Scalable Closed-Boundary Analog Neural Networks

Sied Mehdi Fakhraie, *Member, IEEE*, Hamed Farshbaf, and Kenneth Carless Smith, *Life Fellow, IEEE*

***Abstract*—In many pattern-classification and recognition problems, separation of different swarms of class representatives is necessary. As well, in function-approximation problems, neurons with a local area of influence have demonstrated measurable success. In our previous work, we have shown how intrinsic quadratic characteristics of traditional metal-oxide-semiconductor (MOS) devices can be used to implement hyperspherical discriminating surfaces in hardware-implemented neurons. In this work, we further extend the concept from quadratic forms to more-arbitrary closed-boundary shapes. Accordingly, we demonstrate how intrinsic characteristics of submicron MOS devices can be utilized to implement efficient pattern discriminators for various applications and, through representative simulations, show their success in some typical function-approximation problems. Further, we offer two mathematical interpretations of possible roles for these networks: Geometrically, we show that our networks employ closed hypercone shapes as their discriminating surfaces; analytically, we show that a set of these synapses connected to a common integrating body calculates the distance between their inputs and weight vectors using a power norm. The feasibility of the idea is practically investigated by design, implementation, and test of a three-dimensional (3-D) closed-boundary pattern classifier, fabricated in 0.35-μm complimentary MOS, whose results are reflected in this work.**

***Index Terms*—two-dimensinal (2-D) and three-dimensional (3-D) pattern classification, alpha-rule devices, closed-boundary decision surfaces, function approximation, implementation, neural networks (NNs), quadratic, scalable, submicron complimentary metal-oxide-semiconductor (CMOS).**

I. Introduction

NEURAL-NETWORK (NN) models have shown great promise in providing solutions for many difficult pattern-recognition and function-approximation problems. In addition, they have the advantage of being parallel in nature, based on their use of many simultaneously operating simple processing units. Their practical success has attracted a lot of attention toward their implementation using high-performance parallel hardware. The analog nature of biologic NNs and the availability of easily interconnected digital systems has also inspired the idea that only through massively parallel low-power analog-processing digital-interfaced implementations will the real power of massively parallel artificial NNs (ANNs) be convincingly demonstrated.

Although NN research was originally inspired by imitation of biological NNs, the major emphasis is now on getting higher performance by better application of silicon technology and, correspondingly, better utilization of the resources available in mainstream complimentary metal-oxide-semiconductor (CMOS) processes. Better understanding of the way in which neural signal processing is performed may guide us to invent scalable solutions that are able to survive in a presumably ever-shrinking CMOS submicron world.

In Lippmann's original work [3], he clearly discussed different kinds of discriminating boundaries created by ANNs. He also demonstrated pattern-classification properties of single and multilayer perceptrons (MLPs). After the original theoretical work and such practical clarifications as were made on the theory, MLPs were used in numerous applications in different engineering and scientific fields, namely pattern and optical-character recognition, speech recognition, and function approximation.

In many of these applications, different swarms of clustered data are considered for classification. In addition, some of the limitations of MLPs further inspired researchers to look for other possible alternatives. Among the difficulties encountered, one can mention the problems of MLPs in dealing with large data spaces, local-minimum traps, and the small value of the gradient function in plateaus, which all conspire to further elongate the training process. In addition, the desire for more efficient pattern classifiers and the need for a platform to account for probabilistic data-manipulation cases, among others, has led researchers to the idea of using radial basis gunctions (RBFs) [4], [5]. RBFs are normally two-layer networks with regular inputs, a hidden layer with Gaussian units, and output units that are normally linear. The Gaussian units calculate the distance between an input pattern and the vectors that represent their centers. They then use a Gaussian activation function to provide an output that represents a similarity measure between the inputs and the reference vectors. The norm usually applied is Euclidean distance, which tends to classify the inputs into hyperspherical domains. Various attempts at the implementation of RBFs have been made [6]–[8]. In some, modified RBFs have been proposed to facilitate fabrication in a CMOS process [8], [9]. These include an attempt to use other norms to remove the need for the squaring operation required in the calculation of Euclidean distance [9], [10]. Also, some attempts have been made to obtain local classifiers with closed-boundary discriminating surfaces using clusters of three-layer MLPs, each set acting as a local classifier [11].

Another class of closed-boundary discriminating-surface neurons that several researchers have explored is that of quadratic NNs (QNNs) [12]. In [13], their added capabilities in comparison to MLPs with linear synapses have been shown. In addition, the fact that closed quadratic forms include a wide range of closed shapes, ranging from very-narrow ellipses to fat ones and circles in multidimensions is another possibility

Manuscript received January 23, 2002; revised November 26, 2002.

S. M. Fakhraie and H. Farshbaf are with the Department of Electrical and Computer Engineering, University of Tehran, Tehran, Iran.

K.C. Smith is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M2N 6H9, Canada.

Digital Object Identifier 10.1109/TNN.2004.824415

1045-9227/04$20.00 © 2004 IEEE

that has attracted much attention. This provides an extra degree of freedom that can be utilized to further improve network performance in some applications [14].

Overall, the generalized view of a neuron as an information-fusion-and-decision-making machine is an idea considered by several researchers as a way to further facilitate employment of neural units at different stages of information processing [15], [16]. In [15], the idea of an information-fusion machine is applied to the concept of a feedforward information-processing neuron. In [16], the concept of a conic-section function network (CSFN) is introduced, by which a uniform propagation rule for MLPs and RBFs is provided.

A. *Review of Some Existing Implementations*

Implementation-wise, less-linear more-imperfect multipliers are much easier to construct than ideal ones. However, the major issue—their usefulness—remains a target of many researchers. In [17] and [18], the effects of finite mathematical precision on the training time and the occurrence of output errors in ANNs are reviewed. In [22], the effects of hardware nonidealities have been considered. In [19], it is shown that a nonideal multiplier is able to perform neural operations, while [20] shows that using a proper system design and training algorithm allows an analog network to be trained to perform its duties in the presence of network nonidealities.

As well, the idea of using other distance-measuring norms to replace the more conventional Euclidean distance has been considered by various researchers. The goal of this effort is to provide easier, or faster, implementations of a network. However, it is essential to prove and/or demonstrate that the new system using a different norm is equally capable, or at least can perform the job acceptably. Based on simplified norms or on an attempt to construct almost-perfect quadratic elements, several implementations have been proposed for RBFs. In [6], the quadratic functions required for distance calculation have been approximated with Gaussians that are implemented with CMOS inverters employing floating-gate transistors. In [23], an attempt is made to construct RBFs by using standard metal-oxide-semiconductor (MOS) squaring and exponential circuits. In [24], a compact Euclidean-distance calculator is introduced by using floating-gate transistors. In this work, chip-in-the-loop training processes have been used to solve the problems introduced by hardware nonidealities. In [7], based on a CMOS squaring circuit, a programmable circuit for RBFs is designed, which employs the natural characteristics of MOS devices for squaring operations.

In several other reports, compatibility with very-large-scale integration (VLSI) implementation is a greater concern. In [9], a modified RBF model (MRBF) is proposed, which replaces the standard Gaussian basis functions with piece-wise linear ones for easier current-mode VLSI implementation. The piece-wise linear property of these functions is implied from the replacement of conventional Euclidean distance by a Manhattan absolute-value norm, one in which the sum of the absolute values of the differences between the components of the input and reference vectors is calculated. The feasibility of this approach has been verified by solving practical examples. In [11], from a function-approximation point of view, advantages of local cluster (LC) networks is discussed. To remove the difficulties of RBF implementation, small localized MLPs are formed, each of which has a localized bump in its domain of activity. Then, these networks are combined together to form a larger MLP constructed of smaller localized sub-blocks. In [25], a circuit is proposed for hardware implementation of CSFNs. Simple CMOS circuits for voltage squaring and current summation and mirroring have been used. In [1] and [2], an attempt has been made to use natural quadratic characteristics of MOS transistors to implement quadratic networks.

In the present work, the concept of quadratic networks is further extended to closed-boundary discriminating-surface networks and it is shown how the natural characteristics of velocity-saturated MOS devices can be utilized toward that end.

B. *This Work*

Overall, experiences obtained by many researchers, over many applications and various architectures, have shown that the major factor that accounts for the advantages of RBFs or QNNs over MLPs is related primarily to their closedness property [9], [26]. Accordingly, in this work, we illustrate and compare properties of various closed-boundary architectures at a system level and demonstrate their usage. This has further motivated us to see if some useful features of reduced-feature-size MOS devices could be used to implement scalable closed-boundary ANNs.

Efficiency of implementation of NNs is an important issue. In its pursuit, we have already shown that the quadratic characteristic equation of a large-feature-size MOS transistor can be employed to efficiently implement quadratic neurons [1], [2]. Accordingly, we have shown that MOS compatibility with direct utilization of MOS device characteristics in the implementation of these networks shows great promise. However, the nonscalability of the quadratic I–V–characteristics of an individual MOS device would seem to limit the usefulness of such neurons or, otherwise, makes "quadraticness" expensive to implement. Therefore, the question arises: In reduced-feature-size MOS technologies, which property, if maintained, can possibly be used in the implementation of scalable-MOS-compatible efficient neurons?

In this work, we elaborate on how feature-size scaling affects the quadratic behavior of submicron-implemented neurons. We show that all important properties of QNNs in function approximation, pattern classification, and closed-boundary-region forming are maintained in scaled submicron technologies. However, for submicron networks, closedness of the discriminating surface will be the main feature, rather than the special quadratic form, available in traditional CMOS. In all of this, the critical property that is maintained in scaled enhancement MOS devices is threshold operation. This property, by which a device can provide a current that has a power relationship with the difference between the gate and the threshold voltages, is the major feature that characterizes the networks discussed in this paper.

In Section II, we describe our definition of a CMOS-scalable norm for the operation of VLSI-compatible NNs. We demonstrate analytic and geometric interpretations for the

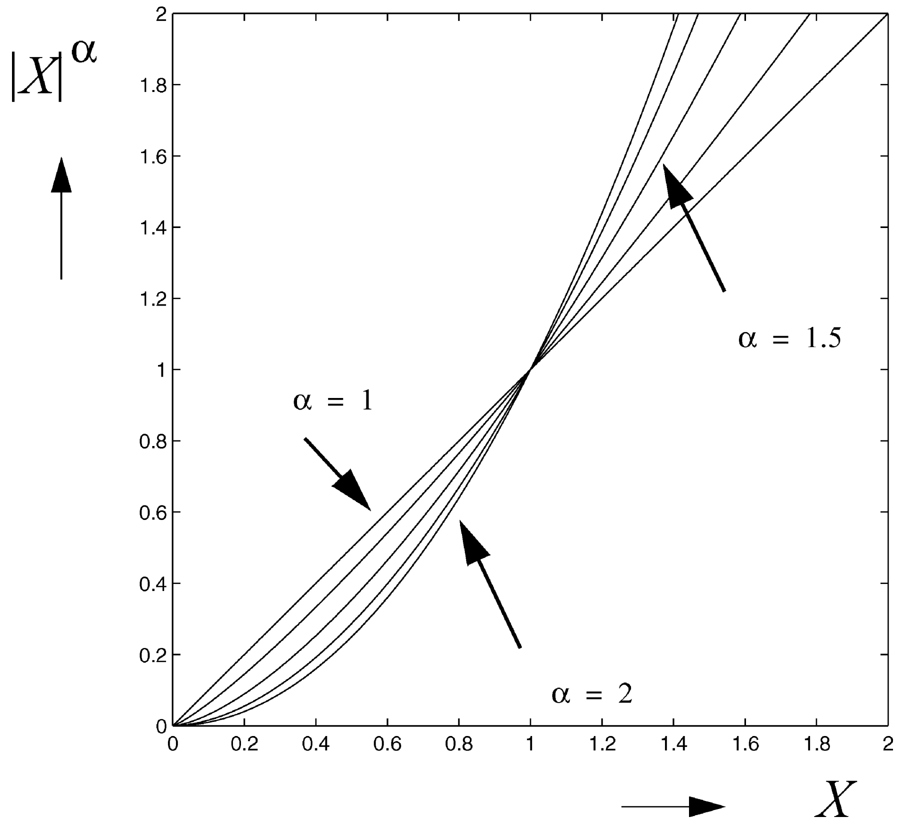


Fig. 1. Geometrical interpretation of the proposed α norms. As seen, a greater difference results in a greater norm for all values of $1 \leq \alpha \leq 2$.

performance of those networks using the introduced property. Then, we investigate the operation and usefulness of these scalable closed-boundary neurons by simulating some neural classifiers and localized-output neurons in typical pattern-classification and function-approximation applications. In Section III, we describe the characteristics of submicron MOS devices and a first-order analytic model for their operation is also discussed. Then, we demonstrate how the functionality of a possible synaptic circuit with possible application to closed-boundary NN scales with submicron devices. Then, circuit design of different neural units are discussed. Section IV describes hardware-implemented one-dimensional (1-D), two-dimensional (2-D), and three-dimensional (3-D) closed-boundary discriminating-surface neurons fabricated in a 0.35 μm CMOS technology. Measured results are shown for 1-D, 2-D, and 3-D pattern discriminators. Conclusions and suggestions for future work complete this paper.

II. Definition and Operation of Closed-Boundary NNs

A. New Generalized Equation

Two different norms are typically used in the implementation of RBFs: unsupervised competitive-learning systems and in general neural pattern classifiers. One is the Euclidean distance $d = \sum_{i=1}^{N}(x_i - w_i)^2$, where x_i are inputs and w_i are stored weights. The other is the Manhattan distance $d = \sum_{i=1}^{N}|x_i - w_i|$. While the Euclidean distance theoretically provides several good noise-cancellation properties, the Manhattan distance measure is much easier to implement and has been used more frequently by VLSI designers.

Here, we approach the problem from another point of view, posing the question: If the characteristic function of the available devices in a deep submicron CMOS technology is used, what networks are naturally available and what are their properties? Our response is based on the fact that $I_D = \kappa \cdot (V_{\mathrm{GS}} - V_{\mathrm{TH}})^\alpha$ is a fairly good first-order representation of deep submicron devices, where κ, V_{TH}, and α are found by fitted measurements [27]. Then, by considering the operation of circuits using such elements, an input–output relation proportional to $(in - w)^\alpha$ is obtained. In this section, we investigate the usefulness of employing such an equation, both in a theoretical development and in two major applications, namely function approximation and pattern classification.

Analytic Interpretation: The generalized deep-submicron CMOS-implementable norm that we consider is

$$\mathrm{distance}(x, w) = ||x - w|| = \sum_{i=1}^{N} |x_i - w_i|^\alpha. \tag{1}$$

For $\alpha = 2$ this converts to the Euclidean distance, while for $\alpha = 1$ this converts to the Manhattan norm of the two vectors. Therefore, generally, this equation analytically expresses an α norm measuring the distance between stored and input patterns. To verify the eligibility of this norm, we must show that it is monotonic; that is, for a larger difference, it provides a larger distance and for smaller difference, it provides a smaller distance. Accordingly, it can legitimately be used as a measurement norm. This fact is illustrated in Fig. 1.

Geometrical Interpretation: Concerning decision boundaries formed by neurons constructed by α-norm synapses,

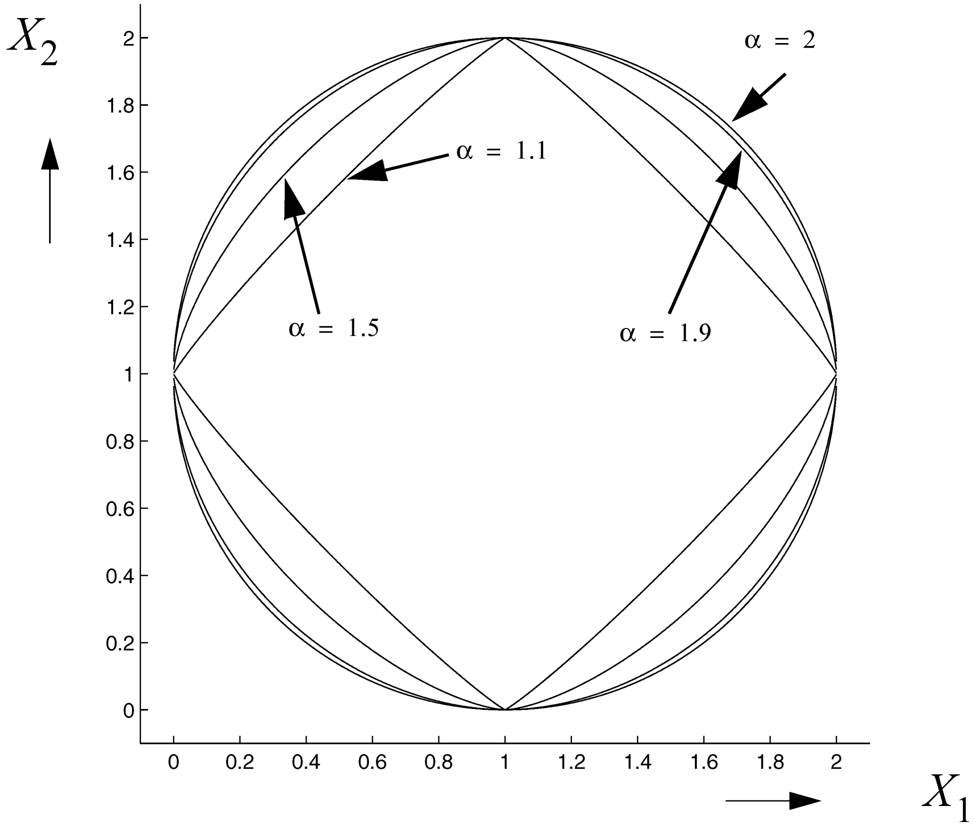


Fig. 2. Discriminating borders formed for various values of α. As seen, closed boundaries are formed for all values. The case $\alpha = 1.1$ is very similar to the linear case of $\alpha = 1$.

we note again that $\alpha = 2$ corresponds to the regular Euclidean distance definition. Accordingly, these boundaries form hyperspherical decision areas in the solution space. For the case $\alpha = 1$, an absolute-value differential amplifier of the kind of our circuit converts the governing equation into $G(x_1, x_2) = |x_1 - w_1| + |x_2 - w2| - k = 0$. Here again, closed-boundary discriminating surfaces are constructed by using interconnected pieces of straight lines (hyperplanes in general), similar to those shown in Fig. 2, for $\alpha = 1.1$.

Now, we consider the situation in which α varies between one and two. As well, we will consider only the case in which the absolute value of the difference signal is used; otherwise, the solution applies differently to different regions of the input space, similar to the cases discusses in [1]. The general governing equation then becomes

$$(G(x_1, x_2) = |x_1 - w_1|^\alpha + |x_2 - w_2|^\alpha - k = 0), \qquad Or, |x_2 - w_2|^\alpha = k - |x_1 - w_1|^\alpha \quad (2)$$

or, equivalently,

$$x_2 = w_2 \pm \sqrt[\alpha]{k - |x_1 - w_1|^\alpha}. \quad (3)$$

The resulting discriminating borders have been plotted in Fig. 2 for several values of α, namely 1.1, 1.5, 1.9, and 2. We foresee that the case $\alpha = 1$ is most likely to be found in very deep submicron technologies.

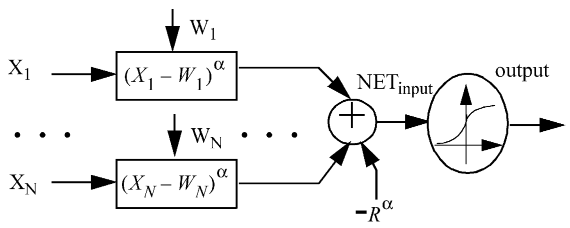


Fig. 3. Block diagram of a general scalable closed-boundary analog NN (SCBANN).

B. Design of the Neural Unit

The neurons that we have designed are composed of three major units, as shown in Fig. 3. The operation of these units is described by

$$\mathrm{NET}_{\mathrm{input}} = \sum_{i=1}^{N} |x_i - w_i|^\alpha - R^\alpha \quad (4)$$

$$\mathrm{Output} = \frac{1}{1 + \exp(-(\eta \cdot \mathrm{NET}_{\mathrm{input}}))}, \qquad \eta = 5 \quad (5)$$

where η is a normalization factor, the higher, the sharper is sigmoid transition region. The inputs are applied to the synaptic units. Each synaptic unit has an associated weight value stored within it. From a systems point of view, the absolute-value difference is computed and then its α power is calculated. These operations constitute an α-power-norm unit. Then the outputs of the synaptic units are added together in a neuron-accumulation

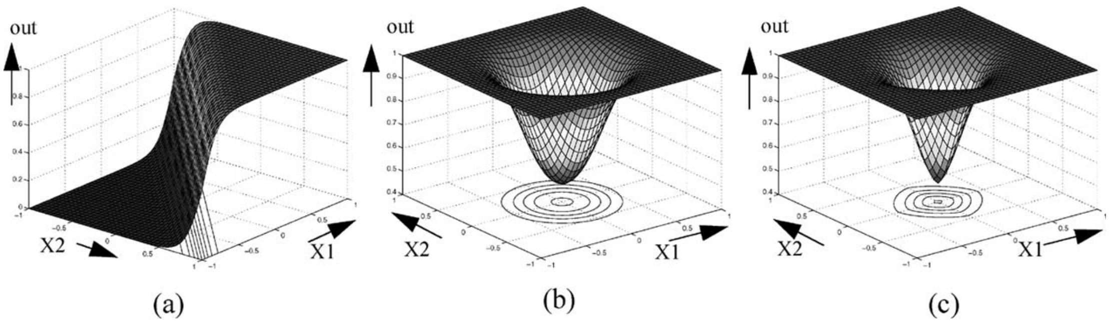


Fig. 4. Output–input graphs and shapes of discriminating borders for three different kinds of NNs with two input variables. (a) Linear-synapse neuron. (b) Quadratic-synapse neuron with $\alpha = 2$. (c) SCBANN with $\alpha = 1.5$. $\mathrm{R} = 0.25$.

unit. As well, a bias term represented by R^{α} is subtracted from the stimulations collected from synapses. This value determines the radius of the isolating area and, in case of QNNs, radius of the discriminating circle. Finally, the total input signal is applied to the output sigmoid unit that provides a saturating output in proportion to its net input received. Based on the η value used, for limited input stimulations, the outputs will remain at values above (below) the theoretical minimum (maximum) value zero (one). While this might look like an inconvenience in comparison to a hard-limiter function, we draw readers' attention to the fact that limited slope of sigmoid input–output characteristic has useful properties in successful chain-rule based training of a network, making derivative propagation and escape from flat plateaus more probable.

C. Training of Scalable Closed-Boundary Analog NNs (SCBANNs)

The α-norm units that we are using (for example, as shown in Fig. 3) have differentiable characteristics except at a bounded number of points. Therefore, we can employ a chain-rule differentiation algorithm to calculate the derivatives of an output with respect to different inside-circuit variables (or, more specifically, weights). This implies that a modified version of the error-back-propagation (BP) algorithm can be used to train these networks. We note that for a hardware-training algorithm, a mixture of chip-in-the-loop and weight-perturbation algorithms should be used to accommodate implementation variations within the chips, as well as environmental changes [21], [22].

D. Operation of Simple SCBANNs

From a geometrical point of view, each kind of neuron provides a specific type of decision boundary. Fig. 4 compares these boundaries for three different kinds of neurons, each with two inputs, one bias term, and one output. As shown, QNNs and SCBANNs both provide outputs with closed boundaries and localized bump functions for efficient function-approximation purposes. In the following sections, we elaborate further on their comparison and the relative efficiencies they provide. Interestingly, the type of boundary having the parameter α closer to one, which we foresee will be naturally obtained by further scaling of CMOS technologies, has been adopted in [9] as the authors' chosen way to ease implementation in conventional CMOS processes.

E. SCBANNs in Pattern-Recognition Applications

In this section, we simulate and compare the operation of MLPs, QNNs, and SCBANNs in separation of four distinct data clusters in a sample-classification problem.

Pattern-Classification Problem: In the pattern-classification problem discussed here, there are four classes of data and each class is associated with one output neuron of the network. If an input of the network belongs to one of the classes, its corresponding output will be "high" and all other outputs will have "low" values. For simulation purposes, the input space is set to be 2-D and data classes are defined as points around (0.3,0.3), (0.3,0.7), (0.7,0.3), and (0.7,0.7) centers, as shown in Fig. 5(a).

The MLP networks employed have two layers. There are two inputs to cover the 2-D input plane and four output neurons, one for each distinct class of data, with four neurons in the hidden layer. As well, for MLPs, networks with six and eight hidden neurons have been investigated. For QNNs and SCBANNs, considering the closedness of their output-discriminating boundaries, networks composed of only four neurons, each with two inputs, are sufficient to isolate four closed areas in the 2-D input space. This further corresponds to a reduced hardware area, power, and complexity.

Fig. 5 shows the definition of data classes and operation of the networks with linear, quadratic, and alpha-power synapses in the solution of a four-member pattern-recognition problem. Table I provides a summary of the results presented in Fig. 5. Obviously, the linear network is able to perform the defined pattern-recognition problem. However, SCBANNs show their capabilities in solving this problem at a reduced cost with only four neurons.

F. SCBANNs in Function-Approximation Applications

NNs are among the best-known function-approximator paradigms. This fact is attributed to their ability to construct localized bumps [30] or, in more rigorous terms, impulse basis functions. They then make the weighted sum of the constructed impulses fit the required function. In fact, any well-defined function can be approximated, with arbitrarily chosen accuracy, as a summing integral of shifted delta functions when they are appropriately weighted [1]. Therefore, we expect that neurons with localized fields of response will perform satisfactorily in this application. However, quantified observation and analysis of the effect of being quadratic remains to be done. Here, the accuracy; number of units required; convergence; and training

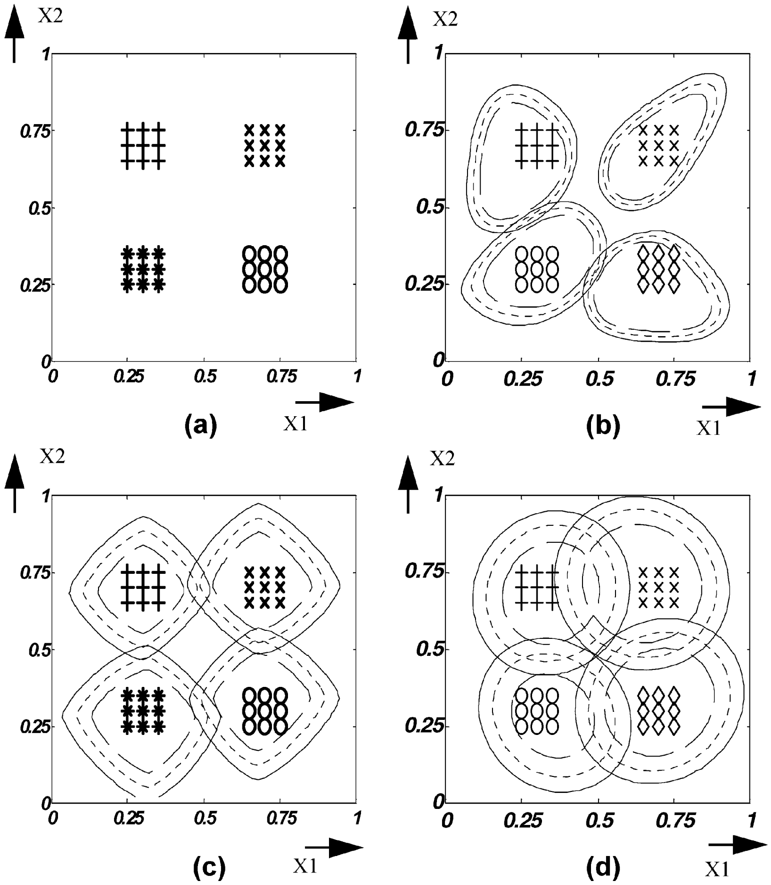


Fig. 5. Pattern-recognition problem. (a) Data classes used for this pattern-recognition problem and equivalue curves of (b) MLP, (c) SCBANN, and (d) QNN.

TABLE I
SUMMARY OF SIMULATION ERRORS FOR THE PATTERN-RECOGNITION PROBLEM

Network type	MLPs (15000 epochs)			QNNs (15000 epochs)	SCBANNs (15000 epochs)
Error	4 hidden neurons	6 hidden neurons	8 hidden neurons	No hidden neurons	No hidden neurons
Neuron one	0.00005	0.00049	0.00015	0.0020	0.0215
Neuron two	0.0423	0.0016	0.00019	0.0030	0.0212
Neuron three	0.0312	0.0224	0.00029	0.0016	0.0227
Neuron four	0.0403	0.0175	0.00010	0.0040	0.0212

cost in terms of time, operation count, and complexity will be investigated. In this case study, we approximate one period of a sinusoidal function by using one of the three above-mentioned NNs, namely MLPs, QNNs, and SCBANNs. Each network consists of two layers, one hidden and one output. The output unit is a linear neuron and the synapses are linear. Fig. 6 shows the target function (a), and the approximations (b), (c), and (d) made by the three NNs. Table II summarizes the simulation results in terms of number of neurons and synapses used, operation count and type, number of training epochs, and the overall residual error. Note, in particular, that the hardware-implementation simplicity of SCBANNs outweighs the mathematical efficiency of QNNs.

III. SUBMICRON IMPLEMENTATION OF CLOSED-BOUNDARY NNS

A. Operation of MOS Circuits in the Submicron Region

Historically, the quadratic model of operation of an MOS transistor in saturation has been used for analytic treatment of MOS circuits. This model, originally proposed by Schockley, predicts a quadratic relation between the so-called "effective voltage," gate-to-source voltage minus a threshold, and drain output current while the device is in saturation. Now, with downscaling of the minimum feature size of MOS devices, basic assumptions of the simpler models have changed. For example, historically, scaling from 2 to 0.8 $\mu\mathrm{m}$ was applied by adopting

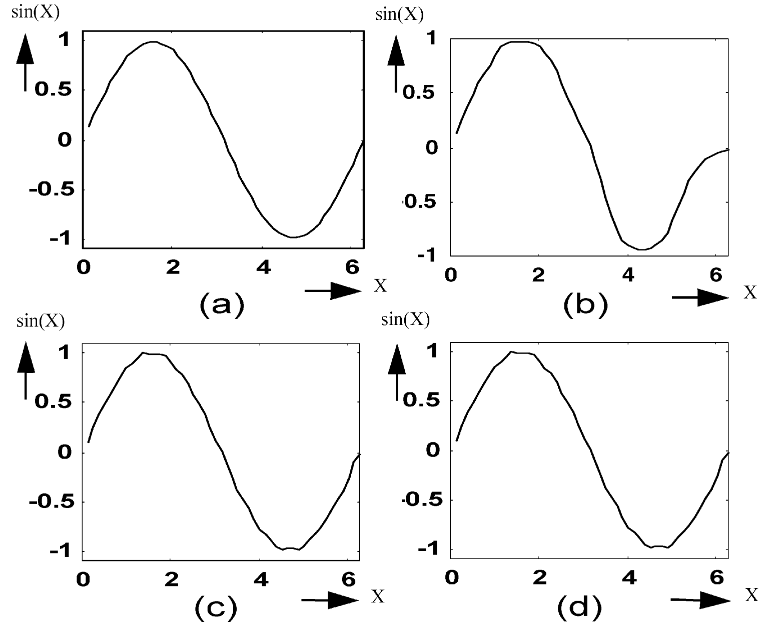


Fig. 6. Comparison of three different types of NNs in a typical function-approximation problem. (a) Sinusoid function used. (b) Approximation made by a ten-hidden-unit MLP. (c) Output of a quadratic network with ten hidden units. (d) Output of an SCBANN with $\alpha = 1.3$ and ten hidden units.

TABLE II
SIMULATION RESULTS FOR NETWORKS USED IN THE APPROXIMATION OF A SINUSOID FUNCTION

Network Type	MLPs	QNNs	SCBANNs
Number of neurons in hidden layer	10	10	10
Epochs	15000	15000	15000
Residual Error	0.0298	0.00016	0.00022

a constant-voltage scaling rule, as a result of which internal fields of the devices have increased. This has led in turn to velocity saturation of the carriers in the channel. Consequently, the "quadraticness" of the current–voltage relation has been lost; rather, a new power relation between them is observed, with a power reduced to between one and two. More recently, with further downscaling of the MOS feature size toward deep submicron values, to reduce the extreme levels of internal electric fields, design attention has shifted toward a constant-electric-field scaling rule. However, the very small dimensions of the resulting MOS architectures has kept the internal electric fields at relatively high values. Following this trend, more accurate yet simple models have been proposed in the literature [27] to account for the characteristics of velocity-saturated CMOS devices. Here, we adopt a simplified version of the nth-order rule model proposed in [27]. Basically, this model accounts for carrier-velocity saturation using an exponent value less than two.

$$V_{\mathrm{TH}} = V_{T0} + \gamma \cdot \left(\sqrt{2\phi_F - V_{\mathrm{BS}}} - \sqrt{2\phi_F}\right) \tag{6}$$

$$V_{\mathrm{DSAT}} = K \cdot (V_{\mathrm{GS}} - V_{\mathrm{TH}})^m \tag{7}$$

$$\begin{aligned} I_D = I_1 &= \left(\frac{W}{L_{\mathrm{EFF}}}\right) \cdot B \cdot (V_{\mathrm{GS}} - V_{\mathrm{TH}})^n \\ &\quad \cdot (1 + \lambda \cdot V_{\mathrm{DS}}),\ (\lambda = \lambda_0 - \lambda_1 V_{\mathrm{BS}}) \end{aligned} \tag{8}$$

$$\begin{aligned} &(\text{in the saturation region when}: V_{\mathrm{DS}} \geq V_{\mathrm{DSAT}}) \\ I_D &= I_1 \cdot \left(2 - \frac{V_{\mathrm{DS}}}{V_{\mathrm{DSAT}}}\right) \cdot \frac{V_{\mathrm{DS}}}{V_{\mathrm{DSAT}}} \\ &(\text{in the linear region when}: V_{\mathrm{DS}} < V_{\mathrm{DSAT}}) \end{aligned} \tag{9}$$

where V_{GS}, V_{DS}, and V_{BS} are gate-, drain-, and bulk-source voltages, respectively. W is the channel width and L_{EFF} is an effective channel length, while V_{TH} is the threshold voltage and V_{DSAT} is drain saturation voltage. Here, V_{T0}, γ, and $2\phi_F$ are parameters related to the calculation of the threshold voltage. Parameters B and n control the saturation region characteristics, while K and m determine the linear-region characteristics. Parameters λ_0 and λ_1 are related to finite drain conductance in saturation. The model is reduced to the Schockley model when we assume that $K = 1$, $m = 1$, $B = 0.5\beta$, and $n = 2$. In this paper, to avoid confusion with the number of inputs of a neuron, we use the notation α in place of n used in (8). As well, we use the above model to extract analytical relations between the output and inputs of the circuits investigated in this work. However, note that in these derivations, we neglect the effect of finite-output impedance in the model by making $\lambda = 0$.

B. Submicron Operation of CMOS Differential and Square-Law Circuits

For analog signal-processing applications, several CMOS square-law circuits have been proposed [7], [28], [29]. Here, we consider submicron operation of the circuit in [29], which is a versatile linear transconductor/square-law-function circuit for large-feature-size devices. Interestingly, when scaled, although using submicron power-law devices, it provides a power relation between its output current and its differential input voltage.

Quadratic-Function Circuit: In [29], it has been shown that with square-law devices, the simplified circuit shown in Fig. 7

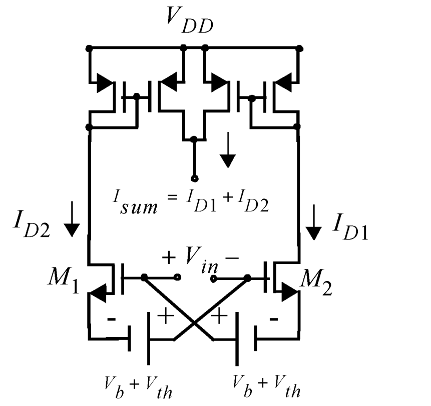


Fig. 7. Symbolic circuit showing the principles of operation of the MOS differential squarer.

produces a square-law function of its inputs. Here, we consider submicron operation of this circuit using α-order power-rule devices.

Here, we assume that M_1 and M_2, fabricated in a submicron CMOS technology, are matched and operating in their saturation region. Also, we assume that to a first-order approximation, the α-order power-rule MOS characteristic equation applies as

$$\begin{aligned} I_{D1} &= \kappa \cdot (V_{\text{GS1}} - V_{\text{TH}})^\alpha \\ I_{D2} &= \kappa \cdot (V_{\text{GS2}} - V_{\text{TH}})^\alpha \end{aligned} \tag{10}$$

where $\kappa = (W/L_{\text{EFF}}) \cdot B$ and W, L_{EFF}, B, and V_{TH} are parameters defined as noted above and are determined by an optimization and curve-fitting process. Then, as seen in Fig. 7, we have

$$I_{\text{sum}} = I_{D1} + I_{D2} = \kappa \cdot [(V_{GS1} - V_{\text{TH}})^\alpha + (V_{GS2} - V_{\text{TH}})^\alpha]. \tag{11}$$

Here, by using equivalent CMOS pairs to replace input transistors and also for the implementation of floating voltage sources as described in Fig. 8, we conclude that the equivalent V_{TH} of the floating voltage source becomes equal to the equivalent V_{TH} of the CMOS pairs replacing MOS transistors [refer to description of (16)–(19) below for proof]. If we write a loop equation for each of the input loops in Fig. 7, we conclude that

$$\begin{aligned} &-V_{\text{in}} + V_{\text{GS1}} - V_b - V_{\text{th}} = 0 \Rightarrow V_{\text{GS1}} = V_{\text{in}} + V_b + V_{\text{th}} \\ &-V_{\text{in}} + V_{\text{th}} + V_b - V_{\text{GS2}} = 0 \Rightarrow V_{\text{GS2}} = V_{\text{in}} + V_b + V_{\text{th}}. \end{aligned} \tag{12}$$

By substituting (12) into (11), we get

$$I_{\text{sum}} = \kappa \cdot ((V_b + V_{\text{in}})^\alpha + (V_b - V_{\text{in}})^\alpha). \tag{13}$$

Now, we use Taylor expansions of the functions $(y+x)^\alpha$ and $(y-x)^\alpha$ for $1 < \alpha < 2$, around point zero. Since the signs of x in the two functions are opposite, odd terms in the corresponding Taylor series will have opposite signs. As a result, only even terms will remain in the summation. As noted

$$\begin{aligned} I_{\text{sum}} = \kappa \cdot \Big(&2V_b^\alpha + 0 + \alpha(\alpha-1)V_b^{\alpha-2}V_{\text{in}}^2 + 0 \\ &+ \frac{2}{4!}\alpha(\alpha-1)(\alpha-2)(\alpha-3)V_b^{\alpha-4} \cdot V_{\text{in}}^4\Big). \end{aligned} \tag{14}$$

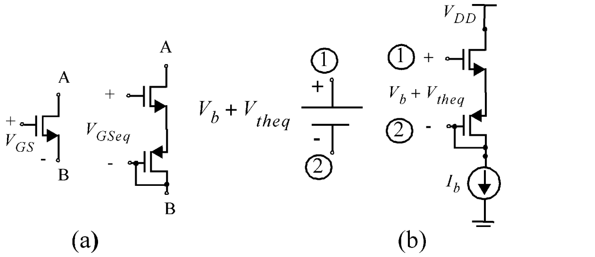


Fig. 8. (a) CMOS pair is substituted for each MOS transistor. (b) This equivalent CMOS pair, when biased with current source I_b, constructs a floating voltage source between terminals 1 and 2.

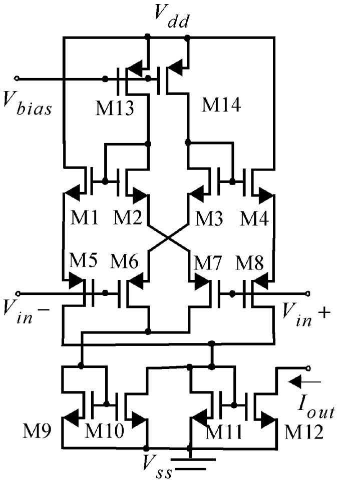


Fig. 9. Schematic diagram of the subtracting-squaring circuit used as the basic synaptic unit.

Considering the small value of the coefficient of the term V_{in}^4, we can neglect that term for small V_{in}. As a result, we see that with nth-order power-rule devices, this circuit continues to provide a square-law output for small V_{in}. Our simulations and measurements on fabricated chips agree with the conclusion that, for small V_{in}, the operation is quadratic and, for larger inputs, the operation becomes nonquadratic. However, the closedness property is preserved. A quantitative analysis is given in Section III-D. However, this analysis vanishes when $\alpha = 1$. In that case, a modified version of this circuit, as shown in Fig. 9, operates as a differential amplifier that is useful in implementation of absolute-value-norm version of SCBANNs.

Now, we consider the fact that when $V_{\text{in}} = 0$ in (13), then $I_{\text{sum}} = 2\kappa V_b^\alpha$, which is the bias current. Therefore, returning to (14), by adding some extra sampling and subtracting circuitry to Fig. 7, if $I_{\text{bias}} = 2\kappa \cdot V_b^\alpha$ is subtracted from I_{sum}, we will obtain a power-law I_{out}

$$I_{\text{out}} = I_{\text{sum}} - I_{\text{bias}} = 2\xi \cdot V_{\text{in}}^2, \text{where } \xi = \alpha(\alpha-1)\kappa V_b^{\alpha-2} \tag{15}$$

which is a current proportional to the square of the input-differential voltage. Note that this simplified circuit has some drawbacks including current flow in the control path and requires the implementation of floating equivalent voltage sources. However, these problems have been solved by replacing M_1 and M_2 with a "CMOS pair," as discussed in [29]. Also, a pair of NMOS and PMOS transistors biased with a current source can be utilized to implement each of the required floating voltage sources,

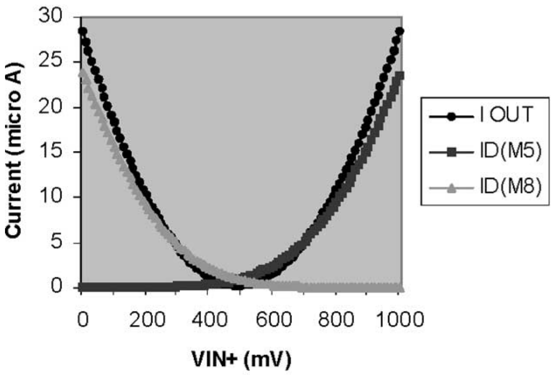


Fig. 10. HSPICE simulation results: output current of the circuit and currents of M5 and M8 transistors in Fig. 9. VIN- $= 500\ \mathrm{mV}$.

which are shown in Fig. 8. For each MOS transistor described by (10), we have

$$V_{\mathrm{GS}} = V_{\mathrm{TH}} + \frac{\sqrt[\alpha]{I}}{\sqrt[\alpha]{\kappa}}. \tag{16}$$

For the substituting CMOS pair, an equivalent gate-source voltage is defined as $V_{\mathrm{GSeq}} = V_{\mathrm{GSn}} + V_{\mathrm{GSp}}$, in which we use κ_n for an NMOS and κ_p for a PMOS transistor. These relations, combined with (16), imply that

$$V_{\mathrm{GSeq}} = V_{\mathrm{thn}} + V_{\mathrm{thp}} + \left(\frac{1}{\sqrt[\alpha]{\kappa_n}} + \frac{1}{\sqrt[\alpha]{\kappa_p}}\right) \cdot \sqrt[\alpha]{I}. \tag{17}$$

If we define $V_{\mathrm{theq}} = V_{\mathrm{thn}} + V_{\mathrm{thp}}$ and $1/\kappa_{\mathrm{eq}} = (1/\sqrt[\alpha]{\kappa_n} + 1/\sqrt[\alpha]{\kappa_p})^{\alpha}$, then V_{GSeq} is given by

$$V_{\mathrm{GSeq}} = V_{\mathrm{theq}} + \frac{\sqrt[\alpha]{I}}{\sqrt[\alpha]{\kappa_{\mathrm{eq}}}}. \tag{18}$$

Returning to Fig. 8(b) and using (16)–(18), we have

$$V_{12} = V_{\mathrm{GSn}} + V_{\mathrm{GSp}} = V_{\mathrm{theq}} + \frac{\sqrt[\alpha]{I}}{\sqrt[\alpha]{\kappa_{\mathrm{eq}}}}. \tag{19}$$

Then, we use V_{12} in place of the floating voltage $(V_b + V_{\mathrm{th}})$ source in Fig. 7. Also, with the previous definition for V_{theq}, V_b becomes equal to $\sqrt[\alpha]{I}/\sqrt[\alpha]{\kappa_{\mathrm{eq}}}$ and is controlled by the current source I_b, shown in Fig. 8(b). We apply these modifications to Fig. 7 and use the fact that I_{bias} in (15) can be obtained by adding the currents in the two floating-source branches $I_{b1} = I_{b2} = \kappa \cdot V_b^{\alpha}$ and $I_{\mathrm{bias}} = I_{b1} + I_{b2} = 2(\kappa \cdot V_b^{\alpha})$. Applying these modifications to Fig. 7 yields the final circuit used in our design, producing the I_{out} current, as shown in Fig. 9. Note that M_9, M_{10}, M_{11}, and M_{12} have been included into Fig. 9 (in comparison to Fig. 7) to sample I_{bias} and subtract it from I_{sum}, yielding the desired I_{out}.

C. Circuit Design

We have used a CMOS 0.35- $\mu\mathrm{m}$ technology to implement sample 1-D, 2-D, and 3-D submicron closed-boundary discriminating-surface artificial NNs. As predicted in our analytic evaluation of the operation of these neurons, their closed-boundary discriminating surfaces are very close to a quadratic shape. We expect that while the quadratic shape will fade away somewhere along the path to deeper submicron technologies, it is demonstrated to be possible to maintain the closedness of the discriminating borders.

Synaptic Units: A synaptic unit has been designed, as described in Section III-B and shown in Fig. 9. Fig. 10, which provides the results of an HSPICE simulation of this synaptic unit. The horizontal axis is the differential input and the vertical axis displays the synaptic output current. BSIM3.3 models have been used in the simulation.

In the 0.35- $\mu\mathrm{m}$ technology employed, the circuits operate with a 3.3-V supply voltage. If the inputs vary within the [0,1] volt interval, satisfactory operation of the circuit with low distortion is maintained. In Section III-D, based on experimental results and theoretical investigations, operation of such neurons is justified.

Neural Accumulation Unit: To facilitate accumulation, synaptic units provide their outputs in the form of a current. Therefore, the current summation of several such outputs is easily achieved by wiring them together. By using this wire adder and a subsequent mirroring operation, the total output current (or any proportion of it) can be transferred to other places. Note, as well, that at the neural level, we have a bias term that, for example in the case of a circle, corresponds to the square of the radius of a discriminating circle or, in general, is equivalent to the width factor of the discriminating boundary. Thus, we need to subtract the net neural contribution of the synapses from this bias term. This operation is achieved by constructing a current accumulating and subtracting unit [1].

Neural Sigmoid Unit: The net output of the neuron is provided in the form of a current. However, the inputs to the synaptic units are applied as voltages. Correspondingly, we need two operations, namely an I–V transformation and a saturating sigmoid transformation having a small linear operating region surrounded by a steadily saturating characteristic, for large positive and negative inputs. Note that we need the steadily saturating characteristics to ease the training process, yet to maintain a finite difference between outputs for two cases with a small change in input.

Fig. 11 shows the circuit adopted for these purposes [20]. In this circuit, *Mp*2 operates in its triode region as a resistor to form the linear part of the sigmoid for small positive or negative inputs. When the voltage developed goes above V_{THn} or below $-|V_{\mathrm{THp}}|$, then the NMOS or PMOS (*Mp*2) transistor, respectively, turns on. Because of their gate bias, these transistors operate in the saturation region and provide an α-power saturating characteristic. Fig. 11(b) shows results of HSPICE simulation of the operation of the sigmoid unit.

D. HSPICE Simulation Results

A basic assumption in deriving (13) has been that both transistors of the input pair work in their saturation regions. This mode of operation leads to cancellation of the odd terms in the Taylor series. Therefore, for small input voltages, one can neglect higher order even terms as well and a square-law relation between the differential input voltage and the output current is implied. If either of the transistors in the input pair leaves its saturation region or the input grows above a few hundred millivolts, the circuit will show nonquadratic behavior. With W/Ls

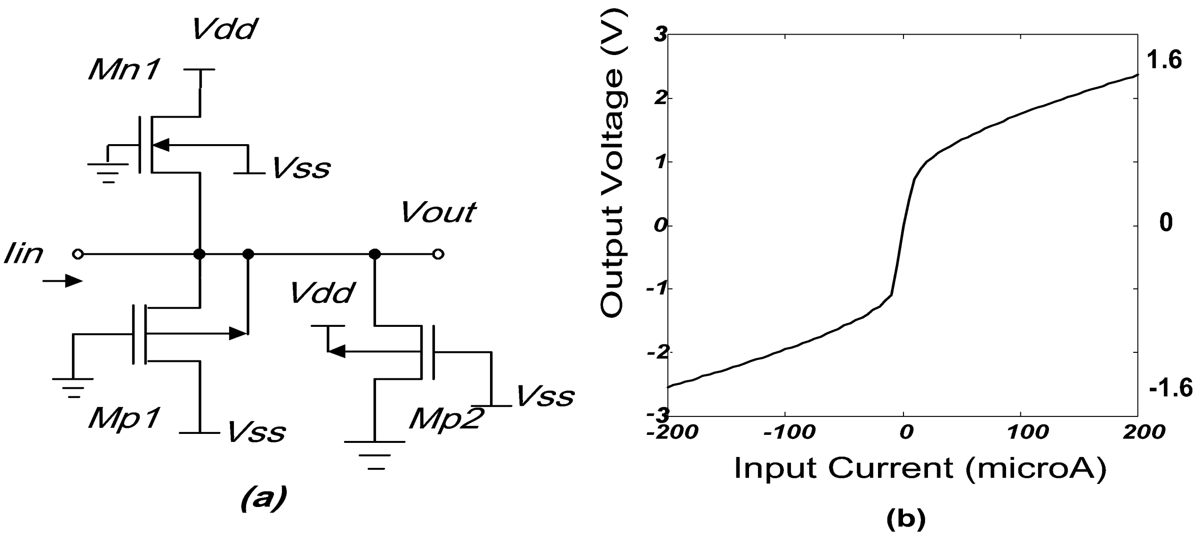
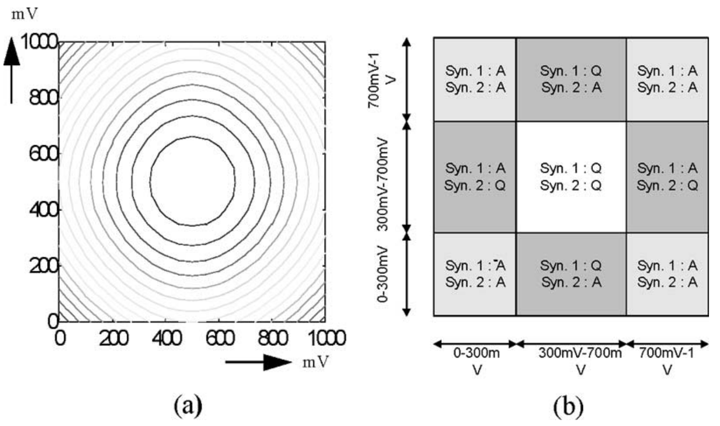


Fig. 11. Sigmoid unit. (a) Circuit diagram and (b) input–output characteristic.



	0-300m V	300mV-700m V	700mV-1 V
700mV-1 V	Syn. 1 : A Syn. 2 : A	Syn. 1 : Q Syn. 2 : A	Syn. 1 : A Syn. 2 : A
300mV-700mV	Syn. 1 : A Syn. 2 : Q	Syn. 1 : Q Syn. 2 : Q	Syn. 1 : A Syn. 2 : Q
0-300mV	Syn. 1 : A Syn. 2 : A	Syn. 1 : Q Syn. 2 : A	Syn. 1 : A Syn. 2 : A

Fig. 12. Results of HSPICE simulation of a two-input neuron. (a) Output-discriminating curves and (b) working regions of the devices.

for all transistors chosen to be equal to 5 μ/0.5 μ, Fig. 10 shows HSPICE simulation results for the circuit of Fig. 9. In this simulation, $V_{\mathrm{IN}-}$ is fixed at 0.5 V and $V_{\mathrm{IN}+}$ is swept from 0 to 1 V. Drain currents of transistors M_5 and M_8 show that, for an input voltage ranging between 300 and 700 mV, the output currents of both transistors are almost quadratic. However, as the input voltage leaves this voltage interval, the behavior of the circuit becomes nonquadratic.

Fig. 12(a) shows output-discriminating curves obtained from an HSPICE simulation for a two-input neuron, in which the circuit of Fig. 9 is used for synapses. A denotes "nonquadratic region" and Q denotes "quadratic region" of operation. As seen in Fig. 12(a), as long as input voltages of the synapses lie between 300 and 700 mV (that is, up to differential level of ±200 V), both synapses show a quadratic characteristic. Correspondingly, the respective parts of the discriminating curves are circular. However, when inputs rise and one or both synapses leave their quadratic regions, output-discriminating surfaces change into a nonquadratic form while maintaining continuity and closedness of the shaped boundaries. The results of simulations on shapes of discriminative curves shown here are consistent with the measurements based on the operation of real devices as, shown in Fig. 15(b). We note that, at the edges of the input space, boundary circles are cut and traverse the hard limits of the input space.

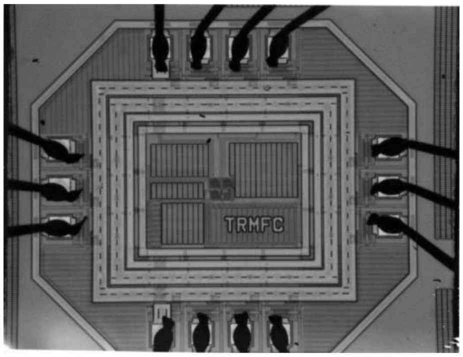


Fig. 13. Photomicrograph of the 3-D closed-boundary pattern-classifier chip, fabricated in a 0.35 $\mu\mathrm{m}$ CMOS technology.

IV. VLSI Implementation and Test Results

A test chip has been fabricated in a CMOS 0.35 $\mu\mathrm{m}$ technology provided to us through the auspices of the Canadian Microelectronics Corporation (CMC). The chip includes three synaptic units, one neural accumulating unit, and one sigmoid unit and can be used to provide 1-D, 2-D, or 3-D closed-boundary-discriminating-surface neurons. The area of each synaptic unit is 45×43 $\mu\mathrm{m}^2$ and the total area of each 3-D pattern-classifier neuron is 103×95 $\mu\mathrm{m}^2$. Fig. 13 shows a photomicrograph of the

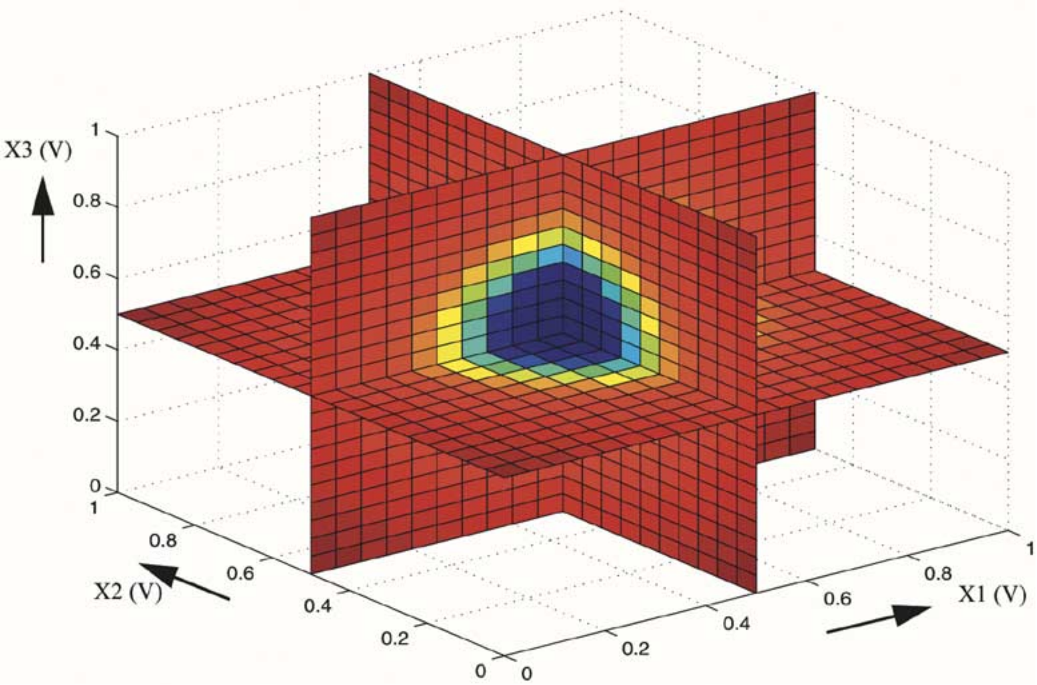
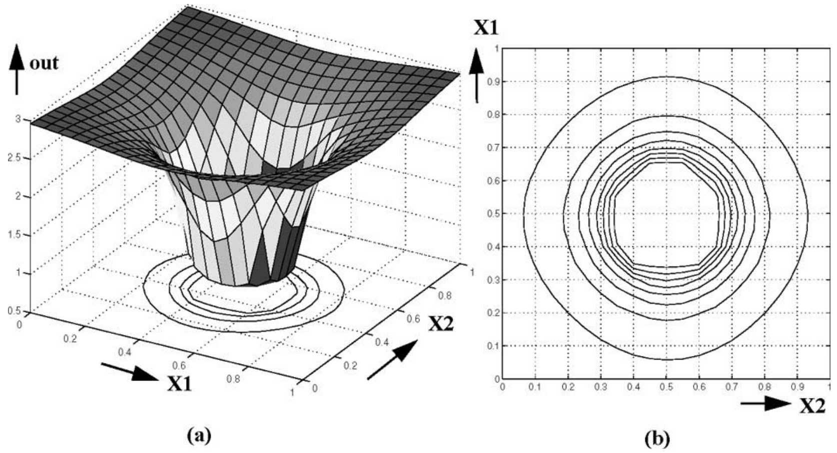


Fig. 14. Measurement results drawn using the 3-D Matlab plotting function. The entire input space has been scanned and 8000 measurements have been used to create this figure. X1, X2, and X3 are input voltages to the chip. The normalized outputs are shown as dots in the 3-D plot. Color code: black=0; dark blue=0.2; yellow=0.5; orange=0.6; light red=0.8; and darker red=0.9 and when approaching one. $R = 0.18$.



(a)

(b)

Fig. 15. (a) 3-D plot of the input–output relation of a closed-boundary 2-D pattern classifier built with the chip in Fig. 13. (b) Plot of equivalue curves showing the discriminating boundaries.

chip. This neuron can be used as the building block of a complex function-approximation or pattern-recognition system. The chip uses a 3.3 V supply voltage, with inputs expected to range from 0 to 1 V. Each synapse has a stored-weight analog input that represents one component of the reference vector to be stored. In a complete system, a dynamic capacitive-storage scheme might be used for weight saving. The voltage applied to the gate of the transistor providing R^{α} component of the neural stimulation is stored in the same way. The other input of each synaptic unit comes from the network inputs.

The density of the current implementation is 516 synapses per mm^2. This will amount to 6300 synapses per mm^2 in a 0.01-μm technology with approximate maximum power consumption of 31 *mW* per mm^2. This will make it possible to have a system with half a million synapses in a 0.01-μm technology with around 2 *W* of power consumption, which is expected to satisfy many real-world complex applications.

A. Measurement Results

The fabricated chips have been tested to verify their general operation. As well, the neuron performance as a 1-D, 2-D, or 3-D classifier has also been tested. We have used a collection of sweeping sources and measurement units to perform the tests. To observe the kind of discriminating surfaces formed by this neuron, several reference vectors are applied to the synaptic units and then the whole 1-D, 2-D, and 3-D input spaces are scanned. For example, in one case, the center of the 3-D discriminating shape was adjusted to (0.5,0.5,0.5) and the three inputs were scanned independently in increments of 0.05 V. This sample experiment provided 8000 test points whose respective output values were recorded. Fig. 14 demonstrates the 3-D closed-boundary pattern-separation borders of this neuron. Fig. 15 shows a 3-D plot of a 2-D pattern classifier. There, the point (0.5,0.5) has been chosen as the pattern center

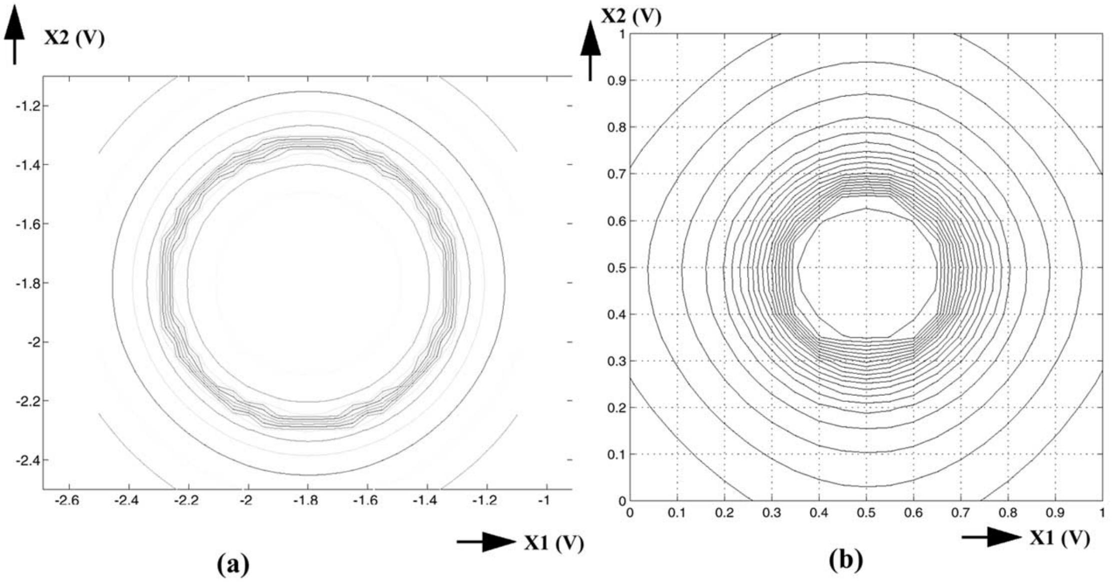


Fig. 16. Comparison of the measured boundaries produced with (a) quadratic neuron manufactured in a 1.2- $\mu\mathrm{m}$ CMOS process and (b) SCBANN neuron built in a 0.35- $\mu\mathrm{m}$ CMOS submicron process. Notice that border circles are cut and fit to the hard input limits.

and the radius is 0.18, as provided by applying a 730 mV bias to the radius-current-injection transistor. Note that equivalue output curves are drawn to show the shape of the discriminating borders. As for the two chosen input variables, one is swept from 0 to 1, while the other is held constant during each sweep. Then, the second input is incremented by 0.05 for the next sweep. This figure can also be interpreted as a collection of 1-D delta functions or 1-D function approximators. Correspondingly, their centers and radii can be tuned by proper adjustment of the current in corresponding transistors.

B. Comparison Between Two Generations of Technologies

Fig. 16 demonstrates and compares the discriminating surfaces obtained from our previous implementation fabricated in a 1.2- $\mu\mathrm{m}$ CMOS technology with $V_{\mathrm{DD}} = 5$ V and the current implementation fabricated in a 0.35- $\mu\mathrm{m}$ CMOS technology using $V_{\mathrm{DD}} = 3.3$ V. As observed, in a long-channel higher-supply-range technology, the quadratic behavior is maintained over a wider operating range of the devices.

V. Conclusion

The effectiveness of neurons with localized outputs has been demonstrated in previously published work. Their implementation in traditional CMOS VLSI technologies has also been shown to be possible. In this work, through diverse means (analytic reasoning, simulation, and measurement of a fabricated sample chip in a 0.35-μm submicron CMOS technology) the theory of operation and feasibility of the idea of using deep-submicron CMOS technology to implement closed-boundary discriminators and efficient function-approximation building blocks have been shown. An α-power model for submicron transistors has been employed and is shown to be satisfactory in predicting the performance of submicron implementation of these circuits. The next step to be taken is to integrate a larger number of these units in several layers by providing the digital and analog interfacing circuits necessary for the application of these networks in efficient high-speed system-level implementations.

Acknowledgment

The authors would like to thank Professors G. Gulak, J. Xu, and C. A. T. Salama of the University of Toronto, Toronto, ON, Canada, for the use of the equipment and facilities in their respective laboratories. Fabrication support of the Canadian Microelectronic Corporation is acknowledged and appreciated.

References

[1] S. M. Fakhraie and K. C. Smith, *VLSI-Compatible Implementations for Artificial Neural Networks*. Boston, MA: Kluwer, 1997.

[2] S. M. Fakhraie, J. M. Xu, and K. C. Smith, "Design of CMOS quadratic neural networks," in *Proc. IEEE Pacific Rim Conf. Communications, Computers, and Signal Processing*, Victoria, BC, Canada, May 1995, pp. 493–496.

[3] R. Lippmann, "An introduction to computing with neural nets," *IEEE Acoust., Speech, Signal Processing Magazine*, vol. 4, p. 4, Apr. 1987.

[4] D. S. Broomhead and D. Lowe, "Multivariable function interpolation and adaptive networks," *Complex Syst.*, vol. 2, pp. 321–355, 1988.

[5] J. Moody and C. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Comput.*, vol. 1, no. 2, pp. 281–294, 1989.

[6] J. Anderson, J. C. platt, and D. B. Kirk, "An analog VLSI chip for radial basis functions," in *Neural Information Processing Systems 5*, S. J. Hansen, J. D. Cowan, and C. Lee, Eds. San Mateo, CA: Morgan Kaufmann, 1993.

[7] S. Churcher, A. F. Murray, and H. M. Reekie, "Programmable analog VLSI for radial basis function networks," *Electron. Lett.*, vol. 29, no. 18, pp. 1603–1605, 1993.

[8] J. Diamond, C. Park, S. C. The, U. Santoni, K. Buckmann, and M. Holler, "Design and implementation of a recognition accelerator," in *Proc. CCVLSI'93*, Banff, AB, Canada, 1993, pp. 6A-8–13.

[9] R. Dogaru, A. T. Morgan, S. Ortmann, and M. Glesner, "A modified RBF neural network for efficient current-mode VLSI implementation," in *Proc. 5th Int. Conf. Microelectronics for Neural Networks*, 1996, pp. 265–270.

[10] A. Zaknich and Y. Attikiouzel, "Modified probabilistic neural network hardware implementation schemes," in *Proc. IEEE TENCON-Digital Signal Processing Applications*, vol. 1, 1996, pp. 167–172.

[11] T. Korner and U. Ruckert, "A VLSI friendly neural network with localized transfer functions," in *Proc. IEEE Int. Conf. Neural Networks*, vol. 1, 1995, pp. 169–174.

[12] D. J. Volper and S. E. Hampson, "Quadratic function nodes, use, structure, and training," *Neural Networks*, vol. 3, pp. 93–107, 1990.

[13] G. S. Lim, M. Alder, and P. Hadingham, "Adaptive quadratic neural nets," in *Pattern Recognition Letters*: North-Holland, 1992, vol. 13, pp. 325–329.

[14] K. F. Cheung and C. S. Leung, "Rotational quadratic function neural networks," in *Proc. Int. Joint Conf. Neural Networks*, Singapore, Nov. 18–21, 1991, pp. 869–874.

[15] S. M. Fakhraie and K. C. Smith, "Generalized artificial neural networks (GANN)," in *Proc. Canadian Conf. Electrical and Computer Engineering*, vol. 1, 1993, pp. 469–472.

[16] G. Dorffner, "Unified framework for MLP's and RBFNs: Introducing conic section networks," *Cybern. Syst.*, vol. 25, pp. 511–554, 1994.

[17] J. L. Holt and J.-N. Hwang, "Finite precision error analysis of neural network hardware implementations," *IEEE Trans. Comput.*, vol. 42, pp. 281–290, Mar. 1993.

[18] T. Szabo and G. Horvath, "Finite word length computational effects of the principal component analysis networks," *IEEE Trans. Instrum. Meas.*, vol. 47, pp. 1218–1222, Oct. 1998.

[19] J. B. Lont and W. Guggenbuhl, "Analog CMOS implementation of a multilayer perceptron with nonlinear synapses," *IEEE Trans. Neural Networks*, vol. 3, pp. 457–465, May 1992.

[20] S. Satyanarayana, Y. P. Tsividis, and H. P. Graf, "A reconfigurable VLSI neural network," *IEEE J. Solid-State Circuits*, vol. 27, pp. 67–81, Jan. 1992.

[21] M. Jabri and H. Flower, "Weight perturbation: An optimal architecture and learning technique for analog VLSI feedforward and recurrent multilayer networks," *IEEE Trans. Neural Networks*, vol. 3, pp. 154–157, Jan. 1992.

[22] S. M. Tam, B. Gupta, H. A. Castro, and M. Holler, "Learning on an analog VLSI neural network chip," in *Proc. IEEE Int. Conf. Systems, Man, Cybernetics*, 1990, pp. 701–703.

[23] S. Wolpert, M. J. Osborn, and M. T. Musavi, "A VLSI-based gaussian kernel mapper for real-time RBF neural networks," in *Proc. 18th IEEE Annu. Northeast Bioengineering Conf.*, 1992, pp. 51–52.

[24] G. Marshall and S. Collins, "A compact analogue radial basis function circuit," in *Proc. 4th Int. Conf. Artificial Neural Networks*, 1995, pp. 471–476.

[25] T. Yildirim and J. S. Marsland, "A conic section function network synapse and neuron implementation in VLSI hardware," in *Proc. IEEE Int. Conf. Neural Networks*, vol. 2, 1996, pp. 974–979.

[26] T. Poggio and F. Girosi, "Regularization algorithms for learning that are equivalent to multilayer networks," *Sci.*, vol. 247, pp. 978–982, 1990.

[27] T. Sakurai and A. R. Newton, "A simple MOSFET model for circuit analysis," *IEEE Trans. Electron. Devices*, vol. 38, pp. 887–894, Apr. 1991.

[28] K. Bult and H. Wallinga, "A class of analog CMOS circuits based on the square-law characteristics of an MOS transistor in saturation," *IEEE J. Solid-State Circuits*, vol. SC-22, pp. 357–365, June 1987.

[29] E. Seevinck and R. F. Wassenaar, "A versatile CMOS linear transconductor/square-law function circuit," *IEEE J. Solid-State Circuits*, vol. SC-22, pp. 366–377, June 1987.

[30] A. Lapedes and R. Farber, "How neural nets work," in *Neural Information Processing Systems*, D. Z. Anderson, Ed. New York: Amer. Inst. Phys., 1988, pp. 442–456.

Sied Mehdi Fakhraie (S'92–M'95) was born in Dezfoul, Iran, in 1960. He received the M.Sc. degree in electronics from the University of Tehran, Tehran, Iran, in 1989 and the Ph.D. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 1995.

Since 1995, he has been an Assistant Professor with the Department of Electrical and Computer Engineering, University of Tehran, where has been Founding Director of the VLSI Circuits and Systems Laboratory. From September 2000 to April 2003, he was with Valence Semiconductor Incorporated as Architect of the Valence gateway processor and also as the Technical Lead of the design team and, subsequently, was Director of ASIC/SoC design in the Markham, Canada, office of Valence. During the summers of 1998, 1999, and 2000, he was a Visiting Professor at the Department of Electrical and Computer Engineering, University of Toronto, where he continued his work on efficient implementation of artificial neural networks. He is coauthor of the book *VLSI-Compatible Implementation of Artificial Neural Networks* (Boston, MA: Kluwer, 1997). He has also published more than 60 reviewed conference and journal papers. He has worked on many industrial IC design projects including design of network processors and home gateway access devices, DSL modems, pagers and one- and two-way wireless messaging systems, and digital signal processors for personal and mobile communication devices. His research interests include system design and ASIC implementation of integrated systems, novel techniques for high-speed digital circuit design, and system-integration and efficient VLSI implementation of artificial neural networks.

Hamed Farshbaf received the B.S. degree in electronics from Sharif University of Technology, Tehran, Iran, in 1999, and the M.S. degree in electrical and computer engineering from the University of Tehran, Tehran, Iran, in 2002.

He is an Instructor with the Department of Electrical and Computer Engineering, University of Tehran and also was responsible for laboratory sessions of an advanced VLSI design course. His research interests include modeling, simulation, and VLSI implementation of artificial neural networks.

Kenneth Carless Smith (S'53–A'54–M'60–SM'76–F'78–LF'96) received the Ph.D. degree in solid-state physics in 1960 from the University of Toronto, Toronto, ON, Canada, where he is presently a Professor Emeritus of electrical and computer engineering, mechanical and industrial engineering, computer science, and information studies. As well, he has extensive industrial experience in the design and application of computers, computer systems, medical instrumentation, and electronic circuits generally, as employee, manager, administrator, entrepreneur, consultant, and educator. His research interests currently include analog VLSI, multiple-valued logic, sensor systems, instrumentation, human-factors engineering, and reliability. He is widely published in these and other areas, with well over 200 journal and proceedings papers, books, and book contributions. As well, he was appointed as an Advisory Professor, Shanghai Tiedao University in 1989. From 1993 to 1998, he also held a position as a Visiting Professor in the Department of Electrical and Electronic Engineering, University of Science and Technology, Hong Kong, where he was the Founding Director of Computer Engineering.