\rm sat}+I_{n,1})\times R_{ij} > I_{\rm sat}R_{ij} \text{ then } x_{ij}(0) = I_{\rm sat}R_{ij} \\ &\text{if } (-I_{\rm sat}+I_{n,1})\times R_{ij} < -I_{\rm sat}R_{ij} \text{then } x_{ij}(0) = -I_{\rm sat}R_{ij}. \end{aligned}$$

One hundred noisy patterns for each character were generated as defined in (33). The RR of a group of m patterns at a fixed standard deviation is the number of sucessful recognitions divided by $100\times m$. The normalized output variable $I_{y_{ij}}(t)/I_{\rm sat}$ is compared with $+1, -1$ for a black$(+1)$ and a white(-1) pixel, respectively. To consider ratio weight variations in a VLSI process, the ratio weight was added with a Gaussian noise having a STD of $\sigma = 0.07$. This is because $3\sigma = 0.21$ is assumed to be the maximum possible noise. This study also assumes that the ratio weight noise can not invert the sign of the ratio weight.

Fig. 11 shows that for smaller gains, the RR under this ratio weight variation is close to the RR based on (32). On the other hand, in the high gain region, (32) is satisfied easily. And the RR under this ratio weight variation is close to the RR without variation. However, in this high gain region, more spurious memory points are generated. Thus, neuron gain g should be tuned large enough to satisfy the necessary condition of (32) and small enough to avoid the generation of many spurious memory points. So an optimal neuron gain exists in the case of ratio weight variations.

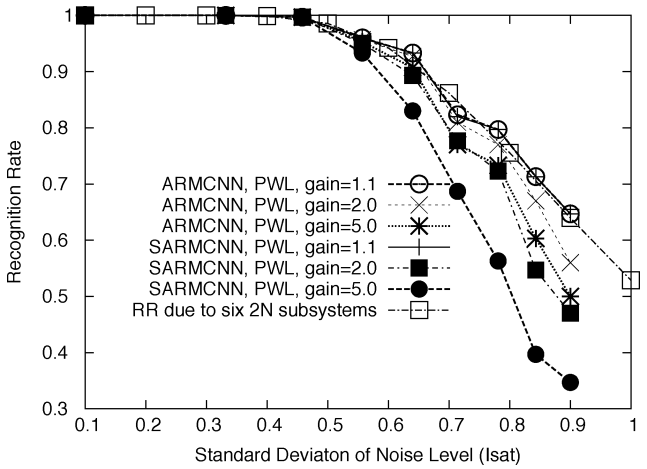

Fig. 12. The RR of recognizing three 9×9 Chinese characters ONE, TWO, and FOUR by ARMCNNs and SARMCNNs at different neuron gains. The RR decreases as the neuron gain increases.

VI. Recognition Rate Comparison

A. Comparison of ARMCNN and SARMCNN

Based upon the mathematical equations of a 9×9-neuron ARMCNN, this study performs behavior simulations for ARMCNNs and SARMCNNs with PWL activation functions at different neuron gains. The input patterns to be learned and recognized are three Chinese characters of ONE, TWO, and FOUR, as Fig. 5 shows. Fig. 4 shows the survived ratio weights. Fig. 4 also shows that no isolated neurons are generated. In Fig. 12, $I_{\rm sat}$ is set to 5.5 μA, C_{ij} is 500 fF, and R_{ij} is 130 kΩ for the practical realization of integrated circuits [17]. One hundred noisy patterns for each Chinese character were generated at a fixed standard deviation of the noise level as defined in (33). The x-axis of Fig. 12 shows that the standard deviations of $I_{n,1}$ and $I_{n,2}$ are in units of $I_{\rm sat}$. The RR of a group of m patterns is the number of successful recognitions divided by $100 \times m$. The output variable $y/I_{\rm sat}$ is compared with the correct one. All the 81 pixels must be correct to produce a successful recognition. Fig. 12 shows that when the standard deviation of the input noise level is smaller than $0.4 \times I_{\rm sat}$, the RR is almost unity for both ARMCNNs and SARMCNNs. Fig. 12 also shows that at larger neuron gains, SARMCNNs have a lower RR compared to ARMCNNs, and at lower neuron gains, both SARMCNN and ARMCNN have approximately the same RR. The RRs for ARMCNNs and SARMCNNs at lower neuron gains are close to RR_{2N}^{6}. This is as expected, because the RR is dominated by the smallest subsystems [7] and there are six 1-D 2N subsytems in Fig. 4. At a larger neuron gain, more spurious memory points are generated. Hence, the RR decreases as the neuron gain increases.

B. Comparison of SARMCNN and SVD-CNN

A potential problem in ARMCNNs without self-feedback is the possibility of isolated neurons. This is caused by low correlation between neurons. In this case, the charge on the neuron capacitor will decay to zero. A self-feedback ARMCNN (SARMCNN) can resolve this issue through the positive self-feedback. The self-feedback ratio weight is preserved for each SARMCNN neuron because each neuron is always positively correlative with itself. This study compares the SARMCNN with the work in [20], which is called the SVD-CNN in this paper. Fig. 13 shows the same four test patterns of $\alpha^1, \alpha^2, \alpha^3, \alpha^4$ from [20]. These four test patterns are expressed as

$$\alpha^1 = [1,-1,1,1,-1,1,1,1,1,-1,-1,1]$$
$$\alpha^2 = [1,1,1,1,-1,-1,1,-1,-1,1,1,1]$$
$$\alpha^3 = [1,-1,1,1,-1,1,1,-1,1,1,1,1]$$
$$\alpha^4 = [1,-1,-1,1,-1,-1,1,-1,-1,1,1,1].$$

The CNN architecture in [20] for this example is $M = 4, N = 3, n = 12$, and $\mathrm{Ra} = 1$. Using the same Sparse Design Procedure with $\mu = 2$ produces the same T matrix as in [20]. The other cases of $\mu = 1.1$ and 3.0 are also generated for comparison. μ is a parameter defined in [20]. Fig. 14 provides simulation results, which shows that the SARMCNN achieves the highest RR. The simulation parameters are the same for SVD-CNN and SARMCNN in that $I_{\rm sat}$ is set to 5.5 μA, C_{ij} is 500 fF, and R_{ij} is 130 kΩ for each neuron. The normalized output variable $y/I_{\rm sat}$ is compared with $+1$ for a black pixel and -1 for a white pixel. Performing the ARMCNN (or SARMCNN) algorithm in Section II generates three isolated neurons. The ARMCNN can not recognize these four patterns due to these three isolated neurons, whereas the SARMCNN can resolve this issue due to the self-feedback of each neuron. These three isolated neurons occur because of low correation between them and their neighbors. Therefore, the RR of SARMCNN is dominated by these three isolated neurons as,

$$\mathrm{RR} \leq \left(\int_0^{+\infty} \frac{\exp^{-(x-1)^2/(2\sigma^2)}}{\sqrt{2\pi}\sigma} dx \right)^3. \qquad (34)$$

Fig. 14 shows that (34) sets an upper bound for the RR of SARMCNN with good accuracy.

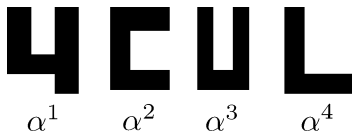

Fig. 13. The four test patterns in [20].

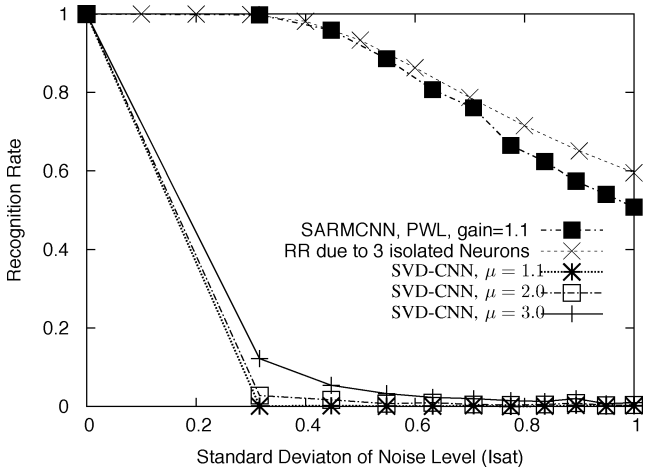

Fig. 14. Simulation results of RR between SARMCNN and SVD-CNN.

VII. Conclusion

This study proposes the stability analysis of the ARMCNN via the Lyapunov theorem. Results show that the ARMCNN can

TABLE I
COMPARISON OF VARIOUS CNN WORKS ON ASSOCIATIVE MEMORY

	This work	[22], [23]	[20], [21]	[24]
$\sigma(\bullet)$	Theorem 1	PWL	PWL	PWL
DOA	Theorem 2	LMI, GEVM	-	-
Analog VLSI	[17]	-	-	-
Synapse Circuit	V/Is,current mirrors	-	-	-
SR Ratio ≤ 1	yes	no	no	no
Weight	Hebbian learning	LMI, GEVM	SVD	pseudoinverse

tolerate large ratio weight variations and the activation function can vary as long as (14) and (15) are satisfied. In addition to the robustness of ARMCNNs, the learning rule is simple and therefore is suitable for analog VLSI implementation. Further, a conservative DOA can easily be determined from the stability proof using a simple graphical method. Table I shows the comparison between ARMCNNs and other CNNs on associative memory. The signal range ratio (SR ratio) is defined as $x/(I_{\rm sat}R)$ divided by $y/I_{I\rm sat}$. As shown in Table I, for ARMCNNs, the generation of weights is from Hebbian learning, which is biology like and was integrated into each CNN cell as analog VLSI. This Hebbian learning depends on the correlation between neighboring neurons. If after learning, many isolated neurons are generated, then the RR is dominated by these isolated neurons. For other CNNs, the generations of weights are from computer programs to perform SVD, LMI, GEVM, and pseudo-inverse operations. To integrate these operations into each CNN cell as analog VLSI is more difficult.

The primary problem for a ARMCNN without self-feedback is the occurrence of isolated neurons due to low correlation between neighboring neurons. Therefore, this study proposes the SARMCNN to solve the problem of isolated neurons. Further, for each subsystem, spurious memory points may exist besides the two binary equilibrium points. The occurence of spurious memory points will reduce the RR. Another issue is that the synapse weight circuit between neighboring neurons is composed of two V/Is and two current mirrors. The layout area is still too large for a high density CNN array.

The above two issues regarding spurious memory points and the layout area of synapse weight circuits originate from building ARMCNNs directly by the dynamic state equations with $g > 1$. However, at $g = 1$, ARMCNNs dynamic state equations approximate heat diffusion equations. And a single MOSFET transistor can be used as the synapse weight circuit. The steady state value of each neuron can be utilized to decide the final binary output image. Therefore, we are currently exploring the diffusion property of ARMCNNs to solve, at least, the above two issues to further improve the performance of ARMCNNs. This will be a future research effort.

REFERENCES

[1] M. Itoh and L. Chua, "Star cellular neural networks for associative and dynamic memories," *Int. J. Bifurc. Chaos*, vol. 14, pp. 1725–1772, 2004.

[2] F. C. Hoppensteadt and E. M. Izhikevich, *Weakly Connected Neural Network*. New York: Springer, 1997, pp. 108–.

[3] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," in *Proc. Natl. Acad. Sci. USA*, 1984, vol. 8, pp. 3088–3092.

[4] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*. Reading, MA: Addison-Wesley, 1991, pp. 21–.

[5] L. O. Chua and L. Yang, "Cellular neural networks: Theory," *IEEE Trans. Circuits Syst.*, vol. 35, no. 10, pp. 1257–1272, Oct. 1988.

[6] W. Gerstner and W. Kistler, *Spiking Neuron Models*. New York: Cambridge Univ. Press, 2002, pp. 356–357.

[7] C.-Y. Wu and S.-Y. Tsai, "Autonomous ratio-memory cellular nonlinear network (ARMCNN) for pattern learning and recognition," in *CNNA*, 2006, pp. 137–141.

[8] Y. Wu and C.-Y. Wu, "The design of CMOS} non-self-feedback ratio memory for cellular neural network without elapsed operation for pattern learning and recognition," in *CNNA*, 2005, pp. 282–285.

[9] C.-H. Cheng and C.-Y. Wu, "The design of cellular neural network with ratio memory for pattern learning and recognition," in *CNNA*, 2000, pp. 301–307.

[10] C.-H. Cheng and C.-Y. Wu, "The design of ratio memory cellular neural network (RMCNN) with self-feedback template weight for pattern learning and recognition," in *CNNA*, 2002, pp. 609–615.

[11] C.-Y. Wu and C.-H. Cheng, "A learnable cellular neural network structure with ratio memory for image processing," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 49, pp. 1713–1723, Dec. 2002.

[12] C.-Y. Wu and J.-F. Lan, "CMOS current-mode neural associative memory design with on-chip learning," *IEEE Trans. Neural Netw.*, vol. 1, pp. 167–181, Jan. 1996.

[13] J.-F. Lan and C.-Y. Wu, "CMOS current-mode outstar neural networks with long period analog ratio memory," in *Proc. IEEE Int. Symp. Circuits Syst.*, 1995, vol. 3, pp. 1676–1679.

[14] J.-F. Lan and C.-Y. Wu, "Analog CMOS current-mode implementation of the feedforward neural network with on-chip learning and storage," in *Proc. 1995 IEEE Int. Conf. on Neural Netw.*, 1995, vol. 1, pp. 645–650.

[15] C.-Y. Wu and J.-F. Lan, "A new neural associative memory with learning," in *IJCNN*, 1992, vol. 1, pp. 487–492.

[16] J.-F. Lan and C.-Y. Wu, "The multi-chip design of analog CMOS expandable modified Hamming neural network with on-chip learning and storage for pattern classification," in *ISCAS*, 1997, vol. 1, pp. 565–568.

[17] W.-T. Chou, "The Design of the Autonomous Ratio Memory Cellular Nonlinear Network Without Elapsed Operation for Pattern Learning and Recognition," Master's thesis, National Chiao-Tung Univ., Tainan, Taiwan, Dec. 2007.

[18] M. Gilli, M. Biey, and P. Checco, "Equilibrium analysis of cellular neural networks," *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 51, no. 5, pp. 903–912, May 2004.

[19] S. Arik and V. Tavsanoglu, "Equilibrium analysis of non-symmetric cnns," *International Journal of Circuit Theory and Applications*, vol. 24, pp. 269–274, 1996.

[20] D. Liu and A. Michel, "Sparsely interconnected neural networks for associative memories with applications to cellular neural networks," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 41, no. 4, pp. 295–307, Apr. 1994.

[21] M. Namba, "Estimating learner's comprehension with cellular neural network for associative memory," in *CNNA*, 2008, pp. 150–153.

[22] J. Park, H.-Y. Kim, Y. Park, and S.-W. Lee, "A synthesis procedure for associative memories based on space-varying cellular neural networks," *Neural Netw.*, vol. 14, pp. 107–113, Jan. 2001.

[23] R. Bise, N. Takahashi, and T. Nishi, "An improvement of the design method of cellular neural networks based on generalized eigenvalue minimization," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 50, pp. 1569–1574, Dec. 2003.

[24] G. Grassi, "A new approach to design cellular neural networks for associative memories," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 44, pp. 835–838, Sept. 1997.

[25] S.-C. Liu, J. Kramer, G. Indiveri, T. Delbruck, and R. Douglas, *Analog VLSI: Circuits and Principles*. Cambridge, MA: MIT Press, 2002.

[26] E. D. Ubeyli, "Eigenvector methods for automated detection of electrocardiographic changes in partial epileptic patients," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, pp. 478–485, Jul. 2009.

[27] G. Cauwenberghs and M. A. Bayoumi, *LEARNING ON SILICON: Adapative VLSI Neural Systems*. Reading, MA: Kluwer Acadamic, 1999, pp. 316–.

[28] S. Espejo, A. Rodriguez-Vazquez, R. Dominguez-Castro, and R. Carmona, "Convergence and stability of the FSR CNN model," in *CNNA*, 1994.

[29] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2002, pp. 318–319.

[30] R. Ziemer and W. Tranter, *Principles of Communication: Systems, Modulation and Noise*, 5th ed. Hoboken, NJ: John Wiley and sons, 2002.

[31] R. Silverman, *Calculus With Analytic Geometry*. New York: Prentice Hall, 1985, pp. 72–73.

Su-Yung Tsai (S'06) was born in Yilan, Taiwan. He received the B.S. degree in control engineering and the M.S. degree in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan in 1993 and 1995, respectively. From 1995 to 1997, he served as an army officer for the obligatory military service in Taiwan. He is currently working toward the Ph.D. degree in National Chiao Tung University, Taiwan.

His current research interests include VLSI implementation of cellular nonlinear networks and neuromorphic VLSI that mimics the biological system.

Chi-Hsu Wang (M'92–SM'93–F'08) was born in Tainan, Taiwan, in 1954. He received the B.S. degree in control engineering from National Chiaotung University, Hsinchu, Taiwan, the M.S. degree in computer science from the National Tsinghua University, Hsinchu, Taiwan, and the Ph.D. degree in electrical and computer engineering from the University of Wisconsin, Madison, in 1976, 1978, and 1986, respectively.

He was appointed Associate Professor in 1986, and Professor in 1990, in the department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan. He is currently a professor in the department of electrical engineering, National Chiaotung University, Hsinchu, Taiwan. His current research interests and publications are in the areas of digital control, fuzzy-neural-network, intelligent control, adaptive control, and robotics.

Dr. Wang is an IEEE Fellow. He is currently serving as an associate editor of IEEE Trans. On Systems, Man, and Cybernetics, Part B and as a member of Board of Governors and Webmaster of IEEE Systems, Man, and Cybernetics Society.

Chung-Yu Wu (S'76–M'76–SM'96–F'98) was born in 1950. He received the M.S. and Ph.D. degrees in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1976 and 1980, respectively.

Since 1980, he has been a Consultant to high-tech industry and research organizations and has built up strong research collaborations with high-tech industries. From 1980 to 1983, he was an Associate Professor with National Chiao Tung University. From 1984 to 1986, he was a Visiting Associate Professor with the Department of Electrical Engineering, Portland State University, Portland, OR. Since 1987, he has been a Professor with National Chiao Tung University. From 1991 to 1995, he served as the Director of the Division of Engineering and Applied Science, National Science Council, Taiwan. From 1996 to 1998, he was bestowed as the Centennial Honorary Chair Professor of National Chiao Tung University. He is currently the President and Chair Professor of National Chiao Tung University. In summer 2002, he conducted postdoctoral research with the University of California at Berkeley. He has authored or coauthored over 250 technical papers in international journals and conferences. He holds 30 patents, including 17 U.S. patents. His research interests are nano-electronics, biomedical devices and systems, neural vision sensors, RF circuits, and computer-aided design (CAD) and analysis.

Dr. Wu is a member of Eta Kappa Nu and Phi Tau Phi. He was a recipient of the 1998 IEEE Fellow Award and a 2000 Third Millennium Medal. He was also the recipient of numerous research awards presented by the Ministry of Education, National Science Council (NSC), and professional foundations in Taiwan.(1999–2003).