Initial Results on Fault Diagnosis of DSN Antenna Control Assemblies Using Pattern Recognition Techniques

P. Smyth
Communications Systems Research Section

J. Mellstrom
Ground Antennas and Facilities Engineering Section

This article describes initial results obtained from an investigation of using pattern recognition techniques for identifying fault modes in the Deep Space Network (DSN) 70-m antenna control assembly. It describes the overall background to the problem, outlining the motivation and potential benefits of this approach. In particular, it describes an experiment in which fault modes were introduced into a state-space simulation of the antenna control loops. By training a multilayer feed-forward neural network on the simulated sensor output, classification rates of over 95 percent were achieved with a false alarm rate of zero on unseen test data. It concludes that although the neural classifier has certain practical limitations at present, it also has considerable potential for problems of this nature.

I. Background and Motivation

Very accurate and precise pointing is a characteristic of the Deep Space Network (DSN) antennas. Some recent pointing problems have led to an interest in investigating automated methods of fault detection and identification within the antenna control assembly (ACA). The ACA for the 70-m antenna is a two-axis (azimuth and elevation) digital control system. In its simplest configuration, each axis controller consists of several servo-valve-controlled hydraulic motors, countertorque motors, gears, analog electronics (power amplifiers, analog compensation, filters), tachometers, an encoder, a digital computer, and various digital interfaces. It gets more complicated if the antenna is operated in precision mode, in which the 70-m antenna position is slaved to that of a precision pointing mechanism called the master equatorial. Clearly, there are many hydraulic, electrical, mechanical, hydromechanical, and electromechanical components that may be subject to wear, degradation, and aging. Identifying the source of pointing degradation within the ACA is not a trivial problem.

Furthermore, although excellent performance of the ACA is critical for good antenna pointing, it is only a part of the complex interaction of people, procedures, and equipment that affects pointing. To track down a pointing problem through all this can sometimes be a very difficult task. As a result, component degradation often goes unnoticed, resulting in suboptimal system performance. No fault identification action is taken until the X-band pointing requirements are no longer met or catastrophic failure occurs. It was recently reported that the antenna subsystem functional requirements for test or diagnostic capabilities have not been fully met [1].

According to the *Deep Space Network System Functional Requirements,* network equipment shall be designed to have a service life of at least 10 years.[1] However, the *Deep Space Network Long Range Plan* indicates that existing antennas will be operating well into the 21st century with greater availability (99 percent), lower crew sizes (35 percent of 1992 levels), and at much higher frequencies (Ka-band).[2] The implications of these goals are that (1) the equipment related to antenna pointing must always operate at near optimal performance levels, (2) scheduled maintenance times must be reduced, (3) equipment failures must be eliminated, and (4) these must be accomplished with a reduction in the personnel available for monitoring, diagnostics, repair, and maintenance. Recognizing this, it was identified in the *Deep Space Network Long Range Plan* that over the next 20–30 years the DSN must develop computer-aided maintenance and expert systems capability.

The objective of maintenance is to keep equipment operating in a nominal condition. Historically, maintenance has meant the periodic inspection, replacement, and rebuilding of equipment that is critical to system performance. However, this strategy is expensive because it results in downtime to replace equipment that may be operating nominally, and it still does not guarantee against catastrophic failure. A more effective strategy is to schedule repairs based on the operating condition of the system. An automatic monitoring system that can detect deviations from the nominal system state and identify the source of the deviation is a more desirable method of scheduling maintenance, maintaining optimal performance, and avoiding catastrophic failure.

As indicated above, a suitable system for an investigation of automated fault detection and identification is the ACA of the 70-m antenna mechanical subsystem (ANT). More fully automated fault detection and identification clearly would assist current DSN operations and is absolutely necessary for future operations.

II. ACA System Model and Fault Simulation

For this investigation, the 70-m antenna azimuth drive was simulated operating in nominal condition and four fault conditions. These simulations were repeated at three different angular velocities: 0.0, 4.0, and 40.0 mdeg/sec. The rates were chosen to emulate the range of rates encountered in the azimuth drive during a spacecraft track. At low elevation angles, the azimuth rate is very small. As elevation angle increases, azimuth rate also increases. The azimuth drive of the 70-m antenna was simulated on MatrixX simulation software. MatrixX is a commercial engineering analysis and control design software package. It incorporates most of the matrix analysis functions in EISPACK and LINPACK. It also has a graphical environment for simulation of discrete and continuous models.

The model described in this article is similar to that reported in [2] and is very briefly described here. For detailed information, readers are referred to the original paper. A block diagram of the model is shown in Fig. 1. The model was a hybrid continuous and discrete time model. The antenna servo controller (ASC) in this simulation consisted of a discrete-time-state feedback control algorithm and a steady-state Kalman filter. Its inputs were the commanded position and position feedback (measured and quantized by a 20-bit encoder). The ASC outputs were the position estimate, rate command, and quantized rate command, a 12-bit digital-to-analog (D/A) conversion labelled DAC Out. The rate loop amplifier represented all the analog electronics, with inputs of rate command and tachometer voltage feedback, and valve current as output. The tachometer voltage feedback represented four tachometers, one for each drive motor. The valve converted an electrical signal to hydraulic flow. Its inputs and outputs were valve (coil) current and valve (hydraulic) flow, respectively. The motor model represented four hydraulic motors. The inputs were valve flow and load torque. The outputs were motor rate, tachometer rate, and differential hydraulic pressure. The structure model was a seventh-order model incorporating the dominant modes of the structure and gearboxes. Its inputs were motor rate and wind disturbance torque. Its outputs were structure position referenced at the encoder and load torque on the axis.

The model incorporates the nonlinearities of static and coulomb friction in the motors, deadband and hysteresis in the valve, position quantization (encoder), and control effort quantization (D/A conversion). At low antenna angular velocities, these nonlinearities are significant and make system analysis very difficult. Since the nonlinearities are discontinuous, it is not possible to get a linear approximation that is valid at low angular velocities. Unfortunately, almost all operation of DSN antennas is at angular velocities from 0.0 to 5.0 mdeg/sec.

[1] *Deep Space Network System Functional Requirements General Requirements and Policies Through 1988,* JPL Document 820–20, vol. 1, Rev. A (internal document), Jet Propulsion Laboratory, Pasadena, California, March 1, 1988.

[2] *Deep Space Network Long Range Plan,* JPL Document 801-1 (internal document), Jet Propulsion Laboratory, Pasadena, California, March 15, 1989.

The faults simulated for this investigation were faults that have actually occurred at one or more of the 70-m antennas. When these faults have occured at the antenna, they have been severe enough to affect antenna pointing, yet subtle enough to be very difficult to diagnose. Part of the difficulty is due to the effect of nonlinearities at operational velocities. Signals obtained at the antenna have such a complex structure in the time domain that it is often very difficult for operations personnel or an engineer to diagnose the fault.

The faults chosen for this investigation, how they were simulated, and their relationships to the actual antenna are described below:

(1) Tachometer failure: This corresponds to a break in a tachometer winding or another electrical connection. There are tachometers associated with each drive motor. Voltage ripple, inherent in any tachometer and/or caused by torque ripple of the motor, is reduced and some failure robustness is achieved by averaging the tachometers. The loss of one tachometer reduces the gain and bandwidth of the rate loop. As a result, the servo will not follow a command as quickly, and disturbances will not be rejected as well. This was simulated by reducing the tachometer voltage by one-fourth.

(2) Increased valve deadband: This corresponds to wear of the surfaces in the hydraulic valve. Very precise machining is required to manufacture a low-deadband valve. Flow of the hydraulic fluid wears these surfaces, especially if the fluid is carrying particulates. Greater deadband increases the limit cycle behavior of the servo. A limit cycle may be unavoidable even in the nominal case, but it reduces pointing performance and increases drive mechanical wear. This was simulated by increasing the deadband in the valve by a factor of 2.

(3) Increased static friction: The significant sources of static friction in the ACA are the valve, the motor, and the gear reducers. It is also caused by wear. The result of increased friction is increased limit cycling. For this investigation, static friction was simulated in the motor. The fault condition corresponded to increasing the static friction by a factor of 2.

(4) Tachometer noise: Tachometer noise corresponds to brush wear and/or bearing wear. It was simulated as additive Gaussian noise with zero mean and standard deviation that increased with velocity.

III. Classification Experiment

As described above, the data for the classification experiment were generated by introducing fault modes into the control-loop simulation model. In addition, data were obtained for normal operation in the absence of any of the four fault modes. Hence, there are in effect five classes. For each class, the system was simulated at three angular velocities, namely, 0, 4, and 40 mdeg/sec over a time span of 20 seconds for each rate, with a sampling resolution of 200 Hz. This yielded $4000 \times 3 \times 5 = 60,000$ data vectors in total. Each data vector has eight components, corresponding to eight system outputs or observable sensors in the simulator. These outputs are antenna rate, differential pressure, valve flow, encoder, rate command, position estimate, valve current, and tachometer voltage. Figure 2 shows a plot of these outputs over 20 seconds at a rate of 4 mdeg/sec under normal operation (no faults). As mentioned above, this corresponds to 4000 data points for each component of the output vector (for a particular class at a given rate). Figures 3, 4, 5, and 6 show output plots at the same rate for the four different faults, namely, tachometer failure, increased deadband, increased static friction, and tachometer noise, in that order. Clearly, the problem of discriminating the individual fault conditions from normal behavior is nontrivial, based on visual inspection of the waveforms. The problem is as follows: given part of the data, say the first 2000 points, derive a classification algorithm that can classify as accurately as possible the remainder of the data.

IV. Feature Generation

Although in principle it would be possible to use the 60,000 input vectors directly as input to a classifier, it is generally considered in the statistical pattern-recognition literature to be a better idea to generate "features" by preprocessing the data. Essentially, the aim is to transform the data into a feature domain, where the features possess greater discriminatory power than the values of the raw data do. Heuristic motivation for this technique comes from the observation that biological systems such as the human visual system use this approach. In addition, there are rigorous statistical arguments that show it is important to make as efficient use of the available data as possible, and transformation to a good feature domain promotes such efficiency. As an example, it might be desirable to transform the data to the frequency domain for a more efficient representation.

Although automated feature discovery systems exist (based on expansions such as the Karhunen-Loeve transform), by and large the technique that works best in practice is manual feature generation. In effect, by defining features thought to possess useful discriminatory power, the classifier is helped up the learning curve; in a statistical sense, this is equivalent to a prior bias on the hypothesis space.

For this particular problem, the initial study chose to define simple time-domain features such as the mean and the range. The motivation for this choice was to investigate how well one could classify the data by only using these very simple statistical indicators; as shall be seen, one can do surprisingly well. An arbitrary choice was made of a window size of 128 over which these features were estimated, which resulted in a reduction of the number of input data vectors from 60,000 to 465. Another advantage of the simple estimators over more sophisticated techniques was their robustness over small sample sizes; i.e., the variance of these estimators could be expected to be lower than Fourier-based estimators for the same amount of data. In turn, more robust estimators would lead to better generalization performance on unseen data.

For each of the differential pressure, valve current, and tachometer voltage outputs, the range, mean, and variance in each window were estimated, giving nine features. The slope of the encoder and position estimate and the mean of the rate command were also estimated, giving a total of 12 features in all. The data from the antenna rate and valve flow outputs were not used in this experiment, as they are not directly measurable in the actual physical control assembly in the stations.

It is instructive to view the discriminatory power of some of these features. In Fig. 7(a), the normalized values of the tachometer-voltage-mean feature as a function of the class values are plotted. The class numbers correspond to the four fault conditions described earlier, with class 5 being the absence of any fault, or normal conditions. It can be seen that this feature contains some discriminatory power for classes 1 and 3, but otherwise not much class information can be distinguished. Figure 7(b) shows a similar plot of the valve-current variance where class 4 (increased tachometer noise) is the only distinguishable class; naturally, the variance-based features possess the capability of discriminating such a class. In general, most of the other features possess even less discriminatory power on their own. Hence, whatever discriminatory power these features possess as a group will only be discovered by a classifier that can effectively combine these features into composite functions; i.e., it would be expected that, say, a simple linear discriminant classifier would not do very well on this problem.

V. Choosing a Classifier

In pattern recognition, there is a wide variety of different algorithms available for generating classification models from data. Among the most widely used methods are nearest-neighbor classifiers, Bayesian models, and, more recently, multilayer feed-forward perceptrons (neural networks). What is perhaps not so well known is that many of these schemes perform equally well across a broad range of problems if evaluated in terms of classification-error performance alone. In other words, the difference between these various schemes in terms of classification accuracy has been empirically shown to be often minimal [3,4]. What often matters then in choosing a classifier technique are other considerations, such as the efficiency of the learning algorithm, ease of implementation, amount of prior knowledge required, etc. For example, the nearest-neighbor classifier is easy to use, but can be very inefficient in terms of memory requirements to implement. The Bayesian approach, for problems involving nondiscrete or continuous-valued data in particular, often requires significant prior knowledge regarding the distribution of the data; for the antenna problem, since the plant under observation is essentially nonlinear, little can be said a priori regarding the distribution of parameters such as the range and variance of the outputs.

Hence, for the initial study at least, a neural network classifier was chosen. The classification of relatively "low-level" time varying waveforms, where there was little prior knowledge about the underlying form of the probability density functions, was considered a suitable problem for the neural approach [5]. Problems that appear to be similar in nature to human perceptual tasks intuitively seem to be typically well matched to connectionist models. In addition, a public-domain algorithm coded in C for exactly this purpose was available (and will be described in more detail in the next section), making it very easy to experiment with the neural approach; i.e., no coding effort was required. It is also worth noting at this point that in this small-scale initial study, the primary interest was in getting an idea of the scale of the problem; e.g., is it possible to classify these waveforms using very simple features?

VI. Conjugate-Gradient Neural Learning Algorithms

The well-known backpropagation algorithm [6] for training multilayer feed-forward neural networks is somewhat wasteful of computational resources, and it is relatively well known that practitioners resort to various unpublished "tricks" to speed up the algorithm in practice. Hence, until recently, although impressive results had been reported in the literature from using this algorithm, it was not practical to experiment with it without a significant investment in initial effort. However, recent results have taken a broader view of the algorithm, and by utilizing prior work in conventional optimization theory and practice, more standard and conventional approaches to backpropagation have developed. In particular, the algorithm used in this experiment is described by Barnard and Cole [7], which in turn is an application of a conjugate-gradient optimization algorithm of Powell [8]. The algorithm will not be described in detail here except to note a few practical points; the interested reader is referred to the original papers. As described by Barnard and Cole, the conjugate-gradient algorithm is usually able to locate the minimum of a multivariate function much faster than a pure gradient-descent technique. In practice, it was found that the algorithm performs consistently well on a variety of classification problems. Of course, with these techniques there is no guarantee of convergence to the global optimum, but again, in practice the algorithm has consistently generated near-optimal solutions.

A factor that is often glossed over in the literature is the choice of neural architecture. This prior choice of a network model is suboptimal in general, and one would prefer to have the algorithm automatically select the appropriate size architecture from the data. A number of research groups are pursuing this goal, but as yet there are no widely accepted robust algorithms available. Hence, in practice, one must choose a network architecture for the problem at hand, i.e., the number of "hidden" layers and number of "hidden units" at each layer. For this experiment, attention is restricted to three-layer models (i.e., one hidden layer). The Appendix describes in more detail the exact nature of the three-layer networks under consideration. Note that there are many other variations of neural network architectures, such as recurrent networks and Boltzmann machines. The three-layer network is the simplest of these models with universal approximation capabilities; i.e., in principle, it can approximate any function, given enough hidden units.

VII. Results of the Classification Experiment

As described earlier, the original simulator output data were preprocessed into 465 feature vectors, with 12 feature components in each vector. This gave 93 data vectors per class. On closer inspection of the data, it was decided that the transient portions of the waveforms could safely be eliminated from consideration. In practice, one would in effect implement a hierarchical classifier, where the data were initially classified as either transient or nontransient. In addition, it was decided that the low-rate case of rate = 0 was a special case, and since large portions of the waveform at this rate contained no information at all, including them in the experiment would not yield meaningful results. Hence, only the nontransient, nonzero-rate data were looked at. This resulted in further data reduction to 260 data vectors.

The experiment consisted of generating two disjoint (roughly equally sized) subsets of the original data, calling one the training set, the other the test set. The conjugate-gradient algorithm was run on the training set, and the resulting three-layer network was used to classify the data in the "unseen" test set. After eight runs of this nature on randomly chosen training and testing disjoint subsets of roughly equal size, the resulting mean classification accuracy was 95.1 percent with almost no variance. Figure 8 shows a so-called "confusion matrix" for one of the networks. The left-hand column denotes the true value of the class; the top row denotes the network's estimate. Hence, a perfect network would have all of its entries in the diagonal; an entry in location i, j indicates the number of test points of class i that were classified as j. Remembering that class 5 is normal behavior, it can be seen that the false alarm rate is zero; i.e., no normal windows are incorrectly classified as a fault condition. In addition, it can be seen that the network has trouble classifying only one class, namely, tachometer failure. The network tends to confuse it with either increased static friction (class 3) or normal mode (class 5). Apart from this class, it performs perfectly.

The results of this simple classification experiment are surprisingly good in the context of pattern recognition. In general, for a given set of features and a class variable, there is a theoretical upper bound (the Bayes optimal rate) on the classification accuracy that is attainable. For example, if the features are completely independent of the class variable, then the optimal strategy is always to choose the most likely class and, hence, the optimal rate is the prior probability of this class. Since in practice the upper bound on performance is often considerably less than 100 percent, a figure of 95 percent is quite respectable for an initial experiment.

VIII. Conclusions

The result of the neural network classification experiment is promising. Even though the faults were only single-mode failures of a simple nature, and only on a simulator, one has reason to believe that the real problem may be amenable to these techniques when one takes into account that the classifier as implemented did not use any of a wide variety of additional information that was available. For example, by treating the data vectors (windows) as random samples, all sequential information in the waveform was ignored; i.e., in practice one would use memory in the classifier to weight the current classification decision based on previous decisions (effectively using a "smoothness" assumption on the occurrence of faults over time).

Of course, the neural network approach has inherent drawbacks also. It may be difficult to ascertain which features, or combinations of features, are contributing most to the classification accuracy, although for three-layer networks there exist visual analysis techniques for this purpose. In addition, training the network on a Sun-3/260 typically consumed about 1.5 hours of computation (with no other processes running except for Unix overhead), while the training data correspond to only 10 seconds of actual elapsed (simulated) time. Hence, it is difficult to see the implementation of actual, practical neural network algorithms, which learn in real time in the field, until very large scale integrated (VLSI) neural hardware becomes available.

IX. Future Work

In general, the problem of real-time predictive and diagnostic monitoring of the antenna control assembly is quite a challenging one. It would be naive to expect that a simple "static" classifier, such as that presented in this paper, would be robust enough to work in the field. In particular, the assumption that there are clearly defined fault classes will probably not hold up in practice, so that approaches such as unsupervised classification techniques (in which the training data have no class labels) will need to be considered. In addition, there are a number of problems, both at the theoretical and implementation levels, with developing an autonomous monitoring system. These include, for example, issues of memory (when should the system discard old data?), validation (how can one verify or quantify the operation of such a system in a nonintrusive manner?), etc. Once these algorithmic issues are dealt with, it may be possible to develop dedicated VLSI hardware specifically for antenna-control-assembly monitoring.

It is proposed that these problems be addressed by using a phased approach, applying existing technologies to prototype systems in-lab, and experimenting with DSS 13 facilities. In this manner, the feasibility of these techniques can be proven without incurring significant risk, and the prototype can be gradually transferred to the DSN operations environment in a relatively low-cost manner.

References

[1] W. O. Wood, "Report on the Maintenance of RF Performance at the DSN 70-Meter Antennas," prepared for the Bendix Field Engineering Corporation, Columbia, Maryland, September 30, 1989.

[2] R. E. Hill, "Dynamic Models for Simulation of the 70-M Antenna Axis Servos," *TDA Progress Report 42-95*, vol. July–September 1988, Jet Propulsion Laboratory, Pasadena, California, pp. 32–50, November 15, 1988.

[3] Y. Lee and R. P. Lippmann, "Practical characteristics of neural network and conventional pattern classifiers on artificial and speech problems," presented at the 1989 IEEE Neural Information Processing Conference, Denver, Colorado, December 1989.

[4] S. M. Weiss and I. Kapouleas, "An empirical comparison of pattern recognition, neural nets, and machine learning classification methods," *Proceedings of IJCAI 1989*, Palo Alto, California, pp. 781–787, 1989.

[5] P. Smyth, "Automated Monitor and Control for Deep Space Network Subsystems," *TDA Progress Report 42-98*, vol. April–June 1989, Jet Propulsion Laboratory, Pasadena, California, pp. 110–120, August 15, 1989.

[6] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing, Vol. 1*, Cambridge, Massachusetts: MIT Press, pp. 318–362, 1986.

[7] E. Barnard and R. Cole, "A neural net training program based on conjugate-gradient optimization," Oregon Graduate Centre Technical Report No. CSE 89–014, Beaverton, Oregon, 1989.

[8] M. J. D. Powell, "Restart procedures for the conjugate gradient method," *Mathematical Programming*, vol. 12, pp. 241–254, April 1977.

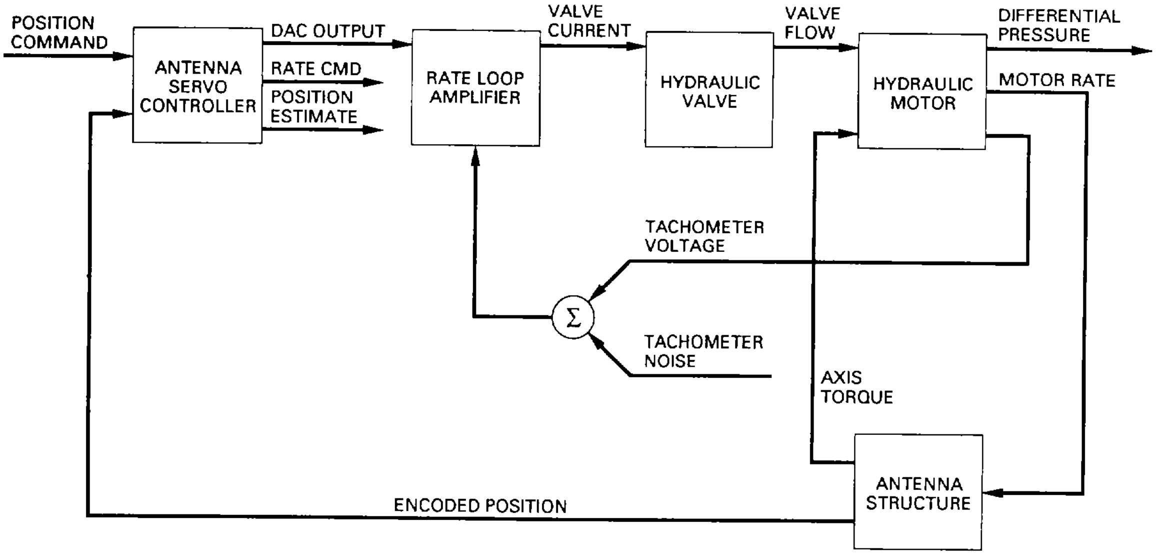

Fig. 1. The 70-m azimuth ACA simulation model.

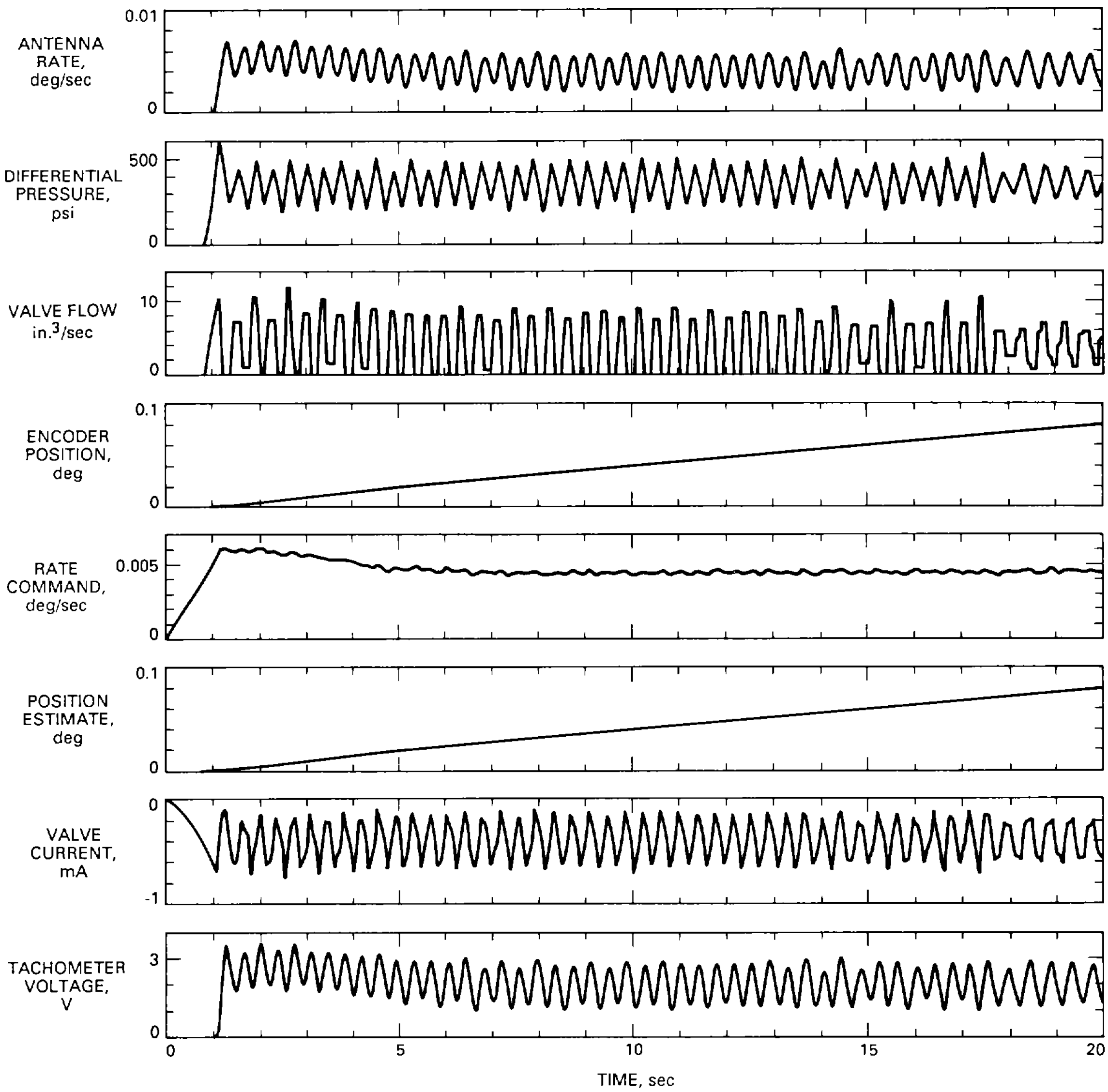

Fig. 2. Plot of each of eight features, sampled at 200 Hz for 20 sec, at rate = 4 mdeg/sec, under normal conditions (no faults).

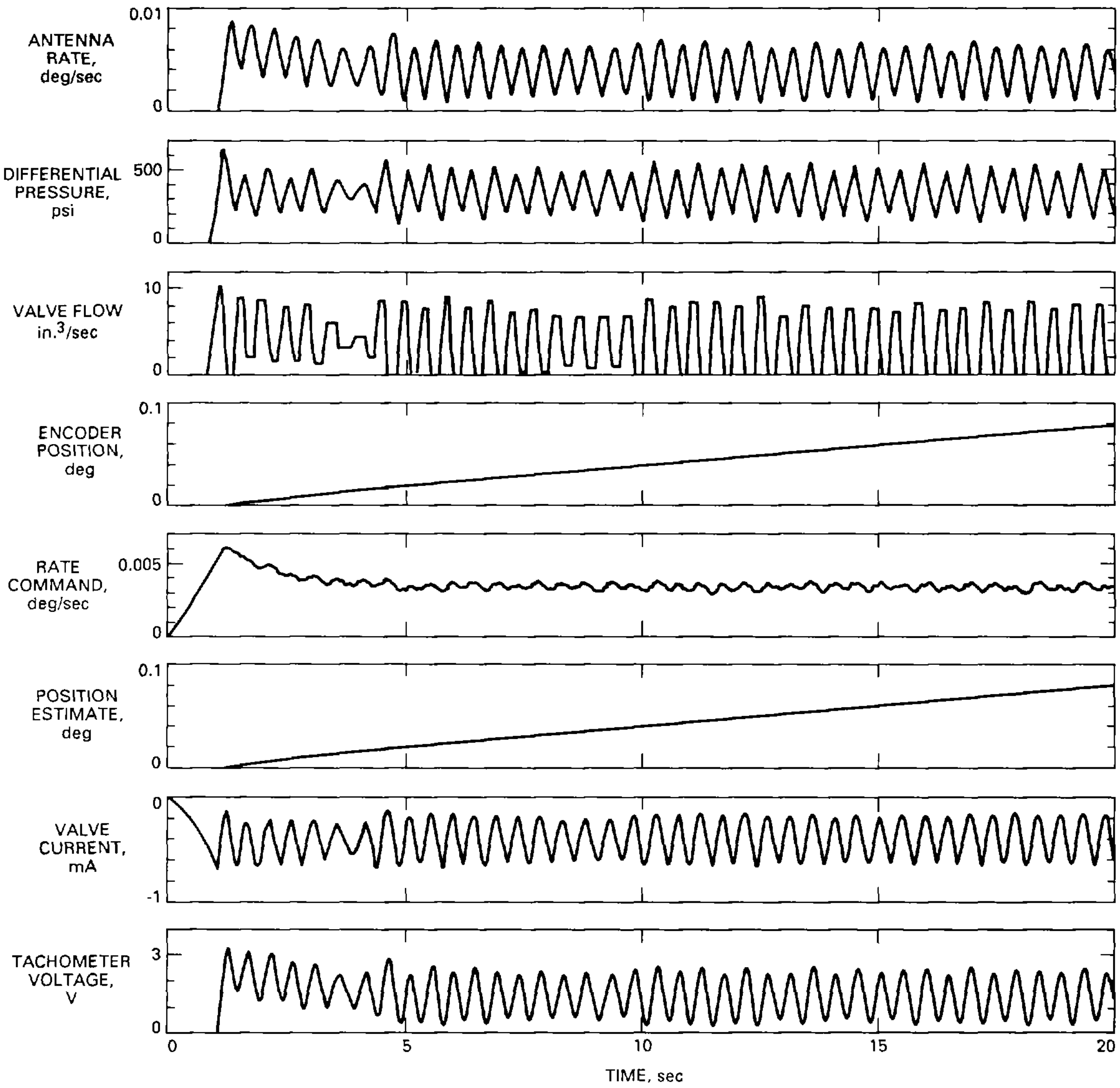

Fig. 3. Plot of each of eight features, sampled at 200 Hz for 20 sec, at rate = 4 mdeg/sec, with fault 1 (tachometer failure).

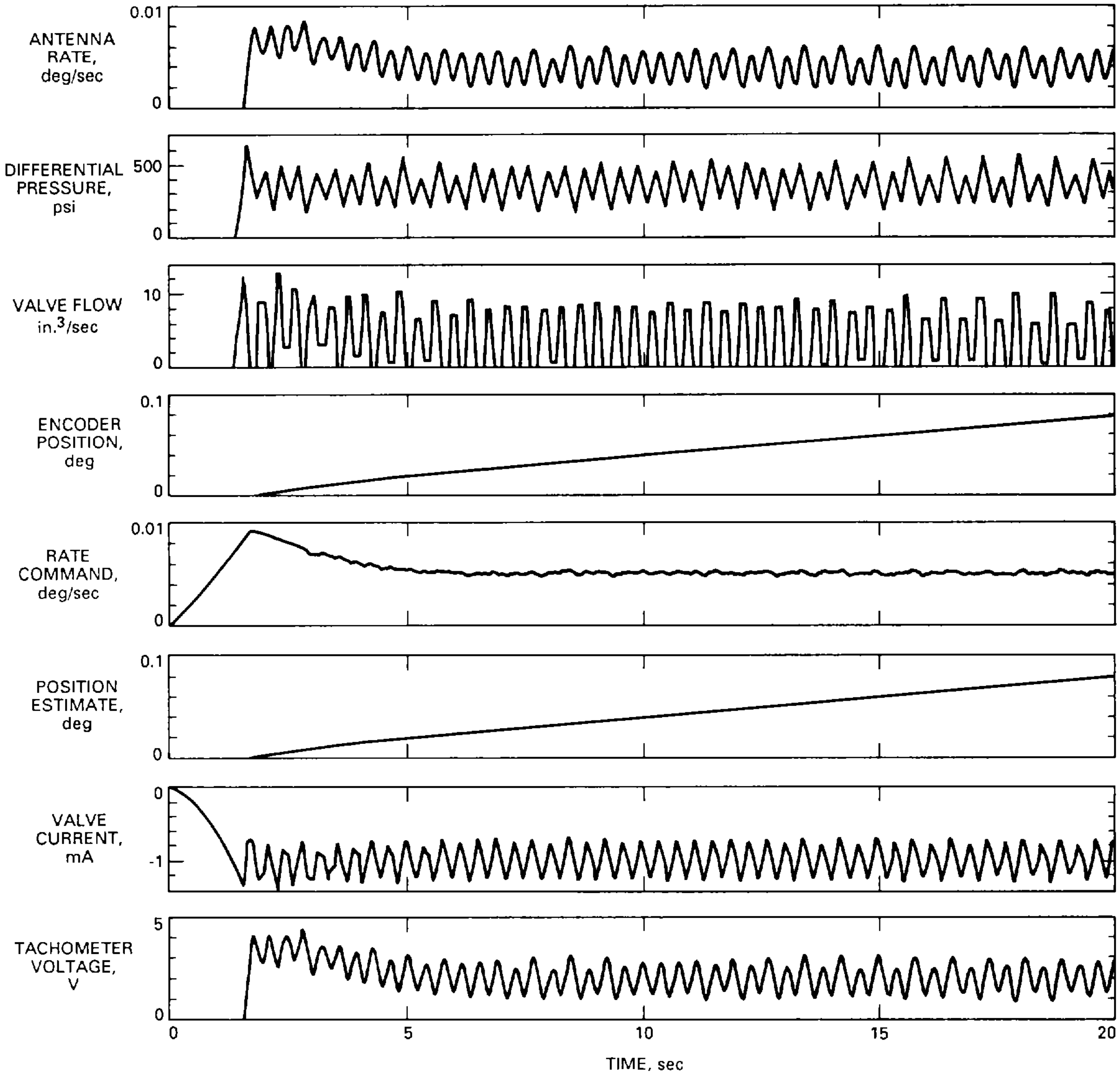

Fig. 4. Plot of each of eight features, sampled at 200 Hz for 20 sec, at rate = 4 mdeg/sec, with fault 2 (increased deadband).

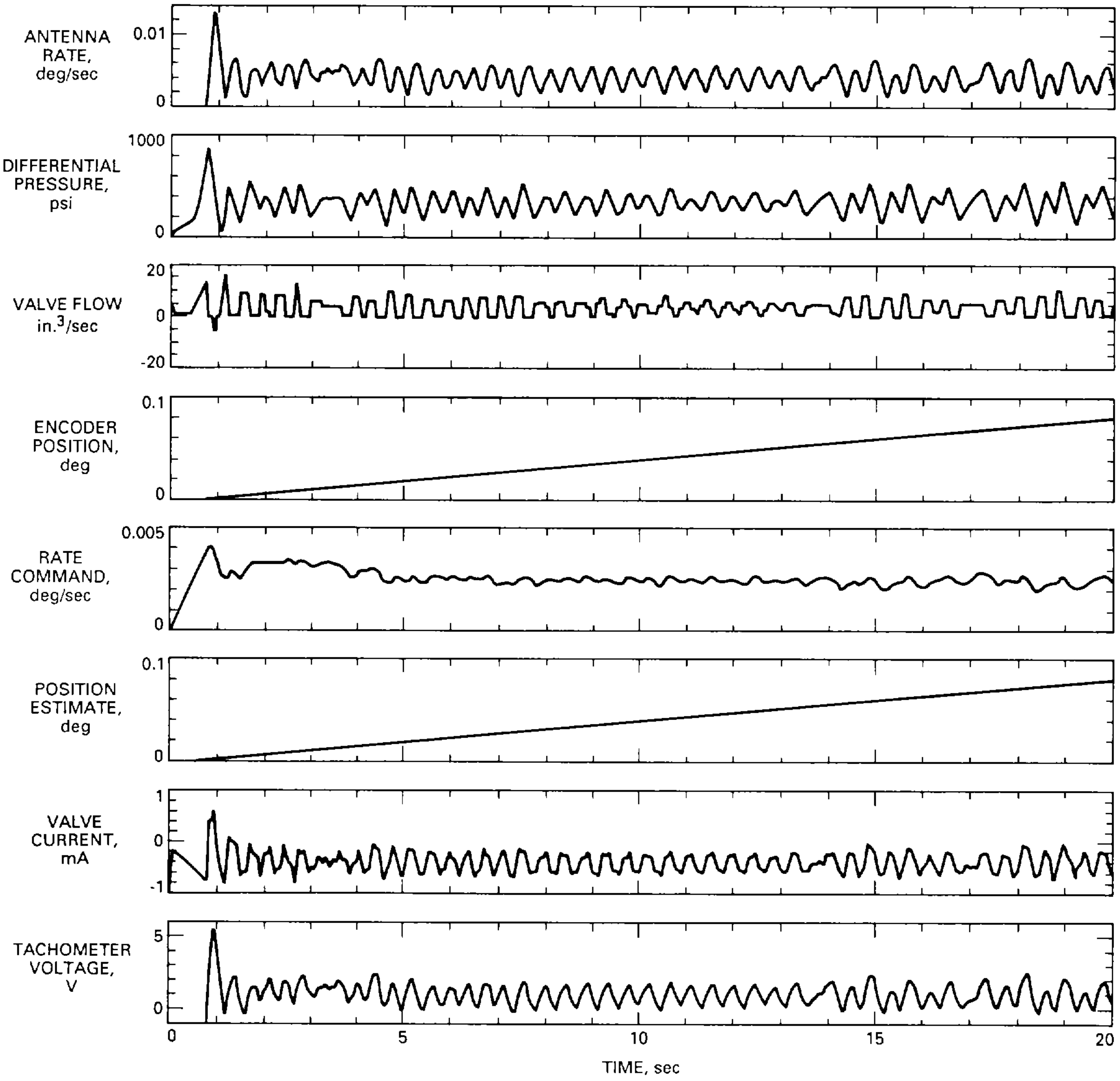

Fig. 5. Plot of each of eight features, sampled at 200 Hz for 20 sec, at rate = 4 mdeg/sec, with fault 3 (increased static friction).

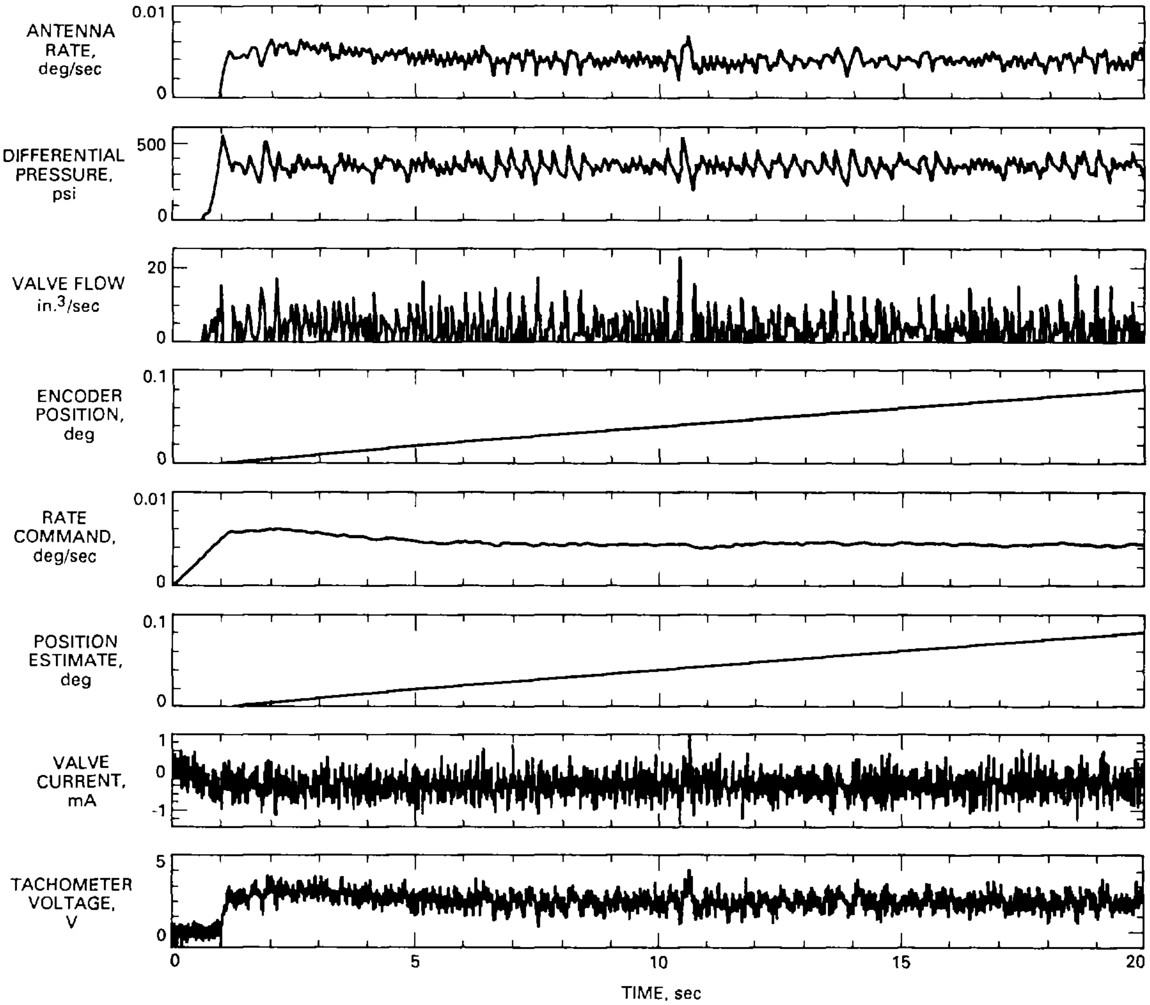

Fig. 6. Plot of each of eight features, sampled at 200 Hz for 20 sec, at rate = 4 mdeg/sec, with fault 4 (tachometer noise).

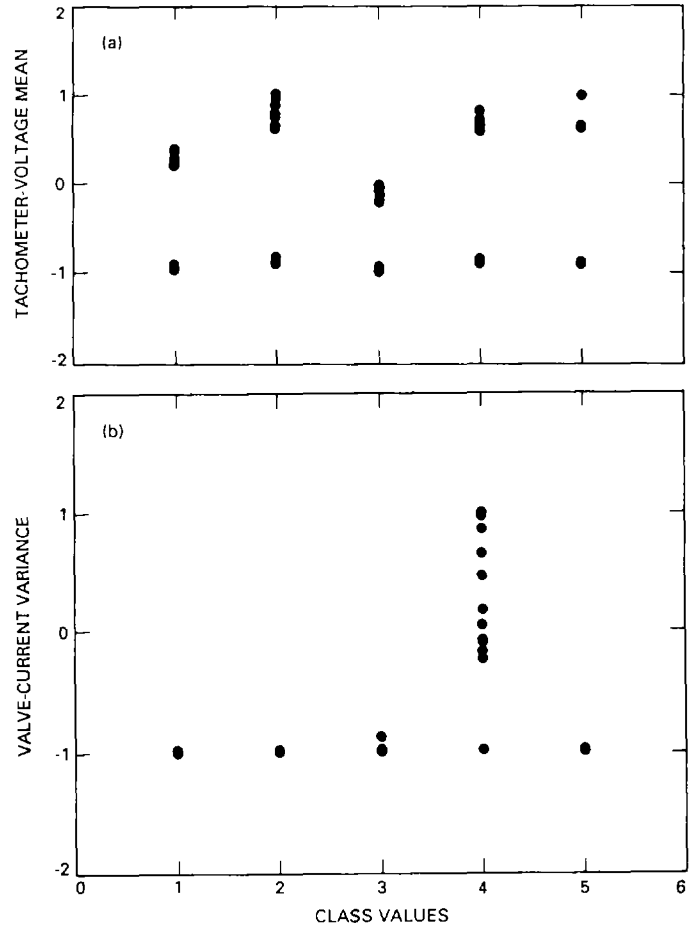

Fig. 7. Plots of (a) tachometer-voltage mean (normalized to ±1) versus class values; and (b) valve-current variance (normalized to ±1) versus class values.

CLASS AS ESTIMATED BY THREE-LAYER NEURAL NETWORK

ACTUAL TEST CLASS	1	2	3	4	5
1	18	0	3	0	3
2	0	24	0	0	0
3	0	0	25	0	0
4	0	0	0	28	0
5	0	0	0	0	25

CLASS 1: TACHOMETER FAILURE
CLASS 2: INCREASED DEADBAND
CLASS 3: INCREASED STATIC FRICTION
CLASS 4: TACHOMETER NOISE
CLASS 5: NOMINAL (NO FAULTS)

Fig. 8. Confusion matrix with 6 errors resulting from testing a particular neural network on independent test data of size 126 (percentage error = 4.8).

Appendix
Three-Layer Networks

Figure A-1 shows an example of a network. The input nodes are labeled n_i, the hidden nodes are labelled h_i, and the output layers are labelled o_i. In general, there are $K+1$ input units, where K is the number of features (12 in this case). The extra node is always in the "on" state, providing a threshold capability. Similarly, there are m output nodes, where $m = 5$ is the number of classes.

The number of hidden units was chosen arbitrarily in these experiments, but an empirically found rule of thumb to have between 1.5 and 2 times the number of input units typically worked well. The size of this hidden layer can influence the classifier performance critically: too many hidden units, and the network overfits the data (i.e., the estimation error will be large), whereas too few hidden units leaves the network with insufficient representational power (i.e., the approximation error term is large). With the weight from input unit n_i to hidden unit h_j as w_{ij}, each hidden unit calculates a weighted sum and passes the result through a nonlinear sigmoid function $F()$, i.e.,

$$a(h_j) = F\left(\sum_{i=1}^{i=K+1} w_{ij}\, a(n_i)\right)$$

where $a(n_i)$ is the activation of input unit i—typically, the actual value of feature i normalized to the range $+1, -1$. The function $F(x)$ is defined as

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

Output unit k, $1 \leq k \leq 5$ calculates a similar weighted sum using the weights w_{jk} between the jth hidden unit and the kth output unit, i.e.,

$$a(o_k) = \sum_j w_{jk} a(h_j)$$

A classification decision is made by choosing the output unit with the largest activation for a given set of inputs (feature values); i.e., choose class k such that

$$k = \arg\max_i \{a(o_i)\}$$

Hence, the optimization problem is to find the best set of weights such that the mean-square prediction error on the training data is as small as possible. Note that strictly speaking, from a statistical point of view, this is not the appropriate criterion, as the error on the training data may be an overly optimistic estimator of the true error of the classifier on unseen samples. Nonetheless, provided the number of free parameters in the network is at least an order of magnitude less than the number of training data points available, this minimization of training error is a reasonably robust procedure in practice.

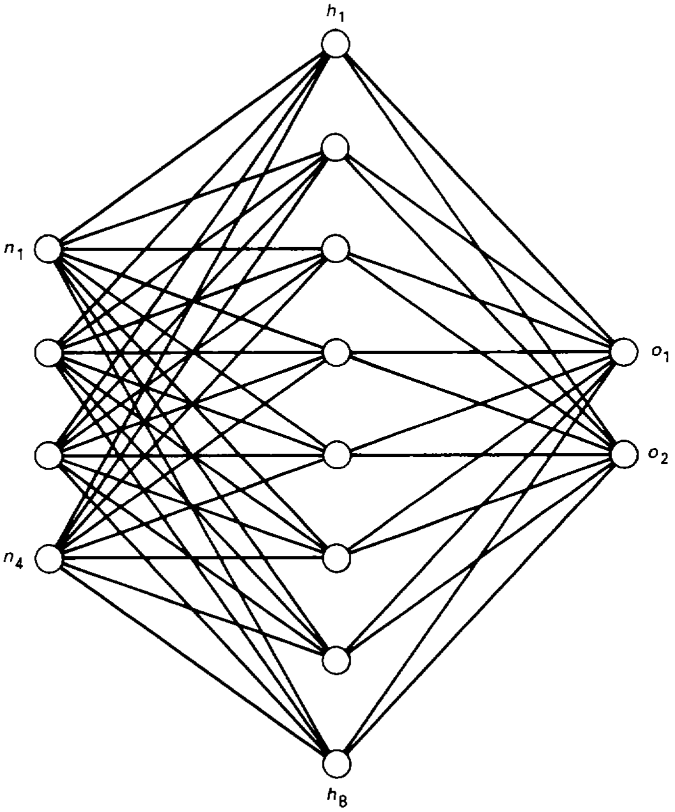

Fig. A-1. An example of a three-layer feed-forward neural network.