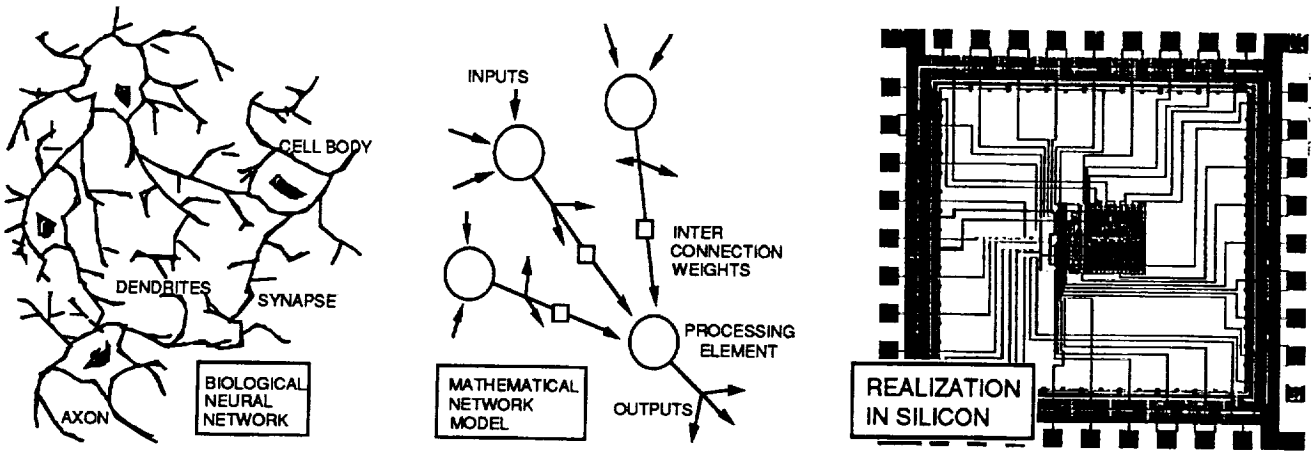
JPL Publication 94-10

A Decade of Neural Networks: Practical Applications and Prospects

May 11-13, 1994

WORKSHOP PROCEEDINGS



Center for Space Microelectronics Technology
Jet Propulsion Laboratory, California Institute of Technology
Pasadena, California

C S M T

This publication was prepared by the Jet Propulsion Laboratory, California Institute of Technology. It was sponsored by the Ballistic Missile Defense Organization, the Army/All Source Analysis System Project Office, the Communication and Electronic Command/Intelligence and Electronic Warfare Directorate, the Naval Surface Warfare Center, and the Office of Naval Research through agreements with the National Aeronautics and Space Administration.

Workshop Proceedings

Terry Cole, Conference Chair, JPL

Sabrina Kemeny, Organizing Committee Chair, JPL

Pat McLane, Local Arrangements Chair, JPL

JPL Organizing Committee

Jacob Barhen

Tien-Hsin Chao

Taher Daud

Daniel Erickson

Sandeep Gulati

Raoul Tawel

Anil Thakoor

Splinter Group Chairs

Azad Madni, Intelligent Systems Technology Inc.

Ken Marko, Ford Motor Company

Demetrios Sapounas, Naval Surface Warfare Center

James Villareal, Johnson Space Center

SPONSORED BY

Ballistic Missile Defense Organization

Army/All Source Analysis System Project Office

Communication and Electronic Command/Intelligence and Electronic Warfare Directorate

Naval Surface Warfare Center

Office of Naval Research

National Aeronautics and Space Administration

Foreword

Welcome to the JPL Neural Network Workshop. Sponsored by NASA and DoD, this workshop brings together sponsoring agencies, active researchers, and the user community to formulate a vision for the next decade of neural network research and application prospects. While the speed and computing power of microprocessors continue to grow at an ever-increasing pace ushering in the era of information supertraffic, the demand to intelligently and adaptively deal with the complex, fuzzy, and often ill-defined world around us remains to a large extent unaddressed. Powerful, highly parallel computing paradigms such as neural networks promise to have a major impact in addressing these needs.

The theme of the workshop is on practical applications. To this end, the workshop begins with a series of invited talks focusing on a variety of applications both in control and signal processing. Following the presentations, we will split into working groups to formulate a road map for future R&D. The splinter groups will identify key application areas for the future and address issues such as technology insertion.

In order to promote the cross-fertilization of ideas and seed discussion, two social events have been planned at the Pasadena Hilton. On Wednesday evening, there will be a welcome reception with hors d'oeuvres and a cash bar at the Hilton patio. On Thursday evening, a sit-down dinner will be served in the Monterey room.

Abstracts and excerpts of presentation materials from the invited talks are included in this booklet. A final report summarizing the workshop and splinter group findings will be published later.

Thank you for your participation in what promises to be an interesting and timely forum.

CONTENTS

Workshop Program

Wednesday, May 11, 1994

12:00 - 2:00 **Registration** **JPL Visitor Center**

Welcome **Location: JPL 180-101** **1:00 - 1:35**

1:00 - 1:10 Welcoming Remarks: *Terry Cole, JPL*
1:10 - 1:30 Keynote Address: *Dwight Duston, BMDO*
1:30 - 1:35 Workshop Logistics: *Sabrina Kemeny, JPL*

Session 1 **Location: JPL 180-101** **1:35 - 5:45** **Chair: *Terry Cole, JPL***

1:35 - 3:05
Missileborne Artificial Vision System (MAVIS)
David K. Andes, James C. Witham, and Michael D. Miles, NAWC
Application of Adaptive Learning to Diagnostics: The Role of Neural Networks in Developing Practical Solutions to Two Major Problems
Kenneth A. Marko, Ford Motor Company
Document Analysis with Neural Net Circuits
Hans Peter Graf, AT&T Bell Laboratories
From Neural-Based Object Recognition toward Microelectronic Eyes
Bing J. Sheu and Sa Hyun Bang, University of Southern California

3:05 - 3:25 BREAK

3:25 - 5:20
VLSI Neuroprocessors
Sabrina Kemeny, Jet Propulsion Laboratory
PHOTONICS: From Target Recognition to Lesion Detection
Michael Henry, Martin Marietta Corporation
3D Artificial Neural Network (3DANN) Technology: A Status Report and Blueprint for the Future: *John Carson, Irvine Sensors Corporation*
Hidden Markov Models and Neural Networks for Fault Detection in Dynamic Systems
Padhraic Smyth, Jet Propulsion Laboratory
Innovation and Application of ANN in Europe Demonstrated by Kohonen Maps
Karl Goser, University of Dortmund

Evening **Location: Pasadena Hilton Hotel - Patio, 150 S. Los Robles, Pasadena**

6:30 - 7:30 Welcome Reception - hors d'oeuvres and cash bar

Thursday, May 12, 1994

Session 2 **Location: JPL 180-101** **8:00 - 12:30** **Chair: *Kris Koliwad, JPL***

8:00 - 8:20 **Keynote Address**: "Neural Nets and Intelligent Control: A Strategic Perspective"
Paul Werbos, NSF

8:20 - 10:15 Neural Network Classification of Clinical Neurophysiological Data for Acute Care Monitoring, *Joseph Sgro, Alacron Inc.*
Application of Neural Networks to Unsteady Aerodynamic Control
William E. Faller, USAF Academy and University of Colorado,
Scott Schreck, USAF Academy, and Marvin Luttges, University of Colorado
Smart Vision Chips: An Overview
Christof Koch, California Institute of Technology
Predictability in Space Launch Vehicle Anomaly Detection Using Intelligent Neuro-Fuzzy Systems, *Sandeep Gulati et al., Jet Propulsion Laboratory*
Neural Networks: Application to Medical Imaging
Laurence Clarke, University of South Florida

10:15 - 10:35 BREAK

10:35 - 12:30 Learning Random Networks for Compression of Still and Moving Images,
Erol Gelenbe, Mert Sungur, and Christopher Cramer, Duke University
Learning to Train Neural Networks for Real-World Control Problems
Lee Feldkamp, G.V. Puskorius, L.I. Davis, Jr., and F. Yuan, Ford Motor Company
An Integrated Optoelectronic ATR Processor
Tien-Hsin Chao, Jet Propulsion Laboratory
Neural Network Applications in Telecommunications
Joshua Alspector, Bellcore
A Neural Network Controller for Automated Composite Manufacturing
Peter F. Lichtenwalner, McDonnell Douglas Aerospace

12:30 - 2:00 **LUNCH (free time)**

Session 3 **Location: Pasadena Hilton Hotel, 150 S. Los Robles, Pasadena**
Splinter Group Discussions **2:00 - 6:00**

2:00 - 2:10 Instructions to splinter groups, Pacific Ballroom
2:10 - 6:00 Control A, Sacramento
Control B, San Jose
Signal and Image Processing A, San Diego
Signal and Image Processing B, Santa Monica

Evening **Location: Monterey room, Pasadena Hilton Hotel, 150 S. Los Robles, Pasadena**

7:00 Dinner, Monterey room
7:45 **Dinner Speaker:** "How Captain Amerika Uses Neural Networks to Fight Crime"
Steven Rogers, Air Force Institute of Technology

FRIDAY, MAY 13, 1994

Session 4 **8:00 - 10:00**
Location: **Pacific Ballroom, Pasadena Hilton Hotel, 150 S. Los Robles, Pasadena**

8:00 - 8:40 Splinter Group Chair Reports
8:40 - 9:55 Sponsor/Industry Panel Assessment
9:55 - 10:00 Closing Remarks, *Sabrina Kemeny, JPL*

P-10

Missileborne Artificial Vision System (MAVIS)

David K. Andes, James C. Witham, Michael D. Miles

Naval Air Warfare Center - Weapons Division

China Lake, CA 93555

ABSTRACT

Several years ago when INTEL and China Lake designed the ETANN chip, analog VLSI appeared to be the only way to do high density neural computing. In the last five years, however, digital parallel processing chips capable of performing neural computation functions have evolved to the point of rough equality with analog chips in system level computational density. The Naval Air Warfare Center, China Lake has developed a real time, hardware and software system designed to implement and evaluate biologically inspired retinal and cortical models.

The hardware is based on the Adaptive Solutions Inc. massively parallel CNAPS system COHO boards. Each COHO board is a standard size 6U VME card featuring 256 fixed point, RISC processors running at 20 MHz in a SIMD configuration. Each COHO board has a Companion board built to support a real time VSB interface to an imaging seeker, a NTSC camera and to other COHO boards. The system is designed to have multiple SIMD machines each performing different Corticomorphic functions.

The system level software has been developed which allows a high level description of Corticomorphic structures to be translated into the native microcode of the CNAPS chips. Corticomorphic structures are those neural structures with a form similar to that of the retina, the lateral geniculate nucleus or the visual cortex.

This real time hardware system is designed to be shrunk into a volume compatible with air launched tactical missiles. Initial versions of the software and hardware have been completed and are in the early stages of integration with a missile seeker.

INTRODUCTION

The onboard processing requirements of air intercept missiles are some of the most demanding imaginable. This is especially true for missiles with imaging focal plane array detectors. Input is measured in megabytes per second. The volume available is a few cubic inches. Decisions are required in milliseconds. The power available is just a few watts and heat dissipation is minimal. Then the system must live in an environment that includes salt air, desert heat, Arctic conditions, high humidity and rapid altitude changes. Aircraft systems have similar constraints but the power, volume and heat dissipation problems are slightly less severe. If we are to survive in a competitive world, however, we must continue to upgrade the internal intelligence of our systems.

Biological systems have met and overcome even greater competitive challenges in real-time embedded computing. Biosystems have similar constraints in power, volume, heat dissipation while requiring high speed computation including high data rate sensors of several varieties. There should be much to learn from the many, highly successful, integrated, real-time biocomputers that surround us every day. The MAVIS project is an attempt to do just that.

Biological Computation Systems

The following is a partial list of some of the salient characteristics of biological computation systems:

1. *Massive parallelism* is the first obvious characteristic. We cannot hope to come even close to the biosystems in this area but at least it gives a definite direction in which to move. Many simple processors working almost independently can clearly achieve great results.

2. Most biocomputation is based only on *locally available information.* Transmitting information beyond a few tenths of a millimeter becomes very expensive.

3. There is a *lack of emphasis on precision* in the elementary processors (neurons). In the cases where more precision is necessary more elementary processors are dedicated to the task.

4. Local computational centers share information with *several* other local centers in a *bi-directional* manner. Computation is shared in a non-hierarchical or only a semi-hierarchical manner. In fact most of the information entering the local processing centers is not raw sensor data but partially processed information from other local centers.

5. The computational components of biosystems are *finely tuned* parts of a whole system. Competition has not allowed much that is inefficient or unnecessary. The processing devoted to sensor data is well matched to the quality and importance of the information.

Corticomorphic Processing

The mammalian vision system has some special structural characteristics which are clearly specialized for the processing of two dimensional image information. An abstraction of the form of this system is used in the MAVIS project and has been given the name Corticomorphic Processing. Although this model is an abstraction of the processing centers of the visual system (such as the retina and patches of visual cortex) it is hoped that models of other areas of the cortex will fit into this general form. The Corticomorphic abstraction is an Artificial Neural Network (ANN) though not of one of the standard forms (e.g. Backpropagation, ART, Hopfield, etc.).

The early processing stages of the visual system (areas like the retina, the Lateral Geniculate Nucleus, primary visual cortex, V2, V3, etc.) have computational forms which are similar. Each area is a "patch" of computational elements laid out in a form which preserves, at least locally, the two dimensional relationships in the original image. Within each of the patches there are various types of neurons arranged in sheets or layers that run throughout the entire patch. Even though the neurons on different sheets perform very different functions the rough topology of the original image is preserved in each sheet. A column cut vertically into a patch through all the sheets will find neurons which only respond to a small local area of the original image. Inputs into each sheet of a patch come in through topology preserving maps from other sheets. Most inputs into a sheet are from sheets within the same patch but some come from sheets within other patches. The strengths of the interactions between neural processing elements can be approximated by the mathematical form of convolution kernels. This is an approximation that is only locally true in real biosystems since it requires exactly the same processing to take place throughout the entire length and width of a patch.

Formalism

The introduction of some formalism may make all this more precise if not clearer. Let

$$O(x,y,i,j,t)$$

be the output value of the neural processing element at the (x,y) position of the image space in the i-th layer of the j-th patch at time t. Then

$$L(m,n) = \{ O(x,y,i,j,t) \} \text{ for } i=m \text{ and } j=n$$

is the m-th sheet or layer in the j-th patch. Note that L(m,n) is a set of neural processing elements. Note also that we have shifted from the more descriptive word "sheet" to the more traditional ANN term "layer". Then let

$$P(i) = \{ L(m,k) \}\ k=i$$

be the i-th patch. Note that P(i) is a set of layers.

Typically the number of layers in a patch runs from three to ten and only a few of the layers in a patch have outputs to layers in other patches. The output value of the neural processing elements of a layer L(i,j) is calculated as follows:

$$O(x,y,i,j,t) = F_{i,j} (\Sigma (a_{i,j,s,p} + g_{i,j,s,p} \Sigma k_{i,j,s,p}(l,m) O(x-l,y-k,s,p,t-b_{i,j,s,p}))) \qquad (1)$$

The first sum is a sum over s and p where p runs over all patches driving this layer L(i,j) and s runs over all layers in p which connect to the layer L(i,j). The second sum is also a double sum over l and m which run through enough positive and negative integers to cover the kernel $k_{i,j,s,p}$.

In this expression:

$F_{i,j}$ is the nonlinear function associated with the neural processing elements of the layer L(i,j).

$k_{i,j,s,p}$ is the kernel weight function which determines the effect of the L(s,p) layer on the L(i,j) layer.

$b_{i,j,s,p}$ is either zero (no time delay) or one (one time step delay) depending on whether the information affecting L(i,j) from L(s,p) is to be current or delayed.

$a_{i,j,s,p}$ and $g_{i,j,s,p}$ are appropriate offset and gain numbers affecting the action of layer L(s,p) on layer L(i,j).

In plain English this amounts to the following: each layer in each patch is calculated by applying a set of kernel convolutions to one or more other layers, summing the results and then passing it through a possibly non-linear function. Gains, offsets and time delays may be applied where necessary.

Although the sums look complex they typically contain only one to three kernel interactions with most of the interactions occurring within the same patch (i.e. j=p). In fact a layer may interact with itself in which case j=p and i=s and $b_{i,j,s,p}$ must be one. This self interaction allows for temporal integration (both point and area).

One more basic construct is useful and that is the idea of a column. Let

$$C(u,v,p)$$

be the symbol for the column centered on the point (u,v) in image space on patch p. Then if

$$R_x(C) \text{ and } R_y(C)$$

are the x and y radii of the column we have

$$C(u,v,p) = \{ O(x,y,u,v,t) \varepsilon L(i,j) \text{ such that } |x-u| < R_x(C) \text{ and } |y-j| < R_y(C) \} \qquad (2)$$

That is a column is the set of all points (outputs of neural processing elements) in pieces of sheets (or layers) from a single patch which are all cut to the same size and all of which are centered at the same place in image space. Note that for C(u,v,p) the values of u, v, $R_x(C)$, $R_y(C)$ need not be integers.

History of Embedded Neurocomputing at China Lake

For the past fifteen years the Office of Naval Research has been funding work at China Lake with the aim of increasing the capability of embedded computational systems for air intercept weapons. Most of the work described in this paper was done under this ONR funding although a significant portion of the early work in several of the areas was started under local funding at China Lake.

In the early 1980's it became clear that traditional Artificial Intelligence techniques had only limited utility for embedded real-time systems in air intercept missiles. This was due mostly to the inability of the hardware of the time to match the severe constraints imposed by these systems. In the mid 1980's the

biologically inspired field of Artificial Neural Networks showed promise of helping to overcome this computational bottleneck. The ideas were amenable to implementation in high speed, parallel, analog circuitry and learning algorithms could be used to circumvent the problems associated with analog imprecision. Early experiments and designs at China Lake led to the development of the Intel ETANN chip [1]. This chip is capable of about three billion operations per second in a fraction of a square inch.

In 1989 the Missileborne Artificial Neural Network Demonstration (MINND) program was initiated to exploit the availability of the new computational power. The MINND program was successfully completed in 1992 with real time demonstrations on real air targets [2]. The architecture of the MINND computer allowed a simple version of the Corticomorphic Processing scheme to be implemented. The fixed form of the analog circuitry, however, put rigid constraints on the types of computations that could be performed. Toward the end of the MINND program it became clear that digital computation was catching up to the analog when total system level computational density was considered. In particular the Adaptive Solutions CNAPS chip [3] had characteristics that allowed us to design the current MAVIS system. MAVIS has system level performance similar to the ETANN based MINND system but without the associated analog problems. Packaging techniques are available which allow the design of the MAVIS system to be reduced enough to fit the constraints of an air intercept missile. The sections of this paper that follow describe the hardware and software components of the MAVIS system.

MAVIS HARDWARE OVERVIEW

The MAVIS system is built around the Adaptive Solutions CNAPS chip. Each chip has 64 fixed point, RISC processors that currently operate at 20 MHz. These processors are designed to operate in an SIMD configuration where several CNAPS chips may be under the control of a single sequencer chip [4]. Each of the 64 processing nodes (PNs) on each CNAPS chip has an adder, a multiplier, a logic unit, 4K bytes of local memory, several general purpose registers, and inter-PN bussing. The system uses the Adaptive Solutions COHO boards [5] each of which mounts four CNAPS chips for a total of 256 PNs per board. The MAVIS system is designed to accommodate several of these COHO boards each of which is used to implement one patch of Corticomorphic processing. A high speed bus intercommunication scheme has been designed to allow high bandwidth injection of sensor data as well as high bandwidth inter-patch communication.

An overview of the initial MAVIS system can be seen in Figure 1. It shows an imaging seeker connected to the MAVIS card cage, a Motorola MVME-147 board (68030 processor), two Adaptive Solutions Inc. COHO boards, two NAWC designed COHO Companion boards, and a NAWC designed Custom I/O board. The diagram also shows two video display monitors and two VCRs used for displaying and recording raw and processed video.

Adaptive Solutions Inc. has a set of integrated tools that can be used to develop and debug code for their COHO board by using a SUN SPARC station connected to the MVME-147 via an ethernet network. Code is developed and compiled on the SUN workstation and then downloaded to the COHO board to run.

Hardware Specifics

COHO Board

The COHO board is a commercially available 6U VME board. The major components of the board are highlighted in Figure 2.

The board has provisions for attaching peripheral devices or memory onto its local bus. The name of this local bus is the CNAPS/VME local bus (CVLB). The CVLB is an implementation of the company's ADAPTbus™ applied to this specific board and its peripherals. There is a 100 pin impedance matched connector on the COHO board which provides access to the CVLB. It is this connector that the COHO board uses to interface to the COHO Companion board.

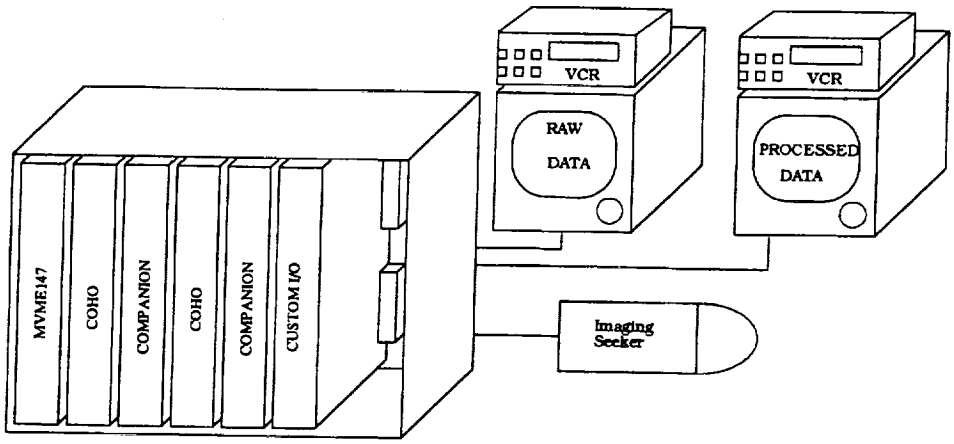


Figure 1

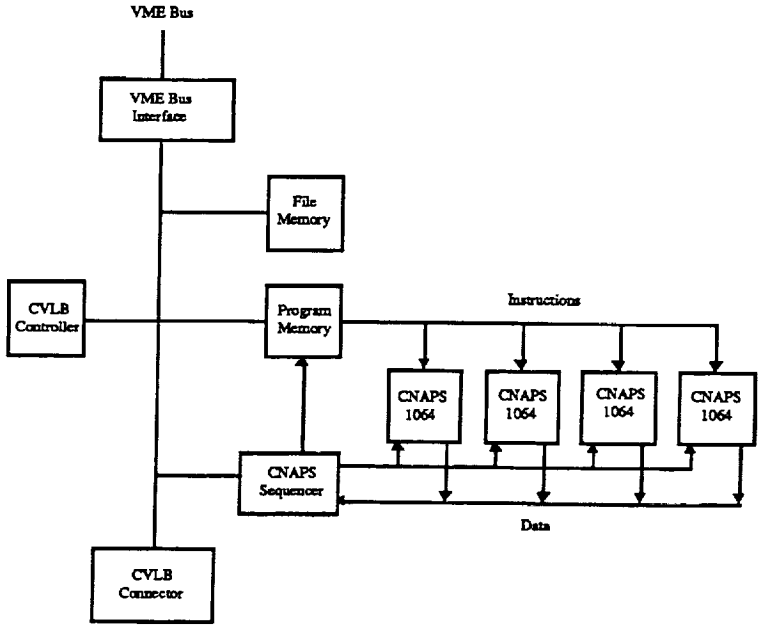


Figure 2

COHO Companion Board

A block diagram for the COHO Companion board is shown in Figure 3. This architecture, made up of two ping-pong memories, was chosen because it allowed images to be read from or written to both memories simultaneously. For instance, as an incoming image is being written into Bank 1, an image can be read out of Bank 2, processed and then written back to Bank 2 without impeding the incoming image. When both tasks are finished the memories are swapped, so that the image in Bank 1 may be processed while a new incoming image may be written into Bank 2.

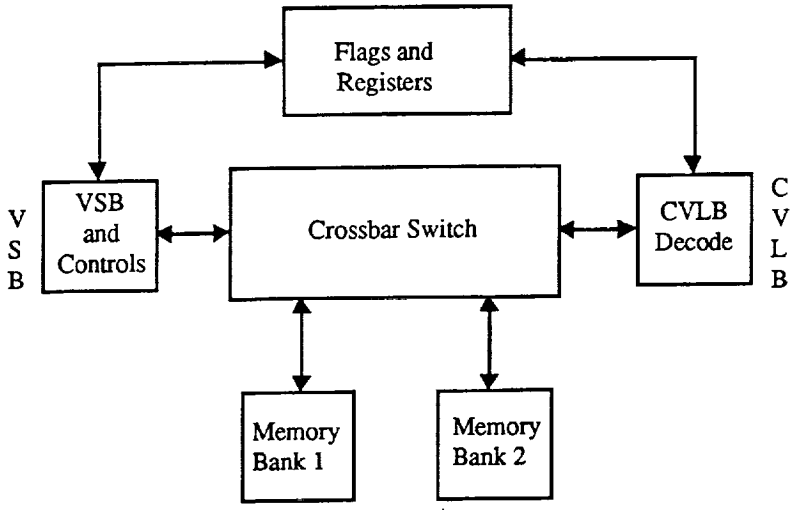


Figure 3.

If one assumes an image patch of 128 by 128 and a frame rate of 60 frames per second the amount of data that is actually passed into the system is approximately 1 MByte per second. With the MAVIS system setup, data is processed on each COHO board (patch) and is available for display only when sent over an interconnection bus. Thus under these assumptions with only a single COHO/COHO Companion board pair the final I/O requirements are only about 2 MBytes/sec. When more than a single pair of boards are used, however, there will be interaction between boards and, with more interaction, more bus bandwidth is required. If larger images or higher video rates are required the bus bandwidth also increases. For these reasons, it was decided to offload the data from the VME bus and use the VSB bus (VME Subsystem Bus). The current implementation is able to move data at 12 MBytes/second over the VSB. Figure 4 shows the buses and the type of data that is transferred on each bus.

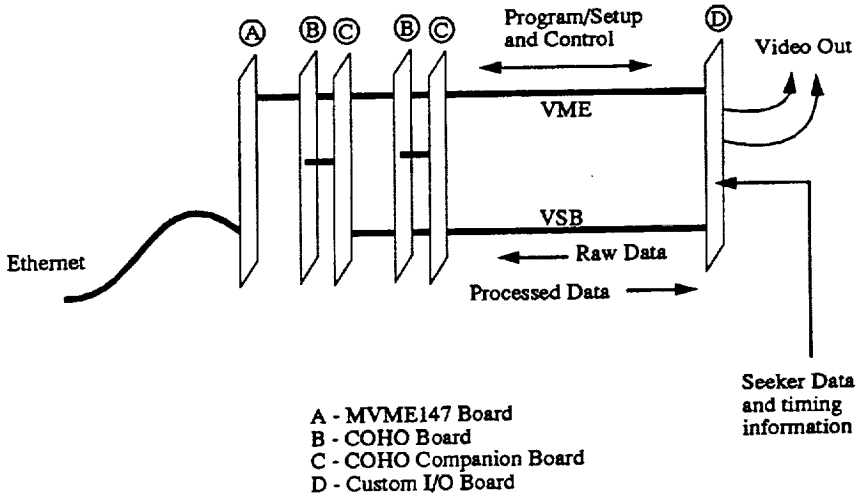


Figure 4

Custom I/O Board

The Custom I/O board was fabricated to comply with the digital video and timing signals for an imaging seeker. The board is also capable of displaying the incoming digital video, plus an extra video channel that

may be used to show the results of processed or intermediate data It is also capable of selecting an Area Of Interest (AOI) of variable size and location, from the incoming video, and transmitting it on the VSB Bus.

As shown in Figure 5 the system is based around a pair of dual ported memories, one for the input, and one for the output. The output video frame's timing is in lock step with the input video frame's timing. This feature could be used to reinsert the processed digital video back into the data stream that it was taken from.

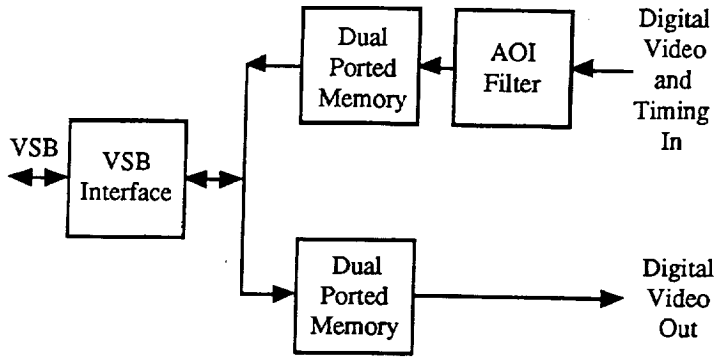


Figure 5

System Options

Having the MAVIS system tied directly to a real missile seeker has many advantages for answering questions related directly to that particular system. There are, however, many disadvantages associated with such a system. A second system option is also being implemented which is much more general than the single seeker system described above. The second system uses a pan/tilt unit with a camera mounted to it in place of the imaging seeker. Several additional boards are required to interface to a camera with a pan/tilt unit: a frame grabber/display board, a D/A (Digital to Analog) board, and a single board computer (SBC). A general purpose microprocessor on the SBC receives information from the COHO board with a target location and generates the angle rates for the pan/tilt unit and sends them out via the D/A board. The microprocessor can also take slave commands from a joystick for external target designation.

MAVIS SOFTWARE OVERVIEW

The system level software is designed to combine flexibility with ease of use in the implementation of a variety of Corticomorphic structures. The system level software is written in C and takes a text file containing Corticomorphic descriptors and produces microcode which is native to the CNAPS processors.

The first step in implementing a Corticomorphic concept is to develop a block diagram of the system to be modeled. Figure 6 shows a relatively simple model of the outer retina. The model itself is broken up into several layers. These layers themselves are idealized models of distinct types of retinal neurons. The boxes labeled with the capital letter K and a number refer to the kernel which will be used in the convolutional interaction between the layers. A kernel is a square matrix made up of integer weights designed to have a specific effect, such as edge enhancement or smoothing.

As shown in equation (1) the creation of each layer is dependent upon several things: the other layers in the model, the kernels with which the layers will be convolved, and the method of combining the results. The software allows for simple definitions of feedback paths both from a layer further along in the model path and from a layer to itself. This self interaction is accomplished by storing a layer in memory when it is created at time t-1, so that it may be used in the creation of a layer at time t.

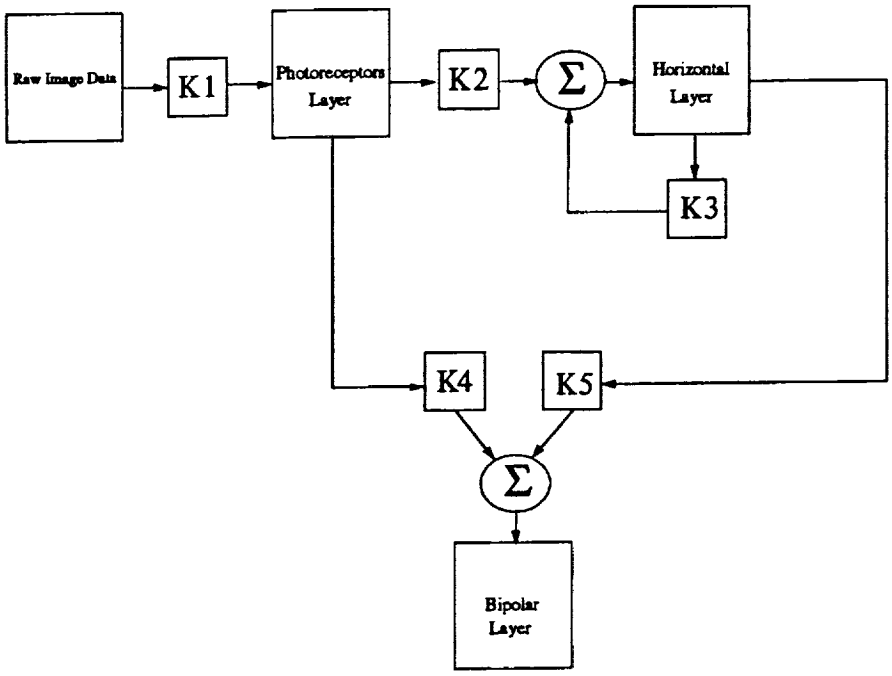


Figure 6

From the block diagram, the user must create a model file and kernel files. A model file is a simple text file containing a description of the elements the user wishes to include in the model. Kernel files are text files containing the dimensions, weights, gains and offsets for a kernel. The system software reads the model file, which references the kernel files as they are needed and uses its' specifications to generate another file containing CNAPS microcode. This microcode is assembled using the CNAPS assembler and then loaded into the COHO program memory space. At this point, the user needs only to assert a start command for the software to assume command of the hardware system.

There are certain details the software must accommodate to implement equation (1). Figure 7 shows the application of a kernel $(k_{i,j,s,p})$ to the intersection of a layer L(i,j) and a column C(u,v,p) as described in equation (2). The pixels surrounding this portion of the column are part of a software construct known as a tile border. As indicated in the figure, the tile border and the column section comprise the tile itself. In order for the kernel to be applied so that the result has the proper correspondence to the pixels along the edges of the column, extra information is required. This extra information is borrowed from neighboring PNs and comprises the tile border. If no tile border was constructed, and the kernel was simply applied as in Figure 8, the result would be the shrinking of the column size as in Figure 9.

The patches referred to in the equations are actually separate COHO boards. The software allows the user to specify which board will act as which patch and which layers the patch will be responsible for processing.

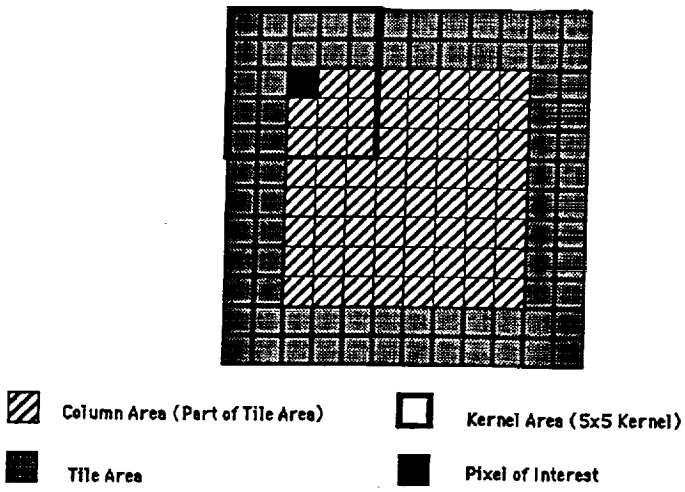


Figure 7

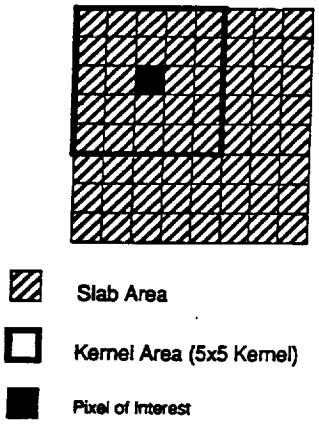


Figure 8

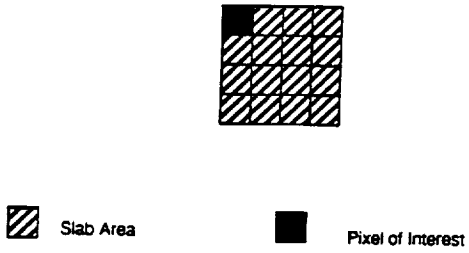


Figure 9

GENERAL NOTES

There are several extensions to the basic Corticomorphic structure which are already planned. None of these require a modification to the form of the hardware.

1. The simplified computational form of equation (1) can be extended to allow the multiplication of convolutions of layers as well as the sum. Sums and products could also be mixed in the same evaluation. This modification has already been tried and is not included in equation (1) mainly because it complicates the formalism and the write-up. Multiplication takes no more time than addition and hence this modification costs nothing in compute time. The same cannot be said of the next two extensions.

2. The terms in the equation (1) which appear as constants (such as kernel weights, gains and offsets) could be made to vary with time since they are stored in memory local to each controller.

3. Time delays of longer than one frame have been implemented. The cost is in local memory and some in compute time.

It is important to note that most of the current image processing schemes (neural net or otherwise) can be put into the form of equation (1) or a minor extension of it as given above. Hence the MAVIS system provides a good real-time test bed for many current image processing ideas.

CONCLUSION

MAVIS is an attempt to produce a computational structure which emulates the form of the processing used in the mammalian vision systems. The eye and the brain are a coupled system which obtains an understanding of the environment by interacting with it. It is hoped that the investigation of this complex interaction will shed light on the functioning of real cortex as well as allowing us to design better sensing systems for both military and non-military applications.

REFERENCES

[1] M. Holler, S. Tam, H. Castro, and R. Benson, "An electrically trainable artificial neural network (ETANN) with 10,240 "floating gate" synapses," in Proc. Int. Joint Conf. Neural Networks (Washington D.C.), 18-22 June 1989, p. II-191.

[2] M. Mumord, D. Andes, and L. Kern, "The Mod 2 Neurocomputer System Design," in IEEE Transactions on Neural Networks, Vol 3., No. 3, May 1992.

[3] D. Hammerstrom, E. Means, M. Griffin, G. Tahara, K. Knorpp, R. Pinkham, and B. Riley, "An 11 Million Transistor Digital Neural Network Execution Engine," in IEEE International Solid-State Circuits Conference, 1991, p. 180-181.

[4] D. Mueller, and D. Hammerstrom, "A Neural Network Component," in IEEE International Conference on Neural Networks (San Francisco), March 1993.

[5] T. Skinner, "Digital Signal Processing on a Multiprocessor System," Electronic Design, Penton Publishing, Cleveland, Ohio, 7 February 1994.

omit

JPL NEURAL NETWORK WORKSHOP "A DECADE OF NEURAL NETWORKS" MAY 1994

ABSTRACT OF ORAL PRESENTATION

APPLICATION OF ADAPTIVE LEARNING TO DIAGNOSTICS: THE ROLE OF NEURAL NETWORKS IN DEVELOPING PRACTICAL SOLUTIONS TO TWO MAJOR PROBLEMS

Kenneth A. Marko
Ford Research Laboratory
Ford Motor Company
Dearborn, Michigan 48121-2054

AN EVOLUTIONARY APPROACH TO PROCESS CONTROL AND DIAGNOSTICS BASED ON ADAPTIVE LEARNING

In previous work, we have examined the application of various Artificial Intelligence (AI) learning paradigms to the problem of diagnosing faults in complex systems in studies to determine whether various learning systems could be properly trained to identify faults in systems under test. The evaluation of these learning paradigms was based upon their performance on large, stable databases which were expected to be fully representative of the data such trained systems would be called upon to classify. These studies therefore proceeded from the assumption that a great deal of information about the systems to be diagnosed was available at the start of the program and that new, incoming information would be very similar to the data upon which the system was trained. In order to develop viable schemes for real applications at manufacturing plants it is necessary to relax these constraints and to construct trainable diagnostic systems when:

1. Very little information from the systems under test is available at the outset of the program.
2. The data from the systems under test changes significantly and in unpredictable ways during the development of the diagnostic system.

3. We wish not only to diagnose faults in the manufactured systems, but also to monitor the manufacturing process to control the quality of the products.

There are several general characteristics of the problem that we can readily identify:

- Our interest is primarily on mechanical faults rather than electronic faults since the products (in this case, automobile engines) at this stage in the manufacturing process are undergoing tests in the absence of their electronic control systems.
- Engines operate only briefly over a restricted range, and all engines are of the same vintage, i.e. this problem is representative of a manufacturing test process rather than a service garage test process, and is, in fact, simpler than the service problem.
- Complete knowledge of all failure modes is not known *a priori,* and new classes of abnormal operation must be identified as data is obtained. Additionally, modifications to the manufacturing process will alter the signature of normal engines on a frequent, but unpredictable, time scale. The system must adapt to these changes as quickly as possible, with the constraint that training data will be very limited, typically a few hundred samples.
- The input data consists of information from only a few sensors, sampled very frequently, making the problem more like pattern recognition in complex waveforms and less like a sensor fusion problem.
- Training data for faulty engines is a tiny fraction of the data available for normal engines and the statistical distributions for very rare abnormalities may never be known very well.
- The diagnostic system must operate continuously, and adapt quickly to changes in the product performance since continuous improvement in the complex manufacturing process must be anticipated.

These characteristics together make the classification problem quite difficult. In particular, our classification system must have a very low false alarm rate, a high accuracy rate for identification of faults, be readily adaptable to changes in the process, and still function as a "novelty" detector to identify engines with new faults not present in training samples. Straightforward application of common learning schemes such as backpropagation in neural networks were not satisfactory for this development program. However, we will demonstrate that a combination of traditional methods and modern learning paradigms, does provide a means of developing a reliable diagnostic system under realistic conditions if we permit the program to evolve as information is gathered. Briefly, our approach is to break the classification task down into modular processes that can be modified to suit each individual application. We utilize traditional classifier systems at the outset, and bring neural networks in later in the process when suitable sample sizes are available. The development of classification systems is also expedited in this process through the use of complexity reduction algorithms such as Principal Component Analysis (PCA) which eliminates the storage and analysis of unneeded or redundant data. Our methods also rely heavily on Monte Carlo simulation to generate statistically representative samples of training data from rather sparse samples of real data. The analysis is applied to engine data obtained from a sample of engines at end-of-line tests conducted as part of a quality assurance program.

DEVELOPMENT OF ON BOARD DIAGNOSTICS FOR EMISSION MONITORING: MISFIRE MONITORS FOR PRODUCTION VEHICLES

The automotive industry is facing a new challenge in meeting regulations mandating that all production vehicles continuously monitor their tailpipe emissions and provide indications to the driver when the vehicles are out of compliance. The task is especially difficult due to the fact that no direct measures of emission gases are available (reliable,

inexpensive sensors have not been developed), so the diagnostics must be inferential. The development of one of the monitors, the misfire diagnostic, provides some insight into how modern adaptive learning methods can be applied to a very complex and demanding task. All auto manufacturers will be introducing hardware and software to meet the statutory requirements beginning this model year. It is useful to note that none of the systems being introduced appear to rely directly on ANS (Artificial Neural Systems) technology. However, at least in our work, ANS methods have played and continue to play an important role in developing means to comply with the legislation. The short development time required for these programs, coupled with the limited capabilities of the on-board microprocessors have certainly had a role in steering the deployed systems away from ANS technology. Yet, these facts do not fully explain why ANS methods are not used in the production systems. Our analysis suggests that "conventional" ANS, in the form of feedforward networks trained by the backpropagation learning schemes, have deficiencies which currently limit the role of these systems in practical applications involving large and complex databases. We have identified several issues which must be addressed and solved before ANS methods can be expected to be employed in developing the solutions to these diagnostic problems. The issues and the identification of possible solutions suggest that ANS methods, properly used, may ultimately provide the best solution to the diagnostic requirements for vehicle systems.

40702

P. 8

JPL NEURAL NETWORK WORKSHOP "A DECADE OF NEURAL NETWORKS" MAY 1994

NEURAL NETWORK APPLICATION TO COMPREHENSIVE ENGINE DIAGNOSTICS

Kenneth A. Marko
Ford Motor Company
Dearborn, Michigan 48121-2054

I. INTRODUCTION

We have previously reported on the use neural networks for detection and identification of faults in complex microprocessor controlled powertrain systems [1,2]. The data analyzed in those studies consisted of the full spectrum of signals passing between the engine and the real-time microprocessor controller. The specific task of the classification system was to classify system operation as nominal or abnormal and to identify the fault present. The primary concern in earlier work was the identification of faults, in sensors or actuators in the powertrain system as it was exercised over its full operating range. The use of data from a variety of sources, each contributing some potentially useful information to the classification task, is commonly referred to as sensor fusion and typifies the type of problems successfully addressed using neural networks.

In this work, we explore the application of neural networks to a different diagnostic problem, the diagnosis of faults in newly manufactured engines and the utility of neural networks for process control. While this problem shares a number of characteristics of the previous studies, there are several significant differences.

- Our interest here is primarily on mechanical faults rather than electronic faults since the engine at this stage in the manufacturing process is undergoing "cold test", i.e. it is connected to an electric dynamometer.
- Engines operate only briefly over a restricted range, and all engines are of the same vintage.
- Complete knowledge of all failure modes is not known a priori, and new classes of abnormal operation must be identified as data is obtained. Additionally, modifications to the manufacturing

process will alter the signature of normal engines on a frequent, but unpredictable, time scale. The system must adapt to these changes as quickly as possible, with the constraint that training data will be very limited.
• The input data consists of information from fewer sensors sampled more frequently, making the problem more like pattern recognition in complex waveforms and less like a sensor fusion problem.
• Training data for faulty engines is a tiny fraction of the data available for normal engines and the statistical distributions for very rare abnormalities may never be known very well.
• We are interested not only in detecting and diagnosing faults, but also in monitoring drifts from nominal in the manufacturing process.

All of these circumstances conspire to make this classification problem quite difficult. In particular, this classification system must have a very low false alarm rate, a high accuracy rate for identification of faults, be readily adaptable to changes in the process and still function as a "novelty" detector to identify engines with new faults not presented in training samples. The simple, brute force application of backpropagation to analysis of raw data did not reliably produce a classifier with these properties. However, the methods we have developed can deal successfully with these circumstances and be applied as well to a wide variety of other classification problems.

Briefly, our approach is to break the classification task down into elemental processes that can be modified to suit each individual application. We choose to utilize traditional classifier systems and neural networks together to obtain optimum performance for this diagnostic problem. The methods also rely heavily on Monte Carlo simulation to generate statistically representative samples of training data from rather sparse samples of real data. These simulations boot-strap information from reasonable assumptions about the underlying statistics which are updated as empirical statistical distributions emerge. Such mathematical artifices permit us to evaluate the expected performance of our classification system early in the development process, before we have an adequate amount of actual data and can be easily adapted to utilize the true statistics of the data.

II. INITIAL STUDIES

Initially we used a 4.0 liter 6 cylinder engine to investigate the feasibility of comprehensive cold test diagnostics on a representative sample of data. Only a single engine was available, and this engine was disassembled and reassembled with deliberately introduced faults to provide the initial database for our investigations. The engine was motored, typically at about 150 rpm, by an electric motor with an in-line torque transducer to measure the dynamic crankshaft torque. Simultaneously, pressure transducers monitored the intake and exhaust manifold pressures, the crankcase air pressure and the oil pressure. Measurements of each parameter were taken every 10 crank angle degrees, and a complete data sample consists of 70 measurements on each trace (2 x 35 samples per revolution due to a 36-1 tooth encoding wheel). Several cycles could be averaged together, but the observed cycle to cycle fluctuations were extremely small and one cycle appeared to be satisfactory. Therefore, the actual data acquisition time for this test was less than 1 second. Typical samples of data from normal and abnormal operation are shown in Figure 1. Visible on these traces are clear features associated with the engine fault, which an expert diagnostician could conceivably use to identify the nature of the fault. These traces were selected to manifest such recognizable features which often lead one to suspect that a simple rule based system could be constructed to perform the diagnostics. However, the engine to engine variability and the need to distinguish not only any one fault from normal operation, but also from all other faults, complicates matters. Closer examination of the traces reveals that in addition to primary discriminating features present at particular points in the trace, additional but smaller correlated features are present elsewhere in the traces. It is desirable to utilize all helpful discriminating features to construct a robust classifier

We used a conventional backpropagation (BP) neural network in a first assault on this problem. However, the raw data from test engine produced an unwieldy test vector with several hundred elements. Data were collected from a test suite of 28 different faults and normal operation (29 classes) and a data base of about 1500 test vectors was obtained. This data was artificially augmented with uncorrelated "noise" in

an attempt to introduce process noise (the variability that could be expected from a larger sample of "identical" engines) into the data set. The data was divided into two equal parts for training and testing. A BP network with a 350-50-29 configuration (350 input nodes, 50 hidden nodes and 29 output nodes) was trained and performed acceptably well on the classification task (>98% accuracy). However, although networks of this size are manageable on small workstations (training was ultimately performed on an IBM RS6000 RISC processor), there are arguably too many free parameters (almost 19,000 trainable weights). No precise rules for selecting sample sizes have appeared in the literature, but we think it prudent to have at least one sample per trainable weight. However, with 19,000 sample vectors and 19,000 trainable weights, proper statistical testing of such a network is beyond the capabilities of modest workstations. We therefore sought to reduce the dimensionality of the input vectors to decrease the input data requirements and the training and testing times.

One approach to dimensionality reduction is to select a set of "features" in the data, based upon an understanding of the physical processes involved (e.g. zero crossing times, peak-to-valley ratios of torque etc.). We elected not to pursue this approach because we wanted to develop a scheme which required as little *a priori* knowledge as possible and therefore was applicable to a wide range of problems. Principal Component Analysis (PCA) is one well-known means of developing a new representation for a sample vector space. Typically, the PCA provides a compact representation of a sample vector space from which effective classifiers can be constructed. A full treatment of PCA is given in a text by Jolliffe, but a few basic features are noted here [3]. PCA is the projection of the vectors from the input samples onto a new set of orthogonal axes which are chosen to represent the largest variance in the sample of data presented. The first principal component is chosen as the direction which accounts for the largest variance in the data. The next (and subsequent) principal component(s) is (are) in the direction associated with the largest remaining variance, subject to the constraint that it is orthogonal to the preceding component(s) For many of our data sets, if we terminate the process after about 99% of the variance is accounted for, we observe more than 10:1 reduction in the dimensionality of input vector space . It is of perhaps more interest to note that the performance of neural network classifiers in these new representations improved over those using the original data representations.

If we apply PCA to our data set and terminate the PCA process after 99% of the variability has been accounted for, we obtain a vector space in the PCA representation with 27 components. A neural network in a 27-16-29 configuration (about 900 trainable weights), trained with 25% of the number of passes through the training set required for the raw data. Combining the smaller number of weight updates required with the smaller number of passes through the data, the use of the PC representation reduced the network training time by a factor of 100.

The PCA analysis could be applied to the complete vector space, in which case the sample space of 350 x 2000 is projected onto a new set of axes. However, the association of the resulting PCA components with any physical measurement is quite difficult and the computational task involves inverting very large matrices. To avoid these difficulties and to provide a means of visualizing and interpreting the PCA representation, we divided the input vector space into several subspaces and performed the PCA on those subspaces. The subspaces were the individual cylinder torque traces, the overall torque trace, the separate overall pressure traces, and finally the deviations of the individual cylinder events from the mean of all events for that engine. The exact details of this subspace decomposition are discussed elsewhere and precise decomposition is problem dependent [4]. The significance of this step is that its reduces the computational task for PCA, it simplifies the interpretation of the PCA and it very often reduces the number of PC's in each subspace to 3 or 4. Figure 2 indicates how various fault signatures appeared in the PCA representation. With the help of 3-D scatterplots and "slicing" in the fourth dimension, or with matrix plots, the PC data within each subspace can be easily visualized [5]. For our purposes, the decomposition of the engine data was into 11 subspaces with 2 to 5 PC's retained for each subspace. The 11 subspaces contained a total of 35 elements which comprised the reduced PC representation of the engine data (nearly a factor of 100 reduction). It is on this vector representation that the classification problem is attacked.

III. ANALYSIS

For a case study on real data, we were presented with data from over 1000 different pre-production engines. This dataset was obtained from a plant survey and lacked a *bona fide* classification for each engine, although very good engines and engines with serious defects were quite evident from the graphs. The problem was to develop a classifier which could identify GOOD from BAD and also identify any faults present in the engines under test. As a first step, we visually scanned all the raw data and identified as many engines as possible as GOOD or BAD and assembled a training set from this manually tagged data. A neural network was trained on this data set until its RMS error ceased to decrease. The classifications of the network were compared with ours and some adjustments were made to our classifications and the network was retrained on the retagged data set. After a few iterations on a training sample of 300 engines, the process converged to agreement between the network classifications and ours. The network was tested on the remaining engines and the results were compared with a technician's analysis of the data. In most cases, the expert technician and the network were in agreement, although the technician was analyzing raw data and the network was analyzing the PCA data.

In reviewing this database, we noticed that sudden changes in the signal spectra took place as a result of changes introduced in the manufacturing process. For example, such an effect could be caused by a change in the lubricating oil in the engine which reduces the turnover torque. This situation caused batches of data within the database to have different means and slightly different variances. Consequently, the amount of real data which would be available to provide examples for training sets seemed likely to be very limited. Further analysis of the PC's revealed that the covariance matrix of the PC data contained off-diagonal terms, indicating that the individual raw signal traces from each engine were correlated. It was noted that the sample means of the PC's varied from production batch to batch, but that the covariance matrix was stable. To re-train a network each time such a shift in the production occurred would require copious quantities of data, which would not be available until some time after each change in the production process. A viable solution to this problem is to utilize the fact that the second-order statistics of the measurement problem are stable and incorporate Monte Carlo methods to generate sufficient data from estimates of the sample means. Unlike our initial study in which we utilized uncorrelated noise, we now needed to generate Monte Carlo data with the same covariance as the real data. A detailed description of the means to carry out this procedure is contained in the Appendix. The Monte Carlo process may be used to generate augmented data sets of both normal and faulty engines if one makes the reasonable assumption that the faulty engines' PC's have covariance matrices similar to that of the normals. This data augmentation process also helps to identify "class clusters" that are easy to separate. In the past, higher success rates for proper class identification of abnormal situations were claimed than could actually be obtained in practice because the variance in the clusters of abnormals was not properly accounted for. In our approach, we base our estimates of the cluster statistics on the historical data and amend the statistics as necessary to be consistent with the incoming data. In most cases, the proper consideration of all the cluster variances diminishes the ability to separate all the fault categories. However, the performance observed in development provides a more accurate gauge of final performance.

In attempting to provide a diagnostic tool which is easy to manage and re-train, we noted that the PCA data, broken down into the 11 subspaces could be very effectively classified as GOOD or BAD by a hard shell classifier defined by elliptical shells centered on the centroids of the distribution of GOOD engines with axes radii determined by the variance of the distributions. Normalization of the distributions to zero mean and unit variance simplifies the classifier boundaries to spherical shells. An ideal engine would be most similar to the best engine identified or the mean of an ensemble of such engines. If the deviation of an engine from such a distribution is larger than an acceptable value, the engine is declared to be unsatisfactory. In the early stages of this functional testing, no empirical data was available for selecting the tolerance boundary. We used Monte Carlo simulations to determine the variations we could expect from a single class of data with the proper covariance matrix. From this simulation we determined that shells with radii shown in Figure 3 would contain virtually all of the Monte Carlo samples. To pass, an engine must fall within all 11 shells constructed for the 11 vector PC subspaces. However, since the Monte Carlo statistics are Gaussian, a fraction the samples will fall outside some spheres. If the values associated with the hard-shell classifiers are selected as shown in Figure 3, we have determined that the GOOD engines should score 9.0 or higher (on a scale of 11) in order to pass 99% of the samples. The histogram of the Monte Carlo data for

the expected distribution of GOOD engines is shown in Figure 4. If the engine falls below the threshold value, then the neural network will be used to identify the failure present. This approach provides an easily understandable, traditional classifier for acceptance and rejection based upon the assumption of a convex data set for the normal engines. The neural network is used for the task it can perform well, fault classification, which may involve very odd-shaped or non-convex sets of data. We anticipate that the class clusters are well-separated, but perhaps not by simple boundaries. The data set from the plant is consistent with this conjecture. Typically, the faulty engines from the production data scored below a 6 or 7, so that we may expect that the distributions of GOOD and BAD are as separable as they were in the initial laboratory study with the 3.0 liter engine. In this situation, we can effectively use standard feedforward networks trained by backpropagation, or utilize Restricted Coulomb Energy (RCE) networks which train much faster.

The process control aspect of this approach is evident if we monitor the engine scores as a function of time. For each major change in the production, the engine test scores dropped until new sample means were calculated. The neural network can provide information on the nature of the problem by indicating the "direction" or the tendency of a fault. For BP, we use one unit in the output layer for each fault class,, and as the data points move in the direction of a known fault, the GOOD output node decreases in value and the FAULT node associated with the class direction in which the data is moving increases in value. Thus, the neural network may be used to provide prognostic information about engines that have not crossed the threshold for outright rejection. We note that the BP network in this situation operates with the full 35 dimensional input space as a fully interconnected network. Investigations are underway to determine if subspace groupings, as used for the hard shell acceptance classifier, applied to the RCE network provide any benefits.

IV. CONCLUSIONS

We have demonstrated how a combination of conventional statistical processing methods and neural networks can be combined to create a classifier system for engine diagnostics. The most significant computational effort is required to compute the PCA and to properly develop the hard-shell classifiers using data sets augmented with Monte Carlo methods. Once these procedures are carried out, the application of neural networks to the data set to obtain the trainable classifier is quite straightforward. We expect that these methods are applicable to a wide range of classification problems.

REFERENCES

[1] K.A. Marko, J. James, J. Dosdall and J. Murphy, "Automotive Control System Diagnostics Using Neural Nets for Rapid Classification of Large Data Sets", Proceedings of IJCNN-89, pp. I I 13-17, Washington D.C., 1989.

[2] K.A. Marko, "Neural Network Application to Diagnostics and Control of Vehicle Control Systems", Neural Information Processing Symposium (NIPS-91) Denver, Colorado, Morgan-Kaufman, pp 337-343, New York 1991.

[3] Jolliffe, I.T. "Principal Component Analysis", Springer-Verlag, Berlin 1986.

[4] K.A. Marko, B. Bryant, N. Soderborg, Ford Technical Report (to be published).

[5] See for example "Data Desk" software for Macintosh Computers, Data Description Inc., PO Box 4555, Ithaca, New York.

[6] Strang, C.G., "Introduction to Applied Mathematics" Wellesley Cambridge Press, Wellesley, MA. 1986.

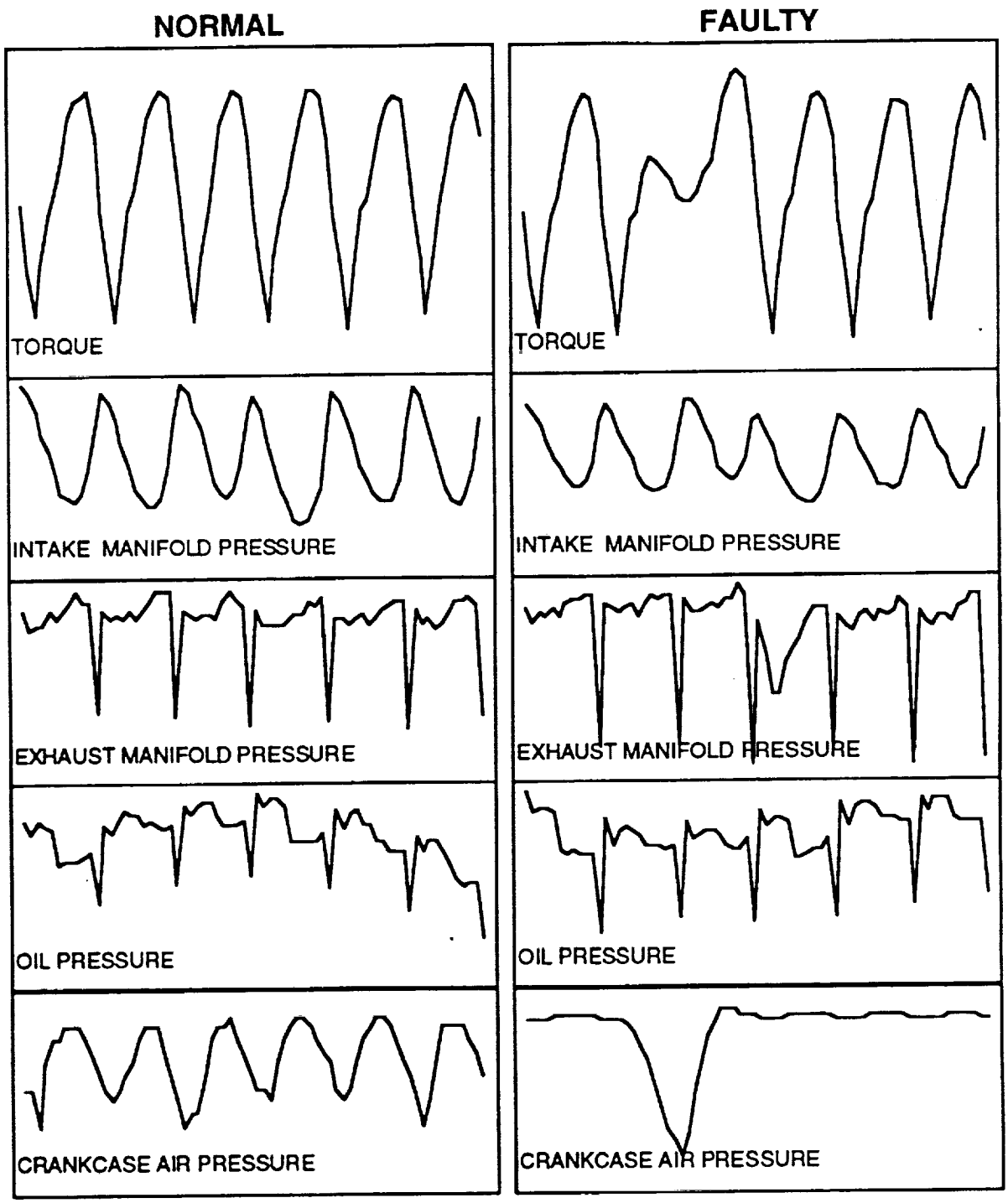


FIGURE 1. Data traces obtained from normal engine (on the left) and from an engine with an easily detectable fault (on the right). The traces are based upon sampling the analog signals one every crankangle degree, so that each trace consists of 720 points.

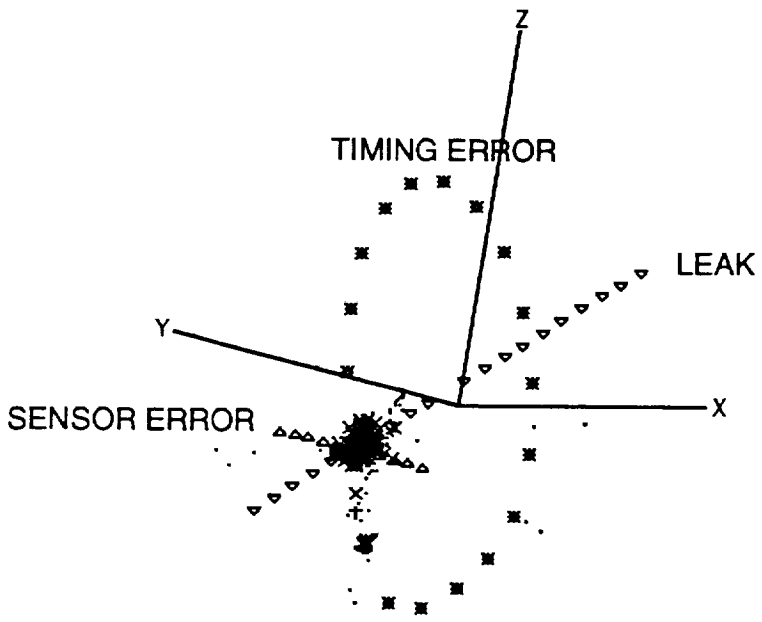


Figure 2. Plot of artifically induced "faults" in PC represenation of exhaust manifold signals. Dense cluster of dots represents "normal" engines. The other other signals indicate the effects of introducing various faults, such as camshaft timing error, or leaks into the engine.

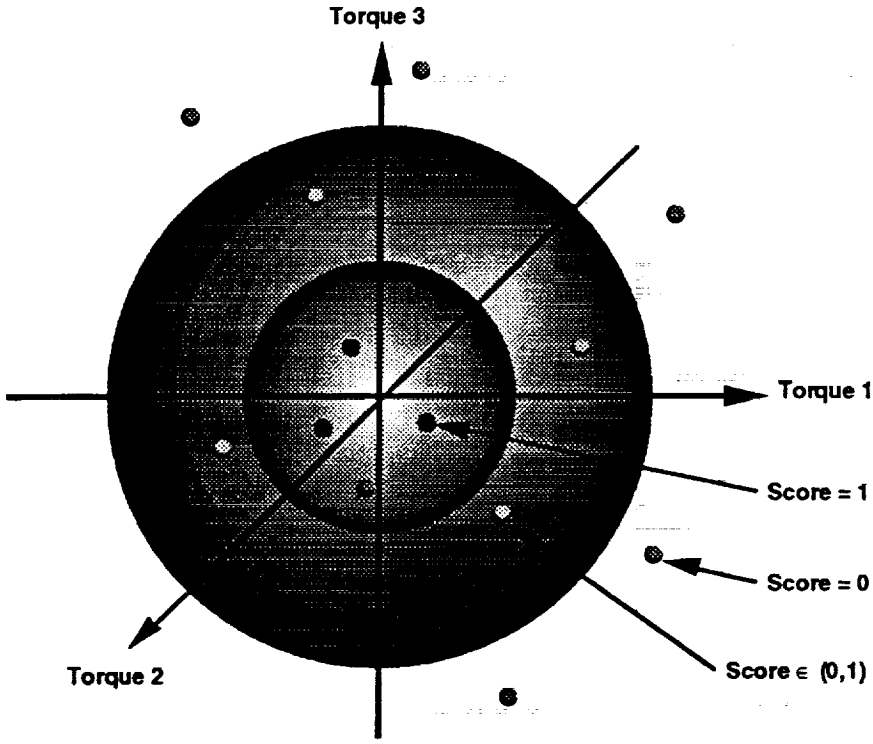


Figure 3. Spheroidal Classifier. Engines are rated according to the location of their data points (shown as small dots) relative to spherical shells whose size and location are determined from a set of nominal good engines. The radii of the shells are proportional to the variance of the empirical distributions. We typically choose 2 times the standard deviation (S.D.) for the inner radius and 3 times the S.D. for the outer radius. The engine test score is determined from 11 such classifiers in the PCA subspaces described in the text. Engines with all points within the inner spheres have a perfect score of 11.

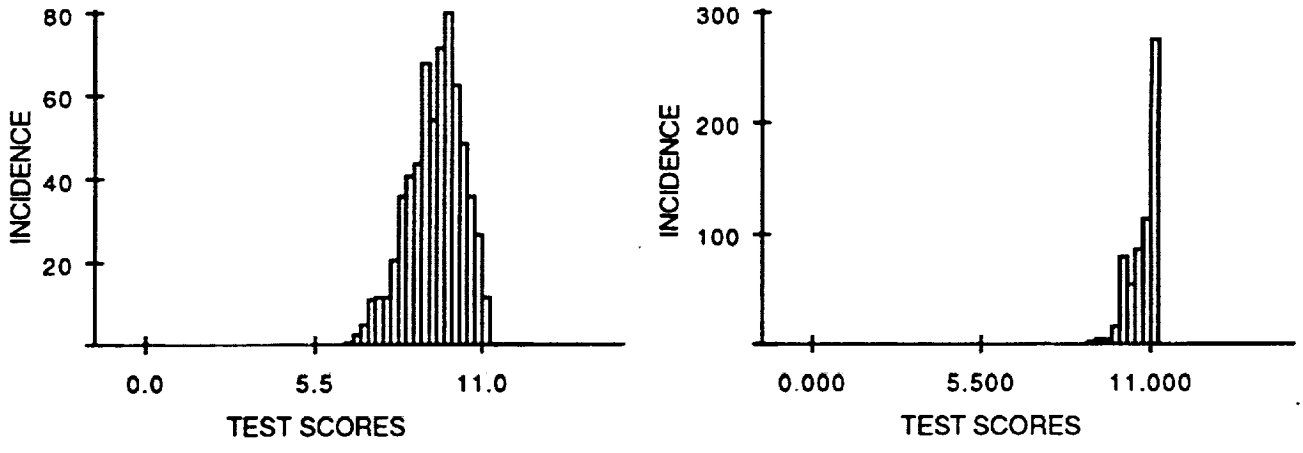


Figure 4. Histograms of the engine scores from the Monte Carlo simulations. The distribution on the left is due to data generated from normally distributed PCA data with a diagonal covariance matrix. The distribution on the right is due the same data transformed to have the covariance obtained empirically from production data. This Monte Carlo simulation of GOOD engines cuts off below a test score of 9.

Document Analysis with Neural Net Circuits

Hans Peter Graf

AT&T Bell Laboratories, Holmdel, NJ 07733

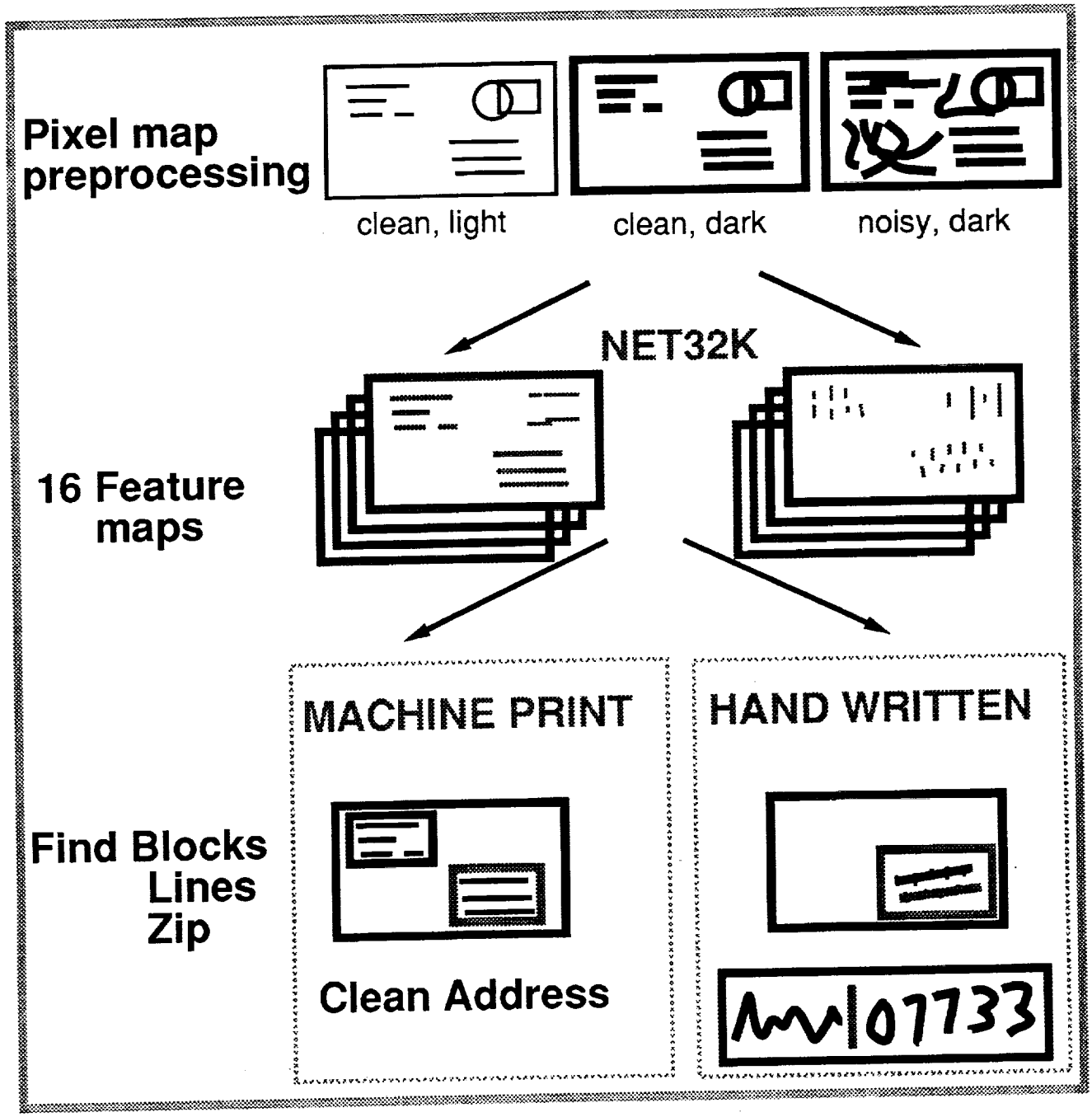
Document analysis is one of the main applications of machine vision today and offers great opportunities for neural net circuits. Despite more and more data processing with computers, the number of paper documents is still increasing rapidly. A fast translation of data from paper into electronic format is needed almost everywhere, and when done manually, this is a time consuming process. Markets range from small scanners for personal use to high-volume document analysis systems, such as address readers for the postal service or check processing systems for banks.

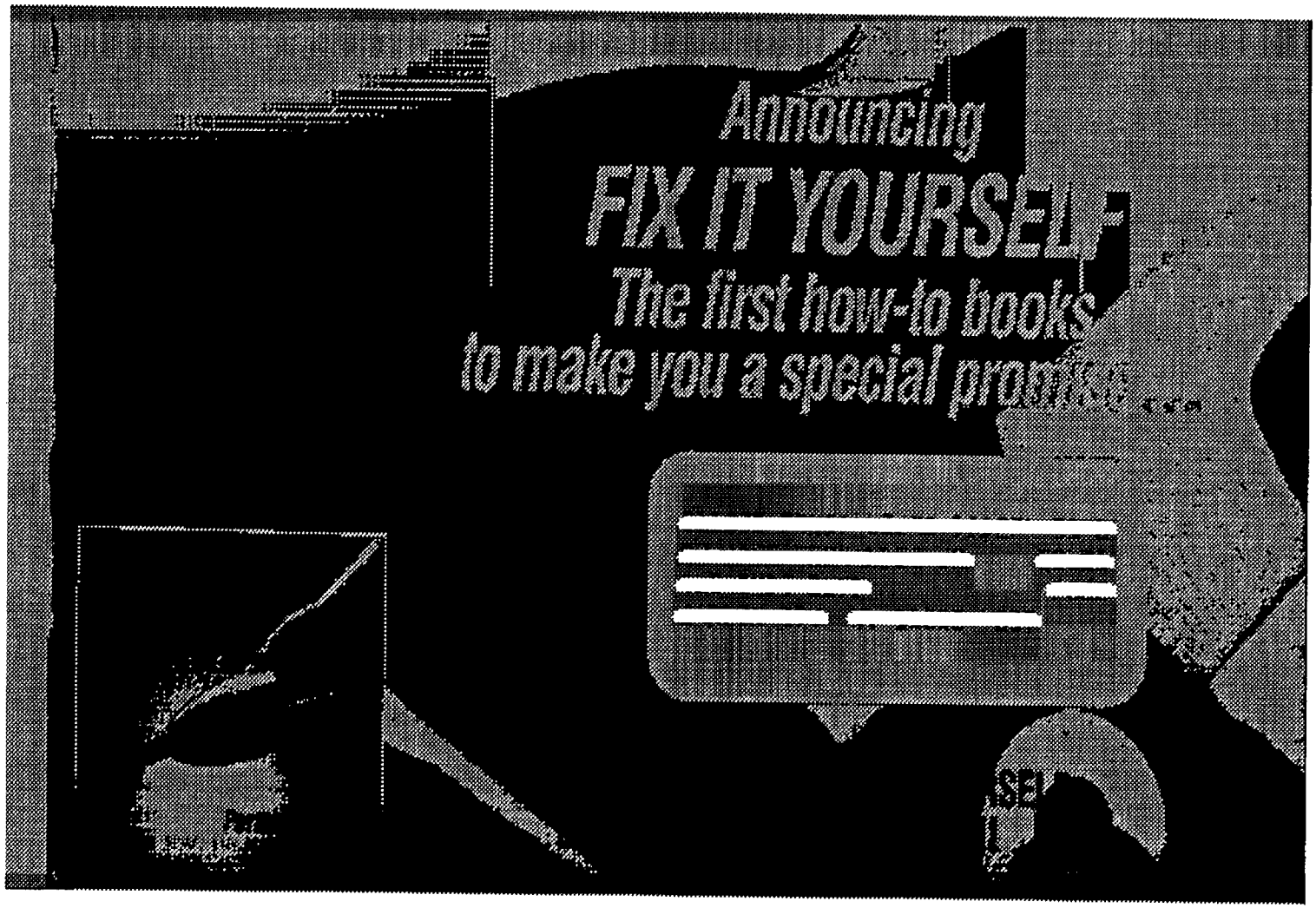
A major concern with present systems is the accuracy of the automatic interpretation. Systems tend to work well, if the image is not too complex and its quality is good, i.e. there is no noise in the image and the print quality is good. Todays algorithms, however, fail miserably when noise is present, when the print quality is poor or when the layout is complex. A common approach to circumvent these problems is, to restrict the variations of the documents handled by a system.

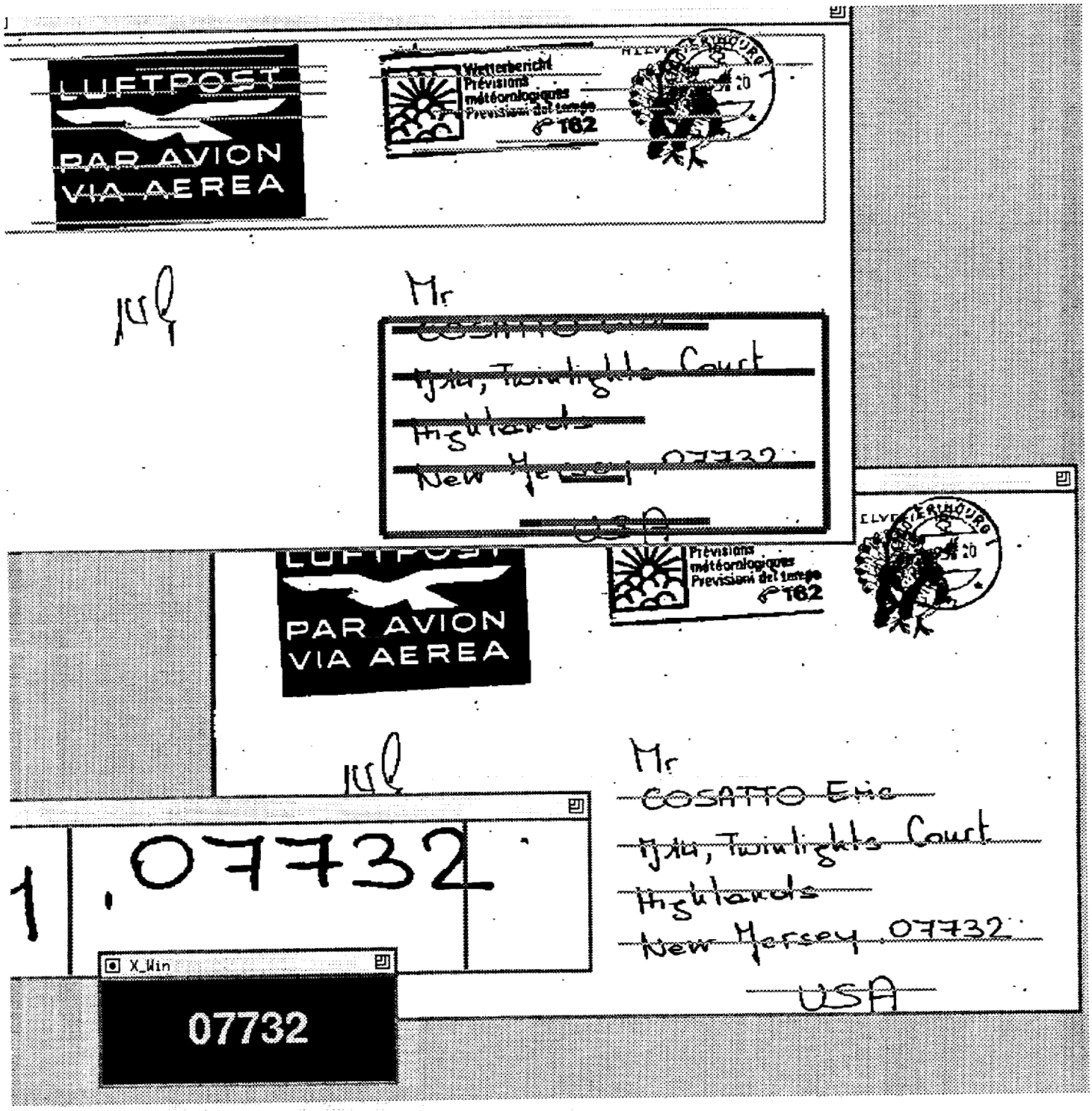
Improving the robustness of algorithms, to deal with a wider variety of conditions, seems always to lead to algorithms requiring an enormous amount of computation. This is a good opportunity for specialized circuits, such as neural net chips. Key for a successful integration of such a circuit into an application is that all the algorithms, from start to end, are taken into account and that the throughput of each stage is well balanced. Often neural net circuits were designed with one particular algorithm in mind, for example the character recognition. But in an application other processing steps, such as the layout analysis or just the discrimination between figures and text, may require more computation and represent the throughput bottleneck. It is clear by now that "pure neural network" solutions are suited for some aspects of document analysis, most notably the recognition of individual characters, but many problems are solved more effectively with other types of algorithms. The main problem for any hardware implementation is that algorithms applied in document analysis are still evolving and are changing rapidly. It is therefore easily possible that by the time a circuit is built and integrated into a system, newer algorithms with better performance and different compute requirements have been developed.

In our laboratory, we had the best luck with circuits implementing basic functions, such as convolutions, that can be used in many different algorithms. To illustrate the flexibility of this approach, three applications of the NET32K circuit are described: Locating address blocks, cleaning document images by removing noise, and locating areas of interest in personal checks to improve image compression. Several of the ideas realized in this circuit that were inspired by neural nets, such as analog computation with a low resolution, resulted in a chip that is well suited for real-world document analysis applications and that compares favorably with alternative, "conventional" circuits.

LOCATING ADDRESS BLOCKS

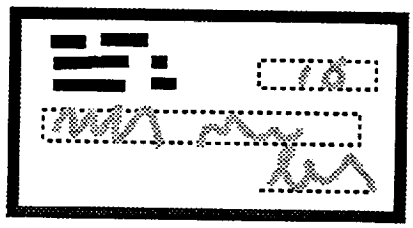



Announcing
FIX IT YOURSELF
The first how-to books
to make you a special

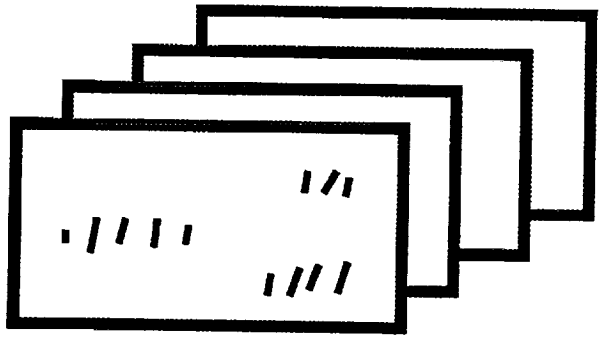

LUFTPOST
PAR AVION
VIA AERIA
Wetterbericht
Prévisions météorologiques
Previsioni del tempo
T62
Mr
Tjkk, Twinlights Court
Highlands
New Jersey 07732
USA
LUFTPOST
PAR AVION
VIA AERIA
Prévisions météorologiques
Previsioni del tempo
T62
Mr
COSATTO Eric
Tjkk, Twinlights Court
Highlands
New Jersey 07732
USA
1 07732
X_Win
07732

Compress check image

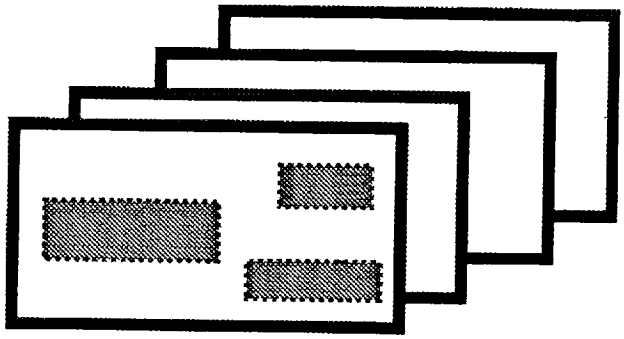
Check with information added by customer



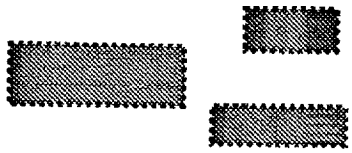
Extract features with NET32K: edges and strokes

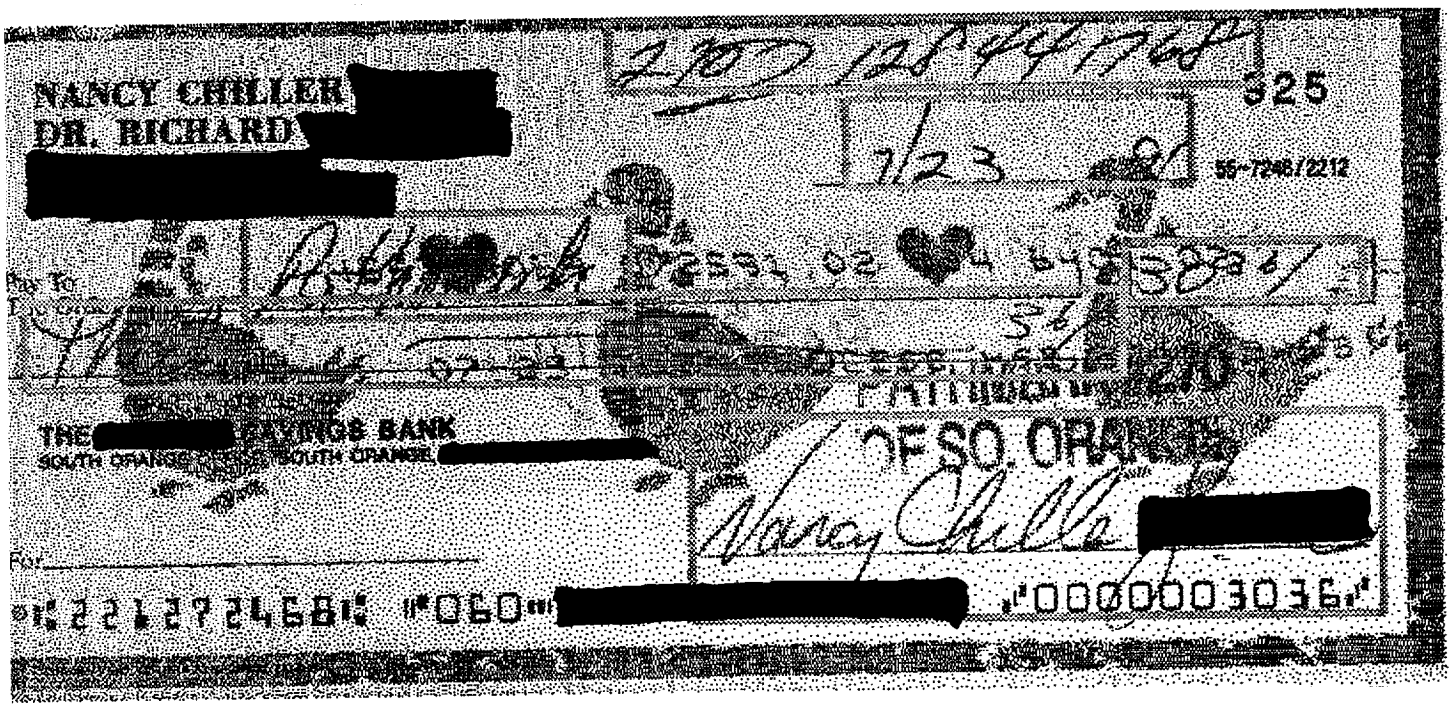


Identify areas with handwritten text.



Transmit only areas of interest.





(Portions of check obliterated to ensure privacy.)

From Neural-Based Object Recognition toward Microelectronic Eyes

Bing J. Sheu, Ph.D. Senior Member, IEEE
Sa Hyun Bang, Ph.D. Student Member, IEEE

Department of Electrical Engineering, Powell Hall-604
University of Southern California, Los Angeles, CA 90089-0271, U.S.A.
Also with Center for Neural Engineering & the Signal and Image Processing Institute

Abstract

Engineering neural network systems are best known for their abilities to adapt to the changing characteristics of the surrounding environment by adjusting system parameter values during the learning process. Rapid advances in analog current-mode design techniques have made possible the implementation of major neural network functions in custom VLSI chips. An electrically programmable analog synapse cell with large dynamic range can be realized in a compact silicon area. New designs of the synapse cells, neurons, and analog processors are presented. A synapse cell based on Gilbert multiplier structure can perform the linear multiplication for back-propagation networks. A double differential-pair synapse cell can perform the Gaussian function for radial-basis network. The synapse cells can be biased in the strong inversion region for high-speed operation or biased in the subthreshold region for low-power operation. The voltage gain of the sigmoid-function neurons is externally adjustable which greatly facilitates the search of optimal solutions in certain networks. Various building blocks can be intelligently connected to form useful industrial applications. Efficient data communication is a key system-level design issue for large-scale networks. We also present analog neural processors based on Perceptron architecture and Hopfield network for communication applications. Biologically inspired neural networks have played an important role towards the creation of powerful and intelligent machines. Accuracy, limitations, and prospects of analog current-mode design of the biologically inspired vision processing chips and cellular neural network chips are key design issues.

I. Introduction

Rapid progresses in the research of intelligent information processing paradigms, architectures, and electronic hardware implementations based on artificial and biologically-inspired neural network models have helped to establish a rich knowledge base for practical applications. Studies of engineering neural network models were motivated by the investigation of human perceptron. The Von Neumann computing approach incorporates a single central processing unit and the main memory unit. It can execute instructions sequentially with a reasonable speed and accuracy for conventional data-processing applications. However, these digital machines, when packaged in a small physical size, can not perform computationally-intensive tasks with satisfactory performance in such areas as intelligent perceptron, including visionary and auditory signal processing, recognition, understanding, and logical reasoning where human being and even living animals can do a superb job.

Recent advances in artificial and biological neural networks research have provided excited evidence for high-performance information processing with a more efficient use of computing resources. The secret lies in the design optimization at various levels of computing and communication. Each neural network system consists of massively paralleled and distributed signal processors with every processor performing very simple operations. Large computational capabilities of these systems are derived from collectively parallel processing and efficient data routing through well-structured interconnection networks. Two different operation modes are associated with a typical neural information processing network: the data retrieving process and the learning process.

II. General Properties

Many important issues need to be carefully addressed in constructing electronic neural network systems:

1. A balanced exploration on the computing algorithms and architectures which are suitable for digital VLSI implementations and analog networks;

2. Emphasis of both artificial neural networks and biologically-inspired neural models; and

3. Solving real-world, large-scale problems.

In electronic implementation, the options are digital, analog, a combination of both, or pulsed-stream forms. Analog approaches can be divided into continuous-time [1, 2, 3], and discrete-time schemes [4, 5]. In continuous-time analog VLSI, some additional options arise relating to the operation mode of transistors: weak inversion [6] and strong inversion [7]. The pulsed-stream approach [8] is more biologically motivated than other approaches. Lyon and Mead [9] described the VLSI implementation of an analog electronic cochlea for speech recognition. Koch et al. [10] reported a real-time chip for computer vision and robotics. Satyanarayana et al. [11] presented a reconfigurable analog VLSI neural chip for general-purpose applications. Hollis and Paulos [12] proposed a current-summing neuron with binary data registers. Boser and Sackinger [13] presented an analog neural chip for hand-written character recognition. Fang, Sheu, et al. [14] presented a mixed-signal neural network processor chip for self-organizing networks.

There are three basic neural network architectures: the iterative networks, the multi-layer perceptron networks, and the self-organizing networks. The iterative neural networks, which are also called recurrent neural networks, are promising for temporal pattern recognition and generation. Recurrent neural networks can solve optimization problems because of their constraint-satisfaction capabilities. Data is retrieved from an iterative network through associative recalling. Representative iterative networks include the Hopfield network [15] and bidirectional associative memory [16]. In a multi-layer perceptron network, supervised learning [17] is used. The effective errors for the output layer and hidden layers are calculated from the actual outputs and expected outputs. Synapse weights are updated according to the delta rules or the derivatives. Layered neural networks are effective for spatial pattern recognition. The multi-layer perceptron networks are widely used in industrial applications.

A self-organizing network consists of two layers of neurons: the input layer and the competitive layer, which is also called the output layer [18]. A winner-take-all function is performed among the neurons in the competitive layer. The self-organizing network has the desirable property of effectively producing spatially organized presentation of various features of the input signals [19]. Competitive learning depends on the competition among the output neural units. Self organization is required in several image and vision processing applications such as pattern recognition, vector quantization for image compression, and motion estimation. In addition, it may be applied in the selection of optimal inference paths in symbolic computers. Such an application can systematically reduce the knowledge inference operation from an NP complete problem to a much simplified problem in a very efficient way.

III. Analog Building Blocks

Power consumption, required silicon area, and the number of packaged pins are also important figures of merit in practical hardware implementation. The required silicon area for a given function will be gradually decreased with the advances of microelectronic fabrication technologies. Therefore, the number of packaged pins for information communication could become a fundamental limitation for information exchange. Each package pin can be shared by several functional outputs through time-multiplexing scheme or frequency-multiplexing scheme.

A. Memory in Synapse Cells

An important component in hardware implementation of learning is memory. In analog neural network processor chips, synapse weight information can be stored in various formats. In the early design, fixed-resistance synapses were implemented with the well regions or an amorphous-silicon layer. Complementary-MOS transmission gates were also proposed to achieve programmable synapse resistance. Continuous-time synthesized resistance [20] is

made of four MOS transistors which are connected in a cross-coupled fashion. The threshold voltage mismatch effect is minimized by using symmetric control voltage.

A basic transconductance amplifier which is made of five MOS transistors requires a simple control signal for the programmable synapses [8]. Such a compact and programmable synapse provides the first- and third-quadrant multiplication capability. The synapse weight can be stored on the gate capacitance and refreshed periodically. A modified wide-range Gilbert multiplier is suitable for general-purpose programmable synaptic operation because it provides four-quadrant multiplication capability [21]. Long-term memory information can be stored in the floating-gate devices fabricated by a special EEPROM technology [22] or by a conventional double-polysilicon technology for analog circuits for over 20 years in room temperature [23].

B. Neurons

The summed synaptic current is converted to the voltage through a current-to-voltage converter. The feedback resistance of the converter can be implemented with six MOS transistors. The voltage gain of the neurons can be controlled continuously to perform the hardware annealing operation [24, 25] for the quick searching of optimal solutions in nonlinear optimization applications. Such a hardware implementation of mean-field annealing can be used in recurrent neural networks and multi-layered perceptron networks to avoid local minima problems.

C. Winner-Take-All Circuit

A high-precision VLSI winner-take-all circuit can achieve high-speed operation by biasing transistors in the strong-inversion region. It uses the cascade configuration to significantly increase the competition resolution and maintain a high speed operation for a large-scale network. The total bias current increases in proportion to the number of circuit cells so that a nearly constant response time is achieved. In addition, a unique dynamic current steering method is used to ensure only a single winner exists in the final output. Experimental results of the prototype chip fabricated by a 2-μm CMOS technology show that a cell can be a winner if its input is larger than those of the other cells by 15 mV. The measured response time is around 50 $nsec$ at a 1-pF load capacitance. This analog winner-take-all circuit is a key module in the competitive layer of self-organization neural networks.

D. Radial-Basis Function Circuit

The circuit schematic diagram and transistor sizes for a Gaussian function synapse cell is shown [26]. This circuit consists of MOS differential pair and several arithmetic computational units in the current-mode configuration. Transistors with non-minimum channel lengths are used to avoid the channel-length modulation effect. The input voltage is applied to the gate terminal of one transistor in the differential pair and the synapse weight value is stored on the capacitance at the gate terminal of the other transistor. Measured results of the Gaussian synapse cell are shown.

IV. Design Methodology

Mixed-signal VLSI implementation is suitable for novel signal processing applications such as image restoration [45] and optical flow computing [46]. The mixed analog-digital circuit design techniques are used to take advantages of efficient numerical computation in analog domain with long-distance communication in digital data bus. The multiplexed scheme can also be used to transmit signals over a long distance in an electronic system. Additional system-level integration results can be found in [47].

Hybrid approach using combined analog dynamics and digital logic represents very powerful and appealing design. For example, the programmable CNNs provide a new quality of artificial neural networks through a kind of analog software, a simple way to solve CNN algorithms. In our design, we give the network instructions and templates information just like we had done with the general-purpose CPU. The whole system will work like a SIMD machine and each local cell will execute the given commands to accomplish the functions we want. There are two distinct portions

but they both use the analog and digital circuits. One part is consisted of global digital control circuits and global analog memory; the other one has one duplications in each local cell which contains small local control circuits and local analog and digital memory. A timing diagram of the global digital circuit is shown in figure 8.

One other novel way to implement the neural network is a hybrid neurocomputer that utilized electro-optic components for the input processing and analog electronics for implementation of the remainder of the transfer function. This type of neurocomputer was shown to be capable of successfully implementing simple Hopfield neural networks with weight values restricted to the set {-1, 0, +1}. B. Soffer et. al also developed a first all-optical neurocomputer [27].

V. Cellular Neural Network

1. General

A cellular neural network (CNN) is a continuous-time or discrete-time artificial neural network that features a multi-dimensional array of neuron cells and local interconnections among the cells. The basic CNN proposed by Chua and Yang [28, 29] in 1988 is a continuous-time network in the form of an n-by-m rectangular-grid array where n and m are the numbers of rows and columns, respectively. However, the geometry of the array needs not to be rectangular and can be such shapes as triangle or hexagon [30]. A multiple of arrays can be cascaded with an appropriate interconnect structure to construct a multi-layered CNN. Structural variations of the continuous-time, shift-invariant, rectangular-grided network include discrete-time CNN [31], CNN with nonlinear and delay-type templates [32], etc. CNN and its variations provide a natural and universal model of analog processor arrays on a geometrical grid. Their local connectivity and regular structure appear most efficient for electronic implementation for high-speed, real-time applications. Several hardware implementations of the CNN have been reported in the literatures [33]-[39].

2. Hardware Annealing

The hardware-based annealing technique [25], has an analogy to the metallurgical annealing in the metallurgy and simulated annealing in the Boltzmann machine, which are the optimal stochastic procedures. It is a paralleled, electronic version of the deterministic mean-field learning rule [42, 43] directly incorporated with the Hopfield neural network or CNN. It is a dynamic relaxation process for finding the optimum solutions in the recurrent associative neural networks such as Hopfield network and CNN. Even with a correct mapping of the cost function onto a neural network, the desired combinatorial solution is not guaranteed because a concave optimization problem always involves a large number of local minima. True combinatorial solutions can be achieved by applying the hardware-based annealing technique with which the global minimum of E is found in a real-time speed.

3. Applications

The CNN's can be used in many computation-intensive, adaptive signal processing applications. Due to its two-dimensional array architecture, CNN's are suitable for real-time image processing applications in the following areas [30].

(a) Image processing: Feature extraction, motion detection & estimation, path tracking, collision avoidance, and mage halftoning,

(b) 3-D surface analysis: Min/max detection and gradient estimation,

(c) Solving partial differential equations,

(d) Non-visual data imaging: Thermographic images, antenna array images, and medical maps and images.

A CNN has similar collective computational behaviors with Hopfield neural networks. Thus, the quadratic nature of the Lyapnov function allows us to map it into optimization problems [41, 43].

VI. Conclusion

There is a strong need to develop new neural network architectures and design techniques to extend the size of electronic implementation to a larger scale for solving real-world problems in science, engineering, and business. Extension of the hardware annealing to large-scale networks for complex problems is highly desirable. Chip-level and system-level packaging technologies will be crucial for future computing machines with one-million-unit neural networks on silicon wafers that interact with the external environment and change the structures adaptively. Reusable software modules and hardware modules are to be invented. For large scientific problems, neural networks with 10 tera connection updates per second will be needed. A flexible framework for representing various kinds of information efficiently and effectively will be the key for successful hardware/software co-designed systems.

Acknowledgement

The authors would like to thank Mr. Tony H.-Y. Wu for preparing some of the figures.

References

[1] B. W. Lee, B. J. Sheu, "Design of a neural-based A/D converter using Hopfield network," *IEEE J. of Solid-State Circuits,* vol. 24, pp. 1129-1135, Aug. 1989.

[2] B. E. Boser, E. Sačhinger, et al., "An analog neural network processor with programmable topology," *IEEE J. Solid-State Circuits,* vol. 26, pp. 2017-2025, Dec. 1992.

[3] M. A. C. Maher, C. A. Mead, et al., "Implementing neural architectures using analog VLSI circuits," *IEEE Trans. on Circuits and Systems,* vol. 36, pp. 643-652, May 1989.

[4] J. E. Hansen, D. J. Allstot, et al., "A time-multiplexed switched-capacitor circuit for neural network applications," *IEEE Int. Symp. on Circuits and Systems,* vol. 3, pp. 2177-2180, 1989.

[5] R. Dominguez-Castro, E. Sanchez-Sinencio, et al., "Analog neural networks for real-time constrained optimization," *IEEE Int. Symp. on Circuits and Systems,* vol. 3, pp. 1867-1870, 1990.

[6] C. A. Mead, et al., "Analog VLSI model of binaural hearing," *IEEE Trans. on Neural Networks,* vol. 2. pp. 230-236, Mar. 1991.

[7] B. W. Lee, B. J. Sheu, "General-purpose neural chips with electrically programmable synapses and gain-adjustable neurons," *IEEE J. of Solid-State Circuits,* vol. 27, pp. 1299-1302, Sept. 1992.

[8] A. Hamilton, et al., "Integrated pulse stream neural networks: results, issues, and pointers," *IEEE Trans. on Neural Networks,* vol. 3, pp. 385-393, May 1992.

[9] R. F. Lyon, C. A. Mead, "An analog electronic cochlea," *IEEE Trans. on Signal Processing,* vol. 26, pp. 1119-1134, July 1988.

[10] C. Koch, et al., "Real-time computer vision and robotics using analog VLSI circuits," *Advances in Neural Information Processing Systems 2,* pp. 750-757, Morgan Kaufmann, 1990.

[11] S. Satyanarayana, Y. Tsividis, H. P. Graf, "A reconfigurable analog VLSI neural network chips," Editor: D. Touretzky, pp. 758-768, Morgan Kaufmann: San Matao, CA, 1990.

[12] P. W. Hollis, J. J. Paulos, "Artificial neural networks using MOS analog multipliers," *IEEE J. Solid-State Circuits,* vol. 25, pp. 849-855, June 1990.

[13] B. E. Boser, E. Sackinger, "An analog neural network processor with programmable network topology," *IEEE Tech. Digest of Inter. Solid-State Circuits Conference,* pp. 184-185, San Francisco, CA, Feb. 1991.

[14] W.-C. Fang, B. J. Sheu, O. T.-C. Chen, J. Choi, "A VLSI neural processor for image data compression using self-organizing networks," *IEEE Trans. on Neural Networks,* vol. 3, pp. 506-518, May 1992.

[15] D. W. Tank, J. J. Hopfield, "Simple 'neural' optimization networks: an A/D converter, signal decision circuit, and a linear programming circuit," *IEEE Trans. on Circuits and Systems,* vol. 33, pp. 533-541, May 1986.

[16] B. Kosko, *Neural Networks and Fuzzy Systems,* Prentice Hall: Englewood Cliffs, NJ, 1992.

[17] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representation by error propagation," in *Parallel Distributed Processing, vol. 1,* Eds. D. Rumelhart & J. McClelland, MIT Press: Cambridge, MA, 1986.

[18] T. Kohonen, *Self-Organization and Associative Memory, 2nd ed.,* Springer-Verlag: New York, NY, 1988.

[19] T. Kohonen, "The self-organizing map," *Proc. of IEEE,* vol. 78, pp. 1464-1480, Sept. 1990.

[20] S. Bibyk, M. Ismail, "Issues in analog VLSI and MOS techniques for neural computing," in *Analog VLSI Implementation of Neural Systems,* Eds. C. Mead, M. Ismail, pp. 116-125, Kluwer Acad, 1989.

[21] B. J. Sheu, J. Choi, C.-F. Chang, "An analog neural network processor for self-organizing mapping," *IEEE International Solid-State Circuits Conference,* pp. 136-137, 266, San Francisco, CA, Feb. 1992.

[22] M. Holler, S. Tam, et al., "An electrically trainable artificial neural network (ETANN) with 10240 'Float gate' synapses," *Proc. IEEE/INNS Inter. Joint Conf. on Neural Networks,* vol. 2, pp. 191-196, Washington, DC, June 1989.

[23] B. W. Lee, H. Yang, B. J. Sheu, "Analog floating-gate synapses for general-purpose VLSI neural computation," *IEEE Trans. on Circuits and Systems,* vol. 38, pp. 654-658, June 1991.

[24] B. W. Lee, Bing. J. Sheu, *Hardware Annealing in Analog VLSI Neurocomputing,* Kluwer Academic Publisher: Boston, MA, 1991.

[25] B. W. Lee, B. J. Sheu, "Paralleled hardware annealing for optimal solutions on electronic neural networks," *IEEE Trans. on Neural Networks,* vol. 4, no. 4, pp. 588-599, July 1993.

[26] J. Choi, B. J. Sheu, C.-F. Chang, "A Gaussian synapse circuit for analog VLSI neural networks," *IEEE Trans. on VLSI Systems,* vol. 2, no. 1, Mar. 1994.

[27] G.J. Dunning, E. Marom, Y. Owechko, B.H. Soffer, "Optical holographic associative memory using a phase conjugate resonator," *SPIE Proc.,* 625, Bellingham WA, Jan. 1986.

[28] L.O. Chua, L. Yang, "Cellular neural network: Theory," *IEEE Trans. Circuits Syst.,* vol. 35, pp. 1257-1272, Oct. 1988.

[29] L.O. Chua, L. Yang, "Cellular neural network: Applications," *IEEE Trans. Circuits Syst.,* vol. 35, pp. 1273-1290, Oct. 1988.

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

[31] H. Harrer, J. A. Nossek, "Discrete-time cellular neural networks," T. Roska, J. Vandewalle, Eds., *Cellular Neural Networks*, West Sussex; England, John Wiley & Sons, 1993.

[32] T. Roska, L.O. Chua, "Cellular neural networks with non-linear and delay-type template elements and non-uniform grids," *Int J. Circuit Theory and Applications,* vol. 20, pp. 469-481, 1992.

[33] J.M. Cruz, L.O. Chua, "A CNN chip for connected component detection," *IEEE Trans. Circuits Syst.,* vol. 38, pp. 812-817, July 1991.

[34] A. Rodriguez-Vazquez, et al., "Current-mode techniques for the implementation of continuous- and discrete-time cellular neural networks," *IEEE Trans. Circuits Syst. Part II,* vol. 40, pp. 132-146, Mar. 1993.

[35] J.E. Varrientos, E. Sanchez-Sinencio, J. Ramirez-Angulo, "A current-mode cellular neural network implementation," *IEEE Trans. Circuits Syst. Part II,* vol. 40, pp. 147-155, Mar. 1993.

[36] H. Harrer, J.A. Nossek, R. Stelzl, "An analog implementation of discrete-time cellular neural networks," *IEEE Trans. Neural Networks,* vol. 3, pp. 466-476, May 1992.

[37] I.A. Baktir, M.A. Tan, "Analog CMOS implementation of cellular neural networks," *IEEE Trans. Circuits Syst. Part II,* vol. 40, pp. 200-206, Mar. 1993.

[38] G.F.D. Betta, S. Graffi, Zs.M. Kovacs, G. Masetti, "CMOS implementation of an analogically programmable cellular neural network," *IEEE Trans. Circuits Syst. Part II,* vol. 40, pp. 206-215, Mar. 1993.

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

[40] B.W. Lee, B.J. Sheu, *Hardware Annealing in Analog VLSI Neurocomputing,* Norwell, MA: Kluwer Academic Publishers, 1991.

[41] S. Bang, B.J. Sheu, "Optimal solutions for cellular neural networks by paralled hardware annealing," submitted for journal publication.

[42] C. Peterson, J.R. Anderson, "A mean field theory learning algorithm for neural networks," *Complex Systems,* vol. 1, no. 5, pp. 995-1019, 1987.

[43] C. Peterson, "Mean field theory neural networks for feature recognition, content addressable memory and optimization," *Connection Science,* vol. 3, pp. 3-33, 1991.

[44] N. Fruehauf, E. Lueder, G. Bader, "Fourier optical realization of cellular neural networks," *IEEE Trans. Circuits Syst. Part II,* vol. 40, pp. 156-162, Mar. 1993.

[45] J.-C. Lee, B. J. Sheu, J. Choi, R. Chellappa, "A mixed-signal VLSI neuroprocessor for image restoration," em Trans. on Circuits and Systems for Video Technology, vol. 2, no. 3, pp. 319-324, Sept. 1992.

[46] J.-C. Lee, B. J. Sheu, W.-C. Fang, R. Chellappa, "VLSI neuroprocessors for video motion detection," *IEEE Trans. on Neural Networks,* vol. 4, no. 2, pp. 178-191, Mar. 1993.

[47] E. Snachez-Sinencio, C. Lau, Eds., *Artificial Neural Networks,* IEEE Press: New York, 1992.

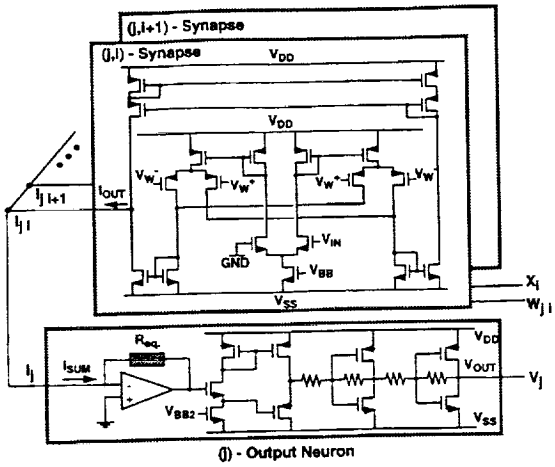


Fig. 1 Circuit schematic of the synapse cell and the output neuron.

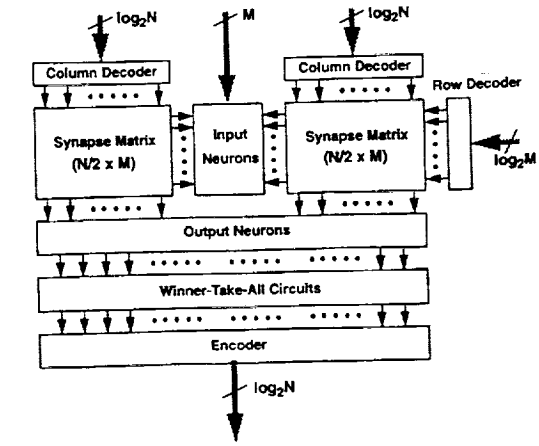


Fig. 2 Schematic diagram of a self-organizing analog neural processor.

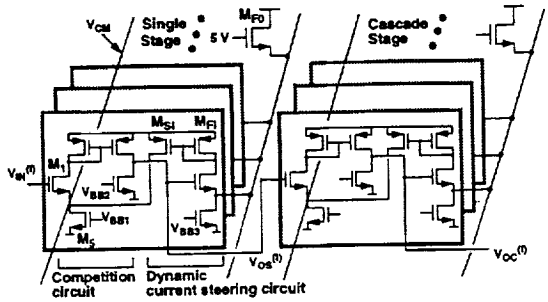
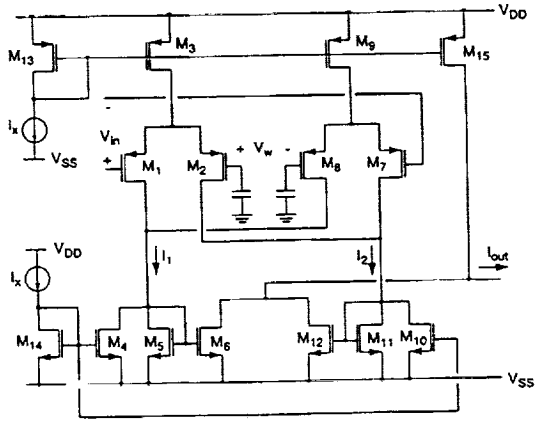
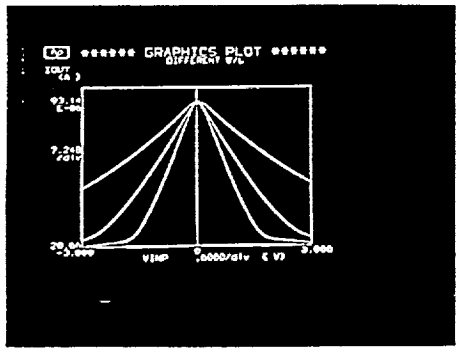


Fig. 3 Circuit schematic of the winner-tale-all function.



(a) Circuit schematic diagram.



(b) Measured results.

Fig. 4 The Gaussian function synapse cell.

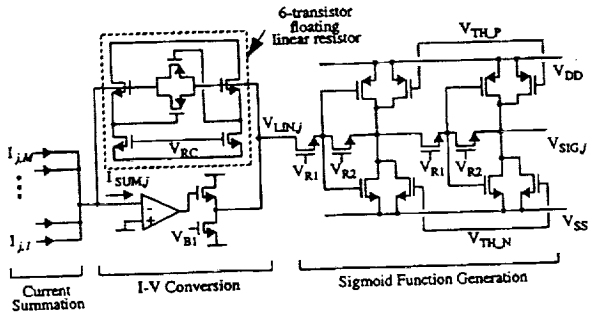


Fig. 5 Circuit schematic of neuron for multi-layered network.

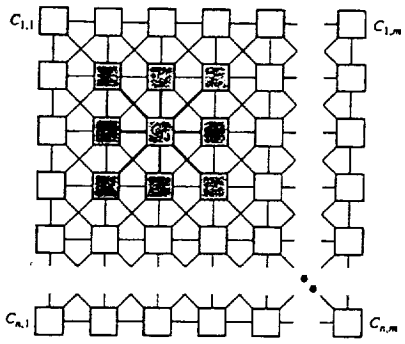


Fig. 6 Cellular neural network.

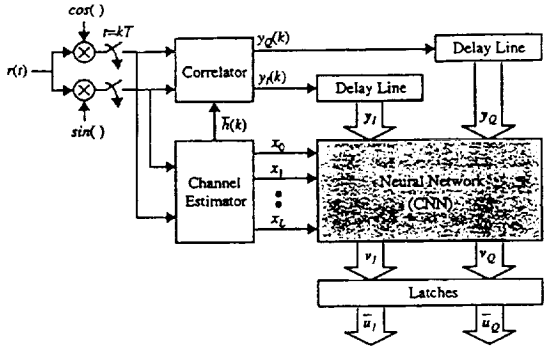


Fig. 7 MLSE application of CNN.

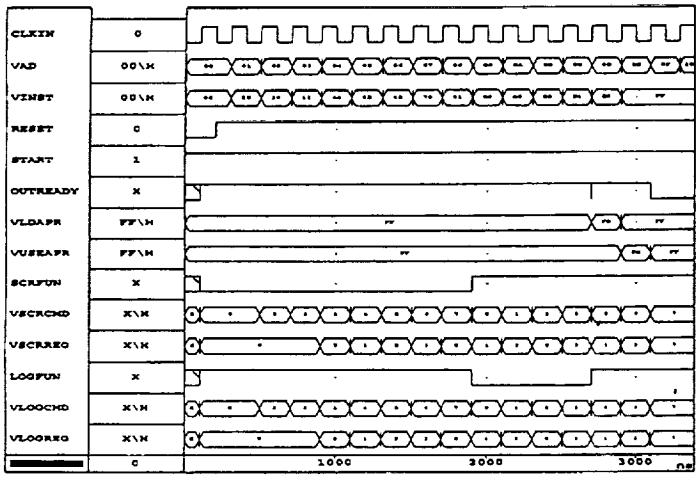


Fig.8 Timing diagram of global control circuit.

VLSI Neuroprocessors

Sabrina Kemeny
Center for Space Microelectronics Technology
Jet Propulsion Laboratory, California Institute of Technology
Pasadena, CA 91109

Abstract

Electronic and optoelectronic hardware implementations of highly parallel computing architectures address several ill-defined and/or computation-intensive problems not easily solved by conventional computing techniques. The concurrent processing architectures developed are derived from a variety of advanced computing paradigms including neural network models, fuzzy logic, and cellular automata. Hardware implementation technologies range from state-of-the-art digital/analog custom-VLSI to advanced optoelectronic devices such as computer-generated holograms and e-beam fabricated Dammann gratings. JPL's Concurrent Processing Devices Group has developed a broad technology base in hardware implementable parallel algorithms, low-power and high-speed VLSI designs and building block VLSI chips, leading to application-specific high-performance embeddable processors. Application areas include high throughput map-data classification using feedforward neural networks, terrain based tactical movement planner using cellular automata, resource optimization (weapon-target assignment) using a multidimensional feedback network with lateral inhibition, and classification of rocks using an inner-product scheme on Thematic Mapper data. In addition to addressing specific functional needs of DoD and NASA, the JPL-developed concurrent processing device technology is also being customized for a variety of commercial applications (in collaboration with industrial partners), and is being transferred to U.S. industries.

This talk will focus on two application-specific processors which solve the computation intensive tasks of resource allocation (weapon-target assignment) and terrain based tactical movement planning using two extremely different topologies. Resource allocation is implemented as an asynchronous analog competitive assignment architecture inspired by the Hopfield network. Hardware realization leads to a two to four order of magnitude speed-up over conventional techniques and enables multiple assignments, (many to many), not achievable with standard statistical approaches. Tactical movement planning (finding the best path from A to B) is accomplished with a digital two-dimensional concurrent processor array. By exploiting the natural parallel decomposition of the problem in silicon, a four order of magnitude speed-up over optimized software approaches has been demonstrated.

Acknowledgments

Anil Thakoor
Taher Daud
Tim Brown
Harry Langenbacher
Tuan Duong
Mua Tran
Silvio Eberhardt
Carlos Villanpando
Helen Tsu

Robert Nixon
Tim Shaw
Eric Fossum
Marc Rieffel
Brad Minch
Doug Kerns

Jointly Sponsored by:
ASAS Program Office, ARPA, NSWC, BMDO, ONR, and NASA

Neural Networks–Cartographic Applications Group

Efforts of the JPL Cartographic Applications Group

Neural Nets & Map Separation

- Extract Roads, Rivers, Urban Areas, etc. out of digitally scanned paper maps & CD-ROM data using trained neural networks. (N. Ritter)

Neural Nets & Multispectral Classification

- Prepare ground-use info and detect man-made features & land-types from hyperspectral airborne AVIRIS data, trained neural networks and ground truth. (N. Ritter)

Concurrent Processing Applications

- TMA: Tactical Movement Analysis finds optimal path over varied terrain using concurrent processing with transputer arrays (T. Kreitzberg)

- LOS: Line of sight computations using tiled data-decomposition with transputers. (T. Kreitzberg)

JPL

Concurrent Processing Devices Group

- Electronic and Optoelectronic Hardware Implementations of Highly Parallel, Neural Network-Inspired Computing Architectures
  - Neurocircuit Simulations of Parallel Architectures
  - Research on Novel Computing and Memory Devices
    amorphous semiconductors, composite oxides, and ferroelectric materials
  - Hardware Implementations of Custom VLSI ICs that can be Embedded into Neuroprocessor Cards

Flexible General purpose Building Block Chips
- Cartographic analysis
- Adaptive Controller for Interceptor
- Space Environmental Test

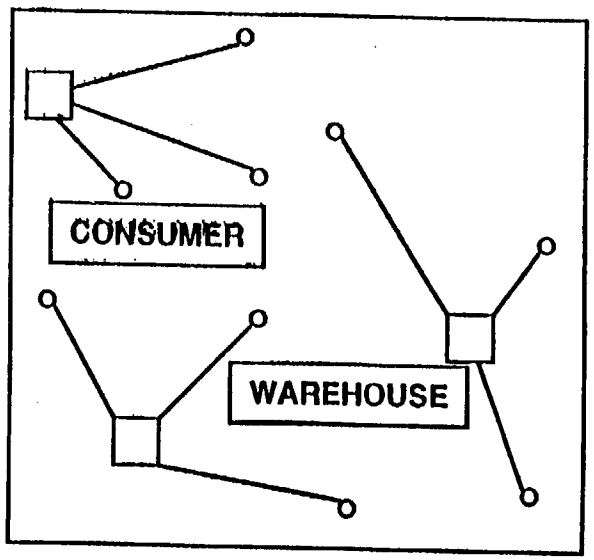
Application Specific ICs
- Resource Allocation
- Path Planning
- Image Compression

- Technology Transfer to U.S. Industry Ongoing
- Development over the Past 8 Years Sponsored By:
  NASA, BMDO/IST, ARPA, CECOM/IEWD, ONR, NSWC, NAWC, CSDL, USA/SSDC, AFOSR, and Industry

RESOURCE ALLOCATION / ASSET MANAGEMENT

- *DEVELOPED A NEW BREAKTHROUGH CONCEPT FOR HARDWARE IMPLEMENTATION OF A NEUROPROCESSOR FOR HIGH SPEED SOLUTIONS TO DYNAMIC ASSIGNMENT PROBLEMS.*

- RESOURCE ALLOCATION
- DYNAMIC ASSIGNMENT
- MESSAGE ROUTING
- TARGET-WEAPON PAIRING
- LOAD BALANCING
- MULTI-TARGET TRACKING
- SEQUENCING / SCHEDULING
- ASSET MANAGEMENT
- REAL-TIME, ADAPTIVE MISSION RE-PLANNING



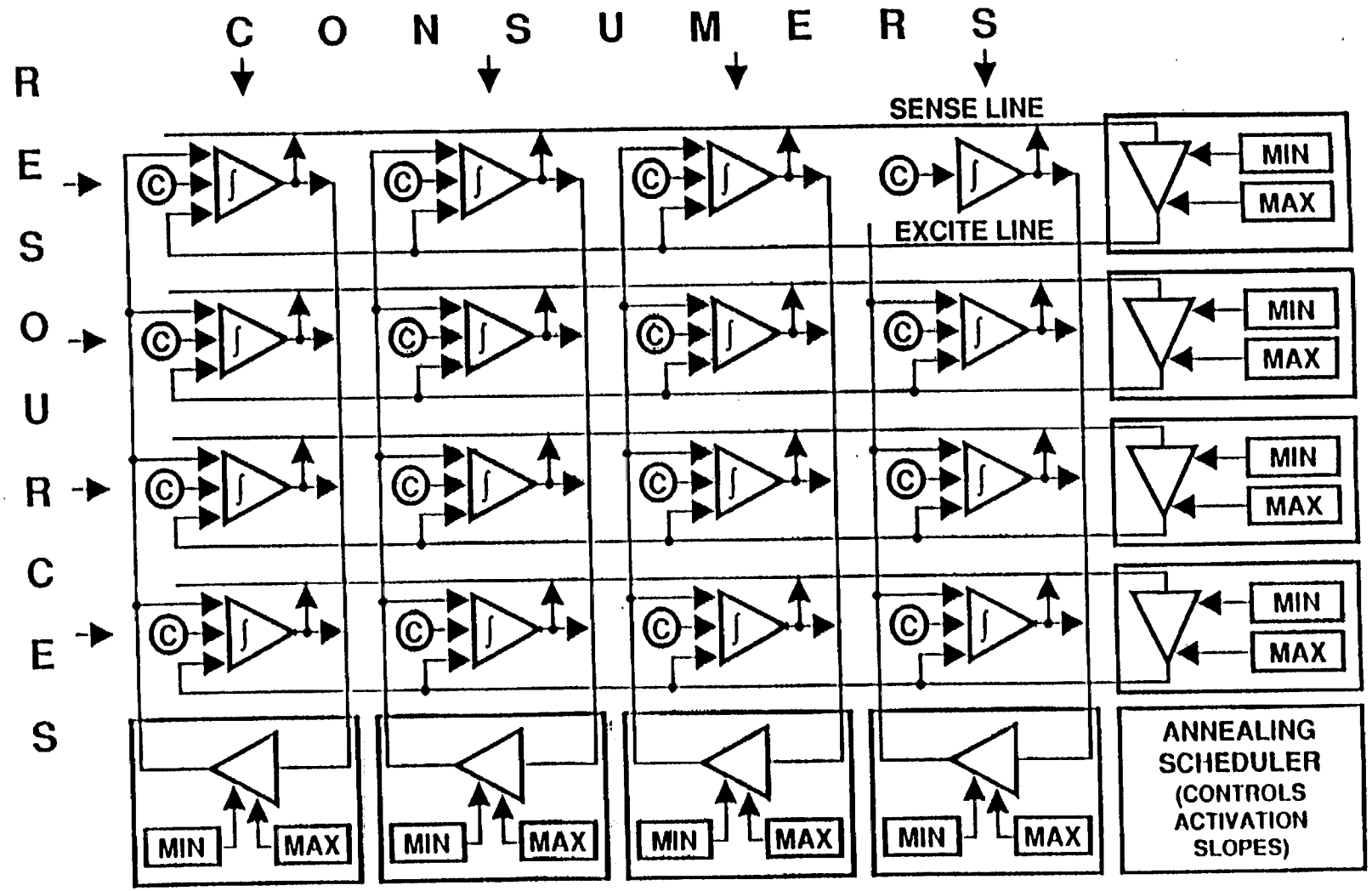
3 WAREHOUSES AND 9 CONSUMERS

ASSIGN

3 CONSUMERS TO EACH WAREHOUSE TO OBTAIN MINIMUM OVERALL COST

- *NEUROPROCESSING APPROACH OFFERS OVER 4 ORDERS OF MAGNITUDE SPEED ENHANCEMENT OVER CONVENTIONAL COMPUTING TECHNIQUES.*
- *ARBITRARY MULTIPLE (MANY - TO - MANY) ASSIGNMENTS ARE MADE, WHICH ARE NOT EASILY ACCOMPLISHED BY CONVENTIONAL TECHNIQUES.*

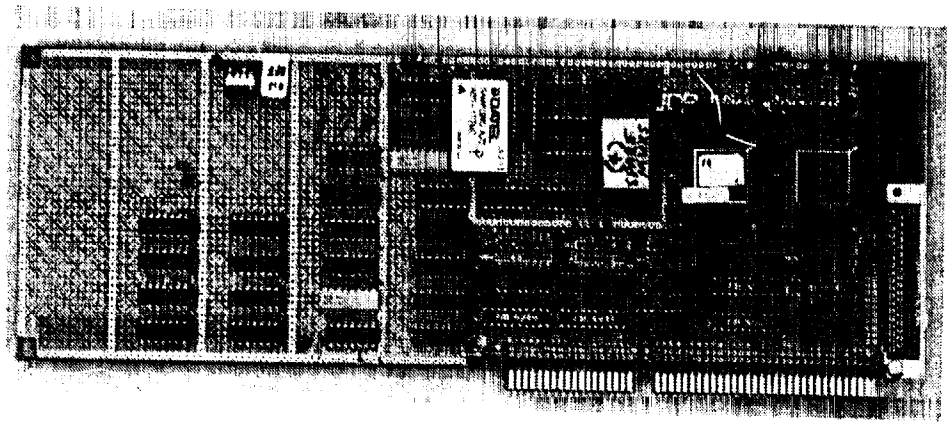
Asset Management Neuroprocessor IC Architecture



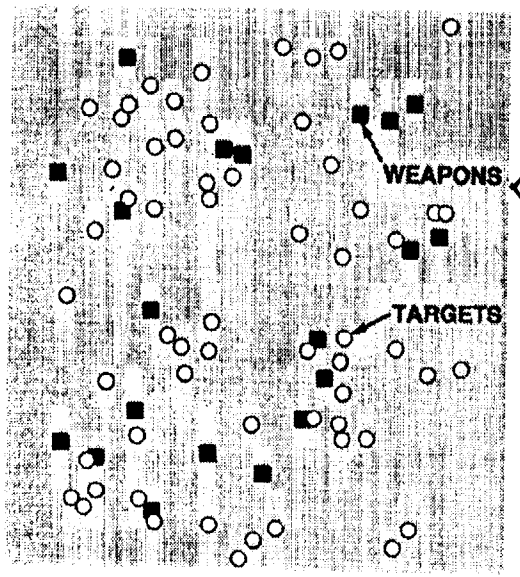
JPL

NEUROPROCESSOR FOR WEAPON-TARGET ASSIGNMENT

PLUG-IN NEUROPROCESSOR CARD

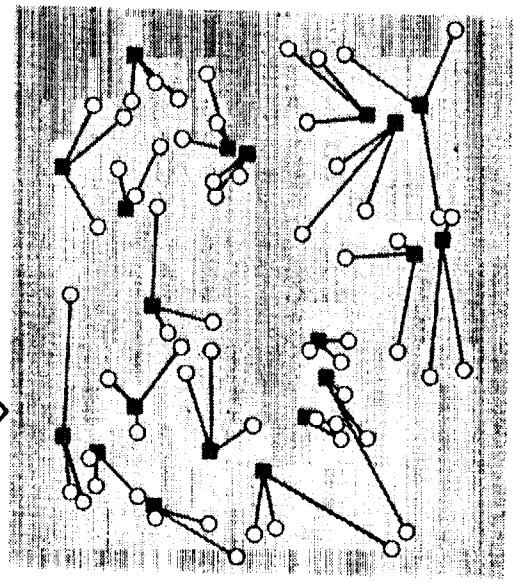


- FULLY PARALLEL ARCHITECTURE OF NEURAL NET HARDWARE PERFORMS DYNAMIC ASSIGNMENT OF RESOURCES
- NEUROPROCESSING SPEED SURPASSES THAT OF AN 8-NODE HYPER-CUBE BY TWO ORDERS OF MAGNITUDE
- A 40X40 ASSIGNMENT CHIP WITH 1600 ANALOG NEURONS GIVES A "GOOD" SOLUTION OUT OF A TOTAL OF OVER 10^{48} POSSIBLE COMBINATIONS
- FOR THE FIRST TIME IT ENABLES MANY-TO-MANY ASSIGNMENTS



A 60 TARGETS TO 20 WEAPONS (3-TO-1) ASSIGNMENT

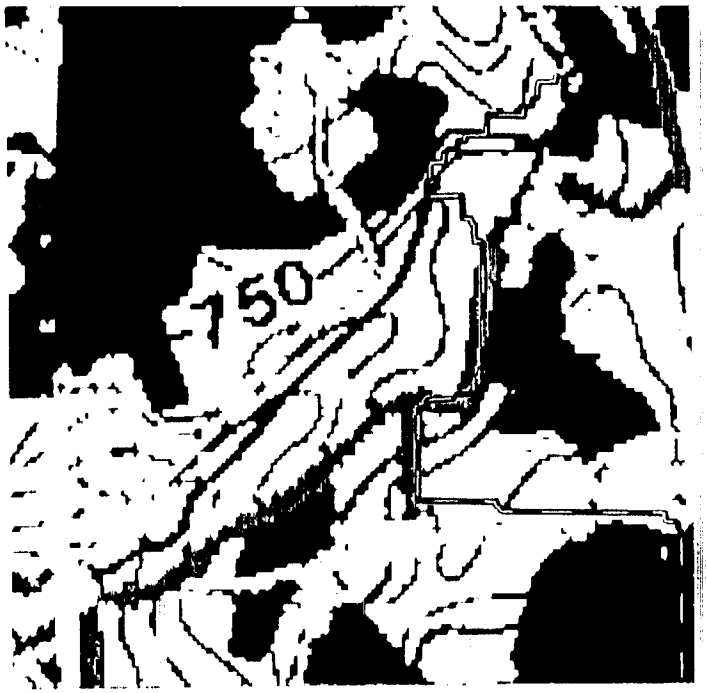
SIMULATION OF HARDWARE PROMISES ASSIGNMENT IN ABOUT 30 MICROSECONDS



JPL

PATH PLANNING

OBJECTIVE: TO FIND LEAST COST (MINIMIZING TIME, FUEL, ATTRITION, ETC.) PATH ACROSS COMPLEX TERRAIN QUICKLY (ms)



APPLICATION AREAS

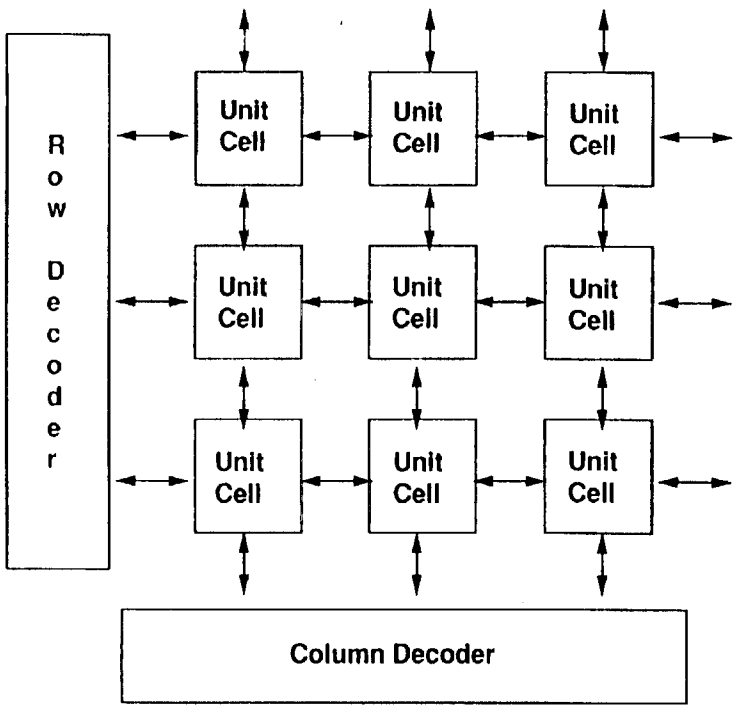
- DEFENSE
  - TACTICAL MOBILITY ANALYSIS FOR BATTLEFIELD SCENARIOS
  - MISSION PLANNING: REAL TIME ROUTE PLANNING AND EN ROUTE RE-ROUTING FOR FLIGHT PLATFORMS
- EMERGENCY DISPATCHING (CIVILIAN AND MILITARY)
- AUTONOMOUS VEHICLE NAVIGATION
- CIRCUIT BOARD WIRE ROUTING/MAZE NAVIGATION

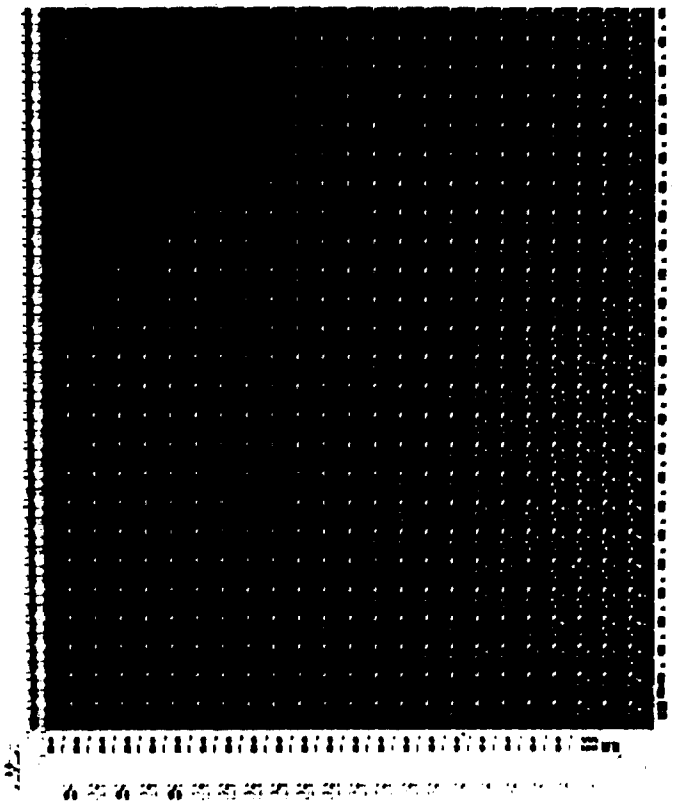
JPL

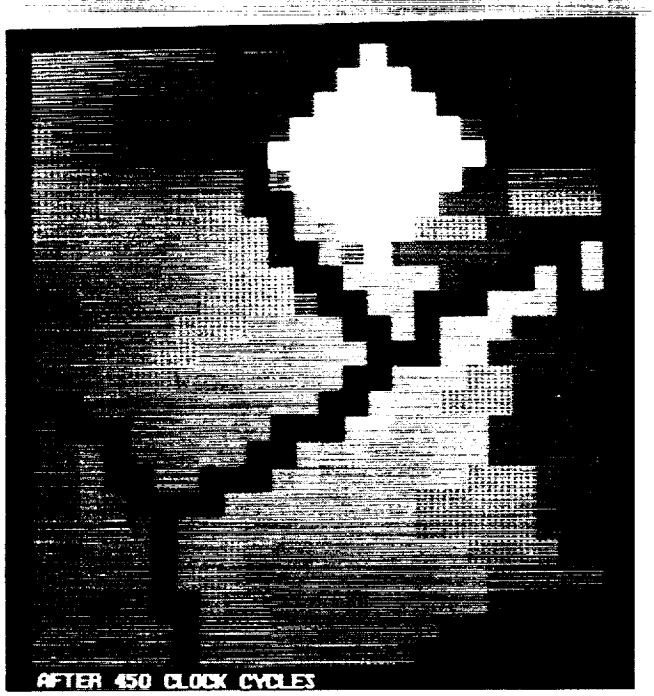
PATH PLANNING APPROACH

PROBLEM: CURRENT SOFTWARE APPROACHES REQUIRE SECONDS TO MINUTES TO COMPUTE LEAST COST PATH

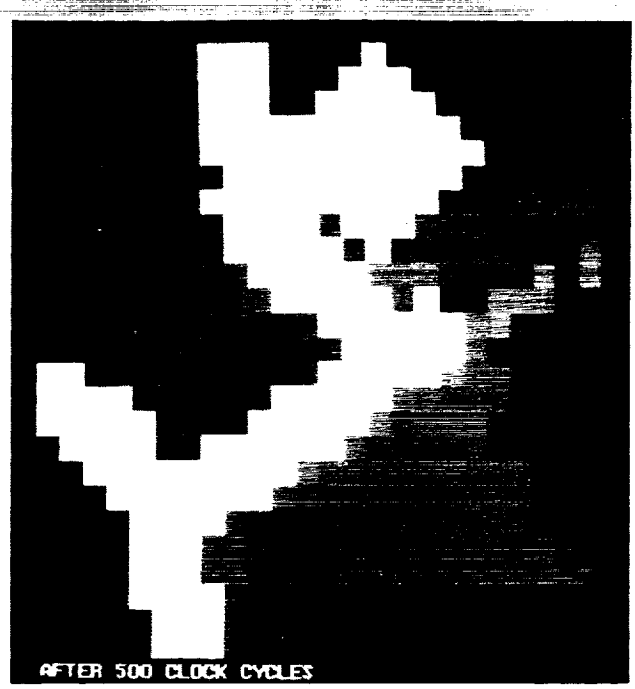
SOLUTION: VLSI INTEGRATION OF A FINE GRAIN PARALLEL PROCESSOR ARRAY PROGRAMMED TO MODEL A GIVEN TERRAIN AND DETERMINE THE LEAST COST PATH IN MILLISECONDS



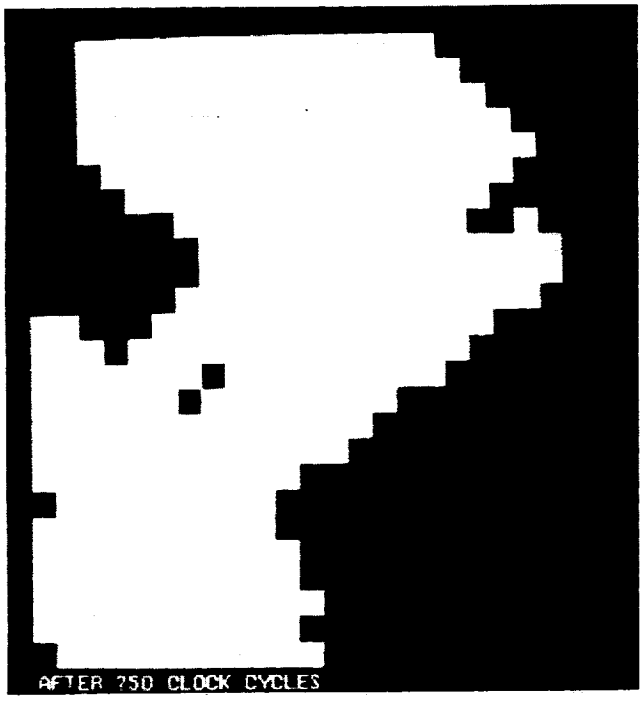




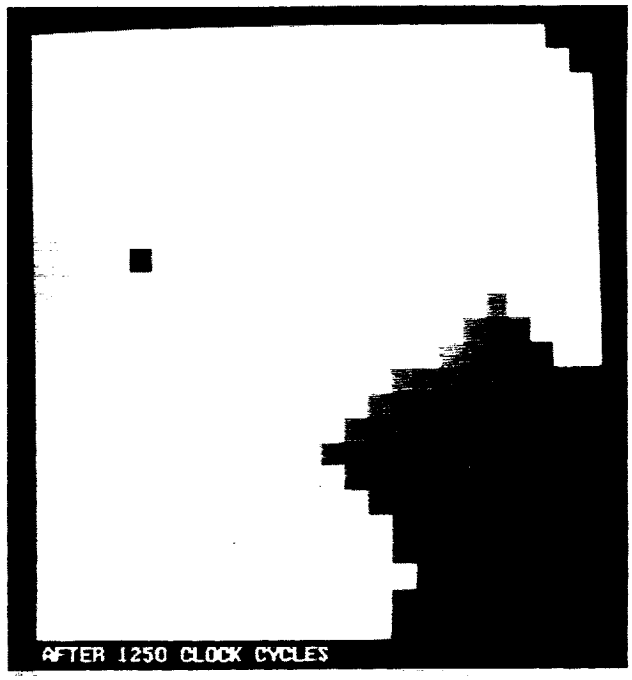
(a)



(b)



(c)

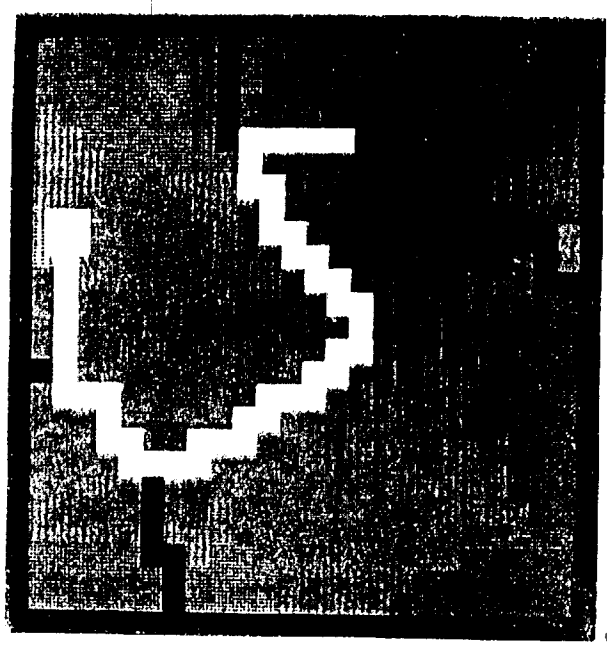


(d)

Signal propagation through array shown in white on map background (black indicates road): a) after 450 clock cycles, b) after 500 clock cycles, c) after 750 clock cycles, and d) after 1250 clock cycles.

JPL PATH PLANNER PERFORMANCE SUMMARY

TYPICAL LEAST COST PATH



IC CHARACTERISTICS

CHIP ARCHITECTURE:	24 x 25 DIGITAL PROCESSOR ARRAY
MAXIMUM CLOCK FREQUENCY:	7 MHz
EQUIVALENT OPERATIONS PER SECOND:	6 BILLION
ORIGINATION NODES:	ONE OR MULTIPLE
COST DYNAMIC RANGE:	256:1
PROCESS:	2 μm CMOS
UNIT CELL (PROCESSOR) SIZE:	296 μm x 330 μm
IC SIZE:	7.9 mm x 9.2 mm

AVERAGE PATH DETERMINATION SPEEDS

TERRAIN GRID DIMENSION	CURRENT SOFTWARE (N^2 ALGORITHM, 4 TRANSPUTERS)	PATH PLANNER IC (7 MHz CLOCK)
24 x 25	—	0.23 ms
64 x 64	4 sec	0.58 ms
128 x 128	17 sec	1.17 ms
256 x 256	74 sec	2.34 ms
512 x 512	320 sec	4.68 ms

JPL

Path Planning

SUN Software Simulation

- Ported Vector-Cost Path Planner Software to SUN (programmable cost for each of 8 directions)
  - **Portable** - will run on any workstation that supports UNIX, Xwindows, and Motif
  - **Modular** - can easily replace software modules with hardware
  - **Friendly Graphics Interface** pushbuttons, pop-up menus, scroll bars

JPL

CONCLUSIONS

- **Neural Networks have Matured, Enabling Solutions to a Variety of Ill-Defined, and/or Computation-Intensive Problems**
- **Ongoing Technology Insertion Strategies**
  - **Embeddable NN Hardware (User Transparent, Inexpensive)**
    - » **PC Co-Processor Boards**
    - » **Small Stand-Alone Packages (e.g. automotive "under the hood", medical devices)**
- **Further Developments in Novel Architectures will Lead to Enabling New Capabilities**
    - » **3-D ULSI/Optoelectronic Implementations**

PHOTONICS: From Target Recognition to Lesion Detection

Dr. E. Michael Henry

Manager, Photonic Systems
Martin Marietta Corporation

(303) 977-7720

MARTIN MARIETTA

Photonics: From Target Recognition to Lesion Detection

Martin Marietta Corporation and Rose Health Care Systems
by Dr. E. Michael Henry, (303)977-7720
Martin Marietta Astronautics, MS FO330
P.O. Box 179, Denver, Colorado, 80201

Introduction -- Since 1989, Martin Marietta has invested in the development of an innovative concept for robust real-time pattern recognition for any two-dimensional sensor. This concept has been tested in simulation, and in laboratory and field hardware for a number of DoD and commercial uses from automatic target recognition to manufacturing inspection. We have now joined Rose Health Care Systems in developing its use for medical diagnostics.

The Concept -- The concept is based on determining regions of interest by using optical Fourier bandpassing as a scene segmentation technique, enhancing those regions using wavelet filters, passing the enhanced regions to a neural network for analysis and initial pattern identification, and following this initial identification with confirmation by optical correlation. The optical scene segmentation and pattern confirmation are performed by the same optical module. The neural network is a recursive error minimization network with a small number of connections and nodes that rapidly converges to a global minimum.

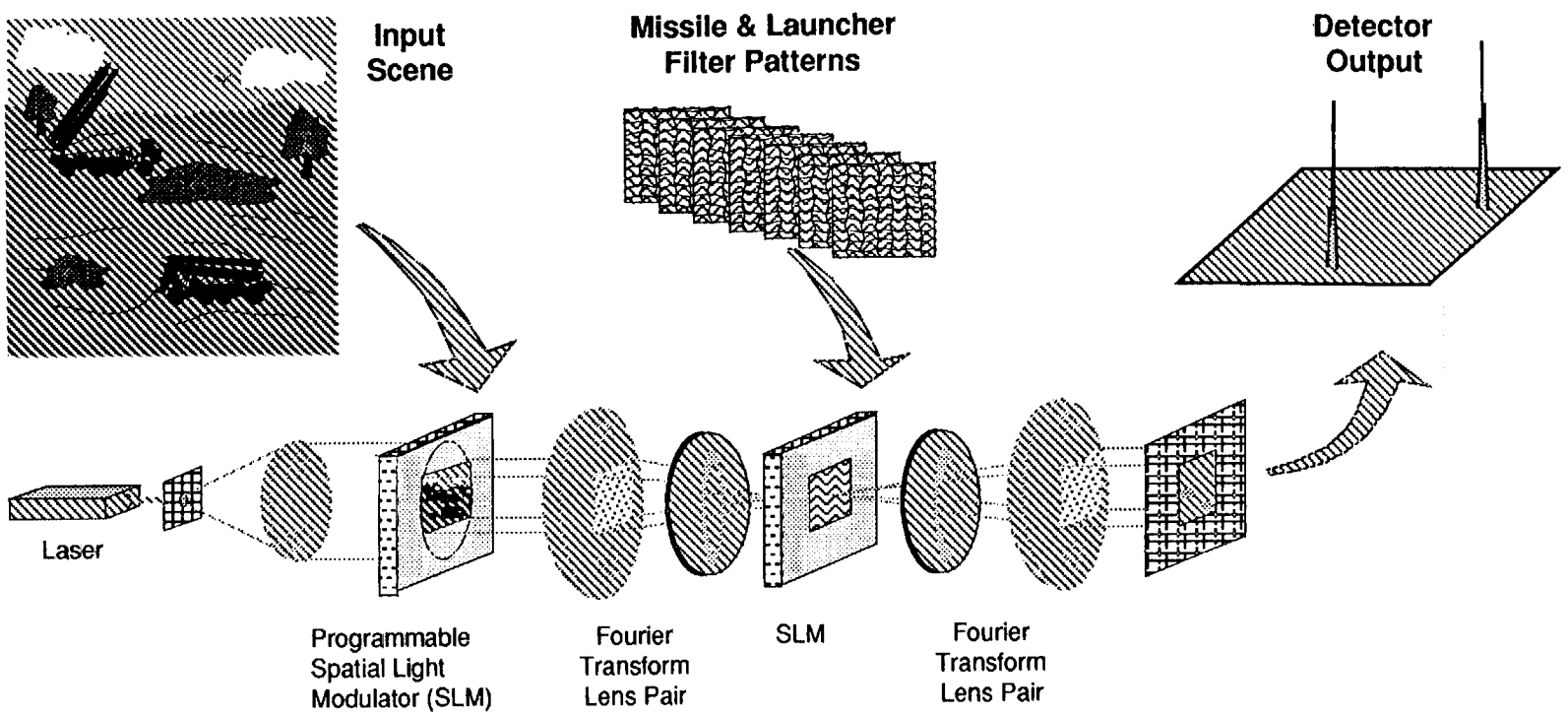
A Specific National Need -- The specific commercial application for which this Defense technology is proposed is a medical diagnostics demonstration in analyzing screening mammograms. Breast cancer is an ever-increasing problem that is striking women at younger and younger ages. Recent statistics indicate that one in eight women will experience breast cancer in their lifetimes--an increase from one in twelve a few years ago. One of the most effective tools in the fight against breast cancer is early detection through the use of mammography. In 1990, 17 million screening mammogram sets were generated. Based on National Cancer Institute and American Cancer Society recommendations, 44 million sets should have been processed. While there are several barriers to greater mammography participation, one barrier is cost. Radiologist reading fees alone for screening mammograms amounted to $652 million in 1990 and are expected to grow to $1 billion by 1996. Statistics also show that early detection of breast cancer not only saves lives, but significantly reduces the cost of the ensuing

treatment as well. Our goal is to reduce screening mammogram fees to increase participation, to aid radiologists in finding a higher percentage of cancerous lesions, and to detect these lesions at least a year earlier than is generally possible with current techniques.

The On-going Effort -- Martin Marietta and Rose Health Care Systems are conducting demonstrations of the concept for mammogram processing. These demonstrations use an optical processor simulator to detect and identify spiculated lesions -- one of three types of potentially cancerous lesions commonly detectable in the human breast, and will be extended to detect the other lesions as well. The effort will then conduct a full proof of concept through simulation and hybrid digital/optical hardware for all three lesion types, prepare a system operational concept, develop a total system prototype for evaluation tests, and prepare for FDA clinical trials and manufacturing readiness. The Martin Marietta/Rose mammogram analysis system has the potential to significantly reduce total mammography costs, while improving the quality of care by ultimately functioning as a radiologist's aid as well as an automatic prescreener or a "second opinion" system. Mammography is only the first of a number of applications to medical diagnostics for which this technology could be key. We expect to expand its use to the analysis of chest imagery, pap smears and other similar image and cytological diagnostics.

The Team -- The team is composed of Martin Marietta Photonic Systems as system developer and team administrator and Rose Health Care Systems as partner and key medical advisor on radiology and operational concepts. Optics and neural network experts from the University of Colorado, the University of Dayton Research Institute, and Tactical Technical Solutions, Inc., are providing technical support in pattern processing. Two nationally-known radiologists provide additional expertise in mammogram analysis techniques, and the Eastern Cooperative Oncology Group, a group of over 3000 cancer research professionals, provides guidance on this and other diagnostic areas for which these techniques apply. Several local suppliers provide assistance in the human-machine interface for medical diagnostic workstations, in clinical evaluation requirements and techniques, and in system packaging.

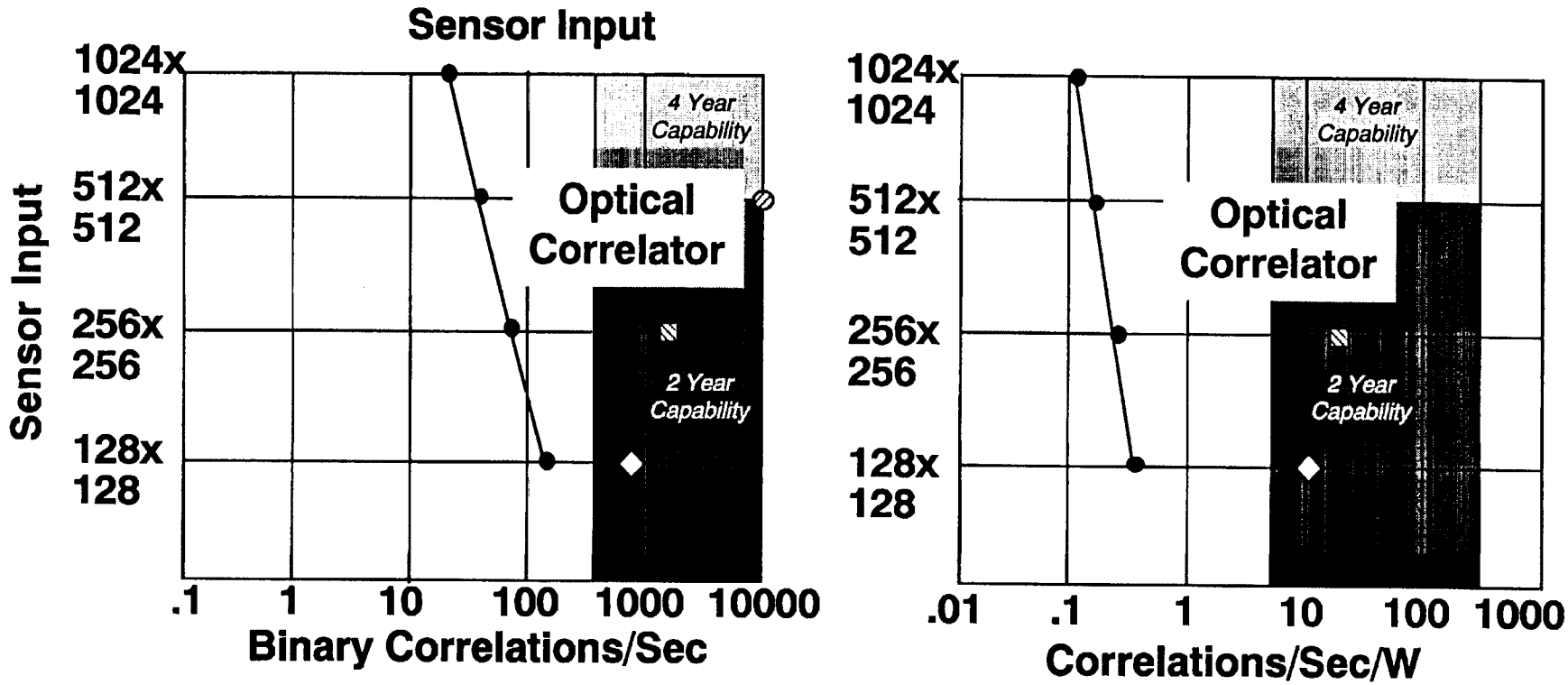
Optical Pattern Recognition



- Inherently Massively Parallel (Entire Frame Simultaneously)
- Excellent Discrimination, Low False Alarm Rates
- Low Power, Light Weight, Small Volume
- Frame Rate Essentially Independent of Sensor Resolution

Optical -- Electronic Correlation Comparison

Photonic Systems



◇ TOPS Martin Marietta Compact Correlator

▧ 1994 Martin Marietta Compact Correlator

● Typical Electronic Parallel Processor

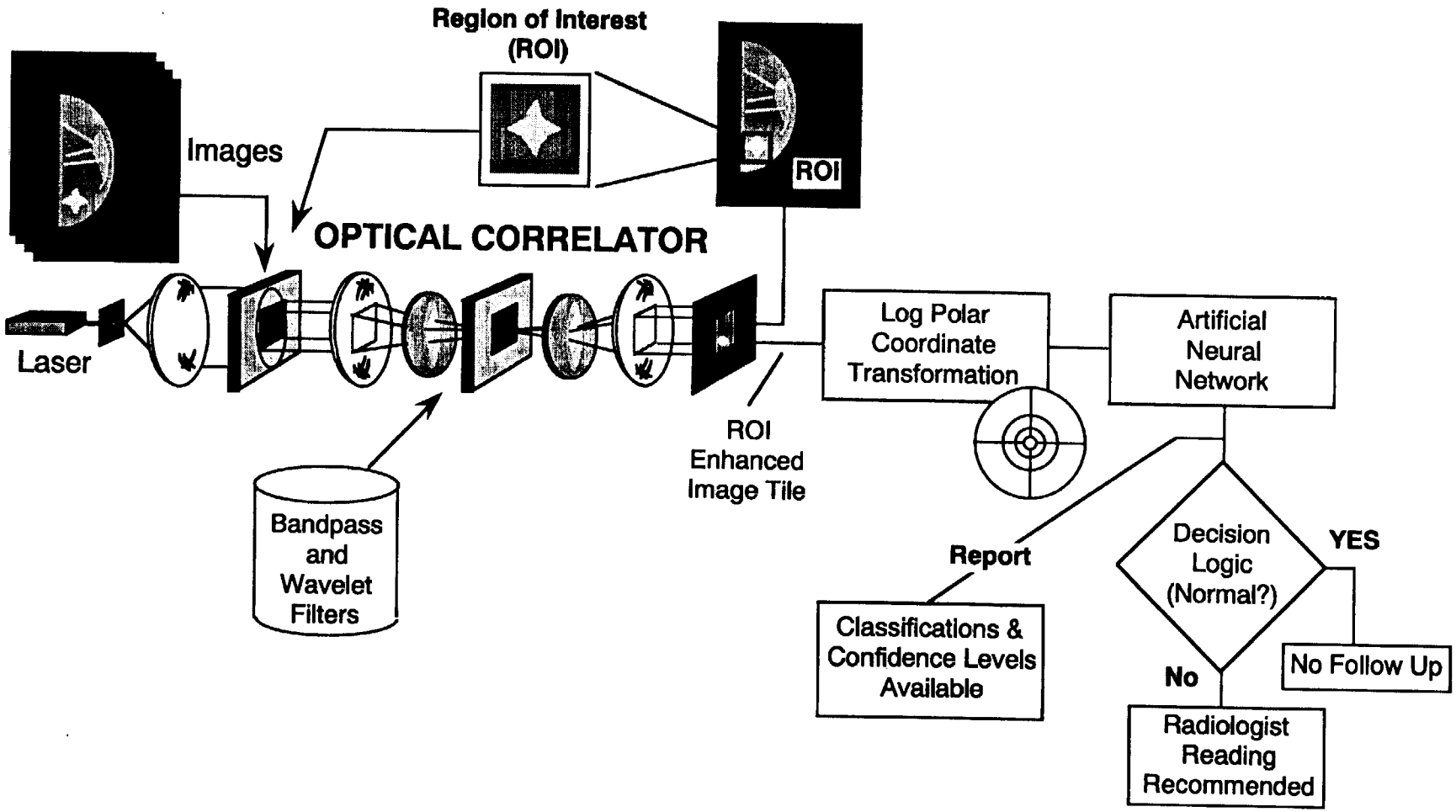
⊘ Approximate Teraflop Throughput

Breast Cancer Detection

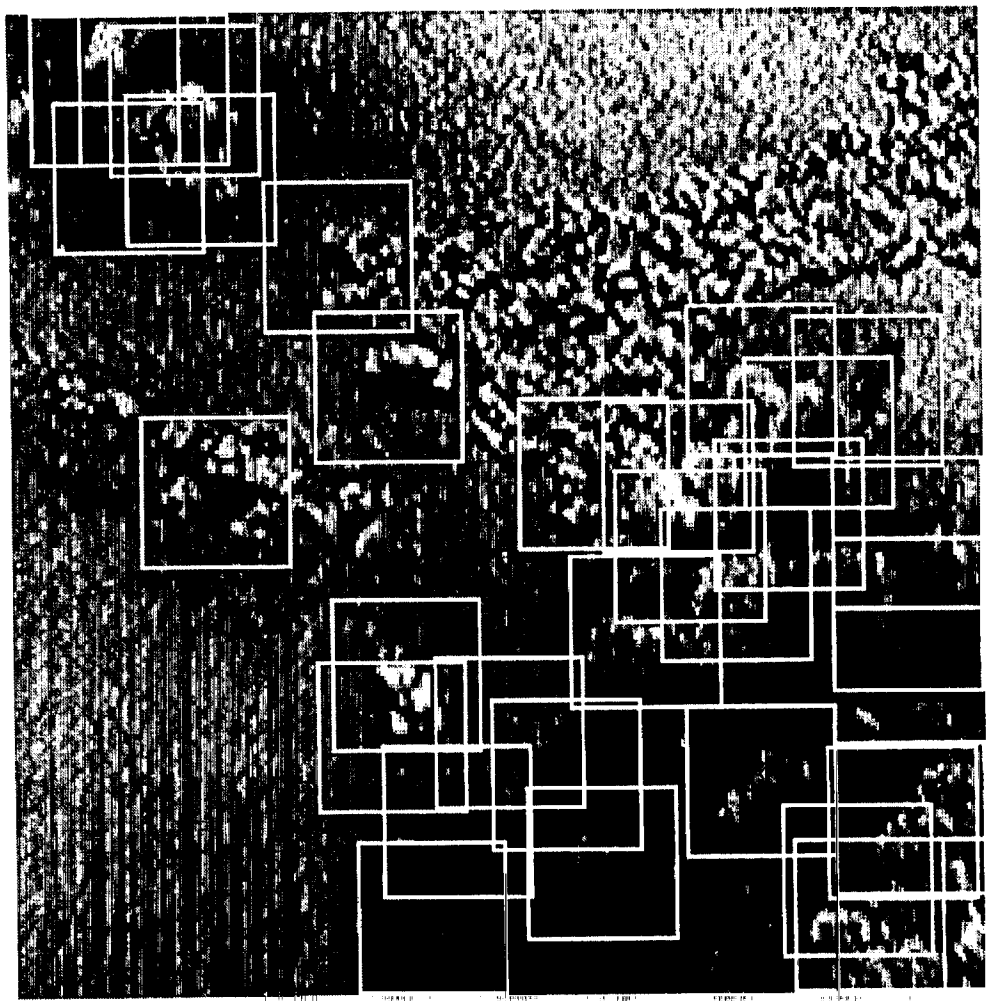
- **Breast Cancer**
  - ~200,000 Cases per year
  - ~50,000 Deaths per year
  - 21 million Screening Mammogram Sets (USA, 1992)
- **Detection**
  - Screening (Mammography and *Interpretation*)
    - » Mammograms (X-ray films)
  - Diagnostics
    - » Alternate Views (X-ray films)
    - » Ultrasound
    - » Biopsy
- **Screening Leads to**
  - Early Detection - Prior to Palpable State
  - Less Radical and Costly Cures
  - Higher Chance of Survival

Screening Mammogram Analysis Concept

Photonic Systems

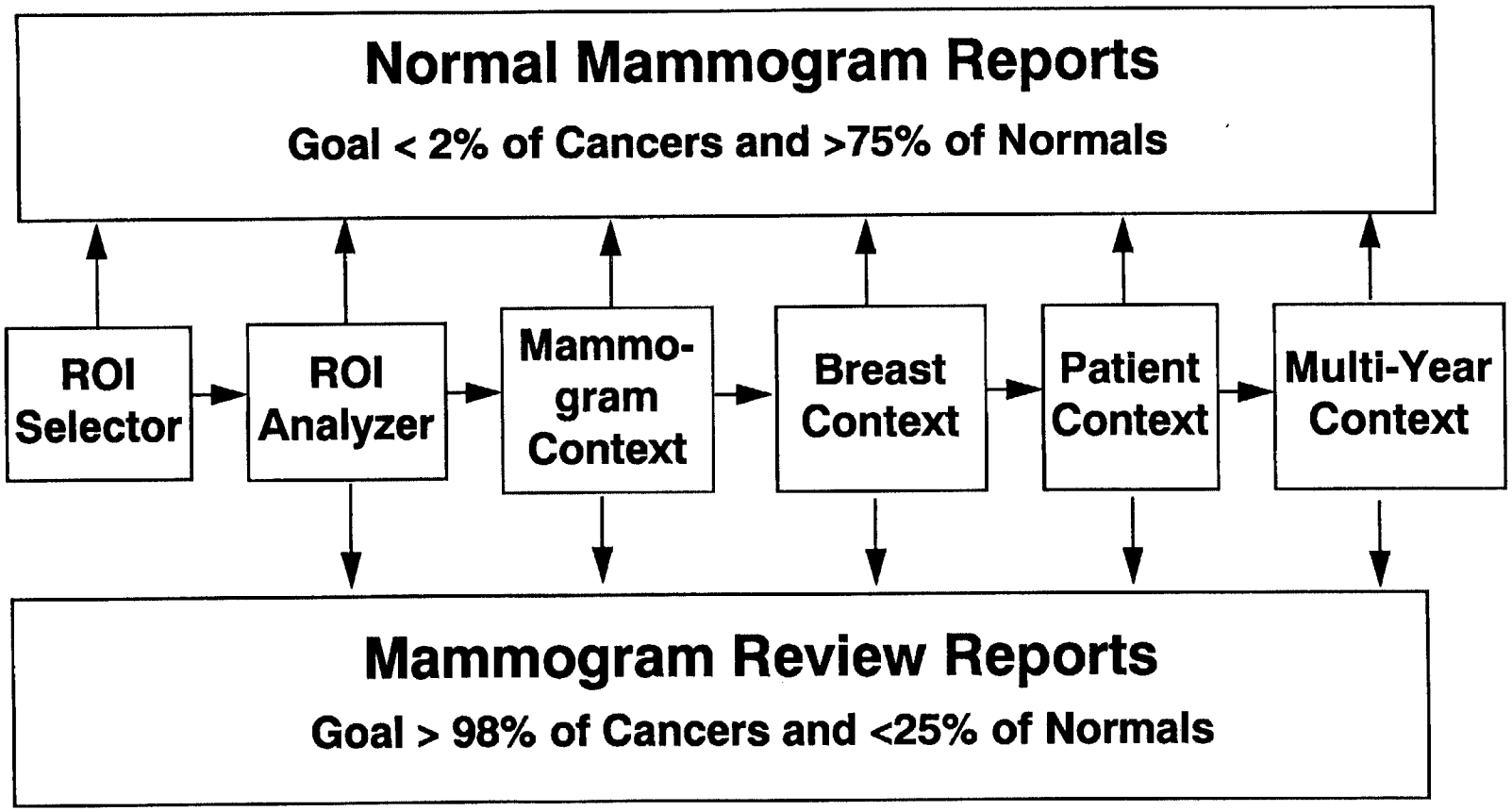


Locating ROIs for ATR



Locating ROIs in Mammograms

System Approach



ROSE HEALTH CARE SYSTEM/MARTIN MARIETTA

Automated Mammogram Screening Project

"If we have the technology sophisticated enough to direct missiles to target thousands of miles away, then we ought to work to have technology sophisticated enough to detect every fatal lump in a woman's breast."

Hillary Clinton
July 19, 1993

3D ARTIFICIAL NEURAL NETWORK (3DANN) TECHNOLOGY

A Status Report And Blueprint For The Future

Irvine Sensors Corporation Presentation to the Workshop:

"A DECADE OF NEURAL NETWORKS: PRACTICAL APPLICATIONS AND PROSPECTS"

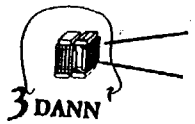
by

John Carson

3D Artificial Neural Network (3DANN) Technology
A Status Report and Blueprint For The Future

Irvine Sensors Corporation (ISC), working closely with JPL under BMDO/ONR sponsorship, is developing a radically new neural computing technology. Primarily aimed at discrimination and target recognition for BMDO missile interceptor applications, it appears to have near term commercial applicability to such problems as handwriting and face recognition, just to name two. In its earliest form it will be able to perform inner product computation using 262 thousand 64x64 templates (weighted synapse arrays) where the 64^5 weights can all be changed every milli-second. Internal switching provides an inherent capability to zoom,. translate, or rotate the templates. The 3D silicon architecture is manufactured on a commercial, high volume DRAM production line at very low cost, enabling its commercialization. Two technology thrusts are beginning: In the first, the 64 layer capability of 3DANN-I will be extended to 1024 layers and beyond. In the second layer size will be shrunk to 2-3 millimeters to reduce layer costs to under fifty cents.

Our workshop goal is to expose this technology to the neural network community as an emerging tool for their use and to obtain their desirement for its future development.



The Silicon Neuron Seeker

BMDO
Ballistic Missile Defense Organization

IRVINE SENSORS CORPORATION

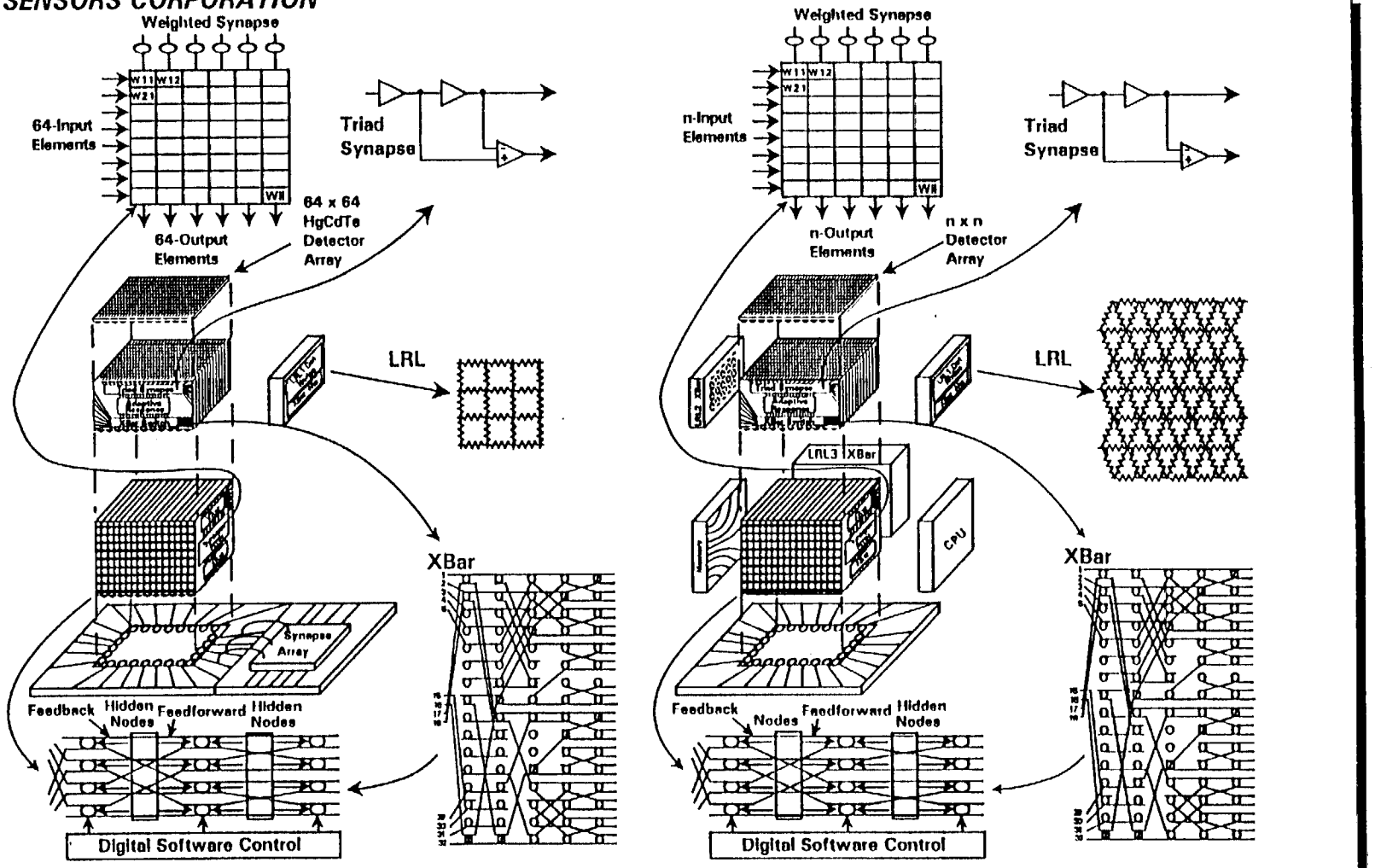
- Frame Rate > 1,000Hz
- Real Time Acquisition, Discrimination & Homing
- Human Level Recognition

The 3DANN FPA Implements Many Neural Network Architectures & Algorithms Real Time

BMDO
Ballistic Missile Defense Organization

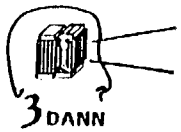
3DANN

IRVINE SENSORS CORPORATION



The 3DANN-I FPA

The General 3DANN FPA Concept



SILICON NEURON SEEKER TECHNICAL OVERVIEW

BMDO
Ballistic Missile Defense Organization

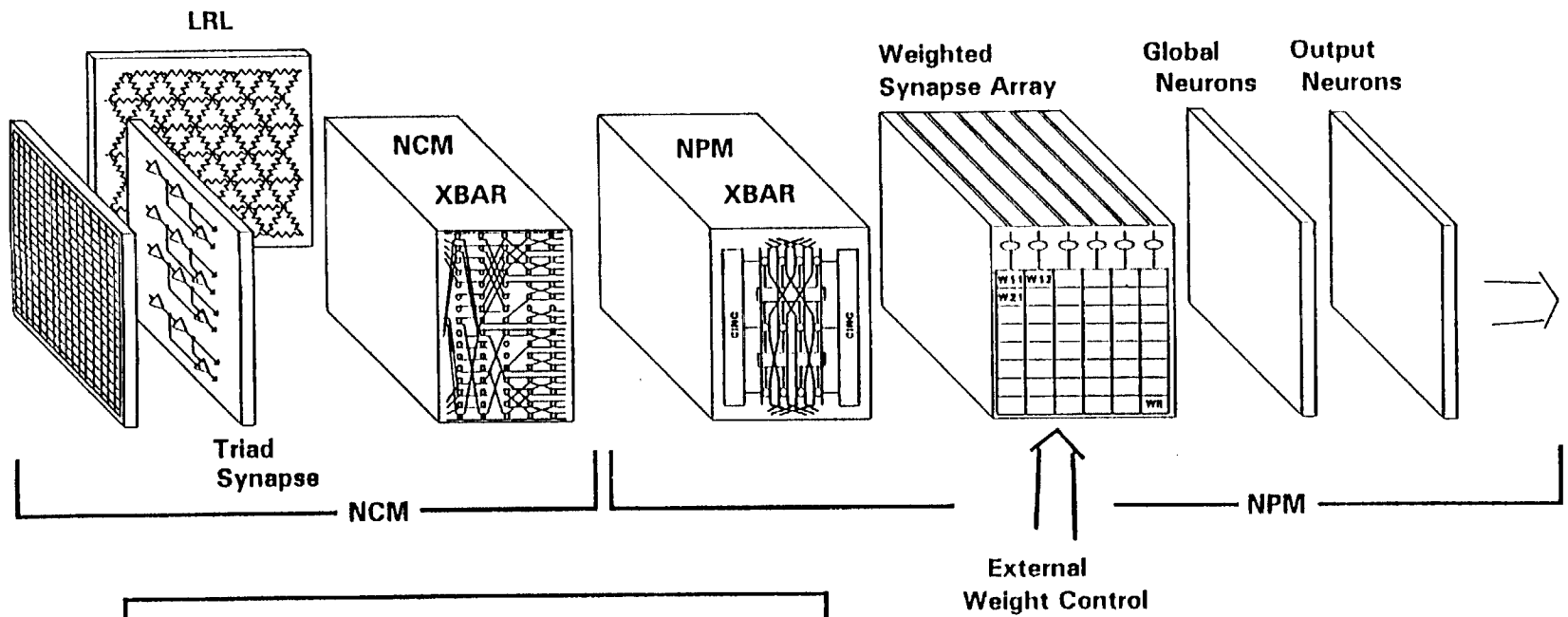
IRVINE SENSORS CORPORATION

- SIGNIFICANT ARCHITECTURAL ISSUES IDENTIFIED AND ADDRESSED
  - Windowing and optimum allocation of weight space impacted the NCM design
  - A promising new approach based on a JPL innovation to be considered for 3DANN-II
- NCM DESIGN NOW COMPLETE AND AT FINAL LAY-OUT
  - Fast cross-bar to facilitate windowing
  - Post-layout simulation results appear excellent
- NPM IC CRITICAL CELLS THROUGH TEST AT JPL
- 3DANN-I HARDWARE EMULATOR NEARING COMPLETION
  - NCM and NPM modules at test
  - Major process hurdles overcome for multi-face processing and bump-bonding
- HARDWARE SIMULATION TOOLS IN PLACE AND DEMONSTRATED

3DANN Architecture

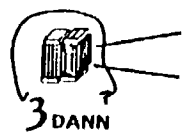
BMDO
Ballistic Missile Defense Organization

IRVINE SENSORS CORPORATION



- n^2 unique paths/weights per pixel
- n^4 x n weight space
- n^3 n x n templates
- n^5 weight changes per frame
- One frame = 10^{-3} seconds

	n = 64	n = 1024
Weights	10^9	10^{15}
Templates (nxn)	262K	10^9



Window Grabber Architecture

BMDO
Ballistic Missile Defense Organization

IRVINE SENSORS CORPORATION

n^3 Weights

n^1 Templates of n x n

Baseline Perfect Shuffle Architecture

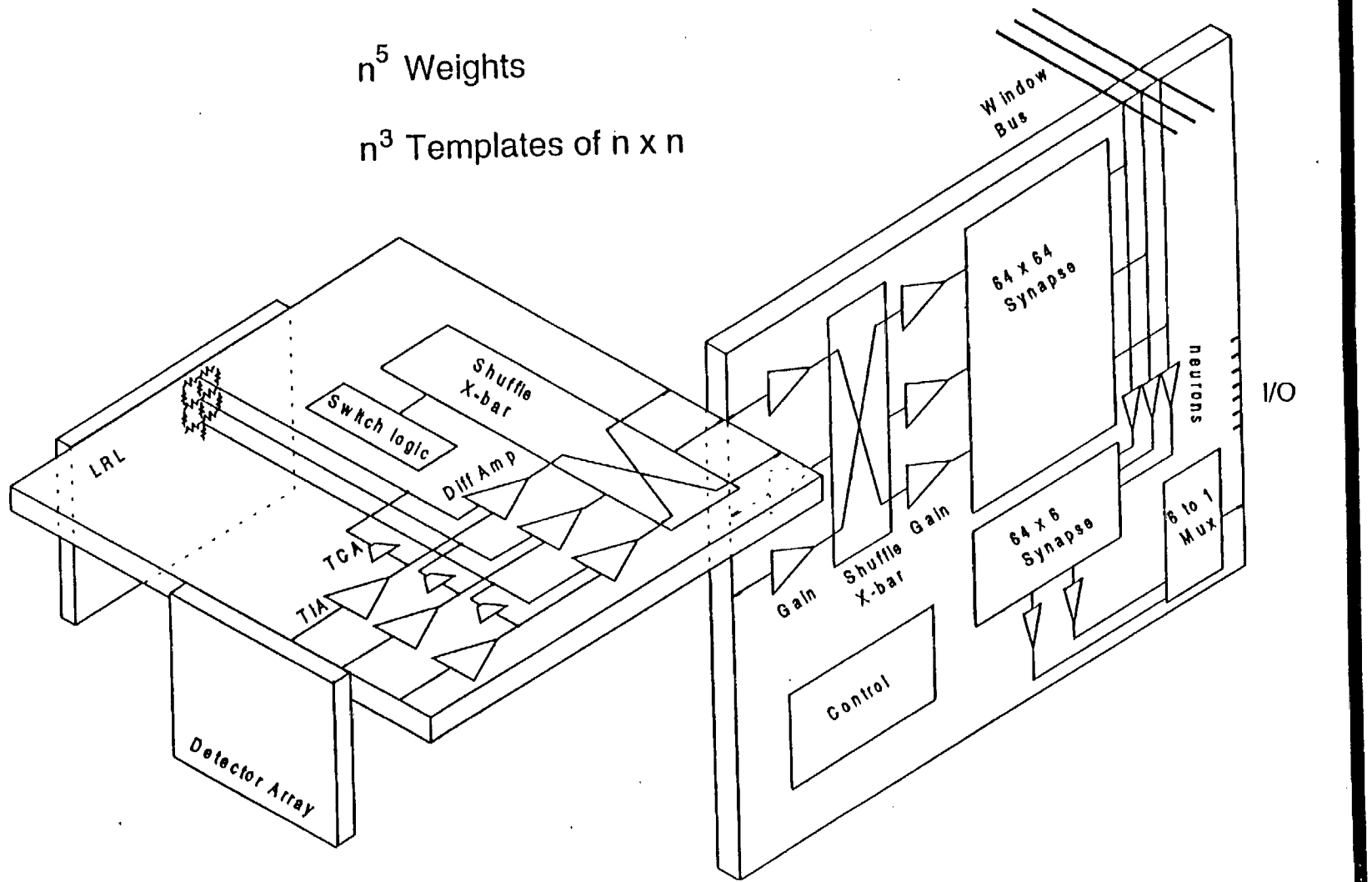
BMDO
Ballistic Missile Defense Organization

3 DANN

IRVINE SENSORS CORPORATION

n^5 Weights

n^3 Templates of n x n



Legacy to Military Applications

BMDO
Ballistic Missile Defense Organization

3DANN

IRVINE SENSORS CORPORATION

Neural Network Technology

Cap Chip Technology

Memory Products

3DANN-I

BMDO

Interface to standard imagers

Flexman

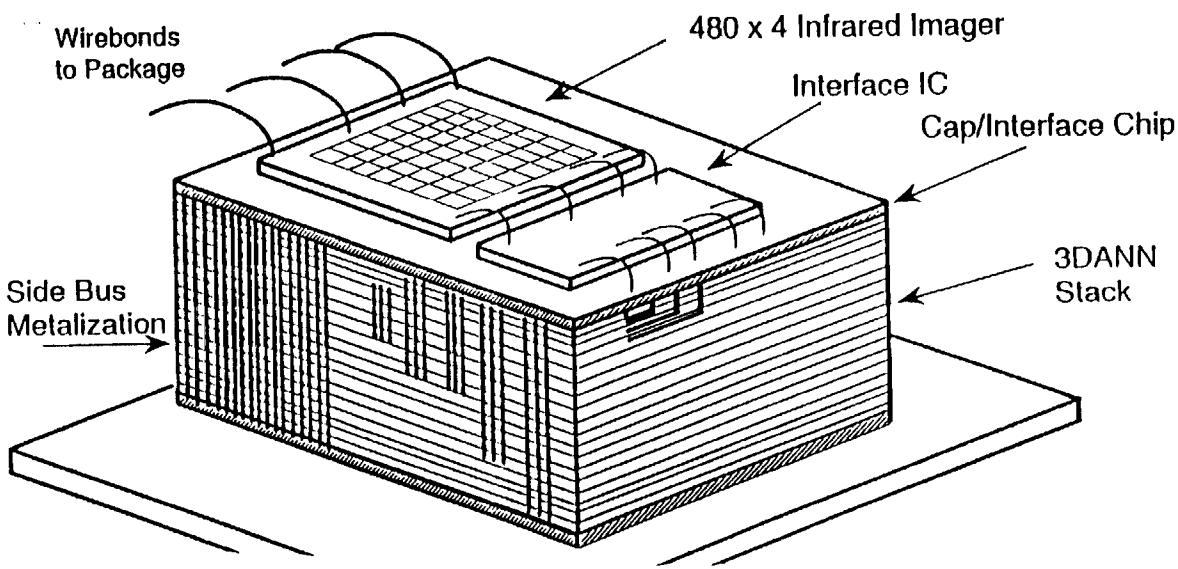
3D SRE

ARPA/Grumman

NASA JPL

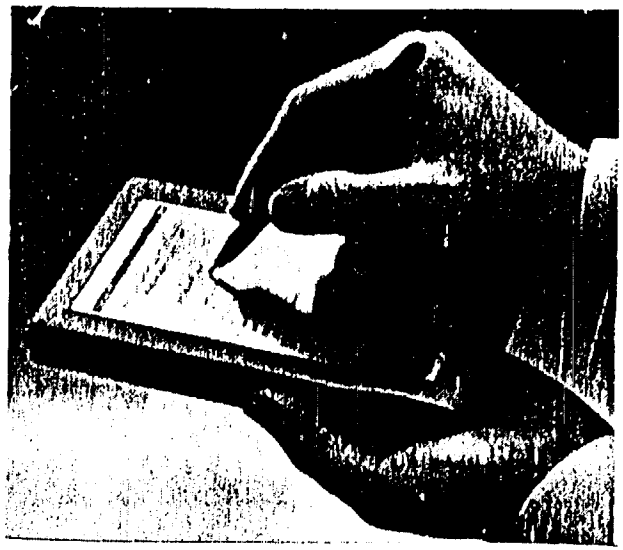
MUSTRS

ARPA/Martin Marietta



THE PDA CHARACTER RECOGNITION APPLICATION

- 3DANN IS AN OVERKILL
- THE ISSUE IS COST
- THE SOLUTION:
  - Miniaturize the IC's to 100x25 mil^2
  - $0.50 per IC
- THE COST $\simeq$ $2M TO PRODUCTION
- 100X100X50 MIL^3 PACKAGE FITS IN STANDARD SOJ PACKAGE



- AS ANYONE WHO OWNS A PDA KNOWS, THIS UNIT WILL OPEN UP A VCR SIZE MARKET
- ONE BILLION DOLLAR SALES IN FIVE YEARS

HIDDEN MARKOV MODELS AND NEURAL NETWORKS FOR FAULT DETECTION IN DYNAMIC SYSTEMS

Padhraic Smyth
Jet Propulsion Laboratory, MS 238-420
California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA 91109

pjs@galway.jpl.nasa.gov

JPL Neural Network Workshop, May 11-13, 1994

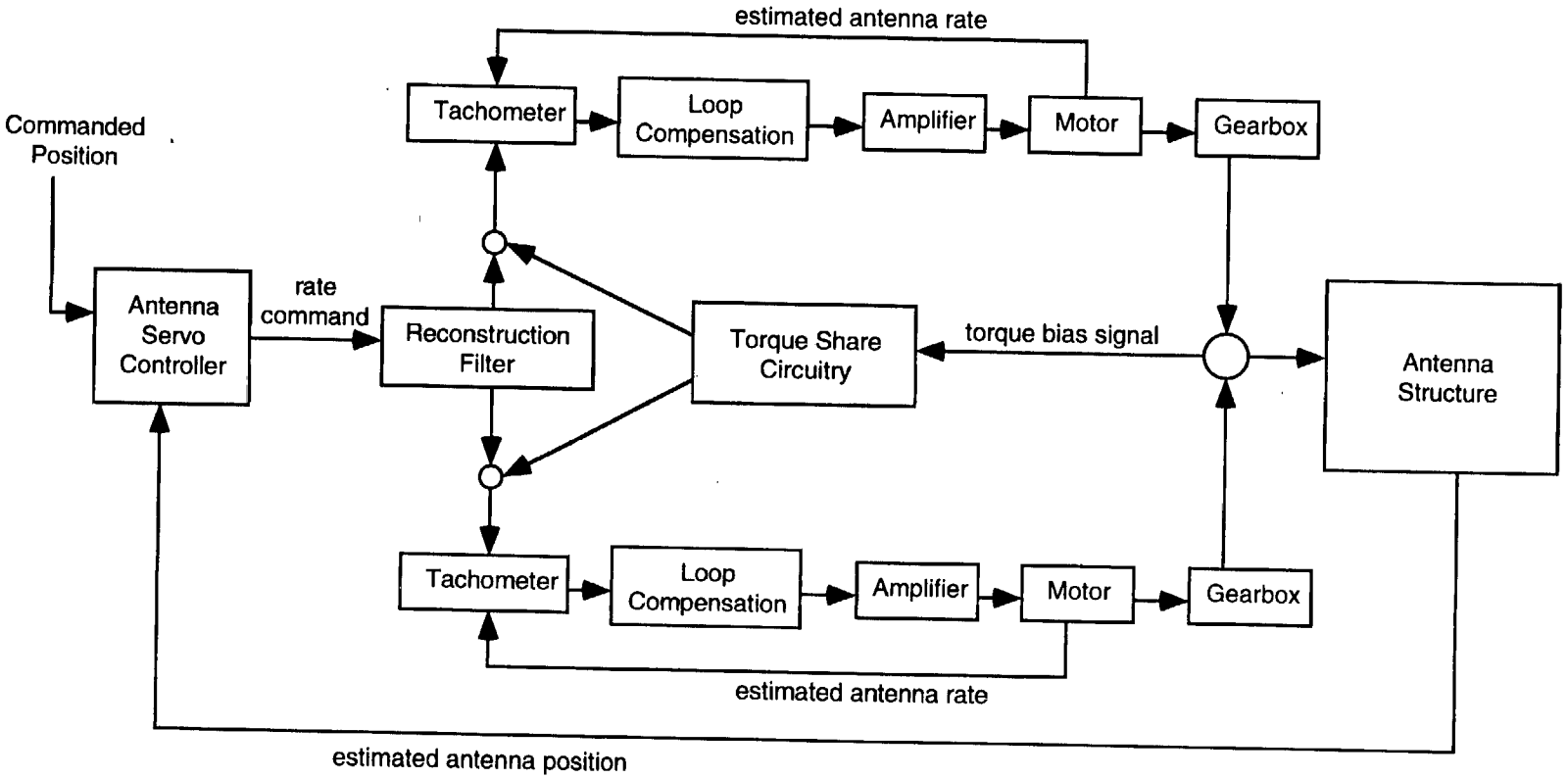
OVERVIEW

- **Neural Network + Hidden Markov models (HMMs):**
  - networks for discrimination and probability estimation
  - embedding networks in HMM's
  - application to fault detection (different from speech)

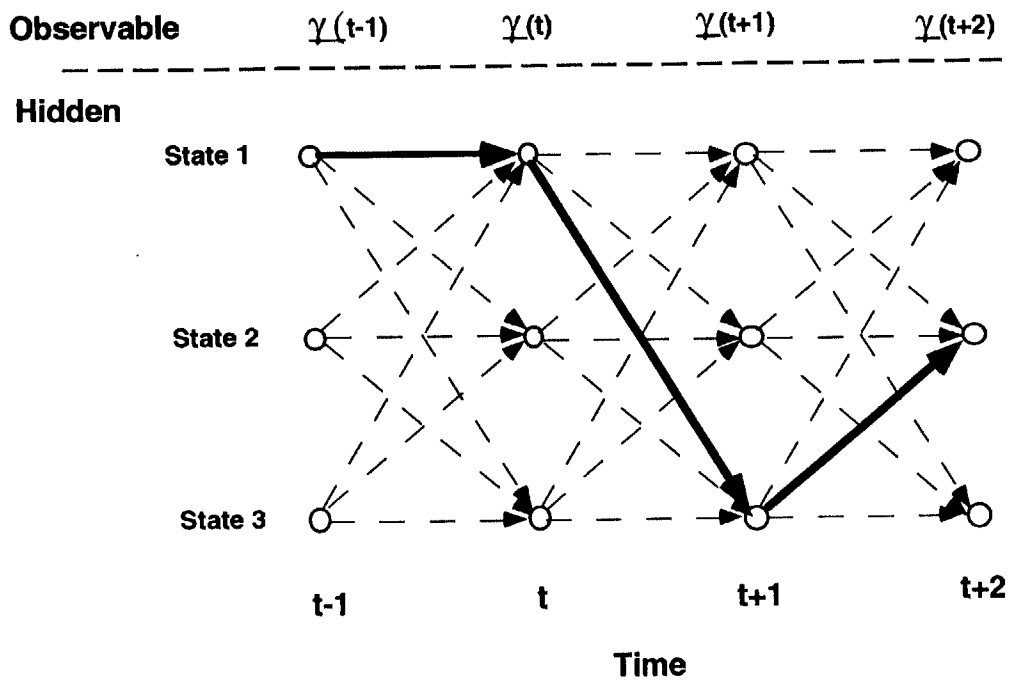
- **Application to Deep Space Network (DSN) Antenna Monitoring:**
  - on-line fault detection in large 34 meter ground antenna
  - discriminative vs. generative models for novelty detection
  - experimental evaluation

- **Conclusions and Application Status**

34 meter Beam Waveguide Antenna Pointing System



HIDDEN MARKOV MODEL BASICS



- **Explicit Assumptions (first order):**

  1. Present state only depends on previous state.

  2. Observables are independent over time *given* the states.

BASIC HIDDEN MARKOV EQUATIONS

Let $\Phi_t = \{\theta_t, \theta_{t-1}, \ldots, \theta_0\}$.

and $\Gamma_{t-k} = \{\theta_t, \theta_{t-1}, \ldots, \theta_{t-k+1}\}$.

- **Probability of Current State given Past Observed Data:**

$$p(\omega_j^t|\Phi_t) = \frac{1}{C_t} p(\theta_t|\omega_j^t) \sum_{i=1}^{m} a_{ij} p(\omega_i^{t-1}|\Phi_{t-1})$$

where

$$C_t = \sum_{j=1}^{m} \left[p(\theta_t|\omega_j^t) \sum_{i=1}^{m} a_{ij} p(\omega_i^{t-1}|\Phi_{t-1}) \right]$$

- **Probability of Past State given Observed Data to Present**

$$p(\omega_j^{t-k}|\Phi_t) = \frac{p(\omega_j^{t-k}|\Phi_{t-k}) p(\omega_j^{t-k}|\Gamma_{t-k})}{\sum_{i=1}^{m} p(\omega_i^{t-k}|\Phi_{t-k}) p(\omega_i^{t-k}|\Gamma_{t-k})}$$

NEURAL NETWORKS FOR PROBABILITY ESTIMATION

- **Theoretical Results:**
  - Theory shows that networks can approximate $p(\omega_i|\text{input features})$
  - Must use appropriate loss function: mean squared error or cross entropy
  - Results are asymptotic, assume global minimum in weight space.

- **Links with Conventional Statistics**
  - Feedforward networks can be considered a generalization of logistic regression: logistic nature of output is appropriate form for approximating posterior probabilities from exponential families.

- **Practical Consequences**
  - Practical results suggest that networks do a decent job of probability estimation.
  - Networks are better at probability estimation than competing nonparametric models (e.g., near-neighbor, decision tree methods).

HYBRID HMM/NEURAL NETWORK MODELS

- **Duality of Observed data term:**
  - Update equations are valid when terms are scaled by constants
  - By Bayes' rule can write:

$$p(\omega_j^t|\Phi_t) = \frac{1}{K_t}\frac{p(\omega_j^t|\theta^t)}{p(\omega_j)}\sum_{i=1}^{m} a_{ij}p(\omega_i^{t-1}|\Phi_{t-1})$$

- **Estimation of $p(\omega_j^t|\theta^t)$ terms:**
  - $p(\omega_j^t|\theta_t)$ = posterior probability of class j given inputs θ_t.
  - Train a feedforward network with MSE or CE loss functions.
  - Simple 12 input, 8 hidden units, 4 output units (normal + 3 fault conditions) feedforward network trained using conjugate gradient descent.
  - Cross-validation indicated that network size was not important.

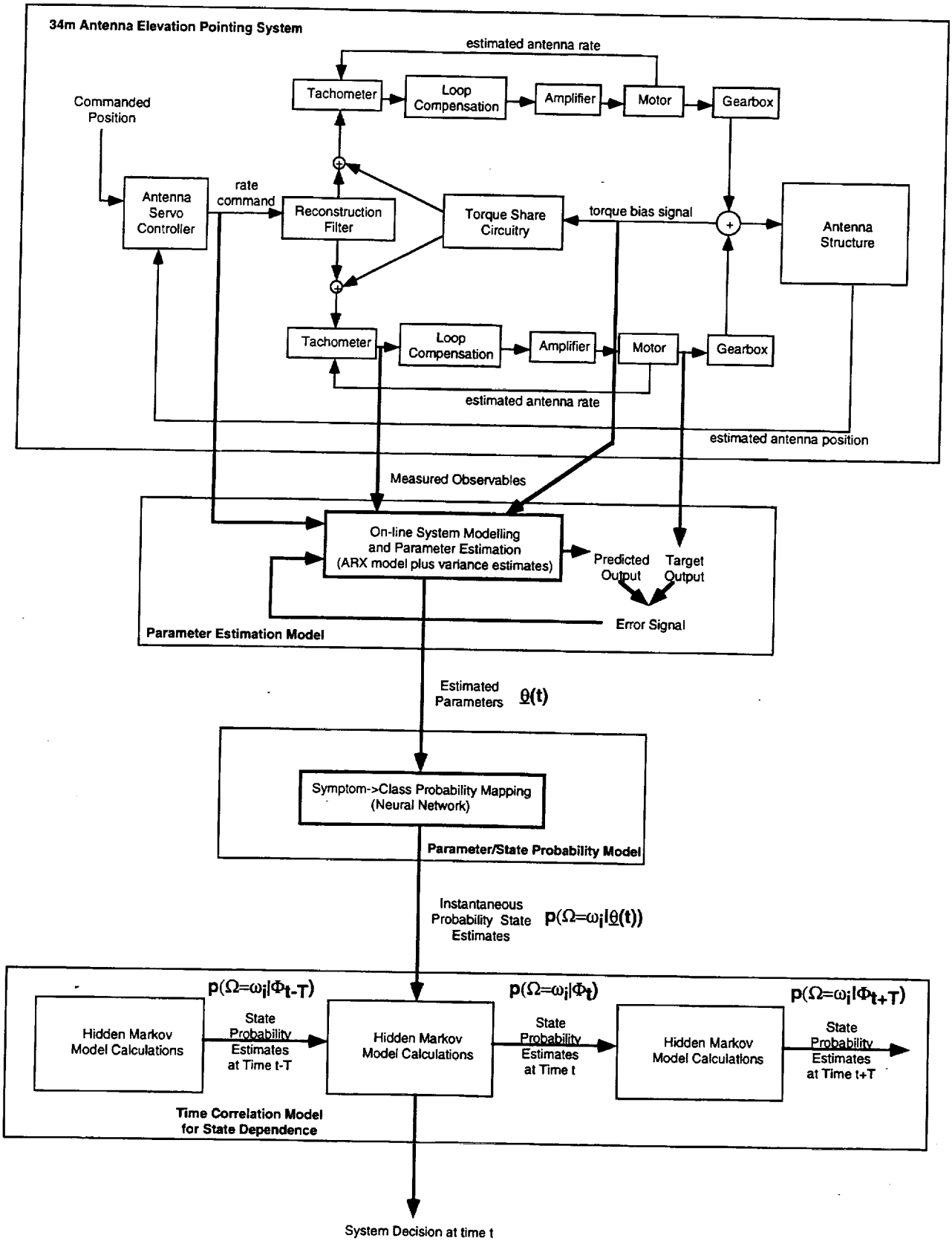
HYBRID HMM/NN FOR FAULT DETECTION

- **Key Ingredients:**
  - States are known a priori, correspond to distinct physical states of system, e.g., normal, fault conditions.
  - Observable-state conditional dependencies, $p(\omega_j^t|\theta_t)$ are learned by neural network from suitably generated training data.
  - HMM transition probabilities are a function of system MTBF and other long term characteristics:

$$a_{11} = 1 - \frac{\tau}{\text{MTBF}}$$

  - Only a single model is used: purpose is to infer "hidden" state sequence, i.e.,

$$\text{estimate } p(\omega_j^t|\Phi_t)$$


34m Antenna Elevation Pointing System
estimated antenna rate
Tachometer
Loop Compensation
Amplifier
Motor
Gearbox
Commanded Position
Antenna Servo Controller
rate command
Reconstruction Filter
Torque Share Circuitry
torque bias signal
Antenna Structure
Tachometer
Loop Compensation
Amplifier
Motor
Gearbox
estimated antenna rate
estimated antenna position
Measured Observables
On-line System Modelling and Parameter Estimation (ARX model plus variance estimates)
Predicted Output
Target Output
Error Signal
Parameter Estimation Model
Estimated Parameters θ(t)
Symptom->Class Probability Mapping (Neural Network)
Parameter/State Probability Model
Instantaneous Probability State Estimates p(Ω=ωi|θ(t))
p(Ω=ωi|Φt-T)
p(Ω=ωi|Φt)
p(Ω=ωi|Φt+T)
Hidden Markov Model Calculations
State Probability Estimates at Time t-T
Hidden Markov Model Calculations
State Probability Estimates at Time t
Hidden Markov Model Calculations
State Probability Estimates at Time t+T
Time Correlation Model for State Dependence
System Decision at time t

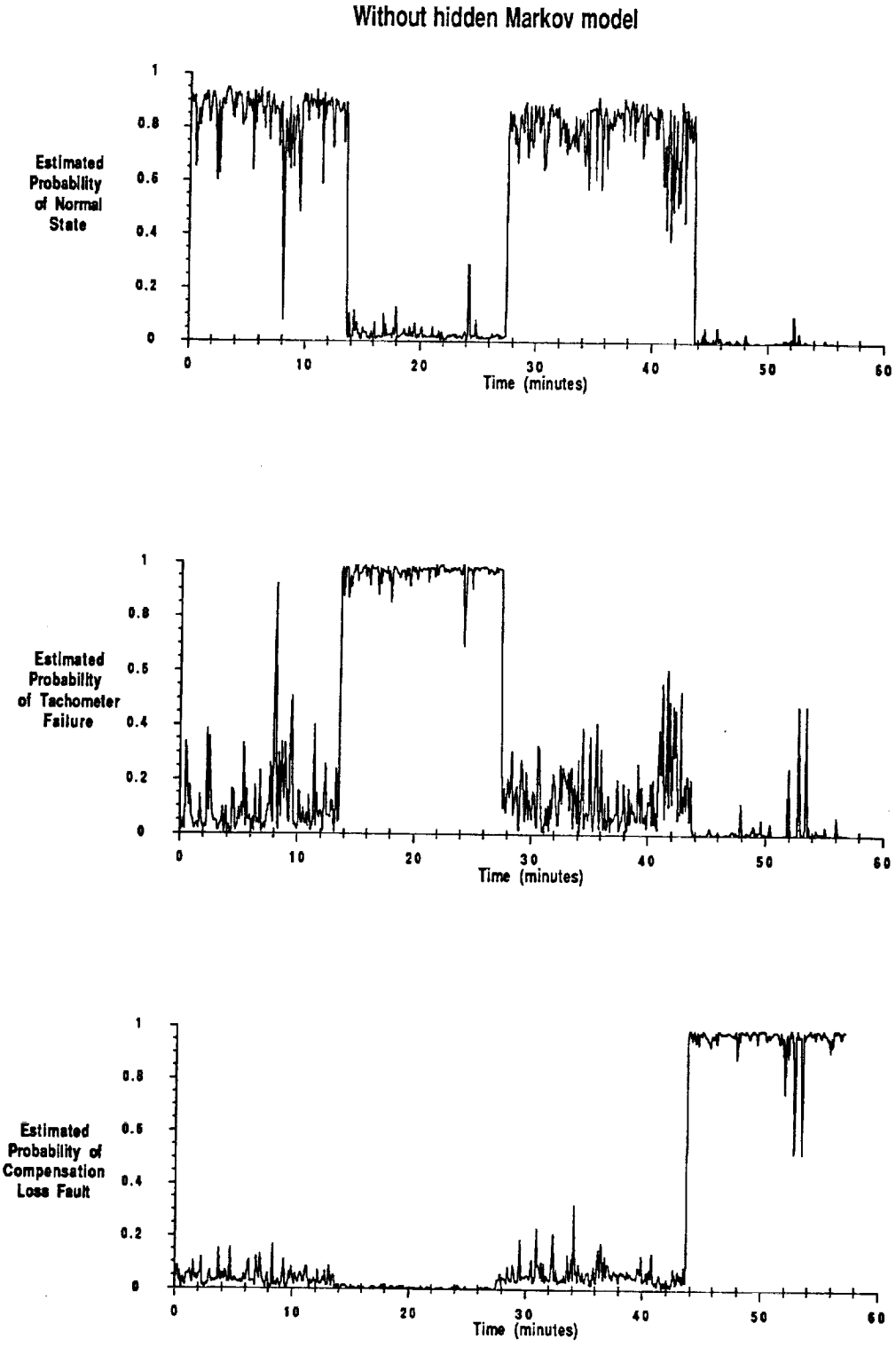
SUMMARY OF EXPERIMENTAL RESULTS OBTAINED AT DSS-13 34M ANTENNA IN REAL-TIME UNDER NORMAL AND FAULT CONDITIONS

Class	Without Markov model		With Markov model	
	Gaussian	Neural	Gaussian	Neural
Normal Conditions	0.36	1.72	0.36	0.00
Tachometer Failure	27.78	0.00	2.38	0.00
Compensation Loss	34.21	0.00	43.16	0.00
All Classes	16.92	0.84	14.42	0.00

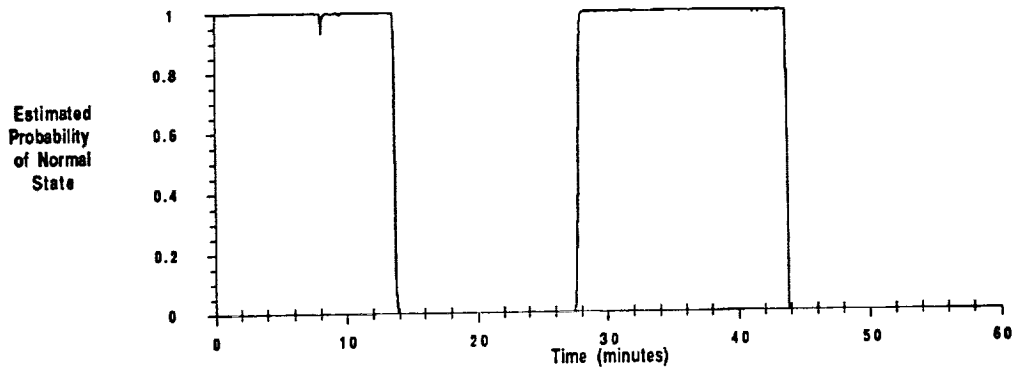
Percentage misclassification rates for Gaussian and neural models both with and without Markov component.

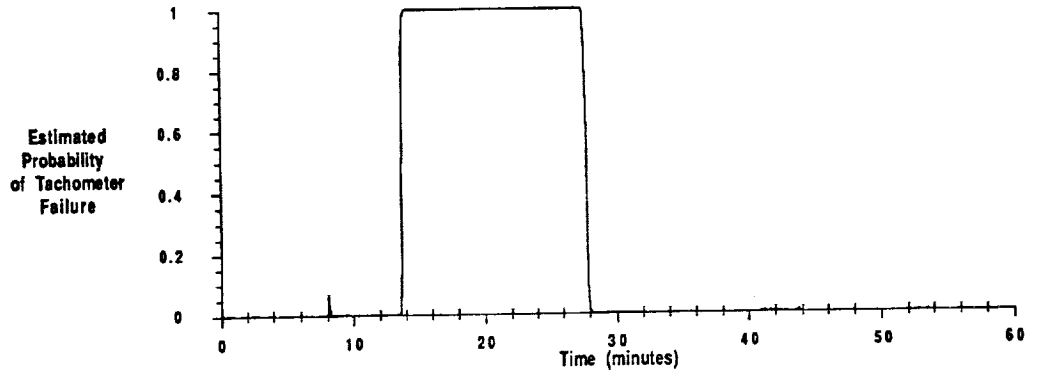
Class	Without Markov model		With Markov model	
	Gaussian	Neural	Gaussian	Neural
Normal Conditions	-2.44	-1.97	-2.46	-4.24
Tachometer Failure	-0.40	-3.52	-0.42	-4.22
Compensation Loss	-0.82	-3.48	-1.39	-4.71
All Classes	-0.87	-2.29	-1.02	-4.34

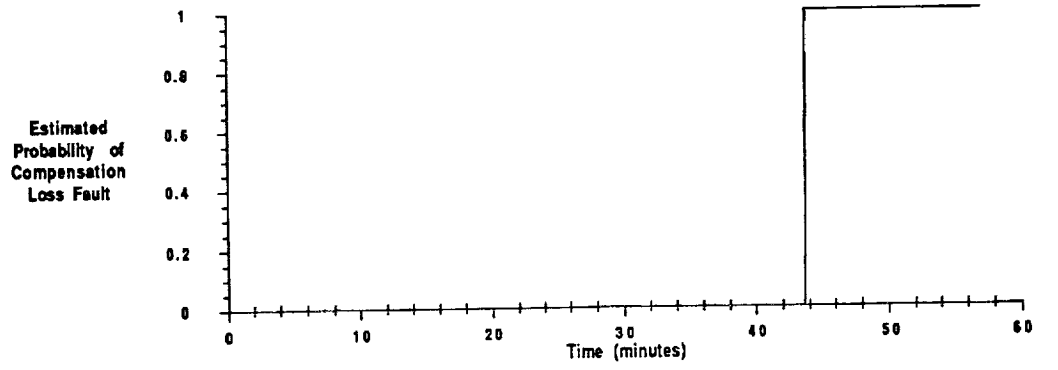
Logarithm of Mean Squared Error for Gaussian and neural models both with and without Markov component (more negative is better).


Without hidden Markov model
Estimated Probability of Normal State
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50
60
Time (minutes)
Estimated Probability of Tachometer Failure
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50
60
Time (minutes)
Estimated Probability of Compensation Loss Fault
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50
60
Time (minutes)

With hidden Markov model







DETECTING NOVEL STATES

- **Basic Problem**
  - In fault detection, it is highly likely that the set of known faults is *not exhaustive.*

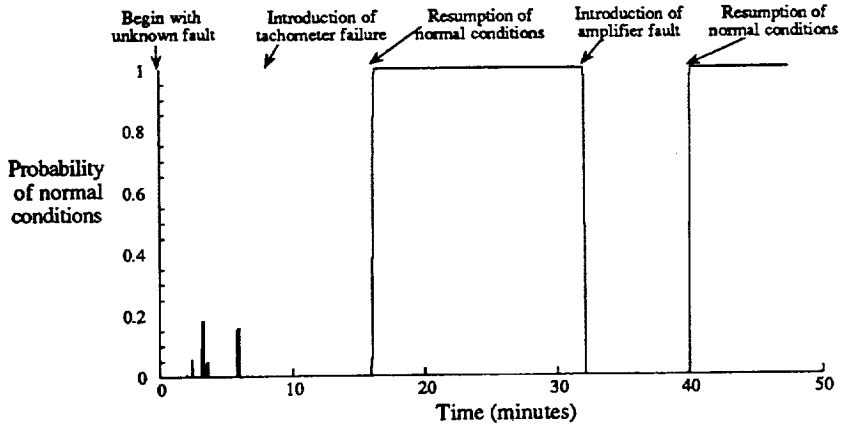
- **Solution**
  - Let ω_{m+1} be the "novel" state
  - Let $p_d(\omega_i|x, \omega_{1,\ldots,m})$ be the discriminative probabilities among the m known states
  - If we can define, $p(x|\omega_{1,\ldots,m})$, $p(x|\omega_{m+1})$, and $p(\omega_{m+1})$, then

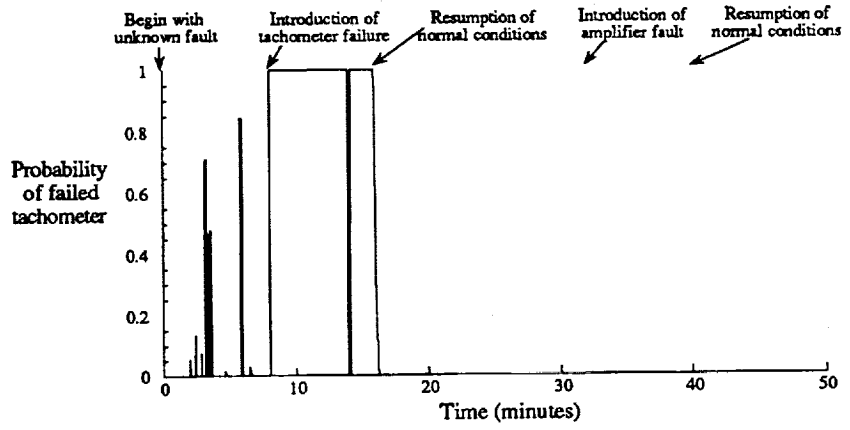
$$p(\omega_i|x) = p_d(\omega_i|x, \omega_{1,\ldots,m})p(\omega_{1,\ldots,m}|x)$$

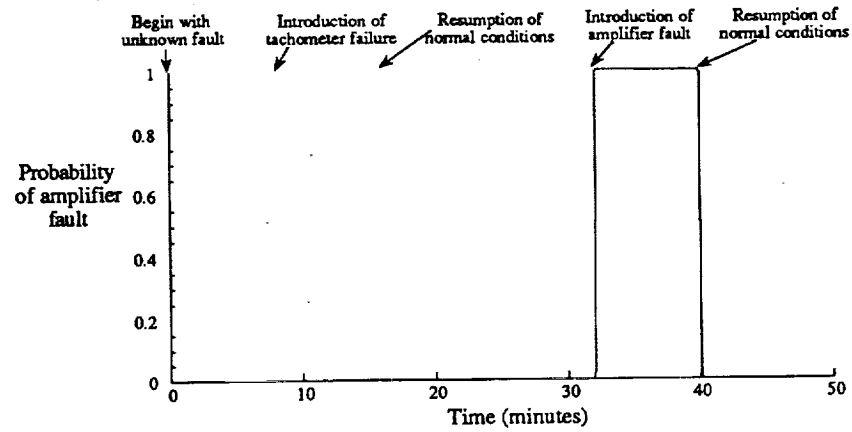
    and

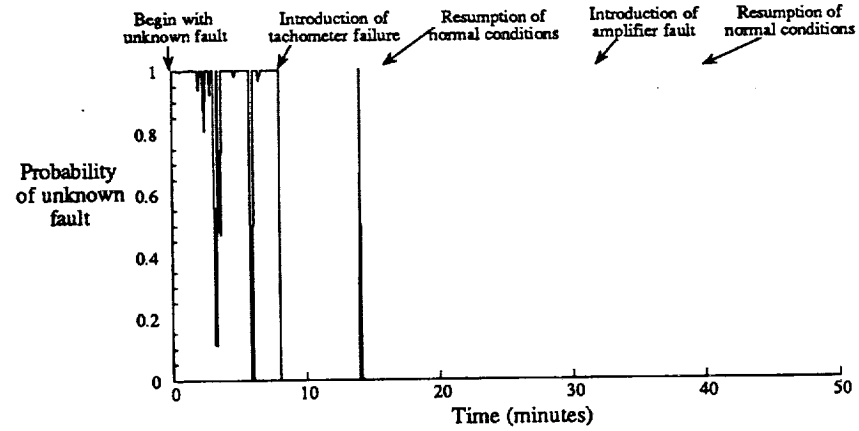
$$p(\omega_{m+1}|x) = 1 - p(\omega_{1,\ldots,m}|x)$$

  - $p(x|\omega_{m+1})$ is determined a priori, e.g., a non-informative prior density.


Begin with unknown fault
Introduction of tachometer failure
Resumption of normal conditions
Introduction of amplifier fault
Resumption of normal conditions
Probability of normal conditions
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50
Time (minutes)


Begin with unknown fault
Introduction of tachometer failure
Resumption of normal conditions
Introduction of amplifier fault
Resumption of normal conditions
Probability of failed tachometer
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50
Time (minutes)


Begin with unknown fault
Introduction of tachometer failure
Resumption of normal conditions
Introduction of amplifier fault
Resumption of normal conditions
Probability of amplifier fault
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50
Time (minutes)


Begin with unknown fault
Introduction of tachometer failure
Resumption of normal conditions
Introduction of amplifier fault
Resumption of normal conditions
Probability of unknown fault
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50
Time (minutes)

CONCLUSION

- **Summary**
  - Neural networks plus HMMs can provide excellent detection and false alarm rate performance in fault detection applications
  - Modified models allow for novelty detection

- **Key Contribution of Neural Network Model:**
  - Excellent non-parametric discrimination capability
  - A good estimator of posterior state probabilities, even in high-dimensions, thus, can be embedded within overall probabilistic model (HMM).
  - Simple to implement compared to other non-parametric models.

- **Application Status:**
  - NN/HMM monitoring model is currently being integrated with the new DSN antenna controller software: will be on-line monitoring a new DSN 34m antenna (DSS-24) by July.

Innovation and Application of ANN in Europe demonstrated by Kohonen Maps

Karl Goser
University of Dortmund
Faculty of Electrical Engineering
D 44221 Dortmund
Fax: x 49 231 755 4450
email: goser@luzi.e-technik.uni-dortmund.de

Extended Summary

One of the most important contributions to neural networks comes from Kohonen, Helsinki/Espoo, Finland, who had the idea of self-organizing maps in 1981. He verified his idea by an algorithm of which many applications make use up to now. The impetus for this idea came from biology, a field where the Europeans have always been very active at several research laboratories. The challenge was to model the self-organization found in the brain. Today one goal is the development of more sophisticated neurons which model the biological neurons more exactly. They should come to a better performance of neural nets with only few complex neurons instead of many simple ones.

A lot of application concepts arised from this idea: Kohonen himself applied it to speech recognition together with a japanese company, but the project did not overcome much more than the recognition of the numerals one to ten at that time. In the last years he is generating artificial music via self-organizing maps. A more promising application for self-organizing maps is process control and process monitoring. In this field Goser, Dortmund, made several proposals which concern parameter classification of semiconductor technologies, design of integrated circuits, and control of chemical processes. His graduates as Speckmann at Tuebingen broadened the field of applications. Ritter applied self-organizing maps to robotics. Germond, MANTRA center at Lausanne, introduced the neural concept into electric power systems.

At Dortmund we are working on a system which has to monitor the quality and the reliability of gears and electrical motors in equipments installed in coal mines. The results are promising and the probability to apply the system in the field is very high. A special feature of the system is that linguistic rules which are embedded in a fuzzy controller analyze the data of the self-organizing map in regard to life expectation of the gears. It seems that the fuzzy technique will introduce the technology of neural networks in a tandem mode. These technologies together with the genetic algorithms start to form the attractive field of computational intelligence. - Von Seelen, Bochum, develops a system with self-organizing maps that can monitor breaks and plugs in cars on this basis, too. Rueckert, Hamburg, and Ultsch, Marburg, try to combine the self-organizing map with an expert system instead of a fuzzy network, so that the total system exploits the advantages of both implicit and explicit rules.

Several research teams try to improve the theory of self-organizing maps, e.g. Cotrell, Paris, published important facts about the consistency of self-organisation, Tryba and Kanstein, Dortmund, are developing a new algorithm which bases on differential equations. Herault and Demartines, Grenoble, developed the vector quantization from the self-organizing concept. The vector quantization shows impressive results at the prediction of catastrophic failures. They also invented the interesting concept of separation of sources by simple neural networks which may find applications in hearing aids and noisy machineries.

A further effort alms to an implementation in hardware: Ramacher at Siemens, Munich, presented the Neural Computer Synapse which has a high flexibility and a remarkable high performance in regard to 10^8 CUPS (Connection Updates Per Second). Siemens AG is introducing Synapse I into the market now. - There are some activities about neural ASICs: Rueping, Dortmund, is representing the interesting concept BISOM in digital technique at which a simplified and adapted algorithm reduces the number of required transistors. Vittoz, Neuchatel, worked out an analog circuit for self-organizing maps which can be used in mobile and portable systems. Del Corso, Turino and Murray, Edinburgh, show that the pulse modulation techniques have decisive advantages for integration in analog technique.

The work on selforganizing maps is supported by national governments and by the European Union, as in the ESPRIT project NERVES, PYGMALION, GALATHEA, ANNIE, NEUFODI, CONNY, ELENA-NERVES II etc. The support includes small companies, too, most of which are in High-Tec centers from which a penetration of the new technology into the established industries should occur.

At the moment there are a lot of conferences in Europe in this field: ICANN, NeuroNimes, MicroNeuro, IWANN, ESANN, and several local workshops. Some conferences are strongly bound to roman and other to anglo-saxon regions. The high number of conferences does absolutely not relate to the number of industrial applications which are quite poor up to now. One reason for so many conferences comes from the role of universities which is far from industrial challenge: the promotion at universities needs papers which can be produced in the most easiest way on an innovative field and on conferences which need participants.

In conclusion we have to say that the industrial situation on the field of artificial neural networks is poor and difficult in Europe. One reason is that there are no or only little activities in the field of classical data processing in Europe. The strategy of many politicians is, however, that Europe gains a better position in a new technology as neuroinformatics, since in classical fields there are barely no chances for newcomers. There are a lot of soft applications of neural nets especially developed at application oriented laboratories as FhG (Seitzer, Hosticka), SICAN (Weinert) and IMS (Hoefflinger) in Germany. At the moment they concentrate their work on the electronic eye and on automotive applications. The academic work far from real economic pressure is overwhelming. We can only hope that the gap between academic and industrial world in Europe will diminish in future and the activity will grow on the industrial side, especially for our own interest to become more successfully in the important economical sector of information technologies.

References

T. Kohonen: Self-Organization and Associative Memory, 2nd Ed., Berlin, Springer-Verlag, 1988.

C.Jutten and J.Herault: Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture, Signal Processing 24(1991),pp.1-10.

P.Demartines and J.Herault: Representation of nonlinear Data Structures through a fast VQP Neural Network, Digest of NeuroNimes'93, pp.411-424.

S. Rüping, U. Rückert, K.Goser, "Hardware Design for Selforganizing Feature Maps with Binary Input Vectors", Int. Workshop on Artificial Neural Networks IWANN'93, Sitges (Barcelona), Juni 1993, pp.243-249.

V. Tryba and K. Goser, "Self-Organizing Feature Maps for Process Control in Chemistry" Proceedings of ICANN-91, June 24 - 28, 1991, Helsinki, pp.847-852.

V. Tryba, K. Goser, "A modified Algorithm for Self-Organizing Maps based on the Schrödinger Equation", Proc. IWANN'91, 17.- 19.9.1991, Granada, pp. 33-47.

A. Kanstein and K. Goser, "Self-Organizing Maps based on Differential Equations", Proceedings of ESANN'94, Brussels, Apr. 20 - 22, 1994.

G.Palm, K.Goser, U.Rückert, A.Ultsch, "Knowledge Processing in Neural Architecture", Oxford, September 1992.

K.Goser, "Kohonen Maps - Their Application and Implementation in Microelectronics", Digest of ICANN'91, June 24 - 28, 1991, Helsinki, pp.I-703-708.

U.Ramacher, W.Raab, J.Anlauf, J.Beichter, U.Hachmann, N.Brüls, M.Weßeling, E.Sicheneder, R.Männer,, J.Gläß and A.Wurz, "Multiprocessor and Memory Architecture of the Neurocomputer SYNAPSE I " , Digest of ICANN'93, Amsterdam, pp. 1034-1039.

D.del Corso, "Hardware Implementations of Artificial Neural Networks", Digest of IWANN'93 in Sitges,Spain, pp. 405-419.

P. Corsi, "What is ESPRIT doing in the Neural Network Field?" Digest of NeuroNimes'92, pp.671-682.

A.F.Murray, D.Del Corso, L.Tarassenko, "Pulse Stream VLSI Neural Networks mixing Analog and Digital Techniques", IEEE Trans.onNeural Networks, 2(1991), pp. 193-204.

H.Ritter, T.Martinetz, K.Schulten, "Neural Computation and Self-organizing Maps", Addison-Wesley, Reading, 1992.

J.D.Nicoud, "The MANTRA Center for Neuro-Mimetic Systems", Digest of Euro-Arch'93, München, Springer Verlag, pp.136-142.

NEURAL NETWORK CLASSIFICATION OF CLINICAL NEUROPHYSIOLOGICAL DATA FOR ACUTE CARE MONITORING

JOSEPH SGRO

Alacron, Inc., 71 Spit Brook Rd., Nashua, NH 03060 and The Neurological Institute of New York, 710 West 168 Street, New York, NY 10032

INTRODUCTION

The purpose of neurophysiological monitoring of the "acute care" patient is to allow the accurate recognition of changing or deteriorating neurological function as close to the moment of occurrence as possible, thus permitting immediate intervention.

EEG MONITORING

The electroencephalogram is a sensitive indicator of cerebral ischemia. Slowing of the EEG in man occurs when regional cerebral blood flow drops to 16-22 ml/100g/min., and severe voltage attenuation results if flow is further reduced to 11-19 ml/100g/min. (Trojaborg & Boysen 1973). This observation has lead to the use of EEG monitoring in clinical settings in which cerebral perfusion is at risk. The utility of EEG monitoring during carotid endarterectomy has been demonstrated (Chiappa and Burke, 1979; Myers et al, 1980), and it is routinely used in some major centers to determine the necessity of shunting. During cardiopulmonary bypass for cardiac surgery, the EEG also has been shown to be a sensitive indicator of the effects of hypotension as well as air embolism (Prior, 1979; Stockard et al, 1964). The Practice Committee of the American Academy of Neurology has advised that "EEG monitoring during complex surgical procedures has become an established procedure to safeguard cerebral perfusion" (Pedley and Emerson, 1984).

Recently, a number of EEG monitoring system have been proposed. These are either primarily displays of data reduced EEG, processed by FFTs (Fast Fourier Transforms) or AR (Autoregressive), or heuristic rule based detectors for specific patterns derived from processed or raw EEG.
In our view, the limitations of automated EEG analysis systems heretofore developed are consequences of either the use of data reduction, which obscures morphological characteristics of EEG waveforms critical for their identification, or the reliance on rule based systems which are limited by their design to detect a limited repertoire of EEG patterns and may have excessive false classification rates.

For an EEG monitoring machine to be clinically acceptable for use in ICU or operating room environments, the following four requirement should be satisfied:

1. It must detect artifacts to avoid false interpretation of EEG waveforms.
2. It must be able to identify unambiguously designated patterns and changes in patterns in the EEG.
3. It must have provision for multiple monitoring channels.
4. It must be able to perform these functions in real-time.

EVOKED POTENTIAL MONITORING

Evoked potentials (EPs) are electrophysiologic markers of transmission of sensory signals through afferent neural pathways in the central nervous system following auditory, visual, and somatosensory stimulation. They are widely used in clinical neurology for detection and localization of neural lesions (Chiappa, 1990). Brainstem auditory evoked potentials (BAEPs) and somatosensory

evoked potentials (SEPs) are relatively resistant to anesthetic agents and levels of patient arousal, and are therefore ideally suited to monitoring the integrity of the central nervous system of patients in "acute care" settings. The purpose of evoked potential monitoring of the "acute care" patient is to allow the accurate recognition of changing or deteriorating neurological function as close to the moment of occurrence as possible, thus permitting immediate intervention.

BAEPs are widely used to monitor acoustic nerve function during surgery in the cerebellopontine angle (CPA), primarily for resection of acoustic neuromas and other CPA tumors, where the surgery threatens auditory nerve function. They are sensitive to mechanical disruption of the auditory nerve, as well as cochlear and eighth nerve ischemia. Intraoperative BAEP monitoring has been recently demonstrated to be associated with significantly decreased postoperative morbidity (Radtke and Erwin, 1988). BAEPs are also sensitive to disruption of and ischemic insult to structures within the brainstem auditory pathways, and hence are employed during other procedures that risk brainstem injury, including surgery for basilar artery aneurysms, posterior fossa arterio-venous malformations, and intrinsic brainstem tumors (Friedman and Grundy, 1987; Radtke and Erwin, 1988; Abramson et. al. 1985).
SEPs are sensitive to parenchymal damage directly involving the posterior columns, as well as compression, mechanical distraction, and cord ischemia. SEP monitoring during scoliosis surgery has become widely accepted, and has virtually replaced the "wake-up" test. SEP monitoring is also employed to monitor the integrity of the spinal cord during cross clamping of the aorta, and neurosurgical procedures involving the spinal cord and its blood supply (Friedman and Grundy, 1987; Loughnan and Hall, 1989; Emerson and Pedley, 1988). Additionally, cortical components of the SEP can be used to assess integrity of the cerebral cortex during procedures requiring temporary occlusion of cerebral arteries (Buchtal and Belopavlovic, 1988).

In order to achieve widespread use and utility, an automated EP monitoring system should have:

1. The ability to detect artifacts to avoid false interpretation of EP waveforms.
2. The ability to unambiguously identify designated EP waveforms.
3. The ability to measure the amplitudes and latencies of designated EP waveforms.
4. The capability of monitoring multiple EP channels in real time.

The Table below lists the major techniques that have been used for automated EP analysis. To date, none of these is in widespread use. This reflects, in large part, their collective sensitivity to artifacts and noise and their inconsistent ability to correctly track the waveform of interest , its amplitude, or latency.

Methods	Disadvantages	Reference
Discriminant methods	Requires a priori definition of features	Clarson Liang (1989)
Template methods	Requires a priori template definition	Childers et al (1987)
Derivative methods	Extremely noise sensitive	Miskiel and Ozdamar (1987)
Rule based methods	Very sensitive to morphology variations	Boston (1989)

NEURAL NETWORKS

INTRODUCTION

PDP networks, also known as *neural* networks, have recently attracted widespread interest and application in diverse areas of computerized pattern recognition, including handwriting, voice and visual pattern recognition systems (Levinson et. al, 1983; Devijer and Kittler, 1982; Blake and Zimmerman, 1987; Lang and Waibel, 1990; Rajavelu et. al., 1989; Buhmann et. al., 1989). Neural networks are structured as arrays of interconnected units which have the capability of "learning" by examples causing functional modification of interconnections. The units have functional properties modeled after neurons, and interconnections modeled after synapses.

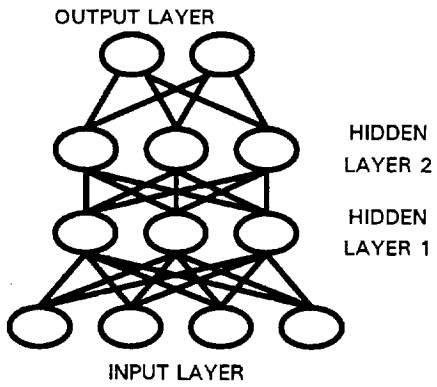
An important feature of neural networks is that it is not necessary to precisely describe the patterns to be recognized. Rather, the network is "trained" by presenting it with examples of patterns to be recognized. While an expert recognition system may be intuitive, or difficult to articulate, the training mechanism only requires examples of classified data (output patterns). In contrast to most other methods, the structure of neural networks allows training to take place in the absence of a specific heuristic method for each feature to be recognized.

The major advantage of neural networks is that they are able generalize, and adapt to distortion or noise without losing their robustness. Neural networks are capable of correctly identifying input patterns that are morphologically similar to but not identical to the patterns on which they were trained. The latter feature makes neural networks ideally suited to EEG and EP analysis which requires correct identification of selected neurally generated signals based upon waveform morphology, and often in the presence of considerable accompanying noise. Neural networks thus have the advantage of allowing an efficient unified system for detection and identification of artifacts, abnormalities, and, EP's waveform latency in the presence of noise. Our results below demonstrate the feasibility of the use of neural networks for EEG/EP analysis.

IMPLEMENTATION

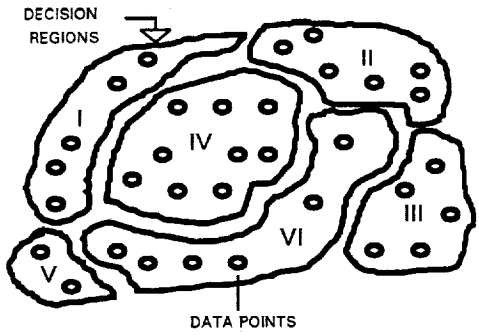
A. NETWORK ARCHITECTURE

We initially implemented a fully interconnected feed forward net with a selectable number of layers and nodes. We used three and four layer networks (i.e. one and two hidden layers) for both EEG and EP analysis. All data processing was performed on AT compatible computer with an Alacron AL860 coprocessor board. The AL860 board uses a 40 MHz Intel i860 RISC processor (80 MFLOPS) and provides 64 MB of memory.



The net initialization is achieved using fixed pseudo-random, unique pseudo-random, seeded pseudo-random or 0 values. The net size, the net structure, the convergence function, the transfer function, and the initialization mode are user selectable at initiation of training. We used nets ranging in size from 512 to 8192 input nodes, hidden layer sizes of between 5 and 500 nodes, and an output layer of less than 20 nodes. The transfer function used was the logistic sigmoid transfer function.

Additionally, we implemented for EP analysis a probabilistic neural network (PNN) as described by Specht (1990) (Figure below), a reduced coulomb energy (RCE) neural network, closely related to PNNs, and a discriminant pattern recognizer (Bow, 1984) .



B. NETWORK TRAINING PARADIGM

Training was achieved using back propagation via modified steepest descent (Rumelhardt, 1987). This entails multiplication of the input values by the interconnection weights, calculation of each layer's output, and propagation of the outputs forward through each successive layer of the network with the calculation of the mean squared error between the output and the desired output. At the end of each training cycle, which consists of a complete presentation of all patterns in the training set, the total calculated error was propagated backwards and the adjustment of the individual weights was made, as outlined in Rumelhardt, 1987. Usually, we obtained an initial pattern match within approximately 50 training cycles using several hundred test patterns, with full convergence taking up to hundred cycles. The network ran entirely in RAM memory on the I860, with an optimized assembly language floating point dot product requiring approximately 10 to 30 minutes per training cycle.

C. NETWORK TESTING PARADIGM

For testing, input data is presented to the network without weight adjustment. The calculated output of the neural network was compared to the expert classification to determine if the classification was successful. Results were then tabulated, and the classification percent correct was calculated.
Separate methods of validation were used for large (>100 epochs) and small (<100 epochs) data sets. For large data sets, the set is split into two subsets - one for training and the other for testing. For small data sets the "holdout" method is employed. A single epoch is held out, and the network is trained on the remaining epochs. The withheld epoch is tested against the trained network. This process is repeated for all epochs in the data set (Specht, 1990; Marchette and Priebe, 1987; Maloney, 1988).

EEG NEURAL NETWORKS

Neural network classification of EEG was investigated using data reduced input via the FFT or an AR model and also raw EEG data.

A. FFT

Input data was decimated to 512 points per channel per 10 second epoch. These data were converted to 512 point power spectra. This is accomplished by applying a standard FFT and taking the squared magnitude of the coefficients. The spectra were then used as input to the neural networks.

B. AR

Input data was initially modeled by a modified covariance ARMA autoregressive moving average model, a Burg model, and a Prony model. The ARMA model was used for classification of EEG because we observed that it consistently produced the most stable and accurate spectra. The ARMA model of EEG consisted of two real coefficients and one hundred complex coefficients. This exceeds the number of coefficients customarily employed to describe EEG spectra (Jansen, 1985). These coefficients were used to compute a 512 point power spectrum. The spectra were the used as inputs to the neural networks.

C. RAW EEG DATA.

A limitation of the use of raw EEG for neural network input is that the data is scale and translation dependent, but EEG interpretation is largely translation and scale independent. Our initial solution to this problem was to train the neural network on rotated and scaled versions of each training epoch. This approach, however, would have resulted in a prohibitive increase in the required number of training epochs. For example, in investigations described below, we used typically 150 training epochs. Each epoch would be transformed into 2560 translated and scaled versions, resulting in a total of 384,000 training epochs [256 translations and 10 amplitude scale levels]. Training the neural network with this number of epochs would not have been practical.

We investigated structural modifications to the neural network to make it immune to translation and amplitude variations in the training set. We implemented a modification of the method of Goggin et al (1991) which preprocesses the epoch into a form that is not effected by translation and amplitude variations. Each epoch contains typically 16 channels, each of which is a time series of 512 data points. Each channel is transformed into a translation and scale invariant form as shown in equation 1, below:

$$Y_i = \frac{\sum_{k=0}^{N-1} X_k \cdot X_{MOD(k+i,N)}}{\sum_{k=0}^{N-1} (X_k)^2}$$

The transformed data is then processed by the back propagation neural network. Neural network employing polynomial transformed data have been named "higher order neural networks" (HONN).

EP NEURAL NETWORKS

In all cases, input to the recognition software consisted of raw 1024 point per channel (both replications). We implemented a fully interconnected feed forward net with a selectable number of nodes (Figure above). The neural network had four layers (i.e. two hidden layers).

The desired outputs were presented to the network as ones and zeros to indicate normal, abnormal, or uninterpretable. Latency and amplitude data were encoded as eight bit binary values. An output of the network was assigned to each bit of the binary value. BAEP and SEP latencies where encoded after multiplying by 10, or 0.1 msec per unit. Amplitude data was encoded as eight bit binary values, 0.1 microvolts per unit.

The interconnection weights of the net were initialized to small random values using a random number generator. We used nets ranging in size from 1024 to 8192 input nodes, and an output layer of less than 100 nodes. First and second hidden layers contained 512 and 256 nodes respectively. The transfer function used was the logistic sigmoid transfer function.

Network training was achieved using back propagation via modified steepest descent as described above. Usually, we obtained an initial pattern match within approximately 50 training cycles using several hundred test patterns, with full convergence taking typically one hundred cycles. The network ran entirely in RAM memory on the I860, with an optimized assembly language floating point dot product requiring approximately 10 to 30 minutes per training cycle, or about 4 to 12 hours for full convergence.

For testing, input data is presented to the network without weight adjustment. The calculated output of the neural network was compared to the expert classification to determine if the classification was successful. Results were then tabulated, and the classification percent correct was calculated. For each data sets the "holdout" method described above was employed.

In addition to back propagation, we also implemented and evaluated RCE and PNN networks.

NEURAL NETWORK RESULTS

EEG CLASSIFICATION RESULTS

All results presented below were obtained using a four layer network (i.e. two hidden layers). We observed that when a sufficient number of nodes were present in the network, training required less than 100 passes over all the epochs in the training set In all cases the net converged and 100% correct identification of the training set was obtained prior to testing.

In all cases, EEG pattern classification using raw EEG was superior to that using FFT or AR input. Furthermore, the HONN outperformed the standard neural network, producing excellent results in all cases. Typical results obtained using the small data set paradigm are illustrated in Table 2, below. In the table, EF refers to eye flutter, IRS to intermittent rhythmic generalized slowing, SH to focal sharp waves, CPD to continuous polymorphic delta, M to muscle artifact and NL to normal. The network size designation in the Table is as follows: number of nodes in the input layer X number of hidden nodes in first hidden layer X number of hidden nodes in the second hidden layer X number of nodes in the output layer.

	EEG Test Patterns					
	EF vs. NL	RS vs. EF	RS vs. EF	SH vs. CPD	SP vs. NL	SP vs. M
Network	512x20x	512x20x	1024x20x	2048x20x	8192x50	8192x50
Size	10x2	10x2	10x2	10x2	x10x2	x10x2
Channels	1	1	2	4	16	16
Data Types	Percent	Correct	Classification			
FFT	57	50	55	52	60	62
AR	52	45	50	48	52	55
Raw EEG	82.5	75	85	75	80	75
HONN AR	75	70	65	60	75	76
HONN FFT	80	65	78	75	78	79
HONN Raw	95	90	95	90	95	95

The above results indicate that superior classification is obtained using raw EEG input when compared to either AR or FFT spectra. We speculate that the inferior performance of AR and FFT based methods is attributable to information loss inherent in these spectral representation of the EEG waveforms. Our results further indicate that use of multiple channels (IRS vs. EF comparisons) improves performance. The best performance, achieving level of EEG pattern recognition accuracy suitable for clinical applications, was obtained using the high order neural network (HONN) methods.

Performance of the our initial, non-translational invariant, network (STD) and the high order neural network (HONN) using raw EEG data was further evaluated using the large data set paradigm to test classification of states of arousal, abnormalities, and artifact identification. For state, 150 sixteen channel test epochs were used. The size of the network was 8192 x 200 x 50 x 3. Results are shown below .

State	% Correct Classification	
	STD	HONN
Wake	82	93
Stage I Sleep	86	97
Stage II Sleep	66	95

Again, using the large data set paradigm, 150 test epochs were classified as normal or demonstrating any of the following "abnormalities": continuous slowing (any type), intermittent slowing (any type), slow alpha, or uninterpretable. The network size was 8192 x 200 x 50 x 5. Results are shown in Table 4, below.

Category	% Correct Classification	
	STD	HONN
Normal	82	98
Intrm slowing	70	93
Cont slowing	70	97
Slow alpha	77	92
Uninterpretable	50	98

Finally, for detection of the presence and classification of types of artifacts, 150 sixteen channel test epochs were used. The size of the network was 8192 x 200 x 50 x 6. Results are shown below .

Artifact	% Correct Classification	
	STD	HONN
None	70	97
Eye Flutter	90	97
Eye Blinks	80	95
Horiz Eye Mnts	66	98
Muscle	73	98
Movements	68	98

The above results confirm the suitability of the HONN network for accurate identification of a wide variety of EEG waveform patterns.

EVOKED POTENTIAL CLASSIFICATION RESULTS

I. LATENCY MEASUREMENT RESULTS

The Table below depicts the latency measurement errors for wave I, III and V of the BAEP, as made by three different neural networks and a discriminant method. All neural network methods performed well, with errors close to human measurement error on BEAPs recordings, which is approximately 0.1 - 0.2 MS or 1-2% of the standard 10 msec sweep. The discriminant methods was not as successful. The most accurate classification was achieved by the back propagation method.

BAEP Latency Error Std Dev

Milliseconds	BP	RCE	PNN	Discr	# Cases
I	0.20	0.22	0.24	1.00	172
III	0.30	0.33	0.40	1.20	168
V	0.30	0.33	0.30	1.50	178

The Table below presents the classification results for median nerve SEP data. The latency measurement accuracy achieved by all neural network methods was excellent. The back propagation performed best. The latency measurement error of the BP network was similar to human measurement errors, which is approximately 0.5 MS, or 1% of the standard 50 msec sweep. Again the discriminant method performed poorly.

SEP Latency Error Std Dev

Milliseconds	BP	RCE	PNN	Discr	# Cases
N9	0.30	0.33	0.45	1.10	221
P14	0.70	0.77	1.05	2.10	218
N20	0.30	0.33	0.45	4.20	213

Similarly, the Table below illustrates classifications for VEPs. Classification accuracy was excellent for all neural network techniques, the best performance being achieved by the back propagation method. The 1 msec error for BP is 0.5% of the standard 200 msec sweep. The discriminant method performed poorly.

VEP Latency Error Std Dev

Milliseconds	BP	RCE	PNN	Discr	# Cases
P100	1.00	1.10	1.50	5.10	270

II. AMPLITUDE MEASUREMENT RESULTS

The Table below presents our amplitude measurement results using BAEP data. Accurate amplitude measurement were made by all neural network methods tested. The best performance was achieved by the back propagation network and the discriminant method performed poorly.

BAEP Amplitude Error Std Dev

micro	BP	RCE	PNN	Discr	# Cases
V	0.08	0.48	0.62	1.01	101

Similarly, the Table below presents our amplitude measurement results for SEP data.

SEP Amplitude Error

micro	BP	RCE	PNN	Discr	# Cases
N9	0.32	0.38	0.47	0.71	105
P14	0.15	0.72	0.75	1.17	105
N20	0.23	0.51	0.50	0.71	105

Our amplitude measurement results are presented in the Table. Again, the back propagation method provides the most accurate amplitude measurement.

VEP Amplitude Error Std Dev

micro	BP	RCE	PNN	Discr	# Cases
P100	1.20	1.23	1.32	2.34	270

III. CLASSIFICATION RESULTS

The Tables below present the accuracy by which the three neural network and the discriminate method classified EP recording of the three modalities and "Normal", "Abnormal" or "Uninterpretable". The best performance was achieved by the back propagation method, which classified 94% of EP studies in agreement with the "expert" reader. Additionally, ninety percent of records that were uninterpretable due to noise contamination were correctly identified.

BAEP

% Correct	BP	RCE	PNN	Discr	# Cases
Result					
Normal	95%	91%	82%	56%	96
Abnormal	92%	87%	80%	54%	91
Uninterpr	90%	80%	80%	60%	10
Overall	93%	89%	81%	55%	197

SEP

% Correct	BP	RCE	PNN	Discr	# Cases
Result					
Normal	97%	89%	84%	64%	155
Abnormal	93%	86%	82%	61%	30
Uninterpr	90%	83%	77%	60%	44
Overall	95%	87%	82%	63%	229

VEP

% Correct	BP	RCE	PNN	Discr	# Cases
Result					
Normal	97%	93%	91%	63%	166
Abnormal	91%	89%	87%	60%	45
Uninterpr	91%	87%	85%	59%	95
Overall	94%	91%	89%	61%	306

IV. MULTICHANNEL RESULTS

The above results were obtained by presenting the neural networks with multiple channels (3 for BAEPs, 4 for SEP, and 6 for VEP). The effect of multiple channels on the performance of neural network classification was examined by omitting channels which did not specifically contain a designated waveform of interest, but provided information which is used in human waveform recognition. Specifically, Ac-Cz and Ai-Ac channels for BAEPs, and SC5-Fpz for SEPs. In all cases, inclusion of these "extra" channels improved classification and measurement results slightly. In some cases, major improvements were linked to the use of extra channels. For examples, use of three channel resulted in a 24% improvement in wave III amplitude measurement.

BAEPs

% Correct	Number of channels		
	1	2	3
Result			
Norm	94%	95%	95%
Abnormal	90%	91%	92%
Uninterp	90%	90%	90%

BAEP Latency Error

msec	Number of channels		
	1	2	3
Wave			
I	0.23	0.21	0.20
III	0.53	0.42	0.40
V	0.32	0.33	0.30

BAEP Amplitude Error

u-Volts	Number of channels		
	1	2	3
Wave			
I	0.32	0.30	0.30
III	0.42	0.33	0.32
V	0.34	0.27	0.26

SEP Classification accuracy

% Correct	Number of channels	
	3	4
Result		
Norm	97%	97%
Abnormal	93%	93%
Uninterp	87%	90%

SEP Latency Error

msec	Number of channels	
	3	4
Wave		
N9	0.31	0.30
P14	0.89	0.75
N20	0.32	0.30

CONCLUSIONS

Our results confirm that:

1. Neural networks are able to accurately identifying EEG patterns and evoked potential wave components, and measuring evoked potential waveform latencies and amplitudes.

2. Neural networks are able to accurately detect EP and EEG recordings that have been contaminated by noise.

3. The best performance was attained consistently with the back propagation network for EP and the HONN for EEGs.

4. Neural network performed consistently better than other methods evaluated.

5. Neural network EEG and EP analyses are readily performed on multichannel data.

BIBLIOGRAPHY

Abramson, M., Stein, B.M., Pedley, T.A., Emerson, R.G. & Wazen, J. *Laryngoscope* 95, 1318-1322 (1985).

Blake, A. & Zisserman. *Visual Reconstruction* (MIT Press, Cambridge MA D 1987., 1987).

Bow, S.T. *Pattern Recognition* 1-323 (Marcel Dekker, New York, 1984).

Buchthal, A. & Belopavlovic, M. *Klin-Wochenschr* 66(suppl 14), 27-34 (1988).

Buhmann, J., Lange, J. & Von Der Malsburg, C. *IJCNN* 1, 155-159 (1989).

Chiappa, K.H. *Evoked Potentials in Clinical Medicine* 1-609-630 (Raven Press, 1990).

Chiappa KH, Burke SR, and Yourn RR. *Stroke*. 381-388 (1979).

Devijver, P.A. & Kittler, J. *Pattern Recognition: A Statistical Approach* (Prentice-Hall, London, 1982).

Emerson, R.G. & Pedley, T.A. *Am J EEG Technol* 28, 251-268 (1988).

Friedman, W.A. & Grundy, B.L. *J Clin Monit* 3, 38-44 (1987).

Goggin SDD, Johnson KM and Gustafson KE. *Neural Information Processing Systems 3. American Institute of Physics, New York*, 313-319 (1991).

Lang, K. & Waibel, A. *Neural Networks* 3, 23-43 (1990).

Levinson, S.E., Rabiner, L.R. & Sondhi, M.M. *An introduction to the application of probabilistic functions of a markov process to automatic speech recognition* 1-1035-1073 (1983).

Loughnan, B.A. & Hall, G.M. *Br J Anaesth* 63, 587-94 (1989).

Maloney P. *Sixth Annual Intelligence Community AI Symposium, Washington DC.* 1988.

Marchette, D. & Priebe, C. *Proc 1987 Triservice Data Fusion Symposium* 1, 230-235 (1987).

Myers RR, Stockard JJ and Saidman LJ. *Stroke*. 8:331-37,(1977).

Pedley TA and Emerson RG. *Recent Advances in Clinical Neurology. Churchill Livingston, New York,* 159-178 (1984).

Prior RF. *Monitoring Cerebral Function.* (JB Lippincott, Philadelphia, 1979).Neurology, T.a.T.A.S.o.t.A.A.o. *Neurology* 40, 1644-1646 (1990).

Radtke, R.A. & Erwin, C.W. *Neurologic clinics* 6, 899-915 (1988).

Rajavelu, A., Musai, M. & Shirvaikar, M. *Neural Networks* 2, 387-393 (1989).

Rumelhart, D.E. & McClelland, J.L. in *Explorations in the microstructure of cognition. Vol 1: Foundations.* (The MIT Press, Cambridge MA, 1987).

Specht, D. *Neural Networks* 3, 109-118 (1990).

Stockard JJ, Bickford RG, Myers RR, Aung MH, Dilley RB, and Schauble JF. *Stroke.*, 5:730-746 (1964).

Trojaborg W and Boysen G. *Electroenceph. Clin. Neurophysiol.* 34:61-69 (1973).

APPLICATION OF NEURAL NETWORKS TO UNSTEADY AERODYNAMIC CONTROL

William E. Faller [1,2], Scott J. Schreck [1] and Marvin W. Luttges [2]

[1] Frank J. Seiler Research Laboratory
USAF Academy, CO

[2] Department of Aerospace Engineering Sciences
University of Colorado, Boulder

Unsteady Fluid Dynamic Models and Control

Problem

Understand → Predict → Control

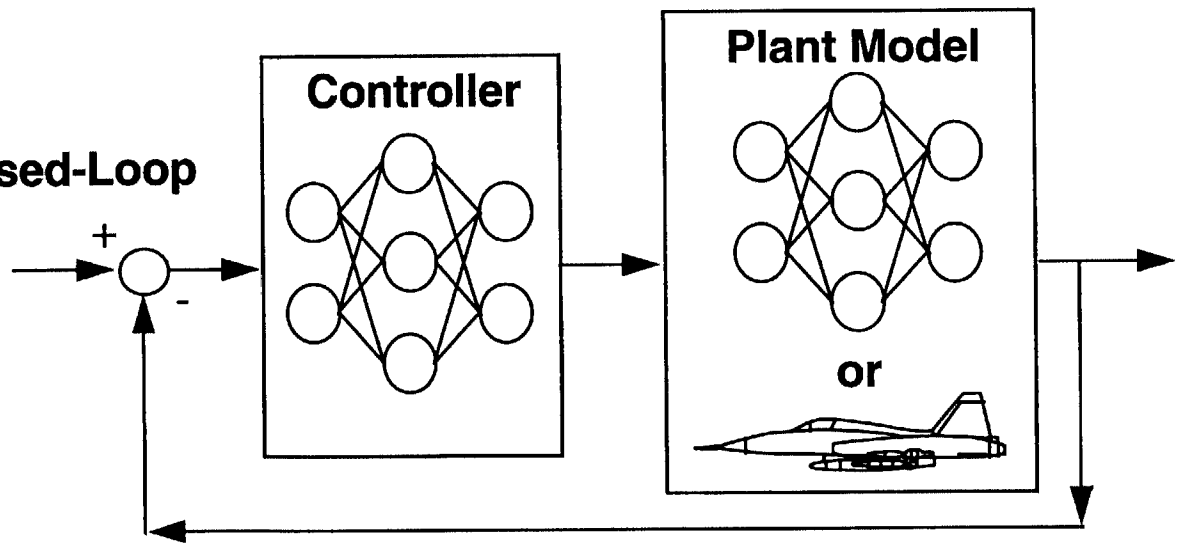
- **Fluid Mechanics of Dynamic Maneuvers**
- **Unsteady Boundary Layers**
- **Vortex Dominated Flows**

Potential Payoffs

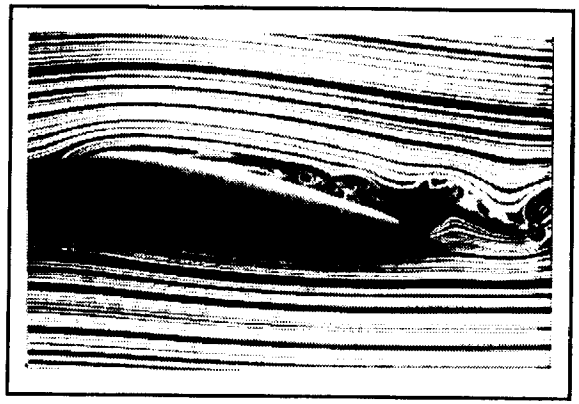
Aircraft, Helicopters, Underwater Vehicles, Wind Turbines ...

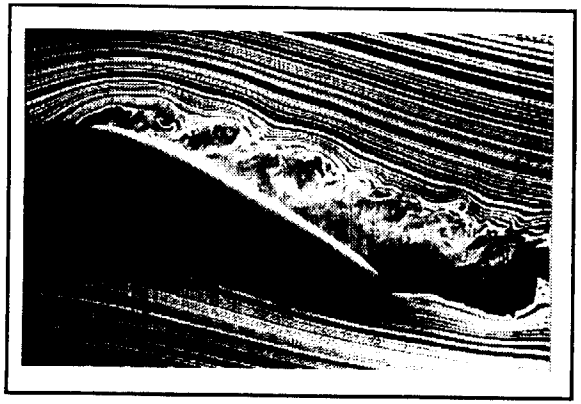
One Solution

Neural Networks: Demonstrate Closed-Loop Control



Pitching Alters Vorticity Generation and Accumulation





STEADY:

Vorticity generated and shed at equal rates

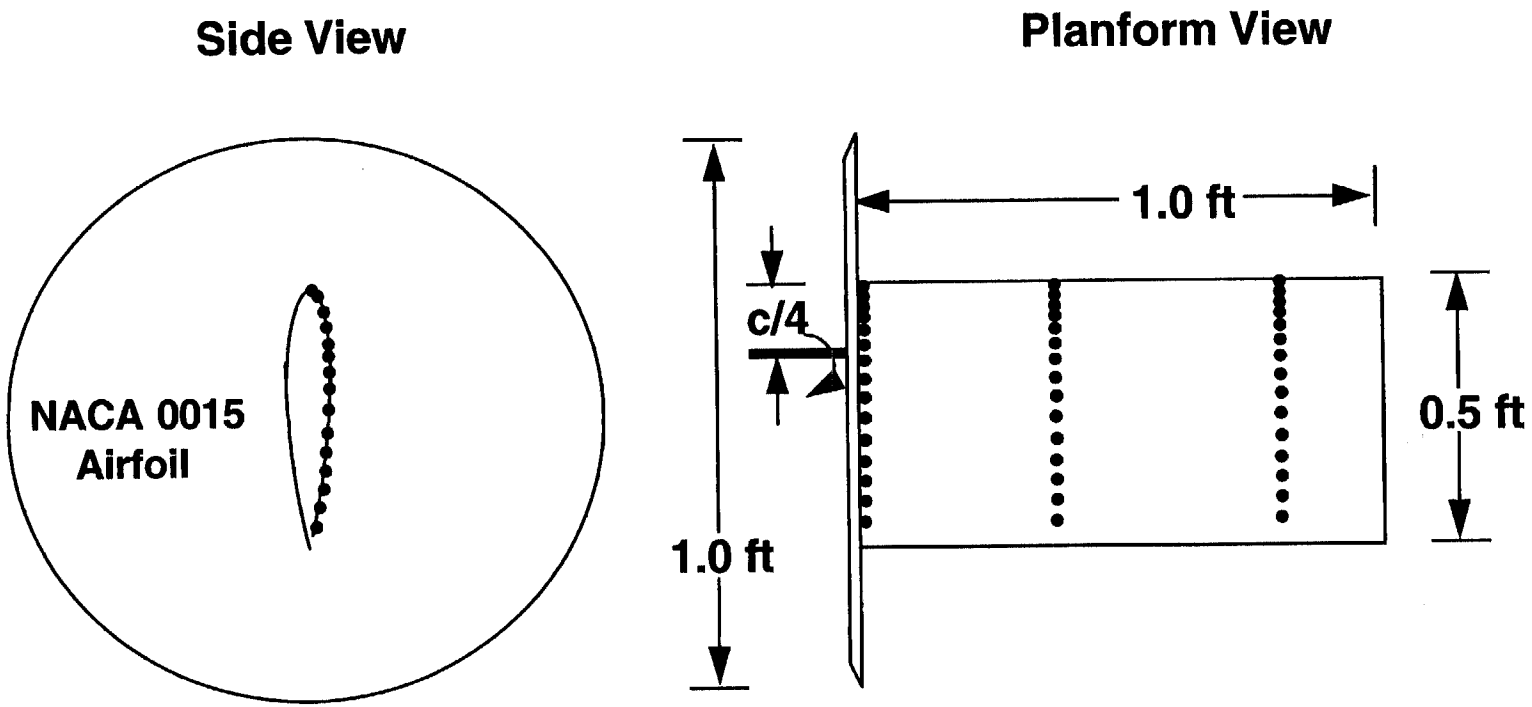
UNSTEADY:

Boundary layer separates, vortex initiated near leading edge

UNSTEADY:

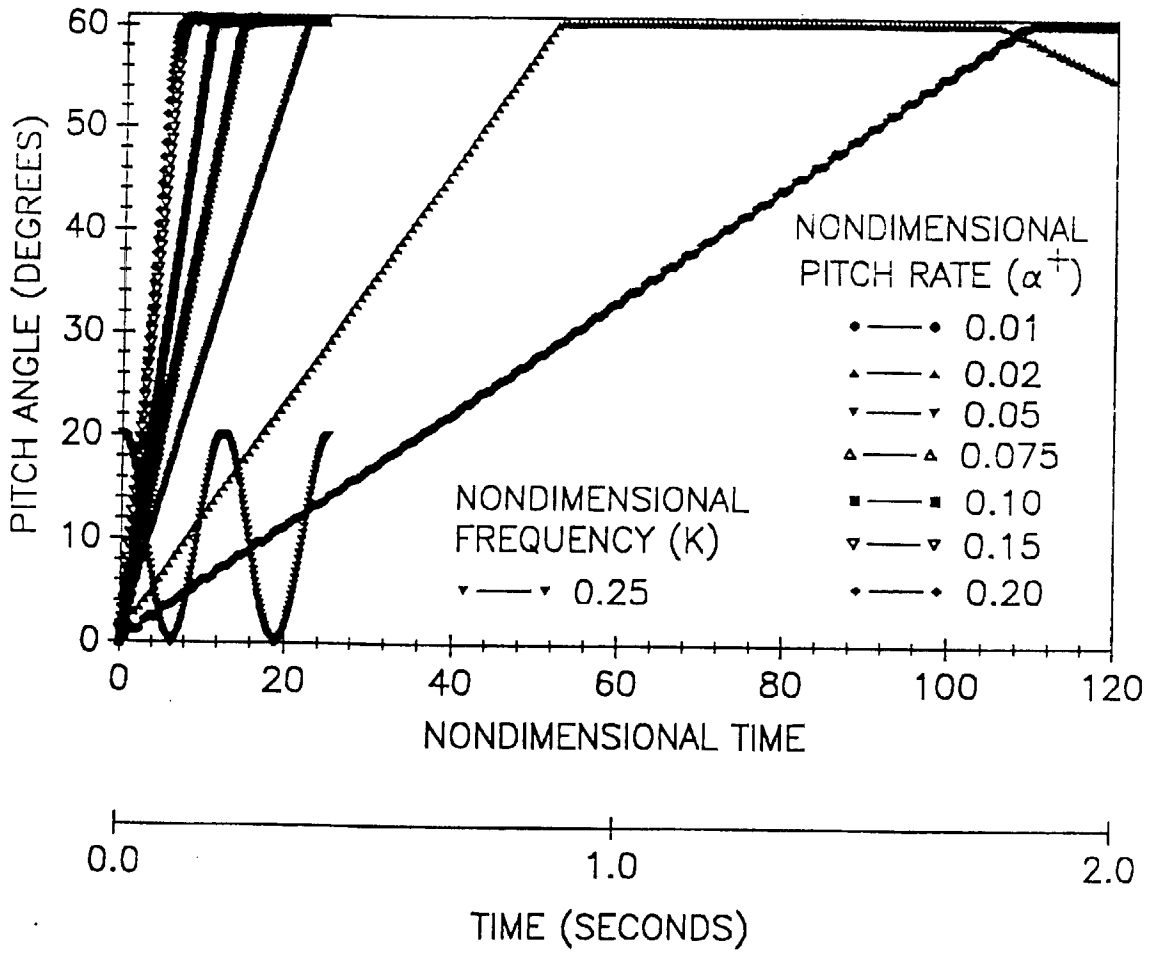
Vorticity accumulates into large, energetic unsteady vortex

Wind Tunnel Wing Model



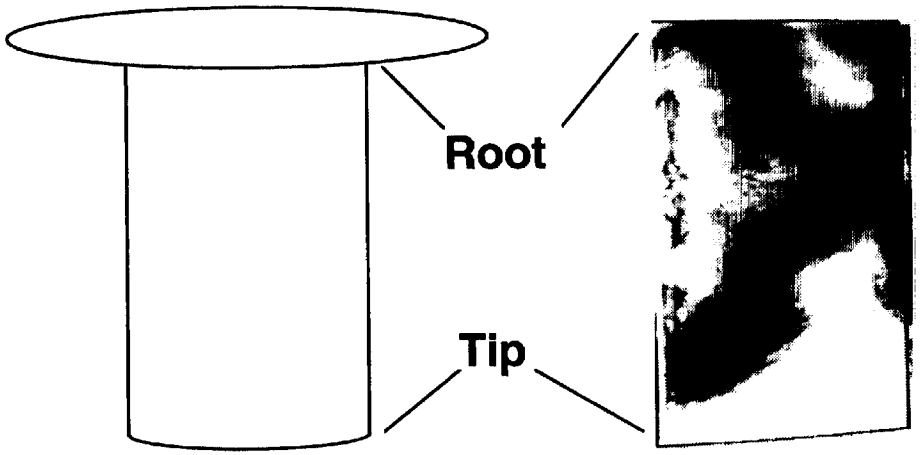
- 15 Pressure Taps (0 to 90% Chord)
- Pressure Transducers Close Coupled With Wing Surface
- 3 Span Locations (Wing Root, 37.5% Span & 80% Span)

Wing Motion Histories



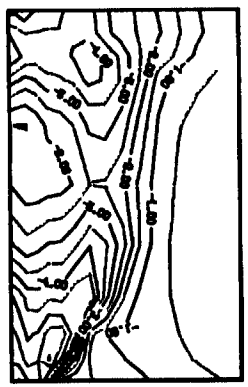
Surface Pressure Topologies and Flow Visualization

- Free wing tip and bounded wing root give 3-D unsteady flow field



Asymmetrical Geometry

Vortex



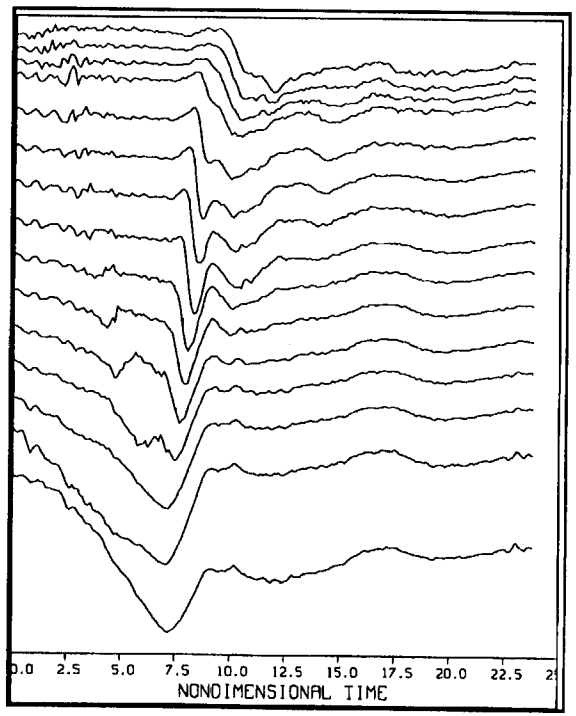
Vortex Signature

Experimental Data Format

- **Data Acquisition Sampling Rate 500 Hz**

90% Chord →

Leading Edge →



Samples

$c_{p15}(t1)$ $c_{p15}(t2)$ .....$c_{p15}(t200)$

$c_{p2}(t1)$ $c_{p2}(t2)$ .....$c_{p2}(t200)$

$c_{p1}(t1)$ $c_{p1}(t2)$ .....$c_{p1}(t200)$

Time

Neural Network Control
Unsteady Aerodynamics

Sensors

REAL-TIME

CONTROLLER

Actuators

Control System Requirements

Constraints

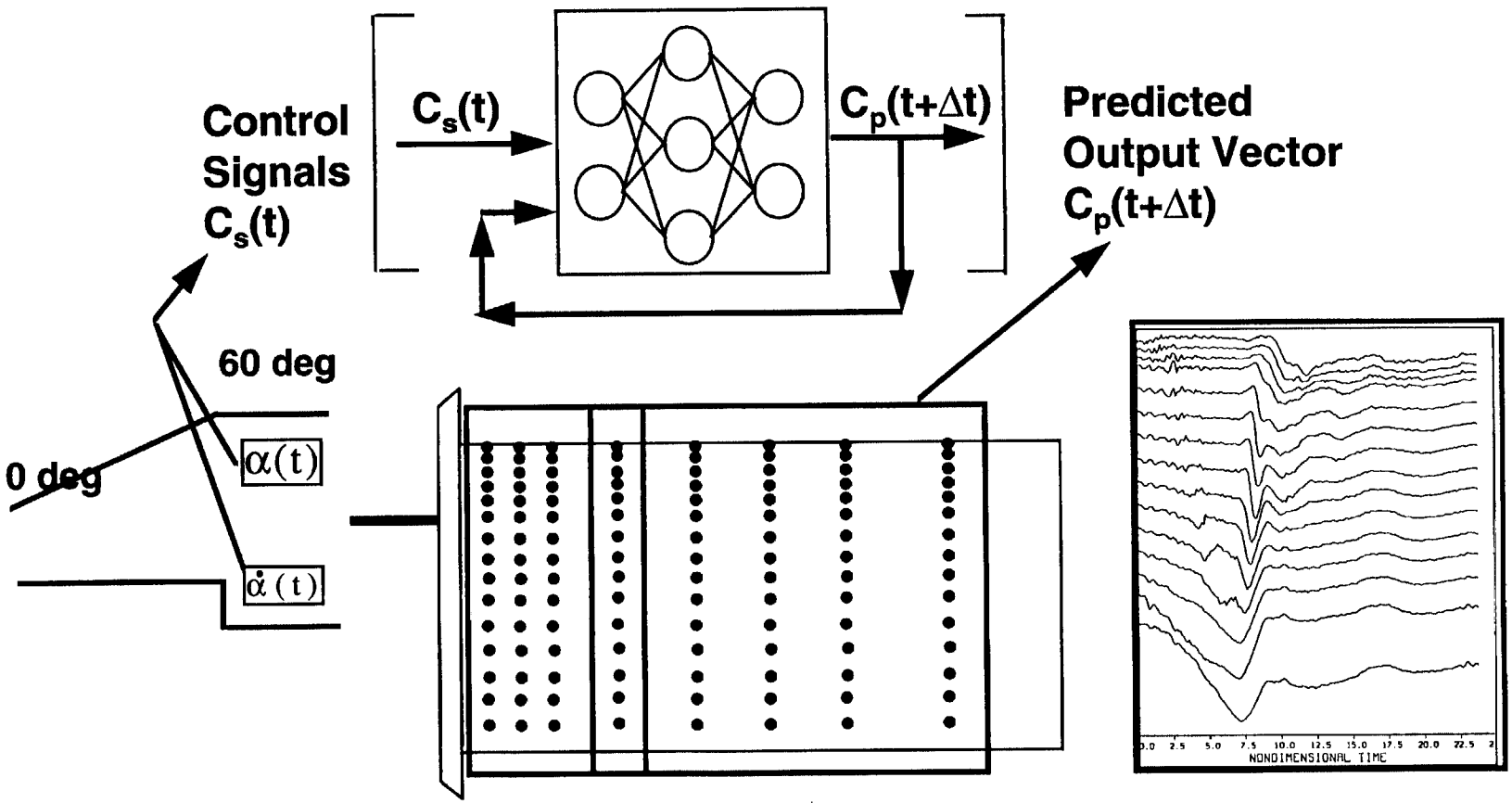
- Plant Output Can Be Highly Nonlinear
- Significant Time Lags Inherent to the System
  (Mechanical, Viscous and Convective)
- System Integration
  (Sensors, Actuators, Controllers, Flow Field Dynamics and the Time-Lags Associated with Each)

Requirements

- Controller for Both Linear and Nonlinear Responses
- Predict the Future State of the Plant / Generate Control Signals with the Required Lead Times
- Many Inputs and Many Outputs in Parallel
- Integrate Multivariate Signals (Sensors, Actuators and Controller)
- Handle Temporal Mismatches (Time-Lags) Automatically

Neural Network Control

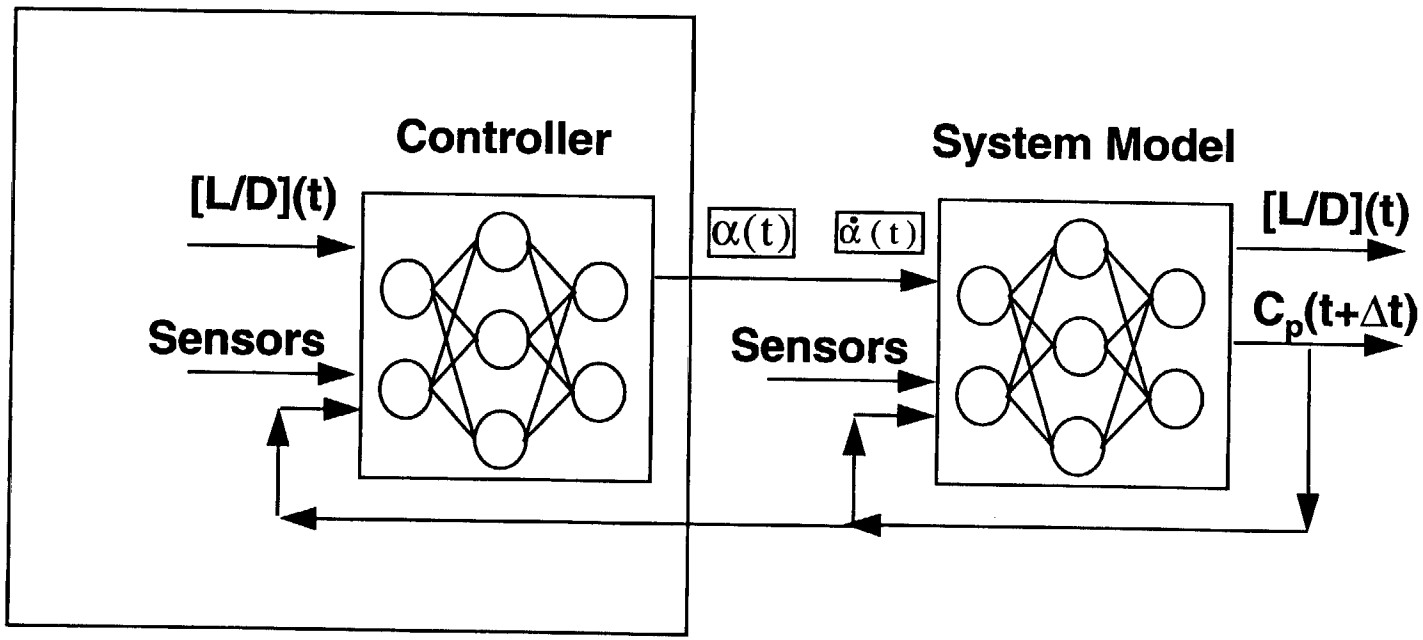
- Develop Neural Network Model of the Plant



- Inputs: Wing Motion History and Recurrent Feedback
- Outputs: Time-Dependent Unsteady Surface Pressures & Forces/Moments

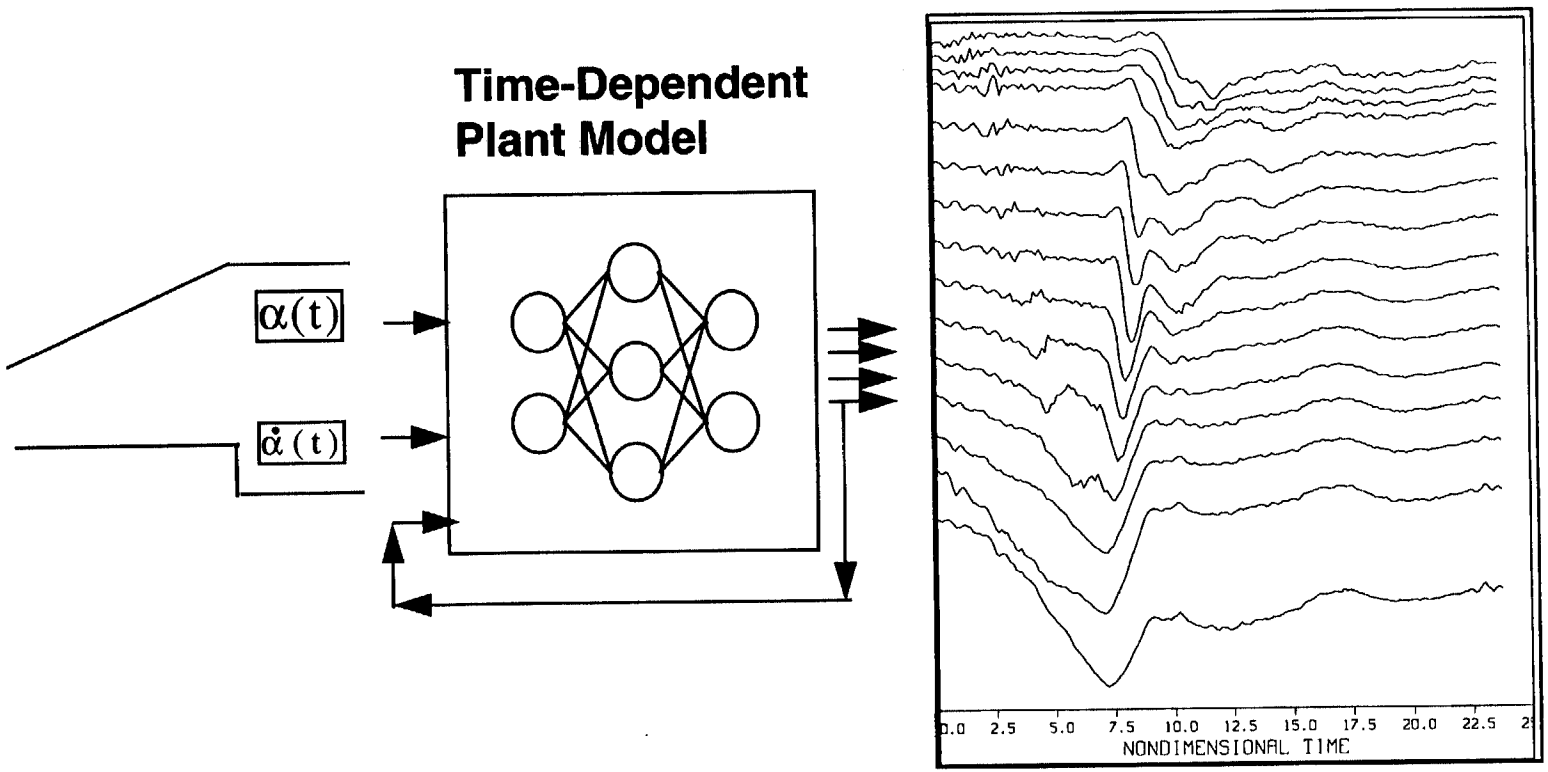
Neural Network Control

- Closed-Loop Control



- Actual system would include sensor inputs to both the plant & controller

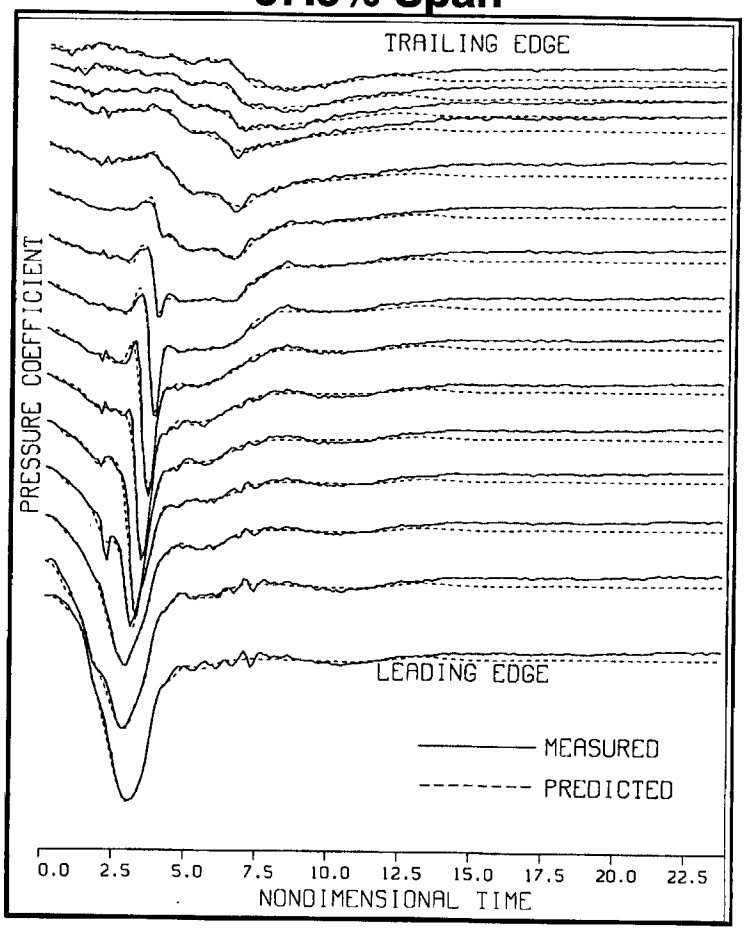
Neural Network Model of Unsteady Flow Field Wing Interactions



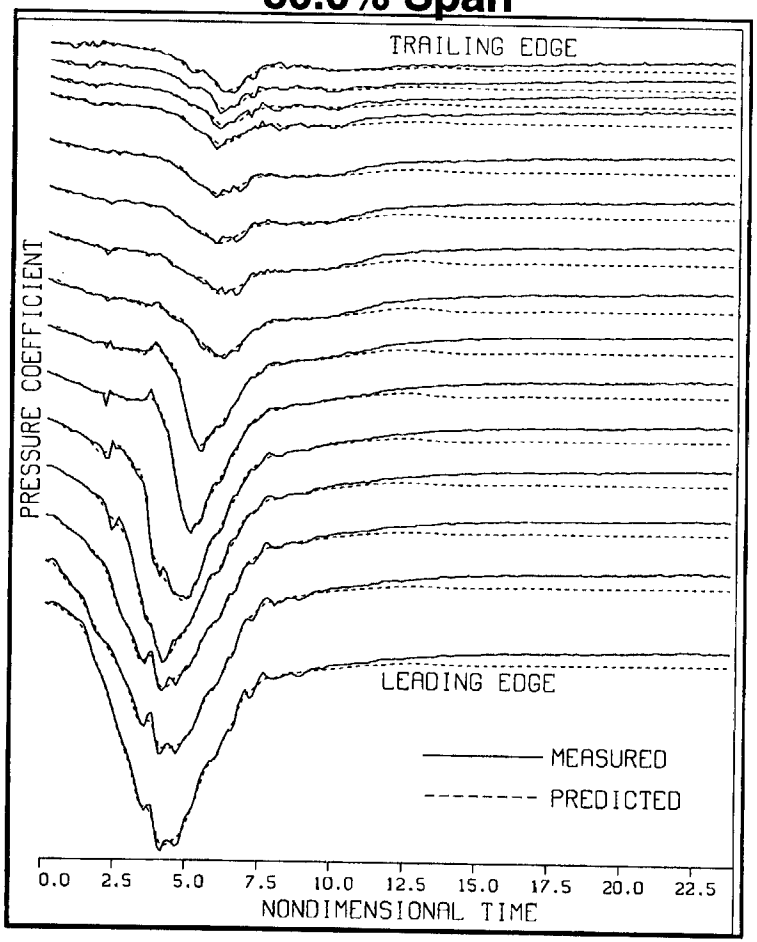
Model Replicates 3-Dimensionality

- **Constant rate pitch motion, $\alpha^+ = 0.20$**

37.5% Span

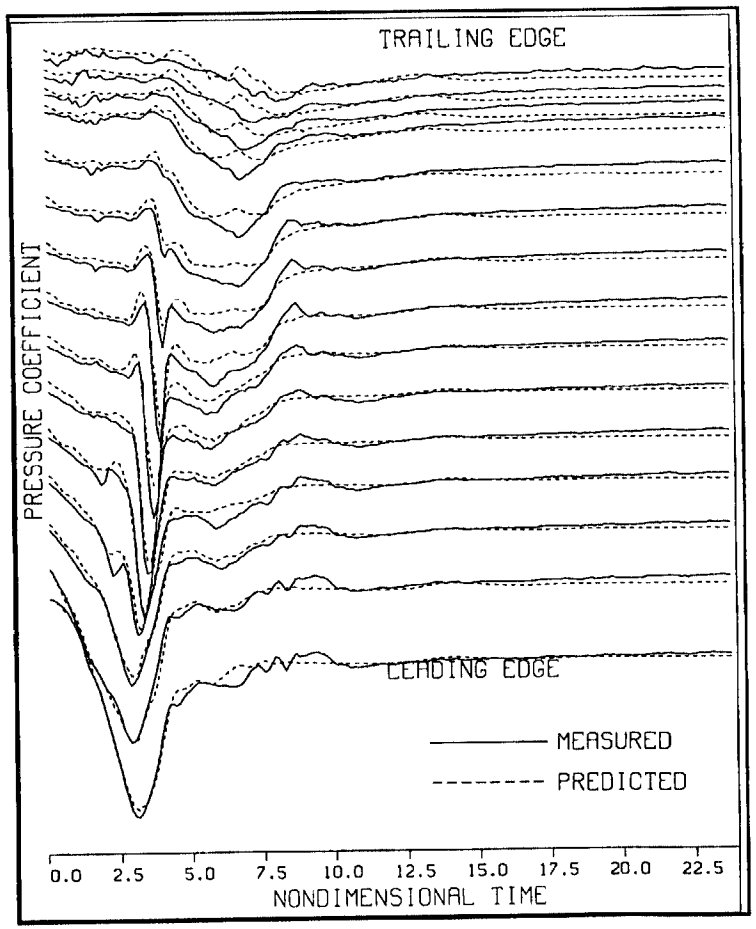


80.0% Span

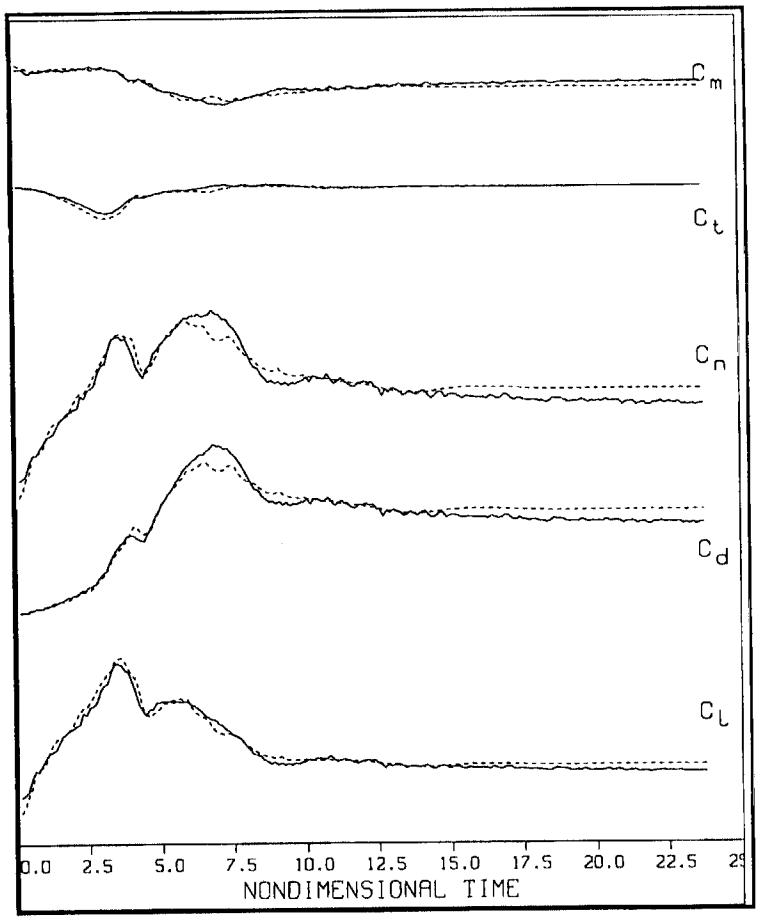


Model Interpolates to Novel Cases

- **Constant rate pitch motion, $\alpha^+ = 0.15$**



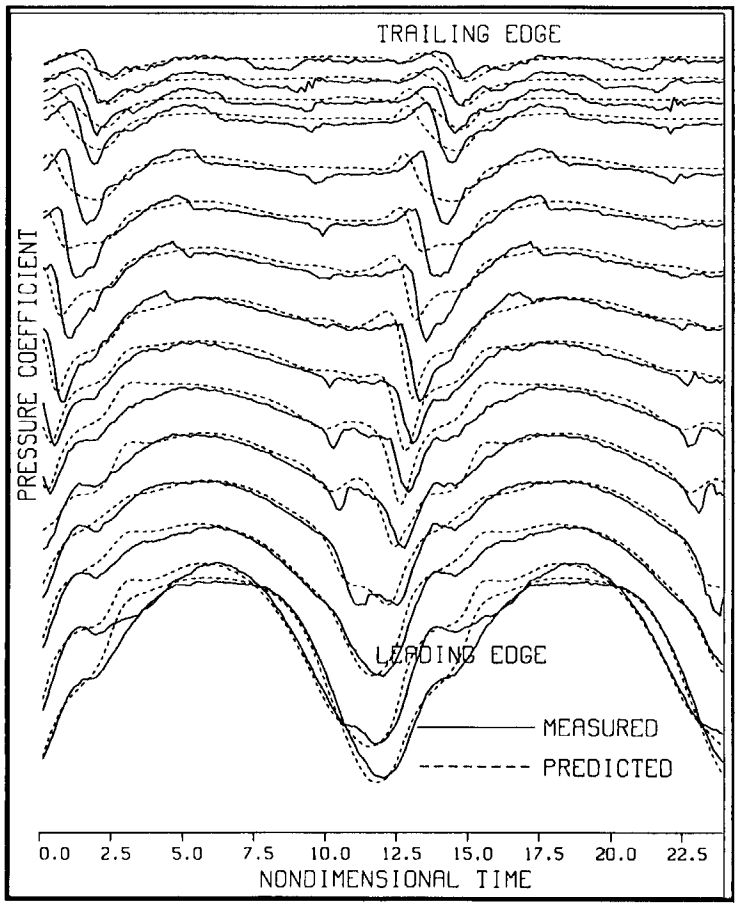
Surface Pressures



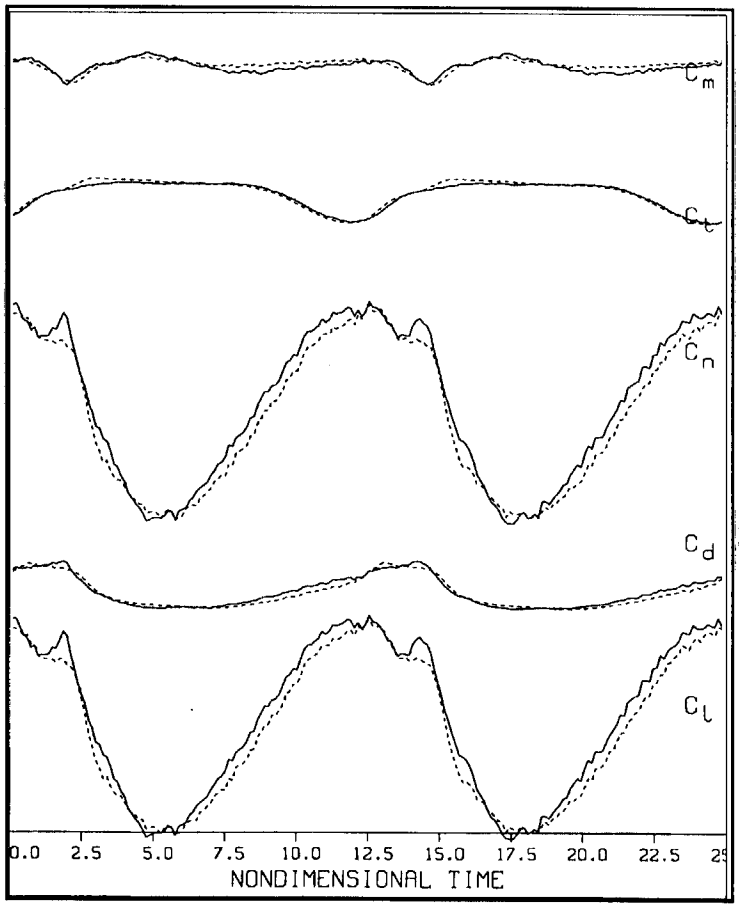
Aerodynamic Coefficients

Model Extrapolates to Novel Cases

- **Harmonic motion history, K = 0.25**

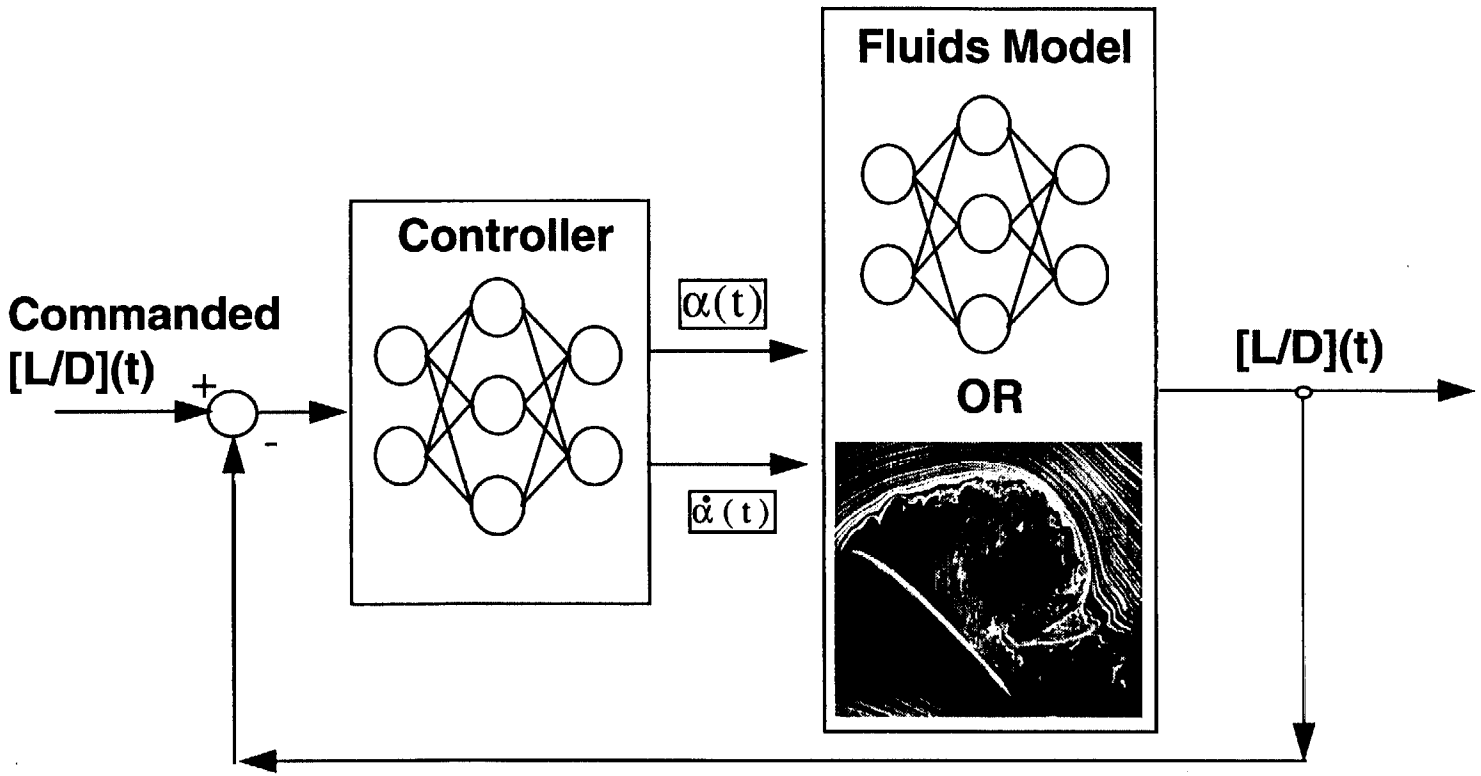


Surface Pressures



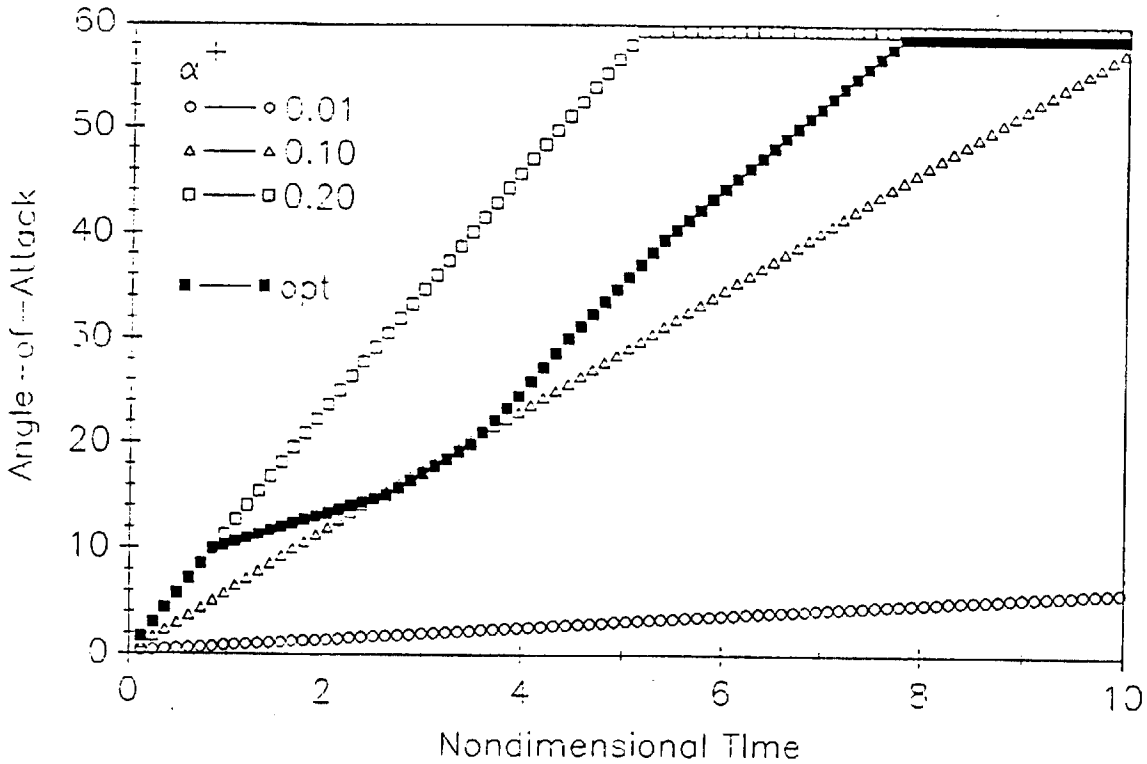
Aerodynamic Coefficients

Closed-Loop Neural Network Control of Unsteady Separated Flows

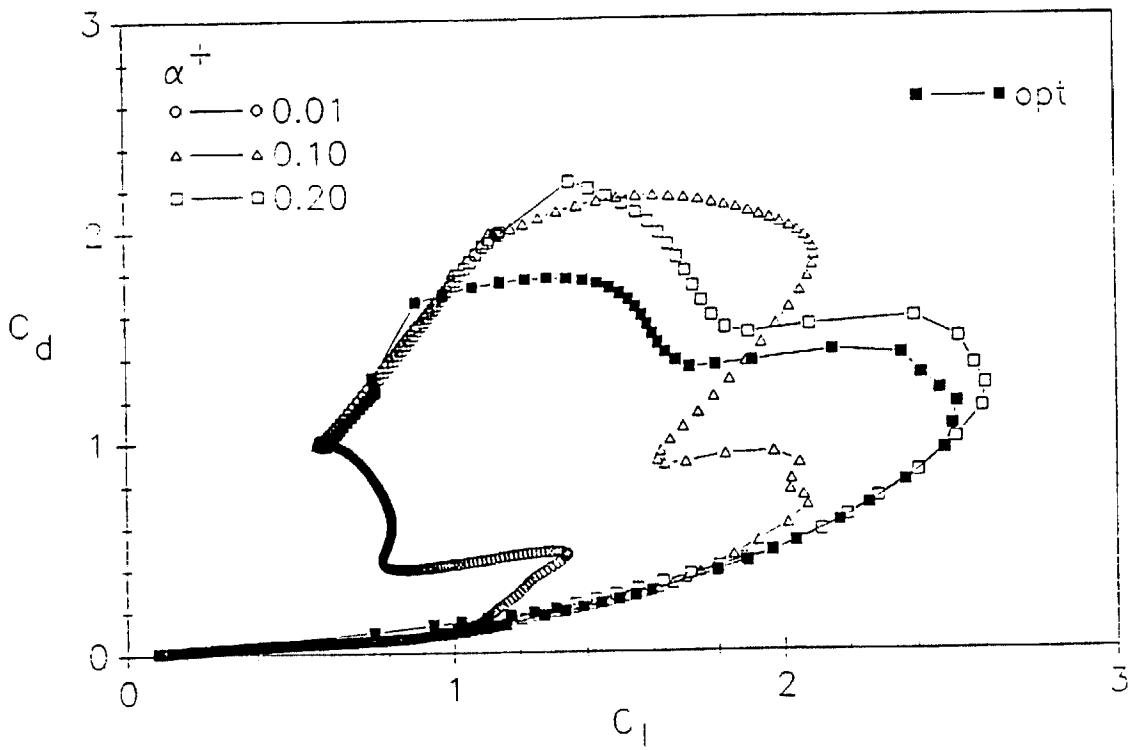


- Demonstrate Optimization and Control of Time-Dependent [L/D]

Neural Network Optimized Wing Motion History for [L/D](t)

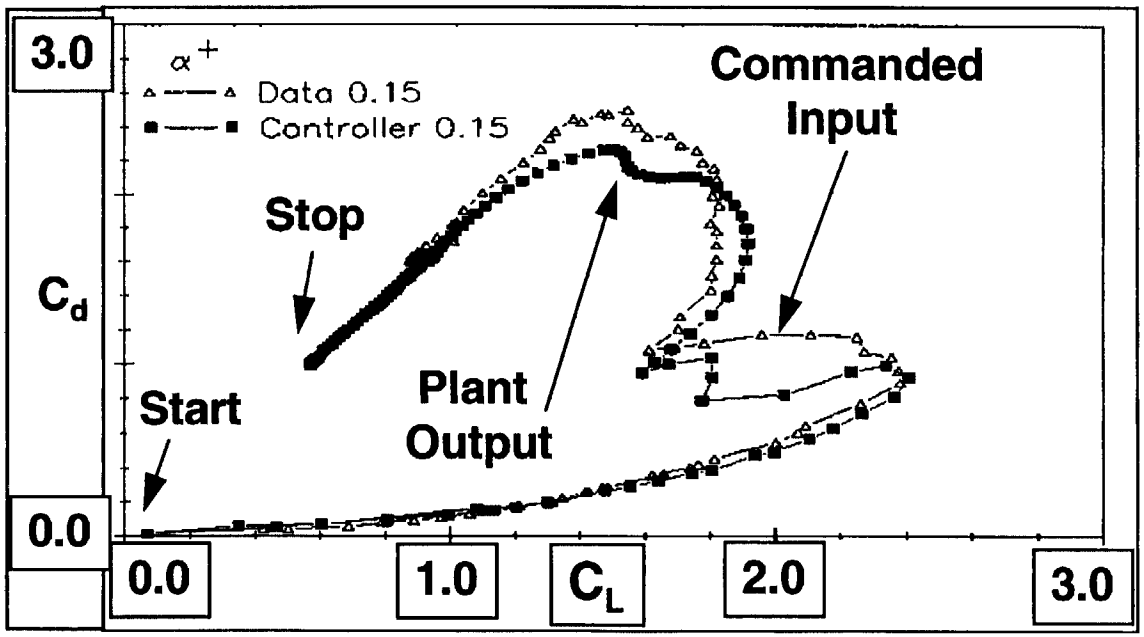


Neural Network Optimized Drag Polar



- Less Than 10% Loss in Lift
- 20% to 40% Decrease in Drag

Neural Network Controller Yields Commanded [L/D](t) for Novel Cases



- Neural Network Controller Trained on Limited Experimental Data
- Neural Network Controller Accurately Interpolates to Novel Cases

CONCLUSION

- Alternative (State-of-the-Art) Solutions

Computational Fluid Dynamics
- Two-Dimensional Solutions (No 3-Dimensionality)
- Computationally Intensive (10's of Hours of Cray Time)

To Date Control Of Unsteady Aerodynamics Has Not Been Demonstrated

- Neural Networks Offer Unique Opportunities:

Simplify Modeling of Three-Dimensional, Vortex Dominated, Unsteady Separated Flow Fields (Practical Geometries & Applications 3-D)

Effective Means for Controlling Unsteady Aerodynamics

Address Integration of Sensors, Actuators, Controllers and Time Lags Into Adaptive Control Systems

Smart Vision Chips: An Overview

Christof Koch
California Institute of Technology

May 1994

1. Four Working Analog VLSI Vision Chips
   (a) Time-Derivative Retina (Delbrück & Mead)
   (b) Zero-Crossing Chip (Bair & Koch)
   (c) Resistive Fuse (Harris & Koch)
   (d) Figure-Ground Chip (Luo, Koch & Mathur)
2. Work in Progress
3. Conceptual and Practical Lessons Learned

Silicon Retina that Computes a Pure Temporal Derivative

T. Delbrück and C. Mead, 1991

- Array of 68 by 43 adaptive, high-gain, logarithmic photoreceptors, implemented in analog CMOS.
- No spatial interactions.
- Array has low offsets and consumes about 4 mW power.
- Array has very **small** fill-factor ($< 3\%$).

1-D Chip that Computes Edges

W. Bair and C. Koch, 1991

- 64 pixel, logarithmic photoreceptors in analog CMOS.
- Each resistive grid implements low-pass filter $\tilde{G}(\omega) = \frac{1}{1+\lambda\omega^2}$. where λ is given by the resistances.
- Chip computes thresholded zero-crossing between two resistive networks (implementing a band-pass filter).
- Output is 63 bit word, indicating presence of edge between adjacent pixels.

Smoothing 2-D Data in the Presence of Discontinuities

J. Harris, C. Koch and J. Luo, 1990

- Algorithmic justification: If values of some variable (for example, depth, hue, intensity) between two adjacent pixels is similar, then smooth away the difference (since it is most likely caused by unavoidable image noise). If the difference is above a threshold, then preserve it, since it is most likely caused by a **discontinuity** between the two locations.
- These constraints can be implemented within a single two-terminal device, the **resistive fuse**.
- Device has nonlinear I-V relationship, similar to an electrical fuse.
- Deterministic annealing can be carried out by dynamically adjusting the I-V relationship.
- Performance of a 20 by 20 pixel analog CMOS chip is shown.

Segregating a "Figure" from "Ground"

J. Luo, C. Koch and B. Mathur 1992

- 48 by 48 pixel resistive grid with configurable switches in analog CMOS.
- Off-chip circuitry detects—possibly incomplete—edges and sets switches appropriately.
- Voltage inside one (or more) figures clearly demarcates them from surrounding pixels.
- Resistive network has natural boundary completion property.

Work in Progress: Computing Motion

- Differential methods to compute velocity (e.g. $v = -I_t/I_x$) are numerically ill-conditioned and require very accurate components.
- Correlation methods to estimate velocity (e.g. $I(x,t) \times I(x+\Delta x, t+\Delta t)$) are robust but expensive in VLSI.
- Computing velocity in the temporal pulse domain appears very promising (Sarpeshkar, Bair and Koch, 1993).
- Special-purpose analog motion sensors can be built for estimating **time-to-contact, observer heading, discontinuities in the optical flow** and other qualitative features of the optical flow field.
- Exploiting Green's theorem

$$\int_A \nabla \cdot \mathbf{V}(x,y)dxdy = \int_C \mathbf{V} \cdot \mathbf{n}ds$$

to compute τ (time-to-collision) in a very robust manner (using a single sensor).

Work in Progress: Neuromorphic Systems

- Carver Mead emphasizes analog VLSI as a medium to model and understand the nervous system (**synthetic neurobiology**).
- Mahowald and Douglas (1991) have successfully built **pyramidal cells** in analog CMOS, including dendritic trees, EPSPs and IPSPs and nonlinear membrane conductances.
- Koch, Douglas, Sejnowski and Lisberger are involved in long-term project to build a complete **oculo-motor system** (including two retinae on movable platform, superior colliculus, brain stem nucleus for eye plant, and cortical areas) based upon the visual system of primates.

What Lessons Have We Learned

- Conception, design and fabrication of smart vision chips **must** go hand-in-hand with the design of the appropriate vision algorithms.
- It is crucial to understand what types of computations map naturally onto analog hardware and which ones are better suited to *Turing universal* digital machines (e.g. motion analysis).
- Important to integrate adaptation and learning abilities at all levels of the circuitry (from photoreceptors to output).

What Should We Do

- Principal limitation of today's circuits is **not** small array size ($< 100 \times 100$ pixels) but lack of further on-chip processing power.
- Do not emphasize development of very costly basic fabrication and circuit technology at the expense of inexpensive algorithmic development and implementation.
- Development of interchip communication protocols (e.g. Mahowald and Mead's event-driven addressing scheme).
- Design not just smart add-on's, but complete, autonomous systems.

Predictability in Space Launch Vehicle Anomaly Detection Using Intelligent Neuro-Fuzzy Systems

JPL *JSC* *McDonnell Douglas* *Lockheed*
Joint Effort

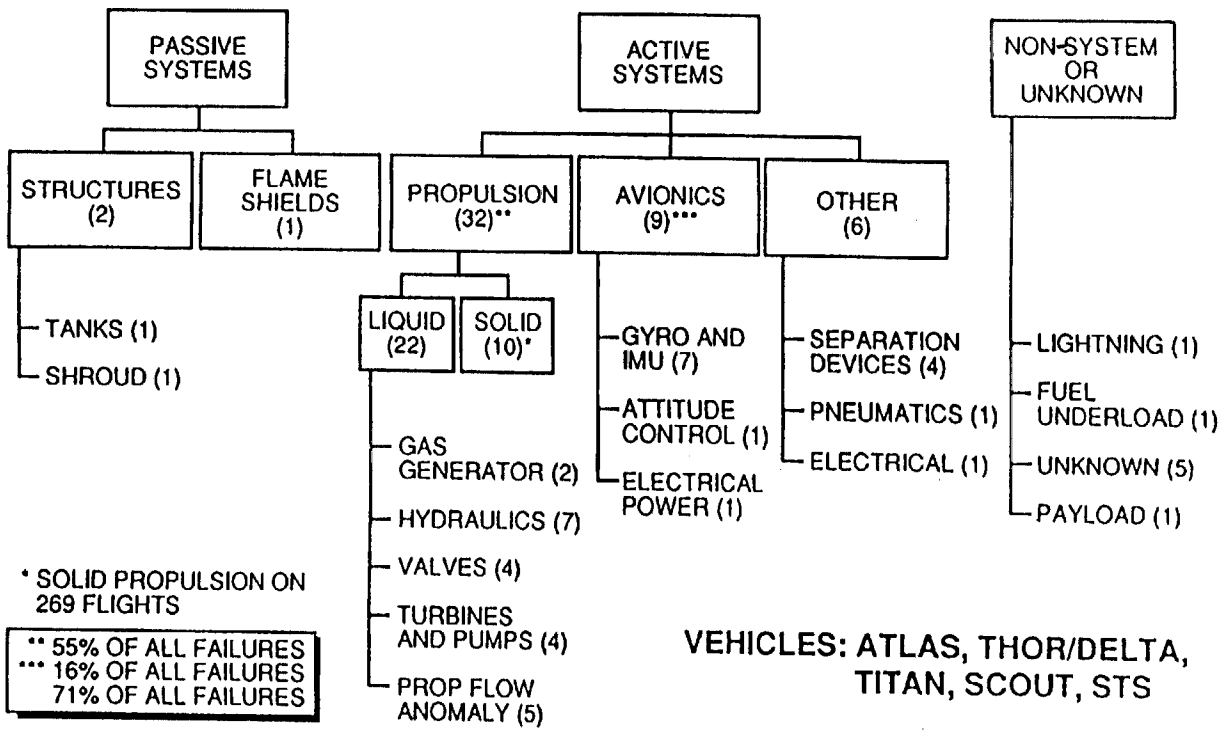
JPL Team

Sandeep Gulati
Raoul Tawel
Nikzad Toomarian
Anil Thakoor
Jacob Barhen
Taher Daud
Ayanna Maccalla

Jet Propulsion Laboratory
California Institute of Technology
Center for Space Microelectronics Technology
Pasadena, CA

INTELLIGENT NEUROPROCESSORS FOR LAUNCH VEHICLE HEALTH MANAGEMENT SYSTEMS

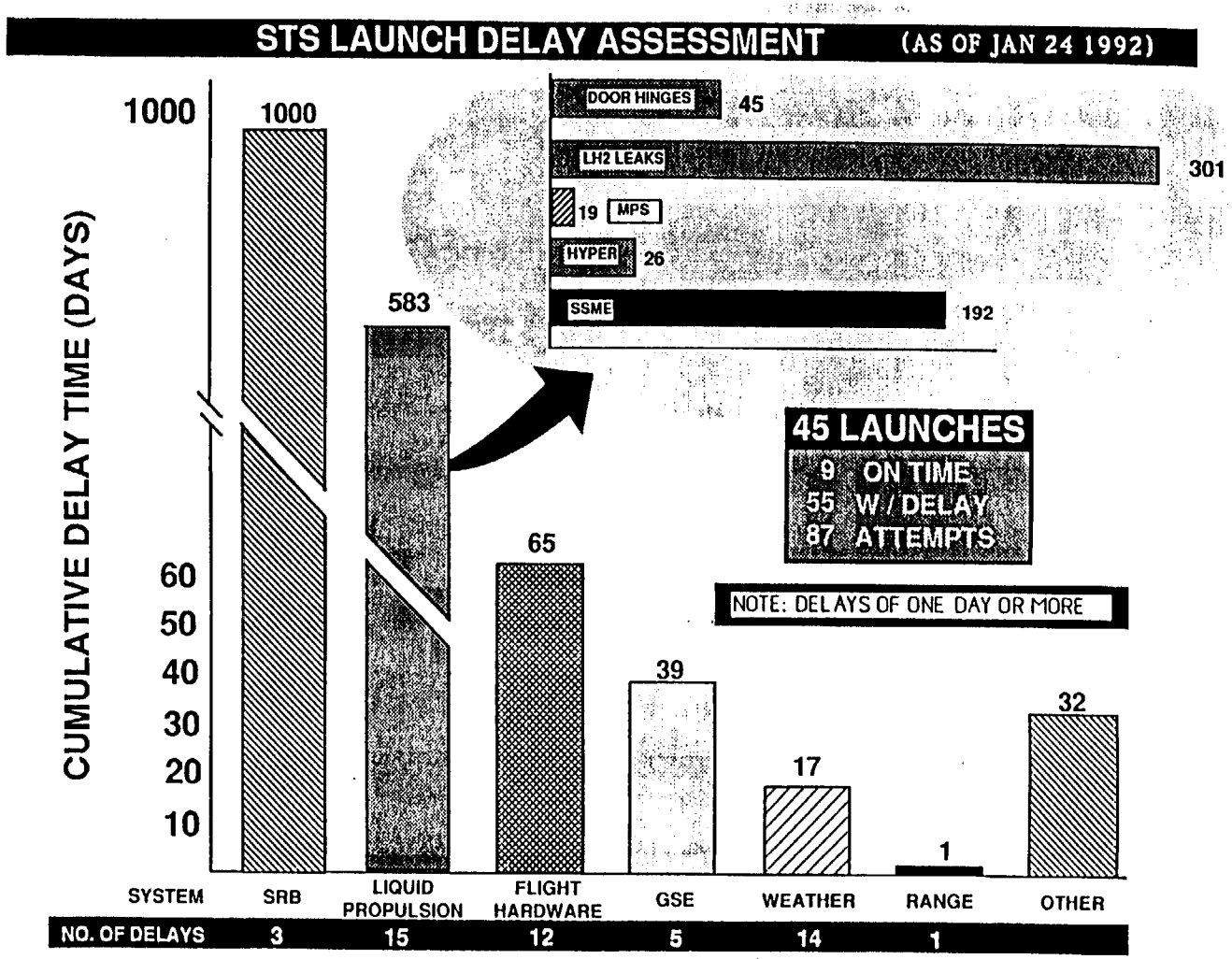
742 TOTAL FLIGHTS (1966-87), 58 failures



Where The Flight Failures Have Been In Launch Vehicles

JPL

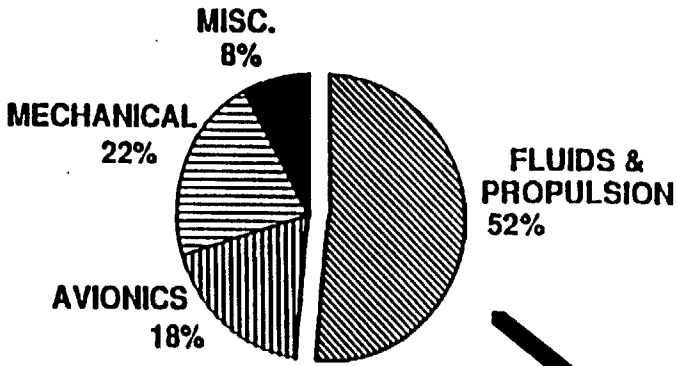
INTELLIGENT NEUROPROCESSORS FOR LAUNCH VEHICLE HEALTH MANAGEMENT SYSTEMS

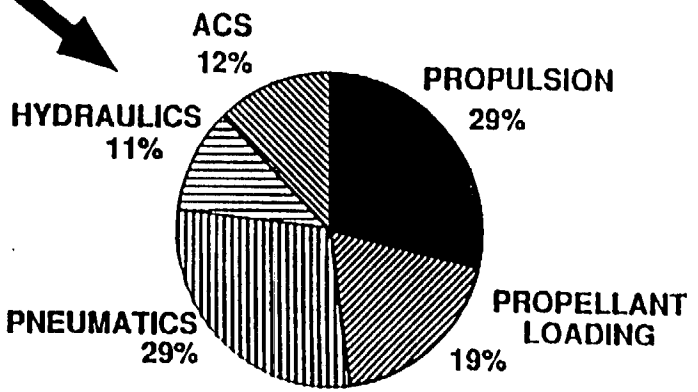


JPL

INTELLIGENT NEUROPROCESSORS FOR LAUNCH VEHICLE HEALTH MANAGEMENT SYSTEMS

Breakdown of Operations Hours





FLUIDS AND PROPULSION SYSTEMS ACCOUNT FOR *HALF* OF ALL TESTING HOURS.

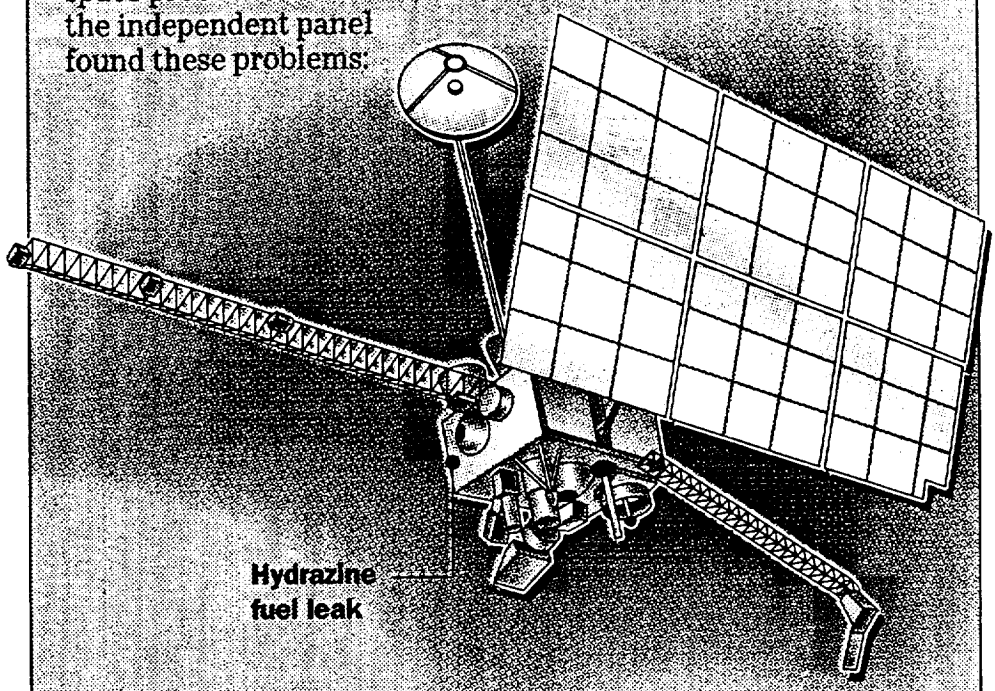
SPACEPORT FLORIDA INFRASTRUCTURE IMPROVEMENT STUDY

JPL

Failure of Mars Probe Blamed on Fuel Leak

Troubled Spacecraft

A federal panel Wednesday announced the findings of its inquiry into the Aug. 21 disappearance of the $980 million Mars Observer spacecraft. Exactly what happened to the space probe is not known, but the independent panel found these problems:



- **Mechanical flaw:** A leak of volatile hydrazine fuel may have caused an explosion when the spacecraft's tanks were pressurized.

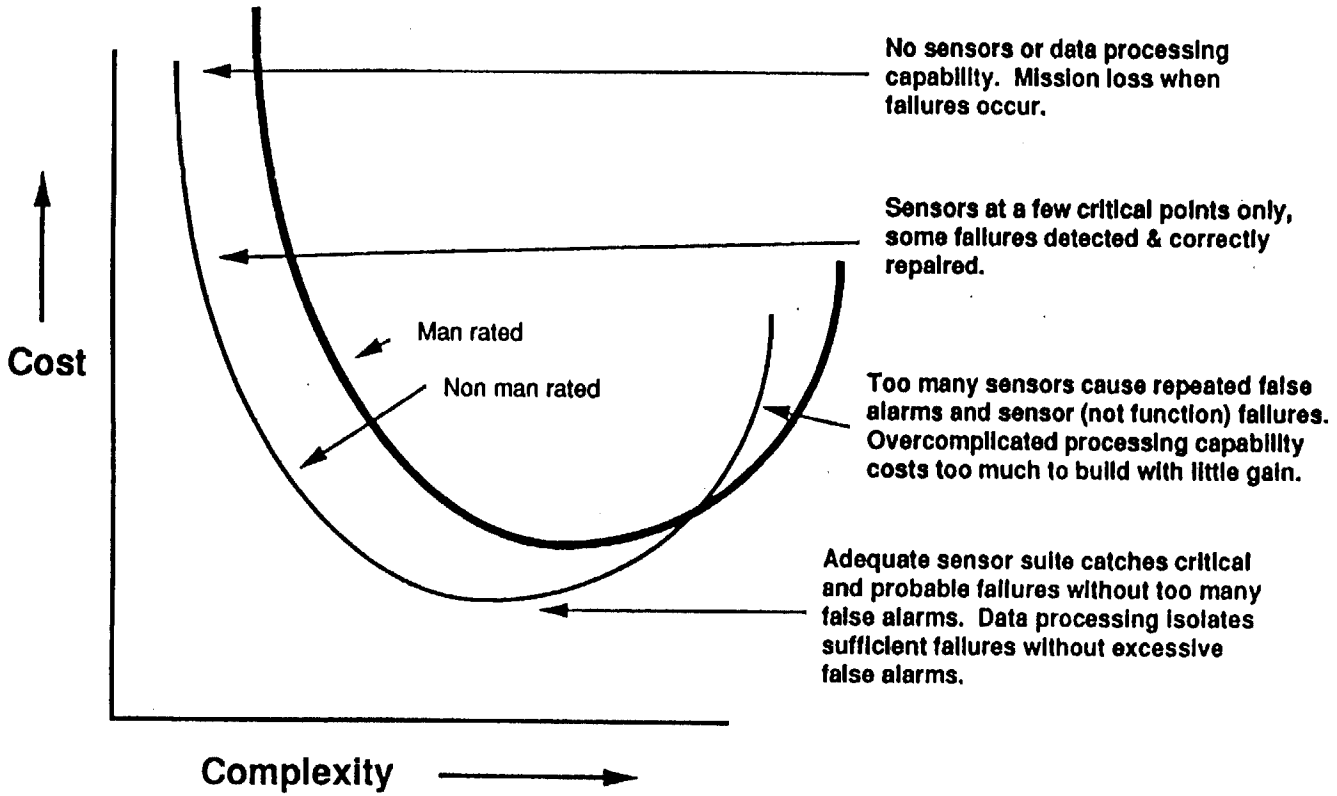
- **Design flaw:** NASA engineers used technology that had been developed for operation in near-Earth orbit but was unsuitable for the more extreme conditions of interplanetary space.

- **Management flaw:** Project managers at the Jet Propulsion Laboratory did not exercise sufficient control over continuing changes in the spacecraft's design and its scientific instruments.

Source: NASA

Los Angeles Times

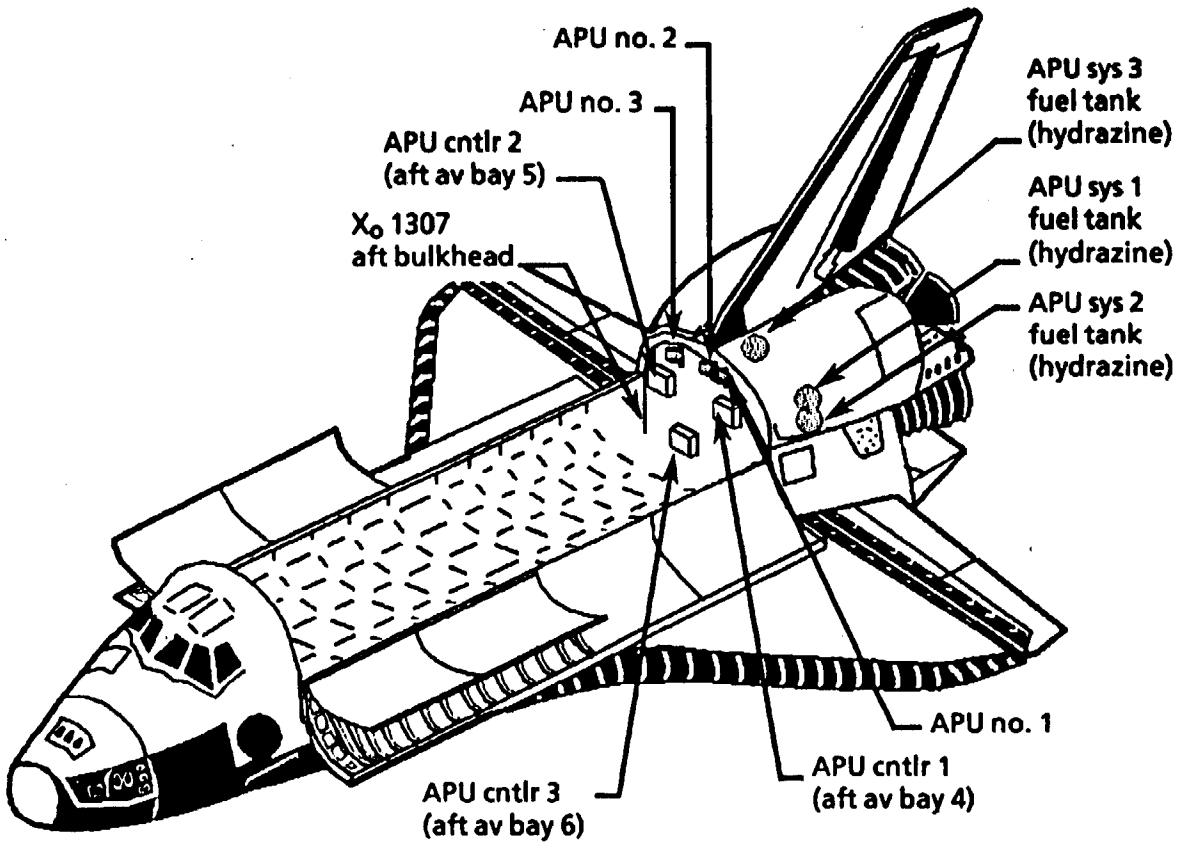
INTELLIGENT NEUROPROCESSORS FOR LAUNCH VEHICLE HEALTH MANAGEMENT SYSTEMS



VHM COST OPTIMIZING CURVE

JPL

INTELLIGENT NEUROPROCESSORS FOR LAUNCH VEHICLE HEALTH MANAGEMENT SYSTEMS



TARGET HMS - STS Auxiliary Power Unit Location

JPL

INTELLIGENT NEUROPROCESSORS FOR LAUNCH VEHICLE HEALTH MANAGEMENT SYSTEMS

AUXILIARY POWER UNIT

- Provide power for the Orbiter hydraulic systems
  - liquid hydrazine -------> mechanical shaft power
- Hydraulic systems
  - actuate the Orbiter aerosurfaces
  - throttle and steer Orbiter main engines
  - deploy and steer landing gear
  - apply landing gear brakes
- Operation Cycle
  - t-5 min to OMS-1 burn
  - deorbit burn and entry to just before landing

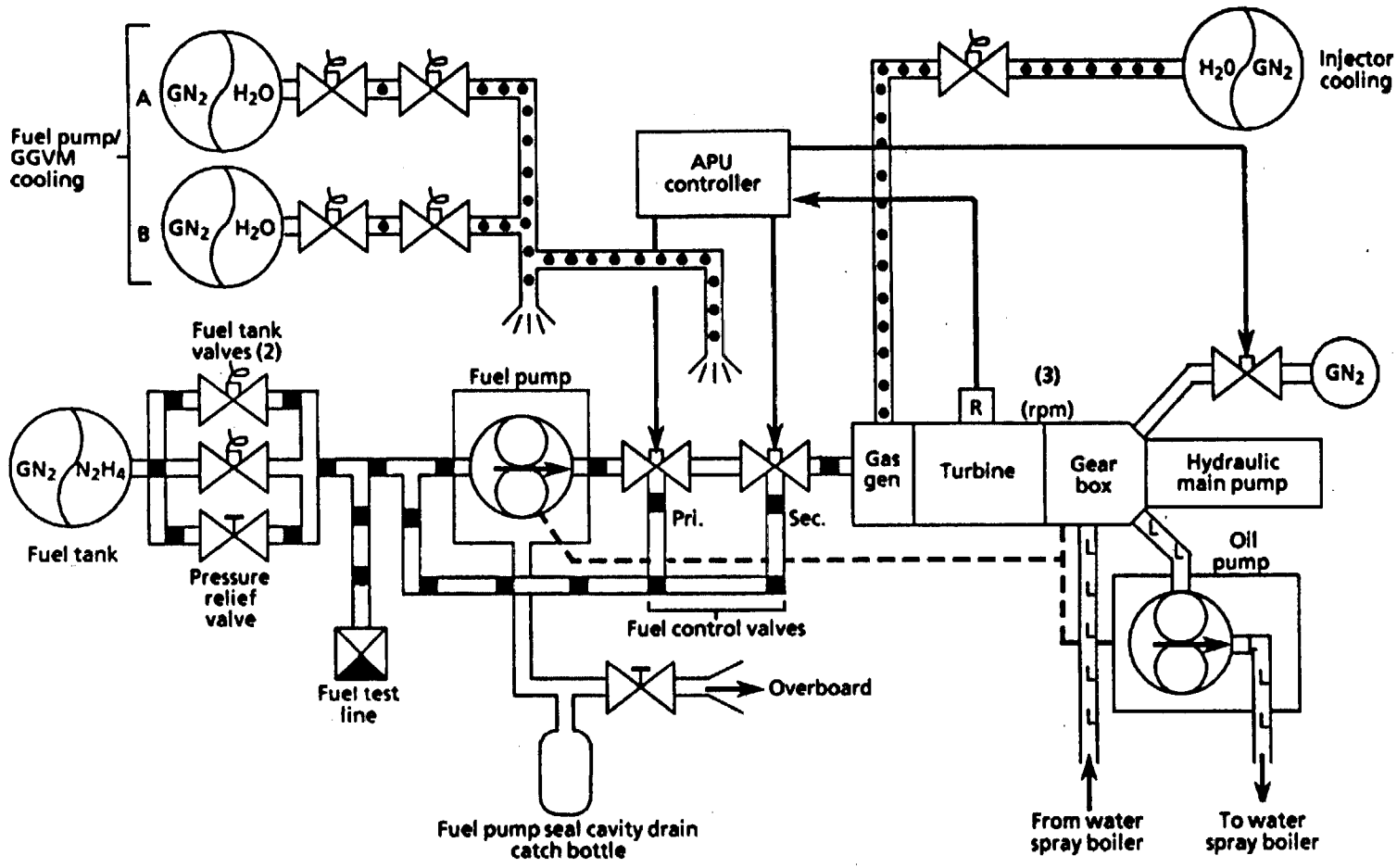
JPL

INTELLIGENT NEUROPROCESSORS FOR LAUNCH VEHICLE HEALTH MANAGEMENT SYSTEMS

- **Monitoring fuel tank isolation, fuel control valves and electronic controller, e.g.,**
  - **valve open for > 2 min in orbit without fuel flow could detonate hyrazine near valve**
  - **leakage detection**
  - **high rmp pulser-type valves**

APU MONITORING AND DIAGNOSIS

INTELLIGENT NEUROPROCESSORS FOR LAUNCH VEHICLE HEALTH MANAGEMENT SYSTEMS



TARGET HMS - STS Auxiliary Power Unit

JPL

INTELLIGENT NEUROPROCESSORS FOR LAUNCH VEHICLE HEALTH MANAGEMENT SYSTEMS

TECHNOLOGY ISSUES

- Engineering alarm limits - critical thresholds which define the acceptable range of engineering values on any telemetry channel

  - determined manually: hardcopy ISOE data, design information on spacecraft, rules of thumb

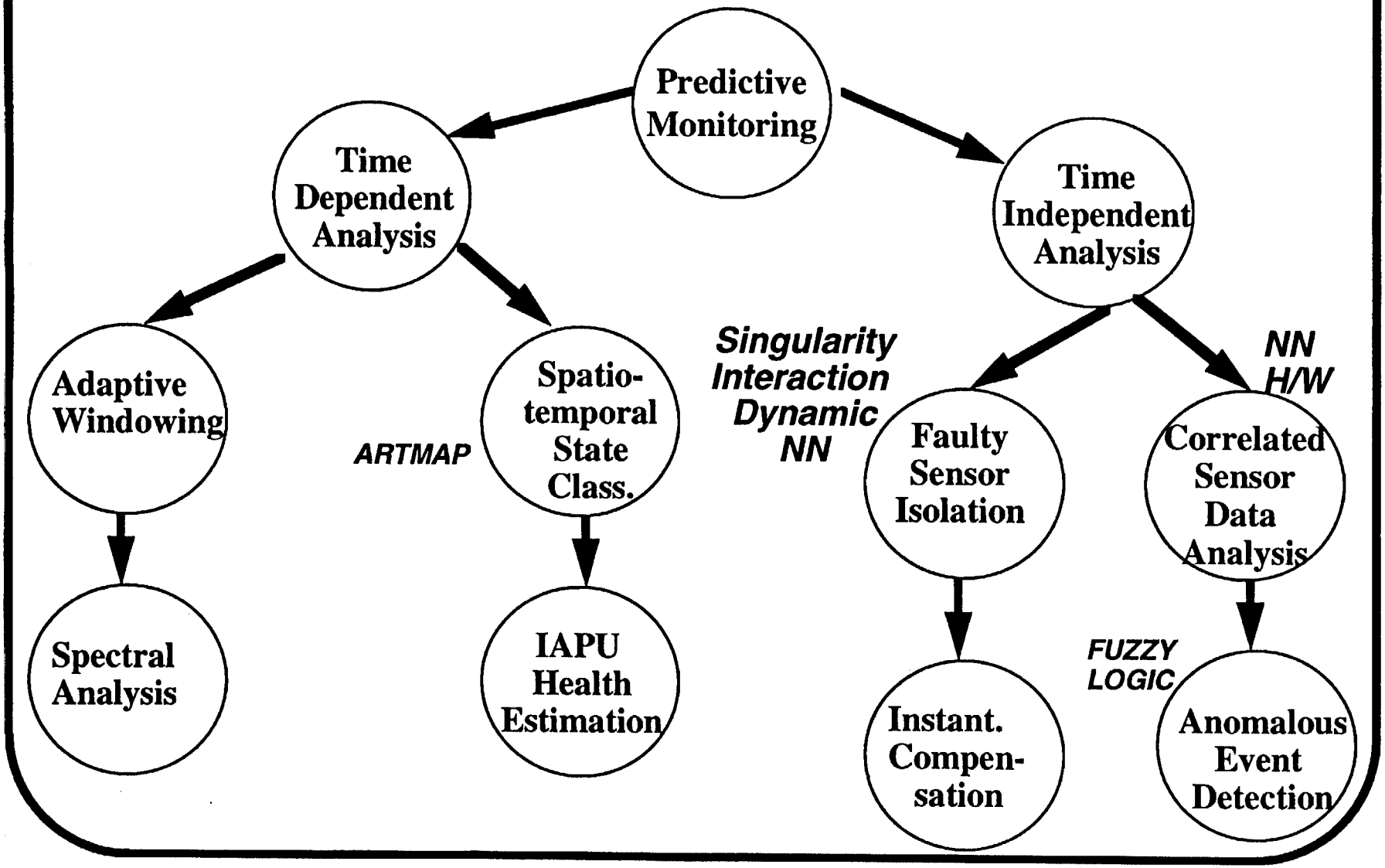
    - Overreliance on domain experts leading to wide thresholds creating a range of undetected anomalies
    - monitoring of individual sensors via redlining approach

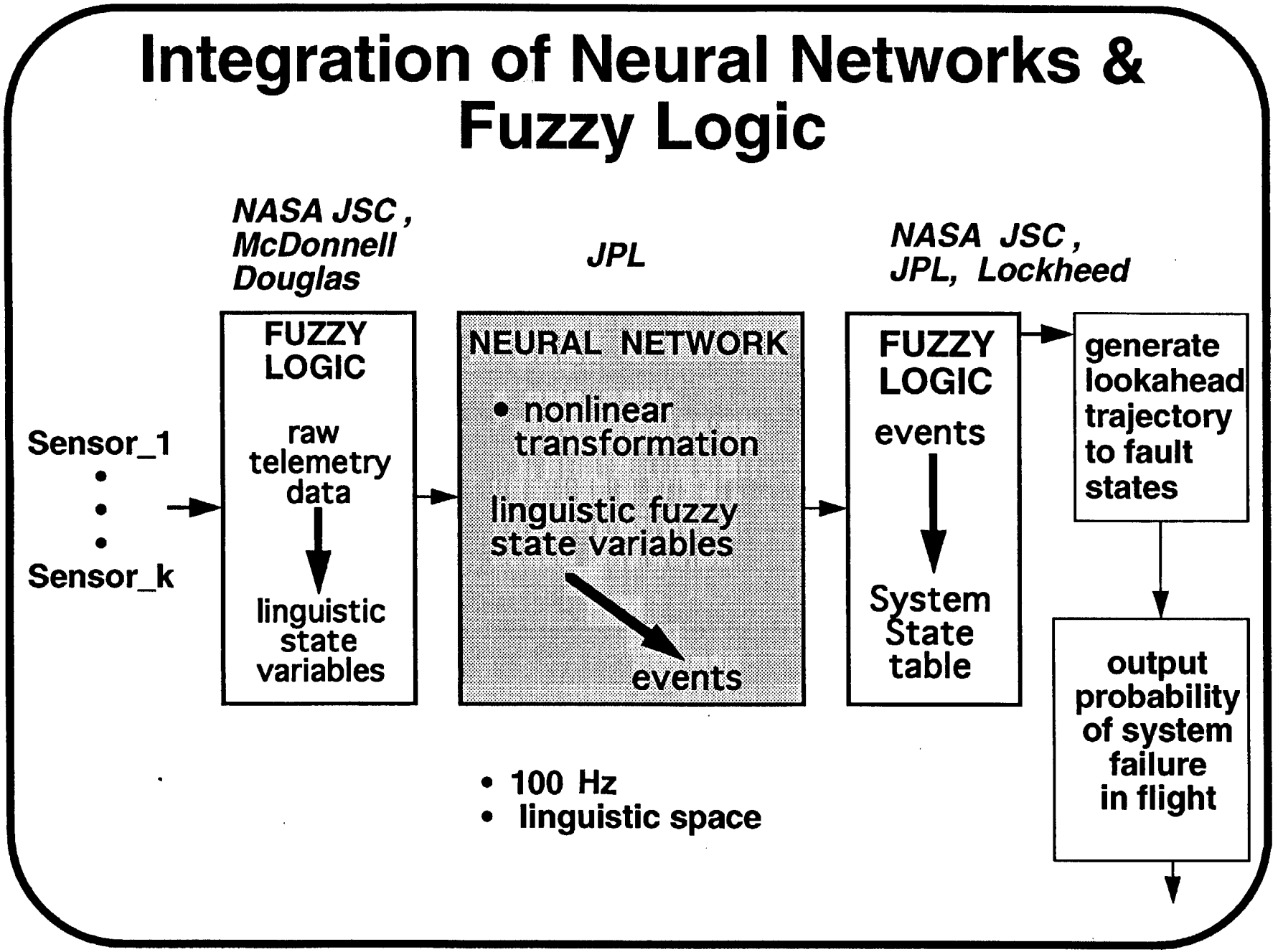
- Access only to snapshots of telemetry due to exploitation of low sensor acquisition rates. Further degradation due to noisy and incomplete data

- Specific diagnostics can be executed only if they were preconceived and preprogrammed

  - cannot currently correlate effects between multiple sensors in real-time
  - fault-detection to engine catastrophy time can be as short as 0.1 sec.

INTELLIGENT NEURO-FUZZY SYSTEM for STS APU Health Monitoring




Integration of Neural Networks & Fuzzy Logic
NASA JSC , McDonnell Douglas
JPL
NASA JSC , JPL, Lockheed
Sensor_1
Sensor_k
FUZZY LOGIC
raw telemetry data
linguistic state variables
NEURAL NETWORK
• nonlinear transformation
linguistic fuzzy state variables
events
FUZZY LOGIC
events
System State table
generate lookahead trajectory to fault states
output probability of system failure in flight
• 100 Hz
• linguistic space

INTELLIGENT NEUROPROCESSORS FOR LAUNCH VEHICLE HEALTH MANAGEMENT SYSTEMS

STS / APU HEALTH MONITORING

- detection of all red line errors currently identified
- real-time correlation of data from multiple heterogeneous sensors
  - faster-than-real-time anomaly propagation to determine probability of failure
  - both with (using NN s/w) and without (using NN h/w) time-lags
- ease of augmenting expert-generated APU fault knowledge base without needing to redesign the system
- isolating failed sensors as against failed subsystem / system
  - reconstruct suspect information and minimize disruption of diagnostic process
- synergistic integration of fuzzy logic and neural networks for real-time diagnostic applications

JPL

INTELLIGENT NEUROPROCESSORS FOR LAUNCH VEHICLE HEALTH MANAGEMENT SYSTEMS

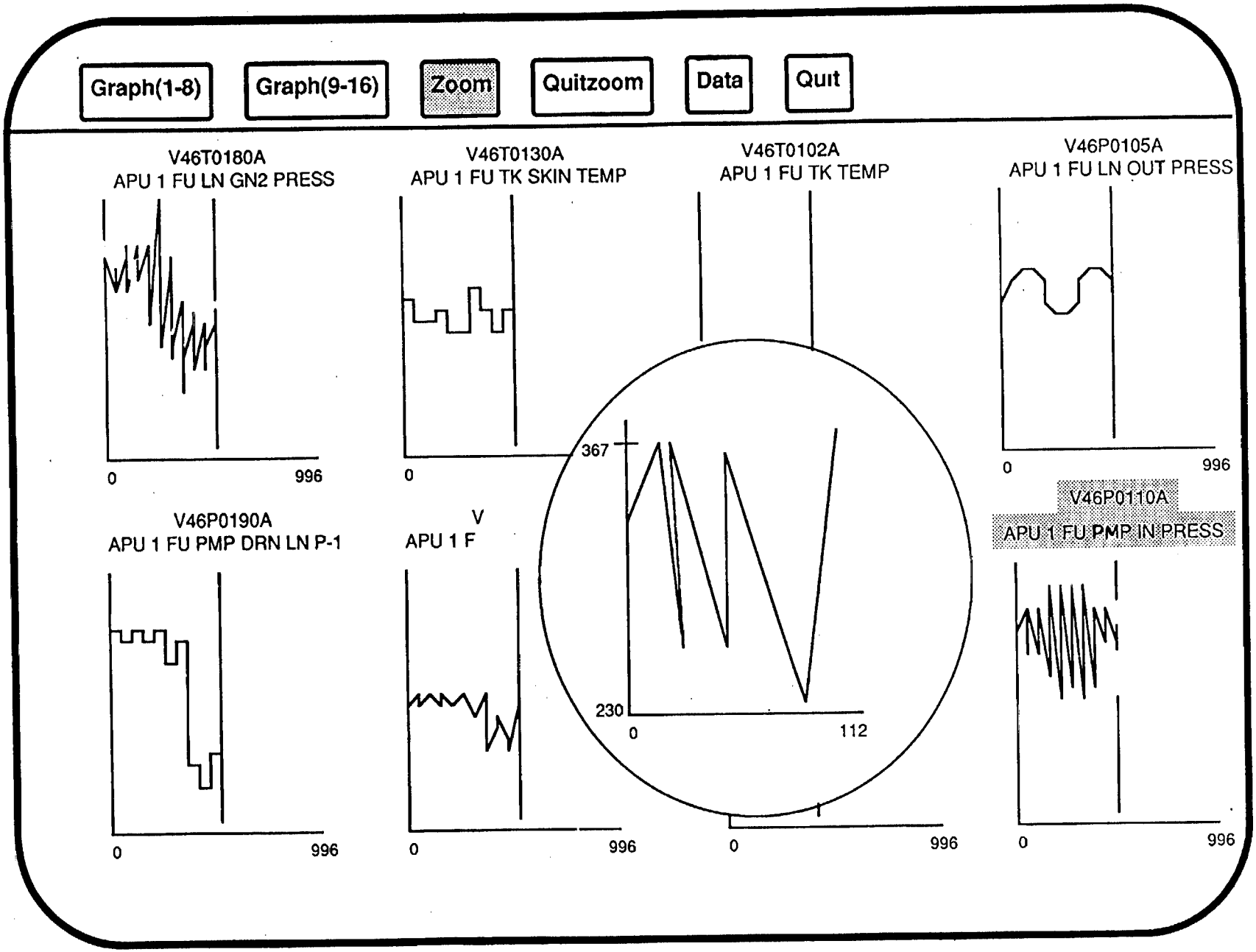
STS / APU HEALTH MONITORING

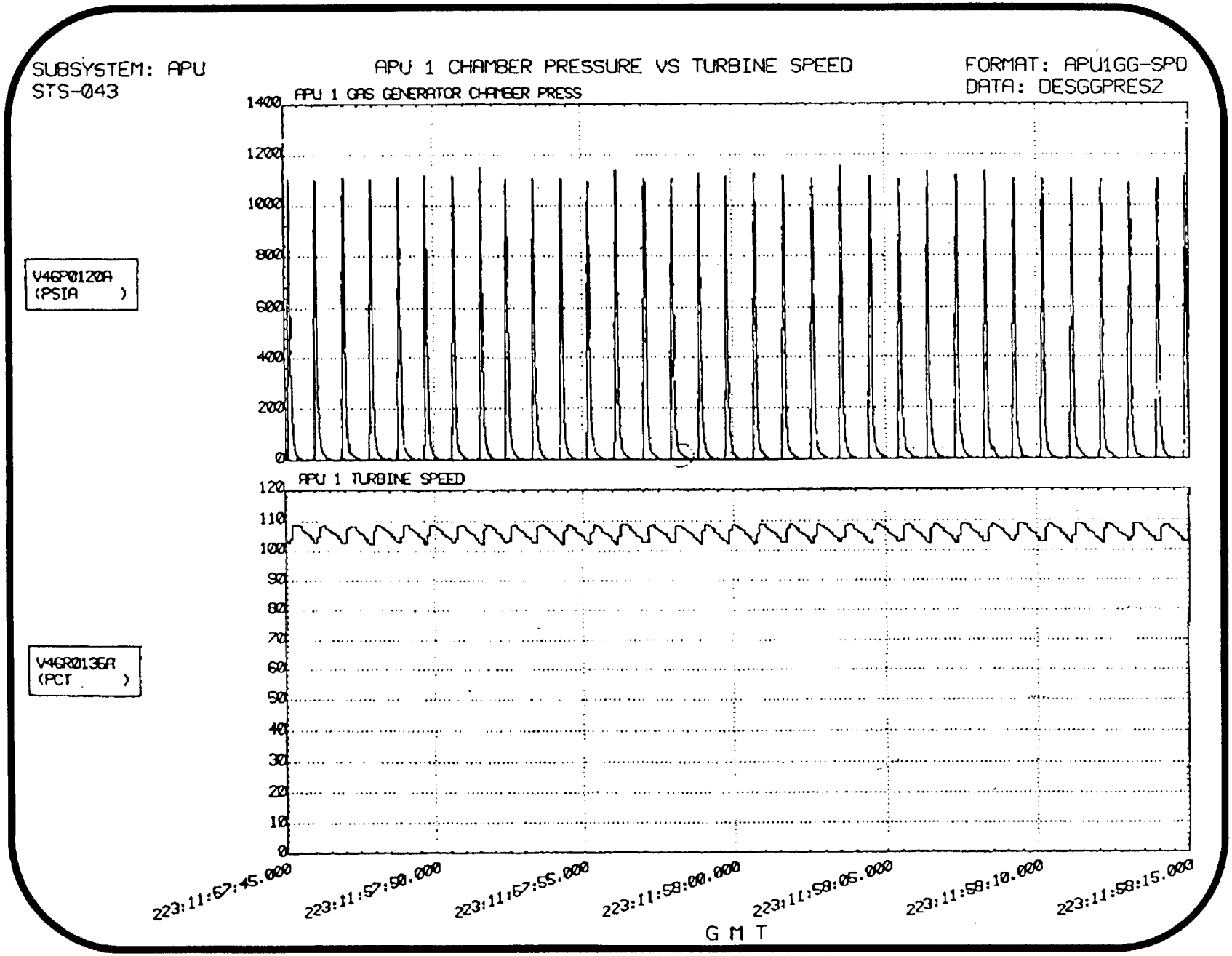
- Startup & mode-switch phases difficult to monitor due to highly complex & nonlinear nature of IAPU dynamics

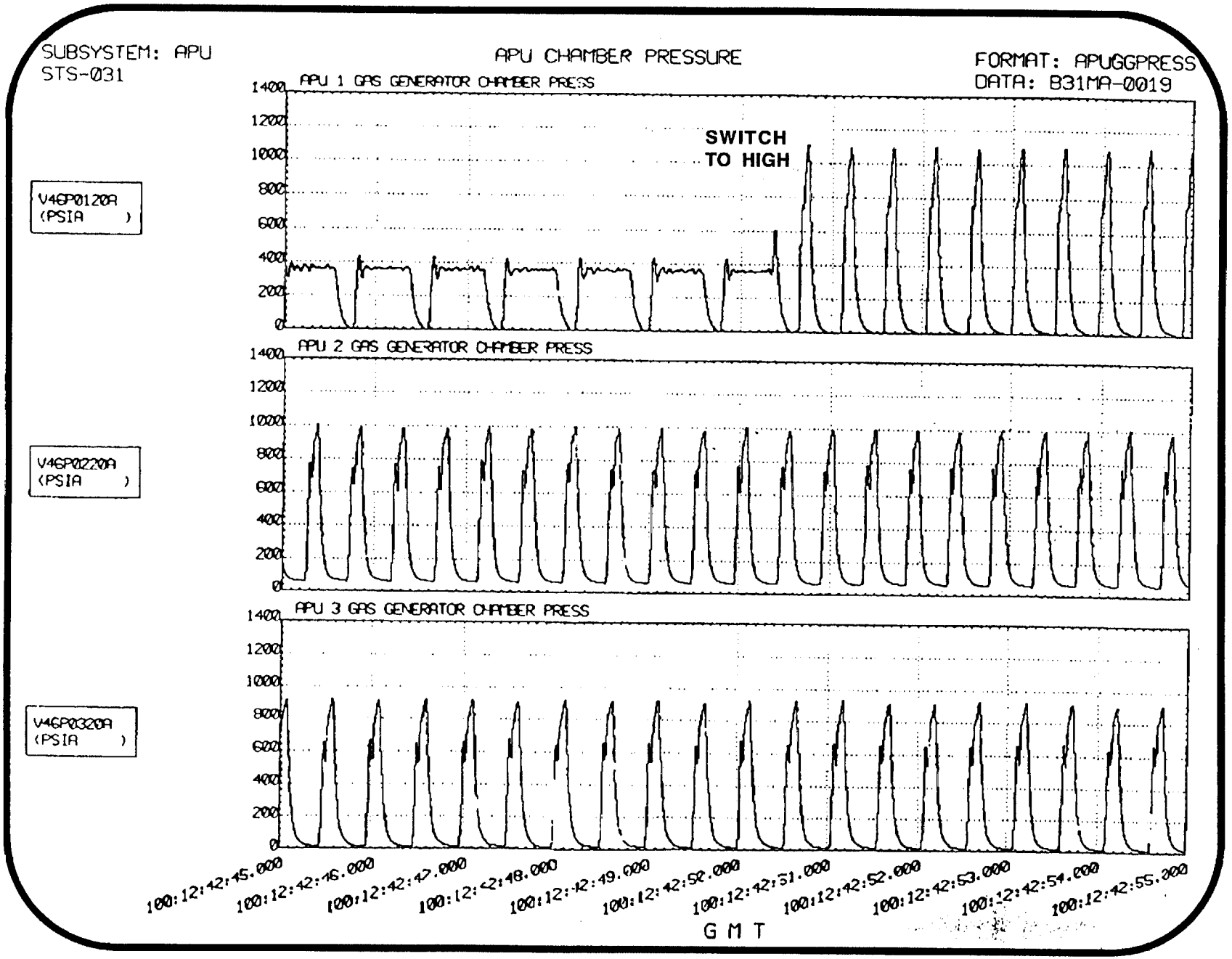
- reduced engine / test stand damage during test firings
  - typically damage 1 APU every 2 weeks

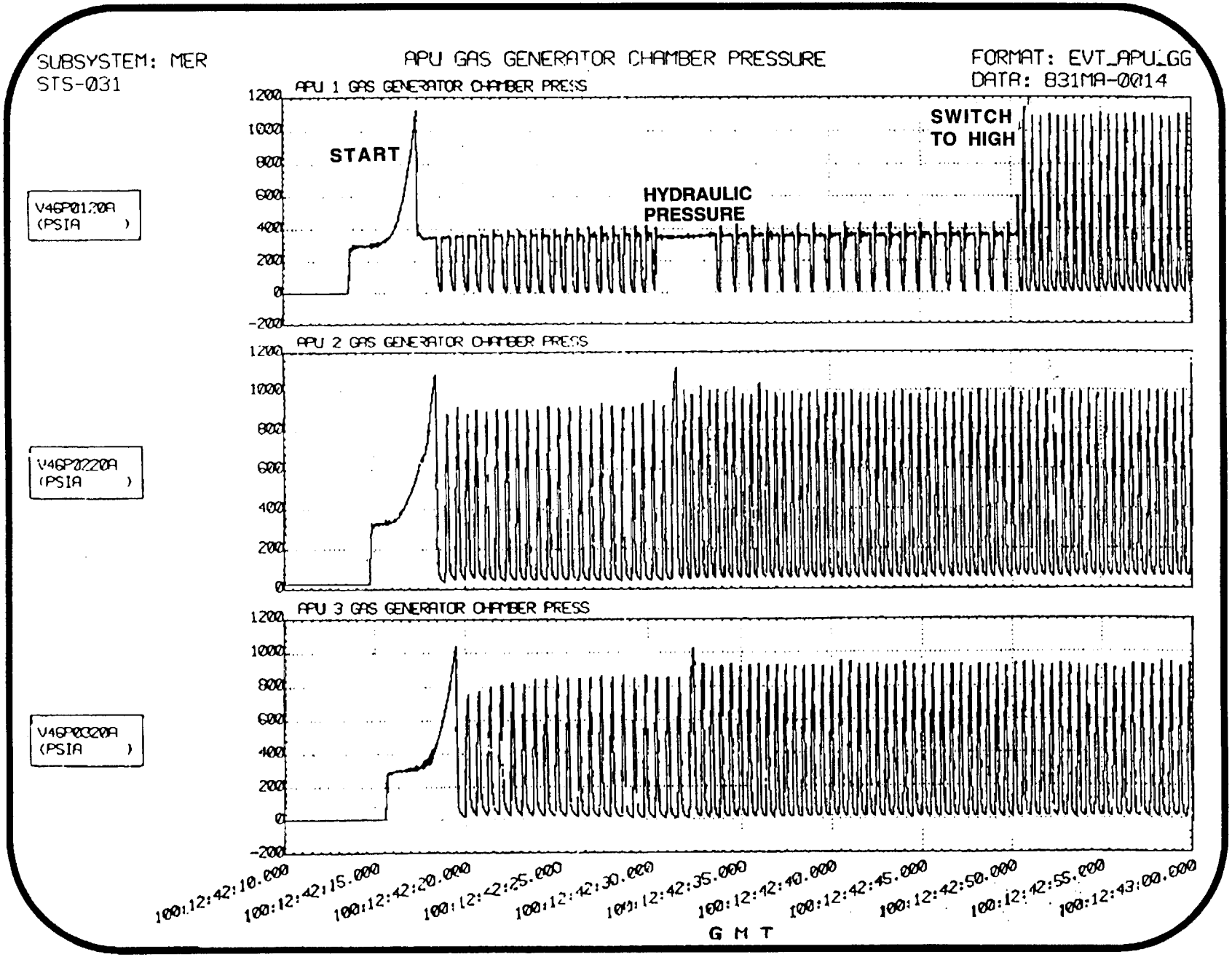
- facilitate post-test diagnostic process
  - tool for APU knowledge engineering

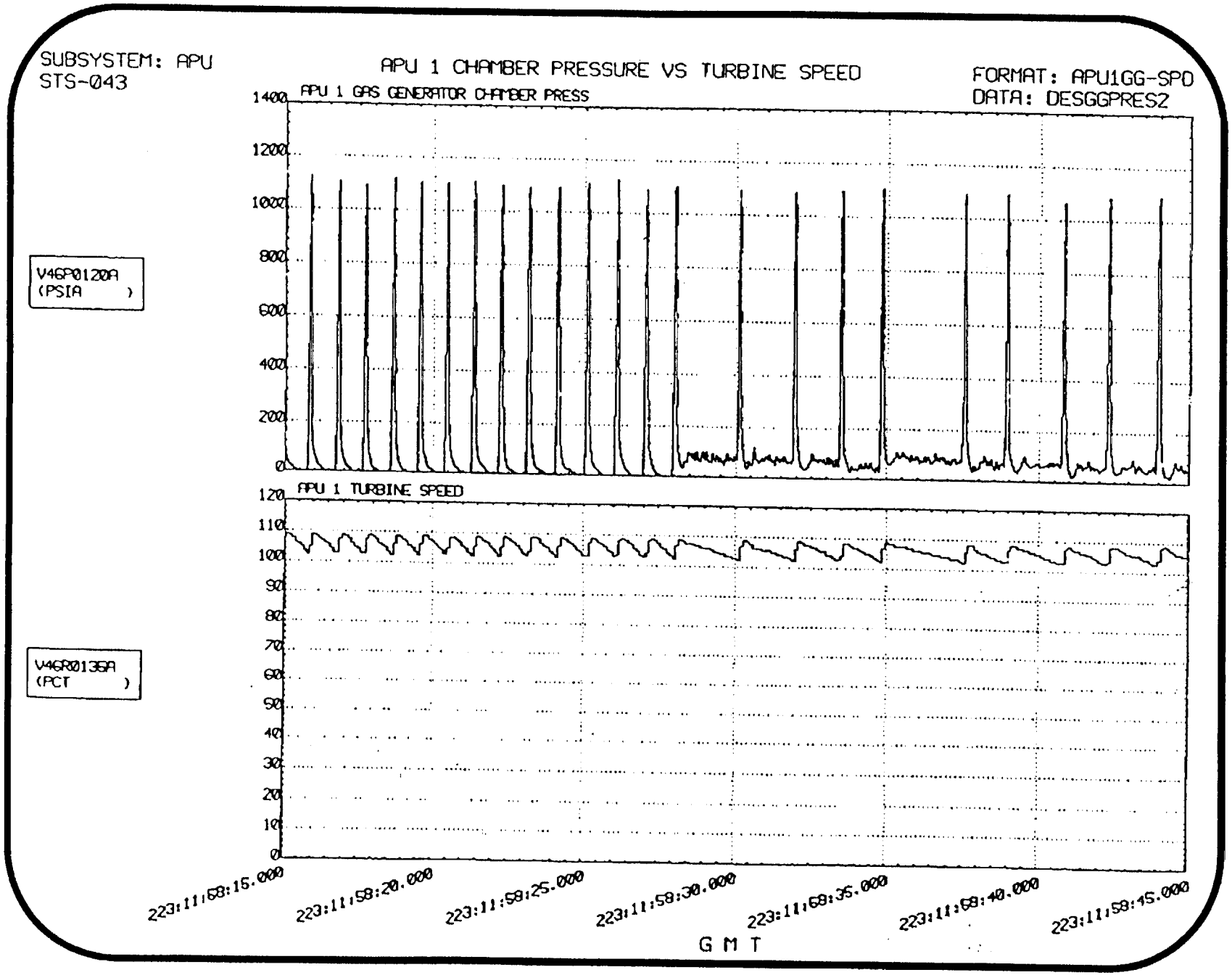
JPL


Graph(1-8)
Graph(9-16)
Zoom
Quitzoom
Data
Quit
V46T0180A
APU 1 FU LN GN2 PRESS
0
996
V46T0130A
APU 1 FU TK SKIN TEMP
0
V46T0102A
APU 1 FU TK TEMP
367
230
0
112
0
996
V46P0105A
APU 1 FU LN OUT PRESS
0
996
V46P0190A
APU 1 FU PMP DRN LN P-1
0
996
V
APU 1 F
0
996
V46P0110A
APU 1 FU PMP IN PRESS
0
996

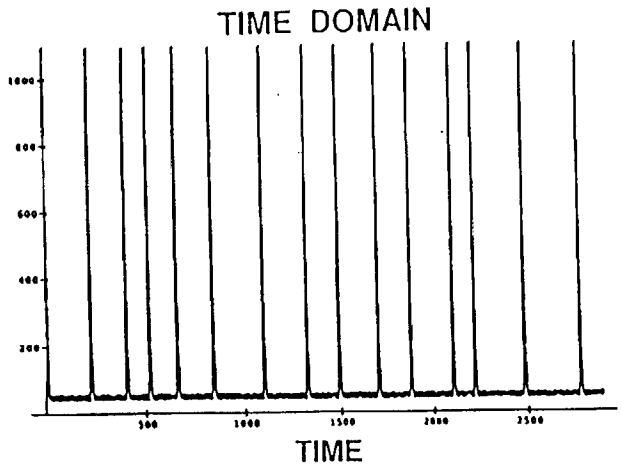

SUBSYSTEM: APU
STS-043
APU 1 CHAMBER PRESSURE VS TURBINE SPEED
FORMAT: APU1GG-SPD
DATA: DESGGPRES2
APU 1 GAS GENERATOR CHAMBER PRESS
1400
1200
1000
800
600
400
200
0
V46P0120A
(PSIA)
APU 1 TURBINE SPEED
120
110
100
90
80
70
60
50
40
30
20
10
0
V46R0136A
(PCT)
223:11:57:45.000
223:11:57:50.000
223:11:57:55.000
223:11:58:00.000
223:11:58:05.000
223:11:58:10.000
223:11:58:15.000
G M T

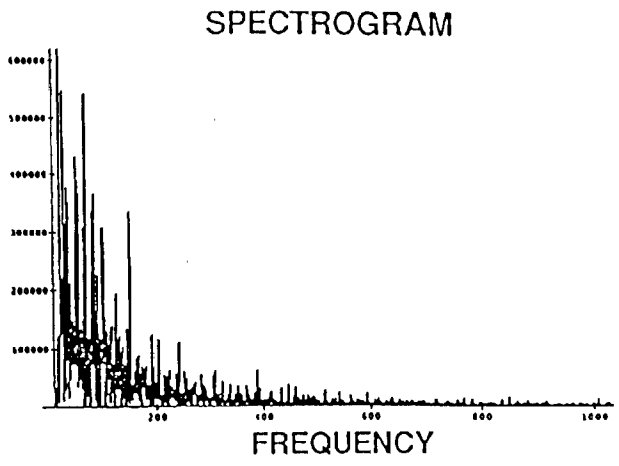

SUBSYSTEM: APU
STS-031
APU CHAMBER PRESSURE
FORMAT: APUGGPRESS
DATA: B31MA-0019
APU 1 GAS GENERATOR CHAMBER PRESS
SWITCH
TO HIGH
V46P0120A
(PSIA)
APU 2 GAS GENERATOR CHAMBER PRESS
V46P0220A
(PSIA)
APU 3 GAS GENERATOR CHAMBER PRESS
V46P0320A
(PSIA)
1400
1200
1000
800
600
400
200
0
100:12:42:45.000
100:12:42:46.000
100:12:42:47.000
100:12:42:48.000
100:12:42:49.000
100:12:42:50.000
100:12:42:51.000
100:12:42:52.000
100:12:42:53.000
100:12:42:54.000
100:12:42:55.000
G M T

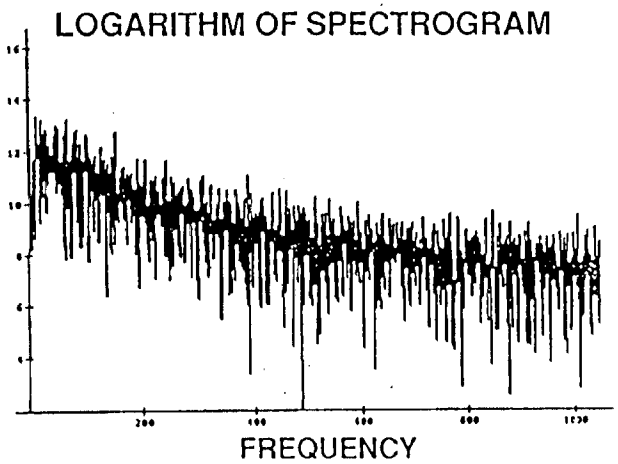

SUBSYSTEM: MER
STS-031
APU GAS GENERATOR CHAMBER PRESSURE
FORMAT: EVT_APU_GG
DATA: 831MA-0014
APU 1 GAS GENERATOR CHAMBER PRESS
APU 2 GAS GENERATOR CHAMBER PRESS
APU 3 GAS GENERATOR CHAMBER PRESS
START
HYDRAULIC PRESSURE
SWITCH TO HIGH
V46P01220A (PSIA)
V46P02220A (PSIA)
V46P03220A (PSIA)
1200
1000
800
600
400
200
0
-200
100:12:42:10.000
100:12:42:15.000
100:12:42:20.000
100:12:42:25.000
100:12:42:30.000
100:12:42:35.000
100:12:42:40.000
100:12:42:45.000
100:12:42:50.000
100:12:42:55.000
100:12:43:00.000
G M T

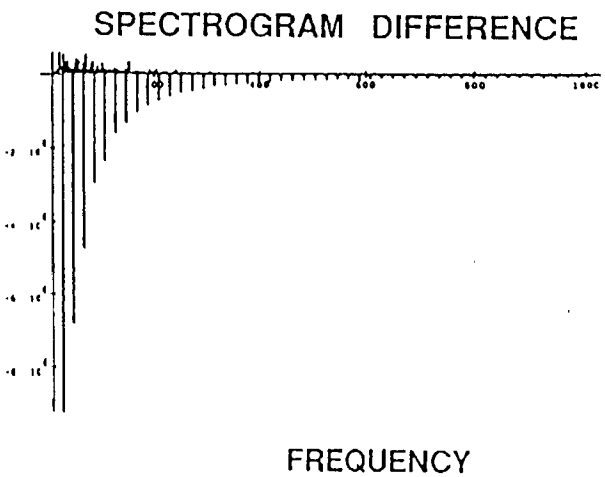

SUBSYSTEM: APU
STS-043
APU 1 CHAMBER PRESSURE VS TURBINE SPEED
FORMAT: APU1GG-SPD
DATA: DESGGPRES2
APU 1 GAS GENERATOR CHAMBER PRESS
V46P0120A
(PSIA)
APU 1 TURBINE SPEED
V46R0136A
(PCT)
223:11:58:15.000
223:11:58:20.000
223:11:58:25.000
223:11:58:30.000
223:11:58:35.000
223:11:58:40.000
223:11:58:45.000
G M T

VHM SENSOR DATA WITH CHANGING FREQUENCY AND ADDITIONAL GROUND NOISE



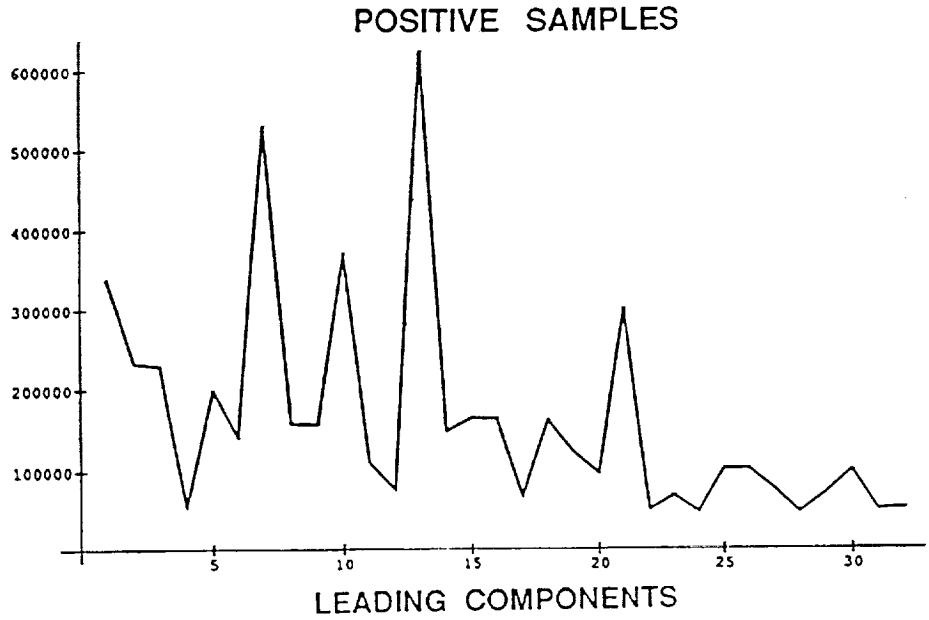


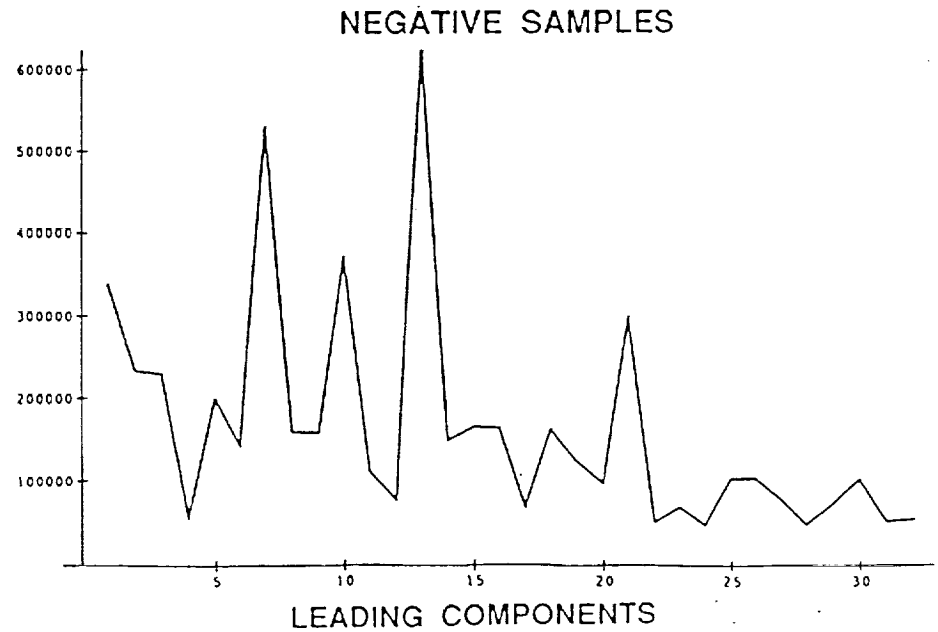




SAMPLED SPECTROGRAM DIFFERENCE

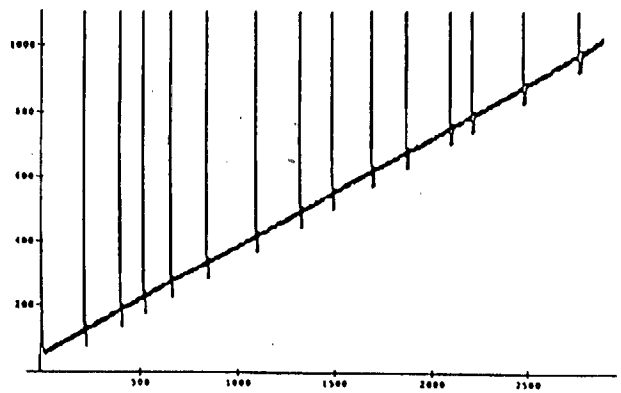
VHM SENSOR DATA WITH VARIATIONS IN FREQUENCY AND GROUND NOISE





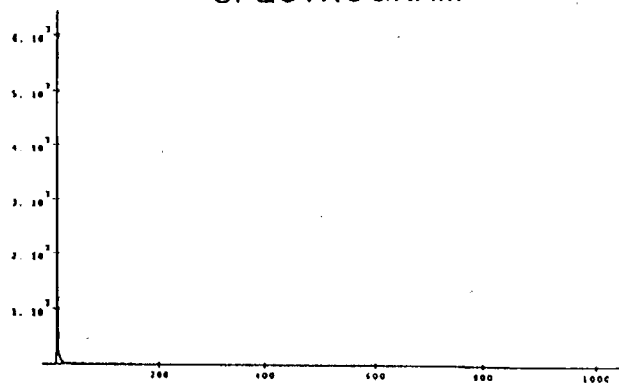
VHM SENSOR DATA WITH CHANGING FREQUENCY AND NOISE BUILDUP

TIME DOMAIN



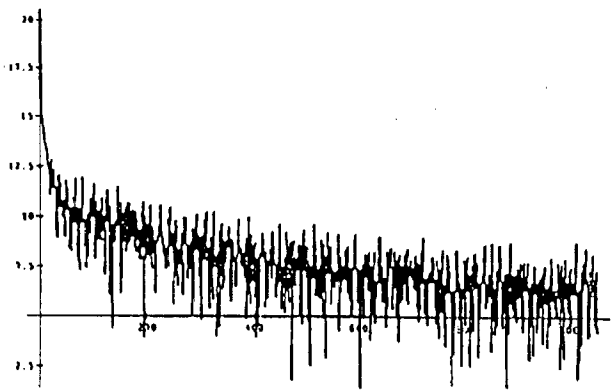
TIME

SPECTROGRAM



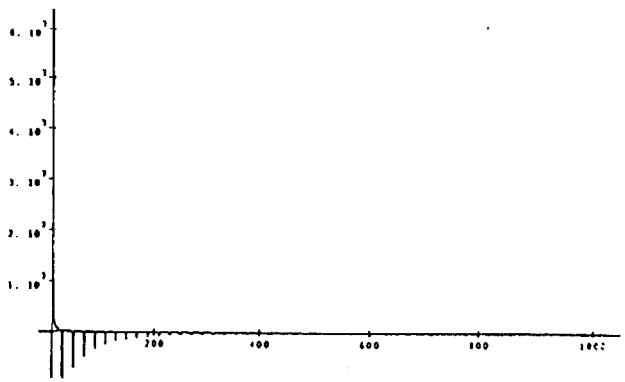
FREQUENCY

LOGARITHM OF SPECTROGRAM



FREQUENCY

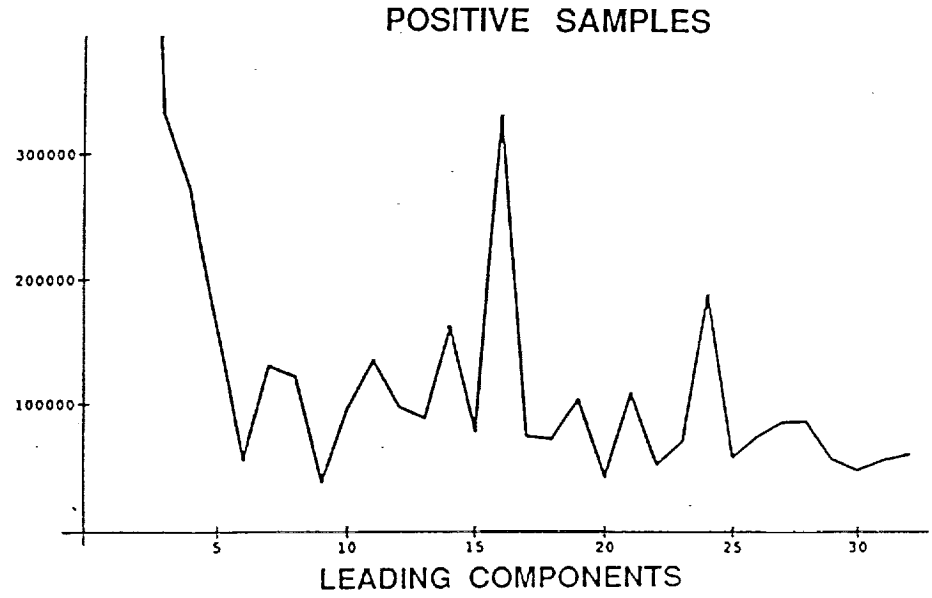
SPECTROGRAM DIFFERENCE

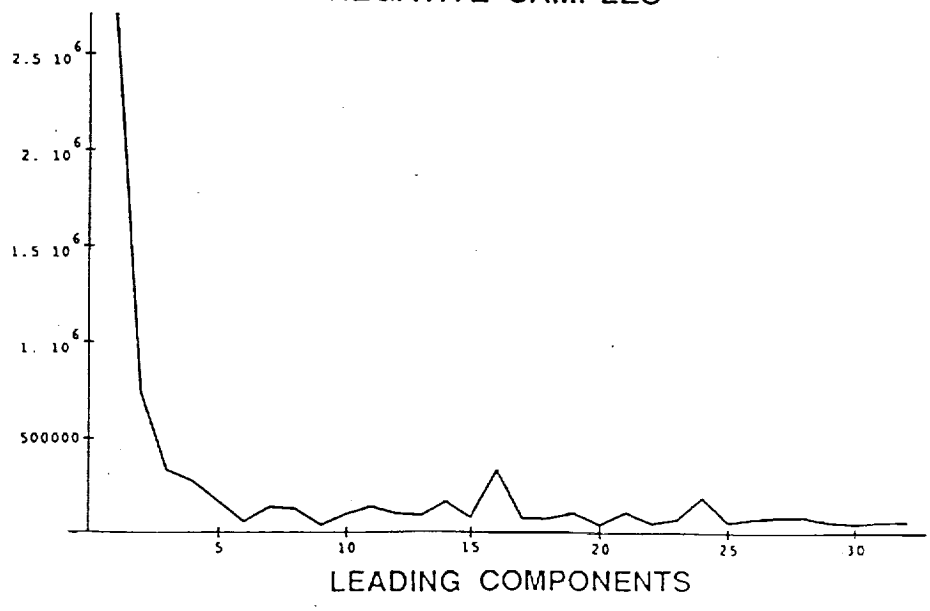


FREQUENCY

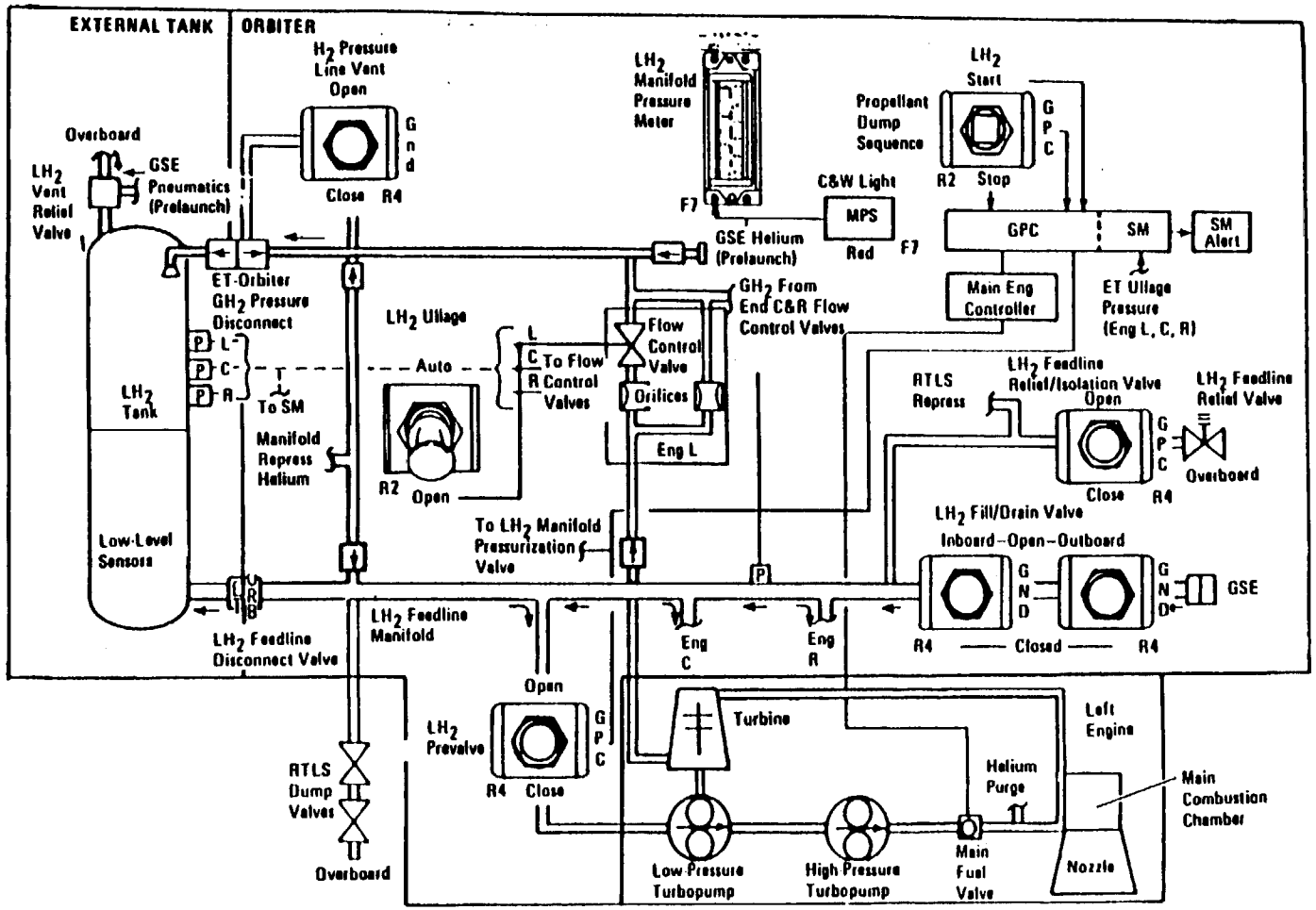
SAMPLED SPECTROGRAM DIFFERENCE

VHM SENSOR DATA WITH VARIATIONS IN FREQUENCY AND BUILDUP NOISE





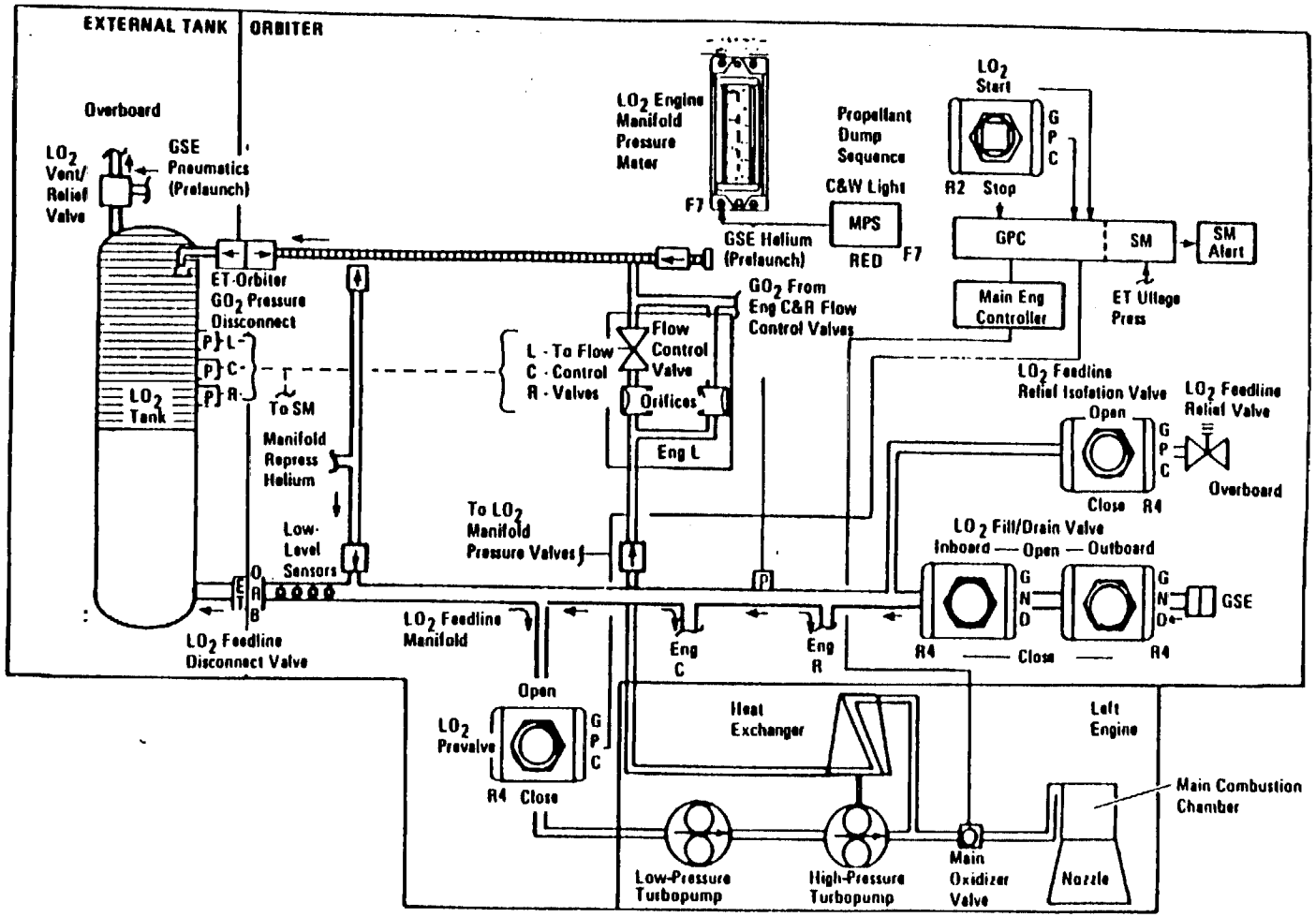
INTELLIGENT NEUROPROCESSORS FOR LAUNCH VEHICLE HEALTH MANAGEMENT SYSTEMS



TARGET HMS - STS / MPS FUEL FLOW

JPL

INTELLIGENT NEUROPROCESSORS FOR LAUNCH VEHICLE HEALTH MANAGEMENT SYSTEMS



TARGET HMS - STS / MPS OXIDIZER FLOW

JPL

JPL Workshop: "A Decade of Neural Networks: Practical Applications and Prospects" May 11th-13th, 1994

Neural Networks: Application to Medical Imaging

Laurence P. Clarke, Ph.D., FAAPM, FSNM
Professor of Radiology and Physics

College of Medicine
and
H. Lee Moffitt Cancer Center and Research Institute
University of South Florida
Tampa, FL 33612-4799

RESEARCH MISSION

•Development of computer assisted diagnostic (CAD) methods for improved diagnosis of medical images including digital x-ray sensors and tomographic imaging modalities.

•The CAD algorithms include advanced methods for adaptive nonlinear filters for image noise suppression, hybrid wavelet methods for feature segmentation and enhancement and high convergence neural networks for feature detection and VLSI implementation of NN for real time analysis. These methods are designed for fully automatic CAD methods that are operator, image and sensor independent for universal application for medical image analysis.

•Implementation of CAD methods on hospital based picture archiving computer systems (PACS) and information networks for central and remote diagnosis i.e. for cost effective health care delivery and standardization of diagnosis.

•Collaboration with defense and medical industry, NASA and Federal Laboratories in the area of dual use technology conversion from defense or aerospace to medicine .

SPECIFIC PROJECTS INVOLVING NEURAL NETWORKS

•Development of computer assisted diagnostic (CAD) methods for breast cancer screening using digital mammography. Projects include NN of different architecture tailored for each project:

1. Automatic detection of microcalcification
2. Detection of masses or parenchymal tissue distortion
3. Recognition of normal vs abnormal mammograms

•Development of nuclear medicine imaging methods for detection of beta particles used for antibody therapy or imaging of positron emitters.

1. Order statistic neural network for image resolution restoration based on systems physical response characteristics

•Development of MRI segmentation techniques using backpropagation and cascade correlation neural networks for tissue characterization.

1. Automatic segmentation of tumor volumes
2. Surgery simulation

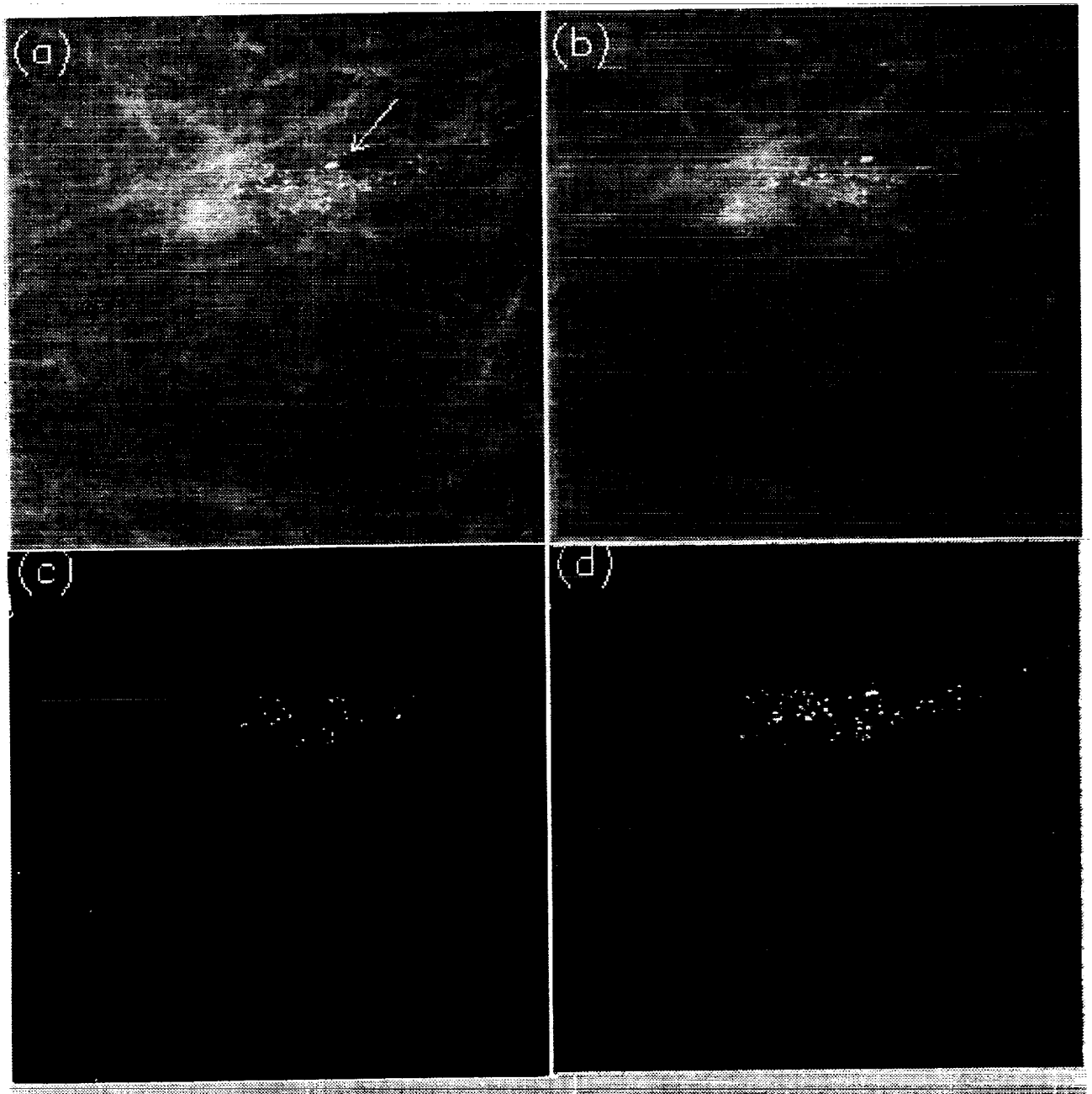


Fig. 1. (a) Section of digitized mammogram with a calcification cluster indicated by arrow.
(b) Smoothed image using the AOSF.
(c) Calcification segmentation using a two-channel TSWT.
(d) Calcification segmentation using a three-channel QMF; the morphology of the calcifications is better preserved supporting our proposed work.

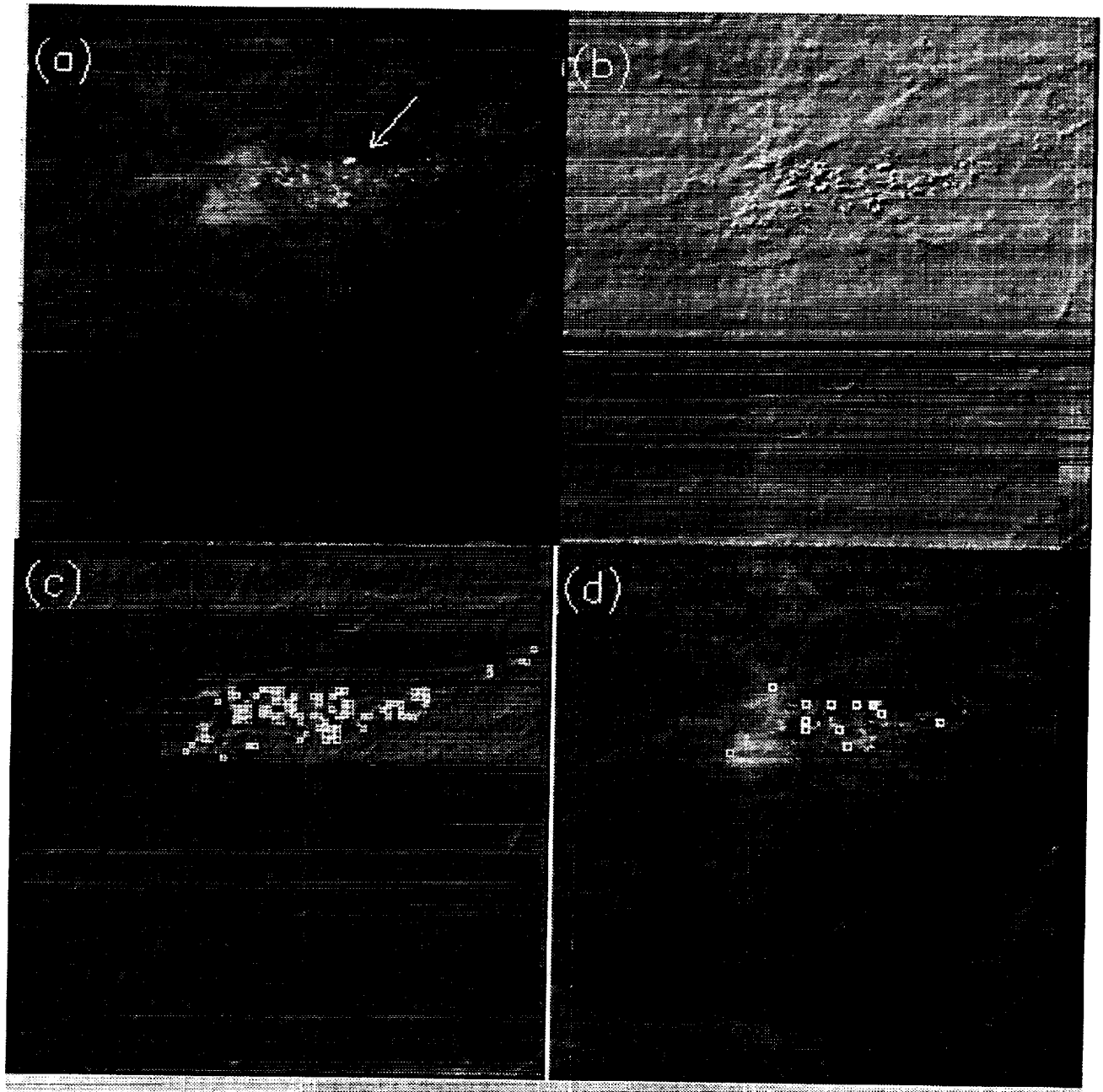


Fig. 2. (a) Section of digitized mammogram with a calcification cluster indicated by arrow
(b) Enhanced image using the AMNF–TSWT filter (linear operation); the extent of the cluster is better defined.
(c) Calcification detection indicated by squares using the enhanced image as input to the ANN.
(d) Calcification detection indicated by squares using the unprocessed image as input to the ANN; several FN detections are observed.

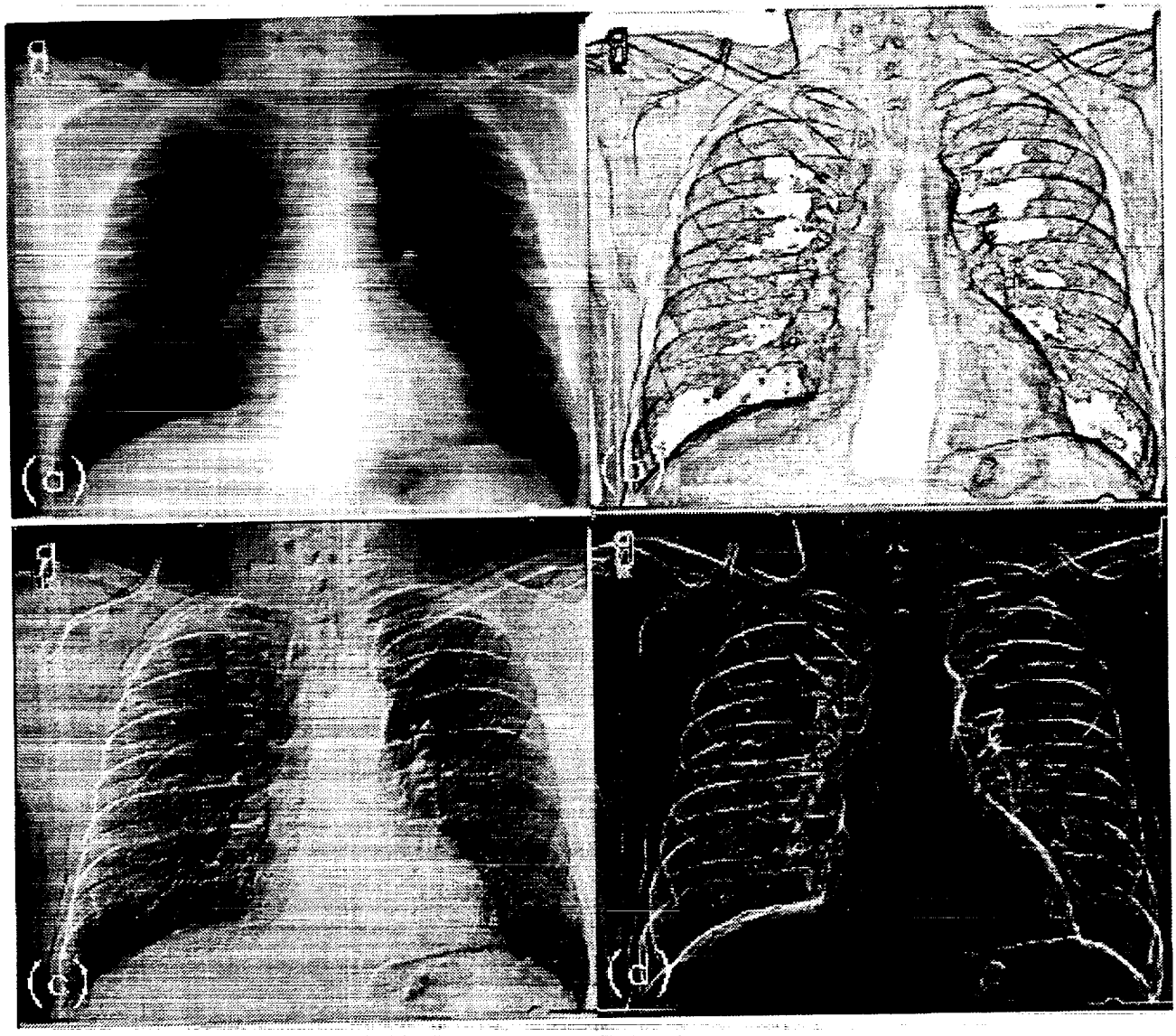


Fig. 3. (a) Digitized unprocessed chest x-ray.
(b) Enhancement by adaptive multistage nonlinear filter with an order statistic operation.
(c) Enhancement by adaptive multistage nonlinear filter with a linear operation.
(d) Processing by a tree-structured nonlinear filter and a dispersion edge detector.

CURRENT COLLABORATORS: FEDERAL/INDUSTRY

1. NASA Jet Propulsion Laboratory (JPL), Pasadena, California. Neuroprocessing and Analogue Computing Devices (NACD).
   Topic: Real time analysis of digital mammograms using VLSI implementation of NNs.

2. NASA Ames. Search for extraterrestrial intelligence (SETI), Moffett Field, California. High Resolution Microwave Survey Project.
   Topic: Detection of weak signals in digital mammograms for microcalcification and tumor detection.

3. DOD. Navy Surface Warfare Center (NSWC), Dahlgren, Virginia. Advanced Computations Technology Group.
   Topic: Pattern Recognition methods in digital mammography for identification of suspicious areas.

4. E-Systems. Garland Division, Dallas, Texas. Information Technology Systems.
   Topic: Algorithm design and real time parameter optimization in digital mammography.

5. Fischer Imaging, Denver, Colorado & Nanoptics, Gainesville, Florida.
   Topic: High resolution direct x-ray digital detection.

REFERENCES

● Hall LO, Bensaid A, Clarke LP, Velthuizen RP, Silbiger ML. A Comparison of Neural Network and Fuzzy Clustering Techniques in Segmenting Magnetic Resonance Images of the Brain. *IEEE Transactions on Neural Networks* 1992; 3(5):672-682.

● Qian W, Kallergi M and Clarke LP. Order Statistic-Neural Network Hybrid Filters for Gamma Camera Image Restoration. *IEEE Trans. Med. Imag.* 1993; 12(1):59-64.

● Clarke LP, Velthuizen RP, Phuphanich S, Silbiger ML, Schellenberg JD. MRI: Stability of Three Supervised Segmentation Techniques. *Magnetic Resonance Imaging* 1993; 11:95-106.

● Bezdek JC, Hall LO and Clarke LP. ***Invited Review***: MR Image Segmentation Techniques Using Pattern Recognition. *Medical Physics* 1993, 20(4):1033-1048.

●Lucier BJ, Kallergi M, Qian W, DeVore RA, Clark RA, Saff EB, and Clarke LP. Wavelet compression and segmentation of digital mammograms. *J Dig Imag* 1993 (to be published).

● Qian W, Clarke LP, Kallergi M and Clark R. Tree-structured nonlinear filters in digital mammography. *IEEE Trans. Med. Imag.* (accepted, to be published March 1994).

● Qian W, Clarke LP, Li HD, Kallergi M, Clark RA and Silbiger ML. Digital Mammography: M-Channel Quadrature Mirror Filters for Microcalcification Extraction. *Computerized Medical Imaging and Graphics* (accepted for publication 1993).

● Clarke LP, Blaine GJ, Doi K, Yaffe MJ, Shtern F, Brown GS, Winfield DL and Kallergi M. Digital Mammography, Cancer Screening: Factors Important for Image Compression. *Proc. of the Space and Earth Science Data Compression Workshop: NASA Conf. Publications (Invited Paper)*, Snowbird, Utah, April 2, 1993.

●Richards DW, Janesick JR, Velthuizen RP, Qian W and Clarke LP. Enhanced Detection of Normal Retinal Nerve-Fiber Striations Using a Charge-Coupled Device and Digital Filtering. *Graefe's Archive for Clinical and Experimental Ophthalmology*, 1993, 231:595-599.

●Yang Z, Kallergi M, Qian W, Clark RA, Lucier BJ, DeVore R and Clarke LP. Digital mammogram compression and processing with hyperbolic and adaptive wavelets, *IEEE TMI* (submitted Jan. 1994).

●Qian W, Clarke LP, Kallergi M. Wavelet-based neural network for multiresolution restoration of bremsstrahlung images. *IEEE Trans. Med. Imag.* (submitted November 1993).

●Clarke LP, Qian W, Kallergi M, DeVore R and Lucier B. **Invited Review:** Review of wavelet applications in medical imaging, *Med. Phys.* (to be submitted March 1994).

●Li HD, Kallergi M, Clark RA, Jain VK and Clarke LP. Markov random field model for tumor detection in digital mammography. *IEEE Trans. in Med. Imag.* (submitted for publication Jan. 1993).

●Priebe CE, Solka JL, Lorey RA, Rogers GW, Poston WL, Kallergi M, Qian W, Clarke LP and Clark RA, 1993. The application of fractal analysis to mammographic tissue classification. *Cancer Letters* (submitted).

●Clarke LP, Zheng B and Qian W. Artificial Neural Network for Pattern Recognition in Mammography. ***Invited Paper*** *by World Congress on Neural Networks* San Diego, CA, June 4-9, 1994.

●Clarke LP, Kallergi M, Qian W, Li HD, Clark RA and Silbiger ML. Tree-structured nonlinear filter and wavelet transform for microcalcification segmentation in digital mammography. *Cancer Letters* 1994 (to be published).

Learning Random Networks for Compression of Still and Moving Images

Erol Gelenbe, Mert Sungur*,
Christopher Cramer
Department of Electrical Engineering
Duke University
Durham, NC 27708-0291
E-mail: erol@ee.duke.edu
Tel: (919) 660–5442, Fax: (919) 660–5293

Summary

Image compression for both still and moving images is an extremely important area of investigation, with numerous applications to videoconferencing, interactive education, home entertainment, and potential applications to earth observation, medical imaging, digital libraries, and many other areas.

In this paper we describe our work on a neural network methodology to compress/decompress still and moving images. We use the "point-process" type neural network model we have developed [12, 13, 16] which is closer to biophysical reality than standard models, and yet is mathematically much more tractable. We currently achieve compression ratios of the order of 120 : 1 for moving grey-level images, based on a combination of motion detection and compression. The observed Signal-to-Noise-Ratio varies from values above 25 to more than 35. Our method is computationally fast so that compression and decompression can be carried out in real-time. It uses the adaptive capabilities of a set of neural networks so as to select varying compression ratios in real-time as a function of quality achieved. It also uses a motion detector which will avoid retransmitting portions of the image which have varied little from the previous frame.

Further improvements can be achieved by using on-line learning during compression, and by appropriate compensation of non-linearities in the compression/decompression scheme. We expect to go well beyond the 250 : 1 compression level for color images with good quality levels.

*Mr Sungur's work was supported by a NATO Science Fellowship at Duke University administrated by The Scientific and Technical Research Council of Turkey (TUBITAK), on leave from Department of Electrical and Electronics Engineering, Middle East Technical University, 06531 Ankara, Turkey

1 Introduction

As the volume of imaging data increases exponentially in a very wide variety of applications – including remote sensing, earth observation, medical imaging, digital libraries and documents, HDTV, entertainment and film, and videoconferencing – and as the needs for storing, retrieving and transmitting images expand, digital image compression is becoming an even more crucial technology. Many of these application areas – including earth observation, videoconferencing and many military applications – deal with sequences of images which represent some form of motion. For instance, sequences of pictures taken by a satellite each time it passes over nearly the same stretch of territory, after appropriate repositioning and compensation, are successive instances of the same scene containing changes due to the motion of objects (vehicles, for instance), or due to changing meteorological conditions. Thus compression can take great advantage of the fact that image sequences need only keep track of *changes* which occur from one frame to the next.

In some areas (such as medical imaging) it is more customary to deal with grey-level images. In other areas of application, one deals overwhelmingly with colour images (as in entertainment). The quality of a processed or compressed image is judged quite differently, whether one deals with grey-level or with colour. In the case of color, acceptable image quality will largely depend on the application. For instance, in HDTV one would be unhappy with a change in skin pigmentation (a greenish face does not look too good ...), while the change in a dress' colour may not matter too much.

Lossless compression is adequate when low compression ratios are acceptable. Very substantial compression ratios can only be achieved with *lossy* compression schemes. Many applications will accept lossy compression, as long as the resulting quality is good. In some critical applications – such as medical imaging and military observation – loss may not be tolerated. However even in those applications, compressed versions of archival images may be conveniently used for remote interrogation and fast access. The aim is of image compression is to encode images or image sequences into as few bits as possible with a decoding mechanism which reconstructs the original image with an acceptable visual and/or informational quality. Another issue in image compression and decompression is its speed, especially in real-time applications, or in those in which the rate at which the source produces data is very high. It is therefore often important to be able to carry out compression and decompression "on the fly" without additional delay in conveying the image.

In this paper we will describe a method for compressing and decompressing still and moving images. For moving image sequences of grey-level images, we obtain better than $110:1$ compression levels with 20 *to* 30 Signal to Noise Ratio (SNR). We use a learning algorithm for the "random neural network" model (Gelenbe 1989, 1990, 1993 [12, 13, 16] [1]) to "teach" a set of networks to compress at different compression levels. A schematic representation of the complete method we propose is shown in Figure 1. The method uses a simple motion detection scheme, together with the set of learning neural networks for compression and decompression.

In the sequel we first describe the problem, then review the literature, after which we describe our method together with measurements describing the resulting compression levels, the SNR of reconstructed images. We also provide an indication of the data transmission rates for the schemes we develop. This last metric is particularly relevant when images are transferred over networks, since the nature of the traffic determines the performance levels which can be expected and the appropriate traffic controls which may have to be imposed.

[1] This model has also been successfully applied to other applications including optimization [15] and image texture analysis and reconstruction [3, 4].

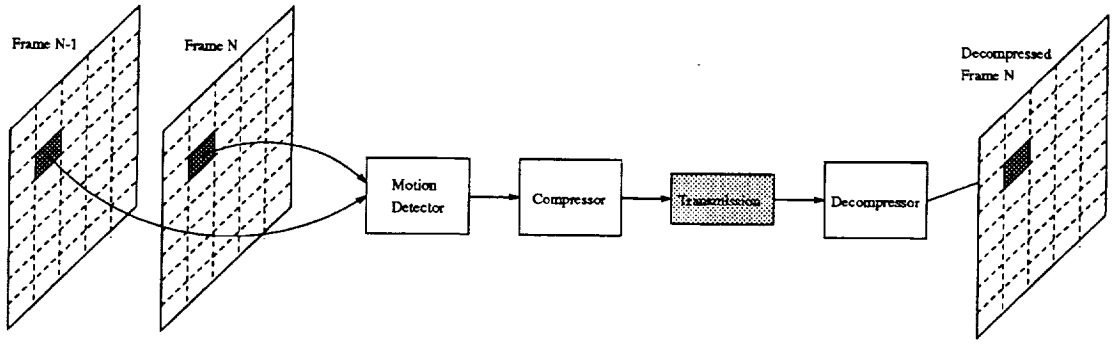


Figure 1: Block diagram of the complete compression scheme.

1.1 The Image Compression Problem

A *digital image* I is described by a function $f : Z \times Z \rightarrow \{0, 1, \ldots, 2^k - 1\}$ where Z is the set of natural numbers, and k is the maximum number of bits to be used to represent the gray level of each pixel. In other words, f is a mapping from discrete spatial coordinates (x, y) to gray level values. Thus, $M \times N \times k$ bits are required to store an $M \times N$ digital image. The aim of digital image compression is to develop a scheme to encode the original image I into *the fewest* number of bits such that the image I' reconstructed from this reduced representation through the decoding process is as similar to the original image as possible: *i.e.* the problem is to design a COMPRESS and a DECOMPRESS block so that $I \sim I'$ and $|I_c| << |I|$ where $|.|$ denotes the size in bits (Figure 2).

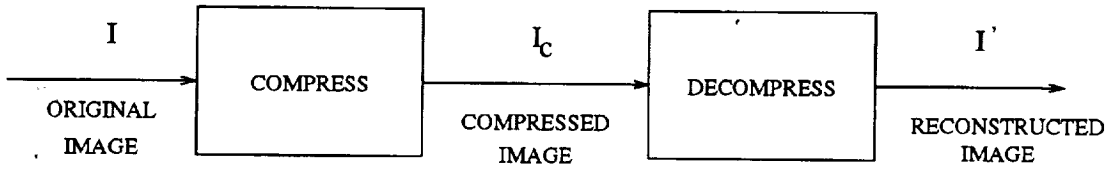


Figure 2: Image Compression Block Diagram

The similarity measure can vary for each application. Some applications may require the reconstructed image to be exactly the same as the original image, in which case the process is called *lossless compression*. In *lossy compression*, the peak signal-to-noise ratio or SNR is used as the measure of similarity or of dissimilarity, although it does not necessarily reflect visual quality. Assuming that the original and reconstructed images are represented by functions $f(x, y)$ and $g(x, y)$ of the pixel plane position (x, y), respectively, the SNR is defined by:

$$\text{SNR} = 10 \log_{10} \frac{(2^k - 1)^2}{e^2_{\text{rms}}} \tag{1}$$

where the root-means-square error is

$$e^2_{\text{rms}} = \overline{e^2} = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\, g(x, y) - f(x, y) \,]^2 \tag{2}$$

When moving images are concerned, the compression ratio may vary dynamically with the specific image or image portion being transmitted, since some advantage will be taken of the existence or non-existence of significant motion in successive image frames. However the SNR metric will still be relevant to the evaluation of the resulting quality.

1.2 State-of-the-Art in Still and Moving Image compression

Image compression research generally addresses the basic trade-off between the reconstruction quality of the compressed image, the compression ratio, and the complexity and speed of the compression algorithm. The two currently accepted standards for still and moving image compression are JPEG ([34]) and MPEG ([25]). These schemes provide high compression ratios with good picture reconstruction qualities. However, the amount of computation required for both is generally too high for real-time applications. MPEG uses the following techniques: 1) RGB color space coding to YCrCb coding, this gives and automatic 2:1 compression ratio, 2) JPEG encoding based on discrete cosine transform and quantization followed by some lossless compression, which yields compression ratios as high as 30:1 with good image quality, and 3) Motion Compensation, in which a frame can be encoded in terms of the previous and next frames. However, these techniques severely limit the speed at which a sequence of images can be compressed.

Two classical techniques for still image compression are transform and sub-band encoding. In transform coding techniques the image is subdivided into small block each of which undergoes some reversible linear transformation (Fourier, Hadamard, Karhunen-Loeve, etc.) followed by quantization and coding based on reducing redundant information in the transformed domain. In subband coding ([35]), an image is filtered to create a set of images, each of which contains a limited range of spatial frequencies. These so-called subbands are then downsampled, quantized and coded. These techniques require much computation. Another common image compression method is vector quantization ([18]) which can achieve high compression ratios. A vector quantizer is a system for mapping a stream of analog or very high rate or volume discrete data into a sequence of low volume and rate data suitable for storage in mass memory, and communication over a digital channel. This technique mainly suffers from edge degradation and high computational complexity. Although some more sophisticated vector quantization schemes have been proposed to reduce edge effects ([30]), the computation overhead still exists. Recently, novel approaches have been introduced based on pyramidal structures [1], wavelet transforms [36], and fractal transforms [20]. These and some other new techniques [24] inspired by the representation of visual information in the brain, can achieve high compression ratios with good visual quality but are nevertheless computationally intensive.

The speed of compression/decompression is a major issue in applications such as videoconferencing, HDTV applications, videophones, which are all likely to be a part of daily life in the near future. Artificial neural networks [31] are being widely used as alternative computational tools in many applications. This popularity is mainly due to the inherently parallel structure of these networks and to their learning capabilities which can be effectively used for image compression.

Several researchers have used the Learning Vector Quantization (LVQ) network [23] for developing codebooks whose distribution of codewords approximates the probabilistic distribution of data which is to be presented. A Hopfield network for vector quantization which achieves compression of less than 4:1 is reported in [27]. A Kohonen net method for codebook compression is demonstrated in [29]; it seems to perform slightly better than another standard method of generating codebooks. Cottrell et al. ([8]) train a two-layer perceptron with a small of number of hidden units to encode and decode images, but do not report encouraging results about the performance of the network on previously unseen images. Using neural encoder/decoders has been suggested by many researchers such as [6]. In [10], the authors present a neural network method for finding coefficients of a 2-D Gabor transform. This 2-way function can then be quantized and encoded to give good images at compression of under 1 bit/pixel, and as low as 0.38 bits/pixel with good image quality in a particular case.

A feed-forward neural network model to achieve 16 : 1 compression of untrained images with $SNR = 26.9dB$ is presented in [26] by using four different networks to encode different "types" of images. A backpropagation network to compress data at the hidden layer and an implementation on a 512 processor $NCUBE$ are discussed in [32]. In [19], the authors perform a comparison of backpropagation networks with recirculation networks and the DCT (discrete cosine transform). The best results reported here are obtained with the DCT, then with recirculation networks and finally with backpropagation networks. An interesting feature

of this paper is that they show the basis images for the neural networks, which allows one to compare the underlying matrix transformations of the neural networks to that of the DCT. In [11], the authors present a VLSI implementation of a neuro vector quantization/codebook algorithm. In [28], the authors use a back-propagation based nested training algorithm to do compression. For images on which the network has already been trained (which is not specifically of practical use) the compression ratios and resulting qualities are as follows: 8:1 (SNR = 22.89dB), 64:1 (SNR=15.15dB) to 256:1 (SNR=10.44dB). For previously "unseen" images, results are given with the following ratios and qualities: 8:1 (SNR=18.13dB) to 64:1 (SNR=12.93dB). Our own results for "unseen" images provide substantially better quality, especially at the lower compression ratios (8:1 and 16:1). In [22], the authors suggest the use of a non-linear mapping function whose parameters are learned in order to achieve better image compression in a standard backpropagation network.

Motion detection and compensation are key issues when one deals with moving images. Motion compensation provides for a great deal of the compression in the MPEG standard. By using motion compensation, MPEG can code the blocks in a frame in terms of motion vectors for the blocks in the previous and/or next frames. To perform motion must be estimated using block matching over the area local to the block under consideration. Exhaustive searches which consider all possible motion vectors yield good results. However for large ranges, the cost of such a search becomes prohibitive and heuristic searches must be used. This also raises the problem that full motion compensation cannot be performed in real time since it requires the future frame to be known in advance. Partial motion compensation, in which blocks may be encoded only in terms of blocks in the previous frame, may be used. One should also note that the MPEG standard does not specify the method of motion compensation to be used and a neural solution to motion compensation problem in two dimensions has been examined. In [9], a neural network for motion detection is presented; however it only works for a one dimensional case and the authors state that problems arise when the approach is extended to two dimensional detection of edge motion. It appears this approach would involve a great deal of research before it could be usefully applied in moving picture compression. In [7], a neural network method for motion estimation is presented. Drawbacks include the assumption that displacement is uniform in the area of interest. This would be a problem in trying to estimate the motion of a human being in which motion vectors differ over subsets of the picture.

2 Still Image Compression with the Random Neural Network

One of the common neural approaches in image compression is to train a network to encode and decode the input data [8], so that the resulting difference between input and output images is minimized. The network consists of an input layer and an output layer of equal sizes, with an intermediate layer of smaller size in between. The ratio of the size of the input layer to the size of the intermediate layer is – of course – the compression ratio. More generally, there can also be several intermediate layers. The network is usually trained on one or more images so that it develops an internal representation corresponding not to the image itself, but rather to the relevant features of a class of images.

In our approach, both the input, intermediate and output image is subdivided into equal-sized blocks and compression is carried block by block (see Figure 3). This has the desirable effect of reducing the network learning time. It also achieves good generalization, since the blocks comprising a single test image are used as the training set. The amount of information representing the compression and decompression algorithm (i.e. the "weights") is also substantially reduced in this manner. We use a feedforward encoder/decoder random neural network with one intermediate layer as shown in Figure 8. The weights between the input layer and the intermediate layer correspond to the encoding or *compression* process, while the weights from the intermediate to the output layer correspond to the decoding or *decompression* process.

Our current results use 8×8 boxes, where each element is a byte. We encode the 8-bit gray level values as real numbers between 0 and 1, *i.e.* we map the $[0, 255]$ interval into the $[0, 1]$ interval since the grey level of each image pixel is transformed into a real-valued excitation level of a neuron (and vice-versa). The network

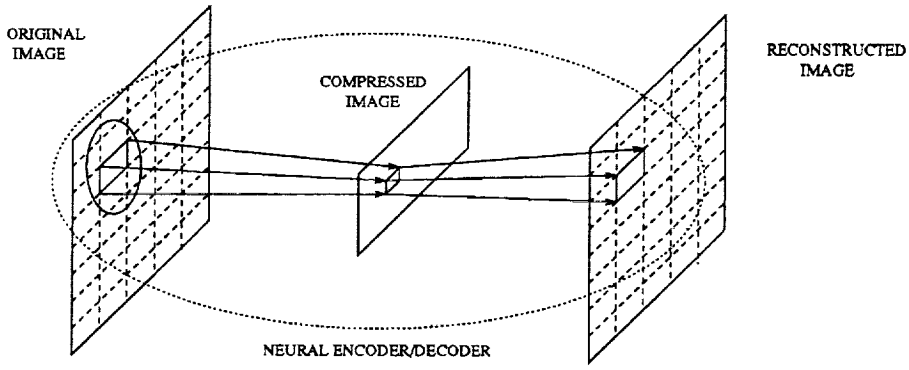
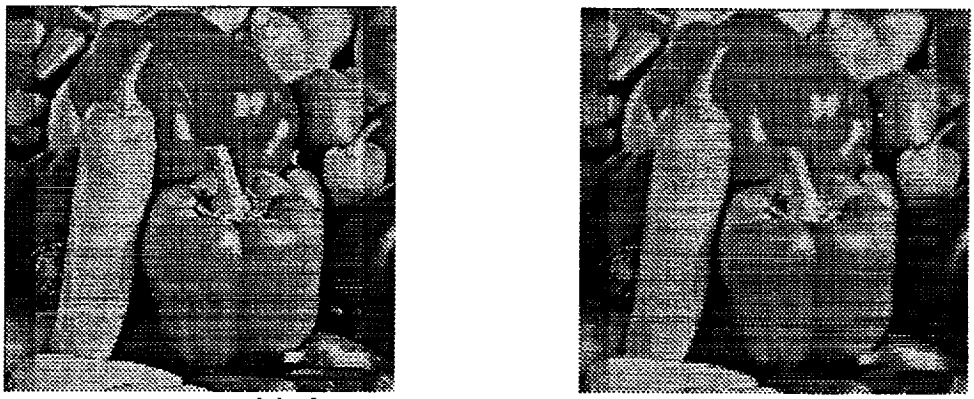


Figure 3: Compression of an arbitrarily large image using a neural encoder/decoder

is trained so as to minimize the squared error between the output and input values, thus maximizing the SNR, with the proviso that the image SNR is measured for quantized values in $[0, 255]$ while the neural network learning uses the corresponding real-valued network parameters. In all the results we report, both in this section and when we deal with moving images, our networks are trained using the algorithm described in [16] using a single image: the well-known 512×512 8-bit *Lena*. Indeed, we have found that *Lena* provides some of the best results for training the network. The network is then tested for a variety of images, and we have observed a reconstruction quality ranging from $SNR = 23dB$ to more than $30dB$ for $16:1$ compression (*i.e.* 0.5 bits/pixel). As an example, Figure 4 shows our results with $16:1$ compression for the 512×512 8-bit *Peppers* image [17].



PEPPERS original SNR = 27.82

Figure 4: Test results for $16:1$ compression (0.5 bit/pixel) with random neural network

2.1 Motion Detection

In many applications such as *videoconferencing*, sequences of image frames representing a moving scene are transmitted. Often, a substantial part of an image, such as the background, basically does not move – except for noise which may originate at various levels, including the imaging devices. On the other hand, the objects in the image move relative to the background, but this displacement be quite small between any two successive frames. We use these facts in order to perform motion detection. Specifically we examine the 8×8 boxes from successive frames F_{i-1}, F_i. Motion is sensed if the average grayscale value of a box in F_i differs from that of the corresponding box in frame F_{i-1} by more than a certain amount d. We have observed experimentally that the difference in the average grayscale value of a block that is perceptable to the human

eye is around around $d = 1$. Note that the box structure used throughout our compression scheme makes this approach possible as long as the box size is small enough. Indeed, a large box size would either make it highly improbable that motion has not occurred within any given box, or would render the detection process insensitive if accompanied by a large value of d.

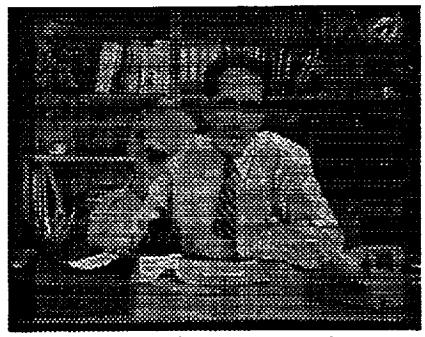
We use the first 101 frames of gray-level image sequences, *Miss America* and *Salesman*, to test our motion detector. Each frame is of size 360 x 288 yielding 1620 8 x 8 boxes. To test the motion detector, we load the first two frames into two arrays. Array 1 contains the frame which is on the screen at the receiving end of the transmission, while Array 2 is the new frame. Each 8 x 8 box in the frames is tested for motion detection. If a box is classified as unchanged, the box in Array 1 is replaced by the box in Array. Once all of the boxes are tested, the next frame is loaded into Array 2, and the process is repeated. Clearly, the parameter d will influence both the compression ratios and the resulting image quality. In order to illustrate its effect on compression we have run a series of tests summarized on Table 1. In the tabulated information note that the "Total Compression Ratio" is derived from the size of the whole video sequence after motion detection, whereas the "Steady State Compression Ratio" is the average compression ratio due to motion detection over all the frames *after* the complete first frame has been transmitted. Both values do *include* the overhead due to the additional bits sent for each box of each frame: two bytes to indicate x and y indices of the block in that frame. For storage applications, a simpler and possibly more efficient scheme with one bit per block can be used: a bit value of "1" means that motion is detected in the box and that it be sent, while "0" means that the box will not be sent (and therefore that the previous frame's corresponding box should be used). However, considering network applications, we will prefer the former header so that the image transmission will not be sensitive to packet losses.

d	Miss America				Salesman			
	Compression Ratio		Frame SNR		Compression Ratio		Frame SNR	
	Total	Steady State	Min	Max	Total	Steady State	Min	Max
0.5	2.25	2.28	38.78	40.83	3.01	3.07	37.38	44.15
1.0	4.44	4.59	36.81	39.51	6.55	6.94	35.04	43.42
1.5	6.06	6.38	35.72	38.07	9.23	10.06	33.66	42.59
2.0	7.25	7.74	34.57	37.48	11.26	12.55	32.77	41.94
2.5	8.42	9.10	33.91	36.92	13.08	14.88	31.99	41.71
3.0	9.53	10.41	33.63	36.68	14.70	17.04	31.41	41.81
3.5	10.60	11.73	33.02	36.43	16.32	19.29	30.84	41.28
4.0	11.71	13.11	32.69	36.23	18.01	21.71	30.60	41.05
4.5	12.82	14.54	32.37	35.80	19.75	24.30	30.05	40.50
5.0	13.96	16.04	32.08	35.55	21.38	26.86	29.77	40.12

Table 1: Compression ratios obtained *only* by motion detection: as a function of difference threshold d

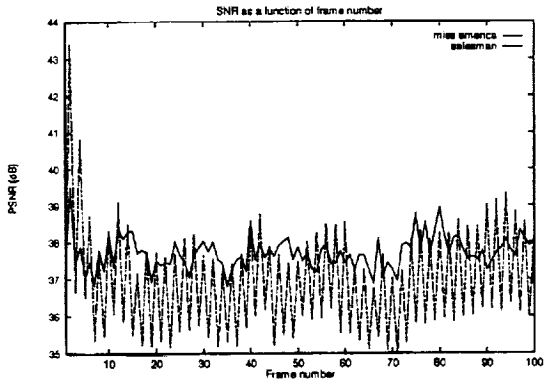
Other results are presented in the form of the actual images before and after motion detection. Figure 5 shows the original and the reconstructed 101st –and last– frame of the sequence with $d = 1$. In Figure 6(a), the SNR is plotted as a function of frame number for $d = 1$. Similarly Figure 6(b) shows the number of bits transmitted as a function of frame number. ¿From these results and other experiments we have run, it appears that a compression ratio of 6 or 7 can be obtained easily with a value of d close to or slightly above 1, with satisfactory image quality, when only motion detection is used for compression. In the next section this scheme will be combined with the actual neural compression of frames in order to achieve high compression ratios and satisfactory image quality.

Original 101st frame

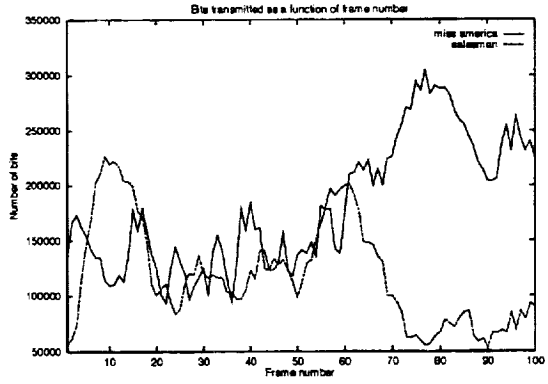


Reconstructed (SNR = 38.21)

Figure 5: Original and reconstructed last frames (101st frames) in the SALESMAN sequence using the motion detection scheme with $d = 1$



(a)



(b)

Figure 6: Experimental results for motion detection with $d = 1$: a) PSNR as a function of frame number, b) Number of bits transmitted as a function of frame number

3 Compression for Moving Images

In this section we will describe and evaluate the complete compression scheme for video sequences of natural images, using a combination of the motion detection scheme described earlier together with our adaptive still block-by-block (Figure 3) random neural network compression/decompression. Specifically, our compression scheme uses three networks:

- The first network scans successive boxes (fixed size portions of the image) in sequence, and identifies those boxes where motion has taken place, as described above. If a box is considered to be identical to the same box in the previous frame, it is not compressed or transmitted.
- The second network carries out compression of the box which is identified by the first network. In fact the second network is a set of distinct neural compression networks C_1, ... , C_L which are designed to achieve different compression levels. Each of these networks compresses the box in parallel. The choice of the compression level to be selected is carried out by the third network.
- The third network simulates the decompression, and provides a measure of the "quality" of the compression-decompression. In fact it is composed of L distinct decompression networks D_1, ... , D_L, where D_i matches C_i.

Then the pair C_i,D_i which yields the highest compression ratio at a quality level of Q or better, chosen to be acceptable for the particular application, is selected and the compressed box is transmitted. For grey-level images Q is formulated as a SNR value. Figure 7 shows the block diagram of the adaptive still image compression network. Note that with the exception of the initial learning phase, all the operations which have been outlined above can be carried out "on-the-fly", i.e. in real-time as each box goes through the transmitter, and as each compressed box goes through the receiver. (See Figure 1 for a block diagram of the total proposed scheme).

Another refinement would be to use the network D_i (which is stored both at the transmitting end and at the receiving end) to further train the network C_i in on-line mode. In this case, D_i's weights will *not* be changed, and only C_i's weights are updated.

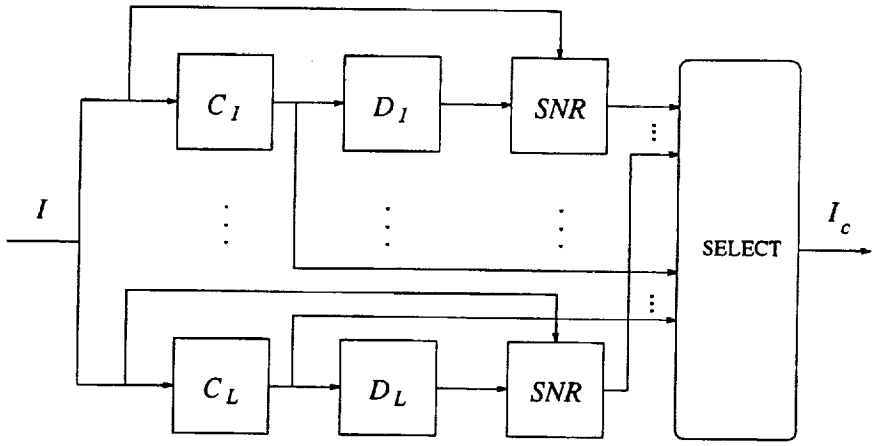


Figure 7: Block diagram of the adaptive still image compression network

At the *"receiving or decompression"* end, if the transmitter has sent a 0 bit to indicate that the current box is identical to the same box in the previous frame, then the previous frame's box is placed in the corresponding position of the output image. Otherwise the compressed box is received. Implicitly (through the box's size) or explicitly (via some variable i which would accompany the box) the compression level used is known to the receiver. We then use the network D_i to decompress the box, which is subsequently placed in appropriate sequence into the output image. The relationship between any two compression/decompression networks C_i, D_i is shown in (Figure 8).

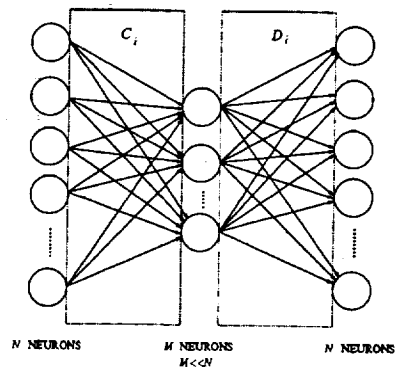


Figure 8: A Neural Network Compression/Decompression Pair

3.1 Experimental Results for Moving Image Compression

We have experimented the combined scheme with three still image compression machines ($L = 3$ with $8:1$, $16:1$ and $32:1$ compression/decompression pairs), and have tested it on the 101-frame *Miss America* and *Salesman* grey-level image sequences. Table 2 summarizes the results we have obtained for $Q = 30$.

d	Miss America				Salesman			
	Compression Ratio		Frame SNR		Compression Ratio		Frame SNR	
	Total	Steady State	Min	Max	Total	Steady State	Min	Max
0.5	21.69	27.35	31.93	33.70	21.46	31.13	26.86	31.13
1.0	32.82	48.12	32.02	34.02	36.82	57.38	28.26	35.83
1.5	38.91	62.68	32.73	34.24	45.38	81.58	28.72	37.94
2.0	42.88	73.79	32.50	34.44	50.90	101.59	28.93	38.75
2.5	46.30	84.65	32.36	34.54	55.02	119.64	28.90	38.96
3.0	48.81	95.35	32.10	34.60	58.26	136.30	28.77	39.07
3.5	51.95	105.89	32.00	34.69	61.22	153.93	28.73	39.05
4.0	54.36	116.55	31.80	34.76	63.96	172.67	28.73	39.14
4.5	56.70	128.03	31.71	34.88	66.52	192.91	28.57	39.05
5.0	58.92	140.01	31.50	34.91	68.74	213.08	28.54	39.00

Table 2: Compression ratios obtained by the combination of motion detection and still image compression with $Q = 30$: as a function of difference threshold d

In Figure 9 we show the original and the reconstructed 101st frame of *Miss America* using the complete scheme described above with $d = 1.5$ and $Q = 30$. Figure 10 indicates the variation of compression ratio over time. Figure 11 shows the running average compression ratios and the running average bits per pixel for a runlength of 1000, based on *Miss America* sequence with $d = 2$ and $Q = 30$. In Figure 12.a, PSNR is plotted as a function of frame number for $d = 2$, $Q = 30$. Figure 12.b shows the number of bits transmitted as a function of frame number.

Original 101st frame

Reconstructed (SNR = 32.83)

Figure 9: Original and reconstructed last frames (101st frames) in the Miss America sequence using the motion detection scheme with $d = 1.5$ combined with still image compression with $Q = 30$

4 Discussion and Conclusions

Many further improvements of the basic method we propose can be thought of and some are certainly worth further work. In particular the following observations can be used to design networks with enhanced

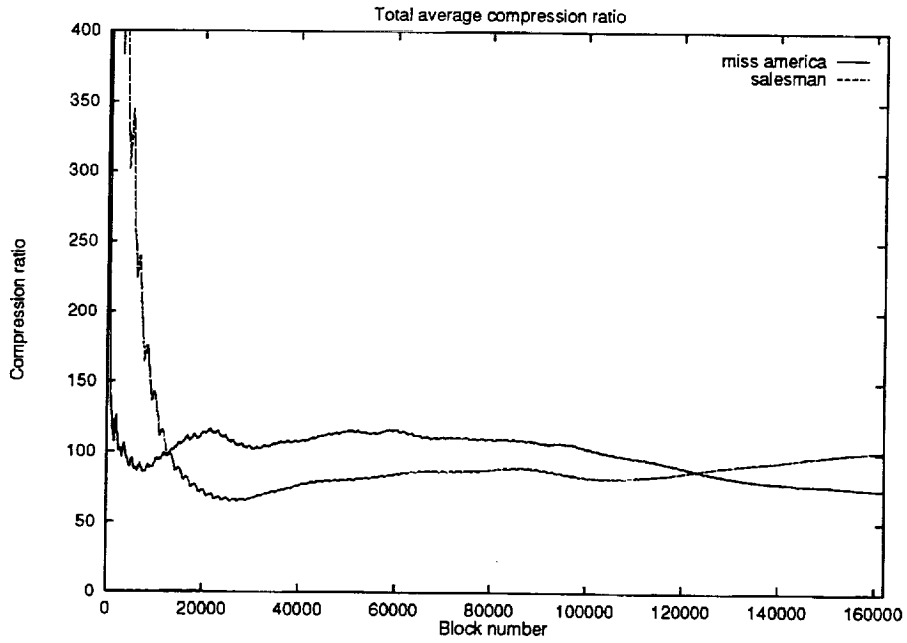
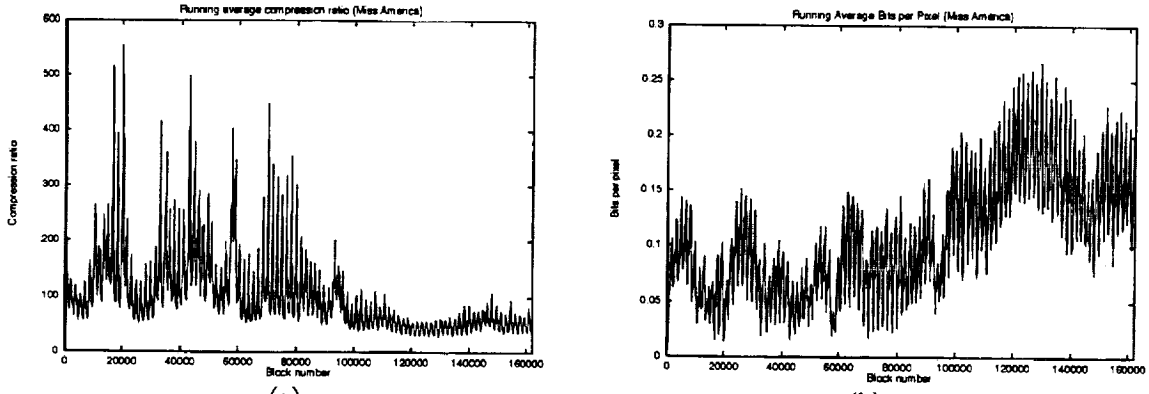
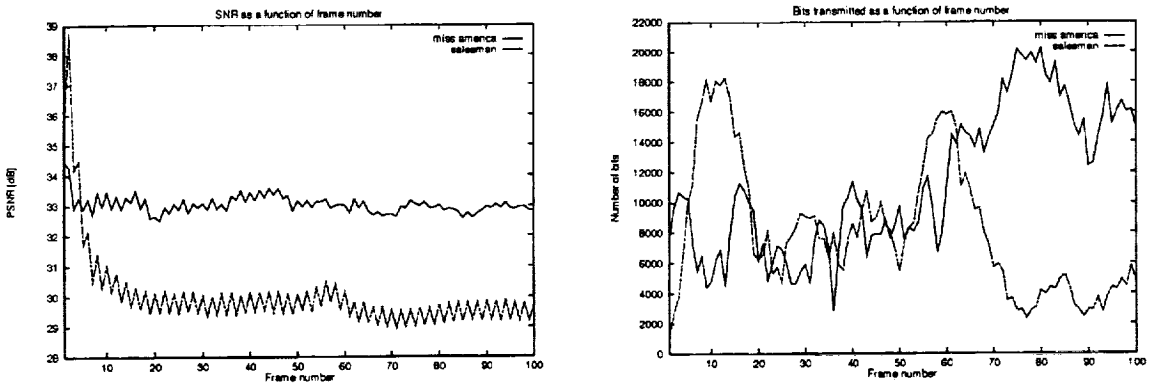


Figure 10: Total average compression ratio as a function of block number for the combined scheme with $d = 2$ and $Q = 30$



(a) (b)

Figure 11: Experimental results with MISS AMERICA sequence using the combined scheme with $d = 2$ and $Q = 30$: a) Running average compression ratio as a function of block number, b) Running average bits per pixel as a function of block number



(a) (b)

Figure 12: Experimental results for the combined scheme with $d = 2$ and $Q = 30$: a) PSNR as a function of frame number, b) Number of bits transmitted as a function of frame number

compression capabilities:

- The random neural network learning algorithm (described in the Appendix) applies to arbitrary recurrent networks. Hence, instead of restricting ourselves to fully feedforward networks, we can use feedback connections between the compressed and input layer, and the output layer and the compressed layer. Further feedback is possible and useful locally within the output layer. Such feedback can help the network find better compression/decompression parameters.
- The quality level (e.g. SNR) predicted at the transmitting end is exactly what the result is for that box, after it is decompressed at the receiver, since the networks $D_1, \ldots, D_L$ are identical both at the transmitter and receiver. Thus we propose to update the weights of the neural networks $C_1, \ldots, C_L$ constantly using gradient descent to improve performance with each individual box. This will be detrimental to the "real-time" nature of the whole approach we propose, but would be worth examining in order to obtain much higher SNR figures.
- It is also possible to store all of the compression networks $C_1, \ldots, C_L$ at the receiver – as well as at the transmitter. Then, on-going improvement via learning as compression/decompression takes place can be carried out periodically for both compression and decompression networks, at the expense of transmitting some uncompressed frames or boxes from time to time.
- Initial learning of weights can be carried out at the transmitter, or receiver, or both at the transmitter and receiver, or off-line. The resulting weights would then be loaded into the transmitter and the receiver. Note that if the sample images used for learning are known both to the transmitter and to the receiver, then the quasi-identical set of weights (to the exception of possible different numerical round-errors) can be obtained both at the transmitter and at the receiver. Thus, the images to be used as a basis for learning can be transmitted from time to time (i.e. infrequently) from one to the other in order to improve the system's compression capabilities.
- All the work described in this paper needs to be extended to colour images. Currently, learning of the weights of each C_i, D_i pair is obtained using gradient descent and the SNR ratio is used as a performance criterion is essentially equivalent to a quadratic cost function. We would use other cost metrics (such as LAB-type measures) to carry out learning for colour images.

In addition to the general scheme described above, we will examine some other enhancements related to the non-linearity of the input-output amplitude mapping of the compression/decompression scheme. We expect to obtain further quality improvement with appropriate compensation of non-linearity. This compensation

can also be part of the learning scheme. Moreover, the adaptive selection of the level of compression to be used at the transmitter side can be improved by making use of the state of the transmission medium – specifically of the network being used. This would be particularly relevant if we are dealing with an ATM (Asynchronous Transfer Mode) network. The adaptive decision can be based on feedback about network state – such as current load on the network – as well as SNR and/or visual quality metrics. For example, in case of little load on the network, we can favor small compression ratios, thus increasing visual quality. Similarly, in case of a heavily loaded network, we can sacrifice visual quality and transmit with maximal compression. This adaptive decision can also be learned.

With some of the improvements described above, we expect to achieve compression ratios better than $250:1$ for grey-level moving image sequences, and still higher levels for colour, with quality levels of the order of $SNR = 30$ for grey level images, and acceptable LAB-type measures and SNR levels for colour images.

5 Appendix: The Random Neural Network Model and its Learning Algorithm

In this appendix we provide a summary of the Random Neural Network Model and of its Learning Algorithm, in order to provide a theoretical background for the techniques which are used in this paper.

5.1 The Random Neural Network Model

In the random neural network model (Gelenbe (1989,90) [12, 13]) signals in the form of spikes of unit amplitude circulate among the neurons. Positive signals represent excitation and negative signals represent inhibition. Each neuron's state is a non-negative integer called its potential, which increases when an excitation signal arrives to it, and decreases when an inhibition signal arrives. Thus, an excitatory spike is interpreted as a "+1" signal at a receiving neuron, while an inhibitory spike is interpreted as a "−1" signal.

Neural potential also decreases when the neuron fires. Thus a neuron i emitting a spike, whether it be an excitation or an inhibition, will lose potential of one unit, going from some state whose value is k_i to the state of value $k_i - 1$.

The state of the n-neuron network at time t, is represented by the vector of non-negative integers $k(t) = (k_1(t), \ldots, k_n(t))$, where $k_i(t)$ is the potential or integer state of neuron i. We will denote by k and k_i arbitrary values of the state vector and of the i-th neuron's state.

Neuron i will "fire" (i.e. become excited and send out spikes) if its potential is *positive*. The spikes will then be sent out at a rate $r(i)$, with independent, identically and exponentially distributed inter-spike intervals. Spikes will go out to some neuron j with probability $p^+(i,j)$ as excitatory signals, or with probability $p^-(i,j)$ as inhibitory signals. A neuron may also send signals out of the network with probability $d(i)$, and $d(i) + \sum_{j=1}^{n} [p^+(i,j) + p^-(i,j)] = 1$. Let $w_{ij}^+ = r(i)\, p^+(i,j)$, and $w_{ij}^- = r(i)\, p^-(i,j)$. Here the "$w$'s" play a role similar to that of the synaptic weights in connectionist models, though they specifically represent rates of excitatory and inhibitory spike emission. They are non-negative. Exogenous (i.e. those coming from the "outside world") excitatory and inhibitory signals also arrive to neuron i at rates $\Lambda(i)$, $\lambda(i)$, respectively.

This is a "recurrent network" model, *i.e.* a network which is allowed to have feedback loops, of arbitrary topology.

Computations related to this model are based on the probability distribution of network state $p(k,t) = \Pr[k(t) = k]$, or with the marginal probability that neuron i is excited $q_i(t) = \Pr[k_i(t) > 0]$. As a consequence, the time-dependent behaviour of the model is described by an infinite system of *Chapman-Kolmogorov* equations for discrete state-space continuous Markovian systems.

Information in this model is carried by the *frequency* at which spikes travel. Thus, neuron j, if it is excited, will send spikes to neuron i at a frequency $w_{ij} = w_{ij}^+ + w_{ij}^-$. These spikes will be emitted at exponentially distributed random intervals. In turn, each neuron behaves as a non-linear *frequency demodulator* since it transforms the incoming excitatory and inhibitory spike trains' rates into an "amplitude", which is $q_i(t)$ the probability that neuron i is excited at time t. Intuitively speaking, each neuron of this model is also a frequency modulator, since neuron i sends out excitatory and inhibitory spikes at rates (or frequencies) $q_i(t)r(i)p^+(i,j)$, $q_i(t)r(i)p^-(i,j)$ to any neuron j.

The stationary probability distribution associated with the model is the quantity used throughout the com-

putations:

$$p(k) = \lim_{t\to\infty} p(k,t), \quad q_i = \lim_{t\to\infty} q_i(t), \quad i = 1,...,n. \tag{3}$$

It is given by the following result:

Theorem 1. *Let q_i denote the quantity*

$$q_i = \lambda^+(i)/[r(i) + \lambda^-(i)] \tag{4}$$

where the $\lambda^+(i), \lambda^-(i)$ for $i = 1,...,n$ satisfy the system of nonlinear simultaneous equations:

$$\lambda^+(i) = \sum_j q_j r(j) p^+(j,i) + \Lambda(i), \quad \lambda^-(i) = \sum_j q_j r(j) p^-(j,i) + \lambda(i) \tag{5}$$

Let $k(t)$ be the vector of neuron potentials at time t and $k = (k_1,...,k_n)$ be a particular value of the vector; let $p(k)$ denote the stationary probability distribution.

$$p(k) \;=\; \lim_{t\to\infty} \text{Prob}[k(t) = k]$$

If a nonnegative solution $\{\lambda^+(i), \lambda^-(i)\}$ exists to equations 4 and 5 such that each $q_i < 1$, then

$$p(k) = \prod_{i=1}^{n} [1 - q_i] q_i^{k_i} \tag{6}$$

The quantities which are most useful for computational purposes, *i.e.* the probabilities that each neuron is excited, are directly obtained from:

$$\lim_{t\to\infty} \text{Prob}[k_i(t) > 0] \;=\; q_i \;=\; \lambda^+(i)/[r(i) + \lambda^-(i)] \qquad \text{if } q_i < 1.$$

5.2 The Learning Algorithm

Let us describe the learning algorithm we use in this study. It is based on the algorithm described in (Gelenbe 93) [16].

The algorithm chooses the set of network parameters $\mathbf{W}$ in order to learn a given set of K input-output pairs $(\iota, \mathbf{Y})$ where the set of successive inputs is denoted $\iota = \{\iota_1,...,\iota_K\}$, and $\iota_k = (\Lambda_k, \lambda_k)$ are pairs of positive and negative signal flow rates entering each neuron:

$$\boldsymbol{\Lambda}_k \;=\; [\Lambda_k(1),...,\Lambda_k(n)], \qquad \boldsymbol{\lambda}_k = [\lambda_k(1),...,\lambda_k(n)]$$

The successive desired outputs are the vectors $\mathbf{Y} = \{y_1,...,y_K\}$, where each vector $y_k = (y_{1k},...,y_{nk})$, whose elements $y_{ik} \epsilon [0,1]$ correspond to the desired values of each neuron. The network approximates the set of desired output vectors in a manner that minimizes a cost function E_k:

$$E_k \;=\; \frac{1}{2}\sum_{i=1}^{n} a_i (q_i - y_{ik})^2, \quad a_i \geq 0$$

If we wish to remove some neuron j from network output, and hence from the error function, it suffices to set $a_j = 0$

Both of the n by n weight matrices $\mathbf{W}_k^+ = \{w_k^+(i,j)\}$ and $\mathbf{W}_k^- = \{w_k^-(i,j)\}$ have to be learned after each input is presented, by computing for each input $\iota_k = (\Lambda_k, \lambda_k)$, a new value $\mathbf{W}_k^+$ and $\mathbf{W}_k^-$ of the weight matrices, using gradient descent. Clearly, we seek only solutions for which all these weights are positive.

Let $w(u,v)$ denote any weight term, which would be either $w(u,v) \equiv w^-(u,v)$, or $w(u,v) \equiv w^+(u,v)$. The weights will be updated as follows:

$$w_k(u,v) = w_{k-1}(u,v) - \eta \sum_{i=1}^{n} a_i(q_{ik} - y_{ik})[\partial q_i / \partial w(u,v)]_k \tag{7}$$

where $\eta > 0$ is some constant, and

1. q_{ik} is calculated using the input ι_k and $w(u,v) = w_{k-1}(u,v)$, in equation 3.
2. $[\partial q_i / \partial w(u,v)]_k$ is evaluated at the values $q_i = q_{ik}$ and $w(u,v) = w_{k-1}(u,v)$.

To compute $[\partial q_i / \partial w(u,v)]_k$ we turn to the expression 3, from which we derive the following equation:

$$\begin{aligned} \partial q_i / \partial w(u,v) \;=\; & \sum_j \partial q_j / \partial w(u,v)[w^+(j,i) - w^-(j,i)q_i]/D(i) \\ & -\mathbf{1}[u=i]q_i/D(i) \\ & +\mathbf{1}[w(u,v) \equiv w^+(u,i)]q_u/D(i) \\ & -\mathbf{1}[w(u,v) \equiv w^-(u,i)]q_u q_i/D(i) \end{aligned}$$

Let $\mathbf{q} = (q_1, ..., q_n)$, and define the $n \times n$ matrix

$$\mathbf{W} \;=\; \{[w^+(i,j) - w^-(i,j)q_j]/D(j)\} \quad i,j = 1, ..., n$$

We can now write the vector equations:

$$\begin{aligned} \partial \mathbf{q} / \partial w^+(u,v) &= \partial \mathbf{q} / \partial w^+(u,v)\mathbf{W} + \boldsymbol{\gamma}^+(u,v)q_u \\ \partial \mathbf{q} / \partial w^-(u,v) &= \partial \mathbf{q} / \partial w^-(u,v)\mathbf{W} + \boldsymbol{\gamma}^-(u,v)q_u \end{aligned}$$

where the elements of the n-vectors $\boldsymbol{\gamma}^+(u,v) = [\gamma_1^+(u,v), \ldots, \gamma_n^+(u,v)]$, $\boldsymbol{\gamma}^-(u,v) = [\gamma_1^-(u,v), \ldots, \gamma_n^-(u,v)]$ are

$$\gamma_i^+(u,v) \;=\; \begin{cases} -1/D(i) & \text{if } u = i, v \neq i \\ +1/D(i) & \text{if } u \neq i, v = i \\ 0 & \text{for all other values of } (u,v) \end{cases}$$

$$\gamma_i^-(u,v) \;=\; \begin{cases} -(1+q_i)/D(i) & \text{if } u = i, v = i \\ -1/D(i) & \text{if } u = i, v \neq i \\ -q_i/D(i) & \text{if } u \neq i, v = i \\ 0 & \text{for all other values of } (u,v) \end{cases}$$

Notice that

$$\begin{aligned} \partial \mathbf{q} / \partial w^+(u,v) &= \boldsymbol{\gamma}^+(u,v)q_u[\mathbf{I} - \mathbf{W}]^{-1} \\ \partial \mathbf{q} / \partial w^-(u,v) &= \boldsymbol{\gamma}^-(u,v)q_u[\mathbf{I} - \mathbf{W}]^{-1} \end{aligned} \tag{8}$$

where $\mathbf{I}$ denotes the n by n identity matrix. Hence the main computational work is to obtain $[\mathbf{I} - \mathbf{W}]^{-1}$. This is of time complexity $O(n^3)$, or $O(mn^2)$ if an m-step relaxation method is used.

We now have the information to specify the complete learning algorithm for the network. We first initialize the matrices $\mathbf{W}_0^+$ and $\mathbf{W}_0^-$ in some appropriate manner. This initiation will be made at random. Choose a value of η, and then for each successive value of k, starting with $k = 1$ proceed as follows:

1. Set the input values to $\iota_k = (\Lambda_k, \lambda_k)$.
2. Solve the system of nonlinear equations 3 with these values.
3. Solve the system of linear equations (8) with the results of (2).
4. Using equation 7 and the results of (2) and (3), update the matrices $\mathbf{W}_k^+$ and $\mathbf{W}_k^-$. Since we seek the "best" matrices (in terms of gradient descent of the quadratic cost function) that satisfy the *nonnegativity* constraint, in any step k of the algorithm, if the iteration yields a negative value of a term, we have two alternatives:
   (a) Set the term to zero, and stop the iteration for this term in this term in this step k; in the next stop $k+1$ we will iterate on this term with the same rule starting from its current null value;
   (b) Go back to the previous value of the term and iterate with a smaller value of η.

This general scheme can be specialized to feedforward networks yielding a computational complexity of $O(n^2)$, rather than $O(n^3)$, for each gradient iteration.

References

[1] Adelson, E.H., Simoncelli, E. "Orthogonal pyramid transforms for image coding", *Visual Communications and Image Processing II*, Proc. SPIE, Vol.845, pp.50–58, 1987.

[2] Anthony D. "A comparison of image compression by a Neural Network and Principle Component Analysis". *Proc. International Joint Conference on Neural Networks (IJCNN'90)*, pp. 339–344. IEEE, 1990.

[3] Atalay V., Gelenbe E., Yalabık N., "The random neural network model for texture generation", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 6, No. 1, pp 131-141, 1992.

[4] Atalay V., Gelenbe E., "Parallel algorithm for colour texture generation using the random neural network model", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 6, No. 2-3, pp 437-446, 1992.

[5] Carrato, S., Marsi, S. "Parallel Structure Based on Neural Networks for Image Compression", *Electronics Letters*, Vol.28, No.12, pp. 1152–1153, June 1992.

[6] Carrato, S. "Neural networks for image compression," in Gelenbe, E. (ed.) *"Neural Networks: Advances and Applications 2"*, Elsevier North-Holland, pp. 177-198, 1992.

[7] Chiang Y.W. "Motion estimation using a neural network." *Proc. IEEE International Symposium on Circuits and Systems*. IEEE, 1990.

[8] Cottrell, G.W., Munro, P., Zipser, D. "Image compression by backpropagation: an example of extensional programming,", in Sharkey, N.E., (ed.) *Models of cognition: a review of cognition science*, NJ:Norwood, 1989.

[9] Courellis S.H. "An Artifical Neural Network for Motion Detection and Speed Estimation". *Proc. International Joint Conference on Neural Networks (IJCNN'90*, pp. 407–421. IEEE, 1990.

[10] Daugman J.G. "Relaxation Neural Network for Non-Orthogonal Image Transformations' '. *Proc. International Conference on Neural Networks*. IEEE, 1988.

[11] Feng W.C. "Real-Time Neuroprocessor for Adaptive Image Compression Based upon Frequency-Sensitive Competetive Learning". *Proc. The International Joint Conference on Neural Networks (IJCNN'91)*, pp 429. IEEE, 1991.

[12] Gelenbe E., "Random neural networks with negative and positive signals and product form solution", *Neural Computation*, Vol. 1, No. 4, pp 502-511, 1989.

[13] Gelenbe E., "Stability of the random neural network model", *Neural Computation*, Vol. 2, No. 2, pp. 239-247, 1990.

[14] Gelenbe, E., Stafylopatis, A., "Global behaviour of homogeneous random neural systems", *Applied Math. Modelling*, **15** (1991), pp. 535–541.

[15] Gelenbe E., Stafylopatis A., Likas A., "An extended random network model with associative memory capabilities", *Proc. International Conference on Artificial Neural Networks (ICANN'91)*, Helsinki, June 1991.

[16] Gelenbe E., "Learning in the recurrent random neural network", *Neural Computation*, Vol. 5, No. 1, pp 154-164, 1993.

[17] Gelenbe, E., Sungur, M. "Image compression with the random neural network", to appear in *Proc. International Conference on Artificial Neural Networks*, North-Holland Elsevier, 1994.

[18] Gray, R.M. "Vector Quantization", *IEEE ASSP Magazine*, Vol.1, No.2, pp.4–29, April 1984.

[19] Huang S.J. "Image Data Compression and Generalization Capabilities of Backpropagation and Recirculation Networks". *Proc. International Symposium on Circuits and Systems*, page 1613. IEEE, 1991.

[20] Jacquin, A.E. "Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations", Vol.1, No.1, p.18–30, January 1992.

[21] Klein S.A. "'Perfect' Displays and 'Perfect' Image Compressionin Space and Time". *Human Vision, Visual Processing and Digital Display*, pp. 190–205. SPIE, 1991.

[22] Kohno R. "Image compression using a neural network with learning capability of variable function of the neural unit." *Visual Communication and Image Processing*, pp. 69–75. SPIE, 1990.

[23] Kohonen, T. *Self Organization and Associative Memory*, Springer-Verlag:Berlin, 1989.

[24] Kunt, M., Benard, M., Leonardi, R. "Recent Results in High-Compression Image Coding", *IEEE Transactions on Circuits and Systems*, Vol.CAS–34, No.1, pp. 1306–1336, 1987.

[25] LeGall, D. "MPEG : A Video Compression Standard for Multimedia Applications. *Communications of the ACM*, Vol. 34, No. 4, pp. 46–58, April 1991.

[26] Marsi S. "Improved Neural Structures for Image Compression". *Proc. International Conference on Acoustic Speech and Signal Processing (ICASSP'91)*, page 2821. IEEE, 1991.

[27] Martine Naillon. *Advances in Neural Processing Systems*. Morgan-Kaufmann, 1989.

[28] Namphol A. "Higher Order Data Compression with Neural Networks". *Proc. The International Joint Conference on Neural Networks (IJCNN'91)*, pp. 55–59. IEEE, 1991.

[29] Nasrabadi N.M. "Vector quantization of images based upon Kohonen self organizing feature maps." *Proc. International Conference on Neural Networks*. IEEE, 1988.

[30] Ramamurthi, B., Gersho, A., "Classified Vector Quantization of Images", *IEEE Transactions on Communications*, Vol.COM–34, No.11, pp.1105–1115, November 1986.

[31] Rumelhart, D.E., McClelland, J.L. and the PDP Research Group (1986) *"Parallel Distributed Processing"*, Volumes 1 & 2, MIT Press, 1986.

[32] Sonehara N. "Image Data Compression Using a Neural Network Model". *Proc. International Joint Conference on Neural Networks (IJCNN'89)*. IEEE, 1989.

[33] Storer, J.A. (1988) *"Data Compression: Methods and Theory"*, Computer Science Press, Rockville, MD, 1988.

[34] Wallace, G.K., "The JPEG Still Picture Compression Standard, *Communications of the ACM*, Vol. 34, No. 4, pp. 30–44, April 1991.

[35] Woods, J., O'neil, S.D. "Subband Coding of Images", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol.ASSP–34, No.5, pp.1278–1288, October 1986.

[36] Zettler, W., Huffman, J., Linden, D.C.P. "Application of Compactly Supported Wavelets to Image Compression", *Image Processing Algorithms and Techniques*, Proc. SPIE, Vol.1244, pp.150–160, 1990.

Learning to Train Neural Networks for Real-World Control Problems

Lee A. Feldkamp
G. V. Puskorius
L. I. Davis, Jr.
F. Yuan

Research Laboratory
Ford Motor Company
MD 3135 SRL
P.O. Box 2053
Dearborn, MI 48121-2053
lfeldkam@smail.srl.ford.com

ABSTRACT

Over the past three years, our group has concentrated on the application of neural network methods to the training of controllers for real-world systems. This presentation will describe our approach, survey what we have found to be important, mention some contributions to the field, and show some representative results. Topics to be discussed include:

1) executing model studies as rehearsal for experimental studies
2) the importance of correct derivatives
3) effective training with second-order (DEKF) methods
4) the efficacy of time-lagged recurrent networks
5) liberation from the tyranny of the control cycle using asynchronous truncated backpropagation through time
6) multi-stream training for robustness

Results from model studies of automotive idle speed control will serve as examples for several of these topics. Experimental results may also be shown.

JPL

An Integrated Optoelectronic ATR Processor

JPL
Neural Network Workshop

May 11, 1994

Tien-Hsin Chao

Center for Space Microelectronics Technology
Jet Propulsion Laboratory
California Institute of Technology

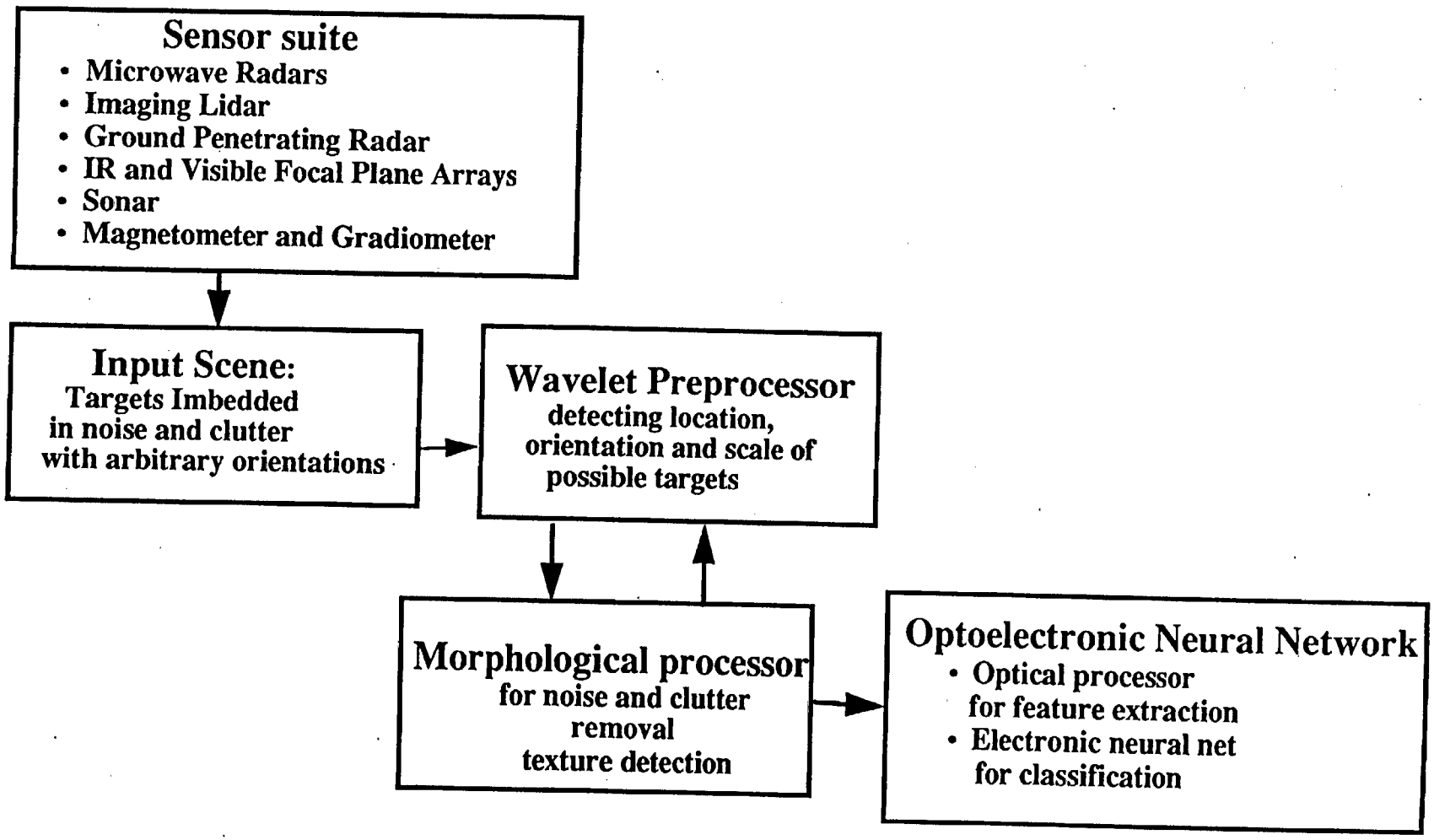
JPL

Optoelectronic ATR Systems Development at JPL

- Technology base developed by more than 8 years of continued DoD and NASA sponsorships
- JPL has extended experience in developing
  - Optoelectronic Neural Network
  - Wavelet Processor
  - Morphological Processor
- Optoelectronic ATR system development work:
  - Algorithm development, architecture design and simulation
  - Innovative optical and optoelectronic hardware development
  - Compact system integration
  - Experimental demonstration

JPL

An Integrated Optoelectronic ATR Processor



JPL

Pay Off

The successful completion of the integrated Optoelectronic ATR Processor will:

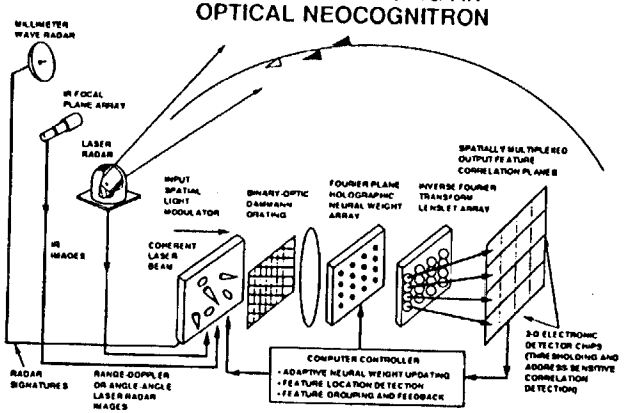
- **Provide an enabling solution to sensor signature processing with high speed, large throughput and compact package**

- **Be readily applicable to many ATR problems for DoD, NASA, EPA, and industry.**

JPL

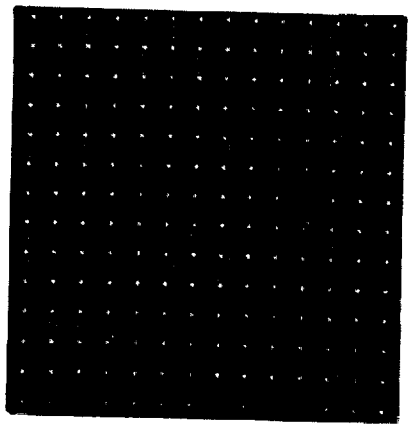
OPTICAL NEOCOGNITRON FOR MULTISENSOR AUTOMATIC TARGET RECOGNITION

MULTISENSOR AUTOMATIC TARGET RECOGNITION USING AN OPTICAL NEOCOGNITRON

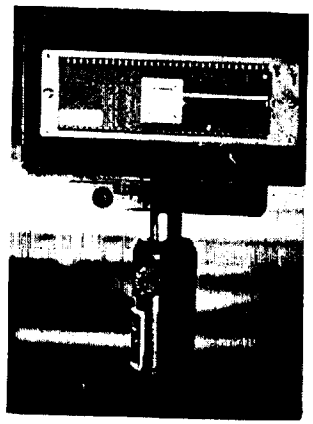


A GENERIC LAYER OF AN OPTOELECTRONIC NEOCOGNITRON

New Optic and Optoelectronic Devices



15 x 15 diffraction pattern of a Dammann grating



picture of a 64 x 64 thresholding photodetector array system

Objective:

To develop an optical neocognitron for high speed, fault tolerant, multisensor automatic target recognition and tracking

Approach:

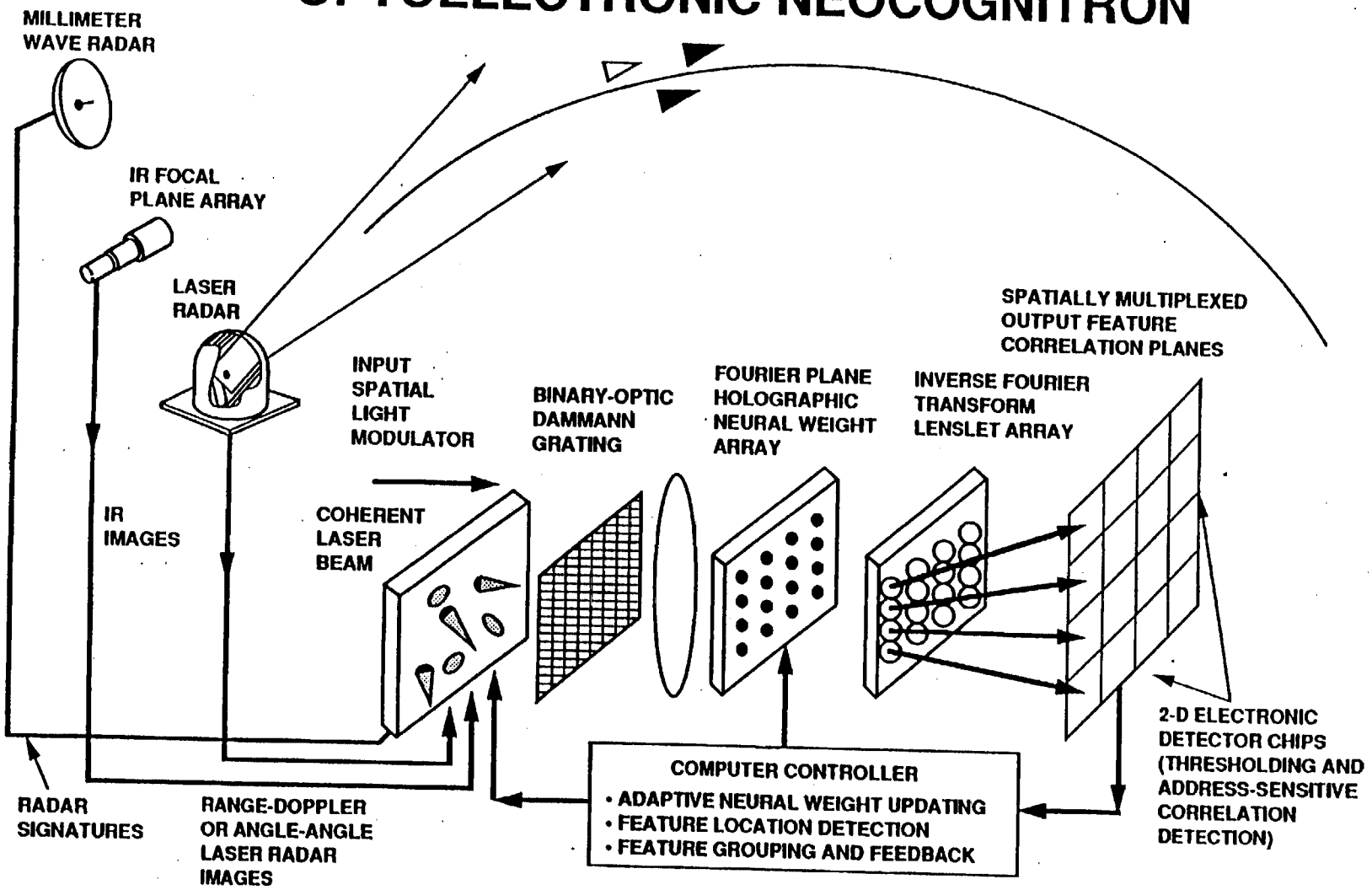
- Develop a multichannel correlator based neocognitron architecture for feature correlations
- Develop a binary-optic Dammann grating for global interconnection
- Develop a custom VLSI photodetector detector chip array for high speed feature detection

Advantages:

- Optically implemented neocognitron neural network possesses the inherent advantages of parallel processing, massive interconnectivity, shift invariance, and distortion invariance
- System processing speed exceeds 10^{14} connections/sec, at least two orders of magnitude faster than that of its state-of-the-art electronic counterpart
- Optically implemented neocognitron is uniquely suitable for 2-D image and sensor data recognition and classification

JPL

MULTISENSOR AUTOMATIC TARGET RECOGNITION USING OPTOELECTRONIC NEOCOGNITRON

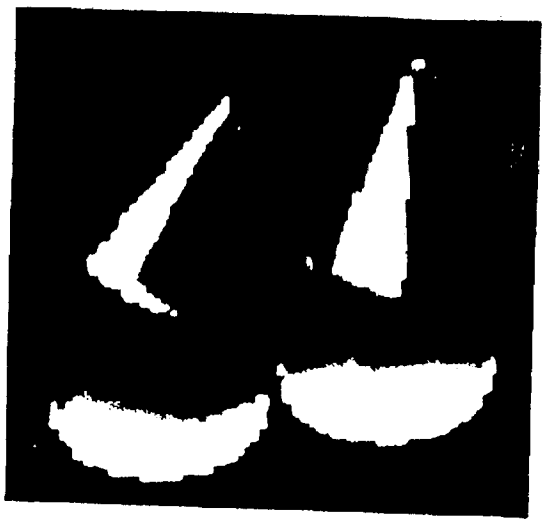


A GENERIC LAYER OF AN OPTOELECTRONIC NEOCOGNITRON

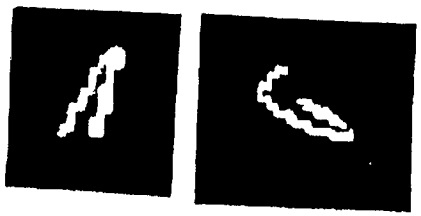
JPL

EXPERIMENTAL DEMONSTRATION: RECOGNITION OF RVS WITH INTRA-CLASS INVARIANCE AND REJECTION OF DECOYS WITH INTER-CLASS DISCRIMINATION

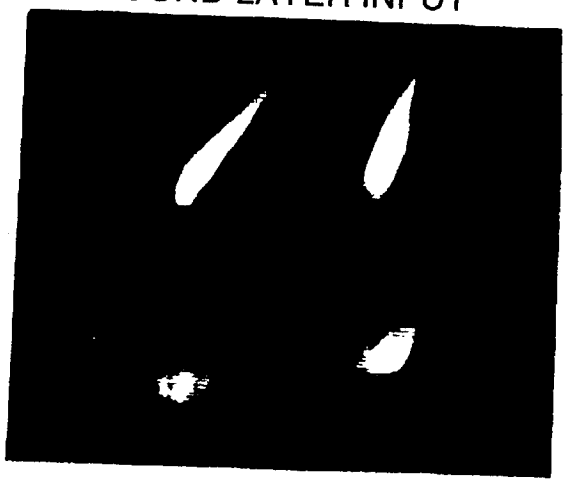
INPUT: SIMULATED LASER RADAR ANGLE-ANGLE IMAGES OF 2 REENTRY VEHICLES AND 2 DECOYS



FIRST LAYER TRAINING FEATURES

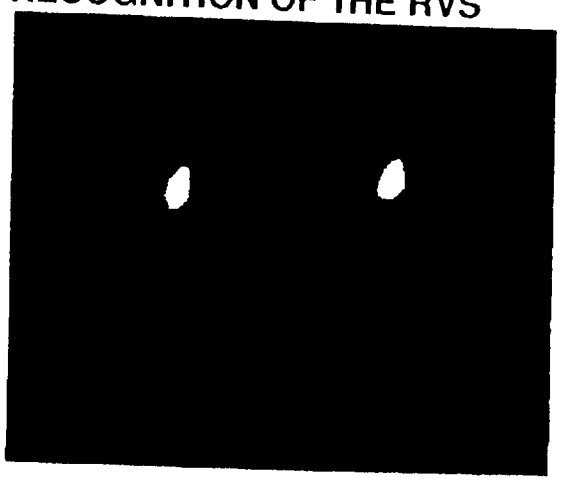


FIRST LAYER OUTPUT AND SECOND LAYER INPUT



SECOND LAYER TRAINING FEATURES

SECOND LAYER OUTPUT - RECOGNITION OF THE RVS

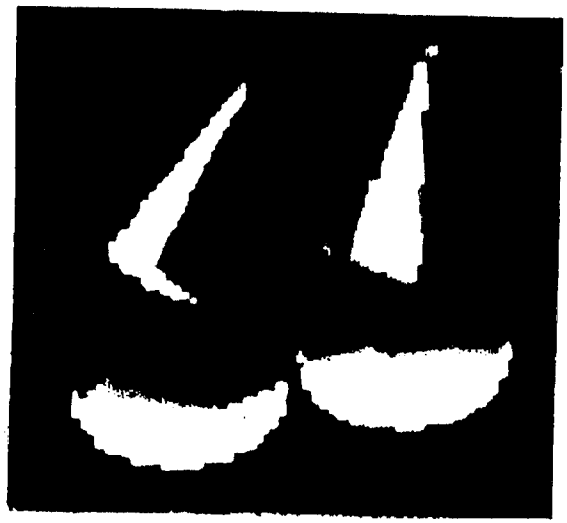


THE TWO RVS WERE RECOGNIZED AND THE TWO DECOYS WERE REJECTED WITH A TWO-LAYER OPERATION

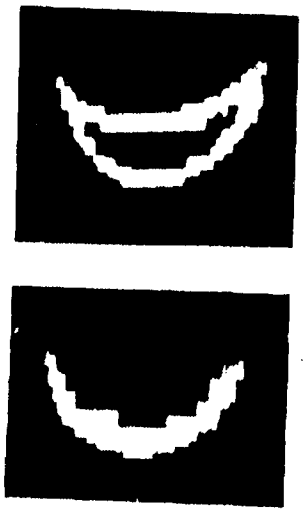
JPL

EXPERIMENTAL DEMONSTRATION: RECOGNITION OF DECOYS WITH INTRA-CLASS INVARIANCE AND REJECTION OF RVS WITH INTER-CLASS DISCRIMINATION

INPUT: SIMULATED LASER RADAR ANGLE-ANGLE IMAGES OF 2 REENTRY VEHICLES AND 2 DECOYS

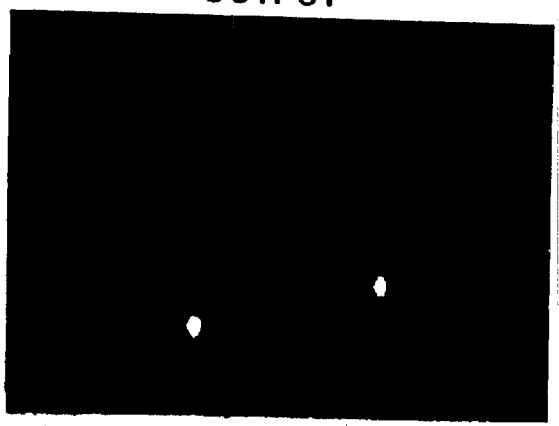


FIRST LAYER TRAINING FEATURES



FIRST LAYER OUTPUT BEFORE THRESHOLDING

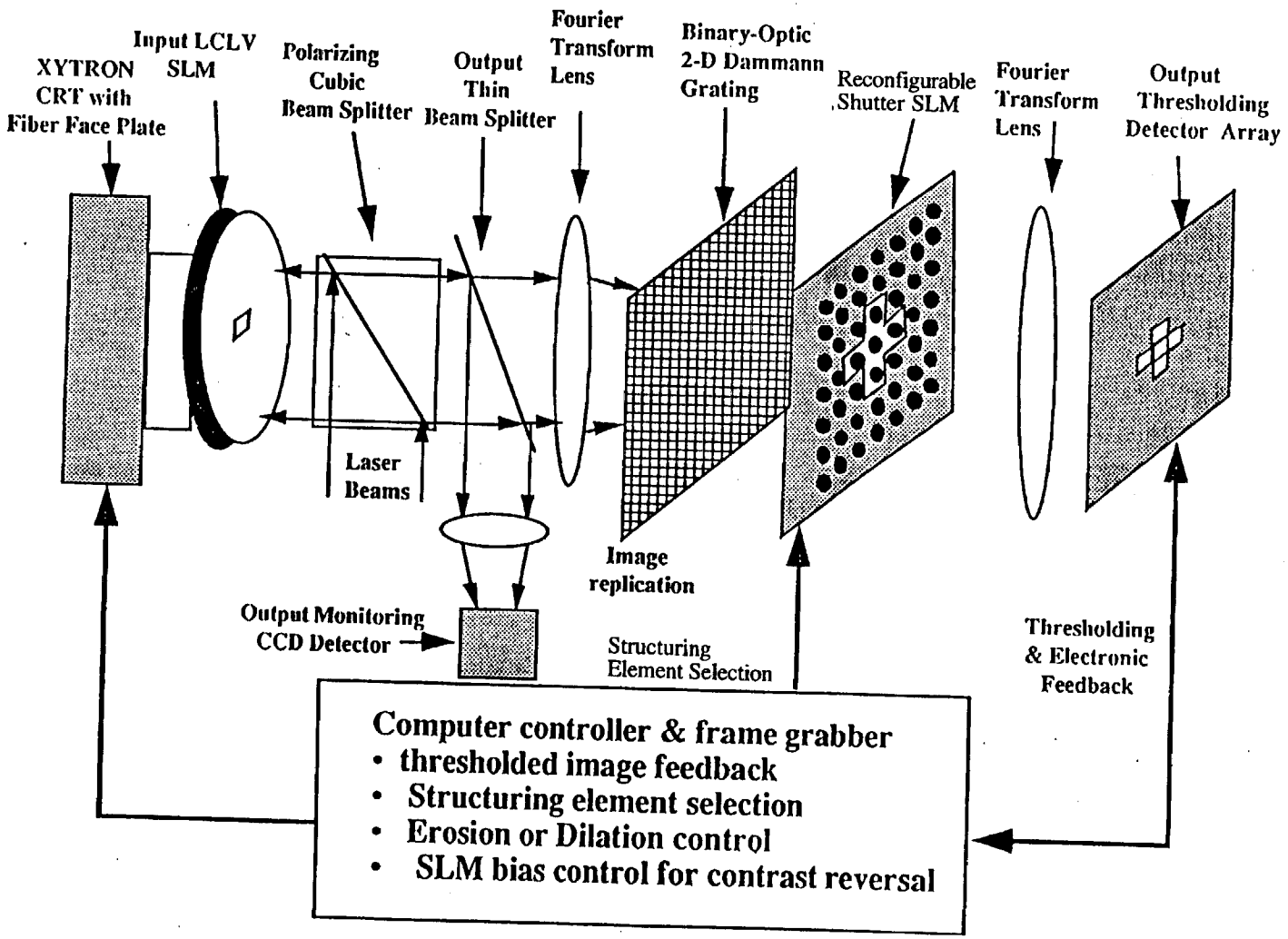
FIRST LAYER THRESHOLDED OUTPUT



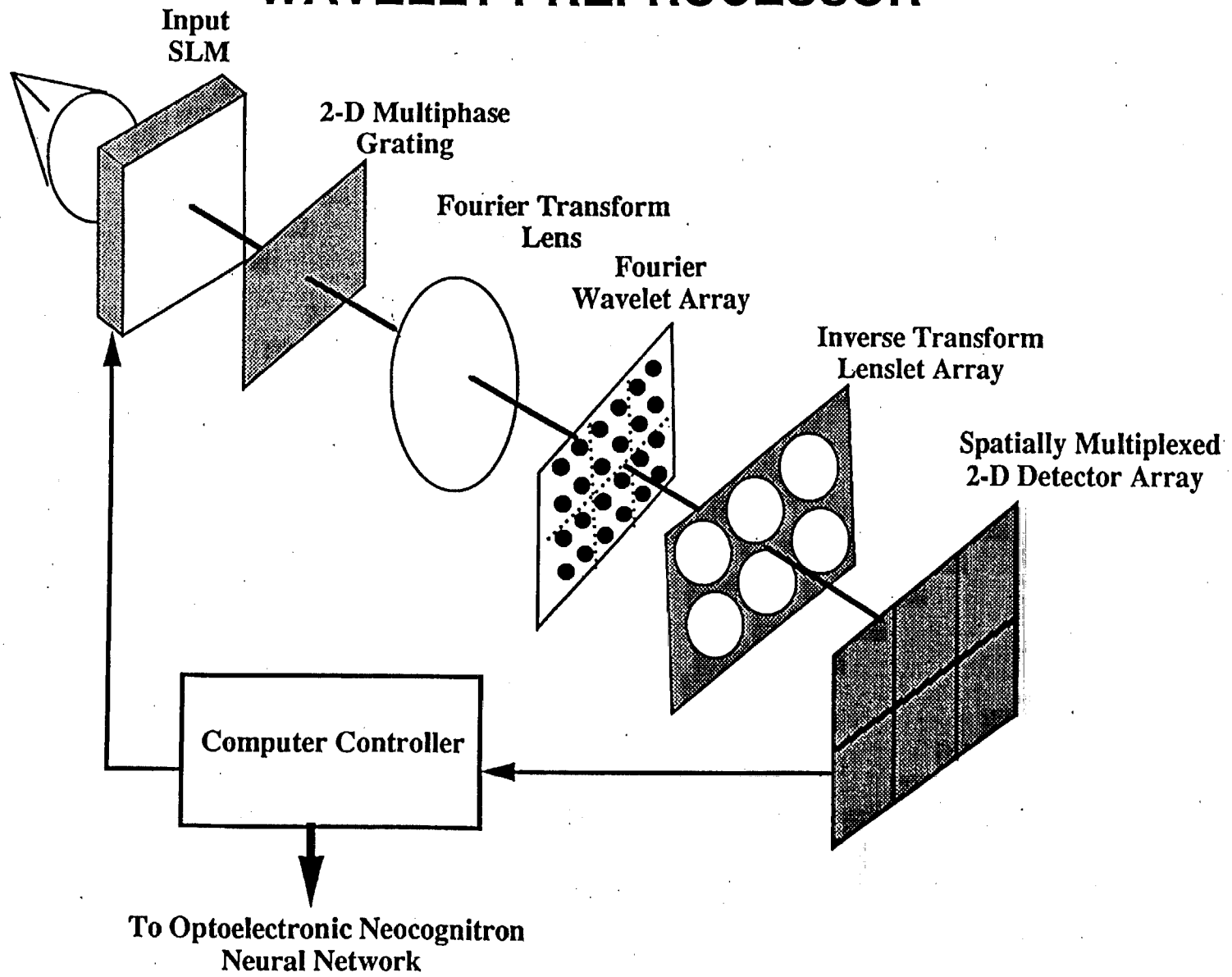
THE TWO DECOYS WERE SUCCESSFULLY RECOGNIZED WHILE THE TWO RVS WERE EFFECTIVELY REJECTED WITH A SINGLE LAYER OF OPERATION

JPL

SYSTEM SCHEMATIC OF AN OPTICAL MORPHOLOGICAL PROCESSOR

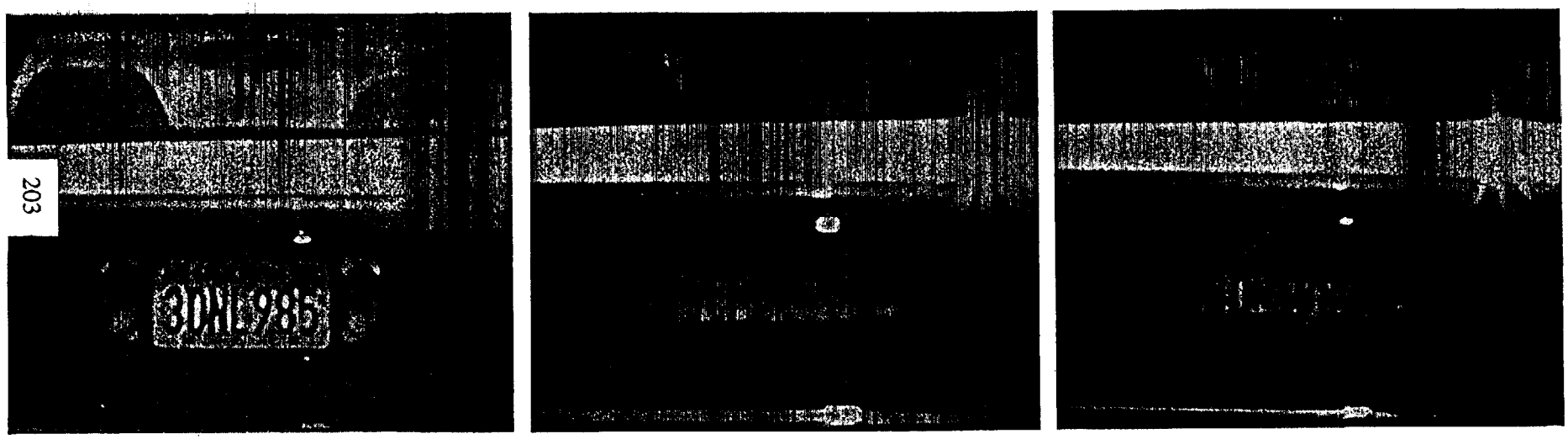


SYSTEM ARCHITECTURE OF AN OPTOELECTRONIC WAVELET PREPROCESSOR



JPL

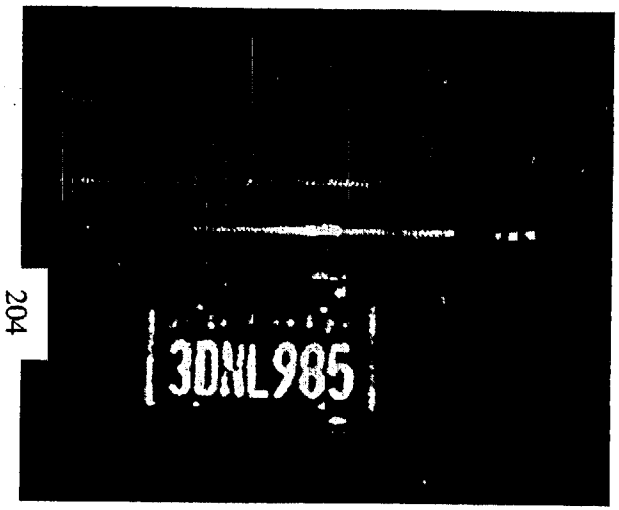
Car License Plate Detection Using Morphological and Wavelet Processing

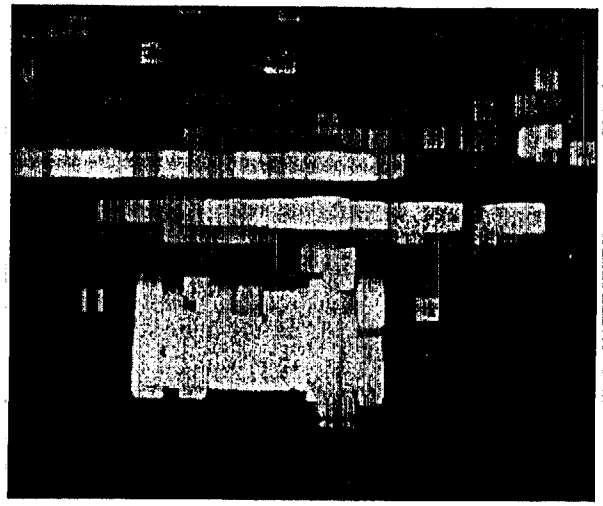


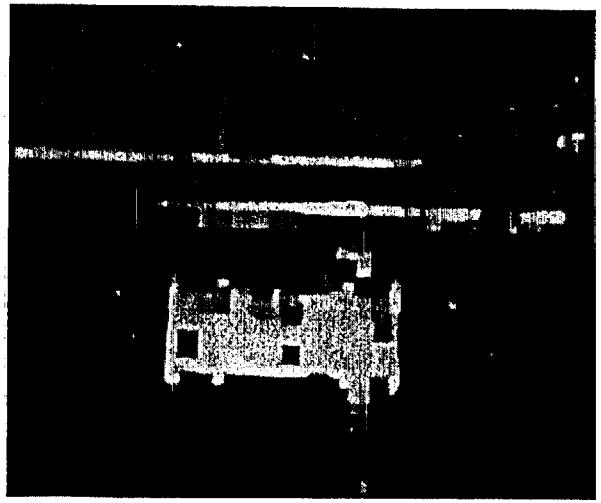
Input-
Rear View of a Car → Morphologically
Dilated → Morphologically
Eroded

JPL

Car License Plate Detection Using Morphological and Wavelet Processing -Continued-







Subtraction Result Between the input and the Morphologically Processed Output

-- License Plate Text Enhancement

→

Second Morphological Dilation

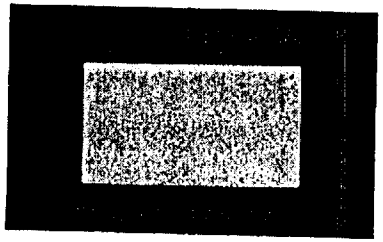
→

Second Morphological Erosion

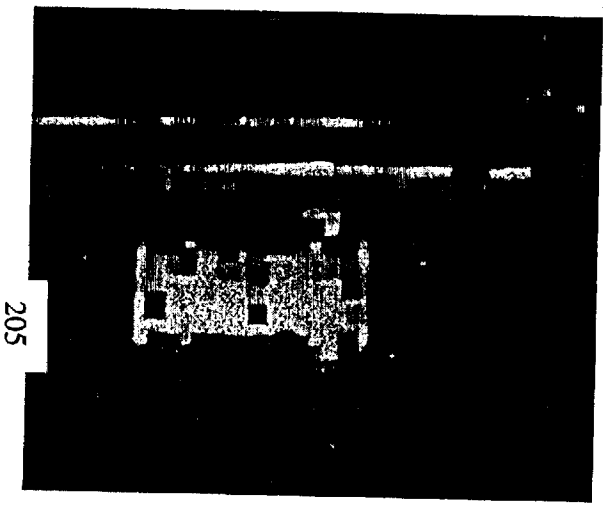
- Location of License Plate Highlighted and Singled-out

JPL

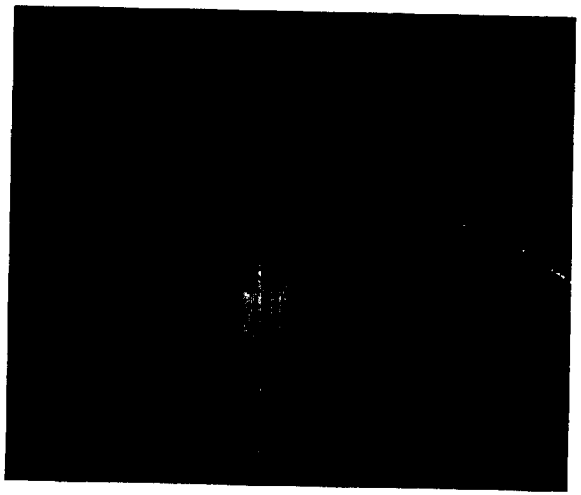
Car License Plate Detection Using Morphological and Wavelet Processing -Continued-



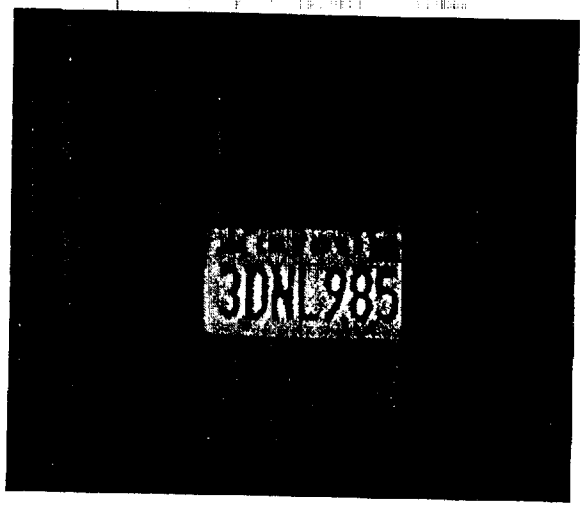
Correlation with a Rectangular Wavelet



Morphologically Processed output - ready for License Plate Location Identification

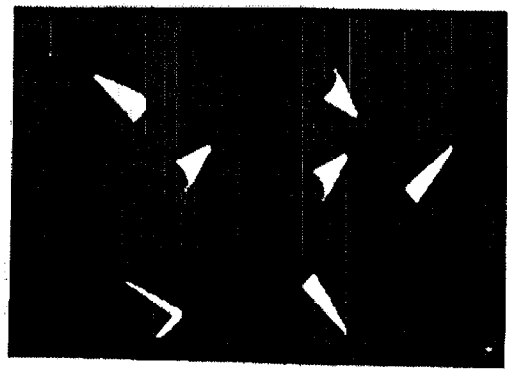


Output Plane Peak Detection Corresponds to Location of License Plate

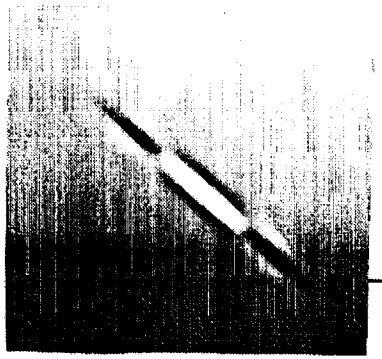


Singled-out License Plate

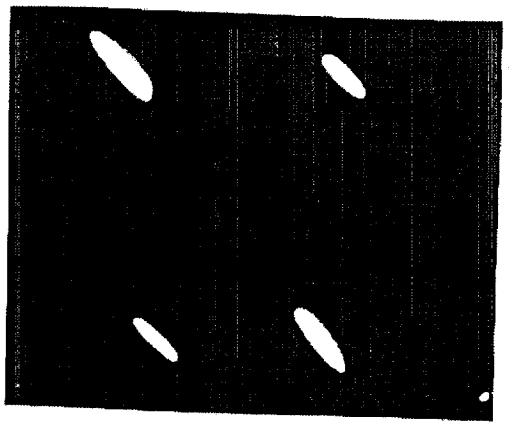
Orientation and Location Sensitive Wavelet Preprocessing

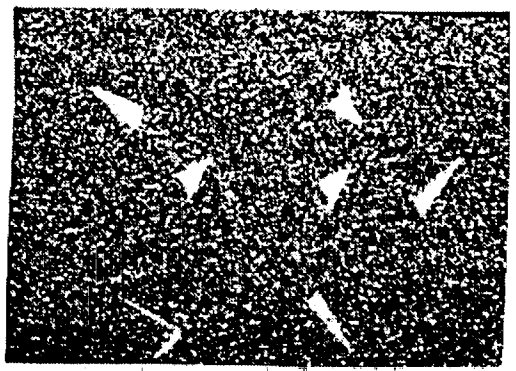


Input scene containing War Heads with different orientations and sizes

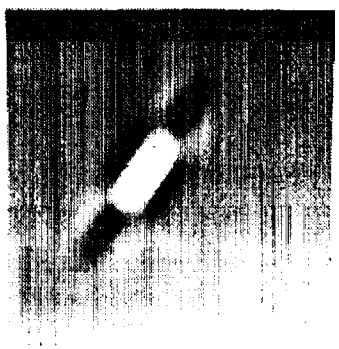


135 degree Morlet Wavelet

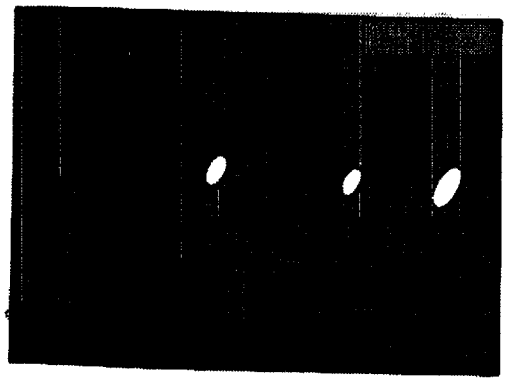




Input with 25% Added Noise

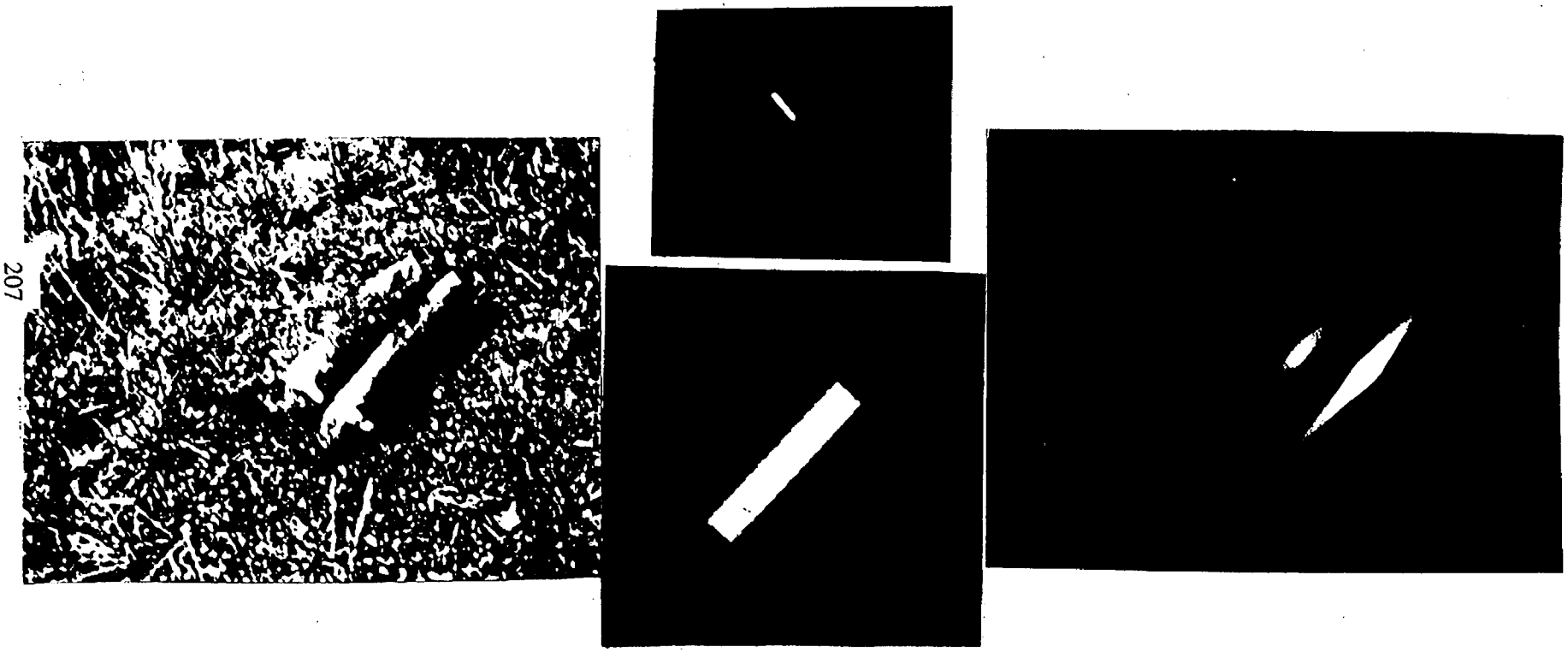


45 degree Morlet Wavelet



Detection of orientations and locations of War Head images

JPL ORDNANCE IDENTIFICATION FROM FOCAL PLANE ARRAY IMAGERY USING WAVELET PROCESSING



TWO RUSTY 105 SHELLS AT THE BLACK HILLS ARMY DEPOT

WAVELET FILTERS

WAVELET PROCESSED OUTPUT- DETECTION OF THE ORDNANCE

CONCLUSIONS

- HARDWARE IMPLEMENTED AUTOMATIC TARGET RECOGNITION PROCESSING SYSTEMS OFFER ENABLING SOLUTIONS TO DATA PROCESSING AS REQUIRED BY AIRBORNE ENVIRONMENTAL MONITORING SENSOR SUITE.
- A BROAD TECHNOLOGY BASE HAS BEEN ESTABLISHED AT JPL.
- JPL'S NEUROPROCESSOR CAN BE READILY INTEGRATED WITH VARIOUS SMART SENSORS FOR OEW DETECTION.

Neural Network Applications in Telecommunications

Joshua Alspector

Bellcore; Morristown, NJ

Neural Network Capabilities

- **Automatic and Organized Handling of Complex Information**
  - Where rules are not known or there are too many for an expert system, neural networks can encode knowledge by learning using training data. This is easy and can give good results quickly.
- **Adapts to Continuously Changing Environment**
  - The adaptive equalizer, a simple neural network, can maintain a high level of performance and is used in millions of modems.
- **Non–linear Modeling**
  - For control, equalization, or modeling of complex systems, neural networks are inherently non–linear in their fit to data. They can give results superior to traditional linear methods.
- **Parallel Implementation**
  - For problems where the data rate is too high for serial processing, parallel hardware based neural nets can provide dramatic speedups.

Bellcore Work on Applications

- **Proprietary Neural Network Hardware**
  - Adaptive equalization
  - ATM admission control
  - Optimization (Packet routing, Channel assignment, Multi-user detection)
- **New Services**
  - Adaptive user interface for information filtering
  - Financial and market prediction
  - Auditory localization for multipoint teleconferencing
- **Operations**
  - Fraud detection
  - Traffic characterization for differential billing
  - Fault identification
  - Software reliability prediction

Learning Chip Computational Function

Current summation

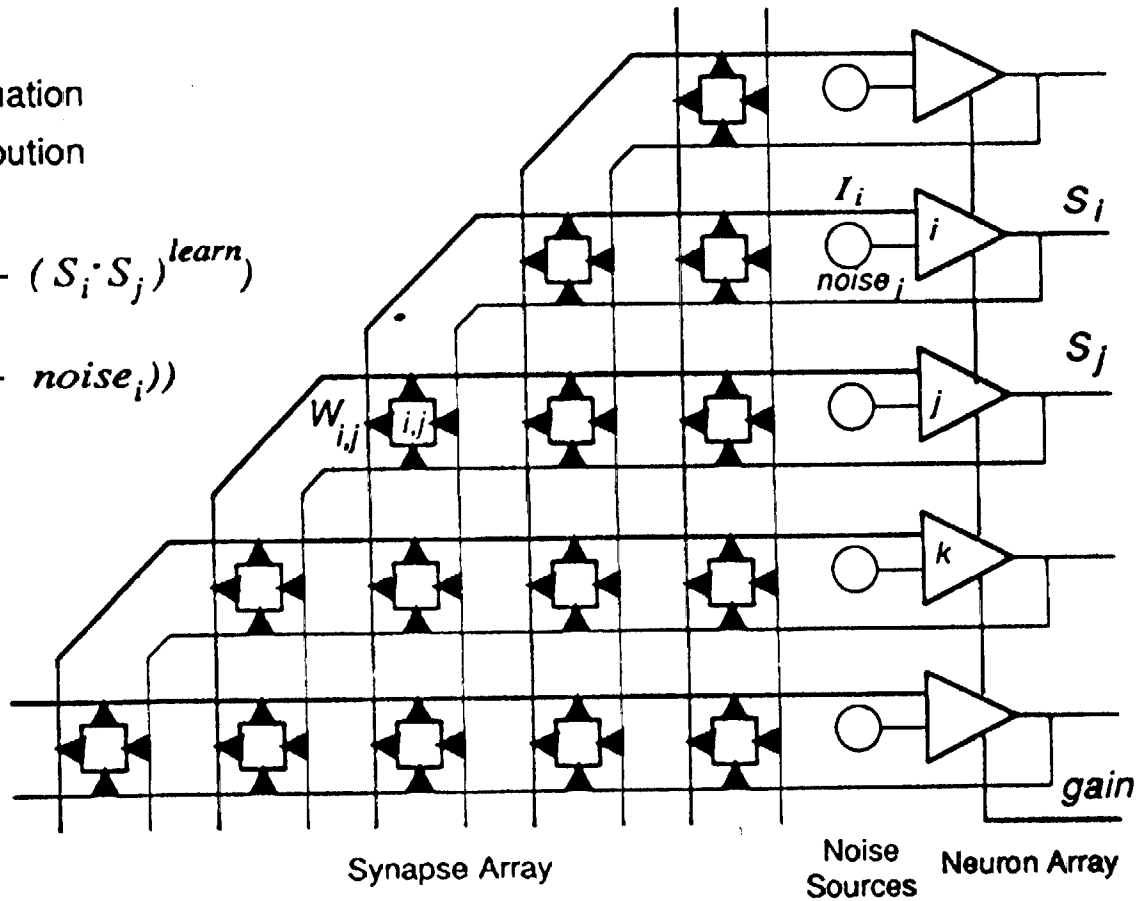
Voltage distribution

$$\Delta W_{ij} = sgn((S_i \cdot S_j)^{teach} - (S_i \cdot S_j)^{learn})$$

$$S_i = f(gain \cdot (netin_i + noise_i))$$

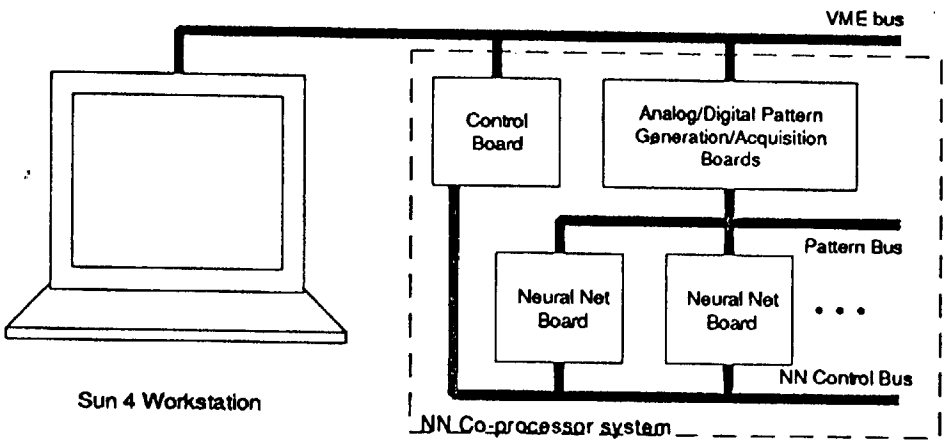
$$I_{ij} = W_{ij} S_j$$

$$I_i = \sum_j W_{ij} S_j$$



Bellcore
Bell Communications Research

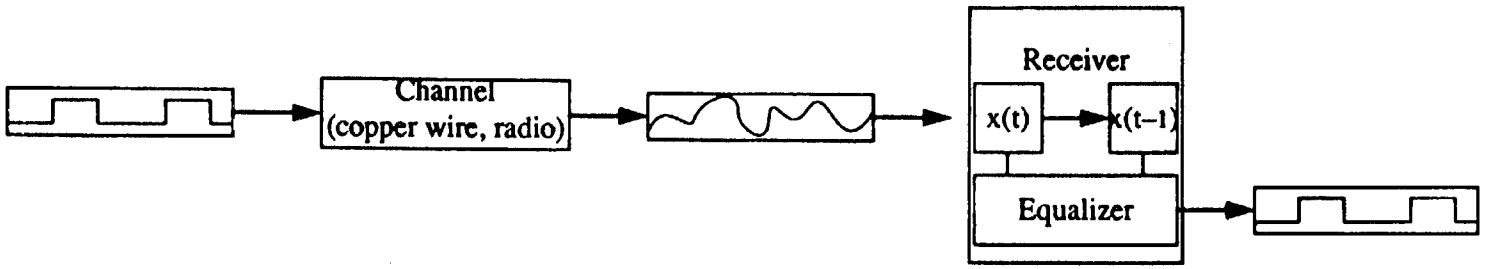
Learning System - Block Diagram



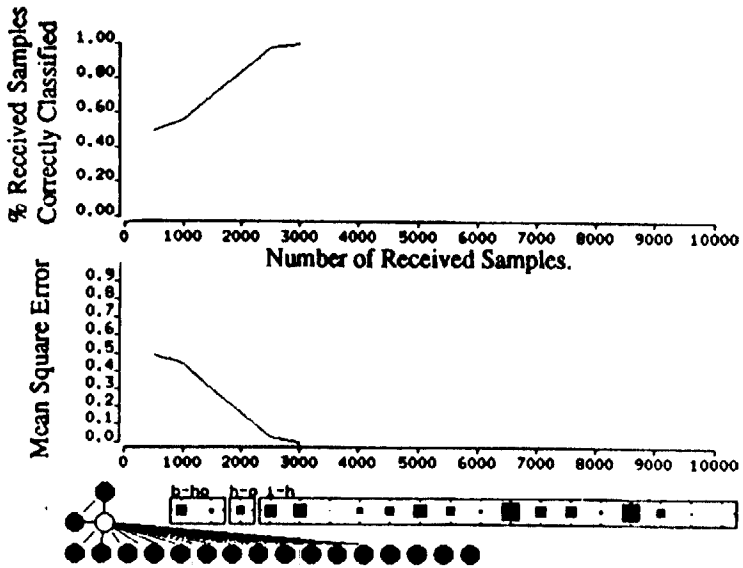
Bellcore
Bell Communications Research

Neural Network Equalization

Equalization



Neural Hardware

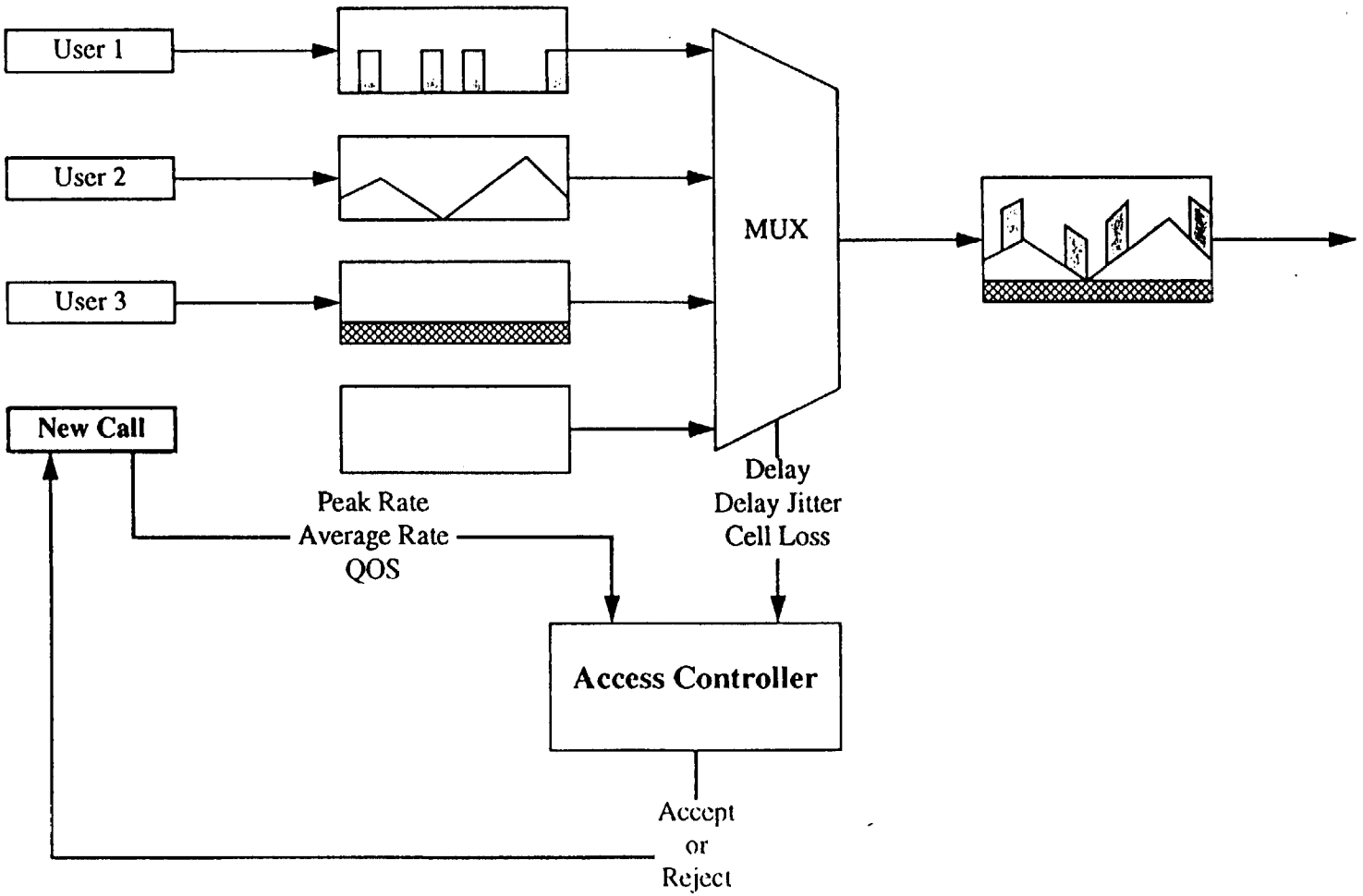


Processing Speed (samples/second)	
Current Test Platform	10,000
Limit of Current Chip	100,000
Chip Dedicated to EQ	1,000,000

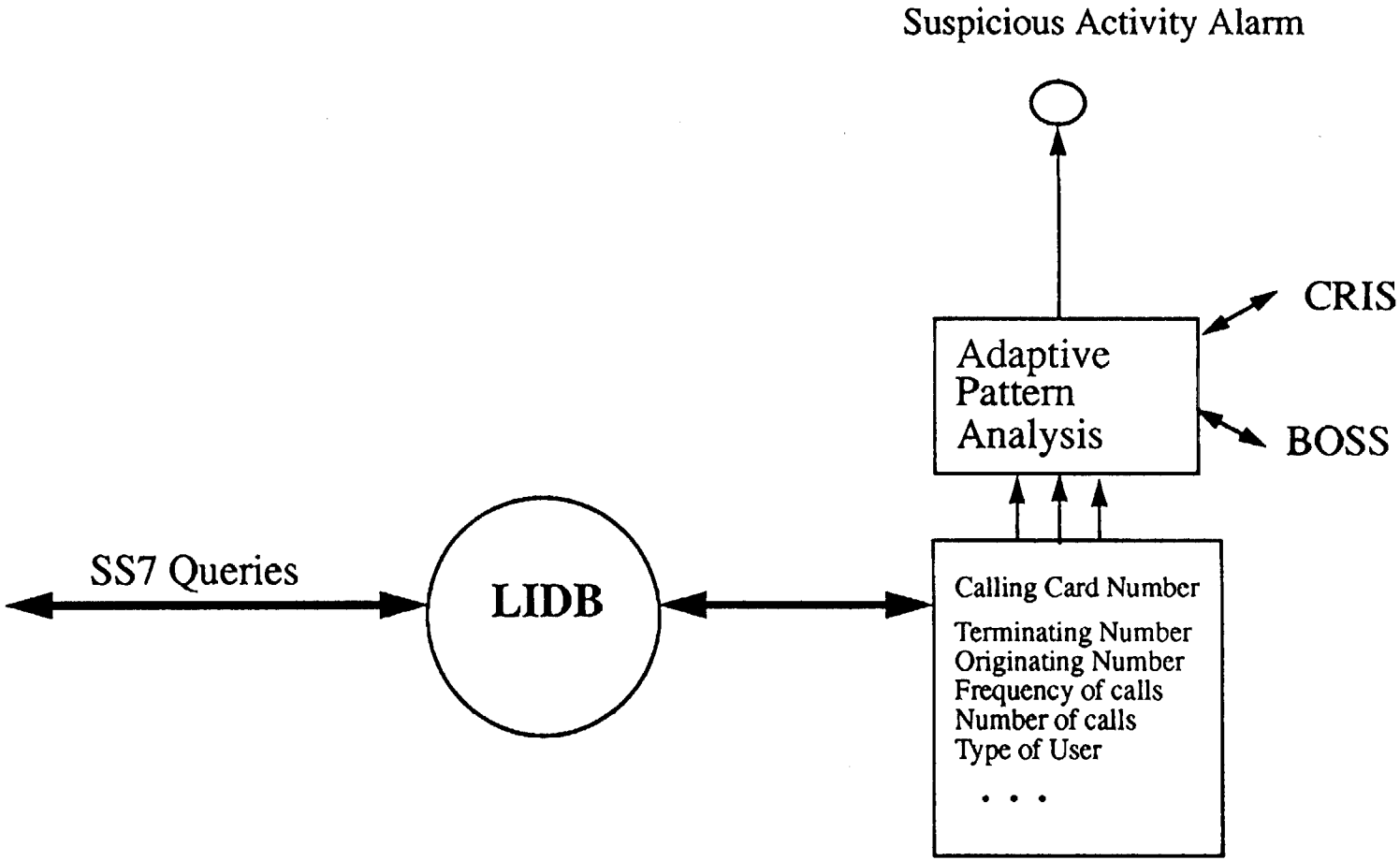
Analog neural network uses 20 times less power than similar speed digital.

Bellcore
Bell Communications Research

Broadband Access Control



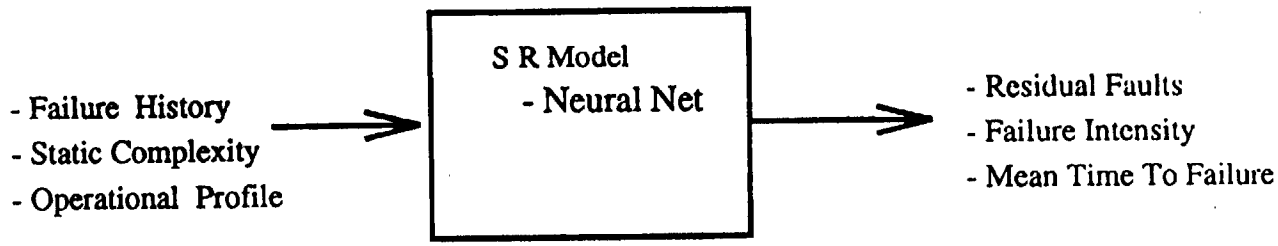
Calling-Card Fraud Detection

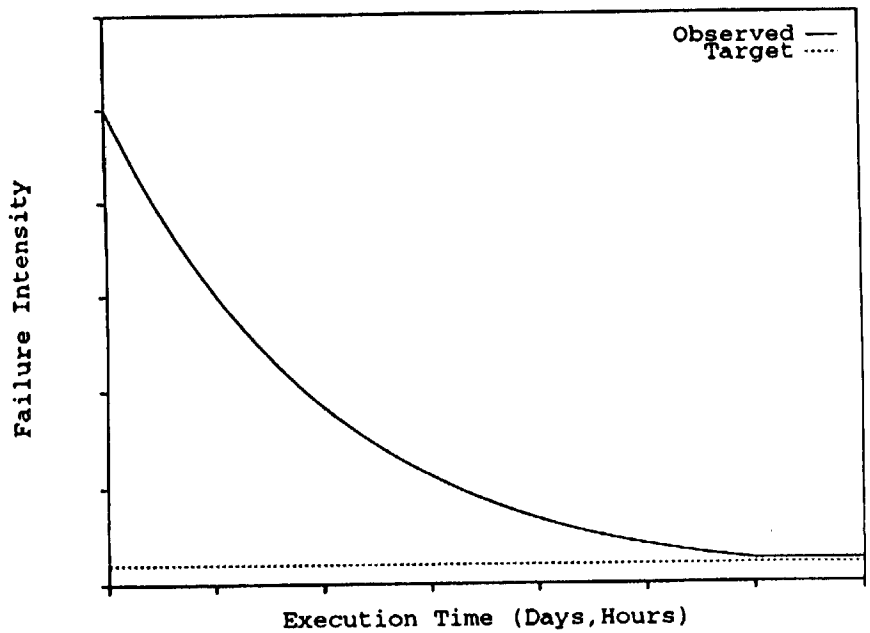


Bellcore
Bell Communications Research

Software Reliability Prediction (Cont.)

An Overview :





Conclusions

- **Quality**
  - Adaptive equalization
  - Fault identification
  - Software reliability
- **Efficiency**
  - Coding, compression
  - Routing, scheduling
- **Operations**
  - Network management
- **Interfaces**
  - Speech and pattern recognition
  - Adaptive information filters
- **Communication systems and their customers benefit from adaptive and intelligent systems**

A Neural Network Controller for Automated Composite Manufacturing

Peter F. Lichtenwalner

McDonnell Douglas Aerospace
New Aircraft and Missile Products
P.O. Box 516, St. Louis, MO 63166
(314) 233-7014
pete@aicenter.mdc.com

At McDonnell Douglas Aerospace (MDA), an artificial neural network based control system has been developed and implemented to control laser heating for the fiber placement composite manufacturing process. This neurocontroller learns an approximate inverse model of the process on-line to provide performance that improves with experience and exceeds that of conventional feedback control techniques. When untrained, the control system behaves as a proportional plus integral (PI) controller. However after learning from experience, the neural network feedforward control module provides control signals that greatly improve temperature tracking performance. Faster convergence to new temperature set points and reduced temperature deviation due to changing feed rate have been demonstrated on the machine. A Cerebellar Model Articulation Controller (CMAC) network is used for inverse modeling because of its rapid learning performance. This control system is implemented in an IBM compatible 386 PC with an A/D board interface to the machine.

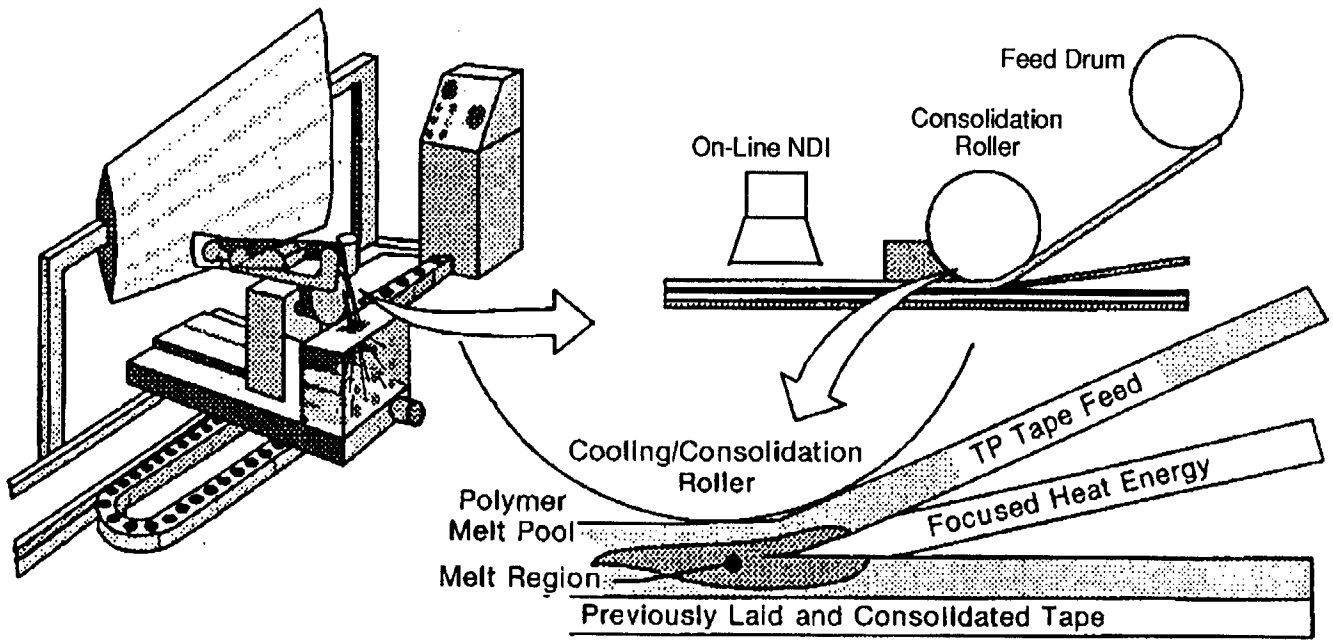
A Neural Network Controller for Automated Composite Manufacturing

JPL Neural Network Workshop
A Decade of Neural Networks:
Practical Applications and Prospects

May 11 - 13, 1994

Peter F. Lichtenwalner
New Aircraft and Missile Products
McDonnell Douglas Aerospace
(314) 233-7014

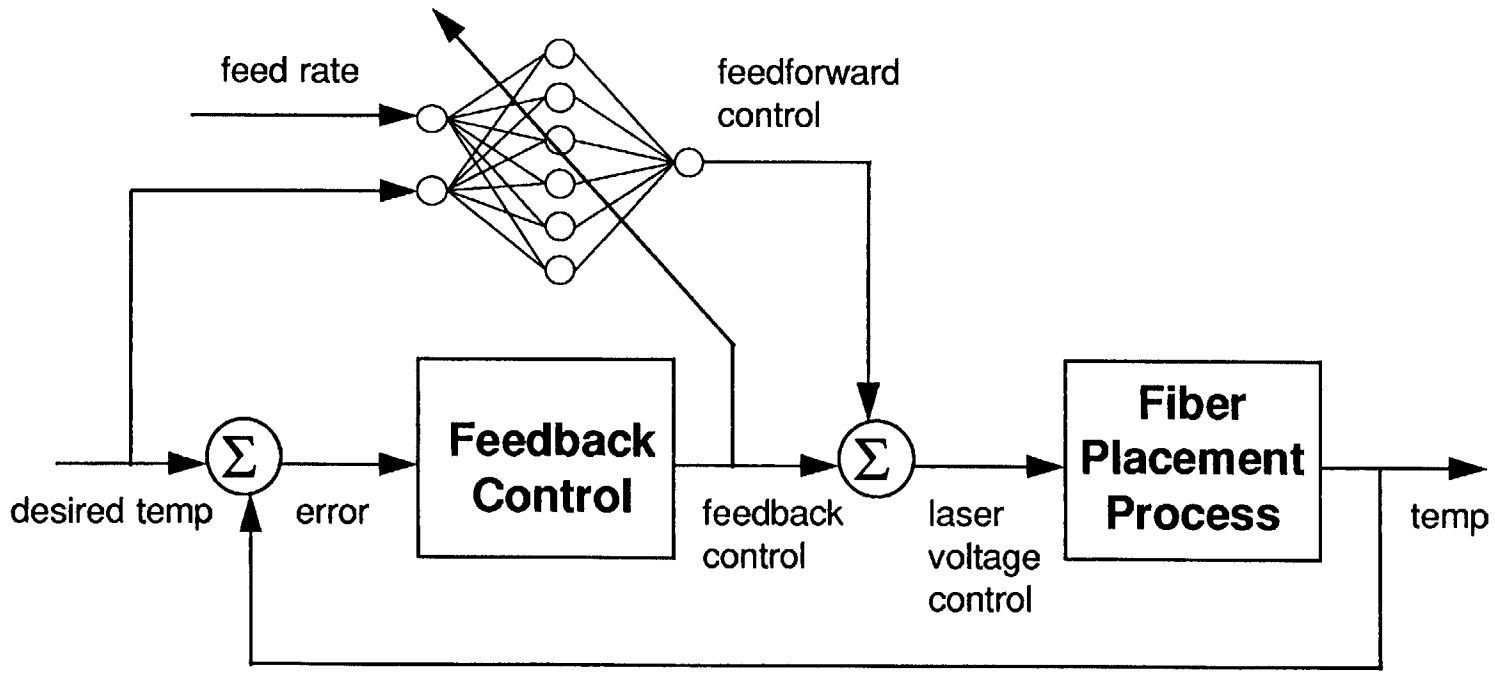
Thermoplastic Composite Fiber Placement Manufacturing Process



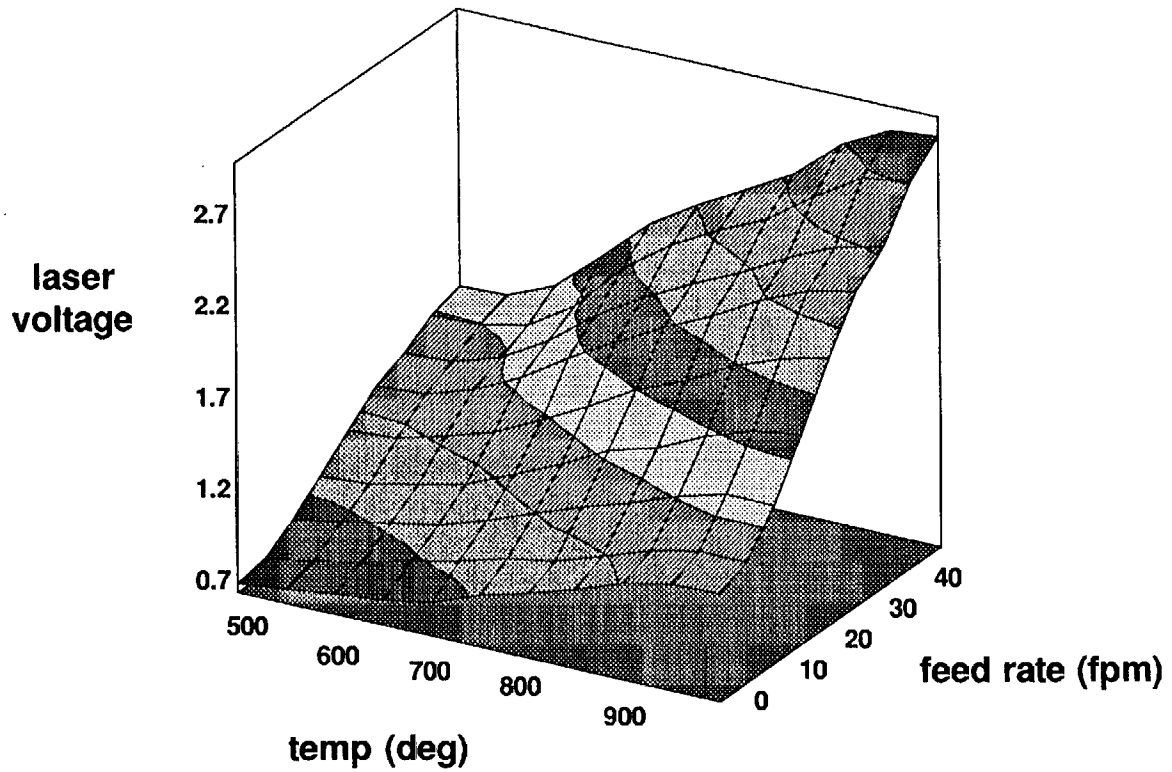
Features of Fiber Placement Process

- **Thermoplastic In-Situ Consolidation Eliminates:**
  - Manual Labor Intensive Material Lay-up
  - Vacuum Bag & Debulk
  - Autoclave
- **Potential for Every Layer Inspection**
  - Reduces Post-Process Inspection
  - Allows On-Line Repair
- **Accurate Process Control is Critical for Quality**
  - Complex Part Geometries Require Intelligent Control

On-Line Learning Neural Control for Fiber Placement Laser Heating

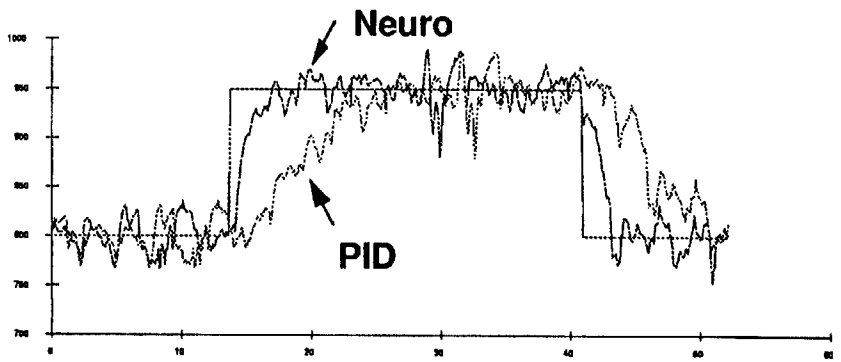


Control Law Learned by Neural Network

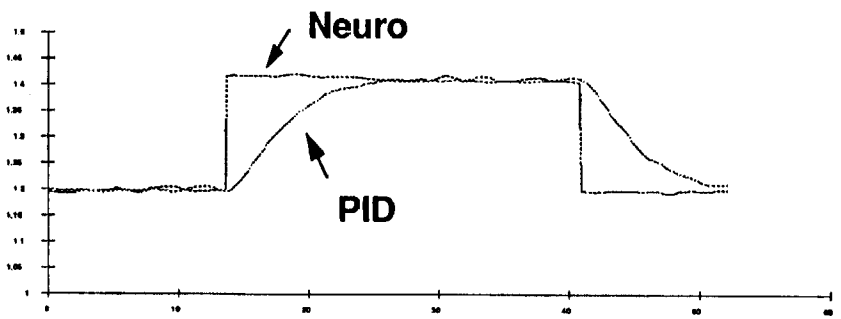


Benefits of Neural Control over PID Control

Faster Convergence to New Temperature Set Points

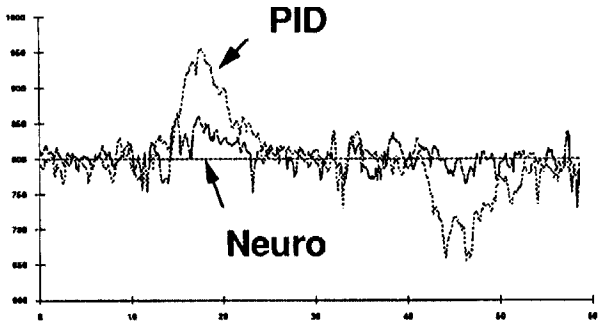


Control Voltage

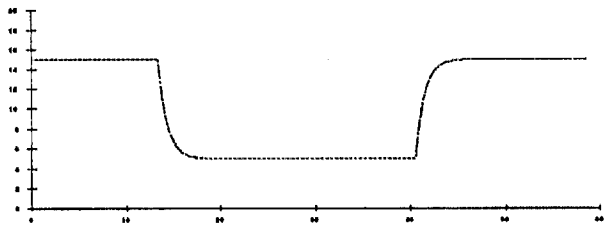


Benefits of Neural Control over PID Control

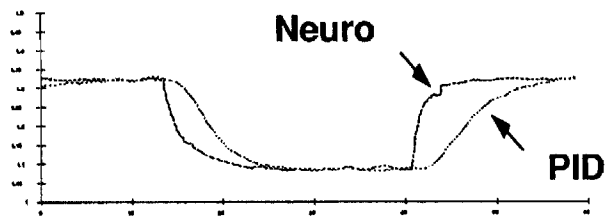
Reduced Temperature Deviation due to Feed Rate Changes



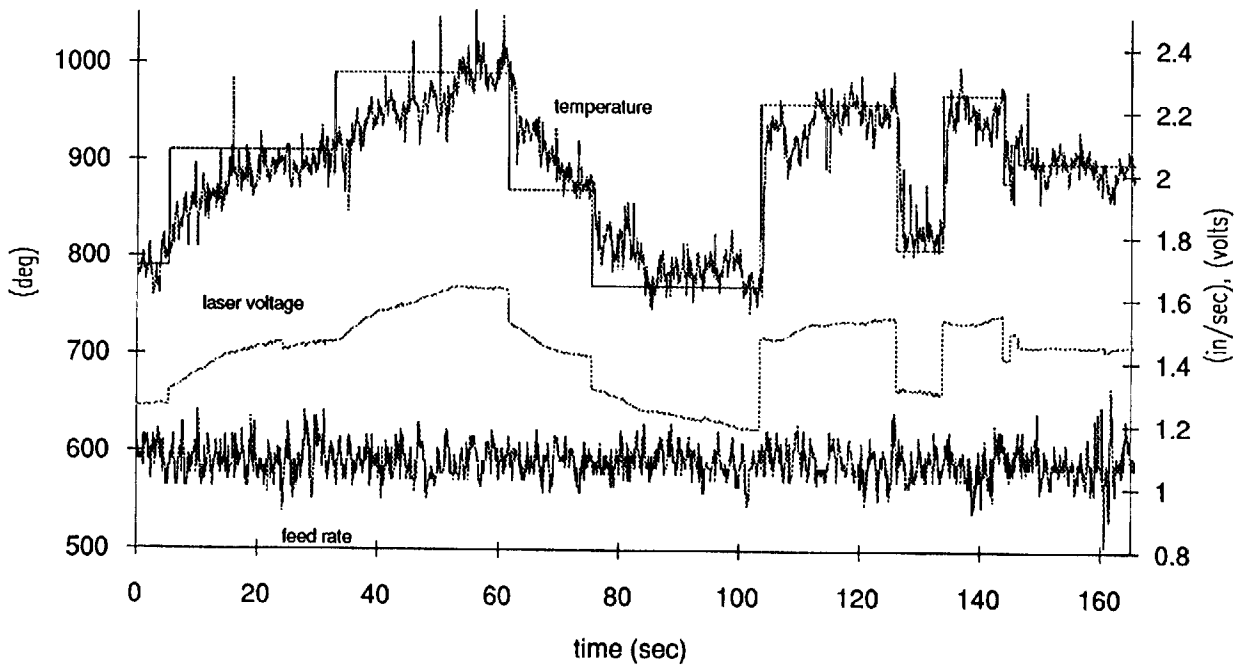
Feed Rate



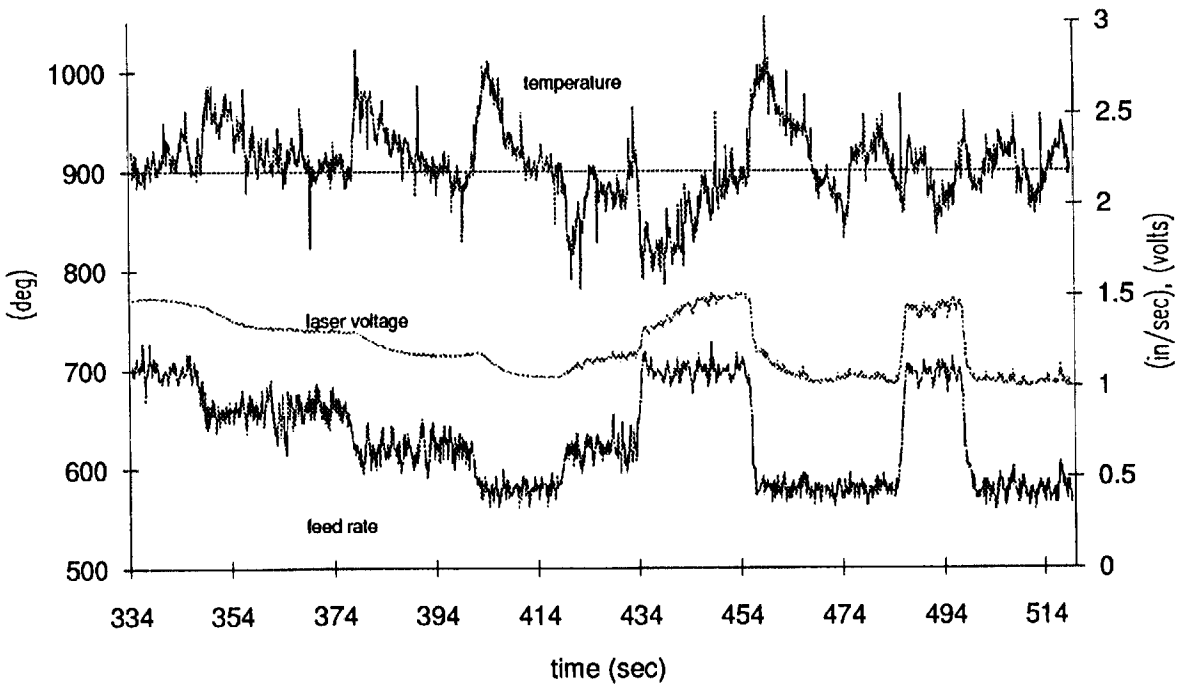
Control Voltage



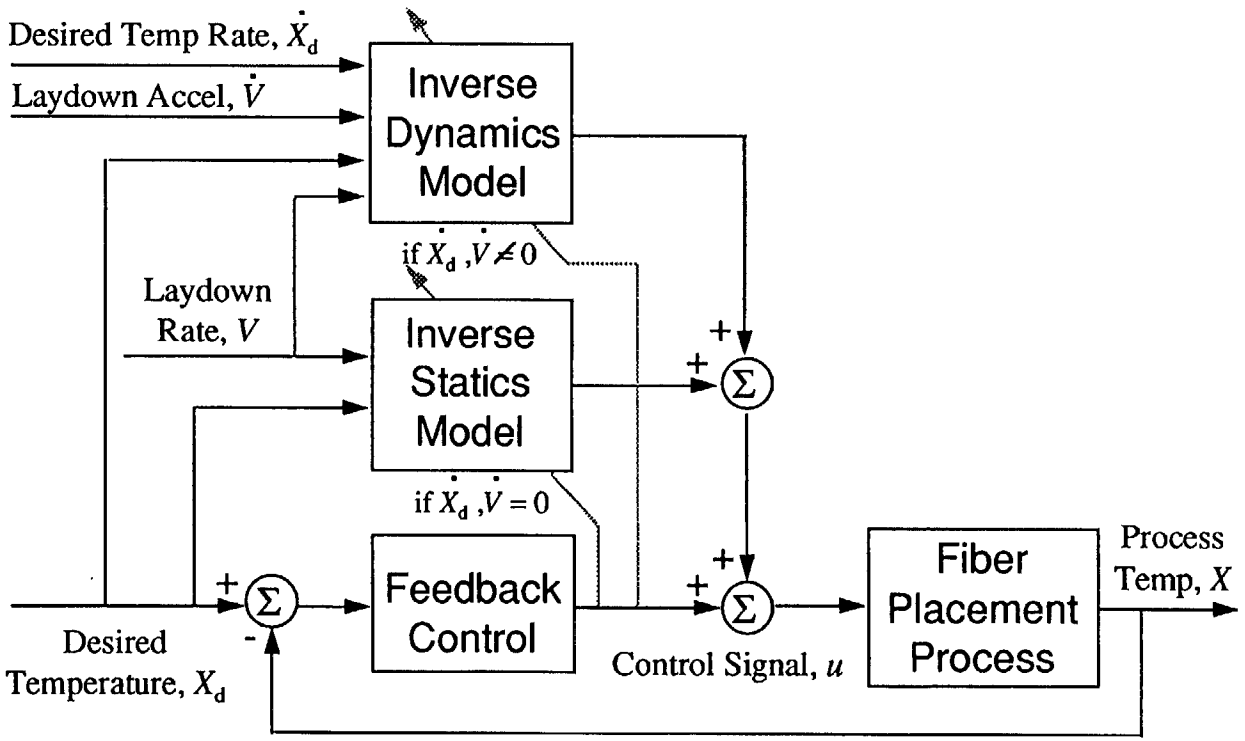
Neural Control Temperature Learning
(Fiber Placement R&D Facility, 4/1/92)



Neural Control Feed Rate Learning
(Fiber Placement R&D Facility, 4/1/92)



Architecture for Improved Dynamics Learning & Compensation



Future Work with Neural Networks

- **Enhance Fiber Placement Process Control**
  - Utilize Dynamics & Geometry Information
  - Implement On-Line Inspection Sensing
  - Integrate Material Process Modeling
- **Intelligent Control for High Speed Machining**
  - On-Line Learning for Precision Path Control
- **Smart Structure Applications**
  - Active Vibration Control for Flutter and Buffet

How Captain Amerika Uses Neural Networks to Fight Crime

Steven K. Rogers, Matthew Kabrisky, Dennis W. Ruck and Mark E. Oxley

Air Force Institute of Technology
Department of Electrical and Computer Engineering
2950 P Street, Wright-Patterson AFB, OH 45433-7765
15 February 1994

Abstract

Artificial neural networks models can make amazing computations (some of which are applicable to fighting crime: recognition of faces; speaker identification; fingerprint recognition). Those models will be explained along with the application of those models into problems associated with fighting crime. Specific problems addressed are identification of people using face recognition, speaker identification as well as fingerprint and handwriting analysis (biometric authentication).

I Introduction

Before getting started it is common to explain the Captain Amerika connection. Captain America comic books describe the superhero as: "born in the U.S.A," that obviously applies to the authors; "endowed with a superhuman physique," once you see the authors at the conference you will make the obvious connection with this point; and finally "fights an ongoing battle for liberty, justice, and the American dream!", who needs Ross Perot? Oh, by the way, you might also notice in the comic book that Captain America's secret identity is "Steve Rogers". The "k" in Captain Amerika is just a copyright infringement worry of that author.

This lecture covers the application of artificial neural network techniques for fighting crime. For example the image of a suspect might be provided to some law enforcement agency for processing, possibly to recognize the person in the image. Image processing usually consist of three stages. The first is the location of regions of interest within the image (segmentation-find the face). The second step is the extraction of a set of numbers which characterize the regions that are extracted (feature extraction-describe the face). The last step is the processing of the features for decision making (classification-decide who it is).

II Crime Fighting Problems

An enormous part of crime fighting is recognition of faces. We will use this problem to demonstrate the application of artificial neural networks to real world problems. During the lecture other problems like fingerprint identification, speaker identification and handwriting analysis will also be addressed. From automatic mugshot matching to border crossing monitoring, law enforcement agencies need an autonomous

face recognition capability. Such a system could also be used to verify users of automatic teller machine cards, or control of login into sensitive computer systems. This capability has also been used to interface handicapped people to computers. To be honest this last application is the one that our group is the most excited about. In this case a young Chicago lady (13 years old) who has cerebral palsy was interfaced to her personal computer by recognizing her facial expressions.

III Segmentation

The finding of regions of interest in an image is called segmentation-find the face in the image. Any errors in this step are preferred to be false acceptance, (passing pixels that may not contain parts of the face), but not false negatives (miss regions that might contain parts of the face). The same concept applies to processing sound. For example, when trying to identify a speaker's voice, sound is recorded. The parts of the recording that need to be identified must be segmented from the rest of the recording. To be of any benefit, this step must significantly reduce the number of pixels or periods of the recording that the next steps of feature extraction and classification must deal with. The processing of the raw pixels to find the regions that might contain the face may be the toughest of the image processing stages. To reduce the amount of computation necessary for the subsequent processing the system should only look in those regions of space, time, frequency, intensity or texture where the face is likely to be located. A one-pass segmentation algorithm filters the raw data to eliminate obvious nonface regions (a function of neighborhood calculations).

Before feature extraction, image preprocessing is usually necessary. The most common preprocessing is some form of energy normalization. The preprocessing is necessary because images have characteristically low contrast and lots of irrelevant structure. To be effective for real world images, the energy normalization is usually based on local neighborhood information. Most segmentation techniques are based on morphological operations, texture analysis and local intensity comparisons or spatial frequency information processing that allow discrimination of regions of interest from the rest of the pixels.

Single neurons can be probed by electrodes and stimulus response measurements made. The results of such measurements show that the system cares about local orientation information and motion direction. Similar more recent measurements have expanded this idea to localized texture information as being the critical first step. To get information from multiple locations, radioactive dyes have been used and clearly show the mapping of the real world onto the visual cortex. One problem with these experiments is that the animal has to volunteer to have its metabolism reduced to zero for the measurements. Only volunteer animals are used of course. Using VLSI technology, multiplexed array cortical electrodes have recently been made and implanted directly onto cortex.

IV Feature Extraction

The processing of the data to extract a set of measurements (describe the face) that represent the gestalt of the information required to decide who is in the image is called feature extraction. There can be no information gained by this step; its purpose is to increase the ratio of pertinent information to irrelevant data. If a perfect classification stage could be accomplished on the raw data, it would achieve the lowest error possible. But, in the problems of interest here, image processing for face recognition, the processing of the raw data (the original images) is not always feasible. The dimensionality alone of such a task make it not an option for some applications. For each region of interest segmented, a set of features must be found to represent the region for classification.

There are several popular methods for obtaining the features to be used. The first is to ask experts in the field of interest. For example in the problem of target recognition some common features include: length-to-width ratio; hot spot intensity; or complexity. Similarly, relevant expert extracted features are used in face recognition, such as the distance between anthropometrically significant features. The distance between the eyes or from the bridge of the nose to the chin. No one believes that computer aides for recognition are useful if human extracted features have to be keyed in. Finding the important parts of the face by using artificial neural networks is a key first step.

The second alternative is to have the segmented regions processed directly by the neural feature extractor. One common neural feature extraction technique uses a layer of artificial neurons with receptive fields in the input raw data. This is similar to the processing discovered in visual striate cortex, V1. The Nobel Prize winning results of Hubel and Wiesel clearly demonstrated that orientation selectivity and motion direction selectivity within the receptive field of a striate neuron exists. The weights for these artificial neurons are either found using a gradient search based learning algorithm, hardwired based on some a priori knowledge (such as a Hubel and Wiesel or the later work of Jones and Palmer) of types of feature extraction that might be useful.

Quite often after classification, questions are asked about which features caused a particular decision to be made. That is, the question of why a particular region of a photograph was called President Clinton and another called Ross Perot. It's not the shoes. It's got to be the ears! A related question is: of the many features that may have been suggested as useful for a given problem which ones are the most important ones for the task of interest? The answer to this question is often used to reduce the set of feature measurements (vector) to a smaller dimension. This is critical in applications where there are only a limited amount of training data available. To reduce the feature vector, the most common statistical and trial-and-error techniques have been augmented with neural feature saliency techniques. Conventional statistical correlation ideas are the most common technique to find how features are related. The discovery of nonobvious relationships between features may be one of the great contributions of neural networks. One of the early applications of neural networks was in loan analysis. The data on the application for the loan were fed into a neural network and the network that had been

trained on historical data on loan defaults would predict whether you would default. For litigation reasons the users of such networks had to be able to determine the application information that the network considered to be the indicator of you eventually defaulting. There also currently exists artificial neural network systems that monitor credit card transactions to detect fraud. They are trained on historical transaction data and analyze current transactions to detect fraudulent transactions.

As a side note, using the biological insight a good set of candidate features can often be found. In the application of speaker identification, measurements of the processing of the pinna and frequency extraction as a function of distance along the cochlea have resulted in models that have been demonstrated useful in sound localization and speaker identification.

V Classification

Once the features that are to be used to decide whether a particular region of interest requires further attention are extracted, they are submitted to the classification stage. This is the area where neural techniques have proven to be most useful. The most common neural techniques require an enormous amount of labeled data. Labeled data has to be hand labeled by experts. It is the experience of these experts that the classification step must learn to encode in the interconnection weights. In the application of face recognition, some expert must feed the network with images and tell the network the identity of the face. Similarly, someone must identify the voice from a training recording before the system can identify the person from a later recording.

It has been proven many times in the literature that the common neural techniques perform as approximators of the Bayes optimal decision elements (minimum probability of error). This allows the user to know that if correctly engineered there are no first order statistical techniques which will outperform the neural algorithms with respect to accuracy. Even with this knowledge the comparison of the neural classification algorithms with statistical techniques such as regression or quadratic discriminant function analysis is useful to ensure that the neural technique is correctly engineered.

VI Future Work

The most important future area of research is in field test and demonstration. Large scale tests will determine whether anything useful will come out of the preliminary exciting results. It will only be by statistically significant improvement in real world applications such as crime fighting that this technology will be proven.

Fundamental work on generalization predictions is also necessary. The question is how much datawill be required in a given application to allow the system to be fielded with some confidence on how well it will perform. How much shrinkage should be expected from the accuracy rate seen in training to the rate that is expected in the real world.

The combination of neural with fuzzy and expert system techniques will also play a key role in driving these solutions to useful applications. Joint conferences, such as the IEEE World Congress on Computational Intelligence, may allow a quick improvement in this area.

One of the most interesting areas of research is in consciousness. Real brains, of course, think about being real brains. The idea of self-awareness as a computation going on within your brain is controversial but true. How does a piece of meat think about being a piece of meat? Could meat ever understand how it does it? Why does human meat seem to be different from that of other animals even though all mammalian brains are constructed to the same basic plan using the same basic parts? There are fundamental limits to the computational capability of the human brain. One way to see the limitations is by the concept of Miller's magical number seven plus or minus two. The human brain is limited to keeping track of about seven things. If keeping track of more than seven things is required to build a stable world society then we have a problem. In the context of this lecture if more "chunks" (more than seven) are required to understand self-awareness then we will never understand how we do it. A puppy dog has fewer chunks than the seven. How many does a chimp have? How can we measure the number of "chunks" for nonverbal animals or if they also can compute their own existence? Series of delay-non-matching-to-sample tests may work here.

The illusion of self awareness is aided and abetted by a series of tricks and lies perpetrated by the human sensory systems; the world is not quite the way it looks, not at all the way it sounds, and the sense of the flow of time is a total confabulation which runs about 200 milliseconds behind real time. The purpose of the brain is to construct as accurate a model of the world as it can given the inevitable limitations of being made out of meat. The results, though, are really amazing; we live inside our own private bags of life which are equipped with a seemingly high fidelity stereo sound system, a 3-dimensional movie display and complete cognizance of touch and smell. We have an enormous content-addressable memory and can keep track of about seven things simultaneously. We can manipulate arbitrary symbols and create the illusion that we are aware of our own existence (and thus compute that it will someday end). Some of the neural hardware forming the sensory systems was described in this lecture but a complete description of how it all works does not exist nor is there any reason to imagine that a human brain could understand it if it did.

VII Conclusions

It has been shown in several areas that artificial neural networks can make a significant impact in fighting crime. The biometric authentication systems are being fielded. The application of neural technology to other crime-related problems is necessary. This will require a joint effort between experts in the law enforcement area with signal processing people. Participation at the professional meetings of each group by the other is critical.

NOTES

NOTES

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No. 94-10	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle A Decade of Neural Networks: Practical Applications and Prospects		5. Report Date May 11, 1994 6. Performing Organization Code
7. Author(s) Sabrina C. Kemeny		8. Performing Organization Report No.
9. Performing Organization Name and Address JET PROPULSION LABORATORY California Institute of Technology 4800 Oak Grove Drive Pasadena, California 91109		10. Work Unit No. 11. Contract or Grant No. NAS7-918 13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address NATIONAL AERONAUTICS AND SPACE ADMINISTRATION Washington, D.C. 20546		14. Sponsoring Agency Code RF 238 0001 PX-644-11-00-00-0

15. Supplementary Notes

16. Abstract

The Jet Propulsion Laboratory Neural Network Workshop, sponsored by NASA and DoD, brings together sponsoring agencies, active researchers, and the user community to formulate a vision for the next decade of neural network research and application prospects. While the speed and computing power of microprocessors continue to grow at an ever-increasing pace, the demand to intelligently and adaptively deal with the complex, fuzzy, and often ill-defined world around us remains to a large extent unaddressed. Powerful, highly parallel computing paradigms such as neural networks promise to have a major impact in addressing these needs. Papers in the workshop proceedings highlight benefits of neural networks in real-world applications compared to conventional computing techniques. Topics include fault diagnosis, pattern recognition, and multiparameter optimization.

17. Key Words (Selected by Author(s))	18. Distribution Statement
Electronics and Electrical Engineering, Mathematical and Computer Sciences, Computer Systems, Information Theory	

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	235	