Proceedings of GT2007
ASME Turbo Expo 2007: Power for Land, Sea and Air
May 14-17, 2007, Montreal, Canada

GT2007-27862

REAL-TIME HEALTH ESTIMATION AND AUTOMATED FAULT ACCOMMODATION FOR PROPULSION SYSTEMS

Liang Tang, Michael J. Roemer and Gregory J. Kacprzynski
Impact Technologies, LLC, 200 Canal View Blvd. Rochester, NY 14623
E-mail: {Liang.Tang, Mike.Roemer, Greg.Kacprzynski}@impact-tek.com

ABSTRACT

This paper presents the development of innovative real-time health estimation and automated fault accommodation techniques for advanced propulsion systems within a Dynamic Decision Support (D^2S) framework. The proposed approach uses dynamic models in a real-time computing environment to not only diagnose system degradation and faults, but also to determine "on the fly" how to accommodate for them. The real-time health estimation modules enhance on-board PHM (Prognosis & Health Management) capabilities with a dynamic system identification algorithm that is capable of detecting faults with a continuously updated dynamic model. In addition, a real-time, self-tuning Kalman filter and fault classification algorithm are combined to provide accurate health estimation. Based on the inferred health condition, mission requirements and flight regime information, the automated fault accommodation module automatically makes decisions regarding control reconfiguration and change of control strategies. The presented techniques have been applied to a generic turbofan engine model with simulated engine component faults and degradation and simulation results are presented. To further raise the technological readiness level, select algorithms have been implemented and evaluated on a PC104 embedded platform. The dynamic modeling capabilities, techniques and tools sets will not only improve the reliability of the propulsion systems, but also greatly enhance maintenance decision support and contingency planning concepts.

INTRODUCTION

Impact Technologies, LLC has developed a novel architecture and a suite of tools for real-time health estimation and automated fault accommodation within a Dynamic Decision Support (D^2S) framework for aircraft propulsion systems. The goal of this research was to create an aircraft-embeddable software system to accommodate propulsion system faults and manage system health and life through the fusion of PHM and intelligent control technologies. The envisioned product is a real-time, modular, hierarchical software system that can be implemented on an embedded platform and integrated with the Full Authority Digital Engine Controller (FADEC) and off-wing avionics hardware to provide real-time fault accommodation functionality to propulsion system components.

The research in health monitoring and fault accommodating control of propulsion systems has in recent years provided a variety of promising approaches to improving the reliability and performance of the propulsion systems [1-5]. Health monitoring and diagnosis of engine performance through engine parameter estimation has been reported in a wide range of works [4,6]. These processes estimate engine performance variables mostly using linearized models and some enhanced with neural networks to compensate for nonlinearity and modeling errors [7]. Romessis and et al. [8] and Ganguli [9] used classifiers to accomplish diagnostic functions. The health estimation algorithms developed here highlights the real-time system identification of the engine dynamic model and the combination of Kalman Filtering and classification methods for accurate and robust health estimation. Unlike the work done in [10], our work uses a Stochastic Pattern Classifier fused with a Probabilistic Neural Network based classifier, and does not require a-priori conditional probability measures as needed for Bayesian Belief Network. The Kalman Filtering algorithm used here is similar to the work done in [7]. However, in this work, the health parameter estimates from the Kalman Filter are not directly used due to that fact that their estimates have essentially become a mathematical artifact for achieving a better match between observation and model output [11]. Instead, the accurate state estimates are fed into fault classifiers to produce a robust health estimation. The automated fault accommodation concepts applied here incorporate an adaptive thrust controller and a damage protecting controller that is similar to the work in [12]. Innovative ways to implement automated fault accommodation are presented including the use of the Stateflow™ toolbox and predicates-based reasoning.

The work reported here establishes a real-time health estimation and automated fault accommodation framework and a suite of efficient algorithms that are embeddable in current COTS hardware platforms. The remainder of this paper is organized as follows. First, an overview of the system architecture is presented. Real-time health estimation modules including a recursive system identification algorithm, self-tuning Kalman filter, and fault classifiers are described in details in the following section. Automated fault accommodation modules and predicates based reasoning concept are explained and the presented techniques are applied to a generic turbofan engine model to detect faults resulting in efficiency losses in the engine high-pressure compressor (HPC) and/or combustor. Simulation results are presented. To raise the technological readiness level of the presented system, select algorithms have been implemented and tested on a PC104 embedded platform and real-time performance was evaluated. This paper concludes with remarks on future work.

SYSTEM OVERVIEW

Jet engine performance varies from engine to engine due to manufacturing tolerances, aging, and deterioration caused by use. Generally the control system developed for the engine is robust enough to keep it operating within acceptable boundaries for several thousand flight cycles if there were no major faults present. However, as the engine ages or faults occur, the baseline engine controller may no longer assure optimal engine operation. The goal of the proposed engine D^2S system is to increase the level of autonomy of the engine operation by bridging the gap between onboard PHM system and baseline control systems. As shown in Figure 1, engine D^2S system consists of two major functions: 1) real-time assessment of engine system health (PHM); 2) Automated Fault Contingency Planning (AFCP).

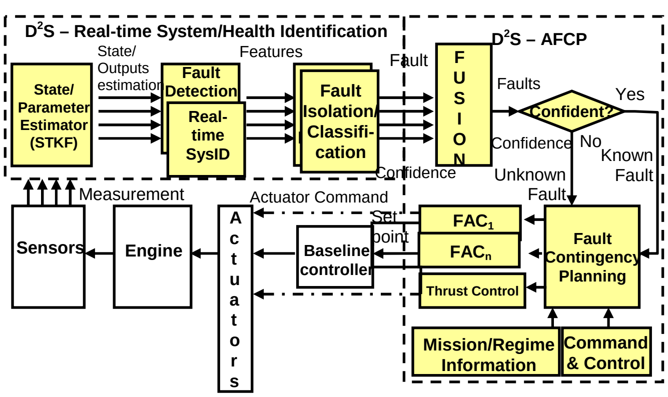


Figure 1: D^2S System for Jet Engines

The PHM modules are enhanced with a set of innovative supporting tools including dynamic modeling and simulation capabilities and diagnostic technologies that are capable of detecting fault and isolating failures in a dynamic system environment. A Self Tuning Kalman Filter (STKF) is utilized to provide accurate engine state estimation even in the presence of component faults and degradation. The health parameter estimated from STKF can be used as a fault indicator. The state estimates and features calculated from fault detection routines are feed into fault classifiers for fault isolation. Information fusion techniques are applied when there are multiple fault classifiers engaged.

Based on health condition, mission requirements and flight regime information, the onboard AFCP module automatically makes decisions regarding control reconfiguration and change of control strategies. Usually when a fault can be confidently classified as a known fault pattern, predetermined optimal scheduling and control logics obtained from offline experiments, if available, can be applied to accommodate the fault. However, when an unknown fault (i.e., a fault that can't be confidently classified as any known fault pattern) is encountered, in the absence of predetermined fault accommodating rules, the objective of achieving required thrust level to meet mission requirement and meanwhile minimizing damage to the engine can possibly be obtained by using adaptive control laws such as thrust control. Since thrust is not directly measurable, a thrust virtual sensor based on STKF was developed.

Depending on the control strategy, the AFCP module may issue direct actuator commands or utilize an outer loop controller to adjust the reference (N1) commands sent to the baseline engine controller (the FADEC). A benefit of using a outer loop controller is that, since most baseline FADEC logic has been verified and validated (V&V), the change of reference fan speed command, which controls the engine via the FADEC, will not drive the engine into an unsafe operational regime.

The ultimate goal of D^2S system is to enable an aircraft propulsion system to operate in an intelligent way featuring self-diagnostic, self-prognostic, self-optimizing, and mission adaptable. At the control level, new engine control techniques that can potentially be leveraged include Life Extending Control [13], Performance Seeking Control [14], and Model Predictive Control [15].

REAL-TIME HEALTH ESTIMATION

To adapt to changing environment, fault conditions and performance degradation, it's very important that the proposed D^2S system possesses the capability to identify/update various models in real-time. Based on the work of Bodson and et al [16,17], real-time recursive system ID algorithms have been developed and demonstrated on an embedded platform.

Real-time Recursive System Identification Algorithm

State space model or equation of motions for a plant can be stated (in discrete format) as,

$$y(n) = \theta(n)^T \phi(n) + v(n) \qquad (1)$$

Depending on the format of the model and different application scenarios, *y(n)* may either be a system output or the derivative of a system state variable, and y(n) is a scalar; When applied to identify a state-space model, the state variables can be obtained from the STKF and derivatives of state variables can be calculated numerically. Each column of $\theta(n)$ contains the parameters corresponding to one component of the vector y(n). $\phi(n)$ is the regressor vector, e.g. $[x^T ; u^T]^T$, the state variables and control inputs. The least square algorithm drives $\theta(n)$ towards its true value, θ^* by a modified least squares algorithm that seeks to minimize an augmented cost function,

$$J(\theta(n)) = \frac{1}{2}\sum_{k=n-N+1}^{n} \lambda^{n-k} \| y(k) - \theta(n)^T \phi(k) \|^2 + \alpha \| \theta(n) - \theta(n-1) \|^2 \quad (2)$$

Where λ is the time-varying forgetting factor, which will be modified adaptively according to an anomaly detection module. α is the weighting coefficient that adjusts the influence of the derivation of the current estimate from the previous estimate. N is the length of the data window used for regression. The parameter estimate can be obtained by

$$\theta(n) = \theta(n-1) + P(n)\phi(n)[y(n) - \phi(n)^T \theta(n-1)] + \alpha\lambda P(n)[\theta(n-1) - \theta(n-2)] \quad (3)$$

where the covariance matrix P(n) is calculated by

$$P(n) = (1/\lambda)P(n-1) - (1/\lambda)P(n-1)C(n)[\lambda I + C(n)^T P(n-1)C(n)]^{-1} C(n)^T P(n-1) \quad (4)$$

where the matrix C(n) is given by,

$$C(n) = [\phi(n), \sqrt{n_p \alpha (1-\lambda)} \cdot e(n)] \quad (5)$$

and e(n) is a sequence of vectors,

$$e(1) = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix},\ e(2) = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix},\ e(n_p) = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix},\ e(n_p+1) = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad (6)$$

n_p is the number of parameters in θ and the number of variables in ϕ. ***C(n)*** is a $n_p \times 2$ vector and ***e(n)*** is a $n_p \times 1$ vector. Only a 2 x 2 matrix inverse is required. The covariance matrix is known to be symmetric, and it is helpful to compute the upper triangular part of the matrix only and to update the lower triangular part by symmetry (or vice versa).

The augmented cost function includes one additional terms that restricts the movement of the estimate, $\theta(n)$, in the temporal dimension. Similarly, another term can be added to restrict the movement of $\theta(n)$ in the spatial dimension, e.g. $\beta \| \theta(n) - \hat{\theta} \|^2$, where $\hat{\theta}$ is an a priori estimation of $\theta(n)$. This formulation will further help when $\theta(n)$ doesn't deviate too much from $\hat{\theta}$. However, at the same time, extra computational burden is introduced.

As an improvement for fast detection and tracking in real-time, an anomaly detection scheme and a time varying forgetting factor have been integrated. When a change in prediction residuals is detected, a smaller forgetting factor is utilized to discount 'old' data and place more weight on the latest measurements.

It's worth pointing out that the identified system is the dynamics of the degraded/faulty system. Since health parameters are not included as state variables, the influence of health degradation/fault is attributed to θ. In case of a state-space model identification (e.g. $\dot{x} = Ax + Bu$), engine component degradation/fault usually changes the A matrix, while actuator fault usually affects the B matrix, which can be directly used for actuator fault diagnosis. However, fault diagnosis based on A matrix changes is not straightforward. The identified degraded system dynamics can be utilized by some advanced controls, such as model predictive control to implement adaptive fault accommodation strategies.

Self Tuning Kalman Filter (STKF)

In order to make the on-board engine model adaptive to the real engine's performance variations due to degradation or anomalies, the STKF is designed with the ability to adjust its performance through the adjustment of health parameters that are added to the system dynamics as "virtual state", which also function as tuning parameters. The tuning parameters are embedded in the Kalman filter design. If sensor outputs deviate from nominal condition values due to component degradation and/or faults, the Kalman filter will attribute the cause of sensor output deviations to the tuning parameters, so that the residuals of state/output estimates will remain small. With this design approach, the STKF can maintain accurate estimation performance when it is applied to aircraft engines at off-nominal conditions.

An aircraft engine can be modeled as a nonlinear dynamical system as follows:

$$\begin{aligned} \dot{X} &= f(X,U) + Gv \\ Y_m &= g_m(X,U) + w \\ Y_u &= g_u(X,U) \end{aligned} \quad (7)$$

where, X: state vector, U: control vector, Y_m: measurable output vector, Y_u: unmeasurable output vector, v: system noise vector, w: measurement noise vector, and G: system noise transfer matrix. By linearizing Eq. (7) at a specific operating point, a linearized engine model is obtained as follows:

$$\begin{aligned} \dot{X} &= AX + BU + Gv \\ Y_m &= CX + DU + w \\ Y_u &= C_u X + D_u U \end{aligned} \quad (8)$$

Assuming noise covariance matrices and mean values:

$$E(vv^T) = Q, \quad E(ww^T) = R, \\ E(v) = 0, \quad E(w) = 0 \quad (9)$$

Maximum likelihood estimates of state, measurable and unmeasurable output vector, $\hat{X}, \hat{Y}_m$ and $\hat{Y}_u$, are given by,

$$\begin{aligned} \dot{\hat{X}} &= A\hat{X} + BU + K(Y_m - C\hat{X} - DU) \\ \hat{Y}_m &= C\hat{X} + DU \\ \hat{Y}_u &= C_u\hat{X} + D_uU \end{aligned} \qquad (10)$$

where,

$$K = PC^T R^{-1} \qquad (11)$$

K is the linear Kalman filter gain matrix. P is the steady-state solution of the Riccati differential equation associated to the covariance matrix of the estimates and can be calculated from

$$AP + PA^T + GQG^T - PC^T R^{-1} CP = 0 \qquad (12)$$

Worth noting is that for STKF, the state vector X is augmented with virtual state variables called health parameters, or tuning parameters. For example, for a gas turbine engine example, define,

$$X_c = [\eta_{comp,} w_{comp}, \eta_{lpt}, w_{lpt}, \eta_{hpt}, w_{hpt}, \eta_{comb}, w_{comb}] \qquad (13)$$

Where, *Xc* is the augmented virtual states including compressor, low and high pressure turbine, and combustor efficiency and flow capability. Since *Xc* is artificial state vector which can be estimated by the Kalman filter, state equation can be defined as,

$$\dot{X}_c = 0 \qquad (14)$$

Or in discrete form,

$$X_c(k+1) = X_c(k) + G_c v_c \qquad (15)$$

If an engine component performance is changed, it causes a difference between measurement and estimation; then the STKF changes Xc until the difference approaches zero. The STKF can accurately estimate output variables, and for state estimation, it usually outperform classic Kalman filter because of the tuning mechanism. The method can be extended to nonlinear system by using the Extended Kalman Filter (EKF).

Fault Classification and Fusion

Diagnostic methods employing statistical inference can be mainly categorized into regression methods and classification methods, depending on the way information is processed. Applying regression techniques such as Kalman filtering on gas turbine engine diagnosis poses some stability problems when few measurements are available [10]; Another problem related to regression methods such as STKF is that, since health parameters are also used as tuning parameters, their estimates tend to become a mathematical artifact for achieving accurate state and output estimates in the presence of faults [11]. The improvements suggested in [11] help to minimize the affect of nonlinearity and modeling error. However, the training of neural networks for the whole flight envelope seems a complicated process and difficult to implement in practice. In this work, the health parameter estimates are used as fault indicators while diagnosis and severity estimation are done by the fault classifiers.

Classification algorithms solve the diagnosis problem in a different way and, therefore, give a different kind of results. Although the results may be less accurate due to the number of the classes available, classification algorithms are often more reliable and usually more stable. Moreover, some classification algorithms allow some qualitative knowledge to be introduced into the classification rule [18]. Those characteristics make classification algorithms very complementary to regression algorithms.

In the D^2S system, synergetic combination of both regression methods (Kalman Filtering) and classification methods is developed. Both model based and data driven approaches can be utilized in the design of fault classifiers, and the outputs from multiple classifiers are further fused to improve accuracy and confidence.

Figure 2 illustrates the application of multiple fault classifiers and information fusion techniques to jet engine fault diagnostics. In this example, both model based and data driven (e.g. the stochastic fault pattern and the Neural Networks classifiers) are implemented, and the results from different classifiers are fused using Dempster-Shafter Theory. Note that when the health is in tune with the expected level of degradation (change in parameters like efficiency in time), it will not be classified as a fault.

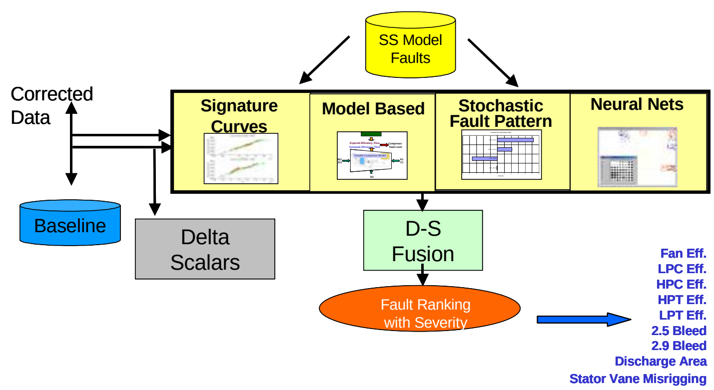


Figure 2: Multiple Fault Classifiers and Information Fusion

Robust fault classifiers were developed based on stochastic fault pattern recognition techniques that rely on matching the current set of deltas (difference between measured and expected steady-state sensor readings) to sets of delta values in a fault pattern database. This approach relies on gauging the proximity of the current deltas to those associated with known fault conditions in the 'N-delta' feature space (defined in the fault pattern database). The proximity of the current set of deltas to a known fault pattern's deltas determines the level of confidence with which one can expect this to be the current system fault. The deltas in sensor measurements are mapped to the health parameters of each component in order to estimate its health. For the proof-of-concept demonstration, the sensor delta feature space is four dimensional and consists of the four most commonly and universally available engine sensor measurements: Fan Speed (N1), Core Speed (N2), Fuel Flow (Wf), and Exhaust Gas Temperature (EGT). The fault pattern database was developed using the deviations/deltas associated with each measurement and the corresponding associated fault. Details regarding the fault classifiers and the Dempster-Shafter method were presented in previous publications [19,12].

AUTOMATED FAULT CONTINGENCY PLANNING

To meet the objective of real-time control reconfiguration and resource management in the D^2S system, the intelligent control system functionality requires the health management modules to provide continuous and up-to-date state awareness. With this information, adaptable control strategies can be implemented conceptually within the hierarchical accommodation strategy shown in Figure 3. In the presence of a fault, the high-level redistribution controller re-routes the available control authority taking advantage of any inherent redundancy in the system. A mid-level set point controller then determines set point values, which maintain stability of the restructured system, possibly at some degraded performance. Finally, the low-level algorithms adjust local controller gains in response to the new set points generated by the mid-level controller.

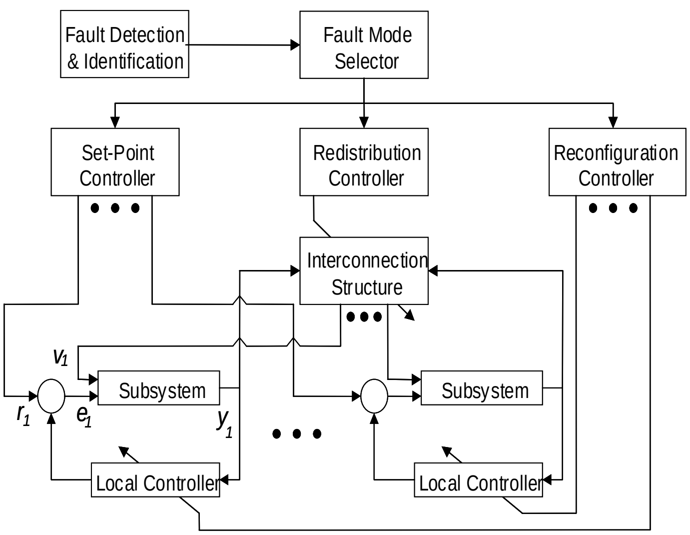


Figure 3: A Conceptual Structure for AFCP

At the propulsion level, one of the goals of the D^2S system is the capability to control (avoid) damaging gas path events such as surge or stall in the compressors, which is of fundamental importance in lengthening the life of those critical components in the engine. Figure 4 shows an automated fault accommodating control system that provides the capability to maintain acceptable performance and stability properties in the case when sensors, actuators, or other components malfunction in the engine.

Particularly, AFCP strategies related to typical engine faults are illustrated in previous work [19,12]. Worth pointing out is that although the D^2S architecture is a generic solution, the AFCP strategies to be implemented on a particular system are often application specific.

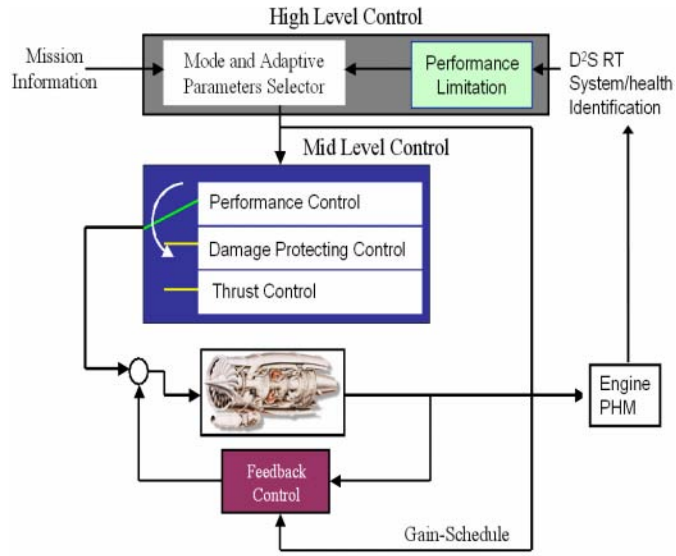


Figure 4: Adaptive Engine Control Strategies

<u>New Methods to Implement AFCP System</u>

A capability predicate model is utilized for development of the contingency management decision support function that uses a hierarchy of predicates for the physical layer, the regulation layer, the maneuvering layer, and the mission layer, as shown in Figure 5. The core part of a dynamic automated contingency management software is to use the information from the fault diagnosis to evaluate the vehicles capability for performing the function at all of these levels and completing its mission. The real-time fault accommodation component will perform the contingency analysis including the assessment of controller capabilities, assessment of flight capabilities, and assessment of mission accomplishment capabilities. The analysis results will form the new constraints for potentially changing operations or re-planning the mission. If the current mission plan can not satisfy the new constraints, mission re-planning may become necessary under of the new constraints for responding efficiently to changes of vehicles status and environments.

The capability predicate modeling concept is widely applicable to any set of interconnected subsystems as is the case for subsystems such as propulsion and multi-channel flight control actuators.

Capability Predicates for Physical Layer Predicates: The adaptive control system uses sensors, actuators, and communication devices. Each of them is modeled as a dynamic predicate, which will return 1 if it is functioning properly and 0 otherwise.

Capability Predicates for Regulation Layer Predicates: The regulation layer assesses the capabilities of the controllers themselves, and depends on the predicates of sensors and actuators in the physical layer. The capability of each controller in the regulation layer is modeled a dynamic predicate with 1 if the control functions are working properly and 0 otherwise.

Capability Predicates for Maneuvers or Key Functions: The vehicles maneuvering ability depends on several control laws in the regulation layer. The function capabilities can be expressed as a vector consisting of dynamic predicates with 1 if the specific maneuvers are possible and 0 otherwise.

Capability Predicates for Missions: Execution of an aircraft's overall mission requires certain maneuvers or a sequence of functions in a more generic case. The mission capabilities can be expressed as predicates with 1 if it functions properly and 0 otherwise.

The predicate model decides when and how the contingency strategies should be activated in an optimal way based on health assessment and mission requirement. This will be implemented within the Stateflow™ development tool for control and supervisory logic. This approach provides clear, concise descriptions of complex system behavior using finite state machine theory, flow diagram notations, and state-transition diagrams, all in the same Stateflow™ diagram.

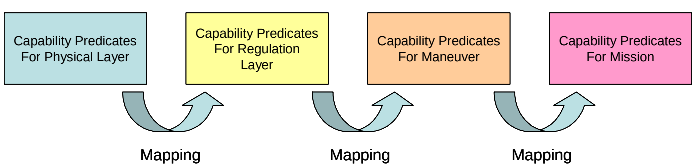


Figure 5: Fault Accommodation Capability Modeling Using Predicates

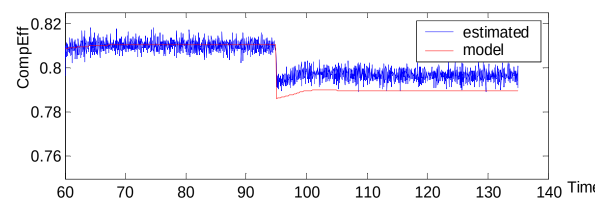
Stateflow™ generates its own C-code to simulate Stateflow™ charts during simulation. With the Stateflow™ Coder, code can be generated for applications built in other environments, such as an embedded environment. With Real-Time Workshop™ tool, code from Simulink and Stateflow can be run as an application on another environment to control a process.

SIMULATION RESULTS

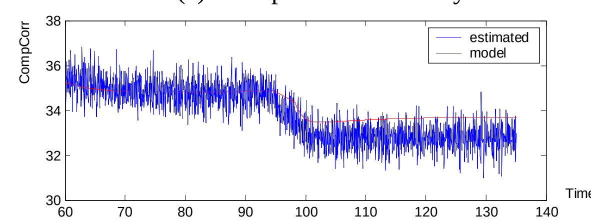
A generic turbofan engine model is utilized as a test bed, which has a bypass ratio of 4.8 and design fan and core speeds of 8,700 and 14,700 RPM and a rated thrust of 8,000 lbs. Mostl turbo-machinery based components, like the fan, compressor and turbine, are based on look-up tables obtained from engine test data. The test bench also contains a fault injection block that simulates engine faults by imparting gains, adding noise, or adding bias to model states. Simulated engine component faults include combustor efficiency loss, compressor efficiency loss, compressor flow capacity loss, fan hub efficiency loss, LPT efficiency loss, fan tip efficiency loss, HPT efficiency loss, degradation, and PT2 sensor failure, and etc.

STKF Simulation Results

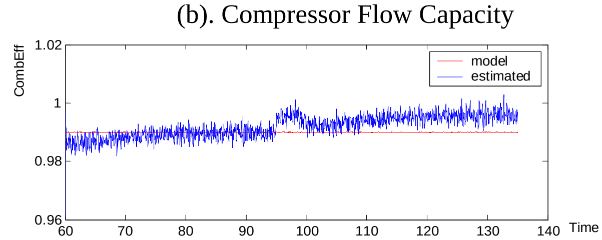
In this study, the state variables are N1, N2, T25, P25, T3, P3, T45, P45, T8, P8; control variable is Wf (fuel flow), output variables are N1, N2, T3, P3, T45, P45, T8, P8 and they are assumed to be measurable. Health parameters are compressor efficiency loss, compressor flow capacity loss, and combustor efficiency loss. Figure 6 shows the simulation results for a simulated 2% compressor efficiency loss introduced at time 95. Changes in the estimated values of the health parameters can be used for anomaly detection purpose. Since they also serve as tuning parameters, the estimated health parameter values might be inaccurate. However, STKF provides accurate state/output estimation in the presence of this fault. The estimation results for compressor efficiency, compressor flow capacity, combustor efficiency, N1, N2 and P3 are also shown in the same figure. Simulated measurement noises are Gaussian noise with standard deviation of .1% of the steady state values. The plots shows response time from time 60 to 140, the transient responses from time 0 to 60 correspond to an engine start procedure and are not shown in the plots. Note that though only simulation results for component faults are presented, the STKF technique is applicable to component degradation as well. Similar results were reported in [7].



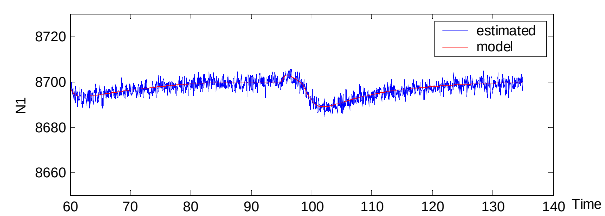
(a). Compressor Efficiency



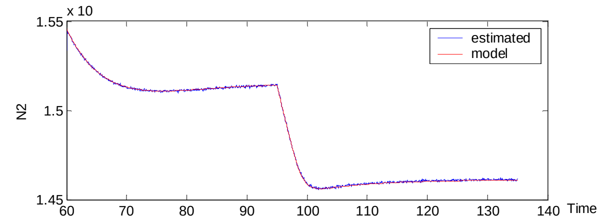
(b). Compressor Flow Capacity



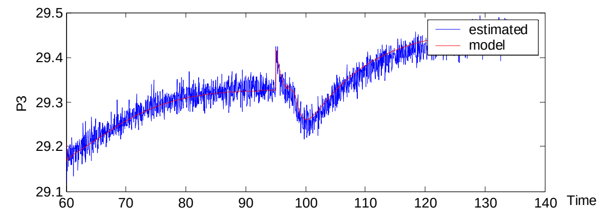
(c). Combustor Efficiency



(d). N1



(e) N2



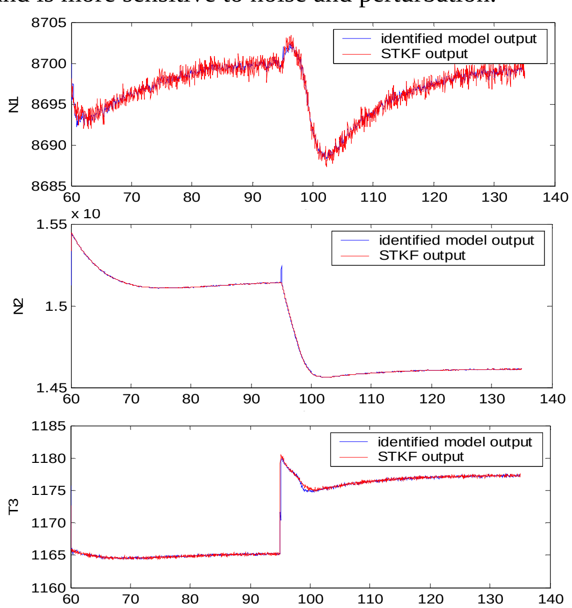
(f) P3

Figure 6: STKF Simulation Results

Real-time Recursive System ID Results

Figure 7 shows the tracking capability of the identified linear model compared with the output estimation from the STKF module. The forgetting factor (λ) and weighting factor (α) were set to 0.99 and 0.1 respectively.

In the above simulation results, the identified linear model could quickly adapt to the fault (2% Compressor Efficiency Loss introduced at time 95). However, there is a big overshoot in the N2, P3, P45 and P8 plots above. Figure 8 shows the P3 and P45 response of the identified linear model when the forgetting factor (λ) is set to 0.95. Decreasing λ value will result in more prompt response to system dynamics changes. However, it may produce less accurate parameter estimation and is more sensitive to noise and perturbation.



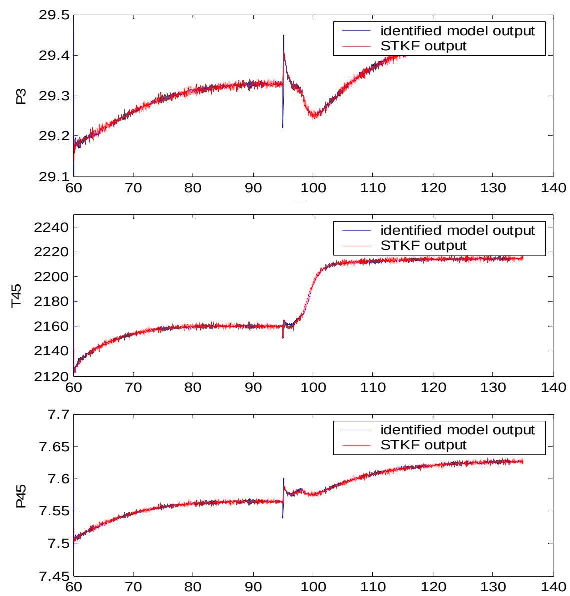


Figure 7: Comparison of Identified Linear Model Outputs and STKF Outputs

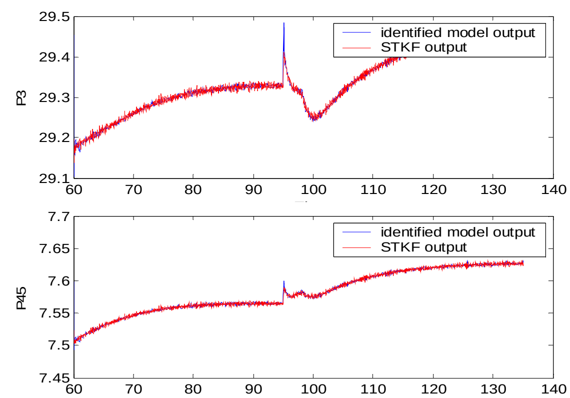
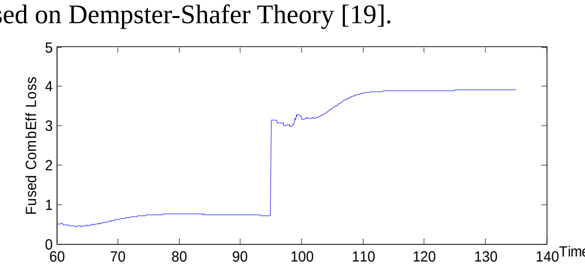


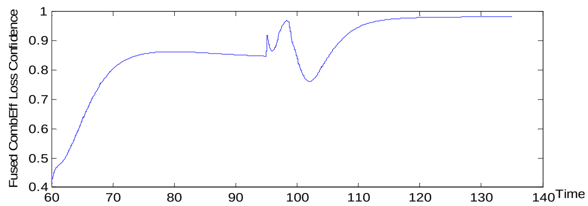
Figure 8: Comparison of Different Forgetting Factor Values

Fault Classification & Fusion Results

Figure 9 shows the fused classification result of a simulated 4% Compressor Efficiency Loss introduced at time 95, where figure (a) shows the fused classification results and (b) the fused confidence level. The classification result is weighted based on the classification confidence for different fault severity levels to produce continuous output values as shown in Figure 9 (a). The fusion algorithms implemented is based on Dempster-Shafer Theory [19].



(a) Fused Classification Result



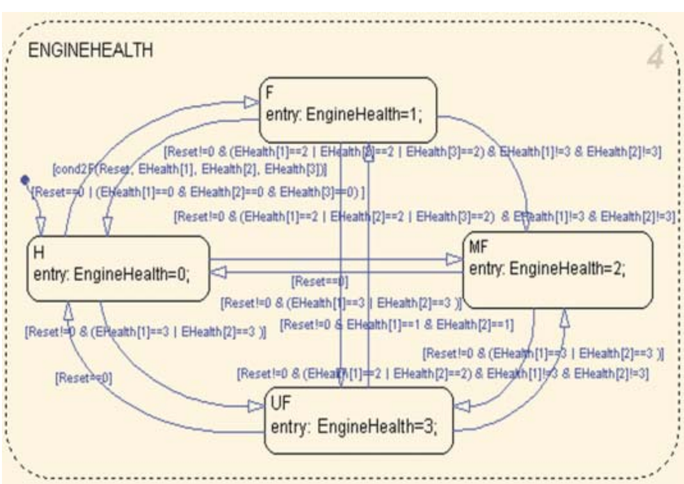
(b) Fused Confidence Level

Figure 9: Fused Classification Result and Confidence Level

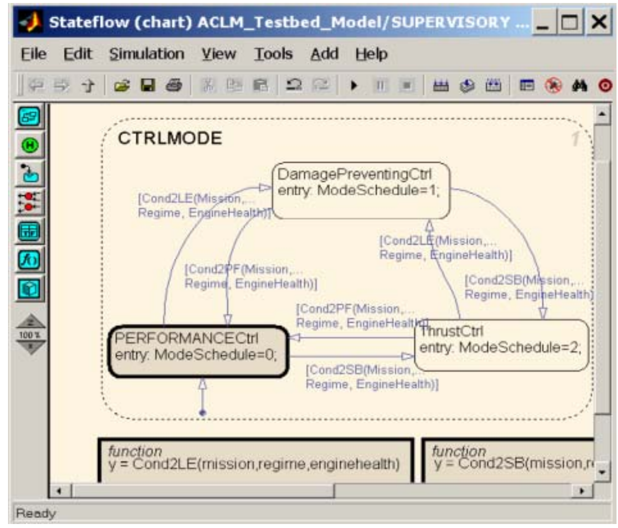
Automated Fault Contingency Planning

The AFCP module was implemented as a Finite State Machine in Stateflow[TM]. The purpose of this AFCP implementation is straightforward; given state awareness (i.e. faults, mission, mode/regime) what can be accomplished autonomously to best fulfill a set of objective functions (i.e. complete the mission, survive, etc.). The flight regimes used in demonstration are take off/descent, cruise, and attack maneuvers, while missions can be sortie, transport and surveillance. As a demonstration case, three control laws, namely performance control, damage protecting control and thrust control, were developed. The decision making regarding control law change is made based on consideration of engine health condition, operating mode, mission requirements, flight regime and command & control orders. Snapshots of part of the Stateflow implementation of the AFCP control mode decision making module and the automatic switching of control laws is shown in Figure 10 for illustration.

To test the performance of the AFCP module, a fault scenario was simulated as described in Table 1 below. Simulation Results are shown in Figure 11, where at time 87, the control law was switched to damage protecting control (lowering N1 setpoint value to reduce further damage), and at time 103 (with a delay due to the transient), the control low was switched to thrust control as shown in see Figure 11 (a). Figure 11 (b), (c) and (d) shows the commanded N1 value, thrust estimation and N1 estimation respectively. Interestingly, the same thrust level was obtained with a lower N1 speed with thrust control.



(a) Mapping of component level health information to engine level

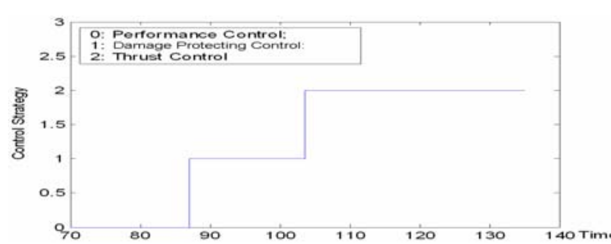


(b) Control decision making

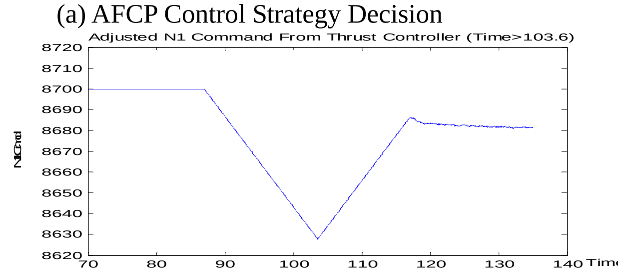
Figure 10: Stateflow TM Based FCM Implementation

Table 1. AFCP Simulated Fault Scenario

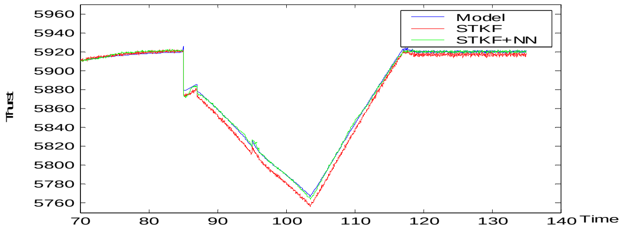
Simulation Time	Simulated Fault
[85 – 135]	3% Combustor Efficiency Loss
[95 – 135]	4% Compressor Efficiency Loss
[95 – 135]	3% Compressor Flow Capacity Loss



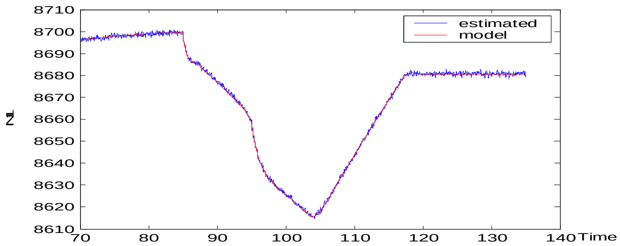
(a) AFCP Control Strategy Decision



(b) N1 Command Reference Signal



(c) Thrust Estimation



(d) Estimated N1 Value

Figure 11: AFCP Simulation Results

EMBEDDED SYSTEM DEVELOPMENT

To raise the technological readiness level of the presented techniques, an embedded system was developed to perform Kalman filtering and System Identification. The PC104 system was used due to its small size and expandable capability. Running Linux, it is capable of establishing TCP/IP connections, receiving data, performing operations on that data, and sending it back to the host. Two blocks were developed in Simulink for sending and receiving data to the PC104. Each time the Simulink model is run, a connection is established with the PC104 and data is sent and received at 25Hz. The hardware setup was shown in Figure 12.

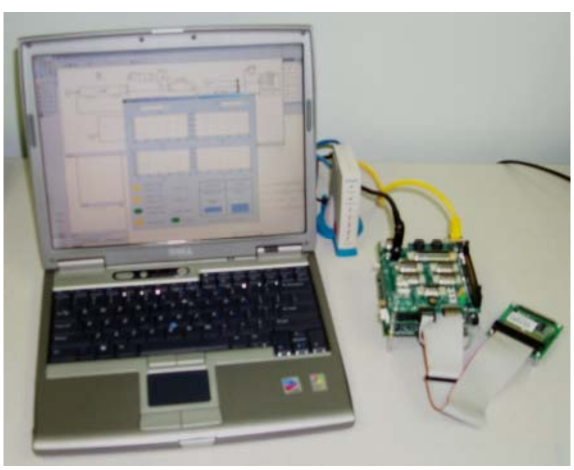


Figure 12: Embedded System Development & Testing Hardware Setup

The Kalman filter and the System ID algorithms were programmed in C and compiled for the PC104 platform. The testing results showed that the real-time recursive SID algorithm could track 72 parameters (the A and B matrix of the engine state space model) in less than 15 ms, allowing the algorithm to run as high as 66Hz on a 100MHz CPU with 32M bytes flash memory.

CONCLUSIONS AND FUTURE WORK

This paper reports the first phase of the D^2S system development process. Impact Technologies has developed and demonstrated a suite of innovative Dynamic Decision Support prototype software modules for real-time assessment of system health and fault contingency planning for potential application on JSF or Navy unmanned platforms systems. The proof-of-concept demonstration provides insight into the problems Impact is trying to address using the D^2S system. The program successfully demonstrates the ability to merge health monitoring and controls in a flexible, adaptable and scalable architecture. The dynamic modeling capabilities, techniques and tools sets will not only improve the reliability of the propulsion systems, but also greatly enhance maintenance decision support and contingency planning concepts.

The logical next step in this development is to expand the D^2S system to cover other engine components and accessories. Impact is working closely with Engine OEMs, to formulate a strategy for focusing the future efforts towards integrating the D^2S system with modular control systems. The next phase of the program will accomplish the following objectives:

1. Identify critical propulsion and power system faults that require detection, prognosis and contingency management in a target engine.
2. Implement the modular, flexible and scalable architecture to create an on-board propulsion D^2S system with health management and intelligent control capabilities.
3. Develop and implement advanced adaptive control strategies using the model identified online in real-time. Model predictive control is an area to be further investigated.
4. Develop algorithms to predict the achievable performance of the degraded system based on the model identified.
5. The combination of Kalman Filtering and classification methods will be further developed. A potential improvement is to use the classifier output as an *a priori* heath estimation for the Kalman Filter to improve the health parameter estimation accuracy.
6. Continue the embedded system development toward the goal of producing the envisioned product, which is a aircraft-embeddable real-time, modular software system that can be integrated with the FADEC and off-wing avionics hardware to provide real-time fault accommodation functionality to propulsion systems.

NOMENCLATURE

AFCP = Automated Fault Contingency Planning

COTS = Commercial off-the-shelf
EFK = Extended Kalman Filter
FADEC = Full Authority Digital Engine Controller
JSF = Joint Strike Fighter
N1 = Fan Rotor Speed, RPM
N2 = Compressor Rotor Speed, RPM
PHM = Prognostic and Health Management
PLA = Power Level Angle
SID = System Identification
STKF = Self Tuning Kalman Filter
V&V = Verification & Validation

ACKNOWLEDGMENTS

This work is supported by the Naval Air Warfare Center in Patuxent River, MD, under SBIR contract No. N68335-06-C-0073. The Impact Technologies project team gratefully acknowledges the support and guidance provided by project managers, Mr. Doug Gass, and Mr. Michael Begin of the NAVAIR Joint Strike Fighter Program Office. In addition, we would like to recognize the contributions of our colleagues, Mr. Mike Koelemay and Mr. Greg Rowe.

REFERENCES

1. Vachtsevanos G., Lewis F. L., Roemer M., Hess A., Wu B., 2006, "Intelligent Fault Diagnosis and Prognosis for Engineering Systems", WILEY.
2. Roemer M., Byington C. S., et al, 2005, "An Overview of Selected Prognostic Technologies with Reference to an Integrated PHM Architecture," ISHEM Forum, Napa Valley, CA, November 7-10.
3. Byington C.S., Watson M., Edwards D., Stoelting P., 2004, “A Model-Based Approach to Prognostics and Health Management for Flight Control Actuators”, *Proc. IEEE Aerospace Conference*, Big Sky MN, paper 1047.
4. Volponi A., Wood B., 2005, “Engine Health Management for Aircraft Propulsion Systems”, First International Forum on Integrated System Health Engineering and Management in Aerospace, November 7-10, Napa, CA.
5. Litt J.S., Simon D.L., Garg S. and et al. 2004, “A Survey of Intelligent Control and Health Management Technologies for Aircraft Propulsion Systems”, Journal of Aerospace Computing, Information, and Communication, vol.1, no.12, Pp. 543-563.
6. Gilyard G. and Orme, J. S., 1993, “PSC: Program Overview and Future Directions”, Technical Report TM 1993-4531, NASA Dryden FRC, Edwards AFB, CA.
7. Volponi A., DePold H., Ganguli R., Daguang C., 2000, “The Use of Kalman Filter and Neural Network Methodologies in Gas Turbine Performance Diagnostics: A Comparative Study”, *Proc. of ASME Turbo Expo*, 2000-GT-547.
8. Romessis C., Mathioudakis K., 2006, "Bayesian Network Approach for Gas Path Fault Diagnosis", Journal of Engineering for Gas Turbines and Power, Vol. 128, Issue 1, pp. 64-72.
9. Ganguli R., 2003, “Application of Fuzzy Logic for Fault Isolation in Jet Engines”, *ASME Journal of Engineering for Gas Turbines and Power*, **125**, pp 617-623.
10. Dewallef P., Romessis C., Léonard O., and Mathioudakis K., 2006, “Combining Classification Techniques With Kalman Filters for Aircraft Engine Diagnostics”, Journal of Engineering for Gas Turbines and Power, Vol. 128, Issue 2, pp. 281-287.
11. Volponi A., 2005, “Use Of Hybrid Engine Modeling For On-Board Module Performance Tracking”, Proceedings of GT2005 ASME Turbo Expo: Power for Land, Sea and Air, GT2005-68169, June 6-9, Reno-Tahoe, Nevada, USA.
12. Kallappa P., Hailu H., 2005, "Automated Contingency And Life Management For Integrated Power And Propulsion Systems", Proceedings of ASME Turbo Expo, Power for Land, Sea and Air, June 6-9, Reno-Tahoe, Nevada, USA.
13. Wiseman. M.W., and Guo, T.H., 2001, “An Investigation of Life Extending Control Techniques for Gas Turbine Engines”, Proceedings of the 2001 American Control Conference.
14. Adibhatla, S., and Lewis, T., 1997, “Model-Based Intelligent Digital Engine Control (MoBIDEC),” Paper AIAA–97–3192, 33rd Joint Propulsion Conference.
15. Fuller, J., 2004, “Model Predictive Control for Gas Turbine Power Systems”, GT–2004–54110, Proceedings of the ASME Turbo Expo.
16. D. Shore & M. Bodson, 2005, “Flight Testing of a Reconfigurable Control System on an Unmanned Aircraft,” AIAA Journal of Guidance, Control, and Dynamics, vol. 28, no. 4, pp. 696-707.
17. Bodson, M., and Groszkiewicz, J., 1997, “MultivariableAdaptive Algorithms for Reconfigurable Flight Control,” IEEE Transactions on Control Systems Technology, Vol. 5, No. 2, pp. 217–229.
18. Romessis, C., and Mathioudakis, K., 2004, “Bayesian Network Approach for Gas Path Fault Diagnosis,” ASME Turbo Expo, ASME Paper No. GT2004-53801.
19. Tang L., Kacprzynski G. J., Roemer M., Vachtsevanos G., and Patterson-Hine A., 2005, “Automated Contingency Management Design for Advanced Propulsion Systems”, Infotech@Aerospace, Arlington, Virginia. 26-29 September.
20. Litt J. S., Turso J. A., Shah N., Sowers T. S., Owen A. K., 2005, “A Demonstration of a Retrofit Architecture for Intelligent Control and Diagnostics of a Turbofan Engine,” Infotech@Aerospace, 26 - 29 September, Arlington, Virginia.

