OPTICAL IMPLEMENTATIONS OF THE ALTERNATING
PROJECTION NEURAL NETWORK

by

ALAN TROWBRIDGE SMITH, B.S. in E.E.

A THESIS

IN

ELECTRICAL ENGINEERING

Submitted to the Graduate Faculty
of Texas Tech University in
Partial Fulfillment of
the Requirements for
the Degree of

MASTER OF SCIENCE

IN

ELECTRICAL ENGINEERING

000000

805

T3

1989

No. 98.

cop. 2

ACKNOWLEDGEMENTS

There are many people who have spent large amounts of time and money to help me obtain a Master's Degree. I would like to thank first my family members for their love and encouragement. Also, I owe Dr. John Walkup many thanks for letting me study under him and for his support during the past six months. Partial support of this research by the Office of Naval Research is gratefully appreciated. I am grateful to Dr. Robert J. Marks II for sharing with me his expertise in the subject of neural networks. Finally, I would like to thank Dr. Sunanda Mitra for convincing me that I could survive and do well in graduate school.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS.....	ii
ABSTRACT.....	vi
LIST OF FIGURES.....	vii
CHAPTER	
I. INTRODUCTION.....	1
II. AN ANALYTICAL STUDY OF THE APNN.....	8
2.1 The Interconnection Matrix.....	8
2.2 Two Computer Examples.....	11
2.3 Convergence Dynamics.....	15
2.3.1 Notation.....	15
2.3.2 The Steady State Solution.....	17
2.3.3 The Eigenstructure of T.....	18
2.4 Geometrical Analysis.....	19
2.4.1 One Stored Library Vector.....	21
2.4.2 More Than One Library Vector.....	23
III. ELECTRICAL IMPLEMENTATION.....	28
3.1 Electronic Modeling of the APNN Neurons...	28
3.2 Potential Problems.....	34
3.3 Grey Level Resolution.....	35
IV. PASSIVE OPTICAL IMPLEMENTATION.....	37
4.1 Optical Matrix-Vector Multiplication.....	38
4.2 Film as a Spatial Light Modulator.....	40
4.3 The Matrix-Vector Multiplier.....	43
4.3.1 Choosing the Library Matrix.....	43
4.3.2 Choosing the Optical Waveguides.....	44
4.3.3 Building the Matrix Mask.....	45

4.4	LED Performance.....	47
4.5	The Detector Array.....	47
4.6	System Integration.....	52
4.6.1	Crosstalk Problems.....	52
4.6.2	Detector Noise Problems.....	54
4.7	System Strengths and Limitations.....	56
4.7.1	Spatial Resolution.....	56
4.7.2	Grey Level Resolution.....	57
4.7.3	Speed of Operation.....	59
4.7.4	Programmability.....	60
V.	ACTIVE OPTICAL IMPLEMENTATION.....	61
5.1	The Hughes Liquid Crystal Light Valve.....	63
5.1.1	LCLV Components.....	63
5.1.2	LCLV Operation.....	65
5.2	The Semetex SIGHT-MOD.....	68
5.3	Improved Detector Array.....	69
5.4	System Integration.....	73
5.4.1	Focusing Optics.....	73
5.4.2	LCLV Linear Operating Range.....	74
5.5	Example With Binary Library Vectors.....	76
5.5.1	Encoding the Interconnection Matrix.....	77
5.5.2	Convergence.....	80
5.6	Example With Continuous-Valued Library Vectors.....	83
5.6.1	Encoding the Interconnection Matrix.....	83
5.6.2	Convergence.....	84
5.7	System Strengths and Limitations	86
5.7.1	Spatial Resolution.....	88
5.7.2	Grey Level Resolution.....	88
5.7.3	Speed of Operation.....	89
5.7.4	Cost.....	90

VI.	CONCLUSIONS.....	91
REFERENCES.....		96
APPENDIX.....		98

ABSTRACT

An analysis of the alternating projection neural network (APNN) along with results from an electronic and two optical implementations are presented in this thesis. After an introduction the APNN is explored both analytically and geometrically. Characteristics such as the speed of convergence are realized through matrix equations and linear algebra. If convergence of the APNN is visualized as a geometrical process based on projection, complementary principles are established, such as how the convergence rate changes as new data is stored in the network. Finally, experimental results from two optical architectures are discussed; both configurations use optical matrix-vector multipliers with feedback, but different spatial light modulators are used.

LIST OF FIGURES

Figure		Page
1.1	Alternating Orthogonal Projections Between Two Convex Sets T and H	3
1.2	Geometrical Operation of the APNN.....	5
2.1	Projecting s in L Space to Derive T.....	9
2.2	Convergence After 5 Iterations.....	12
2.3	Convergence After M Iterations (M = 0,3,30,45).....	13
2.4	Detailed Geometrical Operation of the APNN...	20
2.5	Convergence Rate of the APNN as a Function of M.....	27
3.1	APNN Neurons With Interconnection Weights....	30
3.2	Reduced APNN Network With Interconnection Weights.....	31
3.3	Electrical Circuit for the Third Neuron.....	31
3.4	Oscilloscope Photograph Showing the Convergence of the Third Neuron.....	33
4.1	APNN System Configuration.....	39
4.2	Continuous-Valued Intensity Transmittances...	42
4.3	Intensity Transmittance by Area Modulation...	42
4.4	Pictorial of Mask Pattern.....	46
4.5	LED Characteristics.....	48
4.6	Detector Circuit.....	49
4.7	Areas of Potential Crosstalk.....	53
4.8	Integrated System With Feedback for One Neuron.....	55

5.1	Top View of the Active APNN System Configuration.....	62
5.2	Schematic of the LCLV.....	64
5.3	Twisted Nematic Effect.....	66
5.4	LCLV Off-State.....	67
5.5	LCLV On-State.....	67
5.6	Operation of the SIGHT-MOD.....	70
5.7	A Linear Amplifier.....	72
5.8	FET Current Amplifier.....	72
5.9	System Linearity with SIGHT-MOD 100% Transmissive.....	75
5.10	Intensity Matrix Incident on the LCLV Write Side.....	78
5.11	Oscilloscope Photograph Showing Convergence of the Binary APNN.....	81
5.12	SIGHT-MOD Modulating Characteristic Curve....	85
5.13	Oscilloscope Photograph Showing the Convergence of the Continuous-Valued APNN.....	87

CHAPTER I

INTRODUCTION

The aggregation of research dedicated to the development of neural networks is diverse. Potential applications range from pattern recognition to archival storage and from weather prediction to credit approval. Optimists even predict that neural networks will be the heart of the next generation of high speed computers. Due to their preeminent advantages over sequential computers [1], neural networks will undoubtedly find a place in our increasingly technical world.

The development of neural networks was initiated by biologists studying the mechanics of the brain [2]. They discovered that the human brain can quickly perform many tasks that are almost impossible for the traditional Von Neumann Computer because of the parallelism and high neuron interconnectivity inherent in neural networks. At that point engineers became interested in developing a neural network processor that would operate faster than the sequential computer.

Three fundamental components of neural networks are the neurons, the interconnection matrix, and the state vector. Neurons are discrete, highly interconnected operating units. Their function is to perform a weighted sum of their inputs and pass the result through a specified nonlinearity. The interconnection matrix T defines the manner in which neurons

are linked; element t_{ij} defines the interconnection weight between the i^{th} and j^{th} neurons. If operation of a neural network is considered an iterative, sequential process, then $\mathbf{s}(M)$ denotes the state of all neurons at iteration M . The state of the k^{th} neuron at the next iteration can be found by

$$s_k(M+1) = \eta_k \left[\sum_{i=1}^L t_{ik} s_i(M) \right] \quad (1.1)$$

where η_k is the nonlinearity applied to the sum and L is the number of neurons.

Many artificial neural networks have been developed; each type has its own distinctive qualities. The alternating projection neural network (APNN) is a new class of networks developed by Professor Robert J. Marks II of the University of Washington [3]. As the name implies, the APNN performs by alternately projecting between two constraint sets. Convergence is assured if the sets are convex [4]. A set of vectors, $\mathbf{C}$, is convex if, for all vectors $\mathbf{x}$ and $\mathbf{y}$ in $\mathbf{C}$, the line segment connecting $\mathbf{x}$ and $\mathbf{y}$ is contained in $\mathbf{C}$. Convexity is described mathematically as

$$(1-\alpha)\mathbf{x} + \alpha\mathbf{y} \in \mathbf{C} \quad ; \quad 0 \leq \alpha \leq 1. \quad (1.2)$$

The usefulness of projection theory is illustrated in Figure 1.1. Given an arbitrary vector $\mathbf{s}$ and two convex sets $\mathbf{T}$ and $\mathbf{H}$, alternate orthogonal projections onto these sets asymptotically approach a point common to both. Since only

Figure 1.1. Alternating Orthogonal Projections
Between Two Convex Sets T and H .

two convex sets are projected upon, von Neumann's Alternating Projection Theorem provides the necessary mathematical foundation to describe the APNN's operational dynamics [5]. Given two subspaces **S** and **T** in a general Hilbert space, the orthogonal projection onto the intersection of these subspaces is described by **TS**. In terms of the APNN, the product corresponds to a matrix-vector multiplication, where **s** is the neural network state vector, and **T** is the interconnection matrix.

The convergence process is geometrically represented for a two dimensional (two neuron) case in Figure 1.2. The abscissa and ordinate represent all possible values (states) of the first and second neurons, respectively. The two convex sets are the subset **T** representing the interconnection matrix and the linear variety (translated subspace) $\mathbf{s}_1 = \boldsymbol{\eta}$. Alternate orthogonal projections between these two subspaces approach the point (η, ξ) .

The goal of these projections is to extrapolate ξ ; the APNN is a content addressable memory. η and ξ will hereafter be referred to as the clamped (known) and floating (unknown) components, respectively. Physically, the floating neurons represent the noise or uncertainty in the system. The input stimulus to the network is

Figure 1.2. Geometrical Operation of the APNN.

$$\mathbf{s}(0) = \begin{bmatrix} \eta \\ 0 \end{bmatrix} . \quad (1.3)$$

By von Neumann's theorem, the projection onto $\mathbf{T}$ is accomplished by the multiplication $\mathbf{T}\mathbf{s}(0)$. Forcing the first element of the matrix-vector product back to η performs the projection onto the linear variety and completes the first iteration. A generalization of this process is given by

$$\mathbf{s}(M+1) = \underline{\eta} \mathbf{T} \mathbf{s}(M) \quad (1.4)$$

where $\underline{\eta}$ is the second projection or clamping operator and $\mathbf{s}(M)$ denotes the state of the network at the M^{th} iteration.

Convergence is realized when

$$\mathbf{s}(\infty) = \begin{bmatrix} \eta \\ \xi \end{bmatrix} . \quad (1.5)$$

In practice, of course, a decision threshold must be set for reaching ξ .

The two-neuron network represented by Figure 1.2 will be extended to L dimensions (L neurons) in Chapter II. The stimulus vector will consist of a clamped sub-vector η and a floating sub-vector ξ .

Perhaps the greatest attribute of the APNN is the mathematical completeness with which it can be described. In most cases the network's behavior can be predicted with great rigor using linear algebra and geometry-based insight.

Unfortunately, one pays for this bonus with a relatively low storage capacity and an extensive interconnection pattern. In some ways the APNN is similiar to the linear Hopfield model because the capacity is less than the number of neurons and the interconnection schemes are the same. As with the perceptron model, hidden layers can be added to improve the storage capacity [3]. The APNN can be configured as a content addressable memory or as a classifier. However, the APNN is not a back propagation neural network. The convergence rate is determined solely by the data base, not by the complexity of the decision boundaries. Overall, the APNN is in a class by itself. Hence standard statistical comparisons of performance characteristics between the APNN and other networks do not yield insight into its operational dynamics.

In Chapter II, a detailed analysis of the interconnection matrix with examples is presented to describe convergence. An electrical model is described in Chapter III with the optical architectures appearing in Chapters IV and V. Finally, in Chapter VI, a summary of the results of the thesis is given along with the APNN's application limitations.

CHAPTER II

AN ANALYICAL STUDY OF THE APNN

2.1 The Interconnection Matrix

Suppose $\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_N$ are linearly independent library vectors stored in the APNN in L dimensional space where we assume $N < L$ [6]. Their linear combinations $x_1\mathbf{f}_1 + x_2\mathbf{f}_2 + \dots + x_N\mathbf{f}_N$ fill out a subspace $\mathbf{U}$, which is the space spanned by the vectors. If one forms a matrix

$$\mathbf{F} = [\mathbf{f}_1 : \mathbf{f}_2 : \dots : \mathbf{f}_N], \quad (2.1)$$

the linear combination of the $\mathbf{f}_i$ is the column space of $\mathbf{F}$.

Similiarly, a subspace $\mathbf{V}$ orthogonal to $\mathbf{U}$ can be obtained from another set of linearly independent library vectors $\mathbf{y}_i$ that are orthogonal to the $\mathbf{f}_i$'s. In general, $\mathbf{V}$ will have dimension $L-N$.

Now consider any vector $\mathbf{s}$ in L dimensional space. Figure 2.1 shows $\mathbf{U}$ and $\mathbf{V}$ as perpendicular lines in a plane. Here the subspace $\mathbf{U}$ consists of all multiples of $\begin{bmatrix} \eta \\ \xi \end{bmatrix}$ and the subspace $\mathbf{V}$ consists of all vectors with

$$\begin{bmatrix} \eta & \xi \end{bmatrix} \begin{bmatrix} y^1 \\ y^2 \end{bmatrix} = 0 \quad (2.2)$$

The projection of $\mathbf{s}$ onto $\mathbf{U}$ is $\mathbf{F}\mathbf{x}$ and the projection of $\mathbf{s}$ onto $\mathbf{V}$ is $\mathbf{y}$. From the figure,

$$\mathbf{F}\mathbf{x} + \mathbf{y} = \mathbf{s}. \quad (2.3)$$

Figure 2.1. Projecting $\mathbf{s}$ in L Space to Derive $\mathbf{T}$.

Also,

$$F^T \mathbf{y} = 0 \quad (2.4)$$

since $\mathbf{y}$ is perpendicular to the $\mathbf{f}_i$. Combining the two equations gives the normal equation

$$\mathbf{x} = (F^T F)^{-1} F^T \mathbf{s} . \quad (2.5)$$

Therefore $F\mathbf{x}$, the projection of $\mathbf{s}$ onto U , is

$$F\mathbf{x} = F(F^T F)^{-1} F^T \mathbf{s} = T\mathbf{s} . \quad (2.6)$$

The key result is that

$$T = F(F^T F)^{-1} F^T . \quad (2.7)$$

Geometrically, $F\mathbf{x}$ is the point on U closest to $\mathbf{s}$.

Algebraically, $F\mathbf{x}$ is obtained by multiplying the state vector $\mathbf{s}$ by the $L \times L$ projection matrix T of Equation 2.7.

Some properties of T include:

- (1) $T^T = T$ and $T^2 = T$;
- (2) T has real elements ;
- (3) Rank of T = number of library vectors $\mathbf{f}_i$;
- (4) The eigenvalues of T are either zero or one .

Calculating the interconnection matrix can be computationally expensive and inflexible when wishing to sequentially "teach" the APNN new vectors. An improved iterative method for calculating T based on Gram-Schmidt orthonormalization is discussed elsewhere [5].

2.2 Two Computer Examples

Consider the cross in Figure 2.2 which has grey level integer values ranging from 0 (white) to 4 (black) and consists of 36 blocks, each representing a pixel. The goal is to recover the image from a noisy version after it has been stored in the APNN in accordance with Equations 2.1, 2.7, and 1.4. Since we are only storing the cross, the matrix F is a 36 element vector containing integers that range from 0 to 4.

First, let the floating neurons correspond to pixels with grey level values 0,1, and 2 and the clamped neurons correspond to the pixels with values 3 and 4. Without loss of generality, all floating neurons in the stimulus image are assumed to initially have value 0. The computer program simulating the APNN, which is listed in the Appendix, calculates the state vector $\mathbf{s}(M)$ using Equation 1.4. After each iteration the continuous-valued elements of the state vector are rounded-off and compared to the elements of F . When these numbers match, the APNN has converged and the image has been extrapolated from the noisy representation. The rounding threshold is a parameter that controls the accuracy of $\mathbf{s}(M)$ with respect to the original. In these examples, 0.5 or higher is rounded to 1.0. After only five iterations the APNN recovered the cross within this round-off range.

In the second example of Figure 2.3, the floating

Original

Input M=0

M=5

Figure 2.2. Convergence After 5 Iterations.

Figure 2.3. Convergence After M Iterations ($M = 0, 3, 30, 45$).

neurons correspond to those pixels having grey levels 2,3, and 4, and the remaining pixels were clamped. Again, without loss of generality, the floating neurons initially had a value of 0. Notice that 45 iterations were needed to recover the original cross. This result is intuitively pleasing since the floating neurons have larger values than the clamped neurons. The APNN must work harder in this example.

Geometrically, the slope of the plane T (see Figure 1.2) is larger, and each iteration does not gain as much distance on the ordinate axis. An in-depth analysis of the APNN geometry will be given in Section 2.4.

With the examples just presented, the practical limitations of this neural network become apparent. First, we have assumed that a sufficient part of the data to be recalled is known exactly. This data defines the clamped and floating partition of neurons. In a real-world application, as samples are taken, there will always be uncertainty or noise in the data. Consequently, the APNN must generalize from data outside of the library vector set. Even if the network converges, the result will be an interpolation between library vectors in F . More likely, the APNN state vector $\mathbf{s}$ will oscillate between several solutions; the range of oscillations will depend on how close (numerically and geometrically) the library vectors are to each other. Only in a few limited applications, such as one iteration convergence, will a solution be guaranteed [5]. The manner in

which the APNN generalizes is not yet fully understood and is a good subject for further study.

Second, these examples cannot differentiate between images with the same histogram. This problem can be avoided, however, if somehow components can be added to $\mathbf{f}_i$ to "encode" the i^{th} vector and ensure its uniqueness. These components must arise from a quality that can be sampled and given to the APNN as data; otherwise they are useless.

2.3 Convergence Dynamics

We will now step through the important convergence equations previously derived [3] that will lead to the geometrical study.

2.3.1 Notation

The notation established in this section is useful in proofs and in the derivation of equations. Without loss of generality, let the first P elements of the state vector correspond to the clamped elements of the i^{th} library vector $\mathbf{f}_i$. Thus the vector partitioning notation

$$\mathbf{s}(M) = \begin{bmatrix} \mathbf{f}_i^P \\ \mathbf{s}^Q(M) \end{bmatrix} \quad (2.8)$$

is adopted, where $Q = L - P$ is the number of floating neurons. Similarly, the matrix consisting of the first P rows of F is defined as F_p , where

$$F_P = [\mathbf{f}_1^P : \mathbf{f}_2^P : \dots : \mathbf{f}_N^P] . \quad (2.9)$$

Finally, the interconnection matrix is partitioned as

$$T = \begin{bmatrix} T_2 & T_1 \\ T_3 & T_4 \end{bmatrix} \quad (2.10)$$

where T_2 is $P \times P$ and T_4 is $Q \times Q$. The partitioning scheme is motivated by quadrant location.

Convergence, in a mathematical sense, means that

$$\lim_{M \rightarrow \infty} \|\mathbf{s}(M) - \mathbf{f}_i\| = 0 , \quad (2.11)$$

or that the i^{th} library vector has been restored. The only requirement is that F_P be full rank. This important criterion can be justified with a quick example. Consider the four neuron system used to recall two vectors with

$$F = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} . \quad (2.12)$$

For the case $P=2$, F_P is singular and the APNN cannot "look up" the desired library vector. If $P=3$, however, the network can recover the value of the fourth element given the values of the first three.

Embedded in this criterion on F_P is the condition $P \geq N$, where N is the number of library vectors stored in the network. Thus the capacity limit of the APNN is ascertained. The maximum number of vectors that can be stored is equal to the number of clamped neurons. In practice, this condition is vague because the number of clamped neurons may change from

application to application. If the APNN is to be used as a content addressable memory, the number of stored images should be significantly less than the fewest number of expected clamped pixels. This limit can be raised, however, by adding pseudo-clamped neurons in "hidden layers" [3].

2.3.2 The Steady State Solution

The steady state solution for the APNN can be explicitly derived. Equation 1.3 and its partitioned form are written as

$$\mathbf{s}(M+1) = \eta \mathbf{T} \mathbf{s}(M) \quad (1.3)$$

and

$$\begin{bmatrix} \mathbf{f}_1^P \\ \mathbf{s}^Q(M+1) \end{bmatrix} = \eta \begin{bmatrix} T_2 & T_1 \\ T_3 & T_4 \end{bmatrix} \begin{bmatrix} \mathbf{f}_1^P \\ \mathbf{s}^Q(M) \end{bmatrix} . \quad (2.13)$$

The first P elements of the state vector are clamped; their states are not effected by a weighted input sum. As a result, Equation 2.13 can be reduced to

$$\mathbf{s}^Q(M+1) = T_3 \mathbf{f}_1^P + T_4 \mathbf{s}^Q(M) . \quad (2.14)$$

Letting $M \rightarrow \infty$, the solution becomes

$$\mathbf{s}^Q(\infty) = (\mathbf{I} - T_4)^{-1} T_3 \mathbf{f}_1^P \quad (2.15)$$

where $\mathbf{I}$ is the identity matrix. All of the eigenvalues of T_4 are between zero and one [2]. If F_P is full rank, the vector $\mathbf{s}^Q(\infty)$ represents the converged states of the floating neurons.

A point about the convergence solution should be made

before proceeding further. Equation 2.15 can be used in place of the APNN only if the clamped and floating status of each neuron remains unchanged from application to application.

2.3.3 The Eigenstructure of T

The results derived in this section will serve as the foundation for the geometrical study of the APNN in Section 2.4. Let $\{\mathbf{p}_r \mid 1 \leq r \leq Q\}$ denote the normalized eigenvectors and $\{\lambda_r\}$ denote the eigenvalues of the T_4 matrix. Also, define the matrix

$$P = [\mathbf{p}_1 | \mathbf{p}_2 | \dots | \mathbf{p}_Q] . \quad (2.16)$$

Since the $\mathbf{p}_r$'s are linearly independent and T_4 is symmetric, T_4 can be diagonalized into

$$\Lambda_4 = P^T T_4 P \quad (2.17)$$

where

$$\text{diag } \Lambda_4 = [\lambda_1 \lambda_2 \dots \lambda_Q]^T \quad (2.18)$$

and $P^T = P^{-1}$ is an orthogonal matrix. Define

$$\mathbf{x}(M) = P^T \mathbf{s}^Q(M) \quad (2.19)$$

where $\mathbf{s}^Q(M)$ is the state vector representing the Q floating neurons. From Equation 2.14,

$$P \mathbf{x}(M+1) = T_3 \mathbf{f}_i^P + T_4 P \mathbf{x}(M) \quad (2.20)$$

and

$$\mathbf{x}(M+1) = P^T T_3 \mathbf{f}_i^P + P^T T_4 P \mathbf{x}(M)$$

$$= \mathbf{g} + \Lambda_4 \mathbf{x}(M) \quad (2.21)$$

where

$$\mathbf{g} = \mathbf{P}^T \mathbf{T}_3 \mathbf{f}_1 \mathbf{P} \quad (2.22)$$

is constant once the floating and clamped partition is established. The steady state solution can be found by taking the limit as $M \rightarrow \infty$, leading to

$$\mathbf{x}(\infty) = (\mathbf{I} - \Lambda_4)^{-1} \mathbf{g} . \quad (2.23)$$

Equation 2.21 can also be solved, yielding

$$x_k(M+1) = \sum_{r=0}^M \lambda_k^r g_k , \quad (2.24)$$

where x_k is the k^{th} element of $\mathbf{x}$. With this foundation we are ready to proceed and study the geometry of the APNN.

2.4 Geometrical Analysis

In this section intuitively pleasing results are derived when the APNN geometry is linked to the eigenstructure of $\mathbf{T}$. Figure 2.4 shows the process in a little more detail than does Figure 1.2. Specifically, an angle θ and distance ξ_k are added. In the general case of $L > 2$, η , ξ , and ξ_k are replaced with η , ξ , and $|\xi_k|$. We shall keep in mind that any result derived in the two-dimensional case can automatically be extended to L dimensions.

Figure 2.4. Detailed Geometrical Operation of the APNN.

2.4.1 One Stored Library Vector

Since an infinite number of iterations are needed to completely determine the exemplar ξ ,

$$\xi = \sum_{k=1}^{\infty} \xi_k \quad (2.25)$$

Using trigonometry, one can obtain the significant equivalence

$$\frac{\xi_{k+1}}{\xi_k} = \sin^2 \theta \quad (2.26)$$

If Equation 2.26 is thought of as the APNN "transfer function," then the net energy gained from the next iteration can be found a priori given the angle θ . This ratio, which is constant, is the fundamental measure for the APNN's speed of convergence. Thus, if θ is known, a measure of the APNN's behavior during all iterations can be found by completing the first iteration only. One might conclude that the matrix-vector multiplication and clamping operations are useless. However, we should note that for $L > 2$, $|\xi_k|$ only gives the net energy (or length) of the floating components of the state vector, not the individual component values.

Equation 2.26 will hold true for any number of neurons as long as the $L \times L$ T matrix has a rank of one. Geometrically, T is in L space but has a dimension of one. A generalization

of the "transfer function" for the dimension of T being greater than one is derived later in this section.

The full value of this geometrical analysis is realized when θ is linked to the eigenstructure of T. Let the single column F matrix be written as

$$F = \begin{bmatrix} \eta \\ \xi \end{bmatrix} \quad (2.27)$$

where η and ξ represent the clamped and floating sub-vectors, respectively. The interconnection matrix can be calculated from Equation 2.7 as

$$T = \frac{1}{|\eta|^2 + |\xi|^2} \begin{bmatrix} |\eta|^2 & |\eta||\xi| \\ |\eta||\xi| & |\xi|^2 \end{bmatrix} \quad (2.28)$$

and

$$T_4 = \frac{|\xi|^2}{|\xi|^2 + |\eta|^2} = \lambda_{SR} \quad (2.29)$$

where λ_{SR} is the spectral radius, the largest eigenvalue.

Obviously,

$$\lambda_{SR} = \sin^2\theta \quad (2.30)$$

and the angle θ has been linked to the eigenstructure of T.

For fast convergence, we want

$$\xi_{k+1} \ll \xi_k \quad (2.31)$$

so that the next iteration does not give an appreciable

amount of new information compared to the last iteration.

Since

$$\left| \frac{\xi_{k+1}}{\xi_k} \right| = \frac{|\xi|^2}{|\eta|^2 + |\xi|^2}, \quad (2.32)$$

the condition of Equation 2.31 demands that

$$|\xi| \ll |\eta|. \quad (2.33)$$

This result strengthens our foreknowledge that the more a priori information we give the APNN, the higher its convergence rate will be.

2.4.2 More Than One Library Vector

The APNN can hardly be considered useful unless it can discriminate between more than one library vector. Thus, a generalized APNN "transfer function" similar to Equation 2.11 is developed in this section. Since the T matrix will be up to P dimensional, where P is the number of clamped neurons, geometry fails us and we must return to using the tool of linear algebra.

The primary purpose of Section 2.3 was to obtain the equation

$$x_k(M+1) = \sum_{r=0}^M \lambda_k^r g_k \quad (2.24)$$

where k denotes particular components of the state vector $x(M+1)$, diagonal matrix Λ , and constant vector g . Remember

that the state vector represents the state of the floating neurons and has $Q=L-P$ components. Rewriting Equation 2.8 leads to

$$\mathbf{x}(M+1) = \begin{bmatrix} \sum_{r=0}^M \lambda_1^r g_1 \\ \vdots \\ \sum_{r=0}^M \lambda_Q^r g_Q \end{bmatrix} . \quad (2.34)$$

From Figure 2.4,

$$\xi_k = \mathbf{x}(M+1) - \mathbf{x}(M)$$

$$= \begin{bmatrix} \sum_{r=0}^M \lambda_1^r g_1 - \sum_{r=0}^{M-1} \lambda_1^r g_1 \\ \vdots \\ \sum_{r=0}^M \lambda_Q^r g_Q - \sum_{r=0}^{M-1} \lambda_Q^r g_Q \end{bmatrix} = \begin{bmatrix} \lambda_1^M g_1 \\ \vdots \\ \lambda_Q^M g_Q \end{bmatrix} . \quad (2.35)$$

Thus

$$|\xi_k| = [g_1^2 \lambda_1^{2M} + g_2^2 \lambda_2^{2M} + \dots + g_Q^2 \lambda_Q^{2M}]^{1/2} \quad (2.36)$$

and

$$|\xi_{k+1}| = [g_1^2 \lambda_1^{2M+2} + g_2^2 \lambda_2^{2M+2} + \dots + g_Q^2 \lambda_Q^{2M+2}]^{1/2} \quad (2.37)$$

The APNN "transfer function" immediately follows as

$$\left| \frac{\xi_{k+1}}{\xi_k} \right| = \left| \frac{g_1^2 \lambda_1^{2M+2} + \dots + g_Q^2 \lambda_Q^{2M+2}}{g_1^2 \lambda_1^{2M} + \dots + g_Q^2 \lambda_Q^{2M}} \right|^{1/2} \quad (2.38)$$

For the case of one stored vector, all of the λ_i 's are zero except for one, the spectral radius. Thus

$$\left| \frac{\xi_{k+1}}{\xi_k} \right| = \lambda_{SR} \quad (2.39)$$

as before. For the general case of at most P stored vectors, Equation 2.9 can be written as

$$\left| \frac{\xi_{k+1}}{\xi_k} \right| = \left| \frac{\lambda_1^2}{1 + \dots + \left(\frac{g_P}{g_1} \right)^2 \left(\frac{\lambda_P}{\lambda_1} \right)^{2M}} + \dots + \frac{\lambda_P^2}{1 + \dots + \left(\frac{g_1}{g_P} \right)^2 \left(\frac{\lambda_1}{\lambda_P} \right)^{2M}} \right|^{1/2} \quad (2.40)$$

If, without loss of generality, we define λ_p to be the spectral radius such that

$$\lambda_p > \lambda_i ; i \neq p, \quad (2.41)$$

then

$$\lim_{M \rightarrow \infty} \left(\frac{\lambda_p}{\lambda_i} \right)^{2M} = \infty \quad (2.42)$$

and

$$\lim_{M \rightarrow \infty} \left(\frac{\lambda_i}{\lambda_p} \right)^{2M} = 0 \quad (2.43)$$

The most important result of the thesis follows:

$$\lim_{M \rightarrow \infty} \left| \frac{\xi_{k+1}}{\xi_k} \right| = \lambda_p = \lambda_{SR} \quad (2.44)$$

Since the APNN "transfer function" describes the rate of convergence, Equation 2.44 implies that the rate of convergence can never be better than the magnitude of the spectral radius. Keeping this eigenvalue small is the fundamental method for ensuring quick APNN performance.

An intuitively pleasing observation can be made from Equations 2.40 and 2.44. Figure 2.5 illustrates how the convergence rate starts out quickly and approaches the

spectral radius. Since we want $\left| \frac{\xi_{k+1}}{\xi_k} \right|$ to be small, the convergence rate gets worse as $M \rightarrow \infty$ despite the fact that each iteration brings us new information. The APNN's efficiency decreases as the number of iterations increases. This concept is consistent with the computer examples of Section 2.2. In the second example, the first four grey levels 0, 1, 2, and 3 are extrapolated after 30 iterations. To obtain the black level 4, 15 more iterations are needed. If the APNN is to be used to recall a typical photograph, for example, areas of the photo with low spatial frequencies will be extrapolated first, and the high frequency edge information will be recalled later. Many image processing applications come to mind, such as spatial averaging, filtering, and image subtraction.

Figure 2.5. Convergence Rate of the APNN as a Function of M .

CHAPTER III

ELECTRICAL IMPLEMENTATION

Before introducing the optical portion of this thesis, an APNN consisting of operational amplifiers and resistors is presented. The electrical model has several undeniable advantages over its optical counterparts.

3.1 Electronic Modeling of the APNN Neurons

Consider the library matrix

$$F = \begin{bmatrix} 0 & 1 \\ 3 & 3 \\ 0 & 2 \\ 1 & 1 \end{bmatrix} \quad (3.1)$$

and consequently the interconnection matrix

$$T = \begin{bmatrix} 0.2 & 0.0 & 0.4 & 0.0 \\ 0.0 & 0.9 & 0.0 & 0.3 \\ 0.4 & 0.0 & 0.8 & 0.0 \\ 0.0 & 0.3 & 0.0 & 0.1 \end{bmatrix} \quad (3.2)$$

We chose to challenge the electronic system by storing in it as many grey levels as neurons. If the APNN is viewed at the neuron level, the analog to Equation 1.4 is

$$s_k(M+1) = \eta_k \left[\sum_{i=1}^L t_{ik} s_i(M) \right] \quad (1.1)$$

where s_k is the state of the k^{th} neuron, η_k is the clamping operator acting on the k^{th} neuron, t_{ik} is the corresponding

element of the interconnection matrix, and L is the number of neurons.

Since, like all neural networks, the APNN executes a weighted sum, using summing operational amplifiers is the logical approach to building the machine. Figure 3.1 shows the network consisting of four neurons with the appropriate feedback weights defined by the interconnection matrix. We will let the first two neurons be clamped and initialize the floating neurons at zero such that

$$\mathbf{s}(0) = \begin{bmatrix} 1 \\ 3 \\ 0 \\ 0 \end{bmatrix}. \quad (3.3)$$

The goal for the network is to recover the vector

$$\mathbf{f}_2 = \begin{bmatrix} 1 \\ 3 \\ 2 \\ 1 \end{bmatrix}. \quad (3.4)$$

Once the clamped and floating partition is defined, Figure 3.1 can be reduced to Figure 3.2.

Since the neurons operate independently in this case, only data and results from the third neuron will be discussed. In Figure 3.3, two operational amplifiers are pictured, the first performs a weighted sum and the second inverts the output voltage of the first. The equation for the output voltage from the combined amplifiers is

$$v_{s_3}(M+1) = \frac{R_f}{R_3} v_{s_3}(M) + \frac{R_f}{R_1} v_{s_1} \quad (3.5)$$

Figure 3.1. APNN Neurons With Interconnection Weights.

Figure 3.2. Reduced APNN Network With Interconnection Weights.

Figure 3.3. Electrical Circuit for the Third Neuron.

where V_{s1} is the clamped (constant) voltage from the first neuron. Notice that the output is fed back into R_3 to update the system.

The values for R_1 , R_3 , and R_f are determined by the stimulus vector and Equations 1.1 and 3.5. The only condition on the resistors in the second amplifier is that their ratio be one. Since $S_1=1$ we will let $V_{s1}=1$ V and initialize $V_{s3}(0)=0$. From Equation 3.5,

$$V_{s3}(1) = \frac{R_f}{R_1} \quad (3.6)$$

Executing Equation 1.1 for the first iteration yields

$$\mathbf{s}(1) = \begin{bmatrix} 1.0 \\ 3.0 \\ 0.4 \\ 0.9 \end{bmatrix} \quad (3.7)$$

so that

$$s_3(1) = 0.4 \quad (3.8)$$

Thus

$$\frac{R_f}{R_1} = 0.4 \quad (3.9)$$

The procedure is duplicated for $V_{s3}(2)$ and $S_3(2)$ to find

$$\frac{R_f}{R_3} = 0.8 \quad (3.10)$$

Choosing R_f defines R_1 and R_3 . Repeating the process for any subsequent iteration duplicates Equations 3.9 and 3.10.

In Figure 3.4 we present an oscilloscope photograph

Figure 3.4. Oscilloscope Photograph Showing the Convergence of the Third Neuron.

showing the third neuron converging to a voltage of 1.820 V, as documented by ΔV_2 , the voltage difference between the two crossed points. The rise time, ΔT , is 4 msec, and denotes the convergence time of the neuron. Due to the limitations in the sampling rate of the scope, the voltage convergence looks like a step function instead of the actual exponential characteristic.

3.2 Potential Problems

Although the electrical model worked flawlessly, we were concerned with three potential problems:

- (1) Noise;
- (2) Skew effects;
- (3) Current attenuation.

All resistors generate thermal noise. Since the convergence process depends on feedback, we were worried that the noise would make the current snowball and cause the APNN to diverge. Fortunately, the noise is zero mean and cancels with the succeeding iterations.

Second, we studied the potential problems of skew effects. Since the APNN is described with matrix equations, its operation can be characterized as synchronous and discrete. Supplying pulsed voltages as inputs and timing each iteration will work if some components are added, but the network's response to continuous voltage levels that are

possibly not synchronous was a concern. Luckily, the APNN has been shown to behave indifferently to skewed inputs [5].

Third, we were concerned that normally insignificant attenuation losses might become critical with the feedback. This care was also unnecessary; no losses of this type occurred within the resolution of the measuring equipment.

3.3 Grey Level Resolution

Although this model will be compared and contrasted in great detail with the optical architectures in Chapters IV and V, a key observation about grey level resolution should be made here. The grey level resolution (the number of distinguishable levels needed to represent the interconnection matrix) is generally determined by

$$R = \det[F^T F] + 1. \quad (3.12)$$

One can verify this equation by calculating T according to Equation 2.7. We add 1 to allow for the number 0. In the computer program, the resolution is limited by the number of bits carried on the data bus of the microprocessor. For a spatial light modulator, the limit is set by the modulating element's sensitivity. In the electronic implementation, the limit is set by the resistor ratios R_f/R_1 and R_f/R_3 . Since resistors are cheap and can be made with great ohmic accuracy, the grey level resolution of this electrical model is essentially unlimited. In the optical implementations, we will see that the resolution of the spatial light modulators is

relatively small; hence the elements of T are restricted in range.

CHAPTER IV

PASSIVE OPTICAL IMPLEMENTATION

Incoherent processing is a growing area in the field of optics because of its many advantages over coherent processing. Incoherent processors generally are just as fast and yet are frequently cheaper than coherent processors because of the sources used. Incoherent systems also have reduced dynamic range requirements [7]. Finally, incoherent systems do not suffer as much from noise due to dust and optical aberrations in the imaging equipment since they are insensitive to phase.

In this chapter we present the first of two incoherent optical processors. The first model is termed passive because it is controlled solely by means of the power of the input signal. We will show that the simplicity, reliability, and low cost of this architecture make it the more attractive choice for an optical APNN machine.

Fundamental to the execution of the APNN is Equation 1.4, i.e.,

$$\mathbf{s}(M+1) = \underline{\eta} \mathbf{T} \mathbf{s}(M) \quad . \quad (1.4)$$

In Section 2.3 a computer program was used and in Chapter III the equation was reduced to a weighted summation and discrete electronic components were used. Here we shall attempt to

model Equation 1.4 using optical and electro-optical components.

Figure 4.1 shows a system consisting of an incoherent light emitting diode (LED) array, a matrix mask surrounded by guiding optics to form the matrix-vector multiplier, and a detector array. The LED intensities correspond to the values of the neuron states. Since Equation 1.4 is a recursive expression, feedback is needed between the detectors and the floating LEDs in order to update the neurons. The clamped LEDs have a constant intensity. The heart of the system, however, is the matrix-vector multiplier.

4.1 Optical Matrix-Vector Multiplication

Many matrix-matrix, matrix-vector, and vector-vector processor architectures have been documented [7-9]. The algorithm implemented in this thesis is based on representing the interconnection matrix T in its column space. Consider the multiplication of the $L \times L$ T matrix with the $L \times 1$ state vector $\mathbf{s}$:

$$\begin{aligned}
 T\mathbf{s} &= \begin{bmatrix} t_{11} & \cdots & t_{1L} \\ \vdots & \ddots & \vdots \\ t_{L1} & \cdots & t_{LL} \end{bmatrix} \begin{bmatrix} s_1 \\ \vdots \\ s_L \end{bmatrix} = s_1 \begin{bmatrix} t_{11} \\ \vdots \\ t_{L1} \end{bmatrix} + \cdots + s_L \begin{bmatrix} t_{1L} \\ \vdots \\ t_{LL} \end{bmatrix} \\
 &= \begin{bmatrix} s_1 t_{11} + \cdots + s_L t_{1L} \\ \vdots \\ s_1 t_{L1} + \cdots + s_L t_{LL} \end{bmatrix} .
 \end{aligned}
 \tag{4.1}$$

Figure 4.1. APNN System Configuration.

The goal is to vertically expand s_i onto the i^{th} column of T and to sum the inner product elements horizontally. In the optical configuration, the state of the i^{th} neuron s_i is represented by the intensity of the i^{th} LED. The interconnection element t_{ij} is represented by the intensity transmittance of a photographic transparency. Finally, the vertical expansion and summation operations are accomplished with waveguides.

4.2 Film as a Spatial Light Modulator

To simulate the multiplication of the inner products, the intensity from s_i is modulated by the intensity transmittance t_{ij} :

$$s_i^{\text{out}} = t_{ij} s_i^{\text{in}}$$

where s_i^{out} and s_i^{in} are the intensities after and before the film mask, respectively. The term t_{ij} will be interchangeably denoted as an element of T or as a transmittance because, in both cases, it will obey two rules:

$$(1) \quad 0 \leq t_{ij} \leq 1 ;$$

$$(2) \quad t_{ij} \text{ is real .}$$

If $t_{ij} = 0$, the mask is completely transmissive, and if $t_{ij} = 1$, it is opaque.

Two modulation schemes are available when designing the mask. First, the film negative can take analog form, making

t_{ij} continuous-valued, as shown in Figure 4.2. This method is difficult when many different values of t_{ij} are needed. Each grey level must be faithfully represented on paper in order for a negative to be made. Computer graphic software can only generate up to eight or ten different grey levels. Afterwards, the film development stage must be so controlled and precise that these grey levels do not become distorted. Just a few of the variables involved in this process are the shutter exposure, fix, stop, and rinse times.

A more practical method for generating a quantized intensity transmittance t_{ij} representation is area modulation. In Figure 4.3, the element is subdivided into smaller areas. Each area is either clear or opaque. In this illustration t_{ij} is quantized to one of five transmittances. Since the blocks comprising t_{ij} are binary, Kodak Kodalith film and developer can be used to prevent the transmittance distortion that usually occurs during development. One disadvantage to this method is that the light incident upon the modulating area must be uniform in intensity; otherwise one portion of the area unduly biases the overall intensity. Nevertheless, area modulation is usually the preferred choice and was used for the project described in this thesis.

Figure 4.2. Continuous-Valued Intensity Transmittances.

Figure 4.3. Intensity Transmittance by Area Modulation.

4.3 The Matrix-Vector Multiplier

The entire architecture is centered around the spatial light modulator. In most cases this single component determines the attributes of the entire system. Consequently, once the film mask is designed, most other system parameters are determined, such as the optical waveguide configuration surrounding the mask and, of course, the vectors to be stored in the APNN that directly determine the elements of T.

4.3.1 Choosing the Library Matrix

This first architecture considered is a proof of principle example and we chose the elements of f_i to be binary:

$$F = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}. \quad (4.2)$$

The consequent interconnection matrix is

$$T = \begin{bmatrix} 1/2 & 0 & 0 & 1/2 \\ 0 & 1/2 & 1/2 & 0 \\ 0 & 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 & 1/2 \end{bmatrix}. \quad (4.3)$$

Thus the system consisting of four neurons has two stored vectors and at most two neurons can be floating. Also, in order for F_p to be full rank (refer to Equation 2.9), either the first and fourth or the second and third neurons cannot be floating simultaneously.

Notice that the rank of T is two since $N=2$, (N is the

number of library vectors), and two transmittance values are needed.

4.3.2 Choosing the Optical Waveguides

From Section 4.1 we know that the optics preceding the matrix mask must vertically collimate light from LED s_i so that it is projected onto the i^{th} column of T . The optics which follow the mask must horizontally sum the inner product modulated components to form the resultant. The two most promising choices for the task are lenses and glass microscope slides. In previous work done on optical matrix-vector multipliers [8,9], the preference has been microscope slides aligned vertically in front of the mask and horizontally behind it. Since the ends of the slides are pressed flush against the mask, they are more efficient in terms of allowing scattering. Light from the LEDs propagates through the glass by means of total internal reflection [9].

The physical dimensions of the slides play a key role in building the matrix mask. For this architecture the dimensions are 75 mm x 25 mm x 1 mm. The logical size for the matrix element t_{ij} is 1 mm x 1 mm. If t_{ij} is any smaller the light intensity is overly attenuated and if it is any bigger crosstalk between the inner product terms and the LEDs may be a problem; the light that is not coupled into the slide can

pass right through to the detectors. Steps taken to avoid this problem are outlined in the last section of the chapter.

4.3.3 Building the Matrix Mask

We have established that the interconnection matrix T will have dimension 4×4 , element values of 0 and .5, and an element area of 1 mm x 1 mm. To avoid the problems associated with producing $t_{ij} = .5$, the matrix is scaled to T' , where

$$T' = 2T = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \quad (4.4)$$

and the current in the feedback path updating the LED intensity is electrically divided in half. This scheme will work for any intensity, during any iteration, because the product operation Ts is a linear operation and can be replaced with $0.5T's$.

Figure 4.4 is a reduced version of the graphic constructed to obtain the photographic negative. The most important, yet tedious phase in creating the mask is to scale Figure 4.4 such that the elements of t_{ij} on the film are as close to 1 mm x 1 mm as possible. For this reason the pattern dimensions are proportional to that of a photographic negative. Exposure times ranged from 1/30 of a second to three seconds when pictures of Figure 4.4 were taken. We chose the negative that most closely produced opaque and perfectly transmissive matrix elements.

Figure 4.4. Pictorial of Mask Pattern.

4.4 LED Performance

The two primary requirements for "good" LED performance are:

(1) The LED must have a large enough intensity range for the photosensor to discriminate between the energy differences in succeeding iterations;

(2) This range must be linear.

The curve of Figure 4.5 characterizes the LED in both of these areas. We measured the current generated by the detector array (to be discussed below) as a function of the current through the LED. A library vector element of "1" is assigned to 50 mA to satisfy the above criteria and to keep from unnecessarily overloading the LED and degrading its operating characteristics. Finally, individual differences between LEDs were found to be small enough to ignore.

4.5 Detector Array

The last section of Figure 4.1 to be discussed is the detector array. Two detectors are needed to initiate feedback to the (at most) two floating LEDs. The Hamamatsu S1337-16BR infrared-enhanced PIN silicon photodiode was chosen [11]. The sensing area "matched" the end of the microscope waveguide and its response is maximized at the near infrared wavelength of the LED. The output voltage of the detector circuit of Figure 4.6 is given by

Figure 4.5. LED Characteristics.

Figure 4.6. Detector Circuit.

$$V_o = I_D R_L \left(1 + \frac{R_f}{R_i}\right) \quad (4.5)$$

The magnitude of the photon-generated current, I_D , is on the order of $1 \mu A$ [10]. The resistor combination $R_L \left[1 + \frac{R_f}{R_i}\right]$ must provide an output voltage (hence feedback current) that is large enough to initiate convergence of the APNN. To determine these component values we will step through the network's operation. Recall the library matrix

$$F = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad (4.6)$$

the interconnection matrix

$$T' = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \quad (4.7)$$

and the convergence equation

$$\mathbf{s}(M+1) = 1/2 \mathbf{T}' \mathbf{s}(M) \quad (4.8)$$

We shall recall the vector

$$\mathbf{f} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (4.9)$$

and designate the stimulus vector as

$$\mathbf{s}(0) = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}. \quad (4.10)$$

Execution of Equation 4.1 reveals the state vector

$$\mathbf{s}(1) = \begin{bmatrix} 1 \\ 1 \\ 1/2 \\ 1/2 \end{bmatrix}, \quad (4.11)$$

which means that after the first iteration 25 mA flows in each of the two floating LEDs. After the second iteration the current is 37.5 mA, etc. The physical operation of this network should be clear. The clamped LEDs, which are supplied with a constant 50 mA, pump energy into the system and force the floating LEDs to a steady state value which signals convergence.

The resistances R_L , R_f , and R_i must be chosen such that the feedback current from the first iteration is 25 mA. This is accomplished by assigning

$$R_i = 100 \, \Omega, \quad (4.12)$$

$$R_L = 330 \, \Omega, \quad (4.13)$$

and varying R_f as a potentiometer until this current is reached. A current exceeding 25 mA will cause the APNN to diverge. A current less than 25 mA will cause the APNN to converge short of 50 mA. In the latter case we can nevertheless decide "1" since the alternative is "0," which is no current flow at all. If, however, the elements of the library vector to be recalled are not the same, this error may be too large and a wrong decision might be made. The very

significant problem of noise associated with the detector dark current will be discussed in the next section.

4.6 System Integration

With the subsystems outlined, we are ready to proceed to the most difficult phase of the implementation: system integration. In this section we will discuss problems encountered in coupling the LED light into the waveguide and detecting the elements of the matrix-vector product. A subsection is devoted to each problem area.

4.6.1 Crosstalk Problems

A top view of the system with waveguides in place is given in Figure 4.7. Areas where crosstalk poses the greatest threat to degradation of performance are at the LED-waveguide interface and the waveguide-detector interface. Notice that light from adjoining LEDs can easily become coupled with the wrong waveguide. The same is also true between the waveguides and detectors. To reduce crosstalk among LEDs black light-blocking tape was placed between each source. This tape was also the solution to crosstalk problems at the detector end.

A less threatening crosstalk problem was encountered at the waveguide-film mask-waveguide interfaces. After the mask was developed the transmissive squares were a little larger than 1 mm x 1 mm. This sizing error allowed the uncoupled light to pass straight through the mask and become detected.

Figure 4.7. Areas of Potential Crosstalk Problems.

Affixing tape perpendicularly between adjacent slides solved this difficulty.

4.6.2 Detector Noise Problems

The resistance value R_f needed to generate an LED current of 25 mA after the first iteration is about 300 k Ω .

The total gain of the amplifier, $1 + \frac{R_f}{R_i}$, is 3001. With this large gain dark current is a major problem. We consider dark current to be current generated by the detector without any incident light, due mostly to thermally generated carriers [12]. Since this current is also amplified by the circuitry, the dark current increases linearly with the gain. Noise is a particularly large obstacle for processors with feedback such as neural networks, because with each new iteration new noise is generated and noise from past iterations is reamplified as well. Any such noise will thus most likely cause the APNN to diverge.

A solution for extracting the noise after each iteration was found. If this noise voltage is broken down into AC and DC components, we know that the AC portion has zero mean and the DC portion is signal independent. The cascaded operational amplifier shown in Figure 4.8 is in its summing configuration and subtracts the DC noise voltage after each iteration. The AC noise voltage continuously cancels with successive iterations.

Figure 4.8.. Integrated System With Feedback for One Neuron.

Incorporation of the second amplifier has another advantage. Assume that each LED has a voltage drop of 2 V. Since we are starting the floating LEDs at 0 A, the feedback current initiated from the first iteration must overcome this 2 V drop to "turn on" the LED and still provide 25mA. With the cascaded amplifier we "tune" the added bias voltage to a level such that the LED almost turns on and the feedback current can be exclusively used for APNN convergence. Figure 4.8 shows this amplifier and the voltage as V_{bias} . The feedback circuitry for one neuron is also shown.

4.7 System Strengths and Limitations

In the final section of Chapter IV we compare this passive optical architecture with the computer program of Section 2.2 and the electronic example of Chapter III. For those knowledgeable in optics, familiar comparisons between optics and electronics will be made.

4.7.1 Spatial Resolution

Spatial resolution is discussed first because it directly determines computing power. This term is usually defined in terms of line pairs per unit area or pixels per unit area. Since the LED light is modulated spatially, spatial resolution is a measure of the maximum possible number of neurons that can be fit into the system. Resolution is clearly limited by the physical dimension of the spatial

light modulator, waveguides, and LEDs. When one considers that 16,384 neurons are needed for processing a 128 x 128 image, the optical configuration is not feasible. Some 16,384 LEDs and almost as many detectors would be needed. The most efficient model spacewise is the electronic model. With VLSI semiconductor technology, an APNN with 16,384 neurons can be reliably fabricated in a relatively miniscule area.

A parameter related to the physical dimensions is sturdiness. Although the optical architecture described is lighter than most optical processors, problems with alignment and stability are still present. An optical APNN will not be a reliable processor if it must experience the jolt of a rocket launch or jar of an airplane landing. Either of the two other implementations is a far better choice if movement is to occur.

4.7.2 Grey Level Resolution

As spatial resolution determines computing power, grey level resolution determines computing accuracy. For this discussion we assume that elements of the library vectors f_i correspond to sampled grey level values. These numbers, of course, can represent any sampled quality or characteristic. For the optical APNN, grey level resolution is defined as the number of distinguishable intensity transmittances that can be represented in the interconnection matrix T . As the

elements of f_1 increase in range the elements of T become more difficult to represent. As entire f_1 vectors become close in L space, representing T is even more difficult.

Two schemes have been outlined for constructing the mask that represents T . Assuming area modulation is chosen, grey level resolution is also dependent on the dimensions of the system. From our experience, a 1 mm x 1 mm area can be faithfully subdivided into nine binary blocks, generating the ten transmittances 0, 1/10, 2/10, etc. Such a low resolution severely limits the choice of vectors for F .

As in the case of spatial resolution, optics cannot compete with electronics in the area of grey level resolution. The system of Chapter III has a virtually unlimited grey level resolution because the limit is determined by resistances. Resistors can be fabricated with great range and accuracy. In a microprocessor system resolution is set by the number of bits carried. For an 8-bit data bus, 256 grey levels are available.

As a final note on grey level resolution, we should point out that the limit can be prematurely set by an insensitive detecting circuit that cannot sense modulated intensity differences from two close transmittance values. The predominant trade-off with detectors is sensitivity versus speed [12]. Sensitivity is proportional to detector area since a larger area means more silicon is available to

absorb photons. Response speed is inversely proportional to area since the generated carriers must travel farther to become external current. With this in mind, the Hamamatsu detector is designed with the high sensitivity of 0.5 A/W [11].

4.7.3 Speed of Operation

At last we find an area in which optics has an advantage over electronics. In the optical system the limiting factor is the response time of the detection system. For the circuit of Figure 4.6 the response time is on the order of 100 nsec [11]. We define response speed as the rise time from 10% to 90% of the output current value once a step input of light is incident on the detector [11]. Since the optical matrix-vector multiplier is a parallel processor, the response time will be the same regardless of the dimension of the system.

For the computer program, however, an L dimensional matrix-vector product requires L^2 multiplications and L^2 additions. Assuming 25 and 7 clock cycles are needed for a multiplication and addition [13], respectively, the basic operation time for $L=4$ neurons and a clock speed of 12 MHz takes 43 μ sec. The optical system thus provides the result almost 500 times faster.

As the dimension grows, the time advantage of optics is even greater. Matrix-vector multiplication for a 16384-neuron system takes almost 12 minutes for the microprocessor system,

but still only 100 nsec for the optical system! Remember, also, that these times are for one iteration only. Hundreds of iterations may be needed, depending on the accuracy needed.

Improving this operating speed will come by means of optical feedback. One possibility for an all-optical APNN is matrix-vector multiplication with wave mixing in photorefractive crystals [14,15].

4.7.4 Programmability

A flexible APNN allows for the "real time" updating of the interconnection matrix as library vectors are added or deleted. The optical architecture of this chapter and electronic implementation of Chapter III do not have this feature. The principal advantages of the active architecture to be described in Chapter V are both its programmability and speed.

CHAPTER V

ACTIVE OPTICAL IMPLEMENTATION

In this chapter we present the second incoherent APNN processor. This architecture employs active spatial light modulators which allow for the real time updating of the interconnection matrix as library vectors are added. Although the system is more complicated, optical matrix-vector multiplication and electronic feedback are still its basic functions.

The system of Figure 5.1 consists of two active spatial light modulators, two light sources, two polarizers, focusing optics, a 50% transmissive beam splitter, and a detector array providing feedback. As was the case with the passive configuration, the LED intensities correspond to the neuron

states. The symbols and denote that the orientation of the polarizer/analyzer pair is crossed by 90°. The matrix-vector multiplication is accomplished with a Semetex Iron Garnet H-Triggered Magneto-Optic Device (SIGHT-MOD), a Hughes Liquid Crystal Light Valve (LCLV), a white light source, and associated focusing optics. The interconnection matrix is encoded onto a beam of white light by the SIGHT-MOD and is projected onto the write side of the LCLV. The LED light is vertically expanded by the first set of optics, modulated and reflected from the read side of the LCLV in proportion to the

Figure 5.1. Top View of the Active APNN System Configuration.

light intensity transmitted by the SIGHT-MOD. The reflected light is horizontally summed by a second set of optics to form the matrix-vector product, and is then detected to update the floating LEDs. A section below is devoted to each system component.

5.1 The Hughes Liquid Crystal Light Valve

For most applications the LCLV is an optically read and optically addressed, incoherent-to-coherent real time image converter [16]. In this architecture, however, the LCLV performs the same function as the photographic negative of the last architecture: multiplication of the interconnection matrix element t_{ij} with the state vector component s_i . We will first give a description of the LCLV, briefly describe its operation, and show how it was read and written with incoherent light and used for analog multiplication. A more detailed analysis is provided elsewhere [17-20].

5.1.1 LCLV Components

The pictorial of Figure 5.2 indicates that the LCLV consists of a sandwich of thin films with an AC voltage applied across the entire device. The cadmium sulfide (CdS) photoconductor is sensitive to white light having a wavelength in the range of 520-550 nm. The cadmium telluride (CdTe) light-blocking layer separates the writing and projecting functions. The dielectric mirror reflects the

Figure 5.2. Schematic of the LCLV.

projection light. A nematic liquid crystal layer causes birefringent polarization modulation of the projection light.

Insulating layers provide surfaces onto which the nematic liquid crystal molecules align. If the preferred alignment direction (alignment direction of the long axis of the molecules) between the two layers is offset by 45° , the liquid crystal molecules adjust into a continuous helical form, as shown in Figure 5.3. The polarization direction of linearly polarized light will follow this twist pattern in passing through the liquid crystal.

5.1.2 LCLV Operation

Consider the conditions of Figure 5.4 in which no light is incident on the write side (off-state) and a polarizer/analyzer pair is placed on the read side of the LCLV. Unpolarized light is linearly polarized before entering the liquid crystal. The polarization direction is rotated 45° after traversing through the liquid crystal and is rotated -45° after reflecting from the dielectric mirror and propagating back through such that the polarization directions of light entering and leaving the LCLV are parallel. Thus the crossed analyzer blocks all light.

In the on-state of Figure 5.5, the write light intensity lowers the resistance of the CdS light-sensing layer, switching more of the AC voltage across the liquid crystal layer. In response, the nematic molecules attempt to align

Figure 5.3. Twisted Nematic Effect.

Figure 5.4. LCLV Off-State.

Figure 5.5. LCLV On-State.

themselves with the electric field. Incident linearly polarized projection light becomes circularly polarized due to the properties of birefringence and a portion of the light passes through the analyzer. Since the APNN matrix-vector multiplication is an analog, linear process, the write light intensity from the SIGHT-MOD must linearly modulate the LED projection light intensity. In other words, for a given write light intensity, changing the LED intensity by a factor of α must change the intensity of the transmitted light by the same factor. In Section 5.4, after a discussion of the SIGHT-MOD and detector array, we present data showing the LCLV's linear operating range.

5.2 The Semetex SIGHT-MOD

The SIGHT-MOD is a high-speed electrically addressed two dimensional binary magneto-optic display system [21,22]. In this project the SIGHT-MOD encodes the interconnection matrix onto a white light beam. Like the LCLV, the SIGHT-MOD modulates the polarization of incident light. With an added polarizer, intensity modulation is created.

In making the SIGHT-MOD, a magnetic garnet film is epitaxially deposited onto a nonmagnetic optically transmissive substrate. A two-dimensional 128x128 mesa array is etched by photolithography and X-Y address lines for high speed electronic pixel switching are laid down.

The operation of the SIGHT-MOD can be explained with the aid of Figure 5.6. The polarization direction of vertically polarized light is rotated either clockwise or counter-clockwise by the Faraday effect, depending on the sense of magnetization of the mesa. Ideally this rotation is $\pm 45^\circ$ so that the analyzer completely blocks or passes the light to create binary on/off states. In practice this angle of rotation is approximately 35° . To change the rotation direction, current is passed through the appropriate address lines and the combined magnetic field from the X and Y address lines will switch the polarization rotation of the selected mesa.

In order for this inherently binary device to create continuous intensity levels, area modulation is used. For the 128×128 mesa/pixel array and a 4×4 interconnection matrix, a 32×32 block can be allotted for each t_{ij} element. The ideal grey level resolution is thus established at $32^2 + 1 = 1025$. An AppleII GS computer served as the interface to the SIGHT-MOD, and the T matrix was stored as software. In Section 5.4 we further discuss encoding the matrix.

5.3 Improved Detector Array

While integrating the components of Figure 5.1, it became evident that a better detecting circuit was needed. The polarizers and beam splitter caused significant attenuation of the LED light intensity. Also, operational

Figure 5.6. Operation of the SIGHT-MOD.

amplifiers are low current devices to begin with. Since current must be generated to drive the floating LEDs, a current amplifier stage was cascaded onto the end of the previous detecting circuit. A large current gain can be provided only if the amplifier has a high input impedance, relative to the load impedance. A general linear amplifier with load is shown in Figure 5.7. The current gain of this amplifier, A_I , is

$$A_I = A_V Z_i / R_L \quad (5.1)$$

where $A_V = V_o/V_i$ is the voltage gain, Z_i is the input impedance, and R_L is the load resistance. While the load resistance can be chosen with a fairly large degree of freedom, the voltage gain and input impedance depend mostly on the type of transistor used [23]. For example, a bipolar junction transistor (BJT) characteristically has a lower input impedance than a field effect transistor (FET). Thus the FET was found suitable as a current amplifier when biased in the common-drain configuration of Figure 5.8. The voltage gain is approximately 1 and the input impedance is simply R_G , the gate resistance [23].

Operation in the pinchoff region is assured if the source voltage is greater than the gate voltage and if the drain voltage is greater than the gate voltage by $|V_p|$, the magnitude of the gate-source voltage required to "pinchoff" the FET channel [23]. Since a 2N3819 N-channel FET with $I_D=10$

mA and V_{GS}

Figure 5.7. A Linear Amplifier.

Figure 5.8. FET Current Amplifier.

mA and $V_p = -3.5$ V was used, circuit values of $V_{DD} = 20$ V, $R_s = 5$ k Ω , and $R_G = 100$ k Ω were chosen to assure pinchoff operation and produce a current gain of 100.

5.4 System Integration

Due to the physical dimensions of the components of Figure 5.1, the largest number of neurons that the system can process is four. In this section we will discuss the focusing optics for the LED light, the modulating characteristics of the LCLV, and the encoding of the interconnection matrix onto the white light beam with the SIGHT-MOD.

5.4.1 Focusing Optics

Since a beam splitter must reflect the modulated LED light onto the detector array, using glass microscope slides as waveguides is impractical. Dispersion makes detecting the matrix-vector product impossible. Cylindrical lenses are used instead to first vertically expand the LED light for modulation by the LCLV and second to horizontally sum the partial products to form the result. The LEDs, which have a diameter of 5 mm, are adjoined in a row. Two cascaded cylindrical lenses each having a width and height of 50 mm and 60 mm, respectively, produce four distinct columns of light focused on the LCLV, whose modulating element has a diameter of 50 mm. Four more cylindrical lenses of the same

size converge the reflected beam into one column of light. When an intensity distribution representing the interconnection matrix is projected onto the write side of the LCLV, this column will have four distinct regions representing the four elements of the matrix-vector product.

5.4.2 LCLV Linear Operating Range

We have expressed the need for the LCLV to linearly modulate the LED output intensity so that the network's behavior is predictable. Once the components of Figure 5.1 were in place, measurements to find a linear operating range were made. The curve of Figure 5.9 depicts this characteristic. We varied the current through an LED from 0 to 70 mA and measured the current generated by the detector circuit for a constant write light intensity. The LED intensity from the feedback LED was blocked. To make the data more meaningful, we placed the SIGHT-MOD in the path of the constant write beam and switched every pixel to its transmissive state. The write light intensity was $0.57 \mu\text{W}/\text{mm}^2$.

Note that feedback current was not generated until the source LED current was 34 mA. The most linear portion of the characteristic curve is between source LED currents of 45 and 55 mA. Above 60 mA, the feedback current levels off somewhat and the system appears to saturate. Feedback data for source LED currents greater than 70 mA were not measured for fear of

the LEDs

Figure 5.9 System Linearity with SIGHT-MOD 100% Transmissive.

destroying the LEDs. Ideally, the curve should have a slope of 1 and should pass through the origin so that the generated feedback current is the same as the source current.

The nonlinear behavior can be attributed to either the LED, LCLV, the detecting circuit, or a combination of the three. The LED can be dismissed because of its linear behavior shown in Figure 4.5. For the LCLV, there is no known theoretical or practical reason for this nonlinear behavior. Thus our attention became focused on the detecting circuit. We believe that for LED source currents less than 34 mA, the operational amplifiers and FET stage could not generate enough current to overcome the voltage drop due to the feedback LED in order to "turn it on." Once the LED was on, the operating range was fairly linear until the source LED current reached 60 mA. At that point the operational amplifiers saturated and could not provide any more current to the FET.

The source LED current of Figure 5.9 represents the constant current of the clamped LED. These values must be judiciously chosen to keep the APNN operation in the linear range. In Sections 5.5 and 5.6 we discuss two examples and step through the analysis.

5.5 Example With Binary Library Vectors

The first test of the active optical architecture was to recall the binary library vectors of Equation 4.2, i.e.,

$$F = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}. \quad (4.2)$$

The interconnection matrix,

$$T = \begin{bmatrix} 1/2 & 0 & 0 & 1/2 \\ 0 & 1/2 & 1/2 & 0 \\ 0 & 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 & 1/2 \end{bmatrix}, \quad (4.3)$$

was again changed to

$$T' = 2T = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \quad (4.4)$$

so that the detecting circuit only needed half the gain than if Equation 4.3 was used. The most critical part of getting this system to work is to correctly encode Equation 4.4 onto the SIGHT-MOD.

5.5.1 Encoding the Interconnection Matrix

The function of the SIGHT-MOD is to represent the interconnection matrix as a two-dimensional light intensity distribution. With help from the optics the pattern is focused onto the write side of the LCLV, as shown in Figure 5.10.

We reported that for a 4x4 interconnection matrix, a 32x32 pixel block can be allotted for each t_{ij} matrix element since the spatial resolution of the SIGHT-MOD is 128x128. Actually, a 13x32 pixel block was needed as noted. If the

Figure 5.10. Intensity Matrix Incident on the LCLV Write Side.

white light beam passing through the SIGHT-MOD is focused by the optics such that it circumscribes the working face of the LCLV, each pixel is $50 \text{ mm}/128 = 0.39 \text{ mm}$ wide. If we assume that the pixels are square, then the area of one pixel is $(0.39 \text{ mm})^2 = 0.153 \text{ mm}^2$. The spatial resolution of the entire system can be immediately calculated as

$$\text{SR} = 1/0.153 \text{ pixels/mm}^2 \times \pi(50/2)^2 \text{ mm}^2 \approx 12,868 \text{ pixels. (5.2)}$$

This number is therefore clearly limited by the resolution of the SIGHT-MOD since the LCLV's specified resolution limit is greater than 157,000 pixels [15].

The four vertically expanded LED light beams were 5 mm wide and 5 mm apart. Since each pixel has a width of 0.39 mm, we needed each t_{ij} block to be $5/0.39 \approx 13$ pixels wide and 13 pixels apart. (In the laboratory, three pixels were added to each side to compensate for alignment and measurement errors.) The height of each t_{ij} element block was $128/4 = 32$ pixels.

Notice from Figure 5.10 that the matrix elements t_{11} , t_{14} , t_{41} , and t_{44} do not exactly fit onto the round LCLV working face, causing light from the middle elements to bias the system. A remedy for this problem is to scale the intensity array down to a suitable size. However, no measureable errors from this discrepancy could be detected in our system.

5.5.2 Convergence

With the SIGHT-MOD analysis completed, the system was ready to be tested. We defined the APNN stimulus vector as

$$\mathbf{s}(0) = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad (4.10)$$

so that, without loss of generality, the floating LEDs were initialized at 0 mA. We again defined "1" as 50 mA to keep the convergence process on the most linear portions of Figures 4.5 and 5.9. With the LEDs off we measured the detector dark current and electronically subtracted it with V_{bias} in the operational amplifier of Figure 4.8 until just a few microamperes of current flowed in the floating LEDs. Next, the clamped LEDs were turned on and the floating LEDs were covered so that the current generated from the first iteration, which was just below 25 mA, could be observed. Once the floating LEDs were uncovered they converged to about 63 mA, instead of 50 mA, because of the nonlinearity of Figure 5.9. Nevertheless, the system converged and a "1" can be deduced from this current value.

Figure 5.11 is a photograph from the oscilloscope that measured the voltage of the detector as the LED current converged. The rise time was 0.1 sec and was thus 25 times slower than the electronic implementation since its rise time was 4 msec. However, the speed of this optical APNN is limited by the response time of the detectors and

Figure 5.11. Oscilloscope Photograph Showing Convergence of the Binary APNN.

amplification system. If the amplifiers had been designed more carefully, the speed of the optical implementation would have surpassed the electronic implementation. One should also keep in mind that the speed of the optical system will be constant regardless of the dimension of the system (number of neurons) due to the parallel nature of the architecture. As discussed in Section 4.7.3, the convergence speeds of the electronic implementation and computer program slow considerably as the dimension of the system increases.

Although this test was considered a success because it demonstrated that the system converged, it is equally important to show that it could diverge. When the intensity matrix was

$$T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad (5.3)$$

and was focused onto the write side of the LCLV, the floating LEDs burned out because of the high feedback current generated. When the intensity matrix was

$$T = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (5.4)$$

the floating LEDs converged to 3.6 mA. Some 2.5 mA was generated after the first iteration because the polarization rotation due to the mesas of Figure 5.6 was actually $\pm 35^\circ$. A small amount of light will always pass through the analyzer

from mesas in their off-state. Likewise, mesas in their on-states never pass the full light intensity incident on them. In the next section we discuss how these consequences limit the contrast ratio of the system.

5.6 Example With Continuous-Valued Library Vectors

In this section we incorporate the intensity modulating capabilities of the SIGHT-MOD. The procedure for recalling continuous-valued library vectors is outlined and results from this example are presented.

5.6.1 Encoding the Interconnection Matrix

Consider the library and interconnection matrices

$$F = \begin{bmatrix} 2 & 2 \\ 3 & 0 \\ 2 & 0 \\ 2 & 0 \end{bmatrix} \quad (5.5)$$

and

$$T = \begin{bmatrix} 1.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.53 & 0.35 & 0.35 \\ 0.00 & 0.35 & 0.24 & 0.24 \\ 0.00 & 0.35 & 0.24 & 0.24 \end{bmatrix} . \quad (5.6)$$

Although the vectors in Equation 5.5 can hardly be considered continuous-valued, they are not binary, and the procedure for recalling continuous-valued library vectors is the same regardless of the numbers' range and closeness in magnitude.

The challenge of this example is to represent the elements in the matrix of Equation 5.6 as intensities by

means of SIGHT-MOD area modulation. One might hypothesize that to represent the number 0.53, for example, 53% of the pixels in the t_{22} 13x32 pixel block should be in their transmissive mode. Unfortunately, the encoding procedure is not this obvious. Data from the curve in Figure 5.12 was used to "look-up" the amount of light transmission that was needed. We measured the feedback current, normalized to 50 mA, that was generated from various transmission percentages of a 13x32 pixel block. Since we want 53% of the LED current to be generated in the feedback loop from the t_{22} component, approximately 40% of the pixel block must be transmissive. The same procedure was followed to decide that blocks $t_{32} = 0.35$ and $t_{33} = 0.24$ should be 37% and 32% transmissive, respectively.

5.6.2 Convergence

The library vector

$$\mathbf{f}_1 = \begin{bmatrix} 2 \\ 3 \\ 2 \\ 2 \end{bmatrix} \quad (5.7)$$

was extrapolated from a stimulus vector of

$$\mathbf{s}(0) = \begin{bmatrix} 2 \\ 0 \\ 0 \\ 2 \end{bmatrix} . \quad (5.8)$$

The library vector element "2" was defined as 50 mA so that

Figure 5.12. SIGHT-MOD Modulating Characteristic Curve.

the second and third neurons would converge to 75 mA and 50 mA, respectively. From Equation 1.4,

$$\mathbf{s}(1) = \begin{bmatrix} 2.00 \\ 0.70 \\ 0.48 \\ 2.00 \end{bmatrix}, \quad (5.9)$$

and the corresponding currents in the floating LEDs should be 17.5 mA and 12.0 mA after the first iteration. In the laboratory these currents were 21 mA and 14 mA, respectively, and the neurons converged to currents of 60 mA and 82 mA. As was the case with the binary implementation, these converged current values were larger than ideal because of the nonlinear characteristic of the curve in Figure 5.9. The error in these values is not a concern as long as we can resolve the currents and deduce "2" from 60 mA and "3" from 82 mA.

In Figure 5.13 we present a photograph of the oscilloscope which displayed the output voltage of the detectors as the currents in the LEDs converged. One can easily resolve the two voltage levels to deduce a "2" and "3."

5.7 System Strengths and Limitations

The strengths of this system are its speed and programmability, while its weaknesses are its spatial and grey level resolutions, and its cost.

Figure 5.13. Oscilloscope Photograph Showing the Convergence of the Continuous-Valued APNN.

5.7.1 Spatial Resolution

We reported that the spatial resolution of the system is 12,868 pixels, and is limited by the SIGHT-MOD. Unfortunately, this resolution cannot be fully used because of the relatively large physical size of the LEDs. The vertical LED light beams incident on the read side of the LCLV were focused to widths of 5 mm, or 13 pixels. Thus the first step to improving the computing power of the system is to use smaller LEDs. Unfortunately, the size of the high-power LEDs must currently be approximately three times larger than that of the smaller power LEDs in order to generate large light intensities. Even with three times the number of LEDs, however, the computing capability of the optical system is far below the capability of the electrical implementation. Overall, the spatial resolution of this system is potentially higher than the passive optical architecture but is still lower than the resolution of the electronic model of Chapter III.

5.7.2 Grey Level Resolution

The grey level resolution is directly limited by the spatial resolution since we represented the interconnection matrix elements by means of area modulation. The spatial resolution of the elements, which was 13x32 pixels, defined the grey level resolution as $13 \times 32 + 1 = 417$ levels. The first step in increasing this number is to spread the width

of the vertical LED light beam and use a larger modulating area. Thus, we trade grey level resolution for the number of neurons in the system. One problem associated with the spreading of the LED light and using more of the LCLV face is crosstalk. It is very difficult to align the expanded LED light intensity columns focused on the LCLV read side with the t_{ij} matrix intensity elements incident on the write side. To avoid this problem, we focused the LED light columns 5 mm apart, as shown in Figure 5.10, to allow for alignment errors. If one needs more resolution than 417 grey levels, he must be willing to sacrifice computing accuracy due to the crosstalk problem.

This system's grey level resolution is higher than that of the passive architecture of Chapter IV where only 10 levels could be reliably produced. Compared to a microprocessor with an 8-bit data bus and $2^8 = 256$ levels, this system is also better. However, the electronic example has the highest grey level resolution since the limit is determined by resistor ratios.

5.7.3 Speed of Operation

The time needed for an optical matrix-vector multiplication in this system is practically the same as for the passive system since the speed is determined by the propagating medium for the light. A more interesting parameter is the time needed to update the interconnection

matrix. The SIGHT-MOD's frame-addressing time is 512 μ sec [20] and the LCLV's writing time (the time needed for the nematic molecules to adjust to the incident write light) is less than 50 msec [16]. Thus, the real time updating of the interconnection matrix is possible. This characteristic is mandatory if a system needs to "learn" new library vectors.

An autonomous artificial intelligence system, for example, may remember an event by storing in memory data taken while the event occurred. If the same data is sampled again, the machine may predict that the same event is occurring at that time. In terms of the APNN, this data and event can be stored in the form of a library vector, with the sampled data representing the clamped portion of the vector and the event representing the floating portion. If the events occur at least 50 msec apart, the APNN can "learn" the events and decide if one of them is occurring.

5.7.4 Cost

Until the demand for the LCLV and SIGHT-MOD becomes so great that they are mass produced, the cost of this architecture is prohibitive. The costs of the LCLV and SIGHT-MOD are currently roughly \$20,000 and \$15,000, respectively, while the entire systems presented in Chapters III and IV can be made for less than \$20 each.

CHAPTER VI

CONCLUSIONS

In this thesis we have presented a theoretical, geometry-based analysis and two optical implementations of the alternating projection neural network. The results from these implementations have been compared with the results of a traditional computer simulation and an electronic implementation. We hope that the reader has acquired a good understanding of the APNN and how optics might be used to build an APNN machine.

Compared to other neural networks, the APNN is appealing because of the mathematical completeness with which its convergence can be described. In Section 2.6 we derived the APNN convergence as a function of the eigenvalues of the T_4 matrix and the number of iterations which have been completed. For the case of one stored library vector, this rate is constant. In general, the APNN convergence rate is highest during the first iteration and approaches the spectral radius of T_4 as the number of iterations increases.

Although the APNN is very interesting to study, its applications are limited because of its low capacity for library vectors and the stringent conditions that are placed on the stimulus vector. The number of clamped neurons must be greater than or equal to the number of stored library vectors

if convergence is to be assured. This capacity is small compared to the second order Hopfield model, for example, whose capacity is of the order of the square of the total number of neurons [1]. Also, unrealistic conditions are placed on the neurons. Each neuron must be designated as either clamped or floating before convergence can start, and the state of the clamped neurons must be known exactly. For pattern recognition or image processing applications referred to in this thesis, the requirement this places on data acquisition systems is impractical. The APNN's general behavior from a stimulus not represented in the library matrix is not well understood. Overall, we cannot think of any applications in which the APNN would supercede another network model.

In Chapter III, we presented an electronic APNN consisting of operational amplifiers and resistors. The simplicity and low cost of this model make it worth investigating. Since its contrast ratio and grey level resolution were found to be determined by resistor ratios, the configuration is a top candidate for nonprogrammable applications.

The first optical architecture was presented in Chapter IV. For an optical processor it is very inexpensive. The major problems of the passive model are its low spatial and grey level resolutions since area modulation was used to represent the interconnection matrix elements. For

applications needing high speeds, however, the passive architecture is worth considering.

The second optical architecture added the dimension of programmability to the system. The SIGHT-MOD can be updated within a television frame-rate period. This active APNN also has a potentially higher spatial resolution, assuming the LEDs can be replaced with current-controlled sources having smaller physical dimensions. The biggest shortcoming of this system is its prohibitive cost. In most cases, this single cost factor will override any benefits from the architecture's speed and programmability.

Several improvements can be made to the active implementation presented here. First, laser diodes could replace the LEDs, allowing for the simulation of more neurons due to their low angular dispersion. One must be wary, however, of the low power and the infrared (or near infrared) operation of these devices. High power is necessary to create a large operating range on the curve of Figure 5.9. Since the dielectric mirror is transmissive to wavelengths greater than 800 nm [16], the infrared wavelengths must be filtered, and power must come from wavelengths in the visible range. The Sharp LT030 laser diode, for example, has a mean wavelength of 750 nm and has a fairly linear output power versus drive current characteristic [24].

Second, a better detecting system is needed in terms of broadening the system's dynamic range and its linear

operating range defined in Figure 5.9. The detector should be capable of handling the dynamic ranges of each of the components of Figure 5.1. Otherwise, the system's potential for recalling continuous-valued library vectors is weakened.

In Table 6.1 we present a comparison of the four APNN models. The performance of each type is graded with a number; 1 and 3 denote superior and inferior performance in a particular area, respectively. For the optical implementations, spatial resolution is a less significant measurement since the dimensions of the LEDs currently limit the number of neurons that the system can process. The "sturdiness" category takes into account the size and weight of the system. For readers familiar with optical processing, a familiar conclusion will be made: until a low-cost spatial light modulator with a high contrast ratio and spatial resolution are developed, electronic implementations, such as the one described in Chapter III, will generally give better results. Since a system with a large number of neurons requires a high-dimension matrix-vector multiplication, emphasis should be shifted from parallel spatial modulation to a time-division multiplexing scheme where partial products are generated sequentially instead of in parallel. The tradeoff, of course, is speed of operation, but the physical dimensions of the spatial light modulators required by a large number of neurons can be limited.

Table 6.1. A Qualitative Comparison of APNN Machines.

	Computer Simulation	Implementation		
		Electronic	Passive Optical	Active Optical
Spatial Resolution	1 [†]	1 [†]	3 ^{††}	2 ^{††}
Grey Level Resolution	2	1	3	2
Speed of Operation	3	2	1	1
Cost	3	1	1	3
Program- mability	1	3	3	1
Sturdiness	2	1	3	3

Grading Scale: 1-superior
3-inferior

† The number of neurons that the system can process is unlimited.

†† Spatial resolution is limited by the LED dimensions.

REFERENCES

1. Lippman, R.D., "An Introduction to Computing With Neural Nets," IEEE ASSP Magazine, 4, April 1987.
2. McCulloch, W.S. and Pitts, W., "A Logical Calculus of the Ideas Imminent in Nervous Activity," Bulletin of Mathematical Biologists, 5, 115-133, 1943.
3. Marks, R.J., Oh, S., and Atlas, L.E., "Alternating Projection Neural Networks," IEEE Trans. on Circ. and Systems, 851, June 1989.
4. Stark, H., Image Recovery, Academic Press, New York, 1987.
5. Von Neumann, J., The Geometry of Orthogonal Spaces, Princeton U. Press, Princeton, N.J., 1950.
6. Strang, G., Introduction to Applied Mathematics, Wellesley-Cambridge Press, Wellesley, MA, 1986.
7. Goodman, J.W., Dias, A.R., and Woody, L.M., "Some New Methods for Processing Electronic Image Data Using Incoherent Light," Proc. ICO-II Conf., Madrid, Spain, 1978.
8. Ittycheriah, A.P., Walkup, J.W., and Krile, T.F., "An Optical Matrix-Vector Multiplier," (in press IEEE trans. on Education, 1989).
9. Dias, R.A., "Incoherent Matrix-Vector Multiplication for High Speed Data Processing," Technical Report No. L722-4, June, 1980.
10. Athale, R.A., "Optical Matrix Algebraic Processors: A Survey," Proc. 10th Int. Opt. Comp. Conf., IEEE Computer Society Press, IEEE Cat. No. 83CH1880-4.
11. Hamamatsu Silicon Photocell Catalog No. SC-3-6, 1983.
12. Keiser, G., Optical Fiber Communications, McGraw-Hill Book Company, New York, N.Y., 1983.
13. Intel IAPX 88 Book No. 210200-002, Reston Publishing Co., Reston Va, 1983.

14. Yeh, P., and Chiou, A.E.T., "Optical Matrix-Vector Multiplication Through Four-Wave Mixing in Photorefractive Media," Optics Letters, 138, February 1987.
15. Huignard, J.P., "Wave mixing in Photorefractive Bismuth Silicon Oxide Crystals and Its Applications," Optical Engineering, 586, July/August, 1985.
16. Bleha, W.P., Lipton, L.T., Wiener-Avenar, E., Grinberg, J., Reif, P.G., and Casasent, D., "Application of the Liquid Crystal Light Valve to Real-Time Optical Data Processing," Optical Engineering, 371, July-August, 1978.
17. Grinberg, J., Bleha, W.P., Braatz, P.O., Chow, K., and Close, D.H., "Liquid-Crystal Electro-Optical Modulators for Optical Processing of Two-Dimensional Data," SPIE Vol. 128, 253, 1977.
18. Montgomery, G.P., "Optical Properties of a Liquid-Crystal Image Transducer at Normal Incidence: Mathematical Analysis and Applications to the Off-State," J. Opt. Soc. Am., 287, March, 1980.
19. Montgomery, G.P., "Optical Properties of a Liquid-Crystal Image Transducer at Normal Incidence: On-State," J. Opt. Soc. Am., 843, July, 1980.
20. Soref, R.A., and Rafuse, M.J., "Electrically Controlled Birefringence of Thin Nematic Films," J. Appl. Phys, 2029, May, 1972.
21. Ross, W.E., Psaltis, D., and Anderson, R.H., "Two-Dimensional Magneto-Optic Spatial Light Modulator for Signal Processing," Optical Engineering, 486, July/August, 1983.
22. Davis, J.A., Gamlieli, J., and Bach, G.W., "Optical Transmission and Contrast Ratio Studies of the Magneto-optic Spatial Light Modulator," Applied Optics, 5194, December 15, 1988.
23. Sedra, A.S., and Smith, K.C., Micro-Electronic Circuits, Holt, Rinehart and Winston, New York, N.Y., 1982.
24. Sharp Laser Diode User's Manual, Ref. No. HT493D, September, 1986.

APPENDIX

PRINT "THIS PROGRAM MODELS THE APNN"

```
REM  PROCEDURE:      ENTER LIBRARY VECTORS FROM THE KEYBOARD,
REM                  DESIGNATE THE CLAMPED NEURONS,
REM                  AND THE PROGRAM CALCULATES THE STATE
REM                  VECTOR AFTER EACH ITERATION.
```

```
REM  VARIABLE  LIST:      P-# of rows of  $F_p$ 
REM                        E-dot prod of error vector
REM                        C()-denotes  $s$  as clamped, floating
REM                        M-counting index for X
REM                        S()-stimulus vector
REM                        N-# of iterations
REM                        X-# of columns of  $F_p$ 
REM                        T()-interconnection matrix
REM                        F()-library matrix
REM                         $F_p()$ -library matrix
REM                        corresponding to clamped neurons
REM                         $NFP() = F_p^T(\text{transpose}) * F_p$  to check if
REM                         $F_p$  is full column rank
REM
```

INPUT "HOW MANY NEURONS"; L

```
DIM  T(L,L), I(L,L), E(L), S(100,L), FP(L,L), NFP(L,L)
DIM  B(2*L,2*L), F(L,L), C(L)
```

X=0

```
100  INPUT      "DO YOU WISH TO ENTER A LIBRARY VECTOR"; X$
IF X$ = "N" GOTO 200
X=X+1
PRINT "ENTER THE LIBRARY VECTOR"
FOR J=1 TO L
  INPUT F(J,X)
NEXT J
```

REM CALCULATE THE INTERCONNECTION MATRIX T

```
FOR J= 1 TO L
  I(J,J) = 1
  E(J) = 0
NEXT J
```

```
FOR J = 1 TO L
FOR K = 1 TO L
```

```

        E(J)=E(J) + ( I(J,K)-T(J,K) ) * F(K,X)
NEXT K
NEXT J
E=0
FOR J = 1 TO L
    E = E + E(J) * E(J)
NEXT J

FOR J = 1 TO L
FOR K = 1 TO L
    T(J,K) = T(J,K) + E(J) * E(K) / E
NEXT K
NEXT J

GOTO 100
PRINT

200 PRINT"ENTER THE STIMULUS VECTOR AND 1 IF CLAMPED,0 ELSE"
PRINT"FOR EXAMPLE : (4.5,1)"
    N=0
    M=0
    P=0
FOR J = 1 TO L
    INPUT S(N,J) , C(J)
    IF C(J) = 1 THEN P=P+1
NEXT J
GOTO 1000

REM FIND  $F_p$  TO VERIFY THAT IT IS NONSINGULAR
FOR J = 1 TO X
FOR K = 1 TO L
    M = M + 1
    IF C(K) = 1 THEN FP(M,J) = F(K,J) ELSE M=M-1
NEXT K
M=0
NEXT J

REM FIND  $F_p^T$ (transpose)* $F_p$  AND USE FOR SINGULARITY CHECK
FOR I = 1 TO P
FOR J = 1 TO X
    NFP(I,J)=0
FOR K = 1 TO P
    NFP(I,J)=NFP(I,J) + FP(K,I)*FP(K,J)
NEXT K
NEXT J
NEXT I

REM SINGULARITY CHECK
REM FIND INVERSE OF NFP WITH GAUSS ELIM: if det=0 ,THEN SING
FOR I=1 TO X
FOR J=1 TO X

```

```

        B(I,X+J)=0
        B(I,J)=NFP(I,J)
    NEXT J
        B(I,X+1)=1
    NEXT I
    FOR K=1 TO X
        IF K=X THEN 300
        M=K
        FOR I=K+1 TO X
            IF ABS(B(I,K))>ABS(B(M,K)) THEN M=I
        NEXT I
        IF M=K THEN 300
        FOR J=K TO 2*X
            B=B(K,J)
            B(K,J)=B(M,J)
            B(M,J)=B
        NEXT J
        300    FOR J=K+1 TO 2*X
            IF B(K,K) = 0 THEN 450
            B(K,J)=B(K,J)/B(K,K)
        NEXT J
        IF K=1 THEN 350
        FOR I=1 TO K-1
            FOR J=K+1 TO 2*X
                B(I,J)=B(I,J)-B(I,K)*B(K,J)
            NEXT J
        NEXT I
        IF K=X THEN 400
        350    FOR I=K+1 TO X
            FOR J=K+1 TO 2*X
                B(I,J)=B(I,J)-B(I,K)*B(K,J)
            NEXT J
        NEXT I
    NEXT K
    400    FOR I=1 TO X
        FOR J=1 TO X
            B(I,J)=B(I,X+J)
        NEXT J
    NEXT I

```

REM END OF SINGULARITY CHECK

```

1000 REM FIND THE NEXT STATE S(M+1)
INPUT"HOW MANY ITERATIONS";N
FOR I = 0 TO N-1
    FOR J = 1 TO L
        S(I+1,J) = 0
    FOR K = 1 TO L
        S(I+1,J) = S(I+1,J) + T(J,K)*S(I,K)
    NEXT K
NEXT J

```

```

REM CLAMPING APPROPRIATE NEURONS
FOR J = 1 TO L
    IF C(J) = 1 THEN S(I+1,J) = S(0,J)
NEXT J
NEXT I
PRINT" EXTRAPOLATED LIBRARY VECTOR,ROUNDED TO THOUSANDTHS : "
FOR J = 1 TO L
    PRINT INT(1000*S(N-1,J) + .5)/1000
NEXT J

PRINT"DO YOU WISH TO CONTINUE ? 1-ADD A LIBRARY VECTOR "
PRINT"                                2-ENTER A NEW STIMULUS"
PRINT""
INPUT Z
    IF Z = 1 GOTO 100
    IF Z = 2 GOTO 200
GOTO 500
450    IF B(K,K) = 0 THEN PRINT "Fp NOT FULL RANK"
GOTO 200
500    INPUT    "ENTER <RETURN> TO EXIT";V

```

PERMISSION TO COPY

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Texas Tech University, I agree that the Library and my major department shall make it freely available for research purposes. Permission to copy this thesis for scholarly purposes may be granted by the Director of the Library or my major professor. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my further written permission and that any user may be liable for copyright infringement.

Disagree (Permission not granted)

Agree (Permission granted)

Student's signature

Alan Smith

Student's signature

Date

7/13/89

Date