

Class of continuous level associative memory neural nets

Robert J. Marks II

A neural net capable of restoring continuous level library vectors from memory is considered. As with Hopfield's neural net content addressable memory, the vectors in the memory library are used to program the neural interconnects. Given a portion of one of the library vectors, the net extrapolates the remainder. Sufficient conditions for convergence are stated. Effects of processor inexactitude and net faults are discussed. A more efficient computational technique for performing the memory extrapolation (at the cost of fault tolerance) is derived. The special case of table lookup memories is addressed specifically.

I. Introduction

Hopfield's neural net content addressable memory (CAM)¹ has stirred great interest in the signal processing community. The net has been implemented both optically²⁻⁵ and electronically.⁶ For optical implementation, intensive neural interconnects are possible since light paths can cross without interference. Planar VLSI implementations, on the other hand, are restricted to nearest-neighbor interconnects. The interconnects in Hopfield's CAM are programmed by a set of binary library vectors. Given a noisy subset of one of the library vectors, the neural net ideally converges to the library vector closest to the initialization. The net can operate asynchronously or synchronously. It is also tolerant of both lumped and distributed faults.^{3,6} Thus, analog optical processor inexactitude is of less significance than usual.

The neural net introduced in this paper allows for library vectors with continuous elements. The interconnects are determined analogous to Hopfield's recipe. The net can also operate asynchronously and is fault tolerant. It differs from Hopfield's in that the initially known neural states are imposed on the net each iteration. That is, the known states act as the net stimulus and the remaining nodes catalog the response. A human memory analogy is our ability to recall a well-known painting by continuously viewing only a portion of it.

After a brief introduction to the mathematics of the neural net, we specifically define the extrapolation neural net. Borrowing from some recent results in iterative signal recovery and synthesis,⁷⁻¹¹ important insights into the net's performance are generated. These include sufficient conditions for convergence to the proper library vector and effects of known state perturbations. A short section on fault tolerance contains empirical evidence that the net still works well for both quantized and deleted interconnects. A table lookup net is one where the same P nodes are always used as the net stimulus. Neural net architectures for these specific memory extrapolation problems are presented. Some final remarks tying the net's operation to some other well-known iterative algorithms are made in the conclusions.

II. Preliminaries

Consider a neural net of L nodes. The transmission from the k th to the i th node is t_{ik} . We will assume a symmetric net ($t_{ki} = t_{ik}$) and will allow for autointerconnects ($t_{kk} \neq 0$). The state s_k of the k th node will be assumed to be a function of the sum of its inputs. For synchronous operation (i.e., all delays between node pairs are identical), we have at time M ,

$$\mathbf{i}_M = T\mathbf{s}_M, \quad (1)$$

where \mathbf{s}_M is a vector of the L neural states at time M , \mathbf{i}_M is the vector of the L input sums at time M , and T is the matrix of the t_{ik} terms. Let \mathcal{N} denote the node operator that determines the next set of states from the input sum:

$$\mathbf{s}_{M+1} = \mathcal{N}\mathbf{i}_M. \quad (2)$$

Since the state of the k th node depends only on its input sum, \mathcal{N} must be a pointwise operator. That is, the k th element of \mathbf{s}_{M+1} depends only on the k th element of \mathbf{i}_M .

The author is with University of Washington, Interactive Systems Design Laboratory, Seattle, Washington 98195.

Received 26 August 1986.

0003-6935/87/102005-06\$02.00/0.

© 1987 Optical Society of America.

Substituting Eq. (1) into Eq. (2) gives the state iteration equation:

$$s_{M+1} = \mathcal{N}Ts_M \quad (3)$$

We illustrate with two short examples, saving our memory extrapolation net for a more detailed treatment.

A. Solving Simultaneous Equations

Consider the L linear equations $\mathbf{g} = \mathbf{K}\mathbf{f}$. Given \mathbf{g} and \mathbf{K} , we wish to find \mathbf{f} . Design a neural net with $\mathbf{T} = \mathbf{I} - \mathbf{K}$ and let the neural operator be defined for an arbitrary vector \mathbf{i} by [see Fig. 1(a)] $\mathcal{N}\mathbf{i} = \mathbf{i} + \mathbf{g}$. Thus, the k th node adds g_k to the sum of the node's inputs. Then with initialization $s_0 = \mathbf{g}$, Eq. (3) can be inductively shown to be equivalent to

$$s_M = \sum_{m=0}^M \mathbf{T}^m \mathbf{g}.$$

If $\|\mathbf{T}\| < 1$, we can use a generalized geometric series and write

$$\begin{aligned} s_\infty &= [\mathbf{I} - \mathbf{T}]^{-1} \mathbf{g} \\ &= \mathbf{f}. \end{aligned}$$

The net thus ideally converges to our desired result.¹²

B. Hopfield's Neural Net

Let $\{\mathbf{b}_n | 1 \leq n \leq N\}$ denote N library vectors each with only ± 1 elements. Define the library matrix $\mathbf{B} = [\mathbf{b}_1 : \mathbf{b}_2 : \dots : \mathbf{b}_N]$. From this, we form the interconnect matrix $\mathbf{T} = \mathbf{B}\mathbf{B}^T - \mathbf{N}\mathbf{I}$, where the superscript T denotes transposition. (Note that $t_{kk} = 0$.) Let the node operator be [see Fig. 1(b)] $\mathcal{N} = \text{sgn}$, where sgn performs a signum operation on each vector element. The resulting neural net is Hopfield's CAM. For an initialization \mathbf{g} and $N \ll L$, the net's state many times will converge to the library vector closest to \mathbf{g} in the Hamming sense.

III. Memory Extrapolation Net

Consider a set \mathcal{F} of N continuous level linearly independent vectors of length $L \geq N$: $\mathcal{F} = \{\mathbf{f}_n | 1 \leq n \leq N\}$ and the corresponding library matrix: $\mathbf{F} = [\mathbf{f}_1 : \mathbf{f}_2 : \dots : \mathbf{f}_N]$. We form a neural net with interconnects⁵

$$\mathbf{T} = \mathbf{F}(\mathbf{F}^T\mathbf{F})^{-1}\mathbf{F}^T. \quad (4)$$

Given a portion of one of the library vectors, a memory extrapolator, using the library, will reconstruct the remainder of that vector. For our net, we will divide the nodes into two sets: one in which states are known and the remainder in which the states are unknown. This node partition may change from application to application. That is, any node may be used to stimulate or to respond. Without loss of generality, assume that states 1 through $P < L$ (corresponding to the first P elements in some given $\mathbf{f} \in \mathcal{F}$) are known for a given application. Define the node operator by

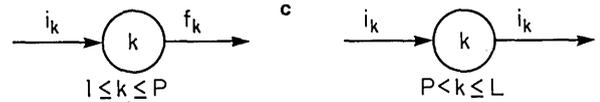
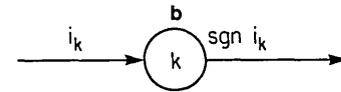
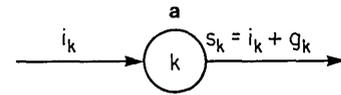


Fig. 1. Three types of nodes used in this paper. The input into the k th node i_k is the sum of the contributions of all L nodes through transmittances t_{ik} . (a) A node useful for linear equation solution and table lookup nets. (b) The node used in Hopfield CAM nets. (c) Nodes useful for our extrapolation net.

$$\begin{aligned} \mathcal{N}\mathbf{i} &= \mathcal{N}[i_1 i_2 \dots i_P : i_{P+1} \dots i_L]^T \\ &= [\delta_1 i_1 \delta_2 \dots \delta_P : i_{P+1} \dots i_L]^T, \end{aligned} \quad (5)$$

where δ_k is the k th element of δ [Fig. 1(c)]. That is, for $1 \leq k \leq P$, the node state is kept at δ_k . Otherwise, the node state is the input sum. The P known states thus act as the input or stimulus to the net and the remaining steady state node states are the response.

In summary, the algorithm is this:

- (1) Initialize with all unknown states set to zero. (If convergence is unique, any initialization will converge to the correct result.) The known states are equated to the known portion of the library vector.
- (2) Multiply the state vector by \mathbf{T} in Eq. (4).
- (3) Replace states 1 through P with their known values.
- (4) Go to step (2) and repeat.

In many cases of interest, we claim that this iterative procedure will converge to the desired library vector. The uniqueness of convergence to the proper library element is addressed in the next section.

IV. Performance Analysis

In this section we derive important convergence properties of the memory extrapolation net and analyze the effects of input uncertainty on the net's performance. Some empirical results on the net's fault tolerance are also discussed.

Insight into the net's performance is gained by viewing the corresponding iterative algorithm in an L dimensional Hilbert space \mathbf{H} . Consider first, the N dimensional subspace \mathbf{T} (also called a closed linear manifold) spanned by the N library vectors (i.e., \mathbf{T} is the closure of \mathcal{F}). The matrix \mathbf{T} in Eq. (4) (orthogonally) projects any vector onto that subspace.¹³ That is, for any $\mathbf{h} \in \mathbf{H}$,

$$\inf\|\mathbf{h} - \mathbf{f}\| = \|\mathbf{h} - \mathbf{T}\mathbf{h}\|$$

$$\mathbf{f} \in T,$$

where $\|\mathbf{a}\|^2 = \mathbf{a}^T\mathbf{a}$. Specifically, note that $T^2 = T$, $TF = F$, and that, for any element \mathbf{b} orthogonal to T , $T\mathbf{b} = \mathbf{0}$ where $\mathbf{0}$ is the zero vector.

To similarly analyze the \mathcal{N} operator in Eq. (5), we adopt the vector partitioning notation

$$\mathbf{h} = \begin{bmatrix} h_p \\ \dots \\ h_q \end{bmatrix},$$

where \mathbf{h}_p is a P and \mathbf{h}_q is a $Q = L - P$ dimensional vector. Then, for example, the zero vector can be written as $\mathbf{0} = [\mathbf{0}_p : \mathbf{0}_q]^T$ and Eq. (5) becomes $\mathcal{N}\mathbf{h} = [\mathbf{f}_p : \mathbf{h}_q]^T$. Note that the operator

$$\mathcal{S}\mathbf{h} = [\mathbf{0}_p : \mathbf{h}_q]^T \quad (6)$$

(orthogonally) projects \mathbf{h} onto the Q dimensional subspace S , spanned by the unit vectors $\epsilon_k = [\mathbf{0}_p : \delta_k]^T$; $1 \leq k \leq Q$, where the vector δ_k is 1 in its k th position and is otherwise zero. Thus, our operator $\mathcal{N}\mathbf{h} = [\mathbf{f}_p : \mathbf{0}_q]^T + \mathcal{S}\mathbf{h}$ projects \mathbf{h} onto the linear variety N which is the translation of S by the vector $[\mathbf{f}_p : \mathbf{0}_q]^T$.

A. Algorithm Convergence

As illustrated in Fig. 2, by alternately projecting between the subspace T and linear variety N , one expects convergence to a point common to both.⁶ Of principal concern is whether our net's iteration,

$$\mathbf{s}_{M+1} = \mathcal{N}\mathbf{T}\mathbf{s}_M, \quad (7)$$

will converge to $\mathbf{f} \in \mathcal{F}$. A sufficient condition for unique convergence is that

$$P \geq N, \quad (8)$$

and the matrix

$$F_p = [\mathbf{f}_{1p} : \mathbf{f}_{2p} : \dots : \mathbf{f}_{Np}] \quad (9)$$

is full rank.

Proof: A fundamental contribution of Youla and Webb⁹ states that alternating projections between two (or more) convex sets converge to a point common to both (all) sets. (A set C is convex if $a\mathbf{a} + (1-a)\mathbf{b} \in C$ for all $\mathbf{a}, \mathbf{b} \in C$ and $0 \leq a \leq 1$.) Since both N (a linear variety) and T (a subspace) are convex, the theorem is applicable here. Furthermore, since both of these sets are linear varieties, convergence is strong.⁹ That is, a vector \mathbf{h} exists in both sets (i.e., $\mathbf{h} \in T$ and $\mathbf{h} \in N$) such that

$$\lim\|\mathbf{s}_M - \mathbf{h}\| = 0,$$

$$M \rightarrow \infty.$$

Clearly, we would like to have $\mathbf{h} = \mathbf{f}$. We can be assured of this if T and N intersect only at a single point. Let us explore this notion. If $\mathbf{h} \in T$, an N dimensional vector \mathbf{a} exists such that $\mathbf{h} = F\mathbf{a}$. Similarly, if $\mathbf{h} \in N$, then $\mathbf{h}_p = \mathbf{f}_p$. Any \mathbf{h} common to both sets must then satisfy

$$F_p\mathbf{a} = \mathbf{f}_p. \quad (10)$$

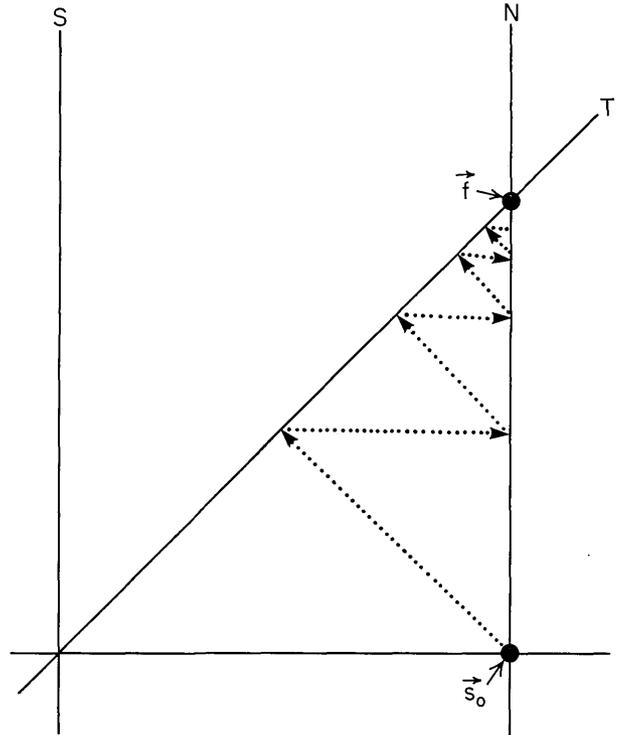


Fig. 2. Illustration of the iterative convergence to the library vector. Beginning with $\mathbf{s}_0 = [\mathbf{f}_p : \mathbf{0}_q]$, we alternately orthogonally project between T and N as shown with the dotted lines. Note that \mathbf{s}_0 is orthogonal to the subspace.

If $P < N$, there is a continuum of solutions. If $P \geq N$, there is at least one solution. If $\mathbf{f} = \mathbf{f}_m$, the solution is $\mathbf{a} = \delta_m$. A sufficient condition for this to be the unique solution is that F_p be full rank.

A more general approach to the question of the degree of subspace intersection, in which our theorem is subsumed, is given by Youla⁷ and by Stark *et al.*⁸

B. Relaxation Parameters

The speed of convergence of the net iteration can be painfully slow. (Consider, for example, when the angle between T and N in Fig. 2 is very small.) One technique to offset this slow convergence is use of relaxation parameters.^{9,14,15} Specifically, we select two constants, λ_T and λ_N , both of which lie on the interval $[0,2]$ and redefine the interconnect and node operators by

$$T_r = (1 - \lambda_T)I + \lambda_T T,$$

$$N_r = (1 - \lambda_N)I + \lambda_N N.$$

The autointerconnects are now $(t_r)_{kk} = \lambda_T(t_{kk} + 1) - 1$ and the remaining interconnects become $(t_r)_{jk} = \lambda_T t_{jk}$; $k \neq j$.

C. Effects of Input Node Operator Error

Consider the perturbed node operator \mathcal{N}_e defined by $\mathcal{N}_e\mathbf{h} = [\mathbf{f}_p + \Delta_p : \mathbf{h}_q]^T$, where Δ_p is a P dimensional error vector corresponding to faulty library information or processor inexactitude. Define $\Delta = [\Delta_p : \mathbf{0}_q]^T$.

If $\Delta \in T$, a perturbed fixed point is clearly at $f + \Delta$. Otherwise, we ask whether the linear variety N_e intersects T . If it does, convergence will be to a common point in each set. If not, we can appeal to a result of Goldburg and Marks¹⁰ who proved that iteration between two nonintersecting finite dimensional convex sets strongly converges to a cycle between two points in each set—each, a closest point in its set to the other convex set. In either case, the fixed point of iteration is not affected by translation of the linear variety in a direction orthogonal to both sets.

D. Fault Tolerance

To obtain an empirical feel for the fault tolerance of the extrapolation net, we used $N = 5$ orthogonal sampled sine wave vectors of length $L = 40$. Each vector had norm $\|f_n\| = \sqrt{20}$. In all cases, we deleted half of a library vector's elements. With only single precision computing error, the mean-square error $e_M = \|s_M - f\|^2$ reduced in ten iterations from $e_0 = 10.5$ to $e_{10} = 0.3$. Quantizing each element of the T matrix to seven quantization levels yielded surprisingly similar results. Doubling the quantization interval resulted in divergence.

A number of simulations were performed wherein a percentage of the elements in T were randomly set to zero. Convergence was strongly dependent on the chosen library vector. Under the scenario above, for example, for 10% of T set to zero, e_{10} typically varied from 0.4 to 0.7. For 20%, $0.7 < e_{10} < 2.8$. A more exhaustive analysis of the fault tolerance is in order.

E. Trade-off of Fault Tolerance with Operations per Iteration

The extrapolation net requires L^2 multiplications per iteration. Note, however, that $R = F^T F$ is a non-negative definite (correlation) matrix, and thus its inverse can be written as $R^{-1} = D^T \Lambda D$, where the diagonal matrix Λ contains the eigenvalues of R^{-1} and D is the corresponding matrix of eigenfunctions. Therefore, Eq. (4) can be written as $T = \Phi \Phi^T$, where

$$\Phi = F D^T \sqrt{\Lambda} \quad (11)$$

is an $L \times N$ matrix. As was done by Marks and Atlas,¹⁶ one iteration can be performed by first, multiplying s_M by Φ^T and second, multiplying this vector result by Φ . Each step costs NL multiplies and, if $N \ll L$, a significant number of multiplies per iteration is saved using this outer product technique at, of course, the loss of fault tolerance and the neural net structure.

V. Table Lookup

An assumption thus far is that any set of P known values in a vector $f \in \mathcal{F}$ can be used to drive the remaining Q nodes. Due to this generality, every node must be connected to every other node. If, on the other hand, the same P nodes are always used as inputs, the number of interconnects can be reduced. Indeed, the states of the P input nodes are not determined by their inputs. Thus, the interconnects to these nodes can be discarded. As we shall see, such

table lookup nets can be reconfigured to $Q < L$ nodes. As with the extrapolation net, the number of operations per iteration can be reduced at the cost of fault tolerance.

A. Table Lookup Net

Again, without loss of generality, assume that the first P elements of some f are our input. Since the first P elements of s_M and f are the same, Eq. (1) can be written as

$$i_M = \begin{bmatrix} i_{M,p} \\ \vdots \\ i_{M,q} \end{bmatrix} = \begin{bmatrix} T_2 : T_1 \\ \vdots \\ T_3 : T_4 \end{bmatrix} \begin{bmatrix} f_p \\ \vdots \\ s_{M,q} \end{bmatrix}, \quad (12)$$

where we have partitioned the T matrix. For the node operator in Eq. (5), we need not be concerned with $i_{M,p}$ since the nodes will transform it to f_p . Thus, the T_1 and T_2 partitions have no contribution to the final result. Such don't-care portions in extrapolation matrices have been noted elsewhere.¹⁷ Setting $s_{M+1,q} = i_{M,q}$, the informational part of Eq. (12) is

$$\begin{aligned} s_{M+1,q} &= [T_3 : T_4] \begin{bmatrix} f_p \\ \vdots \\ s_{M,q} \end{bmatrix} \\ &= g + T_4 s_{M,q} \end{aligned} \quad (13)$$

where $g = T_3 f_p$ can be computed from the library and the memory address f_p . A net for this operation using Q nodes can be formed akin to that discussed in Sec. II. Our interconnect matrix is T_4 and the node operator is defined by $\hat{N}i = i + g$. If the sufficient criteria in (8) and (9) are applicable, $s_{\infty,q} = f_q$ with Q^2 multiplications per iteration. The node used in this net is that in Fig. 1(a).

B. Outer Product Equivalent

The matrix in Eq. (11) can be partitioned as

$$\Phi = \begin{bmatrix} \Phi_p \\ \vdots \\ \Phi_q \end{bmatrix},$$

where Φ_p contains the first P rows of Φ and Φ_q the remaining Q . Then $T_4 = \Phi_q \Phi_q^T$ and Eq. (13) can be written $s_{M+1,q} = g + \Phi_q \Phi_q^T s_{M,q}$. Performing the iteration in this non-net format requires $2NQ$ multiplications per iteration.

VI. Final Remarks

(1) A summary of the operations per iteration for each of the four extrapolation techniques is in Table 1.

(2) The analysis of the extrapolation net drew strongly from results previously derived for signal synthesis and recovery purposes.^{7-12,14,15} In these cases, the equivalent of a library set was chosen either due to a design or constraint motivation rather than for memory purposes. The celebrated Papoulis-Gerchberg algorithm^{7,8,12,14,17-20} (in discrete form), for example, used a similar \mathcal{N} as ours, but chose as a library those vectors whose DFTs were identically zeros in specified bins. The extrapolation net performs this algorithm when the library vectors are the inverse DFTs of the corresponding complementary rows of the DFT ma-

Table I. Multiplications per Iteration for Four Memories

Recall technique	Multiples/iteration
Extrapolation net	L^2
...Outer product technique	$2NL$
Table lookup net	Q^2
...Outer product technique	$2NQ$

Note: For each, there are N library vectors of length L . P elements of one of these elements are used to regenerate the remaining $Q = L - P$. Each memory scheme executes the same restoration algorithm. Thus, in the absence of processor inexactitude, all perform identically.

trix. The continuous form of the Papoulis-Gerchberg algorithm has been performed optically.^{12,21-23}

(3) We have applied the powerful results of convex set projection in our analysis. Any net with a correspondingly convex \mathcal{N} can be similarly analyzed. Also, two or more convex operations can be combined at a node. If, for example, we knew that the library vector's elements were between minus and plus one, the output nodes could perform an additional convex operation which for $P + 1 \leq k \leq L$ is defined by

$$s_k = \begin{cases} 1; & i_k > 1, \\ i_k; & |i_k| \leq 1, \\ -1; & i_k < -1. \end{cases}$$

For $1 \leq k \leq P$, \mathcal{N} is as before. One can view this as a projection onto a (convex) hypercube centered at the origin.

(4) One advantage of the Hopfield CAM net is that a finite number of iterations can result in the exact correct answer, whereas the extrapolation net generally only gets iteratively closer and closer. A step toward a multilevel net, however, can be obtained from the extrapolation net by requiring each library vector to contain only integers. In lieu of Eq. (7), we perform the iteration

$$s_{M+1} = \mathcal{J} \mathcal{N} T s_M, \quad (14)$$

where the vector operator \mathcal{J} rounds each vector element to the nearest integer. Geometrically, \mathcal{J} projects onto the nearest vector with all integer components. Although Eq. (14) generally converges in a finite number of iterations and gets us close to the desired library element, convergence can be to an element not contained in our library. Consider, for example, Fig. 3 where, as in Fig. 2, the subspaces T and N are shown. The lattice of dots denote vectors with integer components. Beginning with the s_0 in the lower right corner, in accordance with Eq. (14), we project onto N and then onto T and finally onto the nearest lattice point. Continuing, we eventually converge to s_∞ shown as the vertex of the steady state (s_∞, b, c) triangle in Fig. 3. Although the process has converged in a finite number of iterations, the result is not our desired f . Note similar steady state triangles (e.g., t in Fig. 3) exist closer to f .

The author gratefully acknowledges the support of this work by the SDIO/IST's Ultra High Speed Com-

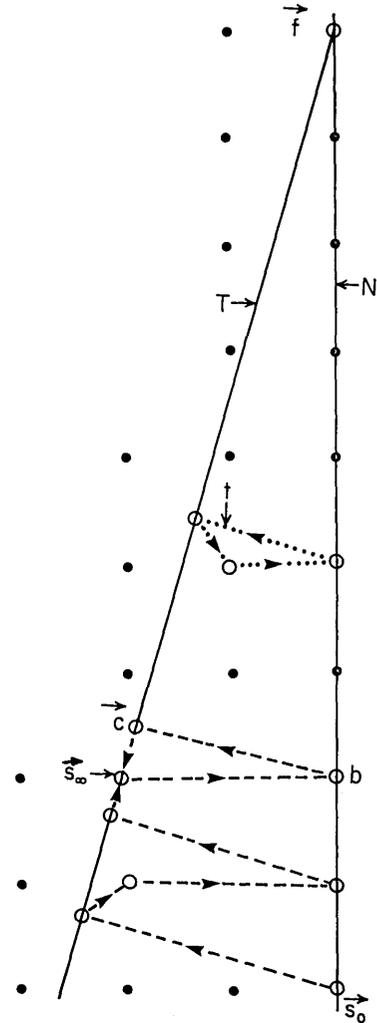


Fig. 3. When rounding the states to the nearest integer, the iteration converges in a finite number of steps—but not to the desired integer vector f .

puting Program administered through the U.S. Office of Naval Research in conjunction with the Optical Systems Laboratory at Texas Tech University and, in part, by the Boeing High Technology Center. Significant contributions to the clarification of result interpretation were made by Dziem Nguyen and Fred Holt at the Boeing High Technology Center. Also appreciated are the stimulating discussions with the author's ISDL colleagues: Les Atlas, Kwan Cheung, and Jim Ritcey.

References

1. J. J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," Proc. Natl. Acad. Sci. U.S.A. 79, 2554 (1982).
2. D. Psaltis and N. Farhat, "Optical Information Processing Based on an Associative-Memory Model of Neural Nets with Thresholding and Feedback," Opt. Lett. 10, 98 (1985).
3. N. Farhat, D. Psaltis, A. Prata, and E. Paek, "Optical Implementation of the Hopfield Model," Appl. Opt. 24, 1469 (1985).

4. D. W. Tank and J. J. Hopfield, "Simple Neural Optimization Networks: an A/D Converter, Signal Decision Circuit and a Linear Programming Circuit," *IEEE Trans. Circuits Syst. CAS-33*, 533 (1986).
5. A. D. Fisher and C. L. Giles, "Optical Adaptive Associative Computer Architectures," in *Proceedings, IEEE 1985 Comcon Spring (IEEE Computer Society Press, 1985)*, pp. 342-344.
6. C. Brown, "Hopfield's Neural Nets Realize Biocomputing," *Electron. Eng. Times* (7 Apr. 1986).
7. D. C. Youla, "Generalized Image Restoration by Method of Alternating Orthogonal Projections," *IEEE Trans. Circuits Syst. CAS-25*, 694 (1978).
8. H. Stark, D. Cahana, and H. Webb, "Restoration of Arbitrary Finite-Energy Optical Objects from Limited Spatial and Spectral Information," *J. Opt. Soc. Am.* **71**, 635 (1981).
9. D. C. Youla and H. Webb, "Image Restoration by the Method of Convex Projections: Part 1—Theory," *IEEE Trans. Med. Imaging MI-1*, 81 (1982).
10. M. Goldburg and R. J. Marks II, "Signal Synthesis in the Presence of an Inconsistent Set of Constraints," *IEEE Trans. Circuits Syst. CAS-32*, 647 (1985).
11. D. C. Youla and V. Velasco, "Extensions of a Result on the Synthesis of Signals in the Presence of Inconsistent Constraints," *IEEE Trans. Circuits Syst. CAS-33*, 465 (1986).
12. R. J. Marks II and D. K. Smith, "Gerchberg-Type Linear Deconvolution and Extrapolation Algorithms," *Proc. Soc. Photo-Opt. Instrum. Eng.* **373**, 161 (1981).
13. G. Strang, *Linear Algebra and Its Applications* (Academic, New York, 1980), p. 116.
14. R. W. Schafer, R. M. Mersereau, and M. A. Richards, "Constrained Iterative Restoration Algorithms," *Proc. IEEE* **69**, 432 (1981).
15. J. R. Fienup, "Reconstruction and Synthesis Applications of an Iterative Algorithm," *Proc. Soc. Photo-Opt. Instrum. Eng.* **373**, 147 (1981).
16. R. J. Marks II and L. E. Atlas, "Content Addressable Memories: A Relation Between Hopfield's Neural Net and an Iterative Matched Filter," submitted for publication.
17. D. Kaplan and R. J. Marks II, "Noise Sensitivity of Interpolation and Extrapolation Matrices," *Appl. Opt.* **21**, 4489 (1982).
18. R. W. Gerchberg, "Super-Resolution Through Error Energy Reduction," *Opt. Acta* **21**, 709 (1974).
19. A. Papoulis, "A New Algorithm in Spectral Analysis and Bandlimited Signal Extrapolation," *IEEE Trans. Circuits Syst. CAS-22*, 735 (1975).
20. A. Papoulis, *Signal Analysis* (McGraw-Hill, New York, 1977), pp. 234-251.
21. T. Sato, S. J. Norton, M. Linzer, O. Ikeda, and M. Hiram, "Tomographic Image Reconstruction from Limited Projections Using Iterative Revisions in Image and Transform Spaces," *Appl. Opt.* **20**, 395 (1981).
22. R. J. Marks II, "Coherent Optical Extrapolation of 2-D Bandlimited Signals: Processor Theory," *Appl. Opt.* **19**, 1670 (1980).
23. R. J. Marks II and D. K. Smith, "An Iterative Coherent Processor for Bandlimited Signal Extrapolation," *Proc. Soc. Photo-Opt. Instrum. Eng.* **231**, 106 (1980).

Pattern continued from page 1997

phase, the heater is lowered to cover only the evaporator, and a shutter is placed in the open ball joint to mask off the ampoule. The charge in the evaporator is warmed until its vapor pressure is $\sim 10^{-1}$ N/m², so that moisture and other volatiles are driven off. The third phase is initiated by removing the shutter, closing the ball joint, and raising the heater so that the top of the free volume in the ampoule is just inside the top of the heater. In this configuration, the charge in the evaporator is melted and heated until the vapor pressure is several hundred N/m², and the distillation process is started. As the distillation proceeds, the heater is moved downward, so that the top of the ampoule cools, thereby causing a crystal of purified material to start growing.

Because the crystal grows down from the top of the ampoule into the heater until it passes the melting-point isotherm, the solid material always has a drop of melt at the bottom end that is held up by surface tension. There is a segregation of impurities as the molten raw material is distilled to this drop and a second segregation as the material in the drop is solidified. When the growing drop reaches a certain size, it falls back into the evaporator, carrying with it impurities rejected in the solidification process. In this way, the solid growth face is constantly washed by fresh molten distillate. Thus the material is effectively purified by multiple zone refining and by distillation.

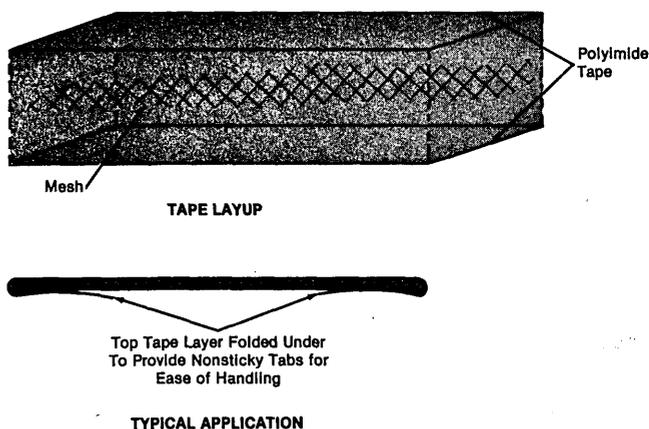
This work was done by Lawrence R. Holland of the University of Alabama, Huntsville, for Marshall Space Flight Center. In accordance with Public Law 96-517, the contractor has elected to retain title to this invention. Inquiries concerning rights for its commercial use should be addressed to: University of Alabama, Huntsville, AL 35899. Refer to MFS-26004.

Fig. 7. Strong tape for low temperatures consists of two layers of polyimide tape with a layer of reinforcing mesh.

Strong adhesive tape for cold environments

An improved tape devised for repairs in space may also find use on earth in polar regions and in superconducting applications. The tape retains its adherence and strength at extreme temperatures, where conventional tapes would fail. Experience on the Space Shuttle had shown that ordinary duct tape loses most, if not all, of its stickiness at the orbital temperatures. Kapton (or equivalent) polyimide tape, for example, maintained adequate adhesion but had to be layered for sufficient strength to resist inadvertent tearing by crew members. Multiple layers, however, make the tape stiffer. The improved tape (see Fig. 7) consists of two layers of the polyimide tape with a reinforcing intermediate layer of thin, open-weave Kevlar (or equivalent) aromatic polyamid. Other mesh materials may also be suitable.

This work was done by Thomas G. Woods of McDonnell Douglas Corp. for Johnson Space Center. Refer to MSC-20924.



continued on page 2015