JOURNAL OF PROPULSION AND POWER
Vol. 18, No. 4, July–August 2002

Application of Autoassociative Neural Network on Gas-Path Sensor Data Validation

Pong-Jeu Lu* and Tzu-Cheng Hsu†
National Cheng Kung University, Tainan 70101, Taiwan, Republic of China

Gas-path analysis holds a central position in the engine condition monitoring and fault diagnostics technique. The success of gas-path analysis, as concluded from previous investigations, depends mainly on the quality of the measurements obtained. For approaches using either a classical Kalman filter method or a contemporary artificial neural network approach, a high success rate of diagnosis can only be guaranteed when a correct set of measurement deltas is available. The objective of the present work is to design a genetic autoassociative neural network algorithm that can perform offline sensor data validation simultaneously for noise filtering and bias detection and correction. The neural network-based sensor validation procedure usually suffers from the slow convergence in network training. In addition, the trained network often fails to provide an accurate accommodation when bias error is detected. To remedy these network training and bias accommodation problems, a two-step approach is proposed. The first step is the construction of a noise-filtering and a self-mapping autoassociative neural network based on the backpropagation algorithm. The second step uses an optimization procedure built on top of these noise-filtering and self-mapping nets to perform bias detection and correction. A nongradient genetic algorithm search is employed as the optimization method. It is shown in the present work that effective sensor data validation can be achieved for noise filtering, bias correction, and missing sensor data replacement incurred in the gas-path analysis. This newly developed algorithm can also serve as an intelligent trend detector. A true performance delta and trend change can be identified with no delay, to assure a timely and high-quality engine fault diagnosis.

Nomenclature

b_j	=	bias added in the jth sensor signal
ERR	=	error incurred as data passing decentralized neural networks (DNNs)
f	=	fitness value of genetic algorithm (GA) search
I_s	=	input of the self-mapping autoassociative neural network (AANN)
K	=	noise-level control parameter
$K0$	=	noise-free data, $K = 0$ in Eq. (1)
$K2$	=	noise level, $K = 2$ in Eq. (1)
l_i	=	lower bound of the ith parameter for GA search
M	=	number of sensors
N_i	=	noise of the ith sensor signal
O_d	=	output of DNN
O_s	=	output of self-mapping AANN
r	=	blending factor used in crossover operator
rand	=	uniformly distributed random number between [0, 1]
S_i	=	signal of the ith sensor
U	=	random number bound by $[u, l]$
u_i	=	upper bound of the ith parameter for GA search
X_i	=	ith selected chromosome in the GA population
$\bar{X}_i$	=	offspring of the ith selected chromosome X_i
x	=	random number between [0, 1]
Y_{SFDIA}	=	sensor failure detection, identification, and accomodation (SFDIA) accommodated performance delta
Y_{TRUE}	=	original true performance delta
y_i	=	ranking of the ith selected chromosome
Λ	=	averaged noise-to-signal ratio
Λ_i	=	noise-to-signal ratio of the ith sensor
σ	=	sensor scatter

I. Introduction

ENGINE condition monitoring (ECM) holds a central position in facilitating safe engine operation and effective on-condition maintenance. Since the pioneering work of Urban,[1] gas-path analysis[2–6] has been developed into practical ECM software systems to help realize the objective of on-condition and preventive maintenance. However, to date there still exist practical difficulties for maintenance mechanics to comprehend and use these software systems. On-condition or preventive maintenance, in fact, is not fully enforced, simply because the engine operators still cannot successfully or easily convert the data gathered into useful information.

In the past, gas-path analysis has been developed based on model-based diagnostic methods.[1,2] The success of these methods relies heavily on the mathematical model that represents the engine characteristics and the number and quality of the parameters measured. For systems with insufficient measurement data or corrupted data with high noise or large bias, the diagnosis results often are not reliable.[7] Aside from these basic mathematical problems, the effort required for executing Kalman filter analysis for ECM diagnosis is so large that it tends to discourage model-based methods from being routinely applied by airlines.

Artificial neural network- (ANN-) based techniques are evolving as another approach for diagnosing engine performance problems.[8–10] The major advantages enjoyed by neural network-based methods are 1) they require no mathematical model to define parameter relationships in the ECM problem and 2) they can tolerate imperfect sensor data while diagnosing engine faults. Unlike the model-based approach, the number of sensors, or mathematically the problem of identifiability of the system, is basically immaterial for ANN-based methods. It has been demonstrated that a four-input limited ECM system is not inferior to an eight-input extended counterpart when a neural network is employed as the diagnostic method.[7] How successful the ANN is depends on how good the quality of the fault samples that are provided are. A previous feasibility study[7] of an ANN-based approach indicates that

Received 3 October 2001; revision received 6 February 2002; accepted for publication 6 February 2002.

*Professor, Department of Aeronautics and Astronautics. Senior Member AIAA.

†Ph.D. Student, Department of Aeronautics and Astronautics.

engine fault diagnosis can achieve a 100% success rate, provided that an unbiased and noise-free set of input data is available. Also indicated is that the success rate decreases as the noise level of the sensor data increases. This finding suggests that a sensor data validation process is of paramount importance for the development of ANN-based diagnostic methods. Both training and testing samples ought to be validated before they are inserted into the diagnostic ANN. A validated training data set represents a more correct and consistent input/output mapping relationship, and usually the training of an ANN using a properly correlated data set has a faster rate of convergence. Similarly, in diagnosing the sensed data, it is particularly important to analyze these data first. Otherwise, noise, bias, or other imperfections contained in the sensor data would easily be mistaken for the physical parameter trend change, resulting in an incorrect diagnostic judgment and, hence, a higher possibility of a false alarm. Hence, it is suggested that a reliable ECM diagnostic tool should contain two elements: a sensor validation preprocessor and a fault diagnostic solver, as illustrated in Fig. 1.

In recent sensor validation studies, the autoassociative neural network (AANN) approach has often been employed.[8,11,12] The present AANN method belongs to the so-called nonlinear principal component analysis category.[13,14] Noise contained in the input nodes is eliminated in the mapping part located between the input and the bottleneck layers. The reconstructed data, however, are generated in the subsequent demapping layers beyond the bottleneck layer. AANNs are currently being used in many real-time monitoring or control problems.[15–17] In these safety-critical processes, data must be first validated and/or accommodated (when data are proven to be invalid) before any work is to be done using these data. Although ANN-based diagnostic tools can tolerate a certain degree of corruption or incompletion in data input, logically, a reliable fault diagnostic method should be able to validate its input data prudently, so that a consistent, high-quality diagnosis can be maintained.

The experiences with AANN in sensor data validation show that noise filtering can be achieved with no basic problems, but bias correction encounters some difficulties.[8] In dealing with bias detection and correction, the robust autoassociative neural network (RAANN), which is an extension of the noise-filtering ANN, was proposed. However, it is found that RAANN is hard to train, and the bias error, in general, cannot be accurately accounted for. To overcome these difficulties, the present work proposes a different design philosophy. Bias correction and noise filtering are decoupled, which avoids the dilemma faced by RAANN, namely, too many objectives are assigned for one network. Bias correction is treated as a parameter identification task in the present approach, and this parameter identification is carried out via a global search using a genetic algorithm (GA). AANs are used only for noise filtering and for fitness evaluation of the GA search process. In the subsequent sections, the elements of this proposed genetic AANN algorithm are introduced and explained. Then a validation exercise is performed to demonstrate that sensor errors including noise or different types of bias errors (hard failure, soft failure, and drift error) can all be treated satisfactorily. The significance of the use of this intelligent sensor data validation algorithm in ECM is also discussed. Finally, a conclusion is drawn and some future suggestions are proposed.

II. Fault Sample Generation and Diagnosis

The diagnostic ability of an ANN is gained through a learning process using training samples. Training samples can be generated via numerical simulation or by collecting identified faults from actual maintenance or repair work. The former, namely, by simulating the ECM fault matrix,[2,7] was adopted in the present study. This fault matrix, in the present research, represents the relationship between the module performance parameters and the measurement deltas of a Pratt and Whitney Aircraft, United Technologies Corporation (PW) 4000-94 engine at cruise condition (engine pressure ratio equal to 1.29). Only a single-fault type is considered at a time. In the generation of the sensor data, data scatters representing engine-to-engine variability and measurement nonrepeatability were supplied by a survey results of 48 PW engines maintained in an engine repair station.[7] The formula used herein states that

$$\text{sensor data} = \text{clean data} + K\sigma[\text{rand} - 0.5] \qquad (1)$$

in which clean data are obtained by multiplying the fault matrix with a single-module fault vector. The constant K is a control parameter governing the noise level and σ is the averaged characteristic sensor scatter of actual engines as listed in Table 1. The variable rand stands for a random number between [0, 1] with a uniform probability density distribution.

For simplicity, there are five engine faults (high-pressure compressor performance loss, low-pressure compressor performance loss, high-pressure turbine performance loss, low-pressure turbine performance loss, and fan performance loss) with fault severity ranging from 1 to 5% performance loss assumed in generating the fault samples. Sensor measurements used in the following studies are exhaust gas temperature (EGT), fuel flow (WF), low-rotor speed (N1), and high-rotor speed (N2), respectively. Note that $K = 2$ corresponds to having noise-contaminated measurements comparable to data acquired from the actual engines.

The second term on the right-hand side of Eq. (1) is not the only representation of data nonrepeatability. For example, one may use $K\sigma$ as the standard deviation of a Gaussian distribution of the probability density function for generating the random number. Equation (1) is only used for convenience, and it is clearly understood that the data fluctuations are bounded within the range of $\pm K\sigma/2$. Figures 2 and 3 show the success rates of two diagnostic networks trained, respectively, using samples generated by a uniformly and a Gaussian-distributed random data set. The variable Λ appearing in Figs. 2 and 3 is a measure of the sensor noise level and is defined by

Table 1 Averaged sensor scatters σ (PW4000)

Parameters	Values
EGT	3.81°C
WF	0.74%
N2	0.15%
N1	0.17%

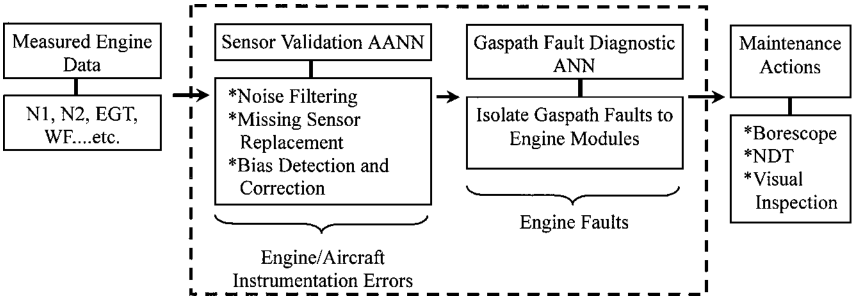

Measured Engine Data
N1, N2, EGT, WF....etc.
Sensor Validation AANN
*Noise Filtering
*Missing Sensor Replacement
*Bias Detection and Correction
Engine/Aircraft Instrumentation Errors
Gaspath Fault Diagnostic ANN
Isolate Gaspath Faults to Engine Modules
Engine Faults
Maintenance Actions
*Borescope
*NDT
*Visual Inspection

Fig. 1 ANN-based fault diagnostic system.

$$\Lambda = \frac{1}{M}\sum_{i=1}^{M}\Lambda_i \tag{2a}$$

$$\Lambda_i = \frac{N_i}{N_i + S_i} \tag{2b}$$

where N and S represent noise and trend change, respectively. The integer M represents the number of sensors, and, presently, we choose $M = 4$ for the four-input network.

There are no qualitative differences existing between these two simulated results. The success rates all decline as the noise level increases. Moreover, Gaussian-distributed noise tends to affect more adversely the success rate because there exist more highly scattered data in the training and testing fault samples. However, the AANN preprocessor, as enforced, is helpful for enhancing the accuracy of diagnosis regardless of what kinds of noise models are used. For this reason, Eq. (1) will be adopted hereafter for generating the noisy data for the subsequent analysis.

III. Genetic AANN Sensor Validation Algorithm

For an engine gas-path fault diagnosis problem, the actual trend change is often of the same order as the sensor bias and the noise level. This situation makes RAANN work only for sensor failure detection and identification, leaving the sensor failure accommodation, when using a neural network-based approach, as a problem yet to be satisfactorily resolved.

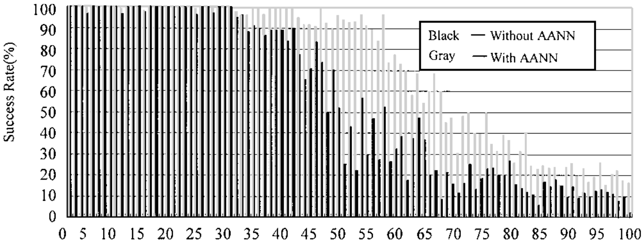


Fig. 2 Success rate vs noise-to-signal ratio for input data with/without AANN filtering (uniformly distributed random data noise model).

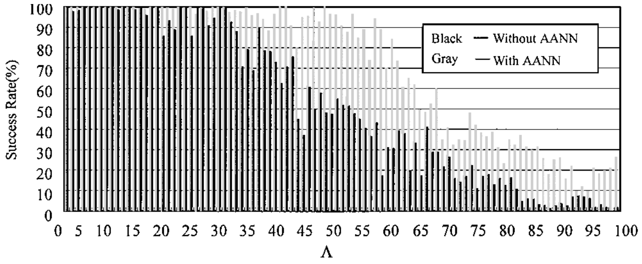


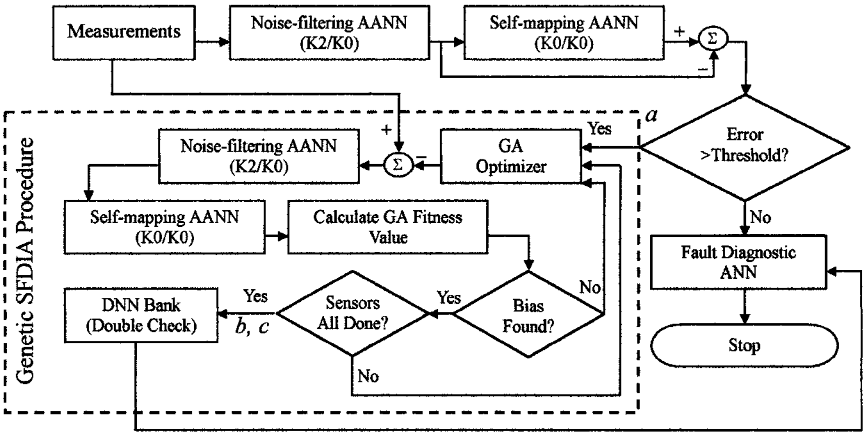
Fig. 3 Success rate vs noise-to-signal ratio for input data with/without AANN filtering (Gaussian-distributed random data noise model).

It has been reported by many researchers that training a RAANN is much more difficult than the training of a noise-filtering AANN. The reason is that we have added an additional requirement (i.e., the elimination of bias error) to the input/output mapping relationship of the neural network under consideration. This convergence problem in training a RAANN is associated with the aforementioned bias elimination capability of a RAANN. To not be limited by this basic drawback associated with RAANN, the present work chooses a strategy that uses noise-filtering AANN only and considers bias detection and accommodation a problem of parameter optimization, as explained in the following.

The present idea in constructing a sensor failure detection, identification, and accommodation (SFDIA) procedure is illustrated in Fig. 4. There are four constituent components that are included in this procedure: 1) a GA optimizer, 2) a noise-filtering AANN, 3) a self-mapping AANN, and 4) a bank of decentralized neural networks (DNNs). The function of the noise-filtering AANN is to remove the sensor data nonrepeatability, thus making the output values, or the input to the subsequent self-mapping AANN, contain mainly the errors induced by the bias effect. The difference between the input/output values of the self-mapping AANN, therefore, is an evaluation of how close the input/output data are. This input/output difference is defined as the fitness function for the GA optimizer placed in front of the noise-filtering AANN shown in Fig. 5. Why such a self-mapping input/output difference can be used as the fitness function for the GA search may be explained as follows. Suppose the correct bias value for a faulty sensor is guessed correctly. The sensor data, as subtracted from their bias value, will encompass trend change plus noise effect only. As these bias-corrected data go through the noise-filtering AANN, the noise effect will be largely eliminated, resulting in a set of data that are intercorrelated almost perfectly with each other. Hence, this corrected data set, when fed into the self-mapping AANN, will map onto itself as evidenced by a very small input/output error norm that is the global minimum the GA optimizer is intended to search for.

Although the GA fitness value can be taken as an index for sensor fault identification, sometimes ambiguities might arise when the fitness value corresponding to certain assumed faulty sensor is not prominently, for example, at least an order of magnitude, lower than the others. DNNs are, therefore, used as a second check to help decide which sensor is indeed faulty. This double-check procedure is a special design feature that can reduce the possibility of a false alarm.

At the present time, a suitable genetic AANN algorithm so developed can perform sensor validation in a sequential manner by searching one-by-one the bias assumed with a candidate sensor. In



a: Sensor Fault Detection b: Sensor Fault Identification c: Sensor Data Accommodation

Fig. 4 Flow chart of genetic AANN sensor validation algorithm.

principle, the GA search procedure may be carried out simultaneously with all of the sensors initially assumed faulty. However, the AANNs trained and used in this optimization procedure are based on the single fault assumption. In other words, there should be only one sensor that is biased as the search procedure truly converges. A simultaneous search should accompany the multiple sensors validation problem. Another practical advantage of a sequential search is the saving of computing time required because the parameter space dimension is only one for a sequential search, resulting in a much faster convergence rate compared to a multiple-parameter search technique.

The block diagram of such a genetic AANN search is illustrated in Fig. 4, for which the search procedure is explained as follows:

1) Detect whether a sensor has failed by feeding the observed measurements into both noise-filtering and self-mapping AANNs and check the error residue. If the error exceeds a certain threshold, the data are deemed faulty and then passed to the neighboring genetic AANN box for SFDIA treatment. Otherwise, the data go to the next fault diagnostic ANN for engine fault isolation.

2) Assume that the ith sensor is faulty, and a population of GA chromosomes is created randomly within the parameter bound defined with the ith sensor.

3) Insert each GA chromosome into the noise-filtering AANN. Data coming out at the output nodes will appear in a form in which the sensor noise is largely filtered out.

4) Feed the output of the noise-filtering AANN into the next self-mapping AANN for fitness evaluation. The closer the suggested bias value is to the true solution, the smaller will be the fitness computed.

5) Go back to step 2 to perform the GA operations reproduction, crossover, and mutation, to generate the next population of chromosomes and repeat these GA evaluations (steps 3 and 4) until a predetermined stop criterion is achieved.

6) Perform GA search from steps 2 to 5 for all of the candidate sensors.

7) Identify the faulty sensor by comparing the lowest fitness values obtained with the assumed candidate sensors. The sensor that is identified as faulty should have a fitness value that is an order of magnitude smaller than those pertaining to the other candidate sensors.

8) Take the outputs of the self-mapping AANN corresponding to the ith GA search as the accommodated ith failed sensor data and feed them into the DNN bank. Check the DNN errors to see whether this ith assumed biased sensor has the lowest error (which should be at least an order of magnitude smaller) when comparing to the other DNN errors.

9) Final SFDIA judgment is decided by comparing the results of steps 7 and 8. If the ith assumed failed sensor passes both the selection criteria of steps 7 and 8, this sensor will be deemed faulty and the corresponding output of the self-mapping AANN is taken as the accommodated sensor data for the subsequent use of engine fault diagnosis.

In the next subsections we will discuss, respectively, the characteristics of each of the aforementioned neural networks and the GA search mechanism used in the optimization procedure. The idea of constructing such a genetic AANN validation procedure will become clearer as the properties of these components are explained and demonstrated.

A. Noise-Filtering AANN

This AANN is trained using a set of fault samples generated by Eq. (1). The noise level is defined using $K = 2$. A single module fault is assumed for each fault type (total of five faults) with deterioration severity ranging from 1.5 to 4.5%. There are a total of 400 sensor data that are generated, in which each fault type contributes 80 samples, spanning uniformly over the deterioration levels.

In the establishment of the weights associated with the noise-filtering AANN, the input nodes are fed with noise-contaminated data ($K = 2$), whereas the output target nodes are noise free ($K = 0$). The structure of this AANN is 4–8–3–8–4, in which three bottleneck nodes are assumed. This AANN is abbreviated as $K2/K0$ AANN in Fig. 4.

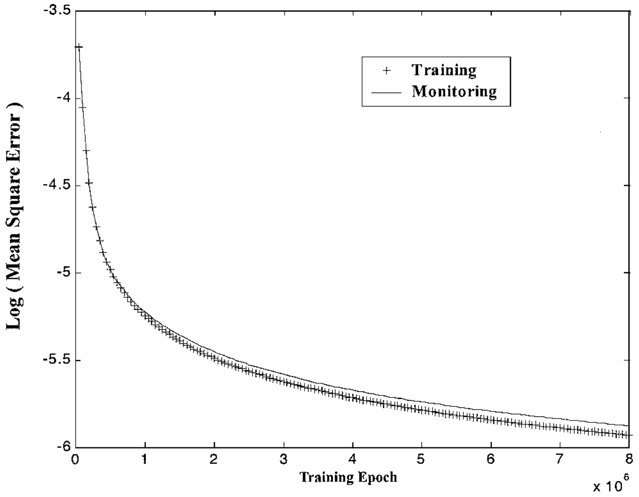


Fig. 5 Convergence histories of noise-filtering AANN.

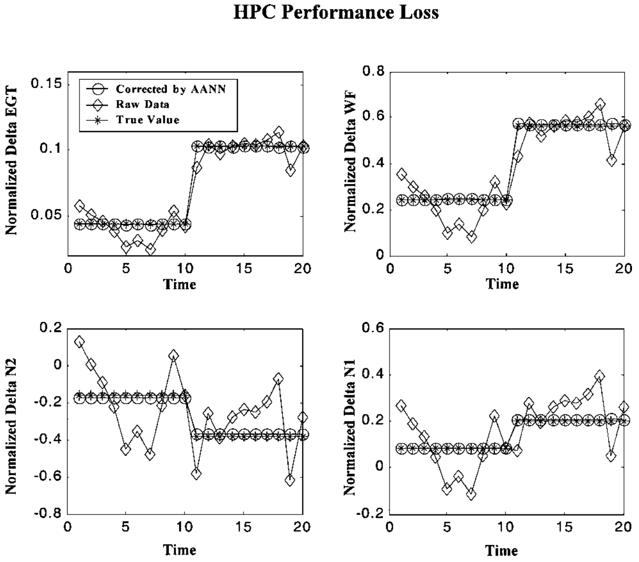


Fig. 6 Noise-filtering capability of AANN.

The convergence histories for training and testing this AANN are illustrated in Fig. 5. Overtraining is monitored while training this noise-filtering AANN. The training stops as the error residue reach a level of 1.16×10^{-6}.

Figure 6 indicates the noise-filtering capability of this AANN. A series of 20 consecutive measurement deltas are tested and presented. A faulty condition occurs at the 11th readout, as evidenced by a trend change appearing in the noise-free data. Noise levels are purposely defined high as compared to the trend changes for N2 and N1 sensors. As observed from the raw data, it is hard to tell whether trend changes occur with N2 and N1 sensors. The filtered data, however, correctly recover the original noise-free values. This demonstrates the noise-filtering ability of the present AANN.

The noise-filtering AANN is then tested with data containing bias error. A representative example is shown in Fig. 7. In this example, an EGT sensor is assumed to have failed with a bias, which is equal to the trend change. The result of this AANN output reveals a smoothed but erroneous temporal distribution for each sensor readout. Sensor nonrepeatability is seen filtered; however, the corrected data do not recover the original true value. For example, the bias of the EGT sensor, as weight summed through the network, affects all output nodes of the AANN, which was trained using nonbiased, noisy samples. It is not surprising to see this behavior because the AANN cannot generalize its mapping ability onto bias correction that was absent from the features possessed by the training samples. The influence of bias, hence, can propagate through the noise-filtering

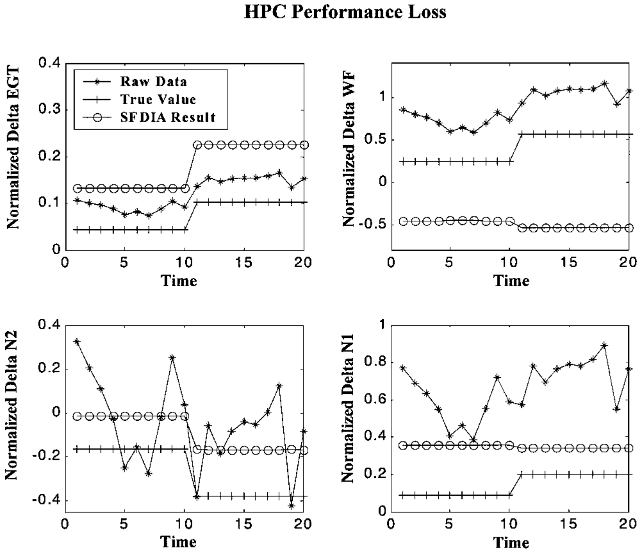


Fig. 7 Influence of bias on noise-filtering AANN.

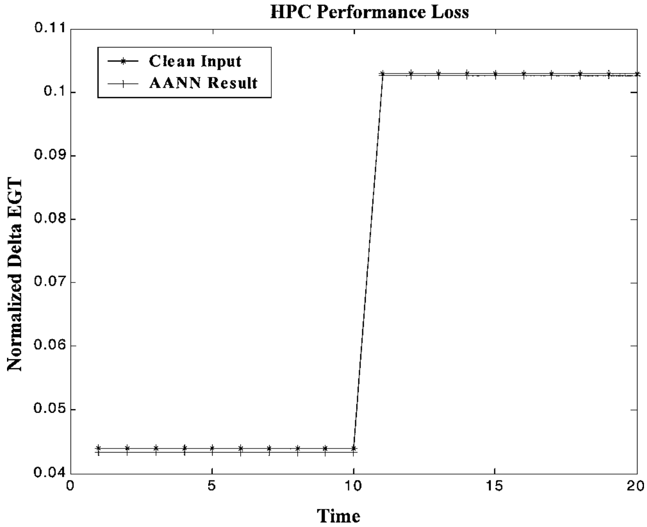


Fig. 8 Identity mapping of self-mapping AANN.

AANN. From the output values shown in Fig. 7, it is observed that the bias effect may sometimes amplify, rather than diminish, when passing through the noise-filtering network.

B. Self-Mapping AANN

Downstream of the noise-filtering AANN is a so-called self-mapping AANN. This particular AANN was trained using clean data ($K = 0$) for both input and output nodes ($K0/K0$ AANN). The network structure has three hidden layers (4–6–3–6–4), and there are 400 samples that were used for the training process. Identity mapping is achieved with this self-mapping AANN, as illustrated in Fig. 8, in which 20 clean samples were tested with very low input/output differences.

This self-mapping AANN has certain bias correction capability. Figures 9 and 10 demonstrate the test results of the bias correction characteristics. A set of data corresponding to a biased EGT sensor was sent through the network. The results computed at the output nodes can almost completely recover the nonbiased values (Fig. 9). A second data set for which an N2 sensor is biased is also checked; however, the bias contained this time can only be partially eliminated as it passes through the self-mapping AANN (Fig. 10). This example indicates that the self-mapping AANN cannot be used alone for bias correction, simply because its performance in eliminating bias effect is not always reliable.

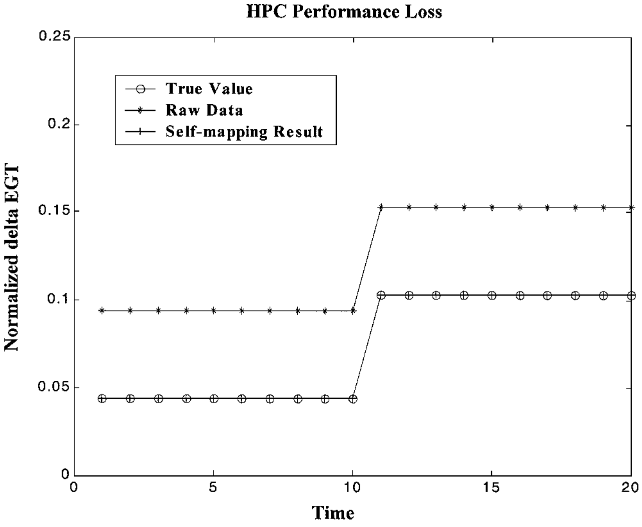


Fig. 9 Bias correction ability of self-mapping AANN (sensor 1 biased).

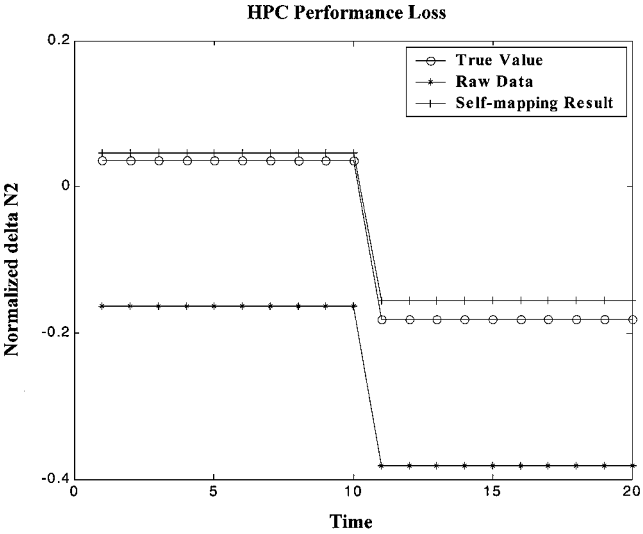


Fig. 10 Bias correction ability of self-mapping AANN (sensor 3 biased).

One can take advantage of this partial self-mapping characteristic to define our fitness function for the GA search:

$$f(b_j) = \sum_{i=1}^{M}[O_s(i) - I_s(i)]^2 \tag{3}$$

in which $O_s(i)$ and $I_s(i)$ are the output and input values corresponding to the ith sensor and M is the total number of sensors. The function $f(b_j)$ represents the fitness of the jth sensor assigned with bias value b_j. Suppose a near-optimal bias is guessed for the correctly selected failed sensor. The output of the noise-filtering AANN will yield an almost, but not exactly, recovered set of sensor data. As this corrected data set enters into the next self-mapping AANN, the fitness will approach its minimum, signifying that the search trend is converging toward the desired true bias value.

Note that the AANN characteristics and the fitness defined for the GA optimization is very critical. Based on our experiences, inappropriate AANN architecture and fitness definition would often lead to incorrect search results.

C. Decentralized ANN

The DNN has been used for fault identification and accommodation.[18] DNN utilizes the redundancy of sensor information and reconstructs the missing sensor data from other available data. Table 2 illustrates the features of the DNNs used in the present work. All of

Table 2 Decentralized ANN definitions for DNN Architecture

Neural network no.	Input			Output	Neural network structure
DNN1	Δ EGT	Δ WF	Δ N2	Δ N1	3-5-5-1
DNN2	Δ EGT	Δ N2	Δ N1	Δ WF	3-5-5-1
DNN3	Δ EGT	Δ WF	Δ N1	Δ N2	3-5-5-1
DNN4	Δ WF	Δ N2	Δ N1	Δ EGT	3-5-5-1

Table 3 Error characteristics of DNN fault identification

Assumed faulty sensor	Candidate faulty sensor			
	EGT	WF	N2	N1
EGT	2.01*e*−3	3.71*e*−2	9.38*e*−2	6.58*e*−2
WF	7.18*e*−2	2.16*e*−3	2.19*e*−1	3.70*e*−2
N2	1.17*e*−1	1.29*e*−1	2.05*e*−3	1.71*e*−2
N1	3.85*e*−2	3.15*e*−2	3.08*e*−2	2.32*e*−3

these DNNs are trained using clean data. Noise effect is not considered because it has already been greatly eliminated in the preceding noise-filtering ANN process.

Error incurred as data passing through this bank of DNNs is, hence, defined as

$$\text{ERR} = \sum_{i=1}^{M} [O_d(i) - O_s(i)]^2 \tag{4}$$

where $O_d(i)$ is the output of the ith DNN and $O_s(i)$ is the ith sensor data coming out of the self-mapping AANN. The summation runs through all of the DNNs defined in the bank. As the GA search takes on a near-optimal bias value for the correctly identified failed sensor, the reconstructed set of data out of the self-mapping and decentralized ANNs in the search process will result in a very small error residue defined by Eq. (4). Otherwise, the influence of bias, produced either by an incorrect guess of the bias value or by an erroneous isolation of the failed sensor, will propagate through the noise-filtering and the self-mapping AANNs as well, yielding a set of relatively poorly intercorrelated inputs to the DNNs and, hence, a larger error residue defined by Eq. (4).

Table 3 indicates the error characteristics of the DNN bank. The diagonal entries of this matrix have error residues that are an order of magnitude smaller than those of the off-diagonal terms. This characteristic, therefore, can be used as an additional fault identification index for the present SFDIA algorithm.

D. GA

For the present optimization problem, there often exist multiple local minima over the parameter bound of the fitness function, as illustrated in Fig. 11. A gradient search may easily lead to an erroneous solution, depending on how the data were initiated. A GA search[19–21] is ideal for a global, rather than a local, search of the optimum of the objective function. Therefore, the GA optimizer is adopted here for the task of bias parameter identification.

Important features of the presently used GA are described in Table 4, and the search space for each parameter is defined in Table 5. Real-valued random population initiation starts the GA evolution. Each generation contains a family of 100 chromosomes. Through the GA operators, namely, reproduction, crossover, and mutation, a next generation is produced. In general, the fittest individuals have a better chance to survive to the next generation. This GA search will stop at a predetermined number of generations, and the parameter corresponding to the fittest value will be taken as the converged solution.

A selection operator is critical for a successful GA evolution. Here, we use tournament selection to generate the parents for offspring reproduction. The upper 60% ranked chromosomes are selected first. These fittest individuals, which are randomly chosen from the fittest pool, will compete with one another. This competition will be done 100 times, and a next generation of 100 winners will be selected to continue the next evolution process. To be specific with this selection method, a group of 200 randomly generated numbers between [1, 60] is first created,

Table 4 Main features of GA

GA parameters	Operators setting
GA type	Real-valued GA
Selection method	Tournament
Crossover method	Blending
Population size	100
Crossover rate	95%
Mutation rate	10%
Tournament selection rate	60 ~ 80%
Blending factor	0.4
Maximum no. of function evaluations	100,000 (1,000 generations)

Table 5 Sensor bias value and GA searching bound

Parameter	Averaged ECM delta	Bias	GA searching bound
Δ EGT	0.04	0.02	[−5.0, 5.0]
Δ WF	0.4	0.2	[−5.0, 5.0]
Δ N2	1.4	0.5	[−5.0, 5.0]
Δ N1	1.1	0.5	[−5.0, 5.0]

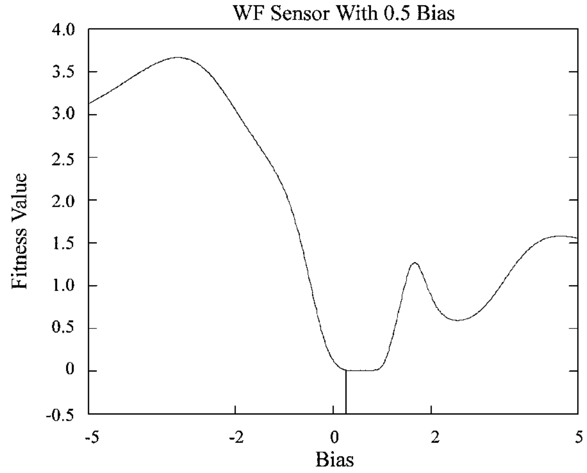


Fig. 11 Fitness value vs bias.

$$y_i = (0.6 \times \text{population-size} - 1) \times x_i + 1, \qquad i = 1, \ldots, 200 \tag{5}$$

in which x_i is a random number between [0, 1]. The integer part of y_i is the ranking of the ith selected chromosome. Competition is then performed with the $(2i-1)$th and the $(2i)$th chromosomes, and the fitter (higher ranking) will survive.

Blending crossover of each two parent X_1 and X_2 chromosomes selected from the tournament selection process is defined here by

$$\bar{X}_1 = r \times X_1 + (1 - r) \times X_2, \qquad \bar{X}_2 = r \times X_2 + (1 - r) \times X_1 \tag{6}$$

to produce the offspring, where variables with an overbar are offspring, and r is the blending factor that could be random or fixed. Here, we choose r as a fixed number, and $r = 0.4$.

Mutation is performed for the randomly selected chromosomes in a population. The mutation ratio here is set to be 10%. For a selected jth chromosome, the mutated new offspring takes on a new parameter:

$$\bar{X}_i = \begin{cases} U(u_i, l_i), & \text{if} \quad i = j \\ X_i, & \text{otherwise} \end{cases} \tag{7}$$

where $U(u_i, l_i)$ is a random number bounded by $[u_i, l_i]$

IV. Sensor Data Validation Tests

In this section we will examine the validity of the presently proposed SFDIA algorithm. First, the detection of sensor error is presented. Then, a bias identification and accommodation will be demonstrated. Sensor errors, including hard failure, soft failure, and drift error will be tested. Finally, a large set of examples consisting of sensor biases with different degrees of failure is to be examined to show the robustness of the present genetic AANN validation algorithm. In all of the cases studied, the sensor data are assumed noisy with data scatters ($K = 2$) comparable to those of the actual engine deltas.

A. Sensor Error Detection

Sensor measurements are first sent into the noise-filtering and self-mapping AANNs for failure detection. The overall fitness defined by Eq. (3) is used as a reference value for claiming whether bias error exists in the sensors. Because the residue of the AANNs exceeds a certain predetermined threshold, the sensor data are taken as unqualified, and error identification and accommodation will be activated before these data are fed into the diagnostic fault isolation ANN.

The sensor failure detection threshold can be determined in accordance with the minimal acceptable success rate of the subsequent fault diagnostic results. Figure 12 shows the success rate of engine fault diagnosis subject to different degrees of sensor bias. The horizontal coordinate is the bias-to-signal ratio where signal refers to the averaged parameter delta. The statistical success rate is computed using 200 samples generated by Eq. (1). It is demonstrated in Fig. 12 that the EGT sensor is most insensitive to the diagnostic result, whereas N2 sensor is most sensitive. Suppose a 90% success rate is the minimal acceptance level, then the error residue of the most influential N2 sensor, corresponding to the success rate no less than 90%, is taken as the threshold for alarming the detection of the sensor failure.

B. Hard Failure Correction

Two types of hard failure are simulated. The first is a complete loss of the sensor data that is represented by a constant of zero throughout the period of data collection. The second hard failure is modeled by a large sudden jump in the observed data. Both of these two hard failures have triggered the error detection alarm. The results after treatment by the present SFDIA algorithm are illustrated in Figs. 13 and 14. There are three curves appearing in Figs. 13 and 14. Raw data refers to the data input into the SFDIA procedure, which contain noise, bias, and trend change. True data are the clean data simulated that have only a trend change and are the target of the present SFDIA optimization. The SFDIA-corrected data are the accommodated data using the present algorithm. It is shown that data can be recovered completely without significant error incurred.

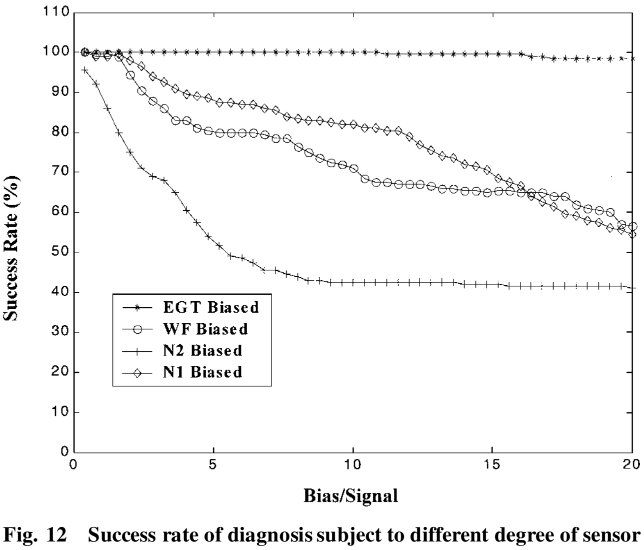


Fig. 12 Success rate of diagnosis subject to different degree of sensor bias.

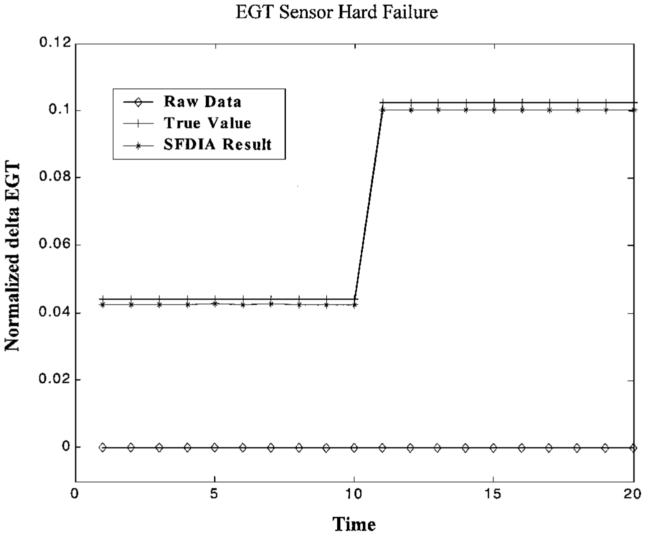


Fig. 13 Sensor hard failure: recovery from a complete loss of sensor data.

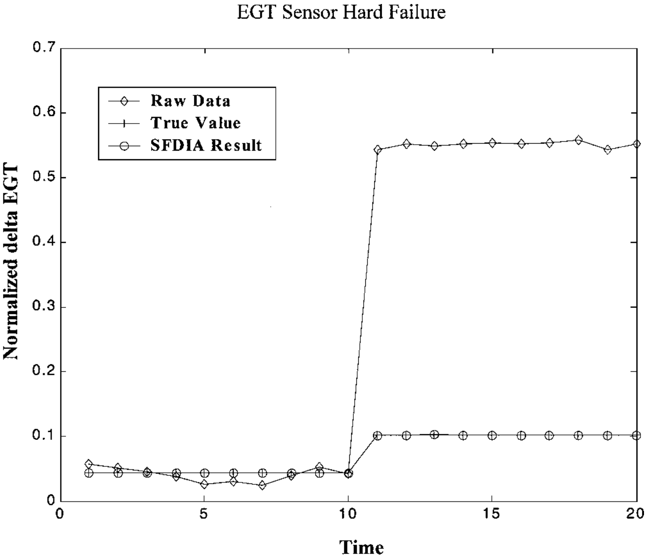


Fig. 14 Sensor hard failure: recovery from a large sudden jump of sensor data.

C. Soft Failure Correction

Usually a soft failure in sensor data is relatively hard to identify when compared to a hard failure. Here, we assume a soft failure in an EGT sensor with a drift of bias varying linearly from 0.003 to 0.06. This migration of sensor error has an amplitude growth from approximately 6.7 to 150% of the averaged parameter delta, and, during the course of sensor drift, a trend change occurs with a jump value of 0.06 appearing at the 11th recording.

The corrected data are shown in Fig. 15. Again, the present validation algorithm has proven to be excellent in recovering the drifting soft failure data. Not only is the bias drift correctly identified, but also the ambiguity in mistaking bias drift for performance trend migration is straightened.

D. Data Recovery Evaluation

To examine the robustness of the present data validation algorithm, a set of test examples are generated. Normal engines parameter scatter ($K = 2$) together with different bias amplitudes ($50 \sim 300\%$ of averaged trend change) are used in the generation of the test examples. For each fixed bias level, there are 200 faults that are generated, spanning uniformly from 1 to 5% performance degradation for each of the five fault types.

The results are shown in Figs. 16a–16d. The vertical coordinate represents the averaged ratio of the recovered data to its original unbiased and noise-free value. The recovery rate (RR) is defined herein by

$$\mathrm{RR} = \left(1 - \frac{|Y_{\mathrm{SFDIA}} - Y_{\mathrm{TRUE}}|}{|Y_{\mathrm{TRUE}}|}\right) \times 100\% \tag{8}$$

where Y_{SFDIA} and Y_{TRUE} represent, respectively, the SFDIA accommodated and the original true performance deltas. Curves appearing in Figs. 16a–16d are the recovery rates corresponding to different candidate sensors that are assumed biased. It is shown in Figs. 16 that a wrong identification of the faulty sensor will lead to a low sensor recovery rate. The present genetic AANN algorithm finds no problem in identifying the faulty sensor. The uniformly high recovery rates demonstrate the accuracy and robustness of the present scheme. It is shown that the SFDIA test has been performed well in terms of noise filtering and sensor bias detection and accommodation for single sensor failure problems.

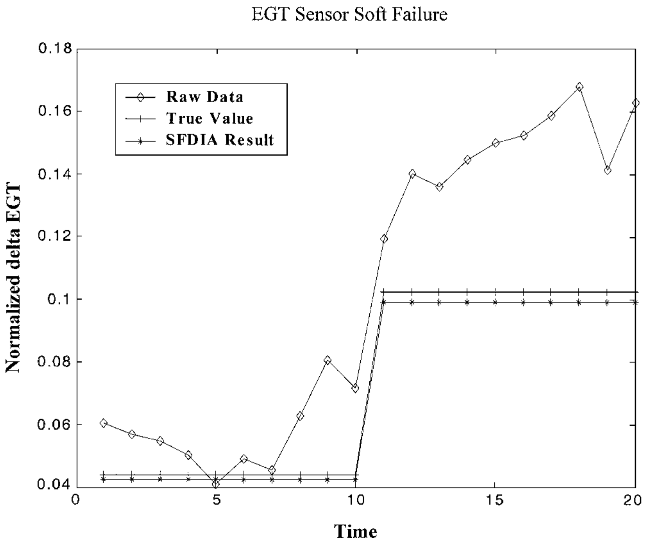


Fig. 15 Sensor soft failure: recovery from a drifting bias of sensor data.

E. Computing Time Evaluation

A GA search is notorious for its long computing time requirements. This is because the search is done in a multiple pointwise manner. The fast computing characteristic of the ANN approach, however, offsets this drawback and reduces the time in evaluating the fitness function. For the cases studied in the preceding subsections, the clock time required for computing a single bias correction case takes approximately 24 s on a Pentium III (450-MHz) processor, for which 3000 generations with each generation having a population size of 100 was considered.

V. Intelligent Trend Detector

In gas-path analysis, the observed measurements are the resultants consisting of several contributing factors as added to the baseline value defined by the manufacturer. Figure 17 illustrates these contributing factors, in which only a portion is associated with the deviant of engine performance. One may attribute these factors to three categories. Category I is the measurement nonrepeatability that is random in nature. Category II accounts for the genuine parameter deltas responsible for the health condition of the gas path. Category III comprises other factors due to sensor bias, installation calibration error, Reynolds number, variable geometry setting deviations (bleed, nozzle, etc.), and baseline shift caused by correction error using inaccurate reference data. To date, how to extract categories I and III from the measurement delta (observed value subtracted from the provided baseline value) is a critical problem not yet solved. Usually, it requires a frequent calibration of sensors and a systematic monitoring and update of the baseline migration.

The presently developed genetic AANN algorithm could be used as an intelligent trend detector and a performance delta sorter. This is because category I can be filtered by AANN, and category III, which can be considered as a generalized sensor bias, can be identified using the present algorithm. This characteristic is particularly valuable for real-life application of the ECM diagnostic techniques.

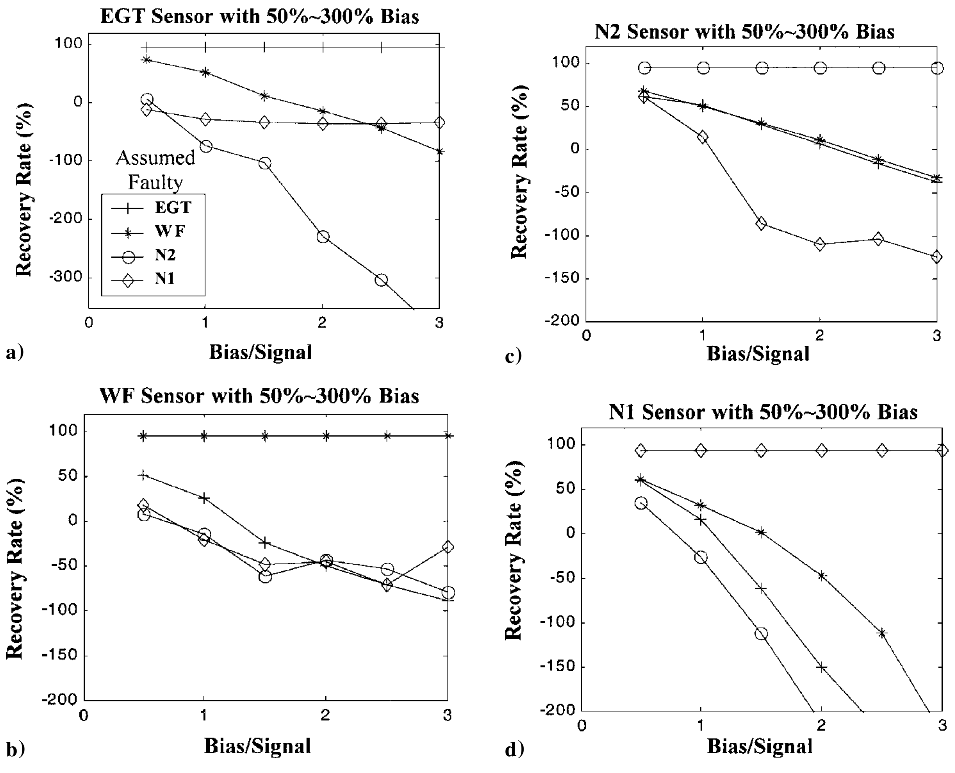


Fig. 16 Sensor recovery rate vs bias amplitude.

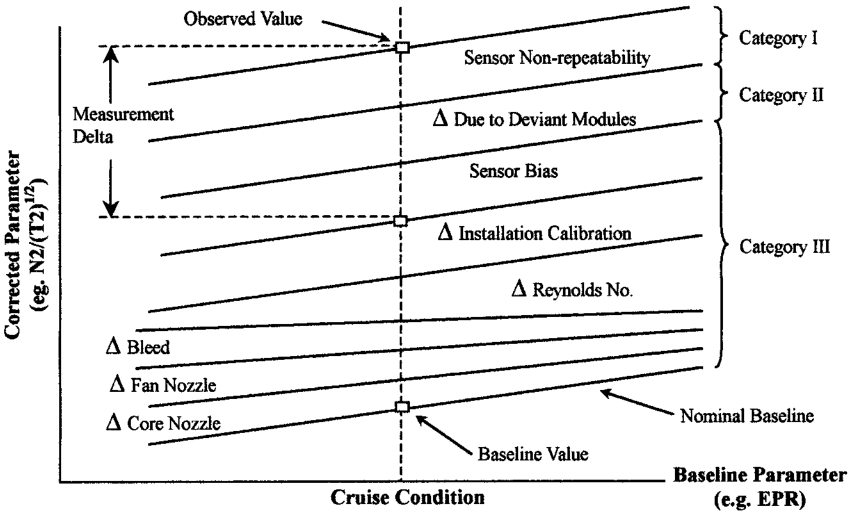


Fig. 17 Contributing factors to observed measurement value.

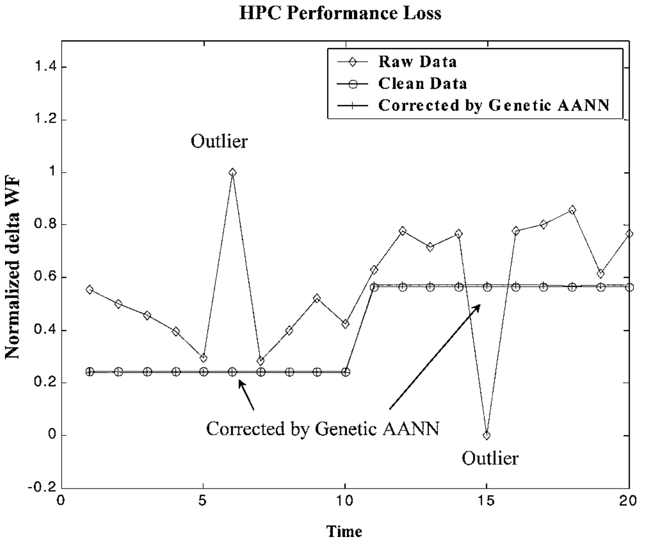


Fig. 18 Data correction and trend detection by genetic AANN algorithm.

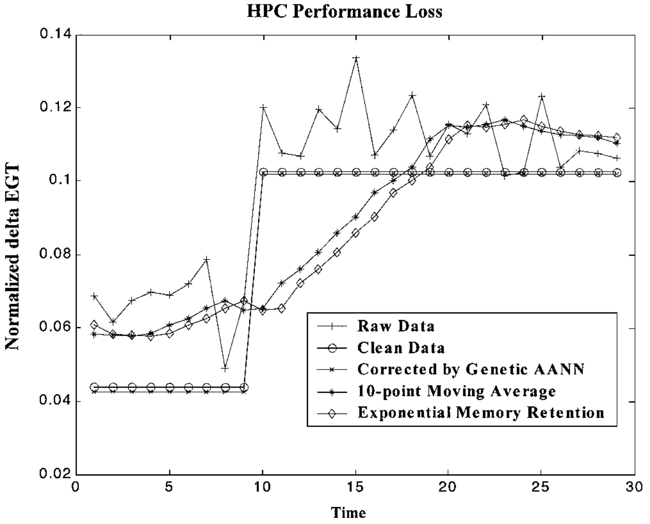


Fig. 19 Comparison of data smoothing methods in trend detection.

Appropriate and timely correction action can be activated soon after a true performance trend change is discovered, and false alarms can, thus, be greatly reduced because only true deltas are used in the diagnostic ANN.

The ability of the present algorithm as an intelligent trend detector is shown in Figs. 18 and 19. A set of consecutive recordings is depicted in Fig. 18, in which the data series contain noise, bias, trend change, and a nonphysical jump value known as outliers. The treated data, however, almost completely recover the original performance delta distribution, and the genuine trend change is correctly identified. Illustrated in Fig. 19 is a comparison of the present data validation method with respect to two conventional smoothing algorithms.[22] Both 10-point moving average and exponential memory retention methods need at least 10 points for the smoothed trend to pick up the jumped value. A hazardous event may happen during this transient pick-up period. Conventional data smoothers are statistical averages that have no idea about what kind of physics are associated with these measured data. However, the present genetic AANN data smoother is trained using educated samples that are generated using the gas-path dynamics. Noise filtering and bias correction are done during the course of searching for an equilibrium point in the intrinsic state-space around which the sensor deltas are properly correlated. In other words, the present genetic AANN algorithm has knowledge about how sensor deltas should be intercorrelated. This explains why they bear the name of intelligent trend detector.

VI. Conclusions

A new offline sensor data validation preprocessor for an engine gas-path diagnostic system is developed. SFIDA can be achieved using a genetic search in conjunction with noise-filtering and self-mapping AANNs. This newly proposed algorithm avoids the fundamental difficulties encountered in constructing the required robust autoassociative network. In the present work, sensor validation works only for single sensor failure on an offline basis. The characteristics associated with the noise-filtering and self-mapping AANNs are explained, and the fitness function defined for the genetic search of the bias quantity is demonstrated as appropriate. A large set of testing samples is inserted into this genetic AANN algorithm for scheme validation, and the results show a high recovery rate over a broad bias range. Moreover, the present algorithm is able to sort out performance-related deltas in the presence of noise, as well as biases of various kinds (soft, hard, and drifting failures).

The present genetic AANN procedure can also serve as an intelligent trend detector. Jump values in the trend can be identified immediately with no delay. Ambiguities in defining measurement deltas can also be straightened using the present data validation method. Deltas arising from installation calibration, Reynolds number, variable geometry settings, and baseline shift may be absorbed into a so-called generalized bias and can be corrected together with the true sensor bias using the present genetic AANN method.

Future work may include 1) evaluation of the present ANN-based diagnostic system using real-engine data, 2) extension of the present offline genetic SFDIA scheme to a real-time or pseudoreal-time basis, and 3) improvement of the present method to cope with multiple sensor failure problems.

Acknowledgments

The present research is supported by National Science Council (NSC) of Taiwan under Contract NSC 89-CS-7-006-006. The authors are particularly grateful for discussions with the Pratt and Whitney Aircraft, United Technologies Corporation collaborators, Hans Depold, Ranjan Ganguli, and Alan Volponi and Chen Daguang and Zhang Jin of Beijing University of Aeronautics and Astronautics.

References

[1] Urban, L. A., “Gas Path Analysis Applied to Turbine Engine Condition Monitoring,” AIAA Paper 72-1082, 1972.

[2] Urban, L. A., and Volponi, A. J., “Mathematical Methods of Relative Engine Performance Diagnostics,” *SAE Technical Paper Series*, Aerotech’92, Anaheim, CA, Oct. 1992, pp. 1–26.

[3] Doel, D. L., “TEMPER-A Gas Path Analysis Tool for Commercial Jet Engines,” *Journal of Engineering for Gas Turbines and Power*, Vol. 116, No. 1, 1994, pp. 82–89.

[4] Doel, D. L., “An Assessment of Weighted Least-Squares-Based Gas Path Analysis,” *Journal of Engineering for Gas Turbines and Power*, Vol. 116, No. 2, 1994, pp. 366–373.

[5] Merrington, G. L., “Fault Diagnosis in Gas Turbines Using a Model Based Technique,” American Society of Mechanical Engineers, Paper ASME 93-GT-13, 1993.

[6] Stamatis, A., Mathioudakis, K., Berios, G., and Papailiou, K., “Jet Engine Fault Detection with Discrete Operating Points Gas Path Analysis,” *Journal of Propulsion and Power*, Vol. 7, No. 6, 1991, pp. 1043–1048.

[7] Lu, P. J., Zhang, M. C., Hsu, T. Z., and Zhang, J., “An Evaluation of Engine Faults Diagnostics Using Artificial Neural Networks,” *Journal of Engineering for Gas Turbines and Power*, Vol. 123, No. 2, 2001, pp. 340–346.

[8] Zedda, M., and Singh, R., “Fault Diagnosis of a Turbine Engine Using Neural Networks: A Quantitative Approach,” AIAA Paper 98-3602, 1998.

[9] Whitehead, B., Kiech, E., and Ali, M., “Rocket Engine Diagnostics Using Neural Networks,” AIAA Paper 90-1892, 1990.

[10] Eustace, R., and Merrington, G., “Fault Diagnosis of Fleet Engines Using Neural Networks,” International Symposium on Air Breathing Engines, Paper 95-7085, 1995.

[11] Mattern, D. L., Jaw, L. C., Guo, T. H., Graham, R., and McCoy, W., “Simulation of an Engine Sensor Validation Scheme Using an Autoassociative Neural Networks,” AIAA Paper 97-2902, 1997.

[12] Mattern, D. L., Jaw, L. C., Guo, T. H., Graham, R., and McCoy, W., “Using Neural Networks for Sensor Validation,” AIAA Paper 98-3547, 1998.

[13] Kramer, M. A., “Non-linear Principal Component Analysis Using Autoassociative Networks,” *AIChE Journal*, Vol. 37, No. 2, 1991, pp. 233–243.

[14] Kramer, M. A., “Autoassociative Neural Networks,” *Computers and Chemical Engineering*, Vol. 16, No. 4, 1992, pp. 313–328.

[15] Romer, M., “Testing of a Real-Time Health Monitoring and Diagnostics System for Gas Turbine Engines,” AIAA Paper 98-3603, 1998.

[16] Hines, J. W., Uhrig, R. E., and Wrest, D. J., “Use of Autoassociative Neural Networks for Signal Validation,” *Journal of Intelligent and Robotic Systems*, Vol. 21, No. 2, 1998, pp. 143–145.

[17] Napolitano, M. R., Windon, D., Casanova, J., and Innocenti, M., “Comparison Between Kalman Filer and Neural Network Approaches for Sensor Validation,” AIAA Paper 96-3894, 1996.

[18] Napolitano, M. R., Neppach, C., Casdorph, V., Naylor, S., Innocenti, M., and Silvestri, G., “Neural-Network-Based Scheme for Sensor Failure Detection, Identification, and Accommodation,” *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 6, 1995, pp. 1280–1286.

[19] Holland, J. H., *Adaptation in Natural and Artificial Systems*, Univ. of Michigan Press, Ann Arbor, MI, 1974.

[20] Goldberg, D. E., *Genetic Algorithm in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989, pp. 2–10.

[21] Hajela, P., “Nongradient Methods in Multidisciplinary Design Optimization: Status and Potential,” *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 255–265.

[22] Depold, H. R., and Gass, F. D., “The Application of Expert Systems and Neural Networks to Gas Turbine Prognostics and Diagnostics,” *Journal of Engineering for Gas Turbines and Power*, Vol. 121, No. 4, 1999, pp. 607–612.