An Intelligent Fault Detection and Isolation Architecture For Antenna Arrays

K. Rahnamai,[1] P. Arabshahi,[2] T.-Y. Yan,[2] T. Pham,[2] and S. G. Finley[2]

This article describes a general architecture for fault modeling, diagnosis, and isolation of the DSN antenna array based on computationally intelligent techniques (neural networks and fuzzy logic). It encompasses a suite of intelligent test and diagnosis algorithms in software. By continuously monitoring the health of the highly complex and nonlinear array observables, the automated diagnosis software will be able to identify and isolate the most likely causes of system failure in cases of faulty operation. Furthermore, it will be able to recommend a series of corresponding corrective actions and effectively act as an automated real-time and interactive system supervisor. In so doing, it will enhance the array capability by reducing the operational workload, increasing science information availability, reducing the overall cost of operation by reducing system downtimes, improving risk management, and making mission planning much more reliable. Operation of this architecture is illustrated using examples from observables available from the 34-meter arraying task.

I. Introduction

The Deep Space Network (DSN) for tracking and communicating with spacecraft is of paramount importance and fundamental utility to NASA's present and future missions. Accurate, reliable, and robust operation of the antenna subsystems, as well as the complex chain of signal processing and communication equipment from the spacecraft down to mission control, is imperative to any mission success. When a problem occurs, its identification, isolation, and rapid and successful solution are critical to mission objectives, whether the fault has occurred in the ground station, the spacecraft, or elsewhere.

The architecture described here provides a cohesive system fault modeling, diagnosis, and isolation procedure applicable to the DSN antenna array. Instead of generally conceived ad hoc approaches, this article describes a systematic procedure using computationally intelligent techniques (neural networks and fuzzy logic). The approach has been applied successfully to complex problems such as actuator/surface failure detection and identification for reconfigurable aircraft flight control systems [2–4]. Its application to antenna array fault diagnosis will involve capture of, and knowledge-discovery on, the valuable and large set of data on array operations gathered during the Galileo mission. It will encompass a suite of intelligent test and diagnosis algorithms in software.

[1] Western New England College, Springfield, Massachusetts. This work was performed under a NASA Summer Faculty fellowship at the Jet Propulsion Laboratory, Communications Systems and Research Section.

[2] Communications Systems and Research Section.

By continuously monitoring the health of the highly complex and nonlinear DSN array, the architecture will be able to identify and isolate the most likely causes of system failure in cases of faulty operation. Furthermore, it will be able to recommend a series of corresponding corrective actions and effectively act as an automated real-time and interactive system supervisor. In so doing, it will enhance the array capability by decreasing the operational workload, increasing science information availability, reducing the overall cost of operation by reducing system downtimes, improving risk management, and making mission planning much more reliable.

II. Background and Motivation

The era of the New Millennium mission series brings JPL to a revolutionary period for spacecraft design, deployment, and tracking. This new paradigm requires reliable tracking with low system failures as well as increased demand on minimizing system downtime. NASA is planning to use microtechnology and instrumentation and more frequently to launch smaller, lower-cost spacecraft with focused mission objectives. Since budget limitations prohibit increases in mission operations staff, this vision sets forth a tremendous challenge to introduce intelligent automation into all aspects of mission control, ground tracking, telemetry equipment, and failure diagnosis. Operating a larger number of spacecraft results in less tracking time devoted to each mission pass. In addition, smaller spacecraft will have much less transmission power, which in turn requires antenna arraying to improve the signal-to-noise ratio (SNR) at the communication receiver. Due to increased system complexity and susceptibility to failures, antenna arraying considerably increases the complexity of the overall tracking and operations.

At the same time, costs associated with maintaining continuous operation of such a sophisticated tracking system are rising as the functionality of the antennas and other equipment degrades as a result of age. Some of the equipment, such as the next generation Block V receiver, already are complicated for station personnel operation, let alone unaided diagnosis in case of faulty operation. Others, such as the large 34- and 70-meter antennas, seldom exhibit continuous uninterrupted operation due to a variety of factors, such as atmospheric effects. With the ever-increasing sophistication of the antennas' electronic subsystems, their life-cycle maintenance costs (related to per-unit testing and diagnosis cost of a faulty system and its recovery rate) could in the near future exceed the corresponding original capital investment.

Presently all telemetry fault detection and isolation tasks are performed by mission staff on a manual case-by-case basis. Most often, due to complexity of the link, diagnosis cannot be performed in real time, resulting in frequent loss of telemetry data or early shutdown of the track. A glance at the operation logs of the antennas reveals daily multiple outages from a few minutes to several hours at times. The impact of such losses of contact with the spacecraft potentially could result in very difficult situations in mission critical operations. Add to this the loss of valuable science data associated with these outages and the recovery cost, and the necessity of development of an intelligent, comprehensive fault diagnosis unit becomes inevitable.

Without such a unit, tracking down the problem after it has happened can be a daunting task for the operator faced with such a complex system. If the problem symptoms cannot be recreated accurately, there is room for misdiagnosis and replacement of the wrong hardware/software module or, at worst, the potential exists for ignoring the problem until it becomes more severe. This last possibility is quite realistic and can manifest itself as a catastrophic failure resulting from a cascade of simpler faults. The solution lies in correct and timely diagnosis of such lower-level faults, and this is precisely where the human operator may have difficulty, due to the apparent transparency of such simpler faults.

The antenna subsystem and its associated signal processing/communication link are, therefore, the single most difficult pieces of equipment to monitor and troubleshoot accurately. They exhibit nonlinear behavior over a wide range of interconnected devices and interdependent and tightly coupled parameters, which is not amenable to standard linear models and associated diagnosis techniques.

The magnitude of the problem is even greater when a multiple number of antennas are electronically arrayed together for a specific mission, as shown in Fig. 1. Such an array was recently completed to improve by a factor of 100 the data return capability from the Galileo spacecraft (using encoding and compression as well). Arraying enables mission developers to reduce spacecraft costs by using smaller spacecraft antennas and transmitters. The issue of proper, rapid, and accurate fault diagnosis and recovery for the array thus becomes even more important (in terms of cost savings and overall mission success) than the single-antenna diagnosis problem.

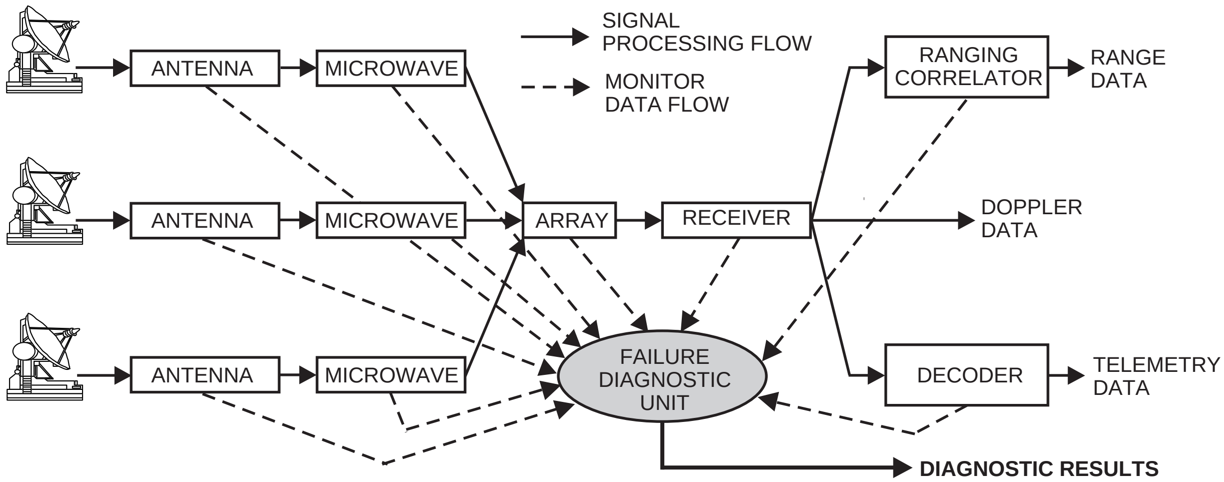

Fig. 1. DSN antenna array.

III. Nonlinear Fault Detection–Isolation Architecture

The intelligent fault detection and isolation system described in this article is based on fuzzy logic and neural networks. Such technologies are capable of nonlinear system modeling based on human expert knowledge. Furthermore, they are well suited to problems in which traditional or regular artificial intelligence-based diagnosis techniques have failed or achieved partial "costly success." A comprehensive approach to the problem of DSN telecommunication system diagnosis using such algorithms will represent by far the most ambitious, far reaching, and useful application of these methodologies, and will put NASA at the forefront of intelligent fault diagnosis and recovery techniques for complex systems.

In the past decade, considerable research has been devoted to the area of fault diagnosis. This trend has been the result of an increase in complexity of various systems and manufacturing processes as well as demand for reduced operational costs. In general, failure detection and isolation (FDI) methodology can be divided into two classes: (1) model based and (2) model free. As the name suggests, model-based techniques use the system model to detect faults. This, for instance, has been investigated and applied to actuator/surface failure detection and identification for reconfigurable aircraft flight-control systems [2–4]. In contrast, model-free methods rely solely on sensor measurements and do not require a priori estimation of knowledge or the system model. An example of this approach is power-plant fault diagnosis [5]. A typical FDI process contains the following subprocesses: (1) fault detection, (2) fault-level estimation, (3) fault isolation, and (4) fault recovery.

At the same time, the field of computational intelligence or soft computing also has advanced dramatically in the past three decades. Many new powerful paradigms, such as neural computing, fuzzy logic, or combinations of the two, have evolved that allow development of intelligent, adaptive, dynamic diagnostic systems. These techniques parallel the ability of the human mind to reason and learn in an environment of uncertainty and imprecision.

Drawing upon advances made in these two fields, Fig. 2 presents a block diagram of an intelligent FDI system with a hierarchical pattern for information flow. The antenna array at the macro-level is a large and complex system with many subsystems and processes; this fact renders a global model-based fault diagnosis approach impractical. Therefore, in this article, a model-free intelligent methodology is presented that takes advantage of the latest developments in the area of soft computing. As shown in Fig. 2, the architecture consists of the following subsystems: (1) a signal preprocessor, (2) a neural network fault detector with statistical validation, (3) a failure classifier, (4) a fuzzy fault-isolation block, and (5) a fuzzy expert system executive block.

It should be noted that a model-based approach is quite viable for well-defined, mathematically tractable, smaller antenna tracking subsystems, and could be used very effectively as a failure isolation strategy. For example, antenna-pointing errors (errors in azimuth and elevation) can be traced through the antenna control system and the measured values. Therefore, while the overall macro-fault detection and isolation unit will not utilize an a priori model of the entire antenna array system, it could rely on recommendations and decisions reached at lower levels using model-based fault detection and isolation methods.

Successful implementation of such an architecture requires continuous access to downlink monitor data. The array control processor (ACP) of the 34-meter array implementation task currently ongoing at JPL is projected to provide such a capability. At present there are no provisions in place to allow almost "live" access to the network monitor and control (NMC) system data over JPL's intranet. Furthermore,

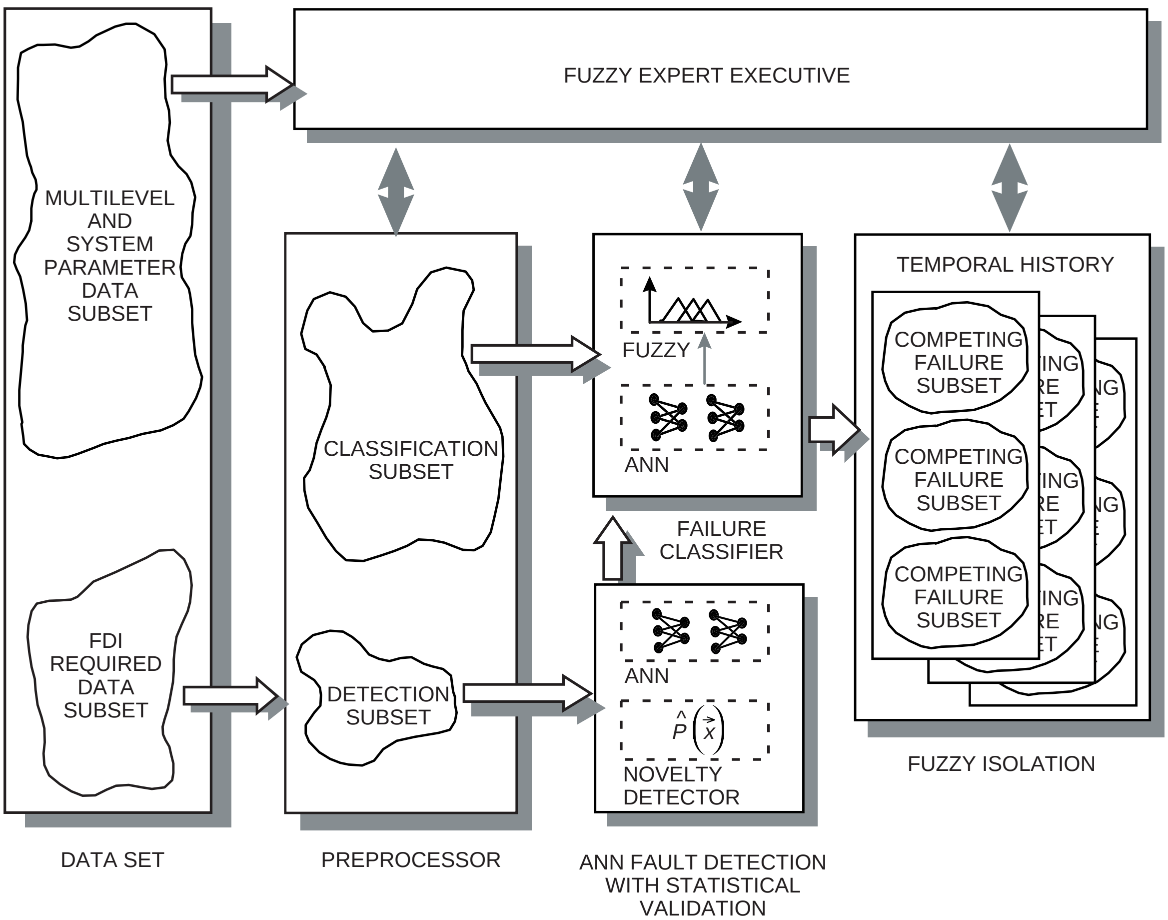

Fig. 2. DSN antenna array failure diagnostic unit.

due to large storage requirements, the telemetry system monitoring data is purged every 18 hours. An automated monitoring system needs to be developed that allows selection and storage of a subset of the real-time telemetry information.

Details of the various subsystems of the intelligent fault detection and isolation unit are presented next, using examples from the 34-meter arraying task.

A. Signal Preprocessor

Types of antenna array preprocessor monitor data include

(1) X-band SNR, P_c/N_0

(2) X-band system noise temperature (SNT)

(3) Ka-band SNR, P_c/N_0

(4) Antenna azimuth

(5) Antenna elevation

(6) Elevation error

(7) Air temperature

(8) Water vapor density

(9) Relative wind direction

(10) Relative wind speed

To establish a temporal correlation between various monitor data signals such as those listed above, all the input data must be synchronized. At present, different sampling rates are used for many of the data acquisition channels. The data from these channels need to be interpolated, or at the minimum properly indexed, for comparison/correlation purposes.

To minimize contamination of the data due to controllable sources, the signal preprocessor also includes various adaptive noise cancellation algorithms. These algorithms, whether conventional or neural network-based, will be used to reduce or remove excess noise on a select subset of the measured monitor signals. Other time-series denoising techniques, such as wavelet transforms or simple lowpass filtering, also may be included.

Another function of this block is to perform statistical estimation of the probability density function (pdf) for a selected monitor data signal. This will be used in the next stage (the neural network fault detection block) to place confidence intervals on decisions reached by the neural network.

B. Fault Detection With Statistical Validation

The second stage is a fault detection subsystem capable of minimizing the false alarm probability while maximizing the detection probability.[3] Failure detection is achieved in one of two ways: using neural networks or a fuzzy rule-based system.

1. Neural Network Approach. In the first approach, a trained neural network monitors critical data, such as the received signal-to-noise ratio (P_c/N_0) for the 8.45-GHz (X-band) and the 32-GHz

[3] Previous work [6] has shown that probabilities of missed detection and false alarm have inverse exponential relationships with each other. For every practical failure-detection application, an acceptable false alarm rate must be selected. This design criterion alone dictates the complexity of the FDI system. For this design, an acceptable false alarm rate will be selected, and the FDI system will be designed to meet or exceed the false alarm and missed-detection criteria.

(Ka-band) telemetry. Figure 3 shows a sample plot of these parameters that was collected in 1993 as part of the Mars Observer Ka-Band Link Experiment (KaBLE). The mission operators consider the system healthy as long as the SNR exceeds a predefined threshold. As can be seen from the figure, Ka-band P_c/N_0 starts degrading approximately 3.5 hours into the mission. This degradation continues and leads to a complete loss of communication and a subsequent corrective action approximately 4.5 hours into the mission. Early detection can be achieved by alerting the operator if the SNR falls below a threshold.

The proposed detection system makes use of a layered-perceptron artificial neural network (ANN).[4] Once trained and online, this network will generate a system health metric, based on observing the standard set of monitor data as they are made available. Based on this criterion, the network will be able to simply detect any degradation in system health. When the system health approaches such a critical point that corrective actions are justified and required, and before complete loss of tracking signal, a fault classification procedure will be initiated (next stage). In addition, a fuzzy failure level signifying the degree of importance of this particular failure will be assigned to the classified fault and will allow the executive block to make a decision to initiate the isolation phase.

An important aspect of the design and implementation of a neural network FDI system is determination of the degree of confidence in the answer the network provides (system health measure). In general, the unconditional probability density function of the input data to the ANN can be viewed as a quantitative

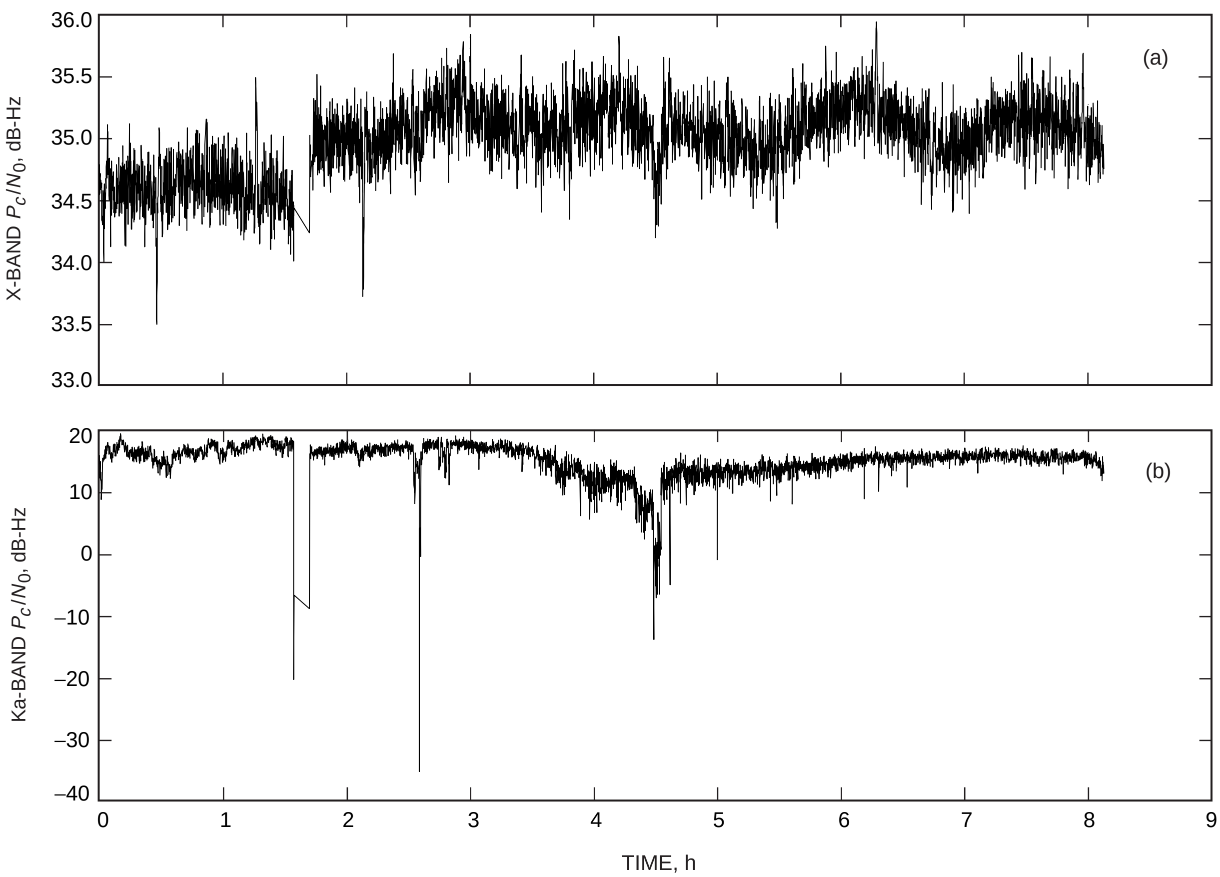

Fig. 3. Downlink tracking SNR time series: (a) X-band and (b) Ka-band.

[4] Appendix A provides an overview of neural computing.

measure of novelty with respect to the training data set [10]. That is, if a new input vector, $\vec{x}_i$, falls in a region of input space for which the density, $p(\vec{x})$, is high, then the network is interpolating effectively between training data points, and the performance generally will be good. In this case, we have a new vector that looks very similar to the training data. However, if the input vector falls in a region of input space for which $p(\vec{x})$ is low, then the input is essentially novel, and the network easily could generate erroneous output.

The procedure for detecting the health of the system, then, is as follows: The data that are used to train the network also are used to construct an estimate, $\hat{p}(\vec{x})$, of the unknown density, $p(\vec{x})$, at the signal preprocessor. Standard cross-validation (using partitions of the training set) then is used during neural network training to optimize the network topology and the values of regularization parameters as well as any smoothing parameters in the density model, $\hat{p}(\vec{x})$.

When the network is online and in use, each new input vector, $\vec{x}_i$, is presented to the network and used as well to evaluate $\hat{p}(\vec{x})$ in order to provide a measure of novelty. It can be shown that the network error is weighted by a variance factor of the form $\sigma(\vec{x}_i) = 1/\sqrt{p(\vec{x}_i)}$, which can be used to assign vector-dependent error bars to the network outputs via the model, $\hat{p}(\vec{x})$.

Depending on the application, there is statistical justification for placing a threshold on $\hat{p}(\vec{x})$ to reject all new data points for which $\hat{p}(\vec{x})$ falls below this threshold. This effectively classifies the input data sequence into two classes: those similar to the training data fall in class C_1, and those that are novel are assigned to class C_2. The training data are assumed to be drawn totally from C_1 with probability $P(C_1)$, and the new data are assumed to be drawn from either class with probabilities $P(C_1)$ and $P(C_2)$, where $P(C_1) + P(C_2) = 1$. Given a new input vector, the aim is to classify it into either class with minimum probability of error. Mathematically, the input vector is declared to belong to C_1 if $P(C_1|\vec{x}) > P(C_2|\vec{x})$ and to C_2 otherwise. Using Bayes theorem, this inequality can be rewritten as

$$p(\vec{x}|C_1) > \frac{p(\vec{x}|C_2)P(C_2)}{P(C_1)}$$

In the above inequality, $p(\vec{x}|C_1)$ is the pdf from which the training data were drawn and which has been modeled as $\hat{p}(x)$. The density $p(\vec{x}|C_2)$ belongs to the novel data. It is reasonable to assume that this density should be smaller in regions of input space where the density $\hat{p}(x)$ is large. Thus, one would choose $p(\vec{x}|C_2)$ to be of the form $f(p(\vec{x}|C_1))$, where f is a monotonically decreasing function. This results in a threshold criterion of the following form:

$$p(\vec{x}|C_1) > G^{-1}\left(\frac{P(C_1)}{P(C_2)}\right) = \text{ constant}$$

where $G(z) = f(z)/z$. In practice, an independent test set not used in the training and cross-validation process generally is used to confirm the network performance. In this case, one can evaluate $\hat{p}(\vec{x})$ for all points in the test set and then choose a value for the threshold such that most or all of these points are classified as belonging to C_1.

In addition, for a given input vector, x_i, the normalized version of $\hat{p}(\vec{x}_i)$ can be treated as a fuzzy novelty classifier signal. This information will be passed to the fuzzy executive and the neuro-fuzzy classifier block to be used as one of the failure classification criteria. For example, consider the following scenario: A failure has been detected by the detection block, and the classifier has determined that the elevation error or the relative wind speed could be the cause (very close output values from the neural net). But the novelty level of the relative wind speed and elevation error are 0.2 and 0.9, respectively. Definitely, this indicates high confidence in the elevation error results, and the executive block can either launch a procedure to check the control system or make a decision based on available information. This example also very simply illustrates the importance of this modularized approach and the need for an executive block.

2. Fuzzy Rule-Based Approach. In this approach, a fuzzy rule-based decision system is designed to detect the presence or impending presence of a fault.[5] In the simple case of a single antenna, where fault detection by the human operator may rely only on observation of the received signal-to-noise ratio, the input parameters to the decision system may be the SNR and the change in the SNR (over the previous time step). Additional inputs may be the SNR time series average over small, medium, and large time steps to determine the overall trend of the series (e.g., if the SNR is going down continuously and reaching lower and lower values, there is the possibility of occurrence of a fault).

Rules may be of the following format:

> IF the SNR is *fair*, AND the change in SNR is *negative big*, AND the SNR trend is decreasing *fast*, THEN the possibility of approaching a fault is *high*.

In the above rule, *fair*, *negative big*, *fast*, and *high* are all fuzzy constructs describing the fuzzy *variables* SNR, the change in SNR, SNR average, and possibility of fault occurrence. They are represented by fuzzy membership functions. A set of such rules needs to be constructed to cover the entire domain of the variables. Fuzzy set theory then is used to mathematically "quantify" such descriptions and results in a decision as to whether or not a fault has occurred or is approaching.

C. A Failure Classifier

This subsystem is responsible for classifying failures. It consists of a multilayer neural network with a fuzzy classifier shell (FCS).

Failure categorization is very similar to pattern recognition, and ANNs have been proven to be excellent pattern classifiers. The ANN takes advantage of the spatial and temporal context of a subset of monitor data, as directed by the executive block, to classify failures by detecting changes in input vector directions (i.e., recognizing the failure signature).

Time series data for each monitor signal are processed through the ANN using a nonoverlapping moving window. Each temporal input to the ANN is a vector in hyperspace and consequently has an associated vector direction. During normal operating mode (no failures), the direction of all the input vectors will remain constant with some statistical variance. Failures cause a change in the direction of the input vectors, forming a failure signature, which in turn is classified by the neural network to a class of failures.

The results of this classification are passed to the fuzzy classifier shell. This information, along with the fuzzy novelty signal generated by the failure detection block, will be used by the FCS to isolate the failures to a subset in the failure vector space.

For instance, assume that the normalized output of the classifier neural network for the elevation error, azimuth error, water vapor density, and relative wind direction are 0.92, 0.93, 0.3, and 0.4, respectively, with confidence levels of 0.95, 0.94, 0.96, and 0.9, respectively. Clearly, elevation error and the azimuth error signals could be the cause of the tracking loss and will be reported to the isolation subsystem along with supporting results. In general, the fuzzy classifier works like a shell around the neural network and identifies a set of likely failure candidates but will not make the final decision.

---

[5] Appendix B provides an overview of fuzzy systems.

D. Fuzzy Fault Isolation

This subsystem is responsible for isolating the failure to a single cause or a smaller subset of the failure space. If the failure has a unique failure signature, its cause can be isolated clearly. On the other hand, if the failure signature is not unique and suggests multiple possible causes, the information about the likely candidates will be passed on to the executive block for further analysis.

Under directives from the fuzzy expert executive block, this block will make the final decision on the most probable cause of the failure and will direct the operators toward corrective actions. In some instances, this block may interact with the operator for additional information or to check or modify specific system settings before making a final isolation decision.

This block also will act as a database for the short-term and long-term fault history data. For example, if multiple competing failure signatures are detected, this block could compile and wait for several time series results from the fault classifier subsystem before isolation of the failure. It also could act as a false alarm recovery module.

E. Fuzzy Executive Block

This executive block is responsible for all the global decision making and performs as a repository of all the expert's knowledge. This methodology provides a robust hybrid hierarchical decision-making platform that ensures low frequency of false alarm and missed-detection. For example, as shown in Fig. 3, approximately 1.5 hours into the mission, change of the modulation index caused a discontinuity in the X-band signal-to-noise ratio, P_c/N_0. This information is used by the executive routine to avoid an unnecessary false alarm.

If we consider the whole system a black box, this block will act as the brain and the nerve of the system. Through a series of interviews with the experts—mission staff and scientists involved in the development of the telemetry system—a fuzzy expert database will be constructed to provide executive directives and supervision for the overall system. This subsystem will be structured such that upgrades can be performed on a continual basis and the knowledge base will be to allowed to expand and improve as the system grows.

In addition, all the logical decisions are made by this block. Many of the monitor data are simply discrete signals by nature, and a fuzzy rule-based decision can be performed to obtain the system configuration status to aid the failure isolation process. A short list of the logical signals could contain the following: number of antennas in the link, modulation index, spacecraft ID, carrier frequency and loop-lock status, subcarrier frequency and loop-lock status, received total power, downlink carrier frequency, uplink carrier band, exciter, decoder type, and the expected carrier prediction error, etc. In summary, all the parameter settings for the receiver block are important factors and should be included in the failure isolation strategy.

IV. Application to the 34-Meter Array Task

The following two scenarios illustrate the use of such an intelligent FDI unit as part of the 34-meter arraying task.

Scenario 1: The decoder indicates that there is an undecodable frame. The FDI unit will retrace the signal-processing chain and examine if upstream components are working properly. For example, the unit would check on the lock status of the Block V receiver; the status of data alignment, combining, and antenna pointing; consistency between signal-to-noise measurements at the array and the receiver; consistency between measured and predicted signal power, etc. The unit then will deduce the cause of the problem. The result of the diagnosis and suggested corrective actions then will be reported to Mission

Operations. If more than one item is identified as a potential cause, they will be ranked according to the likelihood of occurrence.

Scenario 2: The ranging correlator detects low correlation. The FDI unit then examines the operation status of the downlink and uplink components to see if the transmitter or encoder of the ranging block is operating nominally, or if the receiver is locked, or perhaps if a configuration change as indicated in the operating sequence of events, such as one-way/two-way transition, may be to blame.

Many more such cases occur on a daily basis at DSN tracking stations. A well thought out architecture will be able rapidly to identify such problems, trace them to their causes, and recommend corrective action(s). The FDI unit can reside in the ACP of the 34-meter arraying task.

The benefits of such an FDI unit are manyfold. They include

(1) Expediting the troubleshooting process, letting operations personnel focus in rapidly on faulty devices.

(2) Facilitating the process of restoring the system to service by suggesting a series of proper corrective actions, ranked based on the types of faults encountered.

(3) Enhancing system operability by automation and reduction of the level of human expertise needed.

(4) Making mission planning much more reliable by acting as an automated risk management system and real-time, interactive system supervision tool.

(5) Substantially reducing life-cycle costs of the telecommunications system.

(6) Creating a vast body of new knowledge that will pave the way toward future similar tasks at NASA, greatly aiding the design and implementation of other low-cost diagnosis applications.

V. Conclusions

This article described a computationally intelligent fault detection and isolation (FDI) architecture for DSN antenna arrays. The architecture consists of five subsystems: a signal preprocessor, fault detection with statistical validation, a failure classifier, fault isolation, and a fuzzy expert executive block. Each subsystem serves a different function and is designed based on computationally intelligent algorithms. The general architecture has been applied successfully to complex problems such as actuator/surface failure detection and identification for reconfigurable aircraft flight control systems.

The effectiveness of the intelligent FDI system depends on the knowledge base collected during the training process. In general, there are a number of trade-offs involved in the design of a neural network. Since the network design is performed iteratively and concurrently with network training, the number of network nodes should be minimized to improve generalization and speed of training. Furthermore, definition of fuzzy sets also affects the accuracy or the confidence level of the diagnosis. The systematic approach presented in this article provides a solid foundation to develop an intelligent FDI system using neural network and fuzzy logics.

Acknowledgments

The authors would like to thank Kim Simpson for helpful discussion on the development of the diagnostic unit and Dave Watola for providing us with the data for the preliminary study.

References

[1] D. Driankov, H. Hellendoorn, and M. Reinfrank, *An Introduction to Fuzzy Control*, New York: Springer–Verlag, 1993.

[2] K. Rahnamai, A. K. Caglayan, and D. Moerder, "Detection and Identification of Control Impairment in a Self-Repairing Flight Control System," *Proc. National Aerospace and Electronics Conference*, Dayton, Ohio, May 1986.

[3] K. Rahnamai, A. K. Caglayan, and S. M. Allen, "Detection, Identification and Estimation of Surface Damage/Actuator Failure For High Performance Aircraft," *Proc. 1988 American Control Conference*, Atlanta, Georgia, June 15–17, 1988.

[4] K. Rahnamai, S. M. Allen, and A. K. Caglayan, "Failure Detection, Isolation and Estimation Strategy Design For Control Reconfigurable Combat Aircraft Real-Time Simulation," *Final Report No. R8905 for Grumman Aerospace*, Subcontract No. PO 20-46116, Grumman Aerospace, Boston, Massachusetts, April 1989.

[5] G. Guglielmi, T. Parisini, and G. Rossi,"Fault Diagnosis and Neural Networks: A Power Plant Application," *Control Engineering Practic*, vol. 3, no. 5, pp. 601–620, 1995.

[6] G. Lanyi and R. Kahn, "Tone Detection Via Incoherent Averaging of Fourier Transforms to Support the Automated Spacecraft-Monitoring Concept," *The Telecommunications and Data Acquisition Progress Report 42-129, January–March 1997*, Jet Propulsion Laboratory, Pasadena, California, pp. 1–22, May 15, 1997.
http://tda.jpl.nasa.gov/tda/progress_report/42-129/129E.pdf

[7] R. Reed, "Pruning Algorithms—A Survey," *IEEE Trans. Neural Networks*, vol. 4, no. 5, pp. 740–747, September 1993.

[8] P. Arabshahi and R. J. Marks II, "Fourier Analysis and Filtering of a Single Hidden Layer Perceptron," *IEEE Trans. Circuits and Systems II*, May 1998 (to appear).

[9] P. Arabshahi, J. J. Choi, R. J. Marks II, and T. P. Caudell, "Fuzzy Parameter Adaptation in Optimization: Some Neural Net Training Examples," *IEEE Computational Science and Engineering*, pp. 57–65, Spring 1996.

[10] C. M. Bishop, "Novelty Detection and Neural Network Validation," *IEE Proc.: Vision, Image and Signal Processing*, vol. 141, no. 4, pp. 217–222, August 1994.

[11] P. Arabshahi, R. J. Marks II, S. Oh, T. P. Caudell, and J. J. Choi, "Pointer Adaptation and Pruning of Min–Max Fuzzy Inference and Estimation," *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, no. 9, September 1997.

Appendix A
Overview of Neural Computing

There exists a variety of neural computing architectures that differ from each other in training or classification procedure, application, physical layout, and parameterization. The proposed effort uses the layered perceptron neural network architecture to process the desired operation for fault classification and isolation.

It is well known that neural networks are universal classifiers. For any given classification problem, one can construct a neural network that achieves optimum performance and effectively obtain results that are better than or highly competitive with traditional techniques. A properly designed, solid neural network classifier can offer attractive features such as robustness to noise and high classification accuracy. It is the selection of training and the network architecture that determines the effectiveness of the network.

A typical layered perceptron artificial neural network (ANN) is composed of elementary processors (neurons) and the interconnects (weights) that connect the neurons to each other, as shown in Fig. A-1. The operation of each neuron is predetermined and usually involves the application of a nonlinear function to the weighted sum of inputs from a previous layer. The proposed effort uses the sigmoid function as the nonlinearity, which is defined as $\mathrm{sigmoid}(x) = 1/(1 + e^{-x/T})$, where T represents the "temperature" of the system, specifying the degree of hardness or softness of the nonlinearity—it can be varied for each application. In general, the layered perceptron is composed of an input layer, one or more hidden layers, and an output layer, as shown in Fig. A-1 . In this structure, raw or preprocessed data samples are presented to the input layer (bottom). Typically the input layer serves only as a data holder or buffer for the hidden layer (middle), and, thus, no operations are performed on the data at this stage. Each hidden layer neuron then multiplies the input data values by its corresponding weight vector (vector $\vec{w}_3$ is shown by heavier lines in the figure), and passes them through the sigmoidal nonlinearity. Results from the hidden layer are then further weighted and passed on to the output layer, consisting of one or more neurons, where another nonlinear or linear operation may be performed on the received sum.

It is seen, therefore, that the design of the ANN is the specification of the weights. The network essentially performs a highly nonlinear mapping between input data (e.g., signal features) and desired

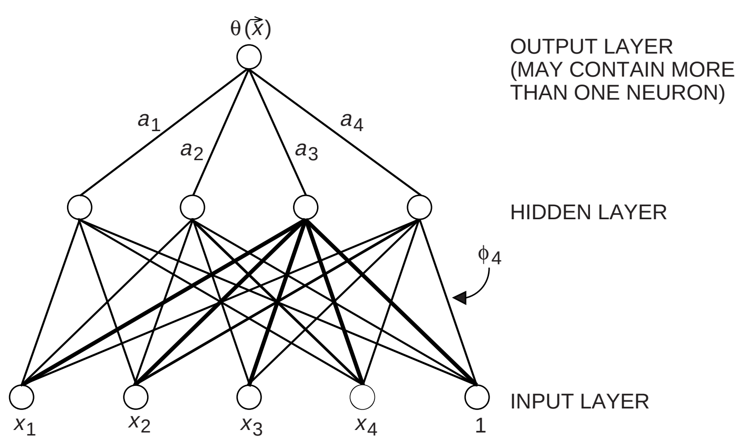

Fig. A-1. Architecture for a feedforward artificial neural network. The weight vector $\vec{w}_3$ is shown by heavier lines. Also shown is the fourth component of the optional bias interconnect, $\vec{\phi}$ (weight vector for unity input).

output (modulation or signal classification). The aim of network training is to find the set of weights for a given input–output data set, such that an error criterion is minimized. Thus, a training set for the ANN is composed of corresponding vectors of input (x_i) and desired output (t_i): $\{(\vec{x}_1, \vec{t}_1), (\vec{x}_2, \vec{t}_2), \cdots, (\vec{x}_N, \vec{t}_N)\}$. The error function is defined as

$$E = \sum_{i=1}^{N} \parallel \vec{t}_i - \vec{\theta}(\vec{x}_i; \vec{w}) \parallel^2$$

where $\vec{\theta}$ represents the nonlinear mapping of the ANN from the input to the output, and $\vec{\theta}(\vec{x}_i; \vec{w})$ is the actual (not desired) value of the network output obtained for a given input vector, $\vec{x}_i$, and weight vector, $\vec{w}_i$. Network training involves a search for a vector of weight values ($\vec{w}$) that minimizes the error, E.

Network design is performed iteratively and concurrently with network training. There are a number of issues that have to be addressed. For instance, the number of nodes, especially at the hidden layer, should be set properly. Different approaches have been studied to optimize the network size and improve generalization. These include selective pruning and weight decay algorithms [7] or Fourier analysis of single hidden-layer perceptrons with linear activation functions for providing insight into the number of hidden-layer nodes to be selected and the proper choice of the node nonlinearity [8].

Another factor that should be considered in network design is speed of training. One should aim to minimize network training time spent off-line or on-line (incremental learning for new, previously unseen input vectors). Previous work has shown the remarkable efficiency of techniques based on fuzzy modeling of network training heuristics that have resulted at times in dramatic speedups in network training [9].

Appendix B

Overview of Fuzzy Set Theory

I. Definitions

Let X denote a universal set known as the universe of discourse. Then a fuzzy subset, A, of X usually is characterized by a membership function, $\mu_A(x)$, which assigns a real number in the closed interval $[0, 1]$ to every element of X. This number $\mu_A(x)$ represents the grade of membership of element x in set A. The membership function thus can be described in terms of the following mapping:

$$\mu_A : X \to [0, 1] \tag{B-1}$$

with larger values of it denoting higher degrees of set membership. A nonfuzzy (crisp) set, therefore, can be viewed as a restricted case of a fuzzy set, where the membership function, μ_A, maps elements of the universe of discourse to the set $\{0, 1\}$.

For example, membership functions representing the meanings of the concepts large negative (LN), medium negative (MN), small negative (SN), zero (ZE), small positive (SP), medium positive (MP), and large positive (LP) are shown in Fig. B-1.

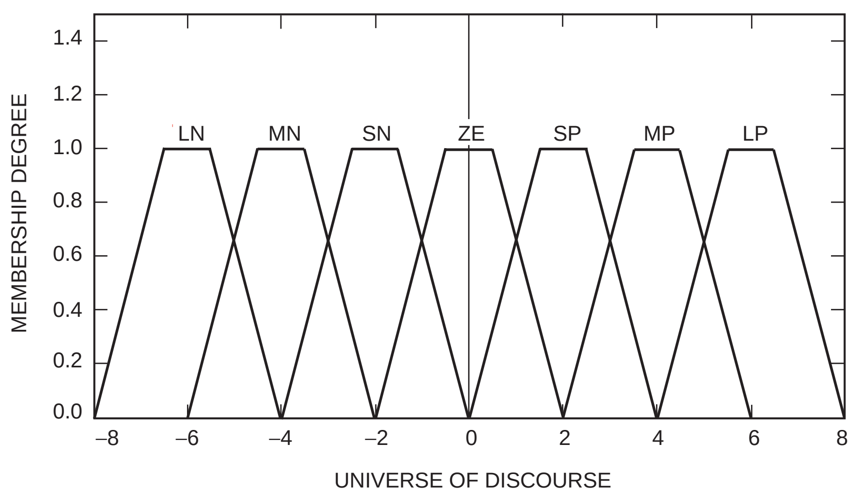

Fig. B-1. Membership functions representing the meanings of the concepts large negative (LN), medium negative (MN), small negative (SN), zero (ZE), small positive (SP), medium positive (MP), and large positive (LP), defined over a continuous-valued universe of discourse [–8,8].

We define a fuzzy variable as a variable that can be described by a number of different fuzzy sets. For instance, if we have a fuzzy variable denoted by height, then it could be described as tall, very tall, not tall, etc. Note that the values that height can take on can be crisp, such as when we say that a person's height is 2 meters. However, a person with that height could be described as tall in a fuzzy way by defining a proper membership function for the concept "tall."

II. Some Basic Operations on Fuzzy Sets

Basic operations related to fuzzy subsets A, B, and C of a universe of discourse X, having membership functions $\mu_A(x)$, $\mu_B(x)$, and $\mu_C(x)$, respectively, where $x \in X$, are summarized here.

Equality:

$$A = B \;\Rightarrow\; \mu_A(x) = \mu_B(x) \qquad \forall x \in X \tag{B-2}$$

Complementation (negation):

$$A = \bar{B} \;\Rightarrow\; \mu_A(x) = \mu_{\bar{B}}(x) = 1 - \mu_B(x) \qquad \forall x \in X \tag{B-3}$$

Containment:

$$A \subseteq B \;\Rightarrow\; \mu_A(x) \leq \mu_B(x) \qquad \forall x \in X \tag{B-4}$$

Union (the connective *or*):[6]

$$C = A \cup B \;\Rightarrow\; \mu_C(x) = \mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)] \qquad \forall x \in X \tag{B-5}$$

Intersection (the connective *and*):[7]

$$C = A \cap B \;\Rightarrow\; \mu_C(x) = \mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)] \qquad \forall x \in X \tag{B-6}$$

III. Fuzzy Inference

Fuzzy inference is based on the concept of the fuzzy conditional statement: IF A THEN B, or for short, $A \Rightarrow B$, where the antecedent A and the consequent B are fuzzy sets. Whereas in classical propositional calculus the expression $A \Rightarrow B$ is interpreted as being equivalent to $\bar{A} + B$ (in the sense that the two statements have the same truth table), a fuzzy conditional statement describes a relation between two fuzzy variables. Consider the following statement:

IF *very far* THEN *expensive.*

which is an abbreviation of the statement

IF a location is *very far* from here, THEN travel to there is *expensive.*

Inference policy of a fuzzy inference system is expressed in terms of a series of rules of the above form. Fuzzy set theory then provides the mathematical formalism to evaluate these rules, assigning degrees of importance to each one, and then combining them to reach a single non-fuzzy decision. Details of the algorithm can be found in [1]. Techniques for adaptive fuzzy inference are covered in [11].

---

[6] Where $\max(a, b) = a(b)$ if $a \geq b(a < b)$.

[7] Where $\min(a, b) = a(b)$ if $a \leq b(a > b)$.