NASA Technical Memorandum 87289

2665

A Real-Time Simulation Evaluation of an Advanced Detection, Isolation and Accommodation Algorithm for Sensor Failures in Turbine Engines

Walter C. Merrill and John C. DeLaat
Lewis Research Center
Cleveland, Ohio

Prepared for the
1986 American Control Conference
sponsored by the Institute of Electrical and Electronics Engineers
Seattle, Washington, June 18–20, 1986

NASA

A REAL-TIME SIMULATION EVALUATION OF AN ADVANCED DETECTION, ISOLATION AND ACCOMMODATION ALGORITHM FOR SENSOR FAILURES IN TURBINE ENGINES

Walter C. Merrill and John C. DeLaat
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

E-2995

SUMMARY

An Advanced sensor failure Detection, Isolation, and Accommodation (ADIA) algorithm has been developed for use with an aircraft turbofan engine control system. In a previous paper the authors described the ADIA algorithm and its real-time implementation. This paper discusses subsequent improvements made to the algorithm and implementation, and presents the results of an evaluation. The evaluation used a real-time, hybrid computer simulation of an F100 turbofan engine.

INTRODUCTION

The ADIA program (ref. 1) is an effort to improve the overall demonstrated reliability of digital electronic control systems for aircraft turbine engines by detecting sensor failures using analytical redundancy. Various redundancy management techniques have been applied to both the total control system and to individual components. The least reliable of the control system components are the engine sensors. Typically, sensor redundancy is required to achieve adequate control system reliability. The ADIA approach uses analytical redundancy to achieve reliability.

Analytical redundancy uses a reference model of the engine and redundant information from dissimilar sensors to provide an estimate of a measured variable. How the estimates and measurements are used to detect failures is one way to differentiate the various analytical redundancy approaches. The ADIA algorithm is based upon hypothesis testing of Kalman filter generated residuals. Considerable work has been accomplished in the application of analytical redundancy to improve turbine engine control system reliability. These accomplishments are surveyed in reference 2.

The ADIA program has been organized into four phases: development, implementation, evaluation, and demonstration. Reference 1 describes the development and implementation phases. This paper describes additional development and implementation details as well as giving the significant real-time hybrid computer evaluation results. The ADIA algorithm will be demonstrated on an F100 engine at the NASA Lewis Research Center altitude test facility during the last half of 1986. The paper will briefly describe the ADIA algorithm and the simplified engine model used in the algorithm. Also included is a description of the hardware and software used to implement the algorithm. Finally, results are presented. These include simplified model accuracy and ADIA performance for hard and soft failures.

ALGORITHM DEVELOPMENT

The ADIA algorithm detects, isolates, and accommodates sensor failures in turbofan engine control systems. The ADIA algorithm was originally developed for Lewis under contract (ref. 3). The algorithm incorporates advanced filtering and detection logic and is general enough to be applied to different engines or other types of control systems. A specific version of the ADIA algorithm was designed for the F100 engine and F100 Multivariable Control. This combination of engine, control, and ADIA algorithm comprises the F100 testbed system shown in figure 1.

The ADIA algorithm consists of three elements: (1) hard failure detection and isolation logic, (2) soft failure detection and isolation logic, and (3) an accommodation filter. These are shown as part of the testbed system in figure 1. The algorithm detects two classes of sensor failures, hard and soft. Hard failures are out-of-range or large bias errors that occur instantaneously in the sensed values. Soft failures are small bias errors or drift errors that accumulate relatively slowly with time.

The algorithm inputs are the measured engine inputs, $u_m(t)$, (fuel flow, nozzle area, compressor inlet guide vane angle, rear compressor variable vane angle, and bleed flow) and the measured engine outputs, $z_m(t)$ (fan speed, compressor speed, burner pressure, augmentor pressure, and fan turbine inlet temperature). The algorithm outputs, $z(t)$, are optimal estimates of the engine outputs, $z(t)$.

The hard failure detection logic uses residuals generated by the accommodation filter to detect and isolate hard failure. The soft failure logic uses six hypothesis filters (one normal mode and five failure modes) to detect and isolate soft failures. Implicit in the soft failure isolation logic and the accommodation filter is a simplified engine model. The algorithm implementation and hybrid evaluation phases of the program have resulted in several improvements to both the structure and operation of the algorithm.

The most significant improvement has been the removal of the sequential structure of the soft detection and soft isolation logic. Originally, the philosophy of the algorithm was to detect the presence of a soft failure with a Weighted Sum of Squared Residuals (WSSR) statistic. Once a failure had been detected, then an hypothesis-based bank-of-filters approach was used to isolate the faulty component. Since soft failures are uncommon events and since the hypothesis-based logic takes a significant computational capability, the original implementation approach was to calculate the WSSR statistic each update cycle and only calculate the isolation logic when required. This forces a higher computational load and thus a longer update interval only during soft isolation. However, the three computer, parallel processing architecture, with which the ADIA algorithm has been implemented, has sufficient processing power to calculate the complete algorithm including the soft isolation logic during each update interval. With this implementation, described in the next section, the update interval does not change during soft isolation. Also, since the hypothesis filters are continuously tracking, the effect of sensor failures begins to accrue from the time of the failure, not after soft detection has occurred. Initial studies with this new configuration showed the isolation logic to be more sensitive to failures than the soft detection logic. In effect, failures were being isolated before they were being detected. This clearly removed the requirement for the WSSR statistic and it was removed from the algorithm.

Another improvement to the algorithm was the incorporation of integral action in the Kalman filter to improve the steady state accuracy of the Fan Turbine Inlet Temperature (FTIT) estimate. One important engine control mode is the limiting of FTIT at high power operation. Because the FTIT sensor is relatively slow, control action is based upon the dynamically faster FTIT estimate. Because the FTIT limiting control has integral action, a high degree of steady-state accuracy in the FTIT estimate is required to ensure satisfactory control. This accuracy is accomplished by augmenting the seven algorithm filters (one for the accommodation filter and six for the hypothesis filters) with the following additional state equation.

$$\dot{b} = K_6 * \gamma$$

$$\hat{FTIT} = \hat{z_5} + b$$

where K_6 is a gain matrix, b is the temperature bias, z_5 is the unbiased temperature estimate, and γ is the vector of residuals from the accommodation filter. The addition of these dynamics, while improving FTIT estimation accuracy, results in a larger minimum detectable FTIT drift failure rate.

Finally, the soft failure detection/isolation threshold was modified. Originally, thresholds were determined by noise statistics and then modified to accommodate modeling error effects. 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, M_{tran}, which is indicative of transient operation. When the engine experiences a transient M_{tran} is set to 4.5, otherwise it is 0. This variable is used as follows to modify the isolation threshold λ_I, as follows.

$$\lambda_I = \lambda_{ISS} * (\lambda_{EXP} + 1)$$

$$\tau * \dot{\lambda}_{EXP} + \lambda_{EXP} = M_{tran}$$

where λ_{ISS} is the steady-state detection/isolation threshold and τ = 2 sec. The values of λ_{ISS}, τ and M_{tran} were found by experimentation to minimize false alarms during transients. The adaptive threshold expansion logic enabled λ_{ISS} to be reduced to 40 percent of its original value.

These three improvements represent the major changes to the ADIA algorithm. The next section describes how the modified algorithm was implemented into commercially available, microprocessor based, hardware and software.

ADIA IMPLEMENTATION

The hardware implementation of the ADIA algorithm requires the integration of the algorithm with the F100 Multivariable Control (MVC) (ref. 4). The F100 MVC had been implemented on an 8086 microcomputer (ref. 5). The ADIA was merged with this MVC implementation to give a full microcomputer implementation of the control algorithm with sensor analytical redundancy. The F100 engine system dynamics require a combined MVC and ADIA update interval of 40 msec or less.

A microcomputer system for real-time controls research has been designed and fabricated at Lewis (ref. 6). The controls microcomputer within the system is based on the Intel 8086 microprocessor architecture. In order to implement the combined MVC/ADIA and satisfy the update interval requirement of 40 msec, a second 8086 based CPU was added to the controls microcomputer (ref. 1). This second CPU runs in parallel with the first. Data is transferred between CPU's through dual-ported memory and synchronization between CPU's was achieved through interrupts. Initially, only the normal-mode accommodation filter and the hard detect logic from the ADIA were implemented. This allowed a straightforward evaluation of the parallel processing mechanism. It was assumed that the soft failure detection and isolation logic could be added to the second CPU at a later date.

During algorithm development, the soft failure isolation logic was only run after a soft failure was detected by the soft failure detect logic. Due to the complexity of the soft failure isolation logic and since it was felt there might be some benefit to running the soft isolation logic in parallel with the soft detect logic, a third CPU was added to implement the soft isolation logic. Data transfers and synchronization were accomplished in the same manner as with the two-CPU implementation. Most recently, the 8086 based CPU's were replaced with 80186 based CPU's. The new CPU's are software compatible with the old CPUs, but are considerably faster. The relative timing for the three CPU's is shown in figure 2.

As shown in the figure, the different parts of the combined MVC/ADIA algorithm are divided among the three CPU's. The MVC is implemented in fixed point assembly language on CPU number 1. At the time the MVC was originally implemented on a microcomputer, it was felt that assembly language programming using fixed point arithmetic was necessary to achieve real-time execution of the algorithm. With the availability of the 8087 floating point coprocessor for the 8086 came the capability of implementing real time controls in floating point arithmetic. The majority of the ADIA algorithm running on CPU's numbers 2 and 3 is implemented using floating point arithmetic and the application oriented language FORTRAN. FORTRAN was chosen because the ADIA was originally coded in FORTRAN, and because a fairly good compiler was available for the 8086-8087.

The advantages of using an application language rather than assembly language include higher programmer productivity, increased readability, and easier maintenance. The primary disadvantage is that it generally produces less efficient object code than the equivalent assembly language.

Execution efficiency is critical for real-time controls. So, for the ADIA, table lookup routines (ref. 7), which are executed frequently in the algorithm, and the hardware interface routines which have no FORTRAN equivalents, are implemented in assembly language. To allow the remainder of the algorithm to remain in FORTRAN, the source code has been optimized to make it run more efficiently (ref. 8). As shown in figure 2, the entire MVC/ADIA algorithm now executes in less than 40 msec.

The programs for each of the CPU's are downloaded into the CPU's using a commercially available disk operating system, CP/M-86. The Microcontroller Interactive Data System (MINDS) is used for data acquisition (ref. 9). This software package runs on CPU number 1 in the spare time that the CPU is not executing the MVC algorithm (fig. 2). The package has both steady-state and

transient data-taking capabilities and can access any variable in the MVC or ADIA algorithms. The data taken can be uplinked to a mainframe computer for off-line processing. In addition, the software has been enhanced to allow plotting of transient data on-line while the control microcomputer is operating (ref. 10). The on-line transient data display of internal MVC and ADIA variables was an indispensable tool in the evaluation process.

ALGORITHM EVALUATION

This section describes the test setup used in the evaluation of the algorithm and gives the more significant results. These include the accuracy of the estimates generated by the simplified engine model and the simulated detection/isolation/accommodation performance of the algorithm.

Test Setup

The algorithm was evaluated using a real-time hybrid computer simulation of the F100 engine. The testbed system consists of the hybrid computer simulation of the engine, the microprocessor based control computer and its interface and monitor (CIM) unit, the Sensor Failure Simulator (SFS) unit, and a data handling utility.

The F100 engine hybrid simulation is a nonlinear, real-time, 32nd order model that includes sensor and actuator dynamics. Differential equations are solved on the analog portion of the hybrid. Component performance information is stored in the digital computer with interpolation and table lookup functions being handled by digital software. A complete description of the simulation and its accuracy performance is given in reference 11.

The Control, Interface, and Monitoring (CIM) unit (ref. 6) contains the microcomputer described in the implementation section. In addition, it contains hardware and cabling to allow the microcomputer to interface to the engine or simulation of the engine being controlled. Lastly, a monitoring system is contained in the CIM unit which allows the signals between the microcomputer and controlled engine to be checked for correctness.

The SFS unit consists of a personal computer driving discrete analog hardware. The SFS can simulate any preprogrammed sensor failure consisting of four basic types: a scale factor change, a bias, a drift, and noise. The SFS allows complete and repeatable control over the failure size and timing. Details of the SFS are given in reference 12.

Data handling is accomplished through the MINDS utility program. This program is described in the implementation section.

Estimate 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 a Kalman filter which incorporates a

simplified engine model. The accuracy of the output estimates for both steady-state and transient operation was evaluated at various engine operating points. An engine operating point is defined by the pilot's power request (Power Lever Angle, PLA), and the altitude (ALT) and Mach number (MN) at which the engine is operating.

Steady-state accuracy was obtained in a straightforward manner. The simulation was "flown" to the desired operating point and allowed to reach steady-state. Then control execution was halted (or frozen). MINDS was then used to sample and store a set of steady-state data. Comparisons of measured and estimated variables for six operating points are given in table I. Table I shows the difference (the residual) between sensed and estimated fan speed, N1, compressor speed, N2, burner pressure, PT4, exhaust nozzle pressure, PT6, and fan turbine inlet temperature, FTIT as a percent of nominal value. From these comparisons it is clear that the estimates exhibit excellent steady-state accuracy (except for PT4 at the 55K point).

Transient accuracy data was obtained in the following manner. Again the simulation was flown to the desired operating point and allowed to reach steady-state. An idle-to-intermediate-power PLA pulse transient was then simulated (fig. 3) at three different operating conditions. MINDS was used to sample and store data throughout the transient. An example plot of sensed and estimated fan speed and its residual, as well as the hypothesis statistic for fan speed are presented in figures 4 to 6. The residual is the difference between the sensed and estimated variables. The hypothesis statistic, H_i, is found from

$$H_i = \gamma_0^T W \gamma_0 - \gamma_i^T W \gamma_i$$

and γ_i is the residual vector from the ith hypothesis filter and γ_0 is from the accommodation filter, and W is a normalizing matrix. Trajectories in figures 4 to 6 give the reader a "feel" for the summarized results of tables II to V. Figure 6 also shows the threshold λ_I, as produced by the adaptive threshold logic. In table II the maximum value of the residuals obtained in response to the reference transient is given for each output at each of the three operating points. In table III the average absolute values of the residuals are given. Since detection performance is determined by the hypothesis statistics, estimate accuracy, interpreted in terms of these statistics, is critical to understanding algorithm performance. The maximum hypothesis values and the average hypothesis values are given in tables IV and V, respectively, to summarize transient accuracy for the reference trajectories. Plots of the hypothesis statistics became the standard tool used by the authors for evaluation and performance prediction. Overall, transient accuracy was considered to be quite good although not as good as steady-state accuracy. It was fairly evident then, that detection performance could be greatly improved if different thresholds for steady-state and transient detection were allowed. This observation led immediately to the implementation of the adaptive threshold logic described previously.

Detection/Accommodation Performance

Two types of failure were considered, hard and soft. Hard failures are defined to be large magnitude, perhaps out-of-range, failures. Because of their size, they are easily detected. Thus, hard failure detection performance, although important to system reliability, is not examined here. The ADIA algorithm exhibits excellent hard detection performance. Soft failures are defined to be small magnitude, in range failures that may accrue over time. Although small in magnitude, these failures, if undetected, may result in degraded or unsafe engine operation. Soft failures are more difficult to detect, and therefore we concentrate on soft failure performance. Two soft failure modes were considered, biases and drifts. Failures due to noise changes are not considered. Performance criteria studied were minimum detectable bias values and drift rates, detection time, steady-state performance degradation, and transient response to failure accommodation.

The procedure followed to obtain performance data was identical to that used to obtain transient accuracy data. Additionally, the SFS was used to inject a sensor failure of the appropriate size and at the desired time. The results obtained are summarized in table VI for minimum detectable biases and in table VII for drifts. In table VI the minimum detectable biases at six different operating points for each engine output are given. Detection times for these biases were essentially instantaneous. In table VII the minimum detectable drift rates are given. These rates were determined by adjusting the drift magnitude such that a failure was detected 5 sec after failure inception. Typical transient responses are given in figures 7 and 8. Since these are nominally steady-state responses, the hypothesis statistic is at its steady-state value. These responses show acceptable failure transient performance with little or no loss in steady-state performance.

CONCLUSIONS

An advanced sensor failure detection, isolation, and accommodation algorithm has been developed, implemented, and evaluated. The development included an adaptive failure detection threshold. The algorithm was implemented using a three-microprocessor, parallel architecture. The evaluation used a real-time hybrid computer simulation of an advanced turbofan engine. Estimate accuracy performance was excellent. Minimum detectable levels of bias and drift type failures were determined at seven operating points for all five sensed outputs. These minimum failure levels represent excellent algorithm detection and accommodation performance. This algorithm performs quite well in the real-time environment and is ready for a full scale engine demonstration. Such an engine demonstration is currently planned for July 1986.

REFERENCES

1. J.C. DeLaat, and W.C. Merrill, "A Real-Time Implementation of an Advanced Sensor Failure Detection, Isolation, and Accommodation Algorithm," AIAA Paper 84-0569, Jan. 1984.

2. W.C. Merrill, "Sensor Failure Detection for Jet Engines Using Analytical Redundancy," J. Guidance, Control, Dynamics, vol. 8, pp. 673-682, 1985.

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

4. B. Lehtinen, R.L. Dehoff, and R.D. Hackney, "Multivariable Control Altitude Demonstration on the F100 Turbofan Engine," AIAA Paper 79-1204, June 1979.

5. J.C. DeLaat, J.F. Soeder, "Evaluation of a Microprocessor Implementation of the F100 Multivariable Control," NASA TM-87130, 1986.

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

7. M.A. Mackin, J.F. Soeder, "Floating-Point Function Generation Routines for 16-Bit Microcomputers," NASA TM-83783, 1984.

8. J.C. DeLaat, "A Real-Time FORTRAN Implementation of a Sensor Failure Detection, Isolation, and Accommodation Algorithm," in 1984 American Control Conference, San Diego, CA: IEEE, 1984, pp. 572-573.

9. J.F. Soeder, "MINDS A Microcomputer Interactive Data System for 8086-Based Controllers," NASA TP-2378, 1985.

10. J.J. Sheskin, J.F. Soeder, "PMID a Software Package for Plotting Interactive Data," Proceedings of the Seventh National Conference on Computers and Industrial Engineering, 1985.

11. J.R. Szuch, K. Seldner, and D.S. Cwynar, "Development and Verification of a Real-Time, Hybrid Computer Simulation of the F100-PW 00(3) Turbofan Engine," NASA TP-1034, 1977.

12. K. Melcher, J.C. DeLaat, W.C. Merrill, L. Oberle, J. Schaefer, and G. Sadler, "A Personal Computer Based Sensor Failure Simulator," NASA TM-87271, 1986.

TABLE I. - STEADY-STATE ESTIMATION ACCURACY RESULTS (PERCENT OF NOMINAL), NO SENSOR FAILURES

Operating point			Estimation error, percent of nominal				
Altitude, ft	Mach number	PLA, deg	N1	N2	PT4	PT6	FTIT
10K	0.6	83	0.43	0.11	-3.16	-0.53	0.11
10K	.9	50	-.06	.16	-.21	1.53	.11
30K	.9	83	.42	.28	-1.36	-.69	.04
45K	.9	60	-.12	-.21	-1.87	-1.45	.04
10K	1.2	83	.17	.11	-1.36	.33	.11
55K	2.2	83	-.33	.54	5.64	-2.48	-.05

TABLE II. - MAXIMUM RESIDUAL VALUE (PERCENT OF NOMINAL) IN RESPONSE TO PLA PULSE INPUT, NO SENSOR FAILURES

Operating point		Estimation error, percent of nominal				
Altitude, ft	PLA, deg	N1	N2	PT4	PT6	FTIT
10K	0.6	3.57	0.81	6.50	12.55	5.78
30K	.9	1.47	.74	4.48	13.08	5.49
10K	.9	4.30	1.13	5.22	14.98	5.68

TABLE III. - AVERAGE RESIDUAL ABSOLUTE VALUE (PERCENT OF NOMINAL) IN RESPONSE TO PLA PULSE INPUT, NO SENSOR FAILURES

Operating point		Estimation error, percent of nominal				
Altitude, ft	PLA, deg	N1	N2	PT4	PT6	FTIT
10K	0.6	0.77	0.24	1.67	2.33	1.44
30K	.9	.63	.28	.92	2.94	1.64
10K	.9	.60	.42	.78	2.98	1.39

TABLE IV. - MAXIMUM HYPOTHESIS VALUE IN RESPONSE TO PLA PULSE INPUT, NO SENSOR FAILURES

Operating point		Maximum hypothesis value				
Altitude, ft	PLA, deg	N1	N2	PT2	PT6	FTIT
10K	0.6	0.8748	0.3420	0.2862	0.5976	0.0797
30K	.9	.2413	.1456	.0755	.3050	.0420
10K	.9	1.0250	.2361	.1616	1.3400	.0590

TABLE V. - AVERAGE HYPOTHESIS ABSOLUTE VALUE IN RESPONSE TO PLA PULSE INPUT, NO SENSOR FAILURES

Operating point		Average hypothesis value				
Altitude, ft	PLA, deg	N1	N2	PT2	PT6	FTIT
10K	0.6	0.0861	0.0566	0.0441	0.0587	0.0100
30K	.9	.0603	.0374	.0060	.0445	.0074
10K	.9	.0881	.0467	.0384	.1240	.0103

TABLE VI. - MINIMUM DETECTABLE BIAS FAILURE (PERCENT OF NOMINAL)

Operating point			Minimum bias failure				
Altitude, ft	Mach number	PLA, deg	N1	N2	PT4	PT6	FTIT
10K	0.6	50	3.47	2.60	6.39	12.02	12.00
10K	.6	83	3.41	2.67	3.85	7.74	8.75
30K	.9	50	3.43	3.10	10.92	18.10	12.36
30K	.9	83	3.25	2.73	6.99	13.24	9.41
10K	.9	50	3.54	2.60	6.17	9.74	12.17
10K	.9	83	2.91	2.66	2.86	6.40	8.72

TABLE VII. - MINIMUM DETECTABLE DRIFT FAILURE (PERCENT OF NOMINAL/SEC)

Operating point			Minimum drift failure				
Altitude, ft	Mach number	PLA, deg	N1	N2	PT4	PT6	FTIT
10K	0.6	50	1.16	0.87	1.28	3.20	5.60
10K	.6	83	1.22	.95	.38	1.55	4.08
30K	.9	50	1.49	.89	2.02	3.95	5.77
30K	.9	83	1.50	.97	1.55	2.65	4.39
10K	.9	50	1.18	.87	1.14	2.13	5.68
10K	.9	83	1.21	.95	.52	1.28	4.07

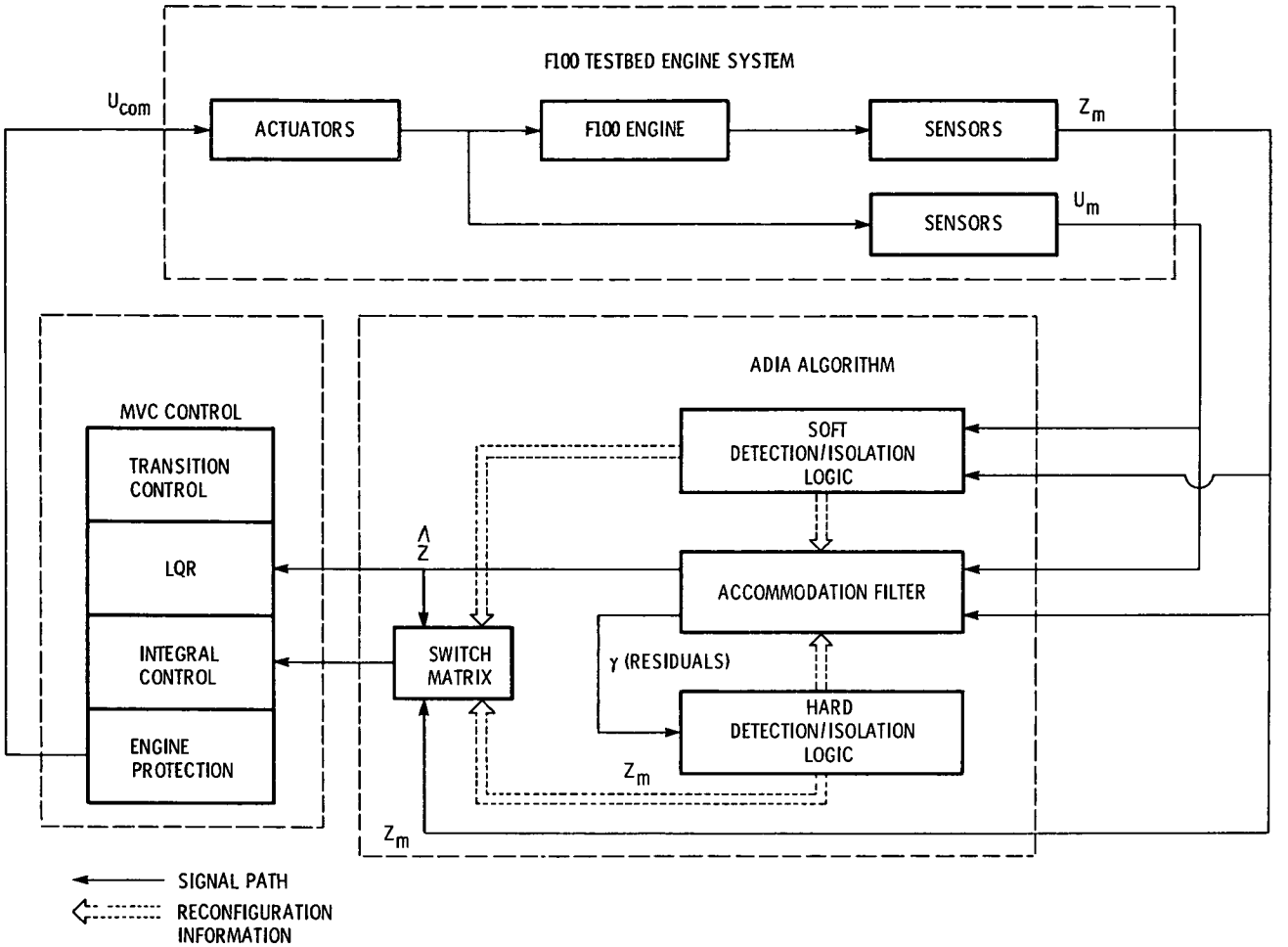

Figure 1. - ADIA/F100 testbed system.

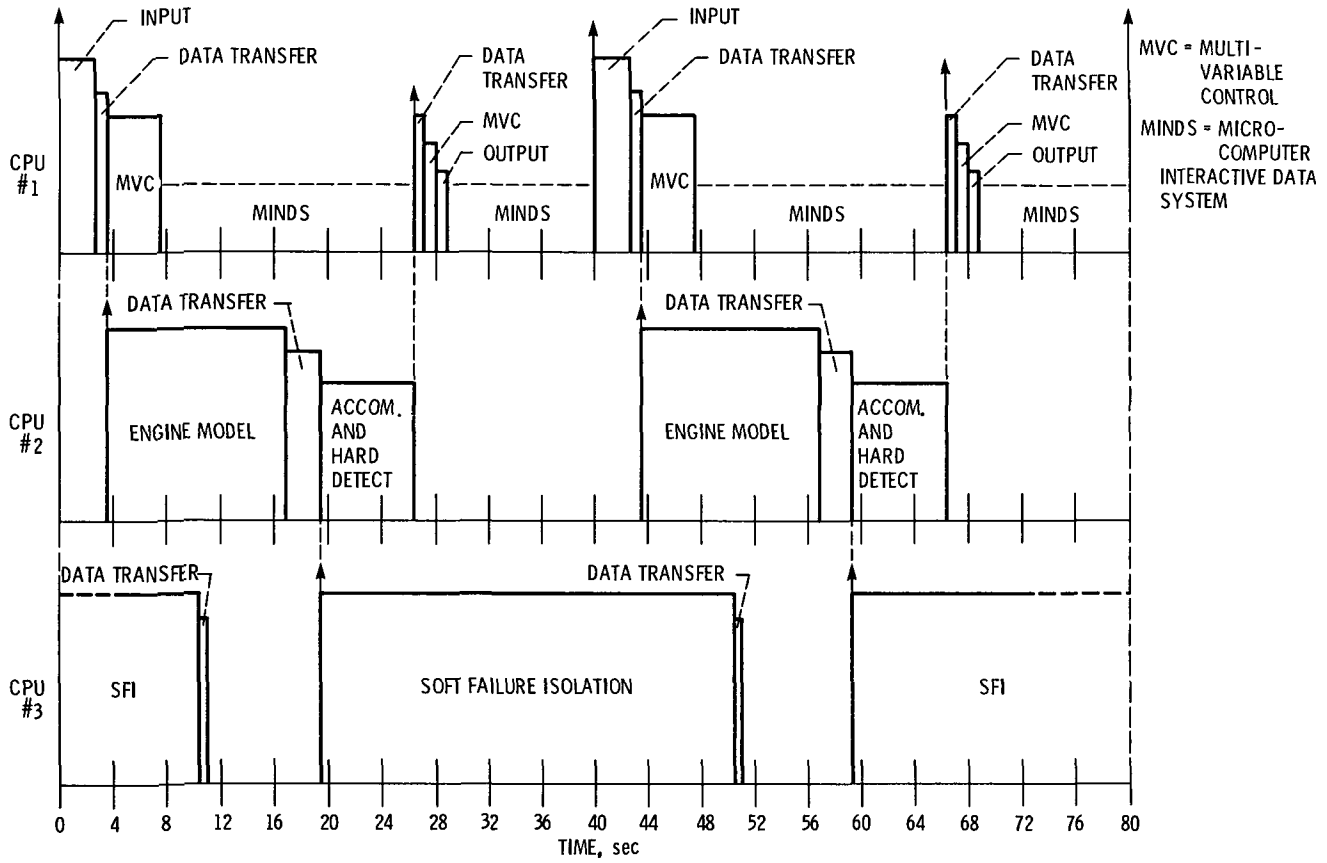

Figure 2. - ADIA timing - 8 MHz MSC 8186.

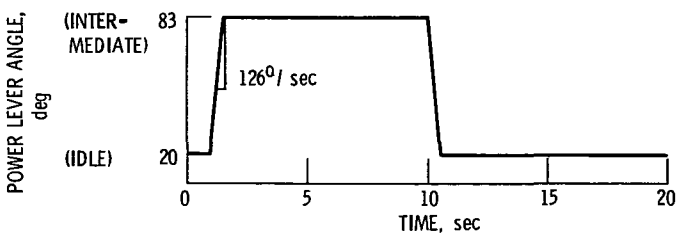

Figure 3. - Idle to intermediate power pulse transient used to generate transient results.

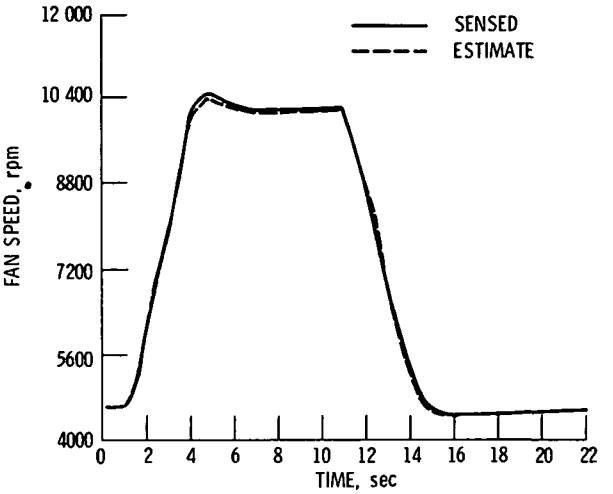

Figure 4. - Sensed and estimated fan speed at a 10K/.6 operating condition.

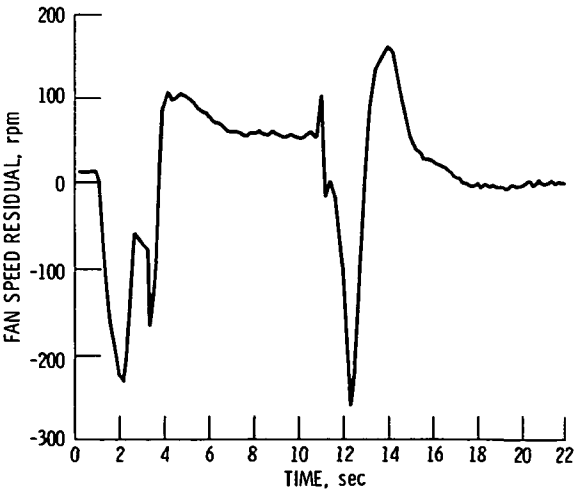

Figure 5. - Fan speed residual at a 10K/.6 operating condition.

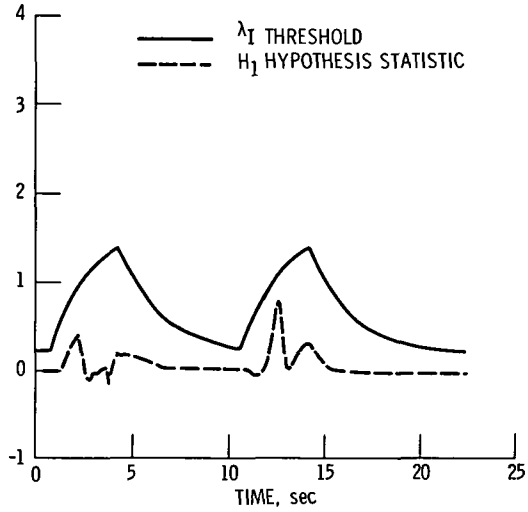

Figure 6. - Fan speed hypothesis statistic and detection threshold for 10K/.6 operating condition.

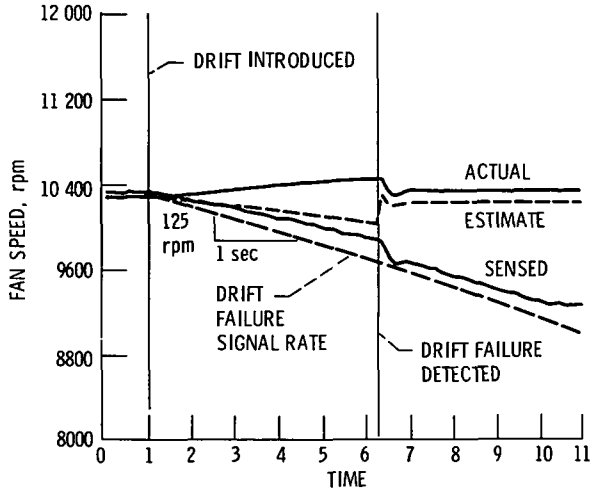

Figure 7. - Sensed estimated and actual fan speed for a drift failure (failure time = 1 sec, detect time = 6.5 sec) operating point is 10K/.9183. Drift rate = 125 rpm/sec (1.21%/sec of nominal).

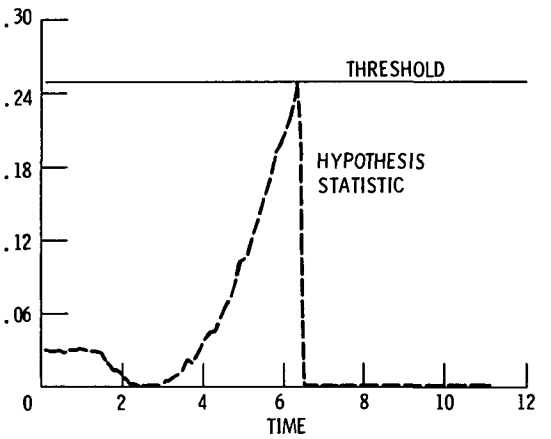

Figure 8. - Hypothesis statistic and threshold for fan speed drift failure of 125 rpm/sec at 10K/.9183 operating point.

1 Report No	2 Government Accession No	3 Recipient's Catalog No
NASA TM-87289		

4 Title and Subtitle	5 Report Date
A Real-Time Simulation Evaluation of an Advanced Detection, Isolation and Accommodation Algorithm for Sensor Failures in Turbine Engines	6 Performing Organization Code: 505-62-01

7 Author(s)	8 Performing Organization Report No
Walter C. Merrill and John C. DeLaat	E-2995
	10 Work Unit No

9 Performing Organization Name and Address	11 Contract or Grant No
National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135	13 Type of Report and Period Covered: Technical Memorandum

12 Sponsoring Agency Name and Address	14 Sponsoring Agency Code
National Aeronautics and Space Administration Washington, D.C. 20546	

15 Supplementary Notes

Prepared for the 1986 American Control Conference, sponsored by the Institute of Electrical and Electronics Engineers, Seattle, Washington, June 18-20, 1986.

16 Abstract

An Advanced sensor failure Detection, Isolation, and Accommodation(ADIA) algorithm has been developed for use with an aircraft turbofan engine control system. In a previous paper the authors described the ADIA algorithm and its real-time implementation. This paper discusses subsequent improvements made to the algorithm and implementation, and presents the results of an evaluation. The evaluation used a real-time, hybrid computer simulation of an F100 turbofan engine.

17 Key Words (Suggested by Author(s))	18 Distribution Statement
Sensor failures; Detection; Isolation; Accommodation; Analytical redundancy; Real-time feedback control; Control systems	Unclassified - unlimited STAR Category 07

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

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