A Modular T-Mode Design Approach for Analog Neural Network Hardware Implementations

Bernabé Linares-Barranco, Edgar Sánchez-Sinencio, *Senior Member, IEEE*,
Angel Rodríguez-Vázquez, *Member, IEEE*, and
José L. Huertas, *Member, IEEE*

Abstract—**A modular transconductance-mode (T-mode) design approach is presented for analog hardware implementations of neural networks. This design approach is used to build a modular bidirectional associative memory (BAM) network. We will show that the size of the whole system can be increased by interconnecting more modular chips together. Also, we will show that by changing the interconnection strategy different neural network systems can be implemented, such as a Hopfield network, a winner-take-all network, a simplified ART1 network, or a constrained optimization network. Experimentally measured results from CMOS 2-μm double-metal, double-polysilicon prototypes (MOSIS) are presented.**

I. Introduction

MANY neural network algorithms have been proposed and studied in the computer science related literature [1]–[14]. Most of these algorithms have been implemented in a software environment. However, it is obvious that for many applications where real-time processing is necessary and/or the size of the complete computing system needs to be reduced, some type of special-purpose hardware implementation needs to be devised. In particular, analog circuits' capability for intrinsic high-speed operation with moderate area and power consumption [15] makes these techniques worthy to be explored in connection to neural networks.

In general, hardware circuit implementations of neural network systems can be made with low-precision components. This property is enhanced in neural systems that include adaptive learning or self-organization [16], because as the system learns to perform a certain function it implicitly compensates for imperfections and nonidealities present in the physical components of which the whole system is made. However, there is a category of circuits, often referred to as being also neural networks, for which the precision of the components is of high importance. These circuits are known as nonlinear programming circuits or constrained optimization circuits [17]–[20]. The outputs of these circuits have, in general, an analog nature, while for the other more conventional neural systems the outputs always have a digital nature, being therefore more immune to imprecise components.

Previous neural network analog VLSI implementations have been specific for particular neural network algorithms. However, if there were a modular hardware implementation able to be reconfigured to realize different neural network systems, it could be integrated with a conventional digital control system and generate very low-cost, very high-efficient, and versatile real-time neural processors. The work we present in this paper belongs in this category [20]–[23], and we explore this in connection to the use of transconductance-mode (T-mode) analog circuit techniques which have been demonstrated to be very well suited for high-speed analog processing in other application contexts [24].

We will present a very simple yet powerful fully analog, continuous-time, T-mode circuit design technique that can be used to implement most of the neural network systems proposed so far in the literature. This implementation technique is modular in the sense that the size of the system can be increased by interconnecting more chips together. No special interface or interchip communication hardware is needed for this. The convergence time of the system is not degraded when increasing its size. In the test prototypes we will present in this paper we use three different types of modular chips—one for the synapses, one for the neurons, and one for the external inputs. We did this mainly for test purposes, but these chips can be reduced to just one single modular chip. Also, the speed of the system can be drastically increased by sacrificing the modular property and integrating the complete system in one single nonmodular chip.

The experimental results we will present in this paper for constrained optimization circuits correspond to hardware realizations built with modular components that were originally designed to implement a bidirectional associative memory (BAM) network. This means that we will use low-precision components to assemble an optimization circuit. Therefore, as we will see, the results generated will be of moderate precision. However, it will serve to illustrate the underlying argument of this paper, which is the great versatility of the proposed implementation technique.

In the next section we will present the analog T-mode circuit design technique to be used in our further imple-

Manuscript received October 8, 1991; revised January 3, 1992.

B. Linares-Barranco, A. Rodríguez-Vázquez, and J. L. Huertas are with the Centro Nacional de Microelectrónica (CNM), 41012 Sevilla, Spain.

E. Sánchez-Sinencio is with the Department of Electrical Engineering, Texas A&M University, College Station, TX 77843.

IEEE Log Number 9107226.

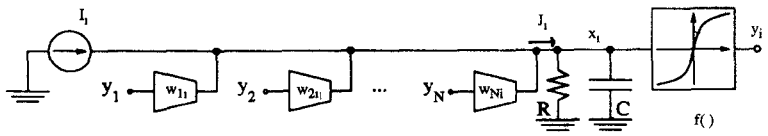


Fig. 1. Implementation of neuron interconnections using transconductance devices.

mentations. Then we will go directly to the experimental results and show the high potential of this technique by giving examples of a 5 × 5 BAM, a 9 × 9 BAM, a five-neuron Hopfield network, a five-neuron winner-take-all network, a 5 × 5 simplified ART1 network, and a moderate precision three-variable three-constraint quadratic constrained optimization network. The prototypes were fabricated in a standard 2-μm double-metal, double-polysilicon CMOS process (through and thanks to MOSIS).

Elsewhere [21], [25] we will demonstrate that this T-mode analog circuit design technique of neural network systems "fits like a glove" for making learning or self-organizing Hebbian type systems with little extra cost. We will also show how to add an on-chip analog memory for each synapse.

II. The T-Mode Neural Circuit Design Technique

Most of the neural network algorithms available in the literature have a short-term memory (STM) whose continuous-time version[1] operation can be described by the following set of nonlinear first-order differential equations:

$$C\dot{x}_i = -\alpha x_i + \sum_{j=1}^{N} w_{ji} f(x_j) + I_i, \qquad i = 1, \cdots, N \tag{1}$$

where x_i is the activity of neuron i, w_{ji} is the weight of the synaptic interconnection from neuron j to neuron i, I_i is the external input to neuron i, α and C are positive constants, and $f(\cdot)$ is a nonlinear, monotonically increasing function with maximum and minimum saturation values.

In some cases the system of equations in (1) is generalized by enriching its dynamics as in the Cohen–Grossberg description [27], or by sophisticating the synaptic interconnectivity like in high-order neural networks [28], or by adding constraint variables like in constrained optimization networks [17]–[20].

The system of equations in (1) can be directly implemented with analog hardware by using transconductance amplifiers as the synaptic interconnections. A transconductance amplifier provides an output current i_o proportional to its input voltage v_i:

$$i_o = g_m v_i \tag{2}$$

where g_m is the transconductance gain of the amplifier.

[1]Grossberg provides a method [26] to map a discrete-time description of a neural system into a continuous-time one, and vice-versa. Therefore, the neural network algorithms reported with discrete-time dynamics can also be represented by (1).

In (1) the output of a neuron $y_j = f(x_j)$ can be represented physically by a voltage signal, the synaptic connection by a transconductance amplifier of input voltage y_j and output current $w_{ji} y_j$, and the external inputs I_i by current variables. The lossy term $-\alpha x_i$ can be implemented by using a resistor of value $R = 1/\alpha$, and the operation $C\dot{x}_i$ can be realized by a capacitor. All this would produce a T-mode circuit representation as is shown in Fig. 1, where the function $f(\cdot)$ is implemented using a nonlinear voltage-to-voltage amplifier.

Assuming the network is stable and it converges to a stable steady state, consider for each neuron the association of the linear resistor R and the nonlinear voltage amplifier $f(\cdot)$ (see Fig. 2(a)). If J_{io} is the steady-state current entering this association and y_{io} is the steady-state output voltage, then

$$y_{io} = f(RJ_{io}) \Leftrightarrow J_{io} = \frac{1}{R} f^{-1}(y_{io}) \tag{3}$$

which can be visualized as a nonlinear resistor with a driving point characteristic $g(\cdot)$ such that

$$J_{io} = g(y_{io}) = \frac{1}{R} f^{-1}(y_{io}) \tag{4}$$

as is shown in Fig. 2(b).

By generalizing the concept of Fig. 2 to the nonsteady-state case as well, the circuit implementation of Fig. 1 is modified into the one shown in Fig. 3. Obviously the dynamics of the system of Fig. 3 is no longer described exactly by the set of equations in (1). The new system is now described by

$$C\dot{y}_i = -g(y_i) + \sum_{j=1}^{N} w_{ji} y_j + I_i, \qquad i = 1, \cdots, N. \tag{5}$$

However, once the steady state is reached, both descriptions are equivalent. In Appendix A we will give a stability proof for the system of equations in (5) as a particular case of a stability proof for quadratic optimization systems. In Appendix B we show that for an equivalent initial state both descriptions will produce the same equivalent final state.

In the next section we will use the T-mode circuit design technique of Fig. 3 to build a set of modular chips intended to implement an arbitrary-size continuous BAM network [5]. Afterwards we will show how to use these modular chips to assemble other neural network systems such as a Hopfield network [1]–[4], a winner-take-all network [11], [29], and a constrained optimization network [18]–[20].

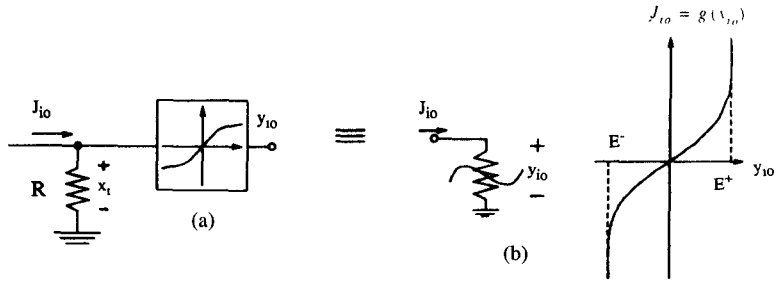


Fig. 2. T-mode simplified neuron implementation.

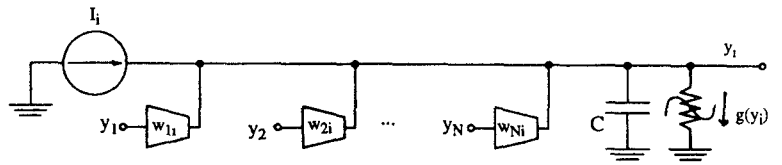


Fig. 3. T-mode simplified implementation for neural networks.

III. Experimental Results

A set of modular chips was designed and fabricated in a 2-μm double-metal, double-polysilicon, 10-V (V_{DD} = +5 V, V_{SS} = −5 V) CMOS process (MOSIS), and used to assemble several neural network systems [30].

The current sources for the implementation of the external inputs I_i of Fig. 3 were realized using the transconductance amplifiers of Fig. 4, which had a size of 20 × 20 μm^2 each. The value of V_{bias} is the same for all current sources I_i. Depending on the sign of I_i it was either $V_{1i} = V_{DD}$ and $V_{2i} = V_{SS}$, or $V_{1i} = V_{SS}$ and $V_{i2} = V_{DD}$.

The neuron is composed of a nonlinear resistor in parallel with a linear resistor and a capacitor. The circuit implementation of the nonlinear resistor is depicted in Fig. 5. If $E^- \leq y_i \leq E^+$ transistors $M1$ and $M2$ are OFF and $J_i = 0$. The only resistor in parallel with the integrating capacitor C of Fig. 3 is the parallel connection of all output impedances of the synaptic multipliers with outputs to this node. The value of this linear resistor is not critical for proper operation, which allows us to rely on parasitic elements for its physical implementation. If $y_i < E^-$ then $M1$ is ON and $M2$ OFF, and J_i is negative and large. If $y_i > E^+$ then $M1$ is OFF and $M2$ ON, and J_i is positive and large. Therefore, the circuit of Fig. 5 with the parallel connection of the output impedances of the synaptic multipliers with outputs to node y_i has driving point characteristics similar to the ones of Fig. 2(b).

For the synaptic transconductors a very simple circuit based on Gilbert's multiplying cell [31] was used, as is shown in Fig. 6. The size of the cell was 50 × 40 μm^2. The top PMOS current mirror was intentionally unbalanced for offset compensation, sacrificing linearity. All synaptic multipliers share the same V_{bias} = −3.77-V voltage, as well as all GND_{top} = −1.00-V and GND_{bottom} = −2.00-V connections. Fig. 7 shows the input output characteristics of the parallel connection of five synaptic multipliers loaded with a 20-kΩ resistor and for W = −2.8, −2.6, −2.2, −2.0, −1.8, −1.6, −1.4, and −1.2 V. A high degree of nonlinearity can be observed, especially around W = −2.0 V. However, as we will see, this will not affect the correct operation of the complete neural network systems.

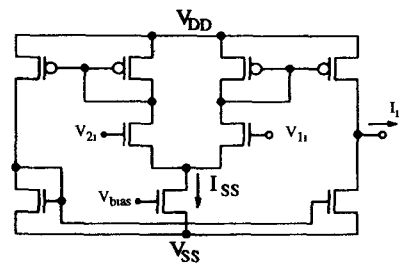


Fig. 4. Circuit implementation of transconductance amplifier.

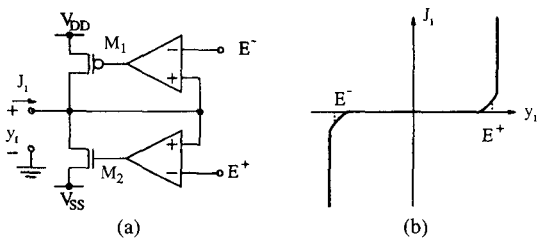


Fig. 5. (a) Nonlinear resistor circuit implementation. (b) Transfer curve.

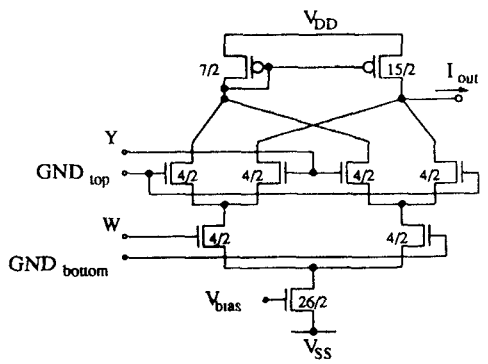


Fig. 6. Actual schematic of transconductance multipliers.

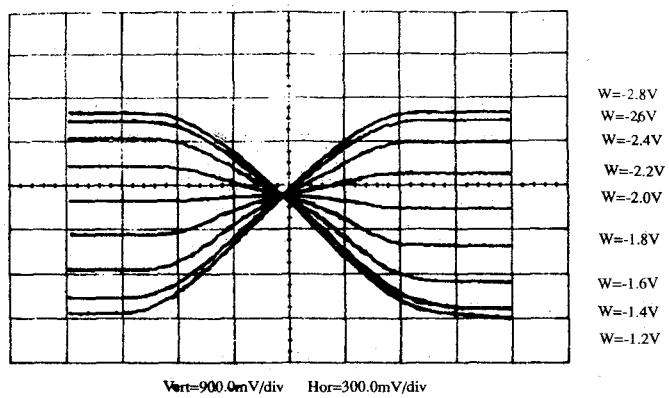


Fig. 7. Measurement of the dc transfer curves of five multipliers in parallel for $V_{bias} = -3.77$ V.

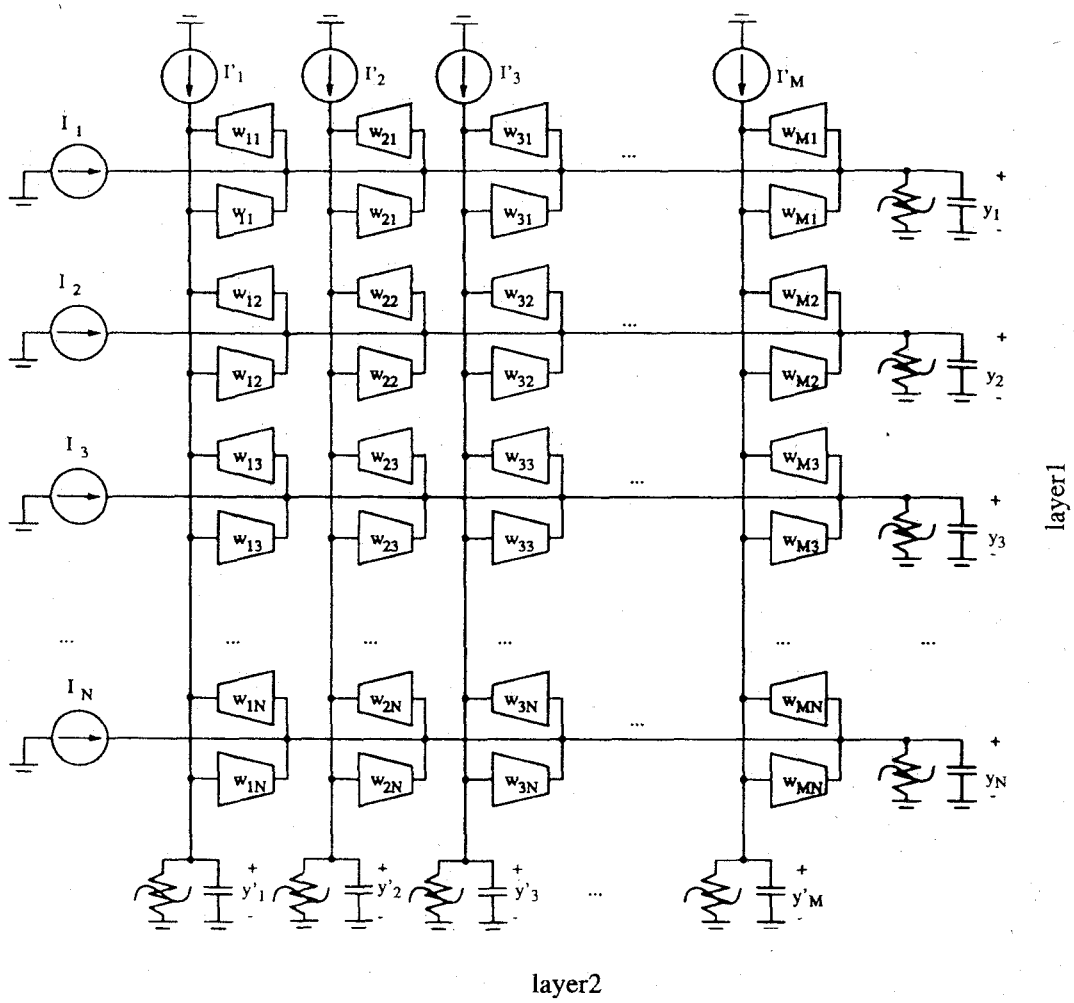


Fig. 8. T-mode circuit implementation of BAM algorithm.

A. BAM Networks

A BAM network is a two-layer neural network in which all neurons in one layer are connected to all neurons in the other layer, and there are no connections between neurons in the same layer [5]. The weight of the synapse that goes from neuron i in layer 1 (y_i) to neuron j in layer 2 (y_j') is the same that the one that goes from neuron j in layer 2 (y_j') to neuron i in layer 1 (y_i), and is denoted w_{ji}.

Using the circuit design technique represented in Fig. 3, a circuit realization of a 5 × 5 BAM network would be as shown in Fig. 8. Three different chips, one for the synaptic matrix, one for the neurons, and one for the external inputs, were fabricated in a 2-μm, double-metal, double-polysilicon CMOS process (MOSIS). Up to three patterns (with correct retrieval) could be stored in this 5 × 5 BAM. We programmed the patterns shown in Fig. 9. The nor-

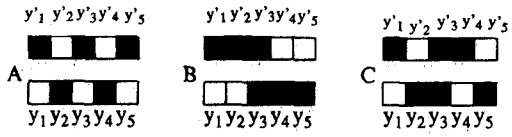


Fig. 9. Three patterns to be stored in the 5 × 5 BAM.

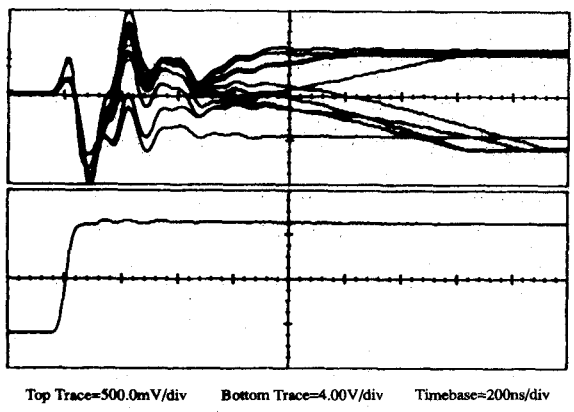


Fig. 10. Convergence to pattern A when the input is pattern A in 5 × 5 BAM. Top traces are neuron outputs; bottom trace is initial conditions triggering signal.

malized synaptic matrix for these patterns is

$$\begin{bmatrix} -3 & 1 & 1 & 1 & 1 \\ 1 & -3 & 1 & 1 & 1 \\ -3 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -3 & 1 \\ 1 & 1 & -3 & 1 & -3 \end{bmatrix} \tag{6}$$

and to program these weights the following voltages were used (see Figs. 6 and 7):

$$W = -2.2 \text{ V}, \quad \text{for } w_{ij} = +1$$
$$W = -1.2 \text{ V}, \quad \text{for } w_{ij} = -3. \tag{7}$$

The nonlinear resistors are biased with $E^+ = -0.5$ V, $E^- = -1.5$ V, and the input current sources with $V_{\text{bias}} = -2.50$ V. Fig. 10 shows the convergence to pattern A when the input pattern is A. For this some switches were added to the neurons in order to set the initial conditions and visualize the transient response. The BAM network also converged correctly to patterns B and C when the input patterns were B and C, respectively. In a continuous-time BAM the way to verify what minimum of the energy surface has been reached is by disconnecting the external inputs after the steady state has been reached. We did this for the case of Fig. 10 (as well as when the inputs where patterns B and C) and the system kept the same final state. When the external inputs are not patterns A, B, or C the BAM might reach in some cases a stable state slightly different from the stored patterns.[2] However, once this steady state is reached and the external inputs are disconnected, the BAM will settle to one of the stored patterns, depending on the hamming distances between the stored patterns and the input.

[2]This discrepancy depends on the ratio between the values of the external current sources and the current levels of the synaptic multipliers.

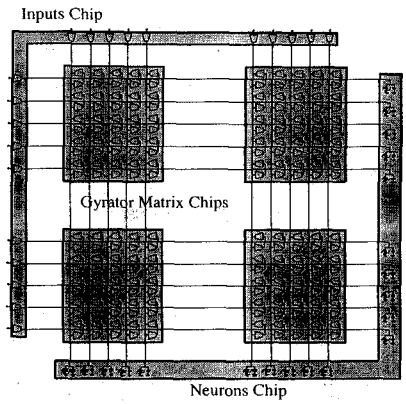


Fig. 11. Illustration of modular capability of T-mode circuit BAM implementation.

Exploiting the modular capability of the T-mode approach, we assembled several of the chips in Fig. 8, as is shown in Fig. 11, and built a 9 × 9 BAM.[3] The patterns shown in Fig. 12 were loaded with the normalized synaptic matrix

$$\begin{bmatrix} -3 & -1 & -1 & -1 & 3 & -1 & 1 & -1 & -1 \\ 1 & 3 & -1 & -1 & -1 & -1 & 1 & 3 & 3 \\ -1 & 1 & -3 & 1 & 1 & 1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 3 & -1 & 3 & -3 & -1 & -1 \\ 1 & 1 & 1 & -3 & 1 & -3 & 3 & 1 & 1 \\ 1 & -1 & -1 & 3 & 1 & 3 & -3 & -1 & -1 \\ 1 & -1 & -1 & 3 & -1 & 3 & -3 & -1 & -3 \\ 1 & 3 & -1 & -1 & -1 & -1 & 1 & 3 & 1 \\ -1 & -3 & 1 & 1 & 1 & 1 & -1 & -3 & -3 \end{bmatrix} \tag{8}$$

using the following weight voltages (see Figs. 6 and 7):

$$W = -2.8 \text{ V}, \quad \text{for } w_{ij} = +3$$
$$W = -2.2 \text{ V}, \quad \text{for } w_{ij} = +1$$
$$W = -1.8 \text{ V}, \quad \text{for } w_{ij} = -1$$
$$W = -1.2 \text{ V}, \quad \text{for } w_{ij} = -3. \tag{9}$$

This larger network is more sensitive to systematic offset in the synaptic multipliers. The value of V_{bias} in Fig. 6 for all the synaptic chips needed to be readjusted in order to minimize this offset and obtain a correct retrieval of the stored patterns of Fig. 12.[4] Fig. 13 shows how the neu-

[3]The voltage of one of the ten neurons in each layer in Fig. 11 was connected to $y_{10} = y'_{10} = \text{GND}_{\text{top}} = -1.0$ V and the external inputs for these neurons were set to $I_{10} = I'_{10} = 0$, so that they would not have any effect on the rest of the network.

[4]Note that here we are compensating a global offset which is similar for each synapse. Due to the nature of neural systems we do not anticipate any misbehavior due to random offsets in the synapses, as long as the mean random offset remains zero.

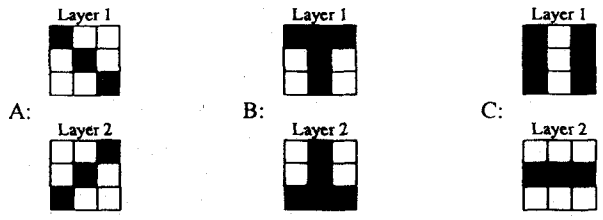


Fig. 12. Patterns to be stored in the 9 × 9 BAM.

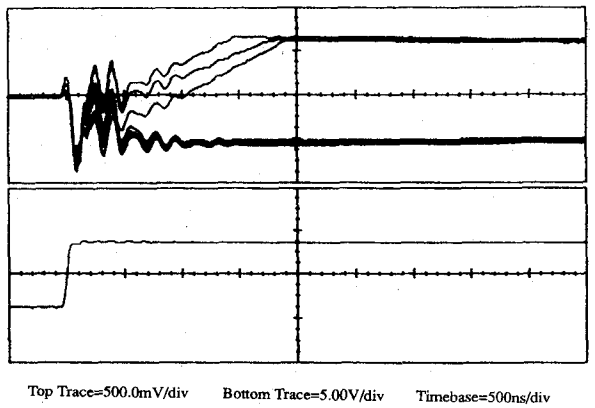


Fig. 13. Convergence to pattern A with input pattern A in 9 × 9 BAM. Top traces are layer 1 neuron outputs; bottom trace is initial conditions triggering signal.

rons of one layer converge to pattern A when the input is pattern A. Note that the convergence time of this network is of the same order as the one shown in Fig. 10, being a new system four times larger. This is because the settling time of this T-mode approach is independent of the size of the system. However, the time response does depend on the g_m's of the synapses, which depend on V_{bias}.

B. Hopfield and Winner-Take-All Networks

In order to demonstrate the versatility of the proposed T-mode technique we also assembled some other neural network algorithms. For example, a Hopfield network is a fully interconnected neural network without self-connections [1]–[4]. Interconnecting the modular chips of Fig. 8, as shown in Fig. 14, a five-neuron Hopfield network can be obtained. Hopfield networks have a very poor pattern capacity ($\approx 0.15 \times N$ for more than one pattern, where N is the number of neurons [1]). Therefore, for five neurons we can only successfully store just one pattern. Note that now each interconnection between neurons is made of two transconductance multipliers in parallel. In order to store the pattern "10101," the following normalized matrix needs to be programmed:

$$\begin{bmatrix} 0 & -1 & 1 & -1 & 1 \\ -1 & 0 & -1 & 1 & -1 \\ 1 & -1 & 0 & -1 & 1 \\ -1 & 1 & -1 & 0 & -1 \\ 1 & -1 & 1 & -1 & 0 \end{bmatrix} \tag{10}$$

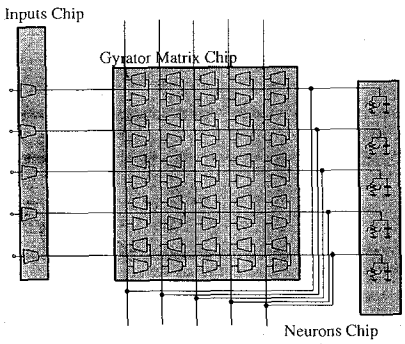


Fig. 14. Hopfield network built with T-mode BAM's modular chip components.

using the weight voltages (see Figs. 6 and 7)

$$W = -2.8 \text{ V}, \quad \text{for } w_{ij} = +1$$
$$W = -2.0 \text{ V}, \quad \text{for } w_{ij} = 0$$
$$W = -1.2 \text{ V}, \quad \text{for } w_{ij} = -1. \tag{11}$$

Fig. 15 shows the stable patterns observed for each input pattern configuration. The stable pattern is always either "10101" or "01010" depending on the hamming distance of the input patterns to the stored pattern.

A winner-take-all network can be considered as a special case of a Hopfield network. It is a fully interconnected network in which all self-connections are excitatory and all interconnections between different neurons are inhibitory. However, now excitatory connections have to remain excitatory always, and so do inhibitory connections. This means that the synaptic multipliers can no longer be four-quadrant ones; here they have to be two-quadrants multipliers. This is accomplished by making GND_{top} in Fig. 6 have the same value as $E^- = -1.5$ V in Fig. 5. The values for the synaptic matrix could be

$$\begin{bmatrix} 0.5 & -1 & -1 & -1 & -1 \\ -1 & 0.5 & -1 & -1 & -1 \\ -1 & -1 & 0.5 & -1 & -1 \\ -1 & -1 & -1 & 0.5 & -1 \\ -1 & -1 & -1 & -1 & -0.5 \end{bmatrix} \tag{12}$$

using the following weight voltages (see Figs. 6 and 7):

$$W = -2.4 \text{ V}, \quad \text{for } w_{ij} = +1/2$$
$$W = -1.2 \text{ V}, \quad \text{for } w_{ij} = -1. \tag{13}$$

Fig. 16 shows the transient response for neurons 1 and 4 when the input to the circuit is the pattern "10010." As can be seen, only one of the two neurons with high inputs is declared the winner.

C. Simplified ART1 Network

A very simplified view of Carpenter and Grossberg's ART1 network is as a BAM network [8] in which the external inputs to one of the layers have been substituted by

Input		Stable Pattern	
(0)	00000	10101	(21)
(1)	00001	10101	(21)
(2)	00010	01010	(10)
(3)	00011	10101	(21)
(4)	00100	10101	(21)
(5)	00101	10101	(21)
(6)	00110	10101	(21)
(7)	00111	10101	(21)
(8)	01000	01010	(10)
(9)	01001	10101	(21)
(10)	01010	01010	(10)
(11)	01011	01010	(10)
(12)	01100	10101	(21)
(13)	01101	10101	(21)
(14)	01110	01010	(10)
(15)	01111	10101	(21)
(16)	10000	10101	(21)
(17)	10001	10101	(21)
(18)	10010	10101	(21)
(19)	10011	10101	(21)
(20)	10100	10101	(21)
(21)	10101	10101	(21)
(22)	10110	10101	(21)
(23)	10111	10101	(21)
(24)	11000	10101	(21)
(25)	11001	10101	(21)
(26)	11010	01010	(10)
(27)	11011	10101	(21)
(28)	11100	10101	(21)
(29)	11101	10101	(21)
(30)	11110	10101	(21)
(31)	11111	10101	(21)

Fig. 15. Measured stable states for Hopfield circuit.

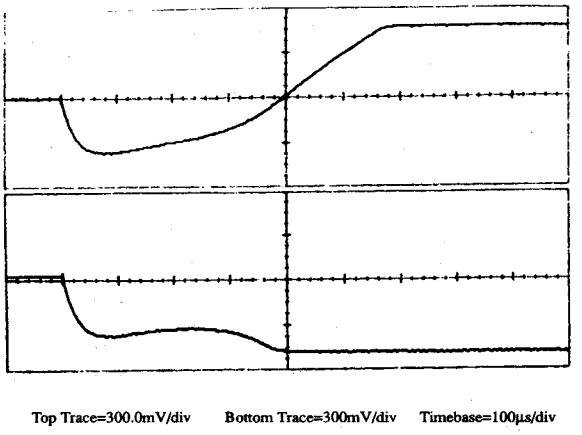


Fig. 16. Winner-take-all T-mode circuit with input (10010): the two traces correspond to neurons 1 and 4. The integrating capacitance of each neuron is 10 nF. Biases are: $V_{bias} = -3.77$ V, $GND_{bottom} = -2.00$ V, $GND_{top} = E = -1.50$ V, and $E^+ = -0.5$ V.

a winner-take-all interconnection matrix. This is illustrated in Fig. 17. The five patterns shown in Fig. 18 were programmed with the normalized synaptic matrix

$$\begin{bmatrix} 1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 \end{bmatrix} \tag{14}$$

using the following weight voltages (see Figs. 6 and 7):

$$W = -2.8 \text{ V}, \quad \text{for } w_{ij} = +1$$
$$W = -1.2 \text{ V}, \quad \text{for } w_{ij} = -1. \tag{15}$$

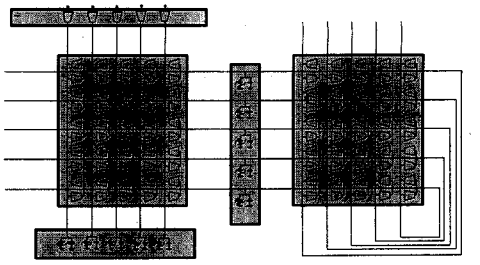


Fig. 17. Topology for simplified ART1 network using the BAM modular chips.

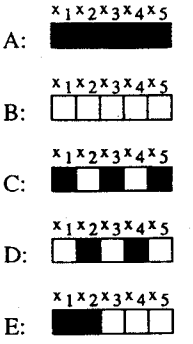


Fig. 18. Five patterns stored in the simplified ART1 network.

The winner-take-all section was biased as indicated in (12) and (13). The output pattern configuration is represented in Fig. 19, where we can see that the network converges to the stored pattern (or patterns) with the minimum hamming distance to the input pattern.

Note that the ART1 network achieves a maximum storage capacity of five patterns, while for the 5 × 5 BAM network the capacity was three, and for the Hopfield network the capacity was one, all of them having 25 programmable synapses.

D. Quadratic Constrained Optimization Network

The general problem of constrained optimization can be formulated as minimizing a given cost function of variables $v_1, v_2, \cdots, v_q$,

$$\Phi(v_1, v_2, \cdots, v_q). \tag{16}$$

subject to p constraints,

$$f_1(v_1, v_2, \cdots, v_q) \geq 0$$
$$f_2(v_1, v_2, \cdots, v_q) \geq 0$$
$$\cdots$$
$$f_p(v_1, v_2, \cdots, v_q) \geq 0 \tag{17}$$

where q and p are two independent positive integers.

Mathematically, the problem can be solved using the Lagrange multiplier method [32] by defining the La-

	Stable Pattern		
Input	Layer 1	Layer 2	
(0) 00000	00000	01000	B
(1) 00001	00001	01000	B
(2) 00010	00010	01100	BC
(3) 00011	00010	00010	D
(4) 00100	00000	01000	B
(5) 00101	10101	00100	C
(6) 00110	01010	00010	D
(7) 00111	11111	10000	A
(8) 01000	01000	01011	BDE
(9) 01001	01010	00010	D
(10) 01010	01010	00010	D
(11) 01011	11111	10000	A
(12) 01100	01110	10010	AD
(13) 01101	11101	10100	AC
(14) 01110	01110	10010	AD
(15) 01111	11111	10000	A
(16) 10000	00000	01000	B
(17) 10001	10101	00100	C
(18) 10010	00010	01010	BD
(19) 10011	10111	10100	AC
(20) 10100	10101	00100	C
(21) 10101	10101	00100	C
(22) 10110	10111	10100	AC
(23) 10111	10111	10100	AC
(24) 11000	11000	00001	E
(25) 11001	11000	00001	E
(26) 11010	01010	00010	D
(27) 11011	11111	10000	A
(28) 11100	11000	00001	E
(29) 11101	11101	10100	AC
(30) 11110	11111	10000	A
(31) 11111	11111	10000	A

Fig. 19. Stable patterns obtained for the simplified ART1 network.

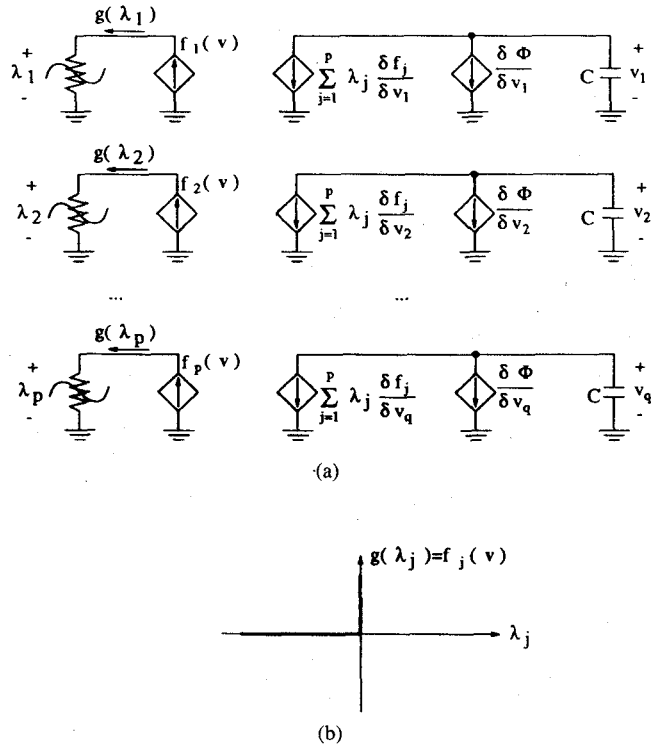


Fig. 20. (a) General constrained optimization circuit. (b) Nonlinear resistor curve.

grange function

$$L(v_1, v_2, \cdots, v_q, \lambda_1, \lambda_2, \cdots, \lambda_p) = \Phi + \sum_{j=1}^{p} \lambda_j f_j \tag{18}$$

where λ_j are called the Lagrange multipliers. The solution is obtained by solving

$$\frac{\partial L}{\partial v_k} = \frac{\partial \Phi}{\partial v_k} + \sum_{j=1}^{p} \lambda_j \frac{\partial f_j}{\partial v_k} = 0$$

$$f_j(\vec{v}) \geq 0, \quad \lambda_j \leq 0, \quad \lambda_j f_j(\vec{v}) = 0 \tag{19}$$

where the unknowns are v_k and λ_j. The circuit of Fig. 20 [17], [18] solves this system of equations, assuming it converges to a stable steady state. In Appendix A we show that this circuit is completely stable under certain conditions.

The quadratic constrained optimization problem is a particular case of the general problem described by (16) and (17), such that

$$\Phi(\vec{v}) = [A_1 \cdots A_q] \begin{bmatrix} v_1 \\ \cdots \\ v_q \end{bmatrix} + \tfrac{1}{2}[v_1 \cdots v_q] \cdot \begin{bmatrix} G_{11} & \cdots & G_{1q} \\ \cdots & \cdots & \cdots \\ G_{q1} & \cdots & G_{qq} \end{bmatrix} \begin{bmatrix} v_1 \\ \cdots \\ v_q \end{bmatrix} \tag{20}$$

and

$$\begin{bmatrix} f_1 \\ \cdots \\ f_p \end{bmatrix} = \begin{bmatrix} B_{11} & \cdots & B_{1q} \\ \cdots & \cdots & \cdots \\ B_{p1} & \cdots & B_{bq} \end{bmatrix} \begin{bmatrix} v_1 \\ \cdots \\ v_q \end{bmatrix} - \begin{bmatrix} E_1 \\ \cdots \\ E_p \end{bmatrix} \geq 0. \tag{21}$$

A T-mode circuit that implements (19) when $\Phi(\vec{v})$ and $f_i(\vec{v})$ are defined by (20) is shown in Fig. 21.

The linear programming circuit [4] is a particular case of the quadratic programming circuit for which $G_{ij} = 0$.

The solution of a general optimization circuit has an analog nature. It does not saturate to a maximum or minimum value as happens in the BAM, Hopfield, winner-take-all, and ART1 networks. This fact implies that higher precision components need to be used. However, for illustration purposes we will use the same modular BAM chips that we have used so far. This means that precision in the solution will be sacrificed to a certain degree.

Let us implement the following three-variable, three-constraint quadratic optimization problem. Minimize

$$\Phi = 2v_1v_3 - 2v_2v_3 + v_3^2 \tag{22}$$

subject to the constraints

$$v_1 \geq 0, \quad v_2 \leq \tfrac{1}{2}, \quad v_3 \geq 0. \tag{23}$$

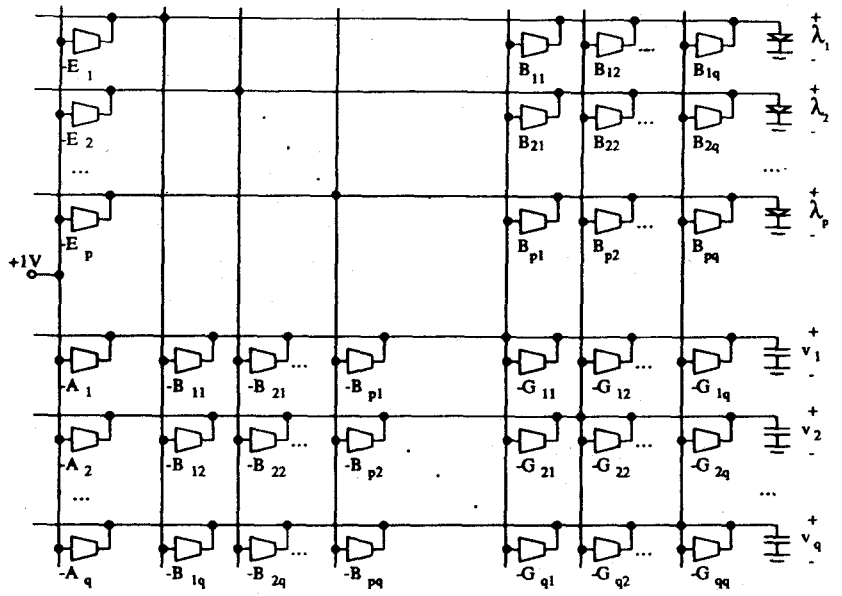


Fig. 21. T-mode implementation of constrained quadratic optimization problem.

The exact solution to this problem is

$$v_1 = 0, \quad v_2 = \tfrac{1}{2}, \quad v_3 = \tfrac{1}{2}. \tag{24}$$

The corresponding normalized matrices and vectors of (20) and (21) are

$$G = \begin{bmatrix} 0 & 0 & 2 \\ 0 & 0 & -2 \\ 2 & -2 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\vec{A} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \vec{E} = \begin{bmatrix} 0 \\ 0.5 \\ 0 \end{bmatrix}. \tag{25}$$

In the actual circuit we will have

$$\begin{aligned} \Phi &= \tfrac{1}{2}G_{33}v_3^2 + G_{13}v_1v_3 + G_{23}v_2v_3 \\ f_1 &= B_{11}v_1 \geq 0 \\ f_2 &= B_{22}v_2 - E_2 \geq 0 \\ f_3 &= B_{33}v_3 \geq 0 \end{aligned} \tag{26}$$

with

$$\begin{aligned} &G_{33} = 2g_o, \quad G_{11} = 2g_o, \quad G_{23} = -2g_o \\ &B_{11} = g_o, \quad B_{22} = -g_o, \quad B_{33} = g_o \\ &\frac{E_2}{B_{22}} = \frac{1}{2} \Rightarrow E_2 = -\frac{g_o}{2} \end{aligned} \tag{27}$$

where g_o is a scaling transconductance. For $V_{\text{bias}} = -3.77$ V, and if $W = -1.2$, -2.8 V, according to Fig. 7, it would be $g_o \approx 30$ μmhos. Since each multiplier in Fig. 21 has a linear range of approximately ± 500 mV (see Fig. 7) we have to make sure that the variables v_1, v_2, v_3 and λ_1, λ_2, λ_3 have a solution within this linear range. In the steady state the circuit satisfies

$$\begin{aligned} &\vec{A}\,\vec{v}^{\,T} + \tfrac{1}{2}G\vec{v}^{\,T} + \tfrac{1}{2}G^T\vec{v}^{\,T} + B^T\vec{\lambda}^{\,T} = 0 \\ &B\,\vec{v}^{\,T} \geq \vec{E}^{\,T}. \end{aligned} \tag{28}$$

However, if in the steady state any of v_i or λ_i is beyond the linear range of the multipliers the solution is not valid and the problem needs to be rescaled. This can be done by defining

$$v_i' = \gamma v_i \quad \lambda_i' = \gamma\lambda_i \quad 0 < \gamma < 1 \tag{29}$$

and in order to keep the problem unchanged (see (28)) we also need to define a new $\vec{E}\,'$ such that

$$E_i' = \gamma E_i. \tag{30}$$

For our case, a factor $\gamma = 1/4$ produced values of v_i' and λ_i' that were within the linear range ± 500 mV of the multipliers.

The circuit configuration assembled with the modular BAM chips is depicted in Fig. 22. The interchip buffers were used to eliminate the bidirectional nature of the synaptic multipliers. The nonlinear resistors were biased using $E^+ = 0$ V and $E^- = -5$ V, so that they would implement the characteristics of an ideal diode. The measured steady-state response of this circuit was

$$\begin{aligned} &v_1' = 90 \text{ mV}, \quad v_2' = 180 \text{ mV}, \quad v_3' = 125 \text{ mV} \\ &\lambda_1' = -300 \text{ mV}, \quad \lambda_2' = -250 \text{ mV}, \quad \lambda_3' = 20 \text{ mV} \end{aligned} \tag{31}$$

which corresponds to the normalized problem solution

$$\begin{aligned} &v_1 = 0.36, \quad v_2 = 0.72, \quad v_3 = 0.50 \\ &\lambda_1 = 1.20, \quad \lambda_2 = 1.00, \quad \lambda_3 = 0.08 \end{aligned} \tag{32}$$

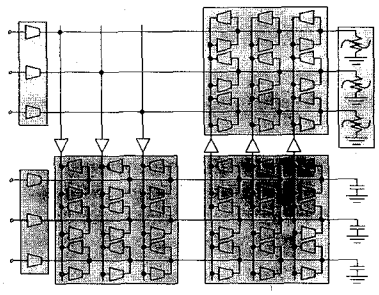


Fig. 22. Interconnection topology for optimization circuit.

while the exact solution should have been

$$v_1 = 0.00, \quad v_2 = 0.50, \quad v_3 = 0.50$$

$$\lambda_1 = 1.00, \quad \lambda_2 = 1.00, \quad \lambda_3 = 0.00. \tag{33}$$

The discrepancy is due to the low-precision components used. When using high-precision analog circuit design techniques, like switched-capacitor circuits, very precise CMOS optimization networks can be achieved [20].

IV. Conclusions

We have proposed, developed, and demonstrated a compact, modular, versatile, cheap, and powerful circuit design technique for the implementation of continuous-time analog neural networks. This technique is based on the use of transconductance synaptic multipliers and neural nonlinear resistors. We have used this approach to design a set of modular chips intended for the implementation of arbitrary-size BAM networks.

These BAM's were successfully tested. Afterwards, we used the same modular chips to assemble other neural network algorithms, such as a five-neuron Hopfield network, a five-neuron winner-take-all network, a 5 × 5 ART1 network, and a moderate precision three-variable, three-constraint optimization network.

This circuit design technique has been extended by including an on-chip Hebbian learning rule for each synapse [21], [30] as well as an on-chip analog dynamic memory for the weight storage of each synapse. The corresponding results will be published elsewhere [25].

In the experimental results given in this paper, absolutely no care was taken to minimize the response times. However, since this is an analog approach, such response times are given by the time constants of the g_m's and capacitors involved, and can be minimized to those values characteristic of other analog circuits.

In summary, this paper demonstrates the high potential and versatility of the proposed T-mode circuit design technique for the analog hardware implementation of neural networks on standard low-cost CMOS processes.

Appendix A

Theorem: The circuit of Fig. 20 is *completely stable* in the sense that it will never oscillate or display other exotic modes of operations [19], assuming the following conditions are satisfied:

- At least one (and maybe more) solution to the problem exists. Consequently, the cost function is bounded from below within the region over which the constraints are satisfied.
- The functions $\Phi(\cdot)$ and $f(\cdot)$ are continuous, and all their first and second derivatives exist and are continuous.

Proof: The equations for the network are

$$C\dot{v}_i = -\frac{\partial \Phi}{\partial v_i} - \sum_{j=1}^{p} \lambda_j \frac{\partial f_j}{\partial v_i}, \qquad i = 1, \cdots, q$$

$$\lambda_j = g(f_j(\vec{v})), \qquad j = 1, \cdots, p. \tag{34}$$

Since $g(\cdot)$, $\Phi(\cdot)$, and $f_j(\cdot)$ are continuous, (34) can be written as

$$\dot{\vec{v}}(t) = \vec{h}(\vec{v}(t)) \tag{35}$$

where $\vec{h}(\cdot)$ is a continuous function from $\Re^q$ to $\Re^q$. Consider the scalar function $E(\vec{v})$: $\Re^q \to \Re$,

$$E(\vec{v}) = \Phi(\vec{v}) + \sum_{j=1}^{p} \int_0^{f_j(\vec{v})} g_j(x)\, dx \tag{36}$$

taking time derivatives yields

$$\frac{dE}{dt} = \sum_{j=1}^{q} \frac{\partial \Phi}{\partial v_i} \dot{v}_i + \sum_{j=1}^{p} \sum_{i=1}^{q} \frac{\partial f_j}{\partial v_i} \dot{v}_i g_j(f_j(\vec{v}))$$

$$= -\sum_{i=1}^{q} C_i \dot{v}_i^2. \tag{37}$$

Therefore, $dE/dt \leq 0$. This implies that $E(t)$ is strictly decreasing unless $\dot{v}_i = 0$ for all $i = 1, \cdots, q$, which corresponds to the steady state. This means that $E(\vec{v})$ is a *Lyapunov function* of the system, which together with the continuity of $\vec{h}(\cdot)$ ensures that the system is completely stable, i.e., any trajectory $\vec{v}(\cdot)$ eventually converges to some equilibrium point $\vec{v}^*$ in $\Re^q$ depending on the initial state $\vec{v}_o$ [19].

Corollary: The fully interconnected T-mode neural network (see Fig. 3) is a particular case of the T-mode constrained quadratic optimization network, and is therefore (via the previous theorem) also completely stable.

The circuit of Fig. 3 can be viewed as the following constrained quadratic optimization network:

$$\Phi(\vec{v}) = [I_1 \cdots I_q] \begin{bmatrix} y_1 \\ \cdots \\ y_q \end{bmatrix} + \tfrac{1}{2}[y_1 \cdots y_q] \cdot \begin{bmatrix} w_{11} & \cdots & w_{1q} \\ \cdots & \cdots & \cdots \\ w_{q1} & \cdots & w_{qq} \end{bmatrix} \begin{bmatrix} y_1 \\ \cdots \\ y_q \end{bmatrix} \tag{38}$$

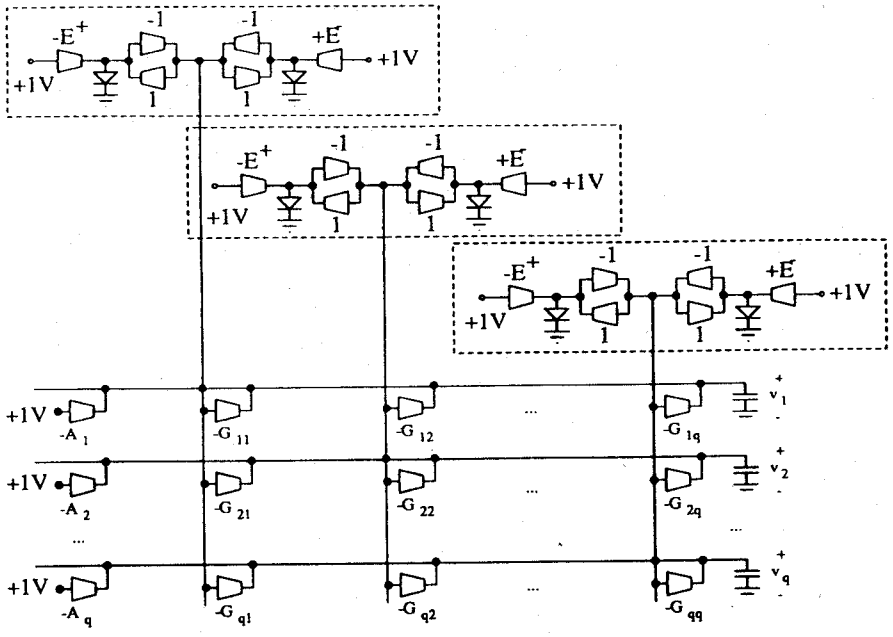


Fig. 23. T-mode neural network as a particular case of constrained optimization circuit.

subject to the constraints

$$\begin{aligned} f_1&: \; y_1 \geq -E^- \\ f_2&: \; y_1 \leq +E^+ \\ f_3&: \; y_2 \geq -E^- \\ f_4&: \; y_2 \leq +E^+ \\ &\cdots \\ f_{2q-1}&: \; y_q \geq -E^- \\ f_{2q}&: \; y_q \leq +E^+. \end{aligned} \tag{39}$$

These constraints equations in matrix form are

$$\begin{bmatrix} f_1 \\ f_2 \\ \cdots \\ f_{2q} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \cdots & & \cdots & & \cdots \\ 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_q \end{bmatrix} - \begin{bmatrix} -E^- \\ +E^+ \\ \cdots \\ +E^+ \end{bmatrix}. \tag{40}$$

The circuit of Fig. 21 for this particular case is shown in Fig. 23. Note that the circuits comprised by broken lines behave like the nonlinear resistors of Fig. 5, and therefore the circuit of Fig. 23 is equivalent to the one in Fig. 3.

Appendix B

Theorem: Given the same equivalent initial conditions for the neural network described by (1) and the neural network described by (5), they will arrive at the same equivalent final state if the weight matrix of the fully interconnected network is invertible.

Proof: This will happen if there exists a set of functions

$$h_{ij}(x_1, \cdots, x_N), \qquad i, j = 1, \cdots, N \tag{41}$$

such that the following perturbative approximation can be made:

$$\begin{aligned} y_i &= f(x_i) + \sum_{j=1}^{N} \dot{x}_j h_{ij}(\vec{x}) \\ &\left| \sum_{j=1}^{N} \dot{x}_j h_{ij}(\vec{x}) \right| \ll f(x_i) \end{aligned} \tag{42}$$

and $h_{ij}(\vec{x})$ does not diverge once the steady state is reached. If (42) are satisfied and if there is a set of trajectories $x_i(t)$ that solves the description of (1) for a given initial condition, then there is also a set of trajectories $y_i(t)$ that solves the description of (5) for the same equivalent initial condition, and both sets of trajectories arrive to the same equivalent steady state. If we can show that the functions $h_{ij}(\cdot)$ do exist then the theorem is proved.

Taking the time derivative in (42) and neglecting the terms in $\dot{x}_j \dot{x}_i$ results in

$$\dot{y}_i = f'(x_i)\dot{x}_i + \sum_{j=1}^{N} \ddot{x}_j h_{ij}(\vec{x}). \tag{43}$$

Also, by (42)

$$f^{-1}(y_i) \approx x_i + \rho(x_i) \sum_{j=1}^{N} \dot{x}_j h_{ij}(\vec{x})$$

$$\rho(x_i) = f'^{-1}(f(x_i)). \tag{44}$$

Substituting the time derivative of (1) in (43), this result together with (44) in (5) yields

$$C[f'(x_i) - 1]\dot{x}_i + \alpha[\rho(x_i) - 1] \sum_{j=1}^{N} \dot{x}_j h_{ij}(\vec{x}) + \sum_{j=1}^{N} \sum_{k=1}^{N} h_{ij}(\vec{x})\, w_{kj} f'(x_k)\dot{x}_k - \sum_{j=1}^{N} \sum_{k=1}^{N} w_{ij} h_{jk}(\vec{x})\dot{x}_k = 0 \tag{45}$$

which, in matrix form, can be expressed as

$$C[f' - 1]_D \dot{\vec{x}}^T + \alpha[\rho - 1]_D H \dot{\vec{x}}^T + HW[f']_D \dot{\vec{x}}^T + WH\dot{\vec{x}}^T = 0 \tag{46}$$

where H is the matrix of elements $h_{ij}(\vec{x})$, W is the one of elements w_{ij}, and $[f' - 1]_D$, $[\rho - 1]_D$, and $[f']_D$ are diagonal matrices of diagonal elements $f'(x_i) - 1$, $\rho(x_i) - 1$, and $f'(x_i)$, respectively. Since (46) has to hold for any $\dot{\vec{x}}^T$, it must be

$$C[f' - 1]_D + \alpha[\rho - 1]_D H + HW[f']_D + WH = 0. \tag{47}$$

The solution of this matrix equation provides the functions h_{ij}, and therefore, the result of (42). If this solution exists and does not diverge, the theorem is proved.

In the steady state $f'(x_i) \approx 0$ and $\rho(x_i) - 1 \approx 0$. Therefore, in the steady state we have

$$H = -CW^{-1}[1]_D. \tag{48}$$

This solution exists and is bounded if the weight matrix is invertible. Since the final state is bounded as well as the initial state and both (1) and (5) are well behaved, the transient response will also be bounded. Furthermore, if there is a solution for (1) with trajectories $x_i(t)$, there is also a solution for (5) with trajectories $y_i(t)$ which can be computed through (47) and (42), and both sets of trajectories have the same equivalent initial and final states defined by

$$y_i = f(x_i), \qquad i = 1, \cdots, N. \tag{49}$$

References

[1] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci. USA*, vol. 79, pp. 2554–2558, Apr. 1982.

[2] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Nat. Acad. Sci. USA*, vol. 81, pp. 3088–3092, May 1984.

[3] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biol. Cybern.*, vol. 52, pp. 141–152, 1985.

[4] D. W. Tank and J. J. Hopfield, "Simple 'neural' optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit," *IEEE Trans. Circuits Syst.*, vol. CAS-33, pp. 533–541, May 1986.

[5] B. Kosko, "Adaptive bidirectional associative memories," *Appl. Opt.*, vol. 26, pp. 4947–4960, Dec. 1987.

[6] B. Kosko, "Bidirectional associative memories," *IEEE Trans. Syst., Man, Cybern.*, vol. 18, pp. 49–60, Jan./Feb. 1988.

[7] B. Kosko, *Neural Networks and Fuzzy Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[8] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural patten recognition machine," *Comput. Vision, Graphics, Image Processing*, vol. 37, pp. 54–115, 1987.

[9] T. Kohonen, "The 'neural' phonetic typewriter," *Computer*, vol. 21, pp. 11–22, Mar. 1988.

[10] T. Kohonen, *Self-Organization and Associative Memory*, 2nd ed. New York: Springer Verlag, 1988.

[11] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Mag.*, Apr. 1987.

[12] J. A. Anderson, "A simple neural network generating interactive memory," *Math. Biosci.*, vol. 14, pp. 197–220, 1972.

[13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[14] G. E. Hinton and R. J. Sejnovski, "Learning and relearning in Boltzmann machines," in *Parallel Distribution Processing*, vol. 1, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: M.I.T. Press, 1986, ch. 7.

[15] E. A. Vittoz, "Future of analog in the VLSI environment," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS90)* (New Orleans), 1990, pp. 1372–1375.

[16] C. Mead, "Neuromorphic electronic systems," *Proc. IEEE*, vol. 78, no. 10, pp. 1629–1636, Oct. 1990.

[17] L. O. Chua and G. N. Liu, "Nonlinear programming without computation," *IEEE Trans. Circuits Syst.*, vol. CAS-31, pp. 182–188, Feb. 1984.

[18] M. P. Kennedy and L. O. Chua, "Unifying the tank and Hopfield linear programming circuit of Chua and Lin," *IEEE Trans. Circuits Syst.*, vol. CAS-34, pp. 210–214, Feb. 1987.

[19] M. P. Kennedy and L. O. Chua, "Neural networks for nonlinear programming," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 554–562, May 1988.

[20] A. Rodríguez-Vázquez, R. Domínguez-Castro, A. Rueda, J. L. Huertas, and E. Sánchez-Sinencio, "Nonlinear switched-capacitor 'neural' networks for optimization problems," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 384–398, Mar. 1990.

[21] B. Linares-Barranco, E. Sánchez-Sinencio, A. Rodríguez-Vázquez, and J. L. Huertas, "VLSI implementation of a transconductance mode continuous BAM with on chip learning and dynamic analog memory," in *Proc. IEEE Int. Symp. Circuits Syst (ISCAS91)* (Singapore), 1991, pp. 1283–1286.

[22] M. Robinson, "Analog CMOS neural Hamming networks," Master's thesis, Texas A&M Univ., College Station, Dec. 1991.

[23] H. Yoneda, "VLSI implementation of neocognitron," Master's thesis, Texas A&M Univ., College Station, Dec. 1991.

[24] B. Linares-Barranco, A. Rodríguez-Vázquez, E. Sánchez-Sinencio, and J. L. Huertas, "CMOS OTA-C high-frequency sinusoidal oscillators," *IEEE J. Solid-State Circuits*, vol. 26, no. 2, pp. 160–165, Feb. 1991.

[25] B. Linares-Barranco, E. Sánchez-Sinencio, A. Rodríguez-Vázquez, and J. L. Huertas, "A CMOS learning BAM with on-chip analog memory," to be submitted to *IEEE Trans. Neural Networks*.

[26] S. Grossberg, "Nonlinear neural networks: Principles, mechanisms, and architectures," *Neural Networks*, vol. 1, no. 1, pp. 17–61, 1988.

[27] M. A. Cohen and S. Grossberg, "Absolute stability of global pattern formation and parallel memory storage by competitive neural networks," *IEEE Trans. Syst., Man, Cybern.*, vol. 13, pp. 815–826, 1983.

[28] C. L. Giles and T. Maxwell, "Learning, invariance, and generalization in high-order neural networks," *Appl. Opt.*, vol. 26, no. 23, Dec. 1987.

[29] C. Mead, *Analog VLSI and Neural Systems*. Reading, MA: Addison-Wesley, 1989.

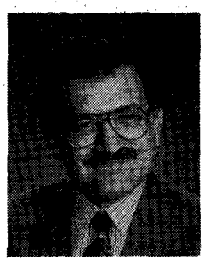
[30] B. Linares-Barranco, "Analog neural network VLSI implementations," Ph.D. dissertation, Texas A&M Univ., College Station, Dec. 1991.

[31] B. Gilbert, "A precise four-quadrant multiplier with subnanosecond response," *IEEE J. Solid-State Circuits*, vol. SC-3, no. 6, Dec. 1968.

[32] D. A. Wismer and R. Chattergy, *Introduction to Nonlinear Optimization*. Amsterdam: Elsevier, 1978.

Bernabé Linares-Barranco was born in Granada, Spain, on November 26, 1962. He did his elementary studies in Germany until 1975. He received the B.Sc. degree in electronic physics in June 1986, and the M.S. degree in microelectronics in September 1987, both from the University of Seville, Sevilla, Spain. He received his first Ph.D. degree in high-frequency OTA-C oscillator design in June 1990 from the University of Seville, Spain, and the second Ph.D. degree in analog neural network design in December 1991 from Texas A&M University, College Station.

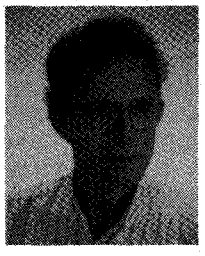
Since September 1991 he has been a Senior Researcher at the Analog Design Department of the Microelectronics National Center (CNM), Sevilla, Spain. His research interests are in the area of nonlinear analog and neural network microelectronic design.



Edgar Sánchez-Sinencio (S'72–M'74–SM'83) received the M.S.E.E. degree from Stanford University, Stanford, CA, and the Ph.D. degree from the University of Illinois at Champaign-Urbana in 1970 and 1973, respectively.

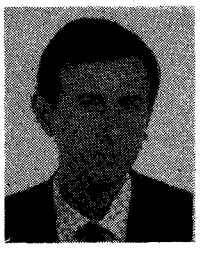
Currently, he is with Texas A&M University, College Station, as a Professor. He is the co-author of *Switched-Capacitor Circuits* (Van Nostrand-Reinhold, 1984), and co-editor of *Artificial Neural Networks: Paradigms, Applications, and Hardware Implementation* (IEEE Press, 1992). His interests are in the area of solid-state circuits, including CMOS neural network implementations, and computer-aided circuit design.

Dr. Sánchez-Sinencio was the General Chairman of the 1983 26th Midwest Symposium on Circuits and Systems. He has been Associate Editor for *IEEE Circuits and Systems Magazine* (1982–1984), for *IEEE Circuits and Device Magazine* (1985–1988), for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS (1985–1987), and for the IEEE TRANSACTIONS ON NEURAL NETWORKS (1990–). He is Guest Editor for IEEE TRANSACTIONS ON NEURAL NETWORKS Special Issues (1991–1993). He was a Committee Member of the Scientific Committee of the Sixth, Eighth, and Ninth European Conferences on Circuit Theory and Design (ECCTD), a Committee Member of the Technical Program of the IEEE International Symposiums on Circuits and Systems (ISCAS) in 1983, 1987, 1989, and 1990, and Chairman of the IEEE CAS Technical Committee on Neural Systems and Applications (1990–1991). He is currently a member of the IEEE CAS Board of Governors, Liason Representative, Region 9, IEEE/CAS, and an IEEE Fellow Member.



Angel Rodríguez-Vázquez (M'80) received the Licenciado en Física degree in 1977, and the Doctor en Ciencias Físicas degree in 1983, both from the University of Seville, Sevilla, Spain.

Since 1978 he has been with the Departamento de Electrónica y Electromagnetismo at the Universidad de Sevilla, where he is currently employed as an Associate Professor. His research interest lies in the fields of analog/digital integrated circuit design, analog integrated neural and nonlinear networks, and modeling of analog integrated circuits.



José L. Huertas (M'74) received the Licenciado en Física degree in 1969, and the Doctor en Ciencias Físicas degree in 1973, both from the University of Seville, Sevilla, Spain.

From 1970 to 1971 he was with the Philips International Institute, Eindhoven, The Netherlands, as a Postgraduate Student. Since 1971 he has been with the Departamento de Electrónica y Electromagnetismo at the Universidad de Sevilla, where he is currently employed as a Professor. His research interest lies in the fields of multivalued logic, sequential machines, analog circuit design, and nonlinear network analysis and synthesis.