# A New Design Method for the Complex-Valued Multistate Hopfield Associative Memory

Mehmet Kerem Müezzinoğlu, *Student Member, IEEE*, Cüneyt Güzeliş, and Jacek M. Zurada, *Fellow, IEEE*

***Abstract*—A method to store each element of an integral memory set $M \subset \{1,2,\ldots,K\}^n$ as a fixed point into a complex-valued multistate Hopfield network is introduced. The method employs a set of inequalities to render each memory pattern as a strict local minimum of a quadratic energy landscape. Based on the solution of this system, it gives a recurrent network of $n$ multistate neurons with complex and symmetric synaptic weights, which operates on the finite state space $\{1,2,\ldots,K\}^n$ to minimize this quadratic functional. Maximum number of integral vectors that can be embedded into the energy landscape of the network by this method is investigated by computer experiments. This paper also enlightens the performance of the proposed method in reconstructing noisy gray-scale images.**

***Index Terms*—Complex-valued Hopfield network, gray-scale image retrieval, linear inequalities, multistate associative memory.**

## I. Introduction

DESIGN of recurrent neural networks as dynamical associative memories has been one of the major research areas in the neural networks literature for two decades, since the pioneering work by Hopfield [1]. This work demonstrated that a single-layer fully connected network is capable of restoring a previously learned static pattern called a memory vector, ensuring its convergence from an initial condition representing the corrupted or incomplete information toward a fixed point. A Hebbian learning procedure was introduced in [1] for the proposed network to be applied for memory sets consisting of bipolar binary memory vectors. Despite its simplicity and biological significance, this learning rule indeed could not ensure asymptotically stable equilibria located at the binary memory. Moreover, it could not avoid some undesired equilibria in the resulting network. The basins of attraction associated with fixed points in the resulting network were also unpredictable and the method did not allow the designer to adjust their sizes. Consequently, the resulting network was indeed far from resembling the nearest-neighbor classifier, i.e., the ideal associative memory, that successfully associates a distorted pattern to the nearest memory vector. Many relatively successful alternative methods [2], [3] then followed to overcome these deficiencies that arise in the design of the Hopfield network as a binary associative memory. However, very few papers have appeared in the literature that generalize the original idea to the nonbinary case, i.e., for cases where the memory vectors are allowed to take integral values other than $-1$ and 1.

To be able to recall $n$-dimensional integral memory vectors in $\{1,2,\ldots,K\}$, the conventional Hopfield model obviously needs to be generalized such that the state space of the network contains $I := \{1,2,\ldots,K\}^n$. A straightforward way to achieve this is through generalizing the conventional bi-state activation function to a $K$-stage quantizer as proposed and analyzed in [4]. By replacing the activation functions of neurons in the conventional Hopfield network with this nonlinearity remarkable steps have been made toward the design of multistate associative memories [5]–[7]. It has also been shown in [8] that the maximum number of integral patterns that can be stored in such a network by any design procedure is proportional to $n \cdot (K-1) \cdot f(K)$, where $f(K)$ is of order one.

An alternative dynamical finite-state system operating on $I$ has been introduced in [9] as the complex-valued multistate Hopfield network. This model employs the complex neuron model [11] employing the complex-signum nonlinearity [10]. Each neuron in this autonomous, single-layer, connectionist network simply takes a complex weighted sum of previous state values and passes it through the complex-signum activation function. This produces its next state, where the complex-signum is a $K$-stage phase quantizer for complex numbers and is defined as

$$\text{csign}_K(u) := \begin{cases} e^0 & 0 \le \arg(u) < \frac{2\pi}{K} \\ e^{i2\pi/K} & \frac{2\pi}{K} \le \arg(u) < \frac{4\pi}{K} \\ \vdots & \\ e^{i2\pi/K(K-1)} & (K-1)\frac{2\pi}{K} \le \arg(u) < 2\pi \end{cases}. \tag{1}$$

Note that, by the virtue of this nonlinearity, each state of the network is allowed to take one of the equally spaced $K$ points on the unit circle of the complex plane (see Fig. 1). Each neuron indicates an integral information modulated as the phase angle of its unit-magnitude complex-valued state, which constitutes an element of the state vector of the dynamical network. Hence, not the original integral vectors, but their transformed versions can be stored and recalled by this network. This injective transformation, which basically maps each entry of a vector in the

Manuscript received December 4, 2002; revised March 25, 2003. The work of M. K. Muezzinoglu was supported in part by The Scientific and Technical Research Council of Turkey, Münir Birsel Foundation, 06100 Ankara, Turkey. The work of J. M. Zurada was supported in part by the Systems Research Institute, Polish Academy of Sciences, 01-447 Warsaw, Poland.

M. K. Müezzinoğluis with the Computational Intelligence Laboratory, Electrical Engineering Department, University of Louisville, Louisville KY 40292 USA and also with the Electrical and Electronics Engineering Department, Dokuz Eylül University, 35160 Buca,İzmir, Turkey.

C. Güzeliş is with the Electrical and Electronics Engineering Department, Dokuz Eylül University, 35160 Buca, İzmir, Turkey.

J. M. Zurada is with the Computational Intelligence Laboratory, Electrical Engineering Department, University of Louisville, Louisville KY 40292 USA.

Digital Object Identifier 10.1109/TNN.2003.813844

1045-9227/03$17.00 © 2003 IEEE

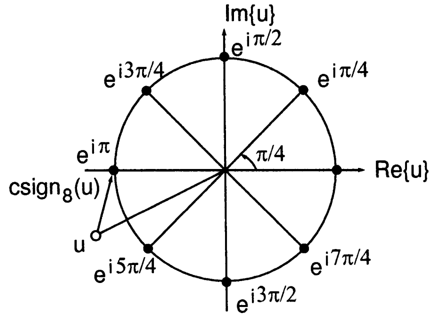



Fig. 1. Illustration of $\text{csign}_8(u)$ for $u = -1.2 - 0.5i$.

integral lattice $I$ as a point on the unit circle of the complex plane, is expressed as

$$\mathbf{p}_K(\cdot) : \{1, 2, \ldots, K\}^n \to \left\{e^{i2\pi/Kj} : j \in \{0, \ldots, K-1\}\right\}^n$$
$$\mathbf{p}_K(\mathbf{u}) := \left[e^{i2\pi/Ku_1} e^{i2\pi/Ku_2} \cdots e^{i2\pi/Ku_n}\right]^T. \tag{2}$$

The range of $\mathbf{p}_K(\cdot)$, which will be called the transformed vectors in the rest of the paper, can also be considered as the codomain of the transformation. In this case, the usage of complex-valued multistate Hopfield network, which actually operates on the transformed vector space, is meaningful in processing integral vectors. Each state of the network can be uniquely transformed to an integral vector in $I$ via $\mathbf{p}_K^{-1}(\cdot)$.

A generalized Hebb rule has been proposed in [9] as a learning procedure for complex-valued multistate Hopfield network to recall some specific phase combinations from their distorted versions. However, as expected, this generalized rule, which constitutes the unique learning procedure for the considered network model, suffers from almost the same limitations as it does in the binary case. This is why an efficient application of this network could not have been proposed yet. On the other hand, another significant qualitative result addressed in [9] is that the state vector of the network necessarily converges to a local minimum of a specific real-valued quadratic functional. This is defined in terms of the network parameters, along the collective operation of the $n$ complex-valued neurons in asynchronous mode, if the complex weight matrix of the network is Hermitian and its diagonal entries are all nonnegative. Such a network will be called *Hermitian* hereafter.

Several design procedures that employ inequalities in the design of recurrent neural networks have been reported, e.g., [12]–[14]. Such attempts mainly focused on embedding fixed points into the conventional Hopfield network and constructed the design inequalities directly from the nonlinear recursion performed by the network. Though a solution of these inequalities gives the desired parameters of the recursion which has fixed points located at the given binary points, networks designed in these ways might not be capable of restoring a memory vector from its distorted versions, since attractiveness is not a design condition in such methods. By posing this property as a constraint, an indirect method to construct the energy landscape of the discrete Hopfield network via solution of homogenous linear inequalities was proposed in [15]. Nevertheless, these effective approaches on designing conventional bi-state network have not yet been extended for multistate associative memories.

Based on the energy minimization performed by the complex-valued multistate Hopfield network, this paper suggests an indirect design procedure. The procedure gives a Hermitian weight matrix such that each transformed memory vector is an attractive fixed point of the resulting finite state system. The proposed method basically employs homogenous linear inequalities to dig a basin for each transformed memory vector in the quadratic energy landscape to ensure that they are all strict local minima. If the system of inequalities is feasible, then its solution provides the desired quadratic form, and finally the complex weights of the network are determined from the Hermitian coefficient matrix of this quadratic.

Feasibility of the inequality system constructed in the design is actually not only sufficient but also necessary for the existence of a Hermitian network that possesses attractive fixed points located exactly at the transformed memory vectors. In other words, if the constructed inequality system is infeasible, no Hermitian network can possess a limit set that contain the transformed memory vectors. This implies that the proposed method reveals the best performance of such a network as a multistate associative memory.

The memory capacity provided by the method has been estimated by intensive computer experiments. The results are presented inSection III-A, which show that the method can be successfully applied for almost every memory set with cardinality less than or equal to the dimension of its elements. As an application of the proposed method, the recall capability of the resulting network has also been demonstrated on gray-scale images. The results presented in Section III-B illustrates the performance of the network in reconstructing some test images from their distorted versions that are corrupted by various amounts of salt-and-pepper noise.

## II. Design Procedure

### A. Complex-Valued Multistate Hopfield Network

Assume a complex-valued multistate Hopfield network consists of $n$ fully connected neurons, whose states at time instant $k$ constitute the state vector $\mathbf{x}[k]$ of the network. Let $w_{ij}$ denote the complex-valued weight associated to the coupling from the state of the $j$th neuron to an input of the $i$th one. The asynchronous operation of the network is characterized as updating the state of a single neuron, say $l$th neuron, at time $k$ according to the recurrence

$$x_l[k+1] = \text{csign}_K\left(e^{i(\pi/K)} \sum_j w_{lj} x_j[k]\right) \tag{3}$$

while keeping all other states unchanged. Here $K$ is the resolution factor of the network, and it determines the cardinality of the finite state-space. Although the term $e^{i(\pi/K)}$ has no effect on the network dynamics theoretically, it provides a phase margin of $\pi/K$ for phase noise of the weighted sum of state vector entries.

The qualitative properties of the proposed network can be investigated by introducing an energy function defined on the state-space in terms of the weight coefficients:

$$E(\mathbf{x}) := -\frac{1}{2}\sum_i\sum_j w_{ij}\bar{x}_i x_j \tag{4}$$

similar to the way followed in the stability analysis of conventional Hopfield network [16]. A sufficient condition on the convergence of the recursion (3) has been reported in [9] as a Hermitian weight matrix ($\mathbf{W} = \mathbf{W}^*$) with nonnegative diagonal entries ($w_{ii} \geq 0$). The proof of this statement is simply achieved by showing that each state transition necessarily causes a decrement in the energy function under these conditions, which also enable us to rewrite (4) in a real-valued quadratic form

$$E(\mathbf{x}) = -\frac{1}{2}\mathbf{x}^*\mathbf{W}\mathbf{x}. \tag{5}$$

Since the network operates in a finite state-space by definition of $\text{csign}_K(\cdot)$, then the domain of (5) is finite. The state transitions therefore ends at a local minimum of (5) in finite time steps for any initial condition. In fact, the domain of the energy function (5) and the state space of the asynchronous recursion (3) are the same spaces. Hence, the energy function, which is quadratic in the states but linear in the weight coefficients, not only establishes the convergence analysis, but also defines attractive fixed points of the network as its strict local minima.

We assume throughout the paper that the update order of the neurons, i.e., the index $l$ in (3), is chosen at random, like usually it is done in the conventional discrete Hopfield network. One can easily verify that the update order is a parameter of the network, in other words the basins of attraction of fixed points may vary with the update order. Therefore, we make use of the term *attractiveness*, instead of conventional stability, to impose nonempty basin of attraction for the fixed point of interest.

*Definition 1:* A fixed point $\mathbf{x}^*$ of recursion (3) is attractive if there exists an open ball $\mathcal{A}(\mathbf{x}^*)$ centered at $\mathbf{x}^*$ such that: for every $\mathbf{y} \in \mathcal{A}(\mathbf{x}^*)$ there exists an update order. When this update order is applied, the state vector $\mathbf{x}$ of recursion (3) converges to $\mathbf{x}^*$ for the initial condition $\mathbf{x}[0] = \mathbf{y}$.

### *B. Design of Quadratic Energy Function Possessing Local Minima at Desired Points*

We restrict ourselves to the synthesis of the complex-valued multistate Hopfield network with Hermitian weight matrix with zero diagonal entries. Note that this assumption not only reduces the amount of parameters that describe the network, but also simplifies the design as it already guarantees convergence. Indeed, the design of the network is equivalent to the design of its energy function in this case, since the parameters (i.e., the Hermitian weight matrix) of the network can be uniquely determined from the coefficients of its energy function and vice versa. Thus, rather than the recursion (3) directly, our design method described in the following mainly focuses on the energy function (5), which is necessarily real-valued by the previous assumption.

Given a set of integral memory vectors $M \subset \{1, 2, \ldots, K\}^n$, let $M_c$ denote the set of complex vectors obtained by transforming elements of $M$ into their complex representation by (2). In order to perform a search for a Hermitian coefficient matrix $\mathbf{W}$, such that the real-valued discrete quadratic form (5) attains a local minimum at each element of $M_c$, we simply apply the definition of a strict local minimum, and impose a set of strict inequalities

$$E(\mathbf{x}) < E(\mathbf{y}), \forall \mathbf{y} \in \mathcal{B}_1^K(\mathbf{x}) - \{\mathbf{x}\} \tag{6}$$

to be satisfied for each $\mathbf{x} \in M_c$. Here $\mathcal{B}_1^K(\mathbf{u})$ is the 1-neighborhood of $\mathbf{u}$ and defined formally as

$$\mathcal{B}_1^K(\mathbf{u}) := \bigcup_{i=1}^{n}\left\{\mathbf{v} : v_i = u_i e^{i2\pi/K} \vee v_i = u_i e^{-i2\pi/K}, v_j = u_j, j \neq i\right\} \cup \{\mathbf{u}\}. \tag{7}$$

By substituting (4) in (6), we express this condition as $2n$ inequalities to be satisfied by the coefficient matrix $\mathbf{W} = [w_{ij}]$

$$\sum_i\sum_j w_{ij}\bar{x}_i x_j > \sum_i\sum_j w_{ij}\bar{y}_i y_j,\ \forall \mathbf{y} \in \mathcal{B}_1^K(\mathbf{x}) - \{\mathbf{x}\}. \tag{8}$$

Incorporating now our initial design considerations $w_{ij} = \bar{w}_{ji}$ and $w_{ii} = 0$, condition (8) can be further expressed in terms of only the upper triangle entries of $\mathbf{W}$

$$\sum_{1\leq i<j\leq n} w_{ij}\left[\bar{x}_i x_j - \bar{y}_i y_j\right] + \bar{w}_{ij}\left[x_i\bar{x}_j - y_i\bar{y}_j\right] > 0 \tag{9}$$

for all $\mathbf{y} \in \mathcal{B}_1^K(\mathbf{x}) - \{\mathbf{x}\}$. We then substitute the identity

$$w_{ij}\bar{x}_i x_j + \bar{w}_{ij}x_i\bar{x}_j = 2\text{Re}\{w_{ij}\}\text{Re}\{\bar{x}_i x_j\} - 2\text{Im}\{w_{ij}\}\text{Im}\{\bar{x}_i x_j\} \tag{10}$$

in (9) and obtain

$$\sum_{1\leq i<j\leq n} \text{Re}\left\{w_{ij}\right\}\left[\text{Re}\left\{\bar{x}_i x_j\right\} - \text{Re}\left\{\bar{y}_i y_j\right\}\right] + \text{Im}\left\{w_{ij}\right\}\left[\text{Im}\left\{\bar{y}_i y_j\right\} - \text{Im}\{\bar{x}_i x_j\}\right] > 0. \tag{11}$$

for all $\mathbf{y} \in \mathcal{B}_1^K(\mathbf{x}) - \{\mathbf{x}\}$. Recall from the definition of transformation $\mathbf{p}(\cdot)$ in (2) that $\text{Re}\{\bar{x}_i x_j\} = \cos(2\pi/K(-\hat{x}_i + \hat{x}_j))$ and $\text{Im}\{\bar{x}_i x_j\} = \sin(2\pi/K(-\hat{x}_i + \hat{x}_j))$ where $\hat{\mathbf{x}}$ is the original integral vector from which the unit-magnitude complex vector $\mathbf{x}$ is obtained. Hence, the design condition (11) could be directly expressed in terms of the original memory vectors, i.e., the elements of $M$, instead of the transformed ones in $M_c$.

We finally gather all inequalities associated to all memory vectors, and formally impose the overall system of inequalities, which have been derived above, as the design condition as follows.

*Corollary 1:* The quadratic form (5) possesses a strict local minimum at each element of $M_c$ if and only if the homogenous inequality

$$\sum_{1\le i<j\le n}\left\{\operatorname{Re}\{w_{ij}\}\left[\cos\left(\frac{2\pi}{K}(\hat{x}_j-\hat{x}_i)\right)-\cos\left(\frac{2\pi}{K}(\hat{y}_j-\hat{y}_i)\right)\right]+\operatorname{Im}\{w_{ij}\}\left[\sin\left(\frac{2\pi}{K}(\hat{y}_j-\hat{y}_i)\right)-\sin\left(\frac{2\pi}{K}(\hat{x}_j-\hat{x}_i)\right)\right]\right\}>0 \quad (12)$$

is satisfied by the Hermitian weight matrix $\mathbf{W}$ for all $\hat{\mathbf{x}}\in M$ and for all $\hat{\mathbf{y}}\in\mathcal{I}_1^K(\hat{\mathbf{x}})-\{\hat{\mathbf{x}}\}$. Here $\mathcal{I}_1^K(\hat{\mathbf{x}})$ is the ball that contains the inverse-transformed versions of the vectors in $\mathbf{B}_1^K(\mathbf{x})$, namely $\hat{\mathbf{x}}$ and all of its "1" neighbors in the integral lattice $\{1,2,\dots,K\}^n$

$$\mathcal{I}_1^K(\mathbf{u}):=\bigcup_{i=1}^{n}\left\{\mathbf{v}: v_i=u_i+1(\operatorname{mod}K)\vee v_i=u_i-1(\operatorname{mod}K)\ v_j=u_j, j\ne i\right\}\cup\{\mathbf{u}\}.$$

To find real and imaginary parts of desired weight coefficients, a solution to this system of $2|M|n$ inequalities is needed to be calculated by an appropriate method. Note that (12) is a linear feasibility problem, because left-hand side of each inequality is linear in the variables $\operatorname{Re}\{w_{ij}\}$ and $\operatorname{Im}\{w_{ij}\}$ for $i,j=1,2,\dots,n$. Due to this property, if (12) is a feasible inequality system for a given $M$, any linear programming procedure, e.g., the primal-dual method [17], or the perceptron learning algorithm [18], would provide a solution, so the complex parameters of the network could be determined by reconstructing $\mathbf{W}$ from this solution. On the other hand, infeasibility of (12) means that the given memory vectors cannot be altogether embedded as strict local minima into (5), and consequently that there exists no Hermitian network which has attractive fixed points located at each of these vectors.

### *C. Elimination of Trivial Spurious Memories*

The goal of the design method described above is only to render each memory vector as an attractive fixed point of the network. Since no additional condition has been imposed on eliminating undesired fixed points that might occur in the resulting network, the Hermitian weight matrix $\mathbf{W}$ obtained by solving (12) by any suitable procedure could also satisfy a set of inequalities, which implies a vector other than the elements of $M_c$ be a strict local minimum of (5), although these inequalities are not explicitly imposed in the design.

Most of the associative memory design methods are known to cause spurious memories. Unfortunately, neither the existence nor the location of many of these points in the state-space of the dynamical network is predictable. Moreover, discrimination of these vectors after the design is very difficult for large $n$ since almost every point in the huge state space of the system should be checked for this purpose. On the other hand, some of the spurious memories are correlated with the memory vectors and their locations can be exactly determined in terms of the memory vectors. For example, the conventional Hebb rule used in the design of binary associative memory introduces many undesired fixed points to the network beyond the desired ones, and most of these points cannot be determined without checking each point in the entire state-space [19]. However, one can easily conclude that if $\mathbf{x}$ is a fixed point of the discrete Hopfield network, then so is $-\mathbf{x}$. This property of network designed by the Hebb rule enables the designer to address some spurious memories in advance, which are directly related to the original memory vectors.

A similar relation can be extracted from our design method by observing from (12) that only the differences between the entries of the integral memory vectors, not their actual values, are used in the construction of the design inequalities. It can be easily verified that the inequality system constructed for an integral memory vector $\hat{\mathbf{x}}\in\{1,2,\dots,K\}^n$ in the way proposed in the previous subsection would be exactly the same one constructed for each vector $\hat{\mathbf{x}}+k\cdot\mathbf{e}\ (\operatorname{mod}K)$, where $k=1,2,\dots,K$ and $\mathbf{e}$ is the $n$-vector with all "1" entries. Hence, the weight matrix calculated from the solution of (12) not only makes each element of $M_c$ an attractive fixed point, but also introduces at least $(K-1)|M|$ additional vectors, namely the transformed versions of the integral vectors obtained by incrementing each element of $M$ in modulo $K$ by $k\cdot\mathbf{e}$, $k=1,\dots,K-1$, as spurious memories to the network. Such vectors are called trivial spurious memories and an extension to the design is proposed in the following to eliminate them.

Let us append an arbitrary integer, say 1, to each memory vector in $M$ as last entry and apply the proposed procedure to obtain the complex-valued multistate associative memory of $n+1$ neurons. Since the last entry of any trivial spurious memory is different than 1 by definition, one can simply exclude their transformed versions from the state-space of the network by restricting the dynamics (3) in the subspace that consists of the vectors whose last entries are equal to $e^{i2\pi/K}$. This is achieved by setting the state of the $(n+1)$st neuron fixed to $e^{i2\pi/K}$ along the recursion. Note that this state is connected to the inputs of other neurons via the weights $\{w_{l,n+1}\}_{l=1}^{n}$, thus this modification on the network model is actually equivalent to introducing a complex threshold $t_l=e^{i2\pi/K}w_{l,n+1}$ to $l$th neuron of the original network (3) for $l=1,\dots,n$, whose dynamical behavior can now be recast as

$$x_l[k+1]=\operatorname{csign}_K\left(e^{i(\pi/K)}\left[\sum_{j=1}^{n}w_{lj}x_j[k]+t_l\right]\right). \quad (13)$$

Although the method avoids the trivial spurious memories, there might still occur some nontrivial spurious memories in the network. It is expected that the number of such attractive fixed points increase with $K$, since the cardinality of the state-space increases with $K$. However, it is ensured by the method that none of these spurious memories is located in $\mathcal{I}_1^K(\hat{\mathbf{x}})$ for all $\hat{\mathbf{x}}\in M$, therefore the resulting network corrects all possible errors caused by incrementing or decrementing a single entry of the memory vectors by one. In other words, correction of the vectors in 1-neighborhood of the memory vectors are guaranteed.

### D. Algorithmic Summary of the Method

A summary of the proposed design method described in Section II-B together with its improvement in Section II-C is given below.

Algorithm 1: Input to the algorithm is $M \subset \{1, 2, \ldots, L\}^n$

Step 0: Set a resolution factor $K \geq L$ for the network. Append 1 to every $\hat{\mathbf{x}} \in M$ as the last entry. Set $\mathbf{A}$ as the empty matrix.

Step 1: For each $\hat{\mathbf{x}} \in M$ and for each $\hat{\mathbf{y}} \in \mathcal{I}_1^K(\hat{\mathbf{x}}) - \{\hat{\mathbf{x}}\}$, calculate the row vector as shown in the first formula at the bottom of the page, where $c_{ij} = \cos(2\pi/K(\hat{x}_j - \hat{x}_i)) - \cos(2\pi/K(\hat{y}_j - \hat{y}_i))$ and $s_{ij} = \sin(2\pi/K(\hat{y}_j - \hat{y}_i)) - \sin(2\pi/K(\hat{x}_j - \hat{x}_i))$, and append it as an additional row to matrix $\mathbf{A}$.

Step 2: Find a solution $\mathbf{q}^* \in \Re^{n(n+1)}$ for the inequality system $\mathbf{A}\mathbf{q} > \mathbf{0}$ by using any appropriate method.

Step 3: Construct the Hermitian matrix as shown in the second formula at the bottom of the page.

Step 4: Extract the parameters of recursion **(13)** from $\hat{\mathbf{W}}$ as $w_{ij} = \hat{w}_{ij}$ for $i, j = 1, 2, \ldots, n$ and $t_j = e^{i2\pi/K}\hat{w}_{i,n+1}$ for $j = 1, 2, \ldots, n$.

As the dimension $n$ of the memory vectors increases, manipulating the energy of each memory vector in the way suggested by the second and third steps of this algorithm becomes time and memory consuming when compared to the generalized Hebb rule. In practice, this procedure is easily realizable for memory sets with resolution factor of order ten and dimension of order ten, sufficient to perform reconstruction of gray-scale images. On the other hand, the performance of the resulting network is much better than that of the one designed by the generalized Hebb rule, as shown at the end of the next section.

## III. Simulation Results

Results of computer experiments are presented below to illustrate the quantitative performance of the method, i.e., the maximum cardinality of an arbitrary memory set that can be successfully embedded into the network by the proposed design method. The recall capability of the resulting network and its application on reconstructing gray-scale images are also demonstrated.

### A. Complete Storage Performance

Any fixed point of an $n$th-order dynamical system can be considered as an $n$-dimensional static information encoded as system parameters. As demonstrated in the previous section, dynamical associative memories are designed from this point of view by determining the parameters of an *a priori* chosen network model such that a given set of static vectors are the fixed points of this system. Hence, an associative memory realizes a dichotomy defined on its state space: some specific points in this space are fixed points (constitute the limit set) of the system, while the rest are not. However, the design of an ideal associative memory in this way is generally not possible for every possible memory set, i.e., not every dichotomy can be implemented, because of limitations of the chosen model, e.g., the number of parameters. In our case, for example, the network model involves $(n^2+n)/2$ complex coefficients (weights and thresholds), however, the number of all possible dichotomies is equal to $2^{K^n}$, which is the number of subsets of the state space $\{1, 2, \ldots, K\}^n$. If it were possible to design the complex-valued multistate Hopfield network as an ideal associative memory for every possible memory set, then this design would be a very efficient compression tool that enables the lossless compression of an arbitrary memory set into $(n^2+n)/2$ complex numbers. However, such a compression seems impossible from the information theory point of view, since the number of free variables, i.e., parameters, is quadratic in $n$, while the number of dichotomies grows exponentially with $n$. Therefore, if the design is based on a network model, which is the case for many neural associative memories, then only some of the possible memory sets can be introduced as fixed points to the network by any design method.

---

$$\left[ c_{12} \;\; s_{12} \;\; c_{13} \;\; s_{13} \;\; \cdots \;\; c_{1,n+1} \;\; s_{1,n+1} \;\vdots\; c_{23} \;\; s_{23} \;\; c_{24} \;\; s_{24} \;\; \cdots \;\; c_{2,n+1} \;\; s_{2,n+1} \;\vdots\; \cdots \;\vdots\; c_{n,n+1} \;\; s_{n,n+1} \right]$$

---

$$\hat{\mathbf{W}} = \begin{bmatrix} 0 & q_1^* + iq_2^* & q_3^* + iq_4^* & \cdots & q_{2n-1}^* + iq_{2n}^* \\ q_1^* - iq_2^* & 0 & q_{2n+1}^* + iq_{2n+2}^* & \cdots & q_{4n-3}^* + iq_{4n-2}^* \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ q_{2n-1}^* - iq_{2n}^* & q_{4n-3}^* - iq_{4n-2}^* & q_{6n-5}^* - iq_{6n-4}^* & \cdots & 0 \end{bmatrix}.$$

TABLE I
PERCENTAGES OF MEMORY SETS THAT YIELDED FEASIBLE INEQUALITY SYSTEMS

| $n$ | $\|M\|$ | $P\%\,(K=5)$ | $P\%\,(K=10)$ | $n$ | $\|M\|$ | $P\%\,(K=5)$ | $P\%\,(K=10)$ |
|---|---|---|---|---|---|---|---|
| 5 | 3 | 100 | 100 | 20 | 10 | 100 | 100 |
| | 5 | 90 | 95 | | 20 | 100 | 100 |
| | 7 | 2 | 5 | | 30 | 3 | 17 |
| 10 | 4 | 100 | 100 | 50 | 25 | 100 | 100 |
| | 10 | 97 | 100 | | 50 | 100 | 100 |
| | 15 | 0 | 12 | | 75 | 9 | 21 |

We say that a memory set $M$ is stored completely by our design method if each element of $M$ constitutes a fixed point in the resulting network. We measure the quantitative performance by the percentage of the number of completely stored memory sets among a collection of memory sets generated randomly. Recall that the complete storage of a memory set is equivalent to the feasibility of the inequality system (12) constructed for this set.

For some $n, |M|$ and $K$ values 100 random memory sets have been generated and checked whether each of these sets yielded a feasible inequality system or not. The number of sets that yielded a feasible inequality system for each experiment is listed in Table I, which shows that almost every set with $|M| \leq n$ can be completely stored independent of the value of $K$.

The effect of $K$ on complete storage performance is also shown in Table I. The probability of complete storage $P\%$ increases as the resolution factor $K$ increases for fixed $n$ and $|M|$. However, this would cause the state space to grow enormously and, hence, possibly cause more nontrivial spurious memories as illustrated in the next subsection.

### B. Application of the Design Procedure

We first give an illustrative example of proposed design procedure and investigate the performance of resulting network.

*Example 1:* Consider the memory set consisting of the following integral vectors:

$$\hat{\mathbf{x}}^1 = \begin{bmatrix} 3 \\ 5 \\ 5 \\ 3 \end{bmatrix} \; \hat{\mathbf{x}}^2 = \begin{bmatrix} 4 \\ 3 \\ 1 \\ 5 \end{bmatrix} \; \hat{\mathbf{x}}^3 = \begin{bmatrix} 4 \\ 4 \\ 5 \\ 4 \end{bmatrix} \; \hat{\mathbf{x}}^4 = \begin{bmatrix} 5 \\ 2 \\ 4 \\ 3 \end{bmatrix}$$

which belong to the integral lattice $\{1, 2, \ldots, 5\}^4$. We have first augmented one to each vector as the last entry and transformed them to their phase-modulated versions by (2)

$$\mathbf{x}^1 = \begin{bmatrix} e^{i6\pi/5} \\ e^{i\pi} \\ e^{i\pi} \\ e^{i6\pi/5} \end{bmatrix}$$

$$\mathbf{x}^2 = \begin{bmatrix} e^{i8\pi/5} \\ e^{i6\pi/5} \\ e^{i2\pi/5} \\ e^{i\pi} \end{bmatrix}$$

$$\mathbf{x}^3 = \begin{bmatrix} e^{i8\pi/5} \\ e^{i8\pi/5} \\ e^{i\pi} \\ e^{i2\pi/5} \end{bmatrix}$$

$$\mathbf{x}^4 = \begin{bmatrix} e^{i\pi} \\ e^{i4\pi/5} \\ e^{i8\pi/5} \\ e^{i6\pi/5} \end{bmatrix}$$

assuming that the resolution factor $K$ is equal to five. The inequality system has been constructed as in (12) and been solved by linear programming to obtain the weight matrix and the threshold vector as shown at the bottom of the page.

It can be verified that for these parameters each transformed memory vector $\mathbf{x}^i$ is a fixed point of the recursion (13). After injecting each 1-neighbor of each memory vector as the initial state vector it has been observed that the network converged to the nearest memory vector for each initial condition. Hence, it can be concluded that the design has been successful. We have also identified the spurious memories by checking the transformed version of each element of the integral lattice $\{1, 2, \ldots, 5\}^4$ and observed that the network has 15 spurious memories, none of which is trivial. Note that the same memory set can be embedded for a larger resolution factor. When the design is repeated for $K = 6$, one can see that the number of spurious memories increases by two.

Since gray-scale images can be represented by integral vectors, reconstruction of such images from their distorted versions constitutes a straightforward application of multistate associative memory, as investigated in [20]. The following example illustrates the performance of the proposed method in performing this task.

*Example 2:* Gray-scale versions of three well-known test images, namely Lenna, peppers, and cups images, have been used in this experiment. Due to computational limitations, the

$$\mathbf{W} = \begin{bmatrix} 0 & 7.4 - i68.4 & -65.6 + i132.8 & 139.7 - i31.7 \\ 7.4 + i68.4 & 0 & 108.4 - i17.8 & -76.5 - i92.6 \\ -65.6 - i132.8 & 108.4 + i17.8 & 0 & 80.9 + i167.1 \\ 139.7 + i31.7 & -76.5 + i92.6 & 80.9 - i167.1 & 0 \end{bmatrix}$$

$$\mathbf{t} = \begin{bmatrix} -73 - i134.1 \\ -82.6 + i46.1 \\ 126.6 - i131.3 \\ 20.7 + i174.4 \end{bmatrix}.$$

Fig. 2. Test images used in image reconstruction example.

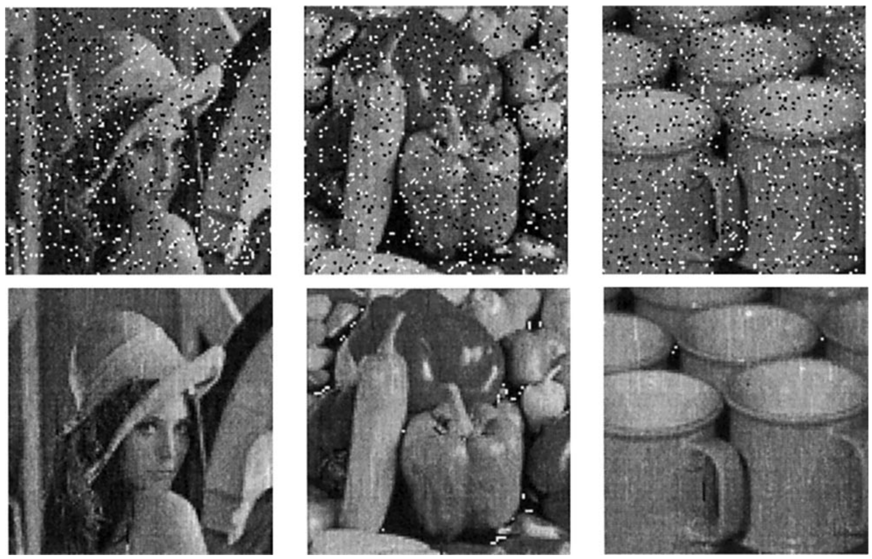


Fig. 3. Images corrupted by 20% salt-and-pepper noise (top) and their reconstructions obtained by the network (bottom).

original high-resolution 256-level images have been rescaled to $100 \times 100$ resolution and their gray-levels have been quantized down to 20 levels. Thus, each image can be considered as a $100 \times 100$ matrix consisting of integral numbers where 0 and 20 denote a black and a white pixel, respectively, and each integer value in between these values indicate a gray tone. These three prototype images are shown in Fig. 2.

Each image has been segmented into 500 20-dimensional vectors as $\mathbf{x}_{uv}^l \in \{1, 2, \ldots, 20\}^{20}$ for $u = 1, \ldots 5$ and $v = 1, \ldots 100$, such that $j$th column of $l$th image is represented by concatenating 5 of these integral vectors, namely $\mathbf{x}_{ij}^l$, $i = 1, \ldots, 5$. Here $l$ denotes the image index: 1 for Lenna, 2 for peppers, and 3 for cups. A 20-neuron complex-valued multistate associative memory has then been designed for each triple of memory vectors $\mathbf{x}_{uv}^1, \mathbf{x}_{uv}^2, \mathbf{x}_{uv}^3$, $u = 1, 2, \ldots, 5$ and $v = 1, 2, \ldots, 20$. Since we have attempted to embed only three vectors into a 20-neuron network by our method, which is far below the actual capacity investigated in Section III-A , all 500 designs have been successful.

After the design phase the distorted versions of the prototype images have been obtained by adding 20% salt-and-pepper noise, as shown in Fig. 3(top). Each of these distorted images was segmented the same way as described above, and then the transformed version of each vector obtained in this way as the initial condition to the corresponding network was applied. After all 500 networks reached their steady states, i.e., fixed points, the integral vectors have been obtained by the inverse transformation $\mathbf{p}_K^{-1}(\cdot)$ and combined in a $100 \times 100$ matrix. The reconstructed images obtained by this procedure for each distorted image are shown in the corresponding column of Fig. 3(b). It can then be concluded that the networks are capable of removing 20% salt-and-pepper noise on each image successfully. In other words, almost none of these 500 networks converges to a spurious memory in this experiment.

As the experiments were repeated for 40% and 60% noise [see Fig. 4(top) and Fig. 5(top), respectively], nontrivial spurious memories became effective in the recall, so reconstruction performance decreased. This can be observed from the recalled images shown in Fig. 4(bottom) and Fig. 5(bottom), respectively.

The tasks performed by a filter and by an associative memory are conceptually different: A filter is usually expected to remove noise on any signal, while an associative memory is designed to filter out the noise on prototype vectors only. However, despite the negative effects of spurious memories, the performance of the network in filtering noisy images is still comparable to that of median filtering, which is known to be one of the most effective methods for filtering out salt-and-pepper noise. This can be verified by Fig. 6(top) and (bottom), showing the reconstructed versions of 40% corrupted images obtained by our method and by median filtering, respectively.

The recall capability of our method with the generalized Hebb rule proposed in [9] was also compared. In this experiment, the three images in Fig. 2 were used as the prototype images in

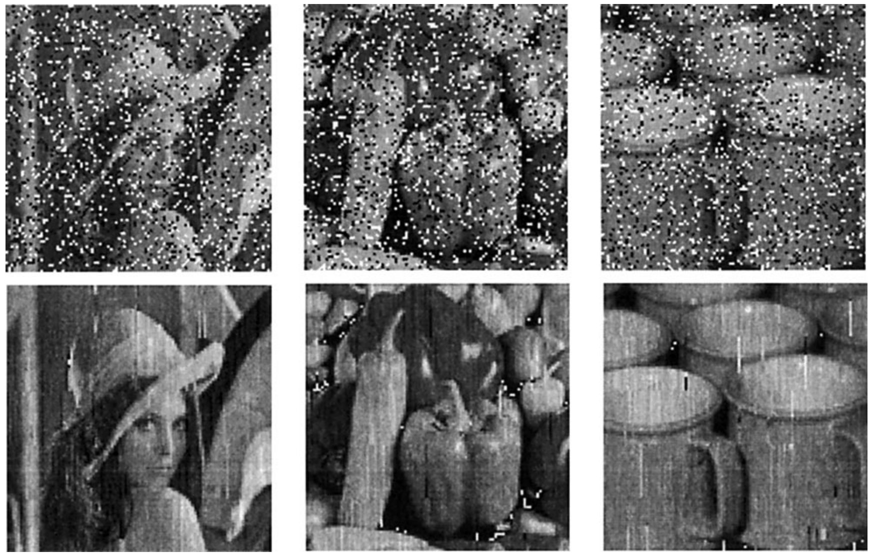


Fig. 4. Images corrupted by 40% salt-and-pepper noise (top) and their reconstructions obtained by the network (bottom).

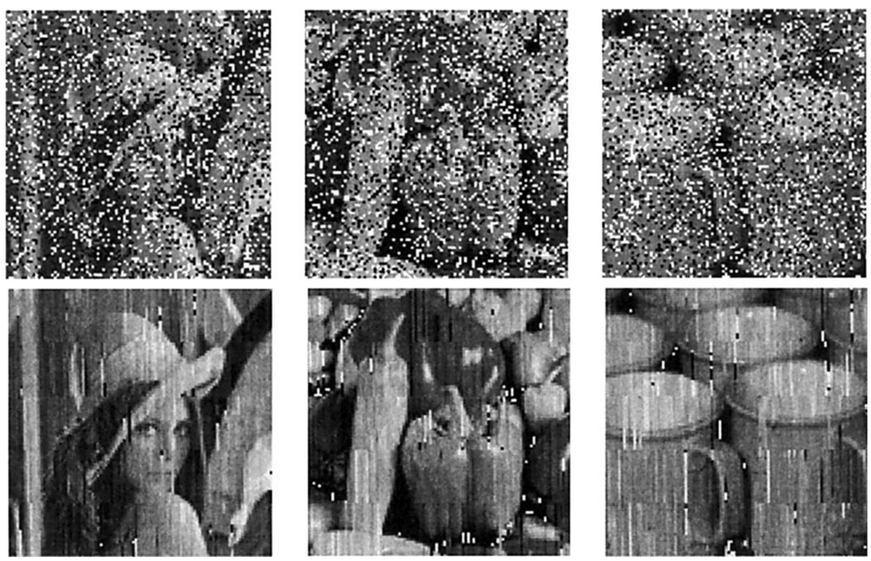


Fig. 5. Images corrupted by 60% salt-and-pepper noise (top) and their reconstructions obtained by the network (bottom).

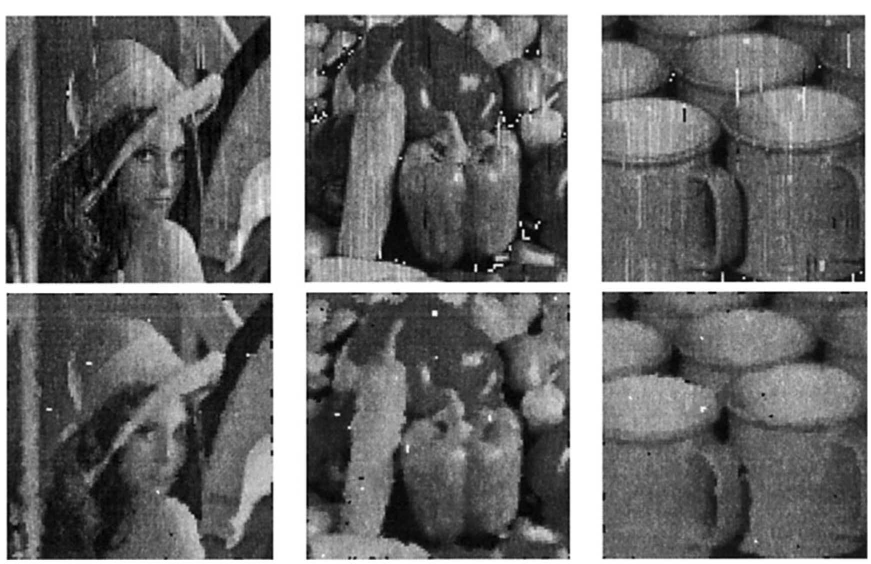
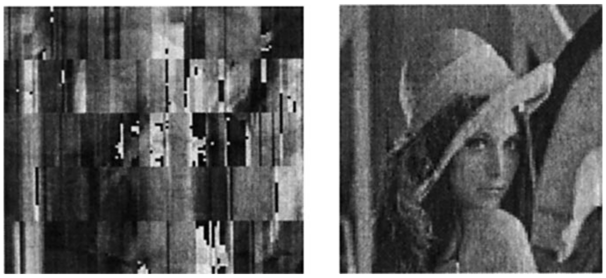


Fig. 6. Filtered images obtained from noisy images with 40% salt-and-pepper noise by the network (top) and by median filtering (bottom).



(a) (b)

Fig. 7. Lenna images obtained by the networks designed by (a) the generalized Hebb rule and (b) by the proposed method, respectively.

generalized Hebb rule. The dominant effect of spurious memories can be visually identified when Lenna image was about to be reconstructed from its 20% distorted version when generalized Hebb rule is used in the design [see Fig. 7(a)]. Our method, on the other hand, enables an almost perfect recall as shown in Fig. 7(b).

## IV. Conclusion

Besides some straightforward generalizations of the conventional Hopfield model, complex-valued multistate Hopfield network can also be an efficient tool to process static integral information. To support this idea, a design method for a subclass of this model has been proposed, and uses Hermitian network model to make it operate as a multistate associative memory. The new method was shown to outperform the generalized Hebb rule, which has yet constituted the only known so far learning rule for this model in associating phase-modulated integral information. The recall performance of the resulting network was illustrated on restoring gray-scale images, and the results have been satisfactory.

## References

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

[2] L. Personnaz, I. Guyon, and G. Dreyfus, "Collective computational properties of neural networks: New learning mechanisms," *Phys. Rev. A*, vol. 34, pp. 4217–4228, 1986.

[3] A. N. Michel and D. Liu, *Qualitative Analysis and Synthesis of Recurrent Neural Networks*. New York: Marcel Dekker, 2002.

[4] J. M. Zurada, I. Cloete, and E. van der Poel, "Generalized hopfield networks with multiple stable states," *Neurocomput.*, vol. 13, pp. 135–149, 1996.

[5] K. Shankmukh and Y. V. Venkatesh, "Generalized scheme for optimal learning in recurrent neural networks," *Proc. Inst. Elect. Eng. Vision Image Signal Processing*, vol. 142, pp. 71–77, 1995.

[6] E. Elizade and S. Gomez, "Multistate perceptrons: Learning rule and perceptron of maximal stability," *J. Phys. A, Math. Gen.*, vol. 25, pp. 5039–5045, 1992.

[7] S. Mertens, H. M. Koehler, and S. Bos, "Learning grey-toned patterns in neural networks," *J. Phys. A, Math. Gen.*, vol. 24, pp. 4941–4952, 1991.

[8] J. P. Nadal and A. Rau, "Storage capacity of potts-perceptron," *J. Phys. I France*, vol. 1, pp. 1109–1121, 1991.

[9] S. Jankowski, A. Lozowski, and J. M. Zurada, "Complex-valued multistate neural associative memory," *IEEE Trans. Neural Networks*, vol. 7, pp. 1491–1496, Nov. 1996.

[10] N. N. Aizenberg and Y. L. Ivaskiv, *Multivalued Threshold Logic* (in Russian). Kiev, Ukraine: Naukova Dumka, 1977.

[11] N. N. Aizenberg and I. N. Aizenberg, "CNN based on multivalued neuron as a model of associative memory for gray-scale images," in *Proc. 2nd Int. Workshop Cellular Neural Networks Applications*, Munich, Germany, 1992, p. 36.

[12] S. Tan, J. Hao, and J. Vandewalle, "Determination of weights for hopfield associative memory by error back propagation," in *Proc. IEEE Int. Symp. Circuits Systems*, vol. 5, 1991, p. 2491.

[13] S. Schwarz and W. Mathis, "Cellular neural network design with continuous signals," in *Proc. 2nd Int. Workshop Cellular Neural Networks Applications*, Munich, Germany, 1992, p. 17.

[14] M. Xiangwu and C. Hu, "Using evolutionary programming to construct Hopfield neural networks," in *Proc. IEEE Int. Conf. Intell. Processing Systems*, vol. 1, 1997, p. 571.

[15] M. K. Müezzinoglu, C. Güzeliş, and J. M. Zurada, "Construction of energy landscape for discrete Hopfield associative memory with guaranteed error correction capability," in Proc. 1st Int. IEEE-EMBS Conf. Neural Engineering, Capri, Italy, 2003.

[16] J. Bruck and J. W. Goodman, "A generalized convergence theorem for neural networks," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1089–1092, Sept. 1988.

[17] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*. Reading, MA: Addison-Wesley, 1973.

[18] F. Rosenblatt, *Principles of Neurodynamics*. New York: Spartan, 1962.

[19] J. M. Zurada, *Introduction to Artificial Neural Systems*. St. Paul, MN: West, 1992.

[20] J. M. Zurada, I. Cloete, and E. P. van der, "Neural associative memories with multiple stable states," in *Proc. 3rd Int. Conf. Fuzzy Logic, Neural Nets, and Soft Computing*, Iizuka, Fukuoka, Japan, 1994, pp. 45–51.

**Mehmet Kerem Müezzinoğlu** (S'01) was born in Ankara, Turkey, in 1976. He received the B.Sc. and M.Sc. degrees from Istanbul Technical University, Istanbul, Turkey, in 1998 and 2000, respectively. He is working toward the Ph.D. degree at Dokuz Eylül University, Izmir, Turkey.

He has been a Research and Teaching Assistant at Dokuz Eylül University since 2000. He held a Visiting Researcher position at the Computational Intelligence Laboratory, University of Louisville, Louisville, KY, in 2002. His research area includes analysis and synthesis of recurrent neural networks and pattern reconstruction.

Mr. Müezzinoglu has been awarded a Ph.D. scholarship by the Scientific and Technical Research Council of Turkey, Münir Birsel Foundation.

**Cüneyt Güzeliş** received the B.Sc., M.Sc., and Ph.D. degrees from Istanbul Technical University, Istanbul, Turkey, in 1981, 1984, and 1989, respectively.

He was a Visiting Researcher and Lecturer in the Department of Electrical Engineering and Computer Science, University of California, Berkeley, from April 1989 to April 1991, and was a Visiting Professor in the Information Laboratory, University of Paris-Nord, Paris, France, in September 1996 and June 1997. He has been a Full Professor at Dokuz Eylül University, Izmir, Turkey, since 1999. His research interests include nonlinear circuits and systems, neural networks and their signal processing applications.

**Jacek M. Zurada** (M'82–SM'83–F'96) is the S. T. Fife Alumni Professor of Electrical and Computer Engineering at the University of Louisville, KY. He is the Co-Editor of *Knowledge-Based Neurocomputing* (Cambridge, MA: MIT Press, 2000) and the author of *Introduction to Artificial Neural Systems* (New York: PWS, 1992) and a contributor to *Progress in Neural Networks* (New York: Ablex, 1994 and 1995) and Co-Editor of Computational Intelligence: Imitating Life (Piscataway, NJ: IEEE Press, 1994). He is the author or coauthor of more than 150 journal and conference papers in the area of neural networks and analog and digital VLSI circuits. Since 1998, he has been Editor-in-Chief of the IEEE Transactions on Neural Networks. He is also an Associate Editor of *Neurocomputing*, and was an Associate Editor of the IEEE Transactions on Circuits and Systems Parts I and II.

Dr. Zurada has received numerous awards for distinction in research and teaching, including the 1993 University of Louisville President's Award for Research, Scholarship, and Creative Activity and the 2001 University of Louisville President's Award for Service to Profession. He is an IEEE Distinguished Speaker.