

NASA/CR—2005-213964



Intelligent Engine Systems

Work Element 1.2: Malfunction and Operator Error Reduction

Matthew Wiseman
General Electric Aircraft Engines, Cincinnati, Ohio

October 2005

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the Lead Center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.

- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at <http://www.sti.nasa.gov>
- E-mail your question via the Internet to help@sti.nasa.gov
- Fax your question to the NASA Access Help Desk at 301-621-0134
- Telephone the NASA Access Help Desk at 301-621-0390
- Write to:
NASA Access Help Desk
NASA Center for AeroSpace Information
7121 Standard Drive
Hanover, MD 21076

NASA/CR—2005-213964



Intelligent Engine Systems

Work Element 1.2: Malfunction and Operator Error Reduction

Matthew Wiseman
General Electric Aircraft Engines, Cincinnati, Ohio

Prepared under Contract NAS3-01135

National Aeronautics and
Space Administration

Glenn Research Center

October 2005

Available from

NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22100

Available electronically at <http://gltrs.grc.nasa.gov>

Contents

1. Executive Summary	1
2. Introduction	1
3. Field Data Analysis	3
4. Real-Time Feasibility Demonstration	6
5. Proposed Architecture	9
6. Conclusion.....	12
7. References	12

Figures

Figure 1. Propulsion 21 Task 1.2 Schedule	2
Figure 2. PS3 Rates of Change	5
Figure 3. EGT Rates of Change	5
Figure 4. N1 Rates of Change	6
Figure 5. N2 Rates of Change	6
Figure 6. Real-Time Demonstration System	8
Figure 7. Real-time Demonstration Setup	9
Figure 8. Integrated Propulsion System PHM Architecture	10
Figure 9. Engine PHM Architecture	11
Figure 10. Integrated PHM System View	11

1. Executive Summary

Jet engines, although highly reliable and safe, do experience malfunctions that cause flight delays, passenger stress, and in some cases, in conjunction with inappropriate crew response, contribute to airplane accidents. On rare occasions, the anomalous engine behavior is not recognized until it is too late for the pilots to do anything to prevent or mitigate the resulting engine malfunction causing in-flight shutdowns (IFSDs), aborted takeoffs (ATOs), or loss of thrust control (LOTC). In some cases, the crew response to a myriad of external stimuli and existing training procedures is the source of the problem mentioned above.

The problem is the reduction of jet engine malfunctions (IFSDs, ATOs, and LOTC) and inappropriate crew response (PSM+ICR) through the use of evolving and advanced technologies.

The solution is to develop the overall system health maintenance architecture, the detection and accommodation technologies, the processes, and the enhanced crew interfaces that would enable a significant reduction in IFSDs, ATOs, and LOTC. This program defines requirements and proposes a preliminary design concept of an architecture that enables the realization of the solution.

This report is presented in two parts. Part 1 is this document, a 20 page summary of the program in report format. Part 2 is a set of annotated presentation charts intended to be presented as part of a preliminary design review. Parts 1 and 2 are distributed electronically as a single file.

2. Introduction

There are many possible problems that can occur during the operation of an aircraft, and these problems have a variety of causes, including operator error, faulty equipment, structural failures, and improper maintenance or repair of the aircraft. Due to the high level of redundancy and safety margin in the design and operation of aircraft, these problems normally do not impact the safe operation of the aircraft. However, on rare occasions, problems do occur that result in an accident.

Some accidents are the result of a single failure, such as a catastrophic structural failure of the airframe or unacceptable ice build-up on a wing; even such single failures are often the result of a number of contributing factors. Frequently, however, an accident is the result of multiple failures, of which any single one might not normally result even in interruption of the flight, much less an accident. It is the compounding effect of multiple problems that results in such an accident.

The propulsion system in an aircraft can malfunction in a number of ways, which may ultimately contribute to a safety issue; but like other flight-critical subsystems, the propulsion system has a high level of redundancy and safety margin. The aircraft system is designed so that a propulsion system malfunction (PSM) has an extremely low likelihood of causing an accident.

It is very important, however, that the crew respond appropriately when a malfunction does occur, whether propulsion-related or otherwise. Failure to react properly, i.e., failure to act in accordance with established operating procedures, is termed inappropriate crew response (ICR) and is a contributing factor in a high percentage of aircraft accidents.

Accidents resulting from a propulsion system malfunction combined with inappropriate crew response (PSM+ICR) account for nearly 1/3 of propulsion-related accidents. A PSM+ICR event is one in which the crew responds inappropriately to an otherwise benign PSM. In such an event, the PSM alone is not severe enough to result in the event, but the event occurs when the PSM is combined with operator error.

The objective of this program is to reduce PSM+ICR related incidents by developing a fault diagnostic system to identify and isolate propulsion system malfunctions, a control methodology to provide fault accommodation where appropriate, and a reliable human/machine interface to provide information to the cockpit.

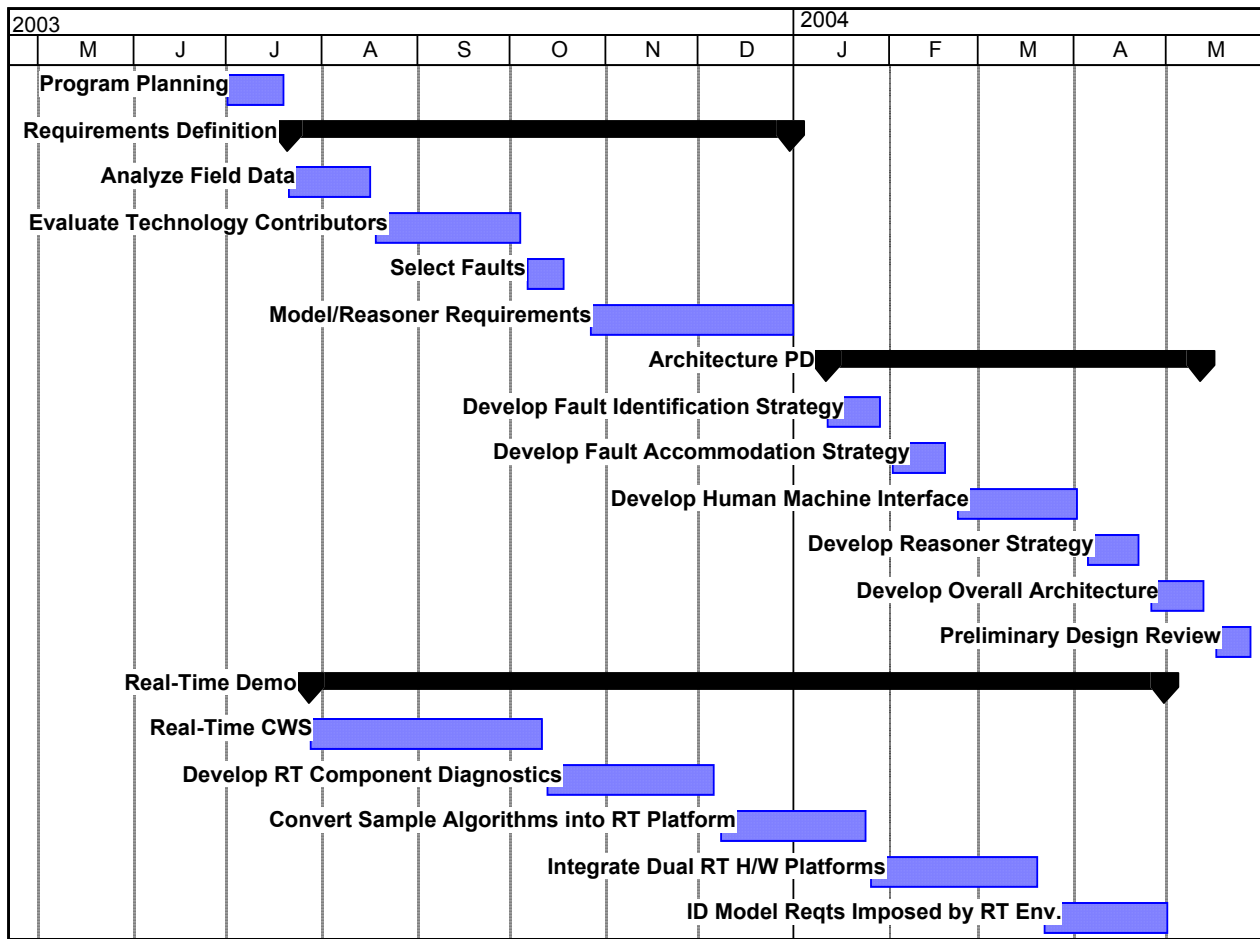


Figure 1.—Propulsion 21 Task 1.2 Schedule.

This program is structured in two phases. Phase 1 is a preliminary design phase, and phase 2 covers detailed design and demonstration. This report covers the phase 1 effort; phase 2 is not funded as of the date of this report. Figure 1 shows a detailed task schedule for phase 1, beginning in July 2003, and concluding in May 2004.

The first task of this program is to identify system requirements. Field event data was analyzed for faults related to in-flight shut down (IFSD), loss-of-thrust-control (LOTC), and aborted-takeoff (ATO) events. All GE commercial engine models for which data was accessible were considered. Other technology programs, such as the NASA-funded Model-based Fault Tolerant Control and Autonomous Propulsion System Technology programs, were surveyed for opportunities to leverage existing and developing technologies. The subsystem diagnostics tasks under Propulsion 21 (Task 1.3) were included in this survey. Human/machine interface (HMI) issues were also investigated, and current state-of-the-art cockpit design was analyzed. Based on the safety-related field events and applicable technologies, a set of faults was selected for use in developing a fault identification, accommodation, and crew interface system. Requirements for gas-path and subsystem models and a reasoner for use in the diagnostic/identification algorithms were specified.

As part of the requirements definition, it was anticipated that real-time model-based diagnostics would be a key component of the final architecture. However, the potential complexity introduced by the diagnostics raises questions about the computational throughput required for such a system. Therefore, under the real-time demo task, studies were performed to address the issue real-time implementation of a complex model-based diagnostic system.

The second task is architecture preliminary design. Under this task, preliminary design activities for the overall system architecture will be conducted. These activities include defining the fault identification and accommodation strategies, as well the human/machine interface. Recommendations for improved crew interface characteristics were solicited from experts in the field, leading to a crew advisory strategy, and a reasoner approach was defined. Finally, an overall system architecture was defined that builds on the strategies defined earlier in the program.

Phase 1 originally included plans for a preliminary design review. This review was to cover the overall architecture and subcomponents of the proposed system. However, this task was removed from the contract late in the program.

3. Field Data Analysis

Field event records were analyzed to highlight potential opportunities for preventing PSM+ICR events. GE maintains a field event and maintenance database called Smart 2000. This database contains narrative on all field events involving GE engines.

In-flight shut down (IFSD), loss-of-thrust-control (LOTC), and aborted-takeoff (ATO) events were extracted from Smart 2000 and analyzed for indications of PCM+ICR. Our objective is to reduce incidents resulting from PSM+ICR. While not all PSM+ICR events result in an accident or incident, the approach taken is to reduce the likelihood of PSM+ICR occurring, thus reducing the opportunities for a resulting accident to occur.

An in-flight shutdown is an event in which the pilot responds to a perceived engine malfunction by shutting an engine down. Specific procedures are defined for dealing with engine malfunctions; depending on the symptoms and the flight situation, it may be appropriate to retard the throttle then re-advance to the desired power setting, shut down the engine, do nothing immediately, or take a variety of other actions. Shutting down the engine when not called for by the operating procedure is classified as an ICR.

Loss of thrust control is an event in which the engine begins producing a level of thrust that is not in line with the thrust demanded by the throttle setting. This includes overspeed, stall, and uncommanded accels or decels. If it cannot be corrected, LOTC may necessitate IFSD.

An aborted takeoff is an event in which the pilot decides to stop the airplane after the takeoff roll has begun but before the airplane has left the ground. However, there is a point at which it is safer to continue the takeoff, go around, and land, than to attempt to stop the airplane immediately. This point is defined as being when the airplane reaches a speed known as “V1.” Federal Aviation Regulations (FAR) Part 1 provides the following definitions of V1:

1. “V1 means the maximum speed in the takeoff at which the pilot must take the first action (e.g., apply brakes, reduce thrust, deploy speed brakes) to stop the airplane within the accelerate-stop distance; [and,]
2. “V1 also means the minimum speed in the takeoff, following a failure of the critical engine at V_{EF} , at which the pilot can continue, the takeoff and achieve the required height above the takeoff surface within the takeoff distance.”

“Critical engine” is defined as “the engine whose failure would most adversely affect the performance or handling qualities of an aircraft,” and V_{EF} is defined as “the speed at which the critical engine is assumed to fail during takeoff.”

Previous definitions of V1 did not state clearly that V1 is the maximum speed at which the pilot must take the first action to reject the takeoff. This means that if an airplane has exceeded V1, it is inappropriate for a pilot to attempt to abort the takeoff. Doing so can result in the airplane running off the end of the runway with sometimes fatal consequences.

The Smart 2000 data over the period 1972 to 2002 was analyzed for root cause. This analysis, together with information from other documents, was used to formulate a list of PSMs to address under this contract. The list is as follows:

1. Surge/Stall
2. Power Loss
3. Reverser Inadvertent Deploy
4. Thrust Asymmetry
5. Fixed Thrust Failure
6. Severe Component Damage
7. Seizure
8. Fuel Leak
9. Fire
10. Severe Engine Vibration

The malfunction labeled Severe Component Damage refers to the traditional engine gas path faults that have been examined in past programs such as IMATE, and IEPHM, and would include FOD and bird ingestion.

The top two PSMs are often associated with startling noise (such as a large bang) that can induce an ICR due to the startling nature of the event. One description found in the references describes the bang associated with engine surge “like a bomb going off”. When someone is properly trained to recognize a startling event, it may be possible to ignore the event and concentrate on flying the plane; however, current training methods do not incorporate realistic conditions, so we do not anticipate a reduction in PSM+ICRs due to these causes unless the propulsion system is properly involved or training techniques are improved.

Based on these results, and keeping to the preliminary design nature of this program, a single root cause was selected for additional analysis. The selected cause is compressor stall. In order to understand better compressor stall, engine test data was gathered from stall tests and other events, such as power interrupts and bird ingestions. The rate of change of certain parameters is compared to a percent of the last value of that parameter. Therefore, the test data was analyzed in the same way. HP and LP speeds, EGT, T25, PS3, and fuel flow were examined. Four of these proved to be good indicators of stall: PS3, EGT, N1, and N2. Stall-indicating limits on the rates of change of these parameters were determined by examining both stall and non-stall events. For each event examined, the rate of change is plotted in the figures 2 to 5. Note that “S” denotes a stall event, and “N” a non-stall event.

Using a threshold of -250 percent, as shown in figure 2 for PS3, would allow correct classification of these all events as either stall or non-stall. However, it can be seen that there are a few near-misses.

A threshold of +10 percent is shown for EGT. Note that nine events would be improperly categorized if EGT rate were the only stall indicator. The threshold shown for N1 is -20 percent. Note that eight events would be improperly categorized if N1 rate were the only stall indicator. The threshold shown for N2 is -5 percent. Seven events would be improperly categorized if N2 rate were the only stall indicator.

An objective of this PSM+ICR study is to improve the information received by the pilots to aid in making decisions. Informing the pilots, in a proper manner that a stall has occurred could lead to a reduction in PSM+ICR accidents. From the analysis, PS3 seems to be a good indicator of stall. However, such an indicator must be extremely accurate, minimizing both missed stalls and false alerts. For this reason, a multi-parameter based stall detection algorithm is desirable. The figures 2 to 5 show that for every stall event, PS3 and at least one of the other three indicators meets the detection criteria. These criteria could form the basis of improved stall detection capability.

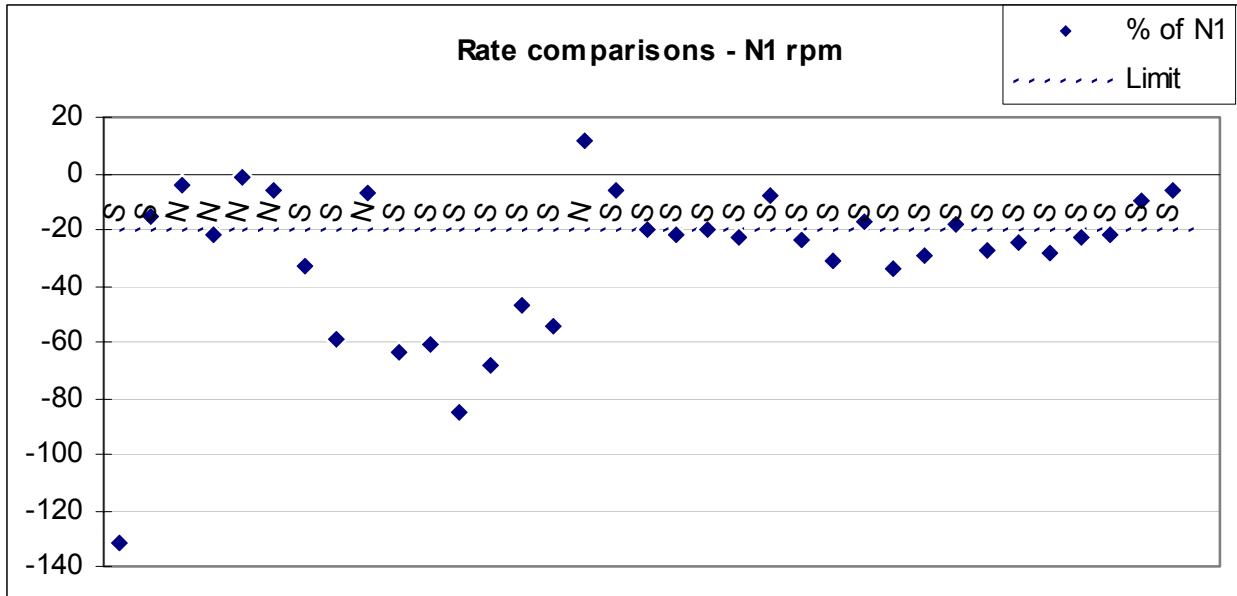


Figure 4.—N1 Rates of Change.

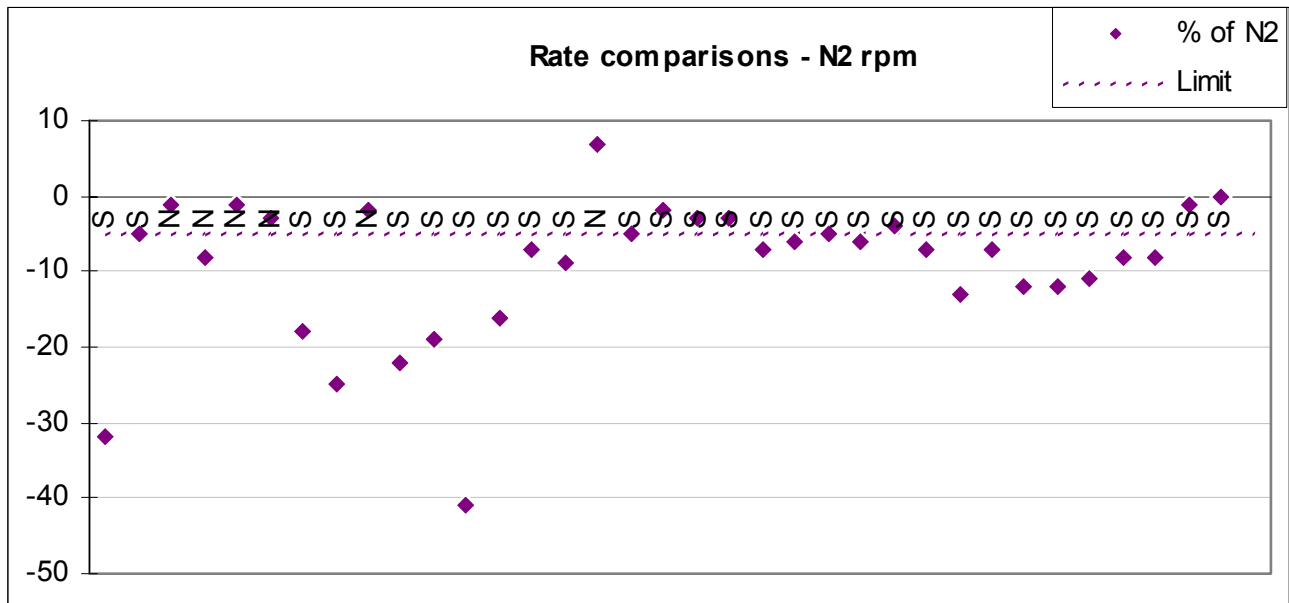


Figure 5.—N2 Rates of Change.

4. Real-Time Feasibility Demonstration

The real-time demonstration effort began to test the feasibility and requirements necessary to develop a functional on-line diagnostic system. A real-time diagnostic system of this type for an aircraft engine is novel, thus there were many questions regarding both the hardware and software requirements of such a system. Knowing beforehand that the algorithms to be implemented are extremely computationally intensive, it was not clear if it was feasible to develop a real-time system using current technology.

Development of the real-time system began in MATLAB and Simulink. This is the environment in which the diagnostic algorithms were originally developed. The diagnostic routines consist of proprietary algorithms originally developed for use with military engine data. These algorithms were implemented in MEX files, using MATLAB's own programming language. This allowed for rapid development and testing of the algorithms but, unfortunately, this code cannot be used in a real-time environment.

The original algorithms consisted primarily of an Extended Kalman Filter (EKF), a tracking filter, and fault detection algorithms. The EKF functions as a state estimator; it uses engine's sensors, such as fan speed, core speed, temperatures and pressures at different stages to determine the current state of the engine. The EKF is an extremely computationally intensive algorithm because of the need to linearize the non-linear engine model. It must linearize the model over the 20 states that are considered in the EKF. This linearization requires a model call for each state, which means 20 model calls per linearization. Given the complexity of the engine model, this linearization process takes a considerable amount of time. The output of the EKF is a set of innovations, or residuals. These residuals are the differences between the state estimation of the model and the actual engine it is modeling. Ideally, these innovations go to zero, but due to noise and engine-to-model mismatch, the innovations often hover around but not exactly at zero. These innovations can then be fed into the fault detection algorithms. By analyzing the differences between the actual engine and the state estimation of a nominal engine, the algorithm can identify trends that predict or uncover faults in the actual engine.

The tracking filter helps to aid the EKF and fault detection algorithms. It also utilizes an EKF, however, the tracking filter's EKF serves a different purpose. Instead of acting as a state estimator, the tracking filter attempts to reduce the mismatch between the actual engine and the engine model. This, in turn, further reduces the innovations from the EKF and promotes the identification of smaller engine faults.

The first step to achieving a real-time demonstration was to choose a real-time environment. Mathworks, the creators of MATLAB and Simulink developed an extension to Simulink called Real-Time Workshop. Real-Time Workshop enables the generation of optimized, portable, and customizable ANSI C source code from Simulink models. An add-on to Real-Time Workshop targets the code generation for xPC, a Mathworks proprietary real-time operating system that is compatible with standard x86 PC hardware. The xPC operating system is installed on a 3.5 in. floppy disk. This disk can then be inserted into any standard PC that contains a network card supported by xPC. During the PC boot-up process, the xPC operating system is loaded from the floppy disk into memory. This results in xPC having total control over the PC and its resources. This allows any program run in the xPC environment exclusive access to the computer's resources. Additionally, the xPC operating system has strict timing constraints and very little overhead, enabling real-time operation to the program loaded into memory. A program can be loaded into memory from a host computer via the local area network and the network card in the xPC system. This same network card can be utilized to transfer data between the xPC machine and the host computer or even other xPC systems.

Since the algorithms were originally written in MATLAB's propriety language, they first had to be rewritten in C in the form of a Type 2 S-function. The Type 2 S-Function is a standard format for the C code that provides an interface to Simulink. This undertaking involved rewriting the Extended Kalman Filter (EKF), the fault diagnostics as well as many matrix math algorithms. Also, since the EKF is heavily reliant on the engine model, an interface to the model was necessary. The engine model is written in FORTRAN, so a driver was created as a Type 2 C S-Function which allowed the model to be run from Simulink. A tracking filter was also included in the initial software suite of tools implemented in MATLAB's MEX code; however, since the tracking filter is not continuously run like the EKF and diagnostic routines, it was not converted to run in real-time. Instead of running continuously, the tracking filter can be run in a batch mode, meaning it can be run on stored engine data during aircraft downtime. By not implementing the tracking filter in real-time, it relieves an unnecessary computational burden.

The real-time demonstration consists of two PCs running the xPC operating system. The first xPC system is set up to run a model of the aircraft engine. Its purpose is to simulate an engine "on-wing". This system simulates the engine itself, as well as the FADEC, which is the engine controller. The engine

model is a physics based model operating in real-time. This model is set up to accept commands to “inject” particular types of engine faults. It supports the simulation of faults of the following types: Foreign Object Damage, High Pressure Turbine damage, VSV failures, EGT sensor failure and CDP sensor failure.

The second xPC system consists of the EKF and fault detection algorithms. Both systems are connected via a local area network. Sensor information from the engine model is sent via the network to the EKF system. A diagram of the setup is shown in figure 6. Host PCs are also attached to the network to monitor and control the xPC systems. The host PCs are used only for a user interface and control of the xPC systems; no computation is performed on them.

Faults can be injected into the engine model via the user interface on the host PC. When faults are injected into the engine model, the engine’s sensors are affected accordingly. These sensor outputs are sent to the xPC system running the EKF and fault detection code. At this point, the residuals from the EKF will reflect a significant difference between the nominal engine model outputs and the actual engine outputs. This difference, when analyzed by the fault detection code matches a signature that indicates the type of fault encountered. A picture of the demo in action is shown in figure 7.

The real-time demonstration development effort was a success. In real-time, a fault can be detected in an engine, even through engine transients. However, the requirements necessary on the xPC system running the EKF were extremely high. As previously mentioned, the linearization requires 20 model calls. Each model call takes roughly 3 percent of the CPU per time step. This brings the computational requirement of the linearization to roughly 60 percent of the CPU, or 75 percent of the EKF and fault detection’s total computation time. This method was known to be computationally inefficient but was implemented because of the flexibility it provides along with its ease of implementation. The EKF system ran on an Intel Xeon processor operating at 3.06 GHz. Using this hardware, the EKF and diagnostic routines utilized 81 percent of the system’s processor. This is an enormous requirement and as of the time of writing, as expected, is unfeasible for implementation in the field. Because of this, several ideas were considered to reduce the computational burden. The standard way of solving this problem involves gain scheduling the Kalman gains or gain scheduling the Jacobian matrix in the EKF. The idea behind gain scheduling is that the gains inside the EKF are pre-computed and functionalized. The gains are the results of the costly linearization, and thus if the gains are pre-computed and scheduled the linearization can be avoided. By removing the linearization, a drastic decrease in the computational requirements is achieved while negligibly increasing the memory requirements of the system. It is estimated by changing to a gain scheduling scheme, the computational time of the EKF and fault detection algorithms would be reduced to 20 percent utilization of the processor. Other standard optimizations of the EKF can be implemented, which from previous experience would likely further reduce the computational requirements by a factor of 2. Preliminary research indicates the gain scheduling induces an initial error in the routines, which after a short amount of time (under a minute) converges to zero. However, further research is necessary to confirm the effectiveness of this approach.

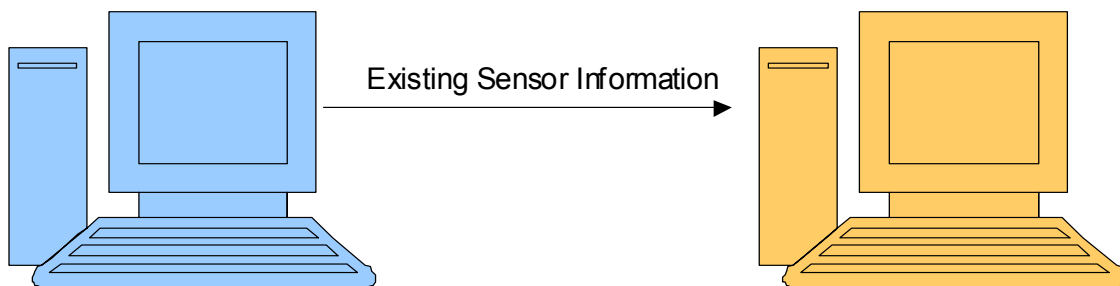


Figure 6.—Real-Time Demonstration System.

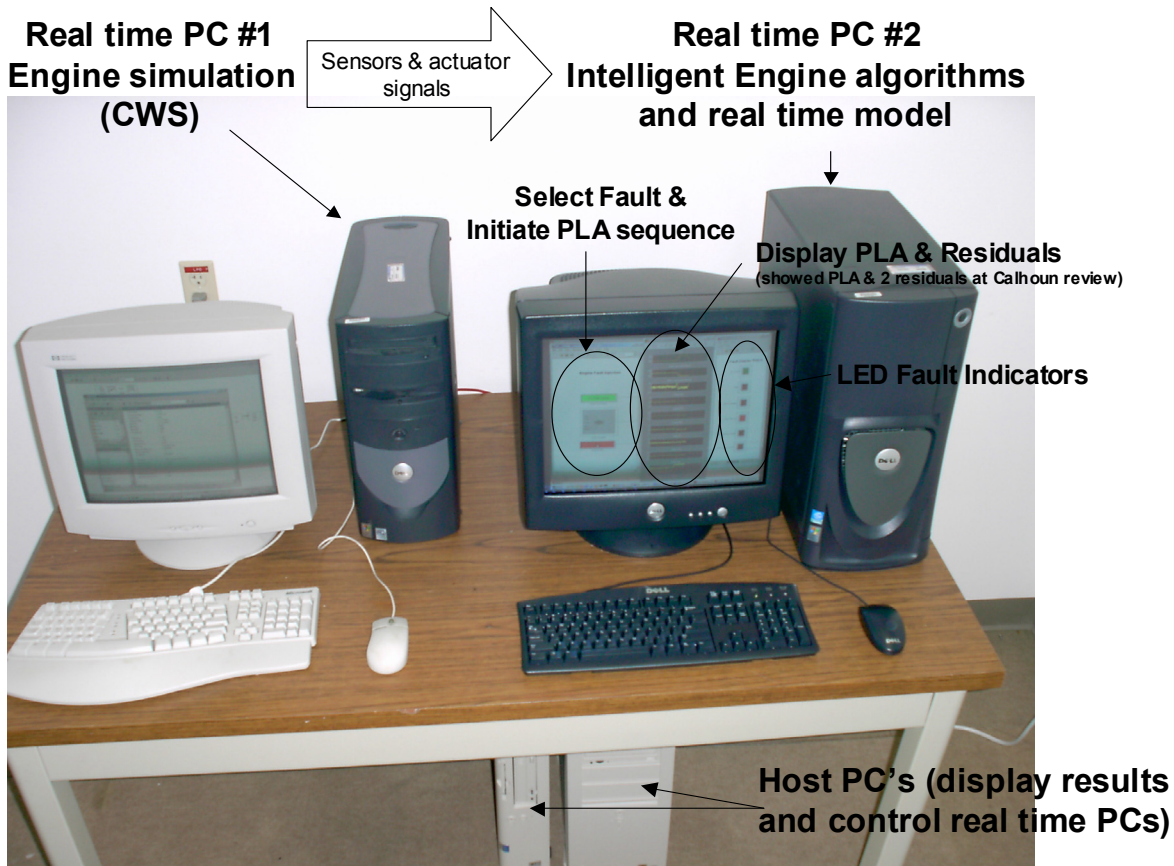


Figure 7.—Real-time Demonstration Setup.

5. Proposed Architecture

Having identified the PSM list, a system architecture must be defined that will address the solution to the problem. In order to facilitate the architecture development, various methods were proposed that would be supportive in solving the problem. The key architectural features that emerged include:

- the propulsion system (vs. engine system)
- state tracking
- prognostics
- PSM prevention
- PSM control

This list drove the creation of a number of key questions that, when answered properly, will provide the required insight into the system architecture. A key concept to understand is that of “true” prognostics - the ability to predict an impending situation prior to its full manifestation. If an imminent PSM can be predicted and prevented from occurring, then by definition, a PSM+ICR has been eliminated.

Figure 8 shows the propulsion system (PS) level architecture. The Integrated PS PHM software architecture combines the control and status information from all engines on the aircraft. Incidents associated with In-Flight Shutdowns (IFSDs) are mitigated by checking Ignition shutdown status against the aircraft state. At a minimum, a crew advisory would alert the crew to the un-safe situation. Providing timely propulsion health status through achievement of V1 mitigates incidents with Aborted Takeoffs.

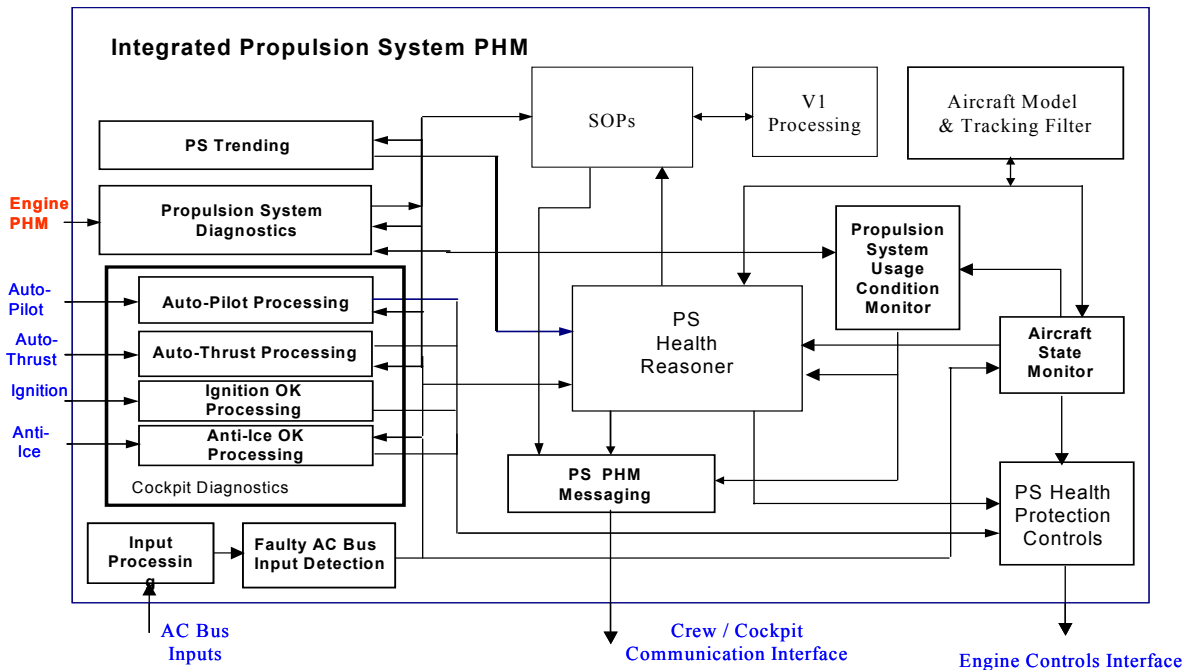


Figure 8.—Integrated Propulsion System PHM Architecture.

Incidents associated with Loss of Thrust Control are mitigated by monitoring the Aircraft state (position, velocity, thrust, attitude) and issuing PS Health Protection controls to compensate for asymmetric thrust; monitoring the auto-pilot and auto-thrust inputs; anti-ice turn-on/shut-off during states of potential stall; and causing a gradual transition from auto-pilot to manual pilot control. When a severe engine fault occurs, the status of the PS as well as the SOPs are updated rapidly so that crew situational awareness or information processing errors do not occur.

The Engine PHM software architecture shown in figure 9 is similar in structure to the Integrated PS PHM architecture. The EPHM differs from the IPSPHM in that:

- the EPHM doesn't include any processing associated with SOPs and V1,
- the real-time, online model is that of the specific engine vs. the aircraft,
- specific engine PHM sensors for vibration, the inlet, the compressor, and the combustor are used to enhance prognostics and detection of severe vibration, foreign object ingestion, compressor stall/surge, and combustor flameout, respectively;
- high bandwidth protection controls are used.

The high bandwidth protection controls (the compressor and combustor) will likely be located in the FADEC for that engine since the delays associated with external processing may degrade control effectiveness.

The health status of the engine is transmitted to the IPSPHM at a periodic rate (~1-2 Hz) in order that an engine advisory is accomplished in a timely fashion.

Finally, figure 10 illustrates how the Engine PHM feeds into the IPS PHM. The IPS PHM needs to know on which side of the aircraft the EPHM information is related to. Decisions and control are partitioned to the lowest appropriate level of responsibility. Impacts to the FADEC concern that of communication bandwidth, and high bandwidth signal and controls processing associated with specific engine subsystems (the compressor and combustor).

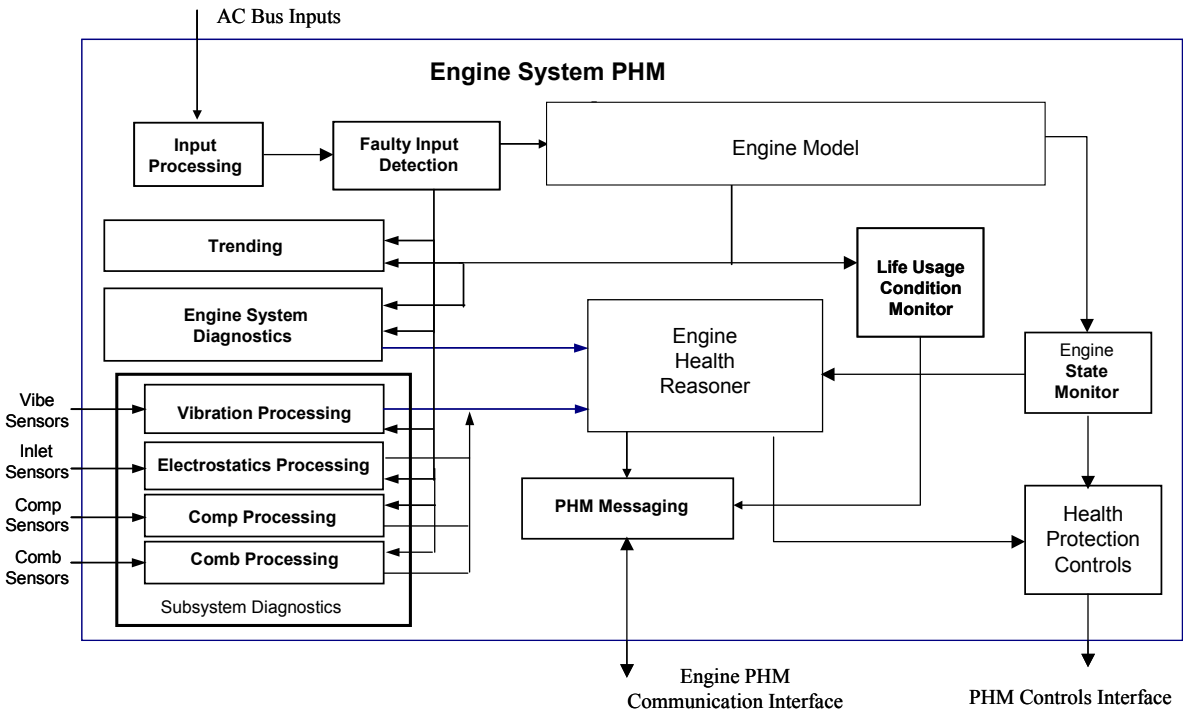


Figure 9.—Engine PHM Architecture.

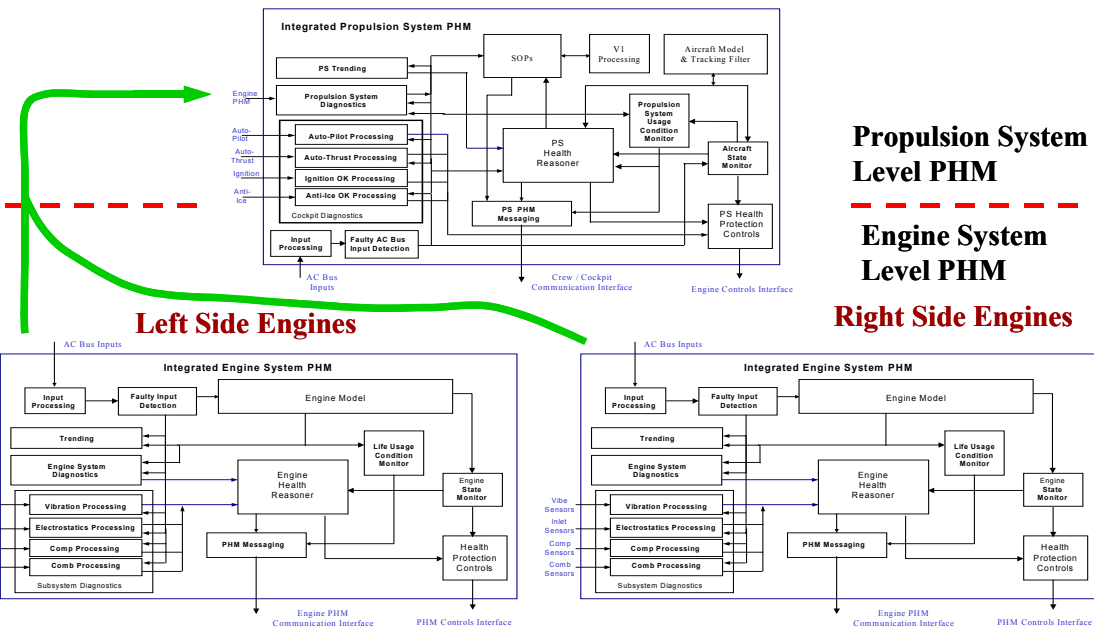


Figure 10.—Integrated PHM System View.

6. Conclusion

Valuable progress has been made in defining requirements and creating an architecture for reducing PSM+ICR events in the commercial aviation industry. A list of key PSMs has been formulated based on the analysis of field events from various sources, and other technology programs have been considered for applicability to the objective. Analyses of PSMs and their root causes drove requirements for the system, leading to the definition of a layered prognostics and health management architecture that addresses the needs of the individual engine, the propulsion system, and the flight deck.

However, significant work remains to be done in this area. In part because of the broad scope of this subject, some areas of the architecture need further development. Additional contact with airframers, airlines, and government agencies is needed to help us better understand some of the crew interface issues, which should lead to additional refinement of the requirements, and thus to a more robust and detailed architecture.

References

1. AIA/AECMA Project Report on Propulsion System Malfunction Plus Inappropriate Crew Response (PSM+ICR); Volume 1; 1 November 1998; G.P. Sallee, AIA Co-Chair; D.M. Gibbons, AECMA Co-Chair.
2. GE Smart 2000 Database.
3. Context of Human Error in Commercial Aviation; December 31, 2001.
4. Intelligent Maintenance Advisor for Turbine Engines (IMATE); Agreement No. MDA 972-98-3-0002; Final Report; Compiled by Malcolm J. Ashby; General Electric Aircraft Engines; August 15, 2000.
5. Integrated Engine Prognostics and Health Management (IEPHM); Agreement No. N00421-01-2-0047; Final Report; Compiled by Malcolm J. Ashby; General Electric Transportation; January, 2004.

REPORT DOCUMENTATION PAGE*Form Approved*
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE October 2005	3. REPORT TYPE AND DATES COVERED Final Contractor Report	
4. TITLE AND SUBTITLE Intelligent Engine Systems Work Element 1.2: Malfunction and Operator Error Reduction			5. FUNDING NUMBERS WBS-22-714-92-50 NAS3-01135	
6. AUTHOR(S) Matthew Wiseman				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) General Electric Aircraft Engines One Neumann Way Cincinnati, Ohio 45215-1915			8. PERFORMING ORGANIZATION REPORT NUMBER E-15280	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001			10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA CR-2005-213964	
11. SUPPLEMENTARY NOTES Project Manager, Clayton L. Meyers, Aeronautics Division, NASA Glenn Research Center, organization code PRV, 216-433-3882.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Category: 07 Available electronically at http://gltrs.grc.nasa.gov This publication is available from the NASA Center for AeroSpace Information, 301-621-0390.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) Jet engines, although highly reliable and safe, do experience malfunctions that cause flight delays, passenger stress, and in some cases, in conjunction with inappropriate crew response, contribute to airplane accidents. On rare occasions, the anomalous engine behavior is not recognized until it is too late for the pilots to do anything to prevent or mitigate the resulting engine malfunction causing in-flight shutdowns (IFSDs), aborted takeoffs (ATOs), or loss of thrust control (LOTC). In some cases, the crew response to a myriad of external stimuli and existing training procedures is the source of the problem mentioned above. The problem is the reduction of jet engine malfunctions (IFSDs, ATOs, and LOTC) and inappropriate crew response (PSM+ICR) through the use of evolving and advanced technologies. The solution is to develop the overall system health maintenance architecture, detection and accommodation technologies, processes, and enhanced crew interfaces that would enable a significant reduction in IFSDs, ATOs, and LOTC. This program defines requirements and proposes a preliminary design concept of an architecture that enables the realization of the solution.				
14. SUBJECT TERMS Propulsion systems (aircraft)			15. NUMBER OF PAGES 20	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT	

