Proceedings of the 1996 IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems

Neural Network Based Sensor Array Signal Processing

Dukki Chung

Dept. of Electrical Engineering and Applied Physics
Case Western Reserve University
Cleveland, Ohio 44106. U.S.A.

Francis L. Merat

Dept. of Electrical Engineering and Applied Physics
Case Western Reserve University
Cleveland, Ohio 44106, U.S.A.

Abstract

An autoassociative memory using neural networks is proposed for sensor failure detection and correction. A classical approach to sensor failure detection and correction relies upon complex models of physical systems; however, a neural network approach can be used to represent systems through training for which mathematical models can not be formulated. In such cases, a neural network autoassociative memory can be used to predict sensor outputs. Differences between measured sensor outputs and sensor outputs estimated by the autoassociative memory can be used to identify faulty sensors. Median filtering or other signal processing schemes may then be used to correct faulty sensor outputs. This technique can be used to process data from MEMS (Micro Electromechanical systems) or other sensor arrays.

1 Introduction

Sensor failure detection and correction is a very important subject for a variety of applications. If one or more sensors in a system fail or are failing, they should be identified and isolated. Whenever a sensor(s) is not functioning correctly, there should be a way to deal with and potentially correct for this inaccurate sensor data. In conventional approaches dealing with inaccurate sensor outputs, there must be an appropriate mathematical model of the system. However, it may not be possible to develop an accurate model for the system, or, it may not be possible to mathematically solve the model. In either case, conventional systems will either perform very poorly or simply be impossible to implement. The rapid development of solid state sensors, and especially MEMS based sensors, allows the development of new signal processing paradigms.

We propose an autoassociative memory using neural networks for sensor failure detection and correction. An artificial neural network can learn the characteristics of a non-linear, unknown/unmodeled system through training samples. Comparison of the autoassociative memory filtered sensor signals with the raw sensor outputs can be used to determine which sensor(s) is(are) faulty. Median filtering or other signal processing schemes may then be used to correct the faulty sensor outputs. Such an autoassociative memory can also be used as a filter to minimize the effects of noise and other external disturbances. This neural network autoassociative memory technique can be successfully used to process data from MEMS or other solid-state sensor arrays.

2 Conventional Approach

Since sensors are typically the least reliable components in a control system and are subjected to harsh conditions, some form of redundancy is necessary to achieve adequate reliability in many control situations.

Hardware redundancy uses multiple sensors to measure system variables. Voting schemes compare multiple output values from the sensors and can be used to detect and isolate faulty sensors. Since multiple sensors are deployed, there can be a substantial increase in weight, cost, and space in a physically redundant system.

Analytical redundancy (AR) uses a reference model for the system and redundant information from dissimilar sensors to provide an estimate of measured variables. Analytic formulas must be developed which describe the system. Differences between measured sensor outputs and estimated sensor outputs, called residuals, are used to identify faulty sensors. This residual generation is typically based on knowledge of the system [1, 2, 3, 4]. If such knowledge is not available, analytical redundancy can not be used or will perform very poorly.

3 Autoassociative Memory Using Neural Networks

Neural networks are commonly used for classification and functional approximation. A neural network with one hidden layer with sufficient nodes can be thought of as a universal function approximator [5, 6]. The network learns a mapping from the given inputs to the desired outputs through training samples. The neural network can learn autoassociation from training samples. The mapping is apparent, i.e. the network output should be equal to the network input at all times. The autoassociative

0-7803-3700-X/96 $5.00

memory is useful because it can correct noise, distortion and/or partial input values.

Several researches have used a three layer network (one hidden layer) to implement an autoassociative memory. Baldi and Hornik [7] showed that a three layer network is equivalent to PCA (Principal Component Analysis) projection. Later, Bourlard [8] showed that the three layer network is not superior to PCA. Recently, several researchers have shown that a five layer (three hidden layers) network can improve on PCA, as a non-linear Principal Component Analysis [9, 10, 11, 12].

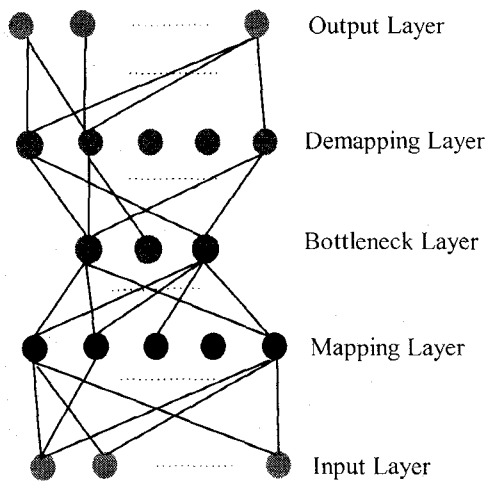


Figure 1. Three hidden layer autoassociative neural network

A three hidden layer autoassociative memory can be viewed as a serial combination of two single hidden layer networks. The input, mapping, and bottleneck layers represent a nonlinear function or mapping, which projects the input data into a lower dimension feature space. The bottleneck layer, demapping layer, and the output layer represent a second network that remaps an approximation of the input from the feature space of the bottleneck layer output.

Since the bottleneck layer has fewer nodes than the input and output layers, the network is constrained to develop a lower dimensional representation in the bottleneck layer. This three hidden layer autoassociative neural network works as a nonlinear compressor-decompressor pair [12, 13]. This type of autoassociative memory is quite difficult to train, since many nonlinear nodes are involved in the training, and it often fails to converge to an acceptable training error. If a large number of inputs are involved in the problem, a three hidden layer autoassociative memory may not be a practical solution.

4 Autoassociative Memory Using Random Vector Enhanced Phasor Neural Networks

For implementing an autoassociative memory, a random vector enhanced phasor neural network (RV-PNN) was used [14]. In an RV-PNN autoassociative memory input patterns are enhanced by the multiplication of the input patterns and randomly chosen vectors. These random vector enhancements are further discussed in [15, 16].

Suppose that there are n original attributes (features, or elements) and j augmented attributes in the real domain pattern $\boldsymbol{x}$. For the augmented attribute x_i, the attribute is defined as

$$x_i = e_i \quad \text{for } i = n+1, \ldots, n+j \tag{1}$$

where $e_i = a_{i1}x_1 + \ldots + a_{in}x_n = \boldsymbol{a}_i^T\boldsymbol{x}$

$\boldsymbol{a}_i = [\, a_{i1} \;\; a_{i2} \;\; \ldots \;\; a_{in} \,]^T$

$\boldsymbol{x} = [\, x_1 \;\; x_2 \;\; \ldots \;\; x_n \,]^T$

and $\boldsymbol{a}_i$ is the random vector. The elements of the vector $\boldsymbol{a}_i$ are randomly generated on the real interval $[-\Omega, \Omega]$.

These augmented patterns are then transformed into complex vectors. One way to convert the pattern attributes (real numbers) to complex numbers is to assign a phase angle to each attribute and give it a unit magnitude. The resulting attributes are on the unit circle in the complex plane (phasors). A phase angle can be assigned by using the Z-score as

$$\theta = \frac{2\pi}{1 + e^{-\frac{x-\mu}{\sigma}}} \tag{2}$$

where μ and σ are, respectively, the mean and standard deviation of the input key vectors. Equation (2) converts unbounded real values into phase angles from 0 to 2π. For the complex number input patterns, complex number random vectors are generated over a magnitude range which avoids saturation of the transfer function. After the enhancement by equation (1), these patterns are fed into a sigmoidal transfer function such as $tanh(real(x)) + i\, tanh(imag(x))$.

These transformed patterns then form the linear associative memory via the Moore-Penrose pseudoinverse. In the linear associative memory model, patterns are linearly transformed by a relation of the form

$$\mathbf{y}_i = \mathbf{M}\mathbf{x}_i \qquad \text{for } i=1, \ldots, p \tag{3}$$

where $\mathbf{M}$ is the memory matrix, $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ is the i-th associated pair of patterns, and p is the total number of associated patterns. In terms of the key vector matrix $\mathbf{X}$ and the recollection vector matrix $\mathbf{Y}$, the associative memory must satisfy $\mathbf{Y} = \mathbf{M}\mathbf{X}$. Generally $\mathbf{X}$ is not square, and, thus, the Moore-Penrose pseudoinverse is used for the memory matrix $\mathbf{M}$ [17]

$$\mathbf{M} = \mathbf{Y}\mathbf{X}^+. \tag{4}$$

If such a pseudoinverse approach is not possible, gradient-descent learning can be used instead [14], as

$$\mathbf{M}^{new} = \mathbf{M}^{old} + \alpha(\mathbf{Y} - \mathbf{M}^{old}\mathbf{X})\mathbf{X}^{*} \quad (5)$$

where * denotes the conjugate transpose, and α is the learning rate (a small positive number). By equation (5), it is possible to construct the pseudoinverse based **M** matrix iteratively. Since the learning is linear, this approach is guaranteed to find the global minimum.

When dealing with large amounts of input data, the training time can be a critical issue for neural networks. Since training a three hidden layer network, or a single hidden layer network, for large number of input nodes is not trivial, RV-PNNs can be used instead of standard backpropagation neural networks to dramatically decrease the necessary training time.

5 Experimental Results

5.1 Overview of Experiments

Photoelasticity can be used to accurately measure surface strains or stresses in a part or structure. A strain sensitive (photoelastic) plastic cylinder is attached to the shaft and illuminated by polarized light [18]. As the shaft torque varies the photoelastic plastic displays the corresponding shaft strain as a 2-D fringe pattern when viewed through an optical polarizer. The strain that causes this observed optical pattern is some complex function of the torque applied to the shaft.

A single hidden layer neural network was trained and tested as a torque estimator using data from a linear optical sensor array which consisted of 32 sensors. There was no known analytical relationship between a given input (optical pattern) and an estimated torque for this sensor array making this sensor array ideal for neural network signal processing. The performance of a single hidden layer neural network torque estimator was quite satisfactory (less than 0.4% average estimation error) [18]. This result is shown in Figure 2.

A stuck-at sensor failure is defined to be a sensor failure where the sensor is stuck at one extreme of its signal range. In practice, this is likely to be an open (stuck-at 0) sensor or a short circuited (stuck-at 1) sensor. Such stuck-at sensor failures can be detected and corrected by using an autoassociative memory. It is a two step process. First, residuals are generated by taking the differences between the output of the sensors and the output of the autoassociative memory. If the residuals of particular sensors are greater than a certain threshold limit, the corresponding sensors are regarded as failed. This can also be verified by comparing the sensor outputs with the operational range of the sensor output. These residuals can be further processed using statistical decision theory, e.g., the sequential probability ratio test [19]. Since failed sensors do not produce any useful data, this can be regarded as a missing data problem where identified stuck-at sensor outputs can be replaced by the mean values for the failed sensors [20, 21, 22]. If the mean values are not known, the medians of the sensor output range can be used instead.

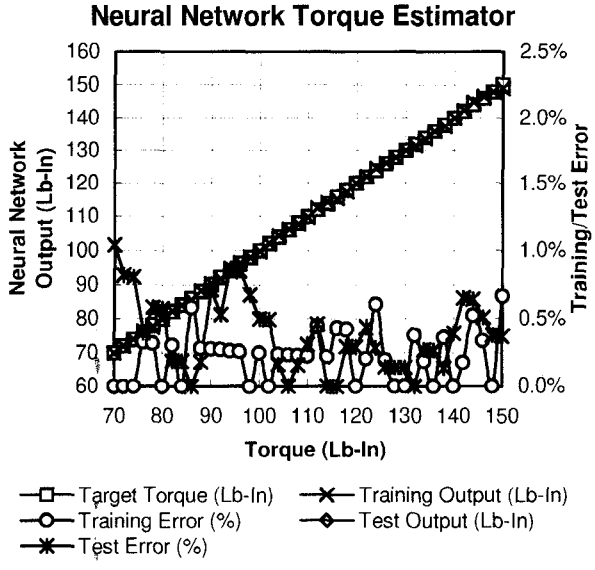


Figure 2. Result of the Neural Network Torque Estimator

5.2 Sensor Failure Detection and Correction

After any substitutions for failed sensors, the sensor data is fed into the autoassociative memory and estimated sensor outputs are reproduced. These reproduced sensor outputs can be used for later processing (e.g., estimation of torque) with more accuracy.

Figure 3 shows sensor failure detection and correction by the autoassociative memory of the individual sensor when 12 out of 32 sensors are stuck-at 1. Two different autoassociative memory architectures (a three hidden layer neural network and an RV-PNN) were implemented for the same data. Without autoassociative memory preprocessing, the neural network torque estimator has an error of 42.9% for this particular input. With the RV-PNN autoassociative memory, the test error decreases to 0.4%.

Figure 4 shows the effect of the autoassociative memory when 16 out of 32 sensors are stuck-at 1. Without the autoassociative memory, the neural network torque estimator shows an average estimation error of 24.3% over the given torque range. With the autoassociative memory used as a filter between the sensor outputs and the neural network torque estimator inputs, the average torque estimation error decreases to 5.2%.

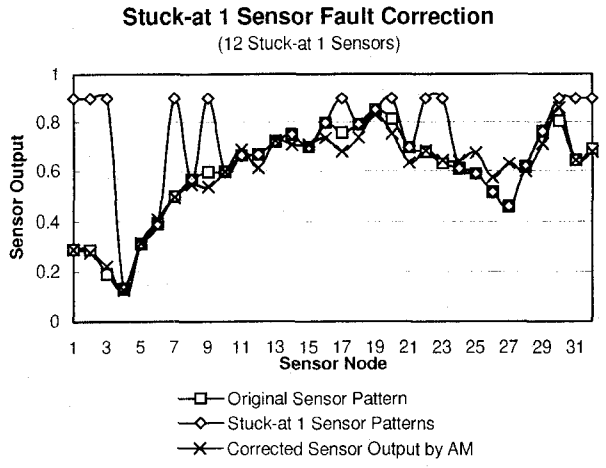


Figure 3. Sensor Failure Detection/Correction for Stuck-at 1 Sensor Faults

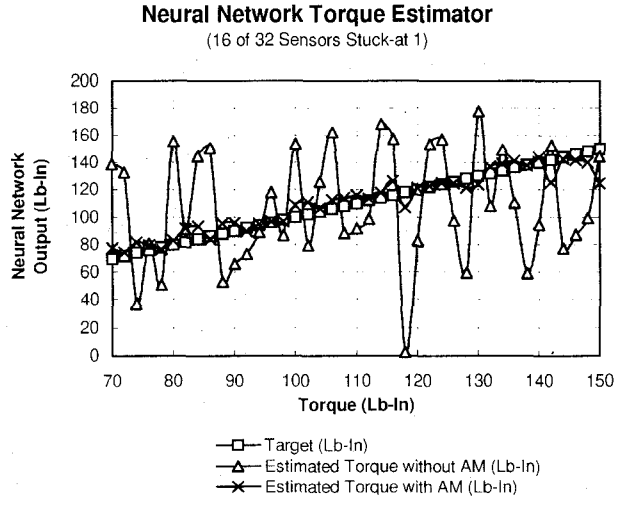


Figure 4. Test Result of Stuck-at 1 Sensor Failure

Table 1 shows the performance of various autoassociative memory architectures for stuck-at 0 failures as a function of the number of stuck-at sensor failures.

Table 2 shows the performances of various autoassociative memory architectures for stuck-at 1 failures as a function of the number of stuck-at sensor failures.

5.3. Sensor Processing in Noisy Environment

Figure 5 shows the method used to test the performance of the various autoassociative memory architectures. Gaussian random noise was added to the relatively noise-free data from the optical sensor array. This noisy data was then filtered by two types of autoassociative memory. The neural network torque estimator was tested for filtered and unfiltered data.

Table 3 shows the noise filtering performance of the RV-PNN autoassociative memory, the three hidden layer neural autoassociative memory, and the neural network torque estimator performance without the associative memory while decreasing the S/N ratio.

Table 1. Stuck-at 0 Performances of Various Autoassociative Memory Architectures

AM Architecture	Number of Stuck-at 0 Sensor Failures			
	4	8	12	16
Torque Estimation with RV-PNN AM	1.4%	2.9%	4.5%	5.7%
Torque Estimation with 3 Hidden Layer AM	1.3%	2.8%	4.3%	5.5%
Torque Estimation with No AM filtering	9.9%	13.2%	15.6%	15.8%

Table 2. Stuck-at 1 Performances of Various Autoassociative Memories

AM Architecture	Number of Stuck-at 1 Sensor Failures			
	4	8	12	16
Torque Estimation with RV-PNN AM	1.4%	2.4%	3.0%	3.7%
Torque Estimation with 3 Hidden Layer AM	1.3%	2.2%	3.2%	5.1%
Torque Estimation with No AM filtering	14.4%	18.6%	20.2%	22.5%

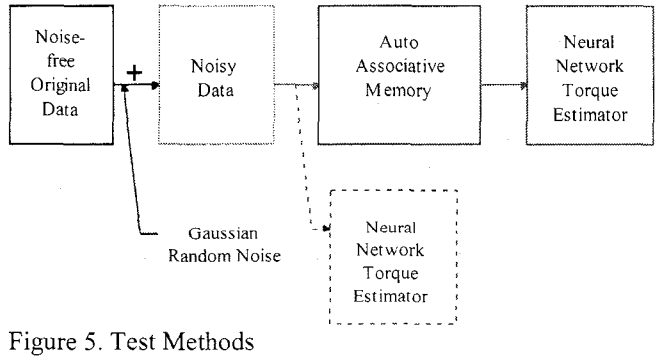


Figure 5. Test Methods

Even at a S/N of -2.2dB, the average estimation error of the neural network torque estimator was 7.1% with the RV-PNN autoassociative memory and 7.5% with the three hidden layer autoassociative memory filters.

Table 3. Noise Filtering of Autoassociative Memory

AM Architecture	S/N			
	3.5dB	0.75dB	-1.0dB	-2.2dB
Torque Estimation with RV-PNN AM filtering	2.2%	4.2%	5.7%	7.1%
Torque Estimation with Three Hidden Layer AM filtering	2.5%	3.9%	5.3%	7.5%
Torque Estimation with No AM filtering	12.2%	17%	17.4%	21.5%

5.4 MEMS Sensor Processing Simulation

Micro Electromechanical Systems (MEMS) are being actively pursued for the realization of high-density array architectures of microfabricated sensors and actuators for distributed monitoring and control of thermal, structural, and aerodynamic parameters within a system. The requirements for such sensor arrays are that they be low-cost, reliable, high density integrated sensor/actuator arrays capable of real-time signal processing and control. We envision an integration of materials and distributed MEMS to locally manipulate the environment for real-time control of thermal, structural, and/or aerodynamic behavior of a system. For example, using a network of microfabricated pressure sensors, local flow instabilities can be detected in real time and suppressed by appropriately opening or closing localized microvalves. In terms of a particular application, we envision an ice detection and removal system for gas turbine engines which utilizes a network of resonant diaphragm microsensors to detect ice formation. This information could then be used to open microvalves which locally route warm compressor bleed air to melt the ice. This ice detection system could also be used for monitoring ice formation on helicopter rotor blades. Sensor array information could also be used to control local shape memory alloy microactuators that flex to break the ice.

The actual development of such MEMS based systems is underway in many laboratories; however, the sensor signal processing and especially the reliability and accuracy of the sensor array is very important to the aircraft applications described. Such applications seem to be especially well suited to neural network signal processing of the type described in the previous section. The principle difference is that envisioned MEMS systems are often two-dimensional. To test the applicability of neural network sensor processing and control in MEMS systems we are developing a test chip which will consist of a two dimensional array of semiconductor heaters and thermal sensors on a silicon wafer. The thermal sensors will be interleaved between the resistive heaters as shown in Figure 6. The long-term goal of this work is to use the heater/sensor array with a 2-D feedback control algorithm to precisely generate two-dimensional thermal profiles even if one or more heater elements have failed. While this chip is being designed and fabricated we have developed simple computer models to test our signal processing and control algorithms.

Only the sensor signal processing aspects of this work will be described in this paper. In our MEMS sensor simulation, we deploy a 4x4 array of thermal sensors between a 3x3 array of resistive heaters as shown in Figure 6. Our goal is to detect and/or correct sensor and heater failures in the simulated chip using autoassociative neural network signal processing. For purposes of simulation we have treated the heaters as impulse type thermal sources and solved the heat equation (6) for the silicon wafer subject to boundary conditions (7), i.e., the boundary temperature is equal to that of environment, and is fixed [23]. A finite element method is used to solve (6).

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = \frac{\partial T}{\partial t} \tag{6}$$

$$T_b = T_{env} = constant \tag{7}$$

This simulation generates a steady state temperature profile of the silicon wafer which is then sampled by the thermal sensors.

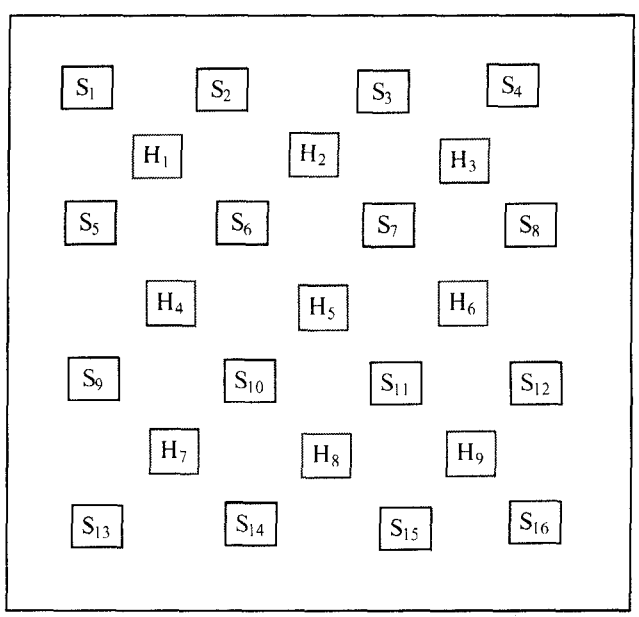


Figure 6. Heater(H) / Sensor(S) configuration

Figure 7 shows a typical temperature profile generated by the simulation. The profile is shown a long time after the heat sources have been turned off and the wafer is in thermal steady state. The issue is whether the

two-dimensional sensor array can be used to detect heater failures.

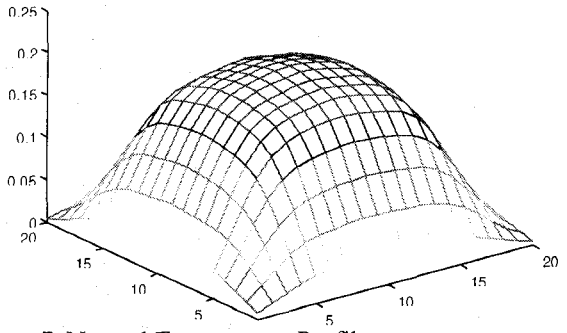
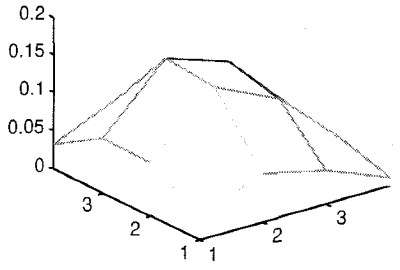


Figure 7. Normal Temperature Profile

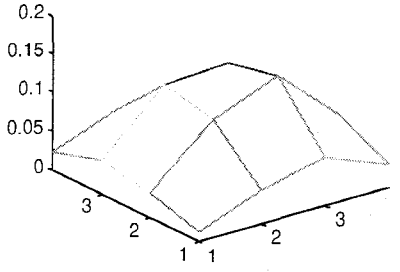
Figure 8 shows temperature profiles as measured by the sensor array for various combinations of individual heater failure. Here a heater failure is defined as an open, i.e., no heating. This is similar to the stuck-at-0 sensor faults described previously.

The steady-state temperature profiles corresponding to all possible heater failures were generated. The sensor signals from these temperature profiles were then used to train a neural network with the goal of detecting heater failure states. A random vector enhanced functional-link net was used in this experiment for faster training [15, 16]. Gaussian random noise was added to the temperature profiles during testing to simulate sensor noise. Table 4 shows the performance of a neural network heater failure detection system as a function of sensor S/N ratio. Note that this particular simulation is only examining heater failures, not sensor failures.

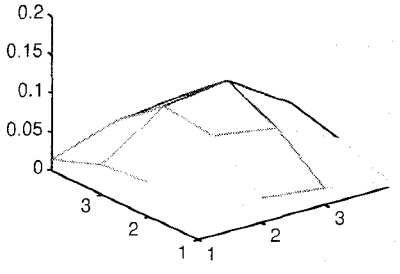
These results are excellent and show that neural network signal processing is promising for MEMS sensor/actuator applications such as in aerospace and smart materials.



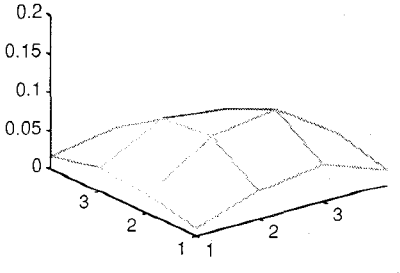
(a) Heater 3, 5 failed



(b) Heater 1, 4, 8 failed



(c) Heater 2, 3, 5, 6 failed



(d) Heater 1, 3, 7, 8, 9 failed

Figure 8. Sensor Array Temperature Profiles for Various Heater Failures

5 Discussion

We have developed a neural-network based sensor processing system which can correct for sensor noise, drift and failure. An autoassociative memory uses redundant information from a sensor array to provided corrected sensor outputs. A second neural network combines the sensor information to estimate the required system parameters. This neural network approach avoids the use of complex analytical models and exploits the information redundancy of a sensor array.

Table 4. Heater Failure Detection Result

Sensor S/N	No noise	4.3dB	-1.7dB	-3.5dB	-4.7dB
Percentage Correct Detection of Failed Heaters	100%	99.9%	91.4%	83.6%	80.2%

In order to detect sensor failures, outputs from the sensors are initially processed by the autoassociative memory and sensor residuals are calculated. Conventional residual generation is typically based on analytical knowledge of the system. However, an autoassociative memory can represent the system through learning and can be used to describe non-linear systems. The residuals can be processed using methods like thresholding or statistical decision theory to identify a particular sensor failure.

We have tested this approach on a one dimensional (i.e., linear) optical sensor array. An autoassociative memory was implemented for a 32 element optical sensor array using real data for functional estimation. With no sensor noise the system was able to estimate torque (the sensed parameter) to an accuracy of about 1%. Computer simulations were done adding independent Gaussian noise to the sensor outputs. Without using autoassociative memory signal processing, the estimation error dropped to 17% for a sensor S/N of 0.75dB. With the autoassociative memory the estimation error was only about 4% under the same conditions.

This neural network approach has several advantages which are important to sensor arrays such as might be implemented using MEMS technology: (1) it dramatically reduces the effects of individual sensor noise; (2) it accommodates sensor-to-sensor variation in arrays by treating the variation as noise; (3) it should be capable of multi-sensor fusion.

We have simulated a MEMS array of thermal sensors and are in the process of collecting data from an actual array of MEMS sensors for comparative proposals. We are also fabricating a specialized silicon chip which will incorporate thermal heaters and sensors.

Our simulations show that neural network signal processing can be used to isolate faulty sensors. Simulations and experiments in noise, sensor failure and sensor tolerance for MEMS sensor arrays are in progress.

If the performance of the neural network sensor array processing can be derived analytically, the proposed method can also be applied to very hard problems, e.g. jet engine or space craft control, which require performance bounds upon the effects of sensor failures.

Acknowledgements

The heater/sensor simulation was performed by Zhou Fu. Parts of this work have been supported by Reliance Electric Division of Rockwell and the US Army under research grant DAAH04-95-1-0097.

References

[1] Chow, E. and A. S. Willsky, "Analytical Redundancy and the Design of Robust Failure Detection Systems", in *IEEE Transactions on Automatic Control,* vol. AC-29, no. 7, pp. 603-614, Jul 1984.

[2] Merril, W. C., J. C. DeLaat, and M. Abdelwahab, "Turbo Engine Demonstration of Sensor Failure Detection," in *J. Guidance*, vol. 14, no. 2, pp. 337-349, Mar-Apr 1991.

[3] Piercy, N. P., "Sensor Failure Estimators for Detection Filters," in *IEEE Transactions on Automatic Control,* vol. 37, no. 10, pp. 1553-1558, Oct 1992.

[4] Zimmerman, D. C. and T. L. Lyde, "Sensor Failure Detection and Isolation in Flexible Structures Using System Realization Redundancy," in *J. Guidance, Control and Dynamics,* vol. 6, no. 3, pp. 490-497, May-Jun 1993.

[5] Hornik, K., M. Stinchombe, and H. White, "Multilayer Feedforward Networks are Universal Approximators," in *Neural Networks,* vol. 2, pp. 359-369, 1989.

[6] Funahashi, K., "On the Approximation Realization of Continuous Mappings by Neural Networks," in *Neural Networks*, vol. 2, pp. 183-192, 1989.

[7] Baldi, P. and K. Hornik, "Neural Networks and Principal Component Analysis: Learning form Examples without Local Minima,", in *Neural Networks*, vol. 2, pp. 53-58, 1989.

[8] Bourlard, H. and Y. Kamp, "Auto-association by Multilayer Perceptrons and Singular Value Decomposition," in *Biol. Cyb.*, vol. 59, pp. 291-294, 1988.

[9] Nampol, A., M. Arozullah, and S. Chin, "Higher Order Data Compression with Neural Networks,", in *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, pp. I-55 - 159, 1991.

[10] Kramer, M. A., "Nonlinear Principal Component Analysis using Autoassociative Neural Networks," in *AIChE Journal*, vol. 37, pp. 233-243, 1991.

[11] Oja, E., "Data Compression, Feature Extraction, and Autoassociation in Feedforward Neural Networks," in *Artificial Neural Networks*, pp. 737-745, Elsevier Science Publishers B. V., 1991.

[12] Kramer, M. A., "Autoassociative Neural Networks," in *AIChE Journal*, vol. 16, pp. 313-328, 1992.

[13] Meng, Z, Investigation of Efficient Optimization through Dimension-Reduction Mapping, Master Thesis, Dept. of Electrical Engineering and Applied Physics, Case Western Reserve University, 1996.

[14] Chung, D. and F. L. Merat, "Associative Memory Using Random Vector Enhanced Phasor Neural Network," in *Proceedings of the OAI Neural Network Symposium and Workshop*, pp. 267-278, 1995.

[15] Pao, Y. H., G. H. Park and D. J. Sobajic, "Learning and Generalization Characteristics of the Random Vector Functional-Link Net,", in *Neuro Computing*, vol. 6, pp. 163-180, 1994.

[16] Igelnik, B. and Y. H. Pao, "Stochastic Choice of Basis Functions in Adaptive Function Approximation and the Functional-Link Net," in *IEEE Transactions on Neural Networks*, vol. 6, no. 6, pp. 1320-1329, 1995.

[17] Penrose, R., "A Generalized Inverse for Matrices," in *Proc. Cambridge Philos. Soc.* (England), LI, pp. 406-413, 1955.

[18] Chung, D., F. L. Merat, F. Discenzo and J. Harris, "Neural Net Based Torque Sensor Using Birefringent Materials," to be submitted to *Sensors and Actuators* , 1996.

[19] Wald, A., *Sequential Analysis*, Wiley & Sons, Inc., 1947.

[20] Beale, E. M. and R. J. A. Little, "Missing Values in Multivariate analysis," in *J. R. Statist. Soc. B*, 37, pp 129-145, 1975.

[21] Jolliffe, I. T., *Principal Component Analysis*, Springer-Verlag, 1986.

[22] Jackson, J. E., *A User's Guide to Principal Components*, John Wiley & Sons, 1991.

[23] White, F. M., *Heat Transfer*, Addison-Wesley, 1984.