PATTERN-BASED FAULT DIAGNOSIS USING NEURAL NETWORKS *

W.E. DIETZ
E.L. KIECH
M. ALI

CENTER FOR ADVANCED SPACE PROPULSION
THE UNIVERSITY OF TENNESSEE SPACE INSTITUTE
TULLAHOMA, TN 37388

ABSTRACT

The detection and diagnosis of faults in real time are active areas of research in knowledge-based expert systems. Several methods of diagnosis have been applied to a variety of physical systems. Rule-based approaches have been applied successfully to some domains. However, encoding knowledge in rule bases raises many difficult knowledge acquisition issues; in addition, rule-based systems are often too slow to be effectively applied in a real-time environment. More advanced diagnostic systems may incorporate a simulation of the physical system in the knowledge base. Although simulation-based expert systems can exhibit powerful capabilities, simulating the domain properly may be difficult and too computationally intensive for real-time diagnosis.

An effort is underway at The University of Tennessee Space Institute to develop diagnostic expert system methodologies based on the analysis of patterns of behavior of physical mechanisms. In this approach, fault diagnosis is conceptualized as the mapping or association of patterns of input data (e.g., from instrumentation) to patterns representing one or more fault conditions. Associative memories and neural networks are being investigated as a means of storing and retrieving fault scenarios, as they offer several powerful and useful features, including 1) general mapping capabilities, 2) resistance to noisy input data, 3) the ability to be trained in a supervised learning mode, and 4) the capability of operation with incomplete input.

Pattern-based fault diagnosis and detection methodologies are currently being applied to jet and rocket engines. These domains are characterized by failure scenarios which may be catastrophic, and may occur over very short time periods. A requirement of the present study is that diagnoses be performed in real time, in order to allow time for effective action to be taken prior to possible engine destruction.

This paper 1) outlines an architecture for a real-time pattern-based diagnostic expert system capable of accommodating noisy, incomplete, and possibly erroneous input data, and 2) presents results from prototype systems applied to jet and rocket engine fault diagnosis.

INTRODUCTION

The real-time diagnosis of faults occurring in complex physical systems is an active area of research in the field of knowledge-based expert systems. Real-time environments present many challenges which must be effectively addressed by the expert system designer. Most importantly, a real-time diagnostic system must be capable of performing an accurate diagnosis quickly enough for effective remedial action to be taken. The diagnostic system must also exhibit resilience when presented with noisy input or failed sensors.

Many diagnostic systems encode domain knowledge in the form of rules. Rule-based systems are an effective approach in many domains, but are often too slow for real-time environments.

Model-based expert systems apply a qualitative or numerical simulation of the problem domain to the diagnostic process (Ref. 1-3). These systems reason on the basis of physical principles, and therefore are capable of performing diagnoses for a wide range of inputs. However, simulation models are usually too slow to be effectively applied in a real-time environment.

*This work was performed within the Center for Advanced Space Propulsion and was supported in part by NASA grant NAGW-1195.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specfic permission.

© ACM 1988 0-89791-271-3/88/0006/0013 $1.50

Knowledge acquisition is a problem area shared by both rule and model-based expert systems. Rule bases must often be tediously hand-encoded and are not suitable for representing non-causal knowledge. The development of reliable simulation models is also generally a difficult and time-consuming process.

An effort has been undertaken at The University of Tennessee Space Institute to develop diagnostic expert systems applied to the jet and rocket engine domains. An approach to real-time diagnosis has been developed and implemented into prototype diagnostic systems which address many of the problem areas encountered in the development of rule and model-based expert systems. The approach, presented herein, encodes behavioral knowledge of jet and rocket engine fault conditions in neural networks, which can then be used to identify fault conditions based on input from a real-time environment.

In the current approach, fault diagnosis is conceptualized as the association or mapping of patterns of input data representing the behavior of a physical system to one or more fault conditions. Behavior of a physical system is often described in terms of the temporal behavior of various physical parameters relating to the system. For instance, the combustion chamber temperature of a rocket engine as a function of time describes in part the behavior of the engine. An expert would recognize some behaviors as normal, while other behaviors would indicate a fault condition.

The diagnostic process employed in the present effort is illustrated in Fig. 1, which depicts a mapping process from patterns of sensor data to a pattern associated with a particular fault condition. Each sensor is associated with a classifying system, which determines which fault condition is indicated by that sensor. A real number between 0 and 1 is output by each classifier for each possible fault condition. An output value near 1 results when the classifier determines that the input behavioral pattern corresponds closely to a known fault behavior. The output of each classifier then is input into an arbitrator, which determines the output of the diagnostic system as a whole. The arbitrator has four outputs; three which are associated with fault conditions, and one output which is activated in the event an unknown pattern is presented to the diagnostic system. In this example, sensors $S1$ and $S3$ have identified a fault as type $F1$, while sensor $S2$ has identified the fault as type $F2$. The discrepancy may be due to several factors, including noise or a faulty sensor. The arbitrator combines the outputs of the classifiers and outputs a pattern of real numbers corresponding to the output of the system as a whole. The arbitrator may simply average the outputs of the sensors, or it may apply appropriate domain knowledge and assign some sensors more weight than others. In Fig. 1, the arbitrator output indicates strongly the occurrence of fault $F1$.

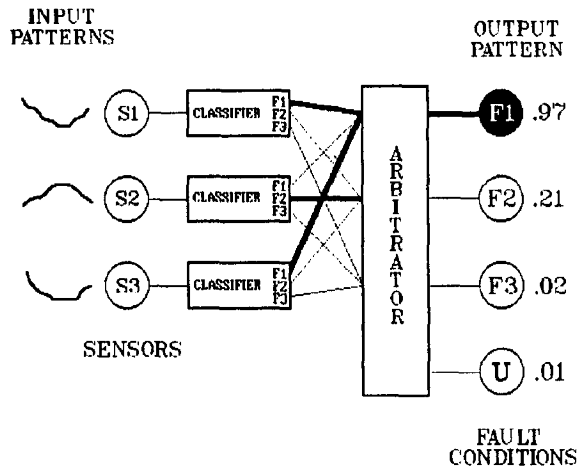


Figure 1. Diagnosis as Behavioral Pattern Mapping

Recently, interest in the application of associative memories and neural networks to problems encountered in expert systems development has increased. There is extensive literature regarding the application of associative memories and neural networks to pattern recognition and mapping (Refs. 2,3). Neural networks appear to offer features which coincide well with the requirements of pattern-based diagnosis. In the present effort, neural networks are applied primarily as devices to perform mappings of patterns of input data to output patterns representing associated fault conditions. Other aspects of neural networks are attractive from the standpoint of diagnosis as well. Knowledge acquisition obstacles generally found in other expert system paradigms appear to be addressed effectively by neural networks, as they can be trained to respond appropriately by example. The actual mapping process, from input patterns to output patterns, requires little computational effort, and is therefore ideally suited for real-time applications. Also, neural networks are resistant to input noise which may be present in data acquisition systems.

NEURAL NETWORK IMPLEMENTATION

Many types of neural networks have been developed and are extensively reviewed in the literature (Ref. 4,5). A basic overview of neural networks and their implementation in the prototype diagnostic systems is presented in this section.

The basic unit of a neural network is a processing element or node, depicted in Fig. 2. Usually, a processing element has one or more inputs and a single output. Each input has an associated activation and weight. The purpose of the processing element is to 1) apply a function (called an activation function) to the sum of the product of the input activations and the weights and 2) output the result. The output of the processing element is called the activation of the element.

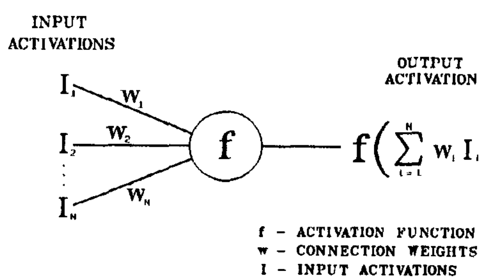


Figure 2. Processing Element

A neural network consists of a system of interconnected processing elements. Many possible connection topologies have been discussed in the neural network literature (Ref. 4,5). The topology of the networks used in the present effort is depicted in Fig. 3. This topology consists of three layers. The first layer consists of input nodes, the outputs of which are connected to a second layer of nodes. The second layer is called the "hidden" layer, because it is only indirectly affected by the application of external stimuli to the input nodes. The outputs of the hidden nodes are connected to the third layer, which consists of the output nodes of the neural network.

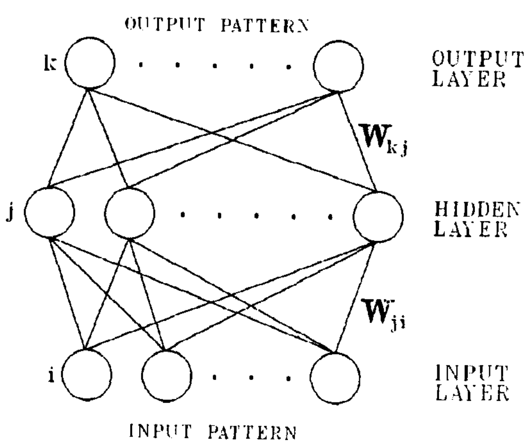


Figure 3. Neural Network Topology

Each connection is associated with a weight, which in the present implementation is a real number. The input nodes will be activated by an external stimulus derived from sensor data. The activations of the input nodes are multiplied by the weights of the connections between the input and hidden layer; the resulting values become the inputs to the hidden layer. The activations of the hidden layer are then multiplied by the weights of the connections between the hidden and output layer; the resulting values become the inputs to the output layer. The activations of the output layer represent the output of the neural network.

The neural network can be represented as a matrix system. If $[I]$ is a vector of activations of the input nodes, and $[H]$ is the resulting activation of the hidden layer, then

$$[H] = \sigma([M][I])$$

where $[M]$ is a matrix of weights assigned to the connections between the input and hidden layer, and σ is the activation function of the hidden layer. A similar system can be defined between the hidden and output layers. If $[N]$ is a matrix of weights of the connections between the hidden and output layers, and ρ is the activation function of the output layer, then the output activation $[O]$ of the entire neural network can be written

$$[O] = \rho\{[N]\sigma([M][I])\}$$

The type of activation function employed in the present study is known as a semilinear activation function, and has the form

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

The activation function has a minimum value of 0 and a maximum value of 1, and is essentially a differentiable approximation to a step function. The rationale for choosing certain forms of activation functions is discussed in Ref. 6.

Calculation of the output of the neural network involves only two matrix multiplications and two applications of the activation functions. As a result, the use of neural networks requires very little computational overhead, and allows diagnoses to be accomplished in real time. In addition, neural networks can be conveniently implemented in computers with parallel architectures, as many of the operations of matrix multiplication can be performed concurrently.

The neural network formulation also impacted the selection of the language used in the present effort. Since the neural networks involve mostly mathematical calculations rather than symbolic manipulation, the diagnostic systems described herein were written in FORTRAN.

KNOWLEDGE ACQUISITION WITH NEURAL NETWORKS

The behavior of the neural networks depends on the weights assigned to the connections between the layers. The assignment of the weights is accomplished when a neural network is trained to associate input patterns representing sensor data to output patterns representing particular fault scenarios. There are various methods of assigning connection weights (Ref. 5), depending on the type of knowledge to be encoded and the topology of the network. The method used in the present study is a gradient- descent method which finds a system of weights which minimizes the error term

$$E = \frac{1}{2}\sum_{j=1}^{N} (T_{jp} - O_{jp})^2$$

where T_{jp} is the desired activation of output node j resulting from an applied input pattern p, O_{jp} is the actual activation occurring at output node j, and N is the total number of output nodes. The method used to perform this minimization is known as the generalized delta rule, or generalized back-propagation algorithm, and is described in Ref.7.

BEHAVIORAL ANALYSIS WITH NEURAL NETWORKS

In some cases, it is sufficient to detect the deviation of an engine parameter from a known normal level for a fault condition to be identified. However, how the parameter deviates from a normal condition may be important in correctly identifying a fault. For instance, whether the combustion chamber temperature of a rocket engine exhibits linear or exponential behavior may be important in not only determining the nature of the fault, but also in determining optimum action to be taken in order to avoid damage (Ref. 8). If the engine behavior were to be represented as sets of curves of engine parameters versus time, this would require analyzing not only the magnitudes of the parameters, but also the shapes of the curves as well.

Fig. 4 illustrates the type of capabilities that neural networks can bring to the diagnostic process. Nine curves are shown that represent three distinct types of behavior of a (fictitious) physical system. The curves labeled as Type A are linear, those of Type B are asymptotic, and those of Type C exhibit a minimum point. Each type of curve may be associated with a particular fault condition. The nine curves comprised a training set for a single neural network, which was trained to activate one of three output nodes when presented with a curve representative of one of the three types of curves in the training set. A noisy linear curve, representing the behavior of a system in a real-time environment, is categorized by the trained neural network as a Type A curve, even though the curve is far from the closest Type A curve in the training set.

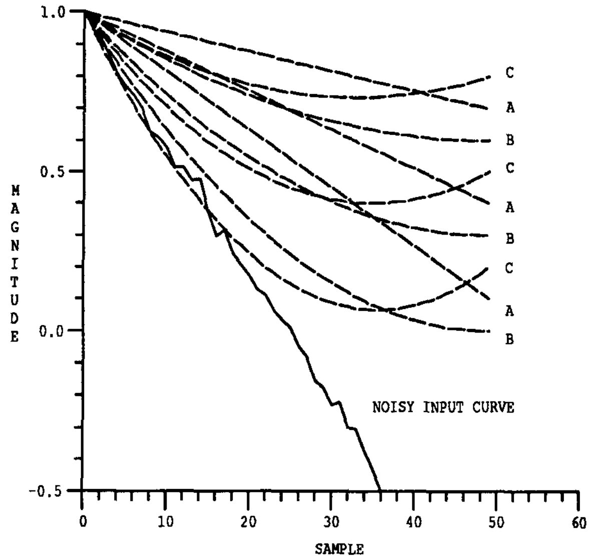


Figure 4. Neural Network Training Set and Input

The important points to be noted in Fig. 4 are 1) the training sets overlap, but do not affect the ability to train the network or retrieve a categorization of an unknown curve, 2) the network categorizes on the basis of qualitative features such as shape, rather than on the basis of strictly quantitative features such as magnitude, 3) an unknown curve does not have to be close to a curve in a training set (in a quantitative sense) to be categorized correctly, and 4) the categorization is resistant to input noise. All of these features are exploited in the prototype diagnostic expert systems presented herein.

ARCHITECTURE OF THE JET ENGINE DIAGNOSTIC SYSTEM

A jet engine is an analog system. As a result, fault conditions can occur with varying degrees of severity, and can develop suddenly or over longer periods of time. Determining the ultimate cause of a fault may require knowledge about the severity or duration of development. For instance, a sudden, severe fuel interruption in a jet engine may indicate a rupture in a fuel line, while a slight interruption developing over a longer period may indicate a blockage. It is therefore important in an analog system to not only identify fault conditions but to also attempt to assess the severity of the fault, and whether the fault condition developed suddenly or slowly. The prototype diagnostic system for the jet engine is designed to identify the type of fault which is occurring, and determine the severity and duration of the fault condition.

The structure of the prototype diagnostic system is depicted in Fig. 5. Sensor data is input into a neural network which has been trained to recognize the difference between behavior exhibited by bearing failures, and that exhibited by fuel interruptions. Once a gross identification of the fault has been made, the sensor data is passed to lower-level neural networks which have been trained to recognize the severity and duration of the fault. Therefore, each sensor has five associated neural networks; one to determine the fault type, and four to determine severity and duration of bearing failures and fuel interruptions.

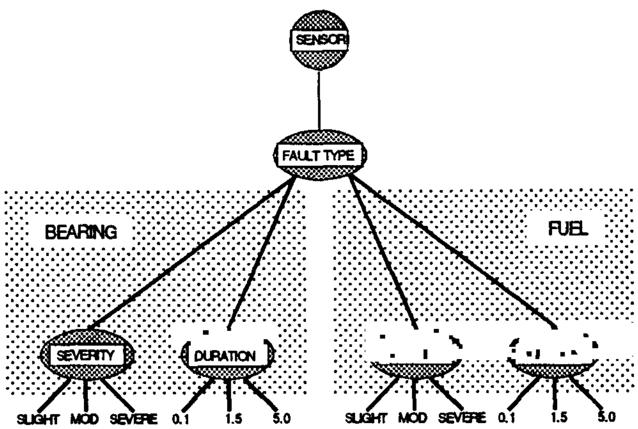
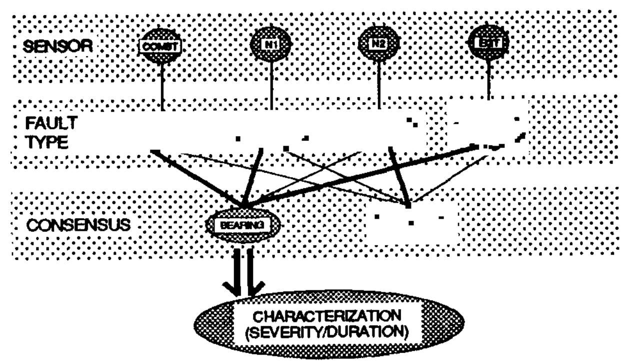


Figure 5. Architecture of Jet Engine Diagnostic System

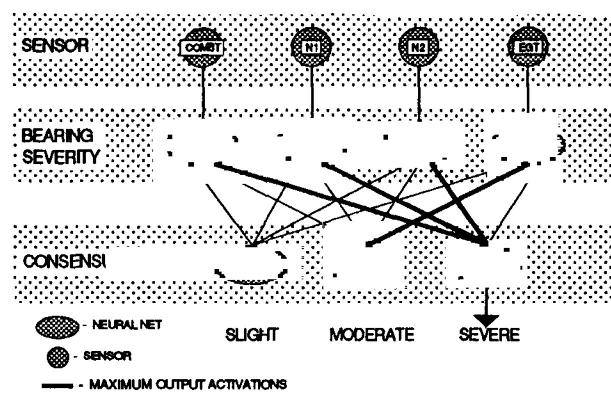
Four sensors are presently employed in the prototype, measuring combustion temperature ($COMBT$), exhaust gas temperature (EGT), low pressure turbine rotational speed ($N1$), and high pressure turbine rotational speed ($N2$). The neural network structure depicted in Fig. 5 is duplicated for each sensor. Therefore, the prototype system consists of a total of 20 neural networks.

In the examples presented herein, top-level networks have two output nodes. One output node is activated if a bearing failure is indicated, while the other is activated if a fuel interruption is detected. The output activations are real numbers between 0 and 1; this output format allows a network to indicate that an input behavioral pattern exhibits features common to both fault scenarios. Lower level networks have three output nodes, each of which identifies either one of three severity levels or one of three durations.

The overall operation of the diagnostic system is depicted in Fig. 6. In Fig. 6 a), data from the four sensors are input into the four top-level neural networks, each of which independently attempts to identify the fault condition. If a network recognizes an input pattern as characteristic of, for example, a bearing failure, the node associated with bearing failures will be highly activated. Corresponding output activations from each top-level network are averaged to yield the overall top-level response to the input patterns. The result is a characterization of the fault as either a bearing failure or a fuel interruption. In the example shown, the neural networks associated with $COMBT$, $N1$, and EGT exhibit maximum output activations indicative of a bearing failure, while the $N2$ network has recognized its input as being indicative of a fuel interruption. On the average, the top-level networks have recognized a bearing failure.



a) Fault Identification



b) Determination of Fault Severity

Figure 6. Diagnostic Process

At this stage no information about fault severity or duration has been obtained. The diagnostic system now attempts to determine the severity and duration of the fault. This is accomplished by applying the same sensor data which was used for the top-level fault identification to the lower-level neural networks. Each sensor will input information into two neural networks; one which determines fault severity, and one which determines duration. Fig. 6 b) depicts the sensor data being input into the networks which determine fault severity. The output activations corresponding to each sensor are averaged, and the result is output as part of the diagnosis. In this example a severe bearing failure has been identified. A similar procedure is followed to determine fault duration.

BEHAVIORAL REPRESENTATIONS

To train and use neural networks, behavioral data must be presented to the inputs of the networks. Two methods of representing behavioral data have been examined in the present effort. The two methods are illustrated in Fig. 7. In the first method, data is presented to the input layer of the neural nets as a vector of continuous real variables. Each node in the input layer corresponds to a point in time since a transient (i.e., a deviation from a known normal level) was detected. The activation of each input node is the magnitude of a parameter at each point in time. Therefore, each input node represents a sampling of the data at different times. In the prototype implementation, 40 input nodes were used, spanning a 4.0 second time interval. As a result, adjacent input nodes contain data 0.1 seconds apart. Ten nodes were utilized in the hidden layer.

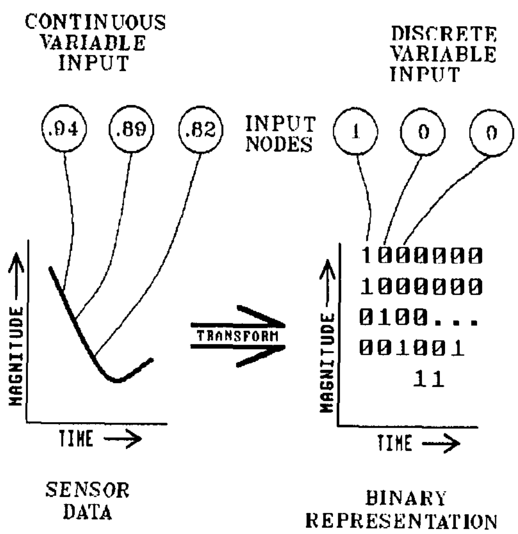


Figure 7. Input Representations

In the second method, parameter behavior is represented in a binary form. The curve describing the temporal behavior of the parameter is represented as a field of 1's and 0's, corresponding roughly to a low-resolution image of the curve. The field is represented as a single binary vector which is presented to the input of the neural nets. Twenty time intervals and 20 magnitude intervals are encoded. As a result, the neural network for this representation requires 400 input nodes. Four hidden nodes were utilized in this network implementation.

The first method of behavioral representation (continuous-variable input) is simply a mapping of sensor data into a 40-component vector. The binary representation of the second method requires an additional step; the 40-component continuous-variable vector must be transformed into a 400-component binary vector. The additional computational overhead does not significantly diminish the speed at which a diagnosis can be obtained.

It is assumed that all engine parameters have attained steady-state values (corresponding to normal operation) prior to the start of a transient indicating the occurrence of a fault condition. All sensor data were normalized with the pre-transient steady-state values. It was found that normalizing the sensor data greatly aided the stability and convergence rate of the training process.

TRAINING THE NEURAL NETWORKS

The jet engine diagnostic system was trained using data from an engine simulation program called ATEST (Advanced Turbine Engine Simulation Technique, Ref. 9), which is capable of calculating both the steady-state and transient response of a jet engine. Many engine simulation programs have been developed over the past 20 years (Ref. 10). ATEST is one of the more recent and advanced engine simulation programs available, and was chosen on the basis of its extensive use, reliability, and accuracy. Fuel interruption scenarios for the training sets were generated by decreasing the fuel flow linearly by amounts of 15, 50, and 75 percent over periods of 0.1, 1.5, and 5.0 seconds. Bearing failures were simulated by extracting energy from the engine turbine shafts linearly over time. Severity levels and time durations used for the bearing failures were the same as those used for the fuel interruption scenarios.

Fig. 8 illustrates the training method applied to the neural networks in the jet engine diagnostic system. In this example, the training process used for a top-level network is shown. The network has two output nodes, each of which is associated with either a bearing failure or fuel interruption. During training, a behavioral pattern representing a fault condition is applied to the input level, while a 1, indicating full activation, is applied to the corresponding output node. In Fig. 8 a), for instance, a behavioral representation of a fuel interruption is applied to the input nodes, while an activation of 1 is imposed on the output node corresponding to a fuel interruption. The generalized back-propagation algorithm is then invoked to adjust the connection weights to be consistent with the imposed input and output patterns.

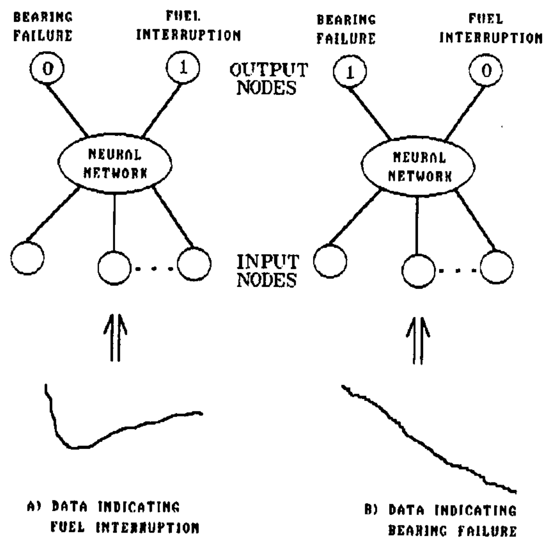


Figure 8. Training Neural Networks

At this point the network has been trained to recognize a single representation of a fuel interruption scenario. The training process is now repeated with a representation of a bearing failure scenario presented to the input nodes, and an activation of 1 imposed on the corresponding output node (Fig 8 b). The connection weights are readjusted to accommodate the new input; in the process, the ability of the network to recognize the first input (the fuel interruption scenario) will be degraded. The network must in effect be repeatedly retrained on the two scenarios until the system of weights converges to steady values.

A training set consists of more than one input pattern to be associated with a particular output. For instance, nine different fuel interruption input scenarios (all combinations of the three severity and duration levels) are associated with the output assigned to identify fuel interruptions. Similarly, nine bearing failure scenarios are associated with the output node signifying the identification of a bearing failure. As a result, each top-level network is trained with 18 input representations, nine associated with bearing failures, and nine associated with fuel interruptions.

The four lower-level networks associated with each sensor are designed to categorize input scenarios according to fault severity and duration. Nine input patterns, consisting of combinations of three severity levels and three durations, are used to train the lower-level networks. Each lower-level network has three outputs, corresponding to the different levels of severity or duration, depending on how the network is trained. The output activations associated with training sets used in the lower-level networks are depicted in Fig. 9. The severity classifiers are taught that all faults of the same severity are to be classified together, regardless of the fault duration, as shown in Fig. 9 a). The networks designed to classify fault durations are taught that all faults of the same duration are to be classified together, regardless of the severity of the fault, as illustrated in Fig. 9 b). The same input patterns are used to train each network; how the training patterns are grouped determines the function of the network.

DURATION	SEVERITY 15% 123	50% 123	75% 123	
0.1	001	010	100	OUTPUT ACTIVATIONS
1.5	001	010	100	
5.0	001	010	100	

A) OUTPUT ACTIVATIONS FOR SEVERITY CLASSIFIER

DURATION	SEVERITY 15% 123	50% 123	75% 123	
0.1	001	001	001	OUTPUT ACTIVATIONS
1.5	010	010	010	
5.0	100	100	100	

B) OUTPUT ACTIVATIONS FOR DURATION CLASSIFIER

Figure 9. Training Sets for Lower Level Networks

Training represents the most computationally intensive aspect of the development of a neural network-based diagnostic system. Time required for training a network can vary widely, depending on the number of nodes in a network and the number of training sets used. However, the similarity of input representations in the training set has the most significant effect on training times; if two input representations are nearly equivalent, but are required to activate different output nodes, the training times can become excessive. In the jet engine diagnostic system, most networks were trained after 1000 to 2000 presentations of input and output patterns. The top-level network for N1, which contained a large number of similar patterns which were required to activate different output nodes, required over 85000 presentations of data for convergence of the weights to be achieved.

RESULTS OF JET ENGINE DIAGNOSES

Testing and evaluation of the jet engine diagnostic system focused on the ability of the system to correctly diagnose fault conditions of varying degrees of severity and duration. Input patterns representing either a bearing failure or fuel interruption scenario were presented to the inputs of the top- level networks. The system was evaluated on the basis of 1) whether the top-level networks correctly identified the fault condition, and 2) whether the lower-level networks correctly identified the severity and duration of the fault. Severity and duration combinations were chosen which were not in the training sets of the neural networks. Severity levels tested ranged from 10 to 80 percent; durations ranged from 0.1 second to 5 seconds.

All real-time environments exhibit some level of noise from instrumentation. The effects of noise on the response of the diagnostic system were assessed by randomly perturbing the inputs to the neural networks. Random perturbations were simulated by selecting values from a set of normally distributed random numbers with a mean of zero and a standard deviation equal to one fifth of the stated noise band value. This ensures that approximately 99 percent of the resultant perturbations fall within the defined noise band. For instance, a 10 percent noise level means that 99 percent of the perturbed values will be within 10 percent of the unperturbed levels.

The diagnostic systems utilizing either the first or second methods of data representation (continuous-variable or binary input, respectively) successfully differentiated between a bearing failure and fuel interruption scenario for all durations and severities tested. Accuracy of diagnoses was maintained under noisy conditions usually encountered in a data acquisition environment. Only extreme noise levels (> 15 percent) began to affect fault identification adversely. Since most data acquisition environments exhibit much lower noise levels, the ability of the system to distinguish between fault conditions was considered to be extremely reliable.

The ability of the diagnostic system to determine the severity and duration of fault conditions is illustrated in Figs. 10-12. Results are shown for the system utilizing the first method of data representation (continuous-variable input). A series of input patterns representing a fuel interruption scenario of 0.75 second duration and severities ranging from 10 to 80 percent was presented to the diagnostic system. The curves indicate the activations of the three output nodes of the lower- level neural networks which were trained to recognize fault severity. The nodes representing 15, 50, and 75 percent fuel interruptions achieve maximum activation levels at the proper severity levels, even though the duration (0.75 seconds) corresponds to a value for which the networks were not trained.

Fig. 11 depicts the response of the diagnostic system to a series of input patterns representing a fuel interruption of 65 percent severity, and durations ranging from 0.1 second to 5.0 seconds. The three curves represent the activations of the output nodes of the lower-level networks which were trained to identify fault duration. The output nodes indicating 0.1, 1.5, and 5.0 second fault durations are activated at the appropriate durations, even though no training set included a 65 percent severity fault scenario.

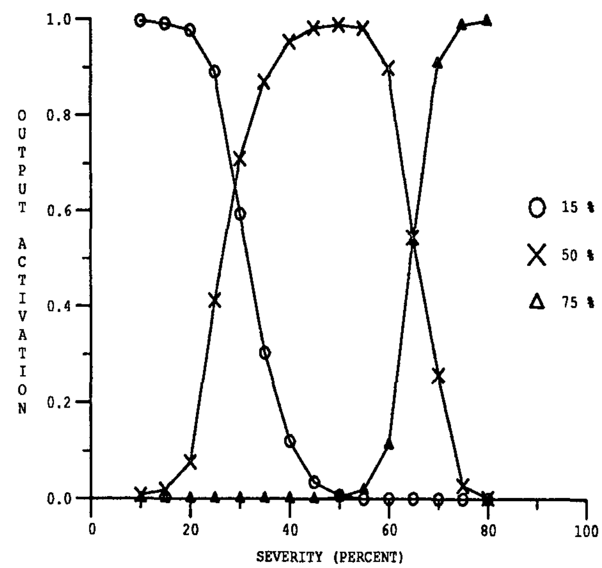


Figure 10. Response of Jet Engine Diagnostic System to 0.75 Second Fuel Interruptions (0 Percent Noise)

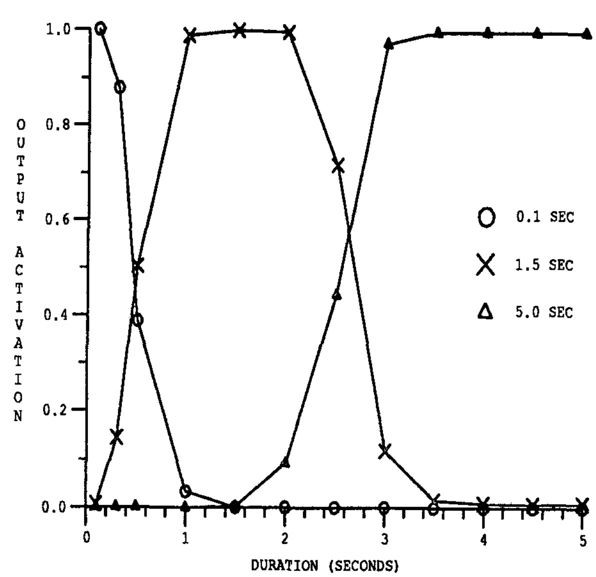


Figure 11. Response of Jet Engine Diagnostic System to Fuel Interruptions of 65 Percent Severity (0 Percent Noise)

The output activations depicted in Fig. 10 show smooth transition between minimum and maximum values. Since each input pattern invoked a unique output activation pattern, the output activations can be used to estimate fault severity.

The output activations depicted in Fig. 11, on the other hand, display wide ranges of maximum activation. For instance, the output activation pattern related to 5.0 second fault duration is nearly identical to that of 3.5 seconds. As a result, a long-duration fault can only be estimated to lie between 3.5 and 5.0 seconds; more precise estimations are limited by the non-uniqueness of the output activation patterns. The cause of non-unique output activations is currently under investigation.

The effect of input noise on the response of the diagnostic system is illustrated in Fig. 12. Random noise patterns of 10 percent intensity were added to the fault scenarios obtained from the engine model; the resulting patterns were then presented to the diagnostic system. Fig. 12 depicts the response of the system to noisy fuel interruption behavioral patterns of 0.75 second duration and severities ranging from 10 percent to 80 percent. The response curves are distorted compared to the noise-free case (Fig. 10), but the overall response is preserved. The activations achieve maximum values at severity levels equal or very close to the correct values. Some plateauing effect is seen near the curve peaks, resulting in some degradation in the ability of the system to correctly estimate fault severity. In general, the diagnostic system is highly resistant to noise levels commonly experienced in data acquisition environments.

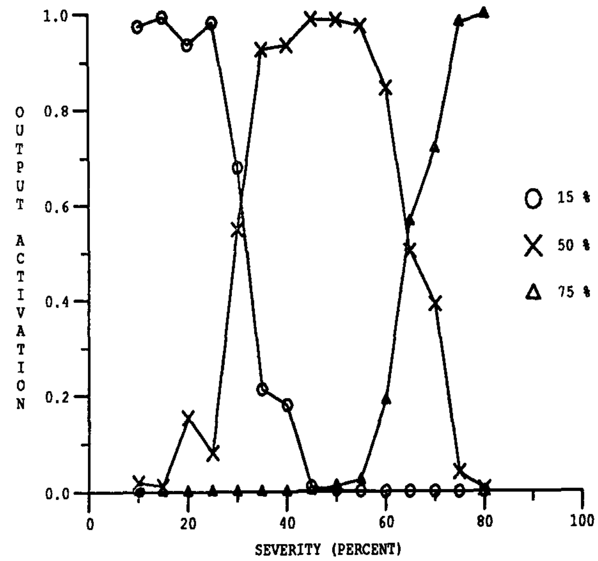


Figure 12. Response of Jet Engine Diagnostic System to 0.75 Second Fuel Interruptions (10 Percent Noise)

The results displayed by the second method of data representation (binary representation) did not differ significantly from the results obtained for the continuous-variable input. The system responses and resistance to input noise corresponded closely for both types of input representations.

The preceding examples are diagnoses based on analyses of data over a period of four seconds. However, diagnostic information can be obtained from analyses of partial input. A "criticality factor", defined as the ratio of fault severity to duration, can be quickly obtained from the neural network implementation utilizing the binary behavioral representation.

The use of a criticality factor is necessary because certain combinations of fault severities and durations will result in initially identical engine behaviors. Consider a fuel interruption with duration of 3.0 seconds and a severity of 20 percent. The resulting behavior of the engine over the 3.0 second interval will be identical to the behavior (in the first 3.0 seconds) caused by a fuel interruption of 6.0 seconds duration and 40 percent severity, assuming the fuel interruption occurs in a linear fashion. The initial behavior of the engine is therefore determined by the rate of change of fuel flow. The average rate of change can be viewed as the ratio of the fault severity to the duration. Therefore, both fault scenarios (20 percent over 3.0 seconds, and 40 percent over 6.0 seconds) have the same criticality factor of 0.067. A diagnostic system could not determine fault severity or duration over the 3.0 second interval; a system could, however, determine a criticality factor.

The binary input representation allows the representation of partial behavioral patterns. For instance, if only the first second of a behavioral pattern is available, the input nodes corresponding to subsequent time intervals are simply set to 0, i.e., a 0 in the binary representation simply means that no information is available, or is unknown. Representation of partial input patterns with the continuous-variable input representation is not possible in the present implementation. A 0 input in the continuous-variable input representation will be interpreted to mean that the corresponding sensor value is known, and known to be 0.

A partial diagnosis is illustrated in Fig. 13, which depicts the criticality factor calculated by the diagnostic system as a function of the number of input samples. The correct criticality factor for a 75 percent, 0.1 second fuel interruption (7.5) is calculated using slightly more than half the input samples usually used by this network implementation.

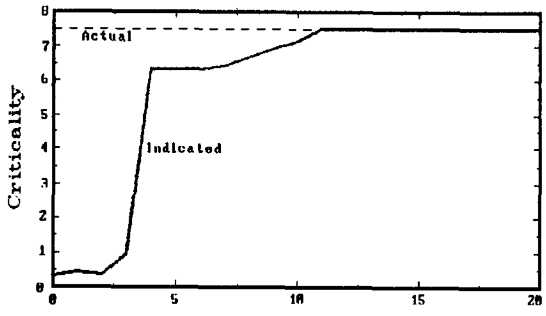


Figure 13. Response to Partial Inputs

THE ROCKET ENGINE DIAGNOSTIC SYSTEM

In another effort, we are implementing neural network-based diagnostic systems for the Space Shuttle Main Engine (SSME). Unlike the jet engine, a numerical simulation model for the SSME is not currently available. Therefore, this effort is examining methods of employing actual experimental data obtained from engine ground tests to the training and evaluation of a network-based diagnostic system.

The SSME has been extensively tested in ground-based facilities; some of these tests have resulted in the occurrence of several different types of fault conditions. The data from these tests were used as training sets for the SSME diagnostic system.

Two types of experimental data were available from SSME tests. The first type was experimental curves, representing the temporal behavior of various engine sensors during testing activities. The second type was summaries of curve features, which included 1) an "excursion interval", defined as the time from detection of a transient to the maximum deviation of a sensor value from the pre-transient level, and 2) an average rate of change of a sensor reading over the excursion interval. Twenty sensors were present on all tests in which a fault condition occurred. The sensors measured various engine parameters, such as main combustion chamber pressure, turbine discharge temperatures, turbine speeds, and valve positions.

In the first prototype system, an architecture similar to the jet engine diagnostic system is being developed for the SSME. To date, investigations have been centered on 1) training the neural networks using noisy experimental data and 2) categorizing input patterns which represent curves similar to the training sets, but with different noise patterns imposed. Because of limited experimental curve data, the neural networks using a binary input representation were trained with one example for each sensor. With this single training example the networks could reliably categorize input patterns exhibiting up to noise levels of 20 percent.

The second prototype system uses the curve feature data summaries as input. Since excursion intervals and average rates of change were available for all twenty parameters, a neural network was implemented with 40 inputs. The engine behavioral patterns for each fault were therefore represented as a vector of 40 real numbers.

Data summaries from 27 tests were available for training the neural networks. The 27 tests resulted in failure scenarios involving six main structural groupings within the SSME: 1) injectors, 2) controls, 3) duct, manifold, and heat exchangers, 4) valves, 5) high pressure oxidizer turbopump, and 6) high pressure fuel turbopump.

A neural network topology was implemented which consisted of 27 output nodes, each identified with a specific test. A network implementation of this type is designed to identify the closest test scenario corresponding to an unknown input.

All 27 data summaries were used as training sets. Testing was done by presenting perturbed versions of the training sets to the networks, and examining which of the 27 test scenarios were activated. Perturbing the data summaries does not necessarily reflect the direct effects of instrument noise, but may include errors which occur extracting curve features. Many conventional feature extraction algorithms are highly susceptible to errors generated not only by noisy sensor data, but also by assumptions inherent in the feature extraction algorithms themselves. In all cases, the perturbed versions of the training sets were identified correctly up to perturbation levels of 30 percent.

CONCLUSIONS

The results of the diagnostic expert systems presented in this paper demonstrate the ability of a neural network-based system to 1) be trained by being presented with behavioral patterns associated with fault conditions, 2) identify faults, and fault severity and duration, when presented with input patterns for which the systems have not been specifically trained, 3) render accurate and reliable diagnoses when presented with noisy data, and 4) produce partial diagnoses when presented with incomplete behavioral representations. Though the response time of the diagnostic systems is already close to the real-time requirement, the matrix structure of neural networks allows further efficiency to be obtained by employing parallel algorithms and machines.

ACKNOWLEDGEMENT

This work was performed within the Center for Advanced Space Propulsion (CASP) and was supported in part by NASA Grant NAGW-1195. The Center for Advanced Space Propulsion is part of The University of Tennessee-Calspan Center for Aerospace Research, a not-for-profit organization located at UTSI. The authors would like to thank the Rocketdyne Division of Rockwell International for their cooperation in the use of rocket engine ground test data. The authors would also like to thank Captain James C. Kinnard, Harry J. Ferber, and Mark Chappell for their assistance during the course of this effort.

REFERENCES

1. Davis, R., "Diagnostic Reasoning Based on Structure and Behavior", pp. 347–410, in "Qualitative Reasoning About Physical Systems", Bobrow, D., editor, The MIT Press, MIT, Cambridge, MA, 1986.
2. Dietz, W. E. and Ali, M., "Qualitative and Temporal Reasoning in Engine Behavior Analysis and Fault Diagnosis", The Third Conference on Artificial Intelligence for Space Applications, Huntsville, AL, Nov. 2-3, 1987.
3. Gupta, U. and Ali, M., "Hierarchical Representation and Machine Learning from Faulty Jet Engine Behavioral Examples to Detect Real Time Abnormal Conditions", The First International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, The University of Tennessee Space Institute, Tullahoma, TN, June 1-3, 1988.
4. Kohonen, T., Oja, E. and Lehtio, P., "Storage and Processing of Information in Distributed Associative Memory Systems", pp. 105–143, in "Parallel Models of Associative Memory", Hinton, G. E. and Anderson, J. A., editors, Lawrence Erlbaum Associates, Hillsdale, NJ, 1981.
5. McClelland, J. L., and Rumelhart, D.E., "Explorations in Parallel Distributed Processing", The MIT Press, Cambridge, MA, 1988.
6. Williams, R. J., "The Logic of Activation Functions", pp. 423–443, in "Parallel Distributed Processing", Vol. I, The MIT Press, Cambridge, MA, 1987.
7. Rumelhart, D. E., Hinton, G. E., and Williams, R. J., "Learning Internal Representations by Error Propagation", pp. 318–364, in "Parallel Distributed Processing", Vol I., The MIT Press, Cambridge, MA, 1987.
8. Cikanek, H. A., "Space Shuttle Main Engine Failure Detection", presented at 1985 American Control Conference, Boston, MA, June 19–21, 1985.
9. Chappell, M. A., "Advanced Turbine Engine Simulation Technique", AEDC-TR-86-19, 1986.
10. Fishbach, L. H., "Computer Simulation of Engine Systems", AIAA-80-0051, 1980.