Turbofan Engine Demonstration of Sensor Failure Detection

Walter C. Merrill,* John C. DeLaat,† and Mahmood Abdelwahab‡
NASA Lewis Research Center, Cleveland, Ohio 44135

In the paper, the results of a full-scale engine demonstration of a sensor failure detection algorithm are presented. The algorithm detects, isolates, and accommodates sensor failures using analytical redundancy. The experimental hardware, including the F100 engine, is described. Demonstration results were obtained over a large portion of a typical flight envelope for the F100 engine. They include both subsonic and supersonic conditions at both medium and full, nonafter burning, power. Estimated accuracy, minimum detectable levels of sensor failures, and failure accommodation performance for an F100 turbofan engine control system are discussed.

Introduction

THE objective of the advanced detection, isolation, and accommodation (ADIA) program was to improve demonstrated reliability of digital electronic control systems for turbine engines by detecting, isolating, and accommodating sensor failures using analytical redundancy methods. This paper discusses the results of an engine demonstration of an analytical-redundancy-based algorithm developed as part of the ADIA program. Over the past 45 years, hydromechanical implementations of turbine engine control systems have matured into highly reliable units. However, there is a trend toward increased engine complexity. Engine control has become increasingly complex and has evolved from a hydromechanical to a full authority digital electronic (FADEC) implementation. These FADEC-type controls have to demonstrate the same or improved levels of reliability as their hydromechanical predecessors.

Various redundancy management techniques have been applied to both the total control system and to individual components. Studies[1] have shown that the least reliable of the control system components are the engine sensors. Sensor redundancy will be required to achieve adequate control system reliability. One important type of sensor redundancy is analytical redundancy[2] (AR), which uses a model to generate redundant information that can be compared to measured information to detect failures. AR-based systems can have cost and weight savings over hardware redundancy.

Considerable progress has been made in the application of analytical redundancy to improve turbine engine control system reliability. However, little has been done to demonstrate AR-based techniques on real engine systems. One exception was a flight test program[3] where selected out-of-range (hard) sensor faults were induced and the resulting actions of the control evaluated. The test program included a ground-thrust stand evaluation and a flight test. The sensors that failed during the flight test included the compressor inlet variable geometry sensor, inlet static pressure, burner pressure, and fan turbine inlet temperature. Most failures were detected and accommodated. However, a recreation of a broken line (hard) burner pressure went undetected. Pilot response to aircraft performance after accommodation was favorable.

Received Sept. 1, 1989; revision received Jan. 23, 1990.

*Senior Research Engineer, Advanced Controls Technology Branch. Member AIAA.
†Research Engineer, Advanced Controls Technology Branch.
‡Senior Operations Engineer, Engine Systems Branch.

This flight program as well as other surveyed accomplishments[2] point to several current technology needs. These needs include 1) the ability to detect small (soft) failures over a wide range of operating conditions, 2) real-time implementations of algorithms capable of detecting soft failures, 3) a comparison of failure detection algorithm complexity vs performance, and 4) full-scale demonstration of a soft failure detection capability. The ADIA program has addressed all of these technology needs.

The ADIA program was organized into four phases: development,[4,5] implementation,[6] evaluation,[7,8] and demonstration. In the first three phases, the algorithm was designed using advanced filtering and detection methodologies, implemented in microprocessor-based hardware, and evaluated using a real-time engine simulation running on a hybrid computer. This paper describes the demonstration of the ADIA algorithm on a full-scale F100 engine in the NASA Lewis Research Center altitude test facility. The objective was to demonstrate the predicted performance of the ADIA algorithm on realistic hardware over a wide range of engine operating condition. These conditions include altitude, Mach number, and power variations.

The paper begins with a description of the test bed used in the demonstration of the ADIA algorithm. Next, a description of the ADIA algorithm is given followed by a description of the implementation hardware. The results of the demonstration are presented and compared with those obtained during a real-time simulation evaluation of the algorithm. Conclusions and recommendations are presented.

Test Bed Description

The algorithm was demonstrated using a test bed system, shown in Fig. 1 which consists of the altitude facility, the engine system, the control computer, which implements the control algorithm and the ADIA algorithm, and the sensor failure simulator. Each of these subsystems is briefly described. Additional details of each subsystem are available.[9]

Altitude Facility

To create the effects of aircraft speed and altitude, the engine is contained within an altitude test facility. This facility, the Propulsion System Laboratory Test Cell 4 (PSL-4), has a working section diameter of 23 ft and is one of two major test cells utilized for air breathing propulsion system testing at the NASA Lewis Research Center. PSL-4 can provide simulated flight environmental conditions ranging from 5000 to 78,000 ft and from 0.0 to 3.0 flight Mach numbers.

F100 Engine System

The engine system consists of an F100 turbofan engine and its associated control actuators and sensors. The basic F100

engine is a 25,000 lbf thrust class Pratt & Whitney engine. It is a low-bypass, high-compression-ratio twin-spool turbofan with a mixed-flow augmentor. A unified fuel control handled the primary and some secondary controlling functions. A supervisory engine electronic control (EEC) provided fine trim for the engine. This combination of unified fuel control and EEC is the standard control delivered with the engine and is called the bill-of-material (BOM) control. The BOM control was used for engine start, engine shutdown, and as a backup control to the ADIA research control.

The test engine has four controlled inputs, five sensed outputs, and four sensed environmental variables. These variables are defined as follows:

Controlled engine inputs, U_{com} and U_m:

WF = main combustor fuel flow
AJ = exhaust nozzle area
CIVV = fan inlet variable vanes
RCVV = rear compressor variable vanes

Sensed engine outputs, Z_m:

N1 = fan speed
N2 = compressor speed
PT4 = burner pressure
FTIT = fan turbine inlet temperature

Sensed environmental variables, E_m:

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

The sensed environmental variables were not covered by the ADIA logic. Although TT25 is a sensed engine output variable, it was used only as a scheduling variable in the control (like TT2). Thus, it was viewed as an environmental variable.

Control Computer

To conduct the engine demonstration, an implementation of the ADIA algorithm was integrated with an existing microcomputer implementation of the F100 multivariable control (MVC) algorithm. To satisfy the control update interval stability requirement of 40 ms, three 80186 microprocessors (μPs) operating in parallel were used. Data were transferred between μPs through dual-ported memory. Synchronization between μPs was achieved through interrupts. The total memory requirement for the three μPs was 54 Kbytes for the algorithm and 17 Kbtes for the real-time executive. In all cases, the code and constants were about 65% and the data or variables about 35% of the total memory required. Additional implementation details can be found in Ref. 6.

Multivariable Control System

The MVC system shown in Fig. 1 is essentially a model-following, proportional-plus-integral control. The MVC[10] was previously demonstrated in an altitude test of an F100 engine.[11] The components of the control were the reference point schedules, the transition schedules, the proportional control logic, the integral control logic, and the engine protection logic. The reference point schedules generate a desired engine operating point in response to the pilot's thrust command, i.e., power level angle (PLA) and sensed engine environment. The transition logic generates rate limited command trajectories for smooth transition between steady-state operating points. The proportional and integral control logic minimize

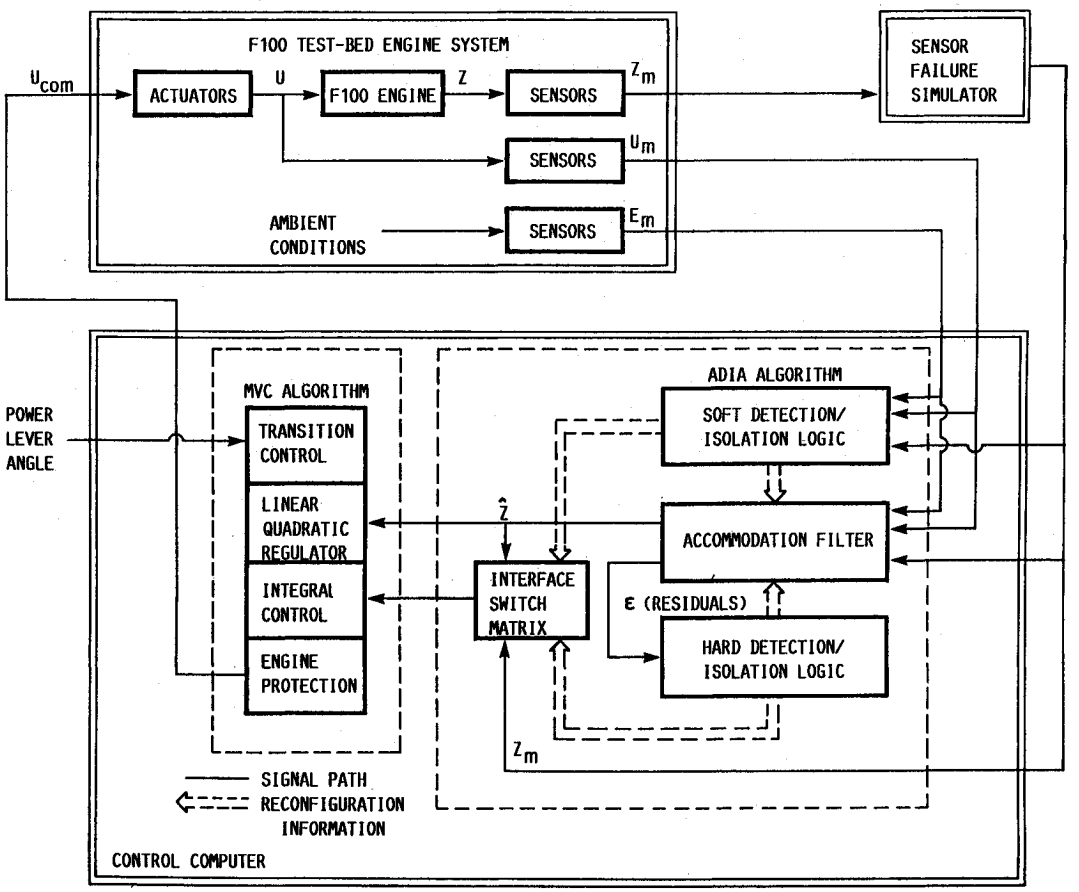


Fig. 1 F100 test bed system.

transient and steady-state deviations from the commanded trajectories. The engine protection logic limits the size of the commanded engine inputs. The normal control mode in the MVC logic uses fuel flow to set engine fan speed and nozzle area to set nozzle pressure (engine pressure ratio). However, at those conditions where limiting was required, fuel flow can be used to limit the maximum FTIT, the maximum PT4, or the minimum PT4.

Algorithm Description

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

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

Accommodation Filter

The accommodation filter incorporates an engine model along with a Kalman gain update to generate estimates of the engine state $\hat{X}$ and the engine outputs $\hat{Z}$ as follows:

$$\frac{d\hat{X}}{dt} = F(\hat{X} - X_b) + G(U_m - U_b) + K\epsilon \quad (1)$$

$$\hat{Z} = H(\hat{X} - X_b) + D(U_m - U_b) + Z_b \quad (2)$$

$$\epsilon = Z_m - \hat{Z} \quad (3)$$

Here, the subscript b represents the base point (steady-state engine operating point) and X is the 4×1 model state vector, U_m the 4×1 sensed control vector, and Z_m the 5×1 sensed output vector. The matrix K is the Kalman gain matrix and ϵ is the residual vector. The F, G, H, and D matrices were the appropriately dimensioned model system matrices. Their individual matrix elements along with those of K were corrected by the engine inlet conditions E_m scheduled as nonlinear functions[5] of Z_b. These nonlinear functions allow continuous correction of the model parameters throughout the flight regime. An improvement that was added to the accommodation filter was the incorporation of integral action to improve steady-state accuracy of the FTIT estimate $\hat{Z}_5$. One important engine control mode was the limiting of FTIT at high-power operation. Because the FTIT sensor was relatively slow, control action was based on the dynamically faster FTIT estimate. Because the FTIT limiting control has integral action, high degree of steady-state accuracy in the FTIT estimate was required to ensure satisfactory control. This accuracy was accomplished by augmenting the filter with the following additional state and output equations:

$$\frac{dT}{dt} = K_6\epsilon \quad (4)$$

$$\hat{\text{FTIT}} = \hat{Z}_5 + T \quad (5)$$

where K_6 is a gain matrix, T the temperature bias, and $\hat{Z}_5$ the unbiased temperature estimate. The addition of these dynamics, while improving FTIT estimation accuracy, results in larger minimum detectable FTIT drift failure. This filter structure, which includes the FTIT bias state, is the structure used in the accommodation filter and all the hypothesis filters used in the soft detection and isolation logic.

Reconfiguration of the accommodation filter after the detection and isolation of a sensor failure was accomplished by forcing the appropriate residual element to zero. For example, if a compressor speed sensor failure (N2) has been isolated, the effect of reconfiguration was to force $\epsilon_2 = 0$. This was equivalent to setting sensed N2 equal to the estimate of N2 generated by the filter. The residuals generated by the accommodation filter were used in the hard failure detection logic.

Table 1 Hard detection threshold magnitudes

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

Hard Failure Detection and Isolation Logic

The hard sensor failure detection and isolation logic is straightforward. To accomplish hard failure detection and isolation, the absolute value of each component of the residual vector was compared to its own threshold. If the residual absolute value was greater than the threshold, then a failure was detected and isolated for the sensor corresponding to the residual element. Threshold sizes were based on the standard deviation of the noise on the sensors. These standard deviation magnitudes were than adjusted, i. e., increased, to account for the effect of modeling errors on accommodation filter estimation accuracy. These adjustments were determined by simulation tests. The hard detection threshold values were chosen as twice the magnitude of these adjusted standard deviations. These magnitudes ares summarized in Table 1.

Soft Failure Detection and Isolation Logic

The soft failure detection logic consists of multiple-hypothesis-based testing. Each hypothesis was implemented using a Kalman filter. The soft failure detection/isolation logic structure is shown in Fig. 2. A total to six hypothesis filters are shown: one for normal mode operation (H_0) and five for the failure modes (one for each engine output sensor, H_1–H_5). The structure for each hypothesis filter was identical to the accommodation filter [Eqs. (1-3)]. However, each hypothesis filter uses a different, reduced set of measurements. For example, the first hypothesis filter (H_1) uses all of the sensed engine outputs except the first, N1. The second uses all of the sensed outputs except the second, N2, and so on. Thus, each hypothesis filter generates a unique residual vector ϵ_i. From this residual, each hypothesis filter generates a statistic or likelihood based on a weighted sum of squared residuals (WSSR).

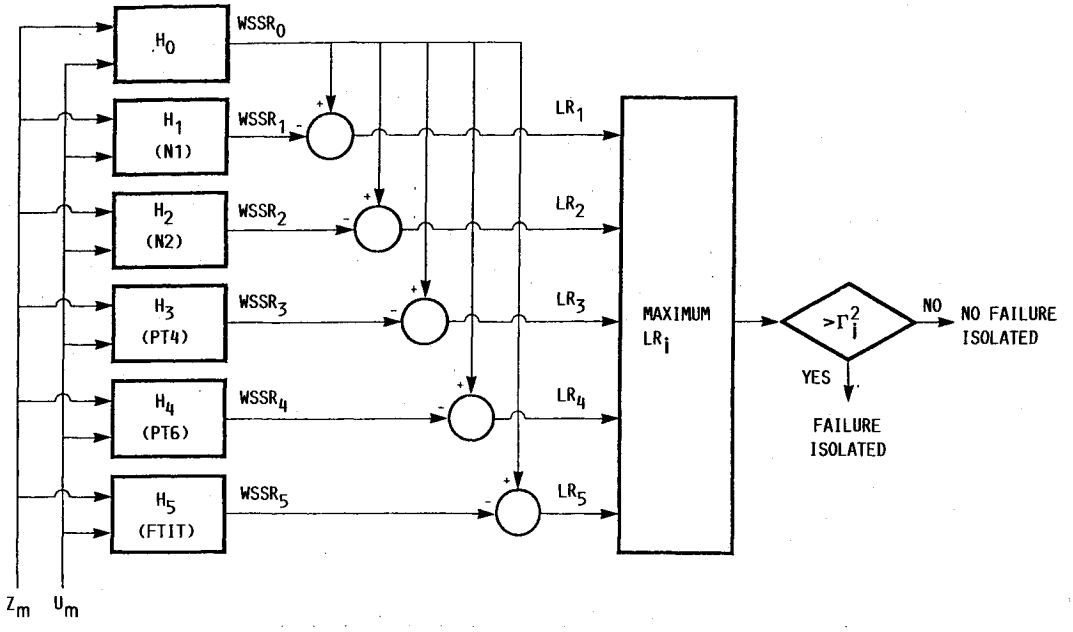


Fig. 2 Soft failure detection and isolation logic structure.

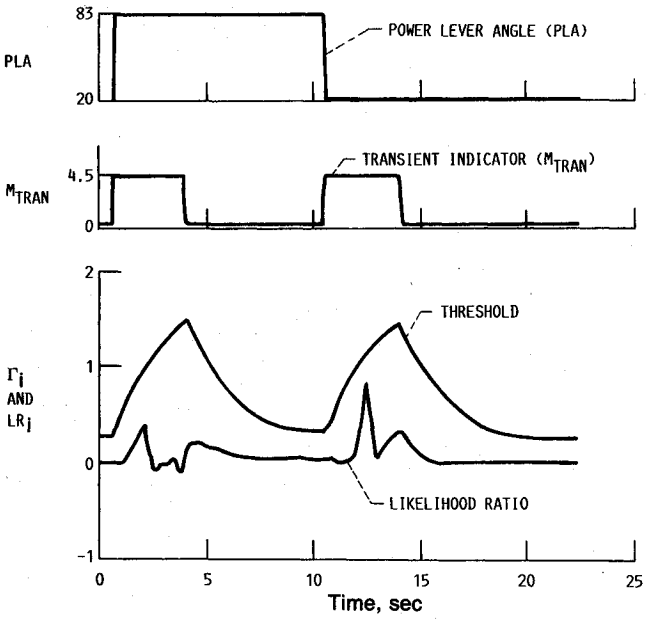


Fig. 3 Soft failure detection adaptive threshold.

Assuming Gaussian sensor noise, each sample of ϵ_i had a certain likelihood or probability p_i given by

$$L_i = p_i(\epsilon_i) = ke^{-\text{WSSR}_i} \tag{6}$$

where k is a constant and $\text{WSSR}_i = \epsilon_i^T \Sigma^{-1} \epsilon_i$ with $\Sigma = \text{diag}(\sigma_i^2)$. Here, T denotes matrix transposition and σ_i are the adjusted standard deviations defined in Table 1. These standard deviation values scale the residuals to dimensionless quantities that can be summed to form a WSSR. The WSSR statistic was smoothed to removed gross noise effects by a first-order lag with a time constant of 0.1 s. Mathematically, when the log of the ratio of likelihoods is taken, then

$$\text{LR}_i = \log(L_i/L_0) = \text{WSSR}_0 - \text{WSSR}_i \tag{7}$$

If the maximum LR_i exceeds the soft failure detection and isolation threshold, then a failure is detected and isolated, and accommodation occurs. If a sensor failure has occurred in N1, for example, all of the hypothesis filters except H_1 will be corrupted by the faulty information. Thus, each LR_i will be small except for LR_1. Thus, LR_1 will be the maximum and it will be compared to the threshold to detect the failure.

Adaptive Threshold

Initially, the soft failure detection/isolation threshold was determined by standard statistical analysis of the residuals to set the confidence level of false alarms and missed detections. Next, the threshold was modified to account for modeling error by simulation analysis. It was soon apparent from initial evaluation studies that transient modeling error was dominant in determining the fixed threshold level. It was also clear that this threshold was too large for desirable steady-state operation. Thus, an adaptive threshold was incorporated to make the algorithm more robust to transient modeling error while maintaining steady-state performance.

The adaptive threshold, defined as

$$\Gamma_i = \Gamma_{iss}\,(\Gamma_{\text{exp}} + 1) \tag{8}$$

$$\tau \frac{d\Gamma}{dt}\,\text{exp} + \Gamma_{\text{exp}} = M_{\text{tran}} \tag{9}$$

was heuristically determined and consists of two parts. One part, Γ_{iss}, is the steady-state detection/isolation threshold, which accounts for steady-state or low-frequency modeling error. The second part, Γ_{exp}, accounts for the transient, or high-frequency modeling error. The adaptive threshold was triggered by an internal control system variable, M_{tran}, which was indicative of transient operation. The values of Γ_{iss}, τ, and M_{tran} were found by experimentation to minimize false alarms during transients. When the engine experiences a transient, M_{tran} was set to 4.5, otherwise it was 0. The threshold time constant $\tau = 2$ s was used. The adaptive threshold expansion logic enabled Γ_{iss} to be reduced to 40% of its original value, which results in an 80% reduction in the detection/isolation threshold Γ_i^2 in steady state. The adaptive threshold logic is illustrated in Fig. 3 for a PLA pulse transient.

Failure Accommodation

For accommodation, two separate steps were taken. First, all seven of the filters (the accommodation filter and the six hypothesis filters) were reconfigured to remove the failed sensor from further use in the filters. Second, if a soft failure was

Table 2 Test matrix: number of sensor failure tests conducted

Test	Altitude (1000 ft)/Mach number/power level angle, deg										
	10/0.6/50	10/0.6/80	30/0.9/50	30/0.9/80	10/0.9/50	10/0.9/80	45/0.9/70	10/1.2/80	50/1.8/80	35/1.9/80	55/2.2/80
Hard	5	6	—	—	—	—	—	—	—	—	—
Soft	7	6	6	5	11	4	5	5	5	7	5
Drift	7	5	6	5	11	4	5	5	5	5	5
Sequence	6	1	—	—	—	—	—	—	—	—	—
Simultaneous	—	1	—	—	—	—	—	—	—	—	—

Table 3 Test matrix: number of PLA transient tests conducted

Test	Altitude (1000) ft)/Mach number			
	10/0.6	30/0.9	45/0.9	10/1.2
Pulse	4	3	2	1
Single	9	4	4	1
All	1	2	1	—

detected, the states and estimates of all seven filters were updated to the values of the hypothesis filter, which corresponds to the failed sensor.

Sensor Failure Simulator

The sensor failure simulator (SFS)[12] is a personal computer-(PC) based device that can repeatably simulate sensor failures in control systems. The PC drives analogy circuitry for simulating sensor failures in real time. The SFS inputs were the sensed engine outputs. The analog interface includes five separate analog signal paths with independent digital control of modifications (i.e., sensor failures) made to the sensed SFS inputs. The SFS can modify any combination of the five ADIA sensor signals and send the modified (or failed) signals to the control system.

Test Definition

The objective of the engine test was to demonstrate the operation and performance of the ADIA algorithm and its implementation over a typical flight envelope of an F100 engine. The flight envelope selected was for an F-15 aircraft. An engine operating condition within this envelope was defined as the operating altitude and Mach number of the engine. An engine operating point was specified by the operating condition and the power setting of the engine given by the PLA setting in degrees.

The test matrix, shown in Tables 2 and 3, summarizes the two classes of tests conducted to demonstrate the algorithm. In Table 2, the different engine operating points (altitude, Mach number, PLA) used for the demonstration are listed across the top of the matrix and the different failure tests conducted at these operating points are listed along the side. In Table 3, the different engine operating conditions (altitude, Mach number) used for the demonstration are listed across the top of the matrix and the different transient tests conducted at these operating points are listed along the side. The numbers in the matrix indicate the number of tests conducted. For example, five hard failure tests were conducted at the 10,000 ft/Mach 0.6/50-deg PLA operating point.

The rationale used to select the test matrix operating points and conditions was to duplicate as many of the points and conditions used in the F100 Multivariable Control Program[11] as possible, to avoid high fan inlet pressures, and to reasonably span the envelope. This rationale was a compromise to take maximum advantage of previous results for comparison, to limit engine operation risk, and to validate over the full operation envelope. Engine power (PLA) settings were selected that represent maximum nonafterburning (intermediate) thrust (PLA = 80) and approximately 50% of intermediate thrust (PLA = 50 and 70, depending on operating condition).

Table 4 Test Definitions

Test	Description
Sensor failure	
Hard	Large magnitude bias failure
Soft	Small magnitude bias failure
Drift	Small magnitude drift failure
Sequence	Sequence of successive sensor failures
Simultaneous	Two simultaneous sensor failures
PLA transient	
Pulse	Idle-to-intermediate-to-idle transient PLA excursion. Intermediate power level is maintained for 10 s.
Single	Pulse test with single sensor failure accommodated before initiating the transient.
All	Same as pulse test except tht minimum power level is raised slightly above idle, maximum power level is decreased slightly below intermediate, and the engine is controlled without using any sensed engine output information in the control.

The different tests used in the demonstration were selected to completely define detection performance for three common failure modes: hard, soft, and drift. Also, tests were conducted to determine engine control performance with and without engine sensor failures. The sensor failure and PLA transient tests are summarized in Table 4.

Algorithm Demonstration

This section describes the demonstration of the ADIA algorithm on a full-scale F100 turbofan engine. The procedure and the results of the demonstration are discussed.

Procedure

The overall test philosophy was based on a building-block approach, First, engine operation under BOM control was demonstrated at the test matrix conditions. Engine control was then transferred to MVC and engine operation was demonstrated. Next, the engine output estimates from the accommodation filter were included in the control. Finally, the remainder of the ADIA logic was enabled. Data were taken at each step in this test procedure. After the ADIA logic was completely enabled, the tests required by the test matrix definition (listed in Table 2 and 3) were accomplished.

Results

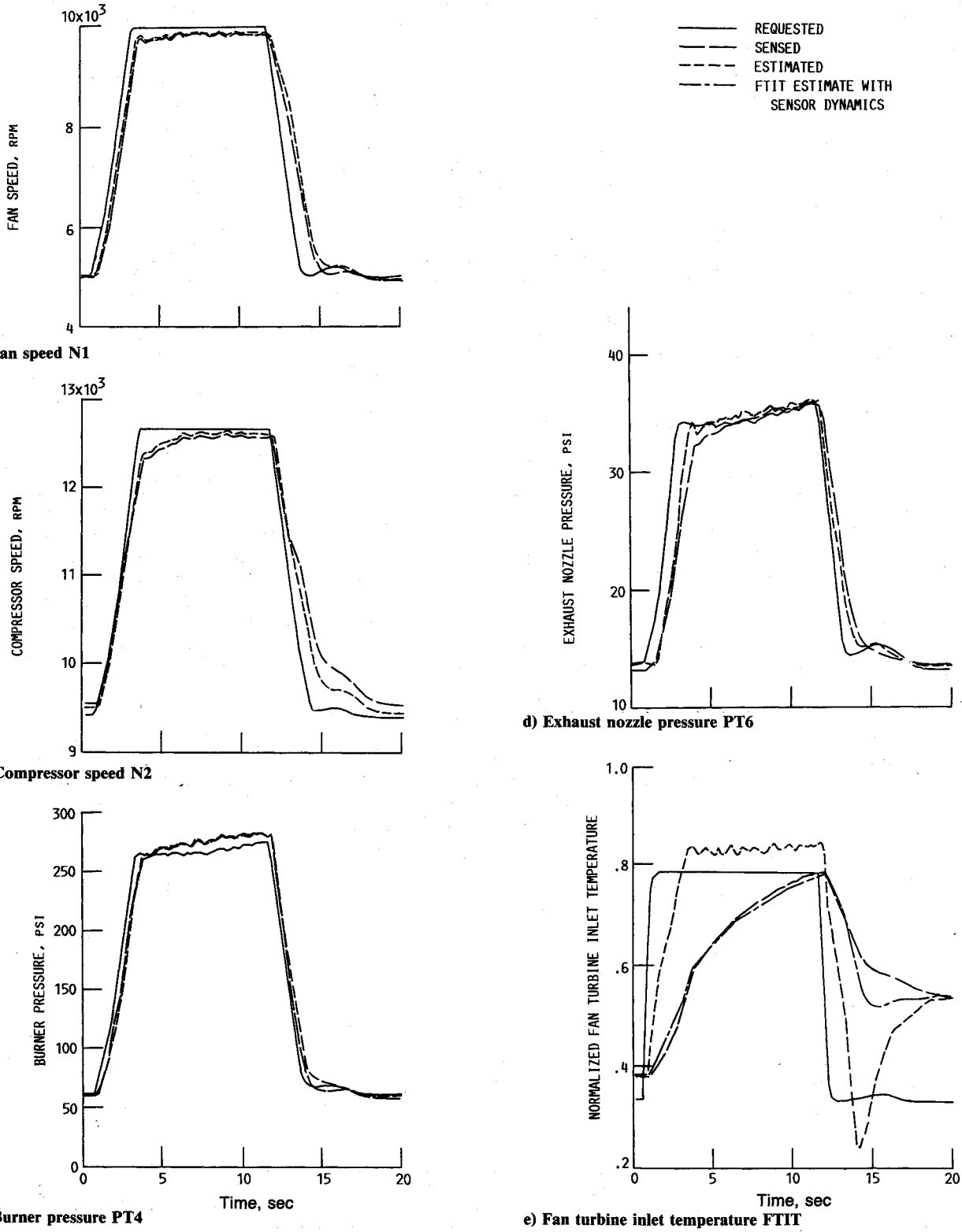
Three types of demonstration results are discussed. The first is engine control performance without the ADIA logic. This establishes a baseline of comparison for engine performance with and without the ADIA logic enabled. Next, the estimation accuracy of the accommodation filter with no sensor failures is described. Excellent estimation accuracy is the principal requirement of high-performance failure detection. The third results section describes the detection, isolation, and

accommodation performances of the ADIA algorithm. The discussion in this section summarizes the extensive results obtained.[9]

Engine Control Performance Baseline

Engine performance for two different control modes was examined. The control modes were MVC only and MVC-ADIA with accommodation filter estimates in the loop. An examination of the difference in performance of the two modes will reveal the impact of using the accommodation filter estimates as feedback information in the control loop.

The steady-state performance of the two control modes was compared by examining the set point deviation of the five engine outputs. The set point deviation of an output was defined as the difference between the steady-state reference point value requested by the control and the sensed value of that output. Steady-state performance with and without the ADIA logic was generally quite similar. In some cases, the set point deviation errors with and without the ADIA logic were in opposite directions, however, the error magnitudes were similar. In the case of burner pressure, some large discrepancies existed. These were the result of a reference point mis-



a) Fan speed N1

b) Compressor speed N2

c) Burner pressure PT4

d) Exhaust nozzle pressure PT6

e) Fan turbine inlet temperature FTIT

Fig. 4 Transient estimation accuracy for various engine outputs (operating condition, 10,000 ft/Mach 0.6).

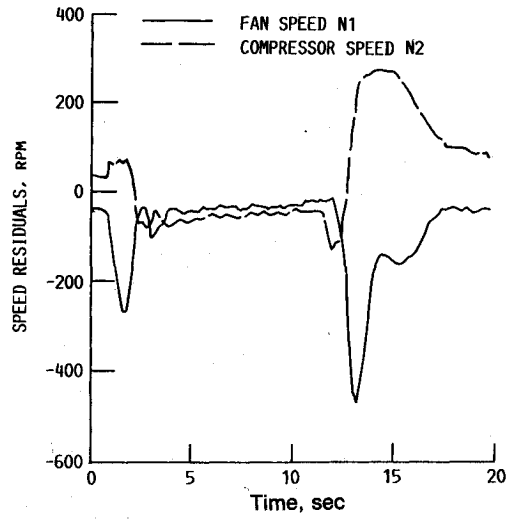


Fig. 5 Residual pulse response for engine speeds (operating condition, 10,000 ft/Mach 0.6).

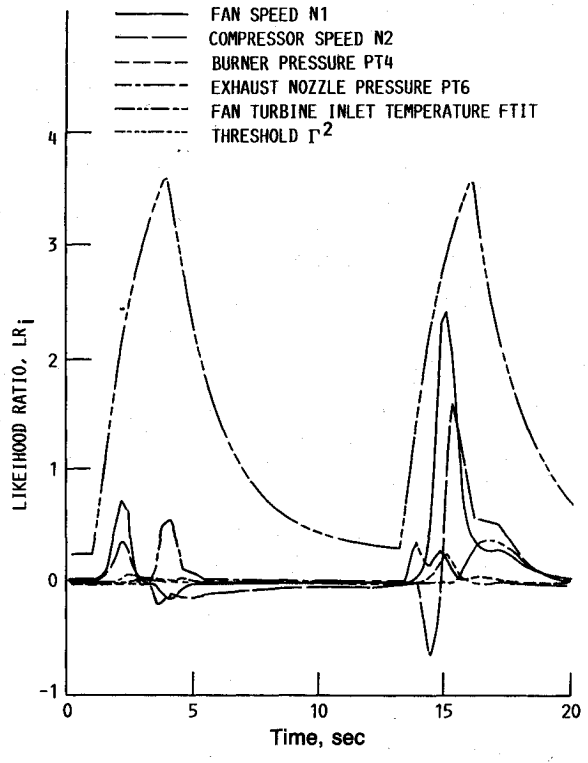


Fig. 6 Likelihood ratio (LR) for various engine outputs (operating condition, 10,000 ft/Mach 0.6).

match. These discrepancies were reduced by a simple correction to the burner pressure reference point schedule. This correction decreased the scheduled value of the PT4/PT2 ratio by 0.75. This same correction was applied to all subsequent engine testing and was viewed as simply a slightly different reference point schedule for PT4. The set point deviation results demonstrated generally good steady-state control performance (i.e., small set point deviation errors) throughout the operating envelope.

Estimation Accuracy

The most important element in determining ADIA algorithm performance was the accuracy of the engine output estimates used in the algorithm. These estimates were generated by the accommodation filter, which incorporates a simplified engine model. Accuracy was studied by examining a residual, that is, the difference between the sensed and estimated output values. Both steady-state and transient accuracy results were studied.

Steady-state accuracy results for each operating condition except 10,000 ft/Mach 0.9 were obtained. The steady-state residual values of N1 and N2 were generally quite good (i.e., <100 rpm) for all conditions and PLA settings. Steady-state estimate accuracy for the MVC and MVC/ADIA control modes, as measured by the N1, N2, PT6, and FTIT residual magnitudes, was essentially the same. A sample of the transient estimation accuracy of the accommodation filter is presented in Fig. 4. This figure shows an idle-to-intermediate-power PLA pulse transient generated at the 10,000 ft/Mach 0.6 condition. The variables shown demonstrate the excellent estimate accuracy achieved. The typical format for these results includes the requested, sensed, and estimated values for each engine output. In the FTIT results, the estimate of FTIT without sensor dynamics was also shown for comparison. This unlagged estimate was the control information used in the integral temperature limiting logic. A more sensitive indication of accuracy was obtained by studying the residuals for the transient presented in Fig. 4. These residuals, presented in Fig. 5, clearly show the obvious deterioration of accuracy during the transient. Additionally, the loss of accuracy seems to be directly related to the frequency of the input.

To complete the study of transient accuracy at the 10,000 ft/Mach 0.6 condition, the likelihood ratios (LR_i) of the engine outputs are presented in Fig. 6. Also shown in this figure is the detection threshold. If a LR_i exceeds this threshold, then a failure is detected. The adaptive nature of the threshold was clearly seen in Fig. 6 as it avoids false failure detections particularly during the engine deceleration. Recalling the results of Figs. 4 and 5, the LR_i of Fig. 6 represent accurate estimation. Note that the adaptive threshold was very conservative (high) in the 45,000 ft/Mach 0.9 case and could be reduced easily for improved detection performance at this condition.

Detection/Accommodation Performance

The criteria used to evaluate detection, isolation, and accommodation performance were 1) minimum detectable bias values and drift rates, 2) elapsed time between sensor failure and detection, 3) steady-state performance degradation after failure accommodation, and 4) transient response of the engine to the filter and control reconfiguration resulting from failure accommodation. Two general failure types were studied, hard and soft sensor failures.

The first type of sensor failure considered in the demonstration testing was a hard failure. Because hard failures are detected easily, they were examined at only one operating condition, 10,000 ft/Mach 0.6. The ADIA algorithm exhibited excellent hard detection performance at this condition. The minimum detectable magnitudes for hard failures were determined by design and given in Table 1. The time to hard sensor failure detection is, because of the logic, always less than two control update intervals or 80 ms. Hard failures were injected in each of the engine sensor output signals. Successful detection and accommodation of the failure was accomplished in each case with no false alarms or missed detections. In addition, no false alarms in the hard detection logic were encountered during the subsequent soft failure demonstration.

The engine performance measure, engine pressure ratio (EPR), which is almost linearly related to thrust, is defined as

$$\mathrm{EPR} = \frac{\mathrm{PT6}}{\mathrm{PT2}} \tag{10}$$

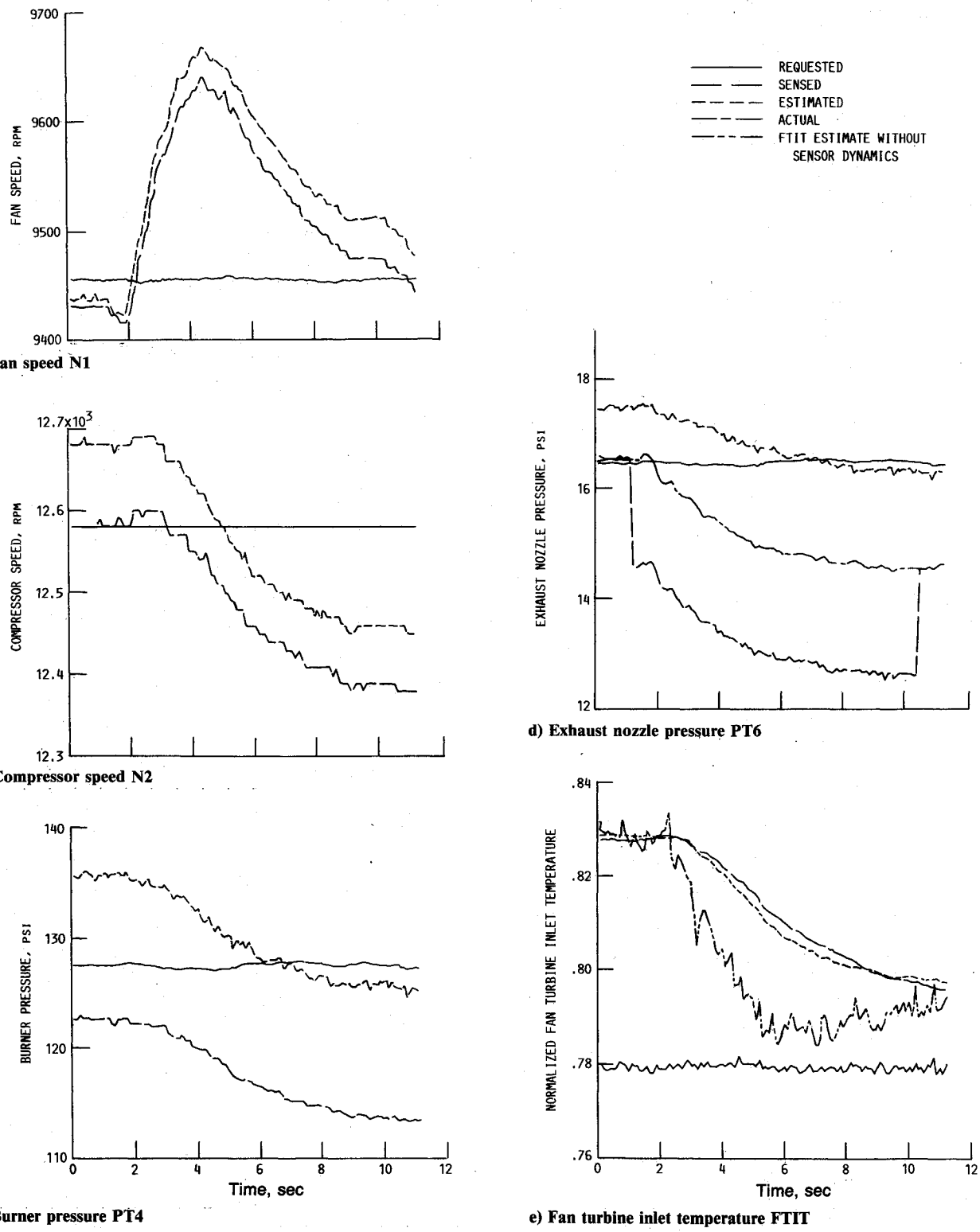
The change in EPR following the accommodated hard sensor failure was used as a measure of accommodation performance. Here, the percent change in EPR

$$\delta\mathrm{EPR} = \frac{100\ (\mathrm{EPR_{T0}} - \mathrm{EPR_{TF}})}{\mathrm{EPR_{T0}}} \tag{11}$$

where EPR_0 is the steady-state engine pressure ratio before the failure and EPR_{TF} is the steady-state EPR after the failure, is defined to be the change in steady-state engine performance. For the hard failure detection and accommodation experiments, the δEPR results were $< 6\%$ in all cases, well below the critical level of 10%. In fact, most of the performance changes were negligible.

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

The minimum detectable magnitudes of soft sensor bias and drift failures were determined for each of the five sensors considered at each of the operating points defined in the test matrix. The minimum detectable drift magnitudes were determined by finding the smallest detectable drift failure such that a failure was detected approximately 5 s after failure inception. In general, there was good agreement between those minimum failure detection magnitudes observed in the test and those predicted by the real-time hybrid evaluation.[8] This



a) Fan speed N1

b) Compressor speed N2

c) Burner pressure PT4

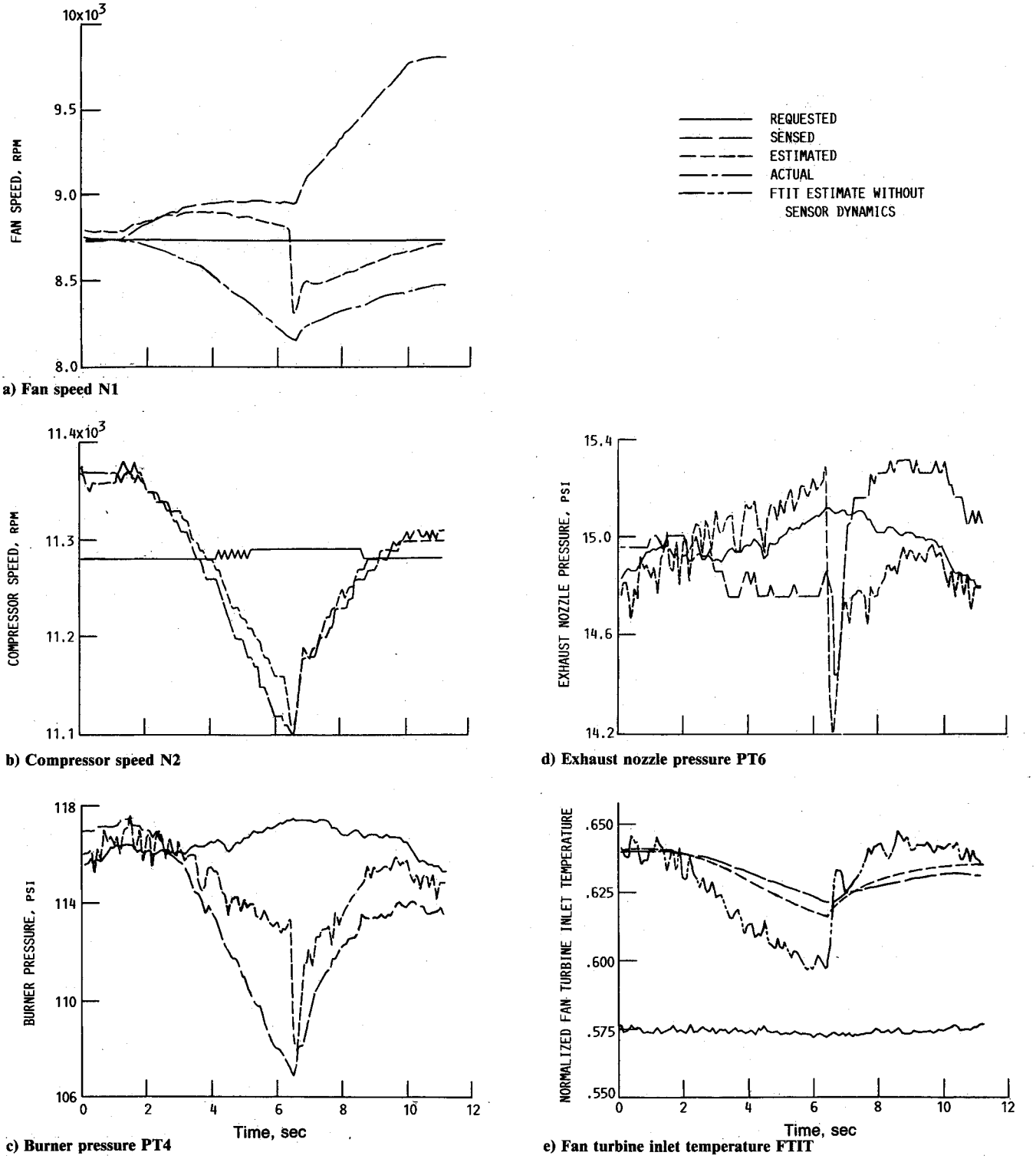
d) Exhaust nozzle pressure PT6

e) Fan turbine inlet temperature FTIT

Fig. 7 Transient response to exhaust nozzle pressure (PT6) bias failure for various engine outputs (operating point, 50,000 ft/Mach 1.8/military power).

agreement demonstrates the excellent fidelity of the model used in the algorithm and the simulation used in the evaluation. At two of the 11 conditions tested, 35,000 ft/Mach 1.9 and 50,000 ft/Mach 1.8, when successfully detecting bias failures for the two engine speeds, false alarm detections, usually on PT4, were encountered. Although clearly undesirable, these false-alarm detections did not significantly degrade engine performance or jeopardize engine safety. Several elements combined to cause these PT4 false alarms: errors in the steady-state schedule value of PT4, failed speed measurements, and a strong dependence of the filter estimates on the engine speed measurements at these high-altitude, high Mach number conditions.

The times to detection of the soft bias failures were all <0.1s. The steady-state accommodation performance of this class of failure was good. Again, percent changes in thrust (EPR) were determined for several operating points demonstrating subsonic and supersonic operation at military and medium power levels. All values obtained were well below the 10% critical level except for operating condition 50,000 ft/Mach 1.8 results, which show a 12% change in thrust for a PT6 sensor failure. This result was due to the low nominal value of PT6 at this condition (16.5 psi). The actual change in PT6 caused by the modeling error in the accommodation filter was only 2 psi and was considered relatively small. This change appears large when compared with the low nominal



a) Fan speed N1

b) Compressor speed N2

c) Burner pressure PT4

d) Exhaust nozzle pressure PT6

e) Fan turbine inlet temperature FTIT

Fig. 8 Transient response to fan-speed (N1) drift failure for various engine outputs (operating point, 30,000 ft/Mach 0.9/medium power).

value. Figure 7 shows the transient response of the PT6 bias failure at the 50,000 ft/Mach 1.8 condition. The nozzle pressure bias of 3 psi was introduced at 1 s with detection occurring within 40 ms. The trajectories shown were the response to the ADIA algorithm's accommodation of the failure. Integral control of PT6 was active at this condition. Thus, any error in the PT6 estimate will result in a trim of the engine operating point. The 1.5–2.0 psi error between the estimate and actual nozzle pressure was evident. At 10 s, the failure bias was removed from the sensed signal. The fan speed, compressor speed, and burner pressure responses showed good estimation accuracy with only small, relatively constant modeling errors present. The burner pressure estimation error was the largest as a percent of nominal and was due to the error in the steady-state schedule at this point. Accuracy for FTIT was good. The estimate of FTIT without sensor dynamics is also shown for comparison. This unlagged estimate was the control information used in the integral temperature limiting logic. This logic was active before the sensor failure. With the engine trim after accommodation, the temperature limiting logic became inactive as engine temperature decreased. Typically, the other bias failure transient responses were better than the results of Fig. 7.

The steady-state accommodation performance results obtained for sensor drift failures were very good with most thrust changes being small and with none larger than the 10% level. Five false alarm detections were obtained during the drift failure testing of the two speed sensors. Compressors speed false alarms were encountered when demonstrating fan speed detection at operating conditions 50,000 ft/Mach 1.8 and 35,000 ft/Mach 1.9. Again, these false alarms were caused by an overdependence of the filters on the speed measurements. Also, burner pressure false alarms were encountered when demonstrating compressor failure detection at conditions 10,000 ft/Mach 0.9, 10,000 ft/Mach 1.2, and 50,000 ft/Mach 1.8. These false alarms were caused by reference point schedule errors. Although false alarms are undesirable, engine safety and performance were not significantly degraded. This was clear since the largest percent change in thrust of any of the five experiments involving false alarms was 6% for the N2 drift failure at condition 50, 000 ft/Mach 1.8. In fact, all of the larger thrust changes resulting from drift failure accommodation (except for the FTIT failure case at condition 10,000 ft/Mach 0.6) were at condition 50,000 ft/Mach 1.8 and were a result of the low nominal value of PT6 at this condition.

Shown in Fig. 8 is a fan-speed drift failure at the 30,000-ft/Mach 0.9 operating condition at medium power. Here, a fan-speed drift failure of 150 rpm/s was introduced at 1 s. Detection and accommodation occurred at 6.5 s (5.5 s after failure initiation) and the engine required about 4.5 s to return to a steady condition. During this experiment, the fan speed and nozzle pressure integral logic was active. The other four output responses show good estimation accuracy and relatively small transient disturbances to the sensor failure accommodation.

Additionally, detection performance for sequential failures was demonstrated. At condition 10,000 ft/Mach 0.6, six different sequences of soft failures were injected into the test bed system at medium power and one was demonstrated at intermediate power. One example of a failure sequence was to fail N1, then 4s later, fail N2, then PT4, and then PT6. Each of the seven sequences was a different permutation of the five sensors taken four at a time. Each sensor failure was a bias of approximately twice the minimum detectable magnitude (except N1, which was 1000 rpm to ensure detection). In each case, the algorithm successfully detected and accommodated each sensor failure in the correct order. The maximum percent change in EPR for the seven tests were all close to the critical 10% level except for the intermediate power case, which was <1%. In each case, these changes were experienced well into the transient when only two of the five sensors remained unfailed. These tests demonstrate the ability of the algorithm to continue to successfully perform even after most of the sensors have failed.

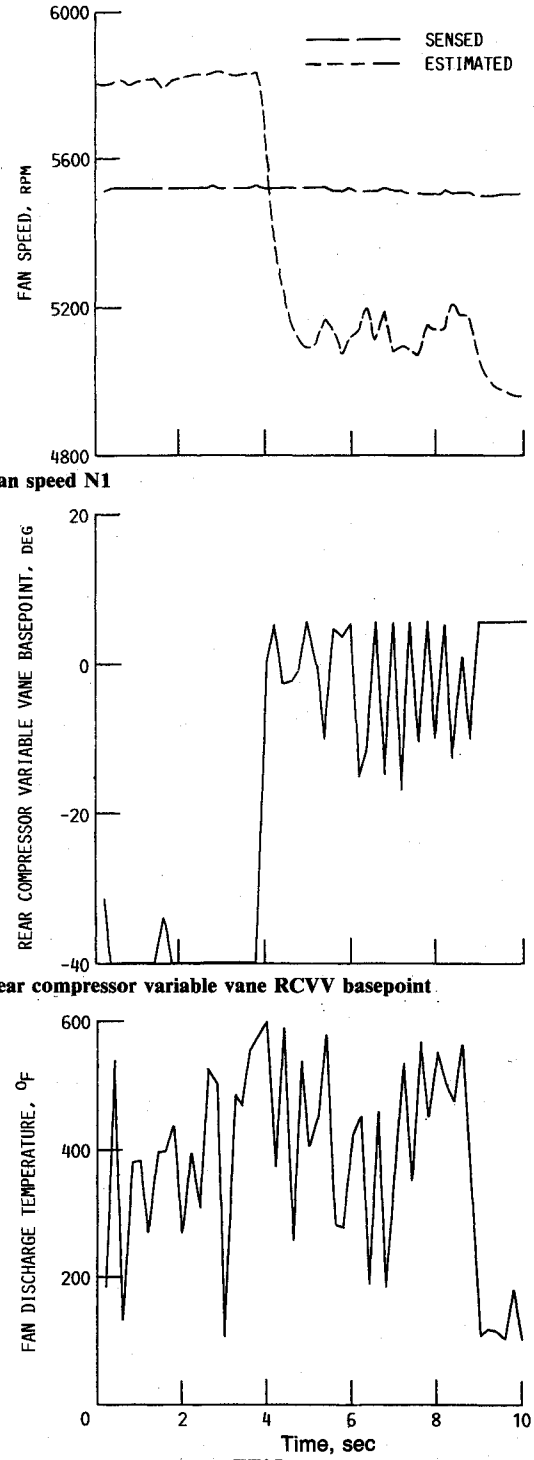
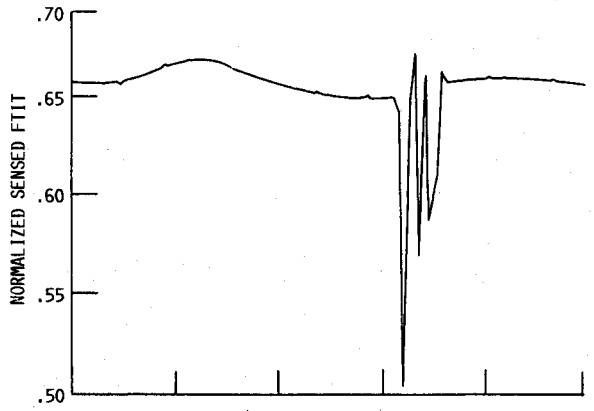


Fig. 9 Transient response to actual fan discharge temperature sensor failure (operating condition, 10,000 ft/Mach 0.6).

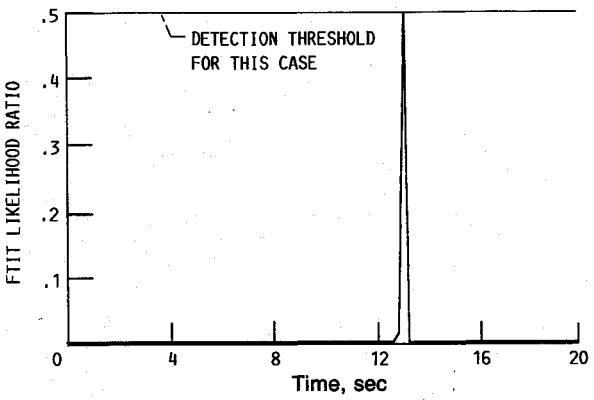
Finally, a simultaneous soft failure of PT4 and PT6 (both failed at the same instant of time) was injected into the engine system. The algorithm, although not designed for this extremely low probability event, successfully detected and accommodated this failure scenario. The resulting engine response to this failure scenario shows a small changes in fan speed (<2%) and EPR (<7%).

Actual Sensor Hardware Failure Detection

The SFS was designed to generate and inject realistic sensor failures into the engine control system. These injected failures



a) Sensed fan turbine inlet temperature FTIT



b) Fan turbine inlet temperature FTIT likelihood ratio

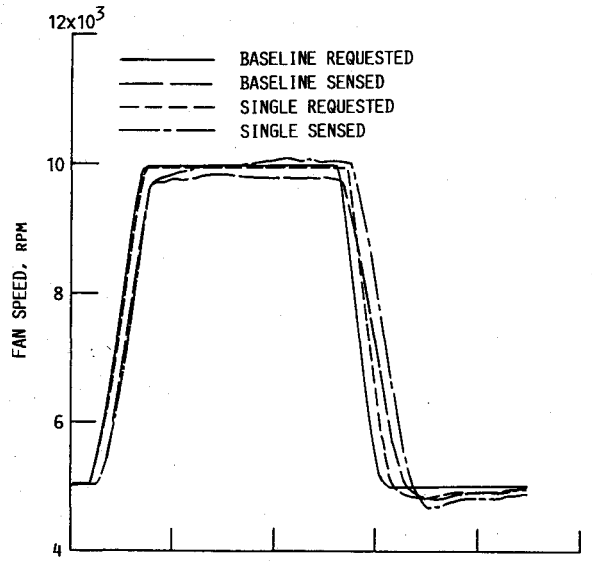
Fig. 10 Transient response to actual fan turbine inlet temperature instrumentation hardware failure.

represent the vast majority of test data. However, during engine testing, two unplanned failures of actual sensor hardware were detected by the ADIA logic. There were no missed detections of sensor hardware failures by the ADIA logic. In the first case, the fan discharge temperature thermocouple, which was not explicitly covered by the ADIA logic, failed. A fan speed false alarm detection was indicated by the algorithm (see Fig. 9a). On examination of the components of the fan speed estimate to determine the cause of the false alarm, it was observed that the RCVV base point value changed dramatically at the time of detection (see Fig. 9b). Because of the strong dependence of the RCVV base point on fan discharge temperature, it was clear that the simultaneous occurrence of these two events (an N1 failure detection and a sudden change in the RCVV base point) indicated a fan discharge temperature sensor failure. Although an examination of the fan discharge temperture measurement signal in Fig. 9c gives no clear indication, the failure was confirmed by inspection of the faulty instrument. A second thermocouple failed at a later time, which resulted in the same algorithm behavior. It was clear from this experience that some very simple, additional logic could be incorporated in the ADIA algorithm to cover fan discharge temperature sensor failures.

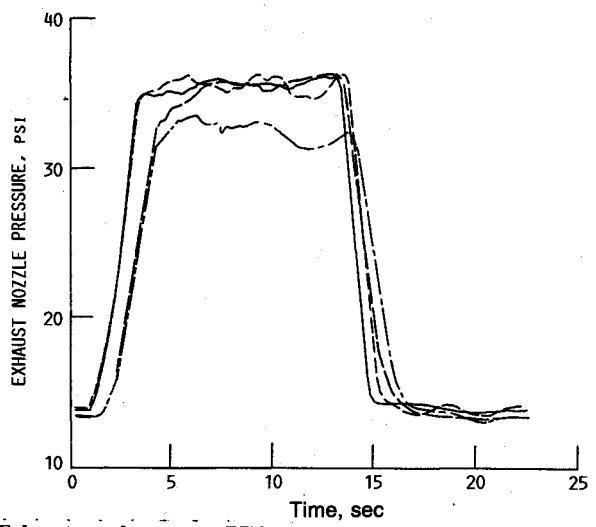
The second sensor hardware failure was associated with the FTIT measurement. About 12 s after the PT6 hard failure detection experiment, an FTIT soft failure was detected. From the sensed FTIT signal shown in Fig. 10a, it was clear that some transient anomaly occurred. The likelihood ratio for FTIT given in Fig. 10b shows the detection taking place at 13 s. The failure mode was indicative of a momentary "singing" of a signal conditioning amplifier. Although conclusive proof of a hardware failure was not obtained because of its nonrepeatability, this failure mode was not observed again once the suspected signal conditioning amplifier was replaced.

Transient Performance

Two PLA transient experiments were used to further demonstrate the successful accommodation, or postfailure performance, of sensor failures. The first experiment consisted of injecting, detecting, and accommodating a single sensor failure and then commanding a PLA pulse transient. Engine performance with this accommodated failed sensor was compared to normal mode engine performance. As seen from Table 2, 18 of these single failure PLA pulse tests were performed at five different operating points. The control performance of each of these experiments was compared to a baseline performance, the normal mode (i.e., no failure) pulse transient response. Here, control performance was the average absolute value of control error over the transient. The baseline average control error serves as a reference point to show relative changes for each of the single failure pulse responses. In general, the changes were small for each experiment. The largest speed error change of about 160 rpm at the 30,000 ft/Mach 0.9/50-deg point is, in fact, quite small when compared to the typical operating range of fan speed (5000–10,000 rpm). Results for the engine PLA pulse response with a single PT6 failure (10,000 ft/Mach 0.6/50 deg) and its baseline response are shown in Figs. 11. Results are shown for both fan speed and exhaust nozzle pressure. In general, performance was good since the desired or request values were



a) Fan speed N1

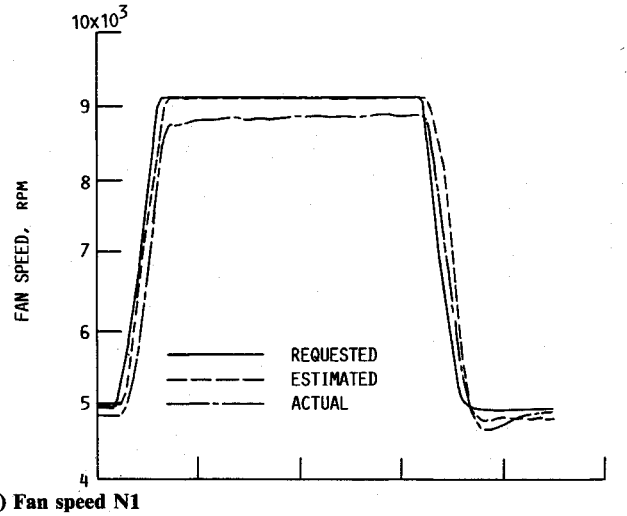


b) Exhaust nozzle pressure PT6

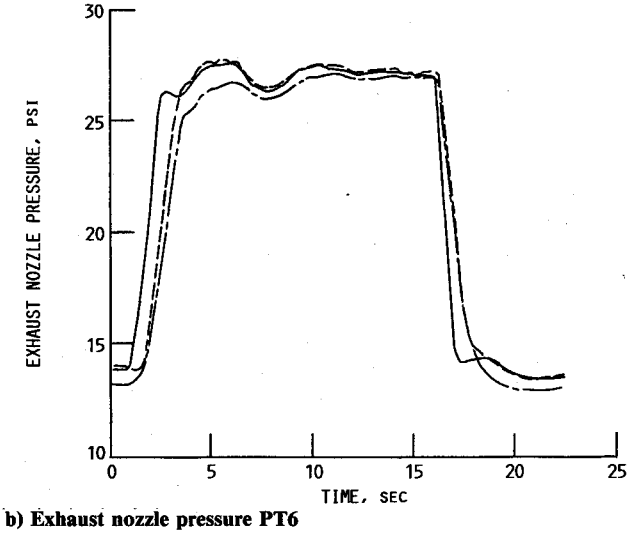
Fig. 11 Power lever angle pulse response to exhaust nozzle pressure single failure compared with baseline response (operating condition, 10,000 ft/Mach 0.6).

closely maintained. A slight drop in actual PT6 can be seen, but this was acceptable. In all other cases, the accommodated single failure transient performance was good. The fluctuations evident in PT6 at the intermediate power level were caused by an airflow interaction between the altitude test cell and the engine.

The second accommodation performance experiment demonstrated the excellent accuracy of the engine model. In this experiment, all of the engine sensors were failed and accommodated. Then, the engine was commanded to respond to a PLA pulse transient. From Table 3, it is seen that four open-loop PLA pulse transient experiments were performed at three different conditions. At condition 10,000 ft/Mach 0.6, the transient was from idle to about 75% of full power. As confidence in the ability to safely control the engine without engine output sensors increased, subsequent open-loop PLA pulse tests went to higher levels of power. The condition 30,000 ft/Mach 0.9 tests went from idle to 90% full power and the test at condition 45,000 ft/Mach 0.9 went from idle to full power. For the condition 10,000 ft/Mach 0.6 experiment, N1 and PT6 results are shown in Figs. 12. Here, excellent performance was demonstrated. Little or no overshoot was observed and engine steady-state performance was good. This demonstrates the capability of safe, predictable engine operation without any sensed engine output information over a slightly restricted power range. Again, the fluctuations in PT6 at high power were caused by an airflow interaction between the facility and the engine.



a) Fan speed N1



b) Exhaust nozzle pressure PT6

Fig. 12 Open-loop (all engine output sensors failed) power lever angle pulse response (operating condition, 10,000 ft/Mach 0.6).

For the condition 30,000 ft/Mach 0.9 experiment, all of the engine output sensors were failed and accommodated except for N1, which remained valid and in the control loop. This was required when, during the first attempt to accelerate the engine at this condition with all engine output sensors failed, the automatic "heal" logic declared the failed N1 sensor valid. The heal logic checks the residual of a failed sensor to determine if the failure signal has been removed. If the residual was sufficiently small for a significant period, the sensor was declared healed (i.e., the fault has vanished). When the heal logic brought the faulty N1 sensed information back into the control loop, the control utilized it as valid information. This incorrect healing was caused by transient modeling error. Although no safety or operational problems resulted from this incorrect logic decision, the failure bias was, nonetheless, removed from the measurement to obtain the results. The fan speed results were quite good, as would be expected. The PT6 response shows a constant error of about 1.5 psi, which was < 10% of nominal.

For the open-loop PLA pulse experiment of condition 45,000 ft/Mach 0.9, the responses were smooth and well behaved. However, there was a large delay or lag in engine performance. Also, there was some offset between requested and actual performance. Considering that there was no performance feedback information being used by the control, this was reasonable behavior. Additionally, engine operation was within safe bounds and stably controlled with the control errors being made in the conservative direction.

Conclusions

Sensor failure detection and accommodation were demonstrated at 11 different operating points, which included subsonic and supersonic conditions and medium and high power operation. The minimum detectable failure magnitudes represent excellent algorithm performance and compare closely to values predicted by simulation. Accommodation performance was excellent. Transient engine operation over the full power range with single sensors failed and accommodated was successfully demonstrated. Open-loop engine operation (all engine output sensors failed and accommodated) over at least 75% of the power range was also demonstrated at two different operating conditions. Engine operation with only one sensor operational (fan speed) was demonstrated at one operating condition.

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

Finally, it is concluded that the demonstrated high-performance detection, isolation, and accommodation capabilities of the ADIA algorithm justify incorporation of this technology in future, advanced propulsion systems.

References

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

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

[3]Myers, L. P., Baer-Riedhart, J. L., and Maxwell, M. D., "Fault Detection and Accommodation Testing on an F100 Engine in a F-15 Airplane," NASA TM-86735, July 1985.

[4]Beattie, E.C., LaPrad, R. F., McGlone, M. E., Rock, S. M., and Akhter, M. M., "Sensor Failure Detection System—for the F100 Turbofan Engine," NASA CR-165515, Aug. 1981.

[5]Beattie, E. C., LaPrad, R. F., Akhter, M. M., and Rock, S. M., "Sensor Failure Detection for Jet Engines," NASA CR-168190, May 1983.

[6]DeLaat, J. C., and Merrill, W. C., "A Real-Time Implementation of an Advanced Sensor Failure Detection, Isolation, and Accommodation Algorithm," NASA TM-83553, Jan. 1984.

[7]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, June 1986.

[8]Merrill, W. C., DeLaat, J. C., and Bruton, W. M., "Advanced Detection, Isolation, and Accommodation of Sensor Failures—Real-Time Evaluation," *Journal of Guidance, Control, and Dynamics*, Vol. 11, No. 6, 1988, pp. 517–526.

[9]Merrill, W. C., DeLaat, J. C., Kroszkewiczs, S. M., and Abdelwahab, M., "Advanced Detection, Isolation, and Accommodation of Sensor Failures—Demonstration Results," NASA TP-2836, Nov. 1988.

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

[11]Lehtinen, B., Costakis, W. G., Soeder, J. F., and Seldner, K, "F100 Multivariable Control Synthesis Program Results-Engine Altitude Tests," NASA TMS-83367, July 1983.

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