

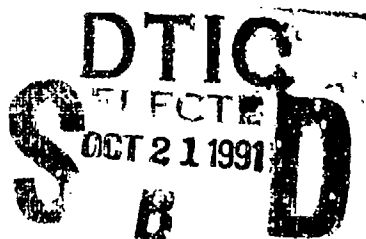
2

AD-A241 765



**Technical Report
926**

Neural Net Sensor Fusion



**R.Y. Levine
T.S. Khoun**

5 September 1991

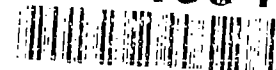
Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS



Prepared for the Department of the Army under
Air Force Contract F19625-90-C-0002.

Approved for public release; distribution is unlimited.

91-13543



This report is based on studies performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. The work was sponsored by the U.S. Army Strategic Defense Command, Department of the Army under Air Force Contract F19628-90-C-0002.

This report may be reproduced to satisfy needs of U.S. Government agencies.

The ESD Public Affairs Office has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Hugh L. Southall, Lt. Col., USAF
Chief, ESD Lincoln Laboratory Project Office

Non-Lincoln Recipients

PLEASE DO NOT RETURN

Permission is given to destroy this document
when it is no longer needed.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

NEURAL NET SENSOR FUSION

R.Y. LEVINE
T.S. KHUON
Group 93

TECHNICAL REPORT 926

5 SEPTEMBER 1991

Approved for public release; distribution is unlimited.

LEXINGTON

MASSACHUSETTS

ABSTRACT

A generic architecture for neural net multisensor data fusion is introduced and analyzed. The architecture consists of a set of independent sensor neural nets, one for each sensor, coupled to a fusion net. Each sensor is trained (from a representative data set of the particular sensor) to map to a hypothesis space output. The decision outputs from the sensor nets are used to train the fusion net to an overall decision. In this report the sensor fusion architecture is applied to the stochastic exclusive-or problem for a benchmark comparison with classical hypothesis testing. The architecture is also applied to a data fusion experiment involving the multi-sensor observation of object deployments during the recent Firefly launches. The deployments were measured simultaneously by X- and L-band and CO₂ laser radars. The range-Doppler images from the X-band and CO₂ laser radars were combined with a passive-IR spectral simulation of the deployment to form the data inputs to the neural sensor fusion system. The network was trained to distinguish pre-deployment, deployment, and postdeployment phases of the launch based on the fusion of these sensors. The success of the system in utilizing sensor synergism for an enhanced deployment detection is clearly demonstrated.

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

ACKNOWLEDGMENTS

We would like to thank Mitch Eggers for providing some of the references and for noting that the optimum FNN is a neural net. Hitoshi Inada and Ken Schultz generated and processed the Haystack and Firepond Firefly data, respectively. Both individuals are also thanked for providing analysis of the Firefly maneuvers and details of image generation for the two radars. Mike Jordan, the author of the passive-IR simulator, is thanked for providing the information necessary to run the code and aiding our interpretation of the output. The reading of the manuscript by Sun Levine and Israel Kupiec is very much appreciated.

TABLE OF CONTENTS

Abstract	iii
Acknowledgments	v
List of Illustrations	ix
1. INTRODUCTION	1
2. SXOR BENCHMARK FOR NEURAL NET DATA FUSION	5
2.1 False Alarm and Detection Probability for SXOR	5
2.2 Back Propagation Neural Net Performance	7
2.3 SXOR Data Fusion	13
3. FIREFLY SENSOR FUSION EXPERIMENT	17
3.1 Firefly Experiment	17
3.2 Firefly Sensor Fusion System	20
4. CONCLUSION	31
REFERENCES	33
APPENDIX A - VARIANCE TRANSITION DETECTION	37

LIST OF ILLUSTRATIONS

Figure No.		Page
1	Generic neural net sensor fusion architecture for distributed sensor processing.	2
2	Schematic of variance transition test mapping: stochastic exclusive-or (SXOR). Input N-sample window variances (χ_1, χ_2) .	3
3	Firefly experiment launch sequence: Phases I (predeployment), II (deployment), and III (postdeployment) for canister-payload and balloon-canister.	4
4	False alarm and detection probability versus threshold γ for SXOR test, with $\sigma_0 = 1$, $\sigma_1 = 2$, and $N = (a) 2, (b) 6, (c) 10$, and $(d) 20$.	7
5	False alarm and detection probability versus threshold γ for SXOR test, with $\sigma_0 = 1$, $\sigma_1 = 4$, and $N = (a) 2, (b) 6, (c) 10$, and $(d) 20$.	8
6	Back propagation neural net for hypothesis testing: P-vector input data and Q-vector hypothesis output, $(0, \dots, 0, 1, 0, \dots, 0) \mapsto H_i$.	9
7	Cost versus iteration curve for back propagation learning of the SXOR map: 2 input, 16 hidden layer, and 2 output neurons in a 100-element training set; $\sigma_0 = 1$, $\sigma_1 = 4$, $N=10$.	10
8	SXOR false alarm and detection probability versus window size N for back propagation NN and classical test: $\sigma_0 = 1$, $\sigma_1 = 2$.	11
9	SXOR false alarm and detection probability versus window size N for back propagation NN and classical test: $\sigma_0 = 1$, $\sigma_1 = 4$.	12
10	Fusion architecture for SXOR test: SNN1 ($\sigma_0 = 1$, $\sigma_1 = 2$), SNN2 ($\sigma_0 = 1$, $\sigma_1 = 4$), and FNN first-order perceptron.	14
11	Performance probabilities for the fusion of two SXOR trained SNNs: P_d , P_f , P_m , and P_{CH_0} versus window size N for $\sigma_1 = 2$ and $\sigma_1 = 4$ SNNs, optimum FNN, and back propagation FNN.	15
12	Passive-IR simulated spectrum over the range $[5 \mu, 25 \mu]$ in W/sr. Metallic object with 1-m ² cross section, $\epsilon = \alpha = 0$.	18
13	Passive-IR simulated spectrum over the range $[5 \mu, 25 \mu]$ in W/sr. Graybody object with 1-m ² cross section, $\epsilon = 0.75$, $\alpha = 0.9$.	19
14	Balloon-canister deployment sequence for FFI launch: predeployment, deployment, and postdeployment phases.	20

LIST OF ILLUSTRATIONS (Continued)

Figure No.		Page
15	Passive-IR simulated spectra $[5 \mu, 25 \mu]$ in W/sr for three balloon-canister phases in FFI launch.	21
16	Formulation of multisensor data fusion for balloon-canister deployment: Haystack and Firepond range-Doppler images and passive-IR simulation of predeployment, deployment, and postdeployment phases.	22
17	Distributed sensor fusion system for FFI balloon-canister deployment detection: Back propagation SNNs for passive-IR, Haystack, and Firepond sensors and BPFNN.	23
18	Fusion neural net cost function C versus training iterations for sensor fusion system training on FFI balloon-canister deployment detection.	25
19	SNN neuron outputs: Novel FFI balloon-canister deployment data for (a) predeployment, (b) deployment, and (c) postdeployment neurons.	26
20	FNN neuron outputs: Novel FFI balloon-canister deployment data for (a) predeployment, (b) deployment, and (c) postdeployment neurons.	27
21	SNN neuron outputs: Novel FFII canister-payload deployment data for (a) predeployment, (b) deployment, and (c) postdeployment neurons.	28
22	FNN neuron outputs: Novel FFII canister-payload deployment data for (a) predeployment, (b) deployment, and (c) postdeployment neurons.	29

1. INTRODUCTION

A highly successful intuitive architecture for hypothesis testing from fused multisensor data consists of distributed single-sensor processors coupled to a fusion processor for an overall decision. Each single-sensor processor outputs a decision based only on the individual sensor data, which forms the input to the fusion processor. Optimal signal processing in a distributed sensor environment based on statistical estimation and hypothesis testing techniques has been considered in Tenney and Sandell [1], Sadjadi [2], Chair and Varshney [3], Thomopoulos et al. [4], Atteson et al. [5], Reibman and Nolte [6], and Dasarathy [7]. As with any Bayesian approach to hypothesis testing, optimum tests for data fusion are a function of the probability distributions of the input data. The design of such tests often involve an assumed model for the observed phenomena to define the data distributions. Alternatively, data-adaptive hypothesis testing results in a test based only on a previously generated training data set [8]. The outcome of a data-adaptive test is estimated from the system performance on a performance set of generated data with known hypotheses. The averaged system performance is simply obtained by applying the testing to an ensemble of training and performance sets. A theoretical treatment of data-adaptive hypothesis testing, with performance estimates based on the statistics of the training set, is given in Levine and Khuon [9]. It should be emphasized that data-adaptive hypothesis testing, while avoiding an assumed model for the data, requires a representative training set for successful definition of the test.

This report applies a particular data-adaptive hypothesis test, the neural net, to the distributed sensor fusion architecture. Relative to the now-conventional neural net taxonomy [8,10], only mapping neural networks such as the multilayer perceptron [11] and back propagation [12-16] are considered. These nets differ from the association Hopfield-type [17,18] by applying supervised learning (adaption) toward the performance of a functional mapping without feedback [8]. In hypothesis testing the desired map is from the input data space to an output hypothesis space. Alternative neural net architectures, such as those employing Kohonen learning [8,19], attempt to store data distributions internally rather than directly performing the data input-hypothesis space output mapping. It has generally been found that neural net classifiers perform as well as conventional techniques on a variety of problems, including linear, Gaussian, and k-nearest neighbor algorithms [10,20-24]. More generally, neural nets have been configured to perform the maximum a posteriori probability [25] and maximum likelihood tests [26] for arbitrary input distributions.

Figure 1 is a generic architecture for distributed multisensor neural net data fusion consisting of a sensor neural net (SNN) for each detector simultaneously observing a stochastic phenomena. Each SNN is trained to the output decision space $\{H_1, \dots, H_Q\}$ from a training set consisting of the corresponding sensor data. The output of the SNN consists of a normalized vector $(a_1, \dots, a_Q)$ where the largest a_i determines the hypothesis H_i . After all SNNs are trained, an independent data set is propagated through the SNNs to form an input training set for the fusion neural net (FNN). The FNN input consists of an analog $Q \times M$ vector, corresponding to Q decisions for each of M sensors. The FNN output consists of the vector $(f_1, \dots, f_Q)$, such that the largest f_i implies an overall system decision for hypothesis H_i . Note that the FNN performs cluster analysis in the QM

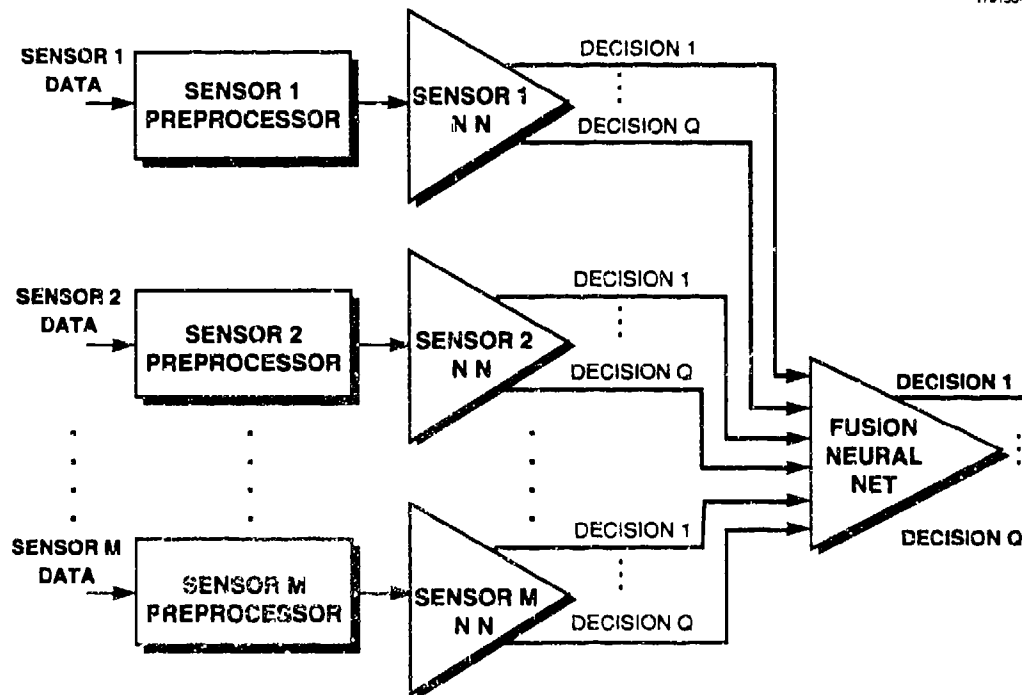


Figure 1. Generic neural net sensor fusion architecture for distributed sensor processing.

dimensional input space, which for hypothesis H_i is clustered to the vector $(0, \dots, 0, \overbrace{1}^i, 0, \dots, 0)$ for each of M sensors.

To motivate the neural net sensor fusion architecture in Figure 1, the system is applied to a problem for which a classical test is formulated. Section 2 discusses the neural net detection of a transition in the standard deviation of Gaussian noise. The process standard deviations before and after the supposed transition are assumed to be sensor dependent. The input to the SNNs consist of windowed sample variances from before and after the (supposed) transition. The mapping is from a χ^2 distributed pair (χ_1, χ_2) to a decision space output $(1, 0)$ for transition and $(0, 1)$ for no transition. Figure 2 is a schematic of the transition test mapping, which is denoted SXOR (for stochastic exclusive-or). It is easily shown that the test requires a classifier bilinear in χ_1 and χ_2 , which is implemented by a second order neural net algorithm [11]. In addition to requiring a nontrivial neural net, the variance transition problem is sufficiently tractable to allow an analytic solution for the classical test performance. False alarm and detection probabilities are expressed in terms of the threshold parameter used in the hypothesis test. An optimum threshold, corresponding roughly by definition to maximum detection and locally minimum false alarm probabilities, is computed

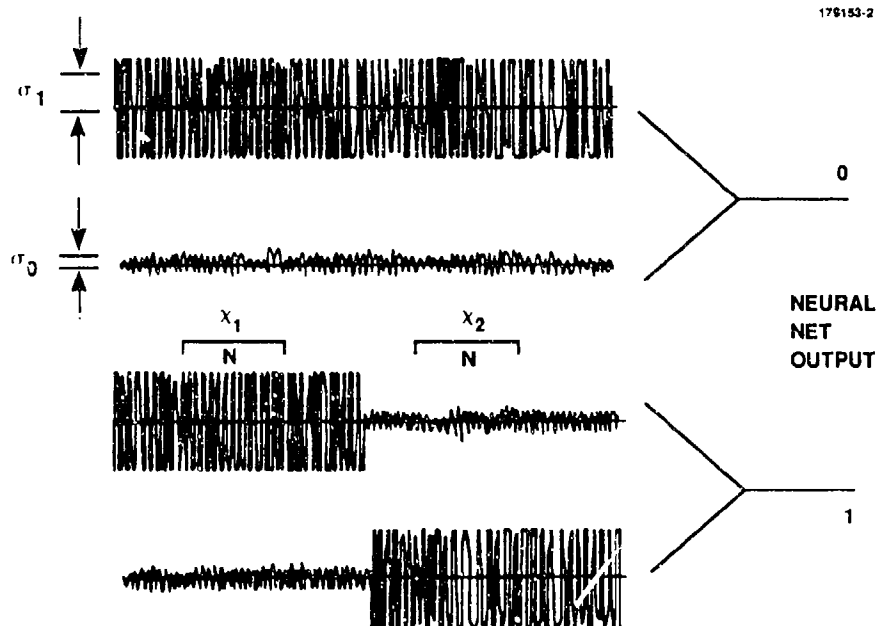


Figure 2. Schematic of variance transition test mapping: stochastic exclusive-or (SXOR). Input N -sample window variances (χ_1, χ_2) .

for a number of different noise and sampling conditions. Section 2 compares the performance of the sensor and fusion neural nets to the classical test probabilities. Chair and Varshney [3] show that optimum data fusion is implemented by a linear combination of the SNN decision outputs followed by the application of a threshold. The optimum weight vector for the linear combination is a function of the performance probabilities of the SNNs. This fusion algorithm is equivalent to a first-order perceptron for which the weight vector can be adapted by the perceptron learning algorithm [5,11]. The performance of the neural net sensor fusion system on the SXOR problem with back propagation SNNs and the optimum perceptron FNN are also discussed in Section 2. Motivated by the fact that the optimum fusion algorithm is a neural net, a back propagation FNN was trained on the SNN outputs. Both the optimum and back propagation FNNs matched or exceeded the higher performing SNN in the data fusion, justifying the use of neural networks in the distributed sensor fusion system.

Section 3 applies the fusion system architecture shown in Figure 1 to the detection of object deployments during the Firefly (FF) launches that occurred on 29 March (FFI) and 20 October 1990 (FFII) from Wallops Island, Virginia (as depicted in Figure 3). The launches presented a rare opportunity for data fusion due to the simultaneous observation by the three Millstone Hill (Westford, Massachusetts) radars: the Haystack X-band imaging, Firepond CO₂ laser, and

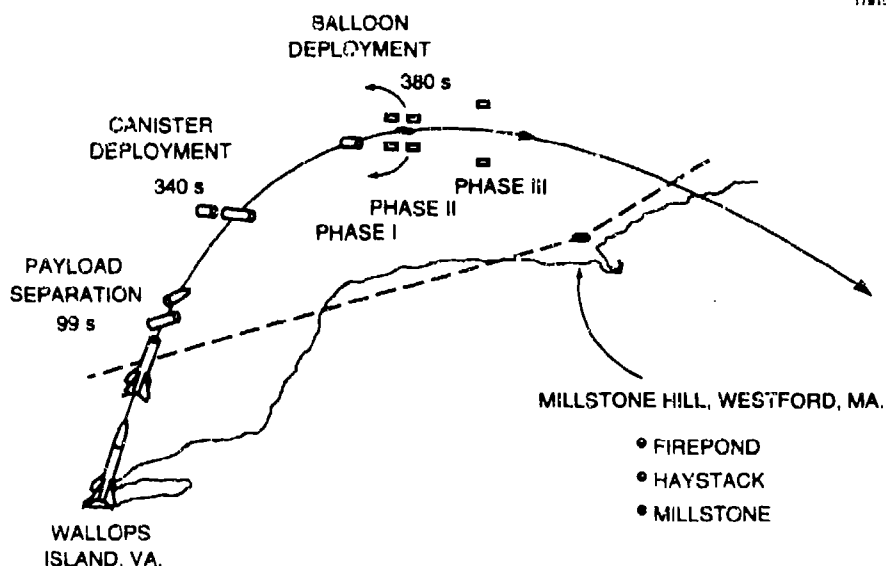


Figure 3. Firefly experiment launch sequence: Phases I (predeployment), II (deployment), and III (postdeployment) for canister-payload and balloon-canister.

Millstone L-band tracking. In applying the sensor fusion architecture to the FF data, two back propagation SNNs were trained on the deployments using range-Doppler images derived from the Haystack X-band and Firepond CO₂ laser radar data. A third SNN had as input the passive-IR spectral simulation of the deployments consisting of the spectral irradiance of the objects in the range $[5 \mu, 25 \mu]$. The range-Doppler images contained information of object segmentation, whereas the passive-IR simulation was sensitive to changes in the exposed object material composition. The sensor fusion system neural output consisted of a decision among the possibilities of predeployment (1,0,0), deployment (0,1,0), and postdeployment (0,0,1). In accordance with Figure 1, the FNN had nine inputs, three sensors with three possible decisions, and three output neurons for an overall deployment decision. The system is applied to deployment detection of an inflated balloon with training and performance data sets from the same launch (FFI). The performance of the entire sensor fusion system is compared to that of the SNN for each sensor to observe evidence of sensor synergism through data fusion. The application of the system to a canister deployment detection, in which the training and performance sets were taken from different launches (FFI and FFII, respectively), is also discussed.

Sections 2 and 3 contain the systematic theoretical and experimental analyses of neural net processing in the increasingly relevant distributed sensor environment. A conclusion follows in Section 4, and the appendix contains the Bayesian analysis of the SXOR test.

2. SXOR BENCHMARK FOR NEURAL NET DATA FUSION

This section considers a quantitative comparison of neural net and classical hypothesis testing in the distributed sensor architecture. The SXOR test map is interesting because it requires a nontrivial neural net implementing at least a second-order classifier and yet is mathematically tractable. In addition, the detection of noise deviation transitions reflects a common situation in nonstationary signal processing [27].

2.1 False Alarm and Detection Probability for SXOR

False alarm and detection probabilities are related to the threshold parameter of the SXOR test and the properties of the noise sampling. The sufficient statistic for a zero mean Gaussian process $\{y_i | i = 1, \dots, N\}$ is the sample variance [28]

$$\chi = \frac{1}{N} \sum_{i=1}^N (y_i - \bar{y})^2, \quad (1)$$

where $\bar{y}$ is the mean. The sample variance is χ^2 distributed with a probability density

$$p(\chi) = \frac{\chi^{\frac{N}{2}-1} \exp(-\frac{\chi}{2\sigma^2})}{2^{\frac{N}{2}} \sigma^N \Gamma(\frac{N}{2})}, \quad (2)$$

where σ is the standard deviation of the Gaussian random process $\{y_i\}$, and Γ is the gamma function. The classic test of distinguishing between two deviations, σ_0 and σ_1 , results from a threshold γ ; χ greater(less) than γ implies noise deviation $\sigma_1(\sigma_0)$.

The computation of performance probabilities for the SXOR requires the conditional probabilities $\{p[(i, j)|(q, m)] | i, j, q, m \in \{0, 1\}\}$ where each pair (i, j) corresponds to a (before, after) variance condition. The index i of 0 or 1 denotes a windowed sample variance from a low (σ_0) or high (σ_1) deviation process, respectively. The conditional probability $p[(i, j)|(q, m)]$ represents the detection of a noise condition (i, j) when the (before, after) windows truly correspond to the condition (q, m) . The hypothesis is tested on two data windows of length N from before and after the supposed variance transition. Assuming independent tests on each window, the conditional probabilities factor according to the equation $p[(i, j)|(q, m)] = p(i|q)p(j|m)$, where $p(j|m)$ denotes the probability of choosing noise deviation σ_j for a single window with deviation σ_m . The pair of decisions necessary to determine a transition is based on the value of χ in Equation (1) for two data windows and the threshold γ (as described above).

The appendix relates the false alarm and detection probabilities for variance transition detection to the conditional probabilities on a single window $p(j|m)$. The conditional probabilities for the transition hypothesis test are shown to be given by

$$P_d = p(\text{transition}|\text{transition}) = p(1|1)p(0|0) + p(0|1)p(1|0) \quad (3)$$

and

$$P_f = p(\text{transition}|\text{no transition}) = p(1|1)p(0|1) + p(1|0)p(0|0), \quad (4)$$

where it is assumed that the four possible noise conditions $\{(i,j)|i,j \in \{0,1\}\}$ have equal prior probability. The conditional probabilities appearing in Equations (3) and (4) are given by

$$p(1|i) = \int_{\gamma}^{\infty} p_i(x) dx \quad (5)$$

and

$$p(0|i) = \int_0^{\gamma} p_i(x) dx, \quad (6)$$

where p_i is the function $p(\chi)$ in Equation (1) with σ equal to σ_i .

As the threshold is varied, the behavior of P_d and P_f in Equations (3) and (4) characterizes the hypothesis test [28], which for this problem is the determination of a high/low or low/high variance transition. The test is a stochastic version of the binary exclusive-or map, which has historically been important in neural net research [11,12]. The central importance of this map derives from the concept of linear separability [11]. Embedding the input sample variances (χ_1, χ_2) to a higher dimensional space $(\chi_1, \chi_2, \chi_1\chi_2)$ enhances the linear separability of the χ^2 distributed input data distributions. This fact suggests that the transition detection classifier is bilinear in the input pair (χ_1, χ_2) and, therefore, that the perceptron realization of the map is necessarily second order [11]. It is emphasized that intermediate single-window variance decisions, high or low, are not performed in the test so that the map is different from the conventional Gaussian classifier.

Figures 4 and 5 plot false alarm and detection probability as a function of threshold γ for σ_0 of one and σ_1 of two and four, respectively. The conditional probabilities were derived for various window lengths N by numerical computation of Equations (3) through (6). Note that for deviation σ_1 and/or window of sufficient size N , the peak of the detection probability occurred near the local minimum in the false alarm probability. The experimental results for neural net performance on the SXOR problem indicate convergence to this region of peak detection and locally minimum false alarm probability.

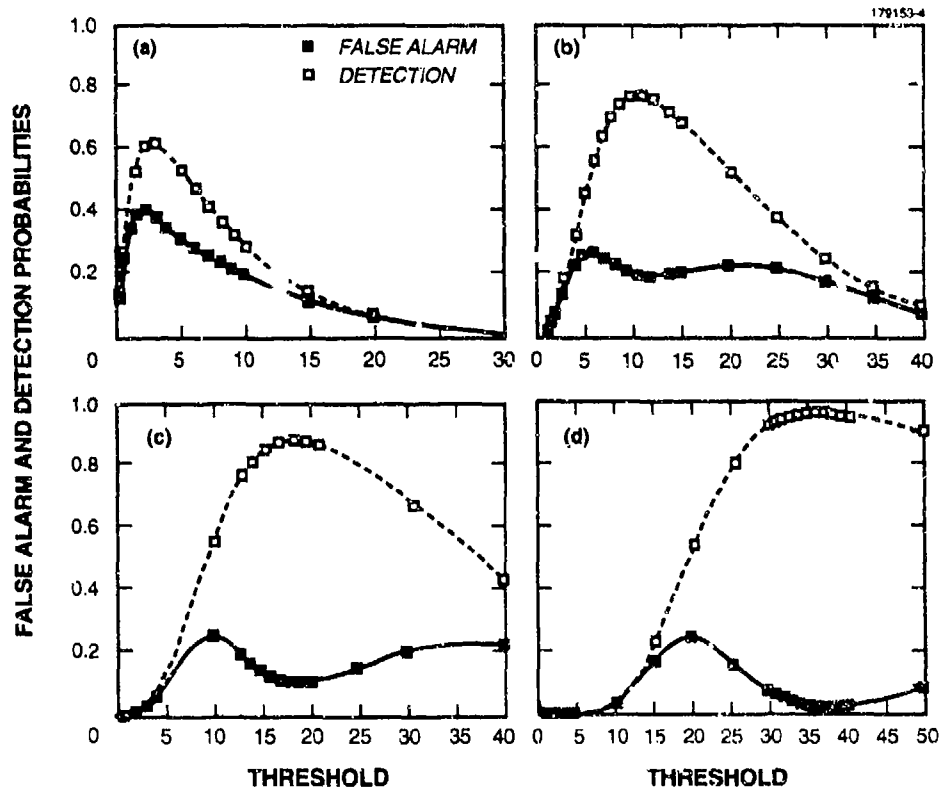


Figure 4. False alarm and detection probability versus threshold γ for SXOR test, with $\sigma_0 = 1$, $\sigma_1 = 2$, and $N = (a) 2$, (b) 6, (c) 10, and (d) 20.

2.2 Back Propagation Neural Net Performance

Figure 6 is a back propagation neural net suitable for hypothesis testing on an input P -vector of data-derived parameters. The desired output for an input vector corresponding to hypothesis

$H_i, i = 1, \dots, Q$ is the vector $(0, \dots, 0, \overset{i}{1}, 0, \dots, 0)$ as obtained from the Q output (deepest layer) neurons. In addition to the input and output neuron layers, the back propagation net contains so-called "hidden" layers. The adjustable parameters on the net consist of a threshold for every neuron in the net and connection weights between neurons on adjacent layers [12]. During forward propagation (left to right) a neuron with threshold θ applies the sigmoid function

$$f(I) = \frac{1}{1 + \exp(-I + \theta)} \quad (7)$$

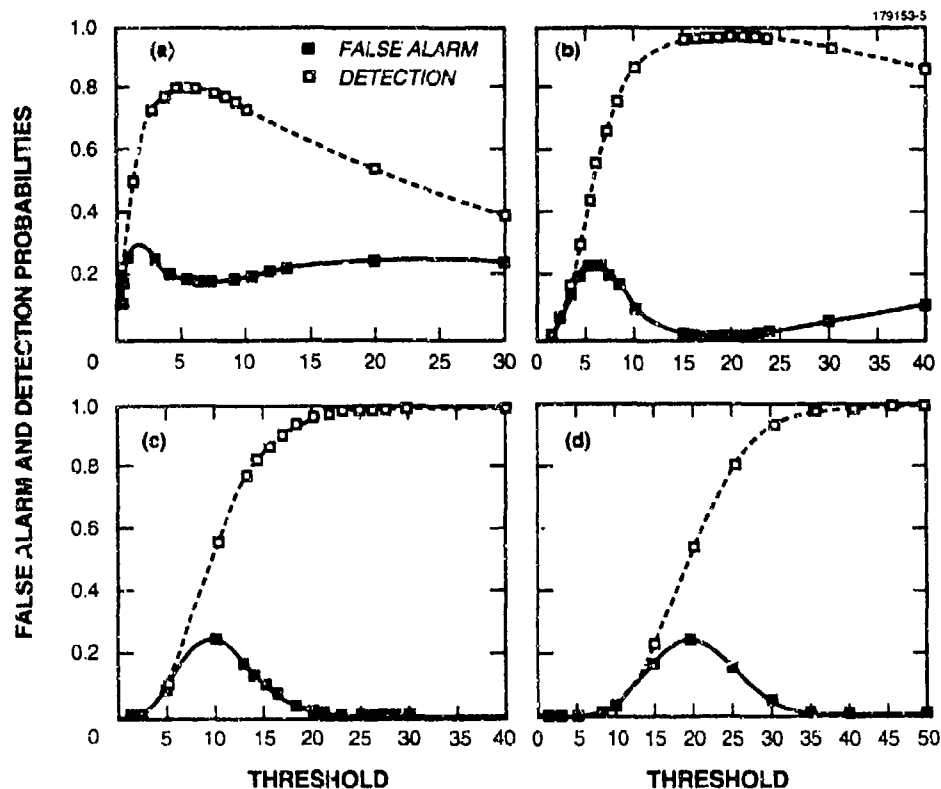


Figure 5. False alarm and detection probability versus threshold γ for SXOR test, with $\sigma_0 = 1$, $\sigma_1 = 4$, and $N = (a) 2$, $(b) 6$, $(c) 10$, and $(d) 20$.

to the input I consisting of the weighted sum of the neuron outputs from the leftward adjacent layer. Net adaption consists of varying the connection weights and thresholds until the output of the deepest layer neurons matches the desired output for all elements of the training set. Details of the back propagation algorithm, which is derived from the gradient descent minimization of the difference between net output and target over the training set, is found in Rumelhart et al. [12].

It has been shown that a three-layer back propagation net is sufficient to implement any reasonable functional mapping between input and output vectors [29]. Note from Equation (7) that an undulation of the mapping is realizable by the equation $(f(I) - f(I + A))$ for a constant threshold A . This roughly suggests that two middle layer neurons are required for each oscillation in the map; however, performance at a Bayesian optimum is not guaranteed by a network that performs an exact mapping for every element in a stochastic training set [9]. To test the performance of the back propagation algorithm on the SXOR map, a net with 2 input neurons, 16 middle layer neurons, and 2 output neurons ($Q = 2$) was used. The training rate and smoothing parameters (η and α

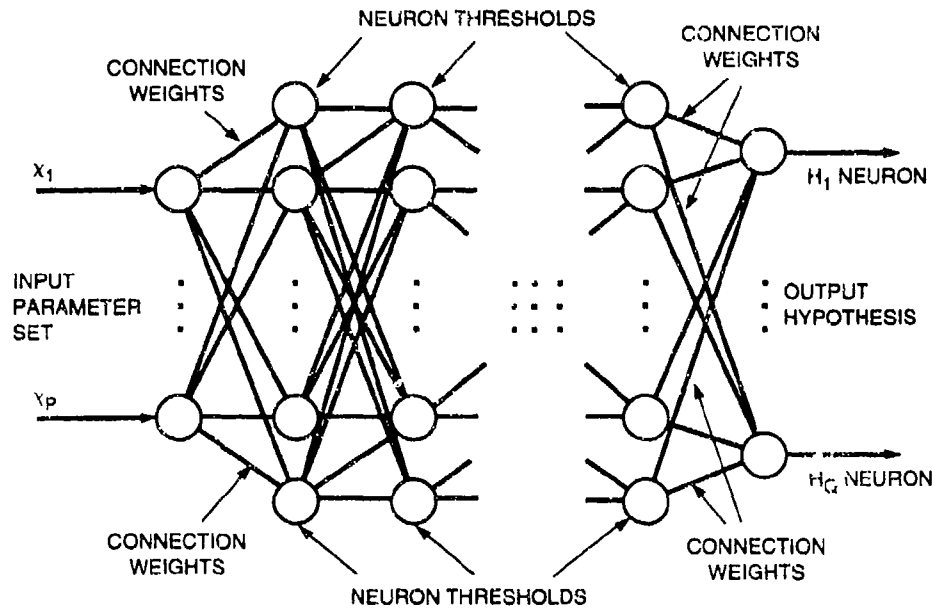


Figure 6. Back propagation neural net for hypothesis testing: P -vector input data and Q -vector hypothesis output, $(0, \dots, 0, 1, 0, \dots, 0) \mapsto H_i$.

in Rumelhart [12]) were chosen to be 0.5 and 0.2 by experimentation with various input sets. The input consisted of sample variances from a training set of Gaussian random noise segments with σ_0 of one and σ_1 of either two or four. The sample variances were computed from windows of length N given by 2, 4, 6, 8, 10, and 15. For each noise pair (σ_0 and σ_1) and window N , two training ensembles each of sizes 400, 800, and 1200 were created with deviation pairs in the order (1,1), (1,0), (0,0), and (0,1). The two third-layer neurons were trained to output values 1 and 0 for the (1,1) and (0,0) inputs, and the output targets were reversed for input corresponding to (1,0) and (0,1). The cost function C , consisting of the summed differences of third-layer outputs and targets, was monitored during training to determine a point beyond which it did not decrease. Figure 7 is a typical cost versus iteration curve for a 100-element training set with σ_1 of 4 and window length N of 10. Also included is the so-called "Hamming error" versus iteration plot, which is defined as the number of decision errors (within 1%) over the training set. As suggested in Hecht-Nielsen [8], nets were trained for a large number ($> 30,000$) of iterations (defined as a single adaption of all net parameters for every element in the training set) and the point of minimum cost was chosen as the optimum. The implementation of the desired training set map, corresponding to $C \rightarrow 0$, was often not attained with 16 middle layer neurons; however, the Bayesian optimum was obtained through the net learning of data biases rather than each undulation in the training set map [9]. A network with too many hidden layer neurons often had plateaus in the cost function in Figure

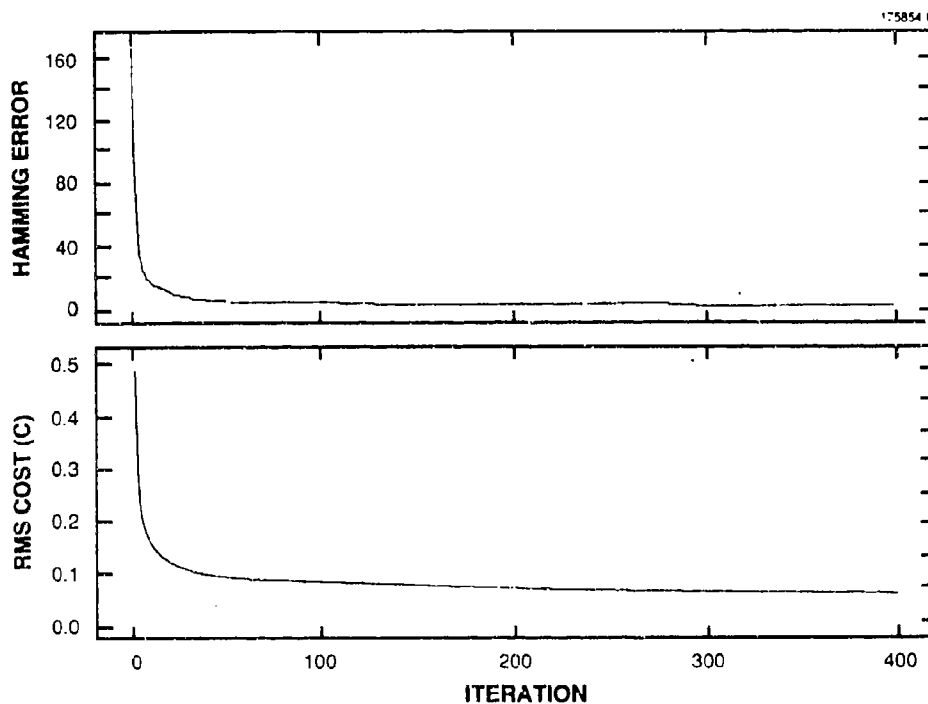


Figure 7. Cost versus iteration curve for back propagation learning of the SXOR map: 2 input, 16 hidden layer, and 2 output neurons in a 100-element training set; $\sigma_0 = 1$, $\sigma_1 = 4$, $N=10$.

7, which was probably due to the phenomenon of "neuron paralysis" [29] that occurs at a neuron when the input is at the tail of the threshold function in Equation (7). In this case, connection and threshold parameter adaption have little effect on the neuron output, hence the cost function remains constant [30].¹ It was found that, whereas extremely long training sometimes resulted in downward jumps in the cost function, the network performance on the test was not improved. Often the only effect was an increase in detection probability with a simultaneous increase in false alarm probability and vice versa. A discussion of techniques to avoid neuron paralysis and other neural computational obstacles is provided in Wasserman [31].

¹On the exclusive-or map a 64-neuron hidden layer had 4% paralyzed runs and a 128-neuron hidden layer had 78% paralyzed runs.

For each parameter set σ_0, σ_1 and window length N , networks trained on ensembles of length 400, 800, and 1200 were performance tested. A performance set with 1200 variance pairs was input to the trained net, and for each pair the largest neuron output determined whether transition or no transition was chosen. The proportion of correctly and incorrectly chosen transitions then determined the detection P_d and false alarm P_f probabilities for the test. The performance probabilities for nets trained on three different sets (of length 400, 800, and 1200) were averaged. The combination of training sets of different size minimized the dependence of the network performance estimate on training set size. Figures 8 and 9 plot P_d and P_f versus N as estimated from the

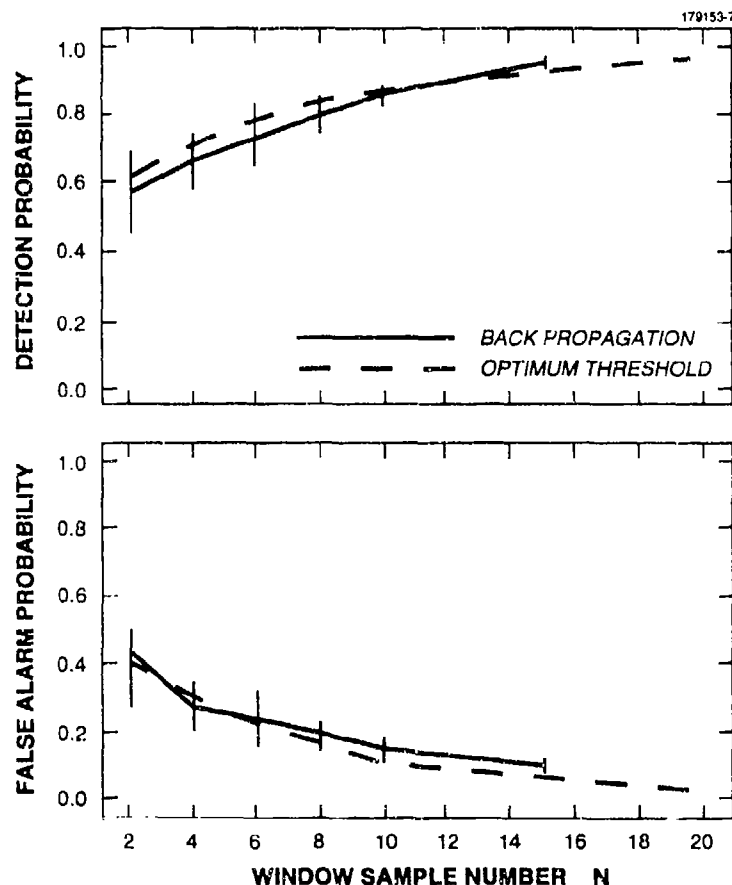


Figure 8. SXOR false alarm and detection probability versus window size N for back propagation NN and classical test: $\sigma_0 = 1, \sigma_1 = 2$.

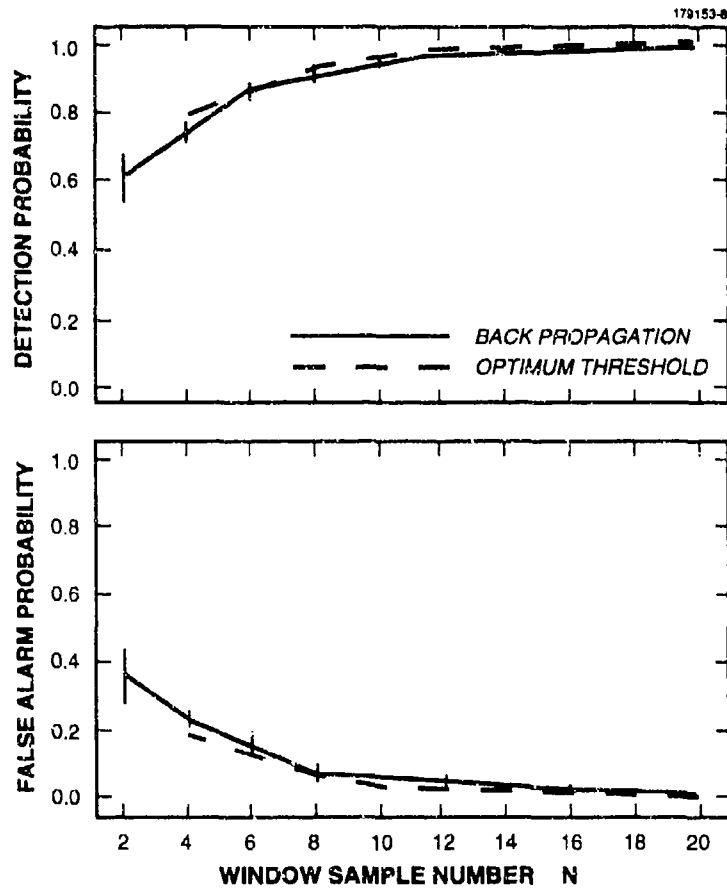


Figure 9. SXOR false alarm and detection probability versus window size N for back propagation NN and classical test: $\sigma_0 = 1, \sigma_1 = 4$.

performance sets for σ_1 of two and four, respectively; the dotted curves correspond to the classical optimum defined in Section 2.1. As seen in Figures 8 and 9, the back propagation network closely approximated the performance at the peak P_d and locally minimum P_f in Figures 4 and 5. This behavior is understood by the equal contribution of H_0 and H_1 errors over the training set in the cost function C [12].

2.3 SXOR Data Fusion

Section 2.2 demonstrated the optimum performance of a back propagation neural net on the SXOR test, which requires a bilinear classification of the input sample variances. This network corresponds to a forward-based SNN in the distributed sensor fusion architecture in Figure 1. A description of FNN training and performance by taking input from two SXOR-trained SNNs follows. The results indicate the enhancement of variance transition performances obtained through distributed sensor data fusion.

In Chair and Varshney [3], an optimum data fusion rule for a binary decision was obtained within the distributed sensor processing architecture. As derived from the log-likelihood ratio test, assuming sensor processor i , $i = 1, \dots, M$, outputs u_i of -1 or $+1$ for decision H_0 or H_1 , the data fusion rule is [3]

$$f(u_1, \dots, u_M) = \begin{cases} +1 & \text{if } a_0 + \sum_{i=1}^M a_i u_i > 0 \\ -1 & \text{otherwise,} \end{cases} \quad (8)$$

where the coefficients a_i , $i = 1, \dots, M$, are given by

$$a_i = \frac{1}{2} \log \left[\frac{(1 - P_{m_i})(1 - P_{f_i})}{P_{m_i} P_{f_i}} \right] \quad (9)$$

and

$$a_0 = \log \frac{P_1}{P_0} + \frac{1}{2} \sum_{i=1}^M \log \left[\frac{P_{m_i}(1 - P_{m_i})}{P_{f_i}(1 - P_{f_i})} \right], \quad (10)$$

with P_0 and P_1 the prior probabilities of H_0 and H_1 , and P_{m_i} and P_{f_i} the miss and false alarm probabilities of the i th sensor processor. The architecture implied by Equations (8) through (10) is, in fact, a first-order perceptron [5,11] that can be realized through the adaption of the connection weights a_i , $i = 0, \dots, M$, by training. To implement perceptron learning for input $u_i = \pm 1$, $i = 1, \dots, M$, define the normalized predicate vector $\hat{\Phi} = (1, u_1, \dots, u_M)/\sqrt{M+1}$ and connection weight $(M+1)$ -vector $A = (a_0, \dots, a_M)$. After a training set element input vector is propagated through each SNN, and the SNN decisions are determined by the largest neuron outputs, the dot

product $\hat{\Phi} \cdot A$ is computed. In the case of a correct FNN decision, $\hat{\Phi} \cdot A > 0 (< 0)$ for $\hat{\Phi}$ corresponding to $H_1 (H_0)$, the connection weight vector is not changed ($A' = A$). For an incorrect FNN decision the connection weight vector is altered by the normalized predicate vector, $A' = A \pm \hat{\Phi}$, where $+$ ($-$) corresponds to $\hat{\Phi} \cdot A < 0 (> 0)$ for $\hat{\Phi}$ corresponding to $H_1 (H_0)$. An iteration of the perceptron adaption algorithm consists of the application of the above algorithm for every element in the training set [11]. Training continues until the FNN performs a correct decision for the entire set or until the FNN performance does not improve.

The architecture for the fusion of two SXOR-trained back propagation SNNs is shown in Figure 10. It is assumed that the high noise deviation σ_1 is sensor-dependent, so that each SNN

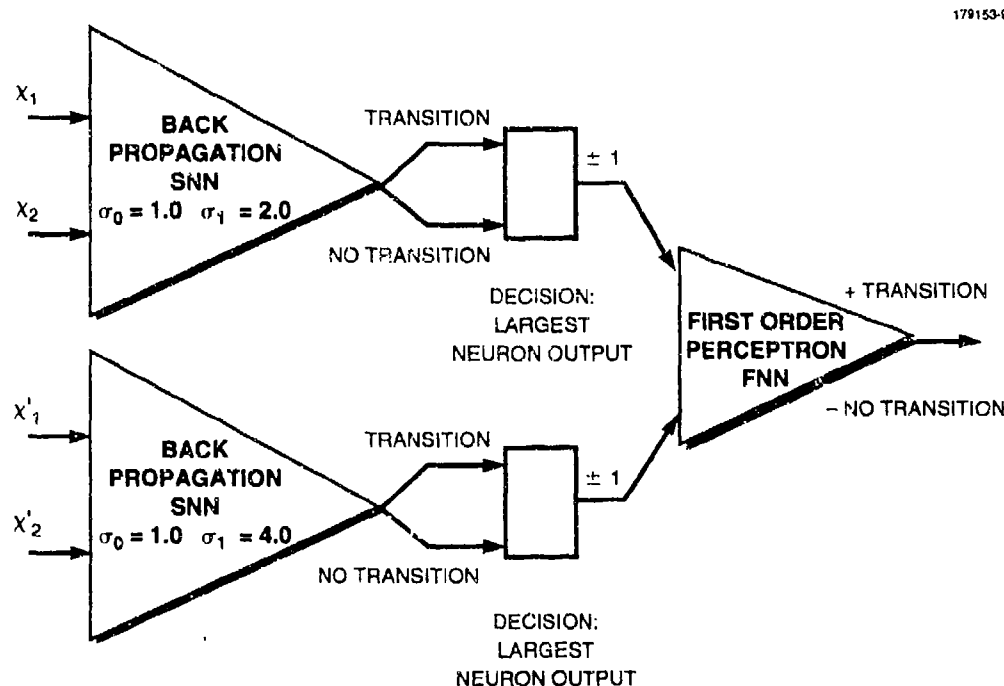


Figure 10. Fusion architecture for SXOR test: SNN1 ($\sigma_0 = 1, \sigma_1 = 2$), SNN2 ($\sigma_0 = 1, \sigma_1 = 4$), and FNN first-order perceptron.

was previously trained on a different variance pair $\sigma_0 (= 1)$ and σ_1 . For each window size N ($= 2, 4, 6, 8, 10$) a pair of SNNs was trained on sample variances with σ_1 of two and four. As in the experiment described in Section 2.2, the SNN target outputs were (1,0) and (0,1) for transition and no transition, respectively. A performance set of 1000 variance pairs each was

used to compute the SNN detection P_d , false alarm P_f , miss P_m , and correct no transition P_{cH_0} probabilities for the test. The SNN decisions were determined by the largest neuron output. A plot of these performance probabilities for the ($\sigma_1 = 4$) SNN and ($\sigma_1 = 2$) SNN as a function of window size N is shown in Figure 11. A determination of the SNN detection and false alarm

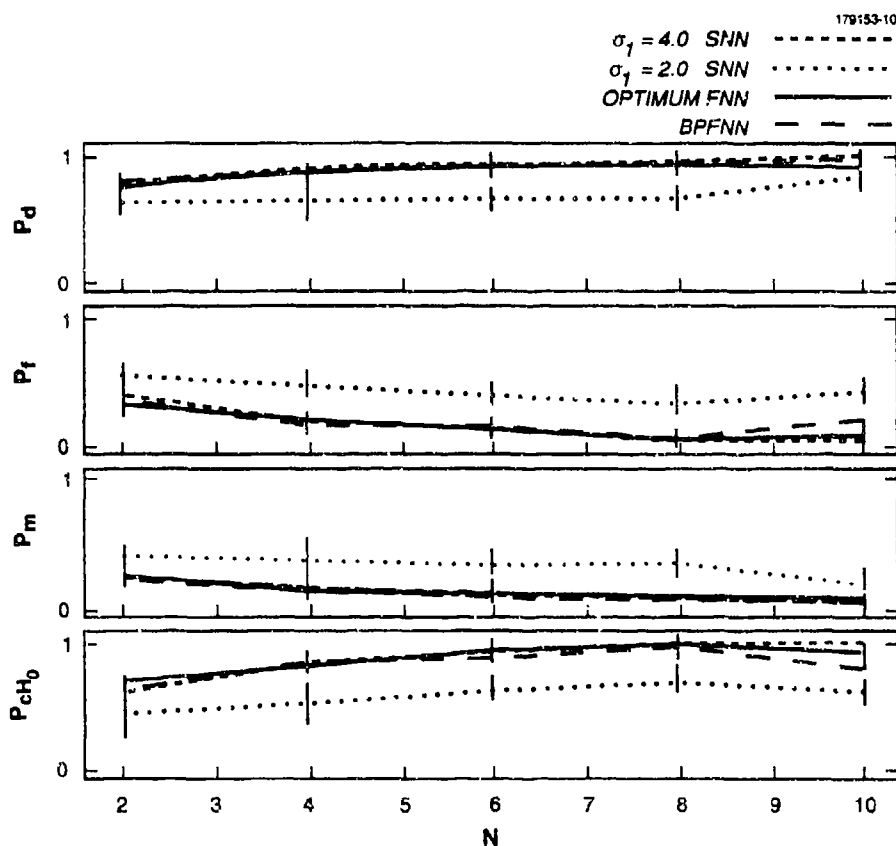


Figure 11. Performance probabilities for the fusion of two SXOR-trained SNNs: P_d , P_f , P_m , and P_{cH_0} versus window size N for $\sigma_1 = 2$ and $\sigma_1 = 4$ SNNs, optimum FNN, and back propagation FNN.

probabilities allowed the definition of an optimum perceptron FNN from Equations (8) through (10). An estimate of the perceptron FNN performance was obtained with 1000 variance quartets $((i, j), (i', j'))$, $i, j, i', j' \in \{0, 1\}$. Quartet $((i, j), (i', j'))$ corresponds to input variance pairs (i, j) and (i', j') for the SNNs with σ_1 of two and four, respectively. Recall that i of 1(0) corresponds to the choice of a high(low) noise deviation in the definition of the sampled variance. The output

SNN decision was converted to $u_i = \pm 1$, $i = 1, 2$ (as in Figure 10) before input to the perceptron FNN. Figure 11 shows the perceptron FNN performance as a function of N as determined from the performance set. Note that the FNN matched the performance of the ($\sigma_1 = 4$) SNN for N of 2, 4, 6, 8, and 10. The ($\sigma_1 = 2$) SNN has a small effect on the optimum FNN due to the generally poor performance of the net ($P_m \simeq P_f \simeq 0.5$).

Motivated by the representation of the optimum FNN as a perceptron, a back propagation FNN (BPFNN) was defined for the data fusion of the two SXOR-trained SNNs. The BPFNN consisted of 4 inputs (2 from each SNN), a 16-neuron hidden layer, and 2 output neurons. The BPFNN was trained on 25 randomly generated variance quartets in the order $((0,0),(0',0'))$, $((1,0),(1',0'))$, $((1,1),(1',1'))$, and $((0,1),(0',1'))$. Each variance quartet was propagated through the SNNs and normalized to define the 4-element input to the BPFNN. As in the case of the SNNs, the BPFNN targets were (1,0) and (0,1) for transition and no transition, respectively. To speed up training for an FNN with only 16 hidden neurons, the variance quartets from the overlapped region of the input domain were removed from the training set. This procedure usually suffices to obtain Bayesian optimum performance through the learning of data biases [9]. After BPFNN training, a performance set of 1000 random variance quartets was generated and propagated through the entire sensor fusion system. A count of correctly and incorrectly detected transitions and no transitions over the performance set determined the conditional probabilities plotted in Figure 11. Note that the trained BPFNN essentially matched the optimum FNN at the performance of the ($\sigma_1 = 4$) SNN for window sizes 2 through 10. These results suggest that the trained distributed sensor fusion system attained at least the performance of the strongest sensor at any time. To demonstrate performance enhancement through data fusion, the fusion of two ($\sigma_1 = 4$) SNNs trained on data of window length 2 was considered. A three-layer BPFNN was trained on the SNN pair outputs from 100 input variance quartets. As in the training above, variance quartets from the overlapped regions of the input domain were discarded from the training set. This procedure required a BPFNN training time of about 30 min on the Silicon Graphics Workstation. The BPFNN performance probabilities were computed from 100 independent performance sets, each consisting of 100 variance quartets. Averaged BPFNN performance probabilities (P_d, P_f, P_m, P_{CH_0}) given by (0.84, 0.11, 0.16, 0.87) were obtained for comparison with the ($\sigma_1 = 4$) SNN performance set (0.76, 0.38, 0.24, 0.62). In applying the same training and performance procedure to two fused ($\sigma_1 = 4$) SNNs with window length 4, BPFNN averaged probabilities given by (0.86, 0.12, 0.14, 0.88) were obtained. These values are compared against a window length 4 ($\sigma_1 = 4$) SNN performance set of (0.84, 0.19, 0.16, 0.81) and, therefore, a BPFNN performance enhancement of up to 70% over the individual sensor nets was demonstrated. The results in Section 3, in which neural net fusion is applied to the FF launches, also support this conclusion.

3. FIREFLY SENSOR FUSION EXPERIMENT

This section applies the distributed sensor fusion architecture described in Section 2 to a three-sensor fusion of measurements during the recent FF launch. The experiment, involving the complicated logistics of three-radar imaging and tracking, provided a rare opportunity to demonstrate the power of neural net sensor fusion.

3.1 Firefly Experiment

The FF experiment consisted of two rocket launches (FFI on 29 March and FFII on 20 October 1990) from Wallops Island into the Atlantic Ocean about 400 km eastward. During the flight the deployment of an inflatable balloon was observed simultaneously by the three Millstone Hill radars at a range of approximately 750 km from the targets. The active sensors were the Haystack X-band ($\lambda = 3$ cm) and Firepond CO₂ laser ($\lambda = 11.2\mu$) imaging radars, and the Millstone L-band ($\lambda = 23.1$ cm) tracking radar.

About 6 min after the launch, a metallic canister (cross section $\simeq 1$ m²) was deployed from a much larger metallic payload. As the payload fell away from the track, the canister ejected four metallic doors and an inflating carbon cloth cone (cross section $\simeq 2$ m²). As shown in Figure 3, the predeployment, deployment, and postdeployment phases are clearly identified for both canister and balloon-canister payloads.

The input data for the sensor fusion system consisted of range-Doppler images from the Haystack and Firepond radars and a passive-IR spectral simulation of the objects in the images. Radar imaging takes advantage of a moving target's aspect angle change to obtain a signal Doppler shift proportional to the scatterer cross-range extent. The Doppler resolution is proportional to the inverse of the signal integration time over which it is assumed that the scatterer has moved a negligible distance and the signal is coherent. Through object motion analysis, the Doppler shift is scaled to a physical cross-range distance [32,33]. This analysis is coupled with an estimate of the range from the signal delay to obtain a 2D range-cross-range image of the object. The range-Doppler technique results in image resolution greater than the limits imposed from the radar aperture and radiation wavelength. Details of range-Doppler imaging theory for the Haystack and Firepond radars is given in Ausherman et al. [34] and Kachemyer [35], respectively. The third sensor input to the sensor fusion system was from a passive-IR simulation of the objects in the images. The Lincoln Laboratory-developed simulator was used to provide a feasibility study for passive-IR deployment detection [36]. Inputs to the simulator included object shape, dimensions, spin/precession rates, and orientation relative to the sun. The input thermal properties were initial temperature, emissivity, interior emissivity, absorptance, thermal mass (density $\times$ heat capacity), and specularity. Finally, a climate and cloud cover-dependent model of earth spectral irradiation through the atmosphere was input. The output from the simulator consisted of the object spectral irradiation into the solid angle over the range $[5 \mu, 25 \mu]$ in Watts/steradians. The spectral irradiation provided information about the material composition of an object. In the range $[5 \mu, 25 \mu]$, this information is indirect

through estimates of relative emissivity and reflectance at the surface. Thus, for example, a metallic object with low emissivity and absorptance ($\epsilon \simeq \alpha \simeq 0$) has a spectrum dominated by reflected earthshine. As seen in Figure 12, the metallic spectrum has notches at the ozone ($\simeq 9.5\mu$) and CO_2

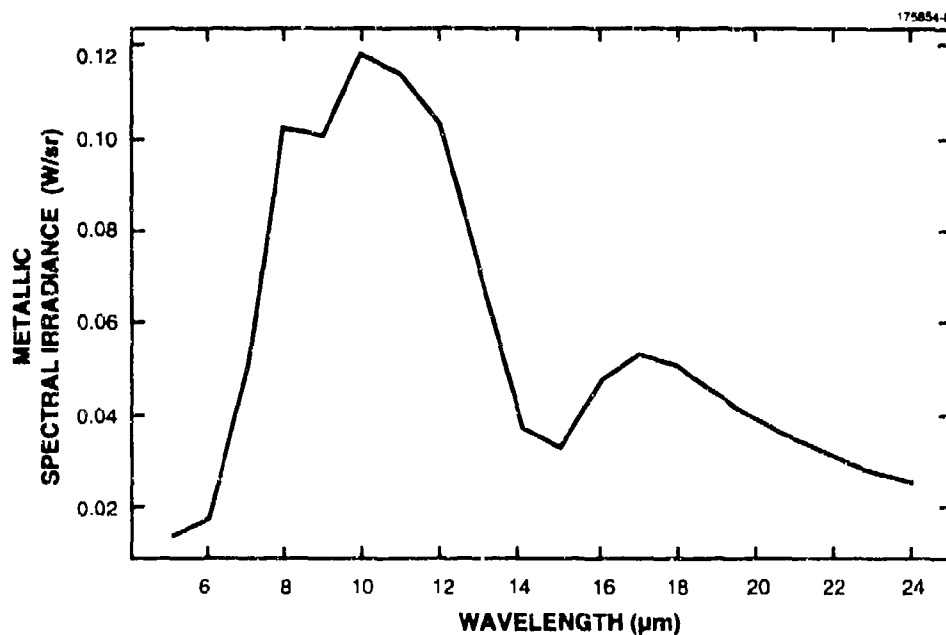


Figure 12. Passive-IR simulated spectrum over the range $[5 \mu, 25 \mu]$ in W/sr. Metallic object with 1-m^2 cross section, $\epsilon = \alpha = 0$.

($\simeq 13 \mu$) wavelengths due to atmospheric absorption of earthshine as contrasted with a graybody object ($\epsilon \simeq 0.75$) in Figure 13, in which a classic blackbody spectrum dominates the spectral irradiance. Note from the spectra in Figures 12 and 13 that the graybody irradiance is about 20 times the reflected component for a 1-m^2 object, suggesting that the existence of a graybody object among a set of metallic targets will dominate the total spectral irradiance. The balloon-canister deployment sequence for the FFI launch, with the identification of the predeployment, deployment, and postdeployment phases, is shown in Figure 14. Figure 15 illustrates the passive-IR simulation from each phase: a reflective earthshine canister spectrum for predeployment, the superposition of metallic door and carbon cloth (graybody) spectrum for deployment, and a graybody carbon cloth spectrum for postdeployment. These simulated spectra form the training set for the passive-IR SNN in the sensor fusion architecture discussed in Section 3.2.

Figure 16 depicts the formulation of the fused sensor decision on balloon deployment from

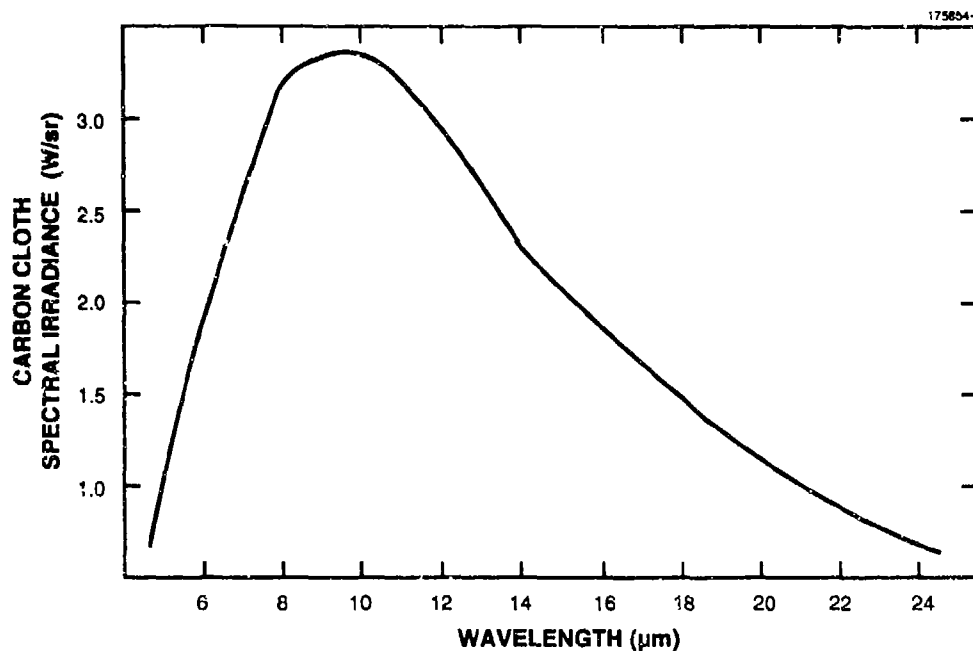


Figure 13. Passive-IR simulated spectrum over the range $[5 \mu, 25 \mu]$ in W/sr. Graybody object with 1-m^2 cross section, $\epsilon = 0.75$, $\alpha = 0.9$.

Haystack and Firepond range-Doppler images and a passive-IR simulation. Due to the longer Haystack coherent integration time, the range and cross-range resolutions of the Firepond and Haystack radar were comparable. The most important differences between the radar images resulted from a Haystack beamwidth about 100 times that of the Firepond, which at 7.5 m at 750 km was sufficient to observe only single targets in a complex scene, whereas the Haystack radar observed a much larger cross-range extent. It should be emphasized that these Firepond properties are beneficial; that is, a shorter integration time allows more rapid image generation (~ 3000 times faster) and a narrow beam is more difficult to detect.

As seen in Figure 16, during the predeployment phase (of about 24 s) Firepond images consisted of only the metallic canister, whereas Haystack images contained returns from the separating payload. The passive-IR spectrum was weak (~ 0.6 W/sr peak) and earthshine-dominated with notches at 9.5 and 13μ . During the 2-s deployment phase the cross-range velocity component of the ejected doors resulted in a rapid loss of images for Firepond. Two of the doors moved roughly in parallel to the inflating balloon so that throughout the deployment the Haystack images consisted of the decoy and nearby doors represented in Figure 16. Note from Figure 15 that in the passive-IR deployment spectra the balloon graybody radiation dominated the structure in the earthshine spectrum from the doors. The postdeployment phase of 30 s was determined from the Firepond images

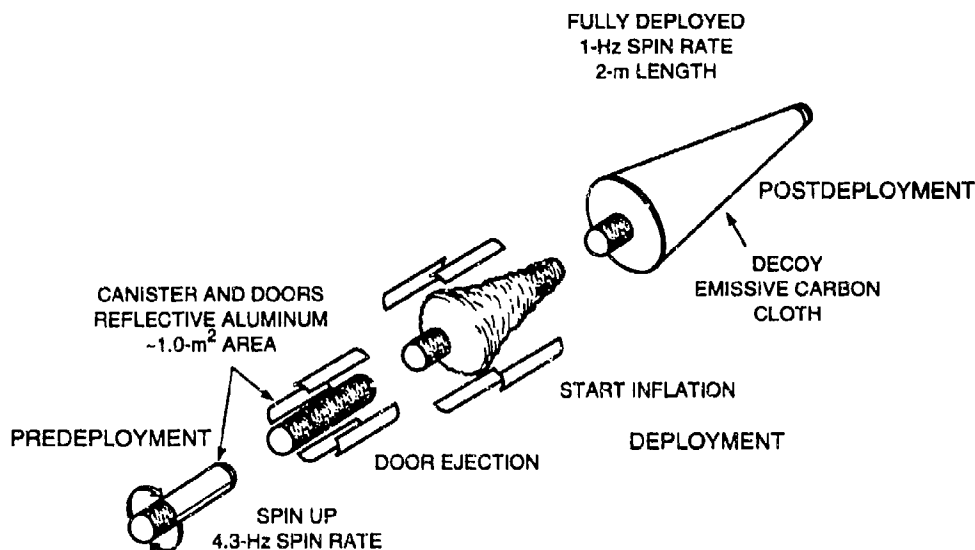


Figure 14. Balloon-canister deployment sequence for FFI launch: predeployment, deployment, and postdeployment phases.

of an inflated carbon cloth cone, Haystack images of the balloon and two sufficiently separated metallic doors, and a passive-IR carbon cloth graybody spectrum. The data represented in Figure 16 were input to the sensor fusion system described in Section 2 for a decision of predeployment, deployment, and postdeployment phases. The irreducible ambiguities inherent in the single sensor data are also observed in Figure 16. The passive-IR sensor discrimination between deployment and postdeployment was weak due to the graybody domination of the reflected earthshine spectra. The Firepond sensor was ambiguous between pre- and postdeployment phases due to the similarity of the canister and balloon range-Doppler images. The shape difference between the cylindrical canister and the cone-shaped balloon is a weak feature in noise-corrupted data. Further image processing, such as intensity averaging and smoothing, may enhance the radar image-based decisions [37]; however, in the sensor fusion experiment preprocessing was limited to single intensity threshold and centroid operations. The Haystack image set was overall the least ambiguous due to the generation of complex scenes. During deployment the radar often lost reflections from the doors and became ambiguous between predeployment and deployment decisions.

3.2 Firefly Sensor Fusion System

Figure 17 shows the distributed sensor fusion system used to analyze the Firefly balloon

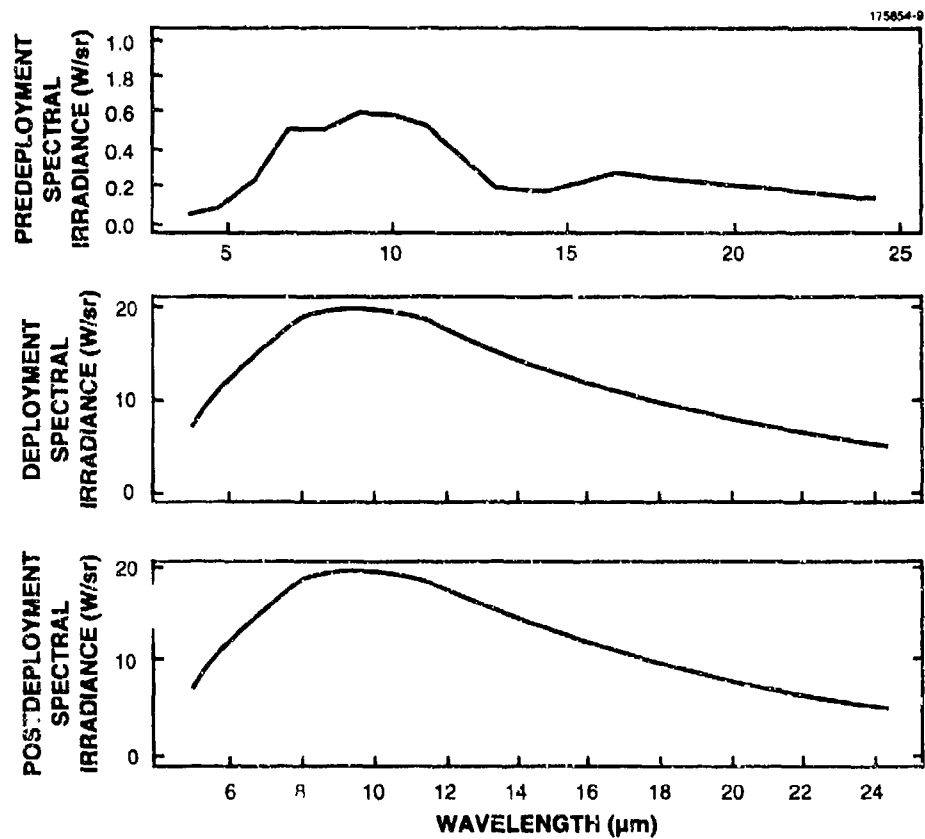


Figure 15. Passive-IR simulated spectra [$5\ \mu$, $25\ \mu$] in W/sr for three balloon-canister phases in FFI launch.

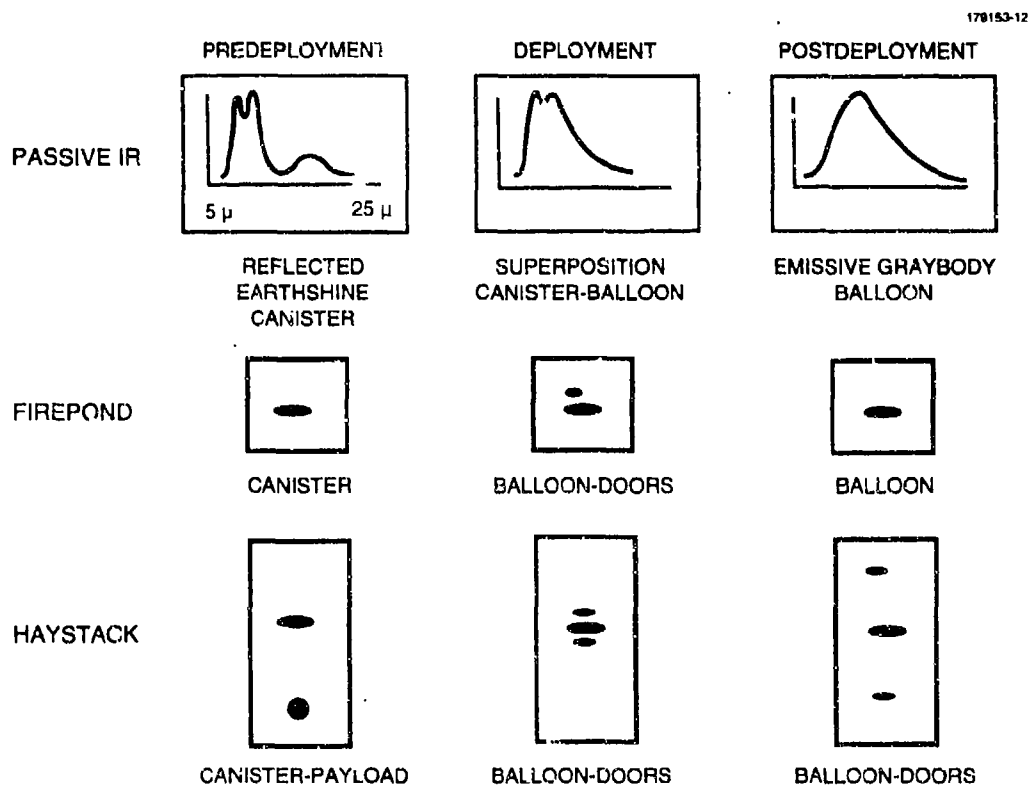


Figure 16. Formulation of multisensor data fusion for balloon-canister deployment: Haystack and Firepond range-Doppler images and passive-IR simulation of predeployment, deployment, and postdeployment phases.

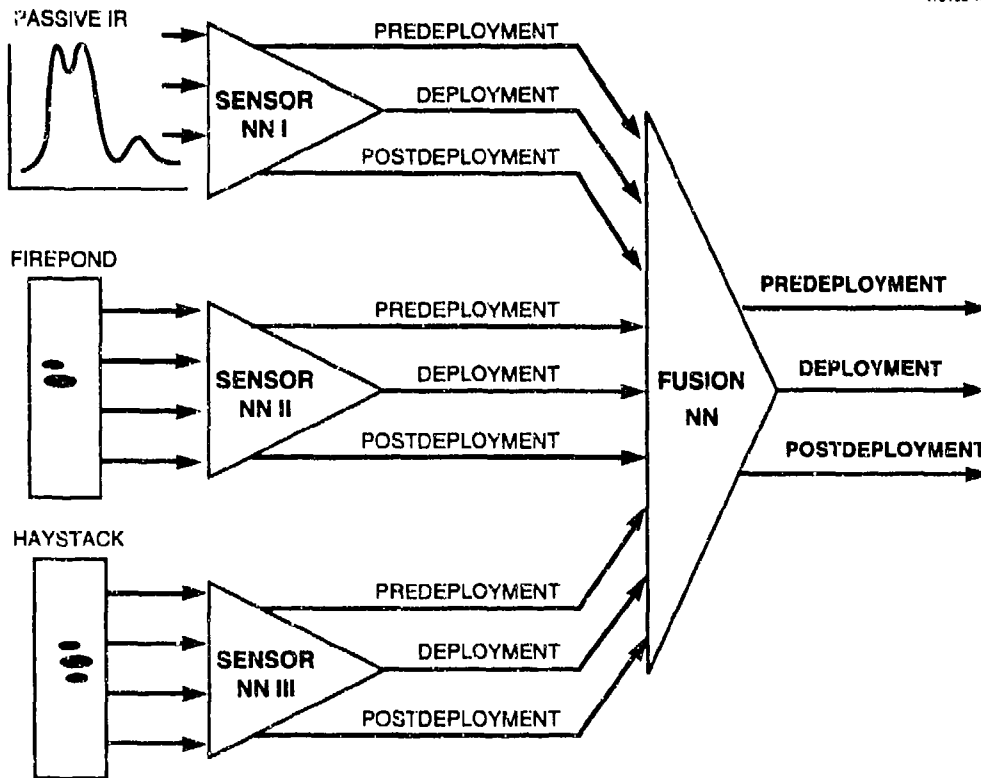


Figure 17. Distributed sensor fusion system for FFI balloon-canister deployment detection: Back propagation SNNs for passive-IR, Haystack, and Firepond sensors and BPFNN.

deployment from Haystack and Firepond range-Doppler images and the passive-IR simulation. Three back propagation SNNs were trained to output a deployment decision based only on the individual sensor data. The three output neurons, corresponding to predeployment, deployment, and postdeployment on each SNN, output an analog value in the range $[0,1]$. The BPFNN took the normalized SNN outputs as input and mapped to an overall decision based on the three neuron SNN outputs for each of the three deployment phases. The SNN and FNN output targets were $(1,0,0)$ for predeployment, $(0,1,0)$ for deployment, and $(0,0,1)$ for postdeployment. The architecture in Figure 17 implies that the FNN was trained to perform a cluster analysis in the 9D space of SNN outputs. The FNN inputs were clustered around $((1,0,0),(1,0,0),(1,0,0))$, $((0,1,0),(0,1,0),(0,1,0))$, and $((0,0,1),(0,0,1),(0,0,1))$ for predeployment, deployment, and postdeployment, respectively. The SNNs for Haystack and Firepond had a 20×200 -pixel input plane, a 4×4 -neuron middle layer, and a third output layer with 3 neurons. The passive-IR SNN and the FNN had 16 neurons in the middle layer, 3 neuron outputs, and input layers of 20 and 9 neurons, respectively. The radar SNN structure was determined in part by the computational complexity of the fully interconnected 2D back propagation net and by the minimum number of neurons required for convergence over the training set of images. The 1D nets (one SNN and the FNN) were not complexity-bound so that the number of hidden neurons was determined by convergence issues discussed in Section 2.2.

The Haystack and Firepond SNNs were trained on 3 to 4 images each from predeployment, deployment, and postdeployment. For each image pair the aggregate passive-IR spectrum was computed based on the objects in the Haystack images. Training each radar SNN on a training set of about 12 images using the back propagation learning algorithm required about 30 min on a Silicon Graphics Workstation. Upon completing SNN training, a set of about 20 images and passive-IR spectra each from the three deployment phases were propagated through the SNNs. The normalized SNN outputs formed a training set for the FNN. It should be emphasized that the training set for the FNN must reflect the uncertainty in decisions from each sensor alone. This was accomplished by using an FNN training set distinct from the SNN training data, for which the performance of each SNN is well-represented. Thus, for example, because the Firepond pre- and postdeployment images were inherently ambiguous, the FNN training set contained Firepond SNN outputs with about 40% error in pre- and postdeployment detection. This procedure was necessary for the FNN to learn the extent that a sensor should be ignored for a given pattern of SNN outputs. Figure 18 plots the FNN cost function C versus iteration during training. The 1D FNN converged after about 90 iterations on a training set with about 60 input nine-vectors. The algorithm ran in approximately 20 s on the Silicon Graphics Workstation.

To test the trained sensor fusion system, a performance set was created that contained between 10 and 20 radar image pairs each from predeployment, deployment, and postdeployment of novel data from the same launch. A simulated passive-IR spectrum was generated for each Haystack image with added random Gaussian noise of deviation 10% of the peak spectral value. The images and spectra were stacked sequentially in time and propagated through the sensor fusion system. Figure 19 shows the neuron outputs of the SNNs over the performance set. Note that for the passive-IR SNN the deployment and postdeployment neurons oscillated in value, reflecting the

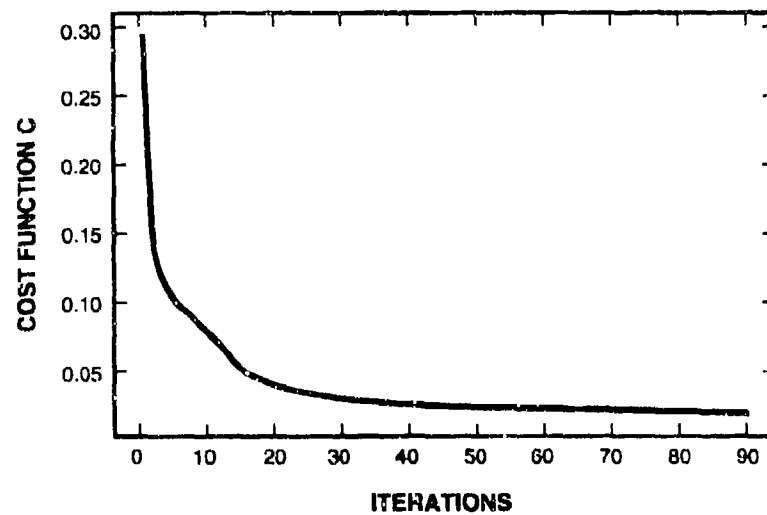
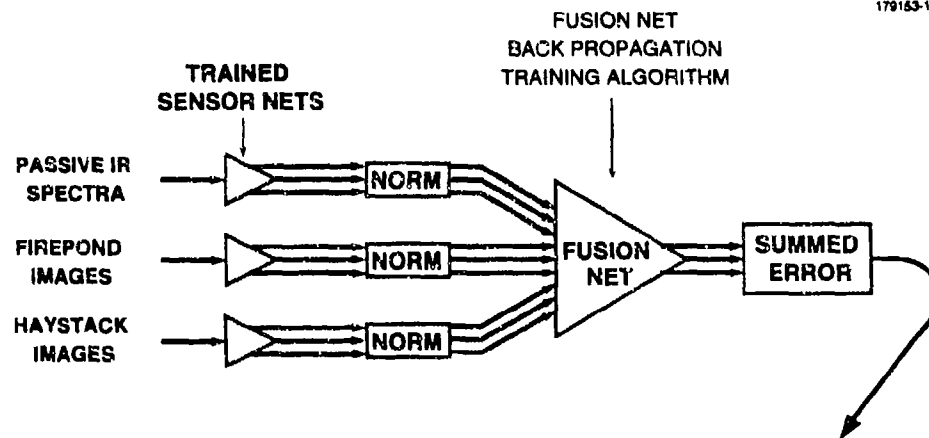


Figure 18. Fusion neural net cost function C versus training iterations for sensor fusion system training on FFI balloon-canister deployment detection.

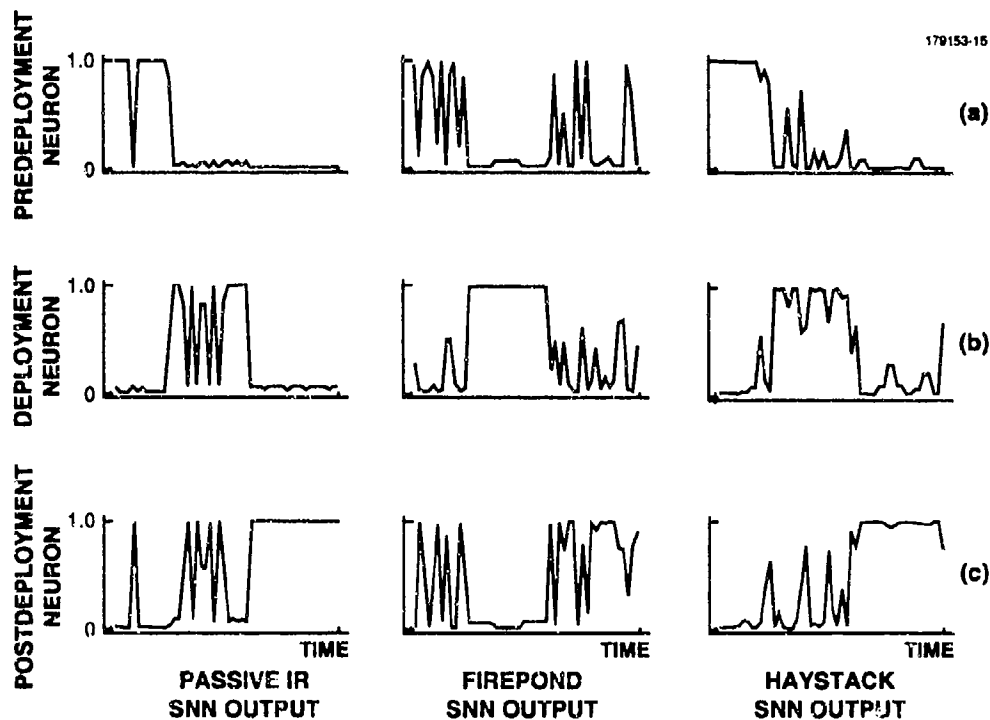


Figure 19. SNN neuron outputs: Novel FFI balloon-canister deployment data for (a) predeployment, (b) deployment, and (c) postdeployment neurons.

ambiguity due to the domination of the reflective door spectrum by the graybody balloon. The Firepond SNN neurons oscillated during the pre- and postdeployment phases due to the similarity of the canister and balloon range-Doppler images. Finally, although the Haystack radar SNN had the best performance overall, there was oscillation during the deployment phase due to the loss of reflections from the ejected doors. Figure 20 depicts the FNN neuron outputs for the performance set, which clearly indicates a performance superior to any of the SNNs. This is the desired evidence of sensor synergism obtained through the fusion of multisensor data.

A procedure similar to the training and performance tests described above was applied to the canister-payload deployment in Figure 3. In this case the training set was generated from the FFI launch, and the system performance was tested on data from the FFII launch. Details of the analysis will not be described, except to note that the passive-IR spectrum was dominated by the large metallic payload (3 W/sr peak) in the predeployment and deployment phases. The radar images contained only the payload during predeployment, the canister and payload during

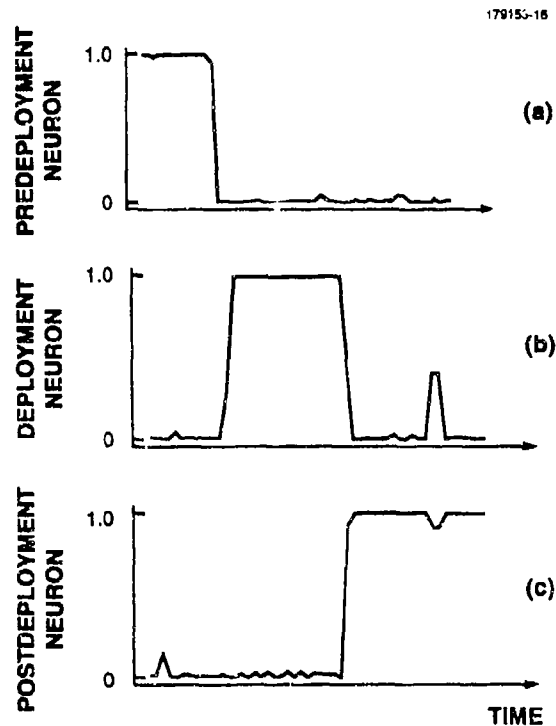


Figure 20. FNN neuron outputs: Novel FFI balloon-canister deployment data for (a) predeployment, (b) deployment, and (c) postdeployment neurons.

deployment, and the canister alone during postdeployment. The radar SNNs, therefore, detected the deployment phases based on image segmentation and payload-canister size differences. The three neuron output values for each of the SNNs from a performance set of about 60 FFII images of the canister deployment is shown in Figure 21. The Firepond SNN performance was poor due to the lack of correct scaling for FFII and a high clutter level in the data. The difficulties in launch-to-launch cross-range scaling resulted from different object spin rates between FFI and FFII. For the most part, the problem can be corrected by further postlaunch image processing. Figure 22 illustrates the three FNN neuron outputs for the performance set of FFII data. As with the balloon deployment results in Figures 19 and 20, there is clear evidence of sensor synergism from the distinct FNN neuron outputs during the different phases.

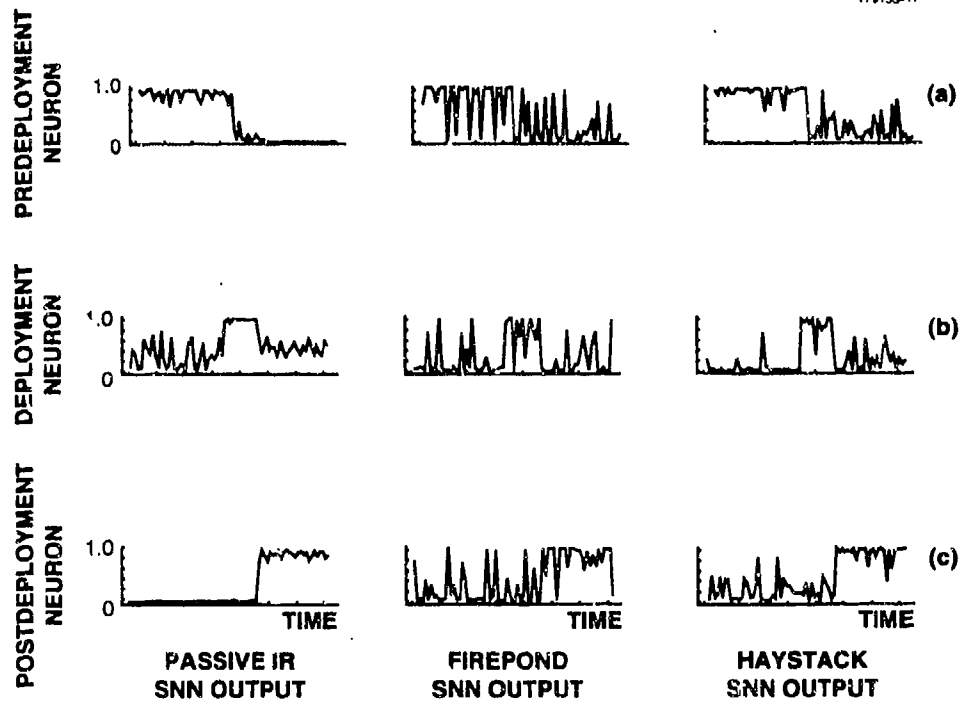


Figure 21. SNN neuron outputs: Novel FFII canister-payload deployment data for (a) predeployment, (b) deployment, and (c) postdeployment neurons.

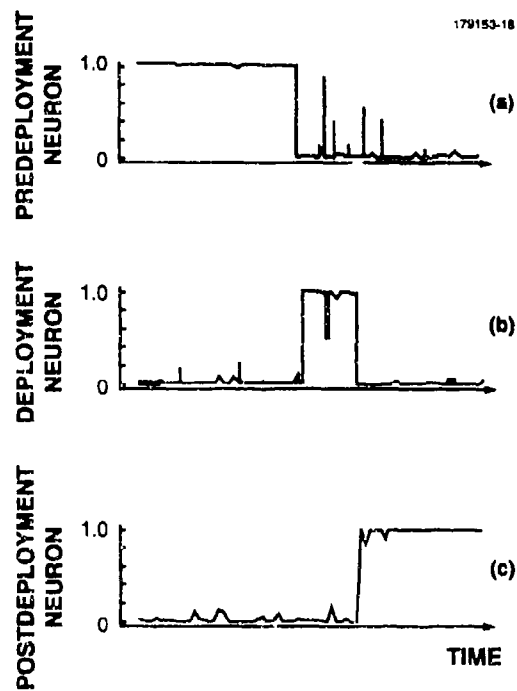


Figure 22. FNN neuron outputs: Novel FFII canister-payload deployment data for (a) predeployment, (b) deployment, and (c) postdeployment neurons.

4. CONCLUSION

This report describes the theoretical and experimental analysis of neural networks in a distributed sensor fusion decision-making environment. The architecture consists of sensor-level decision nodes, which output a decision based on data from a particular sensor. The multisensor decision outputs form the input to a fusion decision node for an overall decision. The fusion node performs cluster analysis in the multisensor decision hypothesis space to obtain the system decision.

The theoretical analysis consisted of the application of neural nets to a benchmark problem, the detection of variance transitions in Gaussian noise, for which a classical hypothesis test is defined. In both the cases of stand-alone single sensor decision making and multisensor fusion, the neural nets matched the performance at the classical optimum. In general, the optimum fusion processor, which is obtained from a log-likelihood test in Chair and Varshney [3], is a perceptron neural net. This fact motivated the use of an adaptive network at the fusion processor in the distributed sensor fusion architecture. It was shown that a back propagation net matched the performance of the optimum fusion processor on the variance transition detection (SXOR) test. The procedure of net training in the distributed sensor architecture, which requires separate representative training sets for the sensor and fusion nodes, was reviewed in its application to the SXOR test.

The experimental analysis of neural net sensor fusion consisted of applying the system to object deployment detection during the Firefly launch. The sensor inputs consisted of range-Doppler images from the Haystack (X-band) and Firepond (CO₂ laser) radars, as well as a passive-IR spectral simulation of the tracked objects. The output decisions were the identification of predeployment, deployment, and postdeployment phases for the release of an inflatable carbon cloth balloon. The fusion neural net performed a 9D cluster analysis (three sensors with three decisions) on the output of independently trained sensor neural nets. The system was trained and performance-tested on data from the first Firefly launch for the detection of balloon deployment. In a more recent experiment the system was applied to the detection of canister deployment using training and performance data from the first and second Firefly launches, respectively. The results clearly demonstrate enhanced fusion performance from the comparison of deployment detection by the fusion and sensor nets. Through the analysis of sensor ambiguities, it was shown that the fusion system employs synergism between the various sensors to provide an optimum overall decision.

Distributed sensor fusion processing is a highly relevant procedure for data-based decision making. The architecture in Figure 1 has built-in robustness against communication failure by allowing decision making at each sensor processor. The system is also robust against single sensor failure through the fusion of multiple sensor decisions. This report demonstrates that the application of neural nets in the architecture takes full advantage of performance enhancements made possible by data fusion.

REFERENCES

1. R.R. Tenney and N.R. Sandell, "Detection with distributed sensors," *IEEE Trans. Aerosp. Electron. Syst.* **AES-17**, 501-509 (July 1981).
2. F.A. Sadjadi, "Hypothesis testing in a distributed environment," *IEEE Trans. Aerosp. Electron. Syst.* **AES-22**, 134-137 (March 1986).
3. Z. Chair and P.K. Varshney, "Optimal data fusion in multiple sensor detection systems," *IEEE Trans. Aerosp. Electron. Syst.* **AES-22**, 98-101 (January 1986).
4. S.C. Thomopoulos, R. Viswanathan, and D.C. Bougoulas, "Optimal decision fusion in multiple sensor systems," *IEEE Trans. Aerosp. Electron. Syst.* **AES-23**, 644-653 (September 1987).
5. K. Atteson, M. Schrier, G. Lipson, and M. Kam, "Distributed decision-making with learning threshold elements," *Proc. IEEE 27th Conf. Decision and Control*, Austin, Texas (December 1988), pp. 804-805.
6. A.R. Reibman and L.W. Nolte, "On determining the design of fusion detection networks," *Proc. IEEE 27th Conf. Decision and Control*, Austin, Texas (December 1988) pp. 2473-2474.
7. B.V. Dasarthy, "Decision fusion strategies in multi-sensor environments," to be published in *IEEE Trans. Syst. Man Cybernetics* **21**.
8. R. Hecht-Nielsen, *Neurocomputing*, Reading, Mass.: Addison-Wesley Publishing Co. (1990).
9. R.Y. Levine and T.S. Khuon, "Performance measures for adaptive decisioning systems," submitted to *Neural Networks* (April 1991).
10. R.P. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Mag.* **4**, 4-22 (April 1987).
11. M. Minsky and S. Papert, *Perceptrons*, Cambridge, Mass.: MIT Press (1969).
12. D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning internal representation by error propagation," in D.E. Rumelhart and J.L. McClelland (eds.) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, 1, Cambridge, Mass.: MIT Press (1986) pp. 318-362.
13. P.J. Werbos, "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences," Harvard University, Cambridge, Mass. (1974).
14. D.B. Parker, "Learning Logic," Center for Computational Research in Economics and Management Science, Massachusetts Institute of Technology, Cambridge, Mass., Technical Rep. 47 (April 1985).

REFERENCES

(Continued)

15. D.B. Parker, "Optimal algorithms for adaptive networks: Second order back propagation, second order direct propagation, and second order learning," *Proc. IEEE 1st Int. Conf. Neural Nets*, San Diego, Calif. (June 1987), pp. II593-II600.
16. A.E. Bryson and Y.C. Ho, *Applied Optimal Control* (Rev. 2nd Printing), New York: Hemisphere Publishing (1975).
17. J.J. Hopfield, "Neural networks and physical systems with emergent collective computation abilities," *Proc. Natl. Acad. Sci. USA* **79**, 2554-2558 (1982).
18. M. Cohen and S. Grossberg, "Absolute stability of global formation and parallel memory storage by competitive neural networks," *IEEE Trans. Syst. Man Cybernetics* **13**, 815-826 (1983).
19. T. Kohonen, *Self-Organization and Associative Memory*, 2nd Ed., Berlin: Springer-Verlag (1988).
20. W.Y. Huang and R.P. Lippmann, "Comparisons between neural net and traditional classifiers," *Proc. IEEE 1st Int. Conf. Neural Nets*, San Diego, Calif. (June 1987) pp. IV485-IV493.
21. W.Y. Huang and R.P. Lippmann, "Neural net and traditional classifiers," in D. Anderson (ed.), *Neural Information Processing Systems*, New York: American Institute of Physics (1988) pp. 387-396.
22. O.J. Murphy, "Nearest neighbor pattern classification perceptrons," *Proc. IEEE* **78**, 1595-1598 (October 1990).
23. H.-C. Yau and M.T. Manry, "Iterative improvement of a Gaussian classifier," *Neural Networks* **3**, 437-444 (1990).
24. I. Sethi and A.K. Jain (eds.), *Artificial Neural Networks and Pattern Recognition: Old and New Connections*, New York: Elsevier (1991).
25. Y. Yair and A. Gersho, "Maximum a posteriori decision and evaluation of class probabilities by Boltzmann perceptron classifiers," *Proc. IEEE* **78**, 1620-1628 (October 1990).
26. L.I. Perlovsky and M.M. McManus, "Maximum likelihood neural nets for sensor fusion and adaptive classification," *Neural Networks* **4**, 89-102 (1991).
27. M. Basseville and A. Benveniste, "Sequential detection of abrupt changes in spectral characteristics of noise," *IEEE Trans. Inf. Theory* **IT-28**, 318-329 (1982).
28. H. Van Trees, *Detection, Estimation, and Modulation Theory, Part I*, New York: J. Wiley and Sons (1971).

REFERENCES

(Continued)

29. R. Hecht-Nielsen, "Theory of back propagating neural nets," *Proc. IEEE 1st Int. Conf. Neural Nets*, San Diego, Calif. (June 1987), pp. I593-I611.
30. R.Y. Levine and T.S. Khuon, private communication.
31. P.D. Wasserman, *Neural Computing: Theory and Practice*, New York: van Nostrand Reinhold (1989).
32. J.L. Walker, "Range-Doppler imaging of rotating object," *IEEE Trans. Aerosp. Electron. Syst.* **AES-16**, 23-52 (January 1980).
33. W.M. Brown and R.J. Fredricks, "Range-Doppler imaging with motion through resolution cells," *IEEE Trans. Aerosp. Electron. Syst.* **AES-5**, 98-102 (January 1969).
34. D.A. Ausherman, A. Kozma, J.L. Walker, H.M. Jones, and E.C. Poggio, "Developments in radar imaging," *IEEE Trans. Aerosp. Electron. Syst.* **AES-20**, 363-400 (July 1984).
35. A.L. Kachelmyer, "Range-Doppler imaging with a laser radar," *Linc. Lab. J.* **3**, 87-118 (1990).
36. M. Jordan, private communication (March 1990).
37. K.I. Schultz, et al., private communication (October 1990).

APPENDIX A

VARIANCE TRANSITION DETECTION

Equations (3) and (4) are derived to relate detection and false alarm probabilities to the quantities $\{p(j|m)|j, m \in \{0, 1\}\}$. Recall that the indexes zero and one correspond to noise deviations σ_0 and σ_1 , respectively. The pair (i, j) denotes a transition from deviation σ_i to deviation σ_j , and the expression $p(x|y)$ denotes the probability of x detection conditioned on y . The relevant probabilities are then given by $P_d = p(\text{transition}|\text{transition})$ and $P_f = p(\text{transition}|\text{no transition})$ for detection and false alarm. The detection probability is given by

$$p(\text{transition}|\text{transition}) = p((1, 0)|\text{transition}) + p((0, 1)|\text{transition}). \quad (\text{A.1})$$

The application of Bayes theorem to Equation (11) yields the result

$$P_d = \frac{p((1, 0), \text{transition}) + p((0, 1), \text{transition})}{p(\text{transition})}, \quad (\text{A.2})$$

where $p(\text{transition})$ represents the prior probability of a transition, which is obtained either by a $(1, 0)$ or a $(0, 1)$ noise deviation pair. Equation (12) can be written in terms of the probability for specific deviation pair detection with the result

$$P_d = \frac{p((0, 1), (1, 0)) + p((0, 1), (0, 1)) + p((1, 0), (1, 0)) + p((1, 0), (0, 1))}{p((1, 0)) + p((0, 1))}, \quad (\text{A.3})$$

where $p((i, j))$ represents the prior probability of a deviation pair (i, j) . Application of Bayes theorem to Equation (13) results in the expression

$$P_d = \frac{[p((0, 1)|(1, 0)) + p((1, 0)|(1, 0))]p((1, 0))}{p((1, 0)) + p((0, 1))} + \frac{[p((0, 1)|(0, 1)) + p((1, 0)|(0, 1))]p((0, 1))}{p((1, 0)) + p((0, 1))}. \quad (\text{A.4})$$

Recall that $p((i, j)|(k, m))$ represents the detection of deviation pair (i, j) conditioned on the pair (k, m) . Assuming that the decision for this occurrence is based on a pair of maximum likelihood tests before and after the transition, the conditional probabilities factorize, that is, $p((i, j)|(k, m)) = p(i|k)p(j|m)$. Application of this property in Equation (14) results in the expression

$$P_d = p(1|1)p(0|0) + p(0|1)p(1|0), \quad (\text{A.5})$$

where $p(i|j)$ is given in Equations (5) and (6). It is interesting that the prior probabilities $p((i, j))$ have cancelled from Equation (15), indicating an overall detection probability independent of the prior distribution of deviation pairs.

The same argument applied to the false alarm probability results in the expression

$$P_f = \frac{p(1|1)p(0|1)p((1, 1)) + p(1|0)p(0|0)p((0, 0))}{[p((0, 0)) + p((1, 1))]/2}. \quad (\text{A.6})$$

In this case the probability depends on the prior probabilities $p((0, 0))$ and $p((1, 1))$ for the ensemble upon which the hypothesis test is applied. An ensemble in which all deviation pairs (i, j) are equally likely results in

$$P_f = p(1|1)p(0|1) + p(0|0)p(1|0). \quad (\text{A.7})$$

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE 5 September 1991		3. REPORT TYPE AND DATES COVERED Technical Report	
4. TITLE AND SUBTITLE Neural Net Sensor Fusion				5. FUNDING NUMBERS C — F19628-90-0002 PE — 12424F, 31310F, 63223C PR — 80	
6. AUTHOR(S) Robert Y. Levine and Timothy S. Khuon					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Lincoln Laboratory, MIT P.O. Box 73 Lexington, MA 02173-9108				8. PERFORMING ORGANIZATION REPORT NUMBER TR-926	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Strategic Defense Command CSSD-HSBD P.O. Box 1500 Huntsville, AL 35807-3801				10. SPONSORING/MONITORING AGENCY REPORT NUMBER ESD-TR-91-093	
11. SUPPLEMENTARY NOTES None					
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.				12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) A generic architecture for neural net multisensor data fusion is introduced and analyzed. The architecture consists of a set of independent sensor neural nets, one for each sensor, coupled to a fusion net. Each sensor is trained (from a representative data set of the particular sensor) to map to a hypothesis space output. The decision outputs from the sensor nets are used to train the fusion net to an overall decision. In this report the sensor fusion architecture is applied to the stochastic exclusive-or problem for a benchmark comparison with classical hypothesis testing. The architecture is also applied to a data fusion experiment involving the multi-sensor observation of object deployments during the recent Firefly launches. The deployments were measured simultaneously by X- and L-band and CO ₂ laser radars. The range-Doppler images from the X-band and CO ₂ laser radars were combined with a passive-IR spectral simulation of the deployment to form the data inputs to the neural sensor fusion system. The network was trained to distinguish predeployment, deployment, and postdeployment phases of the launch based on the fusion of these sensors. The success of the system in utilizing sensor synergism for an enhanced deployment detection is clearly demonstrated.					
14. SUBJECT TERMS neural net sensor fusion range-Doppler imaging passive-IR sensor X-band radar CO ₂ laser radar				15. NUMBER OF PAGES 50	
				16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT SAR		