Proceedings of GT2005
ASME Turbo Expo 2005: Power for Land, Sea and Air
June 6-9, 2005, Reno-Tahoe, Nevada, USA

GT2005-68300

INTEGRATED IN-FLIGHT FAULT DETECTION AND ACCOMMODATION: A MODEL-BASED STUDY

Randal T. Rausch, Kai F. Goebel, Neil H. Eklund, Brent J. Brunell
GE Global Research, Niskayuna, NY 12309
{rausch, goebelk, eklund, brunelbr}@research.ge.com

ABSTRACT

In-flight fault accommodation of safety-critical faults requires rapid detection and remediation. Indeed, for a class of safety critical faults, detection within a millisecond range is imperative to allow accommodation in time to avert undesired engine behavior. We address these issues with an integrated detection and accommodation scheme. This scheme comprises model-based detection, a bank of binary classifiers, and an accommodation module. The latter biases control signals with pre-defined adjustments to regain operability while staying within established safety limits. The adjustments were developed using evolutionary algorithms to identify optimal biases off-line for multiple faults and points within the flight envelope. These biases are interpolated online for the current flight conditions. High-fidelity simulation results are presented showing accommodation applied to a high-pressure turbine fault on a commercial, high-bypass, twin-spool, turbofan engine throughout the flight envelope.

INTRODUCTION

Faults affecting aircraft safety must be detected quickly to avoid undesired events such as engine surge/stall events, power loss, severe vibrations, and loss of thrust control. These events, in addition to inappropriate crew response, may lead to accidents [1]. To reduce the number of accidents NASA established the Aviation Safety Program [2]. Within the scope of that program, this paper investigates a number of faults for which in-flight accommodation is considered.

The goal of a fault accommodation system is to regain operability and maintain commanded thrust after a fault has been detected by biasing aircraft control signals. Aircraft controllers are highly sophisticated systems with strict safety limits based on that many years of design experience. To stay within these safety confines, the control signal biases are limited to established bounds.

Figure 1 shows the top-level conceptual architecture for the detection and accommodation methods used. A model-based approach was employed to aid the diagnostics using 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 output of the diagnostics triggers the accommodation (actuator control adjustments). The top-level inputs to the system are the pilot input throttle resolver angle (TRA), altitude (ALT), Mach number (XM), delta from ISO standard day ambient temperature (DTAMB), as well as engine variation and health parameters (P). Data were generated for a large 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. The modeling and diagnosis techniques are discussed elsewhere [3] in more detail. This paper will focus primarily on the accommodation. In section II, we will first briefly introduce the model and detection schemes used, followed by a short discussion of the faults selected. Section III discusses fault accommodation; section IV concludes with a few final remarks.

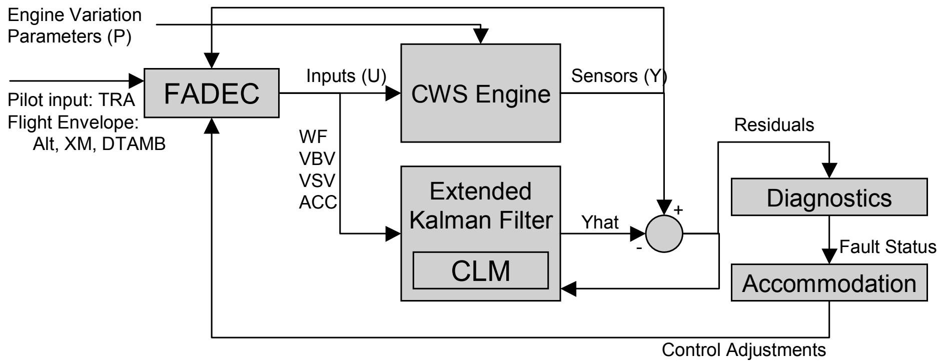

Figure 1: High-level architecture of detection and accommodation strategy

NOMENCLATURE

ACC	=	Active Clearance Control
ALT	=	Altitude
CLM	=	Component Level Model (software) of high bypass, two rotor, turbofan
CWS	=	Cycle WorkStation software model of high bypass, two rotor, turbofan
DEC	=	Digital Engine Control
DTAMB	=	Delta from ISO standard day temperature
EGT	=	Exhaust Gas Temperature
EKF	=	Extended Kalman Filter
FADEC	=	Full Authority Digital Electronic Control
FE	=	Flight Envelope
FP	=	False Positives
FN	=	Net Thrust
GA	=	Genetic Algorithm
GAOT	=	Genetic Algorithm Optimization Toolbox
HPC	=	High-pressure compressor
HPT	=	High-pressure turbine
PS3	=	HPC exit static pressure
SMB	=	Booster stall margin
SMC	=	Core stall margin
SMF	=	Fan stall margin
SVM	=	Support Vector Machine
TRA	=	Throttle Resolver Angle
VBV	=	Variable Bleed Valve
VSV	=	Variable Stator Vane
WF	=	Fuel Flow
XM	=	Mach number

SYSTEM MODELING, DETECTION, AND FAULTS SELECTED

Model

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 cycle workstation (CWS) is a high fidelity physics-based aircraft engine model, which acts as the real engine ("truth"). Variation, deterioration, and fault models can be injected into the CWS model. A simplified, physics-based, component level model (CLM) is used as the embedded model in conjunction with the EKF to aid in fault diagnostics. The CLM takes less time to run and is less accurate than CWS.

The 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 [4]. 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. Residuals are used to detect changes in plant behavior: they should be near zero in fault-free case and non-zero when a fault has occurred. The detection process evaluates the residuals and monitors if and where a fault has occurred.

Source of Variation

In order to realistically represent engine behavior, sources of variation are included as simulation inputs. Specifically, we consider variability between engines, aging, and 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 adds a normally distributed random value (based on measurements of a sample of the engine population) to the nominal value of each parameter. The standard deviation of the parameter variation values ranged from 0.0 to 1.1%.

Deterioration

A deterioration model was created for both the CWS and CLM 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 affected by deterioration. The deterioration model used in the CWS engine model is a uniform random 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 (±) and modeled as normal distribution with a 2-sigma maximum variation.

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 compressor exit pressure (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 Fig. 2 and the parameters and magnitudes varied for each fault are given in Table 1.

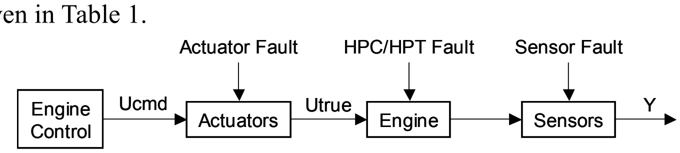

Figure 2: Fault block diagram

Fault Type	Model Parameter Changed	Change from Nominal		
		Small Fault	Medium Fault	Large Fault
HPC Fault	Efficiency	-1.5%	-3%	-5%
	Flow	-1.5%	-3%	-5%
HPT Fault	Efficiency	-1.5%	-3%	-5%
	Flow	1.5%	3%	5%
VSV Fault	Bias	0.8 deg	2.5 deg	5.0 deg
Ps3 Fault	Bias	-7 PSI	-19 PSI	-30 PSI

Table 1: Fault Model Parameters

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. The allowed detection time is based on criticality determined by controls designers. Detection must be completed within a time frame between 106ms and 1.2s, depending on the fault type (Table 2).

Fault	Allowed Detection Time
HPC	1.2 s
HPT	1.2 s
VSV	106 ms
PS3	340 ms

Table 2: Allowed Fault Detection Time

Diagnostics

Specifications about the false positive and true positive rates (or false negative or true negative rates) guide classifier design. 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 (e.g., some military vs. civilian applications).

Model-based information is the foundation of many diagnostic strategies, ranging from simple thresholding to Kalman Filter banks [5]. 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 [6, 7].

The overall classification approach followed the scheme shown in Fig. 3. 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. After a feature selection process, the features are subjected to the 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.

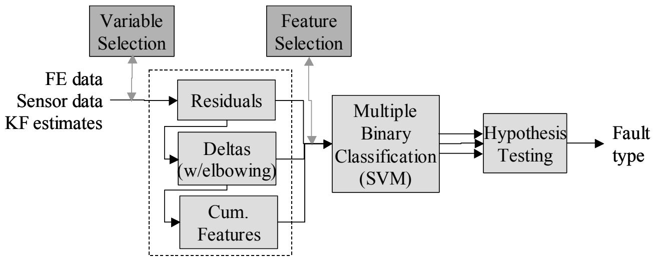

Figure 3: Fault Classification Scheme

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 a mandated zero false positive rate (to avoid any inappropriate accommodation), results are as listed in Table 3 that shows the confusion matrix for the output of the maximum likelihood fault selection logic.

All faults are detected at zero false positive levels upward of 90%. 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.

	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

Table 3: Confusion matrix for rapid detection of selected faults (FP forced to zero) with elbow feature

FAULT ACCOMMODATION

The following sections detail the accommodation of a high-pressure compressor (HPC) fault. The method described gives good results for HPT and VSV faults [3] however; only the HPC results are discussed in this paper.

A successful accommodation strategy will protect steady state operability, limit maximum temperature, and ensure adequate thrust. The critical measures of engine operability are the stall margins for the fan, booster, and high-pressure compressor (SMF, SMB, SMC, respectively). The controller is designed, among other things, to protect minimum stall margins required for safe operation at different points in the flight envelope. The accommodation goal is to achieve steady state stall margins in a faulted engine that are equal to or greater than the pre-fault values.

Recall that the high-pressure compressor fault (such as a blade failure) is modeled as a decrease in the efficiency and flow through the HPC. Small, medium, and large faults are modeled respectively to 1.5%, 3%, and 5% changes in efficiency and flow. The fault occurs as a step change over a 0.01 second interval.

Figure 4 represents a CWS simulation of a large HPC fault occurring at 0.5 seconds and subsequent results over the next 7 seconds. This response is determined by a standard controller without fault accommodation algorithms. The controller attempts to recover fan speed and thus, thrust. As a result of the fault and subsequent controller action, the booster and compressor stall margins drop below their designed (safe) values.

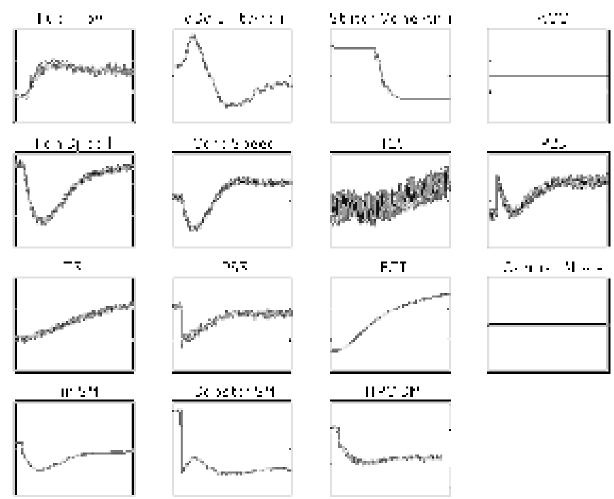

Figure 4: Engine Parameters for Single HPC Fault Simulation

While Fig. 4 shows aircraft operation at a particular point in the flight envelope, one cannot necessarily extrapolate from that to operation at other points in the flight envelope. Figure 5 encompasses an aggregation of many flight envelope points that were chosen to cover the aircraft flight envelope. The histogram shows the frequency of occurrence of a particular percentage change in each parameter. The frequency is calculated using this following equation for 458 flight envelope points:

$$\frac{\vec{Y}_{t_{final}} - \vec{Y}_{t_0}}{\vec{Y}_{t_0}} \qquad (1)$$

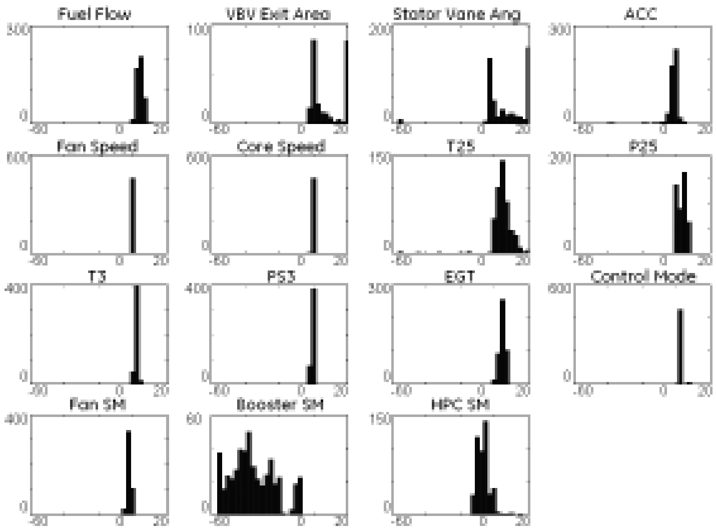

Figure 5: Histogram Representing HPC Fault Trends

As expected, the high-pressure compressor fault causes a drop in the compressor stall margin (SMC) and a significant drop in the booster stall margin (SMB). These values must be restored for safe operation of the aircraft.

Solution Description

The nonlinearity of the FADEC and the engine requires a nontraditional optimization technique to find optimal, or even feasible, accommodation strategies. Genetic algorithms (GAs) are well-suited for this task. Genetic algorithms are a general-purpose optimization method based on the theory of natural selection. GAs make no assumptions about the search space, so they can be applied to almost all optimization problems [8]. However, GAs exchange applicability for speed - although they can be used on a wide variety of problems, they are typically slower to converge to a solution than algorithms designed for a specific problem. The GA Optimization Toolbox (GAOT) [9] was used to implement the GA.

This work assumes that the stall margin measures, SMF, SMB, and SMC, are acceptable measures of operability. This implies these faults are not concerned with lean blow out and that the calculated variables are an accurate representation of the actual margin between the current operating point and the stall line. Finally, it is assumed for steady state evaluation that the rate at which the actuators move to their accommodated values does not matter.

The optimization system takes the outputs of the engine simulation (SMF, SMB, SMC, FN) as inputs. The GA then generates control adjustments, which the FADEC translates to inputs for the engine simulation.

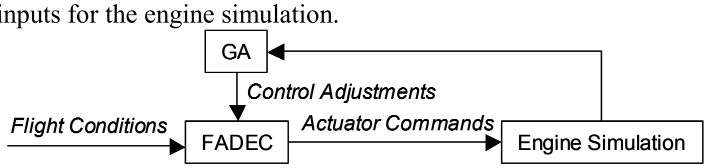

Figure 6: Optimization Architecture

GA Parameters

Run time is a critical factor in this genetic algorithm design, as the CWS engine simulation runs more slowly than real time. In order to maximize the number of runs while still converging on a good solution, the optimization is designed to take approximately 6 hours. The simulations start with a random initial population of 30 members and finish after 30 generations. A crossover rate of 60% and a mutation rate of 5% were chosen.

Solution Representation

Most of the variables that represent a solution are bounded and continuous. These continuous gene values and their bounds are listed in Table 4. Two of the variables correspond to discrete switches, namely:

- Auxiliary Compressor Bleed Switch
- Cowl Anti-Ice bleed Switch

For these switches, a floating-point chromosome representation was used where the values were rounded off. Each individual in the population is represented with a chromosome containing floating-point representations of the values in Table 4.

Solution Variable (Gene)	Minimum Value	Maximum Value
VBV Adjustment Adder	-100	100
VSV Adjustment Adder	-50	50
Max Core Speed Adder	0	1060
Horsepower Extraction	0	Original value
ACC Multiplier	0	100
ACC Adder	0	1
Throttle	97% of original value	103% of original value
Aux Bleed Switch	0	24
Cowl Anti-Ice Switch	0	1

Table 4: Continuous solution variables and bounds

Fitness Function

In every generation, each member of the population is evaluated (each combination of control adjustments is simulated) to determine how good the solution is, and the fitness is appended to the chromosome. The goodness, or fitness, of a solution is based upon the operability and performance measures, namely:

- Fan Stall Margin (SMF)
- Booster Stall Margin (SMB)
- Compressor Stall Margin (SMC)
- Exhaust Gas Temperature (EGT)
- Net Thrust (FN)

As previously defined, an optimal fault accommodation strategy obtains thrust and stall margin levels similar to those prior to the fault's occurrence. A strategy is penalized for exceeding maximum temperatures. To quantify a strategy, J is maximized, where:

$$J = -\sum \begin{bmatrix} Weight_{SM12} * \left(\frac{\Delta SMF}{SMF_{t_0}} \right) \\ Weight_{SM2} * \left(\frac{\Delta SMB}{SMB_{t_0}} \right) \\ Weight_{SM25} * \left(\frac{\Delta SMC}{SMC_{t_0}} \right) \\ Weight_{FN} * \left(\frac{\Delta FN}{FN_{t_0}} \right) \end{bmatrix} - \left(\max \left\{ 0, \quad EGT_{t_{final}} - Limit_{EGT} \right\} \right)^{Exp}$$

$$note: \quad \Delta x = \left| x_{t_0} - x_{t_{final}} \right|$$

and

SM12_WEIGHT = 1,
SM2_WEIGHT = 2,
SM25_WEIGHT = 2,
FN_WEIGHT = 3,
EXPONENT = 2.

Test Points

Table 5 lists a sample of the flight envelope points for which the GA was run.

Run	HPC Fault Level	TRA	Alt (feet)	XM
1	Small	Low	Very Low	Low
2	Large	High	Med	High
3	Small	Med	Low	High
4	Large	Med	Very Low	Low
5	Small	Low	High	High
6	Large	Low	Med	Low
7	Small	Med	Med	Low
8	Large	Med	High	High
9	Large	Med	Low	Med
10	Medium	High	Low	Med
11	Medium	Med	High	Med
12	Medium	Med	Med	Very High
13	Small	Med	Low	Med
14	Medium	Very Low	Low	Med
15	Medium	Med	Very Low	Med
16	Medium	Med	Low	Very Low
17	Medium	Med	Low	Med

Table 5: Flight Envelope Test Points

Accommodation Results

At each point, a nominal, healthy engine was simulated. At 0.5 seconds, a fault was inserted and 2 seconds later, the adjustment parameters were changed to the accommodated variables determined by the genetic algorithm. The runs continued for 20 seconds to allow transients to settle and achieve the steady-state accommodated measures for operability and performance. The temporal responses for a number of test points are shown in figures 7 - 12, illustrating Fan Stall Margin, Booster Stall Margin, Compressor Stall margin, and Net Thrust. The six points shown are dispersed throughout the flight envelope and have different fault severity and throttle settings. In all cases, the stall margins and thrust are restored to their pre-fault values.

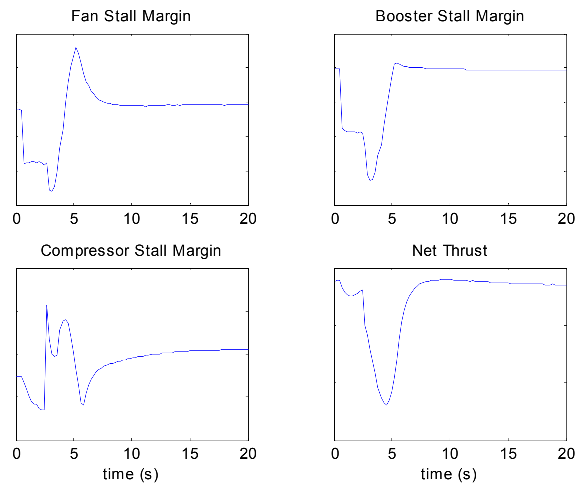

Figure 7: Med. Fault (Med TRA, High Alt, Med Mach)

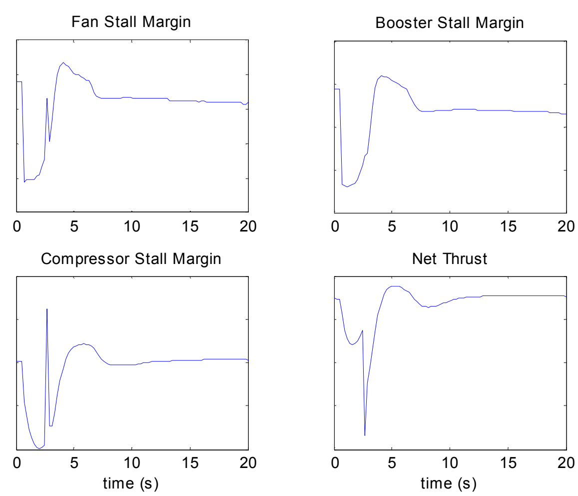

Figure 8: Med. Fault (Very Low TRA, Low Alt, Med Mach)

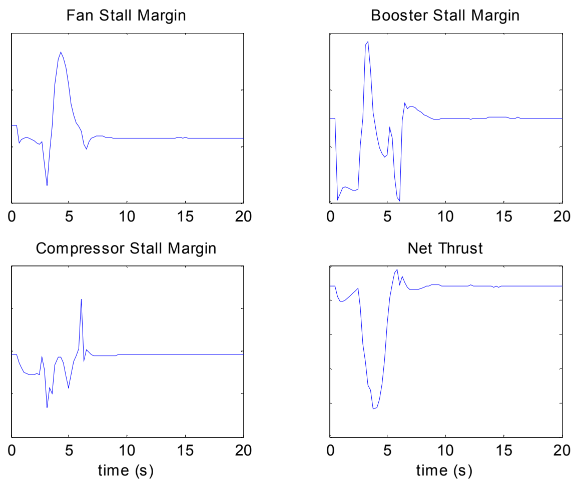

Fig 9: Med. Fault (Med TRA, Med Alt, Very High Mach)

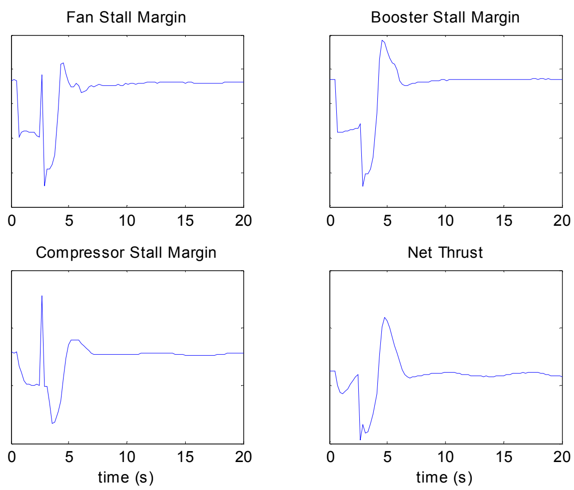

Figure 10: Small Fault (Med TRA, Low Alt, Med Mach)

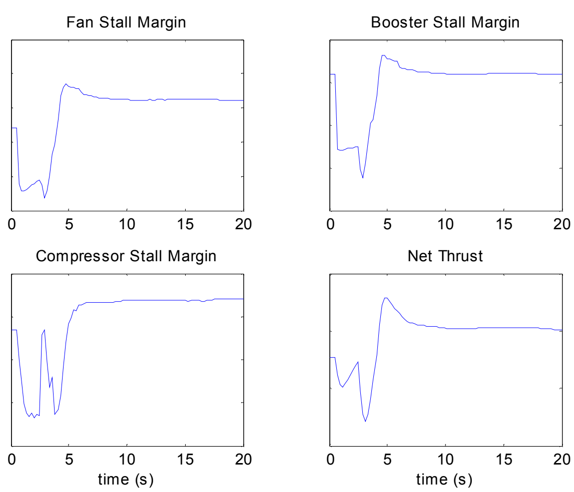

Fig 11: Med. Fault (Med TRA, Low Alt, Very Low Mach)

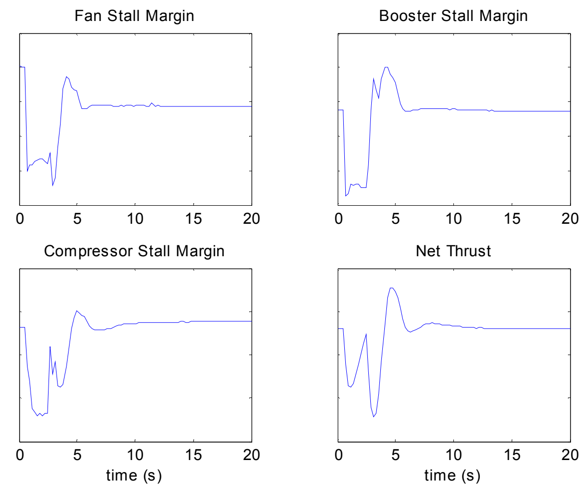

Figure 12: Large Fault (Med TRA, Low Alt, Med Mach)

Solution Analysis

Acceptable accommodation solutions have been found at all flight envelope points tested. This is cautiously taken as indication that certain hardware faults can be accommodated with control adjustments. Some parameters, such as the max core speed adder, are not necessary for accommodation of all points. This is evident in the seemingly random solution provided for these variables from the GA. With the exception of fault level, no statistically significant correlation is found between the flight envelope points and the optimal control adjustments. There are several implications:

- This makes *a priori* scheduling the fault accommodation very difficult. The current optimization strategy is much too slow to run real-time on the FADEC, so the accommodation strategy must be predetermined. This can be accomplished by solving the problem for corner and center points with the minimum number of adjustments possible. These points can be included in a standard schedule where in-between points are calculated using a weighted average of adjustments corresponding to the N closest design points. The weights are proportional to the inverse distances to the design points [3].

- The plurality of optimal solutions suggests that effects of a multivariate nature are not fully taken into account, which leads to apparent lack of solution smoothness. Specifically, the GA generally finds the optimal solution for accommodation which does not necessarily correspond to the best strategy consistent with scheduling. Including a reward in the fitness function for solution similarity to the accommodation strategy of a nearby point in the flight envelope can take care of this issue. Other penalties can be employed such as "low accommodation effort" [3].

The genetic algorithm approach to the problem of finding control adjustments to accommodate faults is successful where hand tuning and derivative approaches have been unable to find suitable results. However, the runtime of the genetic algorithm is orders of magnitude larger than real-time. Faster solving methods exist, but none have yet been successfully employed in real-time to accommodate all flight envelope points. The dual approach of offline GA accommodation and online interpolation between design points has been shown to be successful.

In some cases, the accommodation strategy causes a slight increase in thrust. While this was a permissible change in the experiment (and was bounded to small changes), in practice, this may be just as undesired as loss of thrust. Weighting the importance of thrust change in the fitness function should remedy this problem. Additionally, running the GA for a larger number of generations may be necessary for solution convergence.

Some strategies caused an initial loss of stall margin before converging on the steady state values. While this would be unacceptable in flight, the transient is ignored in the steady state evaluation. Appropriately scheduling the speed at which the actuators slew to the accommodated positions should remedy this problem.

Future Work

This method of discovering accommodation strategies should be very successful in the transient realm. The stall margin numbers are generally used to protect an engine from stall during transient operation. This same technique can easily be used to construct a transient strategy. Steady state accommodation is a necessary first step, but a complete fault accommodation strategy also schedules accommodation transiently. In fact, there are many more handles to adjust, for example, the acceleration and deceleration schedules as well as the actuator gains. The large number of possible accommodation parameters needs to be balanced with the need for solution smoothness as discussed earlier. This paper explored a strategy that relied on first principle understanding of experts to reduce the parameters to a “good” set. A formal method should be explored that guides the user in making these choices.

Other extensions include the investigation of complete systems comprising engines and subsystems. Here, the possible interactions will pose considerable challenges to the expected increase in complexity. An integrated approach might be explored that ensures the overall stability of the system.

Lastly, the accommodation should be examined as part of a comprehensive health management strategy that includes the secondary considerations beyond the immediate operability concerns. While the accommodation strategies presented here do potentially avert undesired system behavior, they do not address the root cause. Dealing with the required maintenance should be considered as part of the overall decision making process, taking into account impact on logistics, parts availability, flight schedule adjustments, shop loading, long term prognostics, etc.

CONCLUSIONS

A method for integrated fault detection and accommodation in turbofan engines via control adjustments was presented. For detection, a scheme was employed that focused on rapid detection of relatively small aircraft engine faults. A bank of binary classifiers established the presence of the faults as determined by a quasi-maximum likelihood hypothesis test within a millisecond range while, at the same time, avoiding false positives. Next, a genetic algorithm was used to identify the control adjustments necessary to maintain operability and performance. Solutions for this highly nonlinear problem were found for all fault levels and operating points tested. A dual method to schedule design adjustments across the flight envelope and interpolate between them for unscheduled operating points proved to be successful. We showed performance results for a high-pressure compressor fault at various levels of severity (small medium, high) and at different points in the flight envelope.

ACKNOWLEDGMENTS

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).

REFERENCES

[1] AIA/AECMA Project Report on PSM+ICR, http://www1.faa.gov/certification/aircraft/engine_psm+icr.doc, 1998.

[2] Belcastro, Christine M., and Belcastro, Celeste M., 2001, “Application of Failure Detection, Identification, and Accommodation Methods for Improved Aircraft Safety,” Proceedings, American Control Conference, Vol. 4, 25-27, pp. 2623 –2624.

[3] Rausch, R., Viassolo, D., Kumar, A., Goebel, K., Eklund, N., Brunell, B., Bonanni, P., 2004, “Towards In-Flight Detection and Accommodation of Faults in Aircraft Engines”, Proceedings, American Institute of Aeronautics and Astronautics, 2004.

[4] Gertler, J. J., 1998, “Survey of Model-Based Failure Detection and Isolation in Complex Plants,” IEEE Control Systems Magazine, pp. 1–11, December 1988.

[5] Kobayashi, T., Simon, D. L., 2003, “Application of a Bank of Kalman Filters for Aircraft Engine Fault Diagnostics”, Proceedings, ASME Turbo Expo, Power for Land, Sea, and Air, June 16-19, 2003, Atlanta, Georgia, USA, GT2003-38550.

[6] Vapnik, V. N., 1995, The Nature of Statistical Learning Theory, Springer-Verlag, New York.

[7] Cawley, G. C., 2000, "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.

[8] Goldberg, D., 2002, "The Design of Innovation: Lessons from and for Competent Genetic Algorithms", Kluwer Academic Publishers, Norwell, MA.

[9] Houck, C., Joines, J., and Kay, M., 1995, "A Genetic Algorithm for Function Optimization: A Matlab Implementation", NCSU-IE TR 95-09.