NASA Technical Memorandum 89880
AIAA-87-2259

Full-Scale Engine Demonstration of an Advanced Sensor Failure Detection, Isolation, and Accommodation Algorithm—Preliminary Results

(NASA-TM-89880) FULL-SCALE ENGINE DEMONSTRATION OF AN ADVANCED SENSOR FAILURE DETECTION, ISOLATION AND ACCOMMODATION ALGORITHM: PRELIMINARY RESULTS (NASA) 14 p Avail: NTIS HC A02/MF A01 CSCL 09C H1/33
N87-22097
Unclas
0072112

Walter C. Merrill, John C. DeLaat, Steven M. Kroszkewicz,
and Mahmood Abdelwahab
Lewis Research Center
Cleveland, Ohio

Prepared for the
Guidance, Navigation, and Control Conference
sponsored by the American Institute of Aeronautics and Astronautics
Monterey, California, August 17-19, 1987

NASA

FULL-SCALE ENGINE DEMONSTRATION OF AN ADVANCED SENSOR FAILURE DETECTION, ISOLATION, AND ACCOMMODATION ALGORITHM - PRELIMINARY RESULTS

Walter C. Merrill, John C. DeLaat, Steven M. Kroszkewicz, and Mahmood Abdelwahab
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

Abstract

The objective of the advanced detection, isolation, and accommodation (ADIA) program is to improve the overall demonstrated reliability of digital electronic control systems for turbine engines. For this purpose, algorithms have been developed which detect, isolate, and accommodate sensor failures using analytical redundancy. In this paper preliminary results of a full scale engine demonstration of the ADIA algorithm are presented. Minimum detectable levels of sensor failures for an F100 turbofan engine control system are determined and compared to those obtained during a previous evaluation of this algorithm using a real-time hybrid computer simulation of the engine.

Introduction

The objective of the advanced detection, isolation, and accommodation (ADIA) program is to improve the overall demonstrated reliability of digital electronic control systems for turbine engines by detecting, isolating, and accommodating sensor failures using analytical redundancy methods. This paper discusses the preliminary results of a full scale engine demonstration of an analytical redundancy based algorithm developed as part of the ADIA program.

Over the past 35 years hydromechanical implementations of turbine engine control systems have matured into highly reliable units. However, there is a trend towards increased engine complexity. This increased complexity is required to meet ever increasing engine performance requirements. Consequently, the engine control has become increasingly complex. Because of this complexity trend and the revolution in digital electronics, the control has evolved from a hydromechanical to a full authority digital electronic (FADEC) implementation. These FADEC type controls have to demonstrate the same or improved levels of reliability as their hydromechanical predecessors.

In an effort to improve the overall reliability of the digital electronic control system, various redundancy management techniques have been applied to both the total control system and to individual components. Studies[1] have shown that the least reliable of the control system components are the engine sensors. In fact some type of sensor redundancy will be required to achieve adequate control system reliability. One important type is analytical redundancy (AR).[2] Analytical redundancy uses a model to generate redundant information that can be compared to measured information to detect failures. Analytical redundancy based systems can have cost and weight savings over other approaches such as hardware redundancy.

Considerable progress has been made in the application of analytical redundancy to improve turbine engine control system reliability. The accomplishments, surveyed in Ref. 2, point to several technology needs. These include: (1) the ability to detect small (soft) failures, (2) real-time implementations of algorithms capable of detecting soft failures, (3) a comparison of failure detection algorithm complexity versus performance, (4) a full scale demonstration of a soft failure detection capability, and (5) an evaluation of the pseudolinearized modeling approach. The ADIA program conducted at the NASA Lewis Research Center addresses all of these technology needs.

The ADIA program is organized into four phases: development, implementation, evaluation, and demonstration. References 3 to 7 describe the development, implementation, and evaluation phases. In the development phase[3,4] the ADIA algorithm was designed using advanced filtering and detection methodologies. In the implementation phase[5] this advanced algorithm was implemented in microprocessor based hardware. A parallel computer architecture (three processors) was used to allow the algorithm to execute in a frame time consistent with stable, real-time operation. In the evaluation phase[6,7] an evaluation of algorithm performance was obtained using a real-time engine simulation running on a hybrid computer. The objectives of the evaluation were to (1) validate the algorithm for sensor failure detection, isolation, and accommodation (DIA) effectiveness, (2) document algorithm performance, (3) validate the algorithm's real-time implementation, and (4) establish a data base for the demonstration phase of the ADIA program. This report describes the demonstration of the ADIA algorithm on a full scale F100 engine in the NASA Lewis altitude test facility.

This paper begins with a description of the test bed system used in the demonstration of the ADIA algorithm. Next, a description of the ADIA algorithm is given followed by a description of the implementation hardware. The results of the demonstration are then presented. Finally, conclusions and recommendations for further work are given.

Test Bed System Description

The algorithm was demonstrated using a test bed system consisting of (1) an engine system, (2) a multivariable control, and (3) the ADIA algorithm. The algorithm will be described in the next section. The test bed system is shown in Fig. 1.

Engine System

The engine system consists of an F100 turbofan engine and its associated control actuators and sensors. The F100 turbofan engine is a high-performance, low-bypass ratio, twin-spool turbofan engine. The test engine has four controlled inputs, five sensed outputs, and four sensed environmental variables.

These variables are defined as follows.

Controlled engine inputs, U_{com} and U_m

WF	Main combustor fuel flow
AF	Exhaust nozzle area
CIVV	Fan inlet variable vanes
RCVV	Rear compressor variable vanes

Sensed engine outputs, Z_m

N1	Fan speed
N2	Compressor speed
PT4	Burner pressure
PT6	Exhaust nozzle pressure
FTIT	Fan turbine inlet temperature

Sensed environmental variables, E_m

P0	Ambient (static) pressure
PT2	Fan inlet (total) pressure
TT2	Fan inlet temperature
TT25	Compressor inlet temperature

Strictly speaking, TT25 is a sensed engine output variable. However, since TT25 is used only as a scheduling variable in the control (like TT2), it is considered an environmental variable and is not covered by the ADIA logic.

Multivariable Control System

The multivariable control (MVC) system shown in Fig. 1 is essentially a model following, proportional-plus-integral control. The MVC control[8] was previously demonstrated in an altitude test of an F100 engine.[9] The components of the control are the reference point schedules, the transition schedules, the proportional control logic, the integral control logic, and the engine protection logic. The reference point schedules generate a desired engine operating point in response to the pilot's thrust command (PLA) and sensed engine environment. The transition logic generates rate limited command trajectories for smooth transition between steady-state operating points. The proportional and integral control logic minimize transient and steady-state deviations from the commanded trajectories. The engine protection logic limits the size of the commanded engine inputs. The normal control mode in the MVC logic used fuel flow to set engine fan speed and nozzle area to set nozzle pressure (engine pressure ratio). However, at those conditions where limiting is required, fuel flow can be used to limit the maximum FTIT, the maximum PT4, or the minimum PT4.

Demonstration Hardware

The ADIA algorithm was demonstrated using an F100 engine in the NASA Lewis altitude test facility. This facility can duplicate a wide range of flight conditions (altitude and Mach number). Reference 10 describes the microprocessor-based control system computer which implemented the MVC and ADIA algorithms, including accompanying interface and monitoring hardware interactive data acquisition software. A separate personal-computer-based system for simulating sensor failures[11] was used. This failure simulator was located between the engine sensors and the control system computer. It simulated sensor failures by adding an appropriate bias voltage to the selected engine sensor outputs signals. All sensor failures (i.e., both hard and soft) were injected in this fashion.

Algorithm Description

The ADIA algorithm detects, isolates, and accommodates sensor failures 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 ADIA algorithm (Fig. 1) consists of three elements: (1) hard sensor failure detection and isolation logic, (2) soft sensor failure detection and isolation logic, and (3) an accommodation filter. Hard failures are defined as out-of-range or large bias errors that occur instantaneously in the sensed values. Soft failures are defined as small bias errors or drift errors that increase relatively slowly with time.

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 ($\hat{Z}$) are used by the control law. When a sensor failure occurs, the detection logic determines that a failure has occurred. The isolation logic then determines which sensor is faulty. This structural information is passed to the accommodation filter. The accommodation filter removes the faulty measurement from further consideration. The filter, however, continues to generate the full set of optimal estimates for the control. Thus the control does not have to restructure for any sensor failure. The ADIA algorithm inputs as shown in Fig. 1 are the sensed engine output variables, Z_m, and the sensed engine input variables, $U_m(t)$. The outputs of the algorithm, the estimates, $\hat{Z}(t)$, of the measured engine outputs, $Z_m(t)$, are used as input to the proportional part of the control. During normal mode operation, engine measurements are used in the integral control to ensure accurate steady-state operation. When a sensor failure 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 $\hat{X}$ and the engine outputs $\hat{Z}$ as follows.

$$\dot{\hat{X}} = F(\hat{X} - X_b) + G(U_m - U_b) + K\gamma$$

$$\hat{Z} = H(\hat{X} - X_b) + D(U_m - U_b)$$

$$\gamma = Z_m - \hat{Z}$$

Here the subscript b represents the base point (steady-state engine operating point) and X is the 4 by 1 model state vector, U_m the 4 by 1 sensed control vector, and Z_m is the 5 by 1 sensed output vector. The matrix K is the Kalman gain matrix and γ is the residual vector. The F, G, H, and D matrices are the appropriately dimensioned system matrices. Their individual matrix elements along with those of K are corrected by the engine inlet conditions E_m and scheduled an nonlinear functions of $\hat{Z}$. These functions are given in Ref. 2.

Reconfiguration of the accommodation filter, after the detection and isolation of a sensor failure, is accomplished by forcing the appropriate residual element to zero. This effectively substitutes the estimate for the feedback (sensed) variable. For example if a compressor speed sensor failure (N2) has been isolated, the effect of reconfiguration is to force $\gamma_2 = 0$. This is equivalent to setting sensed N2 equal to the estimate of N2 generated by the filter. The residuals generated by the accommodation filter are used in the hard failure detection logic.

Hard Failure Detection and Isolation Logic

The hard sensor failure detection and isolation logic is straightforward. To accomplish hard failure detection and isolation the absolute value of each component of the residual vector is compared to its own threshold. If the residual absolute value is greater than the threshold, then a failure is detected and isolated for the sensor corresponding to the residual element. Threshold sizes are initially determined from the standard deviation of the noise on the sensors. These standard deviation magnitudes are then increased to account for modeling errors in the accommodation filter. The hard detection threshold values are twice the magnitude of these adjusted standard deviations. These magnitudes are summarized in Table 1.

Soft Failure Detection and Isolation Logic

The soft failure detection logic consists of multiple-hypothesis-based testing. Each hypothesis is implemented using a Kalman filter. The soft failure detection/isolation logic structure is shown in Fig. 2. A total of six hypothesis filters are shown, one for normal mode operation and five for the failure modes (one for each engine output sensor). The structure for each hypothesis filter is identical to the accommodation filter. However, each hypothesis filter uses a different 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, γ_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 γ_i has a certain likelihood or probability.

$$L_i = p_i(\gamma_i) = k\, e^{-WSSR_i}$$

where k is a constant and

$$WSSR_i = \gamma_i^T \Sigma^{-1} \gamma_i$$

with

$$\Sigma = \mathrm{diag}(\sigma_i^2)$$

The σ_i are the standard deviations defined in Table 1. These standard deviation values scale the residuals to unitless quantities that can be summed to form WSSR. The WSSR statistic is smoothed to remove gross noise effects by a first order lag with a time constant of 0.1 sec. Mathematically, when the log of the ratio of likelihoods is taken, then

$$LR_i = \log\left(\frac{L_i}{L_0}\right) = WSSR_0 - WSSR_i$$

The log of the ratio of each hypothesis likelihood to the normal mode likelihood is calculated. If the maximum log likelihood ratio exceeds the threshold, then a failure is detected and isolated and accommodation occurs. If a sensor failure has occurred in N1 for example, all of the hypothesis filters except H_1 will be corrupted by the faulty information. Thus each of the corresponding likelihoods will be small except for H_1. Thus, LR_1 will be the maximum and it will be compared to the threshold to detect the failure.

Initially, the soft failure 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. It was soon apparent from initial evaluation studies that transient modeling error was dominant in determining the fixed threshold levels. It was also clear that this threshold was too large for desirable steady-state operation. Thus, an adaptive threshold was incorporated. The adaptive threshold is triggered by an internal control system variable which is indicative of transient operation. When the engine experiences a transient, the isolation threshold is expanded. The exact modification was found by experimentation on a simulation to minimize false alarms during transients. The adaptive threshold expansion logic enabled the steady-state detection threshold to be reduced to 80 percent of its original value. Additional details of the algorithm can be found in Ref. 6.

Failure Accommodation

For accommodation two separate steps are taken. First, all seven of the filters (the accommodation filter and the six hypothesis filters) are reconfigured to account for the detected failure mode. Second, if a soft failure was detected, the states and estimates of all seven filters are updated to the correct values of the hypothesis filter which corresponds to the failed sensor.

ADIA Algorithm Implementation

To conduct the test-bed demonstration an implementation of the ADIA algorithm was integrated with an existing microcomputer implementation of the F100 multivariable control (MVC) algorithm. The resulting controls microcomputer system was based on the Intel 80186 microprocessor architecture. To satisfy the control update interval requirement of 40 msec to guarantee engine stability, three processors (CPU's) operating in parallel are used. Data is transferred between CPU's through dual-ported memory. Synchronization between CPU's is achieved through interrupts.

The existing MVC implementation was programmed in fixed point assembly language and was used without change on CPU No. 1. The ADIA algorithm executes on CPU's No. 2 and No. 3 and was programmed using floating point arithmetic and FORTRAN. The total memory requirement for the three CPU's is 57 Kbytes for the algorithm and 15 Kbytes for the executive. In all cases the code and constants were about 65 percent and the data or variables about 35 percent of the total memory required. Additional details of the implementation can be found in Ref. 6.

Demonstration Results

This section describes preliminary results obtained in demonstrating the ADIA algorithm using the F100 engine. The test procedure and the results of the demonstration are discussed. The objective of the engine test was to demonstrate the operation and performance of the ADIA algorithm and its implementation over a substantial portion of the flight envelope of the F100 engine.

Procedure

The test matrix, shown in Fig. 3, summarizes the tests used to demonstrate the algorithm. The different engine operating conditions (altitude/Mach number) used for the demonstration are listed across the top of the matrix and the different tests conducted at these points along the side. Currently, only results for the 10 000 ft altitude, 0.6 Mach number (10 K/0.6) operating condition have been obtained. Engine power (PLA) settings were selected which represent maximum nonafterburning (intermediate) thrust (PLA = 80) and ~50 percent of intermediate thrust (PLA = 50 and 70). The rationale used to select the test matrix operating points was to duplicate as many of the points used in the F100 Multivariable Control Program[7] as possible, to avoid high fan inlet pressures, and to reasonably span the envelope. This rationale is a compromise among taking advantage of previous results for comparison, limited risk engine operation, and full envelope validation.

The type of tests used in the demonstration were selected to completely define detection performance for three common failure modes. Also, tests were conducted to determine engine control performance with and without engine sensor failures. The tests are summarized in Table 2.

Resul'·

Three types of demonstration results will now be presented. The first shows the accuracy of the accommodation filter with no sensor failures. the second shows the detection performance of the ADIA algorithm. Finally, accommodation performance is demonstrated.

Estimator accuracy. The single most important element in determining ADIA algorithm performance is the accuracy of the engine output estimates used in the algorithm. These estimates are determined using the accommodation filter, which incorporates a simplified engine model. A sample of the accuracy of the filter is presented in Fig. 4 which shows an idle-to-intermediate-power PLA pulse transient generated at the 10 K/0.6 condition. The variables shown, fan speed (N1) and exhaust nozzle pressure (PT6), demonstrate the excellent estimate accuracy achieved. These results are typical of the other estimates.

Detection/accommodation performance. Two types of sensor failure were considered, hard and soft. Hard failures, because of their size, are easily detected. Thus, hard failure detection performance, although important to system reliability, was examined at only one operating condition (10 K/0.6). The ADIA algorithm exhibited excellent hard detection performance at this condition. There were no false alarms or missed detections of any hard failures. Hard failures were injected in each of the engine sensor output signals. Successful detection and accommodation of the failure was accomplished in each case. In addition no false alarms in the hard detection logic were encountered during the subsequent soft failure demonstration.

Soft sensor failures, although small in magnitude, if undetected may result in degraded or unsafe engine operation. Soft failures are more difficult to detect then hard failures. Therefore the demonstration concentrated on soft failure performance. Two soft failure modes were studied, bias and drift. The criteria used to evaluate detection, isolation, and accommodation performance were: (1) minimum detectable bias values and drift rates, (2) elapsed time between sensor failure and detection, (3) steady-state performance degradation after failure accommodation, and (4) transient response of the engine to the filter and control reconfiguration of failure accommodation. The minimum detectable levels of bias and drift rate obtained for the 10 K/0.6 condition are summarized in Table 3. The minimum detectable drift rates were determined by adjusting the drift magnitude such that a failure was detected ~5 sec after failure inception.

The demonstration results are compared to the levels obtained during the real-time hybrid evaluation phase of the program. The comparison at PLA = 80° shows an excellent agreement for both bias and drift magnitudes. At PLA = 50° however, the demonstration results are slightly higher than the hybrid evaluation results. In this case the detection threshold has been expanded by a factor of two to study the degradation in performance this would cause. In general, the detection levels are still good except for N1. The algorithm was unable to detect a failure smaller than the hard detection level of 600 rpm.

Additionally, detection performance for sequential failures was studied. Six different sequences of soft failures were injected into the test bed system. One example of a failure sequence was to fail N1, then 4 sec later fail N2, then PT4, and then PT6. In each case the algorithm successfully detected and accommodated each sensor failure in the correct order. These tests demonstrate the ability of the algorithm to continue to successfully perform even after some sensors have failed.

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

Accommodation performance. Two experiments were used to demonstrate the successful accommodation of sensor failures. The first experiment consisted of injecting, detecting, and accommodating a single sensor failure and then commanding a PLA pulse transient. Engine performance with this accommodated failed sensor is compared to normal mode engine performance. Typical results are shown for the PT6 single failure case for fan speed (Fig. 5(a)) and exhaust nozzle pressure (Fig. 5(b)). Performance was good since the desired or request values were closely maintained. A slight drop in actual PT6 can be seen but this is acceptable. In all other cases the accommodated single failure transient performance was good.

The second experiment demonstrated the excellent accuracy of the engine model. In this experiment, first all the engine sensors were failed and accommodated. Then, a PLA pulse transient was generated from idle to about 75 percent of full power. Results for N1 and PT6 are shown in Fig. 6. Again 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 engine feedback information over a broad power range.

Conclusions

Based on results of engine tests obtained so far, several preliminary conclusions have been reached. First, it can be concluded that the ADIA failure detection algorithm works quite well. Sensor failure detection and accommodation were demonstrated at two power conditions. The minimum detectable failure magnitudes represent excellent algorithm performance and compare favorably 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 sensors failed and accommodated) over 75 percent of the power range was also demonstrated. Second, 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, were used. Parallel processing was also used and shown to be an effective approach to achieving a real-time implementation using off-the-shelf (cost effective) computer resources. Finally, it is concluded that the demonstrated high performance detection, isolation, and accommodation capabilities of the ADIA algorithm justifies further demonstration throughout the flight envelope. Pending the anticipated successful outcome of the additional demonstration testing, a flight test evaluation may be justified as future work.

References

1. Baker, L.E., Warner, D.E., and Disparte, C.P., "Design of Fault Tolerant Electronic Engine Controls," AIAA Paper 81-1496, July 1981.

2. Merrill, W.C., "Sensor Failure Detection for Jet Engines Using Analytical Redundancy," Journal of Guidance, Control and Dynamics, Vol. 8, No. 6, Nov.-Dec. 1985, pp. 673-682.

3. Beattie, E.C., Laprad, R.F., McGlone, M.E., Rock, S.M., and Akhter, M.M., "Sensor Failure Detection System - for the F100 Turbofan Engine," PWA-5736-17, Pratt and Whitney Aircraft, East Hartford, CT, Aug. 1981. (NASA CR-165515.)

4. Beattie, E.C., Laprad, R.F., Akhter, M.M., and Rock, S.M., "Sensor Failure Detection for Jet Engines," PWA-5891-18, Pratt and Whitney Aircraft, East Hartford, CT, May 1983. (NASA CR-168190.)

5. Delaat, J.C. and Merrill, W.C., "A Real-Time Implementation of an Advanced Sensor Failure Detection, Isolation and Accommodation Algorithm," AIAA Paper 84-0569, Jan. 1984. (Also, NASA TM-83553.)

6. Merrill, W.C., and DeLatt, J.C., "A Real-Time Simulation Evaluation of an Advanced Detection, Isolation and Accommodation Algorithm for Sensor Failures in Turbine Engines," NASA TM-87289, 1986.

7. Merrill, W.C., DeLatt, J.C., and Bruton, W., "Advanced Detection, Isolation, and Accommodation of Sensor Failures--Real-Time Evaluation," NASA TP- , 1987. (To be published.)

8. Soeder, J.F., "F-100 Multivariable Control Synthesis Program - Computer Implementation of the F-100 Multivariable Control Algorithm," NASA TP-2231, 1983.

9. Lethinen, B., Costakis, W.G., Soeder, J.F., and Seldner, K., "F-100 Multivariable Control Synthesis Program - Results of Engine Altitude Tests," NASA TM S-83367, 1983.

10. DeLatt, J.C., Soeder, J.F., "Design of a Microprocessor-Based Control, Interface, and Monitoring (CIm) Unit for Turbine Engine Controls Research," NASA TM-83433, 1983.

11. Melcher, K.J., DeLaat, J.C., Merrill, W.C., Oberle, L.G., Schaefer, J.H., and Sadler, G.G., "A Sensor Failure Simulator for Control System Reliability Studies," NASA TM-87271, 1986.

TABLE 1. - HARD DETECTION THRESHOLD MAGNITUDES

Sensor	Adjusted standard deviation	Detection threshold
N1	300 rpm	600 rpm
N2	400 rpm	800 rpm
PT4	30 psi	60 psi
PT6	5 psi	10 psi
FTIT	250 R	500 R

TABLE 2. - TEST DEFINITIONS

Name	Description
Sensor failures	
Hard	Large magnitude (hard) bias failure is induced.
Soft	Small magnitude (soft) bias failure is induced.
Drift	Small magnitude (soft) drift failure is induced.
SSF	A sequence of successive sensor failures is induced.
PLA transients	
Pulse	Idle to intermediate to idle transient PLA excursions. The intermediate power level is maintained for 10 sec.
Single	Pulse test with a single sensor failure accommodated before initiating the transient.
Open	Same as the Pulse test except that the minimum power level is raised slightly and the maximum power level is decreased slightly and the engine is controlled without using any sensed engine output information in the control, i.e., all sensors failed.

TABLE 3. - MINIMUM DETECTABLE BIAS AND DRIFT FAILURE MAGNITUDES
[Altitude = 10 K, Mach = 0.6.]

PLA	Sensor	Minimum detectable bias failure			Minimum detectable drift failure		
		Hybrid simulation	Engine demonstration		Hybrid simulation	Engine demonstration	
50	N1	300.0	----	rpm	100.0	200.0	rpm/sec
	N2	300.0	350	rpm	100.0	100.0	rpm/sec
	PT4	12.5	30	psi	2.5	4.0	psi/sec
	PT6	3.0	5	psi	0.8	0.9	psi/sec
	FTIT	50.0	200	deg	70.0	100.0	deg/sec
80	N1	-350.0	-350	rpm	-125.0	-125.0	rpm/sec
	N2	-350.0	-350	rpm	-125.0	-125.0	rpm/sec
	PT4	-12.5	-14	psi	-1.3	-2.5	psi/sec
	PT6	-3.0	-3	psi	-0.6	-0.6	psi/sec
	FTIT	-150.0	-200	deg	-70.0	-100.0	deg/sec

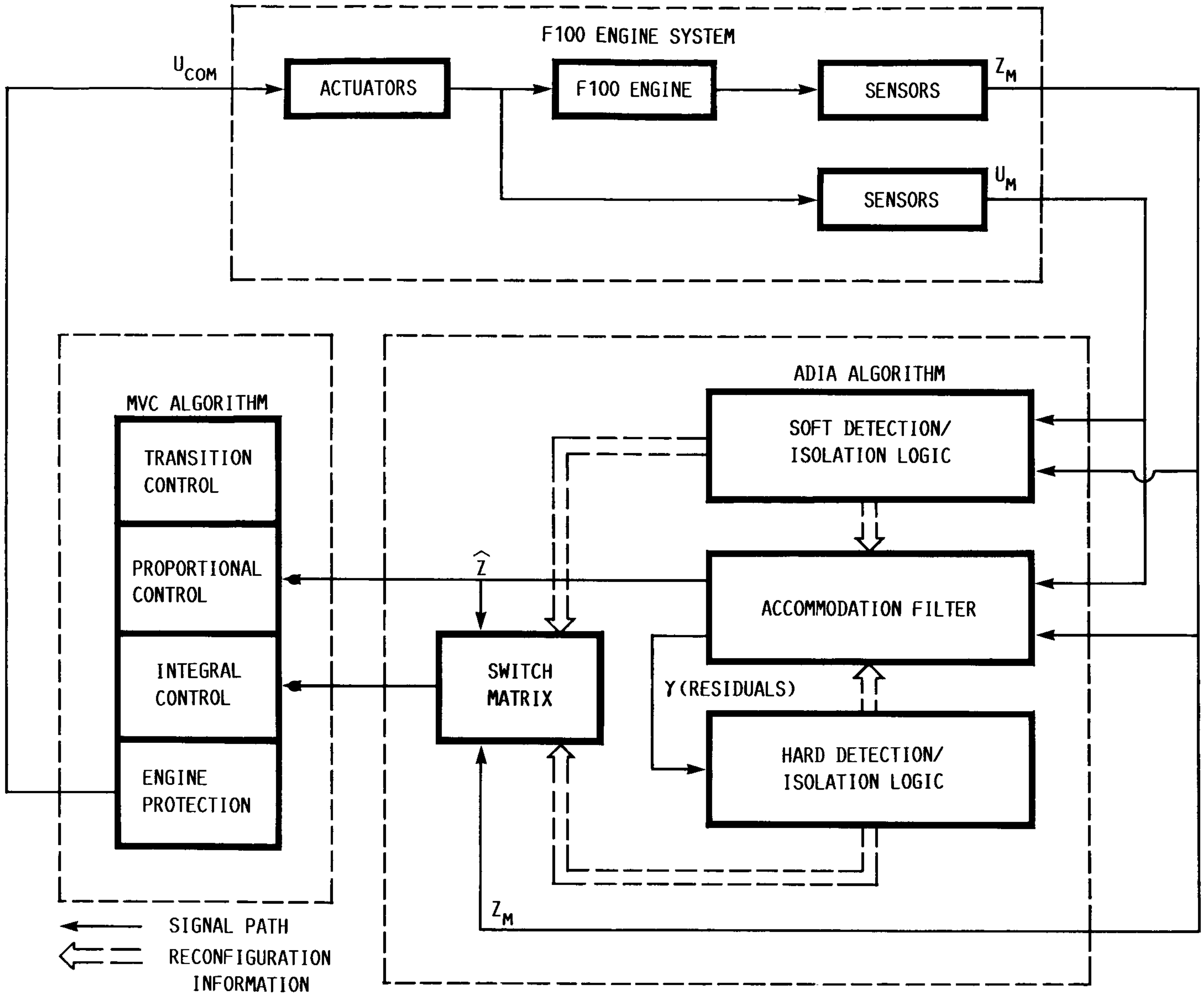


FIGURE 1. - F100 TESTBED SYSTEM.

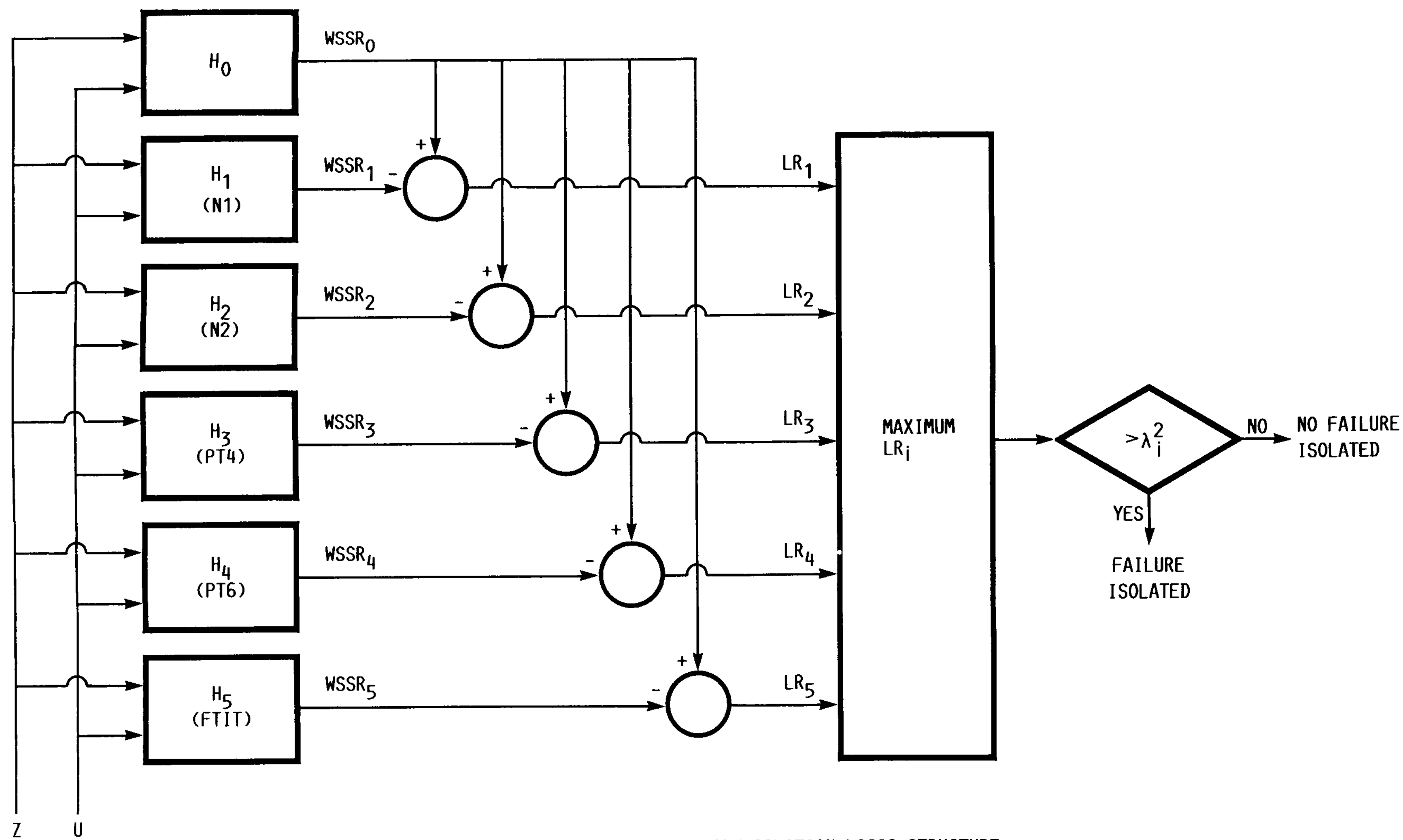


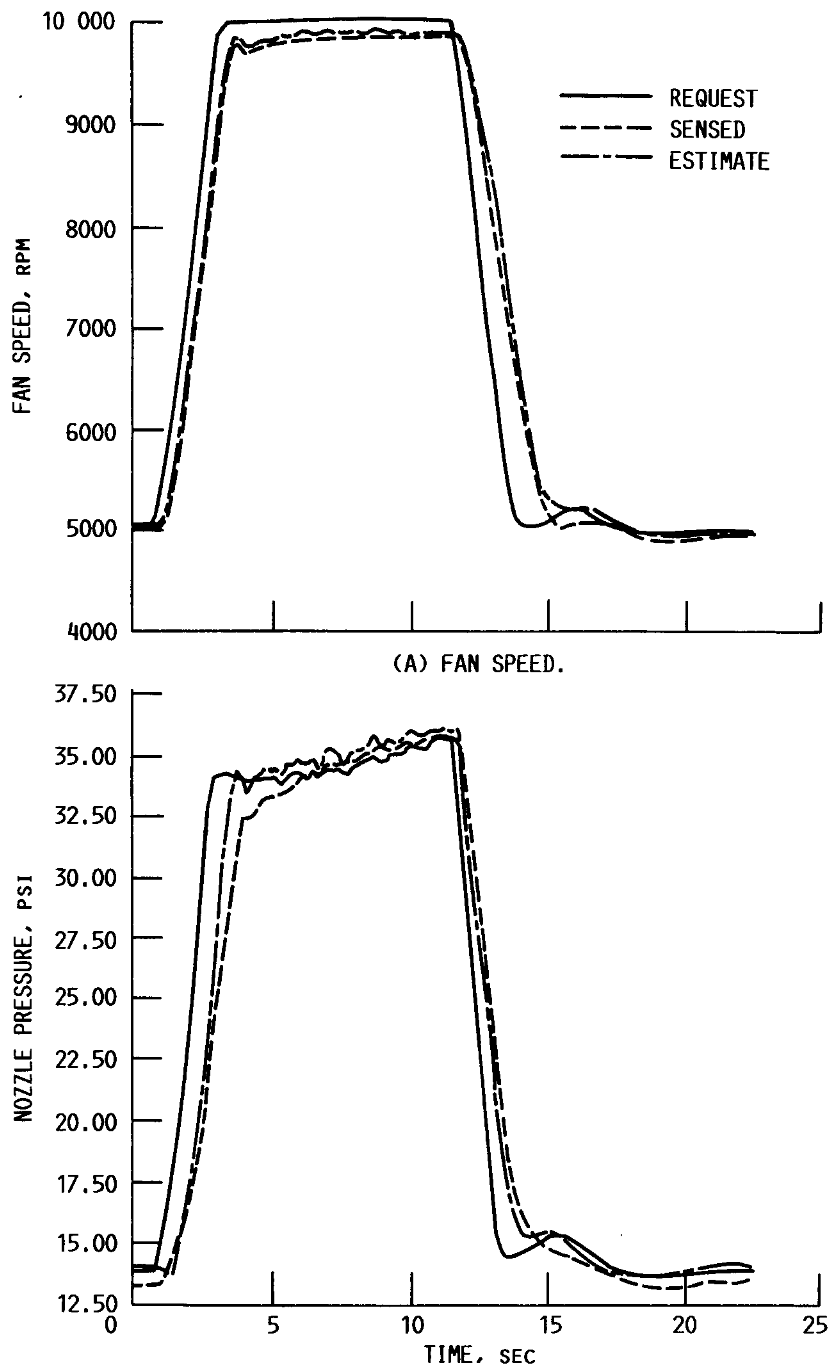
FIGURE 2. - SOFT FAILURE DETECTION/ISOLATION LOGIC STRUCTURE.

	ALTITUDE/ MACH NO.	10K/.6		30K/.9		10K/.9		45K/.9	10K/1.2	50K/1.8	35K/1.8	55K/2.2
PLA, DEG		50	80	50	80	50	80	70	80	80	80	80
TEST TYPE (SEE TABLE II)	HARD	P	P									
	SOFT	P	P	X	X	X	X	X	X	X	X	X
	DRIFT	P	P	X	X	X	X	X	X	X	X	X
	SSF	P	P	X	X							
	PULSE	P		X		X		X	X			
	SINGLE	P		X		X		X	X			
	OPEN	P		X		X		X	X			

KEY: P - TESTS COMPLETED

X - TESTS NOT YET COMPLETED

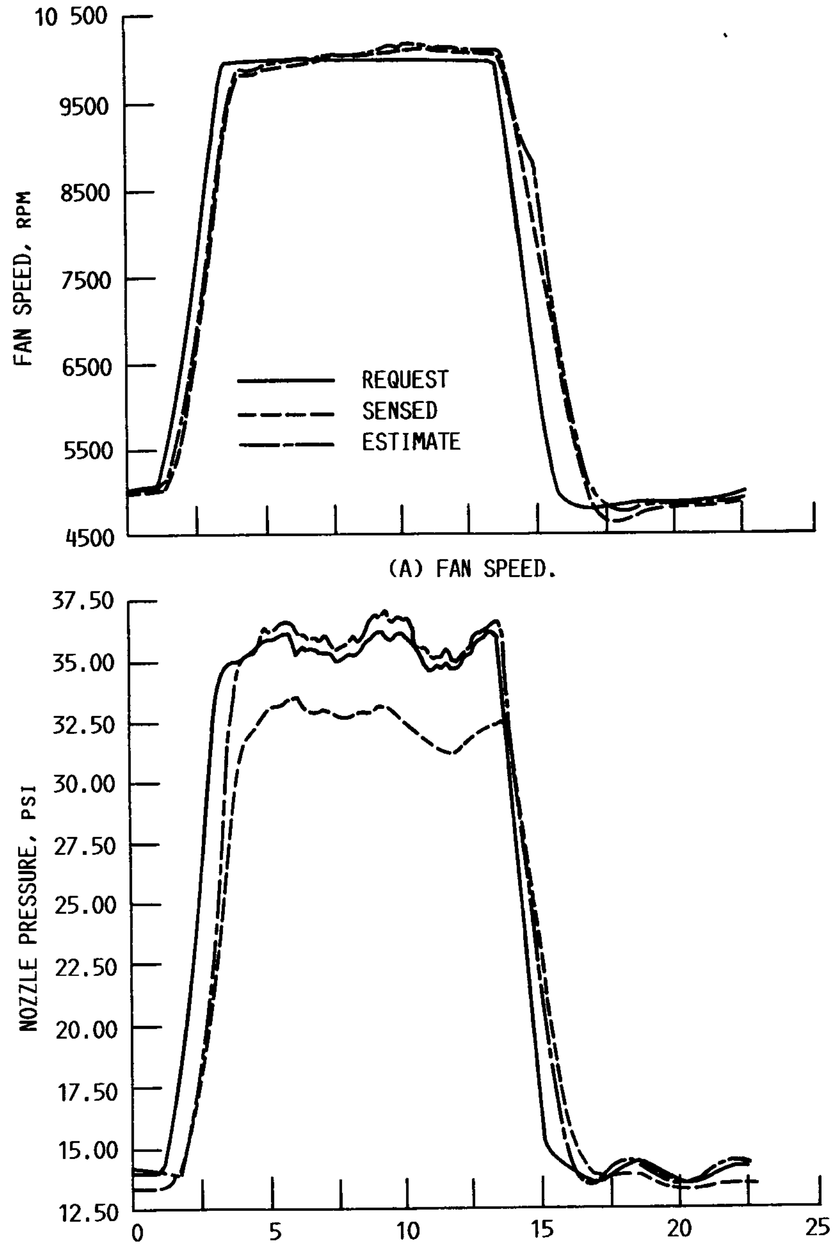
FIGURE 3. - DEMONSTRATION TEST MATRIX.



(A) FAN SPEED.

(B) EXHAUST NOZZLE PRESSURE.

FIGURE 4. - RESPONSE TO A PLA PULSE INPUT - NO SENSOR FAILURES.



(B) EXHAUST NOZZLE PRESSURE.

FIGURE 5. - RESPONSE TO A PLA PULSE INPUT WITH A PT6 FAILURE ACCOMMODATED.

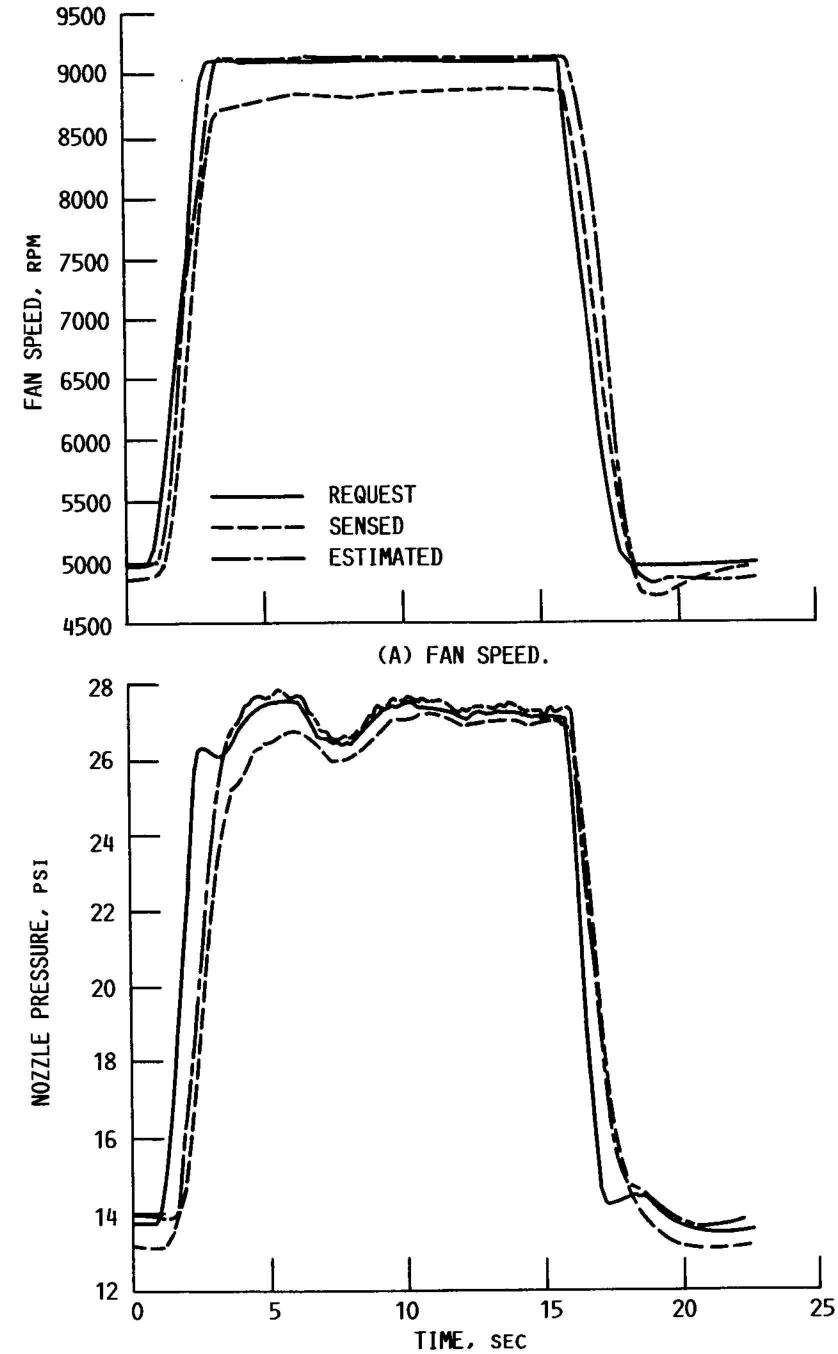


FIGURE 6. - RESPONSE TO A PLA PULSE INPUT WITH ALL SENSORS FAILED AND ACCOMMODATED.

1. Report No. NASA TM-89880 AIAA-87-2259	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Full-Scale Engine Demonstration of an Advanced Sensor Failure Detection, Isolation, and Accommodation Algorithm - Preliminary Results		5. Report Date
		6. Performing Organization Code 505-62-01
7. Author(s) Walter C. Merrill, John C. DeLaat, Steven M. Kroszkewicz, and Mahmood Abdelwahab		8. Performing Organization Report No. E-3561
		10. Work Unit No.
9. Performing Organization Name and Address National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		14. Sponsoring Agency Code

15. Supplementary Notes

Prepared for the Guidance, Navigation, and Control Conference, sponsored by the American Institute of Aeronautics and Astronautics, Monterey, California, August 17-19, 1987.

16. Abstract

The objective of the advanced detection, isolation, and accommodation (ADIA) program is to improve the overall demonstrated reliability of digital electronic control systems for turbine engines. For this purpose, algorithms have been developed which detect, isolate, and accommodate sensor failures using analytical redundancy. In this paper preliminary results of a full scale engine demonstration of the ADIA algorithm are presented. Minimum detectable levels of sensor failures for an F100 turbofan engine control system are determined and compared to those obtained during a previous evaluation of this algorithm using a real-time hybrid computer simulation of the engine.

ORIGINAL PAGE IS
OF POOR QUALITY

17. Key Words (Suggested by Author(s)) Sensors; Failures; Detection; Redundancy; Fault tolerance; Feedback controls; Isolation; Accommodation; Engine test	18. Distribution Statement Unclassified - unlimited STAR Category 33

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price*
Unclassified	Unclassified	13	A02

*For sale by the National Technical Information Service, Springfield, Virginia 22161