Proceedings of the 2001 IEEE Aerospace Conference, Big Sky Montana, March 2001.

Anomaly Detector Fusion Processing for Advanced Military Aircraft[1]

Tom Brotherton
The Intelligent Automation Corporation
San Diego, CA 92131
(858) 679-4140
Tom.Brotherton@iac-online.com

Ryan Mackey
The Jet Propulsion Laboratory
Pasadena, CA 91109
(818) 354-9659
Ryan.M.Mackey@jpl.nasa.gov

Abstract— Automated *Prognostics and Health Management* (PHM) is a requirement for advanced military and commercial aircraft. PHM is the key to achieving true *condition-based maintenance*. PHM processing strategies include modules for the processing of known nominal and fault conditions. However in real operations there will also occur faults and other off-nominal operations that were never anticipated nor ever encountered before. We call these events *anomalies*. Missing the presence of an anomaly could potentially be catastrophic with the loss of the pilot and aircraft. Several different anomaly detectors (ADs) have been developed for advanced military aircraft to solve this problem. Fusion of these ADs can significantly reduce false alarms while at the same time substantially improving detection performance. Fusion is a way of approaching the goal of perfect detection with zero false alarms. We have developed a neural net approach for performing AD fusion. Presented here is a description of that technique and the application to military aircraft subsystem data.

1. Table of Contents

2. 1. Introduction

Automated *Prognostics and Health Management* (PHM) is a requirement for the advanced and commercial aircraft. PHM is the key to achieving true condition-based maintenance. A sophisticated PHM system will potentially save money, aircraft downtime, and lives by providing for the right parts to be in the right place at the right time.

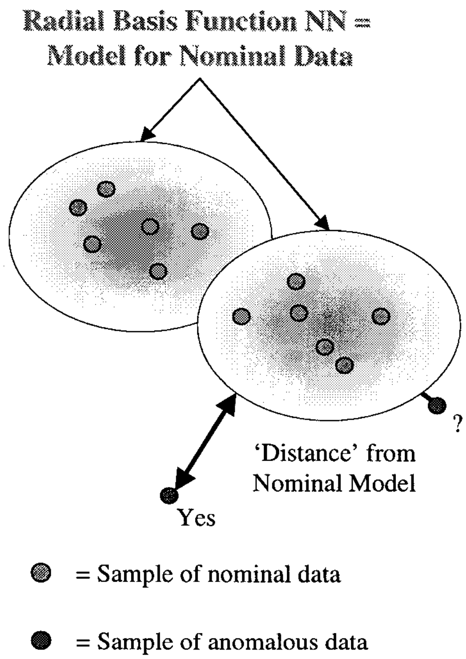
Table 1. Table of acronyms

ACRONYM	MEANING
AD	Anomaly detector
APU	Auxiliary power unit
BEAM	Generalized Cross-Signal Anomaly Detector
BU	Basis unit
CD-RBF	Class dependent – Radial basis function
EGT	Exhaust gas temperature
FF	Fuzzy factor
LMS	Least mean square
MLP	Multi-layer perceptron
NN	Neural net
NNAD	Neural net anomaly detector
PHM	Prognostics and health management
RBF	Radial basis function

Advanced aircraft PHM processing strategies include modules for the detection, diagnosis and prognosis of known fault conditions. However, in real operations, faults and other off-nominal conditions that were never anticipated nor encountered will also occur. We call these events *anomalies*. Treatment of anomalies is particularly important with new aircraft, but it is also a concern for legacy aircraft. Failure to resolve an anomaly could be catastrophic with the potential loss of the pilot and aircraft. An important part of the overall system is the inclusion of *anomaly detection*. The role of the *Anomaly Detector* (AD) is to flag these unanticipated and never before seen events.

In recent work for a new advanced military aircraft three different AD methods have been developed to be included on the aircraft. They are a neural net anomaly detector (NNAD) [1,2] a generalized cross-signal anomaly detector approach [3] (a component of BEAM) and a Hidden Markov Model (HMM) approach [4]. Considered here is the fusion of NNAD and the generalized cross-signal anomaly detector to derive a single AD output.

Figure 1 Simplified example of the NNAD



Fusion technology can combine the results from different processing approaches (such as time-correlation statistics, neural networks, hidden Markov models, and physical models) resulting in superior results. Fusion of multiple approaches has been demonstrated to significantly reduce false alarms while at the same time substantially improving detection and classification performance [5,6]. Each group's AD focuses on different aspects of real data signals when performing a detection. Sometimes the detectors are 'complimentary' and support each other's detections.

In this case, fusion improves confidence of the detections and thus not only improves detection performance but reduces false alarms as well. However, sometimes a particular detector focuses on an aspect of the signals not considered by the other detectors. In this case it provides the only anomaly detection. This expands the class of signals that the fused AD is able to process. Fusion is a way of approaching the goal of perfect detection with zero false alarms.

Table 2 shows a summary of the expected response for the different detectors being developed for advanced military aircraft. The types of anomalies that can be expected are listed on the left. The columns indicate the expected response for each of the detectors; an 'X' indicating that the detector is expected to work well. A '?' indicates the response is not clearly known and depends on the nuances of the data. The goal is to have at least one 'X' in each row. This ensures that no class of anomaly will be missed. Two or more X's ensure increased probability of detection while significantly reducing false alarms.

Table 2. Summary of expected AD detector response

Failure Type	*NNAD*	*BEAM*	*HMM*
Linear transform (gain)	X	?	X
Transient	?	X	X
New 'mode'	X	X	X
Feedback	?	X	?
Sensor failure (in range)	X	X	?
Sensor failure (noise)	?	X	?
Uncorrelated signals	X	?	?
Other	?	?	?

For advanced PHM we have developed a neural net approach for fusion of input ADs. The neural net fuses the individual AD detection outputs as well as "features" that are generated internally by each of the detectors. The result is a single detector output, which has simultaneously improved detection statistics, significantly reduced false alarm rates, and covers a wider range of anomalies then seen by a single detector. In addition to the detector output, the processing gives a measurement of the "difference from nominal" for each of the signals that are input to the individual ADs. These difference measures are then passed down-stream to reasoners that isolate the exact nature of the anomaly.

Presented here are details of the technique and results of AD fusion applied to advanced military aircraft subsystem data. A summary of the individual ADs areas of coverage, synergy, strengths and weaknesses are discussed. Section 3 and 4 give brief summaries of the two anomaly detectors used as inputs to the fusion neural net. Section 3 describes the *Neural Network Anomaly Detector* (NNAD). The NNAD is also used for performing the fusion processing. Section 4 describes the *Generalized Cross-Signal Anomaly Detector* (BEAM). Section 5 discusses fusion processing using the neural net approach. Results from application of the fusion processing to aircraft subsystem data are presented in section 6. Section 7 contains a summary and conclusions.

3. NEURAL NETWORK ANOMALY DETECTION

Presented here is a high level description of the neural net anomaly detector (NNAD). Details of NNAD can be found in a related paper [1]. The NNAD uses radial basis function (RBF) neural nets (NN) to form a statistical model of "nominal" data. As new data enters into the system, it is compared to the RBF NN model. If data falls within the boundaries defined by that model, then it is flagged as "nominal". If is does not, then it is flagged as an "anomaly". Figure 1 shows a simplified example of that processing.

In Figure 1, the input signal data is 2 dimensional. The RBF NN model of the nominal data has two basis functions that are represented by the two ellipses in the figure. Here the basis functions are Gaussian in shape to give a continuous degree of membership measure from each of the basis functions centers. The two ellipses represent constant degree of membership contours that may be used as a detection threshold.

In the figure 1, the small green and red circles represent test samples from nominal and anomaly data respectively. The green circles all fall inside of the detection threshold so they are classified as 'nominal'. The red circles fall outside of the detection threshold and they are declared as anomalies. One of the red circles is clearly far away from the detection threshold ellipse and thus clearly an anomaly. The other red circle is much closer. Is it indeed an anomaly or is it a false alarm?

This describes the basic approach to anomaly detection using the neural net. Of course with the real data we are dealing with many more features (6-50 for the real aircraft applications) and the number of basis functions, particularly for transient data, is much larger (typically 80+) and the basis functions need not be Gaussian in shape, so that the processing becomes more complicated.

Neural Networks

At the heart of the processing are the neural networks used to form the model of nominal signal data. The particular neural net we used is the Radial Basis Function (RBF) neural network (NN) [7,8]. The RBF NN is essentially a nearest neighbor type of classifier. Thus it has several properties that make it ideal for performing anomaly detection. These are not found with multi-layer perceptron neural networks.

The architecture for the standard RBF NN is shown in Figure 2. There are two steps involved with "training" the RBF neural network. The first step is clustering of the input data used to form the hidden-layer *basis unit* functions (BUs). All of the input training vectors for all classes are lumped together at this stage. The data is then clustered into one of several candidate BUs using a clustering algorithm such as the *linear vector quantization* (LVQ) algorithm. There are a variety of techniques to perform to perform this clustering that are included in our program. We have found for anomaly detection the k-means algorithm [9] gives good results in reasonable time.

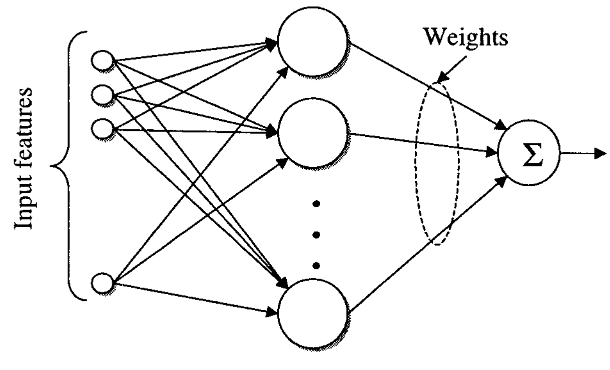


Figure 2 Standard RBF Neural Net Architecture

For NNAD described here these basis functions take the form of multidimensional Gaussian distribution functions. The mean and variance of each dimension is estimated from the data. Following clustering is a *least mean-square* (LMS) weighting of the BU outputs to form the desired function approximation for classification.

During clustering we force the basis units in the RBF NN to be associated with only a single class of data. For the NNAD only nominal data is used so that each of the BUs is used to represent some portion of the overall feature / feature trajectory space. For transient data the number of BUs can be quite large. However for general classification this will include sets of basis units associated with different known fault categories.

The output from the RBF NN can be determined in two ways. The first is simply the final output of the neural net as described above. The second is to select the basis unit that has the maximum activation. The BU with the highest activation will be the BU that's "nearest" to the set of input signals. This is possible because all of the BUs are associated with only the nominal data class. For NNAD we use both methods for getting the neural net output. The LMS output is used for the overall detection and the nearest basis unit is used for the individual signal detections. In effect the RBF NN neural net is a nearest-neighbor classifier with the BUs defining prototype models for different segments of the signal data. As other classes are added, additional BUs are added. These too will be associated with just a single class. We call this architecture a class dependent – radial basis function (CD-RBF) neural net [1].

Compute Signal Distances

When a detection is made, the "off nominal" distance of all the input signals is computed. Figure 3 shows an example of how this processing is done for the two-signal case. In figure 3 the red dot represents the test sample under consideration. Note that no single signal needs to be significantly off nominal for a detection to be made. Rather it is the aggregate signal set that gives rise to the detection.

All of the neural net models developed for the subsystem data have between 6 and 100+ basis units. The first step in the processing is to determine which of those basis units is the "closest" to the sample point being tested. The distance computed is the Mahalanobis distance to the each of the clusters. The Mahalanobis distance is used as it accounts for not only the centers of each of the basis units, but also the spread. In Figure 3, the dark blue arrow represents the basis unit that is closest to the sample point. It is the BU that gives the largest output. This BU is selected for the next step in the processing. The basis unit is the most like the set of input signal in a nearest neighbor sense, and thus gives rise to the minimum off nominal distances. Selecting the closest basis unit for <u>each</u> signal individually is <u>not</u> correct. The detection and distance are a function of the set of signals.

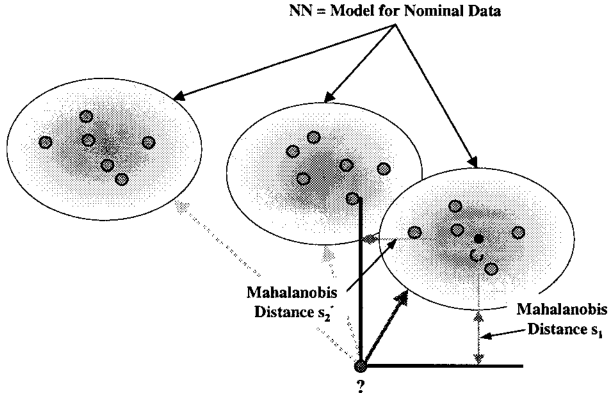


Figure 3 Off-nominal distance calculation

The distance is then computed for each of the individual signals as the Mahalanobis distance from the center of the basis function that was selected. In figure 3 the yellow arrows in the figure indicate the distances for the two input signals to the center of the nearest BU. The red arrows represent the Mahalanobis distance that is reported.

4. Generalized Cross-Signal Anomaly Detection

This section briefly follows the mathematical outline presented in [3], which describes a general method of anomaly detection from time-correlated sensor data. The method is applicable to a broad class of problems and is designed to respond to any departure from normal operation, including faults or events that lie outside the training envelope.

The SIE, or System Invariance Estimator, is a statistical process for examining multi-signal data that was developed as part of the BEAM approach developed at JPL [10]. As input, it receives multiple time-correlated signals as well as a fixed invariant library constructed during the training process (which is itself data-driven using the same time-correlated signals). It returns the following quantities:

- Mode-specific coherence matrix
- Event detection
- Comparative anomaly detection
- Anomaly isolation to specific signals
- Distance measure of off-nominal behavior

As a first step of analysis, this computation makes a decision whether or not a fault is present, and reduces the search space of data to one or a few signals. Time markers are included to indicate the onset of faulted data. These conclusions, which can be drawn for nearly any system, are then passed to other analysis components for further feature extraction, correlation to discrete data events, and interpretation.

To motivate a cross-signal approach, consider that any continuously valued signal, provided it is deterministic, can be expressed as a time-varying function of itself, other signals, the environment, and noise. The process of identifying faults in a particular signal is identical to that of analyzing this function. Where the relationship is constant, i.e. follows previous assumptions, we can conclude that no physical change has taken place and the signal is nominal.

However, the function is likely to be extremely complex and nonlinear. Environmental variables may be unmeasurable or unidentified. Lastly, the interaction between signals may be largely unknown. For this reason it is more efficient to study invariant features of the signals rather than the entire problem.

Because we do have the different signal measurements available, we can consider signal relationships separately and effectively decouple the problem. A good candidate feature is signal cross-correlation. By studying this or a similar feature rather than the raw signals, we have reduced our dependence on external factors and have simplified the scope of the problem.

In the case of the SIE we will use a slightly different feature across pairs of signals. We refer to this feature as the coherence coefficient:

$$\zeta_{ij} = \frac{\left|Cov(S_i, S_j)\right|}{Max\left(Var(S_i), Var(S_j)\right)} \qquad (1)$$

It is chosen instead of the ordinary coefficient of linear correlation in order to take advantage of certain "nice" mathematical properties. This coefficient, when calculated for all possible pairs of N signals, describes an NxN matrix of values. The matrix is referred to as the Coherence Matrix of the system.

The coherence matrix, when computed from live streaming data, is an evolving object in time with repeatable convergence rates. Study of these rates allows us to segment the incoming data according to mode switches, and to match the matrix against pre-computed nominal data.

For the purpose of this discussion, a "Mode" refers to a specific use or operation of the system in which the coherence coefficients are steady. In other words, the underlying physical relationships between parameters may change, but should remain constant within a single mode. These modes are determined from training data for the purpose of detector optimization. Ordinarily they do correspond to the more familiar "modes," which represent specific commands to or configurations of the system, but they need not be identical. Frequently such commands will not appreciably alter the physics of the system, and no special accounting is needed.

Comparison of the runtime coherence matrix to a pre-computed, static library of coherence plots, taking into account the convergence behavior of the computation, is an effective means of anomaly detection and isolation to one or more signals. The complete process is described architecturally in Figure 4 below.

Unfortunately, this comparison is only meaningful if we can guarantee our present coherence values do not reflect mixed-mode data, and so some method of segmentation must be found. For purposes of anomaly detection, mode boundaries can be detected by monitoring the self-consistency of the coherence coefficients. As each new sample of data is included into the computation, a matrix average for the resulting change is extracted and compared against the expected convergence rate. A change in the convergence rate implies a new mode has been entered and the computation must be restarted.

Between detected mode transitions, the difference between the computed and expected coherence allows us to optimally distinguish between nominal and anomalous conditions. Violation of this convergence relationship indicates a shift in the underlying properties of the data, which signifies the presence of an anomaly in the general sense. The convergence rate of this relationship, used for fault detection, is considerably slower than that for data segmentation, though still fast enough to be practical.

Once a fault has been indicated, the next step is to isolate the signals contributing to that fault. This is done using the difference matrix, which is formed from the residuals following coherence comparison against the library.

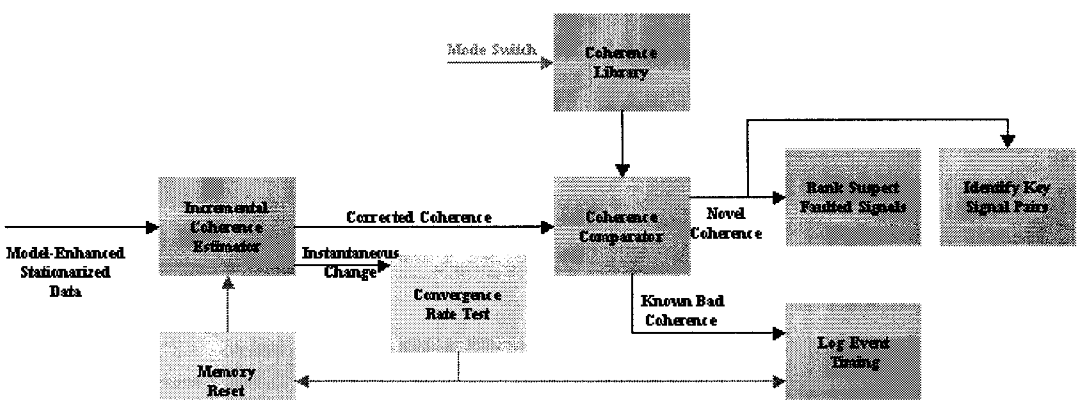
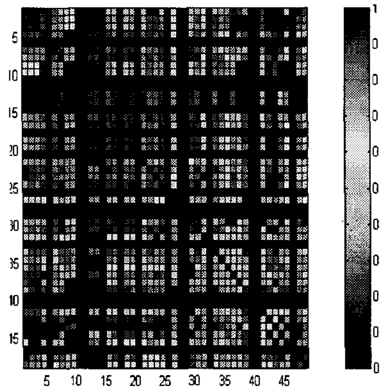


Figure 4: Coherence-Based Detector Architecture

An example illustration is given in Figures 5 through 8 below. In each of these figures, the signal number appears on both X and Y axes, displaying the coherence coefficient for all signal pairs. The coherence values vary between 0 (causally disconnected) to 1 (causally dependent). The difference matrix has values between –1 and 1, with positive values implying the current data shows a loss in coherence compared to the training data.

Figure 5 Snapshot of Evolving Coherence



Given an anomaly affecting one signal, we expect to see the correlation between it and all other signals diminish compared to the expected values. There may be stronger differences with certain pairs than others, but in general all pairs including that signal will decrease. Visually this leads to a characteristic "cross-hair" appearance in the difference matrix. Additional noise, nonlinear behavior, reduced response, sensor drift, and similar phenomena that affect only a single signal will appear in this fashion.

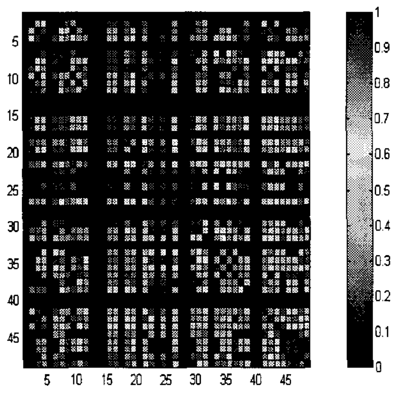
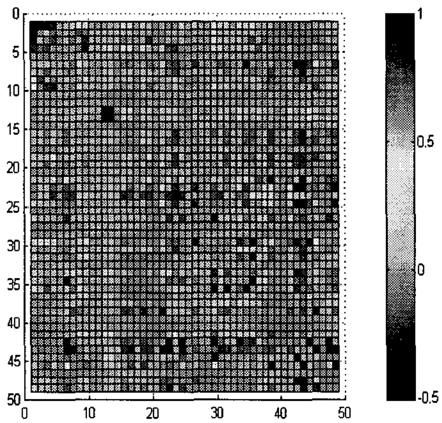


Figure 6 Sample Library Coherence

In general, an anomaly will manifest as a decrease in coherence between signal pairs. However, there are rare cases where coherence will increase. Typically, this is not system-wide but is isolated to a few specific pairs. Such an increase indicates a new feedback relationship occurring in the system, and merits special attention.

The method presented here is applicable to virtually any system producing time-correlated sensor data. Training is conducted using nominal data, or if desired matches can be tested against fault data, should any be available. The detector increases in accuracy as the number of sensors increases; however, computational cost and mode complexity eventually place a practical limit on the size of the system to be treated. This method has been successfully applied to systems as small as four sensors and as complex as 1,600.

Figure 7 Difference Matrix



Another key virtue of this approach is its resilience in the face of novelty. The coherence between signals is a very repeatable property in general, especially as compared to environmental variable or nonlinear terms in the signals themselves. This repeatability allows us to quickly determine whether or not the coherence is consistent with any of the training data, and therefore can be used as an efficient novelty detector, regardless of its cause.

5. Fusion Processing

Fusion processing was performed using the neural net anomaly detector (NNAD) described in section 2 above. Figure 8 shows a high level flow diagram of the processing. The major difference is that instead of using measured sensor signals as input to the detector, the outputs from the first stage detectors are used.

Outputs from NNAD include the detection flag (binary 0 or 1), the raw neural net output, and the off-nominal signal distances for each of the input signals to the first stage of processing. The detection flag equals 1 when nominal data is detected as input to the first stage NNAD. It is 0 when off nominal (i.e. an anomaly) is input. The raw neural net

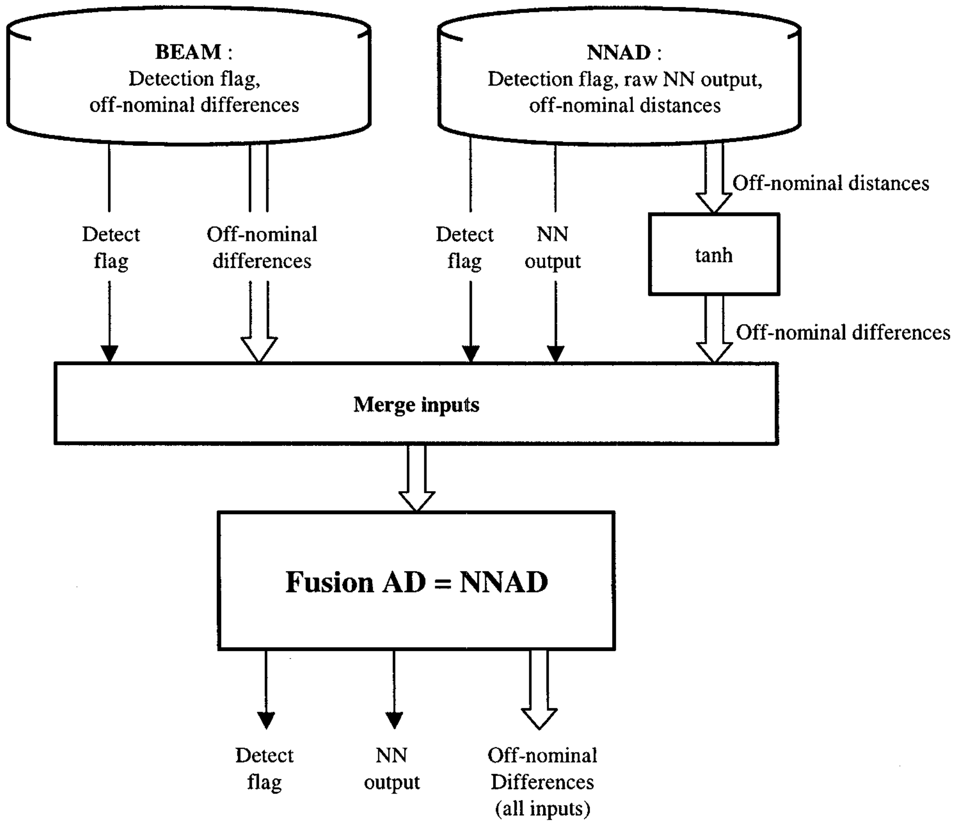


Figure 8 Fusion processing flow diagram

output is a continuous variable that goes approximately between 0 and 1. In some sense it is a measure of how far off nominal the input set of signals is. The set of off-nominal distance measures gives the off-nominal distance of each of the features individually.

Outputs from BEAM include the detection flag (binary 0 or 1) and the off-nominal signal differences. The off-nominal differences are essentially a coherence measure and are naturally restricted to be between 0 and 1.

The off-nominal distance outputs from NNAD can be between 0 and infinity. Those distances are normalized by a hyperbolic tangent function prior to input to the fusion net. This is done to restrict the value to be between 0 and 1 and to approximate the coherence measure that is output by the BEAM processing.

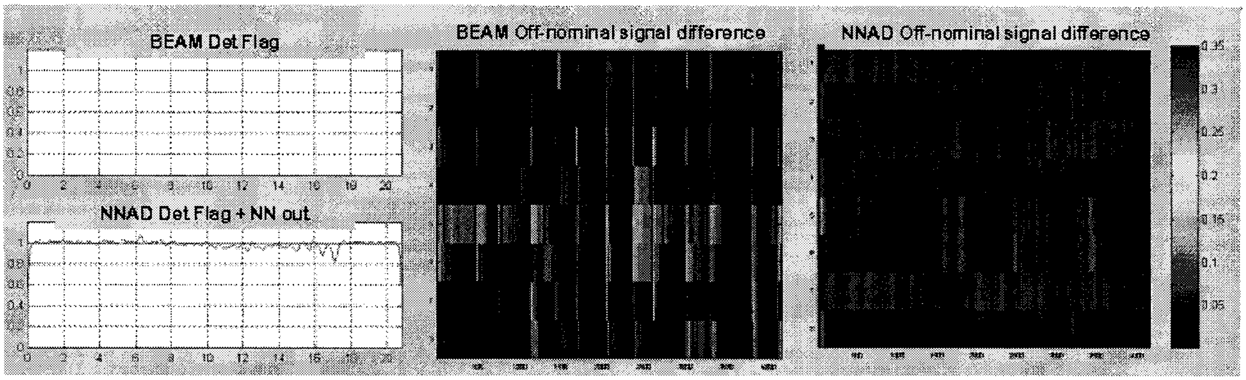
All the first level anomaly detector outputs are merged together to form a vector input to the fusion neural net. Using these inputs processing by the fusion NN is exactly as described for the standalone NNAD.

Generally the first level ADs can be run at a higher false alarm rate then when they are run individually. This is because we can rely on the fusion processing to remove the false alarms. Running the first level ADs at a higher false alarm rate also improves detection performance by including detections that may not be seen when a smaller false alarm rate is required. When training the NNAD we typically allow for a 2% false alarm rate.

6. Application to Advanced Military Aircraft Subsystems

Considered here is processing of two data sets that are related to advanced military aircraft subsystems. One is collected from the hydraulic system for the flight control surfaces. The second is for the auxiliary power unit (APU). Processing of each of the data sets with the standalone NNAD processing is discussed in a companion paper [2]. Processing of the hydraulic data by the generalized Cross-Signal Anomaly Detector is described in [3].

Hydraulic Data



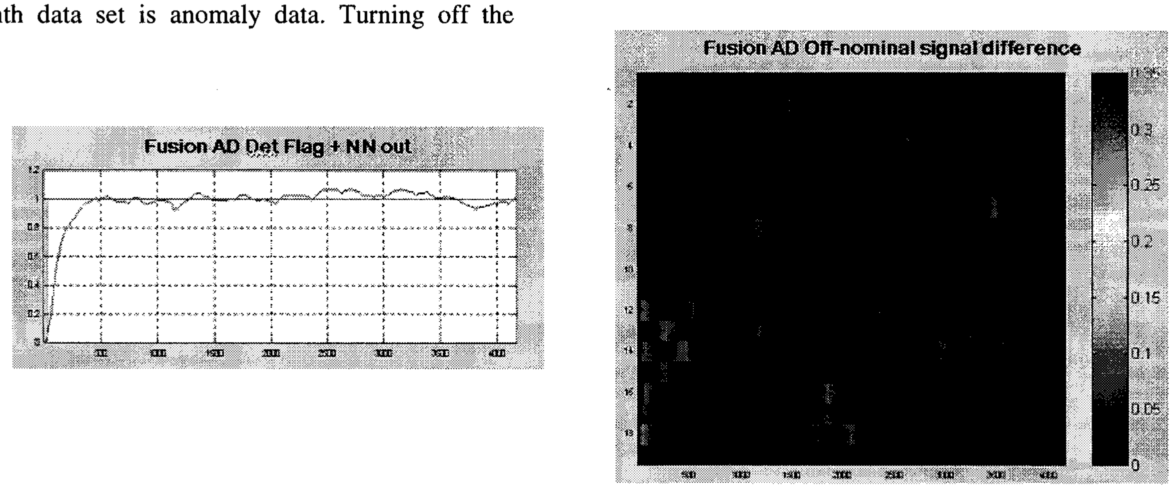


Figure 9 Nominal hydraulic data fusion AD inputs

Figure 10 Nominal hydraulic data fusion AD outputs

The hydraulic data used here consisted of seven different data sets. Six of the data sets represented 'nominal' data. The seventh data set is anomaly data. Turning off the accumulator in the hydraulic system created the 'anomaly' in the hydraulic system. The data sets represent different levels of stick movement by the pilot. The movement varies from 'no movement' to 'severe'. There were 8 channels in the data that correspond to different pressure measurements within the system.

Turning off the accumulator changes time constants in the response of the system to pilot stick movements. This can be seen visually as a change in the second order of the statistics of the data. The nominal and anomaly data sets are similar, varying possibly in their second order statistics.

Five of the nominal data sets were used for training the neural net anomaly detector. One of the nominal and the anomaly data were used for testing the system

Figure 9 shows the inputs to the fusion anomaly detector. The inputs are from nominal (normal) data. There are four separate pictures in the figure. They correspond to:

- The detection flag input from the generalized cross-signal anomaly detection (BEAM). That signal is binary. A '1' implies that the signal is nominal. A '0' that it is an anomaly. Here the flag indicates that the data is nominal for the duration of the signal.

- The detection flag and raw neural net outputs from the neural net anomaly detector (NNAD). The detection flag here is also a binary signal similar to that of the BEAM output. The raw neural net output is a continuous variable the goes between) and around 1 (it can be larger then 1).

- BEAM off-nominal signal differences for all of the signals input to the first level AD. The values go between 0 and 1.

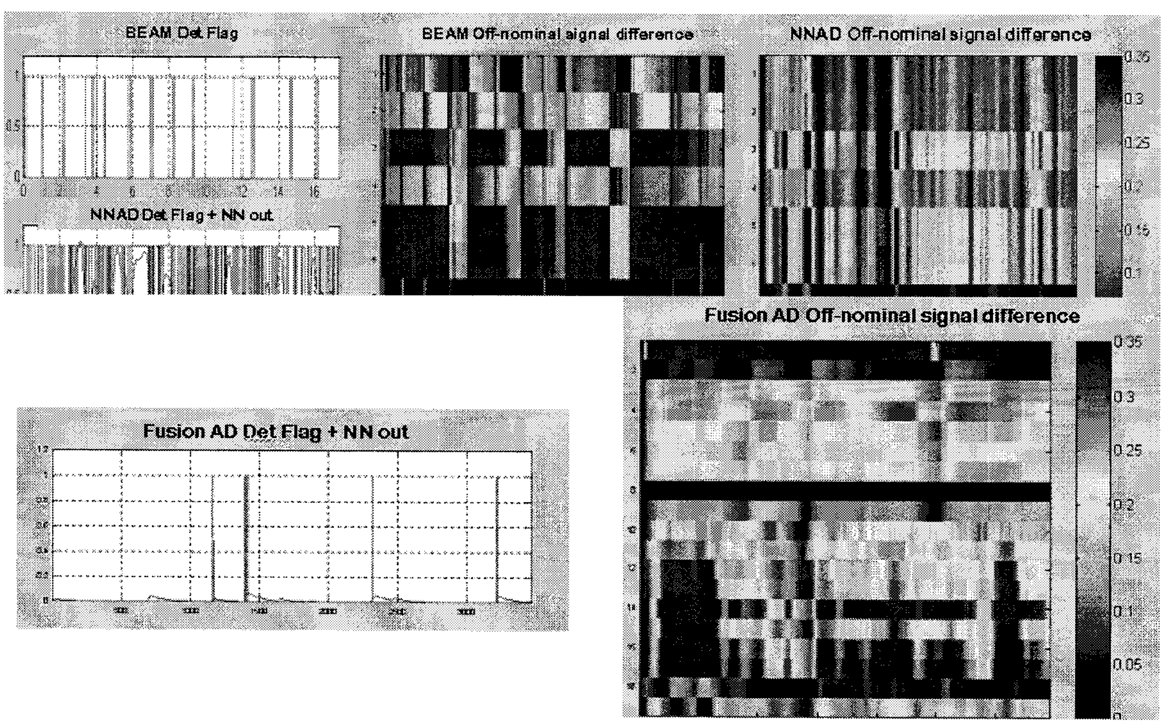
- NNAD off-nominal signal differences for all the signals input to the first level NNAD detector. Recall that these values have been normalized to be between 0 and 1 as well. For both the BEAM and NNAD off-nominal signal differences the color coding indicates how far off-nominal the signals are. The colors are saturated at 0.35 in the plots. 0.35 corresponds to 2 sigma for the NNAD off-nominal individual signal distances used for the stand alone NNAD detector [2]. The saturation threshold is only used for visualization on the input plots shown in Figure 5. The full range values are input to the fusion AD.

As seen in Figure 9, all the outputs from the first level ADs are as expected for nominal data input to the system.

Figure 10 shows the outputs from the fusion AD for the inputs shown in Figure 5. The fusion AD detector flag and raw neural net output are shown in the left plot of Figure 10. As with the inputs the detection flag is binary; 1 for nominal data and 0 for anomaly data. The raw neural net output is a value between 0 and around 1. The second portion of the figure shows the off nominal signal differences for the all of the input signals input to the fusion AD. As such channels 1 to 9 correspond to the BEAM outputs. Channel 1 is the BEAM detection flag. Channels 2 to 9 are for the off-nominal signal differences.

Channels 10 to 19 correspond to the first level NNAD outputs. Channel 10 is the binary detection flag. Channel 11 is the raw neural net output. Channels 12 through 19 are for the individual signal outputs. Initially we were considering merging the individual inputs to give a single off-nominal distance measure for each of the original input signals. However there is information to be gained by keeping them separate due to the different properties of the individual first level ADs.

Figure 11 shows the outputs from the first level ADs used as inputs to the fusion AD when anomaly data (the accumulators are turned off) are input to the fusion AD.



As seen in the figure, both the BEAM and NNAD detectors now flag that an anomaly is present. Notice that both detectors drop in and out of detection. This is as

expected as the anomaly is a function of the stick movement by the pilot; the anomaly is transient in nature and occurs following a stick movement. Also notice that the input signals that have the largest off-nominal distances correspond to each other.

Figure 12 shows the output from the fusion AD. As seen the detector output is now almost continuously indicating that anomaly is present. In addition the indicted signals are consistent as well. The fusion of the two detectors has reduced the dropping in and out of detection substantially.

Auxiliary Power Unit Data

The second data set processed was from an auxiliary power unit (APU). That data contains several cuts of nominal data as well as data that contains real anomalies. Three nominal data sets were used for training. Testing was performed on an independent anomaly data set. One of the training data sets was used for the nominal results presented below.

There were a variety of signals measured from the APU to use for anomaly detection. 6 signal channels were selected for input to the system. They are shown in Table 3. Figure 13 is an example of one of the data sets. As seen in the plot the data is highly non-stationary.

Table 3 APU input signals.

CHANNEL	DESCRIPTION
1	Shaft speed
2	Fuel flow
3	Oil temperature
4	Inlet temperature
5	Exhaust gas temperature (EGT)
6	Compressor Discharge Pressure

Figure 14 shows the outputs from the first level ADs that form the input to the fusion AD when nominal data is present at the input of the system. BEAM has 'perfect' detection results. NNAD has a couple of drop outs. Also notice that the off-nominal signal differences are all with in limits. This is as expected. However also notice that with the exception of start-up transients, all the inputs to

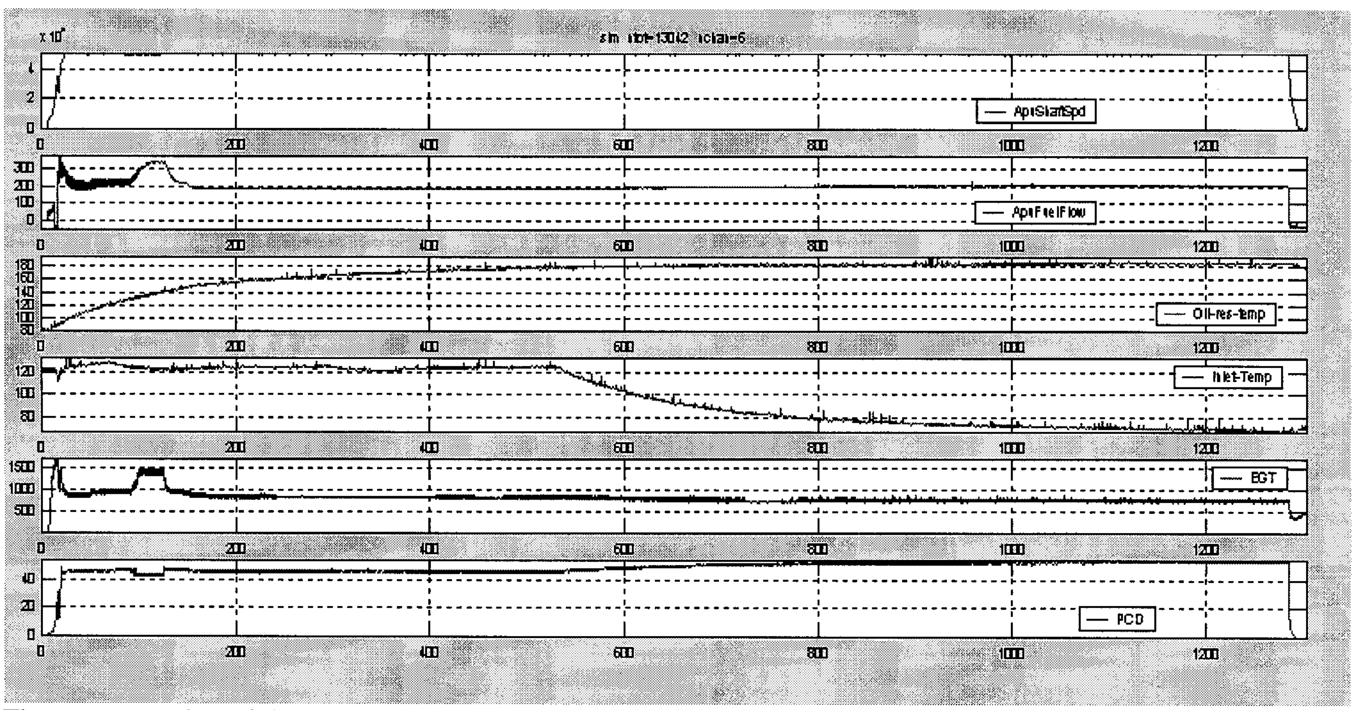


Figure 13 Example of APU data

the fusion neural net appear stationary. This is as expected

Figure 14 Nominal APU fusion AD inputs

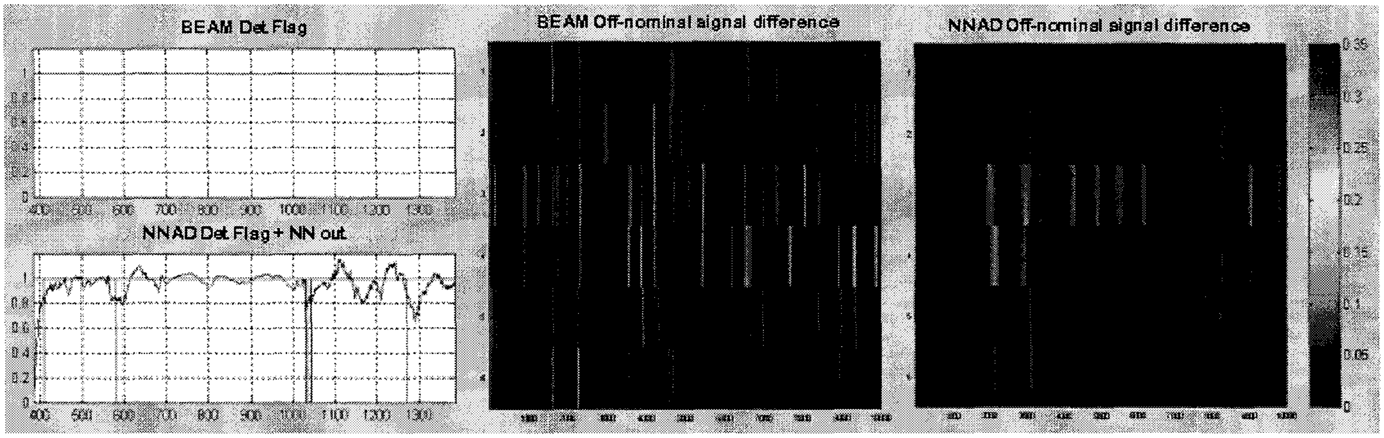
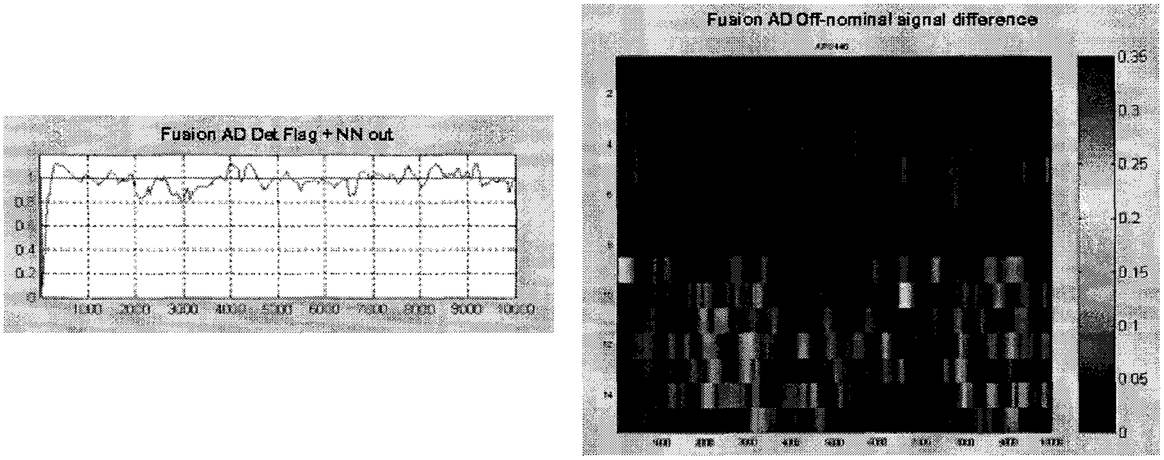


Figure 15 Nominal APU fusion AD outputs



as the first level ADs, when nominal data is input, have

ideal performance of a constant detector output of 1 and 0 off-nominal distance measure. This is true even when the data is highly transient as seen in figure 13. The architecture and processing required for the fusion NNAD is substantially less then that required for the standalone NNAD [2].

Figure 15 shows that output from the fusion processing for nominal data input. As expected no anomalies are detected and the off nominal

Figure 16 shows the inputs for processing of anomaly APU data. The anomaly with this data was a faulty EGT sensor. It gave in-range but anomalous readings.

The outputs from the first level ADs are interesting. Notice that BEAM does <u>not</u> detect the anomaly. This is because the sensor reading, for this data, is uncorrelated with the other signals being monitored; it does not fit the assumptions of the signal properties that BEAM is based on. However, the NNAD has a strong detection. Both the raw neural net output and the detection flag are pegged at '0'. The off-nominal signal differences have the same properties as the detections. In the NNAD off-nominal signal differences the EGT channel has the biggest difference (the thresholding for visualization in Figure 16 can not show this).

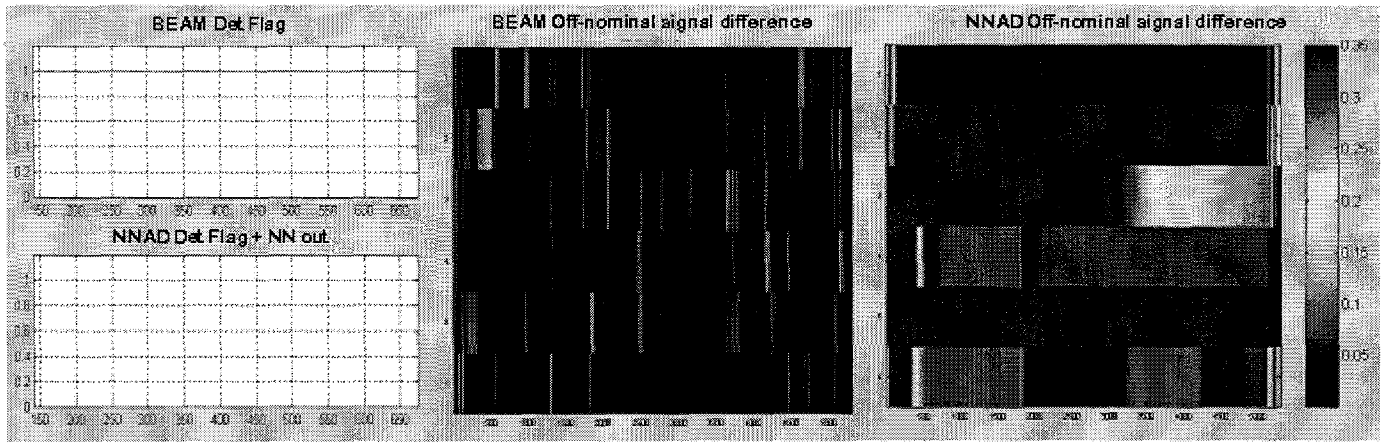


Figure 16 Anomaly APU fusion AD inputs

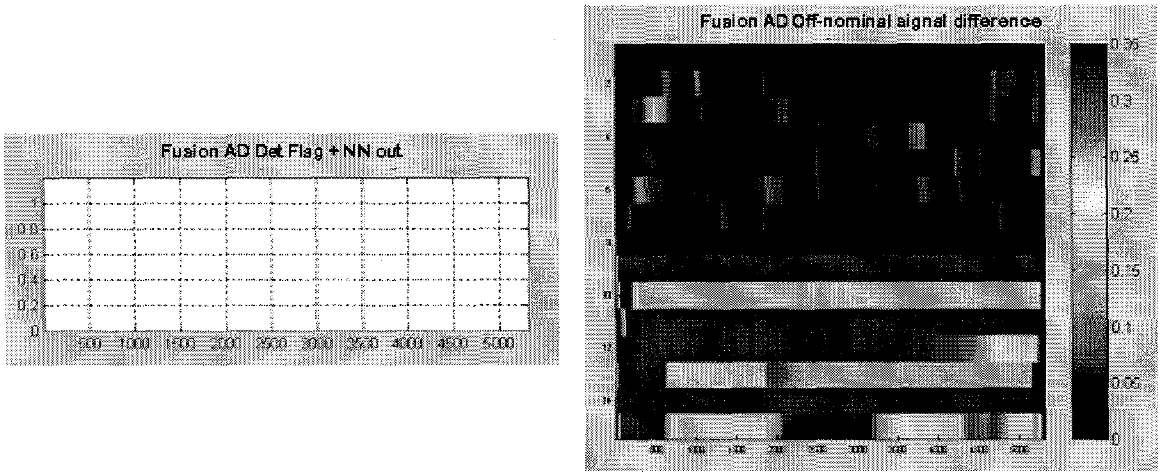


Figure 17 Anomaly APU fusion AD outputs

Figure 17 shows the output from the fusion processing. As seen in the figure, the anomaly is detected. The signals that indicted are those flagged by NNAD. This gives additional information regarding which AD gave rise to the detection and may offer additional information that downstream reasoners may be able to use.

7. SUMMARY AND CONCLUSIONS

In this paper we have presented the results of fusing two different anomaly detectors (ADs) together to form a single AD output. The two input ADs are the Neural Net Anomaly Detector (NNAD) [2] and the Generalized Cross-Signal Anomaly Detector (sometimes called BEAM) [3]. The two different ADs focus on different aspects of real data signals when performing their individual detections.

Sometimes the detectors are 'complimentary' and support each other's detections. In this case fusion improves confidence of the detections and thus not only improves detection performance but reduces false alarms as well.

However, sometimes a particular detector focuses on an aspect of the signals not considered by the other detectors. In this case it provides the only anomaly detection. This expands the class of signals that the fused AD is able to process. Fusion is a way of approaching the utopian goal of perfect detection with zero false alarms.

8. References

[1] Brotherton, T.W., T. Johnson, and G. Chadderdon, "Classification and novelty detection using linear models and a class-dependent elliptical basis function neural network," *Proc. of The Int'l Joint Conf. on Neural Networks* (IJCNN), May 1998, Anchorage, AK.

[2] Brotherton, T. and T. Johnson, "Anomaly Detection for Advanced Military Aircraft using Neural Networks", *The 2001 IEEE Aerospace Conference,* Big Sky, MT, March 2001.

[3] Mackey, R., "Generalized Cross-Signal Anomaly Detection on Aircraft Hydraulic System", *The 2001 IEEE Aerospace Conference,* Big Sky, MT, March 2001.

[4] Owsley, L.M., L.E. Atlas, and G.D. Bernard, "Self-Organizing Feature Maps and Hidden Markov Models for Machine-Tool Monitoring," IEEE Transactions on Signal Processing Special issue on Neural Networks, November, 1997.

[5] Brotherton, T.W. and E. Mears, "Applications of Neural Nets to Feature Fusion," *The 26th Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, Oct. 1992.

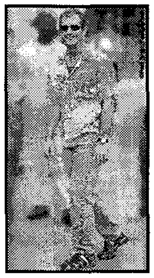
[6] Brotherton, T.W., T.G. Pollard, and D. Jones, "Applications of time-frequency and time-scale representations to fault detection and classification," *Proceedings of the IEEE-SP Int'l Symposium on Time-Frequency and Time-Scale Analysis,* Victoria, British Columbia, Oct. 1992.

[7] Hush, D. R. and B. G. Horne, "Progress in Supervised Neural Nets – What's New Since Lippman?," *IEEE Signal Processing Magazine*, Jan. 1993.

[8] Haykin, S, *Neural Networks: A* Comprehensive *Foundation*, second edition, Prentice Hall, 1999.

[9] Kohonen, T., Self-Organizing Maps, Springer-Verlag, Berlin, 1995.

[10] Mackey, R., M. James, H. Park, and M. Zak, "BEAM: Technology for Autonomous Self Analysis," *The 2001 IEEE Aerospace Conference*, Big Sky, MT, March 2001.



Tom Brotherton received his B.S. degree from Cornell University (1974), an M.A.Sc. from the University of Toronto (1976) and the Ph.D. from the University of Hawaii (1982) all in electrical engineering. He was an assistant professor in the Information and Computer Sciences Department at the University of Hawaii in 1983 and with the Orincon Corp from 1983-1999. He is currently a VP of the Intelligent Automation Corp. (IAC) in San Diego, CA. IAC is involved with the development of aircraft and aircraft equipment monitoring software and hardware. Dr. Brotherton is also on the editorial board for the IEEE Press series on Biomedical Engineering. His interests are in the development of adaptive signal and data fusion techniques for machine condition, fault monitoring, and prognostics as well as medical systems applications.

Ryan Mackey received his B.A. degree from the University of California at Santa Cruz (1993) for Mathematics and Physics, and went on to an M.S. (1994) and Eng. (1997) in Aeronautics at Caltech. He is presently a senior researcher and charter member of the Ultracomputing Technologies Research Group at the Jet Propulsion Laboratory. His research centers upon revolutionary computing methods and technologies for advanced machine autonomy, specifically deep space missions, UAVs and maintainable aerospace vehicles. His interests also include quantum- and biologically-inspired computing.