Original Article

Institution of MECHANICAL ENGINEERS
Journal of AEROSPACE ENGINEERING

Proc IMechE Part G:
J Aerospace Engineering
0(0) 1–14

DOI: 10.1177/0954410012468391
uk.sagepub.com/jaero
SAGE

Multi-sensor information fusion for fault detection in aircraft gas turbine engines

Soumik Sarkar[1], Soumalya Sarkar[2], Kushal Mukherjee[3], Asok Ray[4] and Abhishek Srivastav[5]

Abstract

The article addresses data-driven fault detection in commercial aircraft gas turbine engines in the framework of multi-sensor information fusion and symbolic dynamic filtering. The hierarchical decision and control structure, adopted in this article, involves construction of composite patterns, namely, *atomic patterns* extracted from single sensors, and *relational patterns* representing cross-dependence between a pair of sensors. While the underlying theories are presented along with necessary assumptions, the proposed method is validated on the NASA C-MAPSS simulation test bed of aircraft gas turbine engines; both single-fault and multiple-fault scenarios have been investigated. Since aircraft engines undergo natural degradation during the course of their normal operation, the issue of distinguishing between a fault and natural degradation is also addressed.

Keywords

Gas turbine engines, fault detection, information fusion

Date received: 15 May 2012; accepted: 26 October 2012

Introduction

Since real-time physics-based models of human-engineered complex systems (e.g. aircraft gas turbine engines) are usually computation-intensive, data-driven fault diagnosis is desirable from the perspectives of real-time decision and control. In this research, a dynamic gas-path model, namely the NASA C-MAPSS,[1] has been used to serve as the plant on the simulation test bed that generates different sets of (simulated) sensor time series data. Data-driven techniques for health monitoring of gas turbine engines either use snapshot data at a time instant from various sensors[2] or a window of time series data from selected sensor observations. The data volume is relatively small for the snapshot type of data; consequently, the computational expense is low. However, just by using snapshot data, statistical changes in the acquired information may not be adequately captured, because of possible missed detection of faults. Although this problem could be alleviated by using a window of time series data, it causes another problem of handling time series due to its data volume and the associated computational complexity; therefore, the available data must be appropriately compressed from the high-dimensional information space onto a low-dimensional feature space with reduced loss of class separability. In the previous work,[3,4] the authors proposed a nonlinear feature extraction method, called symbolic dynamic filtering (SDF), for detection of anomalies (i.e. deviations from the nominal condition) in dynamical systems. This method is shown to be particularly useful for feature extraction from time series and has been experimentally validated for real-time execution in different applications (e.g. electronic circuits[5] and fatigue damage monitoring in polycrystalline alloys[6]). Algorithms, constructed in the SDF setting, have been shown to yield superior performance in terms of early detection of anomalies and robustness to measurement noise in comparison with other existing techniques such as principal

[1]Systems Department, United Technologies Research Center, The Pennsylvania State University, University Park, PA, USA
[2]Department of Mechanical Engineering, The Pennsylvania State University, University Park, PA, USA
[3]Systems Department, United Technologies Research Center, Cork, Ireland
[4]Department of Mechanical and Nuclear Engineering, The Pennsylvania State University, University Park, PA, USA
[5]Systems Department, United Technology Research Centre, East Hartford, CT, USA

Corresponding author:
Asok Ray, Pennsylvania State University, 329 Reber Building, University Park, PA 16802, USA.
Email: axr2@psu.edu

component analysis (PCA), neural networks (NN) and Bayesian techniques.[7] Recently, in a two-part paper,[8,9] an SDF-based algorithm for detection and isolation of engine subsystem faults (specifically, faults that cause efficiency degradation in engine components) has been reported and an extension of that work to estimate simultaneously occurring multiple component-level faults has been presented in Sarkar et al.[10] Furthermore, an optimized feature extraction technique has been developed in the semantic framework in Sarkar et al.[11] However, the patterns generated from the time series of a single sensor may not carry sufficient information to correctly diagnose an evolving fault. Moreover, simultaneously occurring faults in a single subsystem or in different subsystems may generate similar signatures as observed from a single sensor. In this regard, a data-driven fault detection tool for aircraft engine systems should have the capability to characterize, quantify and interpret multiple sensor outputs.

Sensor information fusion for an aircraft engine is challenging. First of all, an aircraft engine system has sensors with different modalities (e.g. pressure, temperature, speed and acceleration), which renders data-level information fusion to be difficult due to the inherent scaling problem in sensors of different modality. On the other hand, decision level fusion generally requires an in-depth understanding of the physical system and its failure signatures in different sensor observations. In the literature, Dempster-Shafer evidence theory has been applied for engine fault diagnosis;[12] similarly, the concept of Bayesian belief networks has been used for fault diagnosis in turbofan engines.[13] Both techniques belong to the class of decision fusion at an upper level of hierarchy. Among other data-driven approaches, neural network (NN)-based techniques[14] are very popular. In a life estimation technique for aircraft actuators,[15] an NN has been used for mapping different flight regimes for in-flight aircraft engine fault diagnosis. A hybrid data-driven method involving NN and genetic algorithm (GA) was proposed in Kobayashi and Simon.[16] A regression-based approach was used for detecting anomalies in aircraft performance during cruise flight.[17]

One of the main aspects of the symbolic dynamic filtering (SDF)-based method, adopted in this article, is semantic representation of sensor data, irrespective of modality and other sensor-specific characteristics, where the patterns are represented by probabilistic finite state automata (PFSA). This approach facilitates feature-level fusion of non-homogeneous sensors and possibly other forms of information (e.g. pilot's experience and results of other case studies as semantic information); however, this article focuses on sensor fusion for fault detection in aircraft gas turbine engines. The fusion algorithms are derived based on a *semantic framework* for feature extraction and pattern classification. To handle a large volume of data in real time, a hierarchical framework for information fusion is proposed that progressively leads from machine representations of observed data to fault classification. Quantitative values of cross-dependence between different pairs of sensor observations are used in the fusion algorithm to reduce loss of information.

Deterioration in engine performance usually occurs due to two reasons: (i) natural deterioration, i.e. gradual degradation of engine components due to wear and usage, and (ii) relatively more rapid faults, i.e. abrupt deterioration of turbo machineries due to anticipated events. Therefore, an effective fault detection scheme must be able to distinguish between the faults and usual gradual degradation of a component; this is necessary to reduce the false alarm rate. This article attempts to address this issue by normal degradation/fault class assignment within the general framework of data-driven fault detection. In addition, the usage of semantic sensor fusion leads to: (i) development of a hierarchical decision engine, and (ii) formulation of an optimal sensing system strategy for fault detection in gas turbine engines.

The article is organized in five sections including the present one. Section 'Description of the C-MAPSS simulation test bed and test scenarios' describes the C-MAPSS test bed.[1] Section 'Semantic framework for multi-sensor data interpretation and fusion' explains the hierarchical framework of multi-sensor data interpretation and fusion. Section 'Results and discussion' presents the results of case studies to validate the proposed method on the C-MAPSS test bed. Finally, the article is summarized and concluded in the final section with recommendations of future work.

Description of the C-MAPSS simulation test bed and test scenarios

This section presents the C-MAPSS test bed and simulates two test scenarios. While Scenario I addresses the problem of distinguishing a fault from natural degradation in a single component, Scenario II addresses the problem of simultaneous faults in multiple-component faults. The C-MAPSS simulation test bed[1] was developed at NASA for a typical commercial-scale two-spool turbofan engine and its control system. Figure 1 shows the schematic diagram of a commercial aircraft gas turbine engine used in the C-MAPSS simulation test bed.

The engine under consideration produces a thrust of approximately 400,000 N and is designed for operation at altitude (A) from the sea level (i.e. 0 m) up to 12,200 m, Mach number (M) from 0 to 0.90, and temperatures from approximately $-50\,^{\circ}$C to $+50\,^{\circ}$C. The throttle resolving angle (TRA) can be set to any value in the range between 0° at the minimum power level and 100° at the maximum power level. The gas turbine engine system consists of five major rotating components, namely, fan (F), low

pressure compressor (LPC), high pressure compressor (HPC), high pressure turbine (HPT), and low pressure turbine (LPT), as seen in Figure 1. Apart from the rotating components, three actuators are modeled in the simulation test bed, namely, Variable Stator Vane (VSV), Variable Bleed Valve (VBV), and Fuel Pump (FP) that controls the fuel flow rate (Wf).

Given the inputs of TRA, A and M, the interactively controlled component models in the simulation test bed compute nonlinear dynamics of real-time turbofan engine operation. A gain-scheduled control system is incorporated in the engine system, which consists of speed controllers and limit regulators for engine components.

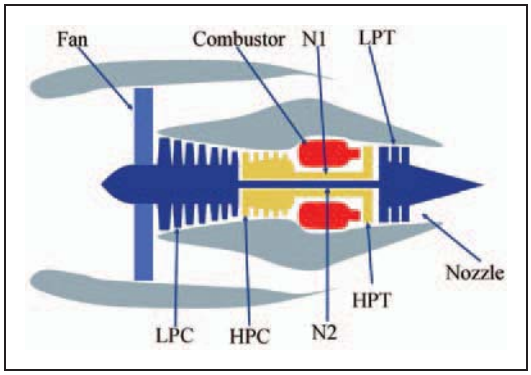

Figure 1. Gas turbine engine schematic.[1]
LPC: low pressure compressor; HPC: high pressure compressor; LPT: low pressure turbine; HPT: high pressure turbine.

Table 1. Sensor suite for the engine system.

Sensors	Description
T_{24}	LPC exit/ HPC inlet temperature
Ps_{30}	HPC exit static pressure
T_{48}	HPT exit temperature
P_{50}	LPT exit pressure
N_f	Fan spool speed
N_c	Core spool speed

LPC: low pressure compressor; HPC: high pressure compressor; HPT: high pressure turbine; LPT: low pressure turbine.

Out of the different types of sensors (e.g. pressure, temperature and shaft speed) used in the C-MAPSS simulation test bed, Table 1 lists those sensors that are commonly adopted in the Instrumentation and Control system of commercial aircraft engines, as seen in Figure 2.

In the current configuration of the C-MAPSS simulation test bed, there are 13 component level health parameter inputs, namely, efficiency parameters (ψ), flow parameters (ζ) and pressure ratio modifiers, that simulate the effects of faults and/or degradation in the engine components. Ten, out of these 13 health parameters, are selected to modify efficiency (η) and flow (ϕ) that are defined[16] as:

- $\eta \triangleq$ Ratio of actual enthalpy and ideal enthalpy changes.
- $\phi \triangleq$ Ratio of rotor tip and axial fluid flow velocities.

For the engine's five rotating components F, LPC, HPC, LPT and HPT, the five pairs (i.e. ten) respective efficiency and flow health parameters are: (ψ_F, ζ_F), (ψ_{LPC}, ζ_{LPC}), (ψ_{HPC}, ζ_{HPC}), (ψ_{HPT}, ζ_{HPT}) and (ψ_{LPT}, ζ_{LPT}). An engine component C is considered to be in the nominal condition if both ψ_C and ζ_C are equal to 1 and fault can be injected in the component C by reducing the values of ψ_C and/or ζ_C. For example, $\psi_{HPC} = 0.98$ signifies a 2% relative loss in efficiency of HPC. Actuator faults can also be injected through the scale shift parameters for the three actuators, VSV, VBV and Wf.

Scenario I: Case study of natural deterioration versus faults in a single component

In the context of gas path health monitoring of aircraft engines, two types of changes in engine

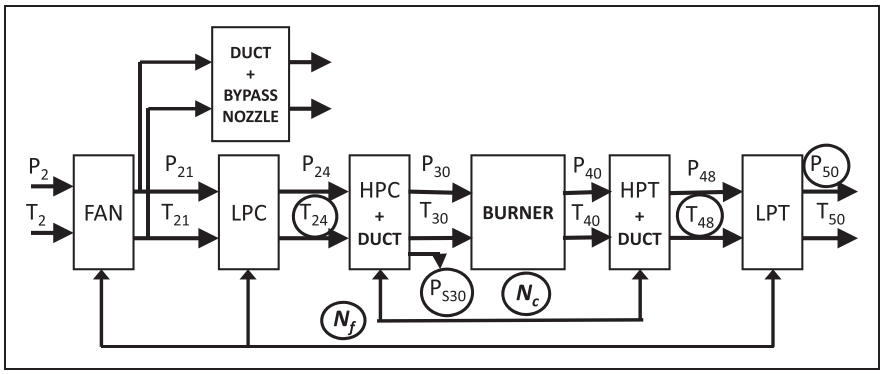

Figure 2. Schematic diagram of the C-MAPSS engine model with sensors.
LPC: low pressure compressor; HPC: high pressure compressor; LPT: low pressure turbine; HPT: high pressure turbine.

performance are considered: (i) natural deterioration, i.e. gradual degradation of engine components due to wear and usage and (ii) rapid faults, i.e. relatively abrupt degradation of turbo-machinery component(s). The fault detection method identifies the health status of an engine component at a particular time epoch, where faults need to be distinguished from the usual gradual degradation of a component. However, to achieve this goal, the health status must be monitored over several such epochs. Given a temporal profile of the health status identified by the current method, the task is to distinguish faults from the natural degradation. Typically, the health of an engine deteriorates at a relatively slow rate for natural degradation while a fault is the cause of a relatively more rapid change in the health status or degradation at a comparatively faster rate. Therefore, the gas path health management system needs to be designed to perform both functions: (1) estimation and trend monitoring of all engine health parameters over the lifetime of the engine; and (2) detection of rapid performance changes to isolate the root cause. In this section, representative simulation studies are performed to validate the capability of the proposed methodology of distinguishing the normal deterioration from faults for a given usage information. Moreover, the performance of the entire (typical) sensor suite is examined over the spectrum of faults in the engine components and actuators. Such an investigation is expected to potentially lead to construction of an optimal sensor suite and a hierarchical procedure of fault detection and isolation.

In the case study, the engines under consideration are assumed to have undergone 1000 flight cycles. Therefore, components of a nominal engine of this type may degrade to some extent from the ideal condition. A stochastic damage model has been constructed and incorporated in the C-MAPSS *Transient Test-case Generator*[18] (developed by NASA), based on the experimental data for trending the natural deterioration of the engine components. The plots in Figure 3 are typical outcomes of the model, which show the natural degradation profiles of efficiency health parameters for the five rotating components of the engine (similar profiles can be generated for the flow health parameters). For example, the performance of an engine after K cycles of flight could be considered as nominal if the changes in health parameters are within the maximum degradation limits calculated by the model after K cycles. Hence, for the nominal data samples of engine operation, health parameters of the rotating components are assumed to be within the prescribed ranges in a uniformly random fashion. Three actuators, Variable Stator Vane (VSV), Variable Bleed Valve (VBV) and Fuel Flow Rate (Wf), are assumed to be in ideal health conditions for a nominally operating engine after 1000 cycles.

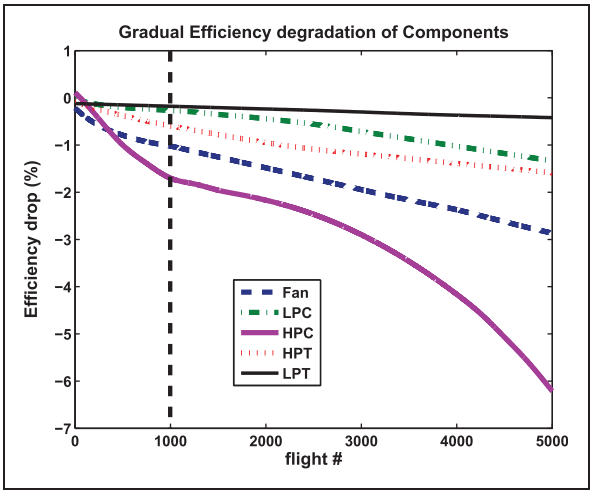

Figure 3. Natural efficiency degradation profiles for rotating components.
LPC: low pressure compressor; HPC: high pressure compressor; LPT: low pressure turbine; HPT: high pressure turbine.

Although injection of rapid/abrupt faults is described in the transient test-case generator code,[18] it is explained in this article for completeness. For all five rotating components, faults exhibit random magnitudes (F_m) and a random health parameter ratio (HPR). While F_m and HPR directly determines the change in efficiency health parameter $\psi(C)$ of a component C, a change in the flow health parameter ζ_C is determined by HPR for a given perturbation in ψ_C. Formally, the following two relations are used.

$$\delta_{\psi_C} = -\frac{F_m}{\sqrt{1+HPR^2}} \quad \text{and} \quad \delta_{\zeta_C} = \delta_{\psi_C} \cdot HPR \qquad (1)$$

where δ_{ψ_C} and δ_{ζ_C} denote the changes in ψ_C and ζ_C, respectively.

In the case study, fault magnitude (F_m) follows a random uniform distribution ranging from 1 to 7. Health parameter ratios (HPR) for Fan, LPC and HPC are uniformly distributed between 1.0 and 2.0, whereas HPRs for HPT and LPT are uniformly distributed between 0.5 and 1.0. The changes in health parameters occur from certain base values of ψ_C and ζ_C. For the present example, base values are taken as the health parameters due to maximum possible natural degradation after 1000 flight cycles. So far the actuator faults are concerned, only those due to scale shift are considered in this example. The ranges for random uniformly distributed scale shift magnitudes ($S_m(A)$ for actuator A) are: (i) 1% to 7% for $S_m(VSV)$, (ii) 1% to 19% for $S_m(VBV)$ and (iii) 1% to 7% for $S_m(Wf)$. Please see Armstrong[18] for detail reasons behind the above parameter choices.

There are 9 health condition classes considered for this study. Apart from the nominal class, 8 fault classes signify (single) faults in 5 rotating components and 3 actuators. One hundred samples are generated

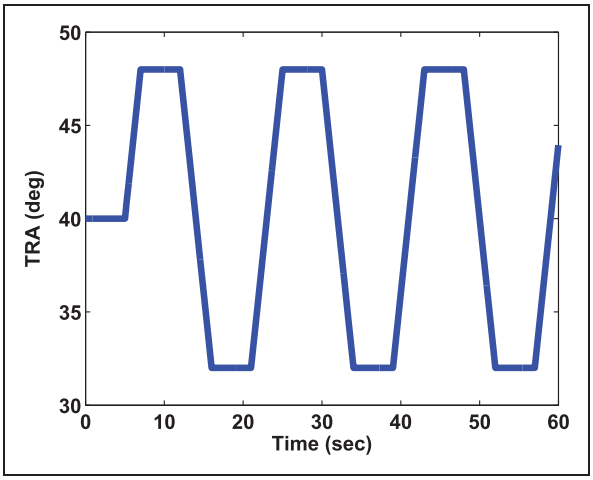

Figure 4. Profile of throttle resolving angle (TRA).

by using the above logic for each of these 9 classes, among which 50 are taken as training samples and the remaining 50 are taken as test samples. Simulation runs are conducted with the TRA inputs having truncated triangular profiles with the mean value of 40°, fluctuations within ±8° and frequency of 0.056 Hz with Mach Number 0.6 and Altitude 30,000 ft (~900 m), as shown in Figure 4. The entire commercially available sensor suite of 6 sensors (N_c, N_f, P_{50}, Ps_{30}, T_{24} and T_{48}) is used for fault detection.

Scenario II: Case study of simultaneously occurring multiple faults

Component-level fault diagnosis in an aircraft gas turbine engine involves identification of the fault type, and location and quantification of the fault level. In the C-MAPSS test bed setting, the physical fault scenarios (e.g. fouling, increased tip clearance and seal wear) are assumed to manifest themselves in affecting the efficiency and flow of the associated engine component(s). In the present case study, a simultaneous fault scenario has been considered involving two major rotating components of the engine, namely, HPC and HPT. Choice of these components has important significance from the perspective of diagnosis of simultaneously occurring faults. As seen in Figure 1, both HPC and HPT are mounted on the core shaft of the engine; hence, they have a strong mechanical interconnection. Besides, they also have electrical interconnections via the control loop. Such a strong electro-mechanical interconnection complicates the fault diagnosis problem. On the other hand, information from an HPC sensor may have strong cross-dependency with an HPT sensor and it may not be reasonable to ignore that for the purpose of fault diagnosis. The present study involves three heterogeneous, non-collocated and commonly used sensors are selected that are placed in the HPC–HPT subsystem as seen in Figure 2. The three sensors

Table 2. Sensors for fault diagnosis in the HPC–HPT subsystem.

Sensors	Description	Noise Std. (%)
Ps_{30}	HPC exit static pressure	0.50
T_{48}	HPT exit temperature	0.75
N_c	Core spool speed	0.25

HPC: high pressure compressor; HPT: high pressure turbine.

Table 3. Fault levels in HPC/HPT.

Fault level	Efficiency range
Low fault	1.0000 to 0.9867
Medium fault	0.9867 to 0.9733
High fault	0.9733 to 0.9600

HPC: high pressure compressor; HPT: high pressure turbine.

are listed in Table 2, where standard deviations of the sensor noise are provided as percentage of the operating point trim values.[19]

Remark 2.1. *The three chosen sensors are of different modalities (pressure, temperature, speed) and while the pressure sensor is located at HPC exit, the temperature sensor is located at HPT exit and the speed sensor is measuring the rotational speed of the core shaft on which both HPC and HPT are mounted. The challenge here is to identify the relational dependencies among these sensor data to enhance the fault diagnosis performance.*

Diagnosis involves both fault localization and fault level identification. The health parameters that define the health status of HPC and HPT, are the efficiency and flow health parameters, namely, ψ_{HPC}, ζ_{HPC} and ψ_{HPT}, ζ_{HPT}. Three similar fault levels are considered for both HPC and HPT. Table 3 shows the approximate ranges of efficiency health parameters under different fault levels. Corresponding flow health parameters are chosen using the same logic used in section 'Scenario I: Case study of natural deterioration versus faults in a single component'. Here, a low fault level indicates very minimal loss in efficiency and flow performance and it includes the absolute nominal health condition (i.e. $\psi_{HPC}=\zeta_{HPC}=1$ or $\psi_{HPT}=\zeta_{HPT}=1$). In the context of the case-study I, the low fault level can be considered as the nominal class as well, i.e. low level faults do not raise any alarm.

In this study, classes are defined as Cartesian products of the ranges of HPC and HPT health parameters. There are 9 (i.e. 3×3) classes of data that can be obtained when a class is uniquely defined by an HPC fault level (a range of HPC health parameters) and an

HPT fault level (a range of HPT health parameters). Hundred simulation runs of the engine system have been conducted for each class to generate data set for analysis among which 50 samples are chosen as the training set and the remaining 50 samples are kept as testing set. HPC/HPT health parameters are chosen randomly from independent uniform distributions for health parameters within the prescribed ranges given in above table. Figure 5 plots the samples generated using the above logic in the two-dimensional parameter space. Different classes of samples are marked in the boxes with respective class numbers.

For each data sample, a time series was collected for all three sensors (given in Table 2) under persistent excitation of *TRA* inputs having the same profile used in case-study I (Figure 4). The ambient conditions are chosen to be at the sea level when the engine is on the ground (i.e. altitude $A = 0.0$, Mach number $M = 0.0$) for fault monitoring and maintenance by the engineering personnel. For each experiment, the engine simulation is conducted at a frequency of 66.67 Hz (i.e. inter-sample time of 15 ms) and the length of the simulation time window is 150 s, which generate 10,000 data points. Figure 6 shows representative examples of time series data from each of the three sensors.

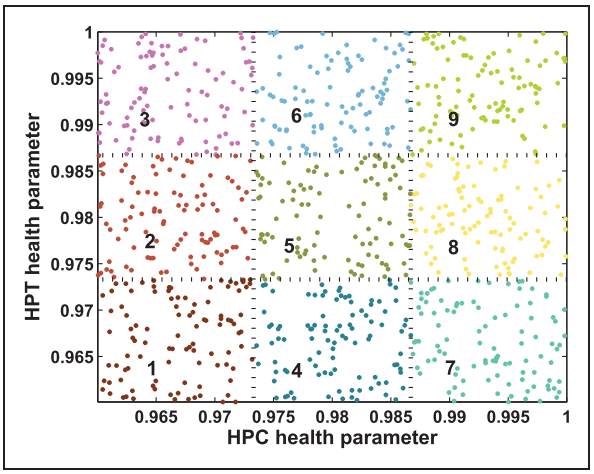

Figure 5. Fault classes for data collection and classification. HPC: high pressure compressor; HPT: high pressure turbine.

Thus, in the above context, the problem of multi-component fault diagnosis in aircraft gas turbine engines is formulated as a multi-class classification problem (in the present scenario, number of classes is 9). The following section presents a semantic framework for analysis and fusion of sensor information for fault diagnosis.

Semantic framework for multi-sensor data interpretation and fusion

A hierarchical (three-layered) semantic framework is proposed in this work for the purpose of multi-sensor data interpretation and fusion. The basic structure of this architecture is inspired by the information fusion model proposed by the Data Fusion Information Group (DGIF).[20] The lowest level of this hierarchy deals with signal conditioning, transformation and finally feature extraction for unimodal sensor data streams. In the present framework, patterns discovered from individual sensors are called atomic patterns and SDF is used to extract them. A brief review of SDF is provided in the following section for the completeness of the report.

Symbolic dynamic filtering for feature extraction

This subsection briefly describes the concepts of symbolic dynamic filtering (SDF) for extracting atomic patterns from single-sensor data. The authors have explored the concepts of symbolic dynamics and time series data partitioning to develop this computationally efficient tool, for anomaly detection in complex dynamical systems.[3,5]

Symbolic feature extraction from time series data is posed as a two-time-scale problem. The *fast scale* is related to the response time of the process dynamics. Over the span of data acquisition, dynamic behavior of the system is assumed to remain invariant, i.e. the process is quasi-stationary at the fast scale. On the other hand, the *slow scale* is related to the time span over which non-stationary evolution of the system dynamics may occur. It is expected that the features extracted from the fast-scale data will depict statistical changes between two different

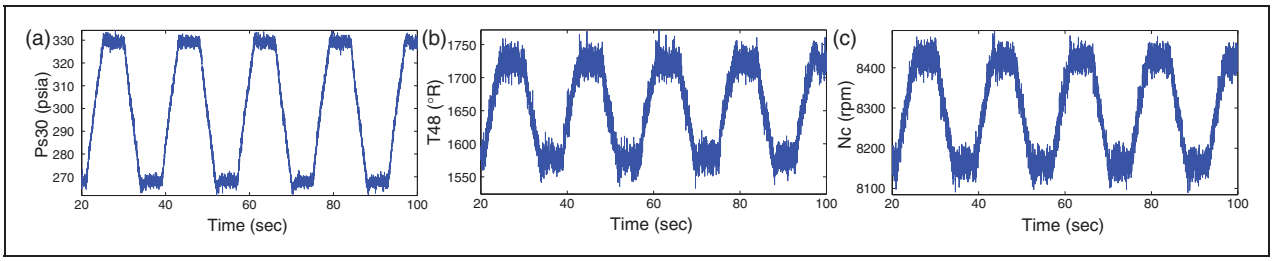

Figure 6. Representative time series observations from different sensors.

slow-scale epochs if the underlying system has undergone a change. The method of extracting features from stationary time series data is comprised of the following steps.

- Sensor time series data, generated from a physical system or its dynamical model, are collected at a slow-scale epoch and let it be denoted as $\mathbf{q}$. A compact (i.e. closed and bounded) region $\Omega \in \mathbb{R}^n$, where $n \in \mathbb{N}$, within which the stationary time series is circumscribed, is identified. Let the space of time series data sets be represented as $\mathcal{S} \subseteq \mathbb{R}^{n \times T}$, where $T \in \mathbb{N}$ is sufficiently large for convergence of statistical properties within a specified threshold. While n represents the dimensionality of the time-series, T is the number of data points in the time series. Then, $\{\mathbf{s}\} \in \mathcal{S}$ denotes a time series at the slow-scale epoch of data collection.
- Encoding of Ω is accomplished by introducing a partition $\mathbb{B} \triangleq \{B_0, \ldots, B_{(|\Sigma|-1)}\}$ consisting of $|\Sigma|$ mutually exclusive (i.e. $B_j \cap B_k = \emptyset \ \forall j \neq k$), and exhaustive (i.e. $\cup_{j=0}^{|\Sigma|-1} B_j = \Omega$) cells, where each cell is labeled by symbols $\sigma_j \in \Sigma$ and $\Sigma = \{\sigma_0, \ldots, \sigma_{|\Sigma|-1}\}$ is called the alphabet. This process of coarse graining can be executed by uniform, maximum entropy, or any other scheme of partitioning. Then, the time series data points that visit the cell B_j are denoted as σ_j $\forall j = 0, 1, \ldots, |\Sigma| - 1$. This step enables transformation of the time series data $\{\mathbf{s}\}$ to a symbol sequence $\{\mathbf{s}\}$, consisting of the symbols σ_j in the alphabet Σ.
- A probabilistic finite state automata (*PFSA*), is then constructed. The *PFSA* considered in this framework is known as D-Markov machine.[3] Formally, a state in a D-Markov machine is a symbol sequence of length D. $\mathcal{Q} = \{q_1, q_2, \ldots, q_{|\Sigma|^D}\}$ is the state set corresponding to symbol sequence $\{\mathbf{s}\}$. States represent all possible words of length D, using the symbol alphabet.

For a suite of N sensors collecting data, each sensor time series is symbolized and the set of states is constructed as described above. Let the sets of symbols describing the N data sets be denoted as $\Sigma_1, \ldots, \Sigma_N$ respectively and the corresponding D-Markov states are denoted as $\mathcal{Q}_1, \ldots, \mathcal{Q}_N$.

Definition 3.1 (D-Markov) *A D-Markov machine*[3] *with depth* $D(\geqslant 1)$ *is defined for the product space of all sensor time-series as a 4-tuple* $\mathcal{M} \triangleq (\mathcal{Q}^P, \Sigma^P, \delta, \tilde{\Pi})$ *such that*:

- $\Sigma^P = \Sigma_1 \times \Sigma_2 \times \cdots \times \Sigma_N$. *Where,* $\Sigma_i = \{\sigma_i^0, \ldots, \sigma_i^{|\Sigma_i|-1}\}$ *is the alphabet set of symbol sequence from the ith sensor.*
- $\mathcal{Q}^P = \mathcal{Q}_1 \times \mathcal{Q}_2 \times \cdots \times \mathcal{Q}_N$. *Where,* $\mathcal{Q}_i = \{q_i^1, q_i^2, \ldots, q_i^{|\Sigma_i|^D}\}$ *is the state set corresponding to symbol sequence from the ith sensor. States represent all possible words of length D, using the symbol alphabet* Σ^P.
- $\delta : \mathcal{Q}^P \times \Sigma^P \to \mathcal{Q}^P$ *is the state transition mapping that maps the transition in a symbol sequence from one state to another upon arrival of a symbol from each sensor.*
- $\tilde{\Pi}$ *is the symbol generation matrix of size* $|\mathcal{Q}^P| \times |\Sigma^P|$ *and encodes the Markovian nature of the data by describing the probability of obtaining a symbol (for each sensor)* $\sigma \in \Sigma^P$ *conditioned on current state of the system* $q \in \mathcal{Q}^P$. *In other words, elements of* $\tilde{\Pi}$ *are the probabilities* $P(\sigma_1 \in \Sigma_1, \sigma_2 \in \Sigma_2, \ldots, \sigma_N \in \Sigma_N | q_1 \in \mathcal{Q}_1, q_2 \in \mathcal{Q}_2, \ldots, q_N \in \mathcal{Q}_N)$

The D-markov PFSA described above is essentially a joint representation of the symbols (Σ^P) observed conditioned on the product space of states ($\mathcal{Q}^P$). However, for most applications, this method suffers from curse of dimensionality. The size of the product spaces of states and symbols is prohibitively large for any meaningful estimation of $\tilde{\Pi}$.

Let $\mathbb{L} = \{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_N\}$ be the universal set of *atomic* patterns. The *atomic* pattern library $\mathbb{L}$ is set of modal footprints identified from individual sensing modalities for various fault classes. Given the atomic pattern library, a popular framework for addressing information fusion is what is called the *set-theoretic* approach. In this framework, higher level patterns or contexts are modeled as subsets of $\mathbb{L}$. Thus a composite pattern, resulting from fusion of atomic patterns, is a collection of elements from $\mathbb{L}$ and the composite pattern library $\mathbb{L}^* \subset 2^{\mathbb{L}}$. The disadvantage of this approach is that it considers only modal footprints for constructing composite patterns as a *bag of atomic patterns*; relational dependencies, if any, between patterns are disregarded. However, the relational dependencies should not be ignored for many problems in practice, e.g. in the present problem of fault diagnosis of simultaneously degrading electromechanically connected aircraft engine components. Therefore, a hierarchical semantic framework for multi-sensor data interpretation and fusion is proposed that involves a common approach to information fusion going from one level to another and to include relational dependencies for composite pattern representation. Thus, the middle layer deals with the relational dependencies among atomic patterns, where relationships are modeled as the cross-dependencies among data streams from different sensors. The cross-dependencies are discovered via relational PFSA that essentially capture the dynamics of state transition in one symbol sequence (obtained from one sensor) corresponding to a symbol appearance in the second symbol sequence (obtained from another sensor). The relational patterns provide symbol-level

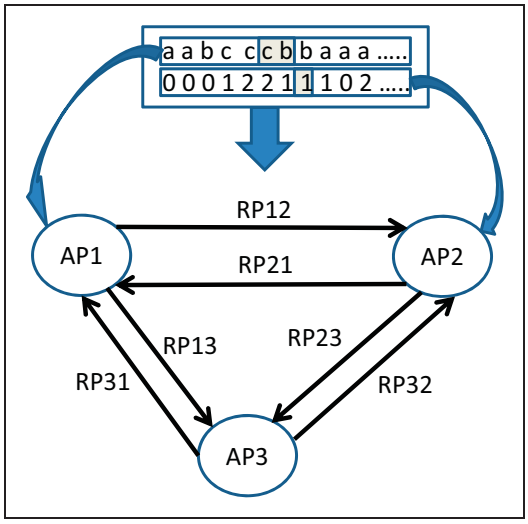

Figure 7. Composite pattern digraph.

cross-dependencies between modalities that are exploited to reduce information loss. While the atomic patterns and relational patterns are constructed based on sensor data on the fast time scale, they are combined to form composite patterns that are labeled at individual epochs on the slow time scale.

Finally, the top layer consists of higher level composite patterns (CP) that will be represented as digraphs where the atomic patterns (AP) are modeled as nodes and dependencies between nodes are modeled as relational patterns (RP). A formal definition is as follows:

Definition 3.2 (Composite pattern representation) *Let* $\mathbb{L} = \{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_N\}$ *be the atomic pattern library. Let* $\mathbb{L}^* \subseteq 2^{\mathbb{L}}$ *be the set of allowable primitives for a class. Then a composite pattern library* $H^r = \{\mathcal{H}_1^r, \mathcal{H}_2^r, \ldots, \mathcal{H}_M^r\}$ *where a composite pattern* $\mathcal{H}_i^r$ *is digraph* $\mathcal{H}_i^r = (\mathcal{L}_{V_i}, \mathcal{E}_{V_i})$; $\mathcal{L}_{V_i} \subset \mathbb{L}$ *with the index set* $V_i \subset \{1, 2, \ldots, N\}$ *and* $\mathcal{E}_i = \{\mathcal{R}_{jk} | j, k \in V_i \times V_i\}$ *is a set of relational PFSAs. The digraph representation is illustrated in Figure 7.*

The relational probabilistic finite state automata (PFSA) are discovered using xD-Markov machine construction to determine cross-dependence; the algorithm is described in the following section.

Construction of relational PFSA: xD-Markov machine

This subsection describes the construction of xD-Markov machines from two symbol sequences $\{\mathbf{s}_1\}$ and $\{\mathbf{s}_2\}$ obtained from two different sensors (possibly of different modalities) to capture the symbol level cross-dependence. A formal definition is as follows:

Definition 3.3 (xD-Markov) *Let* $\mathcal{M}_1$ *and* $\mathcal{M}_2$ *be the PFSAs corresponding to symbol streams* $\{\mathbf{s}_1\}$ *and* $\{\mathbf{s}_2\}$, *respectively. Then a xD-Markov machine is defined as a 5-tuple* $\mathcal{M}_{1\to 2} \triangleq (\mathcal{Q}_1, \Sigma_1, \Sigma_2, \delta_1, \tilde{\Pi}_{12})$ *such that*:

- $\Sigma_1 = \{\sigma_0, \ldots, \sigma_{|\Sigma_1|-1}\}$ *is the alphabet set of symbol sequence* $\{\mathbf{s}_1\}$
- $\mathcal{Q}_1 = \{q_1, q_2, \ldots, q_{|\Sigma|_1^{D_1}}\}$ *is the state set corresponding to symbol sequence* $\{\mathbf{s}_1\}$, *where* D_1 *is the depth for* $\{\mathbf{s}_1\}$
- $\Sigma_2 = \{\sigma_0, \ldots, \sigma_{|\Sigma_2|-1}\}$ *is the alphabet set of symbol sequence* $\{\mathbf{s}_2\}$
- $\delta_1 : \mathcal{Q}_1 \times \Sigma_1 \to \mathcal{Q}_1$ *is the state transition mapping that maps the transition in symbol sequence* $\{\mathbf{s}_1\}$ *from one state to another upon arrival of a symbol in* $\{\mathbf{s}_1\}$
- $\tilde{\Pi}_{12}$ *is the symbol generation matrix of size* $|\mathcal{Q}_1| \times |\Sigma_2|$; *the ijth element of* $\tilde{\Pi}_{12}$ *denotes the probability of finding jth symbol in* $\{\mathbf{s}_2\}$ *while making a transition from ith state in the symbol sequence* $\{\mathbf{s}_1\}$

In practice, $\tilde{\Pi}_{12}$ is reshaped into a vector of length $|\mathcal{Q}_1| \times |\Sigma_2|$ and is treated as the extracted feature vector that is a low-dimensional representation of the relational dependence between $\{\mathbf{s}_1\}$ and $\{\mathbf{s}_2\}$. This feature vector is called a relational pattern (RP). Figure 7 schematically describes the basic concept of the xD-Markov machine. Note, a RP between two symbol sequences is not necessarily symmetric; therefore, RPs need to be identified for both directions. Also, when both symbol sequences are same, the relational patterns are essentially the atomic pattern corresponding to the symbol sequence; i.e. xD-Markov machine reduces to a simple D-Markov machine.

The set-theoretic approach falls at one end of the spectrum of information fusion; here all relationships are excluded and any fusion is solely done in the decision-theoretic sense, where the presence (or absence) of one or more footprints can be used to estimate the probability of the fault class under consideration. The other end of the spectrum is to fuse data at the lowest level and construct machines (PFSAs) working in the product space of all sensors. This approach would be able extract modal dependencies before they are lost when constructing separate machines for individual sensor or modalities. But working in the product space has the danger of state space explosion especially if the sensors and sensing modalities are numerous, as in the case of an aircraft engine equipped with a large number of sensors.

The proposed approach is a trade-off between the two ends of the spectrum and attempts to include relational dependencies between sensing modalities, while keeping it tractable for a practical application. The hierarchical structure ensures that composite patterns are identified only when its constituting units at the lower level have been observed. The current framework constructs relations, taken only two at a time; identification of relations among higher order cliques is a topic of future work.

Results and discussion

This section presents the results obtained from the two case studies by applying the semantic framework of sensor fusion. For pattern classification applications, a time-series from the reference class is partitioned using a partitioning schemes (e.g. uniform partitioning (UP) or maximum entropy partitioning (MEP)).[3,5,21] Then, using the steps described before, a low-dimensional feature is constructed for the reference class. Similar features can be extracted from time-series data of all classes using the same partitioning. Finally a classifier is trained using features of different classes extracted from training data and can be used to classify the features from test data set. There are plenty of choices available for design of both parametric and non-parametric classifiers in Bishop[22] and Duda et al.[23] Among the parametric type of classifiers, one of the most common techniques is to consider up to two orders of statistics in the feature space. In other words, the mean feature is calculated for every class along with the variance of the feature space distribution in the training set. Then, a test feature vector is classified by using the Mahalanobis distance[24] or the Bhattacharya distance[25] of the test vector from the mean feature vector of each class. However, these methods lack in efficiency if the feature space distribution cannot be described by second order statistics (i.e. non-Gaussian in nature). In the present context, Gaussian feature space distribution cannot be ensured due to the non-linear nature of the partitioning feature extraction technique. Therefore, a non-parametric classifier, such as the k-NN classifier may a better candidate for this study[22,23]; however, in general, any other suitable classifier, such as the support vector machines (SVM) or the Gaussian mixture models (GMM) may also be used.

Validation results for Scenario I

Pertinent fault detection results for the problem described in Scenario I is presented in this subsection. For data partitioning, maximum entropy partitioning is used with alphabet size, $|\Sigma| = 5$ for all six sensors (although alphabet size does not need to be the same for different sensors). The depth for constructing *PFSA* states[3] is taken to be, $D = 1$ for both atomic and relational pattern construction and features are classified by a k-NN classifier (with $k = 5$) using the Euclidean distance metric. Table 4 provides the classification errors corresponding to all atomic and relational patterns. The cross-dependence direction is from Sensor 1 to Sensor 2 in the table. Hence, the diagonal elements represent the classification error percentages corresponding to the atomic patterns, where as the off-diagonal elements represent the classification error percentages corresponding to the relational patterns.

Table 4. Comparison of classification error percentages using atomic and relational patterns on test data set (50×9 samples); Cross-dependence direction: *Sensor* 1 → *Sensor* 2.

		Sensor 2					
		Nc	*Nf*	P_{50}	Ps_{30}	T_{24}	T_{48}
Sensor 1	*Nc*	26.22	17.78	16.89	7.11	2.89	5.78
	Nf	19.33	43.56	49.78	12.44	21.11	14.44
	P_{50}	12.00	50.22	52.00	34.22	28.44	29.78
	Ps_{30}	7.56	10.89	32.22	23.78	16.00	15.33
	T_{24}	4.22	17.78	29.78	14.22	43.78	26.44
	T_{48}	8.00	15.78	33.78	5.11	18.67	34.44

It is observed from Table 5 that the relational patterns are able to extract useful information from the perspective of fault diagnosis, which is physically meaningful due to the strong electro-mechanical interconnections among the rotating components and the actuators. Therefore, ignoring these cross-dependencies should affect the fault detection results. Finally all the patterns are concatenated to construct the overall composite pattern. The classification error on the test data set using the composite pattern is found to be 2% and the corresponding confusion matrix is given below. In a confusion matrix C, element C_{ij} denotes the frequency of classifying test sample from class i as a sample from class j.

$$\mathbf{C}_{test}^{\text{All}} = \begin{array}{c|ccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 1 & 50 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 48 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 3 & 0 & 0 & 50 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 50 & 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 & 50 & 0 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0 & 0 & 50 & 0 & 0 & 0 \\ 7 & 0 & 0 & 0 & 0 & 0 & 0 & 50 & 0 & 0 \\ 8 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 44 & 0 \\ 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 49 \end{array}$$

The above exercise is adequate while treating the engine as a black-box and using all sensors blindly for fault detection. However, deeper questions regarding relationships among fault locations and sensor locations or the optimal sensor-suite still remain. These two issues are briefly discussed in the sequel.

Building a hierarchical decision engine. From the physical understanding of the engine model, it is clear that there may be variation in sensitivity of different sensors to different fault conditions (locations). For example, detection accuracy (false alarm) for faults

in components mounted on the core shaft may be higher (lower) while using the *Nc* sensor. However, from the previous observations, it is evident that just using *Nc* may not be enough. This issue will be more clear with the following two concrete examples. Figures 8 and 9 are two bar charts plotting the detection accuracies (in percentage) for HPT and VBV fault conditions respectively while using different atomic and relational patterns. In this case, (as it is seen in the figure) there are 36 different patterns (6 atomic and 30 relational) available for 6 sensors.

Figure 8 shows that the relational patterns involving *Nc* and T_{24} perform very well in detecting HPT faults. This can be explained from the physical understanding of the engine lay out. First of all, *Nc* monitors the speed of the core shaft on which HPT is mounted. On the other hand, T_{24} monitors the temperature at HPC inlet and HPC is mechanically connected to HPT as it is mounted on the core shaft as well. So naturally, relational patterns involving these two sensors should perform well in detecting HPT faults. However, the faults, which do

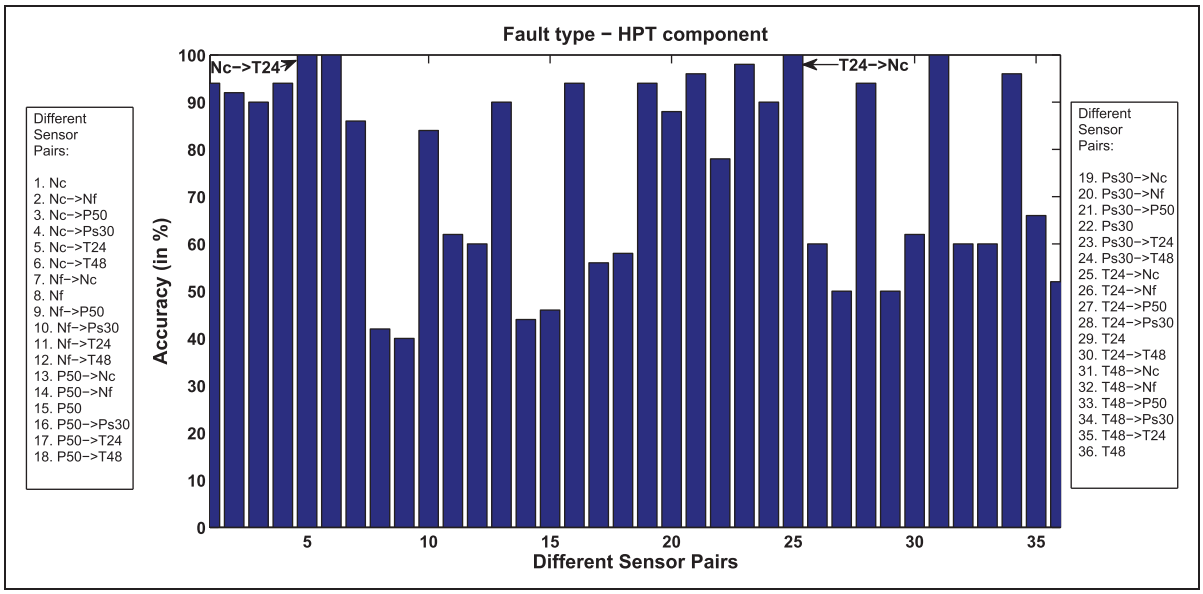

Figure 8. HPT fault detection accuracy for different atomic and relational patterns.
HPT: high pressure turbine.

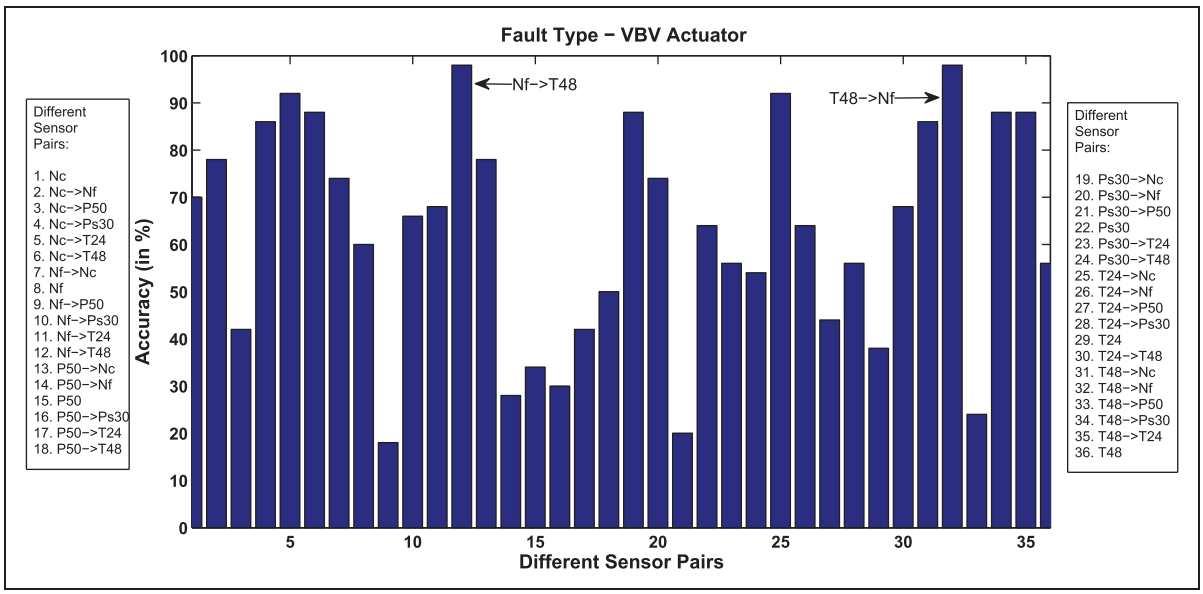

Figure 9. VBV fault detection accuracy for different atomic and relational patterns.
VBV: variable bleed valve.

not affect the outputs of Nc and T_{24} sensors, are not expected to be detected by the respective relational patterns. Therefore, it is necessary to select appropriate sensors for detection of faults of different types.

In the second example, as shown by Figure 9, relational patterns involving Nf and T_{48} perform very well in detecting variable bleed valve (VBV) faults. VBV is used to control the pressure of the gas flowing from LPC to HPC. It is a modulating valve and actuator assembly that bleeds off excess air to the atmosphere as necessary to prevent turbine surge. Therefore, it is natural that Nf, that monitors speed of fan shaft (LPC is mounted on the fan shaft) and T_{48}, that monitors HPT exit temperature (VBV essentially prevents turbine surge) perform very well in detecting VBV faults.

The above discussion leads to the idea of building a hierarchical decision engine. In this framework, there will be a hierarchical strategy of isolating the fault location by going through decisions made by sensor sets that are much smaller in size compared to the full sensor suite. For example, (referring to the two examples above) if patterns from Nc and T_{24} determine that a fault has occurred in VBV, decision confidence can be enhanced based on the decision made by patterns from Nf and T_{48}. And if the decision regarding the fault location is supported by the corresponding most efficient sensor group, then the user does not need to go for investigating remaining patterns. Thus, this procedure can handle data from large sensor suites. However, another interesting question remains: Whether there exists an optimal sensor suite (smaller than the full sensor suite) that performs well enough in all fault conditions. This issue is discussed in the sequel.

Optimal sensor suite. Building a hierarchical decision engine may be very useful when one has considerable physical understanding of the system and the interconnection characteristics among its sub-systems. This approach can also handle the issue of scalability. However, given a large volume of training data, it is possible to identify an optimal sensor suite that performs well enough for all fault conditions. Although, this may require handling large dimensional composite patterns all the time, it does not require much insight regarding the physics of the system. Here is an example optimization procedure for the current problem.

Let the detection accuracy (in fraction, i.e. varies from 0 to 1) of pattern P for a fault class i be denoted as $D_i(P)$ and the minimum allowable accuracy be denoted as D_T. With these notations, a pattern will be called an optimal pattern P^*, if it satisfies the following condition

$$\min_i D_i(P) > D_T \tag{2}$$

This is essentially placing a bound on the worst performance of a pattern. Therefore, the set of optimal patterns $\{P^*\}$ will increase with decrease in the value of D_T. An alternate cost function may be the average performance of a pattern. In the present study, considering $D_T = 0.85$, one can obtain that three relational patterns are optimal, namely Nc to T_{24}, T_{24} to Nc and Nc to Ps_{30}. The following confusion matrix is obtained by using the three sensors involved in these relational patterns. Note that to maintain the structure of a composite pattern, all three atomic and six relational patterns that are generated by these three sensors have been used.

$$\mathbf{C}_{test}^{\text{Opt}} = \left(\begin{array}{c|ccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 1 & 50 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 48 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 3 & 0 & 0 & 50 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 50 & 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 & 50 & 0 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0 & 0 & 50 & 0 & 0 & 0 \\ 7 & 0 & 0 & 0 & 0 & 0 & 0 & 49 & 1 & 0 \\ 8 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 48 & 0 \\ 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 49 \end{array}\right)$$

The overall classification error is reduced from 2% to 1.33%, which is the consequence of leaving out quite a few non-optimal patterns that may be useful for detecting some other faults. This is somewhat counterintuitive from an information theoretic point of view as more patterns should provide more information. However, in the present classification setting, non-optimal patterns may incorporate more ambiguities in decision making.

Validation results for Scenario II

This subsection presents pertinent fault diagnosis results for the Scenario II. The signal processing specifications remains same as in Scenario I. For a particular health parametric condition, three atomic patterns are generated from three sensor observations and six relational patterns are generated by extracting pairwise directed cross-dependencies. Finally all the patterns are concatenated to construct the overall composite pattern. The classification error on the test data set using the composite pattern is found to be 11.56%. Similar to Table 4, Table 5 provides the classification errors corresponding to all atomic and relational patterns.

It is observed from the above table that the relational patterns are able to extract useful information from the perspective of fault diagnosis. The problem in the present study has been posed in such a way that

Table 5. Comparison of classification error percentages using atomic and relational patterns on test data set (50 × 9 samples); Cross-dependence direction: *Sensor* 1 → *Sensor* 2.

		Sensor 2		
		Ps_{30}	T_{48}	Nc
Sensor 1	Ps_{30}	56.44	15.11	41.56
	T_{48}	12.89	56.89	16.00
	Nc	25.11	11.78	52.22

the sensor information from different sensors actually have cross-dependencies due to strong electro-mechanical interconnections between HPC and HPT. Therefore, ignoring these cross-dependencies should affect the fault diagnosis results. This result once again confirms the conjecture and shows that the xD-Markov machine construction can extract those cross-dependencies. It should be noted that the fault diagnosis algorithm is completely data-driven and has no model information. Therefore, the result is significantly encouraging. Representative classification confusion matrices corresponding to atomic patterns from Sensors T_{48}, Nc and their relational patterns of both directions are provided below.

$$\mathbf{C}_{test}^{T_{48}} = \begin{array}{c|ccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 1 & 47 & 1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 18 & 2 & 22 & 3 & 0 & 5 & 0 & 0 \\ 3 & 0 & 1 & 12 & 0 & 15 & 5 & 11 & 6 & 0 \\ 4 & 11 & 11 & 0 & 21 & 5 & 0 & 2 & 0 & 0 \\ 5 & 0 & 6 & 15 & 2 & 15 & 2 & 5 & 5 & 0 \\ 6 & 0 & 0 & 4 & 0 & 0 & 21 & 0 & 16 & 9 \\ 7 & 0 & 5 & 11 & 3 & 17 & 2 & 11 & 1 & 0 \\ 8 & 0 & 0 & 9 & 0 & 3 & 19 & 2 & 13 & 4 \\ 9 & 0 & 0 & 0 & 0 & 0 & 11 & 0 & 3 & 36 \end{array}$$

$$\mathbf{C}_{test}^{Nc} = \begin{array}{c|ccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 1 & 17 & 0 & 0 & 17 & 8 & 0 & 1 & 7 & 0 \\ 2 & 1 & 23 & 1 & 0 & 10 & 6 & 0 & 1 & 8 \\ 3 & 0 & 2 & 40 & 0 & 0 & 6 & 0 & 0 & 2 \\ 4 & 11 & 0 & 0 & 29 & 0 & 0 & 7 & 3 & 0 \\ 5 & 6 & 8 & 0 & 1 & 19 & 0 & 0 & 6 & 10 \\ 6 & 0 & 5 & 16 & 0 & 2 & 17 & 0 & 0 & 10 \\ 7 & 0 & 0 & 0 & 14 & 0 & 0 & 35 & 1 & 0 \\ 8 & 10 & 0 & 0 & 9 & 11 & 0 & 2 & 17 & 1 \\ 9 & 1 & 6 & 4 & 0 & 5 & 13 & 0 & 3 & 18 \end{array}$$

$$\mathbf{C}_{test}^{T_{48}->Nc} = \begin{array}{c|ccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 1 & 47 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 46 & 1 & 0 & 3 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 43 & 0 & 0 & 7 & 0 & 0 & 0 \\ 4 & 1 & 0 & 0 & 44 & 4 & 0 & 1 & 0 & 0 \\ 5 & 0 & 3 & 1 & 0 & 40 & 1 & 0 & 5 & 0 \\ 6 & 0 & 0 & 2 & 0 & 0 & 39 & 0 & 0 & 9 \\ 7 & 0 & 0 & 0 & 4 & 1 & 0 & 43 & 2 & 0 \\ 8 & 0 & 0 & 0 & 0 & 7 & 2 & 1 & 36 & 4 \\ 9 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 5 & 40 \end{array}$$

$$\mathbf{C}_{test}^{Nc->T_{48}} = \begin{array}{c|ccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 1 & 49 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 40 & 2 & 3 & 5 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 48 & 0 & 0 & 2 & 0 & 0 & 0 \\ 4 & 6 & 1 & 0 & 40 & 3 & 0 & 0 & 0 & 0 \\ 5 & 0 & 3 & 0 & 2 & 43 & 0 & 0 & 2 & 0 \\ 6 & 0 & 0 & 3 & 0 & 0 & 45 & 0 & 0 & 2 \\ 7 & 0 & 0 & 0 & 1 & 0 & 0 & 46 & 3 & 0 \\ 8 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 40 & 7 \\ 9 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 2 & 46 \end{array}$$

A close observation reveals similarity of fault signatures on single sensor data for two completely different fault conditions (please see Figure 5 for fault classes), e.g. fault signatures of Class 4 (HPC medium fault, HPT low fault) has been confused with Class 1 (HPC low fault, HPT low fault) for both sensors T_{48} and Nc individually. However, this ambiguity can be removed by using relational pattern directed from T_{48} to Nc.

Summary, conclusions and future plans

Summary and conclusions

This article presents feature-level fusion of multiple sensor data, which is very important for data-driven fault detection techniques. A symbolic dynamic filtering (SDF)-based methodology has been adopted for data-driven detection of component level faults and actuator faults in aircraft gas turbine engines via multi-sensor data interpretation and fusion. In the proposed method, the abstract semantic representation of sensor data enables feature level fusion of heterogeneous and disparate sensors. It is also shown that identification of cross-dependencies among different sensors mitigates loss of significant information compared to set-theoretic information fusion methods. The hierarchical architecture of this method reduces computational complexity, allowing real-time operability.

Scalability in terms of fault classes and number of sensors is always a critical issue for data-driven methods of fault diagnosis. The current algorithm is formulated generally enough to accommodate high number of fault classes possibly at the expense of a larger alphabet size and depth. On the other hand, the basic hierarchical architecture of this framework allows a tractable solution to the problem of dealing with a large number of sensors and mitigates the problem of dimensionality explosion, because upward movements in the knowledge-base levels may occur only after the structure of the lower levels has been identified.

The fault detection method described in this article identifies the health status of an engine component at a given slow-scale epoch. In a real life scenario, faults need to be distinguished from the usual gradual degradation of a component. Typically, the engine health deteriorates at a slow rate for usual degradation while a fault is characterized by a relatively more rapid change in the health status and/or degradation at a comparatively faster rate. A method for distinguishing natural degradation from faults has been developed on a data-driven framework.

Future research directions

Although the method presented in the report has been successfully validated by execution on the C-MAPSS test bed, various operating conditions need to be investigated in real time for potential in-flight applications. Apart from this important issue, the following research areas, specific to semantic sensor fusion, need to be validated on a real-life engine test bed.

- Development of algorithms to extract relational dependencies among three or more symbol sequences.
- Exploration of other statistical analysis tools (e.g. Copula distribution[26]) as an alternative to extraction of relational dependencies.
- Investigation of the effects of mis-synchronization among sensor observations on the proposed xD-Markov machine.
- Comparative evaluation of semantic information fusion framework with other information fusion techniques (e.g. Dempster-Shafer or Bayesian network) approaches.
- Construction of a *Hierarchical Decision Engine* to enhance successful detection and reduce false alarm rates while using a sensor suite.
- Identification of *Optimal Sensor Suite* (i.e. a small number of strategically placed sensors) that can serve the purpose of system fault detection and isolation maximally.
- Fault Diagnosis during *Take-off*, *Climb and Landing* using C-MAPSS transient test case generator model developed at NASA.[18]

Funding

This work has been supported in part by NASA under Cooperative Agreement No. NNX07AK49A, by the U.S. Army Research Laboratory and the U.S. Army Research Office under Grant No. W911NF-07-1-0376.

Acknowledgments

The authors acknowledge the benefits of discussion with Don Simon and Jeff Armstrong at NASA Glenn Research Center. Any opinions, findings and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

References

1. Frederick DK, DeCastro JA and Litt JS. *UserŠs guide for the commercial modular aero-propulsion system simulation (C-MAPSS)*. NASA/TMŮ2007-215026, October, 2007.
2. Saxena A, Celaya J, Balaban E, et al. Metrics for evaluating performance of prognostic techniques. In: *Proceedings of International Conference on Prognostics and Health Management (PHM08)*. Denver, CO, USA, 2008, pp.1–17.
3. Ray A. Symbolic dynamic analysis of complex systems for anomaly detection. *Signal Process* 2004; 84(7): 1115–1130.
4. Jin X, Gupta S, Mukherjee K, et al. Wavwlet-based feature extraction using probabilistic finite state automata for pattern classification. *Pattern Recogn* 2011; 44(7): 1343–1356.
5. Rajagopalan V and Ray A. Symbolic time series analysis via wavelet-based partitioning. *Signal Process* 2006; 86(11): 3309–3320.
6. Gupta S, Ray A and Keller E. Symbolic time series analysis of ultrasonic data for early detection of fatigue damage. *Mech Syst Signal Process* 2007; 21(2): 866–884.
7. Rao C, Ray A, Sarkar S, et al. Review and comparative evaluation of symbolic dynamic filtering for detection of anomaly patterns. *Signal, Image Video Process* 2009; 3(2): 101–114.
8. Gupta S, Ray A, Sarkar S, et al. Fault detection and isolation in aircraft gas turbine engines: Part i - underlying concept. *Proc IMechE, Part G: J Aerospace Engineering* 2008; 222(3): 307–318.
9. Sarkar S, Yasar M, Gupta S, et al. Fault detection and isolation in aircraft gas turbine engines: Part II - validation on a simulation test bed. *Proc IMechE, Part G: J Aerospace Engineering* 2008; 222(3): 319–330.
10. Sarkar S, Rao C and Ray A. Statistical estimation of multiple faults in aircraft gas turbine engines. *Proc IMechE, Part G: J Aerospace Engineering* 2009; 223(4): 415–424.
11. Sarkar S, Mukherjee K, Jin X, et al. Optimization of symbolic feature extraction for pattern classification. *Signal Process* 2012; 92(3): 625–635.
12. Basir O and Yuan X. Engine fault diagnosis based on multi-sensor information fusion using dempsterŮ shafer evidence theory. *Inform Fusion* 2007; 8: 379–386.
13. Romessis C and Mathioudakis K. Bayesian network approach for gas path fault diagnosis. *J Eng Gas Turb Power* 2006; 128: 64–72.

14. Volponi A, Brotherton T, Luppold R, et al. Development of an information fusion system for engine diagnostics and health management. In: *NASA/TMŮ2004-212924, 39th Combustion/21th Airbreathing Propulsion/21st Propulsion Systems Hazards/ 3rd Modeling and Simulation Joint Subcommittee Meeting*. Colorado Springs, CO, USA, 1–5 December, 2003.
15. Byington C, Watson M and Edwards D. Data-driven neural network methodology to remaining life predictions for aircraft actuator components. In: *Proceedings of IEEE Aerospace Conference*. Big Sky, MT, USA, March, 2004.
16. Kobayashi T and Simon DL. A hybrid neural network-genetic algorithm technique for aircraft engine performance diagnostics. In: *37th Joint Propulsion Conference and Exhibit cosponsored by the AIAA, ASME, SAE and ASEE*. Salt Lake City, UT, USA, 2001.
17. Chu E, Gorinevsky D and Boyd S. Detecting aircraft performance anomalies from cruise flight data. In: *AIAA Infotech Aerospace Conference*. Atlanta, GA, USA, April 2010.
18. Armstrong J. *Users guide for the transient test case generator*. NASA GRC Internal Report, September 2009.
19. Simon D and Garg S. Optimal tuner selection for Kalman filter-based aircraft engine performance estimation. *J Eng Gas Turb Power* 2010; 132(3): 031601.
20. Blasch E, Kadar I, Hintz K, et al. Resource managemanet coordination with level 2/3 fusion issues and challenges. *IEEE A&E Sytems Magazine*, March 2008, pp. 32–46.
21. Subbu A and Ray A. Space partitioning via hilbert transform for symbolic time series analysis. *Appl Phys Lett* 2008; 92(8): 084107–1–084107–3.
22. Bishop CM. *Pattern recognition and machine learning (Information science and statistics)*. Secaucus, NJ: Springer-Verlag, 2006.
23. Duda R, Hart P and Stork D. *Pattern classification*. New York, NY: John Wiley, 2001.
24. McLachlan GJ. *Discriminant analysis and statistical pattern recognition*. Wiley Series in Probability and Statistics. New York, NY: Wiley-Interscience, 2004.
25. Choi E and Lee C. Feature extraction based on the Bhattacharyya distance. *Pattern Recogn* 2003; 36: 1703–1709.
26. Trivedi P and Zimmer D. Copula modeling: An introduction for practitioners. *Foundations Trend Econom* 2005; 1(1): 1–111.