# Space Shuttle Main Engine Fault Detection
# Using
# Neural Networks

**NETROLOGIC**

Thomas Bishop
Dan Greenwood
Kenneth Shew
Fareed Stevenson

5080 Shoreham Place, Suite 201
San Diego, California 92122
(619) 587-0970

## ABSTRACT

A method for on-line SSME anomaly detection and fault typing using a feedforward neural network is described. The method involves the computation of features representing time-variance of SSME sensor parameters, using historical test case data. The network is trained, using backpropagation, to recognize a set of fault cases. The network is then able to diagnose new fault cases correctly. An essential element of the training technique is the inclusion of randomly generated data along with the real data, in order to span the entire input space of potential non-nominal data.

## 1. Introduction

NETROLOGIC has devised a new system that uses neural networks for on-line detection of fault conditions in the Space Shuttle Main Engine (SSME). In order to recognize danger signs early enough to shut down the rocket engine and minimize damage resulting from unforeseen malfunctions, an SSME fault detection system needs to be faster and more accurate than existing systems. Even with the current failure response systems which utilize automatic redlining, redundant sensor and controller voting logic, and human monitoring, post test analysis shows the emergence of anomalous engine behavior well before a shutdown sequence is initiated. Neural networks can provide improved test-stand SSME fault detection with natural extensions to in-flight monitoring.

A fast SSME diagnostic method is essential since a large number of simultaneous sensor measurements (over 200 are available) are input to a test shutdown decision module at a high sampling rate. Sensor data fusion and evaluation are complicated issues since clues to engine performance may involve subtle combinations of sensor measurements varying through time. There is a high cost associated with unnecessary shut-downs (false alarms) as well as missed detections (failure to detect an impending catastrophe).

A detection system should not alter the current engine or control system and should utilize all existing data. Since the SSME's major components are line replaceable units, ideally a fault detection system should be independent of engine-to-engine performance variation and of older engine failure signatures.

Neural networks can contribute to an effective solution since they are

1) fast, especially if implemented on parallel hardware;

2) capable of discovering subtle patterns of input data without being explicitly taught what combinations are significant;

3) capable of generalizing based on previously learned examples; and

4) robust --- relatively insensitive to noisy data.

## 2. Data Source an' Description

We used the well-known backpropagation to train our three layer feedforward network with training examples from sensor data from actual SSME test cases (see Figure 1), conducted between 1981 and 1989. Most of the data resulted from recordings of cases in which faulty engine performance occurred. We restricted our attention to time periods after the SSME reached full power, since steady-state fault diagnosis is a sufficiently difficult and important problem, and the use of data from periods of transient SSME operation would introduce considerable complications. We will investigate the application of neural nets to failure detection during the transient phase in the near future. Neural nets can recognize distinctive time series such as temperature transients, and will be useful for rocket engine transient analysis.

The six fault cases that we have used represent failures of various types, caused by malfunctions in different hardware components, such as a fuel leak in the main combustion chamber outlet neck in one case, and a cracked liquid oxygen post in another. Although this provides a variety of data for training and testing, it also means that there is not enough fault data to generalize about any particular failure type.

In each of the fault cases we observed that there was a relatively long period during which the SSME functioned normally prior to malfunctioning, consequently, there was an abundance of nominal sensor data. However, there was a very limited amount of fault data in three cases, because the interval between the fault-declare time and the time of the last sensor measurements was very short (as short as 0.2 seconds).

The fault-declare time for each of the fault cases was based on an analysis of failure investigation reports which showed the time when sensors started to indicate signs of problems or faulty performance. We determined the time when a fault-detection system should have been able to declare that something was wrong enough to warrant shutting down the SSME. Sensor samples taken before the fault-declare time are considered nominal, and samples taken after that time are considered fault data.

We only used a subset of the total number of different sensor measurements, referred to as Parameter Identifiers (PIDs). These PIDs were sampled 25 times per second. We selected twelve PIDs (see Figure 2) for use in our current study. Selection of this subset of data was based on two factors:

1) Availability for all cases under investigation. Different test cases were inconsistent in which sensors were installed and functioning. Since a fundamental objective is to combine data from different test cases, and generalize to other cases, data must have the same format for all cases. Therefore we only chose a PID if it was available for nearly all of the cases used in our study. However, this is not an absolute restriction: if a particular PID is missing from a particular test case, it is possible to use null values for that PID in that case. In fact, it is essential that our method should accommodate missing, faulty, or "dead" sensors.

2) Significance for diagnosis. Analysis of fault case profiles shows that, for a given case, some sensors show strong early symptoms of faulty operation, while other sensors appear to have less value for diagnosis. Naturally we chose PIDs which were significant in the cases under investigation.

## 3. Pre-Processing of Data

The inputs to the network were derived from PID values. Each sample fed into the network corresponded to a particular point in time. However, the input values were not simply the raw values for each PID at that time. The nature of the variation in PID values over time may be more indicative of faulty performance than the value of the PIDs at any isolated moment. For example, in case 901-331, fault symptoms included an increasing HPOT discharge temperature concurrent with a decreasing MCC pressure. Therefore, for each point in time, three features were calculated for each PID, which take into account the medium, long, or short-term history of that PID leading up to that time. These features are described in Figure 3.

Thus, the total number of simultaneous inputs to the network for each point in time was three times the number of PIDs. We have used twelve PIDs and 36 input units. In future studies, more features will be computed for each sample, to provide more detailed input of time-variation of PIDs, or to explicitly input features which code relationships between other features. In theory, the network is capable of performing any computation on the inputs, so such compound features would be superfluous. In practice, however, it might prove to be useful to input such features explicitly in order to encourage the network to learn in a way that will lead to better generalization. The three features currently used are minimal, yet appear to be sufficient for the tasks attempted so far.

## 4. Network Architecture

We used a feedforward neural network model consisting of a layer of input units, plus one or more layers of hidden units, plus a layer of output units. Units are analogous to neurons. The connections between them are analogous to synapses. In the feedforward model, each of the input units is connected to each of the hidden units, and each of the hidden units is connected to each of the output units. Each of the connections is characterized by a weight, which is the strength of the connection. In the basic operation of the network, connections are one-way, going from inputs to outputs (hence the name feedforward). Each unit attains a level of activation by taking the weighted sum of its inputs. It then produces its own output, which is a function of its activation. We have used the logistic function given by $f(x) = 1 / (1 + \exp(-x))$.

Feedforward networks can be trained to associate arbitrary input patterns with arbitrary output patterns, and they have the ability to categorize and generalize, so that similar inputs are mapped to similar outputs, and new input patterns (different from those on which the network has been trained) will be mapped to outputs based on their similarity to training patterns. Training is accomplished by the generalized delta rule (backpropagation of error). After each input sample is fed forward through the network, the output is compared with the desired output. The weights are then adjusted iteratively to reduce any discrepancies (for a detailed description of backpropagation, please see [6])

The choice of how many hidden layers to use, and how many units to have in each layer, is dictated by two opposing factors. On the one hand, it is generally easier for a network to perform an exact mapping from a set of inputs to a desired set of outputs, if there are more hidden units. On the other hand, if there are too many hidden units, the network is liable to "over-learn" the training data, and may be less successful at generalizing to new data. We have found that a single hidden layer of three to six units is sufficient for the network mappings we have attempted so far.

## 5. Assignment of roles to output units

The output of the network represents its evaluation of the input data. The activations of the output units are all floating-point numbers, which take on values anywhere between zero and one. We currently use three output units, each of which represents a different diagnosis category. The three categories are:

1) Nominal

2) Fault (of a type previously witnessed)

3) Deviant (anything that departs from nominal).

For each output unit, activation levels near 1.0 mean "yes", and levels near 0.0 mean "no". Intermediate levels of activation may be regarded as the degree of confidence in that diagnosis.

The first priority of an SSME fault detection method must be to decide when to shut down the engine to minimize damage leading to a potential catastrophe. To the extent that this is a yes-or-no decision, we only need to know whether or not the engine's performance is nominal. This may be described as anomaly detection. Beyond this, however, it may be necessary to distinguish between different failure types. This will be true if different shut-down or safety procedures are employed depending on failure type. Also, if the neural network forms a part of a larger fault detection system, it may be of value for the network to report what failure type it perceives, thus providing a more useful input to the rest of the system.

Fault detection should involve the notification of a failure, the isolation of the type of failure, and the estimation of the severity. The detecti on of a failure which would warrant a shutdown sequence was emphasized, the isolation and estimation functions were secondary. Further study for isolation and estimation will also be pursued, however, a system which emphasizes detection during testing would alleviate some of the complexity or computational burden associated with pursuing all three goals of fault detection simultaneously.

Under the constraint of limited fault data, and keeping in mind the primary importance of shut-down decision making, we focused on anomaly detection rather than fault-typing, and employed only a single output unit for the "fault" category. In the future, when more fault data (real or simulated) becomes available, our method may be extended with no fundamental changes to incorporate more output units for individual failure types.

Using only historical nominal and fault data, the network can be trained to distinguish nominal and fault data that it is trained on, but when we ask it to generalize to new cases (cases that have not been used for training), the results may be disappointing. Unless a new case is very similar to one of the training cases, this new fault data will not resemble the old fault data any more than it resembles the old nominal data. In our experience, the network output "nominal" for all samples in the new faault cases, both before and after the fault-declare time. Evidently the problem was that the fault data in the training cases were too limited, involving only particular PIDs with specific time profiles. A network trained to recognize a particular small set of fault cases cannot be expected to recognize a new fault case, which is likely to involve different PIDs indicating degraded performance with completely new behavior.

In order to train a network to distinguish nominal data from all possible non-nominal data, we needed a source of non-nominal data. Fault data from real fault cases were insufficient for this purpose since, even if we used all the fault data currently available, it would still not span the entire space of potential non-nominal data. Therefore, we experimented with using random data evenly distributed throughout the total input space of the network. We called these data "deviant." The network was given a combination of nominal, fault, and deviant data, and trained to recognize each type. The extra task of recognizing deviant data forced the network to learn the boundaries of the nominal data.

## 6. Training Method and Initial Results

Our usual method was to train a network on data from several SSME test cases shuffled together with randomly generated "deviant" data, test the network on the training cases, and also test on new cases. In three of the cases there were very low proportions of fault data. Therefore, in order to train the network on a balanced set of samples, the fault samples in those cases were duplicated a hundred times in the training data file before it was shuffled.

When we trained and verified the network on actual fault cases, we found that the network was capable of learning the training data with very high accuracy. It would output "nominal" when fed nominal data, and "fault" when fed fault data. When learning was not quite perfect, the incorrect outputs always occurred for data immediately before or after the fault-declare time. This showed that the transition period around the fault-declare time was the most difficult to learn, as it should be if the network was using criteria involving the continuous progression of PID values through time.

The only case which presented some difficulty was case 249. It is not clear from post-test analysis what fault-declare time is appropriate for this case. Proposed times range from as early as 320 seconds to as late as 405 seconds after start-up. When we used an early declare time and combined case 249 with other cases for training, the net had difficulty reconciling this with the other cases used during training. Apparently, the data in the middle period of 249 is too similar to other data which is nominal, so that it could only be learned as a fault through overlearning, that is, by paying too much attention to distinguishing details with no relevance to fault symptoms.

Our initial results with generalizing to new cases were very promising. The network was able to diagnose new fault cases correctly without training. As expected for these cases, none of the data was evaluated as faulty. Data before the fault-declare time was classified by the network as nominal, and data after the fault-declare time was classified as deviant. The fault-declare times for untrained fault cases determined by the networks have been remarkably consistent with the fault-declare times established on the basis of expert post-test analysis. In case 249, mentioned above, a network (which had been trained on cases 259, 331, 436, and random data) diagnosed the data as deviant after 331 seconds; our proposed fault-declare times ranged between 320 and 405 seconds. The same network, when tested on case 340, output strongly deviant after 283 seconds. Our fault-declare times ranged between 280.3 and 290 seconds.

## 7. Other Failure Detection Systems

A typical tradeoff consideration for failure detection is detection performance versus filter behavior under normal conditions. A design specific to certain failures may provide failure isolation at the expense of performance in detecting nominal data. Certain detection filters take into account such a tradeoff. Under normal or nominal conditions, the bandwidths of the Kalman filters used in detection filters will be increased to be sensitive to the failure isolation designs, yet this increase makes the system more susceptible to sensor noise. With the incorporation of the deviant output, neural nets do not have to be trained to detect specific failures and detection performance will not be hindered under normal conditions. Normal operation should not degrade, since neural nets can be insensitive to sensor noise.

Another failure detection system involves voting schemes. Such schemes can efficiently rule out faulty sensors and are very useful for false alarms, but often pay the price of hardware redundancy for a reliable means of failure detection. Failures such as thermal effects and power failures can also affect the "like" sensors utilized by voting systems in the same way. Since failure detection involves voting between these like sensors, a problem which affects all the sensors will not be detected.

Multiple hypothesis filter-detectors can be too complex for a practical failure detection system [8], [9]. Multiple hypothesis filter-detectors are considered to yield the best performance in the widest class of field for detection, isolation, and estimation, but the complexity can be of major concern. These filters involve the computation of probabilities of all the types of failures under consideration, which may require much time and storage capabilities. Neural nets, on the other hand, are not considered very complex in terms of what the network or implementor has to do. Storage and time considerations are not a problem with neural nets either. When implemented in massive parallelism or by an accelerator board, neural nets are able to respond quickly. Very little computational overhead exists since nets require only two matrix multiplication and two activation applications. The matrices involved in the computation to determine the output are the interconnection matrix between the input and hidden layer and the interconnection matrix between the hidden and output layer. Since only two layers are needed for a successful neural network, only two activation applications are required also. Moreover, neural nets should be able to perform well for SSME fault detection. Some other failure sensitive filters can also become oblivious to new sensor outputs by learning the data too well. In these cases, the Kalman filter and the precomputed covariance utilized become too small and, therefore, oblivious to new data.

Innovations-based detection systems, such as the generalized likelihood ratio (GLR) test, can be sensitive to modeling errors [5], [9]. The GLR test may provide fast failure recovery, but it is imperative for a good estimation of failure parameters that the model is accurate. Neural nets are not considered very complex and the creation of accurate models is not difficult.

The key issues to be addressed in discussing the merits of one system compared to another are complexity in implementation, performance with respect to false alarms and delays in detection, and robustness, such as modeling errors and sensitivity concerns. Our initial results indicate that neural nets do very well in resolving these issues in comparison with other methods.

## 8. Acknowledgement

We gratefully acknowledge Claudia Meyer and Larry Cooper of NASA Lewis, for their assistance in obtaining data and explaining the subtleties of interpreting it, Dr. Chris Bowman of Ball Corporation for helping configure the neural network we used, and James Villareal of NASA Johnson Space Center for his programmatic support.

## Figure 1: SSME (Space Shuttle Main Engine) TEST CASES

### *Six Fault Cases*

*Case 901-331* *July 15, 1981*
*LOX Post Fractures, Erosion-MCC*
*Time 152 - 233.48; Fault-Declare 232.3*
*2010 nominal, 28 fault, 2038 total samples*

*Case 902-249* *September 21, 1981*
*Power Transfer Failure, Turbine Blades*
*Time 261.96 - 450.56; Fault-Declare 320*
*1451 nominal, 3265 fault, 4716 total samples*

*Case 901-340* *October 15, 1981*
*Turn Around Duct Cracked/Torn*
*Time 201.96 - 300; Fault-Declare 280.6*
*1966 nominal, 486 fault, 2452 total samples*

*Case 901-364* *April 7, 1982*
*Hot Gas Intrusion to Rotor Cooling*
*Time 131.96 - 230; Fault-Declare 210*
*1951 nominal, 501 fault, 2452 total samples*

*Case 901-436* *February 14, 1984*
*Coolant Liner Buckle*
*Time 551.96 - 611.08; Fault-Declare 610.55*
*1471 nominal, 8 fault, 1479 total samples*

*Case 750-259* *March 27, 1985*
*MCC Outlet Manifold Neck, Fuel Leak*
*Time 41.96 - 101.50; Fault-Declare 101.3*
*1485 nominal, 4 fault, 1489 total samples*

### *Two Nominal Cases*

*Case 902-457* *November 1988*
*Time 100 - 250*
*3751 nominal samples*

*Case 902-463* *February 1989*
*Time 101.96 - 238.16*
*3405 nominal samples*

## Figure 2:

## PIDs (Parameter ID's) for SSME (Space Shuttle Main Engine)

*18 (566) MCC CLNT DS T*
*Main Combustion Chamber Coolant Discharge Temperature B*

*24 (371) MCC FU INJ PR (MCC HG IN PR)*
*Main Combustion Chamber Hot Gas Injector Pressure A*

*40 OPOV ACT POS*
*Oxidizer-Preburner Oxidizer Valve Actuator Position A*

*42 FPOV ACT POS*
*Fuel Preburner Oxidizer Valve Actuator Position A*

*52 (459) HPFP DS PR*
*High Pressure Fuel Pump Discharge Pressure A*

*63 MCC PC*
*Main Combustion Chamber Pressure Average*

*209 (302) LPOP DS PR*
*High Pressure Oxidizer Pump Inlet Pressure A*

*231 (663) HPFT DS T1 A*
*High Pressure Fuel Turbine Discharge Temperature A*

*232 (664) HPFT DS T1 B*
*High Pressure Fuel Turbine Discharge Temperature B*

*233 HPOT DS T1*
*High Pressure Oxidizer Turbine Discharge Temperature A*

*234 HPOT DS T2*
*High Pressure Oxidizer Turbine Discharge Temperature B*

*261 (764) HPFP SPEED*
*High Pressure Fuel Turbopump Shaft Speed*

*These are all CADS sensor measurements taken 25 times per second. Numbers in parentheses are corresponding facility measurements.*

Figure 3: Features computed for each PID for each sample

(1) $$( \mathrm{AVG2}(t) - \mathrm{AVG1}(t) ) / s$$

(2) $$( \mathrm{AVG2}(t) - \mathrm{AVG2}(t0) ) / s$$

(3) $$( X(t) - \mathrm{AVG1}(t - .08) ) / s$$

*Where,*

*AVG2(t) is the mean value of the PID for the 2 seconds (50 samples) leading up to time t.*

*AVG1(t) is the mean value of the PID for the 0.08 seconds (3 samples) leading up to time t.*

*s is the standard deviation of the PID value.*

*t0 is time soon after SSME reaches steady-state operation.*

*X(t) is the value of the PID at time t.*

*These three features are intended to encode the essential history of each PID value, providing sufficient information for the neural network to perform fault diagnosis. They represent the degree of change (positive or negative) over medium, long, and short periods of time.*

*The time t0 is used to calculate a base average value for each PID, to provide an unchanging reference point for measuring the long-term change in the PID value. We have simply used the first 2 seconds of data in the time-slice used for each test case to compute AVG2(t0).*

*In order to make all of the network inputs fall within the same range, all three features are scaled according to the standard deviation of the PID. The standard deviation does not depend on the particular test case; for each PID, a standard deviation is calculated on the basis of all available test cases combined.*

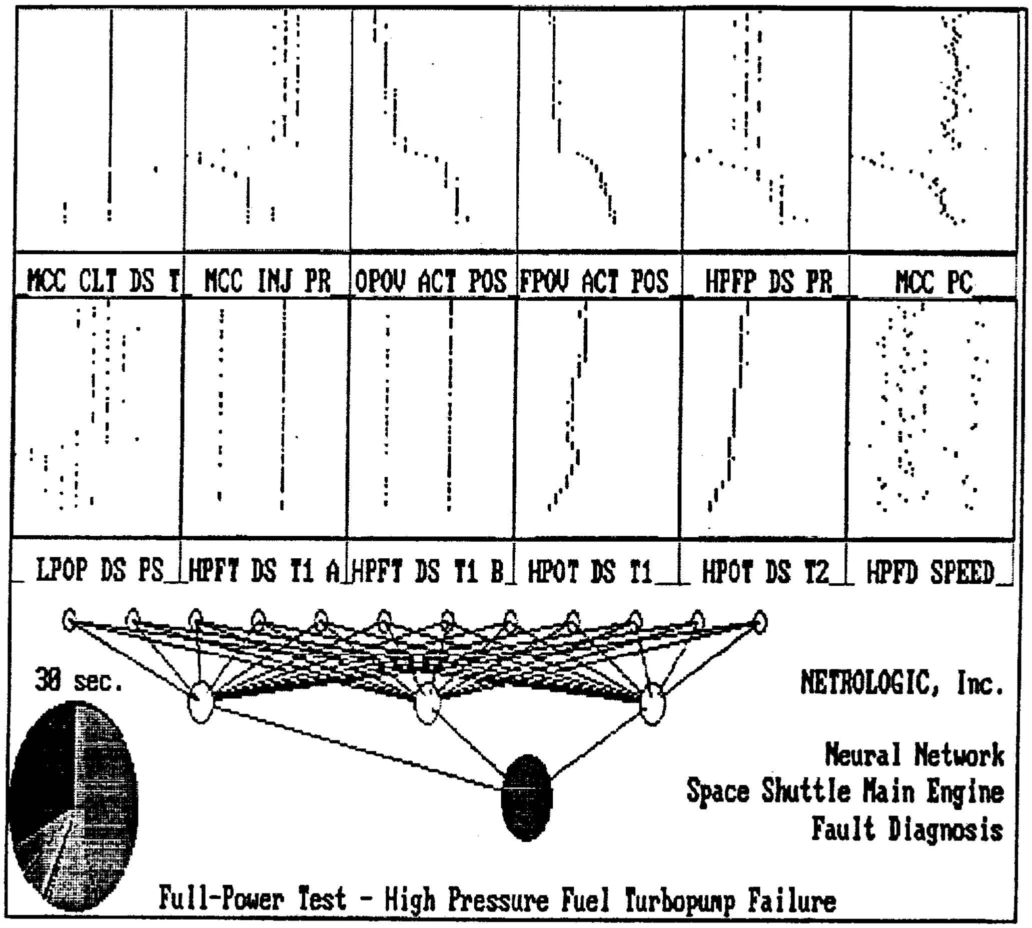


Figure 4: Conceptual Diagram

*This is a computer-screen image of our demonstration program. The windows at the top of the picture are graphs of the twelve PID values varying with time. The schematic diagram conceptually portrays the neural network units and connections. Twelve inputs, three hidden units, and a single output unit are shown (note that our current approach actually employs 36 input, 6 hidden and 3 output units).*

fig. 5

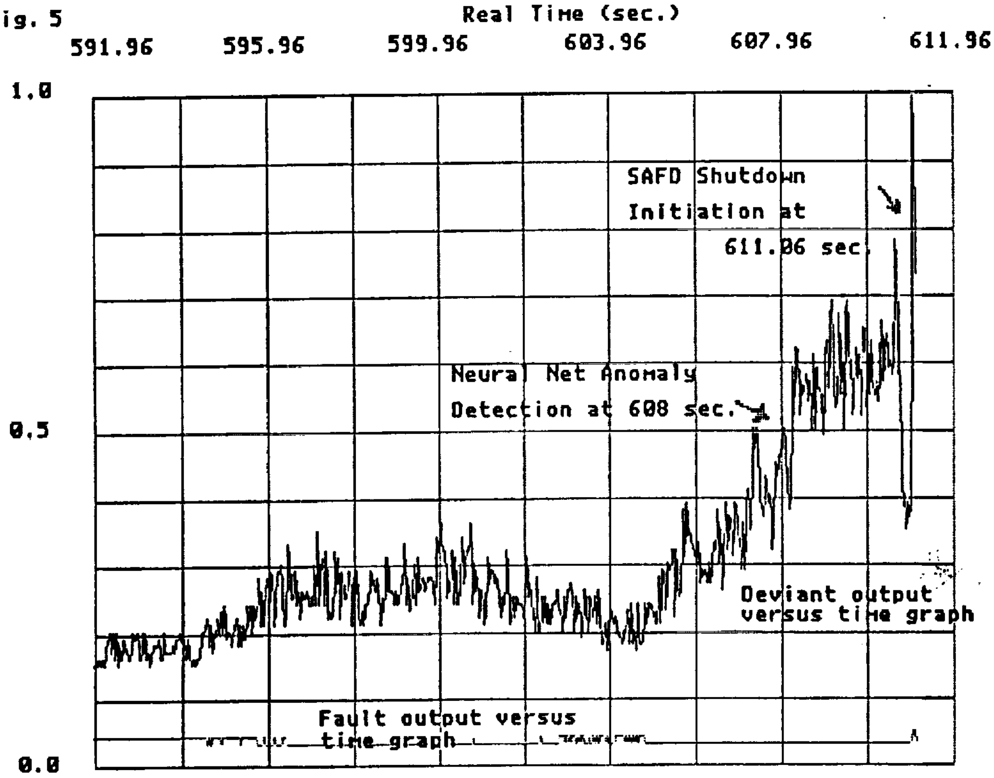


Figure 5: Graph of Neural Network Output

*This shows the results of training the neural net on a case where the primary and secondary faceplates burned causing a problem in the main combustion chamber (901-331), a case where cracks were found in the high pressure fuel turbopump (901-340), and a case where a hotgas intrusion to rotor cooling occurred from a breach in a kaiser helmet (901-364). After training, the network was tested on case 901-436, where the high pressure fuel turbopump was massively damaged. The graph shows that the neural net provided earlier fault detection than that of the SAFD results provided in the "Failure Control Techniques Report For The SSME," by Rocketdyne. The graph of the third output unit, which indicates nominal data, is not shown. The nominal output is simply the reflection of the deviant output around the horizontal axis labelled 0.5.*

## References

[1] Cikanek, Harry A.: "Space Shuttle Main Engine Monitoring Experience And Advanced Monitoring Systems Development."

[2] Cikanek, Harry A.: "SSME Failure Detection." Proceedings of the American Control Conference, June 19-21, 1985, NASA MSFC.

[3] Dietz, Kiech, and Ali: "Neural Network Models Applied to Real-Time Fault Diagnosis." Journal of Neural Network Computing, Vol. 1 No. 1, 1989.

[4] Kelley, Glover, Teeter, and Tischer: "Diagnostic Needs of the Space Shuttle Main Engine." SAE Technical Paper Series, Oct. 15-18, 1984.

[5] Kerr, T.H.: "The Controversy Over Use of SPRT and GLR Techniques and Other Loose-Ends in Failure Detection." American Control Conference, Proceedings, Vol. 3, pp.966-977, 1983.

[6] McClelland, J.L., and Rumelhart, D.E., "Learning Internal Representations by Error Propagation," pp. 318-364, in *Parallel Distributed Processing*, Vol. I., The MIT Press, Cambridge, MA, 1987.

[7] Mehra and Peschon: "An Innovations Approach to Fault Detection and Diagnosis in Dynamic Systems." Automatica, Vol.7, pp. 637-640, 1971.

[8] Taniguchi, M.H.: "Literature Review Results." Failure Control Techniques For The SSME, Rocketdyne, Phase I report. April, 1987.

[9] Willsky, Alan S.: " A Survey of Design Methods For Failure Detection in Dynamic Systems." Automatica, Vol. 12, pp. 601-611, 1976.

# Space Shuttle Main Engine Fault Detection
# Using
# Neural Networks

NETROLOGIC

Neural Networks and Fuzzy Logic Workshop
NASA, Johnson Space Center
April 11-13, 1990

## GOALS AND NATURE OF PROBLEM

- USE TRAINABLE PATTERN CLASSIFIERS FOR SPACE SHUTTLE MAIN ENGINE ANOMALY DETECTION

- PROVIDE EARLIER AND MORE ACCURATE ON-LINE ANOMALY DETECTION (PREVIOUS DETECTION SYSTEMS - REDLINES, HUMAN MONITORING - MISSED EARLY SIGNS OF ENGINE FAILURE)

- IMPROVE TEST STAND MONITORING, EXTEND TO IN-FLIGHT MONITORING

- SHUTDOWN DECISION MODULE MUST INTEGRATE AND EVALUATE LARGE NUMBER OF SIMULTANEOUS SENSOR MEASUREMENTS AT HIGH RATE

- HIGH PENALTY FOR

  - FAILURE TO DETECT IMPENDING CATASTROPHE (TEST-STAND DAMAGE AS HIGH AS $26 MILLION FOR A SINGLE FAILURE; FAILURE IN FLIGHT, IF IT EVER OCCURS, MAY CAUSE LOSS OF HUMAN LIFE)

  - UNNECESSARY SHUT-DOWN (FALSE ALARM) (COSTS THOUSANDS OF DOLLARS ON TEST STAND; IN FLIGHT, EMERGENCY LANDING WITH ENGINE SHUT DOWN UNNECESSARILY MAY ENDANGER LIFE)

- AS SHUTTLE ENGINE FIRING IN PROGRESS, "RAW" INPUT TO ANOMALY DETECTION SYSTEM IS SEQUENCE OF VECTORS

  $$P(\tau_i) \quad i = 0, 1, \ldots, s\text{-}1$$

  ($s$ = # SAMPLES TAKEN SO FAR)

- TIME STARTS FROM LAUNCH: $\tau_0 = 0$

- SAMPLES TAKEN AT REGULAR RATE

  (TYPICAL SAMPLING RATE 25 PER SECOND,
  OR ONE SAMPLE EVERY 0.04 SECONDS)

- FOR EACH POINT IN TIME $\tau$, EACH COMPONENT OF $P(\tau)$ IS THE VALUE OF A PARTICULAR SENSOR MEASUREMENT

  $$P(\tau) = (P_1(\tau), P_2(\tau), \ldots, P_N(\tau))$$

  ($N$ = # SENSORS EMPLOYED)

- SENSORS $P_1, P_2, \ldots, P_N$ REFERRED TO BY PARAMETER IDENTIFICATION NUMBERS, OR "PIDS"

- OVER 200 PIDS AVAILABLE
- TEST FIRING DATA NOT CONSISTENT:
  FOR MOST TEST FIRINGS, SOME PIDS NOT PRESENT OR NOT VALID
  (SENSORS NOT BUILT INTO EARLY VERSIONS OF ENGINES OR FAILED SENSORS)
- CRITERIA FOR INITIAL CHOICE OF PIDS
  - SUBSET OF PIDS USED IN ROCKETDYNE'S SAFD ALOGORITHM
  - SIGNIFICANT FOR DIAGNOSIS IN ANOMALOUS FIRINGS UNDER INVESTIGATION
  - AVAILABLE FOR MOST TEST FIRINGS UNDER INVESTIGATION
    (DESIRABLE FOR GENERALIZING FROM ONE FIRING TO ANOTHER, BUT NOT ABSOLUTE REQUIREMENT - MISSING OR FAILED SENSORS MUST BE TAKEN INTO ACCOUNT ANYWAY)
- METHOD ALLOWS FOR USING MORE PIDS IN FUTURE

TWELVE PIDS USED IN CURRENT STUDY

$P_{18}$ = MCC CLNT DS T
(Main Combustion Chamber Coolant Discharge Temperature B)

$P_{24}$ = MCC FU INJ PR
(Main Combustion Chamber Hot Gas Injector Pressure A)

$P_{40}$ = OPOV ACT POS
(Oxidizer-Preburner Oxidizer Valve Actuator Position A)

$P_{42}$ = FPOV ACT POS
(Fuel Preburner Oxidizer Valve Actuator Position A)

$P_{52}$ = HPFP DS PR
(High Pressure Fuel Pump Discharge Pressure A)

$P_{63}$ = MCC PC
(Main Combustion Chamber Pressure Average)

$P_{209}$ = LPOP DS PR
(High Pressure Oxidizer Pump Inlet Pressure A)

$P_{231}$ = HPFT DS T1 A
(High Pressure Fuel Turbine Discharge Temperature A)

$P_{232}$ = HPFT DS T1 B
(High Pressure Fuel Turbine Discharge Temperature B)

$P_{233}$ = HPOT DS T1
(High Pressure Oxidizer Turbine Discharge Temperature A)

$P_{234}$ = HPOT DS T2
(High Pressure Oxidizer Turbine Discharge Temperature B)

$P_{261}$ = HPFP SPEED
(High Pressure Fuel Turbopump Shaft Speed)

- TEST FIRING MAY LAST OVER TEN MINUTES, SO NUMBER OF SAMPLES s MAY REACH TENS OF THOUSANDS

- VECTORS $P(\tau_i)$, i = 0, 2, ..., s-1 FORM s x N MATRIX (N = # PIDS)

- THIS POTENTIALLY HUGE MATRIX MUST BE EVALUATED QUICKLY (PREFERABLY BEFORE NEXT SAMPLE) PROVIDING STRONG MOTIVATION FOR EXTRACTING MANAGEABLE (AND CONSTANT) NUMBER OF FEATURES FROM MATRIX, USING FAST CLASSIFICATION ALGORITHMS AND MACHINERY, ESPECIALLY PARALLEL PROCESSING

- IDEALLY, SSME PERFECTLY UNDERSTOOD, HEALTH STATUS DETERMINED FROM SENSOR MEASUREMENTS BY APPLICATION OF THEORETICALLY DEDUCED RULES

- BUT SSME IS COMPLICATED, ITS BEHAVIOR NOT ENTIRELY PREDICTABLE

- MAIN RESOURCES FOR CREATING DIAGNOSTIC SYSTEM ARE

  - EXPERT KNOWLEDGE (MUCH OF THIS IN FAILURE INVESTIGATION SUMMARIES)

  - DATA ACCUMULATED FROM PREVIOUS NOMINAL & ANOMALOUS SSME FIRINGS

- USE TRAINABLE PATTERN CLASSIFICATION SOFTWARE TO LEARN TO CLASSIFY TRAINING DATA, ATTEMPT TO GENERALIZE CORRECTLY TO NOVEL DATA

- NEURAL NETWORKS OFFER

  - SPEED, ESPECIALLY IF IMPLEMENTED ON PARALLEL HARDWARE

  - AUTOMATIC LEARNING OF SUBTLE FEATURES IN LARGE QUANTITIES OF DATA

  - CAPABILITY OF GENERALIZING BASED ON PREVIOUSLY LEARNED EXAMPLES

## SSME TEST FIRING DATA EMPLOYED FOR CLASSIFIER TRAINING AND TESTING

(FIRINGS CONDUCTED ON GROUND BETWEEN 1981 AND 1989)

- TWO NOMINAL FIRINGS (902-457, 902-463)
- SIX ANOMALOUS FIRINGS REPRESENTING VARIOUS FAILURE TYPES
  - (901-331) CRACKED LIQUID OXYGEN POST
  - (902-249) POWER TRANSFER FAILURE, TURBINE BLADES
  - (901-340) TURN AROUND DUCT CRACKED/TORN
  - (901-364) HOT GAS INTRUSION TO ROTOR COOLING
  - (901-436) HIGH PRESSURE FUEL TURBOPUMP COOLANT LINER BUCKLE
  - (750-259) FUEL LEAK IN MAIN COMBUSTION CHAMBER OUTLET NECK

(MORE TEST FIRINGS TO BE ADDED IN FUTURE)

FAULT-DECLARE TIMES BASED ON FAILURE INVESTIGATION REPORTS FOR EACH FIRING, PLUS AS OUR OWN ANALYSIS OF SENSOR DATA

- FAULT-DECLARE TIME IS TIME WHEN SENSORS FIRST SHOW SYMPTOMS OF FAULTY ENGINE PERFORMANCE, SO THAT AN ANOMALY DETECTION SYSTEM IDEALLY SHOULD HAVE BEEN ABLE TO INITIATE SSME SHUT-DOWN

- FOR NETWORK TRAINING, SENSOR SAMPLES TAKEN BEFORE FAULT-DECLARE TIME CONSIDERED NOMINAL DATA, SAMPLES TAKEN AFTER THAT TIME CONSIDERED ANOMALOUS DATA (HOWEVER SOME SAMPLES MAY BE LEFT OUT OF THE TRAINING SET IF IN DOUBT WHETHER TO CONSIDER ANOMALOUS)

- WHEN TESTING NETWORK PERFORMANCE, FAULT-DECLARE TIMES USED FOR COMPARISON

## ATTENTION INITIALLY RESTRICTED TO PERIODS OF STEADY-STATE OPERATION

EXPLANATION FOR NON-ROCKET EXPERTS: THE SSME OPERATES AT VARIOUS POWER (THRUST) LEVELS, MEASURED BY THE MAIN COMBUSTION CHAMBER PRESSURE, $P_{63}$. NORMALLY A FIRING HAS A SCHEDULED SEQUENCE OF POWER LEVELS. PERIODS DURING WHICH THE POWER LEVEL IS HELD APPROXIMATELY CONSTANT ARE CALLED "STEADY-STATE", AND MAY LAST A FEW SECONDS OR A FEW MINUTES. IN BETWEEN THE STEADY-STATE PERIODS ARE INTERVALS OF THROTTLING, KNOWN AS "TRANSIENTS". TRANSIENTS USUALLY LAST ONLY A FEW SECONDS.

- MOST MAJOR FAILURES OCCURRED DURING STEADY-STATE
- TAILORING METHOD TO STEADY-STATE DATA ALLOWS USEFUL ASSUMPTIONS:
  - SENSOR VALUES NOT EXPECTED TO CHANGE SIGNIFICANTLY (ALTHOUGH IN PRACTICE THEY CHANGE CONSIDERABLY)
  - UNCHANGING VALUES CAN BE CONSIDERED NOMINAL
  - SAME CRITERIA FOR ENGINE HEALTH SHOULD APPLY REGARDLESS OF AMOUNT OF TIME ELAPSED IN STEADY-STATE PERIOD
- TRANSIENT ANOMALY DETECTION INHERENTLY MORE DIFFICULT:
  - SENSOR DATA CHANGE IN COMPLICATED WAYS
  - PATTERNS OF CHANGE MAY DEPEND ON EXACT NATURE OF TRANSIENT (START & FINISH POWER LEVELS, RATE OF THROTTLING, ETC)
- NOT APPROPRIATE TO GENERALIZE ACCROSS SAMPLES TAKEN AT DIFFERENT TIMES DURING TRANSIENTS
- IN FUTURE, MOST TECHNIQUES WE EMPLOY FOR STEADY-STATE COULD BE EXTENDED TO APPLY TO TRANSIENT ANOMALY DETECTION (RECURRENT NEURAL NETWORKS PARTICULARLY PROMISING)

- AT EACH TIME $T_i$, MOST RECENT SAMPLE $P(T_i)$ IS KEY DATA FOR DIAGNOSIS

- SAMPLES $P(T_j)$, $j < i$, ALSO PROVIDE IMPORTANT INFORMATION

  - FOR DETECTING SIGNIFICANT CHANGES OR RECOGNIZABLE "FAULT SIGNATURES" IN THE GRAPHS OF PID VALUES AS FUNCTIONS OF TIME

  - FOR MEASURING DURATIONS OR COUNTING REPETITIONS OF POSSIBLY ANOMALOUS CONDITIONS

  - FOR COMPUTING MOVING AVERAGES, TO SMOOTH OUT "NOISE"

  - IN ORDER TO CONSTRUCT AN ANOMALY DETECTION SYSTEM WHICH IS GENERAL ENOUGH TO WORK ON VARIOUS ENGINES AT VARIOUS POWER LEVELS, IT MAY BE DESIRABLE TO USE DATA FROM ONE TIME INTERVAL IN A GIVEN FIRING AS A POINT OF REFERENCE FOR EVALUATING DATA FROM LATER TIME INTERVALS IN THE SAME FIRING

- PRE-PROCESSING OF PID VALUES: CALCULATION OF FEATURES

  - CONSOLIDATE RAW DATA FROM HUGE s x N MATRIX

    (s = # SAMPLES $P(T_i)$, $i = 0, \ldots, s-1$)

    (N = # PIDS IN EACH SAMPLE)

  - ENCODE ESSENTIAL TIME INFORMATION

  - COMPOUND FEATURES MAY ALSO BE FORMED FROM PIDS BY CALCULATING DIFFERENCES BETWEEN PIDS, AVERAGES OF PIDS, SPECIAL FORMULAS TO COMBINE REDUNDANT PIDS, ETC
    (SOME OF THE PIDS ARE IN FACT ALREADY COMBINATIONS OF THIS TYPE, BUT WE HAVE NOT CREATED ANY NEW FEATURES IN THIS WAY)

  - SCALE AND TRANSLATE FEATURES SO

    - ALL CENTERED AROUND SAME VALUE (E.G. ZERO)

    - ALL VARY WITHIN SAME APPROXIMATE RANGE (E.G. BY SCALING ACCORDING TO STANDARD DEVIATIONS)

WE CURRENTLY CALCULATE TWO FEATURES FOR EACH PID

- RECENT CHANGE

$$\frac{Avg1(t)-Avg2(t)}{\sigma}$$

- LONG-TERM SMOOTHED CHANGE

$$\frac{Avg2(t)-Avg2(t_s)}{\sigma}$$ doesn't really use short σ

WHERE

$Avg1(t)$ = MEAN PID VALUE FOR 0.12 SECONDS (3 SAMPLES)

$Avg2(t)$ = MEAN PID VALUE FOR 2 SECONDS (50 SAMPLES)

(AVERAGES CALCULATED OVER TIME INTERVAL ENDING AT TIME $t$ )

$\sigma$ = STANDARD DEVIATION OF PID VALUE

(MEASURED OVER ALL STEADY-STATE DATA FROM ALL AVAILABLE FIRINGS)

$t_s$ = TIME 3 SECONDS AFTER START OF CURRENT STEADY-STATE INTERVAL

- THESE FEATURES RESEMBLE CALCULATIONS USED IN ROCKETDYNE'S SAFD ALGORITHM

- RESULT OF PRE-PROCESSING IS D-DIMENSIONAL FEATURE VECTOR

  $$X(\tau_i) = (X_1(\tau_i), X_2(\tau_i), \ldots, X_d(\tau_i))$$

  WHICH IS FUNCTION OF PID SAMPLES $P(\tau_i)$, $i = 0, 1, \ldots, s$

- FEATURE VECTORS X HAVE FOLLOWING PROPERTY: THE ORIGIN OF D-DIMENSIONAL FEATURE SPACE

  $$0 = (0, 0, \ldots, 0)$$

  WHERE ALL D FEATURES ARE ZERO, IS "MOST NOMINAL" OF ALL POSSIBLE SAMPLES, SINCE IT INDICATES ALL SENSORS REMAINING AT CONSTANT LEVEL DURING STEADY-STATE OPERATION

- NON-ZERO VALUES OF FEATURES INDICATE DEVIATIONS FROM CONSTANT VALUE

- TWELVE PIDS, WITH TWO FEATURES EACH, YIELD TWENTY-FOUR INPUTS TO PATTERN CLASSIFICATION SOFTWARE

NEURAL NETWORK ARCHITECTURE:

THREE LAYER FEEDFORWARD NETWORK TRAINED BY BACKPROPAGATION

- BIOLOGICAL ANALOGY: UNIT = NEURON, CONNECTION = SYNAPSE
- LAYER OF INPUT UNITS
  (ONE FOR EACH FEATURE = 24 INPUT UNITS IN CURRENT MODEL)
- LAYER OF HIDDEN UNITS
  (8 - 12 UNITS IN A SINGLE LAYER FOUND TO BE SUFFICIENT SO FAR)
- LAYER OF OUTPUT UNITS
  (ONE FOR NOMINAL-VS-ANOMALOUS DIAGNOSIS, OTHERS FOR FAULT TYPING)
- EACH INPUT UNIT CONNECTS TO EACH HIDDEN UNIT, AND EACH HIDDEN UNIT CONNECTS TO EACH OUTPUT UNIT
- CONNECTIONS BETWEEN UNITS CHARACTERIZED BY WEIGHTS
  (CONNECTION STRENGTHS): EXCITATORY OR INHIBITORY
- CAPABLE OF PERFORMING ANY MAPPING FROM INPUTS TO OUTPUTS
- TRAINING ACCOMPLISHED BY BACKPROPAGATION OF ERROR
  (WEIGHTS CHANGED AFTER EACH TRAINING PASS ACCORDING TO GENERALIZED DELTA RULE)

- NOTE: CHOICE OF HOW MANY HIDDEN UNITS DETERMINED BY
  - NOT ENOUGH HIDDEN UNITS: IMPOSSIBLE FOR NETWORK TO PERFORM DESIRED MAPPING ON TRAINING DATA
  - TOO MANY HIDDEN UNITS: NETWORK MAY OVER-SPECIALIZE ON IDIOSYNCRACIES OF TRAINING DATA, FAILING TO FIND MORE GENERAL FEATURES DISTINGUISHING DATA CATEGORIES

NETWORK OUTPUT = CLASSIFICATION OF INPUT DATA

- SINCE FIRST PRIORITY OF DIAGNOSTIC SYSTEM IS SHUT-DOWN DECISION MAKING, ESSENTIAL CLASSIFIER OUTPUT HAS ONLY TWO VALUES:
    - ANOMALOUS (RECOMMEND SHUTTING DOWN ENGINE) OR
    - NOMINAL (RECOMMEND PROCEEDING AS USUAL)
- MORE COMPLEX FORMS OF EVALUATION MAY PROVIDE
    - DESCRIPTION OF ANOMALY, WHETHER OF KNOWN FAILURE TYPE
    - WHICH ENGINE PARTS ARE INVOLVED
    - ESTIMATE OF SEVERITY
    - SIMILARITY TO DATA FROM PREVIOUS FAILURES
    - DEGREE OF CONFIDENCE IN DIAGNOSIS
- ANOMALY DETECTION VS. FAULT TYPING
    - FAULT TYPING REQUIRED IF SHUT-DOWN PROCEDURES DEPEND ON FAILURE TYPE, OR NETWORK FORMS PART OF LARGER DIAGNOSTIC SYSTEM (WHICH CALLS FOR MORE SPECIFIC DIAGNOSIS BY NETWORK)
    - WE HAVE EXPERIMENTED WITH FAULT-TYPING, TREATING EACH ANOMALOUS TEST FIRING IN TRAINING SET AS REPRESENTING ONE FAULT TYPE
- CURRENT NETWORK CONFIGURATION HAS
    - AN OUTPUT UNIT TRAINED TO FIRE LOW IF NOMINAL AND HIGH IF ANOMALOUS
    - ADDITIONAL OUTPUT UNITS FOR EACH FAULT TYPE (I.E., ONE FOR EACH ANOMALOUS TEST FIRING IN TRAINING SET)
    - THUS WHEN TRAINING ON DATA INCLUDING FIVE ANOMALOUS FIRINGS, WE EMPLOY SIX OUTPUT UNITS IN FEEDFORWARD NETWORK

AVAILABLE NOMINAL AND ANOMALOUS DATA CURRENTLY VERY LIMITED

- ONLY A HANDFUL OF TEST FIRINGS TO USE FOR TRAINING
  (MORE NOMINAL DATA CAN EVENTUALLY BE OBTAINED FROM NASA, BUT ANOMALOUS FIRINGS ARE RARE -- FORTUNATELY!)

- EACH FIRING PROVIDES MANY DATA SAMPLES. HOWEVER SAMPLES FROM A GIVEN FIRING TEND TO LIE ON A TRAJECTORY, EACH SAMPLE BEING CLOSE TO PREVIOUS SAMPLE

- IMPOSSIBLE FOR THIS LIMITED QUANTITY OF DATA TO COME CLOSE TO SPANNING ENTIRE 24-DIMENSIONAL POTENTIAL INPUT SPACE
  (IN 24-DIMENSIONAL SPACE MOST POINTS ARE VERY FAR APART. THE NUMBER OF QUADRANTS IN 24-SPACE IS $2^{24}$ = 16,777,216)

- GENERALIZATION TO NEW DATA REQUIRES BOTH INTERPOLATION AND EXTRAPOLATION

  - COMPLETE DECISION BOUNDARY BETWEEN NOMINAL AND ANOMALOUS REGIONS CANNOT BE UNIQUELY DETERMINED FROM ANY FINITE AMOUNT OF TRAINING DATA

  - NETWORK MUST BE TRAINED APPROPRIATE RESPONSE TO UNPRECEDENTED INPUT DATA

  - UNLESS NEW ANOMALOUS FIRING VERY SIMILAR TO ONE OF TRAINING FIRINGS, NEW ANOMALOUS DATA WILL NOT RESEMBLE OLD ANOMALOUS DATA ANY MORE THAN IT RESEMBLES OLD NOMINAL DATA

  - NEED TO MAKE ASSUMPTIONS ABOUT SHAPE OF NOMINAL REGION TO BE MAPPED OUT BY ANOMALY DETECTION SYSTEM, IMPOSE THESE ASSUMPTIONS ON TRAINABLE CLASSIFIER

  - A BASIC ASSUMPTION WILL LEAD TO DETECTION OF NEW FAULT TYPES: ANY NEW DATA SUFFICIENTLY DIFFERENT FROM ALL PREVIOUSLY ENCOUNTERED NOMINAL DATA TO BE CONSIDERED ANOMALOUS

  - TO FORCE FEEDFORWARD NEURAL NETWORK TO CATEGORIZE NEW DATA IN ACCORDANCE THIS ASSUMPTION, IT HAS BEEN FOUND ADVANTAGEOUS TO ADD IMITATION NOMINAL AND ANOMALOUS TRAINING DATA TO TRAINING DATA FROM ACTUAL SSME FIRINGS

- GENERATE IMITATION DATA RANDOMLY DISTRIBUTED THROUGHOUT SUITABLE PART OF INPUT SPACE
    - IMITATION ANOMALOUS DATA EITHER RANDOMLY DISTRIBUTED (WHICH PLACES IT GENERALLY FAR OUT IN INPUT SPACE) OR WITH VALUES OF SOME COMPONENTS NEAR KNOWN FAULT READINGS)
    - IMITATION NOMINAL DATA WITHIN EXPECTED RANGES OF NOMINAL FEATURES (CURRENTLY LIMITED EXPERIENCE WITH ADDING GENERATED NOMINAL DATA)
- COMBINE RANDOM DATA WITH GENUINE NOMINAL AND ANOMALOUS DATA FOR TRAINING
- TRAIN NETWORK TO CATEGORIZE GENERATED ANOMALOUS DATA AS ANOMALOUS, GENERATED NOMINAL DATA AS NOMINAL
    - TASK OF RECOGNIZING GENERATED DATA FORCES NETWORK TO LEARN BOUNDARIES OF EXPECTED NOMINAL REGION

SOME FINDINGS AT INTERMEDIATE STAGE IN OUR RESEARCH

- NEURAL NETWORK CLASSIFIER IS ALWAYS CAPABLE OF LEARNING TRAINING DATA WITH VIRTUALLY 100% ACCURACY, OUTPUTTING "NOMINAL" WHEN FED NOMINAL DATA, AND "ANOMALOUS" WHEN FED ANOMALOUS DATA

- GENERALIZING TO NEW (UN-TRAINED) ANOMALOUS FIRINGS HAS BEEN SYSTEMATICALLY UNDERTAKEN ACCORDING TO SINGLE HOLD-OUT PRINCIPLE:

    - TRAIN NETWORK ON ALL TRAINING DATA (TO INCLUDE GENUINE DATA FROM NOMINAL AND ANOMALOUS TEST FIRINGS AS WELL AS SOME IMITATION ANOMALOUS DATA), EXCEPT FOR DATA FROM ONE TEST FIRING DELIBERATELY WITHELD

    - TEST SAME NETWORK ON DATA FROM FIRING WHICH WAS WITHELD FROM TRAINING

- NETWORK HAS DEMONSTRATED ABILITY TO CORRECTLY CLASSIFY THIS DATA THAT IS NEW TO IT AS NOMINAL UP UNTIL FAULT-DECLARE TIME, AND ANOMALOUS THEREAFTER

- POSITIVE RESULT OF GENERALIZATION IS CONTINGENT ON TRAINING WITH RANDOM IMITATION ANOMALOUS DATA (OTHERWISE NEW DATA IS ALWAYS CLASSIFIED AS NOMINAL)

- FAULT-TYPING (ACTIVATIONS OF ADDITIONAL OUTPUT UNITS) IS LEARNED CORRECTLY FOR TRAINING DATA, BUT NEW DATA IS NEVER CLASSIFIED AS BELONGING TO ANY PREVIOUS FAULT-TYPE

- NOW WHEN GENERALIZATION IS NOT SUCCESSFUL, CHIEF PROBLEM IS FALSE ALARMS (CLASSIFICATION OF NEW NOMINAL DATA AS ANOMALOUS)

- AN APPROACH HAS BEEN FOUND FOR RECOGNIZING WHEN A FALSE-ALARM IS DEPENDENT ON FEATURES CORRESPONDING TO SINGLE PID, AND IMMEDIATELY DETERMINING WHICH PID IS RESPONSIBLE:

    - MULTIPLE COPIES (ONE FOR EACH PID) OF EACH FEATURE VECTOR ARE SEPERATELY FED THROUGH NETWORK

    - EACH COPY IS ALTERED BY HAVING FEATURES CORRESPONDING TO ONE OF PIDS REPLACED WITH ZEROS (REMEMBER THAT FOR CURRENT FEATURES, ZERO MEANS NO-CHANGE, AND NON-ZERO INDICATES DEVIATION FROM CONSTANT STEADY-STATE VALUE)

    - NETWORK OUTPUTS FOR EACH COPY SHOW WHAT CLASSIFICATIONS WOULD BE IF EACH PID IN TURN INDICATED NO CHANGE

    - ZEROING OUT PID RESPONSIBLE FOR FALSE ALARM RESULTS IN CORRECT CLASSIFICATION AS NOMINAL UP UNTIL FAULT-DECLARE TIME, AND ANOMALOUS THEREAFTER

    - SUCH RESULTS SUGGEST POSSIBILITY OF INCORPORATING VOTING SCHEME INTO MAKING CLASSIFIER OUTPUT MORE ROBUST WITH RESPECT TO FALSE ALARMS CAUSED BY ANY SINGLE FEATURE, IF IT IS FOUND APPROPRIATE TO REQUIRE MORE THAN ONE PID TO MANIFEST SYMPTOMS BEFORE MAKING AN ANOMALOUS CLASSIFICATION, OR SIMPLY AS AID TO ISOLATING POSSIBLE SENSOR FAILURES

WORK IN PROGRESS

- FURTHER TRAINING AND TESTING OF FEEDFORWARD NEURAL NETWORKS, EMPLOYING SEVERAL NEW KINDS OF SIMULATED OR MODIFIED SUPPLEMENTARY TRAINING DATA:

  - GENERATE SIMULATED / MODIFIED DATA DYNAMICALLY DURING TRAINING, RATHER THAN PUTTING INTO TRAINING DATA FILE AND USING REPEATEDLY (MUCH MORE EVEN COVERAGE OF FEATURE SPACE)

  - RESTRICT RANDOM SIMULATED ANOMALOUS DATA TO STAY OUTSIDE OF REGIONS ASSUMED TO BE NOMINAL (REQUIRE MINIMUM LENGTH FOR ANOMALOUS FEATURE VECTORS, ETC --- MAY DECREASE FALSE-ALARMS)

  - USE RANDOMLY GENERATED NOMINAL DATA CLOSE TO ORIGIN (JUSTIFICATION: NO GENUINE ANOMALOUS FEATURE VECTORS HAVE BEEN OBSERVED WITHIN A CERTAIN RADIUS OF ORIGIN, BUT FALSE ALARMS HAVE OCCURRED THERE)

  - MODIFY GENUINE NOMINAL FEATURE VECTORS BY REPLACING SOME COMPONENTS WITH ZERO VALUES (TO PREVENT FALSE ALARMS DUE TO MISSING SENSORS, AND TO FILL OUT NOMINAL REGION IN ACCORDANCE WITH ASSUMPTION THAT IN STEADY-STATE CONTEXT, UNCHANGING SENSOR VALUE SHOULD NOT CAUSE FEATURE VECTOR TO BE REGARDED AS ANOMALOUS)

  - MODIFY GENUINE ANOMALOUS FEATURE VECTORS IN SAME WAY (TO MAKE ANOMALY DETECTION MORE ROBUST, NOT DEPENDENT ON ANY SINGLE PID, TO GUARANTEE DETECTION EVEN USING TESTING METHOD SUGGESTED ABOVE IN WHICH APPARENT ANOMALY DUE TO ONLY ONE PID MAY NOT BE ENOUGH TO WARRANT ENGINE SHUT-DOWN)

- EXPERIMENTING WITH VARIATIONS IN TRAINING TECHNIQUE AND NETWORK ARCHITECTURE, ESPECIALLY RECURRENT NETWORKS:
    - RECURRENT NETWORKS DESIGNED TO CLASSIFY TIME SERIES DATA
    - ACTIVATIONS OF HIDDEN UNITS FEED BACK TO RETAIN MEMORY FOR CLASSIFYING SUBSEQUENT INPUTS IN TIME SERIES CONTEXT
    - AUTOMATICALLY LEARNED INTERNAL FEATURES OF RECURRENT NETS MAY BE USEFUL ADDITION OR ALTERNATIVE TO OUR EXPLICITLY COMPUTED CHANGE-MEASURING FEATURES

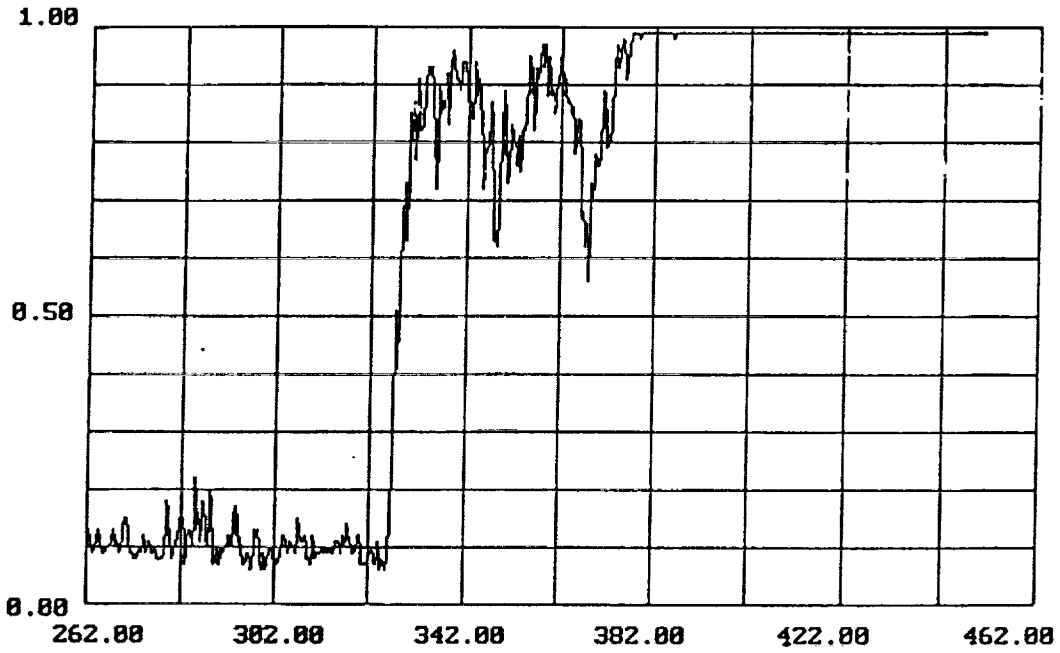
- USING SOME GEOMETRICAL PERSPECTIVES ON THE PROBLEM, EXPERIMENTING WITH PLAUSIBLE ALTERNATIVE METHODS FOR EXTRAPOLATING FROM TRAINING DATA TO DETERMINE BOUNDARIES OF NOMINAL REGION IN 24-DIMENSIONAL VECTOR SPACE:

  - LENGTHS OF FEATURE VECTORS (I.E. DISTANCE FROM ORIGIN) FOUND TO BE GOOD INDICATORS OF TRANSITIONS FROM NOMINAL TO ANOMALOUS DATA

  - NOMINAL REGION COULD BE CHARACTERIZED BY ESTABLISHING MAXIMUM LENGTH FOR NOMINAL FEATURE VECTORS IN ANY GIVEN DIRECTION

  - DETERMINE THESE MAXIMUM LENGTHS FOR TRAINING DATA, GENERALIZE TO NOVEL DATA BY VARIATION ON NEAREST NEIGHBOR PRINCIPLE, DEFINING NEARNESS ACCORDING TO ANGLES BETWEEN VECTORS

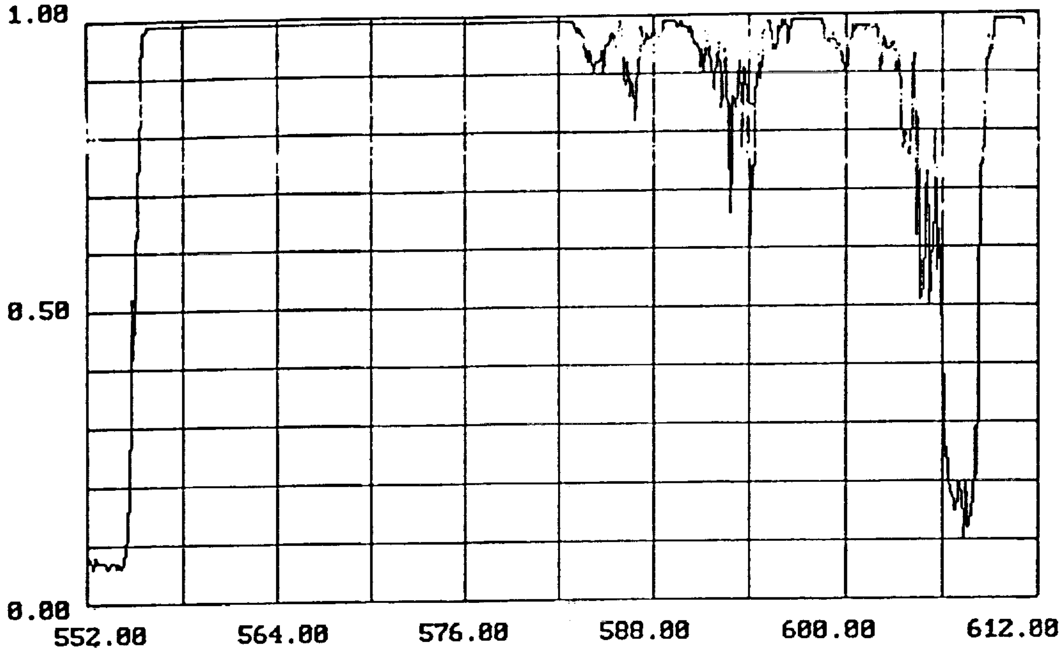
  - INITIAL IMPLEMENTION OF THIS APPROACH USES SEQUENTIAL ALGORITHMS, COULD BE IMPLEMENTED IN PARALLEL (ALONG SIMILAR LINES AS THE PROBABILISTIC NEURAL NETWORK, WHICH ALSO RESEMBLES NEAREST NEIGHBOR CLASSIFIER)



Graph of Neural Network Output for Novel Data

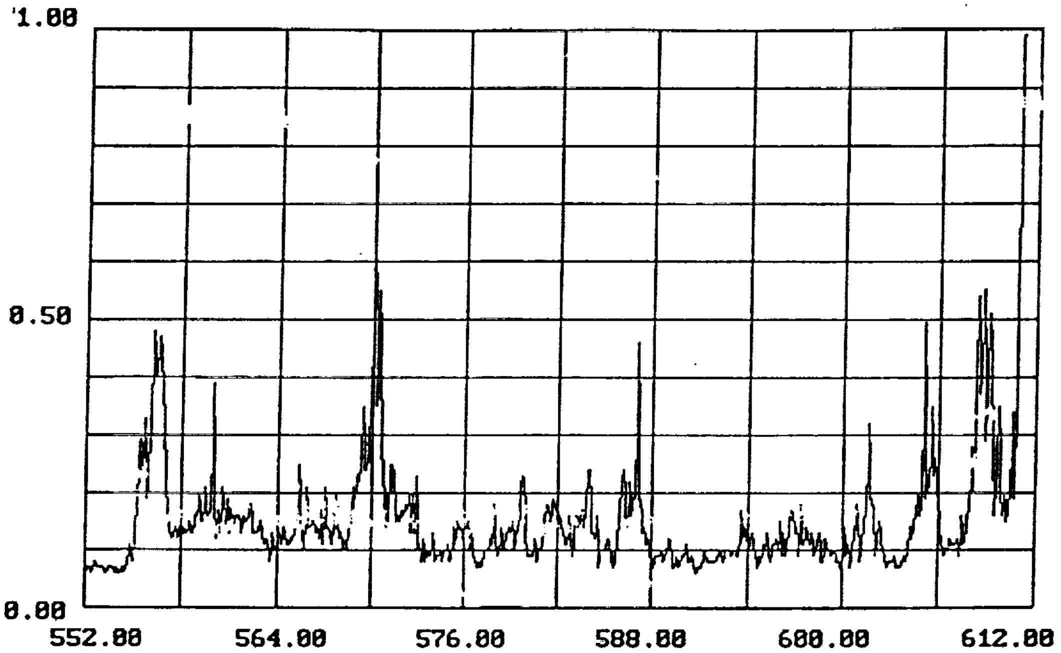
A neural network was trained on data from all test firings except 901-249, plus randomly generated anomalous data. The graph shows the activation of the nominal-versus-anomalous output unit when the network was tested on firing 901-249.

The network clearly begins to detect an anomaly around 328 seconds, a few seconds after symptoms began to occur according to Failure Investigation Summary. The SSME was not actually shut down until 450.58 seconds, after massive damage had occurred.



Example of a "False-Alarm" in Generalization to Novel Data

Network was trained by holding out only the anomalous firing 901-436, and tested on that firing. The actual fault did not occur until 610 seconds, and early warning as early as shown on this graph does not appear to be realistic. Therefore this must be regarded as a false alarm.



Result of zeroing out features for PID 24
in the same "False Alarm" case

The time at which the graph of the "deviant" output unit finally goes above .6 is now precisely the fault-declare time determined by analysis for the novel anomalous firing 901-436. (PID 24, and the two features calculated were in fact out of range for the training firings.)