

Rapid Detection of Faults for Safety Critical Aircraft Operation¹²

Kai Goebel, Neil Eklund and Brent Brunell
GE Global Research
One Research Circle
Niskayuna, NY 12309
{goebelk, eklund, brunelbr}@research.ge.com

Abstract—Fault diagnosis typically assumes a sufficiently large fault signature and enough time for a reliable decision to be reached. However, for a class of safety critical faults on commercial aircraft engines, prompt detection is paramount within a millisecond range to allow accommodation to avert undesired engine behavior. At the same time, false positives must be avoided to prevent inappropriate control action. To address these issues, several advanced features were developed that operate on the residuals of a model based detection scheme. We show that these features pick up system changes reliably within the required time. A bank of binary classifiers determines the presence of the fault as determined by a maximum likelihood hypothesis test. We show performance results for four different faults at various levels of severity and show performance results throughout the entire flight envelope on a high fidelity aircraft engine model.

TABLE OF CONTENTS

1. INTRODUCTION	1
2. SYSTEM MODELING.....	2
3. MODEL-BASED DETECTION.....	3
4. DIAGNOSTICS	4
5. SELECTED FAULTS.....	5
6. ELBOWING.....	6
7. RESULTS AND DISCUSSION	7
8. ACCOMMODATION.....	10
9. CONCLUSIONS	11
ACKNOWLEDGEMENT.....	11
REFERENCES	11
BIOGRAPHY	12

1. INTRODUCTION

Fault detection typically assumes a sufficiently large fault signature and enough time to come up with a reliable decision. However, faults that affect aircraft safety must be detected quickly. Otherwise, undesired consequences may

arise such as engine surge/stall events, power loss, severe vibrations, power loss, and no thrust response to commanded power. These events in addition to inappropriate crew response may lead to accidents [1]. Since the air traffic is projected to increase in the long term, and at the current level of reliability, the number of accidents would increase as well. To reduce this projected number of accidents NASA established the Aviation Safety Program [2]. Within the scope of the program this paper investigates a number of faults for which in-flight accommodation is considered.

Based on the review of an engine events database of the past 20 years of DEC and FADEC operated aircraft a list of root cause faults was compiled. From this list a set of representative cases from different categories including sensor faults, actuator faults, and gas-path component damage was chosen. More specifically, compressor damage, turbine damage, variable stator vanes faults, and ps3 sensor loss were selected. The results of these failures can lead to compressor stall, loss of power, loss of thrust control, and flameouts. Requirements for time to detection were established to range from roughly 100ms for some actuator faults to 1.2s for gas path faults. This means that in the worst case, only a few samples of sensor data are available to perform the fault detection. With these time constraints, the full magnitude of the fault signature will in some cases not be seen in the measurement data until well after the limits for accommodation, even if the fault mode is represented as a step change. In addition, noise, engine-to-engine variation, deterioration, model uncertainty, and closed loop controller effects further encumber the detection effort. Because it is envisioned to ultimately develop automatic fault accommodation, another requirement imposed on the detection includes a zero false positive rate to avoid taking remedial action when no fault exists.

To allow for the detection of small changes in engine components, a suite of techniques was employed to address certain aspects of these issues in parallel.

¹ 0-7803-8155-6/04/\$17.00© 2004 IEEE

² IEEEAC paper #1360, Version 6, Updated December 17, 2003

Figure 1 shows the top-level architecture for the detection methods used. In particular, a model-based approach was chosen employing extended Kalman filter (EKF) techniques with a component level engine model (CLM) featuring nonlinear simulation of a commercial high-bypass twin-spool turbofan engine. The top-level inputs to the system are the pilot input throttle resolver angle (TRA), flight conditions including altitude (ALT), mach number (XM), and delta from ISO standard day ambient temperature (DTAMB), variation and health parameters (P), and fault inputs. Data were generated for an exhaustive set of cases within the flight envelope. For all of the data sets random variations were added to the process and measurements corresponding to engine-to-engine variation, deterioration, sensor accuracy bias, and sensor noise.

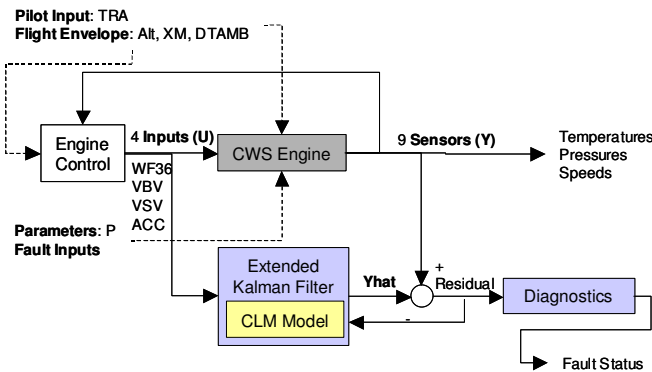


Figure 1 - Diagnostic architecture

Since emphasis was placed on the quick detection, feature extraction underwent special attention. First, the effects of fault signature variability of the tracking residuals due to operating conditions were investigated using a generic baselining approach. In addition, several advanced features were developed that indicate a deviation from normal operation including some techniques that are more commonly found in data trending. Finally, a bank of binary classifiers determines the presence of the fault as resolved by a maximum likelihood hypothesis test. We show performance results for four different faults at various levels of severity (small, medium, and high) at steady state with no change in actuator inputs or flight conditions and continuously for several seconds after fault introduction.

2. SYSTEM MODELING

For this study two nonlinear models of an advanced commercial high-bypass twin-spool turbofan engine were used. Both models can be run in a transient or steady state mode. The first is the truth model that will act as the real engine. The model used for this purpose is the cycle workstation (CWS). CWS is a high fidelity physics-based aircraft engine model. Variation, deterioration, and fault models can be injected into the CWS model. A component level model (CLM) is used as the embedded model in conjunction with the EKF to aid in the fault diagnostics. The

CLM is also physics-based but runs faster and is less accurate than CWS. The top-level inputs and outputs to the CLM are shown in Figure 2

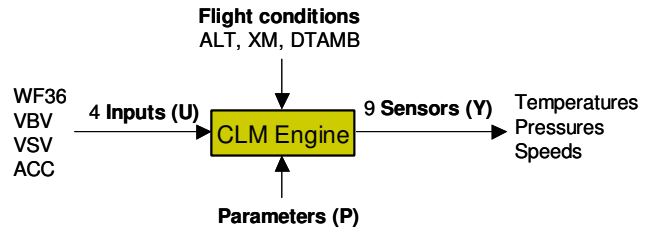


Figure 2 - CLM inputs, parameters, and sensors

The inputs to the CLM are the four actuator inputs (U),

1. fuel flow – WF36,
2. variable bleed valve – VB,
3. variable stator vanes – VSV,
4. active clearance control valve – ACC,

the flight conditions, and the variation and health parameters (P),

1. efficiency adders on fan, booster, high pressure compressor (HPC), high pressure turbine (HPT), low pressure turbine (LPT),
2. flow scalars on the fan, booster, HPC, HPT, LPT,
3. seal leakage scalars,
4. turbine clearance adders.

The outputs from the CLM are the nine sensors (Y) consisting of the following physical measurements,

- temperatures,
- pressures,
- speeds.

Variation, Deterioration, Bias, Noise

In order to represent more realistically engine behavior sources of variation are included as inputs to the simulations. Specifically, we considered variability between engines, as engine age, and through measurement processes.

Engine-to-Engine Variation

An engine-to-engine variation model was created for the CWS engine model. The engine-to-engine variation accounts for manufacturing and assembly variation found in new engines. These variations can be described or modeled by adding variation to the efficiency adder, and flow scalar parameters on the rotating components. The model used for engine-to-engine variation applies a normal distribution to each of the parameters about their nominal value. In particular, each of the parameters is assigned a random normally distributed variation based on measurements of the engine population. On a 1 sigma basis the parameter variation values were found to range between 0 to 1.1%.

Deterioration

A deterioration model was created for both of the engine models. Some of the known deterioration mechanisms are seal and bushing leakage, clearance increases, and other main or secondary flow leaks. After evaluating analytic engine teardowns, production test data, development test data, and overhaul engine findings it was determined that the rotating component efficiencies and flow parameters, as well as leakage and clearances are effected by deterioration. The deterioration model used in the CWS engine model is a random uniform distribution from new to fully deteriorated. The process is centered for the CLM engine model by running it only with a 50% deterioration model.

Sensor Accuracy

There are several factors that affect the measurement accuracy of any given sensor. Factors considered here are;

- signal conditioning – accounts for excitation, a/d conversion, filtering,
- sensor bias – accounts for sensor to sensor variation,
- profile error – accounts for radial and circumferential variation in the measured parameters,
- noise – accounts for other noise in the system.

All accuracies are two-sided (\pm) and modeled as normal distribution with maximum equal to 2-sigma variation.

3. MODEL-BASED DETECTION

The basic idea of a model-based approach is to compare the actual plant output to the output estimated from a mathematical model that attempts to mimic the system behavior [3]. An optimal estimate of the state of the model in the presence of process noise, sensor noise, and initial condition mismatch for a nonlinear plant can be created using extended Kalman filtering (EKF) techniques. The difference between the plant output and the estimated output is the residual. The residuals are then used to detect changes in the plant behavior. Residuals should be near zero in fault-free case and deviate from zero when a fault has occurred. The decision process evaluates the residuals and monitors if and where a fault has occurred. Classifiers that operate on the residuals can be used to deal with the decision process. Typical problems with the Kalman filter-based state estimation approach are that the filter does not account for unmodeled dynamics and other sources of variation including engine-to-engine variation, deterioration, and sensor biases. Furthermore, the response to an abrupt change may be sluggish if the filter gain is small. We will outline below how these issues were addressed.

The advantages of the EKF are that it is nonlinear, operates over the full envelope, accounts optimally for sensor and

process noise, and that it provides state estimates which might assist in the failure isolation and accommodation. Faults such as blockage in the PS3 sensor line, which manifest themselves as a delay in the transient response, could be more visible with a dynamic estimator. The disadvantages vis-à-vis the Kalman filter (KF) are the computational complexity of the EKF, the fact that non-trivial parameter estimation or bias estimation needs to be provided and that tuning is more complicated.

The EKF for the engine is based on the *Component-Level Model (CLM)*. This model is parameterized by states x and z , inputs u , and parameters p . Thus,

$$\begin{aligned}\dot{x}_t &= f(x_t, p_t, u_t) + w_t \\ y_{k\Delta t} &= h(x_{k\Delta t}, p_{k\Delta t}, u_{k\Delta t}) + v_{k\Delta t}\end{aligned}$$

where the measurement y arrives at every Δt seconds and the white noise variables w and v represent the process and measurement noise (respectively). This is a continuous-time dynamical system with discrete-time (sampled) measurements. The state, x , comprises the spool speeds and temperatures — 9 states in total — followed by several clearance model states and a few sensor states. Its dimension is 94. The input, u , has 4 control inputs WF36, VBV, VSV, ACC, the environmental variables of altitude (ALT), mach number (XM), and ambient temperature (DTAMB). The CLM output, y , contains the 9 sensed outputs. The parameter vector, p , contains the coefficients representing the variation and engine wear parameters. This should capture the engine deterioration and the engine-to-engine variability.

For the purposes of the EKF, we run the CLM in discrete time, with time index $t = k \, dt$. The time-update section of the EKF is given in standard form for the x -part of the equations

$$\hat{x}_{k+1|k} = \hat{x}_{k|k} + dt \, f(\hat{x}_{k|k}, u_k, p_k).$$

Here we have used Euler integration to move from continuous to discrete time.

The measurement update section requires the linearized model for the computation of the Σ -matrix using the partial derivatives

$$F_k = \frac{\partial f}{\partial x}(\hat{x}_{k|k}, u_k, p_k), \quad H_k = \frac{\partial h}{\partial x}(\hat{x}_{k|k-1}, u_k, p_k).$$

The Kalman gain is computed as follows.

$$\begin{aligned}K_k &= \Sigma_{k|k-1} H_k^T (H_k \Sigma_{k|k-1} H_k^T + R_k)^{-1} H_k \Sigma_{k|k-1} + Q_k, \\ \Sigma_{k+1|k} &= F_k \Sigma_{k-1|k} F_k^T - F_k \Sigma_{k|k-1} H_k^T (H_k \Sigma_{k|k-1} H_k^T + R_k)^{-1} H_k \Sigma_{k|k-1} F_k^T + Q_k.\end{aligned}$$

The state estimate measurement update is

$$\begin{aligned}\hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k (y_k - \hat{y}_{k|k-1}), \\ &= \hat{x}_{k|k-1} + K_k (y_k - h(\hat{x}_{k|k-1}, u_k, p_k)).\end{aligned}$$

The state estimate $\hat{x}_{k|k}$ is then used in the output equations to obtain the estimated sensed values (\hat{Y}) that is used in the residual calculations.

4. DIAGNOSTICS

Classifier design is in practice most of the times guided by particular performance requirements. That is, there are particular specifications about the false positive and true positive rates (or false negative or true negative rates). The actual specification is driven by the application and the domain where the classifier is executed. For example, it might be extremely undesirable to issue false positives in certain domains while (even for the same application) it is more important to avoid false negatives in a different domain (such as in some military vs. civilian applications). At any rate, the classifier designer would follow a path of data pre-processing, feature extraction and selection, classifier design and optimization, and classifier evaluation [4]. If the classifier performance is not satisfactory, the steps of classifier selection and optimization as well as feature selection and extraction may be revisited to improve on the results (see Figure 3).

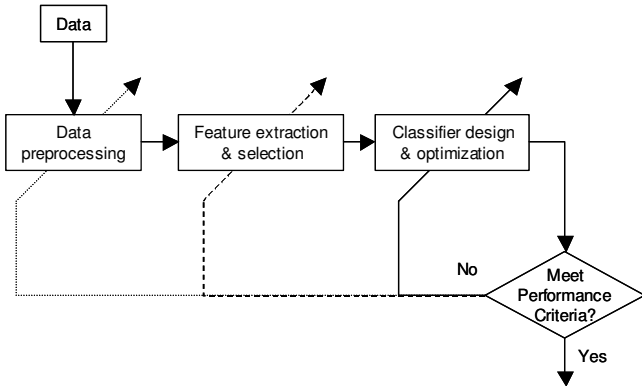


Figure 3 - Typical classifier design process

Typically there are a number of classifiers that might be considered for a given problem and each classifier will exhibit a performance ceiling. This ceiling might prevent the overall goals of the problem to be achieved. A convenient way to display the classifier performance for a binary problem over the range of performance criteria is the receiver operating characteristic (ROC) curve. ROC analysis is an established method of measuring classification performance in various domains such as medical imaging. Originated from the field of signal detection to depict tradeoffs between hit rate and false alarm rate [5] ROC analysis and its associated indices have been extended for use in evaluating performance of 2-class classifiers [6, 7]. The ROC space is a coordinate system that is used for

visualizing classifier performance by plotting the true positive rate (TPR) against the false positive rate (FPR). By varying the decision threshold of the classifier, a series of points (FPR, TPR) can be obtained and a ROC curve can be generated by connecting these points. Typical ROC curves are shown in Figure 4 where the three curves represent three different classifiers. Classifiers with ROC curves located in the upper-left corner in ROC space are better because they represent classifiers that have lower false positive rate and higher true positive rate than the classifiers below them.

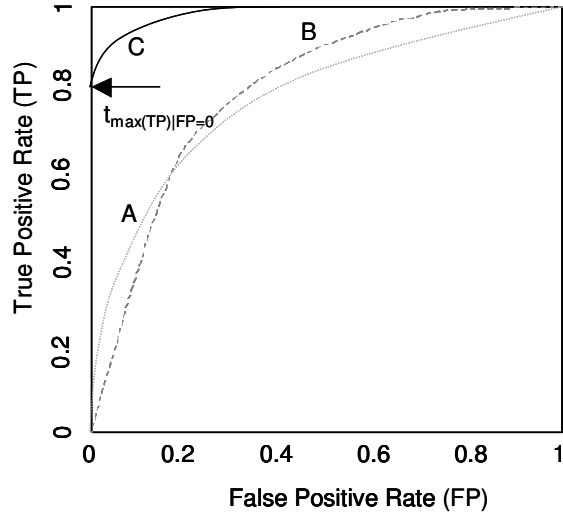


Figure 4 - ROC curves for classifiers A, B, and C.

The three ROC curves in Figure 4 correspond to three different classifiers A, B, and C. One can easily conclude that classifier C is better than or at least as good as the other two classifiers for all possible cost and class distributions since curve C dominates others over the entire range. Determining which of the two classifiers (A and B) is better, on the other hand, would not be so straightforward unless a specific performance requirement is given. For example, given the maximum acceptable false positive rate, one would simply draw a vertical line at the specified maximum FPR, and rank the classifiers based on TPR at the intersection of ROC curves with this vertical line. Similarly, one would use a horizontal line to rank the classifiers if the maximum required true positive rate is given. There is always a trade-off between false FPR and TPR and typically, only one parameter can be set. However, there has been a general desire to increase the overall accuracy of classifiers that has in part driven the development of classifier fusion systems. For the application at hand, the requirements were set to a zero false positive rate and the highest true positive rate possible. This results in a setpoint for notional classifier C as indicated by the horizontal arrow labeled $t_{\max(TP)|FP=0}$ in Figure 4. That is, the best true positive rate that can be accomplished by notional classifier C under these constraints is about 80%. We will display the results for the fault detection in the context of safety critical faults in the same way in the results section.

There are many strategies how to take advantage of the model based information and how to obtain the desired diagnostic information. These strategies range from simple thresholding to Kalman Filter banks [8]. To maximize the diagnostic information, we add a classifier to the model-based detection. Specifically, we consider here (after an initial of downselect as part of the classifier design step) a Support Vector Machine (SVM) as the classification engine [9, 10]. A SVM is a maximum margin linear classifier in a high dimensional feature space $\Phi(x)$ defined by a positive definite kernel function $k(x, x')$ specifying an inner product in the feature space $\Phi(x)$. $\Phi(x') = k(x, x')$ given a labeled data set

$$D = \{(x_i, y_i)\}_{i=1}^l, x_i \in X \subset R^d, y_i \in Y = \{-1, +1\}$$

The Gaussian radial basis function can be used as a kernel function.

$$k(x, x') = e^{-\frac{\|x-x'\|^2}{2\sigma^2}}$$

The function implemented by a support vector machine is given by

$$f(x) = \left\{ \sum_{i=1}^l \alpha_i y_i k(x_i, x) \right\} - b$$

where the optimal coefficients α can be maximized by the functional

$$W(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j k(x_i, x_j)$$

in the non-negative quadrant $0 \leq \alpha_i \leq C, i = 1, \dots, l$

subject to the constraint $\sum_{i=1}^l \alpha_i y_i = 0$

C controls the compromise between maximizing margin and training error. The optimization conditions (Karush-Kuhn-Tucker conditions) are:

$$\begin{aligned} \alpha_i = 0 &\Rightarrow y_i f(x_i) \geq 1 \\ 0 < \alpha_i < C &\Rightarrow y_i f(x_i) = 1 \\ \alpha_i = C &\Rightarrow y_i f(x_i) \leq 1 \end{aligned}$$

A set of Lagrange multipliers α^0 satisfies these conditions. The bias b is set s.t. the second optimization condition is met. Input patterns for feasible Lagrange multipliers are called support vectors [11].

5. SELECTED FAULTS

One of the goals of this project is to create general diagnostic technologies for the different components of an aircraft engine system. To that end we wanted to select actuator, engine, and sensor faults. We selected the variable stator vanes (actuator), high-pressure compressor and high-pressure turbine damage (engine), and ps3 sensor as the components that are the most suitable to investigate for this

project based on an investigation of an engine events database of the past 20 years of DEC and FADEC controlled engines. The fault block diagram is shown in Figure 5 and the parameters and magnitudes varied for each fault are given in Table 1.

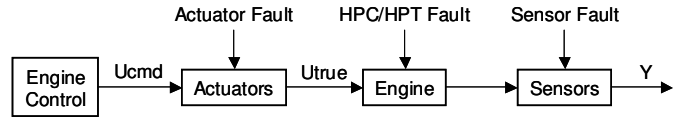


Figure 5 - Fault block diagram

Table 1 - Fault Model Parameters

	CLM Input	Small Fault	Medium Fault	Large Fault
VSV Fault	VSV bias	2.50%	5.80%	10%
HPC Fault	Efficiency	-1.50%	-3%	-6%
	Flow	-1.50%	-3%	-6%
HPT Fault	Efficiency	-1.50%	-3.50%	-6%
	Flow	0.15%	0.35%	0.60%
ps3 Leak	Vector	0.9	1.4	1.75

These faults may have by definition safety-averse consequences such as loss of throttle control, compressor stalls, aborted takeoffs, in-flight shutdowns, etc. It is therefore crucial to overcome these situations through proper accommodation. A prerequisite for accommodation is detection before the negative effects take place. Assuming a worst case scenario where the fault has a rise time of less or equal to one time sample (in our case 15 ms), the detection and final fault finding will need to be complete within a time frame of between 106ms and 1.2s, depending on the fault in question. Table 2 summarizes the allowed fault detection time.

Table 2 – Allowed fault detection time

Fault	Allowed detection time
HPC	1.2s
HPT	1.2s
VSV	106ms
ps3	340ms

The following sections will detail the selected fault for each component, describe how the faults were modeled, and investigate the resulting fault signature or characteristic.

Variable Stator Vane (VSV) fault

For the variable stator vanes position errors and other losses of system actuation account for the majority of the VSV system failures. The remaining faults result from localized failures at individual lever arm/vane sites. The VSV position error failures are modeled as a steady in-range bias and a drifting in-range bias. Lever arm failures were not

modeled since they would require three dimensional flow models and because the available VSV model is a lumped system level model.

HPC/HPT fault

There is a large number of different causes recorded in relation to the total number of cases. However, not all root causes are truly known. A number of faults are hidden in the catch-all “blade failure” without distinction to whether it is material fatigue, DOD, or some other cause. Based on the granularity of the fault mode, this may distort a Pareto evaluation. Irrespective of these caveats, this study examined blade failure as the cause to be characterized, modeled, and later detected and accommodated. There are a few assumptions under which the modeling needs to be performed. One is that control actions exist that can avert safety critical damage to compressor and turbine components. Here it is important to note that only events with precursors can be possibly averted. For some faults only secondary damage may be prevented.

The HPC and HPT damage models are taken from experience based on the Navy SECAD program [3]. Data and testing on the SECAD program validated that these failures can be modeled as changes to the health parameters of the affected components. In this case these are the flow and efficiency health parameters in the compressor and turbine.

ps3 Sensor fault

The failure mode for the ps3 (compressor discharge static pressure) sensor is leaks in the pressure line feeding the sensing elements. Leakage occurs for example due to a hole in the ps3 sensor tube. The nature of the ps3 fault has strong implications for detecting them. A leak is detectable in transient or steady state conditions, but (assuming constant sensor noise) will be most detectable at relatively high values of ps3.

6. ELBOWING

As mentioned earlier, requirements for time to detection were established to range from 106ms for VSV actuator faults to 1.2s for gas path faults. This implies that there may be only a few samples available for fault detection. With these time constraints, the full magnitude of the fault signature will in many cases not be seen in the measurement data until well after the limits for accommodation, even if the fault is injected as a step change. Noise, engine-to-engine variation, deterioration, and model uncertainty further weigh down the detection algorithms. In addition, the zero false positive requirement makes reliable detection even harder. It was therefore deemed prudent to focus some effort on development of features that can aid in the detection process.

Within this context, we introduce here an elbowing operator that responds to abnormal changes. Because changes do not show up as a step change, the detection is encumbered. This is true for most measured parameters because mechanical and thermodynamical inertia acts as a dampening agent such that changes come to light with some lag only. What hampers reliable early recognition further is the noise in the system, which is the primary potential source for false positives. It is vital to find the point at which the change may have been initiated. Traditional regression techniques do not do well because only a few data points are available for the change detection and small changes are drowned in the noise. That in turn may result in drastically different regression results as well as very unreliable determination of the elbow point. Generally these techniques can be found within the trending domain where similar problems of quick detection occur (although the sampling time may differ by several orders of magnitude). To address quick change recognition, we carry out a quasi sliding hypothesis test that evaluates whether a change over a window of observations meets certain change criteria. These criteria are based on change persistence as well as change magnitude. If these criteria are met, an “elbow” point, i.e., the point at which the potential change has first occurred, is identified and retained. The difference between the elbow point and the current smoothed observation is then used as a feature for the classifier. The window size for smoothing is determined by the time to detection requirements. One advantage of this feature is that it is always baselined to 0 because it operates on the differences only. The operative equation for the elbowing variable $x(k)$ is

$$x(k) = m(k) - x_{elbow}$$

with

$$x_{elbow}(k) = \begin{cases} m(k) & \text{if } |res(k) - res(k - k_{elbow} + 1)| < d_{th} \\ x_{elbow}(k-1) & \text{otherwise} \end{cases}$$

where

$$m(k) = \text{mean}(res(k), res(k-1), res(k-2), res(k-3), res(k-4))$$

$$res(k) = \hat{y}(k) - y(k)$$

k_{elbow} is the time at which elbowing was observed

d_{th} is the threshold for elbowing

Figure 6 shows the temperature residual and corresponding elbowing where the fault was injected at time $t=0.6$ seconds.

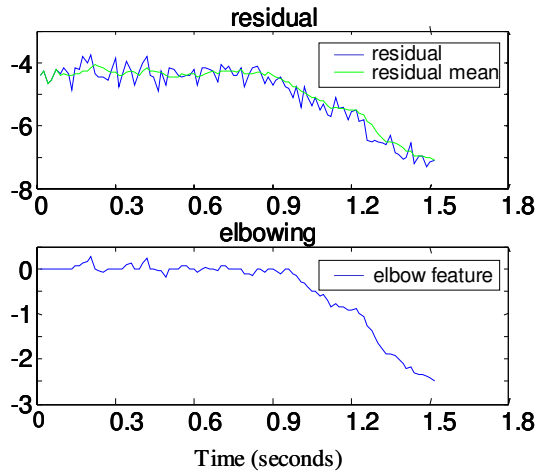


Figure 6 - small HPT fault example; (a): residual and mean; (b) “elbow” feature

7. RESULTS AND DISCUSSION

The overall classification approach followed the scheme shown in Figure 7. Inputs are the flight envelope (FE) data, sensor data, and Kalman filter estimates. After a preliminary variable selection, residuals are computed from which further features are calculated including the elbowing features. After a further feature selection process, the features are subjected to the classification (here multiple binary classification). The classifiers were trained as binary classifiers using the fault data as one class and the normal and other fault classes as the other class. The last step is the hypothesis test that selects the final fault state.

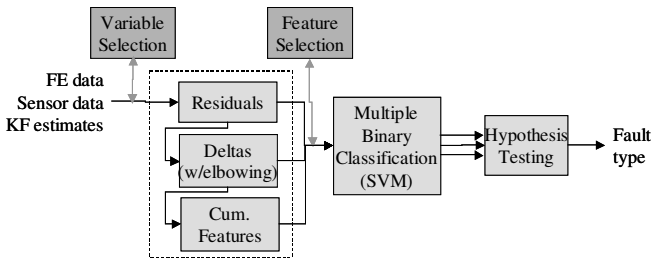


Figure 7 - Fault Classification Scheme

Following are a number of ROC curves that summarize fault specific results. Because performance is generally good, we show only a small portion of the ROC curve from the true positive (TP) point where the false positive (FP) rate was zero. This means that the axes of the curves are different in the cases presented to provide a better appreciation of the performance.

The ROC in Figure 8 shows the results for detection within allowed detection period. The HPC fault needed to be detected within 1.2s For a false positive rate of zero, the TP rate is 0.998. The somewhat choppy line is due to the number of cases used which at this level of granularity is not

sufficient to provide a smooth curve.

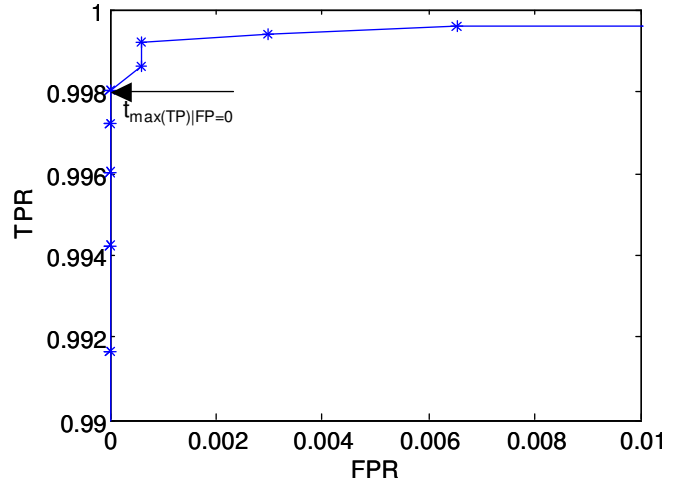


Figure 8 - ROC curve for HPC fault classification

The ROC in Figure 9 shows the HPT fault which needed to be detected within 1.2 s. For a false positive rate of zero, the TP rate is 0.9998;

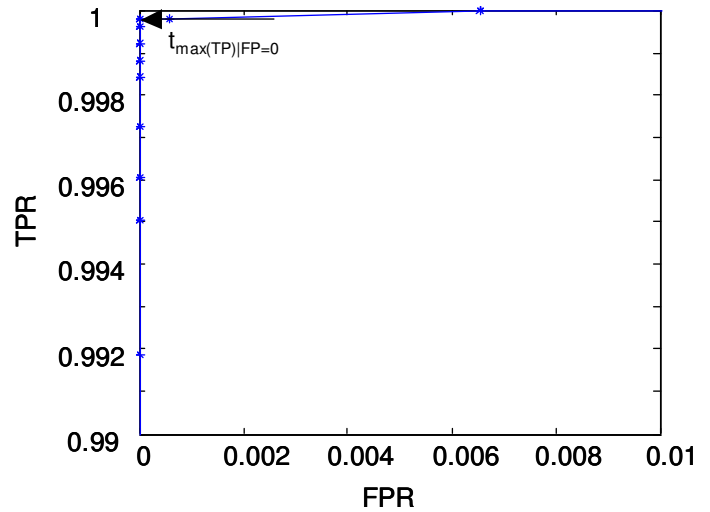


Figure 9 - ROC curve for HPT fault classification

The VSV fault needed to be detected with 106ms. For a zero false positive rate, the corresponding TP rate is 0.953. The ROC in Figure 10 shows the corresponding ROC.

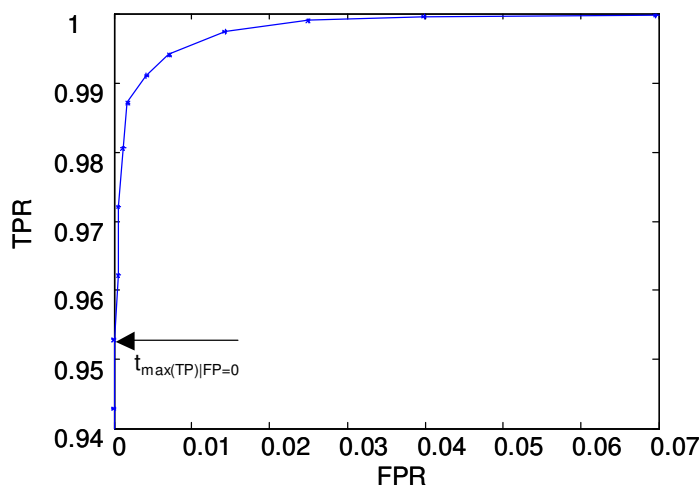


Figure 10 - ROC curve for VSV fault classification

The ps3 fault needed to be detected within 340ms. For zero false positives, the TP rate is 0.52. Figure 11 shows the associated ROC.

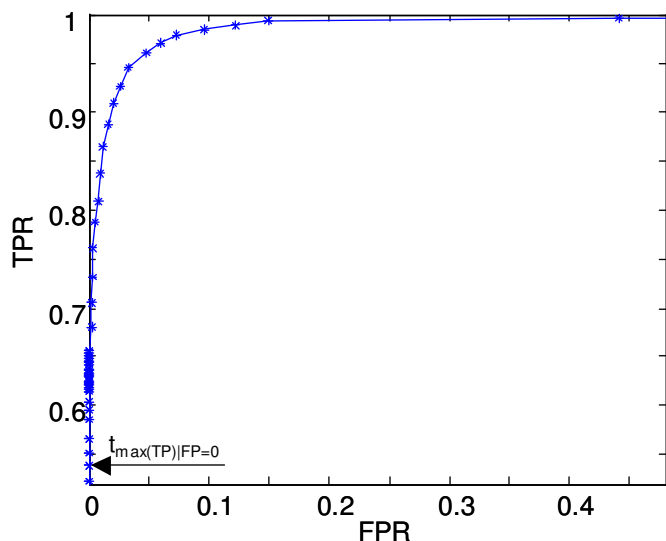


Figure 11 - ROC curve for ps3 fault classification

One of the main distinctions of this study is that results are compiled for faults that can occur at any point in the flight envelope with any level of noise and engine deterioration. With the mandated zero false positive rate (to avoid any inappropriate accommodation), results are still very reasonable as shown in Table 3 that shows the confusion matrix for the output of the maximum likelihood fault selection logic with sensor residuals alone.

Table 3 - Confusion matrix for rapid detection of selected faults (FP forced to zero) – sensors only

	No fault	HPC	HPT	VSV	ps3
No fault	1	0	0	0	0
HPC	0.200	0.799	0.001	0	0
HPT	0.166	0.009	0.825	0	0
VSV	0.152	0	0	0.848	0
ps3	0.480	0	0	0	0.520

The performance improves considerably over the detection with sensor measurements alone when the elbowing feature is used as well. Results are summarized in the confusion matrix in Table 5. Specifically, all faults are detected at zero false positive levels upward of 90%, where in particular the ps3 fault improved from 0.52 to 0.951 TP rate. At the same time, all falsely classified faults drop considerably. In both cases, only HPT and HPC faults are misclassified. If accommodation mandates that no misclassifications be made at all, then the results will deteriorate considerably because the particular combination of deterioration, flight envelope, and fault signature are completely overlapping. However, the misclassifications occur in large part only at the smallest fault level with very few misclassifications at the medium fault level and no misclassifications at the large fault level. That also implies that larger faults can be accommodated safely. Results for that case are summarized in Table 4.

Table 4 - Confusion matrix for rapid detection of selected faults (FP forced to zero) with elbow feature

	No fault est.	HPC est.	HPT est.	VSV est.	ps3 est.
No fault	1	0	0	0	0
HPC	0.036	0.959	0.005	0	0
HPT	0.041	0.007	0.952	0	0
VSV	0.090	0	0	0.910	0
ps3	0.049	0	0	0	0.951

There are some fundamental assumptions on which this work was based:

- 1.) The faults, although relatively small, occur as a step change. While that may be a valid assumption for certain failure modes, it is not true necessarily for all failure modes. More gradual fault signatures (“soft faults”) may result in a more gradual fault signature that might not be picked up within the allowable accommodation period – or it may not be picked up at all. Theoretically, the model based approach has an advantage here since it operates on the residuals which tend to amplify the fault signature. On the other hand, normal deterioration may obfuscate the picture further by blurring the line between fault and normal

deterioration. Methods to tackle the deterioration issue – such as tracking filters – may not be able to help here and may in fact impede the detection capability because they might latch on to small changes. On the other hand, accommodation may not necessarily have to deal with soft changes within the same detection window as abrupt faults since the fault magnitude is by definition smaller in the beginning. These issues should be addressed in the future.

- 2.) This work was done at steady state due to simulator restrictions. Technically, the Kalman Filter should be able to track system changes through transients as well. However, there is a real chance that the fault signal gets lost over time. Again, accommodation should have addressed the fault at that time and simple gating mechanisms can prevent losing the fault presence. However, this should not be meant to play down the significance of issues involved in performing fault detection and accommodation during transients, in particular since most fault will likely occur during transients such as take off.
- 3.) While we evaluate the data after the maximum allowable persistence period, no close attention has been paid to the intermediate region between fault initiation and the end of the maximum persistence period. Within that region, the fault logic may be erratic in the sense that it may not initially latch on the correct fault. Some reasons are due to closed loop effects in conjunction with deterioration effects and environmental conditions. This results in dissimilar initial fault directions for a given fault. More work needs to be done to examine this region and the potential effects on accommodation.

Figure 12 shows the individual fault indicators as well as the overall fault indicator for HPC fault. At time $t=0.5s$, a HPC fault is injected. The desired detection window is indicated by the two dotted vertical lines, the first at fault initiation, the second at the maximum allowable detection period. It can be seen that the fault indicator shows the presence of the fault within the desired time window.

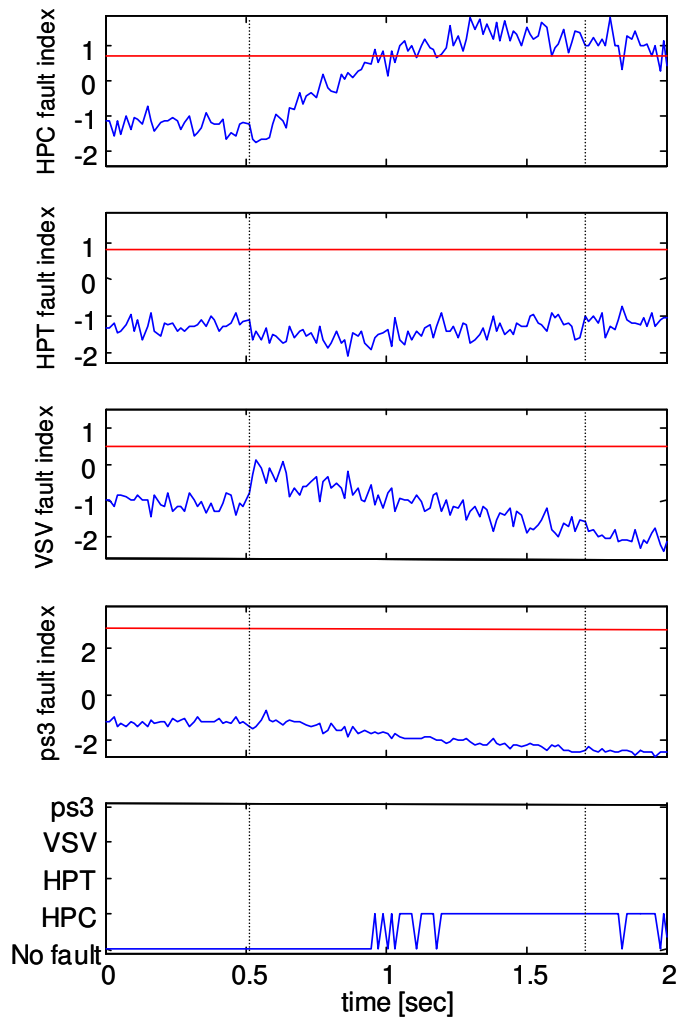


Figure 12 - HPC fault injected at 0.5 s.

Figure 13 shows a VSV fault injected at 0.5 s. It is notable that the detection commences rapidly within only a few samples and well within the required detection interval.

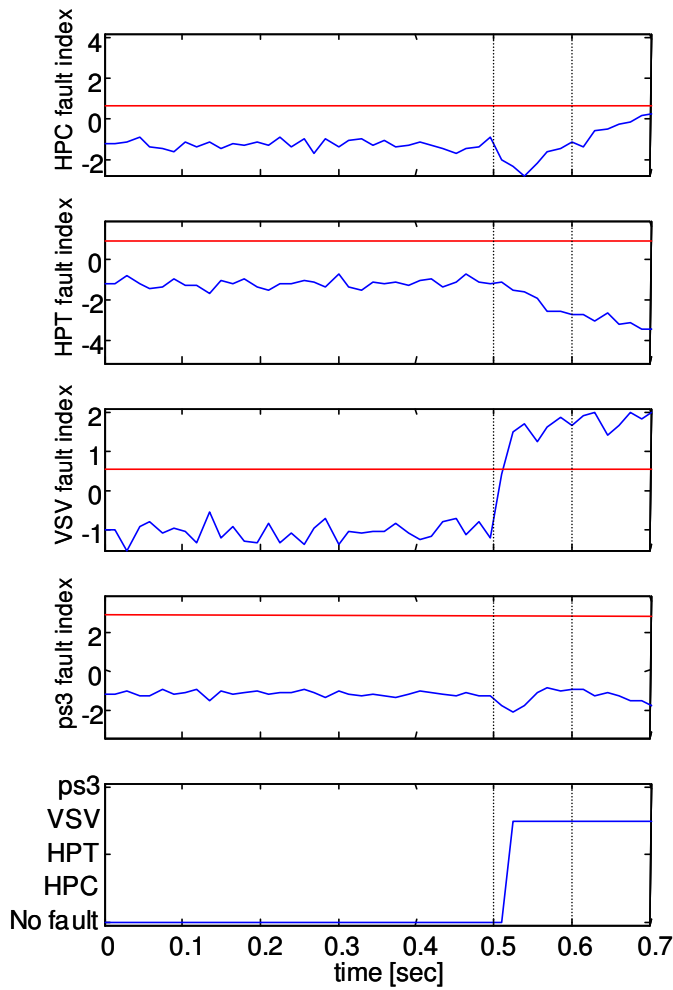


Figure 13 - VSV fault injected at 0.5 sec.

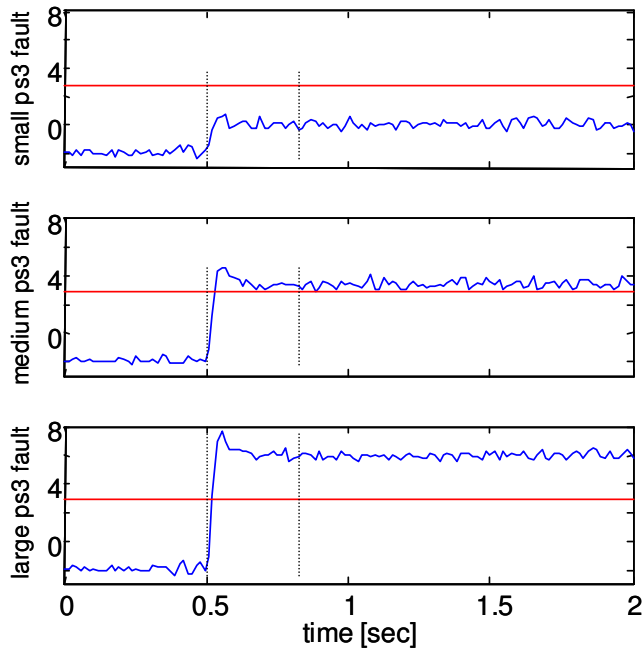


Figure 14 - Effect of fault size on detection capability: ps3 for same flight envelope point

Figure 14 shows the effect of fault size on detection capability. The three graphs show a small, medium, and larger VSV fault. In this particular instance, a fault at a point in the flight envelope with noise conditions was chosen for display that could not be picked up for small faults because it falls under the detection threshold. Recall that small faults are modeled as a 2.5% bias. Medium and large faults are detected as seen in the second and third graph of Figure 14. Not surprisingly, fault magnitude seems to have a strong impact on detection capability.

8. ACCOMMODATION

Although the focus of this paper is not on accommodation, the rapid detection was motivated by the desired to achieve accommodation. We will therefore start the discussion with a few introductory thoughts here and address accommodation more fully in a forthcoming paper.

Each detected fault needs to be appropriately accommodated. Related programs [12, 13] have tried to address aircraft safety in a context of reducing engine vulnerability to combat damage by taking the appropriate control action. While these “survivable engine controls” do not try to predict imminent engine failure as a result of catastrophic damage, intermediate levels of damage is being targeted. In these cases, the engine component efficiencies are reduced beyond normal operating points, and the engine might operate at a greatly reduced power stage. A large number of these types of damage can go undetected by FADEC fault detection logics [12]. The work presented in this paper here addresses fault levels at a much smaller level. This is an advantage because the physical damage is smaller and therefore might be more easily accommodated. On the other hand, the impact on the controller is more subtle as well and which might make accommodation somewhat harder.

Sensor Accommodation

Sensor accommodation can be accomplished the easiest. Once the fault is detected, either the other hardware redundant sensor can be used or, in case of a dual failure, the analytic redundant model can be used.

Actuator Accommodation

The actuator accommodation will depend on the failed position. If the VSV is failed full open it may not be possible to run the engine above an idle condition and maintain operability. But if the VSV fails to an intermediate position one should be able to modify the engine operation through changing operating constraints and opening the

bleed valve to accommodate the failure and keep the engine running. Next, modifications to the control logic need to be defined that will move the references and the constraints to their necessary accommodation values when each of the faults are detected and isolated.

HPT/HPC Accommodation

HPC and HPT accommodation is more involved than the other two categories. In part this is true because HPC and HPT are large, complex systems, that may operate reasonably well in the presence of fault due to controller based accommodation. Accommodation would need to make changes in the controller. However, due to the highly nonlinear nature of the engine controller and the fact that it is implemented as a large collection of computer modules (typically over 100) that employ a variety of one- and two-input tables, switching variables, logical elements, limiters, priority-select logic, etc., the control design space is high-dimensional, highly nonlinear, multimodal, and discontinuous. To find an optimal accommodation, it is very important, yet non-trivial, to define the performance metric in a flexible and non-analytical manner. This is necessary in order to properly account for such diverse requirements as maintaining stall margins above certain limits, minimizing both peak temperatures and the time spent above a certain temperature, and obtaining short rise times in response to changes in demand values. Furthermore, the changes must be accomplished over a wide range of flight conditions and disturbance inputs. Finding a set of parameters to change or controls modes that make a substantial impact is not trivial [13]. Some possible changes include establishing a new exit temperature schedule consistent with the level of airflow loss [12].

9. CONCLUSIONS

We presented a scheme that focused on rapid detection of relatively small aircraft engine gas path faults, sensor faults, and actuator faults. Fault detection typically assumes a sufficiently large fault signature and enough time to come to a reliable decision. However, for a class of safety critical faults detection within a millisecond range is necessary to allow accommodation to avert undesired engine behavior. At the same time, false positives must be avoided to prevent inappropriate control action. To address these issues, several advanced features were developed that operate on the residuals of a model based detection scheme. We showed that these features pick up system changes reliably within the desired time period. A bank of binary classifiers determines the presence of the fault as determined by a quasi maximum likelihood hypothesis test. We showed performance results for four different faults at various levels of severity (small, medium, and high) and throughout the entire flight envelope.

Future work should address the issue of soft faults and faults occurring during transients. Future work should also take advantage of the engine model and use calculated engine parameters as input to the decisioning model. These parameters should include the turbine and compressor efficiencies as well as other quality parameters.

ACKNOWLEDGEMENT

This work was supported by NASA grant NAS3-01135 Task # 3. The authors greatly acknowledge the support of Don Simon and Tak Kobayashi (both with NASA). The authors also acknowledge the help in the fault selection by Aaron Dentinger (with GE Global Research).

REFERENCES

- [1] AIA/AECMA Project Report on PSM+ICR, http://www1.faa.gov/certification/aircraft/engine_psm+icr.doc, 1998.
- [2] Christine M. Belcastro, Celeste M. Belcastro, "Application of failure detection, identification, and accommodation methods for improved aircraft safety", *Proceedings of the American Control Conference*, Vol. 4, 25-27, pp. 2623–2624, June 2001.
- [3] J. J. Gertler, "Survey of model based failure detection and isolation in complex plants," *IEEE Control Systems Magazine*, pp. 1–11, December 1988.
- [4] W. Yan, K. Goebel, and J. Li, "Classifier Performance Measures in Multi-Fault Diagnosis for Aircraft Engines", *Proceedings of SPIE, Component and Systems Diagnostics, Prognostics, and Health Management II*, vol. 4733, pp. 88-97, 2002.
- [5] Egan, J.P., *Signal detection theory and ROC analysis, Series in Cognition and Perception*. New York: Academic Press, 1975.
- [6] Bradley, A.P., "The use of the area under the ROC curve in the evaluation of machine learning algorithms", *Pattern Recognition*, Vol.30, No.7, pp1145-1159, 1997.
- [7] Downey, T.J., Meyer, D.J., Price, R.K., and Spitznagel, E.L., "Using the receiver operating characteristic to assess the performance of neural classifiers", *IJCNN '99 – International Joint Conference on Neural Networks*, Vol.5, pp3642-3646, 1999.
- [8] Takahisa Kobayashi, Donald L. Simon, "Application of a Bank of Kalman Filters for Aircraft Engine Fault Diagnostics", *Proceedings of ASME Turbo Expo 2003, Power for Land, Sea, and Air*, June 16-19, 2003, Atlanta, Georgia, USA, GT2003-38550, 2003.
- [9] V.N. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.
- [10] Cawley, G. C. "Support Vector Machine Toolbox v0.50 beta, <http://theoval.sys.uea.ac.uk/~gcc/svm/toolbox>, University of East Anglia, School of Information Systems, Norwich, Norfolk, U.K. NR4 7TJ, 2000.

- [11] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization", *Advances in Kernel Methods - Support Vector Learning*, (Eds) B. Scholkopf, C. Burges, and A. J. Smola, MIT Press, Cambridge, Massachusetts, chapter 12, pp 185-208, 1999.
- [12] C. E. Frankenberger, III "Survivable Engine Control Algorithm Development (SECAD)", *Proceedings IEEE Aerospace Conference, 2002. Volume: 6 ,9-16. pp. 6-3015 - 6-3020*, 2002.
- [13] Matthew Wiseman, Ten-Huei Guo, "An investigation of life extending control techniques for gas turbine engines", *Proceedings of the 2001 American Control Conference*, Vol. 5 , 25-27, pp. 3706 –3707, 2001.

BIOGRAPHY

Kai Goebel received the degree of *Diplom-Ingenieur* from the Technische Universität München, Germany in 1990. He received the M.S. and Ph.D. from the University of California at Berkeley in 1993 and 1996, respectively.



Dr. Goebel joined General Electric's Corporate Research and Development facility in Schenectady, NY in 1997 as a computer scientist after working as a visiting postdoctoral fellow at UC Berkeley from 1996 to 1997. He has carried out applied research in the areas of artificial intelligence, soft computing, and information fusion. His research interest lies in advancing these techniques for real time monitoring, diagnostics, and prognostics. He has fielded numerous applications for aircraft engines, transportation systems, medical systems, and manufacturing systems.

Dr. Goebel is an adjunct professor of the CS Department at Rensselaer Polytechnic Institute (RPI), Troy, NY, since 1998 where he teaches classes in Soft Computing.

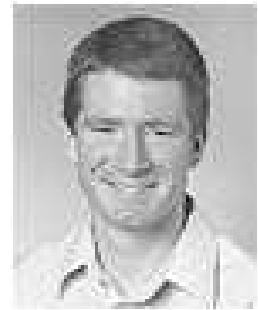
Neil Eklund received B.S. in 1991, two M.S. degrees in 1998, and a Ph. D. in 2002, all at the Rensselaer Polytechnic Institute.



Dr. Eklund was a research scientist at the Lighting Research Center from 1993 to 1999. He was in the network planning department of PSINet from 1999 to 2002, before joining General Electric Global Research in Niskayuna, NY in 2002. He has worked on a wide variety of research projects, including early detection

of cataract using intraocular photoluminescence, multiobjective bond portfolio optimization, paint inspection lighting optimization, multiobjective optimization of permanent virtual circuits for a frame relay service, and the influence of lighting on office worker performance. His current research interests include color measurement, color and spatial vision, genetic algorithms for multiobjective optimization, and applied soft computing.

Brent Brunell received a B.S. in 1994 from Rensselaer Polytechnic Institute and a M.S. degree from Massachusetts Institute of Technology in 2000.



Mr. Brunell was on the Technical Leadership program at General Electric Aircraft Engines in Lynn Massachusetts from 1995 to 2000. He joined General Electric Global Research in Niskayuna NY in 2000. He has been involved in several projects related to control and diagnostics of turbo machinery. His current role is project leader for advanced model-based controls and diagnostics with responsibilities that include leading the NASA Aviation Safety Program. His current research interests are dynamic modeling, model-based control and model-based diagnostics for real time applications.