NASA Technical Paper 2740

July 1987

NASA-TP-2740 19870015898

Advanced Detection, Isolation, and Accommodation of Sensor Failures—Real-Time Evaluation

Walter C. Merrill,
John C. DeLaat, and
William M. Bruton

NASA

NASA
Technical
Paper
2740

1987

Advanced Detection, Isolation, and Accommodation of Sensor Failures—Real-Time Evaluation

Walter C. Merrill,
John C. DeLaat, and
William M. Bruton

Lewis Research Center
Cleveland, Ohio

National Aeronautics
and Space Administration

**Scientific and Technical
Information Office**

Summary

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 using analytical redundancy to detect sensor failures. The results of a test-bed evaluation of an analytical-redundancy-based algorithm developed as part of the ADIA program are presented in this report. The ADIA program is organized into four phases: development, implementation, evaluation, and demonstration. This report describes the evaluation phase. This includes a validation of the ADIA algorithm for sensor failure detection, isolation, and accommodation effectiveness, documentation of algorithm performance, validation of the algorithm's real-time implementation, and establishment of a data base for the demonstration phase of the ADIA program.

The algorithm was evaluated in a test-bed system that consisted of the engine system, the multivariable control, and the ADIA algorithm. The engine system, including actuators and sensors, was simulated in real time on a hybrid computer. The multivariable control used was an existing control design based on linear quadratic regulator theory. The ADIA algorithm is based on hypothesis testing and can detect, isolate, and accommodate hard and soft sensor failures.

The evaluation is defined by a test matrix. The test matrix consists of engine evaluation operating conditions along one axis and the type of test performed along the other axis. Control performance with and without the ADIA algorithm was evaluated. Eight operating points were considered, and one or more of the 13 possible tests were performed at these operating points. Control performance was documented as was sensor failure detection, isolation, and accommodation performance. Minimum detectable levels of bias and drift failures were determined at all eight operating points for all sensors considered. Conclusions and recommendations based on the evaluation also are presented.

Introduction

Over the past 35 years hydromechanical implementations of turbine engine control systems have matured into highly reliable units. However, there is a trend toward greater engine complexity to meet ever-increasing engine performance requirements. Consequently the engine control too has become increasingly complex (fig. 1). 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 controls have to demonstrate the same levels of reliability as their hydromechanical predecessors, or better.

Thus 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 individual components. One of the least reliable of the control system components is the engine sensor. In fact some type of engine sensor redundancy will be required to achieve adequate control system reliability. One important type is analytical redundancy (AR). AR-based systems can have cost and weight savings over other redundancy approaches such as hardware redundancy.

Considerable work has been done in applying analytical redundancy to improve turbine engine control system reliability. Reference 1 surveys these accomplishments and defines several technology needs. These needs include (1) the ability to detect small (soft) failures, (2) real-time implementations of algorithms capable of detecting soft failures, (3) a comparison of algorithm complexity versus performance, (4) a full-scale demonstration of a soft-failure

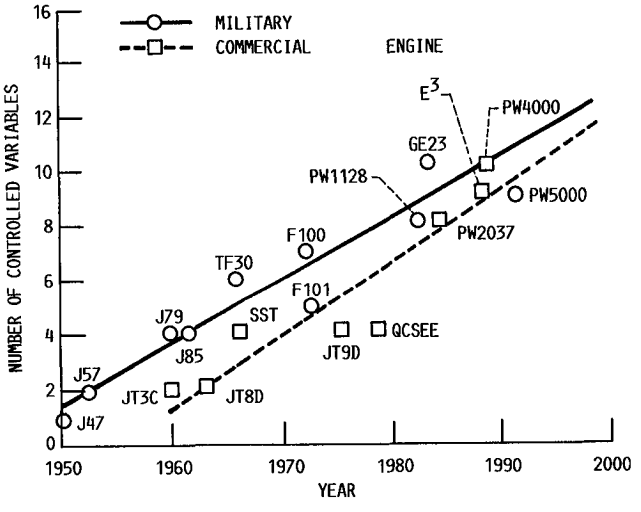


Figure 1.—Trends in control complexity of aircraft turbine engines.

detection capability, and (5) an evaluation of the pseudo-linearized modeling approach. The ADIA program addresses all of these technology needs.

The ADIA program is organized into four phases: development, implementation, evaluation, and demonstration. In the development phase (refs. 2 and 3) the ADIA algorithm was designed by using advanced filtering and detection methodologies. In the implementation phase (refs. 4 and 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 timeframe consistent with stable, real-time operation. This report describes the evaluation phase. In this phase algorithm performance was evaluated by using a real-time hybrid computer engine simulation. The objectives of the evaluation were to validate the algorithm for sensor failure detection, isolation, and accommodation (DIA) effectiveness, to document algorithm performance, to validate the algorithm's real-time implementation, and to establish a data base for the demonstration phase of the ADIA program. The ADIA algorithm will be demonstrated on a full-scale F100 engine in the NASA Lewis Research Center altitude test facility.

The report begins with a description of the test-bed system used in evaluating the ADIA algorithm. Then the ADIA algorithm and the implementation hardware are described. Next the results of the evaluation are presented. Finally conclusions and recommendations for further work are given.

Test-Bed System

The ADIA algorithm was evaluated in a test-bed system (fig. 2) consisting of the engine system, the multivariable control algorithm, and the ADIA algorithm. The ADIA algorithm is described in the next main section.

Engine System

The engine system consisted of an F100 turbofan engine, the control actuators, and the sensors. The F100 turbofan engine is a high-performance, low-bypass-ratio, twin-spool turbofan engine. The test-bed engine has five controlled inputs, five sensed outputs, and four sensed environmental variables. These variables are defined as follows:

Controlled engine inputs $\mathbf{U}_{com}$ and $\mathbf{U}_m$

WF	main combustor fuel flow
AJ	exhaust nozzle area
CIVV	fan inlet variable vanes

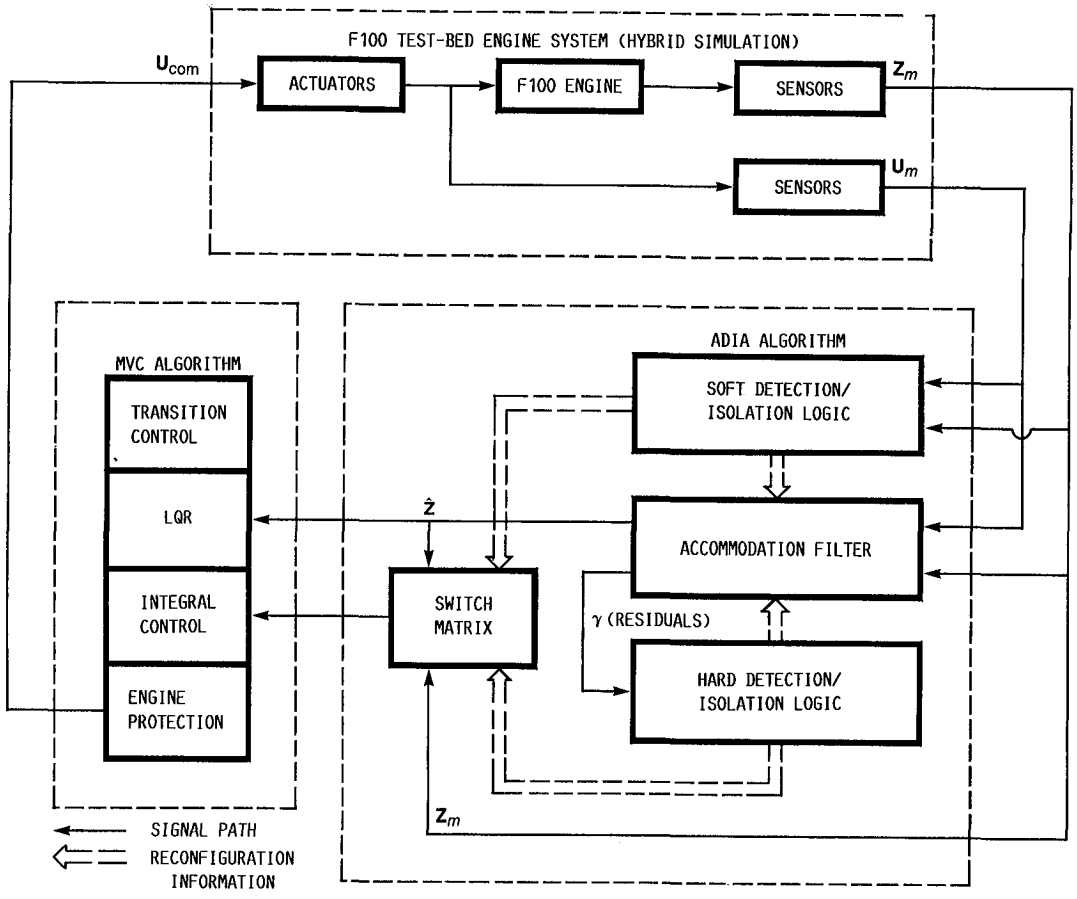


Figure 2.—Test-bed system.

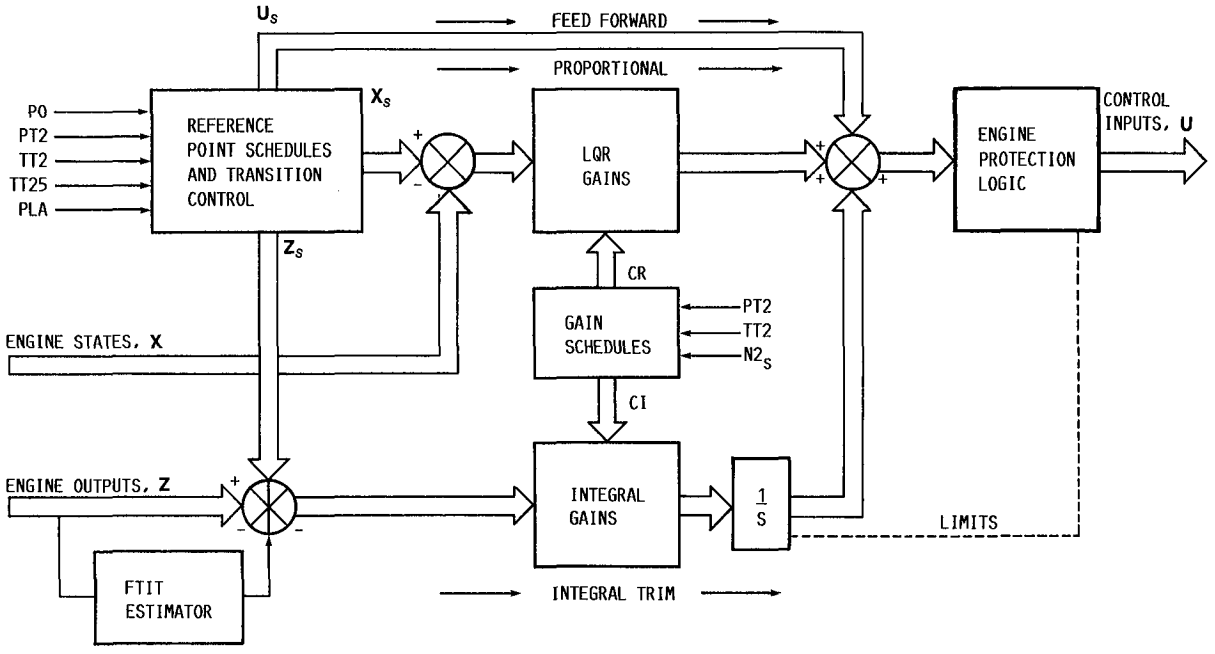


Figure 3.—Structure of F100 multivariable control.

RCVV rear compressor variable vanes
BLEED compressor bleed

Sensed engine outputs $\mathbf{Z}_m$

N1 fan speed
N2 compressor speed
PT4 combustor pressure
PT6 exhaust nozzle pressure
FTIT fan turbine inlet temperature

Sensed environmental variables $\mathbf{E}_m$

P0 ambient (static) pressure
PT2 fan inlet (total) pressure
TT2 fan inlet temperature
TT25 compressor inlet temperature

Strictly speaking, TT25 is an engine output variable. However, since TT25 is used only as a scheduling variable in the control (like TT2), it is called an environmental variable. Also, TT25 sensor failures are not covered by the ADIA logic.

Multivariable Control System

The multivariable control (MVC) system (fig. 3) is essentially a model following proportional plus integral control. The components of the control are the reference point schedules, the transition control schedules, the proportional control logic, the integral control logic, and the engine protection logic. The reference point schedules generate a desired engine operating point given the pilot's commanded power lever angle (PLA). The transition logic generates rate-limited command trajectories for smooth transition between steady-state operating points. The proportional and integral control logic minimizes transient and steady-state deviations from the commanded trajectories. The engine protection logic limits the size of the commanded engine inputs. This control is more completely described in reference 6. The control modes in this logic normally use fuel to set engine fan speed and use 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.

Algorithm

The ADIA algorithm detects, isolates, and accommodates sensor failures in turbofan engine control systems. It was originally developed for NASA Lewis under contracts NAS3-22481 and NAS3-23282 by Pratt & Whitney Aircraft with subcontractor Systems Control Technology (refs. 2 and 3). The algorithm incorporates advanced filtering and detection logic and is general enough to be applied to different engines or to other types of control systems.

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 test-bed system in figure 2. 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 general concept is shown is block diagram form in figure 4. Here, in a 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 $\mathbf{Z}$ are used by the control law. When a sensor

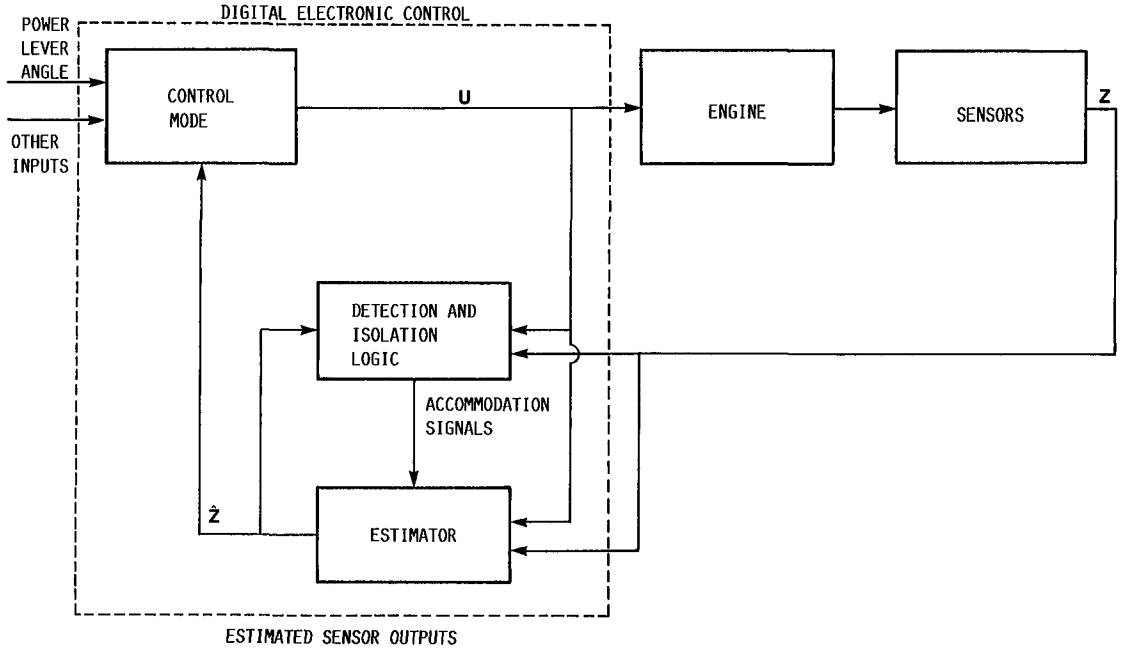


Figure 4.—Advanced detection, isolation, and accommodation concept.

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 estimator. The estimator then removes the faulty measurement from further consideration. The estimator, 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 failure.

The ADIA algorithm inputs are the measured engine inputs $\mathbf{U}_m(t)$ (fuel flow, nozzle area, compressor inlet guide vane angle, rear compressor variable vane angle, and bleed flow) and the measured engine outputs $\mathbf{Z}_m(t)$ (fan speed, compressor speed, combustor pressure, augmentor pressure, and fan turbine inlet temperature). The algorithm outputs $\hat{\mathbf{Z}}(t)$ are optimal estimates of the engine outputs $\mathbf{Z}(t)$. The measured environmental variables $\mathbf{E}_m$ are also used to schedule engine model parameters. The outputs of the algorithm, the estimates $\hat{\mathbf{Z}}(t)$, are used as input to the proportional (linear quadratic regulator, or LQR) part of the control. During normal-mode operation engine measurements are used in the integral control. 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.

Engine Model

The performance of the accommodation filter and the detection and isolation logic is strongly dependent on a model of the engine. The model used has a linear state-space structure, and the base points are nonlinear functions of various engine variables.

$$\dot{\mathbf{X}} = F(\mathbf{X} - \mathbf{X}_b) + G(\mathbf{U} - \mathbf{U}_b)$$

$$\mathbf{Z} = H(\mathbf{X} - \mathbf{X}_b) + D(\mathbf{U} - \mathbf{U}_b) + \mathbf{Z}_b$$

Here the subscript represents the base point (steady-state point) and $\mathbf{X}$ is the 4×1 model state vector, $\mathbf{U}$ the 5×1 control vector, and $\mathbf{Z}$ the 5×1 output vector. The F, G, H, and D matrices are the appropriately dimensioned system matrices. The system matrices and the model base points were determined at 109 operating points throughout the flight envelope. Three variables are sufficient to completely define an operating point—power lever angle (PLA), altitude, and Mach number. An alternative definition set is PLA, inlet pressure (PT2), and inlet temperature (TT2). Figure 5 shows some of those 109 points as a function of the altitude/Mach number envelope at 83° PLA. In figure 6 the same points are shown at 83° PLA as a function of engine inlet conditions. The second envelope is the more appropriate format for ensuring that all significant model dynamics are considered by adequately spanning the entire envelope with model points. Once system matrices are determined at all of the 109 operating points, the individual matrix elements are corrected by the engine inlet condition $\mathbf{E}_m$ and scheduled as nonlinear functions of $\mathbf{Z}$. These functions are given in reference 2.

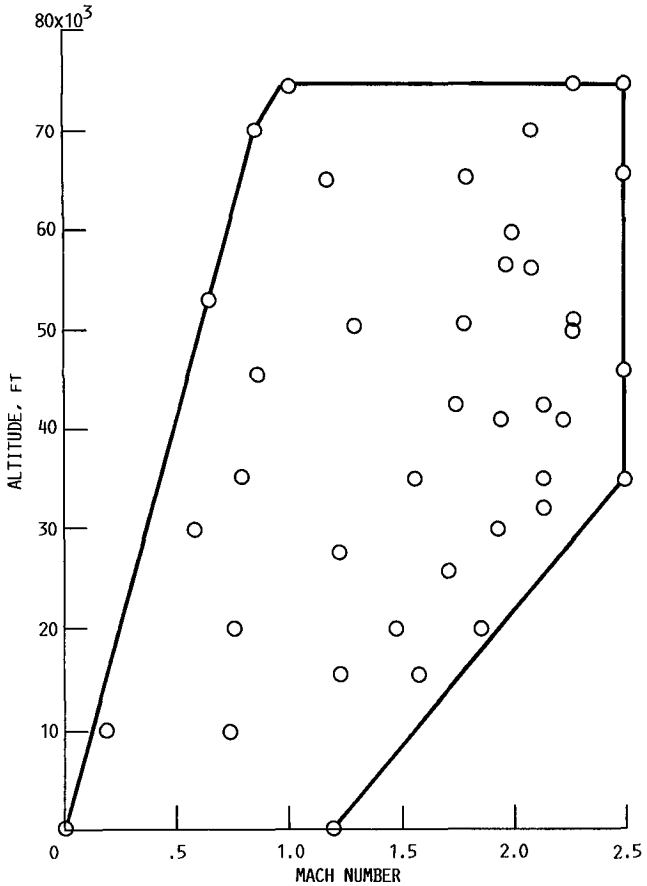


Figure 5.—Model operating points at 83° PLA altitude/Mach number envelope.

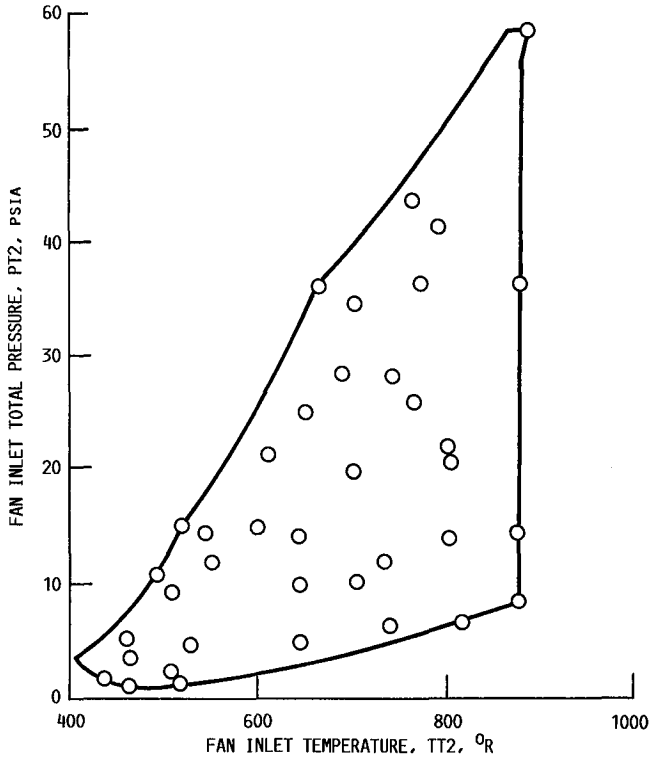


Figure 6.—Model operating points at 83° PLA PT2/TT2 envelope.

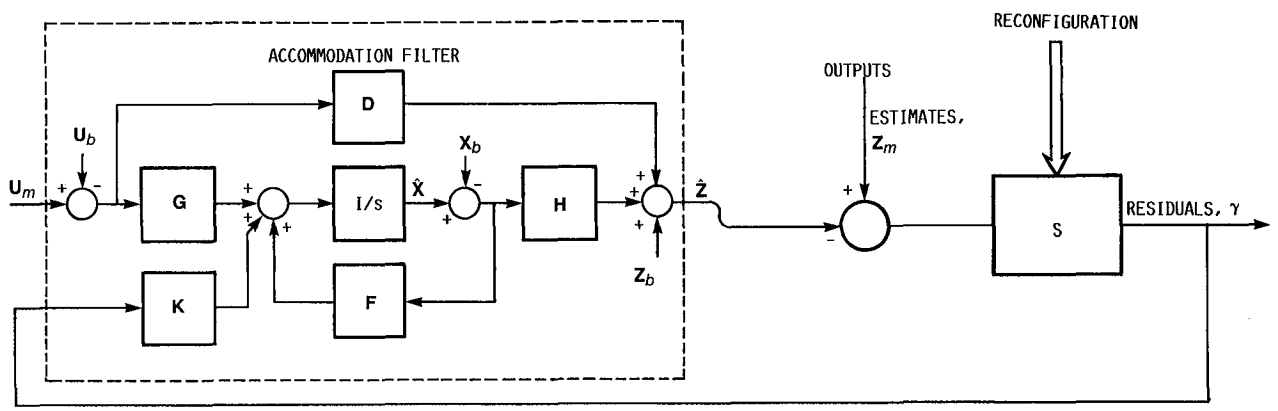


Figure 7.—Accommodation filter structure.

Accommodation Filter

The accommodation filter (fig. 7) incorporates the engine model along with a Kalman gain update to generate estimates of the engine outputs $\hat{\mathbf{Z}}$ and states $\hat{\mathbf{X}}$ as follows:

$$\dot{\hat{\mathbf{X}}} = F(\hat{\mathbf{X}} - \mathbf{X}_b) + G(\mathbf{U}_m - \mathbf{U}_b) + K\gamma$$

$$\hat{\mathbf{Z}} = H(\hat{\mathbf{X}} - \mathbf{X}_b) + D(\mathbf{U}_m - \mathbf{U}_b) + \mathbf{Z}_b$$

$$\gamma = \mathbf{Z}_m - \hat{\mathbf{Z}}$$

where K is the Kalman gain matrix and γ is the residual vector. Like the system matrices the elements of K are corrected by $\mathbf{E}_m$ and scheduled as nonlinear functions of $\mathbf{Z}$. An improvement that was added to the accommodation filter was the incorporation of integral action to improve steady-state accuracy of the FTIT estimate $\hat{\mathbf{Z}}_5$.

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 on the dynamically faster FTIT estimate. The FTIT limiting control has integral action; therefore a high degree of steady-state accuracy in the FTIT

estimate is required to ensure satisfactory control. This accuracy is accomplished by augmenting the filter with the following additional state and output equations:

$$\hat{b} = K_6\gamma$$

$$\text{FTIT} = \hat{Z}_5 + b$$

where K_6 is a gain matrix, b is the temperature bias, $\hat{Z}_5$ is the unbiased temperature estimate, and γ is the vector of residuals from the accommodation filter. The addition of these dynamics, although improving FTIT estimation accuracy, results in a larger minimum detectable FTIT drift failure rate. Concatenating the temperature bias state to the filter state vector yields the same filter equations with the following replacements:

$$\hat{\mathbf{X}} \leftrightarrow \left[\frac{\hat{\mathbf{X}}}{b}\right]$$

$$F \leftrightarrow \left[\begin{array}{c|c} F & 0 \\ \hline 0 & 0 \end{array}\right] \qquad \left[G \leftrightarrow \frac{G}{0}\right]$$

$$K \leftrightarrow \left[\frac{K_6}{K}\right]$$

$$H \leftrightarrow \left[H \middle| \frac{0}{1}\right]$$

$$D \leftrightarrow D$$

This filter structure, which includes the FTIT bias state, is the structure used in the accommodation filter and all the hypothesis filters in the soft-failure detection and isolation logic.

After the detection and isolation of a sensor failure the accommodation filter is reconfigured by a switching matrix (fig. 7). This matrix is defined as

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

When a sensor failure has been isolated, the filter is reconfigured by setting the appropriate diagonal matrix element to zero. For example, if a compressor speed sensor failure (N2) has been isolated, the switch matrix becomes

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The effect of this reconfiguration is to force γ_2 equal to 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-failure detection and isolation logic (fig. 8) compares the absolute value of each component of the residual with its own threshold. If the residual absolute value is greater than the threshold, 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-failure detection threshold values (table I) are twice the magnitude of these adjusted standard deviations.

The failure is accommodated by reconfiguring the switch matrices in the accommodation filter and all of the hypothesis filters in the soft-failure detection logic.

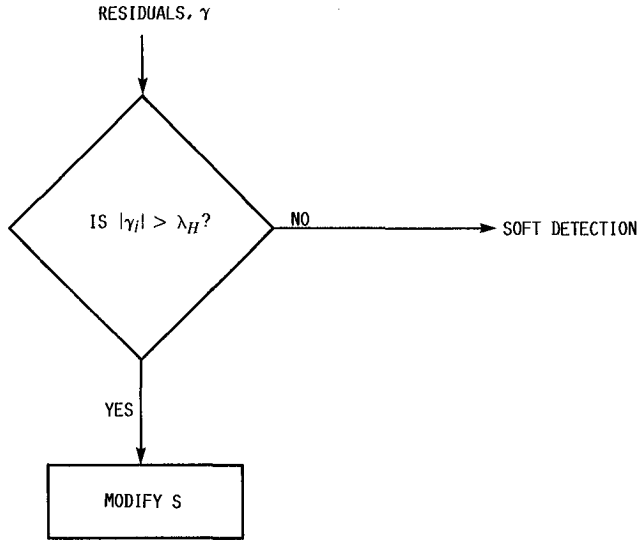


Figure 8.—Hard-failure detection logic.

TABLE I.—HARD-FAILURE DETECTION THRESHOLD MAGNITUDES

Sensor	i	Adjusted standard deviation, σ_i	Detection threshold, λ_H
N1	1	300 rpm	600 rpm
N2	2	400 rpm	800 rpm
PT4	3	30 psi	60 psi
PT6	4	5 psi	10 psi
FTIT	5	250 °R	500 °R

Soft-Failure Detection and Isolation Logic

The soft-failure detection and isolation logic consists of multiple-hypothesis-based testing. Each hypothesis is implemented by using a Kalman filter. The soft detection and isolation logic structure (fig. 9) consists of six hypothesis filters, one for normal mode operation and five for the failure modes (one for each engine output sensor). 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. Each hypothesis filter generates a statistic or likelihood called the weighted sum of squared residuals (WSSR) statistic, which is defined below. This statistic is subtracted from the normal-mode WSSR filter statistic. The maximum of the results is compared with the soft-failure detection and isolation threshold. If the threshold is exceeded, a failure is declared. 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 the H_1 likelihood will be the maximum, and it will be compared with the threshold to detect the failure.

Each hypothesis filter is identical in structure (fig. 10) to the accommodation filter except for the switch matrix S_i. Each hypothesis filter generates a unique residual vector γ_i. Assuming Gaussian sensor noise, each sample of γ_i has a certain likelihood or probability

$$L_i = p_i(\gamma_i) = ke^{-\text{WSSR}_i}$$

where k is a constant and $\text{WSSR}_i = \gamma_i^T \Sigma^{-1} \gamma_i$ with $\Sigma = \text{diag}\,(\sigma_i^2)$. The σ_i are the adjusted standard deviations defined in table I. These standard deviation values scale the residuals to unitless quantities that can be summed in the WSSR statistic. The WSSR statistic is smoothed to remove gross noise effects by a first-order lag with a time constant of 0.1 sec. When the log of the ratio of likelihoods is taken,

$$\text{LR}_i = \log\left(\frac{L_i}{L_o}\right) = \text{WSSR}_o - \text{WSSR}_i$$

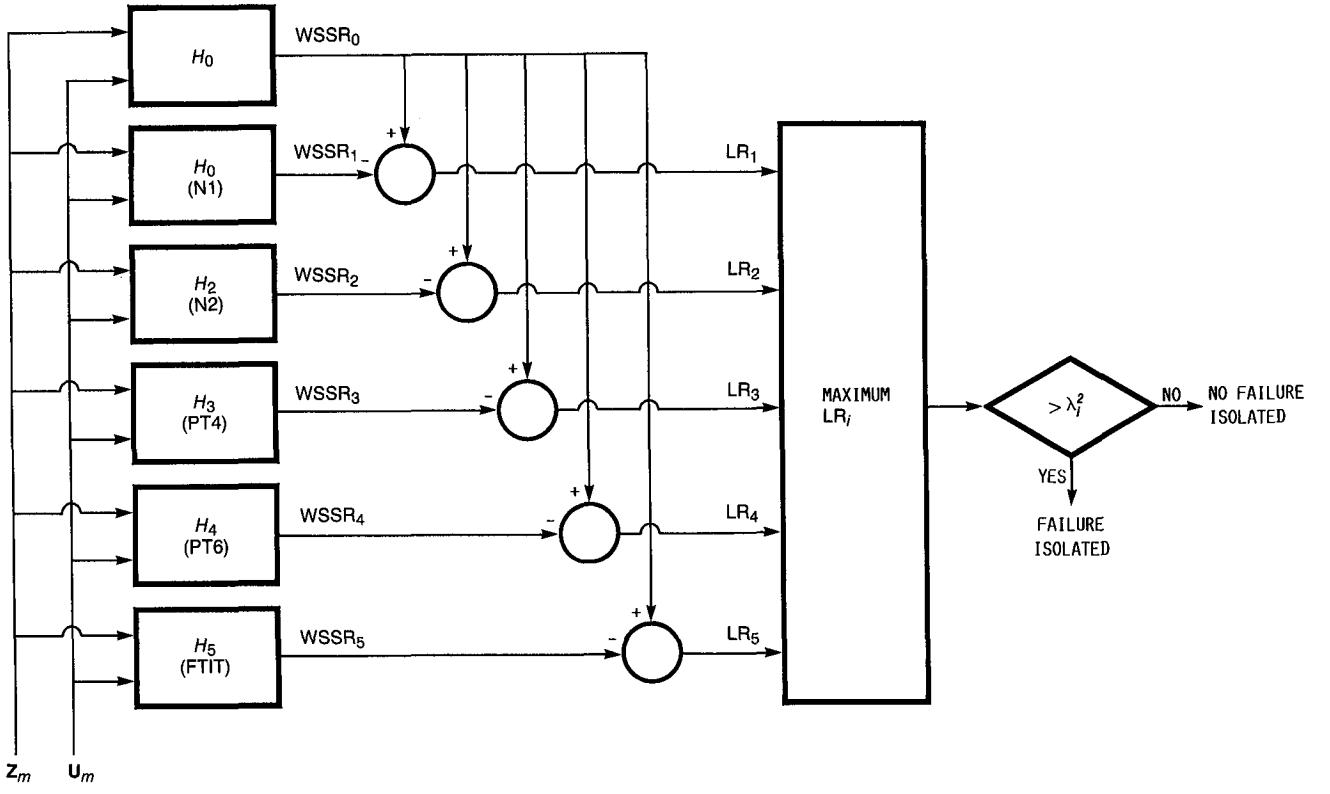


Figure 9.—Soft-failure isolation logic.

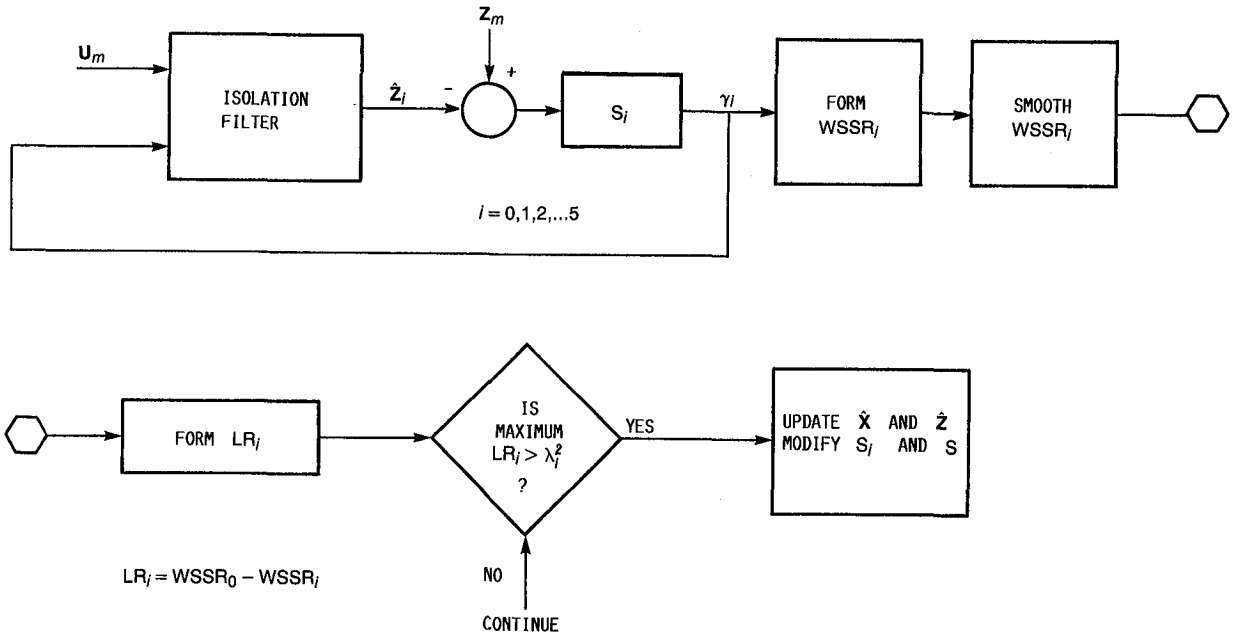


Figure 10.—Hypothesis filter structure.

If the maximum log likelihood ratio exceeds the threshold, a failure is detected and isolated and accommodation occurs. Three steps are taken for accommodation. First, all seven of the filter (one accommodation and six hypothesis) switching matrices are reconfigured to account for the detected failure mode. Second, the states and estimates of all seven filters are updated to the correct values of the hypothesis filter that corresponds to the failed sensor. Third, the interface switch matrix is reconfigured.

Adaptive Threshold

Since the WSSR statistic is the sum of Gaussian variables squared, it has a chi-squared distribution. Initially the soft-failure detection and isolation threshold is determined by standard statistical analysis of this distribution to set the confidence level of false alarms and missed detections. Next the threshold is 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 level. 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 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 $\tau = 2$ 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. This resulted in an 80 percent reduction in the detection and isolation threshold λ_i^2. The adaptive threshold logic is illustrated in figure 11 for a PLA pulse transient.

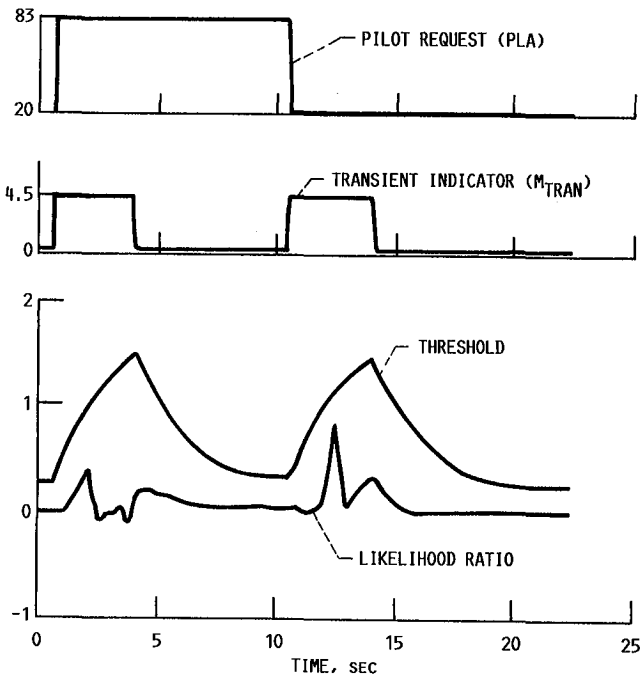


Figure 11.—Soft-failure detection threshold.

Evaluation System

The ADIA algorithm was evaluated by using a real-time hybrid computer simulation of the F100 engine, a microprocessor-based control computer including accompanying interface and monitoring hardware and interactive data acquisition software, and the sensor failure simulator (SFS).

Hybrid Computer Simulation

The F100 engine hybrid simulation is a nonlinear, real-time, 32nd-order model that includes sensor and actuator dynamics. Differential equations, which are based on lumped-parameter thermodynamic and mechanical conservation 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. The simulation was derived from a digital computer program developed by the engine manufacturer and from engine test data obtained during the MVC test program. Simulation accuracy is 2 to 4 percent of nominal for steady-state performance and is good for transient performance. A complete description of the simulation and its accuracy performance is given in reference 7.

Control Computer

The control, interface, and monitoring (CIM) unit contains the microcomputer used to implement the combined MVC–ADIA algorithm in real time. The CIM unit also contains hardware and cabling to provide a flexible interface to and from the engine or engine simulation being controlled. A monitoring system in the CIM unit allows the signals between the microcomputer and the controlled engine to be checked for correctness. The interface and monitoring functions of the CIM unit are described in detail in reference 8. The remainder of this subsection describes the control microcomputer hardware and software.

Implementing the MVC–ADIA algorithm required integrating the ADIA algorithm with an existing microcomputer implementation of the F100 multivariable control (MVC). The update interval of the microprocessor-based MVC implementation was 22 msec. The F100 engine system dynamics required that the combined MVC–ADIA algorithm update interval be 40 msec or less.

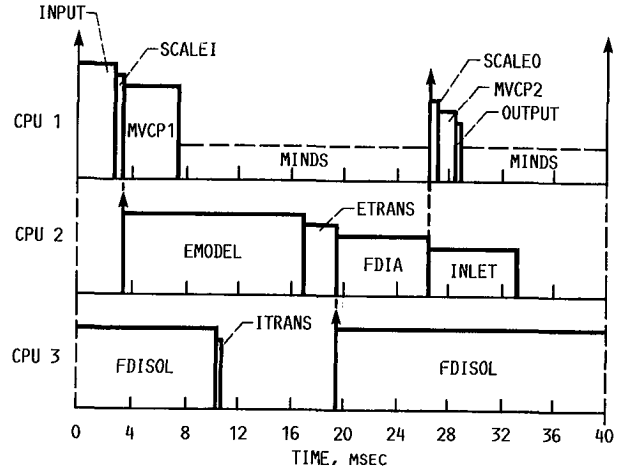
The microcomputer implementation of the MVC algorithm had been developed by porting the minicomputer implementation of the MVC algorithm used for the F100 MVC program to an Intel 8086 microprocessor-based control microcomputer. The ADIA algorithm was then merged with this MVC implementation to give a full microcomputer implementation of the control algorithm with sensor analytical redundancy. The resulting control microcomputer was also based on the Intel 8086 microprocessor architecture. However, in order to implement the combined algorithm and satisfy the update interval requirement of 40 msec necessary for stable engine operation, multiple processors operating in parallel were used.

Initially only the normal-mode accommodation filter and the hard-failure detection logic of the ADIA algorithm were added to the MVC algorithm. For this initial configuration a second 8086-based central processing unit (CPU), running in parallel, was added to the CPU used to implement the MVC alone. The CPU's used were Intel 86/30 single-board computers. Data were transferred between CPU's through dual-ported memory, and synchronization between CPU's was achieved through interrupts. The software for the combined MVC–ADIA algorithm was partitioned so that the ADIA software ran on the second CPU while the MVC algorithm remained intact on the first CPU. This straightforward way to partition the algorithm allowed the parallel-processing mechanism to be thoroughly evaluated. It was assumed that the soft-failure detection and isolation logic would 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 detection logic. Because the soft-failure isolation logic is complex and since we felt that there might be some benefit to running the soft isolation logic in parallel with the soft detection logic, a third CPU was added to implement the soft isolation logic. The soft detection logic was added to the second CPU. Data were transferred and synchronized in the same manner as with the two-CPU implementation. Most recently, the three 8086-based CPU's were replaced with 80186-based CPU's. These are Monolithic Systems MSC 8186 single-board computers. The new CPU's are software compatible with the old CPU's but are considerably faster. The relative timing for the three CPU's is shown in figure 11.

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 1. When the MVC was implemented on a microcomputer, assembly language programming using fixed-point arithmetic was necessary to achieve real-time execution of the algorithm. With the development of efficient floating-point coprocessing hardware, in this case the Intel 8087, came the capability of implementing real-time controls in floating-point arithmetic. Thus most of the ADIA algorithm running on CPU's 2 and 3 is programmed in floating-point arithmetic and the application-oriented language Fortran. Fortran was chosen because the ADIA as developed was coded in Fortran and because a fairly good compiler was available for the 8086–8087. The advantages of using floating-point arithmetic and an application language such as Fortran rather than programming in fixed-point assembly language as was used for the MVC are well known.

The primary disadvantage to using an application language is that it generally produces less efficient object code than the equivalent functions programmed in assembly language. Execution efficiency is critical for real-time control systems



CPU 1

INPUT	A/D CONVERSION AND SCALING OF ENGINE MEASUREMENTS
SCALEI	CONVERSION OF MEASUREMENTS TO FLOATING POINT AND TRANSFER TO CPU 2
MVCP1	MVC PART 1: REFERENCE POINT SCHEDULE, TRANSITION CONTROL, AND GAIN SCHEDULE CALCULATIONS
MINDS	INTERACTIVE DATA SYSTEM (SPARE TIME CALCULATION)
SCALEO	TRANSFER OF ESTIMATES AND RECONFIGURATION INFORMATION FROM CPU 2 AND CONVERSION TO FIXED POINT
MVCP2	MVC PART 2: PROPORTIONAL CONTROL, INTEGRAL CONTROL, AND ENGINE PROTECTION LOGIC
OUTPUT	UNSCALING AND D/A CONVERSION OF ENGINE INPUTS

CPU 2

EMODEL	ENGINE MODEL MATRIX AND BASE POINT CALCULATION
ETRANS	TRANSFER OF EMODEL INFORMATION TO CPU 3
FDIA	ACCOMMODATION FILTER AND HARD DETECTION LOGIC CALCULATIONS
INLET	MACH NUMBER AND ALTITUDE CALCULATION

CPU 3

FDISOL	HYPOTHESIS FILTERS AND SOFT ISOLATION LOGIC CALCULATIONS
ITRANS	TRANSFER OF SOFT ISOLATION INFORMATION TO CPU 3

Figure 12.—ADIA timing for 8-MHz MSC 8186.

including the MVC–ADIA. Thus for the ADIA, table lookup routines, which are written to take advantage of the 8087 architecture (ref. 9) and are executed frequently in the algorithm, and the hardware interface routines, which have no Fortran equivalent, 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. 10). As shown in figure 12, the entire MVC–ADIA algorithm now executes in less than the required 40 msec.

The programs for each of the CPU's are downloaded into the CPU's by using a commercially available disk operating system, CP/M–86. The *M*icrocontroller *IN*teractive *D*ata System (MINDS) is used for data acquisition (ref. 11). This software runs on CPU 1 in the spare time when the CPU is not executing the MVC algorithm (fig. 12). The package has both steady-state and transient data-taking capabilities and can

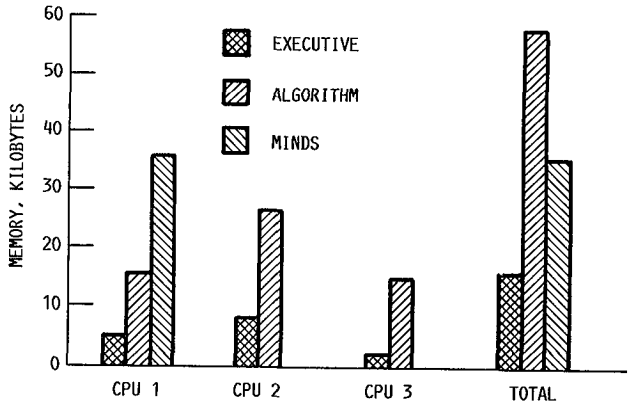


Figure 13.—MVC–ADIA memory requirements. Algorithm code requires 65 percent of total algorithm memory. Total memory, 110 kilobytes.

access any variable in the MVC or ADIA algorithm. 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. 12). The on-line transient data display of internal MVC and ADIA variables was an indispensable tool in the evaluation process.

The memory requirements for each of the three CPU's are shown in figure 13. Each CPU has, in addition to its share of the MVC–ADIA algorithm, an executive routine that maintains correct real-time operation of the total algorithm. The memory requirements for the algorithm and for the executive are shown for each CPU. In addition, the memory requirement for MINDS is shown for CPU 1. In all cases the code and the constants were about 65 percent, and the data and the variables about 35 percent, of the total memory required. Lastly figure 13 shows the total memory requirements for all executives, the total algorithm, and MINDS for all three CPU's combined.

Sensor Failure Simulator

The sensor failure simulator (SFS) provides an efficient means of modifying engine sensor signals to simulate sensor failures. The SFS unit consists of a personal computer driving discrete analog hardware. The personal computer allows a menu-driven, top-down approach to failure scenario retrieval, creation, editing, and execution. The SFS can simulate any of four basic sensor failure modes: scale-factor change, bias, drift, and noise. These failure modes are implemented in analog electronic hardware that is controlled by the personal computer. The SFS allows complete and repeatable control over the failure size and the timing of failure injection. Details of the SFS are given in reference 13.

Real-Time Evaluation

This section describes the evaluation of the ADIA algorithm using a hybrid-computer-based, real-time simulation of an

F100 engine. The objectives, the procedure, and the results of the evaluation are discussed.

Objectives

The first objective of the evaluation was to validate the operation and performance of the ADIA algorithm and its implementation. It was especially important to conduct this validation in a real-time environment in order to establish the feasibility and practicality of the implementation. The second objective was to document the performance of the algorithm over the envelope of the engine. The third objective was to establish a data base for comparison with results obtained during the demonstration phase of the program.

Procedure

The procedure for evaluating the algorithm is defined by the test matrix (table II). The different operating conditions (altitude/Mach number) used during the evaluation are across the top of the matrix, and the different tests conducted at these points are along the side. Both MVC only and ADIA–MVC evaluation tests are shown.

Operating conditions.—The rationale used in selecting the test matrix operating conditions was to duplicate as many conditions as possible used in the F100 Multivariable Control Program (ref. 6), to avoid high fan inlet pressures, and to reasonably span the envelope. This rationale was a compromise between taking advantage of previous results for comparison, limited-risk engine operation, and full-envelope validation. The test conditions selected are plotted on the engine face condition envelope in figure 14.

Test definitions.—The tests used in the evaluation were selected to completely define detection performance for five common failures modes. Also, tests were conducted to determine engine control performance with and without the ADIA algorithm and with and without engine sensor failures. The tests are summarized in table III.

TABLE II.—EVALUATION TEST MATRIX

Test	Operating condition, altitude (1000 ft)/Mach number							
	10/0.6	30/0.9	10/0.9	45/0.9	10/1.2	50/1.8	35/1.9	55/2.2
	Number of tests							
Sensor failure test:								
Hard	10							
Soft	10	10	10	5	5	5	5	5
Drift	10	10	10	5	5	5	5	5
Noise	2	2						
Scale	2	2						
Sequence	12	12						
Pulse	1	1	1	1	1			
Open	1	1						
Random	4	4						
Frequency	4	4						
MVC tests:								
MVC SS	7	3	3	3	2	1	1	1
MVC pulse	1	1	1	1	1			
Original/CIM comparison	1	1	1	1	1			
Single	5							

TABLE III.—TEST DEFINITIONS

Test	Description
ADIA-MVC evaluation	
Hard	Large bias failure
Soft	Small bias failure
Drift	Small drift failure
Noise	Random noise failure
Scale	Scale-factor bias change
Sequence	Sequence of successive output sensor failures
Pulse	Minimum-to-maximum-to-minimum transient power excursions. Maximum power level is maintained for 10 sec.
Open	Same as pulse test except that minimum power level is raised slightly, maximum power level is decreased slightly, and engine is controlled without using any sensed engine output information.
MVC-only evaluation	
MVC	Same as pulse test except that control is run exclusively with engine sensed output. ADIA estimates are not used in control.
MVC SS	Steady-state data at operating point
Bleed	Pulse test with bleed control disabled
Single	Pulse test with single sensor failure accommodated before initiating transient
Alt/Mach	Altitude/Mach number excursion from 10 000 ft/Mach 0.6 to 45 000 ft/Mach 0.9

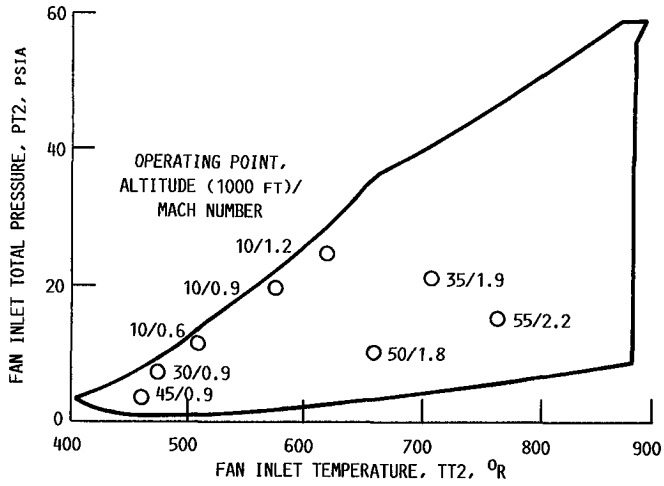


Figure 14.—Evaluation test conditions.

Results

Three types of real-time evaluation results are presented. The first shows the performance of the microprocessor-based MVC control. The second shows the accuracy of the Kalman-filter-based estimator. Finally the performance of the ADIA algorithm itself is given.

MVC performance.—The performance of the MVC control was evaluated without the ADIA logic. This evaluation demonstrated that the microprocessor-based implementation of the MVC control algorithm could accurately and safely control the test-bed engine. Steady-state and transient results were obtained at the test matrix operating conditions. In every case both the steady-state and transient accuracy results were good. The steady-state results are summarized in table IV.

Minimum and maximum steady-state error results at seven points in the 10 000-ft/Mach 0.6 operating condition are given graphically in figure 15 for N1 (minimum error) and in

TABLE IV.—STEADY-STATE MVC PERFORMANCE RESULTS

Operating condition			Engine output					
Altitude, ft	Mach number	Power lever angle, PLA, deg	N1	N2	PT4	PT6	FTIT	EPR
			Value, percent of nominal					
10×10^3	0.6	50	0.07	0.66	1.94	−0.18	6.71	−0.79
	.6	83	−.40	.08	−.31	−.48	.24	1.53
	.9	50	−.17	−.36	−2.66	.00	1.49	.13
	.9	83	−.72	.17	.75	−.07	−.41	−.07
	1.2	70	−.08	−.21	−.15	−1.35	.32	−.07
30	.9	50	−.07	−.52	−.11	−.37	7.20	−1.39
	.9	83	.09	−.20	.94	−.24	1.16	−.24
35	1.9	83	−.34	−.31	−2.47	−.86	−1.96	−.51
45	.9	70	−.16	−.31	0	−.71	−9.89	−1.45
50	1.8	83	−1.20	−2.01	−5.50	−2.18	−3.78	−1.93
55	2.2	83	.28	1.37	4.97	.11	.81	.11

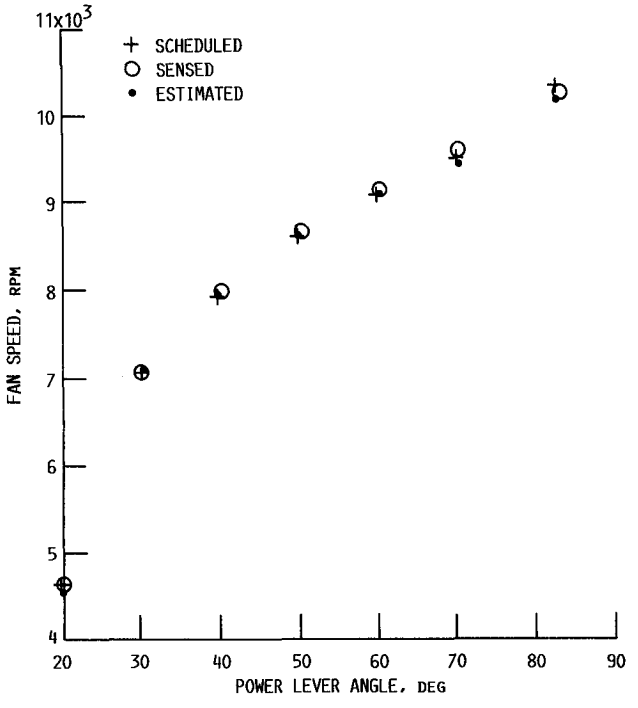


Figure 15.—Steady-state accuracy comparison (minimum error) for fan speed N1 at 10 000-ft/Mach 0.6 operating condition.

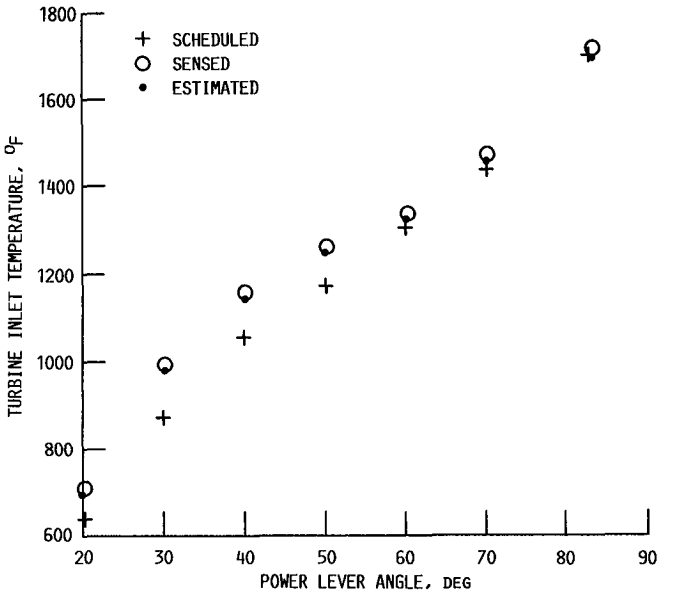


Figure 16.—Steady-state accuracy comparison (maximum error) for fan turbine inlet temperature FTIT at 10 000-ft/Mach 0.6 operating condition.

figure 16 for FTIT (maximum error). The error magnitudes for FTIT at PLA of 50° and 70° from table IV represent some inaccuracy in the steady-state control schedules. However, these inaccuracies will have no effect on control performance since the FTIT control schedule information is not used in the control at these PLA settings. Typical transient response examples at the 45 000-ft/Mach 0.9 operating condition are shown in figure 17 for N1 and in figure 18 for PT6 for pulse tests. Of the five pulse responses the 45 000-ft/Mach 0.9 operating condition represents worst-case engine control performance. It is, however, completely acceptable control response. Additionally the pulse test results at the five transient operating conditions were compared with results obtained during the original real-time evaluation of the MVC algorithm (ref. 7).

In this original evaluation a different computer (a minicomputer) was used to implement the control. Both good steady-state reference point accuracy and transient trajectory-following accuracy were demonstrated throughout the engine operating envelope. Although the minicomputer-based implementation used a slightly different principal control mode (nozzle area sets the airflow Mach number rather than the engine pressure ratio), this did not significantly affect the comparison. Thus if the two implementations compared closely, an additional level of confidence in the ability of the new implementation to accurately control the test-bed engine

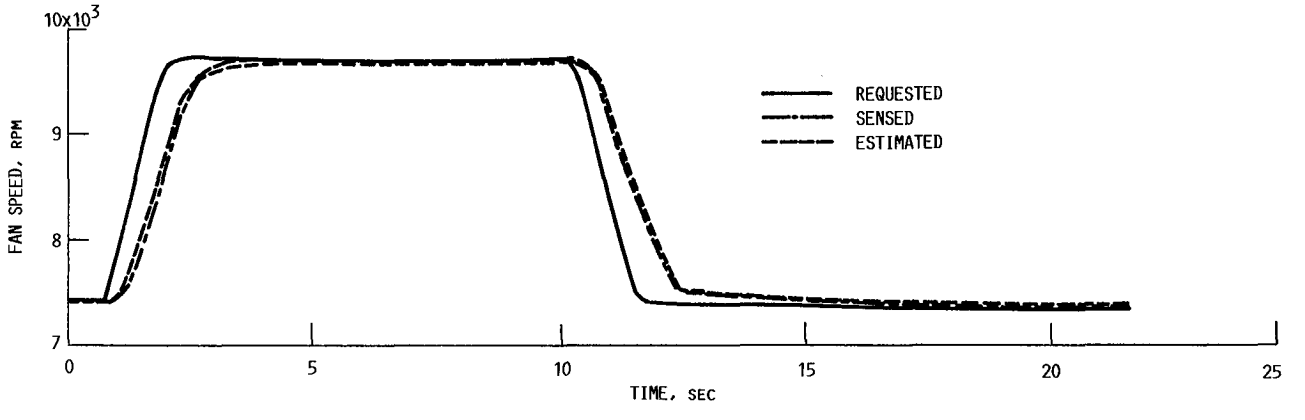


Figure 17.—Multivariable control performance showing 45 000-ft/Mach 0.9 pulse transient response for fan speed N1.

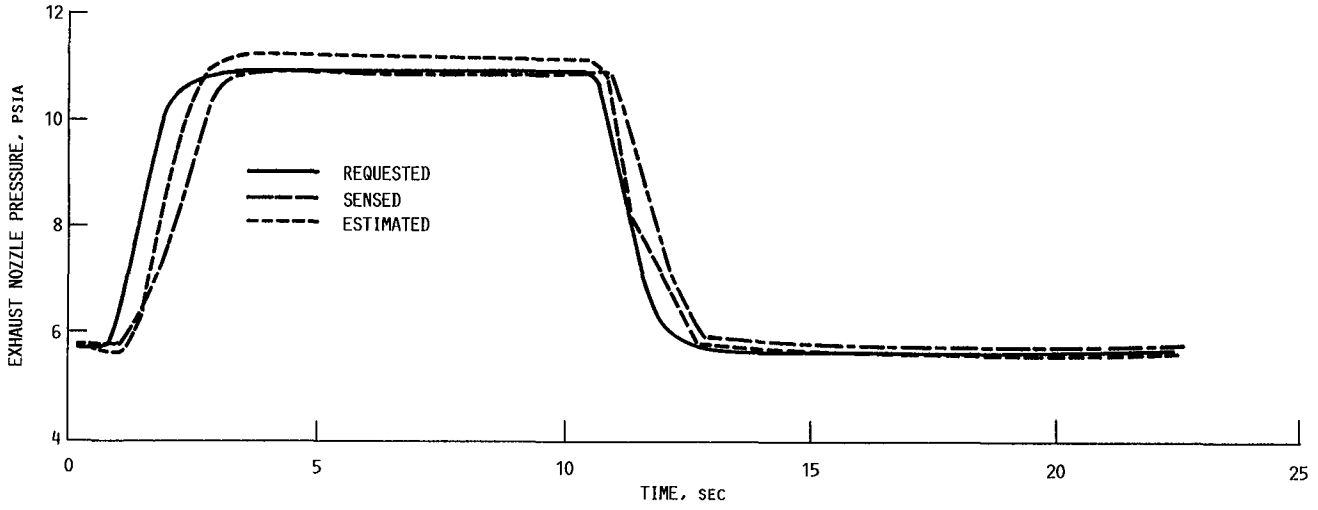


Figure 18.—Multivariable control performance showing 45 000-ft/Mach 0.9 pulse transient response for exhaust nozzle pressure PT6.

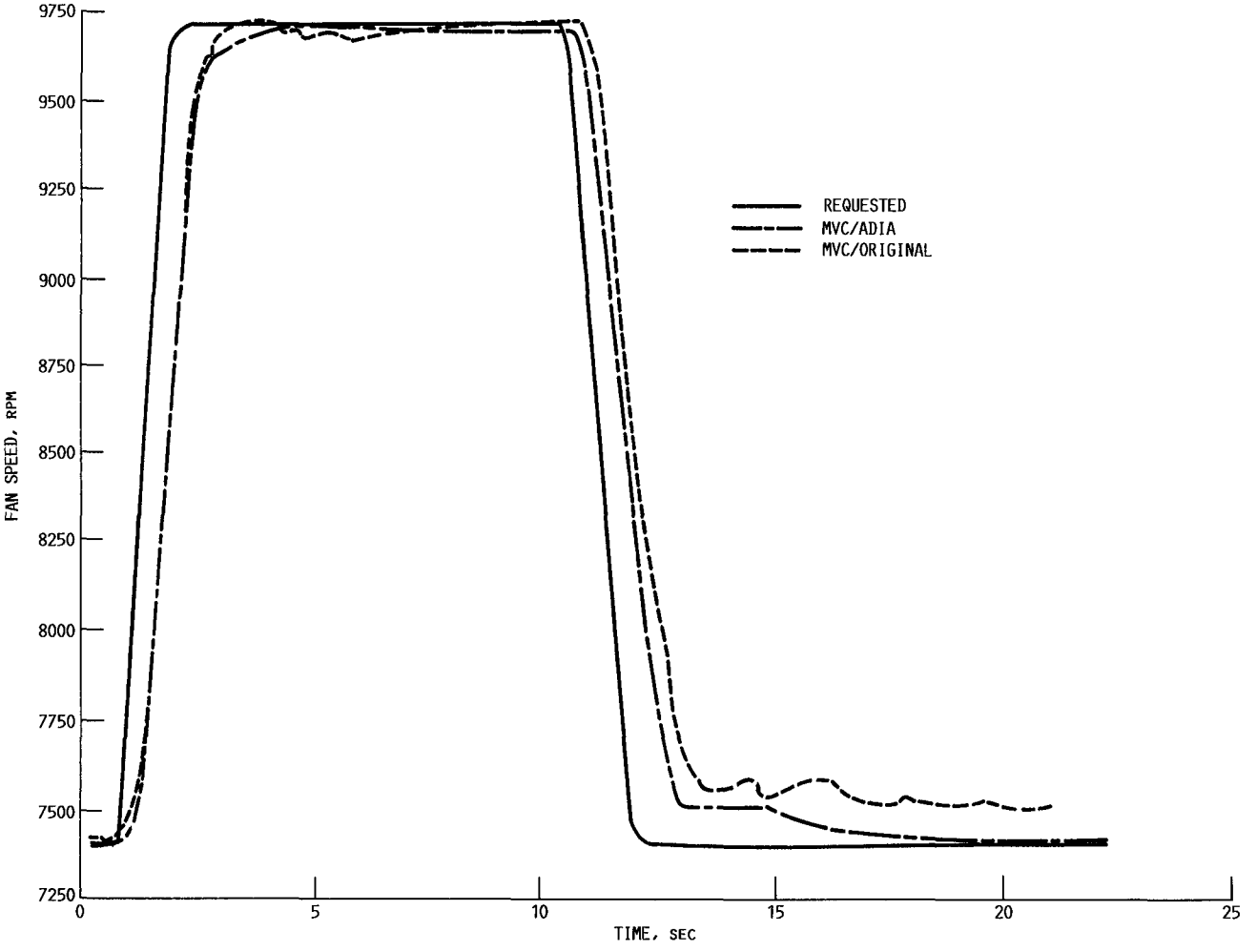


Figure 19.—Multivariable control response to 45 000-ft/Mach 0.9 pulse transient for fan speed N1—comparison for two implementations.

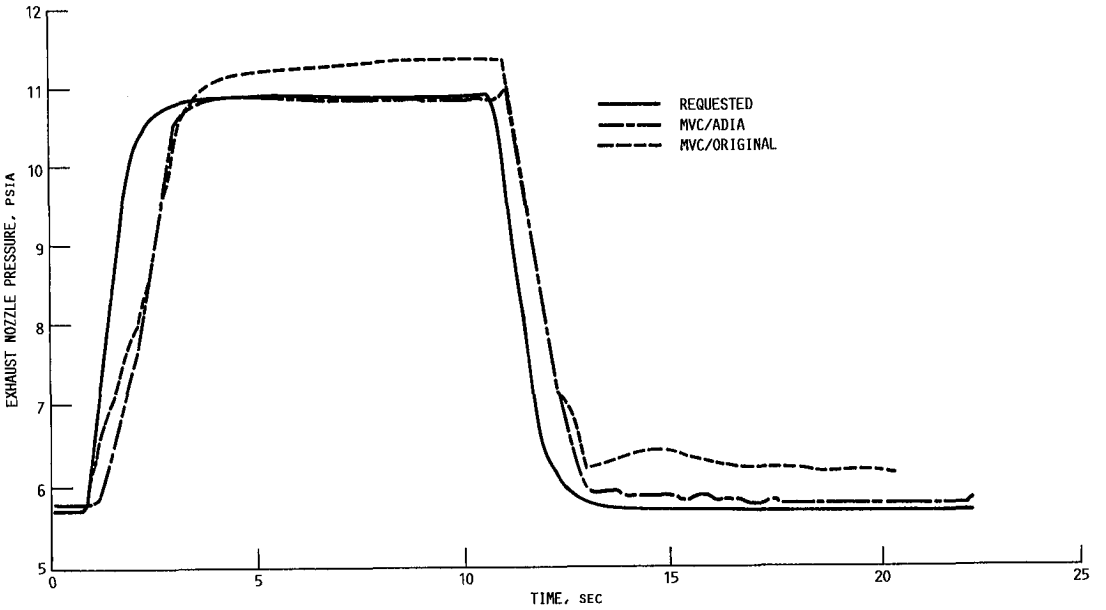
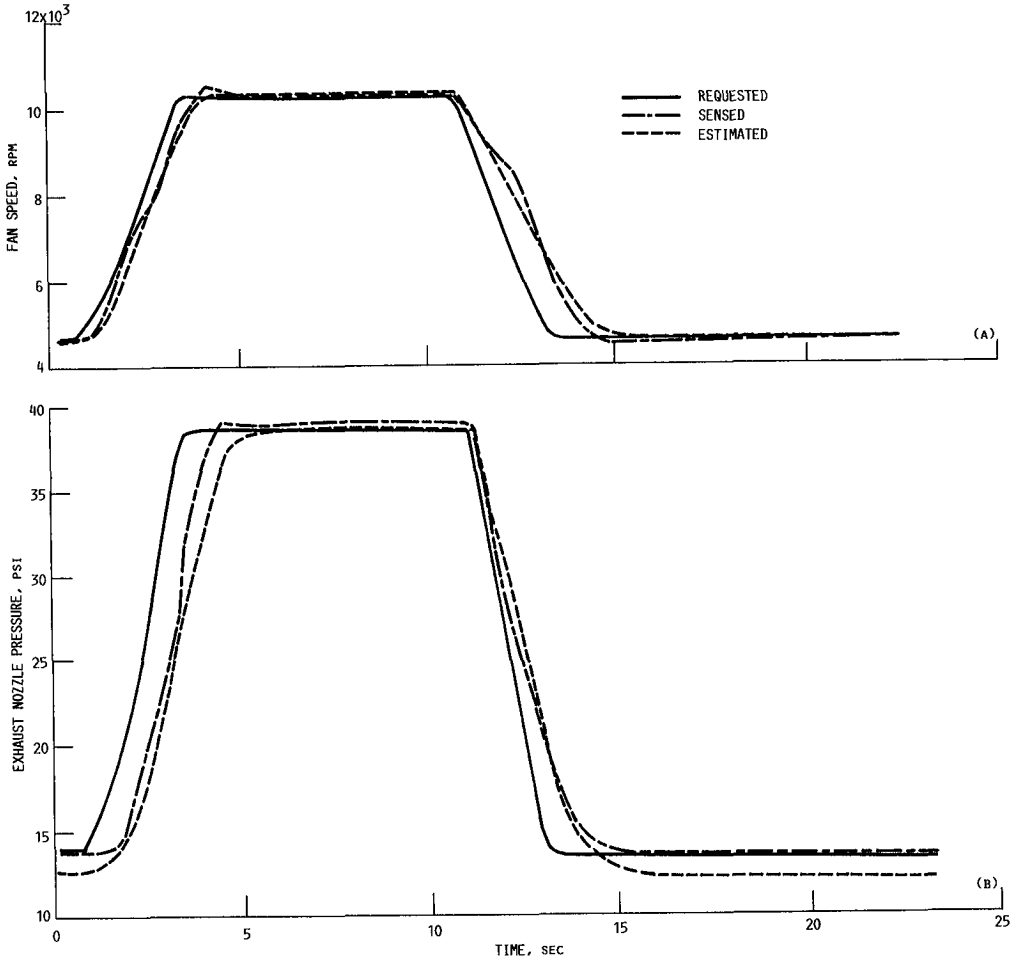
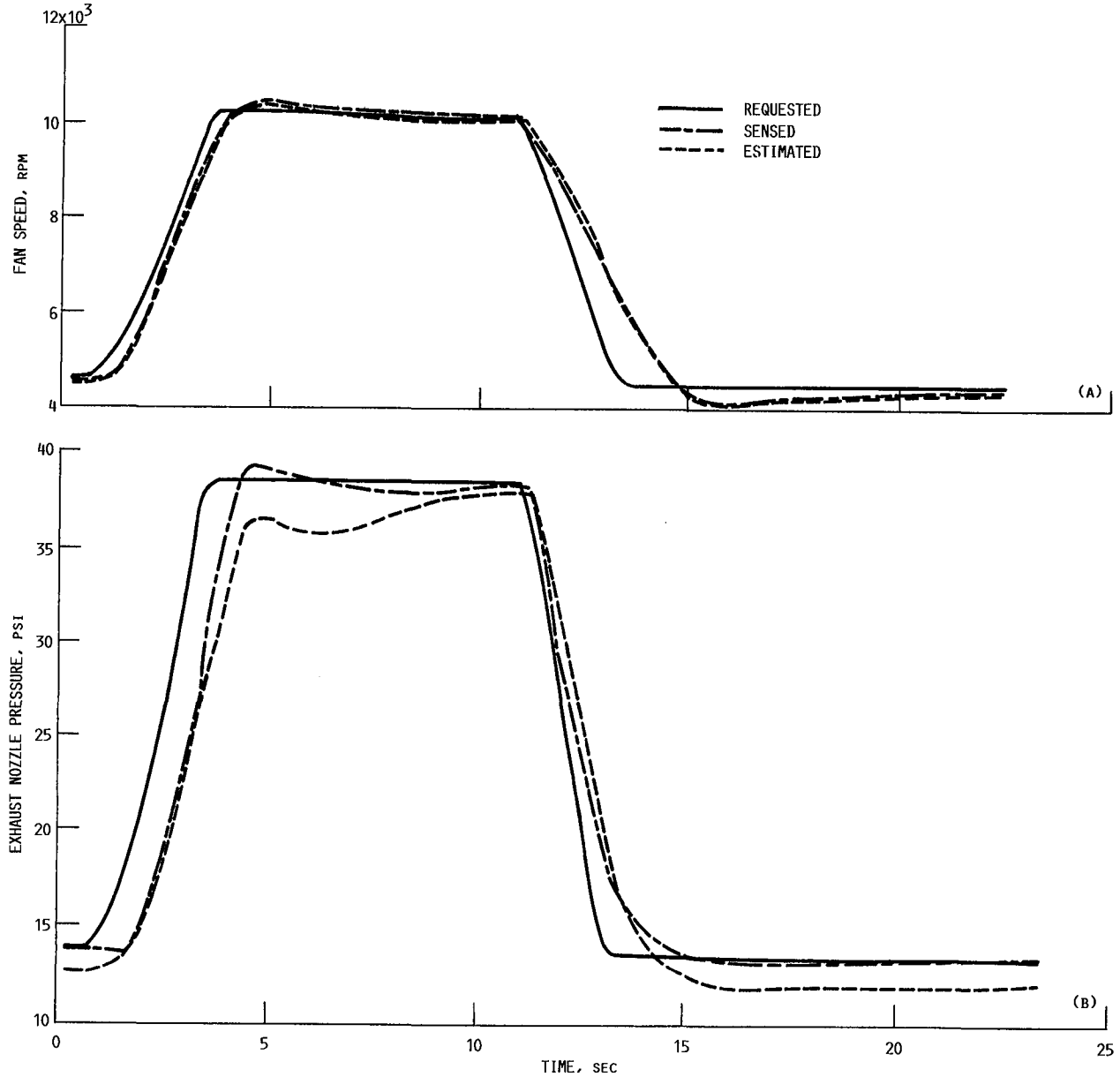


Figure 20.—Multivariable control response to 45 000-ft/Mach 0.9 pulse transient for augmentor pressure PT6—comparison for two implementations.



(a) N1 pulse response.
(b) PT6 pulse response.

Figure 21.—ADIA–MVC performance at 10 000-ft/Mach 0.6 operating condition.



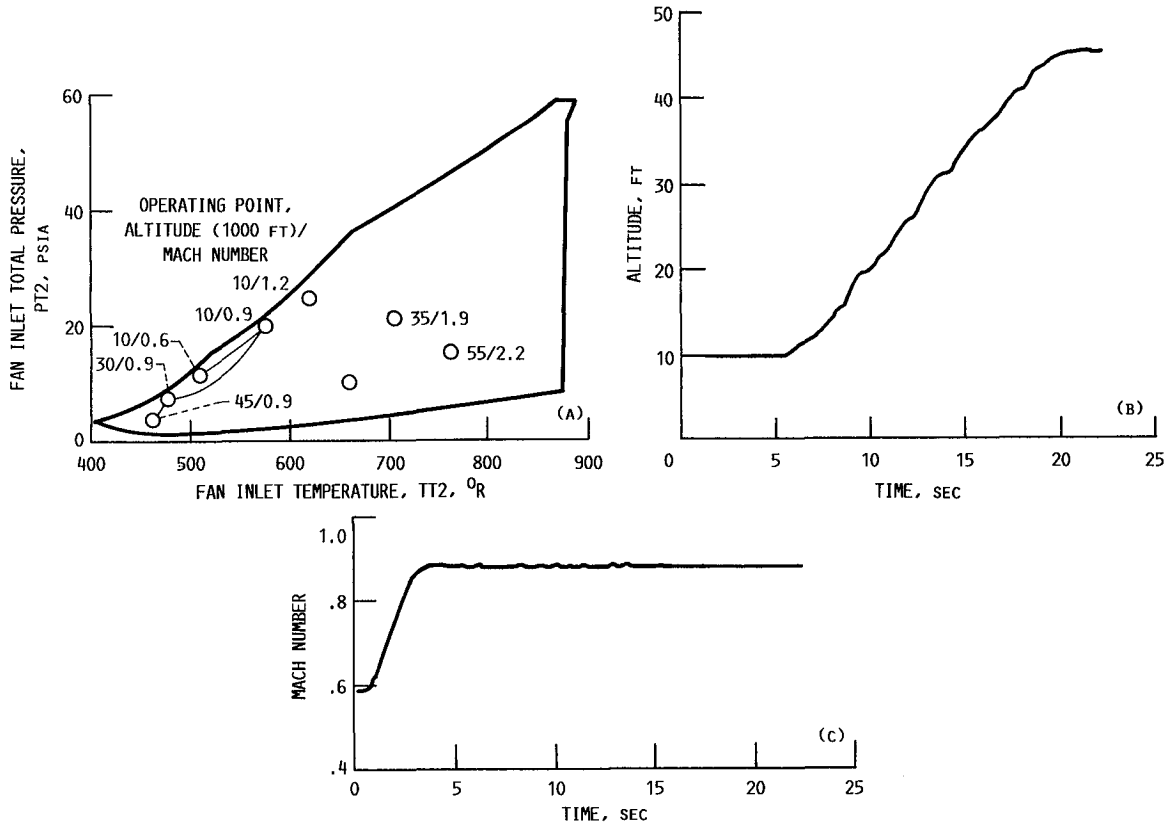
(a) N1 pulse response.
(b) PT6 pulse response.

Figure 22.—ADIA-MVC performance at 10 000-ft/Mach 0.6 with a PT6 sensor failure.

system was obtained. As seen in the typical responses of figure 19 for N1 and in figure 20 for PT6, the comparison was quite good. This was typical for all five of the transient test matrix conditions.

Additionally the control was evaluated to determine if successful engine operation could be obtained without compressor bleed. In these tests the simulation was subjected to the pulse transient with the compressor bleed fixed in the closed position. The transient was simulated at the 10 000-ft/Mach 0.6 operating condition. These results were compared with those for the nominal configuration. The comparison shows no discernible difference between engine control operation with and without compressor bleed.

Control performance was also evaluated given that a single sensor failure had occurred. The purpose of this test was to evaluate control performance after a single sensor failure had been accommodated. In this case five pulse transients at the 10 000-ft/Mach 0.6 operating condition were simulated. In each case a single, but different, output sensor failure was accommodated before the pulse transient was initiated. The normal-mode responses and the failure transient responses are compared for an N1 failure in figure 21 and for a PT6 failure in figure 22. Control performance was good for all five failure-mode cases. Additional information about estimate accuracy during these tests is given in the next section.



(a) Model operating points for PT2/TT2 envelope.
(b) Altitude excursion.
(c) Mach number excursion.

Figure 23.—Altitude and Mach number excursions.

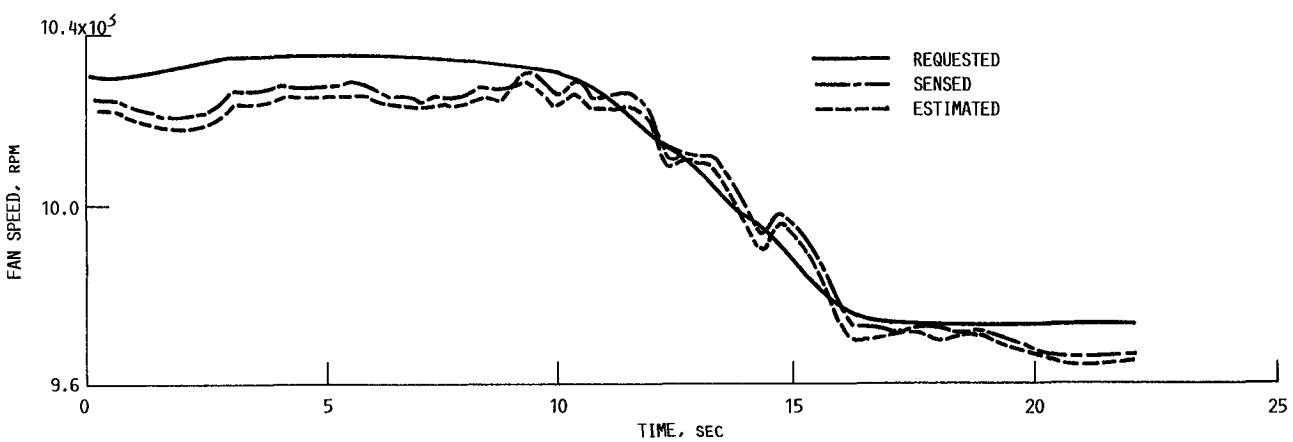


Figure 24.—ADIA-MVC performance—N1 response to altitude and Mach number excursions of figure 23.

Finally control performance was evaluated for an altitude and Mach number excursion. The ADIA-MVC control performed acceptably during this excursion. The excursion went from the 10 000-ft/Mach 0.6/83° operating condition to the 45 000-ft/Mach 0.9/83° operating condition as shown in figure 23. Data showing N1 and PT6 control for this transient are given in figures 24 and 25, respectively. Control performance for this transient was quite good.

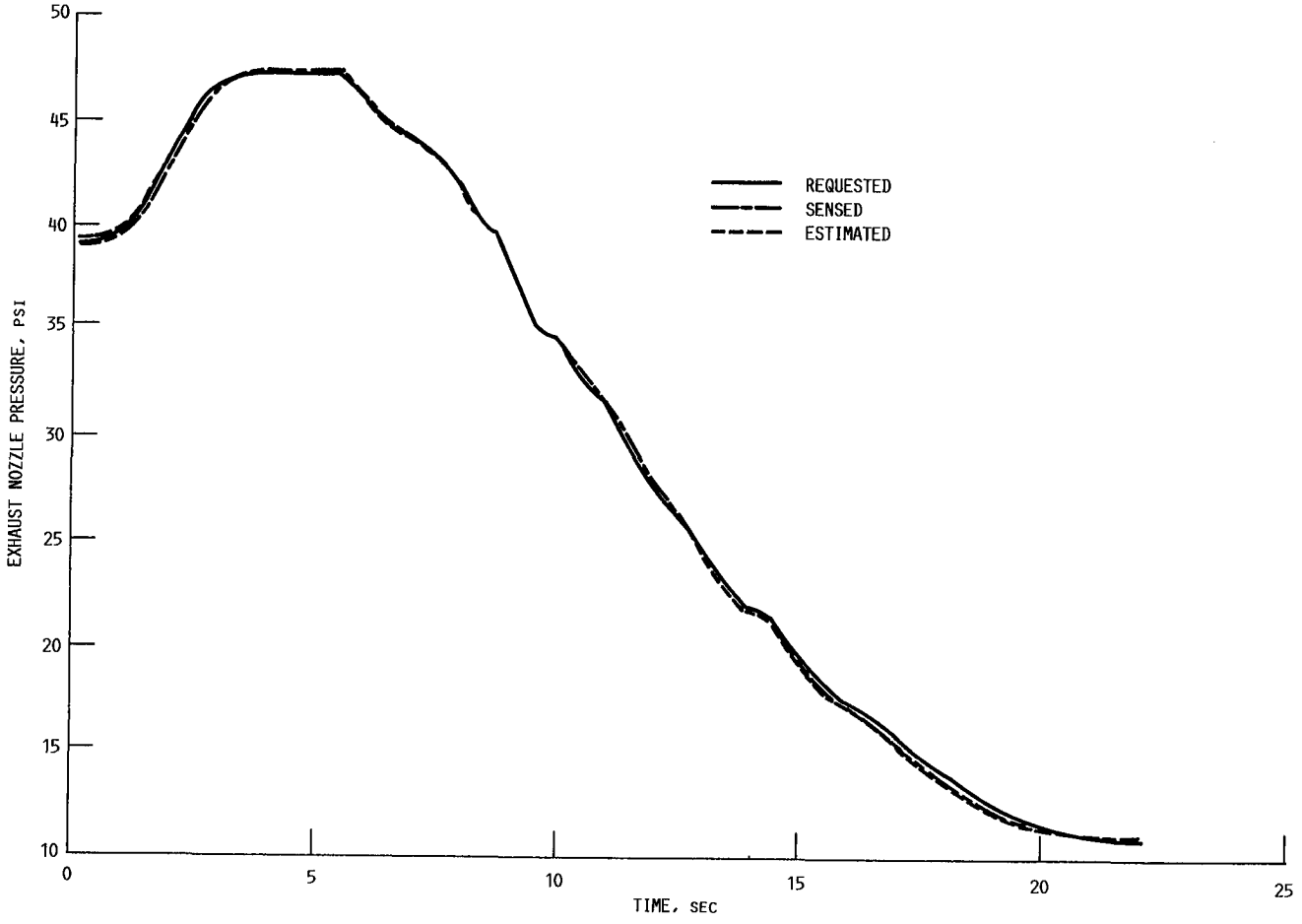


Figure 25.—ADIA–MVC performance—PT6 response to altitude/Mach number excursions of figure 23.

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 by the accommodation 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 conditions. An engine operating condition 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. The accuracy of the estimates is presented in two parts, steady-state accuracy and transient accuracy.

Steady-state accuracy was obtained in a straightforward manner. The simulation was "flown" to the desired operating condition 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. Measured and estimated variables for seven operating conditions are compared in table V by showing the difference (the residual) between sensed and estimated fan speed N1, compressor speed N2, combustor pressure PT4, exhaust nozzle pressure PT6, and fan turbine inlet temperature FTIT as a percentage of the nominal value.

TABLE V.—STEADY-STATE ESTIMATION ACCURACY RESULTS WITH NO SENSOR FAILURES

Operating condition			Engine output				
Altitude, ft	Mach number	Power lever angle, PLA, deg	N1	N2	PT4	PT6	FTIT
			Accuracy, percent of nominal				
10×10^3	0.6	83	0.43	0.11	3.16	0.53	0.11
30	.9	50	.06	.16	.21	1.53	.11
10	.9	83	.42	.28	1.36	.69	.04
45	.9	60	.12	.21	1.87	1.45	.04
10	1.2	83	.17	.11	1.36	.33	.11
55	2.2	↓	.33	.54	5.64	2.48	.05
35	1.9	↓	.03	.32	.92	5.12	.01
50	1.8	↓	.39	.49	3.12	2.48	.04
		Average	0.24	0.28	2.21	1.83	0.06
		Maximum	.43	.53	5.64	5.12	.11

From these comparisons it is clear that the estimates exhibit excellent steady-state accuracy. Maximum error magnitudes occurred for PT4 at the 55 000-ft point and for PT6 at the 35 000-ft point. These error magnitudes can be easily reduced by straightforward adjustment of the base-point schedules used in the algorithm. The average and maximum steady-state accuracy results are summarized in bar graph form in figure 26.

Transient accuracy data were obtained in the following manner: Again the simulation was "flown" to the desired operating condition and allowed to reach steady state. An idle-to-intermediate-power PLA pulse transient was then simulated (see fig. 27) at five different operating conditions. MINDS was used to sample and store data throughout the transient. Example plots of sensed and estimated fan speed and its residual, as well as the likelihood ratio for N1, are presented in figures 28 to 30.

These trajectories give the reader a "feel" for the summarized results of tables VI to IX. In table VI the maximum value of the residuals obtained in response to the reference transient is given for each output at each of the five operating conditions. In table VII the average absolute values of the residuals are given.

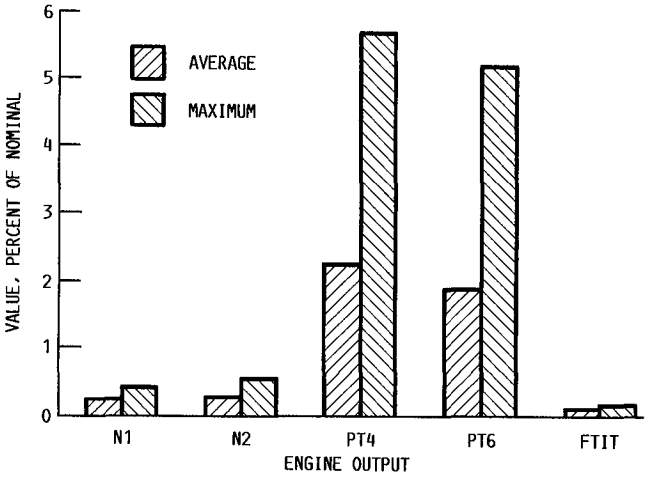


Figure 26.—Steady-state estimation accuracy.

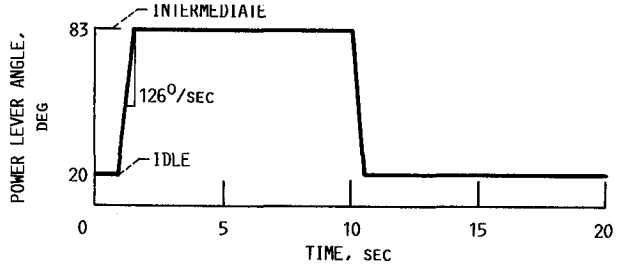


Figure 27.—Idle-to-intermediate power pulse transient used to generate transient results.

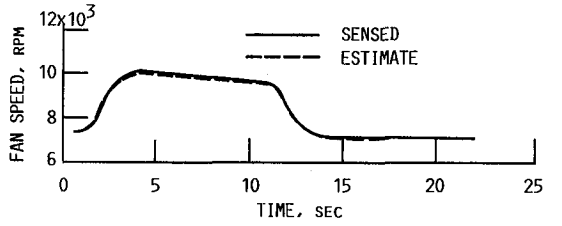


Figure 28.—Example of pulse response to sensed and estimated fan speed for 10 000-ft/Mach 0.6 operating condition.

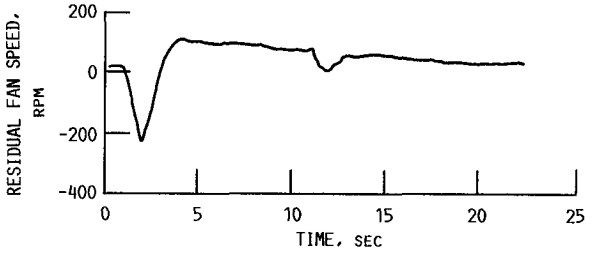


Figure 29.—Example of pulse response to fan speed residual for 10 000-ft/Mach 0.6 operating condition.

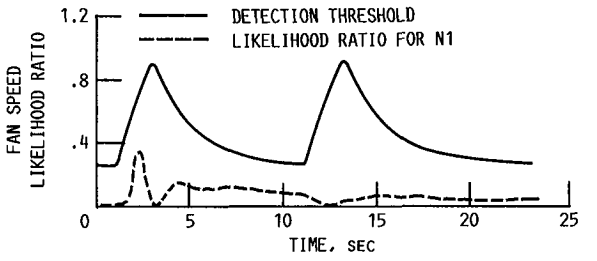


Figure 30.—Example of pulse response to fan speed likelihood ratio for 10 000-ft/Mach 0.6 operating condition.

Since detection performance is determined by the likelihood ratios, estimate accuracy interpreted in terms of these statistics is critical to understanding algorithm performance. The maximum ratio values and the average ratio values are given in tables VIII and IX, respectively, to summarize transient accuracy for the reference trajectories.

TABLE VI.—MAXIMUM RESIDUAL VALUE IN RESPONSE TO PLA PULSE INPUT (NORMAL MODE)

Operating condition		Engine output				
Altitude, ft	Mach number	N1	N2	PT4	PT6	FTIT
		Value, percent of nominal				
10×10^3	0.6	3.57	0.81	6.50	12.55	5.78
30	.9	1.47	.74	4.48	13.08	5.49
10	.9	4.30	1.13	5.22	14.98	5.68
45	.9	2.89	1.86	7.81	19.30	4.44
10	1.2	1.54	1.24	5.02	9.21	3.50
	Average	2.75	1.16	5.81	13.82	4.98
	Maximum	4.30	1.86	7.81	19.30	5.78

TABLE VII.—AVERAGE RESIDUAL ABSOLUTE VALUE IN RESPONSE TO PLA PULSE INPUT (NORMAL MODE)

Operating condition		Engine output				
Altitude, ft	Mach number	N1	N2	PT4	PT6	FTIT
		Value, percent of nominal				
10×10^3	0.6	0.77	0.24	1.67	2.33	1.44
30	.9	.63	.28	.92	2.94	1.64
10	.9	.60	.42	.78	2.98	1.39
45	.9	.49	.36	2.37	5.78	1.21
10	1.2	.21	.23	1.21	.75	.85
	Average	0.54	0.31	1.39	2.96	1.31
	Maximum	.77	.42	2.37	5.78	1.64

TABLE VIII.—MAXIMUM LIKELIHOOD RATIO IN RESPONSE TO PLA PULSE INPUT (NORMAL MODE)

Operating condition		Engine output				
Altitude, ft	Mach number	N1	N2	PT4	PT6	FTIT
		Maximum likelihood ratio				
10×10^3	0.6	0.875	0.342	0.286	0.598	0.080
30	.9	.241	.146	.076	.305	.042
10	.9	1.025	.236	.162	1.340	.059
45	.9	.400	.238	.009	.059	.042
10	1.2	.126	.253	.355	.401	.041
	Average	0.533	0.243	0.177	0.540	0.053
	Maximum	1.025	.342	.355	1.340	.080

TABLE IX.—AVERAGE LIKELIHOOD RATIO IN RESPONSE TO PLA PULSE INPUT (NORMAL MODE)

Operating condition		Engine output				
Altitude, ft	Mach number	N1	N2	PT4	PT6	FTIT
		Average likelihood ratio				
10×10^3	0.6	0.086	0.057	0.044	0.059	0.010
30	.9	.060	.037	.006	.045	.007
10	.9	.088	.047	.038	.123	.010
45	.9	.044	.034	.003	.007	.007
10	1.2	.015	.107	.124	.020	.006
	Average	0.059	0.056	0.023	0.051	0.008
	Maximum	.088	.107	.044	.123	.010

Plots of the likelihood ratios became the standard tool used for evaluation and performance prediction. Transient accuracy was considered to be quite good overall 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 earlier.

Estimator accuracy results were also obtained for the single-failure pulse tests described in the MVC evaluation. These results are given in table X for maximum residual values and in table XI for average residual values. These tests show very little degradation in estimator accuracy performance even when it was operating with only four sensors. Thus ADIA performance will not degrade significantly after a single sensor failure.

Estimator accuracy was also studied during the altitude/Mach number excursion of figure 23. As an example sensed and estimated N1 and PT6 are compared in figures 24 and 25, respectively. In each case the accuracy was excellent.

TABLE X.—MAXIMUM RESIDUAL ERRORS FOR PULSE TRANSIENTS AT 10 000 FT/MACH 0.6 WITH SINGLE SENSOR FAILURE

Output	Failure mode					
	None	N1	N2	PT4	PT6	FTIT
	Maximum residual error					
N1	3.570	0	2.931	2.592	2.481	2.877
N2	.810	.979	0	.823	.784	.791
PT4	6.500	7.513	5.211	0	6.454	6.550
PT6	12.550	10.740	12.220	11.480	0	11.880
FTIT	5.780	6.383	5.704	5.650	5.731	0

TABLE XI.—AVERAGE RESIDUAL ERRORS FOR PULSE TRANSIENTS AT 10 000 FT/MACH 0.6 WITH SINGLE SENSOR FAILURE

Output	Failure mode					
	None	N1	N2	PT4	PT6	FTIT
	Average residual error					
N1	0.770	0	0.755	0.472	0.571	0.483
N2	.240	.249	0	.250	.265	.270
PT4	1.670	1.479	.693	0	1.067	1.260
PT6	2.330	2.211	2.486	2.432	0	2.431
FTIT	1.440	1.305	1.417	1.406	1.455	0

Detection, isolation, and accommodation performance.— Two types of sensor failures 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. The ADIA algorithm exhibited excellent hard-failure detection performance at this condition. There were no false alarms or missed detections of any hard failures at the operating condition studied. Hard failures were simulated in each of the engine sensor outputs. The failure was successfully detected and accommodated in each case. In addition, no false alarms in the hard-failure detection logic were encountered during the subsequent soft-failure evaluation.

Soft sensor failures, although small in magnitude, if undetected, may result in degraded or unsafe engine operation. Soft failures are more difficult to detect. Therefore the evaluation concentrated on soft-failure detection and isolation performance. Four soft-failure modes were considered: bias, drift, noise, and scale factor. Algorithm performance for the bias and drift failure modes was studied extensively. Performance for the noise and scale-factor modes was studied at a limited number of conditions. 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 simulate a sensor failure of the appropriate size and at the desired time. The results obtained are summarized for the minimum detectable level of bias in table XII.

In table XII the minimum detectable biases at 11 different operating conditions for each engine output are given. The detection times for the minimum detectable biases were essentially instantaneous. The results are presented as a percentage of full scale, a percentage of nominal, and in engineering units. Full-scale values are constant, but nominal values can vary throughout the operating range. Note that the size of the failures detected (in units or equivalently as a percentage of full scale) was essentially constant over the operating range.

TABLE XII—MINIMUM BIAS FAILURE MAGNITUDES

(a) Units

Operating condition			Engine output				
Altitude, ft	Mach number	Power lever angle, PLA, deg	N1, rpm	N2, rpm	PT4, psi	PT6, psi	FTIT, °F
			Failure magnitude				
10×10^3	0.6	50	300	300	12.50	3.00	150
	.6	83	350	350	12.50	3.00	↓
30	.9	50	300	↓	13.50	2.75	↓
	↓	83	325	↓	13.50	3.00	↓
10	↓	50	300	↓	13.50	2.75	↓
	↓	83	300	↓	11.00	3.00	↓
45	↓	70	200	400	18.00	3.00	250
10	1.2	70	200	400	20.00	3.50	150
50	1.8	83	300	350	12.50	3.00	↓
35	1.9	83	300	300	19.00	3.00	↓
	2.2	83	250	500	25.00	3.00	↓

(b) Nominal

Operating condition			Engine output				
Altitude, ft	Mach number	Power lever angle, PLA, deg	N1	N2	PT4	PT6	FTIT
			Failure magnitude, percent of nominal				
10×10^3	0.6	50	3.47	2.60	6.39	12.02	12.00
	.6	83	3.41	2.67	3.85	7.74	8.75
30	.9	50	3.43	3.10	10.92	18.10	12.36
	↓	83	3.25	2.73	6.99	13.24	9.41
10	↓	50	3.54	2.60	6.17	9.74	12.17
	↓	83	2.91	2.66	2.86	6.40	8.72
45	↓	70	2.12	3.34	21.33	29.96	18.18
10	1.2	70	2.11	3.21	5.71	7.78	10.00
50	1.8	83	2.94	2.71	8.33	17.44	9.06
35	1.9	83	2.95	2.31	5.94	7.69	8.94
55	2.2	83	2.57	3.76	16.34	15.54	8.80

(c) Full-scale bias

Operating condition			Engine output				
Altitude, ft	Mach number	Power lever angle, PLA, deg	N1	N2	PT4	PT6	FTIT
			Failure magnitude, percent of full scale				
10×10^3	0.6	50	2.92	2.29	3.85	7.74	8.75
	.6	83	3.41	2.67	3.85	7.74	↓
30	.9	50	2.92	2.67	4.15	7.10	↓
	↓	83	3.17	2.67	4.15	7.74	↓
10	↓	50	2.92	2.29	4.15	7.10	↓
	↓	83	2.92	2.67	3.38	7.74	↓
45	↓	70	1.95	3.05	5.54	7.74	4.59
10	1.2	70	1.95	3.05	6.15	9.04	8.75
50	1.8	83	2.92	2.67	3.85	7.74	↓
35	1.9	83	2.92	2.29	5.85	7.74	↓
55	2.2	83	2.44	3.81	7.69	7.74	↓

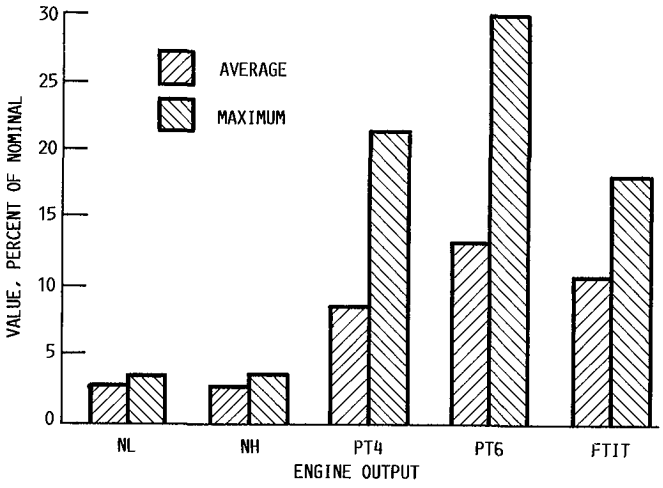


Figure 31.—Minimum detectable bias failure.

Note also that the highest detectable bias (29.96 percent) as a percentage of nominal occurred for PT6 at 45 000-ft/Mach 0.9/70° because of the low nominal value of PT6 at this condition. However, this failure as a percentage of scale (7.74 percent) was about average. The minimum detectable bias magnitudes were small overall and represented excellent performance. This performance is summarized in bar graph form in figure 31.

The minimum detectable drift rates (table XIII) were determined by adjusting the drift magnitude such that a failure was detected approximately 5 sec after its inception. As in the bias case the highest minimum detectable drift rate as a percentage of nominal occurred at the 45 000-ft/Mach 0.9/70° operating condition. However, as before, this value as a percentage of nominal was well below the maximum value as a percentage of full scale. The percentage-of-full-scale values are all similar in magnitude. In general these results, which are summarized in the bar graph of figure 32, were excellent.

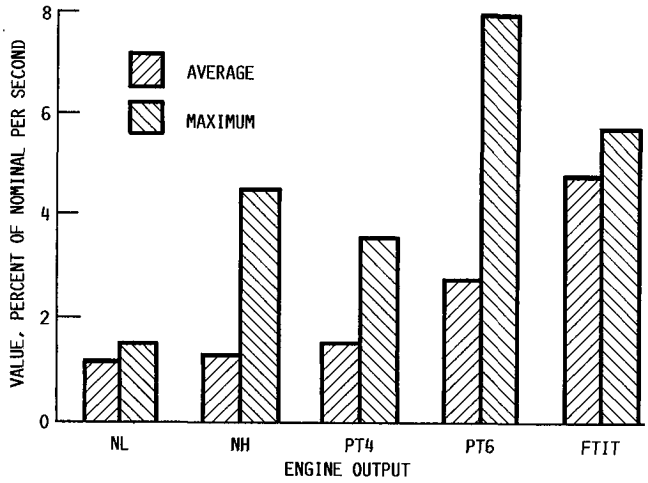


Figure 32.—Minimum detectable ramp failure.

TABLE XIII.—MINIMUM DRIFT FAILURE MAGNITUDES

(a) Units

Operating condition			Engine output				
Altitude, ft	Mach number	Power lever angle, PLA, deg	N1, rpm/sec	N2, rpm/sec	PT4, psi/sec	PT6, psi/sec	FTIT, °F/sec
			Failure magnitude				
10×10^3	0.6	50	100	100	2.50	0.80	70
	.6	83	125	125	1.25	.60	
30	.9	50	130	100	2.50		
		83	150	125	3.00		
10		50	100	100	2.50		
		83	125	125	2.00		
45		70	100	150	3.00	.80	75
10	1.2	70	25	150	3.50	1.20	70
50	1.8	83	150	150	1.75	.20	70
35	1.9	83	125	50	3.00	.20	70
55	2.2	83	75	215	4.50	.60	75

(b) Nominal drift

Operating condition			Engine output				
Altitude, ft	Mach number	Power lever angle, PLA, deg	N1	N2	PT4	PT6	FTIT
			Failure magnitude, percent of nominal				
10×10^3	0.6	50	1.16	0.87	1.28	3.20	5.60
	.6	83	1.22	.95	.38	1.55	4.08
30	.9	50	1.49	.89	2.02	3.95	5.77
		83	1.50	.97	1.55	2.65	4.39
10		50	1.18	.87	1.14	2.13	5.68
		83	1.21	.95	.52	1.28	4.07
45		70	1.06	1.25	3.55	7.99	5.45
10	1.2	70	.26	1.20	1.00	2.67	4.67
50	1.8	83	0	.01	.01	.03	.05
35	1.9	83	1.23	.38	.94	.51	4.17
55	2.2	83	.77	1.61	2.94	3.11	4.40

(c) Full-scale drift

Operating condition			Engine output				
Altitude, ft	Mach number	Power lever angle, PLA, deg	N1	N2	PT4	PT6	FTIT
			Failure magnitude, percent of full scale				
10×10^3	0.6	50	976	0.76	0.77	2.70	4.08
	.6	83	1.22	.95	.38	1.55	
30	.9	50	1.27	.76	.77		
		83	1.46	.95	.92		
10		50	.97	.76	.77		
		83	1.22	.95	.62		
45		70	.97	1.14	.92	2.07	4.38
10	1.2	70	.24	1.14	1.08	3.10	4.08
50	1.8	83	0	.01	.01	.03	.04
35	1.9	83	1.22	.38	.94	.51	4.17
55	2.2	83	.73	1.64	1.38	1.55	4.38

TABLE XIV.—STEADY-STATE RESULTS OF SLOW DRIFT FAILURE TRANSIENTS FOR ORIGINAL ADIA ALGORITHM

Operating condition			Failure parameter	ADIA algorithm				
Altitude, ft	Mach number	Power lever angle, PLA, deg		Parameter bias before DIA	Change in thrust before DIA, percent	Time for DIA, sec	Comments	Performance[a]
0	0	24	P6	7.4 psi (42%)	−4.5	0.490		A
	0	40	N1	1333 rpm (12.1%)	−44.5	1.994		U
	0	83	PT4	46.5 psi (12.7%)	−.1	3.080	Filter noisy during ADIA	A
	1.2	83	FTIT	90 F (5.2%)	−2.2	2.53		↓
10×10^3	.75	50	PT4	40.5 psi (19.6%)	−.2	2.664		
	.75	83	PT6	9 psi (21.8%)	−1.5	.572		↓
20	.3	40	N2	Undetected	----	Undetected	Unstable diverging	U
	.3	83	N2(−)	Undetected	----	Undetected	Unstable	U
	.3	↓	N2	1415 rpm (11.4%)	−4.5	3.518		A
25	1.0		PT4	46.5 psi (18.1%)	−.2	3.066		A
	2.2	↓	PT4	False alarm	----	----	PT4 and PT6 false alarms prior to failure	U
40	.6	40	N2	Miss	−48	----	2000-rpm drift miss	U
	.6	83	PT6	6.75 psi (63.7%)	−.5	.448	System oscillatory after failure induced	A
45	2.2	↓	P4	−56.4 psi (−24.5%)	−.16	3.750		A
60	1.2		N2	2000 rpm (15.8%)	−19.4	5.016	Drift caused system to go unstable	U
65	2.5	↓	P6	−3.75 psi (−27.4%)	+24.7	.400	FTIT false alarm	U

[a]A = acceptable; U = unacceptable.

To place these results in perspective, the soft-failure detection performance of this improved version of the algorithm was compared with the original algorithm (ref. 1). In this comparison drift failures were injected at 17 "edge of the envelope" points. Performance data for the original algorithm (table XIV were obtained from a nonlinear, digital simulation of the engine. The performance of the improved algorithm is presented for comparison in table XV.

Some of the operating conditions of table XIV are only approximated in table XV. The hybrid computer could not successfully attain these conditions because of scaling limitations. In every case but one, the improved version of the algorithm had a smaller parameter bias before detection than the original version. In all failure cases the improved algorithm allowed continued engine operation; in some cases the original algorithm would have required an engine shutdown. The drift failure rates used in this comparison are given in reference 1.

Estimation accuracy and false alarm performance were also evaluated for the altitude/Mach number excursion defined by figure 23. Likelihood ratio and detection threshold responses are given in figure 33 for the PT4 and PT6 engine outputs. The likelihood ratios for N1, N2, and FTIT were all smaller than those for PT4 and PT6. Here the accuracy was quite good and there were no false alarms.

TABLE XV.—DRIFT FAILURE RESULTS

Operating condition			Failure parameter	Parameter bias	Change in thrust before DIA, percent	Time for DIA, sec
Altitude, ft	Mach number	Power lever angle, PLA, deg				
0	0	24	P6	3.7 psi	0	0.5
	0	40	N1	263.0 rpm	1	.3
	0	83	PT4	15.2 psi	0	.9
	1.2	83	FTIT	124.0 °F	−4	2.5
10×10^3	.75	50	PT4	16.4 psi	0	.9
	.74	83	PT6	2.7 psi	−8	3.7
20	.3	40	N2	248.0 rpm	−1	.3
	.3	83	N2(−)	−234.0 rpm	1	.5
	.3	↓	N2	265.0 rpm	−1	.3
25	1.0		PT4	16.8 psi	0	1.0
	2.0	↓	PT4	17.8 psi	−1	1.2
40	.6	40	N2	216.0 rpm	−5	.5
	.6	83	PT6	2.6 psi	−11	1.6
45	2.1	↓	PT4	15.7 psi	0	.5
60	1.2		N2	274.0 rpm	−15	.8
65	2.3		PT6	3.7 psi	−5	2.3
45	2.1	↓	PT4(−)	−15.0 psi	0	1.5

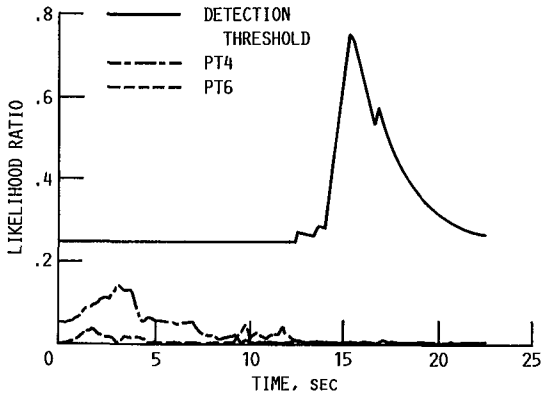
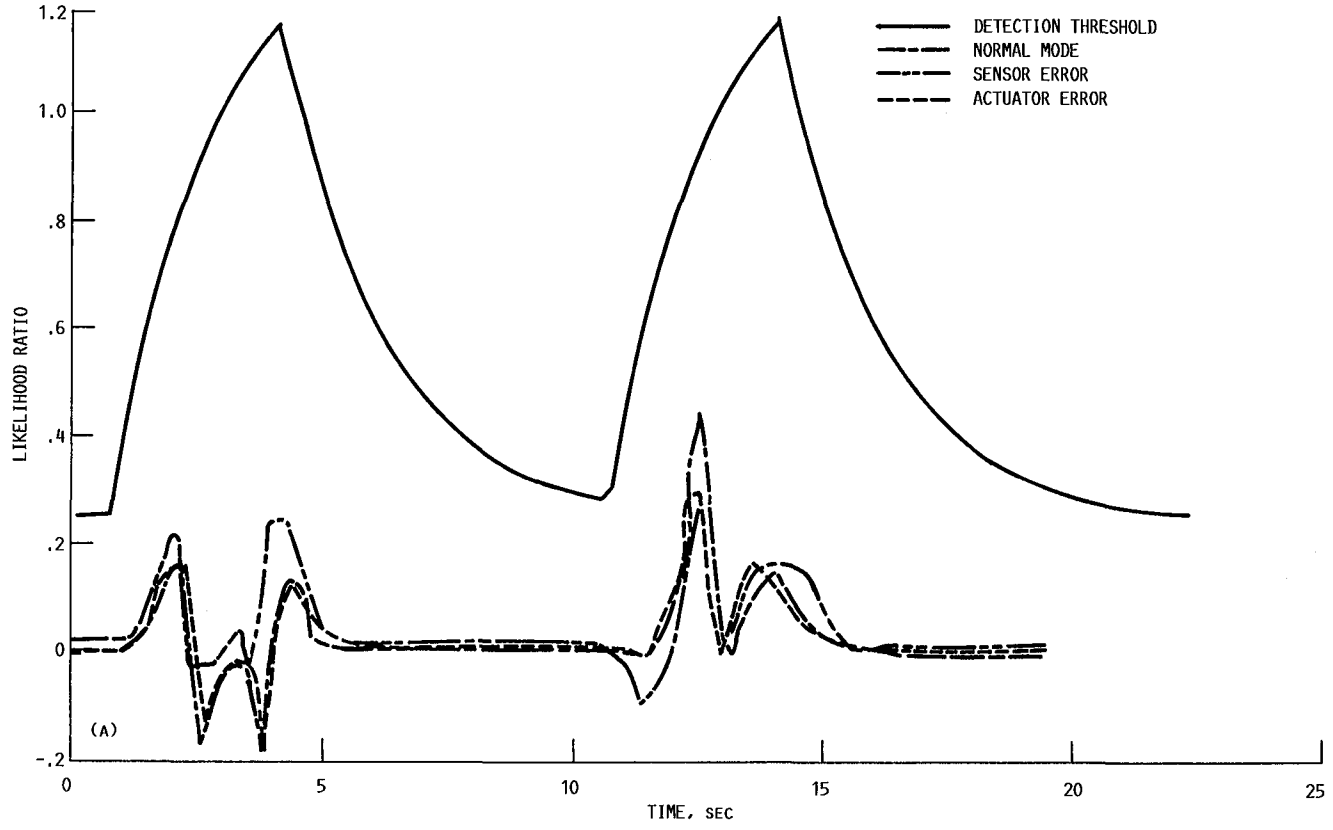


Figure 33.—Likelihood ratio response during altitude/Mach number excursion.

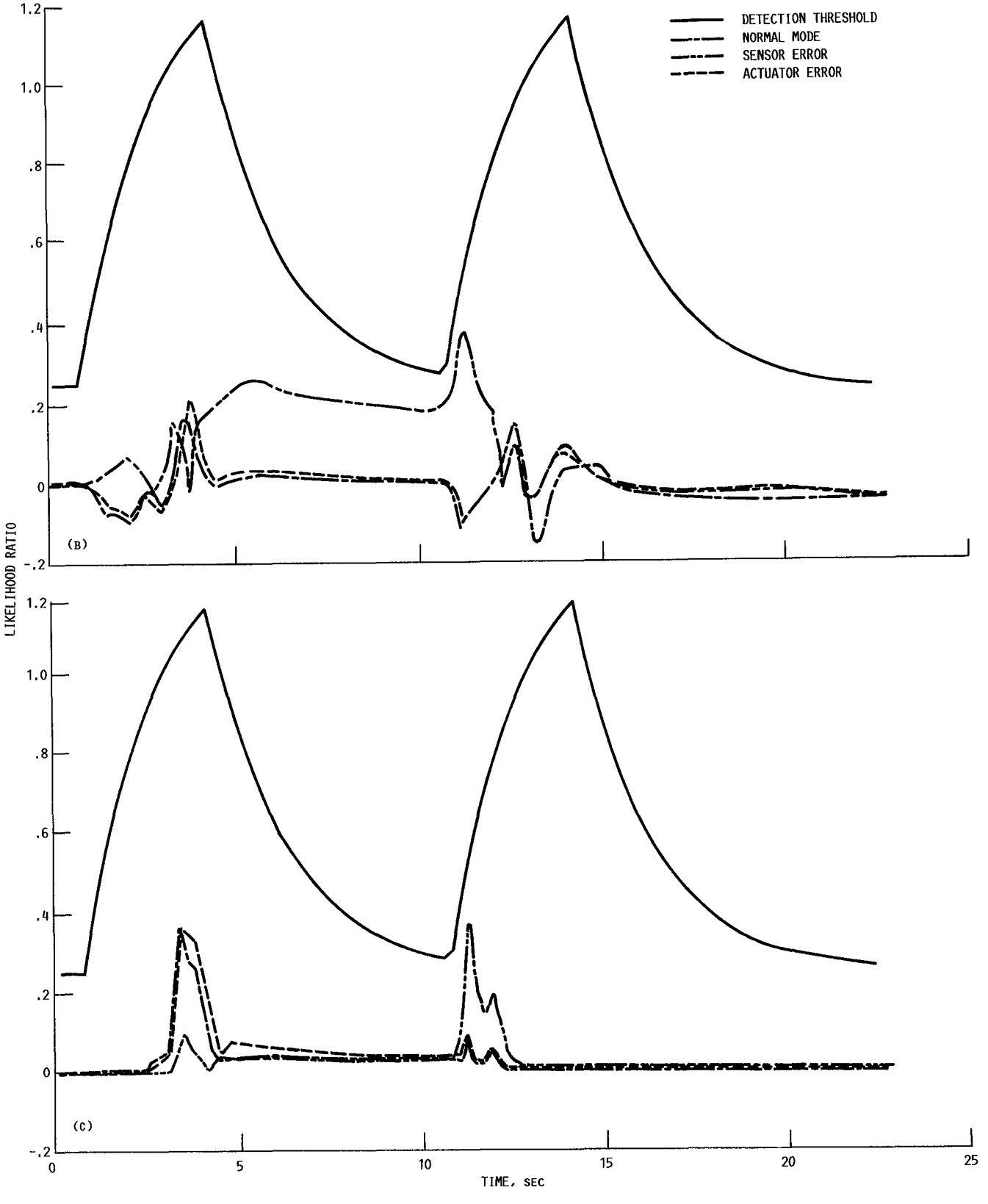
Actuator Modeling Error Evaluation

The effects of fuel flow actuator and fuel flow feedback sensor modeling errors were also evaluated. Since the algorithm estimates depend on fuel flow information and since fuel flow is the primary control variable, fuel flow actuation or measurement errors could significantly degrade detection performance. Likelihood ratio results for the PLA pulse test at the 10 000-ft/Mach 0.6 operating condition are shown in figure 34 for normal operation, for operation with a 10 percent change in the fuel flow actuator gain, and for operation with a 10 percent change in the fuel flow feedback measurement. In each case no false alarms were encountered. In general, detection performance was not significantly degraded by the modeling errors. However, some effect was seen for a feedback sensor error on the N2 and PT4 likelihood ratios. For PT4 the difference occurred only during an engine acceleration and would have only a limited effect on detection performance. For N2, however, a steady-state error occurred that in the worst case would result in approximately a 50 percent increase in minimum detectable failure magnitudes.



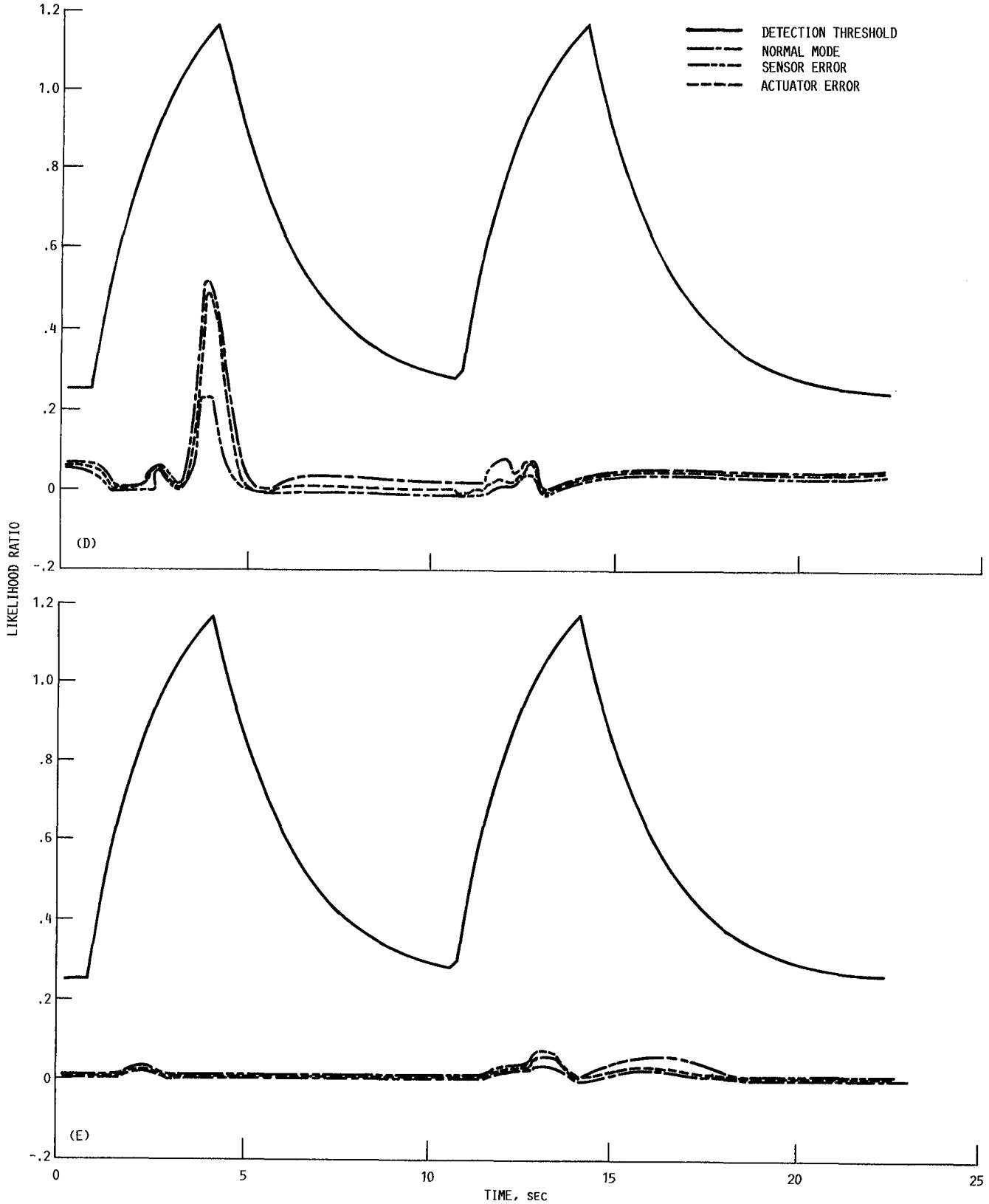
(a) N1 likelihood ratio.

Figure 34—Fuel flow modeling error effects for various likelihood ratios.



(b) N2 likelihood ratio.
(c) PT4 likelihood ratio.

Figure 34.—Continued.



(d) PT6 likelihood ratio.
(e) FTIT likelihood ratio.

Figure 34.—Concluded.

Conclusions

As a result of this real-time evaluation study several conclusions have been reached.

1. The advanced detection, isolation, and accommodation (ADIA) failure detection algorithm works and works quite well. Sensor failure detection and accommodation were demonstrated over a broad range of operating conditions and power conditions. The minimum detectable failure magnitudes represent excellent algorithm performance.

2. The algorithm is implementable in a realistic computer environment and in an update interval consistent with real-time 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 multiplier of computational resources.

3. The ADIA algorithm and MVC control microprocessor-based implementations are ready for demonstration. The ADIA algorithm will be demonstrated on a full-scale F100 engine in the Lewis Research Center's altitude test facility.

Lewis Research Center
National Aeronautics and Space Administration
Cleveland, Ohio, April 14, 1987

References

1. 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.
2. Beattie, E.C., et al.: Sensor Failure Detection System for F100 Turbofan Engine. (PWA-5736-17, Pratt & Whitney Aircraft; NASA Contract NAS3-22481) NASA CR-165515, 1981.
3. Beattie, E.C., et al.: Sensor Failure Detection for Jet Engines. (PWA-5891-18, Pratt & Whitney Aircraft; NASA Contract NAS3-23282) NASA CR-168190, 1983.
4. 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.)
5. Merrill, W.C.; and DeLaat, 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.
6. Lehtinen, B., et al.: F100 Multivariable Control Synthesis Program—Results of Engine Altitude Tests. NASA TM S-83367, 1983.
7. Szuch, J.R.; Seldner, K.; and Cwynar, D.S.: Development and Verification of a Real-Time, Hybrid Computer Simulation of the F100-PW-100(3) Turbofan Engine. NASA TP-1034, 1977.
8. DeLaat, J.C.; and Soeder, J.F.: Design of a Microprocessor-Based Control, Interface, and Monitoring (CIM) Unit for Turbine Engine Controls Research. NASA TM-83433, 1983.
9. Mackin, M.A.; and Soeder, J.F.: Floating-Point Function Generation Routines for 16-Bit Microcomputers. NASA TM-83783, 1984.
10. DeLaat, J.C.: A Real-Time FORTRAN Implementation of a Sensor Failure Detection, Isolation and Accommodation Algorithm. Proceedings of the 1984 American Control Conference, vol. 1, IEEE, 1984, pp. 572-573.
11. Soeder, J.F.: MINDS: A Microcomputer Interactive Data System for 8086-Based Controllers. NASA TP-2378, 1985.
12. Sheskin, T.J.; and Soeder, J.F.: PMID: A Software Package for Plotting Interactive Data. Computers and Industrial Engineering, vol. 9, suppl. 1, 1985, pp. 415-419.
13. Melcher, K.J., et al.: A Sensor Failure Simulator for Control System Reliability Studies. NASA TM-87271, 1986.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TP-2740		

4. Title and Subtitle	5. Report Date
Advanced Detection, Isolation, and Accommodation of Sensor Failures—Real-Time Evaluation	July 1987
	6. Performing Organization Code: 505-62-01

7. Author(s)	8. Performing Organization Report No.
Walter C. Merrill, John C. DeLaat, and William M. Bruton	E-3479
	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 Paper

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

15. Supplementary Notes

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 by using analytical redundancy to detect sensor failures. In this report the results of a real-time hybrid computer evaluation of the ADIA algorithm are presented. Minimum detectable levels of sensor failures for an F100 engine control system are determined. Also included are details about the microprocessor implementation of the algorithm as well as a description of the algorithm itself.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Sensors; Failure; Detection; Isolation; Accommodation; Real-time operation; Feedback control; Control; Simulation; Fault tolerance	Unclassified – unlimited STAR Category 08

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No of pages	22. Price*
Unclassified	Unclassified	29	A03

NASA FORM 1626 OCT 86

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