in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 1992, vol. III, pp. 401–404.

[8] T. Hinamoto, H. Yamada, and Y. Zempo, "Weighted sensitivity minimization synthesis of 2-D filter structures using Fornasini-Marchesini second model," in *Proc. IEEE Int. Symp. Circuits Syst.*, 1995, vol. 3, pp. 2039–2042.

[9] U. Helmke and J. B. Moore, "L^2 sensitivity minimization of linear system representations via gradient flows," in *Proc. M.T.N.S. Conf.*, Kobe, Japan, 1990.

[10] G. Li and M. Gevers, "Optimal synthetic FWL design of state-space digital filters," in *Proc. IEEE Conf. Acoust., Speech, Signal Processing*, San Francisco, CA, Mar. 1992.

[11] M. Gevers and G. Li, "Parametrizations in control, estimation and filtering problems: Accuracy aspects," *Communication and Control Engineering Series*. London, England: Springer-Verlag, 1993.

[12] E. Fornasini and G. Marchesini, "Doubly-indexed dynamical systems: State-space models and structural properties," *Math. Syst. Theory*, vol. 12, pp. 59–72, 1978.

[13] P. Moroney, A. S. Willsky, and P. K. Houpt, "The digital implementation of control compensators: The coefficient word length issue," *IEEE Trans. Automatic Contr.*, vol. AC-25, pp. 621–630, Aug. 1980.

[14] R. E. Crochiere, "A new statistical approach to the coefficient word length problem for digital filters," *IEEE Trans. Circuits Syst.*, vol. CAS-22, pp. 190–196, Mar. 1975.

[15] M. Iwatsuki, M. Kawamata, and T. Higuchi, "Statistical sensitivity and minimum sensitivity structures with fewer coefficients in discrete time linear systems," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 72–80, Jan. 1989.

Cloning Template Design of Cellular Neural Networks for Associative Memories

Derong Liu

***Abstract*—This brief presents a synthesis procedure (design algorithm) for cellular neural networks with space-invariant cloning template with applications to associative memories. The design algorithm makes it possible to determine in a systematic manner cloning templates for cellular neural networks with or without symmetry constraints on the interconnection weights. Two specific examples are included to demonstrate the applicability of the methodology developed herein.**

***Index Terms*—Associative memories, cellular neural networks.**

I. Introduction

Cellular neural networks, first introduced in 1988 [4], have found many applications in image processing [14], [17], pattern recognition [5], [15], [16], [18], [20], and associative memories [2], [7]–[12], [19]. The recent increasing interest in cellular neural networks is partly due to the fact that such networks are among the easiest to implement in hardware. In this brief, we consider the realization of *associative memories* via a class of cellular neural networks introduced in [4]. The goal of associative memories is to store a set of desired patterns as stable memories such that a stored pattern can be recognized when the input pattern (or the initial pattern) contains sufficient information about that pattern. In practice the desired memory patterns are usually represented by bipolar vectors (or binary vectors).

Manuscript received November 27, 1995; revised April 16, 1996. This work was supported by Stevens Institute of Technology, Hoboken, NJ. This paper was recommended by Editor J. Nossek.

The author is with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030 USA (e-mail: dliu@stevens-tech.edu).

Publisher Item Identifier S 1057-7122(97)03483-1.

In [2], a synthesis procedure for associative memories using a class of discrete-time cellular neural networks with learning and forgetting capabilities is presented. In [7], an analytical method is presented for finding weight matrix for cellular neural networks formed of Hopfield networks. In [8]–[10], a synthesis method for associative memories using cellular neural networks is developed based on the so called *eigenstructure method* (cf. [6], [13]) where a set of linear equations are solved using singular value decomposition method. In [12], a design method for cellular neural networks with space-varying cloning template is developed using linear relaxation. In [19], a design algorithm is developed for a class of inputless cellular neural networks to possess prescribed stable and unstable memory points. The algorithm consists of formulating and solving a set of linear inequalities using linear programming and takes into account the robustness with respect to changes of the network parameters.

This brief makes contribution to cellular neural networks for associative memories. In particular, the results of the present brief will complement those of [2], [7]–[12], and [19] by developing an algorithm for the design of *space-invariant cloning template* for cellular neural networks. In Section II, we introduce preliminary existing results which will be used in Section III to develop an algorithm for the cloning template design. In Section IV, two specific examples are considered to demonstrate the applicability of the present results.

II. Preliminaries

We first introduce the model of a class of cellular neural networks used in the present brief.

A class of *two-dimensional continuous-time cellular neural networks* is described by equations of the form (cf. [3], [4])

$$\begin{cases} \dot{v}_{xij} = -\dfrac{1}{R_x} v_{xij} + \displaystyle\sum_{C(k,\,l)\in N_r(i,\,j)} W_{ij,\,kl}\, v_{ykl} + I_{ij} \\ v_{yij} = \mathrm{sat}\,(v_{xij}) \end{cases} \tag{1}$$

where $1 \le i \le M$, $1 \le j \le N$, $R_x > 0$, $\mathrm{sat}\,(v_{xij}) = \frac{1}{2}\,(|v_{xij}+1| - |v_{xij}-1|)$, and v_{xij} represents the states of the network, $\dot{v}_{xij}$ denotes the derivative of v_{xij} with respect to time t, and v_{yij} represents the outputs of the network. The basic unit in a cellular neural network is called a *cell*. In (1), there are $M \times N$ such cells arranged in an $M \times N$ array. The cell in the ith row and the jth column is denoted by $C(i,\,j)$ and an r-*neighborhood* $N_r(i,\,j)$ of the cell $C(i,\,j)$ for a positive integer r is defined by $N_r(i,\,j) \triangleq \{C(k,\,l)\colon \max\{|k-i|,\,|l-j|\} \le r,\, 1 \le k \le M,\, 1 \le l \le N\}$. $W_{ij,\,kl}$ denotes the connection weight from cell $C(k,\,l)$ to cell $C(i,\,j)$. I_{ij} is the bias term for cell $C(i,\,j)$.

In order to write (1) in a matrix format, we introduce the following notation. We define a matrix $Q = [Q_{ij,\,kl}] \in \Re^{MN\times MN}$ as

$$Q_{ij,\,kl} = \begin{cases} 1, & \text{if } C(k,\,l) \in N_r(i,\,j) \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

We let $S = Q = [S_{ij}] \in \Re^{n\times n}$, where $n = M \times N$, and we call S an *index matrix*. For $H = [H_{ij}] \in \Re^{n\times n}$, we define $H|S = [H^s_{ij}] \in \Re^{n\times n}$, where

$$H^s_{ij} = \begin{cases} H_{ij}, & \text{if } S_{ij} = 1 \\ 0, & \text{otherwise.} \end{cases}$$

We denote $W = [W_{ij,\,kl}] \in \Re^{MN\times MN}$, where $W_{ij,\,kl}$ is defined in (1), and let $T = W = [T_{ij}] \in \Re^{n\times n}$. Equation (1) can now be rewritten as

$$\begin{cases} \dot{x} = -Ax + T\,\mathrm{sat}\,(x) + I \\ y = \mathrm{sat}\,(x) \end{cases} \tag{3}$$

where $x = [v_{x11}, v_{x12}, \cdots, v_{x21}, \cdots, v_{xMN}]^T \in \Re^n$ is the state vector, $y = [v_{y11}, v_{y12}, \cdots, v_{y21}, \cdots, v_{yMN}]^T \in D^n \triangleq \{x \in \Re^n : -1 \le x_i \le 1, i = 1, \cdots, n\}$ is the output vector, $A = \mathrm{diag}\,[1/R_x, \cdots, 1/R_x] \in \Re^{n\times n}$ is the state transition matrix, $T = [T_{ij}] = T|S \in \Re^{n\times n}$ is the coefficient (or connection) matrix, $I \in \Re^n$ is a bias vector, and $\mathrm{sat}\,(x) = [\mathrm{sat}\,(x_1), \cdots, \mathrm{sat}\,(x_n)]^T$.

The dynamic rule of a cellular neural network can be completely specified by a $(2r+1)\times(2r+1)$ matrix which is called *cloning template*, or simply, *template*, and a bias current. For example, for a cellular neural network with a neighborhood radius $r = 1$, the cloning template of this network is of size 3×3; thus, in order to completely describe the dynamics of the network, we only need to specify the *nine* connection weights in the template (cf. [4]), and *one* bias term. In this case, the cloning template and the bias term are *space-invariant*. Space-varying cloning template design for associative memories has been considered, for example, in [10], [12], and [19]. One of the advantages of considering space-invariant cloning template for cellular neural networks is the small number of different connection weights required for describing a cellular neural network which can simplify the hardware implementation process [1], [3].

A. Analysis

We will call a vector α a *(stable) memory vector* (or simply, a *memory*) of system (3) if $\alpha = \mathrm{sat}(\beta)$ and if β is an asymptotically stable equilibrium point of system (3). We use B^n to denote the set of n-dimensional *bipolar vectors*, i.e., $B^n \triangleq \{x \in \Re^n\colon x_i = 1$ or $-1, i = 1, \cdots, n\}$. For $\alpha \in B^n$, we define $C(\alpha) = \{x \in \Re^n : x_i\alpha_i > 1,\ i = 1, \cdots, n\}$.

A result established in [8], [10] states the following.

Theorem 1: If $\alpha \in B^n$ and if $\beta = A^{-1}(T\alpha + I) \in C(\alpha)$, then (α, β) is a pair of stable memory vector and asymptotically stable equilibrium point of (3).

For the perturbation model of system (3) given by

$$\dot{x} = -Ax + (T + \Delta T)\,\mathrm{sat}\,(x) + (I + \Delta I) \tag{4}$$

where A, $T = T|S$, I, and $\mathrm{sat}\,(\cdot)$ are defined as in (3) and $\Delta T = \Delta T|S \in \Re^{n\times n}$ and $\Delta I \in \Re^n$ represent parameter perturbations, the following robustness analysis result has been established in [8], [11]. We will make use of the notation $\delta(x) = \min_{1\le i\le n}\{|x_i|\}$ for $x \in R^n$.

Theorem 2: Suppose that $\alpha^1, \cdots, \alpha^m \in B^n$ are desired stable memory vectors of system (3), and suppose that $\beta^1, \cdots, \beta^m$ are asymptotically stable equilibrium points of system (3) corresponding to $\alpha^1, \cdots, \alpha^m$, respectively, i.e., $\beta^i = A^{-1}(T\alpha^i + I)$ for $i = 1, \cdots, m$. Let

$$\nu = \min_{1\le k\le m}\{\delta(\beta^k)\} > 1. \tag{5}$$

Then, $\alpha^1, \cdots, \alpha^m$ are also stable memory vectors of (4) provided that $\|A^{-1}\Delta T\|_\infty + \|A^{-1}\Delta I\|_\infty < \nu - 1$, where $\|\cdot\|_\infty$ denotes the matrix norm induced by the l_∞ vector norm.

Theorem 2 implies that the asymptotically stable equilibrium points $\overline{\beta}^i$ corresponding to α^i after parameter perturbations satisfy that $\overline{\beta}^i = A^{-1}[(T+\Delta T)\alpha^i + (I+\Delta I)] \in C(\alpha^i)$ for $i = 1, 2, \cdots, m$ provided that $\|A^{-1}\Delta T\|_\infty + \|A^{-1}\Delta I\|_\infty < \nu - 1$. We recall that the matrix norm induced by the l_∞ vector norm for a matrix $F = [f_{ij}] \in \Re^{m\times n}$ is defined by $\|F\|_\infty = \max_{1\le i\le m}\{\sum_{j=1}^n |f_{ij}|\}$.

B. Synthesis

We summarize in the following the design procedure developed in [8]–[10].

Design Problem: Given an $n \times n$ index matrix $S = [S_{ij}]$ for the cellular neural network described by (3) and m vectors $\alpha^1, \cdots, \alpha^m$ in B^n, choose $\{A, T, I\}$ with $T = T|S$ in such a manner that $\alpha^1, \cdots, \alpha^m$ are stable memory vectors of system (3). ■

Summary of the Design Procedure:

1) Choose matrix A as the identity matrix.
2) Choose a real number $\mu > 1$ and set $\beta^i = \mu\alpha^i$ for $i = 1, \cdots, m$.
3) Compute the $n \times (m-1)$ matrix $Y = [\alpha^1 - \alpha^m, \cdots, \alpha^{m-1} - \alpha^m]$.
4) Determine matrix T with the constraints $T = T|S$ from $TY = \mu Y$.
5) Compute the bias vector $I = \beta^m - T\alpha^m$.

Then, $\alpha^1, \cdots, \alpha^m$ will be stored as stable memory vectors for system (3) with A, T, and I determined as above. The states β^i corresponding to α^i, $i = 1, \cdots, m$, will be asymptotically stable equilibrium points of the synthesized system ■

It is readily seen from Theorem 2 of Section II that the synthesis procedure presented above guarantees that $\alpha^1, \cdots, \alpha^m$ are also stable memory vectors of system (2) provided that $\|A^{-1}\Delta T\|_\infty + \|A^{-1}\Delta I\|_\infty = \|\Delta T\|_\infty + \|\Delta I\|_\infty < \mu - 1$. ■

Remark 1: If ones wishes to design a network with $I = 0$ (in Step 5), Step 3 above should be modified to $Y = [\alpha^1, \cdots, \alpha^m]$ ■

Remark 2: The design procedure summarized above is developed in [8]–[10] in the same spirit as the one developed in [6] for *fully connected neural networks*. It has been pointed out in [6, Theorem 5.2] that the design method developed therein (hence, the present one) does indeed make attempts to decrease the number of spurious (superfluous) equilibria. ■

III. Cloning Template Design

For a cellular neural network described by (3) with $n = M \times N$ cells and with neighborhood radius r, we need a $(2r+1)\times(2r+1)$ (space-invariant) cloning template to describe the connection matrix T. We use P_k, $k = 1, 2, \cdots, (2r+1)^2$, to denote these elements and we write

$$P = \langle P_k\rangle = \begin{bmatrix} P_1 & P_2 & \cdots & P_{(2r+1)} \\ P_{(2r+1)+1} & P_{(2r+1)+2} & \cdots & P_{2\times(2r+1)} \\ \vdots & \vdots & \ddots & \vdots \\ \cdots & \cdots & \cdots & P_{(2r+1)^2} \end{bmatrix} \in \Re.^{(2r+1)\times(2r+1)}$$

If a cellular neural network (3) has a cloning template $\langle P_k\rangle$, the connection matrix T will consist of only zeros and P_k's. The index matrix S can be expressed as

$$S = \sum_{k=1}^{(2r+1)^2} S^k \tag{6}$$

where $S^k = [S_{ij}^k] \in \Re^{n\times n}$ and for $k = 1, 2, \cdots, (2r+1)^2$

$$S_{ij}^k = \begin{cases} 1, & \text{if } T_{ij} = P_k\text{, i.e., if the } ij\text{th element} \\ & \text{of } T \text{ corresponds to the element } k\text{th} \\ 0, & \text{of the template otherwise.} \end{cases} \tag{7}$$

S^k describes how the $(2r+1)^2$ elements of the cloning template are distributed in the connection matrix T. For the A, T, and I determined by the Design Procedure summarized in Section II with

$\mu > 1$, let us compute

$$F_k = \left(\sum_{i,j} S_{ij}^k \right)^{-1} \sum_{S_{ij}^k = 1} T_{ij}, \qquad \text{for } k = 1, 2, \cdots, (2r+1)^2. \tag{8}$$

Let $G = [G_{ij}] \in \Re^{n \times n}$, where

$$G_{ij} = \begin{cases} F_k, & \text{if } S_{ij}^k = 1 \text{ for } k = 1, 2, \cdots, (2r+1)^2 \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Let $\Delta T = G - T$. Then, $T + \Delta T = G$ is a matrix with cloning template $\langle F_k \rangle$. From Theorem 2 of Section II, we note that if

$$\|A^{-1} \Delta T\|_\infty = \|G - T\|_\infty, \qquad < \mu - 1 \tag{10}$$

and if $\Delta I = 0$, the neural network (4) will also store all the desired patterns as stable memories, with connection matrix $T + \Delta T$ which has a cloning template $\langle F_k \rangle$.

The above observation gives rise to the possibility of designing a cellular neural network (3) with connection matrix T specified by a cloning template. In the following, we develop an *iterative algorithm* (design procedure) which in most cases will result in a cellular neural network (3) with connection matrix specified by a cloning template. In doing so, we apply Theorems 1 and 2 of Section II *iteratively*. Let G be defined in (9) and $\Delta T = G - T$, where $T = T|S$ is obtained by the Design Procedure summarized in Section II with $\mu > 1$. For the given μ, suppose that $\|\Delta T\|_\infty \geq \mu - 1$. We can find a λ, $0 < \lambda < 1$, such that $\lambda \|\Delta T\|_\infty < \mu - 1$, and we let $T_1 = T + \lambda \Delta T$. We use this T_1 as the *new* connection matrix for our cellular neural network (3). According to Theorem 2 we see that $\alpha^1, \cdots, \alpha^m$ are still stable memory vectors of system (3) with coefficient matrix T_1. Using Theorem 1 we can determine the corresponding asymptotically stable equilibrium points as $\overline{\beta}^k = A^{-1}(T_1 \alpha^k + I)$ for $k = 1, \cdots, m$. Clearly, we have $\overline{\beta}^k \in C(\alpha^k)$. Using Theorem 2, we can determine the upper bound $\nu - 1$ for the permissible perturbation ΔT from (5), where we use $\overline{\beta}^k$ instead of β^k. We *repeat* the above procedure, until we determine a coefficient matrix T which is specified by a cloning template $(T = G)$ or arrive at $\nu \leq 1 + \eta$ (where η is a small positive number).

We summarize in the following our design procedure.

CNN Cloning Template Design Procedure: Given positive integers r, n, M, and N with $n = M \times N$, and m vectors $\alpha^1, \cdots, \alpha^m$ in B^n which are to be stored as stable memory vectors for cellular neural network (3). We proceed as follows.

1) Compute the matrix Q as in (2) and denote $S = Q = [S_{ij}] \in \Re^{n \times n}$. Compute S^k for $k = 1, 2, \cdots, (2r+1)^2$, as in (7).
2) According to the Design Procedure summerized in Section II, we choose A as the identity matrix and we determine $T = T|S$ and I for cellular neural network (3) with a $\mu > 1 + \eta$ (e.g., $\mu = 30$, $\eta = 0.001$).
3) Compute G as in (8) and (9).
4) If $T = G$ or $\mu \leq 1 + \eta$, stop. Otherwise go to Step 5.
5) Compute $\Delta T = G - T$. If $\|\Delta T\|_\infty < \mu - 1$, choose $\lambda = 1$. Otherwise, choose

$$\lambda = \frac{\mu - 1}{\|\Delta T\|_\infty} - \varepsilon$$

where ε is a small positive number (e.g., $\varepsilon = 0.01$). Compute $T_1 = T + \lambda \Delta T$.
6) Compute $\overline{\beta}^k = A^{-1}(T_1 \alpha^k + I)$ for $k = 1, \cdots, m$, and compute $\nu = \min_{1 \leq k \leq m} \{\delta(\overline{\beta}^k)\} > 1$.
7) Replacing μ by ν and T by T_1, go to Step 3. ■

If we end up with $T = G$, we have found a solution for our cloning template design problem, and $\alpha^1, \cdots, \alpha^m$ will be stored as stable memory vectors for system (3) with A, T, and I determined as above. If we end up with $\nu \leq 1 + \eta$ and $T \neq G$, our design procedure is not successful in solving the given cloning template design problem. Experimental results indicated that this procedure will frequently succeed in determining a matrix T which is indeed specified by a cloning template.

Remark 3: The above design procedure will usually result in a space-invariant cloning template and space-varying bias terms. Space-invariant bias can be achieved by either designing a cellular neural network with $I = 0$ (cf., Remark 1), or modifying th CNN Cloning Template Design Procedure so that similar iterative procedure (Steps 3–7 above) is used for vector I by initially choosing $\Delta I = [\Delta I_1, \Delta I_2, \cdots, \Delta I_n]^T$ with

$$\Delta I_i = \frac{1}{n} \sum_{i=1}^{n} I_i - I_i.$$

■

Remark 4: The CNN Cloning Template Design Procedure presented above will usually result in a nonsymmetric coefficient matrix T since the resulting cloning template is usually not with the special structure to render a symmetric T [refer to, e.g., (11). We can apply the *Symmetric Design Procedure* developed in [8] to the matrix T obtained by the Design Procedure of Section II to determine a *symmetric* connection matrix T_s which also satisfies that $T_s|S = T_s$, and then starting with T_s and using the CNN Cloning Template Design Procedure of the present section we can determine a connection matrix T which is *symmetric* and which is also *specified by a cloning template*. Such capability is of *great* interest since cellular neural network (3) will be *globally stable* when T is symmetric (cf. [4]). ■

Remark 5: The F_k, $k = 1, 2, \cdots, (2r+1)^2$, computed in (8) denote the target values for the cloning template with each element to be the average of all the elements appearing at the locations corresponding to the same template element in matrix T. We note that there may be other different ways for computing F_k's. Simulation results showed that the choice in (8) works very well (cf., Example 1). ■

IV. Simulation Study

Example 1: The CNN Cloning Template Design Procedure is used to design cellular neural networks (3) with $n = 12$ ($M = 4$, $N = 3$) and $r = 1$ (neighborhood radius). The design process are repeated 20 times using different sets of desired vectors to be stored as memory vectors. Each set contains $m = 4$ vectors in B^n which are generated randomly. For each given set of vectors, we synthesize system (3) using the CNN Cloning Template Design Procedure. There were only *two* tests out of 20 in which we did not succeed in finding a space-invariant cloning template. Table I summarizes our findings. Also shown in Table I are the results for system (3) synthesized by the Design Procedure summarized in Section II for the same sets of desired vectors to be stored as memories.

From Table I we see that in the present example, cellular neural networks with space-invariant cloning template for associative memories have more spurious memories than the network with space-varying cloning templates. ■

Example 2: In order to compare the present result with one of the existing space-varying cloning template design methods [12], the same example as in [12] is used. Consider a design of a cellular neural network (3) with 36 neurons ($n = 36$) with $M = 6$, $N = 6$, and $r = 1$. The objective is to store the six patterns shown in Fig. 1 as stable memories (black = 1 and white = −1).

TABLE I

	The space-varying design (out of 20)	The space-invariant design (out of 18)
average of total number of memory vectors in B^n	12.1	52.4
average of total number of undesired memory vectors in B^n	8.1	48.4
total number of desired patterns which were not stored as memory vectors	0	0

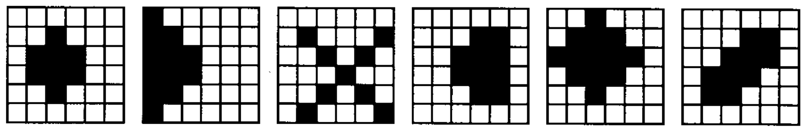

Fig. 1. The six desired memory patterns.

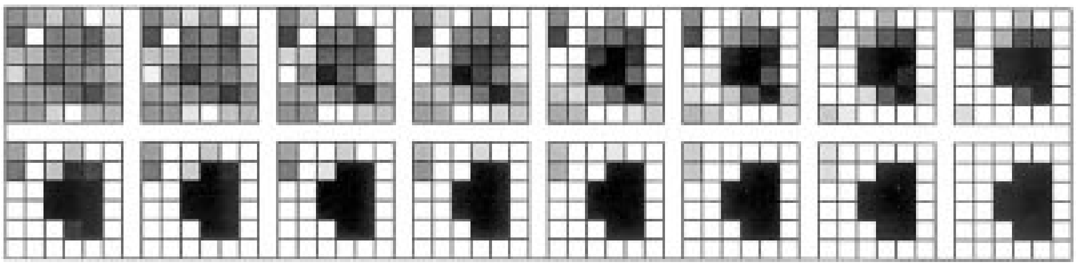

Fig. 2. A typical evolution of pattern no. 4.

Case I—Nonsymmetric Design: We utilize the CNN Cloning Template Design Procedure of Section III to design a cellular neural network (with $I = 0$). The matrix T obtained has a cloning template

$$\langle P_k \rangle = \begin{bmatrix} 0.6582 & 0.9458 & 0.5621 \\ 0.3777 & 4.9381 & 0.4499 \\ 0.5230 & 0.9710 & 0.6495 \end{bmatrix}.$$

A comparison study between the present space-invariant cloning template design and the space-varying designs [10], [12] indicated that the cellular neural network with space-invariant cloning template has smaller basins of attraction for each desired pattern than those with space-varying cloning template designed as in [10] and [12]. This is consistent with the conclusion of Example 1.

Case II—Symmetric Design: Using the Symmetric Design Procedure developed in [8], we first determine a symmetric matrix T_1 for the present design. Starting with this symmetric T_1, we obtain a matrix T which is also symmetric using the CNN Cloning Template Design Procedure. Matrix T in this case has a cloning template given by

$$\langle P_k \rangle = \begin{bmatrix} 0.7139 & 1.0520 & 0.5912 \\ 0.4517 & 5.2901 & 0.4517 \\ 0.5912 & 1.0520 & 0.7139 \end{bmatrix}. \tag{11}$$

It can be verified by Theorem 1 that the four desired patterns are also stable memory vectors for system (3) with a symmetric connection matrix. Notice that the structure of the cloning template in (11) is a typical one for $r = 1$ which renders a *symmetric interconnection matrix*. We can see that this cloning template requires symmetric connections in vertical, horizontal, as well as diagonal directions from any cell unit.

The performance of this network is illustrated by means of a typical simulation run of (3), shown in Fig. 2. ■

References

[1] M. Anguita, F. J. Pelayo, A. Prieto, and J. Ortega, "Analog CMOS implementation of a discrete time CNN with programmable cloning templates," *IEEE Trans. Circuits Syst. II,* vol. 40, pp. 215–218, Mar. 1993.

[2] M. Brucoli, L. Carnimeo, and G. Grassi, "Discrete-time cellular neural networks for associative memories with learning and forgetting capabilities," *IEEE Trans. Circuits Syst. I.,* vol. 42, pp. 396–399, July 1995.

[3] L. O. Chua and T. Roska, "The CNN paradigm," *IEEE Trans. Circuits Syst. I,* vol. 40, pp. 147–156, Mar. 1993.

[4] L. O. Chua and L. Yang, "Cellular neural networks: Theory," *IEEE Trans. Circuits Syst.,* vol. 35, pp. 1257–1272, Oct. 1988.

[5] ——, "Cellular neural networks: Applications," *IEEE Trans. Circuits Syst.,* vol. 35, pp. 1273–1290, Oct. 1988.

[6] J.-H. Li, A. N. Michel, and W. Porod, "Analysis and synthesis of a class of neural networks: Linear systems operating on a closed hypercube," *IEEE Trans. Circuits Syst.,* vol. 36, pp. 1405–1422, Nov. 1989.

[7] B. Ling and F. M. A. Salam, "A cellular network formed of Hopfield networks," in *Proc. 35th Midwest Symp. Circuits Syst.,* Washington, DC, Aug. 1992, pp. 256–259.

[8] D. Liu and A. N. Michel, *Dynamical Systems With Saturation Nonlinearities: Analysis and Design.* New York: Springer-Verlag, 1994.

[9] ——, "Cellular neural networks for associative memories," *IEEE Trans. Circuits Syst. II,* vol. 40, pp. 119–121, Feb. 1993.

[10] ——, "Sparsely interconnected neural networks for associative memories with applications to cellular neural networks," *IEEE Trans. Circuits Syst. II.* vol. 41, pp. 295–309, Apr. 1994.

[11] ——, "Robustness analysis and design of a class of neural networks with sparse interconnecting structure," *Neurocomput.,* vol. 12, pp. 59–76, June 1996.

[12] G. Martinelli and R. Perfetti, "Associative memory design using space-varying cellular neural networks," in *Proc. 2nd Int. Workshop Cellular Neural Networks Their Applicat.,* Munich, Germany, Oct. 1992, pp. 117–122.

[13] A. N. Michel and J. A. Farrell, "Associative memories via artificial neural networks," *IEEE Contr. Syst. Mag.,* vol. 10, pp. 6–17, Apr. 1990.

[14] T. Matsumoto, L. O. Chua, and R. Furukawa, "CNN cloning template: Hole-filler," *IEEE Trans. Circuits Syst.,* vol. 37, pp. 635–638, May 1990.

[15] T. Matsumoto, L. O. Chua, and H. Suzuki, "CNN cloning template: Connected component detector," *IEEE Trans. Circuits Syst,* vol. 37, pp. 633–635, May 1990.

[16] ——, "CNN cloning template: Shadow detector," *IEEE Trans. Circuits Syst.,* vol. 37, pp. 1070–1073, Aug. 1990.

[17] T. Matsumoto, L. O. Chua, and T. Yokohama, "Image thinning with a cellular neural network," *IEEE Trans. Circuits Syst.,* vol. 37, pp.

638–640, May 1990.
[18] G. Seiler and J. A. Nossek, "Winner-take-all cellular neural networks," *IEEE Trans. Circuits Syst. II,* vol. 40, pp. 184–190, Mar. 1993.
[19] G. Seiler, A. J. Schuler, and J. A. Nossek, "Design of robust cellular neural networks," *IEEE Trans. Circuits Syst. I,* vol. 40, pp. 358–364, May 1993.
[20] T. Szirányi and J. Csicsvári, "High-speed character recognition using a dual cellular neural network architecture (CNND)," *IEEE Trans. Circuits Syst. II,* vol. 40, pp. 223–231, Mar. 1993.

Weighted Low-Rank Approximation of General Complex Matrices and Its Application in the Design of 2-D Digital Filters

W.-S. Lu, S.-C. Pei, and P.-H. Wang

***Abstract*—In this brief we present a method for the weighted low-rank approximation of general complex matrices along with an algorithmic development for its computation. The method developed can be viewed as an extension of the conventional singular value decomposition to include a nontrivial weighting matrix in the approximation error measure. It is shown that the optimal rank-*K* weighted approximation can be achieved by computing *K* generalized Schmidt pairs and an iterative algorithm is presented to compute them. Application of the proposed algorithm to the design of FIR two-dimensional (2-D) digital filters is described to demonstrate the usefulness of the algorithm proposed.**

***Index Terms*—2-D digital filters, singular value decomposition.**

I. INTRODUCTION

As one of the basic and important tools in numerical linear algebra, the singular value decomposition (SVD) [1]–[3] has found numerous scientific and engineering applications in the past. An excellent outline on its applications in linear algebra and linear systems can be found in [4]. Sample applications of the SVD in automatic control, robotics, image processing, reduced-rank signal processing, and design of two-dimensional (2-D) digital filters can also be found in [5]–[16]. In a filter design context, the SVD method [10]–[16] starts with a complex matrix F obtained by sampling the desired frequency response, and the application of SVD to F allows one to decompose a complex 2-D design task into a set of simple 1-D design tasks with guaranteed design accuracy. An important property of the SVD utilized in this regard is that the SVD of F of rank r offers a series of optimal low-rank approximations of F in both Euclidean and Frobenius norm sense. That is, if

$$F = U\Sigma V^H = \sum_{i=1}^{r} \sigma_i u_i v_i^H \tag{1}$$

is a SVD of F, then for *any* K between 1 and r,

$$\min_{\mathrm{rank}(\hat{F}_K)=K} \|F - \hat{F}_K\|_{2,F} = \|F - F_K\|_{2,F} \tag{2}$$

where

$$F_K = \sum_{i=1}^{K} \sigma_i u_i v_i^H. \tag{3}$$

Although the SVD method has become a successful design tool, a weak point of the method is that it treats all entries of the sampled frequency response matrix *equally,* which could in some cases lead to degraded designs. In order to discriminate between the important and unimportant portions of the matrix, we seek to find a low-rank approximation of F such that for a fixed K with $1 < K < r$, the rank K matrix

$$F_K = \sum_{i=1}^{K} \sigma_i u_i v_i^H \tag{4}$$

best approximates F in the *weighted* Frobenius norm sense. That is

$$\min_{\mathrm{rank}(\hat{F}_K)=K} \|W \circ (F - \hat{F}_K)\|_F = \|W \circ (F - F_K)\|_F \tag{5}$$

where W is a weighting matrix with the same size as F, $W \circ Y$ denotes the entrywise multiplication of W with Y, which is often termed as Hadamard or Schur product in the literature. In the rest of the brief, we shall call (4), (5) a weighted rank K approximation of F.

In the literature the weighted low rank approximation (WLRA) problem was considered by Shpak [16] in a filter design context for a real matrix F. His approach is to treat (5) as a numerical minimization problem so that the conventional optimization techniques [17], [18] can be used to find a solution. However, the optimization involved requires a large amount of computation, particularly when u_i and v_i are of high dimension. The objectives of this brief are twofold. First, we investigate in Section II the WLRA for a general complex matrix $F \in C^{m\times n}$. It is shown that for a fixed K (which is the rank of F_K approximating F), the WLRA can be characterized by K generalized Schmidt pairs which are nonlinear extension of the conventional Schmidt pairs obtained by the SVD of F. We present an iterative algorithm for numerical computation of the generalized Schmidt pairs. Convergence and computation complexity issues of the algorithm are addressed. Also proposed in Section II is a suboptimal solution to the WLRA problem. This suboptimal WLRA (S-WLRA) is obtained by computing one pair of vectors u_i and v_i at a time, leading to considerably reduced computation complexity and hence offers a feasible solution to those approximation problems where the matrix F is of high dimension. As the second objective of the brief, the S-WLRA is applied to design FIR 2-D digital filters. In Section III, the S-WLRA is applied to design linear phase FIR 2-D filters. An example is included to illustrate the design algorithm and to compare the WLRA-SVD method with the conventional SVD method.

II. WEIGHTED LOW-RANK APPROXIMATION OF COMPLEX MATRICES

A. Preliminaries

The singular value decomposition of a rectangular complex matrix $F \in C^{m\times n}$ is the decomposition (1) where $U \in C^{m\times m}$, $V \in C^{n\times n}$ are unitary and

$$\Sigma = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix}$$

where $\Sigma_r = \mathrm{diag}\{\sigma_1, \ldots, \sigma_r\}$ with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$, $r = \mathrm{rank}(F)$. Writing $U = [u_1 \cdots u_r \cdots u_m]$ and

Manuscript received April 25, 1995; revised September 10, 1996. This work was supported by the National Science Council under Grant NSC84-2213-E002-076. This paper was recommended by Associate Editor H. Reddy.

W.-S. Lu is with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, B.C., Canada V8W 3P6.

S.-C. Pei and P.-H. Wang are with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.

Publisher Item Identifier S 1057-7122(97)03515-0.