CMOS CIRCUIT IMPLEMENTATIONS FOR NEURON MODELS

Bernabé Linares-Barranco[1,2], *Edgar Sánchez-Sinencio*[1],
Angel Rodríguez-Vázquez[2]

[1]Texas A&M University
Department of Electrical Engineering
College Station, Texas 77843-3128

[2]Dpto. Diseño Analógico
Centro Nacional de Microelectrónica
4104 Sevilla, SPAIN

ABSTRACT

Discussion of the mathematical neuron basic cells employed in popular neural network architectures and algorithms are presented. From the fundamental Mc Culloch–Pitts model up to the resultant model from Carpenter–Grossberg Neural Nets are discussed. We consider the viability of the implementation of these models in CMOS technology using transconductance and current–mode techniques. Simulations and experimental results from discrete and CMOS test–chips are presented.

I. INTRODUCTION

Artificial neural networks research was very active in the 60's. In the last few years, a reviving interest in this field [1] has rapidly grown. One of the main reasons of this renovated interest is the potential of hardware implementations of these nets. Artificial neural networks (ANNs) mainly consist of basic cells (neurons) heavily interconnected through variable weighted links (synapses). The neuron outputs are typically weighted (via synapses) and injected (through dendrites) to the neuron inputs. The ANNs can be trained to implement different tasks (i.e., classifiers, image, speech recognition). This training is usually accomplished by changing the weights (of the synapses) using strategic learning algorithms.

In this paper, we specifically deal with the most popular neuron models (without training) used in neural network architectures and algorithms (NNA). The focus will be on hardware implementation of neuron models used in NNA, and in emulation of biological systems.

In this highly interdisciplinary field of NN, authors have not agreed in selecting one unique neuron model for all applications. Therefore, regarding hardware implementations we should also consider the implementations of the different neuron models.

Two attractive circuit implementation techniques [2,3]: current–mode and transconductance–based will be considered. In some convenient cases, some extra current to voltage (or vice versa) converters will be included in the implementations.

II. NEURON MODELS

We will discuss several common neuron models in this section. Mathematical descriptions and block diagram representations will be utilized. This approach is technology–independent. Non–oscillatory and oscillatory models are discussed.

Basic Neuron Cell

One of the most conventional basic neuron cells [4] is shown in Fig. 1. The inputs X_{ik} and weights W_{ik} are related to the linear output S_k by [1]

$$S_k = \sum_{i=1}^{n} W_{ik} X_{ik} \tag{1}$$

the nonlinear ouput y_k is

$$y_k = f_n(S_k) \tag{2}$$

where in general $f_n(\cdot)$ is a limiting or threshold function. Several common limiting (also called activation) functions are shown in Fig. 2. The original McCulloch–Pitts neuron used only the binary (hard–limiting) function. Later development of learning algorithms required having a differentiable limiting function [1,13] such as Fig. 2(b) and the sigmoid of Fig. 2(c). One variation of the basic McCulloch–Pitts neuron involving additional dynamics is next discussed.

Basic Dynamic Neuron Cell

This cell is often used in solving optimization problems. Hopfield [5,6] has used it and made it popular in his optimization network solutions. The mathematical description of the dynamic neuron cell is given by

$$C_k \frac{dS_k}{dt} = I_{ink} - \frac{S_k}{R_k} + \sum_{i=1}^{n} W_{ik} f(S_i) \tag{3}$$

where I_{ink} is an independent input signal and $1/R_k$ is the self–relaxation term which makes the output S_k zero for no input. W_{ik} are the weights for the different inputs $f(S_i)$, and $f(\cdot)$ is a sigmoidal limiting function as shown in Fig. 2(c). The corresponding block diagram of (3) is shown in Fig. 3. The summation term in (3) is sometimes split into two, to consider the excitatory and inhibitory inputs. This yields

$$C_k \frac{dS_k}{dt} = I_{in} - \frac{S_k}{R_k} - \sum_{i=1}^{n_I} W_{ik} f(S_i) + \sum_{i=1}^{n_E} W_{ik} f(S_i) \tag{4}$$

Note that the growth rate (slope) of the activation function of Fig. 2(b) or 2(c) is not a fixed value for all applications.

The most general neural network topology is claimed by Grossberg [14]. According to this topology the differential equation describing the activity S_k of each of the neuron cells in the system is given by:

$$\begin{aligned} \frac{dS_k}{dt} = &-A_k S_k + (B_k - C_k S_k)\left[I_k + \sum_{k=1}^{n} D_{ik} f_i(S_i)\right] \\ &- (U_k S_k + F_k)\left[J_k + \sum_{k=1}^{n} G_{ik} g_i(S_i)\right] \end{aligned} \tag{5}$$

where I_k, J_k are the external inputs, D_{ik} and G_{ik} describe the weight of the interconnections between neurons, $f_i(\cdot)$ and $g_i(\cdot)$ are nonlinear functions, $A_k S_k$ is the self or forgetting term[2], and the constants B_k, C_k, U_k and F_k are responsible for the shunting properties of this net that provide automatic gain control and total activity normalization [14].

Equation (5) can be rewritten in order to merge the positive and negative terms as follows

[1] Note that a threshold value might be introduced by making one of the inputs equal to one, and the corresponding weight equal to the threshold value.

[2] A_k is related to the previous models by $A_k = 1/R_k C_k$.

CH2868-8/90/0000-2421$1.00 © 1990 IEEE

$$\frac{dS_k}{dt} = -A_k S_k + (H_k - L_k S_k)\left[E_k + \sum_{i=1}^{n} z_{ik} f_i(S_i)\right] \quad (6)$$

This equation can be represented schematically by the block diagram of Fig. 4. Most of the known neural network architectures and algorithms can be considered as special cases of this net (note that we are not considering the learning mechanism but only the short term memory STM). For example, the ART1 [15] and ART2 [16] STM equations correspond precisely to the block diagram in Fig. 4; additive models such as the ones by Hopfield [5], Anderson [17], McCulloch–Pitts [4] or Ackley [18] are simply obtained by eliminating the shunting terms in Fig. 4., i.e., $H_k = 1$, $L_k = 0$; feedforward nets like the back propagation algorithm [19] are obtained by properly making zero some of the weights z_{ij}. As mentioned before, different shapes can be used for the nonlinear function $f_i(S_i)$.

So far, we have discussed only what we call non–oscillatory neuron models, where the output of the neuron V_k is a nonlinear monotonic function of the internal potential S_k. Let us consider the other general case.

Analog Neural Oscillator Cell

For biological oscillatory neurons, the output is a firing sequence of pulses whose frequency is a nonlinear monotonic function of the internal potential. In this case, we have to substitute the function $f_k(\cdot)$ in Fig. 4 by a voltage–controlled–oscillator (VCO) whose output frequency is controlled nonlinearly by S_k, as shown in Fig. 5.

One of the closest models resembling biological neurons is the Hodgkin and Huxley model [7]. We discuss two simplified versions of the Hodgkin and Huxley model. First the *hysteretic neural type oscillator* cell [8–10] is presented. In this case the VCO of Fig. 5 is implemented according to the block diagram of Fig. 6. The corresponding characterization of Fig. 6 is given by

$$C_k \frac{dX_k}{dt} = -\frac{X_k}{R_k} - G_H H(X_k) - S_k \quad (7)$$

where $H(X_k)$ describes the hysteresis nature of the model

$$H(X_k) = \begin{cases} H^+ & \text{if} \quad X_k \le S_+ \\ -H^- & \text{if} \quad X_k \ge -S_- \end{cases} \quad (8)$$

and F_k is the oscillation frequency of the signal V_k. Thus F_k will have an oscillatory frequency dependent on S_k. Other alternatives to feed S_k into the VCO instead of connecting it to the summing input, as shown in Fig. 6, are to either make R_k or H_+ and H_- dependent on S_k [9]. Also note that in this model for oscillating neurons all the signals V_1 to V_n have equal amplitude but the oscillating frequencies are different.

The next model presented is the *FitzHugh–Nagumo*, which is one of the most complex, with potential application in emulation of biological systems. This cell can be characterized by a second–order system of the following form:

$$C_{k1} \frac{dX_{k1}}{dt} = -\frac{X_{k1}}{R_{k1}} + \frac{X_{k2}}{a_{k1}} + S_{k1} \quad (9a)$$

$$C_{k2} \frac{dX_{k2}}{dt} = -\frac{X_{k1}}{R_{k2}} - f(X_{k2}) + S_{k2} \quad (9b)$$

Where S_{k1} and S_{k2} represent the summation and integration of inputs coming from other neurons and/or independent inputs, and $f(X_{k2})$ was originally [11] suggested as a cubic polynomial. It has been shown [12] that $f(X_{k2})$ can be approximated as a piece–wise linear function as shown in Fig. 7(a). The block diagram representing eqs. (9a) and (9b) is shown in Fig. 7. The solution of (9) yields a Van der Pol when the time constants associated with (9a) and (9b) are equal.

III. Neuron Model Implementations:

In order to exploit the advantages and to avoid the drawbacks of VLSI CMOS integration we will consider two different approaches of implementation, the transconductance–mode and the current–mode. Both avoid the use of resistors and allow use of tunable synapses in the network. Also, we will not consider any shunting mechanism in our future discussion, therefore $H_i = 1$ and $L_i = 0$ in Figs. 4 and 5.

A. Non–Oscillatory

A.1. Transconductance Mode: This mode allows a very simple implementation for the integration device of the neuron (see Fig. 3), namely a capacitor as shown in Fig. 8. The saturation of the output voltage is accomplished by a limiting device (a simple inverter), whose characteristic is also shown in Fig. 8. Observe that $W_{ik} = G_{ik}$ can take any polarity [3].

A.2. Current–Mode: The current–mode comparator [9] shown in Fig. 9 allows the implementation of the limiting function of Fig. 9b. This element is symbolically represented in Fig. 9c. The complete implementation of eq. (3) using this technique is given in Fig. 10. Note that the weights of the interconnections are determined by the $(\frac{W}{L})$ ratio of the synaptic transistors M_i. The resitance R_k that implements the self–term of eq. (3) can be made using an MOS transistor working in its triode region. R_k can also be assumed composed as the parallel combinaiton of the output resistors of the transistors.

B. Oscillatory Neurons

B.1: Transconductance–Mode: The fundaments of this neural modeling that is based on a hysteretic element were presented elsewhere [9]. According to Fig. 5, the equation relating the input (S_k) and output V_k of the VCO would be of the type

$$\frac{dX_k}{dt} = -g_1 X_k - H(X_k) - g_2 S_k, \quad V_k = H(X_k) \quad (10)$$

where g_1 and g_2 are constants and $H(\cdot)$ is a hysteresis function. The frequency is controlled by S_k. We denote the value of this frequency by $F_k = f(S_k)$. A slight modification of this circuit in order to obtain a suitable CMOS implementation is shown in Fig. 11 [13]. Here $g_2 = 0$ and s_k is fed into the VCO through g_1, which is made to depend nonlinearly on S_k. This hysteretic model can be included in the more general model proposed by FitzHugh–Nagumo [11], by noting that in reality a hysteretic element is a simplification of a second order system [20]. An OTA–C implementation of FitzHugh–Nagumo's differential equation using nonlinear OTA–C techniques [3] has been presented elsewhere [21]. The set of differential equations is, according to Fig. 12,

$$\begin{aligned} g_{m1} S_k + g_{m1} X_k - g_{m3} V_k &= C_1 \dot{V}_k \\ - g_{m2} V_k - f_n(X_k) &= C_2 \dot{X}_k \end{aligned} \quad (11)$$

Where g_{m1}, g_{m2}, C_1 and C_2 are constants, X_k is an internal variable (same as X_{k1} in eq. (9)) of the VCO, V_k is the output (same as X_{k2} in eq. (9)) and $f_n(\cdot)$ is an N–shaped nonlinear function as shown in Fig. 12 together with the OTA–C implementation.

B.2: Current–Mode: Based on the current–mode comparator of Fig. 4, it is very simple to implement a current–mode hysteretic element [22] as shown in Fig. 13. A complete hysteretic–type current mode VCO model for Fig. 5 is shown in Fig. 14. It is essentially composed by a hysteretic element and a nonlinear transconductor, and it works under the same principle as the previous circuit of Fig. 11.

IV. Experimental and Simulated Results

The circuit of Fig. 11 was fabricated in a $3\mu m$ CMOS process (MOSIS). Fig. 15 shows the switching of the output between the on (oscillations) and off (no oscillations) states depending on the input, S_k, being above or below threshold.

At this time we built a discrete prototype for the FitzHugh–Nagumo circuit of Fig. 12. We can observe in Fig. 16 how the input (lower trace) makes the VCO to change between the firing state (on) and the non–firing (off) state.

Only simulation results are available for the current–mode circuits of Fig. 13 and 14. In Fig. 17, we can see the hysteresis loop generated by the circuit of Fig. 13, and in Fig. 18, we can see the on–off operation of the circuit of Fig. 14.

V. Conclusions

A discussion of neuron models using mathematical descriptions and block diagrams which are suitable for CMOS integration was presented. Two approaches have been used to

address this problem: the transconductance and the current-mode. Both of these techniques are suitable for IC designs. Several experimental and simulated results of the prototypes for $3\mu m$ CMOS processes (MOSIS) were presented.

REFERENCES

[1] J. A. Anderson, and E. Rosenfeld, editors. *Neurocomputing: Foundations of Research.* The MIT Press, Cambridge, MA 1988.

[2] K. A. Boahen, P. O. Pouliquen, A. G. Andreou, and R. E. Jenkins, "A Heteroassociative Memory Using Current-Mode MOS Analog VLSI Circuits", *IEEE Trans. Circuits Syst.*, Vol. 36, pp. 747-755, (Special Issue on Neural Networks), May 1989.

[3] E. Sánchez-Sinencio, B. Linares-Barranco, J. Ramírez-Angulo, A. Rodríguez-Vázquez, "OTA-Based Nonlinear Function Syntheses," *IEEE Journal of Solid-State Circuits*, December 1989.

[4] W. S. McCulloch, and W. Pitts, "A Logical Calculus of the Ideas Imminent in Nervous Activity", *Bulletin of Mathematical Biophysics*, Vol. 5, pp. 115-133, 1943.

[5] D. W. Tank, J. J. Hopfield, "Simple 'Neural' Optimization Networks: An A/D Converter, Signal Decision Circuit, and a Linear Programming Circuit", *IEEE Trans. Circuits Sys.*, Vol. CAS-33, No. 5, pp. 533-541, May 1986.

[6] J. J. Hopfield, D. W. Tank, " 'Neural' Computation of Decisions in Optimization Problems", *Biol. Cybern*, Vol. 52, pp. 141-152, 1985.

[7] A. L. Hodgkin, A. F. Huxley, "A Qualitative Description of Membrane Current and Its Application to Conduction and Excitation in Nerves", *J. Physics*, Vol. 77, pp. 500-544, 1952.

[8] G. Kiruthi, R. C. Ajmera, R. Newcomb, T. Yanis, H. Yazdain, "A Hysteretic Neural-Type Pulse Oscillator", *IEEE Int. Symp. Circ. Syst.*, pp. 1173-1175, 1983.

[9] B. Linares-Barranco, E. Sánchez-Sinencio, A. Rodríguez-Vázquez, and J.L. Huertas, "A Programmable Neuron Oscillator Cell", *IEEE Trans. Circuits Syst.*, (Special Issue on Neural Networks) Vol. 36, pp. 7576-761, May 1989.

[10] B. Linares-Barranco, E. Sánchez-Sinencio, R.W. Newcomb, A. Rodríguez-Vázquez, J.L. Huertas, "A Novel CMOS Analog Neural Oscillator Cell", *IEEE Int. Symp. Circ. Syst.*, pp. 794-797, May 1989.

[11] R. FitzHugh, "Mathematical Models of Excitation and Propagation in Nerves", *National Institute of Health, Technical Report*, 1967.

[12] J.P. Keener, "Analog Circuitry for the Van der Pol and FitzHugh-Nagumo Equations", *IEEE Trans. on Systems*, Man and Cybernetics, Vol. SMC-13, Sept./Oct. 1983.

[13] T. Kohonen, *Self-Organization and Associative Memory*, Springer-Verlag, 1988.

[14] S. Grossberg, "Nonlinear Neural Networks: Principles, Mechanisms, and Architectures," *Neural Networks*, Vol. 1, pp. 17-61, 1988.

[15] G. A. Carpenter and S. Grossberg, "A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine," *Computer Vision, Graphics and Image Processing*, Vol. 37, pp. 54-115, 1987.

[16] G. A. Carpenter and S. Grossberg, "ART2: Self-organization of Stable Category Recognition Codes for Analog Input Patters," *Applied Optics*, Vol. 26, pp. 4921-4930, December 1987.

[17] J. A. Anderson, J. W. Silverstein, S. R. Ritz and R. S. Jones, "Distinctive Features, Categorical Perception, and Probability Learning: Some Applications of a Neural Model," *Psychological Review*, Vol. 84, pp. 413-451, 1977.

[18] D. H. Ackley, G. E. Hinton and T. J. Sejnowski, "A Learning Algorithm for Boltzmann Machines," *Cognitive Science*, Vol. 9, pp. 147-169, 1985.

[19] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning Internal Representatiions by Error Propagation," E. E. Rumelhart and J. C. McClelland (Eds.), *Parallel Distributed Processing*, Cambridge, MA:MIT Press.

[20] L. O. Chua, C. A. Desoes and E. S. Kuh, *Linear and Nonlinear Circuits*, McGraw-Hill, New York, 1987.

[21] E. Sánchez-Sinencio and B. Linares-Barranco, "Circuit Implementation of Neural Fitzhugh-Nagumo Equations," *Proc. Midwest Symp. on Circuits and Systems*, 1989.

[22] Z. Wang and W. Guggenühl, "Novel CMOS Current Schmitt Trigger," *Electronics Letters*, 24, pp. 1514-1516, 1988.

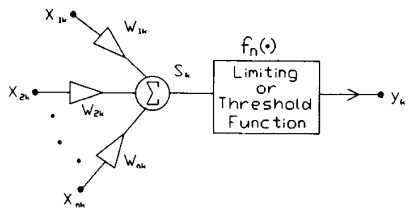

Fig. 1. Basic Neuron Cell

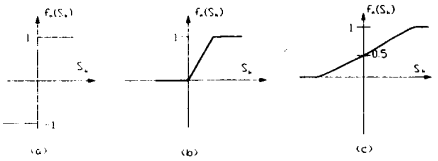

Fig. 2. Activation Functions: (a) Binary, Hard-Limiting or Signum, (b) Threshold Logic, (c) Signoidal

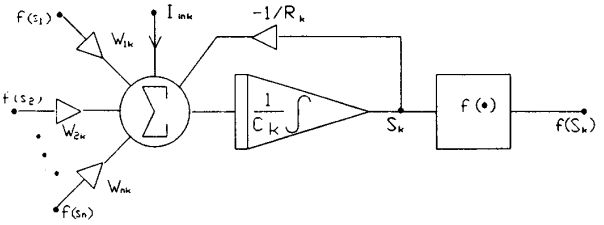

Fig. 3. Dynamic Basic Neuron Cell Architecture

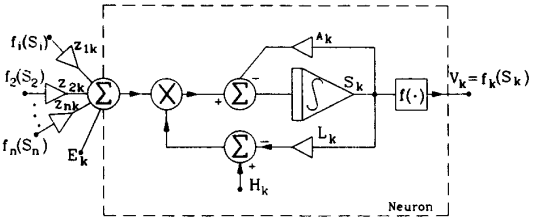

Fig. 4. Block Diagram for General Non-Oscillatory Neuron Model

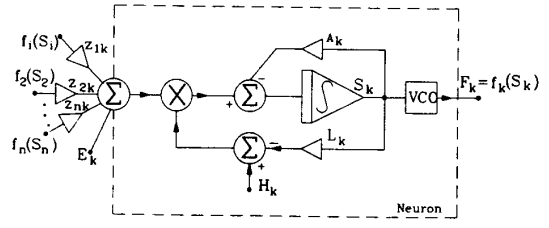

Fig. 5. Block Diagram for General Oscillatory Neuron Model

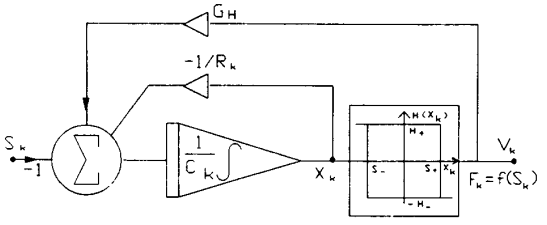

Fig. 6. Hysteretic Neural Oscillator Cell

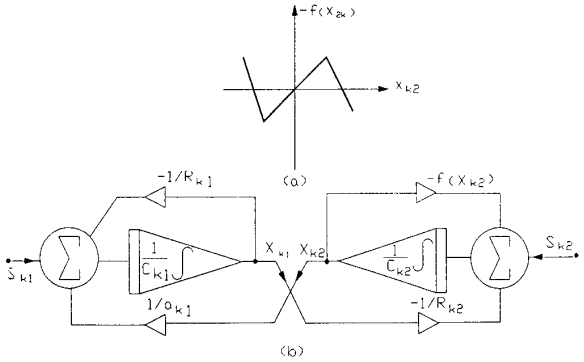

Fig. 7. Fitzhugh–Nagumo Neuron; (a) Nonlinear Function $-f(X_{sk})$, (b) Block Diagram or Neuron

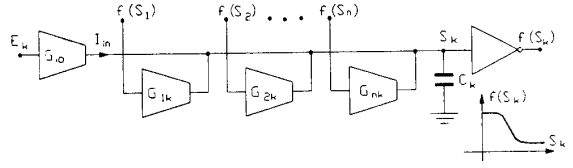

Fig. 8. OTA–C Implementation of Non–Oscillatory Neural Network

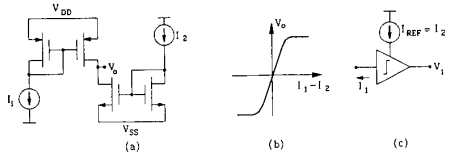

Fig. 9. Current–Mode Comparator; (a) Circuit, (b) Transfer Function, (c) Symbol

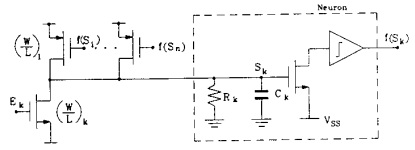

Fig. 10. Current–Mode Implementation of Nonoscillatory Neural Network

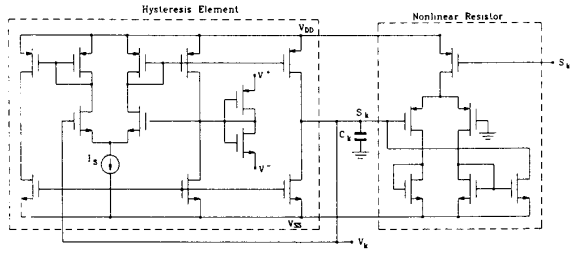

Fig. 11. OTA–C CMOS Implementation of Oscillatory Neuron

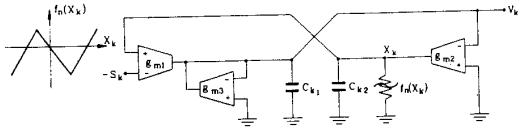

Fig. 12. OTA–C Implementation of Oscillatory FitzHugh–Nagumo Neuron

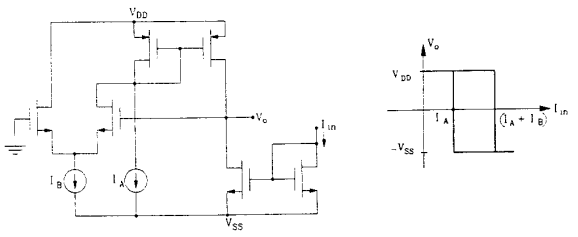

Fig. 13. Current–Mode Hysteresis Element

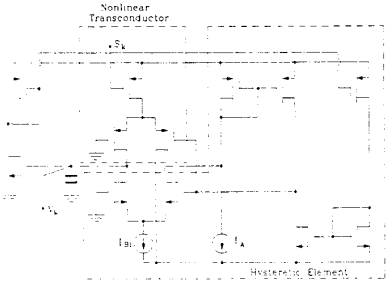

Fig. 14. Current–Mode Oscillatory Neuron

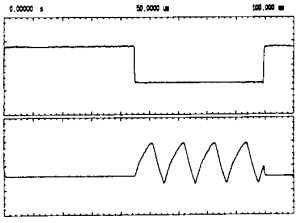

Fig. 15. Experimental Results for CMOS OTA–C Hysteretic Neuron

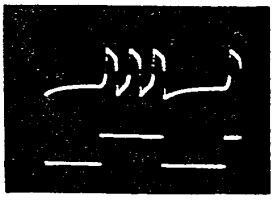

Fig. 16. Experimental Results for OTA–C Fitzhugh–Nagumo Neuron

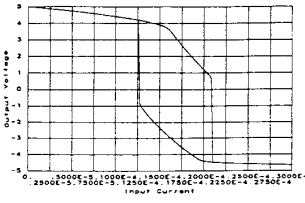

Fig. 17. Simulated Current–Mode Hystersis Comparator

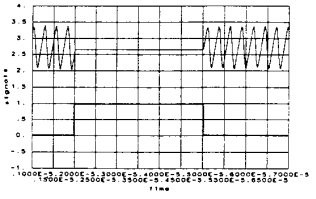

Fig. 18. Simulated Results for Current–Mode Oscillatory Neuron