CMOS CIRCUIT IMPLEMENTATIONS FOR NEURON MODELS

Bernabe Linares-Barranco\textsuperscript{1,2}, Edgar Sánchez-Sinencio\textsuperscript{1},
Angel Rodríguez-Vázquez\textsuperscript{2}
\textsuperscript{1}Texas A\&M University
Department of Electrical Engineering
College Station, Texas 77843-3128
\textsuperscript{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 will be utilized. From the fundamental McCulloch–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 reenewed interest is the potential of hardware implementations of these nets. Artificial neural networks (ANN's) 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

$$S_{k} = \sum_{i=1}^{n} W_{ik}X_{ik} \quad (1)$$

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.

The nonlinear output $y_k$ is

$$y_k = f_n(S_k) \quad (2)$$

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

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

where $I_{in}$ 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

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

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

$$\frac{dS_k}{dt} = -A_k S_k + (B_k - C_k S_k) \left[ I_k + \sum_{i=1}^{n} D_{ik} f_i(S_i) \right]$$

$$-(U_k S_k + F_k) \left[ J_k + \sum_{i=1}^{n} G_{ik} g_i(S_i) \right] \quad (5)$$

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\textsuperscript{2}, and the constants $B_k, C_k, U_k$ and $F_k$ are responsible for the shunting properties of the 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

\textsuperscript{2} $A_k$ is related to the previous models by $A_k = 1/R_k C_k$. 

\textsuperscript{CH12868-9/90/0000-2421$\text{IEEE}$ © 1990 IEEE
\[ \frac{d s_k}{dt} = -A_k s_k + (H_k - L_k s_k) \left[ E_a + \sum_{i=1}^{n} z_{ik} f(S_i) \right] \]  

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 \( z_{ik} \). As mentioned before, different shapes can be used for the nonlinear function \( f(S_i) \).

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 \( f(S_k) \) 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:

\[ C_k \frac{d X_k}{dt} = X_k \frac{G_k}{R_k} \frac{H(X_k) - S_k}{H(X_k)} - G_k H(X_k) \]  

where \( H(X_k) \) describes the hysteresis nature of the model.

\[ H(X_k) = \begin{cases} H^+ & \text{if } X_k \leq S_k \\ H^- & \text{if } X_k \geq -S_k \end{cases} \]  

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 \( H_k \) dependent on \( S_k \). Also note that in this model for oscillating neurons all the signals \( V_k \) to \( V_l \) 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:

\[ C_{G_k} \frac{d X_k}{dt} = -X_k + \frac{X_k^3}{a_k} + S_k \]  

\[ C_{G_k} \frac{d X_k^2}{dt} = -X_k^2 + f(X_k) + S_k \]  

Where \( S_k \) and \( S_k^2 \) represent the summation and integration of inputs coming from other neurons and/or independent inputs, and \( f(X_k) \) was originally [11] suggested as a cubic polynomial. It has been shown [12] that \( f(X_k) \) can be approximated as a piece-wise linear function as shown in Fig. 7a. 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 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 \( H_k = 1 \) and \( L_k = 0 \) 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. 8) 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,

\[ g_{m1} s_k + g_{m1} X_k - g_{m2} V_k = C V_k \]  

\[ -g_{m2} V_k - f(X_k) = C X_k \]  

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 \( f(.) \) is an \( N \)-shaped nonlinear function as shown in Fig. 12 together with the OTA–C implementation.

**B. Oscillatory Neurons**

B.1. **Transconductance Mode:** The fundamentals 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

\[ \frac{d X_k}{dt} = -g_1 X_k - H(X_k) - g_2 S_k + V_k = H(X_k) \]  

where \( g_1 \) and \( g_2 \) are constants and \( H(.) \) is a hysteresis function.

B.2. **Current–Mode:** Based on the current–mode comparator of Fig. 4, it is very simple to implement a current–mode hysteric 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 transconductance, 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 3μm 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 Figs. 13 and 14. In Fig. 17, 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 3μm CMOS processes (MOSIS) were presented.

REFERENCES


