# Generalized Cross-Signal Anomaly Detection on Aircraft Hydraulic System[1,2]

Ryan Mackey
Mail Stop 126-147
California Institute of Technology
Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109
818-354-9659
Ryan.M.Mackey@jpl.nasa.gov

*Abstract*— This paper outlines the mathematical foundation for a general method of anomaly detection from time-correlated sensor data. This method is a component of BEAM [1], described elsewhere, but as an individual algorithm is capable of fault detection and partial classification. The method is applicable to a broad class of problems and is designed to respond to any departure from normal operation, including faults or events that lie outside the training envelope. We will also consider training of the detector and interface to a larger diagnostic system. Lastly we will examine a brief illustration taken from aircraft testing that demonstrates the power and versatility of this method.

TABLE OF CONTENTS



## 1. INTRODUCTION

Aircraft diagnostics is an old subject but one with obvious room for improvement. Traditional diagnostic approaches are hindered by a fundamental limitation; namely, the class of faults a system can experience is never fully understood during the design phase. Typically the space of fault coverage is no better than 50-70% by first flight at best. This is complicated by system improvements, multiple configurations and modular design, and part replacement during routine maintenance. Such is true of many systems besides aircraft, and work here is equally applicable to and mirrors efforts for autonomous space vehicles and similar systems.

Modern aircraft proposals are increasingly concerned with total cost, and the aircraft system's ability to *affordably* meet the requirements of is customers. Reliability and sustainability is central to cost, and so the issue of diagnostics is key. In order to directly address this problem, Boeing has assembled a team to research a component of the vehicle health management system to improve diagnostics over time. This is the *Anomaly System*, where we have defined an anomaly simply as "off-nominal system behavior," of which the broader class of faults can be considered a subset. The anomaly system is dedicated to observing system performance, identifying characterizing anomalous episodes, and returning this information to the ground. This system defines a rigorous, repeatable, and sustainable process to improve the diagnostics over the course of the aircraft's operational lifetime.

The process of identifying anomalies, or to be more specific, *Novelty*, is not at all unusual among the spacecraft community. Spacecraft tend to be complex and unique devices, often facing new environments and phenomena that are poorly understood. In order to safely monitor and control a spacecraft, one must be able to sense new behaviors and understand or correct them. JPL has a major interest in robotic space exploration and has sought mathematical means to aid spacecraft controllers. One new system intended for this duty is BEAM (Beacon-based Exception Analysis for Multimissions) [1], which is an end-to-end method of data analysis intended for real-time fault detection and characterization. It provides a generic system analysis capability for potential application to deep space

---


[2] Updated September 15, 2000

probes and other highly automated systems. JPL has worked to support the Boeing team in order to leverage spacecraft experience against similar aircraft challenges.

In this paper, we will describe the architecture and operating theory of a purely signal-based anomaly detection method and its application to a typical aircraft system. The component we will focus on is referred to as the System Invariant Estimator (SIE) component of BEAM. It receives multiple time-correlated signals as input, and compares their cross-signal behavior against a fixed library of invariants. The library is constructed during the training process, which is itself data-driven using the same time-correlated signals. The SIE returns the following quantities, which will be explained in detail:

- Mode-specific coherence matrix
- Event detection
- Comparative anomaly detection
- Anomaly isolation to specific signals
- Quantitative measure of off-nominal behavior

This method seems simplistic and it can be improved through a variety of methods, such as integration with model-based and symbolic reasoning components as in the full BEAM formulation [1], or through fusion with other detection methods as in [2]. Nonetheless, the detector as presented is broadly applicable, easily trainable, exhibits excellent false-alarm characteristics, and performs a reliable first stage of fault isolation, even in response to novel conditions. This will be clearly shown in the example case.

## 2. CROSS-SIGNAL MOTIVATION

The process of anomaly detection described here is motivated by a simple reality of aircraft systems. Modern aircraft provide a wealth of sensor data coming from performance sensors – i.e. pressures, temperatures, RPM measurements, etc. – a list that is gradually evolving to include advanced sensors, high-frequency signals, and "virtual" sensors or preprocessed signals. Furthermore, weight, space, and power considerations discourage additional diagnostic-only sensors. Instead, the available data must be studied more carefully than in the past.

Because quantitative information is so readily available, approaches grounded in signal processing are likely to be effective. The method described here has two distinct advantages. The first is its broad range of applicability -- the module described here has been used to successfully fuse sensor and computed data of radically different types, on numerous systems, without detailed system knowledge and with minimal training. The second is its ability to detect, and with few exceptions correctly resolve, faults for which the detector has not been trained. This flexibility is of prime importance in systems with low temporal margins and those with complex environmental interaction.

Let us approach the problem from a mathematical standpoint. Consider a continuously valued signal from an electromechanical system, sampled uniformly. Provided this signal is deterministic, it can be expressed as a time-varying function:

$$S_i = f\left(\{S_i(t-dt)\},\{E(t)\},\varepsilon(t)\right) \qquad (1)$$

In the above expression, we have identified the signal as a function of itself and other signals, as expressed by $\{S_i(t)\}$, and of the environment, which may contain any number of relevant parameters $\{E(t)\}$. There is also a noise term $\varepsilon(t)$ included to reflect uncertainties, in particular actual sensor noise that accompanies most signals in practice.

The process of identifying faults in a particular signal is identical to that of analyzing the function $f(t)$. Where this relation remains the same, i.e. follows the original assumptions, we can conclude that no physical change has occurred for that signal, and therefore the signal is nominal. Such is the approach taken by model-based reasoning schemes.

However, the function $f$ for each signal is likely to be extremely complex and nonlinear. The environmental variables may be unknown and unmeasurable. Lastly, the specific interaction between signals may also be unknown, for instance in the case of thermal connectivity within a system. The sheer complexity of the problem precludes model-based techniques in many cases. For this reason, it is more efficient and more generally applicable to study invariant features of the signals rather than the full-blown problem.

One excellent candidate feature for study is cross-correlation between signals. By studying this computed measurement rather than signals individually, we are reducing the dependence on external factors (i.e. environmental variables) and thus simplifying the scope of the problem.

Cross-correlative relationships between signals, where they exist, remain constant in many cases for a given mode of system operation. The impact of the operating environment, since we are dealing with time-correlated signals, applies to all signals and thus can be minimized. This approach is essentially the same as decoupling the expression above, and choosing to study only the simpler signal-to-signal relationships, as follows:

$$S_i = f\left(\{S_i(t-dt)\}\right)\circ g\left(\{E(t)\}\right)\circ \varepsilon(t) \qquad (2)$$

This hypothesis is not strictly true, but it tends to be a good approximation for most realistic systems. In most cases, relationships between signals that represent measured quantities are readily apparent. The environmental contribution can be considered an external input to the system as a whole rather than being particular to each signal. The sensor itself is the source of most of the noise, and it too can be separated.

We must remember to consider the *operating mode* of the system, as hinted at above. For the purpose of this discussion, a mode implies a particular set of relational equations that govern each signal. In other words, the operating physics of the system can differ between modes but is assumed to be constant within a mode. These modes are ordinarily a direct match to the observable state of the system – i.e. inactive, startup, steady-state, etc. Mode differs from the external environment in that it is a measure of state rather than an input to the system's behavior.

Provided we can correctly account for operating mode, we then have a much simplified set of relations to study, namely those between pairs of signals, or in the more general sense each signal versus the larger system. Faults in the system can be expected to manifest themselves as departures from the expected relationships. For this reason, the study of correlations between the signals is singularly useful as a generic strategy.

## 3. State Estimation and Anomaly Detection

Two common measures of second-order cross-signal statistics are the Covariance and the Coefficient of Linear Correlation. Covariance is a good measure of similar behavior between arbitrary signals, but it suffers from a number of difficulties. One such problem is that a covariance matrix will be dominated by the most active signals, viz. those with the greatest variance. In order to avoid this, covariance is typically normalized by the relative variances of the signals, as in the Correlation Coefficient. However, this is often overly simplistic and leads to the inverse problem, as a correlation matrix tends to become ill-conditioned in the presence of signals with relatively low variances.

Returning to the original goal, we are interested in comparing signals. This should take into account both the covariance and the relative variances of the signals. This leads us to the expression for the *coherence coefficient* given below:

$$\zeta_{ij} = \frac{\left|Cov\left(S_i, S_j\right)\right|}{Max\left(Var\left(S_i\right), Var\left(S_j\right)\right)} \tag{3}$$

We have used the familiar definitions:

$$Cov\left(S_i, S_j\right) = \frac{1}{t}\int\left(S_i - \overline{S}_i\right)\left(S_j - \overline{S}_j\right)dt \tag{4}$$

$$Var\left(S_i\right) = \frac{1}{t}\int\left(S_i - \overline{S}_i\right)^2 dt \tag{5}$$

The maximum variance is used in the denominator to guarantee a coherence value normalized to [-1, 1]. Furthermore, the absolute value is taken because the sign of the relation is of no importance for arbitrary signals, only the existence or nonexistence of a causal connection. A coherence value close to 0 implies no relationship between signals, whereas a value approaching 1 indicates a very strong relationship.

Given n data streams, this calculation defines an n x n matrix where each entry represents a degree of causal connectivity. Where relationships between signals are fixed, i.e. during a single mode of system operation, the coherence coefficient between those two signals will remain constant within the bounds of statistical uncertainty. Provided the coherence coefficient converges (see Section 4), this calculation is repeatable, and so it can be used as a basis for comparison between training and run-time data.

Admittedly the above assertions are too strict for a real-world example. Systems may indeed contain signals with fluctuating or drifting relationships during nominal operation, or the appearance of such due to nonlinear relationships. Additionally, the requirement to maintain a countable number of modes may force us to simplify the system state model, to the detriment of repeatability. We will mitigate these concerns, but for now let us press on.

Having understood that cross-channel measurements are an effective method of signal analysis, we next explore how to best apply the calculation above. The first question to ask is how the data should be gathered. From the discussion above, it is clear that we must avoid applying this operator to mixed-mode data. Such data represents a combination of two separate sets of underlying equations for the signals, thus mixed-mode correlations are not necessarily repeatable. Most other cross-signal applications avoid this issue through one of the following methods:

- Compute only correlations having a fixed, mode-independent relationship.

This method is effective in reliable fault detectors, however the system coverage is typically very limited. This approach is restricted to well-understood signal interactions and is not generalizable. (However, see Section 4, where we attempt to redress this philosophy.)

- Window the data until we can assume quasi-steady-state operation.

This procedure also carries significant inherent limitations. Because the computation (of coherence coefficient or any other method) is statistical in nature, selection of a fixed window size places a hard limit on latency and upon confidence of detection. This also does not directly address the core problem – we still do not have pure single-mode operation.

- Window the computation according to external state information, such as commands.

This is the best approach, and it is used in the full formulation of BEAM. However, it too has limits. External state information may not be available. Additionally, there may not be a perfect alignment between discrete "operating modes" and observable shifts in the system – it may not be one-to-one.

Our solution to the mixed-mode problem is based upon mathematical properties of the computation. Consider a pair of signals with a fixed underlying linear relationship, subject to Gaussian (or any other zero-mean) random noise. The coherence calculation defined in (3) will converge to a fixed value, according to the following relationship:

$$\zeta_{ij}(t)-\zeta_{ij}(t-1)\sim\frac{1}{t^2} \qquad (6)$$

This follows from the squared terms in the denominator of (3). The *exact* rate of convergence depends on the relative contribution from signal linear and noise components as well as the specific character of signal noise. However, in practice, it is much easier to determine the relationship empirically from training data.

Given the convergence relationship above, we can define a data test in order to assure single-mode computation. By adopting this approach, we can successfully separate steady-state operation from transitions. This means:

- Transition detection is available for comparison to expected system behavior.

A "transition" in this case is a switch from one mode to another. Most of these are predictable and nominal. On the other hand, a broad class of system faults can be considered transitions, particularly those involving sudden electrical failure or miscommand scenarios. Unexpected events in the system immediately merit further analysis.

- Calculated coherence uses the maximum amount of data available to make its decisions, which optimizes sensitivity and confidence.

Use of the convergence rate establishes a time-varying estimate of confidence in the calculation, which is transparent to the final output of the detector. The time-variance also applies to the values of the computed coherence, which we will study in further detail.

The quantity $p(t) = \zeta_{ij}(t) - \zeta_{ij}(t-1)$ is referred to as the *coherence stability*. This single parameter is a good indicator of steady-state behavior.

One observation regarding the coherence stability is that its convergence rate is quite fast. This allows us to make confident decisions regarding mode transitions with relatively little data to study. This also lends credibility to more complex and subtle fault detection using a coherence-based strategy.

Next we will use a similar strategy to differentiate between nominal and anomalous data, where the fault manifests itself as a drift rather than a transition. Such a fault case is more physically interesting than a sudden transition, since we are concerned about a lasting effect upon the system rather than an instantaneous data error. Suppose we have a current $\zeta_{ij}(t)$ estimate that we are comparing to a precomputed estimate called $\zeta_0$. As we accumulate more data, the estimate is expected to converge at the following rate:

$$\left|\zeta_{ij}(t)-\zeta_0\right|\sim\sqrt{\frac{1}{t}} \qquad (7)$$

This relationship determines the accuracy of the calculation's raw value, which is representative of the underlying physical relationship between the two signals. It is conceptually similar to the error in estimated mean for a statistical sampling process. We can use this relationship to detect a shift in the equations, much in the manner that events are detected above.

The computed quantity $\zeta_{ij}(t) - \zeta_0$ is referred to as the *coherence deviation*. When compared with the base convergence rate, it is a measurement of confidence that the coherence relationship is repeating its previous (nominal) profile. Between detected mode transitions, this relationship allows us to optimally distinguish between nominal and anomalous conditions. Violation of this convergence relationship indicates a shift in the underlying properties of the data, which signifies the presence of an anomaly in the general sense.

Note that the convergence rate of this relationship is considerably slower, though still fast enough to be practical. Because of this it is particularly valuable to adapt a variable-windowing scheme where data is automatically segmented at mode boundaries.

## 4. IMPLEMENTATION

In the previous section we defined a method of generic cross-signal computation and identified properties that facilitate decisions about the data. In this section we will examine how to best apply these properties to a realistic system.

The convergence properties above are written for each individual signal pair. In order to apply this approach in general to a system with N signals, we have $O(N^2)$ signal pairs to process. At first glance, the approach does not appear to lend itself to scaling. For this reason, most cross-signal approaches focus on preselected elements of the matrix, which cannot be done without considerable system knowledge or examples of anomalous data from which to train.

In general, we may not know a priori which signal pairs are significant. Additionally, there are likely to be numerous interactions for each signal, which may vary depending on the mode of operation. Only in rare cases will individual elements of the matrix be the sole points of interest. Typically we are concerned with signal behavior versus the entire system, which corresponds to an entire row of the coherence matrix.

Because we are more concerned with the overall system performance, we should instead consider a single global measure based on the entire matrix. This requires some sort of matrix norm.

Many matrix norms exist, but we shall use the following, where M is an arbitrary N-by-N matrix:

$$\|M\| = \frac{1}{N^2} \sum_i \left| \sum_j M_{ij} \right| \tag{8}$$

The norm chosen here differs from the simple matrix average in one detail, namely the absolute value and its placement. An absolute value is used because the convergence test is only concerned with the magnitude of differences, rather than their sign. (An exception to this: Following the detection of an anomaly, for purposes of identification the sign can be important, as faults that cause an *increase* in coherence are typically more physically complex and more interesting.) The choice to average row totals rather than each individual element is motivated by the inherent structure of the coherence matrix, specifically the fact that each row represents a single signal's total contribution. By averaging the rows prior to their summation we hope to counteract noise present in the calculation, whereas differences due to a significant shift are likely to be of the same sign.

We can substitute the norm into the convergence relationships (6) and (7) without changing their character:

$$\|\zeta_{ij}(t) - \zeta_{ij}(t-1)\| \sim \frac{1}{t^2} \tag{9}$$

$$\|\zeta_{ij}(t) - \zeta_0\| \sim \sqrt{\frac{1}{t}} \tag{10}$$

The stability and deviation on the left are now indicative of the entire matrix, i.e. we are now tracking only two parameters, regardless of system size. This produces a tradeoff between individual pair sensitivity and false-alarm reduction, while at the same time greatly reducing computational cost.

A further adaptation of this approach is to consider separate weighting of different pairs. It is clear that some signal pair relationships will be highly repeatable while others will be pseudorandom. Additionally, we have adopted the concept of multiple modes to handle different relationships at different phases of system operation. This can become an unbounded problem, and a mechanism is needed to guarantee a small number of modes.

Let us introduce a weighting matrix $W_{ij}$ into the convergence relationships above:

$$\|W_{ij}\zeta_{ij}(t) - W_{ij}\zeta_{ij}(t-1)\| \sim \frac{1}{t^2} \tag{11}$$

$$\|W_{ij}\zeta_{ij}(t) - W_{ij}\zeta_0\| \sim \sqrt{\frac{1}{t}} \tag{12}$$

The matrix $W_{ij}$ is a companion to the training matrix $\zeta_0$ and is computed as part of the training cycle. For a general application, i.e. an application for which no signal relationships are known or suspected, it is computed by normalizing each signal-pair coherence by the observed variance in that coherence. This normalization matrix, along with the model coherence $\zeta_0$ and the uncertainty in the training set, can be later combined with other coherence/normalization pairs in order to combine modes or enhance training data results with new data.

Once a fault has been detected, the next step is to isolate the responsible signals. This is done by studying the difference matrix:

$$D_{ij} = W_{ij}\left(\zeta_{ij}(t) - \zeta_0\right) \tag{13}$$

Given an anomaly on one signal, we expect to see the correlation between this signal and all others diminish compared to the expected values. There may be stronger shifts between some signals and others, but in general the coherence values will decrease. Visually this leads to a characteristic "cross-hair" appearance on the rendered difference matrix.

The total deviation for each signal is computed by summing the coherence difference (absolute values) over each row of the matrix. Ranking of these deviations determines the most likely contributors to the faults. This channel implication is passed to interpretive elements of BEAM and to single-signal analysis modules.

In general an anomaly will manifest as a decrease in coherence between signals. However, there are rare cases where coherency will increase. Typically this is not system-wide but is isolated to a few specific pairs. Such an increase in coherency is indicative of a new feedback relationship occurring in the system, and it must be given special attention.

These special cases, physically, define previously unknown modes of the system. This mode may be nominal or faulty. In the former case, such detection implies that the training data used to tune the detector does not adequately cover the operations space, and must be expanded. In the latter case, knowledge of what specific signals or pairs are anomalous can directly lead to better understanding of the problem, particularly in cases where causal or physical models are available to the diagnostic engine.

## 5. Basic Architecture

Figure 1 displays the computational embodiment of this process. This architecture is an exploded view of the SIE box contained in the overall BEAM architecture, found in [1]. In the case of this experiment, this architecture stands alone, which has some minor consequences on the architecture as described below.

Each sample of time-correlated, stationarized data is passed to the Incremental Coherence Estimator, where equation (3) is updated for each signal pair. The coherence stability is computed over the matrix, and is checked against relationship (11) in the Convergence Rate Test. If this test fails, the coherence estimate is reset and a new data window is begun.

A typical coherence matrix is presented in Figure 2. The colormap shows values of the matrix from 0 (dark blue) to 1 (red) for each signal pair, with different signal numbers on the X and Y axes. Notice that, as expected, the matrix is symmetric, and there is a stripe of 1's along the diagonal. Diagonal entries represent autocoherences, which are always 1. An exception to this: In cases where a signal has no variance, the coherence value is uniquely zero except on the diagonal. However, we have adopted the convention that for a truly constant signal, we will retain a zero on the diagonal until its variance is positive. Thus diagonal entries represent signals that are temporarily constant and therefore invalid for this computation.

After the test above, we are guaranteed a coherence estimate free of mixed-mode data. The estimate is compared against the expected coherence supplied by the Coherence Library, as selected by the symbolic model and command data. The match is checked against relation (12).

If we have a mismatch that compares favorably to an abnormal library coherence, we have a known fault, which will be flagged according to the fault number and passed to the interpreter. If we cannot find a suitable match, as is more frequently the case, the differenced coherence, computed by equation (13), is examined to extract the key actor signals and pairs. A typical difference matrix is displayed in Figure 3. We have adopted the convention that positive values on the difference matrix indicate a loss in coherence relative to the training data, and negative values indicate an increase in coherence.

At the end of this operation, we will have successfully identified normal versus anomalous operation of the system as a whole. For those cases where anomalous conditions are

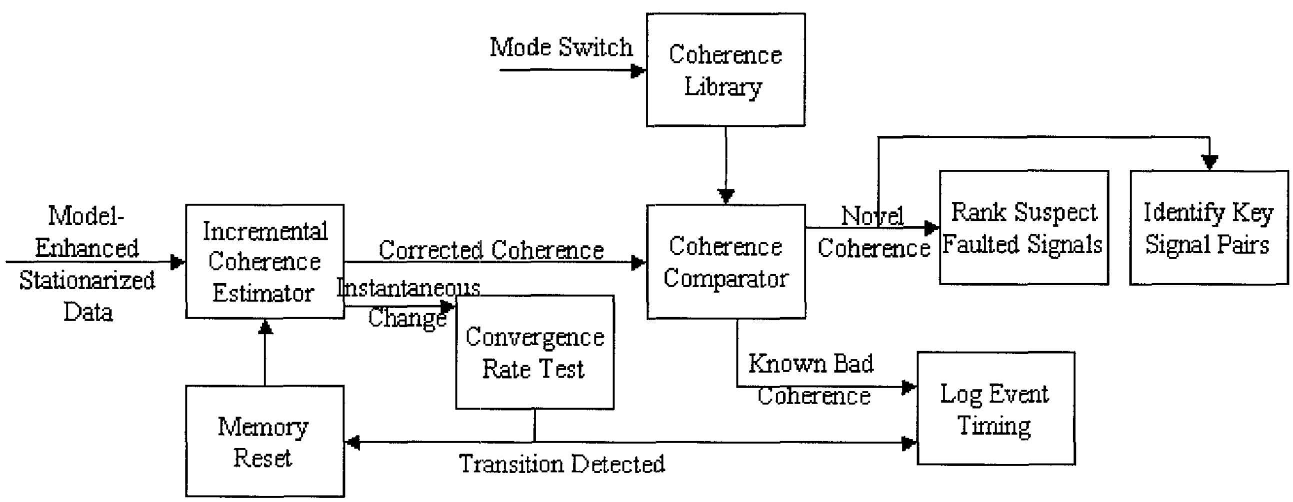


Figure 1: SIE Block Architecture

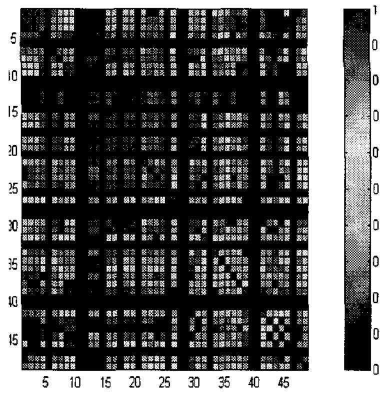


Figure 2: Sample Coherence Matrix

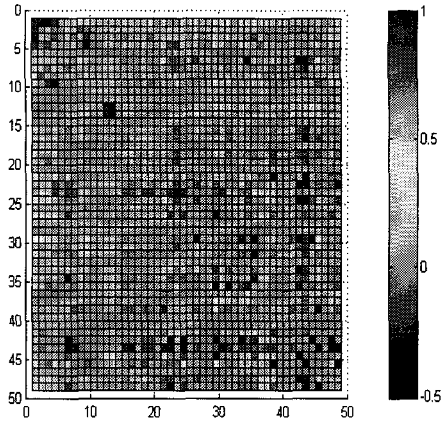


Figure 3: Sample Difference Matrix

detected, we have isolated the effect to a known case or to the key measurements that led us to that conclusion. This has, in essence, digitized the problem into terms that an automatic interpreter can understand.

For this particular stand-alone application, and for purposes of strict anomaly detection, we are lacking three features usually built into the detector. We have made some changes to the usual architecture to accommodate them. These are:

- Stationarization of sensor data.

As mentioned in section 3., it is unlikely but possible that the coherence coefficient may not converge for nonstationary data. Because we are applying this method blindly, we rely upon an approximate stationarization method, namely differencing of sensor values against their previous values. In other words, the detector is applied to the changes from sample to sample of each individual sensor. This has the practical effect of guaranteeing a zero mean for the incoming signals. In ordinary practice, a more sophisticated stationarization can be constructed given some knowledge of the incoming signals.

- Mode selection by command or state information.

For this example, we have no command or state information available. Thus we cannot preselect a specific mode-indexed training set for comparison. For this experiment, we will compare current results against the entire nominal training set, using the closest match for comparison. The practical effect of this is to desensitize the detector. However, should it remain effective, we have further demonstrated its broad applicability and sensitivity.

- Faulted training data.

For this example, we will only train with nominal data. Therefore, faulted data will be "anomalous," i.e. novel to the detector. This test will examine the detector's ability to sense true anomalies as well as its ability to characterize events completely outside the training envelope.

## 6. Hydraulic System Test Case

To illustrate this approach, we will consider processing results on data acquired from an aircraft hydraulic system. The system in question is fairly typical of aircraft systems, in that we have a small number of continuously valued signals uniformly sampled. Specifically, we have eight sensors, all of them pressure sensors, sampled at 200 Hz. The sensors reflect pressure sampled at different points in the hydraulic system.

To complicate matters, we do not have any discrete state information available. The hydraulic system does not have any directly definable "modes," either. It is an accommodating system that is indirectly affected by the amount of control stick activity directed by the pilot. We do not have any visibility into these stick commands aside from a rough classification given to individual datasets – data is labeled as representing "Light," "Moderate," "Heavy," or "Violent" stick activity.

The data provided for this experiment covers eleven different observations, which vary from approximately 10 to 20 seconds in length (2000 to 4000 samples). Nine of these indicate nominal system operation, and two indicate failures; this information is provided prior to the test.

Failure in the hydraulic system was induced by attenuating the accumulators. Because this is an accommodating system, it is difficult to see the effects of this change. One run of failure data was provided with "Moderate" stick activity, and one with "Violent" stick activity.

Example datasets are displayed in Figures 4 and 5. These are the nominal and faulty "Violent" sets. All eight signals are plotted on the same axis.

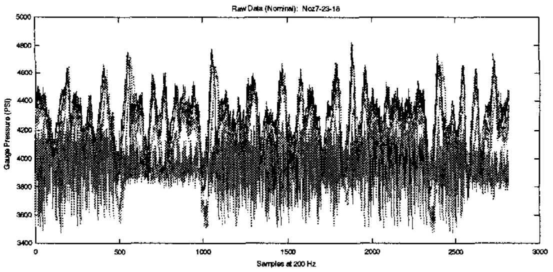


Figure 4: Nominal Hydraulic Data, Violent Stick Activity

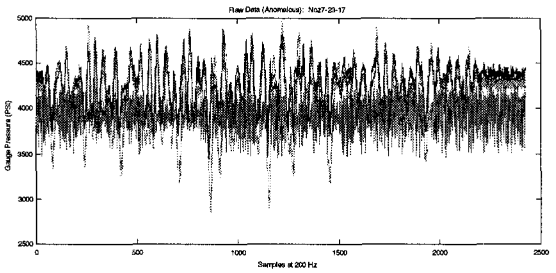


Figure 5: Anomalous Data, Violent Stick Activity

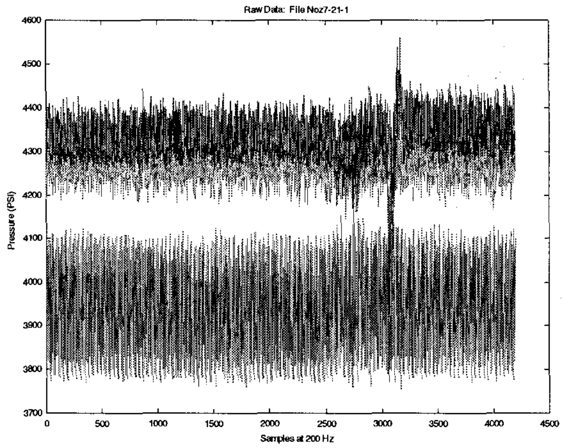


Figure 6: Nominal Data, Light Stick Activity

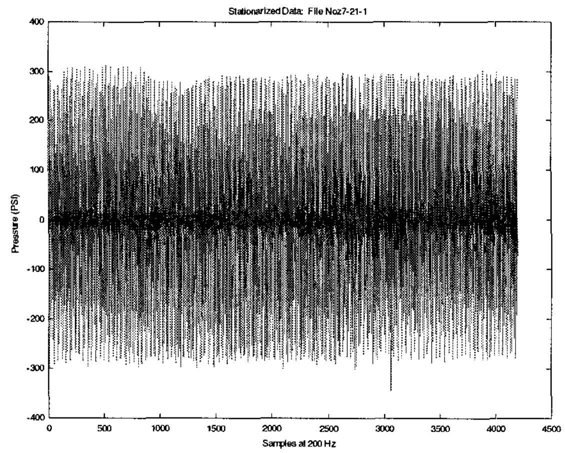


Figure 7: Stationarized Data from Figure 6

Figure 6 is an example of "Light" stick activity with nominal performance. Axes and signals are the same.

In our opinion, the data fail the all-important "eyeball test" – that is to say, it is not obvious to an untrained observer that there are visible features in the data to distinguish nominal from anomalous conditions. It is also not clear how the influence from control stick actuation complicates the problem, but there appears to be some effect.

Given this and only this information, we must prove the performance of the detector. This means we must answer the following questions:

- Can we distinguish nominal from anomalous data
- Can this distinction be made without training on anomalous data
- What is the sensitivity vs. false-alarm performance
- How much nominal data is required to train the detector
- How many "modes" are required for good false-alarm performance
- Is there a distinction between "Light" and "Severe" stick activity in nominal or anomalous data
- How accurately can the detector isolate this anomaly

The detector was trained using a subset of the available nominal data, permitting some nominal data to be withheld for false-alarm testing. The nominal data was rotated and retried to ensure that no single nominal datafile was essential for claimed false-alarm performance. In addition, the detector, once trained, was tested on both anomalous files. We are not only interested in the false-alarm performance (though this is of key importance), but also its sensitivity.

## 7. Detection Results

Because we are considering the SIE as an isolated component, we must begin by stationarizing the data as described in Section 5. This was done as part of the detector, by storing previous sensor values and subtracting them from incoming values. A plot of the effect on this data is given in Figure 7. This represents the same data shown in Figure 6 after this step. In practical use, this step would create a latency of detection of one sample, which is deemed to be acceptable.

The detector is trained by running it in a non-comparative mode. This means the coherences are computed and mode boundaries are sensed, and one "training" coherence is stored for each segment of the data. Along with the training coherence, a weighting matrix is computed based upon its repeatability. The training is repeated for each file, after which training results can be merged or left alone.

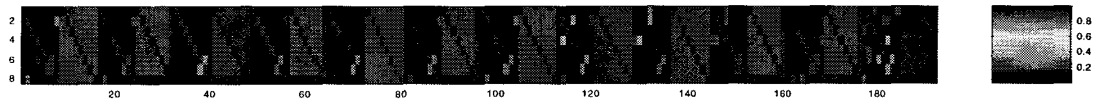


Figure 8: Raw Training Results (Twelve Sets) From Nominal, Light Stick Activity Data

A sample raw, uncombined training result is graphed in Figure 8. This is the result from the nominal file graphed in Figure 6, which contains twelve separate segments. For each segment, we have a pair of 8x8 matrices, which represent the training matrix and weighting matrix pair for each segment. Coherence training matrices are the left member of each pair, followed by the corresponding weighting matrix. To describe this result, each segment is characterized by a very diagonal coherence, as indicated by the red values close to 1 on the diagonal, and blue values close to 0 otherwise, and a weighting matrix that is uniformly close to 1. Values approaching 1 in the weighting matrix indicate stability in the corresponding training matrices, which is characteristic of a well-tuned detector. Very low values typically result when the convergence relationships are poorly tuned, which results in frequent mixed-mode segmentation.

There is one additional bit of information not presented in this graph, namely the confidence in each, as determined by the length in samples of each segment. The last segment, in particular, is shorter than the others (the file simply runs out of data), and therefore its confidence is quite low; this matrix will not be useful for the computation and may be discarded. The other eleven segments show good repeatability.

Note that the training data shows relatively low coherence values throughout. This is typical of poorly correlated systems. Given that the hydraulic system is adaptive and perhaps underinstrumented, it is not implausible that under normal operating conditions, there is little propagation of minor effects from one sensor to another.

Let us begin our experiment with a consideration of detection rates. One of the outputs of the detector is a simple, Boolean flag, updated at every sample, that indicates the presence or absence of anomaly. We should make clear that this flag is only set if we can confirm the presence of anomaly at that particular sample. There is no "latching," either for a momentary dip below the convergence threshold or a reset of the calculation at a window boundary. Latching of some sort would probably be implemented in a fielded system, but we will be very clear about the detector performance in this experiment.

Using any one nominal set as training data did not yield perfect false-alarm characteristics, but performance was impressive. An example result is shown in Table 1. The training was rotated through each of the files, but for this specific example, we have used the data of Figure 8 to train and applied the detector to all other files, nominal and anomalous. We refer to the files as nominal or anomalous, and according to stick activity.

In other words, both anomalous files are detected, and the flag is convincingly set for both. However, we have a slight false-alarm reading with the last nominal file. This is doubtless because we have trained with "Light" stick activity, and the false alarm occurs during "Violent" activity.

Table 1: False-Alarm and Anomaly Detection Results for Detector Trained on One Nominal File

| | | | | | | |
|---|---|---|---|---|---|---|
| File Type | (Trained) Light Nominal | Light Nominal | Moderate Nominal | Moderate Nominal | Moderate Nominal | Heavy Nominal |
| Anomaly Counter | 0 | 0 | 0 | 0 | 0 | 0 |
| File Type | Heavy Nominal | Violent Nominal | Violent Nominal | Moderate Anomalous | Violent Anomalous | |
| Anomaly Counter | 0 | 0 | 13 / 2817 | 3188 / 3473 | 2200 / 2424 | |

Table 2: False-Alarm and Anomaly Detection Results for Detector Trained on Two Nominal Files

| | | | | | | |
|---|---|---|---|---|---|---|
| File Type | (Trained) Light Nominal | Light Nominal | (Trained) Moderate Nominal | Moderate Nominal | Moderate Nominal | Heavy Nominal |
| Anomaly Counter | 0 | 0 | 0 | 0 | 0 | 0 |
| File Type | Heavy Nominal | Violent Nominal | Violent Nominal | Moderate Anomalous | Violent Anomalous | |
| Anomaly Counter | 0 | 0 | 0 | 2970 / 3473 | 2044 / 2424 | |

The solution is to add another mode corresponding to increased stick activity, giving us a total of two – one for "Light Stick" and one for "Significant Stick" nominal behavior.

Using training data from any two files, excepting only both "Light" activity files, we are able to achieve a zero false alarm rate for this test. A typical example, using "Light" and "Moderate" stick activity for training, is given in Table 2 above. Selection of different nominal files to train the algorithm produced very similar results to those shown in Tables 1 and 2.

Based on these results, we have answered all but one of the questions facing the detector:

- It is capable of distinguishing nominal and anomalous behavior.
- This distinction can be made without training on the anomalous data.
- Sensitivity vs. False-Alarm is perfect on a file-by-file basis for this data set.
- Two files of nominal data, comprising ~4000 samples, are sufficient to fully train the detector.
- Only two modes are necessary for this level of performance.
- There is a minor distinction between different levels of stick activity, usually too small to resolve.

The question that remains concerns isolation of the anomaly. Thus far we have only considered the detection across the entire system. We have not studied the individual signal implications following the detection.

The signal-specific results are output from the detector at every sample when an anomaly is indicated. This result is a number, again normalized between 0 and 1, indicating the distance a particular signal is from the training data. In general, a value of 0.1 is a large departure, with 0.5 being the practical maximum. Figure 9 is an example of a nominal file result forced to output at every sample for purposes of comparison. There is a slight bump present, which does *not* indicate an anomaly because it immediately follows a segmentation boundary, and confidence is correspondingly low.

Figures 10 and 11 show the results using the training data of Table 2, applied to both anomaly files. The horizontal axis is time, in samples, while the vertical axis counts the signals from 1 to 8. The colormap indicates the distance for each signal at each sample.

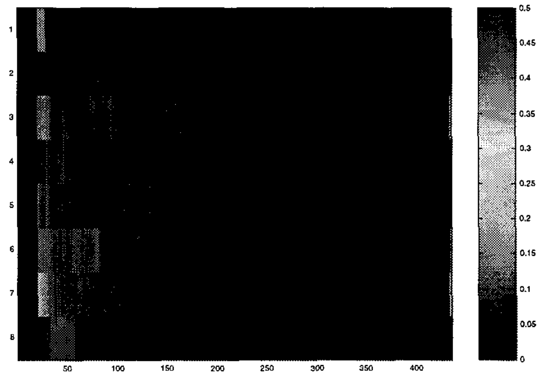

Figure 9: Signal Distances, Nominal Light Activity

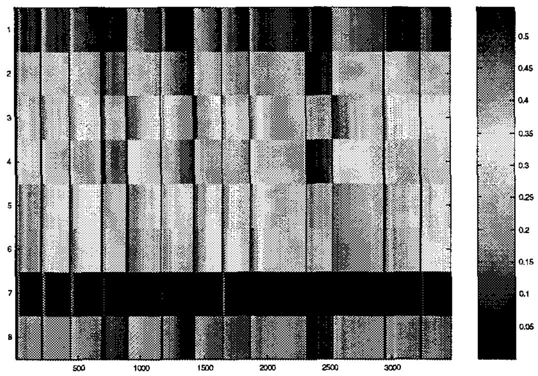

Figure 10: Signal Distances, Anomalous Violent Activity

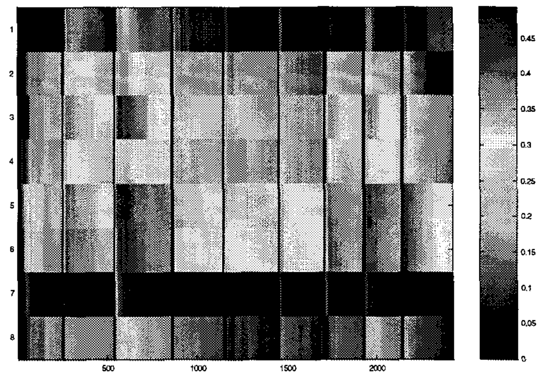

Figure 11: Signal Distances, Anomalous Heavy Activity

Two things are clear from the plots above. First, the anomaly detection is consistent and very strong. Second, the anomaly affects nearly the entire system. Such a result is expected given the connected and accommodative nature of the hydraulic system. There is an exception, though. The seventh signal is almost completely immune to the anomaly. After questioning the system experts, this result was explained by the fact that the seventh signal represents a slightly different type of measurement, in this case pressure at the APU, whereas the other measurements are very similar to each other, having been taken at similar points downstream in the system. In other words, the localization of the anomaly is plausible, even in this pathological case where nearly every sensor is affected and the process begins to saturate.

In a production environment, results such as these may or may not be of immediate use to system operators. Surely such displays are too sophisticated for a pilot, but they might provide additional insight to a system expert seeking to upgrade the diagnostic system in response to a newfound anomaly, as in [3]. Because the results are quantitative, repeatable, and tied directly to the data, they can be sent to autonomous components for further processing, as in [1] and [2], or they may be directly analyzed by human experts.

## 8. Conclusion

We have presented here a purely signal-based method of fault and anomaly detection suitable for use with nearly any instrumented system. Its flexibility allows it to be trained and maintained with relative ease, and it exhibits excellent characteristics with respect to sensitivity and false-alarm rates. It can be applied alone, as in the aircraft hydraulic example presented here, or as part of a larger and more sophisticated monitoring system.

Advanced processing such as this can be conducted on-board most aerospace systems, as processor resources and available sensors are usually more than adequate to support such analysis. Benefits of such processing translate directly into cost savings in terms of safety, maintenance, reduction of CND (CanNot Duplicate) conditions, and readiness. Furthermore, extracting knowledge from the raw sensor data is essential as the system becomes more and more autonomous.

In order to make the greatest use of available sensor data, processing should take place close to the source – subsystem by subsystem, performed on-board as much as possible. The benefits of advanced health management can only be fully realized if a comprehensive system in put into motion, and considerations of novelty and upgrading the diagnostics themselves are planned well in advance.

## Acknowledgement

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Adaptation of this algorithm to aircraft systems and integration was funded by The Boeing Company.

## References

[1] R. Mackey, M. James, H. Park, and M. Zak, "BEAM: Technology for Autonomous Self-Analysis," *The 2001 IEEE Aerospace Conference*, Big Sky, Montana, March 2001.

[2] T. Brotherton and R. Mackey, "Anomaly Detector Fusion for Advanced Military Aircraft," *The 2001 IEEE Aerospace Conference*, Big Sky, Montana, March 2001.

[3] G. Bloor et. al., "Anomaly and Prognostic Reasoning for Advanced Aircraft," *The 2001 IEEE Aerospace Conference*, Big Sky, Montana, March 2001.

[4] G. Box, G. Jenkins, G. Reinsel, *Time Series Analysis*, New Jersey, Prentice Hall, 1994.

[5] M. Wold, *A Study in the Analysis of Stationary Time Series*, Almqvist and Wiksell, Uppsalla, Sweden, 1938 (Second Edition 1954).

**Ryan Mackey** received his B.A. degree from the University of California at Santa Cruz (1993) for Mathematics and Physics, and went on to an M.S. (1994) and Eng. (1997) in Aeronautics at Caltech. He is presently a senior researcher and charter member of the Ultracomputing Technologies Research Group at the Jet Propulsion Laboratory. His research centers upon revolutionary computing methods and technologies for advanced machine autonomy, specifically deep space missions, UAVs and maintainable aerospace vehicles. His interests also include quantum- and biologically-inspired computing.