

Improved Engine Health Monitoring Using Full Flight Data and Companion Engine Information

Allan J. Volponi and Liang Tang
Pratt & Whitney

ABSTRACT

Engine module performance trending and engine system anomaly detection and identification are core capabilities for any engine Condition Based Maintenance system. The genesis of on-condition monitoring can be traced back nearly 4 decades, and a methodology known as Gas Path Analysis (GPA) has played a pivotal role in its evolution. GPA is a general method that assesses and quantifies changes in the underlying performance of the major modules of the engine (compressors and turbines) which directly affect performance changes of interest such as fuel consumption, power availability, compressor surge margins, and the like. This approach has the added benefit in that it enables anomaly detection and identification of many engine system accessory faults (e.g., variable stator vanes, handling and customer bleeds, sensor biases and drift). Legacy GPA has been confined to off-board analysis of *snapshot* data averaged over a stable flight conditions when the engine is in steady state operation. This discrete data point approach, while fairly accurate and repeatable, comes with a price tag which is a time latency to detect (and subsequently isolate) a faulty condition. In this paper, we explore the use of streaming full flight data which includes both transient and steady state operation. This type of data stream, when properly processed, allows faster anomaly detection, credible fault persistency checks and timely fault identification. This paper outlines the use of companion engine data to achieve a cross-wing comparative diagnostic method.

CITATION: Volponi, A. and Tang, L., "Improved Engine Health Monitoring Using Full Flight Data and Companion Engine Information," *SAE Int. J. Aerosp.* 9(1):2016, doi:10.4271/2016-01-2024.

INTRODUCTION

For nearly four decades, GPA has provided a valuable tool for tracking long term performance deterioration of the major modules of a turbofan gas turbine engine in support of the on-condition maintenance philosophy that has evolved during that period. As such, it has been useful in assessing general engine health, in support of fleet-wide engine trending and depot maintenance logistics, as airlines transitioned from *scheduled* maintenance based on hours flown to a more Condition Based Maintenance (CBM) operation based on individual engine health. The original formulations have evolved and expanded to include a diagnostic capability for the Fault Detection and Isolation (FDI) of engine sub-system components whose deterioration or failure result in observable performance changes in the engine's gas path and as such has had an impact on line maintenance practices as well. These sub-systems include engine bleed systems, actuated variable geometry, turbine clearance control, instrumentation, and the like. The literature abounds with descriptions of engine monitoring data capture systems, applicable algorithms for the analysis of the gathered data as well as the perceived cost benefit of these practices. The interested reader can find a summary discourse and extensive bibliography on the subject in references [1,2,3], and as such, we will not repeat a detailed description of legacy methods in this paper. However, to familiarize the reader, who may be unacquainted with these methods, we will provide a short summary

of the salient concepts in order to motivate the central theme of this paper, which is the extension of these methods to leverage full flight data in the emerging world of *big data*.

The format of the paper begins with a summary section on historical perspective, the benefits derived from considering full flight data, the technical challenges imposed, the motivation for utilizing companion engine data and finally a particular approach that utilizes this *cross-wing* strategy to mitigate these challenges. Simulation studies will be discussed in detail along with pertinent metrics to measure the efficacy of this approach.

HISTORICAL PERSPECTIVE

Engine diagnostic practices are almost as old as the gas turbine itself. Monitoring and analysis methods have progressed in sophistication over the past decades as the gas turbine evolved in form and complexity. Beginning with mainly empirical methods centered on monitoring the mechanical integrity of the machine, the evolution of engine diagnostics has benefited from advances in sensing, electronic monitoring devices, increased fidelity in engine modeling and analytical methods. The principal motivation in this development is, not surprisingly, cost. The ever increasing cost of fuel, engine prices, spare parts, maintenance and overhaul, all contribute to the cost of an engine over its entire life cycle. Prognostics and diagnostics can be viewed as a means to mitigate risk in decisions that impact

operational integrity. This can have a profound impact on safety, such as In-Flight Shut Downs (IFSD) as well as the economic impact caused by Unscheduled Engine Removals (UERs), part life, maintenance and overhaul and the overall logistics of maintaining an aircraft fleet.

Engine Health Monitoring began in earnest with the advent of commercial high bypass turbofan engines introduced in the 1970s. Onboard data collection systems initially focused on collecting data from the existing instrumentation available at the time which was primarily gas path sensors (speeds, temperatures and pressures) that were resident on the machine for engine control purposes. The development of Gas Path Analysis began in earnest as commercial airlines and engine manufacturers used this data to trend engine performance. The data used for this purpose was stable aircraft cruise data that would be averaged (over a short time window) to produce a *snapshot* of various engine and flight parameters for subsequent analysis and trending off-board.

Classical Gas Path Analysis (GPA) was targeted to track long term engine module performance degradation as measured by thermodynamic changes in efficiency and flow parameter for each major module (compressors and turbines). It was later modified to accommodate additional functionality such as fault event detection and isolation to address abrupt events and failures such as FOD, DOD, bleed leaks & failures, variable geometry anomalies, actuator & instrumentation faults, and the like. This legacy gas path fault detection and isolation relied on the ability to recognize abrupt shifts in steady state snapshot cruise data. Parameter Δ s from a (fleet) normalized reference model provided the raw ingredients to enable both the detection and identification process. Long and short term averages of the parameter Δ s were compared to determine if a divergence was present (indicating that an abrupt shift in the parameter had occurred) and a *persistence* check was made to determine that the shift was real and not due to an operational excursion. If a persistent shift was observed, parameter Δ s would be calculated between the initial (pre-shift) Δ value and its final (shifted) Δ value to form a *signature* across all dependent measured gas path parameters which in turn could be used to isolate the underlying fault through a pattern matching strategy under the hypothesis of a *single fault* scenario. The details of this process can be found in [4].

Over the subsequent 4 decades since its inception, GPA has been the topic of much research which has extended its capabilities from its early formulation. Its success has contributed to its longevity, and it remains as one of the central techniques supporting health management to this day.

FULL FLIGHT DATA

Since the inception of engine condition monitoring four decades ago, the on-board monitoring hardware has dramatically improved in both capacity and capability. With enhanced multi-processor CPU throughput and massive storage capacity at hand, *both* on-board analyses of engine data is within reach as well as the ability to provide streaming data for off-board processing to ultimately replace the legacy snapshot analyses in current practice. The latter path is part of the *big data* phenomena we hear so much about. The choice of

deploying on-board or supporting the *big data* approach has many considerations, e.g., implementation complexities, certification issues, data management issues, analysis maturation, life cycle cost, to name a few. What is important, from the authors' perspective, is that the analyses methodologies created should be configured to be as generic as possible to allow either on-board or off-board implementation with minimal change. If we restrict ourselves, for the purpose of this paper, to gas path data, what does a full flight perspective offer us beyond what the legacy snapshot approach already provides, given its stated success and longevity? The simple answer is, *time*, i.e., a *reduction in the time latency* to provide credible fault detection and isolation with minimal false alarms and missed detections. Figure 1 illustrates the problem faced with *snapshot* data. The area in the plot that is shaded in pink indicates the observational emergence of an event, i.e. rapid deterioration of a module or failure (full or partial) of an engine sub-system (such as a bleed) or instrumentation fault. The portions of the plot shaded in yellow signify the effect on the measurement delta due to long term (gradual) component degradation. Recall, that each point in this plot would signify a single (averaged) snapshot data point taken from steady state cruise flight and the time axis is measured in flights (i.e., days rather than seconds). Thus, such a plot would gradually emerge with each subsequent flight and that a decision would have to be made as each data point was processed. In this hypothetical plot, there are 4 data points depicted in the pink area. The event detection problem must weigh the consequences of a false alarm versus the consequences of waiting too long to be assured that an event has actually occurred versus a set of rouge points (outliers). This is the general problem of determining *persistence*. The detection methodology employed must minimize false alarms due to outliers while reducing the risk of catastrophic failure which might occur on subsequent flights if the underlying fault was critical and persistence was not established fast enough to allow its determination.

The use of full flight (streaming) data offers the potential for many more data points (within the current flight) to establish a *persistent* shift and launch a single fault isolation procedure immediately. The plurality and fusion of such analyses will mitigate the risk of false alarms and missed detections as well as reducing the potential for collateral (i.e. secondary) damage that might occur if the FDI process were allowed to span several flights, as is the case with *snapshot* data usage.

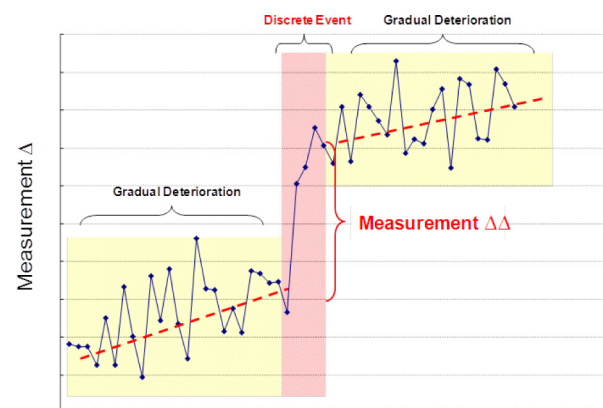


Figure 1. Snapshot parameter Δ trending and event detection

Technical Challenges

The use of streaming data offers a number of technical challenges to afford the same FDI functionality provided with legacy snapshot approach. The ability to perform this same functionality on-board with streaming data, both steady state and transient, requires a number of changes in basic strategy. The first centers on the *reference* model (used to generate the measurement parameter Δs) and the long term performance tracking. The traditional discrete Kalman Filter typically used in the snapshot analysis must be re-cast as a dynamic tracking filter in closed loop operation with a transient (physics) model of the engine. The latter can be a piecewise linear State Variable Model (SVM) or a more sophisticated non-linear component level model and the Kalman Filter (KF) can be derived first from an SVM and later applied to a non-linear model if required. The KF states are extended to contain health parameters (efficiencies and flow capacities) that act as *tuners* to track health deterioration of the engine over the long term. Unfortunately, these tuners can be corrupted if there is any modeling error, instrumentation bias and engine-to-engine variation. This mismatch between the specific monitored engine and physics model is apparent in the data and is most severe during transient operation. This necessitates that a normalization of the model be completed to remove these inconsistencies and allow the KF to track health deterioration in an uncorrupted manner. As we will see in a moment, this is also a necessary (but not sufficient) condition to enable proper anomaly detection and isolation of abrupt faults. One strategy for accomplishing this is to construct an empirically derived difference model between the monitored engine and the model and then use this to initialize the physics model [5,6]. This will remove the steady state and transient errors and allow the tuners to track health over the life-cycle of the engine. This stabilization between the engine observations and the model reference afforded by the normalization permits more accurate anomaly detection and reduces false alarms. Other methodologies can be found in [7,8].

Event (or anomaly) detection and subsequent fault identification is another capability to address in the streaming data environment. This issue is just as insidious as the model normalization task in the technical difficulties that it presents. Fault events (of the same type that are addressed in the snapshot analysis world) must be detected and isolated during transient conditions as well as steady state operation. Extensions of the methods typically applied in steady state, seem to perform adequately enough for near-steady state conditions when streaming data are applied. Transients, however, do not fare as well. The legacy detection methods do not work well and identification methods miss the target more often than not. This paper describes an approach that addresses these issues and will allow extensions of these same methods to be utilized in situations where there are two (or more) engines present on the airframe.

Companion Engine Data

The basic idea is to use information available from the companion engine(s) through a cross-wing comparison. Since companion engines may have different performance levels (FADEC plug class), may operate at slightly different power levels (due to performance or other operational considerations, e.g. airframe control trim, etc), may have different instrumentation errors or bias anomalies, a direct

comparison of engine parameter values is not possible (or desired). The inclusion of an engine model for each engine, properly tuned or initialized to represent the individual engines would provide a means to mitigate these concerns. Residual differences (Δs) for each gas path parameter relative to its individual normalized model removes the performance differences, instrumentation error differences, plug class differences, power level differences, etc., and permits a cross wing comparison of the individual parameter Δs to be made for anomaly detection purposes. In essence, the companion engine with its attendant normalized model becomes the *reference* for the monitored engine under analysis. Deviations due to environmental changes, flight maneuvers, and long term deterioration will effectively cancel out in the cross wing comparison. The latter item, assumes that after initial calibration and model normalizations, both engines will essentially deteriorate at *approximately* the same rate. This assumption is made on the basis that both engines are exposed to the same environment, flight maneuvers, and power excursions. In this way deviations not common to both engines will become visible in the cross wing comparison. In the next section, we explore the cross-wing analysis architecture and describe its attendant algorithms.

CROSS-WING ANALYSIS ARCHITECTURE

Before introducing the proposed methodology, it is important to understand the scope under which it is to be employed. The goal is to support engine health management decisions and enable faster and more accurate fault detection and isolation to support maintenance activities. Although the process and algorithms to be discussed can be executed on-board, it is not intended, at least at this stage of initial development, to offer real-time cockpit annunciation or accommodation. As such, the final analysis alerting will occur at the *end of flight* to take full advantage of the information fusion afforded by considering full flight data.

The proposed architecture for this cross-wing comparative FDI is depicted in Figure 2. For simplicity, we have depicted a two engine aircraft configuration. It should be noted, however, that this can be generalized to multi-engine (>2) aircraft configurations by applying this framework pairwise across remaining engines.

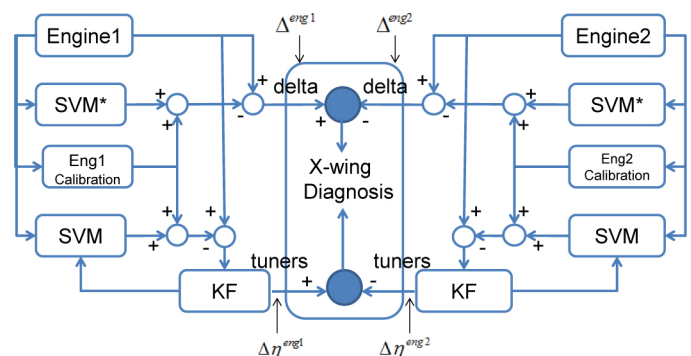


Figure 2. Cross-Wing Analysis Architecture.

The architecture is clearly symmetric. The entity labelled SVM is a State Variable Model and there are four *identical* instances of this model depicted in the Figure. The asterisk (*) that appears on two of them is only intended to distinguish it from its neighbor (for a given engine, say Eng1) in that while identical it may be operating in a

different state since it does *not* have a feedback from the attendant Kalman Filter (KF) estimator. The blocks labelled *Calibration* provide a normalization that is *unique* to each engine in order to align the model more closely to the engine at installation. It is depicted as a bias adder in the Figure as a generality to allow any form of model (linear or non-linear) to be substituted for the SVM without need for changing the architecture or the FDI process to be described herein.

The FDI process, in overview, can be explained as follows. The cross wing comparison $\Delta\Delta_s$, (denoted $\Delta\Delta^* = \Delta^{eng1} - \Delta^{eng2}$), for two *calibrated* engines would be (on the average) zero during any operation. If one of the engines suffers a rapid deterioration due to FOD or DOD or is the victim of an abrupt engine subsystem or instrumentation failure (full or partial), the cross-wing $\Delta\Delta^*$ will diverge in consequence. Comparing the $\Delta\Delta^*$ to an appropriately chosen threshold and applying a persistency check criteria, will flag the anomalous behavior. This provides a detection method across the full flight regime. It does not, however, determine which of the two engines is in fault. To determine this, a second identical normalized model (per engine), fitted with a Kalman Filter observer (used for long term performance deterioration tracking) is required. The faulted engine's Kalman Filter *tuners* will react (i.e. diverge) as a result of the fault event, whereas the non-faulted engine's Kalman Filter tuners will not. This allows the identification of which engine has sustained the faulty condition. Furthermore, the cross-wing $\Delta\Delta^*$ across all measured engine parameters, provide a *signature* for subsequent fault isolation. In the next sections, we will explore some of the details inherent in this architecture.

Engine Model Normalization

It should be no surprise that the engine model plays a pivotal role in any model-based FDI method. It forms the fundamental reference framework representing the nominal state from which (observed) parameter deviations can be quantified as the engine degrades and/or faults occur. Variation in the form of production/overhaul engine-to-engine differences, modeling infidelities/simplifications, instrumentation biases, etc., is ever present and must be accounted for in any model-based diagnostic/prognostic approach. Minimizing this variation is critical to accurate engine health monitoring analyses. A fundamental practice to mitigate this variation is engine model *calibration* (or *normalization*). This is the process of aligning the model to the actual engine under observation for diagnostic systems that perform, at least some of its functions, on-board in a quasi-realtime environment. This alignment can be achieved in a variety of ways through adjustments to the model in the form of corrections or normalization factors whose net effect is to reduce the differences between model output and actual (sensed) observations. In any event, the efficacy of the calibration method and process will depend on the amount (and type) of data available, time allotted to accomplish the calibration, robustness of the process, (i.e., can it be applied generically to any model formulation), and of course, the level of error that can be tolerated in the calibrated model for the specific application at hand. It is outside the scope of this paper to describe, in detail, a specific calibration procedure, rather, we will explore some of the high level properties and consequences of a calibration within the diagnostic process.

Quasi-Steady-State

Aero-engines exhibit a wide-ranging level of dynamic behavior. It is fair to say, from a model error perspective, that steady state operation is undoubtedly an easier mode of operation to model than the almost infinite transient maneuvers that an engine might experience during flight. As a consequence, model calibration for *steady state* operation is a far easier task than a model calibration that encompasses all possible engine transients. As a matter of practicality, a *reasonable* model calibration would capture something in-between these two extremes. We will use the term *Quasi-Steady-State (QSS)* to designate the *collection of conditions for which an acceptable alignment (i.e., calibration) can be made*. The term acceptable, will depend on many factors, including model type, maximum tolerable error, time to calibrate, and other elements germane to our particular application. We will assume, without loss of generality, that *QSS* consists of a collection of engine operating and flight regimes that contains steady state operation and some subset of transient behavior. Whatever the form the calibration process takes, we will assume that the following two (fairly obvious) properties apply: a) If *A* is an operating regime that can be calibrated, then any subset of *A* can be calibrated, and b) the union of calibrated regimes, can also be calibrated. With those assumptions, it is clear that *QSS* will consist of a union of calibrated operating regimes (A_k).

The form of A_k will vary according to the specifics of the calibration process (and model configuration) in use, but in general will be a function of ambient conditions, flight conditions, and engine operating parameters, (e.g., power level, variable geometry, internal bleed status, etc.). It is apparent from these two properties that calibration can be a sequential process which can occur incrementally over a number of flights.

What is important, for FDI, is the ability to recognize whether data being processed (or collected) is within a particular regime that is contained in the *QSS* collection. This question can only be answered if we know the particular calibration process being applied. We will defer this question for discussion in the Simulation Verification section later in this paper, and first explore some consequences of having an already defined *QSS* calibration regime and how it is exploited in performing an engine FDI.

FDI Process

Before we can identify a fault, we must know that one has actually occurred, i.e., we must first *detect* the presence of, (or consequence of), a fault. In this context, we assume a fault to be some part of the engine system or one of its subsystems which have ceased to function as intended. This may be a total failure or a partial failure of the system or subsystem element which (typically) occurs over a short period of time, thus distinguishing itself from normally expected long term degradation or deterioration. To be detectable, it must also affect some set of parameters that we have available to us to observe, i.e. sense. Since failed elements do not repair themselves, their effect on what we can observe must also be *persistent*. This latter requirement lessens the likelihood of a misdetection due to spurious operational events and noise. Lack of model calibration can also lead to misdetection if the model infidelity during certain operational maneuvers provides sufficient departure of model response from actual