IN-07
51770
P-8

NASA Technical Memorandum 105303

Real-Time Fault Diagnosis for Propulsion Systems

Walter C. Merrill, Ten-Huei Guo, and John C. DeLaat
Lewis Research Center
Cleveland, Ohio

and

Ahmet Duyar
Florida Atlantic University
Boca Raton, Florida

Prepared for the
International Federation of Automatic Control Symposium
on Fault Detection, Supervision and Safety for
Technical Processes – SAFEPROCESS '91
Baden-Baden, Germany, September 10–13, 1991

NASA

E-6650

REAL-TIME FAULT DIAGNOSIS FOR PROPULSION SYSTEMS

Walter C. Merrill, Ten-Huei Guo, John C. DeLaat
NASA Lewis Research Center, Cleveland, Ohio 44135 USA

Ahmet Duyar
Mechanical Engineering Department, Florida Atlantic University, Boca Raton, Florida 33431 USA

ABSTRACT Current research toward real-time fault diagnosis for propulsion systems at the National Aeronautics and Space Administration's Lewis Research Center is described. The research is being applied to both airbreathing and rocket propulsion systems. Topics include fault detection methods including neural networks, system modeling, and real-time implementations.

INTRODUCTION

Motivated by the need for high performance and high reliability aerospace and nuclear power generation systems, many methods have been investigated for improved fault detection. In this paper the topic of discussion is restricted to real-time fault detection where real-time is defined by the interaction of the fault detection logic with control accommodation. Much of the literature in real-time fault detection has been directed to sensors and actuators and has been summarized by Frank[1]. Additionally, fault detection research specifically directed to propulsion systems has been summarized by Merrill[2].

This paper describes current research toward real-time fault diagnosis for propulsion systems at the National Aeronautics and Space Administration's Lewis Research Center. This research is of importance because an analytically-based redundancy approach can result in substantial control system cost and weight savings over a hardware-based redundancy approach in a high performance propulsion system. Research with applications to both airbreathing and rocket propulsion systems are discussed. In particular, this paper first discusses the results of a program[3] to detect sensor faults in a high-performance turbofan engine using an advanced algorithm based on analytical redundancy. Next a program to integrate traditional rocket control concepts with real-time fault diagnostic capabilities[4] is described.

SENSOR FAULT DETECTION

The objective of the Advanced Detection, Isolation, and Accommodation(ADIA) program was to improve demonstrated reliability of digital electronic control systems for turbine engines by detecting, isolating, and accommodating sensor faults using analytical redundancy methods. The ADIA program was organized into four phases: development, implementation, evaluation, and demonstration. In the first three phases the algorithm was designed using advanced filtering and detection methodologies, implemented in microprocessor based hardware, and evaluated using a real-time engine simulation running on a hybrid computer. In the fourth phase the algorithm was demonstrated on a full scale F100 engine in the Lewis Research Center altitude test facility. The test objective was to demonstrate the predicted performance of the ADIA algorithm on realistic hardware over a wide range of engine operating conditions. These conditions included altitude, Mach number, and power variations.

Algorithm Description

The ADIA algorithm detects, isolates, and accommodates sensor faults in an F100 turbofan engine control system. The algorithm incorporates advanced filtering and detection logic and is general enough to be applied to different engines or other types of control systems. The algorithm detects two classes of sensor faults, hard and soft. Hard faults were defined as out-of-range or large bias errors that occur "instantaneously" in the sensed values. Soft faults were defined as small bias errors or drift errors that increase relatively slowly with time. The ADIA algorithm of Figure 1 consists of four elements: (1) hard sensor fault detection and isolation logic; (2) soft sensor fault detection and isolation logic; (3) an accommodation filter; and (4) the switch matrix.

In the normal or unfailed mode of operation, the accommodation filter uses the full set of engine measurements to generate a set of optimal estimates of the measurements. These estimates(Z(t)) were used by the control law. When a sensor fault occurs, the detection logic determines that a fault has occurred. The isolation logic then determines which sensor is faulty. This structural information is passed to the accommodation filter. The accommodation filter now removes the faulty measurement from further consideration. The accommodation filter, however, continues to generate the full set of optimal estimates for the control. Thus the control mode does not have to restructure for any sensor fault. The ADIA algorithm inputs as shown in Figure 1 were the sensed engine output variables, $Z_m(t)$, and the sensed engine input variables, $U_m(t)$. The outputs of the algorithm, the estimates, Z(t), of the measured engine outputs, $Z_m(t)$, were used as input to the proportional part of the control. During normal mode operation, engine measurements were used in the integral control to ensure accurate steady-state operation. When a sensor fault is accommodated, the measurement in the integral control is replaced with the corresponding accommodation filter estimate by reconfiguring the interface switch matrix.

Accommodation Filter. The accommodation filter incorporates an engine model along with a Kalman gain update to generate estimates of the engine states X and the engine outputs Z as follows.

$$\dot{X} = F(X - X_b) + G(U - U_b) + K\epsilon \qquad (1)$$

$$Z = H(X - X_b) + D(U - U_b) + Z_b \qquad (2)$$

$$\epsilon = Z_m - Z \qquad (3)$$

Here the subscript b represents the base point(steady-state engine operating point) and X is the model state vector, U_m the sensed control vector, and Z_m is the sensed output vector. The matrix K is the Kalman gain matrix and ε is the residual vector. The F, G, H, and D matrices were the appropriately dimensioned model system matrices. Their individual matrix elements along with those of K were corrected by the engine inlet conditions E_m and scheduled as nonlinear functions of Z_b. These nonlinear functions allow continuous correction of the model parameters throughout the flight regime. This filter structure is the structure used in the accommodation filter and all the hypothesis filters used in the soft detection and isolation logic. Reconfiguration of the accommodation filter after the detection and isolation of a sensor fault was accomplished by forcing the appropriate residual element to zero. The residuals generated by the accommodation filter were used in the hard fault detection logic.

Hard Fault Detection and Isolation Logic. The hard sensor fault detection and isolation logic is straightforward. To accomplish hard fault detection and isolation the absolute value of each component of the residual vector was compared to its own threshold. If the residual absolute value was greater than the threshold, then a fault was detected and isolated for the sensor corresponding to the residual element. Threshold sizes were based on the standard deviation of the noise on the sensors.

Soft Fault Detection and Isolation Logic. The soft fault detection logic consists of multiple-hypothesis-based testing. Each hypothesis was implemented using a Kalman filter. The soft fault detection/isolation logic structure is shown in Figure 2. A total of six hypothesis filters are shown, one for normal mode operation(H_0) and five for the fault modes(one for each engine output sensor, H_1-H_5). The structure for each hypothesis filter was identical to the accommodation filter(Eqs. (1) to (3)). However, each hypothesis filter uses a different, reduced set of measurements. For example the first hypothesis filter(H_1) uses all of the sensed engine outputs except the first, N1. The second uses all of the sensed outputs except the second, N2, and so on. Thus, each hypothesis filter generates a unique residual vector, ϵ_i. From this residual each hypothesis filter generates a statistic or likelihood based upon a Weighted Sum of Squared Residuals(WSSR). Assuming Gaussian sensor noise, each sample of ε_i has a certain likelihood or probability p_i given by

$$L_i = p_i(\epsilon_i) = ke^{-WSSR_i} \qquad (4)$$

where k is a constant and $WSSR_i = \varepsilon_i^T \Sigma^{-1} \varepsilon_i$ with $\Sigma = diag(\sigma_i^2)$. Here T denotes matrix transposition and σ_i are the sensor noise standard deviations. These standard deviation values scale the residuals to dimensionless quantities that can be summed to form a WSSR. The WSSR statistic was smoothed to remove gross noise effects by a first order lag with a time constant of .1 sec. Mathematically, when the log of the ratio of likelihoods is taken, then

$$LR_i = \log(\frac{L_i}{L_0}) = WSSR_0 - WSSR_i \qquad (5)$$

If the maximum LR_i exceeds the soft fault detection and isolation threshold, then a fault is detected and isolated, and accommodation occurs. If a sensor fault has occurred in N1 for example, all of the hypothesis filters except H_i will be corrupted by the faulty information. Thus each LR_i will be small except for LR_i. Thus, LR_i will be the maximum and it will be compared to the threshold to detect the fault.

Adaptive Threshold. Initially, the soft fault detection/isolation threshold was determined by standard statistical analysis of the residuals to set the confidence level of false alarms and missed detections. Next, the threshold was modified to account for modeling error by simulation analysis. It was soon apparent from initial evaluation studies that transient modeling error was dominant in determining the fixed threshold level. It was also clear that this threshold was too large for desirable steady-state operation. Thus, an adaptive threshold was incorporated to make the algorithm more robust to transient modeling error while maintaining steady-state performance. The adaptive threshold was heuristically determined and consists of two parts. One part is the steady-state detection/isolation threshold which accounts for steady-state, or low frequency modeling error. The second part accounts for the transient, or high frequency modeling error. The adaptive threshold expansion logic enabled the threshold steady-state value to be reduced to 40 percent of its original value which results in an 80 percent reduction in the detection/isolation threshold in steady-state. The adaptive threshold logic is illustrated in Figure 3 for a PLA pulse transient.

Fault Accommodation. For accommodation two separate steps were taken. First, all seven of the filters(the accommodation filter and the six hypothesis filters) were reconfigured to remove the failed sensor from further use in the filters. Second, if a soft fault was detected, the states and estimates of all seven filters were updated to the values of the hypothesis filter which corresponds to the failed sensor.

Detection/Accommodation Performance

The criteria used to evaluate detection, isolation, and accommodation performance were: (1) minimum detectable bias values and drift rates, (2) elapsed time between sensor fault and detection, (3) steady-state performance degradation after fault accommodation, and (4) transient response of the engine to the filter and control reconfiguration resulting from fault accommodation. In the engine test sensor faults were injected directly into the control feedback signal and consequently effected engine operation until detection and accommodation. The engine studied was a full scale, high performance, turbofan engine. The engine was tested in an altitude facility over a wide range of operating conditions. Two general fault types were studied, hard and soft sensor faults.

The first type of sensor fault considered in the demonstration testing was a hard fault. Because hard faults are easily detected, they were examined at only one operating condition, 10 000 ft/Mach 0.6. The ADIA algorithm exhibited excellent hard detection performance at this condition.

The chosen engine performance measure, engine pressure ratio(EPR), is almost linearly related to engine thrust. The change in EPR following the accommodated hard sensor fault was used as a measure of accommodation performance. Here the percent change in engine pressure ratio(EPR)

$$\delta EPR = \frac{100(EPR_{TF} - EPR_{TO})}{EPR_{TO}} \qquad (6)$$

where EPR_{TO} is the steady-state engine pressure ratio before the fault and EPR_{TF} is the steady-state EPR after the fault, is defined to be the change in steady-state engine performance. For the hard fault detection and accommodation experiments, the δEPR results were less than 6 percent in all cases, well below the critical level of 10 percent. In fact most of the performance changes were negligible.

The other type of sensor fault studied was the soft fault. Undetected soft sensor faults, although small in magnitude, may result in degraded or unsafe engine operation. Because of their small size, soft faults were more difficult to detect than hard faults. Two soft fault modes were studied, bias and drift.

The minimum detectable magnitudes of soft sensor bias and drift faults were determined for each of the five sensors considered at each of the operating points defined in the test matrix. The minimum detectable drift magnitudes were determined by finding the smallest detectable drift fault such that a fault was detected approximately 5 seconds after fault inception. In general there was good agreement between those minimum fault detection magnitudes observed in the test and those predicted by the real-time hybrid evaluation. This agreement demonstrates the excellent fidelity of the model used in the algorithm and the simulation used in the evaluation.

The times to detection for the soft bias faults were all less than .1 sec. The steady-state accommodation performance for this class of fault was good. Again percent changes in thrust(EPR) were determined for several operating points demonstrating subsonic and supersonic operation at military and medium power levels. All values obtained were well below the 10 percent critical level except for operating condition 50 000 ft/Mach 1.8 results which show a 12 percent change in thrust for an exhaust nozzle pressure (PT6) sensor fault.

The steady-state accommodation performance results obtained for sensor drift faults were very good with most thrust changes being small and with none larger than the 10 percent level. Shown in Figure 4 is a fan-speed drift fault at the 30 000-ft/Mach 0.9 operating condition at medium power. Here, a fan speed (N1) drift fault of 150 rpm/sec was introduced at 1 sec. Detection and accommodation occurred at 6.5 sec.(5.5 sec after fault initiation) and the engine required about 4.5 sec to return to a steady condition. During this experiment the fan speed and nozzle pressure integral logic was active. The other four output responses show good estimation accuracy and relatively small transient disturbances to the sensor fault accommodation.

Additionally, detection performance for sequential faults was demonstrated. At condition 10 000 ft/Mach 0.6 six different sequences of soft faults were injected into the test bed system at medium power and one was demonstrated at intermediate power. Each of the seven sequences was a different permutation of the five sensors taken four at a time. In each case the algorithm successfully detected and accommodated each sensor fault in the correct order. These tests demonstrate the ability of the algorithm to continue to successfully perform even after most of the sensors have failed.

Finally, a simultaneous soft fault of burner pressure (PT4) and PT6(both failed at the same instant of time) was injected into the engine system. The algorithm, although not designed for this extremely low probability event, successfully detected and accommodated this fault scenario.

Actual Sensor Hardware Fault Detection. During engine testing, two unplanned faults of actual sensor hardware were detected by the ADIA logic. There were no missed detections of sensor hardware faults by the ADIA logic. In the first case the fan discharge temperature thermocouple, which was not explicitly covered by the ADIA logic, failed. This fault was a hard fault. The second sensor hardware fault was associated with the fan turbine intermediate temperature measurement and was a soft fault associated with the signal conditioning amplifier for the sensor. In each case the algorithm was able to detect the fault.

Transient Performance. Two power level (PLA) transient experiments were used to further demonstrate the successful accommodation, or post-fault performance, of sensor faults. The first experiment consisted of injecting, detecting, and accommodating a single sensor fault and then commanding a PLA pulse transient. Engine performance with this accommodated failed sensor was compared to normal mode engine performance. Eighteen of these single fault PLA pulse tests were performed at 5 different operating points with excellent results.

The second accommodation performance experimênt demonstrated the excellent accuracy of the engine model. In this experiment all the engine sensors were failed and accommodated. Then, the engine was commanded to respond to a PLA pulse transient. For the condition 10 000 ft/Mach 0.6 experiment, N1 and PT6 results are shown in Figure 5. Here excellent performance was demonstrated. Little or no overshoot was observed and engine steady-state performance was good. This demonstrates the capability of safe, predictable engine operation without any sensed engine output information over a slightly restricted power range. The fluctuations in PT6 at high power were caused by an airflow interaction between the facility and the engine.

ADIA Summary

Sensor fault detection and accommodation were demonstrated at eleven different operating points which included subsonic and supersonic conditions and medium and high power operation. The minimum detectable fault magnitudes represent excellent algorithm performance and compare closely to values predicted by simulation. Accommodation performance was excellent. Transient engine operation over the full power range with single sensors failed and accommodated was successfully demonstrated. Open loop engine operation(all engine output sensors failed and accommodated) over at least 75 percent of the power range was also demonstrated at two different operating conditions. Engine operation with only one sensor operational(fan speed) was demonstrated at one operating condition.

The algorithm is implementable in a realistic environment and in an update interval consistent with stable engine operation. Off-the-shelf microprocessor based hardware and straightforward programming procedures, including FORTRAN and floating point arithmetic, can be used. Parallel processing was used as an effective approach to achieving a real-time implementation using off-the-shelf computer resources.

INTELLIGENT FAULT DIAGNOSIS

Intelligent Control Systems (ICS) are defined as those control systems which integrate traditional control concepts with real-time fault diagnostic and prognostic capabilities[4]. Currently, the Lewis ICS research program is developing a real-time diagnostic capability for reusable rocket engines, specifically the Space Shuttle Main Engine (SSME). Here, diagnostics is defined as the detection of a fault, the isolation of the fault to a specific component, and the estimation of the severity and likelihood of the fault. Critical engine components as well as sensors will be monitored for faults. A framework of the ICS, given in Figure 6, clearly shows the interaction of the control with the diagnostic subsystems. Coordinator subsystems arbitrate the potentially conflicting objectives of the life extension control modes, which accommodate engine faults, and the overall mission and engine performance objectives. The multilevel implementation architecture for the diagnostic system[5] shown in Figure 7 exploits parallelism to achieve high data rates at the condition monitoring and fault detection layers. Each of the layers will now be described with some discussion about current research associated with each of the layers.

1. Sensor Layer

The Sensor layer includes all of the instrumentation required by the diagnostic system. Measurements obtained in this layer are communicated to the Condition Monitoring layer. Efforts to develop special purpose sensors for improved engine diagnostic capabilities is an ongoing research area[6].

2. Condition Monitoring (CM) Layer

The Condition Monitoring layer in the diagnostic hierarchy is responsible for signal processing, signal conditioning, feature extraction, and other necessary data abstraction for decision making purposes. The extracted features needed are defined by the experts according to the selected fault modes to be detected. Typical condition monitoring functions are: trending of a selected sensor over a specific time period, FFT processing of bearing deflectometer data, and calculation of functions of measurements, pump suction specific speed, for example, and model-based-residual generation[7].

3. Fault Mode Detection (FMD) Layer

The Fault Mode Detection layer responds to those conditions which indicate the possibility of a defined fault mode. Each detector module attempts to find conditions of a single fault mode. The emphasis at this layer is to minimize missed detections even at the possible expense of increased false alarms. As discussed in the next section, the diagnostic expert system (DES) layer will address eliminating false alarms. Relevant information that will be transferred to the DES layer includes an estimate of the probability and severity of the detected fault mode.

If several fault modes are extremely similar and distinguishing them is either impossible or unnecessary at this stage then a single module will be used to find conditions indicating the group. Similar fault modes which must be distinguished will be resolved by the next layer in the hierarchy. The FMD layer contains the first layer of diagnostic knowledge. It uses the compressed sensor information obtained from the condition monitors to make the first diagnostic detection of fault modes. An example fault mode of this module is Turbine Interstage Seal Wear which can be deduced from CM layer outputs: Turbine Discharge Temperature, Pump Shaft Speed, Pump Head vs. Flow Coefficient, and Rotor Dynamic Instability Indicator. Another example is the diagnosis of actuator faults using neural networks[8].

4. Diagnostic Expert System (DES) Layer

The diagnostic expert system serves as the high layer health monitor. This layer resolves any conflicting reports from the fault mode detectors, determines the priorities of the detected fault modes, does the health status assessment of the overall system, and issues the status report to the intelligent coordination control layer (see Figure 6). This layer also monitors the dynamic status (operating condition) of the system which may play an important part in determining engine health. The DES has to gather all the information, including the probability of detection and the severity of the detected modes and then decide on the status of the system. The probability of the correctness of this decision, as well as the severity layer of the fault mode are transmitted to the coordination layer. The ICS coordinator, based on current mission status, then decides what, if any, accommodation strategy to employ.

4.a Emergency Logic (EL) subsystem

It is recognized that many fault modes will result in rocket engine shutdown. Many of these situations require a nearly immediate engine shutdown to prevent further damage to the engine. The decentralized functionality and parallelism of the expected implementation of the diagnostic system will provide substantial computational throughput. However, to account for those situations where a direct, irrevocable engine shutdown is required, an Emergency Logic subsystem is included in the diagnostic system.

Example Diagnostic System

As an example of the multilevel diagnostic system just described, a two level, neural network based architecture has been developed to detect and diagnose engine actuator faults in the SSME. The two level architecture would be representative of the Fault Detection Layer of the diagnostic architecture and would be driven by model-based residuals generated at the Condition Monitor Layer. The residual generation process is described first.

Residual Generation

Typically, complete nonlinear dynamic simulations of propulsion system (SSME) performance are available. However, due to their size and complexity (40 min. CPU time for 20 sec. of real time operation with a VAX 8800 for the SSME), these nonlinear simulations cannot be used to generate residuals in real time to describe the normal mode of operation. Alternative approaches, such as the linked linear model approach, have been applied to accurately, yet simply, model the performance of the SSME[9]. Here a system identification algorithm and the data generated from the nonlinear performance simulation are used to obtain linear models of the SSME at twenty five different operating points. The inputs of these models are the rotary motion of the valve actuator outputs of the oxidizer preburner oxidizer valve (OPOV), β_{OPOV}, and fuel preburner oxidizer valve (FPOV), β_{FPOV}. The models have measurable state variables which simplifies the model structure. The outputs which are also the state variables, are the chamber inlet pressure, P_C, mixture ratio, MR, high pressure fuel turbine speed, S_{HPFT}, and high pressure oxidizer turbine speed, S_{HPOT}. The linear models, which predict the output of the nonlinear simulation with very good accuracy, are linked to obtain a simplified, quasi-linear model of the SSME, valid within its full range of operation. The coefficients of the linear models are regressed with the parameters which determine the nominal operating conditions, mixture ratio and the chamber pressure. A comparison of the responses of the linked model and the nonlinear simulation indicate good agreement as shown in Figure 8.

With the observed state variables, residuals are generated between the measured output and the output obtained using the state-space model as:

$$residuals = y_{measurement} - y_{model} \qquad (7)$$

These residuals are generated by inducing stuck valve faults in the nonlinear dynamic simulation of the SSME. Both the OPOV and the FPOV are considered for this purpose. In this paper a stuck fault is defined such that a valve may not move above a certain angle, called the fault severity. However, the valve may move as commanded below this angle. Data are obtained at various stuck fault severity angles. Using the residuals generated by the Condition Monitoring Layer, a two layer, neural network based architecture has been developed to detect and diagnose engine actuator faults.

Neural Network Architecture

The Neural Network architecture employed in this study, as shown in Figure 9 is a two level architecture:

1) The Classifier level where the faults are actually classified as belonging to a particular category (fault detection);

2) The Severity level where the severity(magnitude) of the fault that was identified in the Classifier level is estimated.

Classifier Level

The Classifier level consists of two networks, one associated with the chamber pressure residual and the other with the mixture ratio residual. Each of these two networks are three layer (including the input and output layers) feedforward networks with non-linear hidden and output units. The weights in these networks are assigned using the generalized back propagation algorithm.

Each of these networks consist of 200 input nodes, 20 hidden nodes and 2 output nodes. The input to one network is a time sequence of the chamber pressure residual of length 200. The time step between residuals is 0.04 seconds with the total sequence time representing 8 secs. Similarly the other network receives the mixture ratio residual sequence as its input. For each of these two networks, one output node is associated with the OPOV stuck condition while the other output node corresponds to the FPOV stuck condition. In short, one output node is activated if an OPOV stuck condition is activated; the other is activated if an FPOV stuck condition is activated. The output activations are real numbers between 0 and 1.

Training:

For network training, six fault scenarios were generated from the non-linear dynamic simulation for the following fault conditions:

a) The OPOV valve stuck at 45, 47 and 50 degrees respectively.

b) The FPOV valve stuck at 53, 57 and 59.5 degrees respectively.

Chamber pressure and mixture ratio residuals for these scenarios were generated as in Equation (7) for a time span of 8 seconds in steps of 0.04 seconds (a 200 length sequence). That is, the residual data are generated as the difference of the actual output of the SSME non-linear simulation and the output generated by the linked model.

During training, a residual pattern representing a fault condition is applied to the input level (200 nodes) and a 1 (indicating full activation) is applied to the corresponding output node. For instance, the chamber pressure residual corresponding to an OPOV stuck valve condition is applied to one of the Classifier networks and an activation of 1 is imposed on the output node corresponding to the OPOV stuck condition. The network weights are then adjusted invoking the back propagation algorithm, thus enabling the neural network to learn the imposed input-output pattern. Each of the Classifier networks is trained using all six fault scenarios.

Severity Level

The Severity level consists of four (4) networks, two associated with the chamber pressure residual (one for OPOV severity and the other for FPOV severity) and the other two with the mixture ratio residual. Once again, each of these networks is a three layer feedforward network where the weights are assigned using the back propagation algorithm.

Each of these networks consist of 200 input nodes, 20 hidden nodes and 3 output nodes. Two of these networks receive the chamber pressure residual sequence as its input, while the other two networks receive the mixture ratio residual sequence as its input. The three output nodes correspond to the three severity levels (OPOV stuck at 45, 47 and 50 or FPOV stuck at 53, 57 and 59.5 degrees).

Training:

The training is similar to that used in the Classifier level. Once the input pattern is applied to a network, the node corresponding to the severity level of the input pattern is fully activated and the network weights learned through back propagation. For instance, the chamber pressure residual, corresponding to the 45 degrees OPOV stuck valve fault scenario, is applied as the input and the output node corresponding to a 45 degree severity level is given an activation of 1. However, for the Severity level networks, only those residual sequences that correspond to the appropriate network are used to train the network. For example only mixture ratio residual sequences are used to train the two mixture ratio severity networks. As a result, each of these Severity level networks is trained with three input representations rather than six as in the Classifier level.

Results

Test data with severity levels not used in training were used to test both the Classifier level and the Severity level networks. The network architecture woiks as follows. Consider a fault scenario of the OPOV valve stuck at 47.5 degrees. The two Classifier level networks use their corresponding inputs (one uses the chamber pressure residual while the other uses the mixture ratio residual) to give an output activation corresponding to the fault (in this case the output node corresponding to the OPOV fault condition is activated in each of these two networks). Figure 10 illustrates the results obtained for this case.

Once the decision has been made as to what the fault is, the Severity level networks are used to detect the severity of the fault. With the OPOV stuck condition, 2 Severity level networks swing into action (each corresponding to one of the two residuals) to estimate the severity of the OPOV stuck valve. The other two Severity level networks are dormant as they are trained to estimate the FPOV fault severity levels.

Consider another test case for the fault scenario of the FPOV valve stuck at 55 degrees. Again the network architecture would woik as in the previous case with the outputs of the Classifier and Severity level networks shown in Figure 11. Note that in this case the network actually approximates the severity level since the weighted sum of the output activation magnitudes is 55.5. In both cases the networks correctly identify both the fault types and their severity.

CONCLUSIONS

This paper discusses some of the ongoing research in fault detection and diagnosis for aerospace propulsion systems sponsored by the National Aeronautics and Space Administration's Lewis Research Center. The research reported is grouped in two general areas, sensor fault detection and intelligent fault diagnosis.

Sensor fault detection and accommodation for a high performance turbofan engine was demonstrated at eleven different operating points which included subsonic and supersonic conditions and medium and high power operation. Excellent detection and accommodation performance was shown. Transient engine operation over the full power range with single sensors failed and accommodated was successfully demonstrated. Open loop engine operation(all engine output sensors failed and accommodated) over at least 75 percent of the power range was also demonstrated at two different operating conditions.

In the area of intellignet fault diagnosis a proposed architecture for real-time, decentralized fault diagnosis was proposed. Within this four layer architecture research into model based fault detection and diagnosis using neural networks was presented. This included a design of a model based fault detection and diagnosis system for the space shuttle main engine. The engine is modeled using a discrete time, quasi-linear state-space representation whose model parameters are determined by identification. Residuals generated from the model are used by a neural network to detect and diagnose engine component faults. Fault diagnosis is accomplished by training the neural network to recognize the pattern of the respective fault signatures. Preliminary results obtained from a nonlinear dynamic simulation of the space shuttle main engine for two failed oxidizer valve scenarios were presented and indicate that the developed approach can be used for fault detection and diagnosis. Unequivocal classifications of fault type were obtained along with accurate estimation of fault severity for scenarios not included in the training set.

REFERENCES

1. Frank, P. M.: Fault Diagnosis in Dynamic Systems Using Analytical and Knowledge-based Redundancy-A Survey and Some New Results. Automatica, Vol. 26, No. 3, pp 459-474, 1990.

2. Merrill, Walter C.: Sensor Failure Detection for Jet Engines. *Control and Dynamic Systems - Advances in Aerospace Systems Dynamics and Control Systems*, Vol. 33, Academic Press, Inc., 1990.

3. Merrill, Walter C.; DeLaat John C.; Kroszkewicz, Steven M.; and Abdelwahab, Mahmood: Advanced Detection, Isolation, and Accommodation of Sensor Failures - Engine Demonstration. NASA TP 2836, July 1988.

4. Merrill, Walter C. and Lorenzo, Carl F.: A Reusable Rocket Engine Intelligent Control. NASA TM 100963, AIAA-88-3114, July 11-13, 1988. Presented at the 1988 Joint Propulsion Conference, Boston, MA.

5. Guo, Ten-Huei and Merrill, Walter C.: A Framework for Real-Time Rocket Engine Diagnostics. Presented at the

1990 Conference on Advanced Earth-to-Orbit Propulsion Technology.

6. Advanced Earth-to Orbit Propulsion Technology 1990, NASA Conference Publication 3092, May 1990.

7. Duyar, Ahmet; Guo, Ten-Huei; and Merrill, Walter C.: Identification of Space Shuttle Main Engine Dynamics. IEEE Control Systems Magazine, Vol. 10, No. 4, pp 59-65, June 1990.

8. Duyar, Ahmet; and Merrill, Walter C.: A Failure Diagnosis System Based on a Neural Network Classifier for the Space Shuttle Main Engine. IEEE Conf. on Decision and Control, Dec. 1990.

9. Duyar, A., Eldem, V., Merrill, W. C. and Guo, T.-H., "A Simplified Dynamic Model of the Space Shuttle Main Engine," Submitted to ASME Journal of Dynamic Systems Measurement and Control, June 1990.

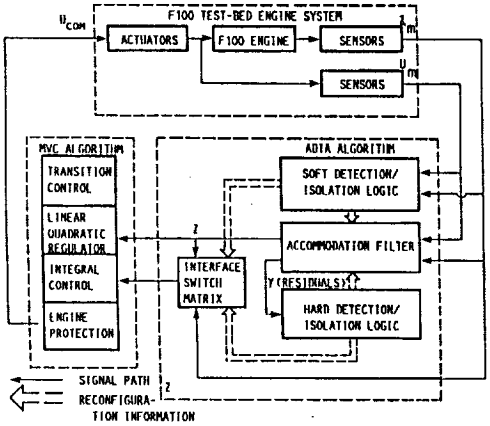

Figure 1 Advanced failure detection, isolation, and accommodation algorithm

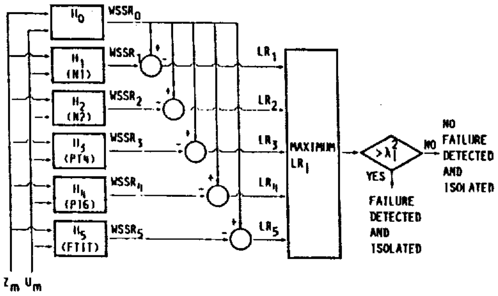

Figure 2 Soft fault detection and isolation logic

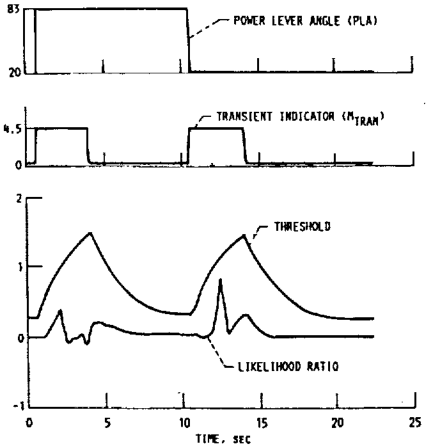

Figure 3 Soft fault adaptive threshold

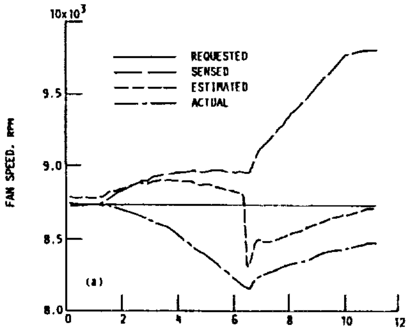

Figure 4 Fan speed drift fault detection results

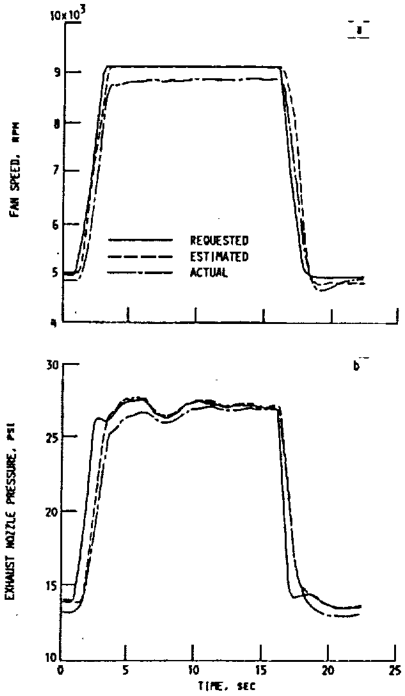

Figure 5 Open-loop engine transient response

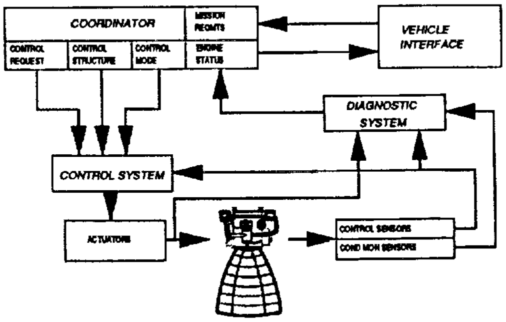

Figure 6 Intelligent Control System framework

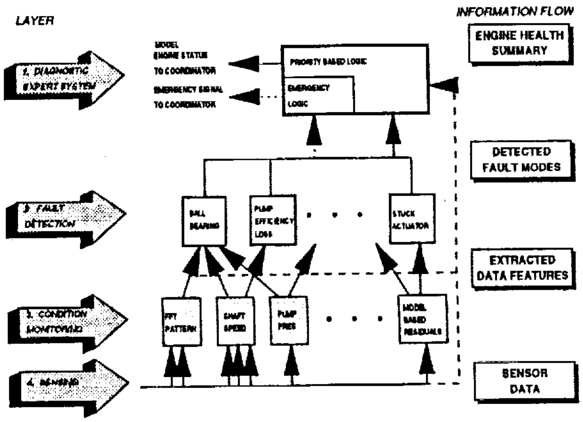

Figure 7 Real time diagnostic system architecture

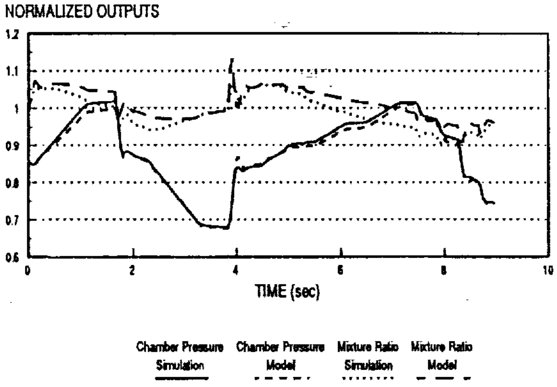

Figure 8 Linked model accuracy results.

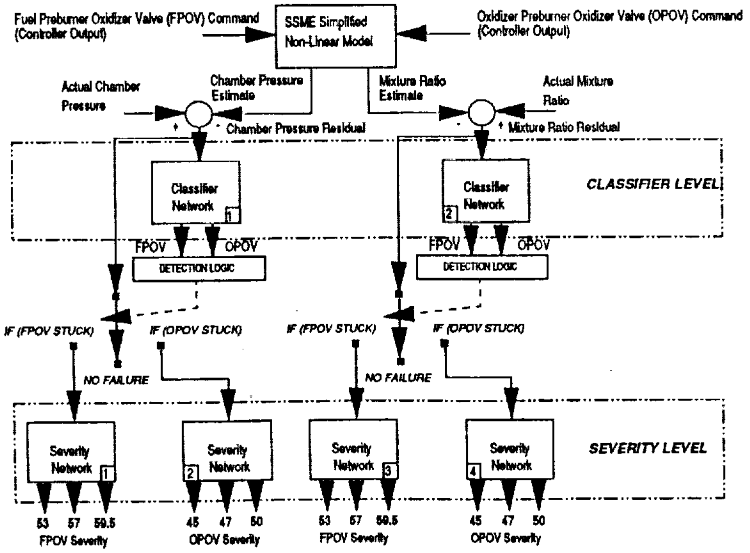

Figure 9 Neural network classifier architecture

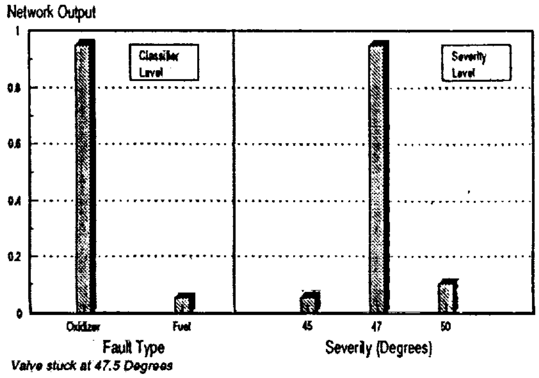

Figure 10 Neural network classifier results for OPOV actuator fault

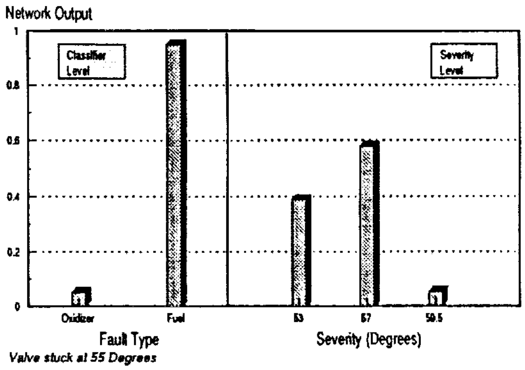

Figure 11 Neural network classifier results for FPOV actuator fault

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	**2. REPORT DATE** 1991	**3. REPORT TYPE AND DATES COVERED** Technical Memorandum

4. TITLE AND SUBTITLE Real-Time Fault Diagnosis for Propulsion Systems	**5. FUNDING NUMBERS** WU-582-01-11
6. AUTHOR(S) Walter C. Merrill, Ten-Huei Guo, John C. DeLaat, and Ahmet Duyar	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191	**8. PERFORMING ORGANIZATION REPORT NUMBER** E-6650
9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, D.C. 20546-0001	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER** NASA TM-105303

11. SUPPLEMENTARY NOTES

Prepared for the International Federation of Automatic Control Symposium on Fault Detection, Supervision and Safety for Technical Processes – SAFEPROCESS '91, Baden-Baden, Germany, September 10-13, 1991. Walter C. Merrill, Ten-Huei Guo, and John C. DeLaat, NASA Lewis Research Center; Ahmet Duyar, Florida Atlantic University, Mechanical Engineering Department, Boca Raton, Florida 33431. Responsible person, Walter C. Merrill, (216) 433-6328.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Category 07	**12b. DISTRIBUTION CODE**

13. ABSTRACT ***(Maximum 200 words)***

Current research toward real-time fault diagnosis for propulsion systems at the National Aeronautics and Space Administration's Lewis Research Center is described. The research is being applied to both airbreathing and rocket propulsion systems. Topics include fault detection methods including neural networks, system modeling, and real-time implementations.

14. SUBJECT TERMS Fault detection; Diagnostics; Propulsion systems; Soft fault	**15. NUMBER OF PAGES** 8
	16. PRICE CODE A02

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT**