INSTRUMENT SURVEILLANCE AND CALIBRATION VERIFICATION THROUGH PLANT WIDE MONITORING USING AUTOASSOCIATIVE NEURAL NETWORKS

DARRYL J. WREST
J. WESLEY HINES
ROBERT E. UHRIG

The University of Tennessee,
Department of Nuclear Engineering,
Knoxville, Tennessee, 37916, (423) 974-3110

Abstract

The approach to instrument surveillance and calibration verification (ISCV) through plant wide monitoring proposed in this paper is an autoassociative neural network (AANN) which will utilize digitized data presently available in the Safety Parameter Display computer system from Florida Power Corporations Crystal River #3 nuclear power plant. An autoassociative neural network is one in which the outputs are trained to emulate the inputs over an appropriate dynamic range. The relationships between the different variables are embedded in the weights by the training process. As a result, the output can be a correct version of an input pattern that has been distorted by noise, missing data, or non-linearities. Plant variables that have some degree of coherence with each other constitute the inputs to the network. Once the network has been trained with normal operational data it has been shown to successfully monitor the selected plant variables to detect sensor drift or failure by simply comparing the network inputs with the outputs. The AANN method of monitoring many variables not only indicates that there is a sensor failure, it clearly indicates the signal channel in which the signal error has occurred.

1. INTRODUCTION

Traditional approaches to instrument calibration at nuclear power plants are expensive both in labor and money. These approaches vary from calibration by replacement, to transfer calibration using standard instruments. Technical Specifications require specific instruments be calibrated on time tables that coincide with the original fuel cycle of the plant. These calibrations require that the instrument be taken out of service and be falsely-loaded to simulate actual in-service stimuli. This can lead to damaged equipment and incorrect calibrations due to adjustments made under non-service conditions. While proper adjustment is vital to maintaining proper plant operation, a less invasive technique is desirable.

As utilities move to 24 month fuel cycles, there is a needfor performance based calibration requirements. When implementing performance based calibrations, the instruments are calibrated only when necessary. Monitoring instruments for calibration performance will allow nuclear utilities to reduce the efforts necessary to assure the instruments are calibrated. Benefits include an industry wide cost savings, less time for reactor startup, and easier compliance with NRC Generic Letter 91-04 for extending calibration intervals.

The use of AANNs for plant wide monitoring was developed by the University of Tennessee and reported in NUCLEAR TECHNOLOGY.[1] This work, using data from the Experimental Breeder Reactor II, has demonstrated the practicality of this application. Related work includes monitoring of the Borssele Nuclear Power Plant using AANN techniques.[2] Similar work using AANN applied to chemical process systems have also been reported.[3,4,5] This work further advances the methodology by

introducing a robust training method and use of the sequential probability ratio test (SPRT) as a sensor failure detection device.

An autoassociative neural network (AANN) is one in which the outputs are trained to emulate the inputs over an appropriate dynamic range. Many plant variables that have some degree of coherence with each other constitute the inputs. During training to make each output equal to the corresponding input, the interrelationship between the variables is embedded in the connection weights. As a result, any specific output shows virtually no change when the corresponding input pattern has been distorted by noise, missing data, or non-linearity's. This characteristic allows the AANN to detect drift or failure by comparing the sensor output with the corresponding network estimate.

2. PARAMETER SELECTION

Previous studies indicate that the degree of correlation between the input variables is relatively significant for an autoassociative neural network to function as a complex system monitoring device.[1,3] Under these circumstances, changes in one input variable due to drift, channel deterioration, or failure will not significantly change the corresponding value of the output because the output is related to all the other correlated input variables through a large number of paths and weights.

Parameters were selected based on plant safety points that require close monitoring (labeled Tech spec.), and the degree of coherence between the parameters statistically determined using linear correlation coefficients given by:

$$S(i,j) = \frac{C(i,j)}{\sqrt{C(i,i)C(j,j)}}$$

where C is the covariance matrix.

The coefficients range from ± 1 with the highly correlated parameters at the boundaries of the range. A value near ± 1 means there is a direct relationship between the change of a parameter with respect to another. If two signals showed no change with time (flat), there correlation would be near zero. The average coefficient value for the network was 0.28. Table 1 shows the 22 plant parameters selected for this study.

This grouping primarily consist of primary side parameters such as flows, pressures, and temperatures. The network includes 14 parameters listed as "Tech spec" safety parameters that must be monitored closely.

3. NETWORK ARCHITECTURE

The network architecture selected is a three hidden layer feedforward AANN proposed by Kramer[3]. It consist of an input layer, 3 hidden layers, and an output layer. The architecture is shown in Figure 1. The first of the hidden layers is the *mapping layer* with dimension greater than the number of input/outputs. The second hidden later is called the *bottleneck layer*. The dimension of this layer is required to be the smallest in the network. The third hidden layer is called the *demapping* layer that has the same dimension as the mapping layer. Kramer points out that five layers are necessary for such nets in order to model non-linear processes.

The three hidden layers form a "feature detection" architecture in which the bottleneck layer plays the key role in the identity mapping. The mapping layer maps from the input data space to the non-linear principle component score space (bottleneck layer), and the demapping layer map from the non-linear principle component score space to the data space (network output) corrected by the non-linear principle components.

The bottleneck layer prevents a simple one-to-one mapping from developing during network training. If the network learned the identity function exactly, it would be of little value. The bottleneck layer provides an internal constraint that prevents the net from strictly learning the identity mapping. Essentially the bottleneck layer functions like a non-linear principle component analysis (NLPCA) filter that gives a richer representation of the data: using a lower dimension to explain maximum information. The outputs of the bottleneck layer are non-linear principal components, which have a clear interpretation in theory.

The mapping-bottleneck-demapping combination forces the network to develop a compact representation of the

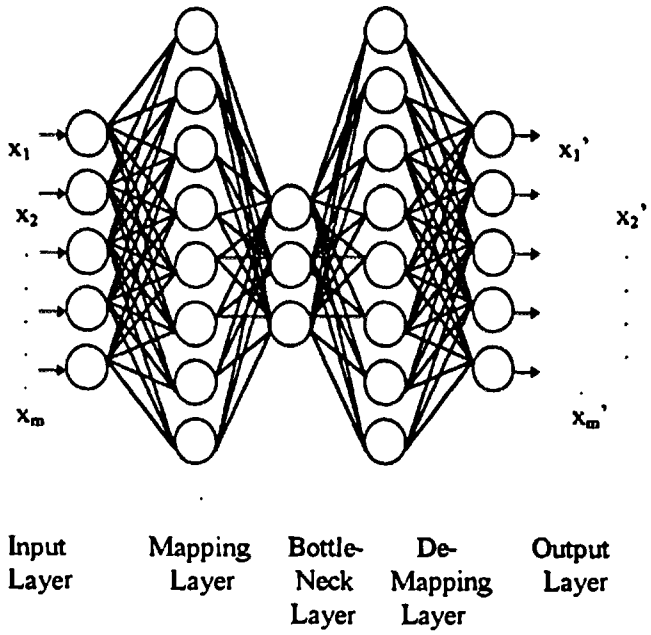

Figure 1. Architecture of the AANN

training data that better models the underlying system parameters.

The non-linear transfer function of the three hidden layers are sigmoidal logistic functions given by:

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

The network uses a linear output layer which allows for a mathematical regression technique (SVD) to solve for the output weights.

3.1. Selection of Hidden Layer Nodes

In the combined network, let m be the number of nodes in the input and output layers, f the number of nodes in the bottleneck layer, and M the number of nodes in the mapping and de-mapping layers. The number of input and output nodes (m) depends on the number of input/output parameters in the network. However, there is no definitive method of deciding a priori the number of nodes in the mapping and de-mapping layers (since the number of mapping nodes is equal to the number of de-mapping nodes, from this point on they will be referred to as mapping nodes). Generally, the number of mapping layer nodes is related to the complexity of the non-linear parameters that can be modeled by the network. To few nodes and the accuracy could be low due to the limited representational capacity

(to few degrees of freedom) of the network. If there are to many nodes, the network is prone to overfitting, or learning the stochastic variations in the data rather than the underlying functions. A simple approach for determining the number of mapping layer nodes is restrict the number of weights in the network to a fraction of the number of constraints imposed by the network. For the combined architecture, assuming all nodes have biases, the number of adjustable parameters in the network is given by the inequality[4]:

$$2M << m(n - f)/(m + f + 1)$$

where n is the number of training patterns. Kramer gives two other methods that may be used to determine the optimum number of mapping layer nodes; the Final Prediction Error (FPE), and Akaike's Information theoretic Criterion (AIC).

Recall that the bottleneck layer of the network functions as a Non-Linear Principle Component Analysis filter (NLPCA). The bottleneck layer requires as many nodes as there are non-linear factors in the parameters that are modeled. Therefore, the number of bottleneck nodes can be determined using an algorithm that statistically determines the principle components in a data set.

The non-linear approach is the same as the linear approach except that the data is summarized with a smooth curve (instead of a straight line) which is determined by non-linear relationships among all the variables. Given a data set X which contains n samples of m parameters can be expressed in terms of l non-linear principal components (number of f non-linear bottleneck nodes) as:

$$X = F(T) + E$$

where T is defined as the non-linear principal component scores, and F is defined as the non-linear principle component loading functions[5]. An example of such an algorithm to determine the non-linear principle components can be found in Hastie and Stuetzle[6].

Due to repeated training and testing trials, empirical formulas have been arrived at to determine the optimal number of hidden nodes in each of the hidden layers. They are given as ratios of one layer to another (for example input to bottleneck) and are listed below. Theses are only in approximate ratios based on the parameter groupings used this project.

$$M/m \cong 1.25 \qquad M/f \cong 1.6 \qquad m/f \cong 1.3$$

These were used in conjunction with Kramer's inequality equation given above to determine the number of hidden layer nodes for each of the networks.

The dimensions of the network presented here are (22-27-18-27-22), the number of inputs, mapping, bottleneck, demapping, and output nodes respectively.

4. NETWORK TRAINING

The network was trained using MATLAB'S[7] fast backpropagation training algorithm which incorporates an adaptive learning rate and momentum. Input and target vector were linearly scaled between 0.1 and 0.9 to simplify training. 425 representative training patterns were selected from a possible 3344 patterns recorded every 15 minutes (5 weeks of operation) during full power plant operation. Two transients occurred during the full power operation and were trained into the network.

Since a linear output layer was used, training was greatly accelerated by solving for the output layer weights using a singular valued decomposition mathematical technique (SVD) instead of iterative training. By using the SVD method to approximate the optimal weights, not only do we reduce training time but the weights are vastly superior to what would be attained by random iterative methods. The provide an excellent starting point for iterative methods applied to all weights later in the

backpropagation algorithm. A robust training technique will also be discussed which vastly improves the fault detection capabilities of the network.

4.1. Network training (stopping) criteria

It was found early on in this study that the training error goal of the network is a very important consideration for a robust network that can generalize well. Real plant operational data contains a small amount of noise (typically 2 to 3 percent) from the sensors and other electronic equipment. If a network is trained to a low error value, it will tend to model the noise in the data and not the overall functions behind the data. Typically this is known as overfitting the training set. This creates a network that has very poor generalization ability and practically no use for its intended purpose. Two methods are used to prevent overtraining the network. The first method is to statistically determine the amount of noise in the signals and calculate an overall training error goal from the noise estimates. The second technique used is cross validation training[8] to verify the results of the first method.

4.2. Singular Valued Decomposition (SVD)

Since a linear output layer was used, training was greatly accelerated by solving for the output layer weights using a singular valued decomposition mathematical technique (SVD) instead of iterative training[8]. By using the SVD method to approximate the optimal weights, not only is training time reduced, but the weights are vastly superior to what would be attained by random iterative methods. They provide an excellent starting point for iterative methods applied to weights in the hidden layers optimized using a backpropagation algorithm.

In this method only the most relevant information is retained to compute the weights, or it's principle component vectors. The least important information is discarded because it is most likely nothing more than noise.

4.3. Robust Network Training

A robust network is one that will estimate the correct output for a respective input that contains an error or missing data, without disturbing the output estimates of the other parameters in the network. Robustness to errors is not automatically a property of AANN's. Feed forward networks generally have poor extrapolation properties. Sensor failures have no precedent in the training set, therefore the networks estimate to an error in the input is unpredictable. An error introduced into a single sensor may be detected, but at the risk of compromising the remaining signals in the network. We know that an error has occurred in the system but we may not be able to isolate which sensor is faulted.

To achieve true robustness, specifically to produce a non-corrupted output value for inputs containing errors, the network must be trained on exemplars that represent the input/output behavior. To achieve the desired behavior, the original training set is augmented by new patterns with training inputs $X' = X + \delta I_j$ $j=1, ...m$ (m is number of input signals), I_j is the jth column of the identity matrix, and original target outputs Y. For each original training example, each sensor is corrupted several times using different random values of δ ranging between ± 10 percent of the original signal value. If there are n original training samples, there will be $2n$ samples in the robust training set.

Using the corrupted input set X', a robust network will be trained to produce the non-corrupted output Y.

A network trained using a robust training set forces the network to rely on all other parameters equally for a correct output estimate during a fault condition, instead of a select few.

5. SENSITIVITY ANALYSIS

Sensitivity is defined as the change of the output over the corresponding change in the input. Given the input vector x and output vector y:

$$S_x^y = \frac{\partial y}{\partial x}$$

The sensitivity analysis of the network is shown by a cumulative bar plot which shows the change in each of the networks outputs due to a 5% perturbation in each of it's inputs. One plot show s the sum of the output change for each signal and the other shows a normalized output change which indicates the contribution of each of the perturbed inputs in the network to the corresponding output change in each signal. This is shown by horizontal bars in each of the outputs.

Figure 2 shows the results of the sensitivity analysis for the non-robust trained network. For the non-robust case a perturbation of 5% in each input of the network results in a change in the output of between approximately 10% up to 900% with the contribution of other signals varying greatly to the overall change in each output (as noted by the 22 horizontal bars in each of the 22 signals). Due to the variation in the other network parameters in the network due to a small perturbation, the network could not perform well under input fault conditions.

The sensitivity results for a network trained with a robust training set is shown in Figure 3. A substantial improvement can be seen over the non-robust network. Each of the 22 outputs in the network respond equally to a 5% perturbation with the output sum change varying from 2% to 5%. We also note that the cumulation of sensitivities of each of the channels is approximately the same. Inputs 2 and 20 vary the most at approximately 5%. These signals (hotwell temps A and B) are the least linear correlated signals in the network. Thus, the sensitivity analysis proves to be a useful tool for network parameter selection. The robust network is stable to perturbations to the inputs, thus we would expect the robust network to perform well under input fault conditions. This is shown in a later section.

6. SEQUENTIAL PROBABILITY RATIO TEST (SPRT)

The SPRT technique, which was originally developed by Wald[9], uses a statistical method to determine if a sensor has failed. It does this by calculating the residual between a sensor's estimated value and its actual output and determining if this residual statistically significant: if the mean is more probably zero or some faulted setpoint. Rather than computing a new mean and variance at every new sample, the SPRT continuously monitors the sensor's performance by processing the residuals. This SPRT based method is optimal in the sense that a minimum number of samples is required to detect a fault existing in the signal.

The residual signals, which are the difference between the sensor measurements and the estimates from the trained autoassociative neural networks, are used to generate a likelihood ratio. When a sensor is operating correctly, the residual should have a mean of zero, and a variance comparable to that of the sensor (due to the filtering characteristics of an AANN). If there is sensor drift, the residual mean shifts. Due to the shift in residual mean, the likelihood ratio increases. If the likelihood ratio increases above a certain predefined boundary (user specified by false and missed alarm probabilities), the residuals are more likely to be from the faulted distribution than from the unfaulted distribution, and an alarm is initiated.

7. RESULTS

Two networks consisting of the signals listed in Table 1 with the same architecture (22-27-18-27-22) were created using different training methods. One network was trained using the standard non-robust

training set and the other was trained using an augmented robust training set. Other than different training sets, all network parameters were the same. The two networks were compared using a sensitivity analysis, and a system simulation using a ISCV monitoring program created with SIMULINK[10] so that the response of all 22 signals could be monitored simultaneously. For the system simulation tests, a ramp error of +0.1% per day of the instruments maximum scale deflection was artificially created in sensor R234 ,reactor loop flow A.

7.1. SIMULINK ISCV monitoring program

To monitor the performance of the network the sequential probability ration test (SPRT)[11] was implemented using SIMULINK. The system block diagram is shown in Figure 4. The AANN estimate is compared to the measured signals and residuals are formed. The residuals are then sent to parallel (one for drift error detection, and one for gross error detection) SPRT blocks which output the status of each sensor (0 = good, 1 = bad) based on the variance of the residuals, a given faulted mean, a preset false alarm probability of 0.01% and a missed alarm probability of 10%. The faulted mean values were optimized using data that contained several transients. The output of the SPRT's are then processed by a logic filter with a 2 out of 4 delayed status voting scheme to eliminate false alarms due to spurious spikes in the plant data.

7.2. Network Simulation

Data for the test consisted of 4 days of full power plant operation sampled every 2 minutes, resulting in 2880 test patterns. Initially, a test was performed using plant data that contained several transients to verify that no false alarms occurred with error free plant data. A comparison of network performance was done to demonstrate the advantages of a robust training set. Finally, simulations were done to show the ISCV ability to detect drift as well as gross fault errors.

7.2.1. Robust vs. Non-robust Training

A simulation was performed using the non-robust trained network with an artificially induced drift of 0.1% of the instruments full scale deflection introduced into sensor 11 (reactor loop flow A). A residual variance of 0.01 (determined with MATLAB) with a faulted mean of 0.35 was used. The network did identify the drift in sensor 11 about 1 day into the drift (0.18% of the full scale value) but alarms in other channels at slightly greater times were also noticed (channels 4,10,12,15,17,19). This is in direct agreement with the sensitivity analysis results: a drift (perturbation) in the network input caused other parameters in the system to become vary. Although a drift was detected, the channel in which the drift occurred in is difficult to identify.

Figure 5 shows the results of the simulation using a network trained on a robust data set. We can see that the results are vastly improved. The network correctly identified the drifting sensor at approximately time interval 1250 and continued to detect the drift throughout the test with no false alarms recorded in other channels. The network correctly identified a drift at an error of 0.14% while the remaining channels in the system did not vary. The results are in direct agreement with the sensitivity analysis which showed that a robust trained network was insensitive to input perturbations. In addition to testing for faults in the above sensor (11) multiple test were performed on other sensors with similar results.

7.2.2. Drift Error Detection

A drift error is defined as a slow rate of change in a signals expected nominal value. To test the performance of the networks, both high and low drift faults of 0.1% per day of the instruments

maximum scale value was artificially created in each of the 22 network channels. Simulations were performed to see how soon the AANN ISVC system could detected the fault with a minimum of false alarms. For each simulation, the time until the fault is first detected, the percent error of the drift (with respect to the full scale deflection of the signal) at the time of detection, and the number of false alarms in all the channels was recorded for both high and low drift fault scenarios. In each test case, faults were initiated at time zero.

Figures 5, 6, and 7 are plots of typical drift test cases in sensors R234 (reactor loop flow A), R212 (reactor outlet temp A), and R225 (reactor outlet pressure B) respectively. The top plot for each case shows the actual drifting signal and the neural network estimate (note the network filtering), the middle plot shows the residual between the two, and the bottom plot shows the SPRT fault hypothesis index. In all test cases, no false alarms were recorded. Table 2. below summarize the results for the three sensors. It list the computer point tag ID, the SPRT faulted mean value, the calculated residual variance, and the percent (of maximum scale deflection) detected drift error.

Table 2. Selected drift simulation results

Computer Pt. (Tag ID)	Set SPRT faulted mean	SPRT residual variance	% Detected drift error
R234	0.35	0.01	0.14
R212	0.2	0.006	0.15
R225	7.0	4.0	2.39

As the table shows, the system performed very well. The detection time generally depended on the noise level of the signal, the more noise, the longer the longer the detection times. The average detection time for a low noise sensor (temperature for example) was approximately 0.20% in less than a day. For a high noise sensor, (pressure) it increases to an average of approximately 2.4% in about one day. A minimum of false alarms was recorded. Throughout the entire testing scenario, only three false alarms were recorded, two of them in the same parameter (R200 -pressurizer level). This was primarily due to a sustained increase in noise in the signal for a 20 minute period in the test data.

On average, all parameters performed equally well despite of the fact that some were highly correlated within a network (reactor temperature) and some had practically no correlation at all (pressureizer level). This can be understood by the robust training method used. Robust training (faulted input/normal output) forces each parameter in the network to rely on all the other parameters equally, not just a select few (or possibly one as shown in the sensitivity plots).

7.2.3. Gross Error Detection

Gross faults are defined in this study as drastic changes in the parameters value. A plant scenario would be a circuit that opens or shorts, where all loss of information is encountered. Gross faults are simulated by failing the sensor to it's maximum or minimum full scale deflection, representing gross fault "high", or gross fault "low" respectively. Maximum and minimum scale deflection is listed for each parameter in Table 1.

Depending on how "gross" the signal fails, the other parameters in the network may, or may not vary. A large drop (or rise) in a signals value may cause other parameters in the network to vary in an attempt to compensate for such a large loss (or gain) of information. Experimental results have shown that a robust network can effectively compensate for a loss of approximately 25% of any one particular signal value. A larger fault can create false alarms in other channels due to the networks reliance on a select few parameters. The residuals may change to a degree that they are greater than the pre-set faulted mean values of the SPRT's. While the other parameter residuals may vary, it is only a fraction

of the amount that the signal that contains the gross fault varies. For example, if we simulate sensor 4 as the gross failed sensor, then the residual for sensor 4 may be around 100, while the other variables are only around 10.

Since the drift detection SPRT's are optimized to detect incipient changes in the sensor, the obvious solution is a less sensitive SPRT. As can be seen in Figures 4 a "gross" detection SPRT block used for detection of gross sensor faults is used in parallel with the drift detection SPRT block. The gross SPRT's are identical to drift detection SPRT's except that they have faulted mean values that are considerably less constrained. The same variance values as the drift detection modules are used in the gross SPRT's.

The faulted mean values for the gross SPRT's were typically set to 15 times the drift fault means. Depending on the magnitude of the gross fault, some faulted means were set at higher value for optimal performance. The greater the fault, the more the other parameters in the network vary.

Like the drift test scenario, all variables were tested under gross fault conditions. Each variable was gross faulted high and low (max and min scale deflection values for each variable) with the time of detection and the number of false alarms in other channels recorded. In each case, the fault was initiated at time stamp 500, or 1000 minutes into the simulation.

Figure 8a shows an example of a signal gross fault low artificially created in sensor 9, (R222 Reactor outlet pressure A). At time sample 500, the sensor drops from it's median value of 2130 PSIG to it's minimum scale deflection of 1700 PSIG (drop of approximately 400 PSIG), with an associated change in residual mean of approximately 400 PSIG. This greatly exceeds the faulted mean value of the SPRT, thus an alarm is initiated at time sample 500 (immediately). Figure 8b shows the change of signal 16 (R224 Reactor outlet pressure B) due to the gross fault in sensor 9. A slight change in residual value can be seen when the fault was initiated (time sample 500), but not large enough to set off the SPRT due it's less stringent faulted mean setting. The residuals of the remaining 20 parameters during the gross fault condition changed to approximately ±10, less than their set faulted means. Thus the system accurately detected a gross fault in sensor 9 with no false alarms recorded in other channels.

8. CONCLUSIONS

The results of this study have shown that a plant wide sensor calibration monitoring system using autoassociative neural networks is not only feasible but very practical.

A methodology has been described to implement such a system. This includes parameter selection utilizing linear correlation coefficients and sensitivity analysis, a feature detection network architecture, and a fast reliable training method to create a robust network.

To detect the faults, a drift and gross fault detection system has been implemented using the sequential probability ratio test (SPRT). The complete ISCV system has been integrated using Matlabs SIMULINK software.

The SPRT detection module proved to be an excellent detection tool for incipient drift faults as well as gross faults. Results show that the ISCV system using autoassociative neural networks could clearly detect a fault or drift in a single channel without affecting the other channels being monitored. Thus the network not only detects the fault, but isolates the channel in which the fault has occurred.

ACKNOWLEDGMENTS

This project was sponsored by the DOE through Sandia National Laboratories under contract document AQ-6982. We would also like to thank Florida Power Corporation for supplying the data for the project.

REFERENCES

1. B.R. Upadhyaya and E. Eryurek, "Application of Neural Networks for Sensor Validation and Plant Monitoring," *NUCLEAR TECHNOLOGY*, Vol. 97, pp. 170-176, February, (1992).
2. K. Nabeshima, K. Susuki and T. Turkan, "Real-Time Nuclear Power Plant Monitoring With Hybrid Artificial Intelligence Systems," *9th POWER PLANT DYNAMICS, CONTROL & TESTING SYMPOSIUM*, Vol 2, pp. 51.01, Univ of Tennessee-Knoxville, May 24-26, (1995).
3. M.A. Kramer, "Nonlinear Principal Component Analysis Using Autoassociative Neural Networks," *AIChE Journal*, Vol. 37, No. 2, pp. 233-243, February, (1991).
4. M.A. Kramer, "Autoassociative Neural Networks," *Computers in Chemical Engineering*, Vol 16, No. 4, pp. 313-328, (1992).
5. D. Dong and T.J. McAvoy, "Sensor Data Analysis Using Autoassociative Neural Nets," *Proceedings Of World Congress On Neural Networks*, Vol 1, pp. 161-166, San Diego, June 5-9, (1994).
6. Hastie, T & Stuetzle, W. "Principle Curves," *Journal of the American Statistical Association*, 84:(406), pp. 50 - 516, (1989)
7. Howard Demuth and Mark Beale, *Neural Networks Toolbox*, The MATHWORKS Inc., Natick, Mass, (1994)
8. Timothy Masters, *Practical Neural Network Recipes in C++*, Academic Press, San Diego, (1993).
9. A. Wald, "Sequential Tests of Statistical Hypotheses," *Annals of Mathematical Statistics*, Vol. 16, pp 117-186, (1945)
10. Simulink Dynamic Systems Simulation Software Users Guide, The MATHWORKS Inc., Natick, Mass, (1992)
11. B.R. Upadhyaya, F.P. Wolvaardt and O. Glockler, "An Integrated Approach for Sensor Failure Detection in Dynamic Systems," *Research reported for the Measurement & Control Engineering Center, Report No. NE-MCEC-BRU-87-01*, March 1987.

Table 1. Selected network Parameters

Network B Signals								
Sensor #	Tag ID	Signal Description	Units	Tech Spec	Redundant	Variance	Median	Range
1	R214	Reactor Pump Suction Temp - A1 (Narrow)	DEG F			0.008	557.8	520-620
2	R215	Reactor Pump Suction Temp - A2 (Narrow)	DEG F			0.009	557.5	520-620
3	A216	Hotwell Temp - A	DEG F			0.118	112.8	32-220
4	P208	Linear Power Channel NI - 5	PERCENT		P209,P210,P21	0.021	100.5	0-125
5	R200	Pressurizer Level (L1) (Uncomp)	INCHES	YES	R201,R202	0.107	143	0-320
6	R208	Reactor Pressure - A (Wide)	PSIG	YES	R209	3.758	2139	0-2500
7	R209	Reactor Pressure - A (Wide)	PSIG	YES	R208	4.119	2141	0-2500
8	R212	Reactor Outlet Temp - A (Narrow)	DEG F	YES	R226,R227	0.005	600.5	520-620
9	R222	Reactor Outlet Pressure - A (Narrow)	PSIG	YES	R223	3.758	2129	1700-2500
10	R223	Reactor Outlet Pressure - A (Narrow)	PSIG	YES	R222	3.397	2137	1700-2500
11	R234	Reactor Loop Flow - A (CHA)	MLB/HR	YES	R236,R238,R240	0.008	74.1	0-80
12	R327	T-hot Loop - A	DEG F	YES		0.047	601.2	120-920
13	R328	T-hot Loop - B	DEG F	YES		0.043	600.3	120-920
14	R210	Reactor Pressure - B (Wide)	PSIG	YES		4.067	2162	0-2500
15	R213	Reactor Outlet Temp - B (Narrow)	DEG F	YES	R228,R229	0.014	600.9	520-620
16	R224	Reactor Outlet Pressure - B (Narrow)	PSIG	YES	R225	3.339	2151	1700-2500
17	R225	Reactor Outlet Pressure - B (Narrow)	PSIG	YES	R224	3.421	2134	1700-2500
18	R235	Reactor Loop Flow - B (CHA)	MLB/HR	YES	R237,R239,R241	0.005	72.7	0-80
19	E202	Unit 3 Electrical Generation	MW			1.962	875.7	0-1200
20	A217	Hotwell Temp - B	DEG F			0.169	116.4	32-220
21	R216	Reactor Pump Suction Temp - B1 (Narrow)	DEG F			0.008	557.8	520-620
22	R217	Reactor Pump Suction Temp - B2 (Narrow)	DEG F			0.008	558.5	520-620

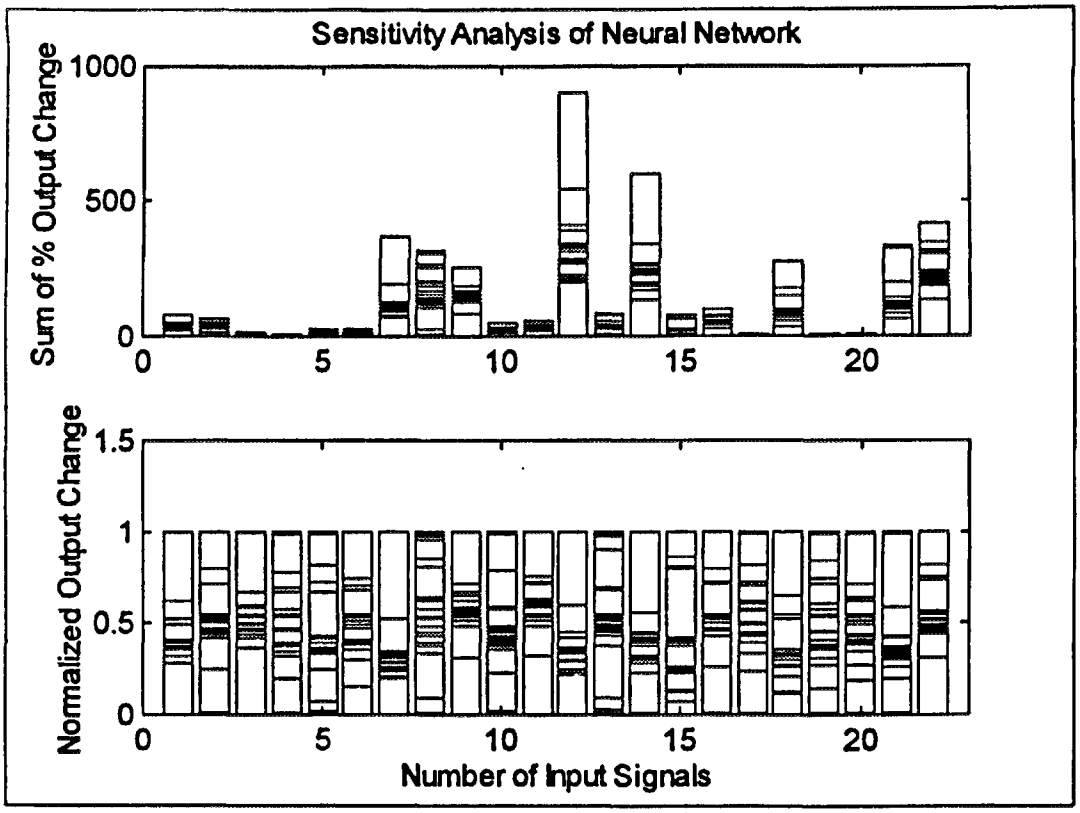

Figure 2. Sensitivity analysis of network trained with non-robust training set when a 5% perturbation is introduced into each of the inputs

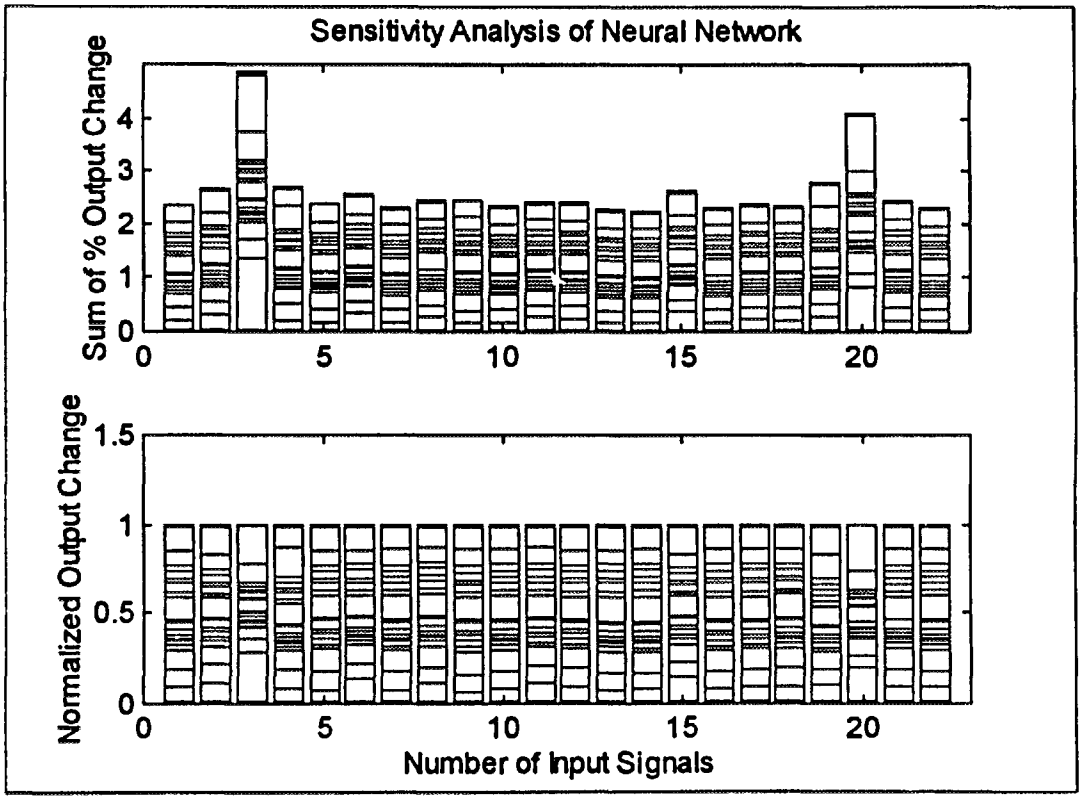

Figure 3. Sensitivity Analysis of network trained with robust training set when a 5% perturbation is introduced into each of the inputs

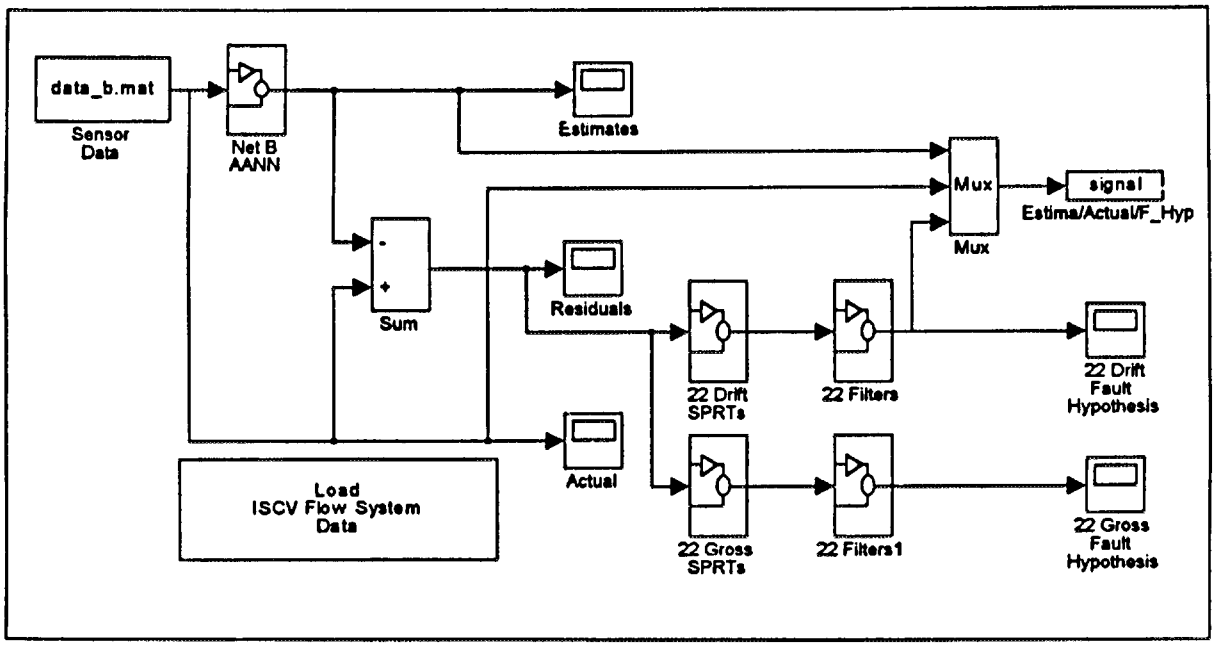

Figure 4. SPRT ISCV monitoring system

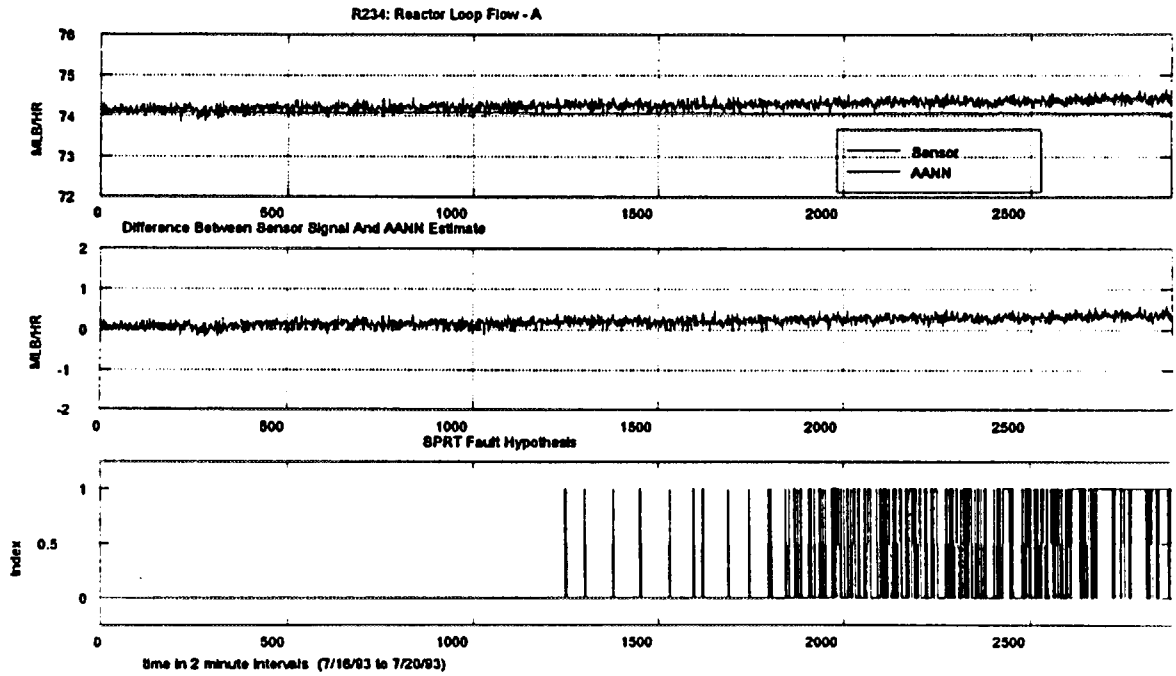

Figure 5. ISCV system detecting a fault drift high in the reactor loop flow when a drift of 0.1% per day of the instruments maximum scale deflection is introduced into the sensor signal

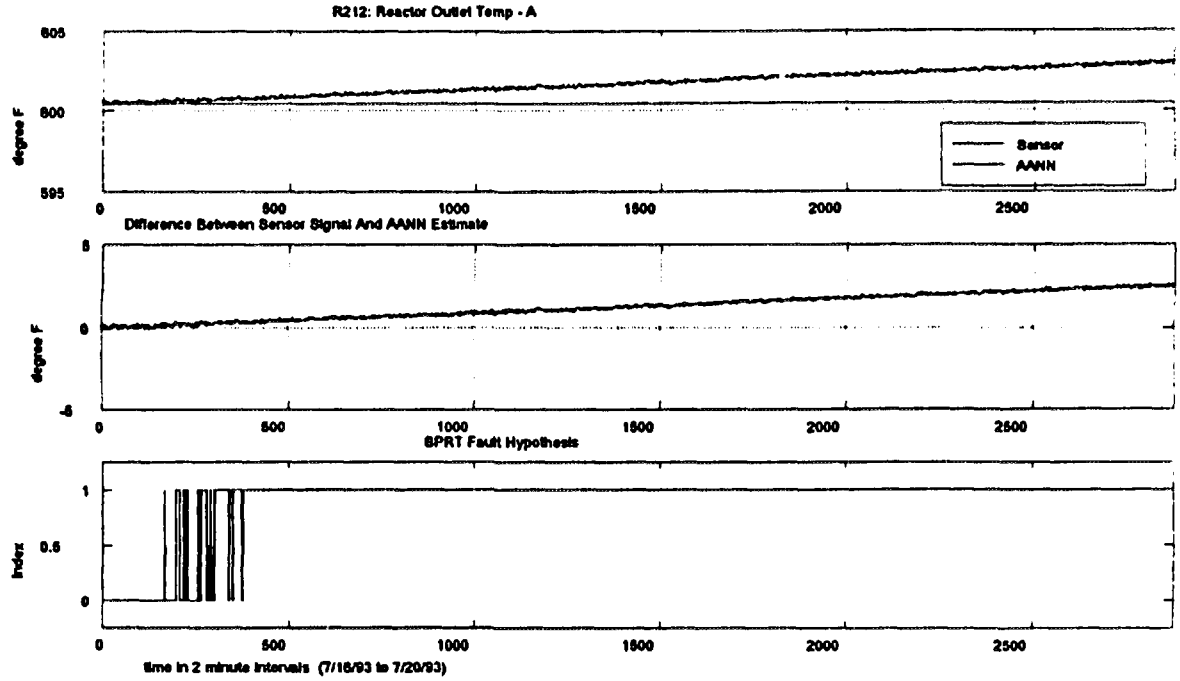

Figure 6. ISCV system detecting a fault drift high in the reactor outlet temperature when a drift of 0.1% per day of the instruments maximum scale deflection is introduced into the sensor signal

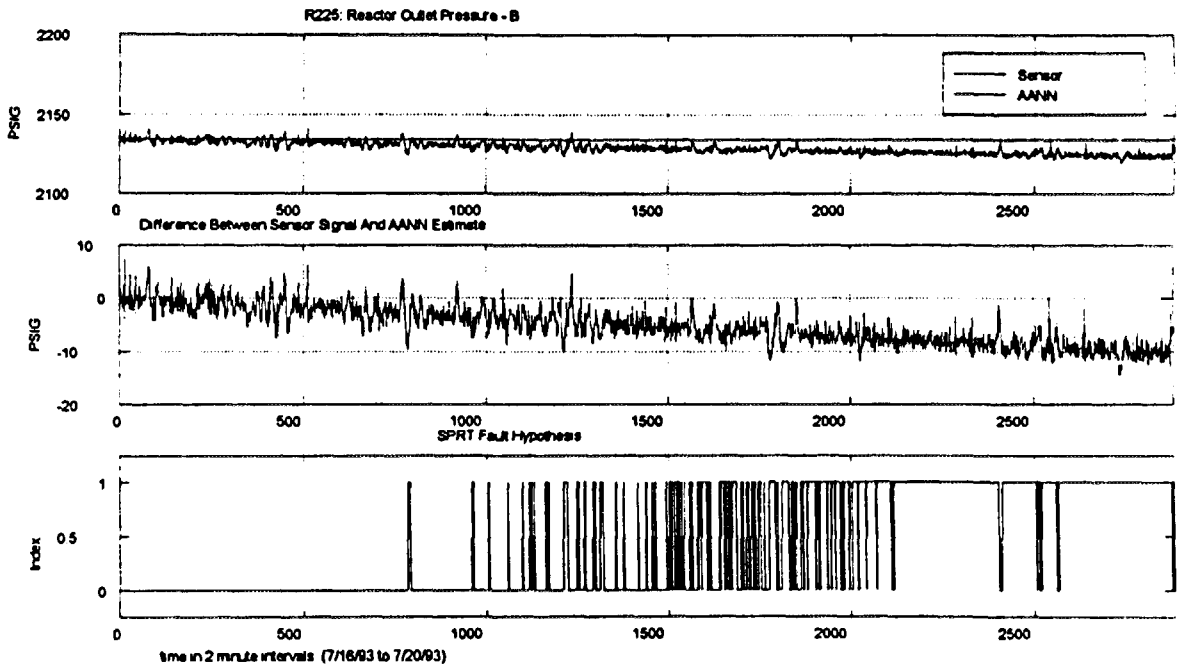

Figure 7. ISCV system detecting a fault drift low in the main steam pressure when a drift of 0.1% per day of the instruments maximum scale deflection is introduced into the sensor signal

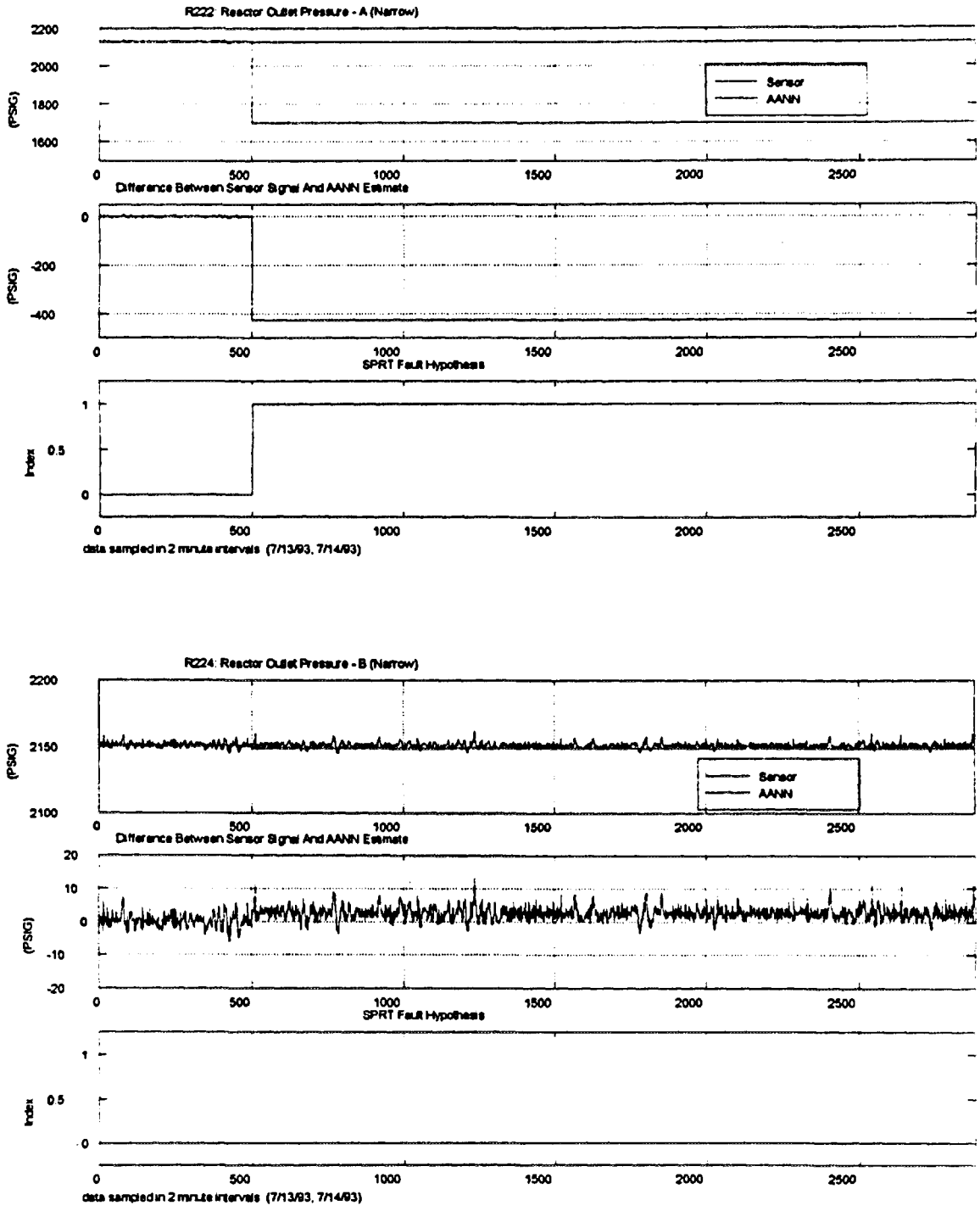

Figure 8. Response of "gross" fault initiated at time sample 500 a.) variable gross fault was initiated in b.) typical response of companion network variable