JPL NEURAL NETWORK WORKSHOP
"A DECADE OF NEURAL NETWORKS"
MAY 1994

NEURAL NETWORK APPLICATION TO COMPREHENSIVE ENGINE DIAGNOSTICS

Kenneth A. Marko
Ford Motor Company
Dearborn, Michigan 48121-2054

I. INTRODUCTION

We have previously reported on the use neural networks for detection and identification of faults in complex microprocessor controlled powertrain systems [1,2]. The data analyzed in those studies consisted of the full spectrum of signals passing between the engine and the real-time microprocessor controller. The specific task of the classification system was to classify system operation as nominal or abnormal and to identify the fault present. The primary concern in earlier work was the identification of faults, in sensors or actuators in the powertrain system as it was exercised over its full operating range. The use of data from a variety of sources, each contributing some potentially useful information to the classification task, is commonly referred to as sensor fusion and typifies the type of problems successfully addressed using neural networks.

In this work, we explore the application of neural networks to a different diagnostic problem, the diagnosis of faults in newly manufactured engines and the utility of neural networks for process control. While this problem shares a number of characteristics of the previous studies, there are several significant differences.

- Our interest here is primarily on mechanical faults rather than electronic faults since the engine at this stage in the manufacturing process is undergoing "cold test", i.e. it is connected to an electric dynamometer.
- Engines operate only briefly over a restricted range, and all engines are of the same vintage.
- Complete knowledge of all failure modes is not known a priori, and new classes of abnormal operation must be identified as data is obtained. Additionally, modifications to the manufacturing

process will alter the signature of normal engines on a frequent, but unpredictable, time scale. The system must adapt to these changes as quickly as possible, with the constraint that training data will be very limited.

- The input data consists of information from fewer sensors sampled more frequently, making the problem more like pattern recognition in complex waveforms and less like a sensor fusion problem.
- Training data for faulty engines is a tiny fraction of the data available for normal engines and the statistical distributions for very rare abnormalities may never be known very well.
- We are interested not only in detecting and diagnosing faults, but also in monitoring drifts from nominal in the manufacturing process.

All of these circumstances conspire to make this classification problem quite difficult. In particular, this classification system must have a very low false alarm rate, a high accuracy rate for identification of faults, be readily adaptable to changes in the process and still function as a "novelty" detector to identify engines with new faults not presented in training samples. The simple, brute force application of backpropagation to analysis of raw data did not reliably produce a classifier with these properties. However, the methods we have developed can deal successfully with these circumstances and be applied as well to a wide variety of other classification problems.

Briefly, our approach is to break the classification task down into elemental processes that can be modified to suit each individual application. We choose to utilize traditional classifier systems and neural networks together to obtain optimum performance for this diagnostic problem. The methods also rely heavily on Monte Carlo simulation to generate statistically representative samples of training data from rather sparse samples of real data. These simulations boot-strap information from reasonable assumptions about the underlying statistics which are updated as empirical statistical distributions emerge. Such mathematical artifices permit us to evaluate the expected performance of our classification system early in the development process, before we have an adequate amount of actual data and can be easily adapted to utilize the true statistics of the data.

II. INITIAL STUDIES

Initially we used a 4.0 liter 6 cylinder engine to investigate the feasibility of comprehensive cold test diagnostics on a representative sample of data. Only a single engine was available, and this engine was disassembled and reassembled with deliberately introduced faults to provide the initial database for our investigations. The engine was motored, typically at about 150 rpm, by an electric motor with an in-line torque transducer to measure the dynamic crankshaft torque. Simultaneously, pressure transducers monitored the intake and exhaust manifold pressures, the crankcase air pressure and the oil pressure. Measurements of each parameter were taken every 10 crank angle degrees, and a complete data sample consists of 70 measurements on each trace (2 x 35 samples per revolution due to a 36-1 tooth encoding wheel). Several cycles could be averaged together, but the observed cycle to cycle fluctuations were extremely small and one cycle appeared to be satisfactory. Therefore, the actual data acquisition time for this test was less than 1 second. Typical samples of data from normal and abnormal operation are shown in Figure 1. Visible on these traces are clear features associated with the engine fault, which an expert diagnostician could conceivably use to identify the nature of the fault. These traces were selected to manifest such recognizable features which often lead one to suspect that a simple rule based system could be constructed to perform the diagnostics. However, the engine to engine variability and the need to distinguish not only any one fault from normal operation, but also from all other faults, complicates matters. Closer examination of the traces reveals that in addition to primary discriminating features present at particular points in the trace, additional but smaller correlated features are present elsewhere in the traces. It is desirable to utilize all helpful discriminating features to construct a robust classifier

We used a conventional backpropagation (BP) neural network in a first assault on this problem. However, the raw data from test engine produced an unwieldy test vector with several hundred elements. Data were collected from a test suite of 28 different faults and normal operation (29 classes) and a data base of about 1500 test vectors was obtained. This data was artificially augmented with uncorrelated "noise" in an attempt to introduce process noise (the variability that could be expected from a larger sample of "identical" engines) into the data set. The data was divided into two equal parts for training and testing. A BP network with a 350-50-29 configuration (350 input nodes, 50 hidden nodes and 29 output nodes) was trained and performed acceptably well on the classification task (>98% accuracy). However, although networks of this size are manageable on small workstations (training was ultimately performed on an IBM RS6000 RISC processor), there are arguably too many free parameters (almost 19,000 trainable weights). No precise rules for selecting sample sizes have appeared in the literature, but we think it prudent to have at least one sample per trainable weight. However, with 19,000 sample vectors and 19,000 trainable weights, proper statistical testing of such a network is beyond the capabilities of modest workstations. We therefore sought to reduce the dimensionality of the input vectors to decrease the input data requirements and the training and testing times.

One approach to dimensionality reduction is to select a set of "features" in the data, based upon an understanding of the physical processes involved (e.g. zero crossing times, peak-to-valley ratios of torque etc.). We elected not to pursue this approach because we wanted to develop a scheme which required as little *a priori* knowledge as possible and therefore was applicable to a wide range of problems. Principal Component Analysis (PCA) is one well-known means of developing a new representation for a sample vector space. Typically, the PCA provides a compact representation of a sample vector space from which effective classifiers can be constructed. A full treatment of PCA is given in a text by Jolliffe, but a few basic features are noted here [3]. PCA is the projection of the vectors from the input samples onto a new set of orthogonal axes which are chosen to represent the largest variance in the sample of data presented. The first principal component is chosen as the direction which accounts for the largest variance in the data. The next (and subsequent) principal component(s) is (are) in the direction associated with the largest remaining variance, subject to the constraint that it is orthogonal to the preceding component(s) For many of our data sets, if we terminate the process after about 99% of the variance is accounted for, we observe more than 10:1 reduction in the dimensionality of input vector space . It is of perhaps more interest to note that the performance of neural network classifiers in these new representations improved over those using the original data representations.

If we apply PCA to our data set and terminate the PCA process after 99% of the variability has been accounted for, we obtain a vector space in the PCA representation with 27 components. A neural network in a 27-16-29 configuration (about 900 trainable weights), trained with 25% of the number of passes through the training set required for the raw data. Combining the smaller number of weight updates required with the smaller number of passes through the data, the use of the PC representation reduced the network training time by a factor of 100.

The PCA analysis could be applied to the complete vector space, in which case the sample space of 350 x 2000 is projected onto a new set of axes. However, the association of the resulting PCA components with any physical measurement is quite difficult and the computational task involves inverting very large matrices. To avoid these difficulties and to provide a means of visualizing and interpreting the PCA representation, we divided the input vector space into several subspaces and performed the PCA on those subspaces. The subspaces were the individual cylinder torque traces, the overall torque trace, the separate overall pressure traces, and finally the deviations of the individual cylinder events from the mean of all events for that engine. The exact details of this subspace decomposition are discussed elsewhere and precise decomposition is problem dependent [4]. The significance of this step is that its reduces the computational task for PCA, it simplifies the interpretation of the PCA and it very often reduces the number of PC's in each subspace to 3 or 4. Figure 2 indicates how various fault signatures appeared in the PCA representation. With the help of 3-D scatterplots and "slicing" in the fourth dimension, or with matrix plots, the PC data within each subspace can be easily visualized [5]. For our purposes, the decomposition of the engine data was into 11 subspaces with 2 to 5 PC's retained for each subspace. The 11 subspaces contained a total of 35 elements which comprised the reduced PC representation of the engine data (nearly a factor of 100 reduction). It is on this vector representation that the classification problem is attacked.

III. ANALYSIS

For a case study on real data, we were presented with data from over 1000 different pre-production engines. This dataset was obtained from a plant survey and lacked a *bona fide* classification for each engine, although very good engines and engines with serious defects were quite evident from the graphs. The problem was to develop a classifier which could identify GOOD from BAD and also identify any faults present in the engines under test. As a first step, we visually scanned all the raw data and identified as many engines as possible as GOOD or BAD and assembled a training set from this manually tagged data. A neural network was trained on this data set until its RMS error ceased to decrease. The classifications of the network were compared with ours and some adjustments were made to our classifications and the network was retrained on the retagged data set. After a few iterations on a training sample of 300 engines, the process converged to agreement between the network classifications and ours. The network was tested on the remaining engines and the results were compared with a technician's analysis of the data. In most cases, the expert technician and the network were in agreement, although the technician was analyzing raw data and the network was analyzing the PCA data.

In reviewing this database, we noticed that sudden changes in the signal spectra took place as a result of changes introduced in the manufacturing process. For example, such an effect could be caused by a change in the lubricating oil in the engine which reduces the turnover torque. This situation caused batches of data within the database to have different means and slightly different variances. Consequently, the amount of real data which would be available to provide examples for training sets seemed likely to be very limited. Further analysis of the PC's revealed that the covariance matrix of the PC data contained off-diagonal terms, indicating that the individual raw signal traces from each engine were correlated. It was noted that the sample means of the PC's varied from production batch to batch, but that the covariance matrix was stable. To re-train a network each time such a shift in the production occurred would require copious quantities of data, which would not be available until some time after each change in the production process. A viable solution to this problem is to utilize the fact that the second-order statistics of the measurement problem are stable and incorporate Monte Carlo methods to generate sufficient data from estimates of the sample means. Unlike our initial study in which we utilized uncorrelated noise, we now needed to generate Monte Carlo data with the same covariance as the real data. A detailed description of the means to carry out this procedure is contained in the Appendix. The Monte Carlo process may be used to generate augmented data sets of both normal and faulty engines if one makes the reasonable assumption that the faulty engines' PC's have covariance matrices similar to that of the normals. This data augmentation process also helps to identify "class clusters" that are easy to separate. In the past, higher success rates for proper class identification of abnormal situations were claimed than could actually be obtained in practice because the variance in the clusters of abnormals was not properly accounted for. In our approach, we base our estimates of the cluster statistics on the historical data and amend the statistics as necessary to be consistent with the incoming data. In most cases, the proper consideration of all the cluster variances diminishes the ability to separate all the fault categories. However, the performance observed in development provides a more accurate gauge of final performance.

In attempting to provide a diagnostic tool which is easy to manage and re-train, we noted that the PCA data, broken down into the 11 subspaces could be very effectively classified as GOOD or BAD by a hard shell classifier defined by elliptical shells centered on the centroids of the distribution of GOOD engines with axes radii determined by the variance of the distributions. Normalization of the distributions to zero mean and unit variance simplifies the classifier boundaries to spherical shells. An ideal engine would be most similar to the best engine identified or the mean of an ensemble of such engines. If the deviation of an engine from such a distribution is larger than an acceptable value, the engine is declared to be unsatisfactory. In the early stages of this functional testing, no empirical data was available for selecting the tolerance boundary. We used Monte Carlo simulations to determine the variations we could expect from a single class of data with the proper covariance matrix. From this simulation we determined that shells with radii shown in Figure 3 would contain virtually all of the Monte Carlo samples. To pass, an engine must fall within all 11 shells constructed for the 11 vector PC subspaces. However, since the Monte Carlo statistics are Gaussian, a fraction the samples will fall outside some spheres. If the values associated with the hard-shell classifiers are selected as shown in Figure 3, we have determined that the GOOD engines should score 9.0 or higher (on a scale of 11) in order to pass 99% of the samples. The histogram of the Monte Carlo data for the expected distribution of GOOD engines is shown in Figure 4. If the engine falls below the threshold value, then the neural network will be used to identify the failure present. This approach provides an easily understandable, traditional classifier for acceptance and rejection based upon the assumption of a convex data set for the normal engines. The neural network is used for the task it can perform well, fault classification, which may involve very odd-shaped or non-convex sets of data. We anticipate that the class clusters are well-separated, but perhaps not by simple boundaries. The data set from the plant is consistent with this conjecture. Typically, the faulty engines from the production data scored below a 6 or 7, so that we may expect that the distributions of GOOD and BAD are as separable as they were in the initial laboratory study with the 3.0 liter engine. In this situation, we can effectively use standard feedforward networks trained by backpropagation, or utilize Restricted Coulomb Energy (RCE) networks which train much faster.

The process control aspect of this approach is evident if we monitor the engine scores as a function of time. For each major change in the production, the engine test scores dropped until new sample means were calculated. The neural network can provide information on the nature of the problem by indicating the "direction" or the tendency of a fault. For BP, we use one unit in the output layer for each fault class,, and as the data points move in the direction of a known fault, the GOOD output node decreases in value and the FAULT node associated with the class direction in which the data is moving increases in value. Thus, the neural network may be used to provide prognostic information about engines that have not crossed the threshold for outright rejection. We note that the BP network in this situation operates with the full 35 dimensional input space as a fully interconnected network. Investigations are underway to determine if subspace groupings, as used for the hard shell acceptance classifier, applied to the RCE network provide any benefits.

IV. CONCLUSIONS

We have demonstrated how a combination of conventional statistical processing methods and neural networks can be combined to create a classifier system for engine diagnostics. The most significant computational effort is required to compute the PCA and to properly develop the hard-shell classifiers using data sets augmented with Monte Carlo methods. Once these procedures are carried out, the application of neural networks to the data set to obtain the trainable classifier is quite straightforward. We expect that these methods are applicable to a wide range of classification problems.

REFERENCES

[1] K.A. Marko, J. James, J. Dosdall and J. Murphy, "Automotive Control System Diagnostics Using Neural Nets for Rapid Classification of Large Data Sets", Proceedings of IJCNN-89, pp. I I 13-17, Washington D.C., 1989.

[2] K.A. Marko, "Neural Network Application to Diagnostics and Control of Vehicle Control Systems", Neural Information Processing Symposium (NIPS-91) Denver, Colorado, Morgan-Kaufman, pp 337-343, New York 1991.

[3] Jolliffe, I.T. "Principal Component Analysis", Springer-Verlag, Berlin 1986.

[4] K.A. Marko, B. Bryant, N. Soderborg, Ford Technical Report (to be published).

[5] See for example "Data Desk" software for Macintosh Computers, Data Description Inc., PO Box 4555, Ithaca, New York.

[6] Strang, C.G., "Introduction to Applied Mathematics" Wellesley Cambridge Press, Wellesley, MA. 1986.

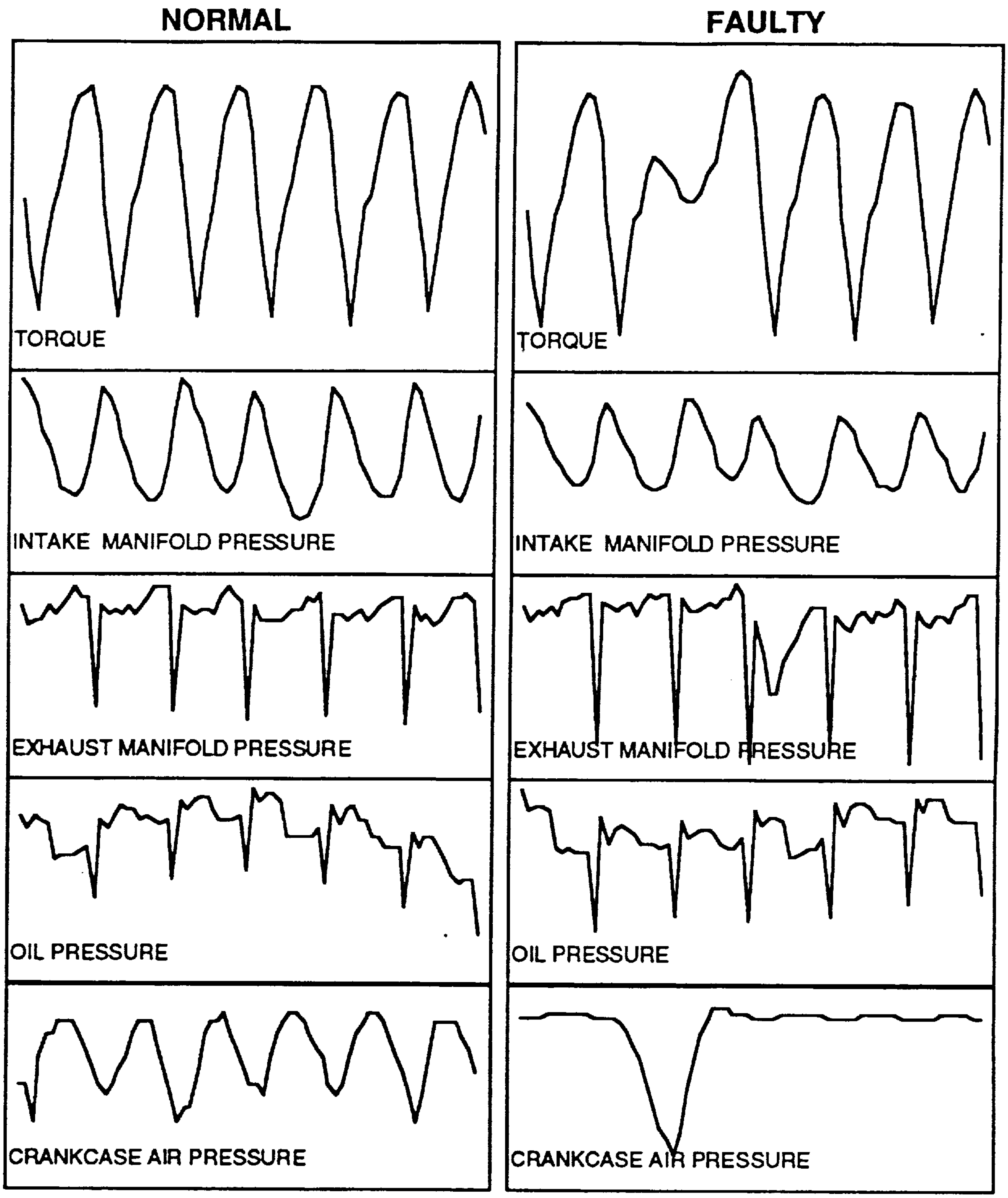

FIGURE 1. Data traces obtained from normal engine (on the left) and from an engine with an easily detectable fault (on the right). The traces are based upon sampling the analog signals one every crankangle degree, so that each trace consists of 720 points.

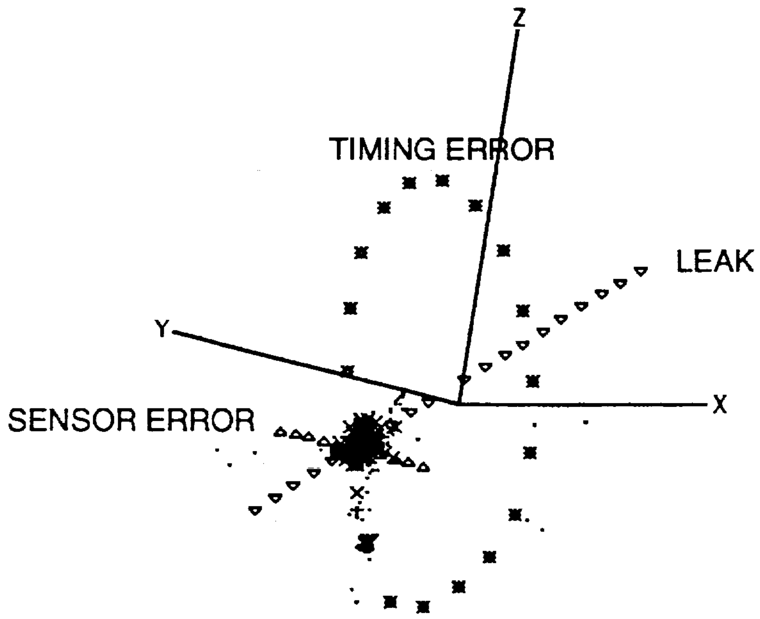

Figure 2. Plot of artifically induced "faults" in PC represenation of exhaust manifold signals. Dense cluster of dots represents "normal" engines. The other other signals indicate the effects of introducing various faults, such as camshaft timing error, or leaks into the engine.

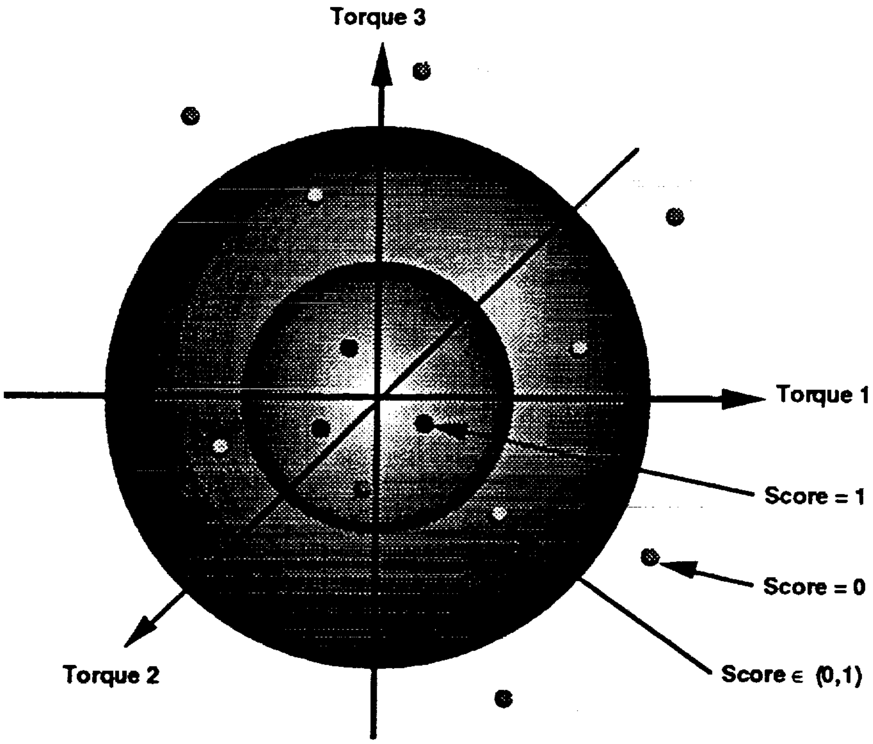

Figure 3. Spheroidal Classifier. Engines are rated according to the location of their data points (shown as small dots) relative to spherical shells whose size and location are determined from a set of nominal good engines. The radii of the shells are proportional to the variance of the empirical distributions. We typically choose 2 times the standard deviation (S.D.) for the inner radius and 3 times the S.D. for the outer radius. The engine test score is determined from 11 such classifiers in the PCA subspaces described in the text. Engines with all points within the inner spheres have a perfect score of 11.

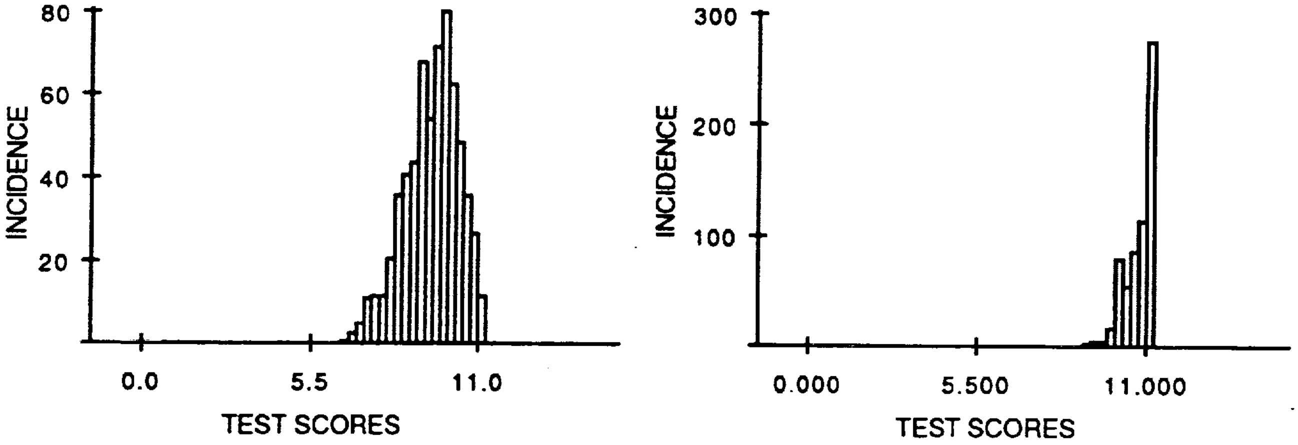

Figure 4. Histograms of the engine scores from the Monte Carlo simulations. The distribution on the left is due to data generated from normally distributed PCA data with a diagonal covariance matrix. The distribution on the right is due the same data transformed to have the covariance obtained empirically from production data This Monte Carlo simulation of GOOD engines cuts off below a test score of 9.