This is a repository copy of *Gas turbine engine condition monitoring using Gaussian mixture and hidden Markov models*.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/135531/

Version: Published Version

Gas Turbine Engine Condition Monitoring Using Gaussian Mixture and Hidden Markov Models

William R. Jacobs [1], Huw L. Edwards [2], Ping Li [3], Visakan Kadirkamanathan [4] and Andrew R. Mills[5]

[1,3,4,5] *Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, S1 3JD, UK*
w.jacobs@sheffield.ac.uk
a.r.mills@sheffield.ac.uk

[2] *Rolls-Royce Plc, Derby, Uk*
Huw.Edwards@controlsdata.com

ABSTRACT

This paper investigates the problem of condition monitoring of complex dynamic systems, specifically the detection, localisation and quantification of transient faults. A data driven approach is developed for fault detection where the multidimensional data sequence is viewed as a stochastic process whose behaviour can be described by a hidden Markov model with two hidden states — i.e. 'healthy / nominal' and 'unhealthy / faulty'. The fault detection is performed by first clustering in a multidimensional data space to define normal operating behaviour using a Gaussian-Uniform mixture model. The health status of the system at each data point is then determined by evaluating the posterior probabilities of the hidden states of a hidden Markov model. This allows the temporal relationship between sequential data points to be incorporated into the fault detection scheme. The proposed scheme is robust to noise and requires minimal tuning. A real-world case study is performed based on the detection of transient faults in the variable stator vane actuator of a gas turbine engine to demonstrate the successful application of the scheme. The results are used to demonstrate the generation of simple and easily interpretable analytics that can be used to monitor the evolution of the fault across time.

1. INTRODUCTION

As modern engineering systems become increasingly complex, there has been a significant growth in the need for sophisticated condition monitoring procedures to ensure reliable operation. Condition monitoring can provide information to support condition-based (rather than schedule-based) maintenance so as to optimize operations and equipment uptime, and maximize cost efficiency. This may involve a variety of tasks such as fault detection and isolation, degradation and anomaly identification, or prediction of impending failures.

The detection of incipient transient faults - faults that are observed over a short time scale, the trend of which grow in magnitude over time - is a particular challenge in condition monitoring due to the short time scales and low magnitudes by which a system deviates from its normal behaviour. Detection and monitoring of such faults is of great importance because they are often observed as a precursor to failure, which may result in an unscheduled withdrawal from service to perform a maintenance action. Successful condition monitoring can lead to pre-emptive fault diagnosis and accurate time to failure estimates and hence reduce asset downtime by the optimisation of maintenance schedules.

Given the perfect analytical model of a system, detection of abnormal transient behaviour can be easily performed by residual analysis, see *e.g.* (Chen & Patton, 1999; Ping Li & Kadirkamanathan, 2001; Ding, 2013; Isermann, 2011). In practice, however, such systems often display complex nonlinear behaviour and it is often difficult, if not impossible, to design a reliable analytical system model due to system complexity, high dimensionality and multiple operating conditions. In such a case, it may be more appropriate (or essential) to take a data-driven approach.

Data-driven techniques, often named as data mining or machine learning, make use of data collected during normal operation, or specifically designed experiments, in order to build statistical models of system behaviour. To detect anomalous behaviour, many data-driven techniques require the existence of target data - data known to contain faults - such techniques are named supervised learning. However, it is often the case, such as for safety critical systems (aero gas turbine engines (GTEs), industrial power generation *etc.*) that a vast quantity of data under nominal conditions is available but target

data is very limited, or no such data exists. In this case we must consider un-supervised/self-supervised learning. Under this scenario the system's normal behaviour is characterised, deviation from which indicates a system fault, referred to as novelty detection.

In recent years, data driven anomaly detection techniques have become increasingly popular for transient fault detection (Chandola, Banerjee, & Kumar, 2009; Ge, Song, & Gao, 2013; Pimentel, Clifton, Clifton, & Tarassenko, 2014). A variety of schemes exist which can be broadly separated into three categories: Reconstruction based, boundry based and density based. Reconstruction based methods attempt to model the underlying system behaviour. Fault detection is achieved through monitoring the reconstruction error. Models can either be physics based or data-driven. Recently, much interest has been seen in the use of deep neural networks (Yan & Yu, 2015; Liao, Jin, & Pavel, 2016). The majority of DNN methods require labelled training data, however, self supervised transient fault detection has been achieved using deep auto-encoders (Suh, Chae, Kang, & Choi, 2016; Fan, Xiao, Zhao, & Wang, 2018). Deep auto-encoders attempt to encode the input signals into a latent space which is then reconstructed at the output. Reconstruction based methods suffer from the presence of unmeasured disturbances and complex system behaviours that may lead to residuals that are larger than the magnitude of the fault behaviour to be detected, limiting performance in many real world systems.

Boundary based methods construct a decision boundary to classify normal/abnormal samples. These methods have been dominated by sparse vector machines (SVMs) (Jena & Panigrahi, 2014; Görnitz, Lima, Müller, Kloft, & Nakajima, 2017), where the boundary is constructed as the smallest hypersphere that contains the (majority) of the training data. Density based methods test if a sample data point belongs to the underlying data distribution in a probabilistic manner. Simple aproaches are based on statistical outlier detection (Barnett & Lewis, 1974). For more complex data distributions mixture models (McLachlan & Peel, 2004) and kernel density estimators (Vincent & Bengio, 2003) can be used. In both boundary and density based methods, the temporal characteristics of anomalous data is not naturally incorporated. A popular approach to include such time dependencies is the hidden Markov model (HMM) (Bishop, 2006) where the system is assumed to be a Markov pocess with hidden states. The HMM has wide applicability and remains a topic of active research (Görnitz, Braun, & Kloft, 2015; Li, Pedrycz, & Jamal, 2017).

Fleet level monitoring of asset condition poses a challenge for all transient fault detection approaches. Each asset may be designed to operate in the same manner, however, factors such as ageing and environmental operating conditions lead to significant variation in the statistics of collected data across assets and through time. Defining nominal behaviour, from which small transient deviations indicate a fault, hence becomes problematic. A further problem is that the collected data is too large to be manually checked for faults and so cannot be assumed to be healthy. In order to detect such faults, data driven methods are required which do not depend on supervised learning, can be tailored to represent each data set individually, and do not rely on the existence of a healthy data set.

This paper is concerned with the problem of developing methods for the detection and localisation of previously unseen transient faults in unbalanced data sets, where normal behaviour can be well defined, but no a priori information about fault modes is available. Training data may therefore contain fault data samples. Significant variation in normal behaviour is observed across data sets, such that a global model cannot be defined. The problem is addressed by the development of a machine learning based method using Gaussian-Uniform mixture models (G-UMMs) and HMMs. The proposed scheme avoids the problems associated with time series models, either first principles or data driven, and unmeasured disturbances are characterised by the variance in the G-UMM. Temporal information relating to anomalous behaviour is retained through the application of the HMM. The method relies on the assumption that transient faults are observed as outliers to the normal data distribution.

The developed condition monitoring scheme is demonstrated using a case study based on the detection of transient faults in the variable stator vane of an aero gas turbine engine. The method is demonstrated to perform well for the detection of synthetic faults incorporated into a real data set. Further results are presented showing the performance of the method on a healthy data set as well as the detection of a real fault event.

The rest of paper is organized as follows. In Section 2 the G-UMM and HMM are introduced, and the transient detection scheme is detailed. In Section 3 the variable stator vane system is introduced and the relevant transient fault modes are described. The results of the case study are given in Section 4. Finally, concluding remarks are made in Section 5.

2. METHODOLOGY

The method developed in this paper uses machine learning-based techniques (Bishop, 2006) to perform transient fault detection for complex systems. Specifically, Gaussian mixture classification and HMMs are employed. In the remainder of this section, these two techniques are introduced first, and then the new fault detection method is presented.

2.1. Gaussian mixture models (GMMs)

A mixture model is a statistical model for representing data sets which display behaviour that cannot be well described by a single standard distribution. It allows a complex probability distribution to be built from a linear superposition of simpler components. Gaussian distributions are the most common choice as mixture components because of the mathematical simplicity of parameter estimation as well as their ability to perform well in many situations, particularly in the presense of measurement noise (Dempster, Laird, & Rubin, 1977). A GMM is a linear superposition of Gaussian densities of the form (Bishop, 2006):

$$P(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \boldsymbol{\Sigma}_k) \qquad (1)$$

where $P(\mathbf{x})$ denotes the probability density function over the data $\mathbf{x}$, $\mathcal{N}(\mathbf{x}|\mu_k, \boldsymbol{\Sigma}_k)$ denotes the k'th Gaussian distribution with mean μ_k and covariance $\boldsymbol{\Sigma}_k$. π_k is a weighting on each component named the mixing coefficient. K is the total number of Gaussian mixtures. The GMM can be interpreted in terms of discrete hidden or latent variables where the discrete hidden variables can be viewed as defining assignments of data points to specific components of the mixture, and hence can be used to classify data as belonging to each distribution with a given probability or 'responsibility' denoted $r_{i,k}$ for the i'th data point in $\mathbf{x}$ belonging to the k'th component.

The GMM can represent an arbitrarily complex distribution given a sufficient quantity of Gaussian components. However, care must be taken to avoid over-fitting to the training data, such that K should be kept sufficiently small (McKenzie & Alder, 1994).

2.1.1. Robust classification with G-UMMs

Gaussian distributions and hence Gaussian mixture components are well known to be non-robust (Mt, 2005). Outliers in the data will skew the estimated distributions. Furthermore, the transient behaviour to be detected will be observed as an outlier to the normal data distribution by its very definition. One method for enforcing robustness to outliers is to use a mixture of Student-t distributions rather than Gaussians (Peel & McLachlan, 2000), however, as shall be seen shortly, there is a significant advantage in classifying outliers as separate from the normal data distribution. On this basis a modification is made to the mixture model by including a single uniform distribution component such that equation (1) becomes

$$P(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \boldsymbol{\Sigma}_k) + \pi_{K+1}\mathcal{U}(\mathbf{x}|a, b), \qquad (2)$$

where $\mathcal{U}(\mathbf{x}|a, b)$ denotes the uniform distribution over the interval $[a, b]$. The Gaussian-Uniform mixture model has the flexibility to assign outlying data points to the uniform distribution with a large probability. The statistics of the Gaussian distributions can then be estimated with little influence from the outlying data points. The addition of the uniform hence makes the model robust to the presence of outliers as well as correctly classifying them as such.

The probability that a point is assigned to the uniform distribution, $P_{\mathcal{U}} = \mathcal{U}(\mathbf{x}|a, b)$, is determined by the parameter π_{K+1}, to be estimated, as well as the choice of the interval $[a, b]$. However, it is simpler both mathematically and intuitively to directly assign a probability value. When working with probability distributions within the exponential family, it is often convenient to consider log probabilities in order to simplify the computations. This is the case with expectation maximisation (EM) for GMMs (discussed in the following section), as such it is also convenient to consider a log probability for the uniform distribution, $log(P_{\mathcal{U}})$, which acts as a tuning parameter which, along with the weight π_{K+1}, affects how outlying data points are classified. Increasing $log(P_{\mathcal{U}})$ may cause more points to be assigned to the $K + 1$'th mixture, and hence as outliers, and vice-versa.

The addition of the uniform distribution to the GMM has the further property of collecting all outlying data points into a single cluster. This property is necessary for the particular implementation of the HMM used here. The interaction of the uniform component and the HMM is discussed in Section 2.2.

An example of a G-UMM applied to a complex distribution formed by the linear superposition of three bivariate Gaussians in the presence of outliers is shown in Figure 1. The true distribution (Left panel) is formed by sampling from each Gaussian distribution and outliers are drawn from a uniform distribution, underlying distributions are represented by 95% confidence intervals (Black line). 200 data points are drawn from each Gaussian distribution and 60 outliers are drawn from a uniform distribution leading to a toal of 660 training data points. Gaussian distributions estimated by the G-UMM (middle panel) are consistent with the true distribution, leading to accurate assignment of outliers. The GMM (right panel) fails to identify one of the true distribution components leading to an extra Gaussian distribution taking responsibility for some of the outliers. The example serves to demonstrate the robust estimation of the GMM in the presence of outliers.

2.1.2. Parameter estimation

Training the G-UMM requires estimating the model parameters π_k, μ_k and Σ_k for $k = 1, \ldots, K + 1$. A popular, and commonly used method, for finding the maximum likelihood solution for the parameter estimates of a GMM is the EM algorithm (Dempster et al., 1977). Although many alternative techniques exist (Redner, Walker, Mathematics, & Review, 1984; Nasios & Bors, 2006), the EM algorithm has significant
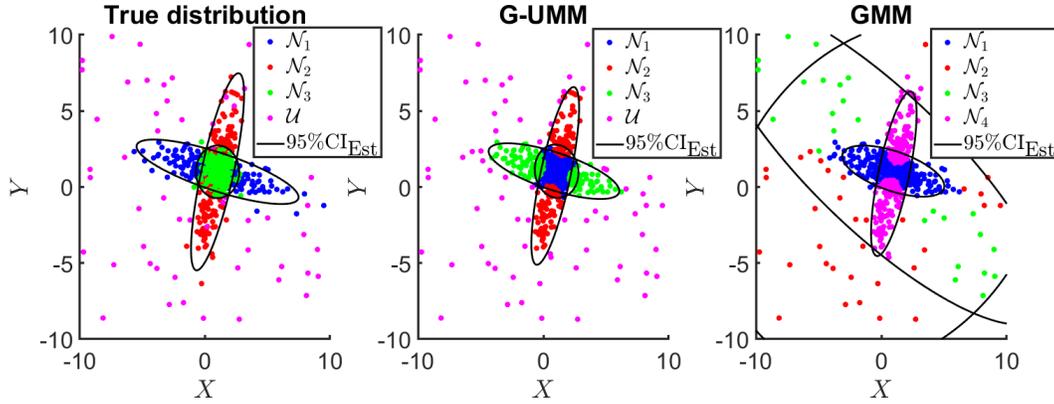
Figure 1. The Gaussian-Uniform mixture model can remove the influence of outliers from the estimation of the Gaussian distributions. A complex distribution in the presence of outliers is well characterised by a G-UMM allowing outliers to be detected. The GMM fails to identify part of the data distribution.

advantages in its simplicity, computational complexity and convergence properties (Xu & Jordan, 1996). Furthermore the EM algorithm can be simply extended to estimate the parameters of the Gaussian-Uniform mixture model (Coretto & Hennig, 2009).

The EM algorithm is guaranteed to converge to a local maximum but not a global one. The choice of initialisation can have an influence on which local maxima the algorithm converges to. It is therefore important to either make a sensible choice for the initial mixture component assignment — *i.e.* one that is close to some local optimum — or to randomly initialise and perform multiple runs over the algorithm. A sensible choice for the initial mixture components may be available from prior knowledge of the data, otherwise it is common to use a K-means based clustering algorithm such as the well known K-means++ (Arthur & Vassilvitskii, 2007). A number of alternatives and their advantages, including options for random initialisation, are discussed in (Blömer & Bujna, 2016). The optimality of the local maxima on convergence can be assessed by observation of the log likelihood, such that multiple runs over the same data set can be directly compared.

Once the model has been trained by estimating the model parameters, the classification of each data point into the different G-UMM clusters can be considered as the observed state of the system.

2.2. Hidden Markov models (HMMs)

The HMM was first developed for speech recognition (Rabiner, 1989), since then, it has been extended to a wide variety of applications such as fault detection (Smyth, 1994), pattern recognition (Baldi & Brunak, 2001; Bishop, 2006) and health prognosis (Liu, Dong, & Peng, 2012). A HMM is a statistical model for a doubly embedded stochastic process with an underlying stochastic process (Markov chain) being

unobservable (i.e. hidden), and this underlying hidden process can only be observed through another stochastic process which produces a sequence of observations (Rabiner, 1989), here referred to as classes. In the context of fault detection and condition monitoring, the state sequence of the system to be monitored is the assumed hidden process that needs to be estimated for each data point, *i.e.* the health state of the system - healthy or faulty. The classes may be some quantifiable operating condition of the engine, or it could be the output of some classification procedure (such as a mixture model).

The HMM is justified under the following two conditions / assumptions on the system:

i   The Markov property of the hidden states, i.e. the hidden state $\mathbf{X}_i$ of the system at sample i depends only on the hidden state $\mathbf{X}_{i-1}$ of the system at the previous sample $i-1$.

ii The conditional independence of the class given the system hidden state, i.e. the class $\mathbf{Y}_i$ at current sample i depends only on the hidden state $\mathbf{X}_i$ at the current sample i and not on the past system hidden states or observations.

Conceptually, this means that the current system hidden state is linked to the previous hidden state by the conditional distribution $P(\mathbf{X}_i|\mathbf{X}_{i-1})$, and hence to all previous hidden states. The two properties can be expressed formally as follows:

$$P(\mathbf{X}_i|\mathbf{X}_{i-1}, \cdots, \mathbf{X}_1) = P(\mathbf{X}_i|\mathbf{X}_{i-1}) \tag{3}$$

and

$$P(\mathbf{Y}_i|\mathbf{X}_i, \cdots, \mathbf{X}_1, \mathbf{Y}_{i-1}, \cdots, \mathbf{Y}_1) = P(\mathbf{Y}_i|\mathbf{X}_i) \tag{4}$$

The HMM is illustrated by considering the pictorial representation given in Figure 2 within the context of fault detection. The system's hidden states to be estimated are shown by the black circles. There are two hidden states: either 'Nominal' or 'Faulty'. The coloured circles represent the classes.
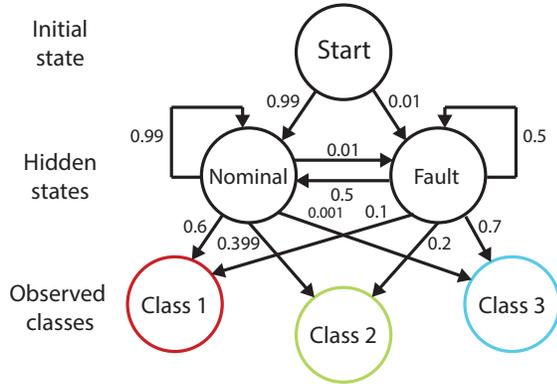
Figure 2. Pictorial representation of the Hidden Markov Model in the context of fault detection

The black arrows between the hidden states represent the probability of a *transition* between hidden states at any given sample, conditional on the hidden state at the previous sample. These probabilities are collected into the transition matrix, A, with elements

$$a_{jp} = P(\mathbf{X}_i = s_p | \mathbf{X}_{i-1} = s_j) \quad j, p = 1, \cdots, n. \quad (5)$$

where n is the number of hidden states, in the current example $n = 2$ with $s_1 = $ 'Nominal' and $s_2 = $ 'Fault'.

The black arrows connecting the Hidden states to the classes represent the *emission* probabilities - the probability that an observation is made, conditional on the hidden state the system is in at the current sample. These probabilities are collected in the emission matrix B with elements

$$\begin{aligned} b_{jk} = P(\mathbf{Y}_i = o_k | \mathbf{X}_i = s_j) \\ j = 1, \cdots, n \quad \text{and} \quad k = 1, \cdots, m \end{aligned} \quad (6)$$

where m is the number of the distinct observation symbols, in the current example, $m = 3$ with $o_1 = $ 'Class 1', $o_2 = $ 'Class 2' and $o_3 = $ 'Class 3'.

Finally, the probability of starting in any given hidden state is represented by the black arrows connecting the initial state labelled 'start' and the hidden states. These probabilities are given by the vector S_0

For the example illustrated in Figure 2 the transition and emission matrices and the initial state vector are given by

$$A = \begin{bmatrix} 0.99 & 0.01 \\ 0.5 & 0.5 \end{bmatrix}, \quad B = \begin{bmatrix} 0.6 & 0.399 & 0.001 \\ 0.1 & 0.2 & 0.7 \end{bmatrix}, \quad (7)$$

$$S_0 = \begin{bmatrix} 0.99 \\ 0.01 \end{bmatrix} \quad (8)$$

respectively.

A HMM is defined by these three matrices (A, B, S_0) which can be estimated from a training data set using, for example, the EM algorithm among others. However, if there is no la-

belled training data available, an alternative method for the choice of these matrices shall have to be used. This choice of matrices is problem specific and is discussed below.

Given the required matrices and a sequence of classes the hidden states can be estimated. Two common methods for performing the state estimation are the forward backward algorithm (Rabiner, 1989) and the max-sum algorithm, also named the Viterbi algorithm in the context of HMMs (Viterbi, 1967).

2.3. G-UMM - HMM based fault detection

The G-UMM and the HMM are combined to form an effective method for the detection of transient faults. Although the method could be applied in the case where labelled training data is available, here it assumes that this is not the case and all learning must be self-supervised (i.e. the HMM matrices are set based on labels generated by the G-UMM). Furthermore, we assume that the process data originates from a variety of different assets whose data characteristics may vary. The variation may be due to different degradation levels, environmental variables or operating conditions. Such a scenario is a common one in the real world: for example, a fleet of aircraft may be built to the same specifications but will operate with varied temperatures, levels of maintenance and component age.

Firstly, the number of Gaussian components, K, is chosen, and the G-UMM is initialised and trained on an input data set (that may or may not contain any transient faults) in the space of the raw signals or in an appropriate feature space. K should be chosen sufficiently large, such that the G-UMM can accurately describe the distribution of the data, while remaining small enough to avoid over-fitting (Kim & Seo, 2014; McKenzie & Alder, 1994) and produce physically interpretable clusters where possible to aid the setting of the parameters of the HMM in the following step. The HMM is used in order to include the temporal relationship into the fault detection scheme. The G-UMM responsibilities, r_{ik}, are the observed state of the HMM. In order to estimate the HMM hidden states, the emission, transition and initial states must be determined. The transient nature of the faults to be detected implies that they are infrequent and occur over small time scales. Even in a data set containing faults, the assumption has been made that a large amount of healthy training data exists such that there is significantly more healthy data than fault data. Based on this, is is assumed that the number of faulty data points is negligible in comparison to the number of healthy data points. The emission matrix values relating to the healthy state are then given by the proportion of data points in each of the classes, given by the G-UMM mixing coefficients π_k.

The elements of the transition and emission matrices, A and B respectively, act as tuning parameters for the algorithm.

Increasing the probability of the transition between states affects the flexibility of the algorithm to assign the fault state to a data point and vice-versa. The sensitivity of the algorithm to these parameters is limited by incorporating the probabilistic state assignment of the G-UMM into the HMM such that, if the G-UMM determines that a data point is highly likely to be in a certain state, then this large probability outweighs those in the transmission and emission matrices. To achieve this, a probabilistic class observation is considered. The class observation is used as past of the propagation through the forwards-backwards algorithm. The probability that a given class is observed at sample i is

$$P(o_i = k) = P(Y_{i,k} = 1|\mathbf{x}) = r_{i,k}, \qquad (9)$$

exactly the G-UMM responsibility.

The probability of the system being in the j'th hidden state is then given by

$$
\begin{aligned}
P(O_i = j) &= \sum_k P(o_i = k)P(Y_{i,k}|X_{i,j}) \quad (10)\\
&= \sum_k r_{i,k}b_{j,k} \quad (11)
\end{aligned}
$$

The i'th hidden state probabilities are represented in matrix form by a diagonal observation matrix, O_i with diagonal elements $P(O_i = j)$, given by

$$O_i(r_i) = diag(B\mathbf{r}_i^T), \qquad (12)$$

where $\mathbf{r}_i = [r_{i,1}, r_{i,2}, ..., r_{i,K+1}]$. The forward-backward algorithm then proceeds as normal, as described in (Rabiner, 1989).

The elements of the emission matrix, B, for the faulty state as well as those for the transition matrix, A, are chosen based on the application. Typically, the probability of observing an outlier when in the faulty state will be much higher, guiding the choice of values. The initial state vector is simply chosen so as to be almost certain that the state sequence will start in the healthy state.
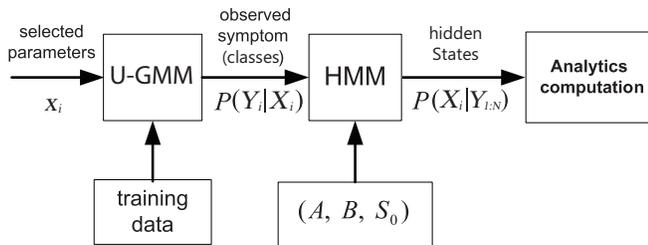

Figure 3. G-UMM -HMM based fault detection

2.4. Summary of the transient fault detection scheme

The proposed fault detection procedure is shown pictorially in Figure 3 and is summarised as follow:

1. Select appropriate system signals that well define the operating behaviour.

2. Initialise the G-UMM by assignment of each data point to an initial cluster.

3. Train the G-UMM in order to characterise the operating region in the space of the selected signals and estimate the G-UMM responsibilities

4. Assign HMM transition, emission and initial sate matrices

5. Apply the HMM algorithm to the set of classes found by the G-UMM classification in step 3 to estimate healthy and faulty hidden states

3. CASE STUDY: VSVA TRANSIENT FAULT DETECTION

The system of interest for this study is the variable stator vane actuator (VSVA) used in aero GTEs. Transient faults have been observed in test bed data for the VSVA system in the presence of a build up of friction, providing an example of the type of fault of interest in this work. The VSVA shares many behavioural characteristics, and hence fault modes, with other controlled actuation systems components giving a wider applicability to the current study.

3.1. VSVA operation

The VSVA controls the position of the variable stator vanes (VSVs) in a GTE in order to manage the efficiency and stability of the compressor system. Figure 4 shows a block diagram of a VSVA for VSV position control. The operation of the VSVA can be summarised as follows:

1. The demanded position of the VSV is set by the electronic engine controller (EEC) control architecture according to estimates of air mass flow.

2. The VSVA position is measured by a linear variable displacement transformer (LVDT) inside the VSVA and continuously transmitted to the EEC.

3. The error between the demanded and measured position is calculated and fed into a PID controller that sets the demanded torque motor current (TMC).

4. The VSV torque motor (TM) controls the servo valve allowing pressurised fuel to be fed to either side of the actuator ram, moving the position of the VSV.

5. When the VSV reaches the correct angle the TM is instructed, via the EEC, to set the spool valve into the null position by producing a holding current, preventing further movement of the actuator.

The actuator can be considered to have three distinct operating modes, extending (moving in the forward direction),
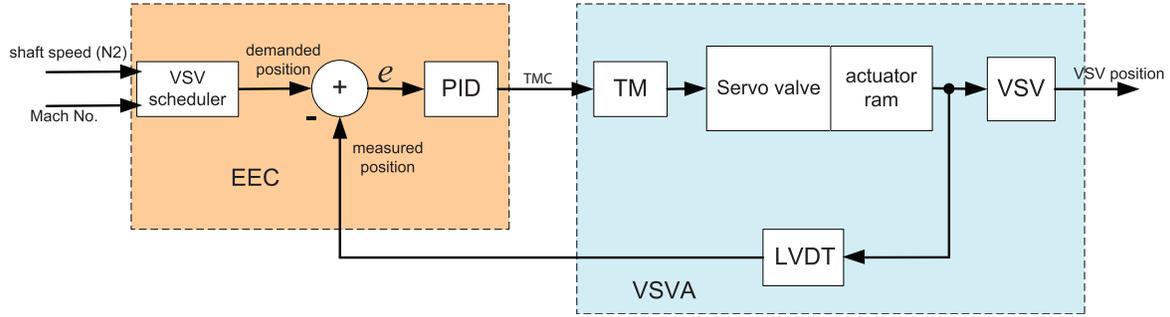
Figure 4. Block diagram of a VSVA

retracting (backwards direction) and approximately stationary. Note that a current is still required when the actuator is stationary and to reject disturbances. The actuator provides a larger force when it is extending than when it is retracting such that the actuator moves at different velocities when moving in different directions. The velocity of the VSVA is also dependant on the high pressure (HP) shaft speed.

3.2. Problem statement

Faults can occur in the components of the VSVA shown in Figure 4. One of the potential fault modes that has been identified by an industrial partner relates to a build up of mechanical friction in the system causing intermittent sticking. A transient deviation from normal behaviour is observed when the VSV becomes temporarily 'stuck' and unresponsive to its driving control signal. Such faults occur in the event of an actuator jam, due to mechanical friction, or the VSV control valve sticking, due to the build up of sediment in the valve. In both cases, the error between the demanded and measured VSVA position increases causing a large tracking error. This results in the feedback control architecture increasing the TMC with no resultant change in the measured VSVA position signal. The fault mode is of particular interest because it is common to many components of a GTE that involve position tracking. Low levels of sticking are expected and are observed in the healthy system. The fault detection task must quantify the magnitude of a transient fault in order to differentiate a developing fault from the baseline.

An example of such a transient fault observed in test-bed data is shown in Figure 5. At the onset of the sticking event, the demanded and measured position signals diverge, which results in the TMC decreasing to its minimum value, as shown in Figure 5.

The G-UMM HMM method developed in this work is employed in order to localise transient faults to achieve a high detection resolution, such that low magnitude/developing sticking events can be identified. Once the faults have been localised in the time series data then analytics can be generated. The interpretability of these analytics is essential such that they can be used as an aid in diagnosis.

3.3. Choice of engine signals

It is important to select signals that maximise the amount of information relating to the fault of interest, while also keeping the dimensionality as small as possible. Two of the signals that are highly sensitive to the fault are the tracking error (difference in the demanded and measured position of the actuator) and the VSV velocity (estimated from the measured VSV position). The final signal that is chosen is the VSV Torque Motor Current which drives the actuator movement.

3.4. G-UMM initialisation

The G-UMM is initialised by selecting the number of clusters, K, and the initial assignment of data points to each cluster. Here, the choice of initialisation is made using domain knowledge. It is clear from a preliminary study of the data that there are approximately three different underlying distributions in the signal space chosen in Section 3.3, corresponding to the three operating modes discussed in section 3.1. The vast majority of the data is in the stationary mode; centred around zero velocity, zero tracking error and the holding current (approx. 16mA). The rest of the data falls into the extending and retracting modes depending on whether the velocity is positive or negative. For this reason, the G-UMM algorithm is initialised by assigning each data point into $K = 3$ clusters defined by the following thresholds on the VSV velocity: VSV velocity < -0.01, $-0.01 \leq$ VSV velocity ≤ 0.01 and VSV velocity > 0.01, see Figure 6: Middle Panels. The velocity thresholds are simply chosen by observation of the VSVA velocity signal, such that positive, negative and near zero velocities are separated into different clusters. It should be noted that it is not necessary to fine tune this value, as it has been found that a range of values within this region will converge to the same local maxima. A random selection of 100 data points are assigned to the outlier distribution to ensure algorithmic stability.

The log probability level of the uniform distribution is set to $log(P_{\mathcal{U}}) = -20$, the value is chosen by trial and error for
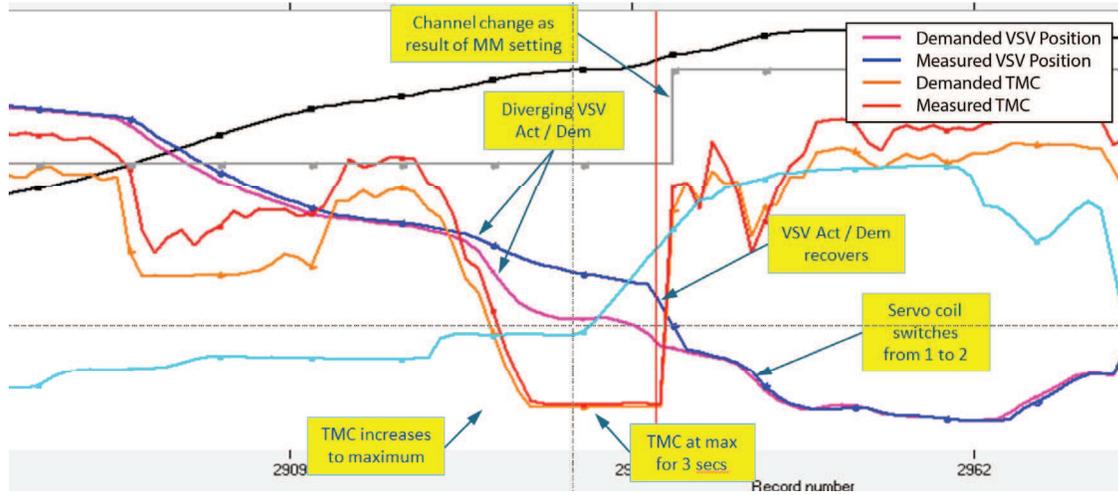
Figure 5. True 'Sticking' event - TMC (Red line) reaches its minimum value for over 3 seconds due to the large error in the demanded (Pink line) and measured (Blue line) position signals.

one data set and then validated by observation on additional data sets. It was found that values in this region (± 10) led to almost identical results. The weighting of the data points performed by the algorithm is observed to have a large effect on the distribution assignment, such that the algorithm is not sensitive to small changes in the uniform probability distribution parameters.

3.5. Choice of HMM matrices

The matrices that define the HMM are chosen so that they reflect the fault detection scenario as discussed in Section 2.3. The matrices were chosen as follows

Transition matrix:

$$A = \left[\begin{array}{cc} 0.99 & 0.01 \\ 0.5 & 0.5 \end{array} \right] \tag{13}$$

Emission matrix:

$$B = \left[\begin{array}{cccc} \pi_1 & \pi_2 & \pi_3 & \pi_4 \\ 0.00001 & 0.05 & 0.05 & 0.89999 \end{array} \right] \tag{14}$$

Initial state matrix:

$$S_0 = \left[\begin{array}{c} 0.9999 \\ 0.0001 \end{array} \right] \tag{15}$$

The choices are justified as follows: The transition matrix reflects a system in the healthy state is likely to remain in that state, by choosing $A_{11} \gg A_{12}$, and has equal probability of remaining or leaving the unhealthy state with $A_{21} = A_{22}$. The elements of the emission matrix in the fault state are cho-

sen from knowledge of the system. The probability of a data point in the fault state being in the outlier cluster ($B_{2,4}$) is large because transient faults are observed as outliers to the nominal distribution of the data. There is a smaller chance that the fault will be found in the clusters at negative/positive speeds ($B_{2,2}$ and $B_{2,3}$) and very little chance it will be in the stationary cluster ($B_{2,1}$, because these data points have a low TMC and are not being commanded to move). It is assumed that the system starts in the healthy state with large probability ($S_{0,(1,1)} \gg S_{0,(1,2)}$).

4. RESULTS OF VSVA TRANSIENT FAULT DETECTION

In this section results of a transient fault detection study are given. Results are first given for a synthetically generated fault and then a healthy data set. A simple analytic is chosen to quantify the severity of the fault. The results of the initial studies provide a threshold value for the analytic and assess the ability to reject false positives. The threshold is then used in order to detect a sticking event in real data. Training of the G-UMM is performed in the space of the raw signals.

4.1. Synthetic fault study

In this section, results and discussion are given for the developed fault detection scheme applied to a real data set into which synthetic faults have been injected.

4.1.1. Synthetic fault generation

Synthetic faults are injected into a time series data set containing 15498 data points for a single real flight, see Figure 7 - Lower panels. A visual comparison to the true fault event observed in Figure 5 shows that a good match is found between the synthetic faults and the true scenario
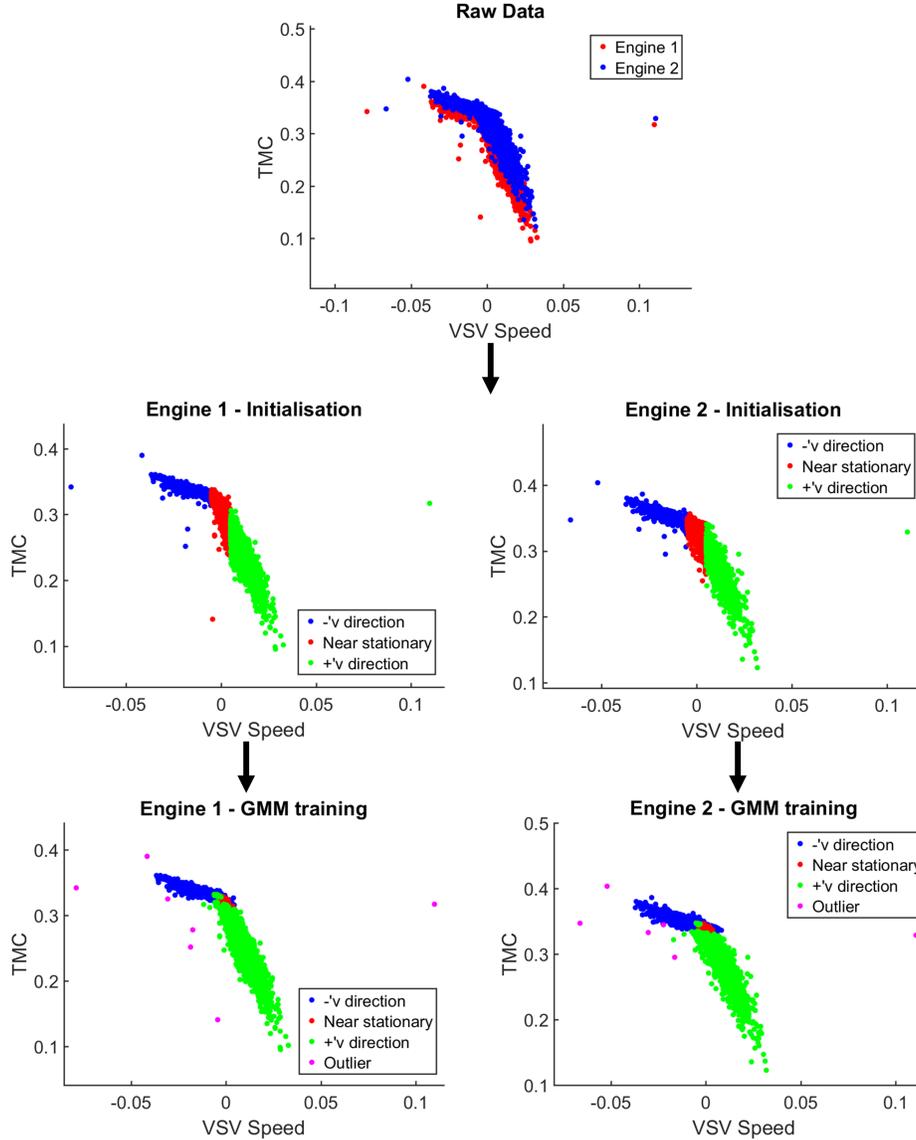
Figure 6. Initialisation and training of the G-UMM - Clustering nominal data

Four faults of increasing duration are injected into the healthy data set with a length of $1, 2, 4$ and 5 seconds ($4, 8, 16$ and 20 data points respectively). Two example of the generated synthetic faults are given in Figure 7. The magnitude of the fault is dependent on the true behaviour of the demanded signal at the fault location. It can be seen from Figure 8 that the faults generate data points that fall outside of the normal operating region of the engine in the space of the signals. The shortest synthetic sticking event ($1s$) causes a divergence from expected behaviour that is of similar magnitude to observations in real, healthy data and is not expected to be differentiable. Note that some of the data points do fall inside this normal operating region. In order to accurately compute the length of the sticking event it is necessary to be able to identify these data points.

4.2. Fault quantification and thresholding

Two analytic are used to describe the potential fault events detected by the G-UMM HMM algorithm. The first analytic is the length of the detected event, defined as the number of consecutive samples for which the system is in the fault state for each potential fault instance. The second is the cumulative sum (CUSUM), over a potential fault location, which quantifies the magnitude of a sticking event. The CUSUM analytic is designed to provide an interpretable measure of the magnitude of the individual sticking event such that it can be monitored over time. The CUSUM is used to remove potential sticking events of low magnitude, that are considered to be part of the normal operating behaviour of the system. A simple thresholding procedure is applied to both analytics,
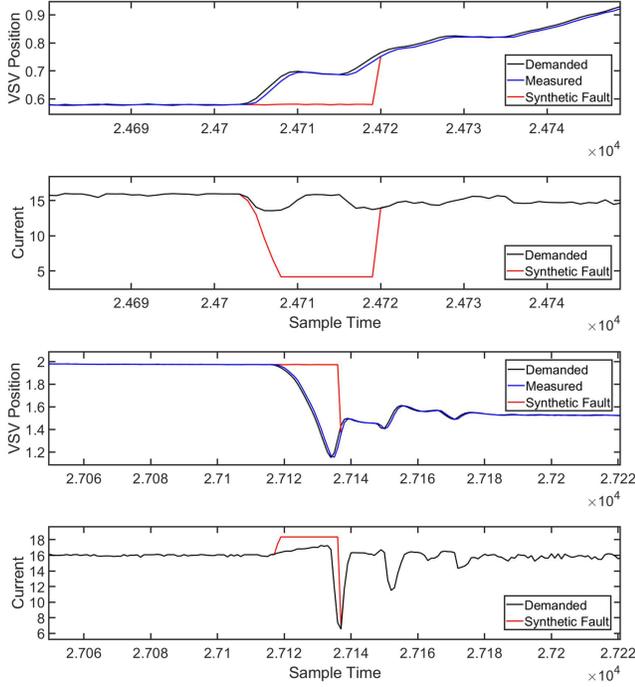
Figure 7. Two examples of a synthetic fault injected into a flight test time series. The measured signal is held constant over the region of 'stick' and the TMC increases/decreases in response to the increased tracking error.

the thresholds are chosen based on observation of the data.

4.2.1. Synthetic fault results

The G-UMM is trained on the real data set with injected synthetic faults using initialisation based on the VSVA velocity, as described in Section 3.4. The elements of the HMM initial state, transition and emission matrices are assigned as in Section 3.5 such that they are given by equations (13)-(15), where the upper row of the emission matrix, B, relating to the emission probabilities in the healthy state is calculated as part of the G-UMM training phase.

The HMM is then applied to the series of classes estimated by the G-UMM and the hidden states are estimated. The result is shown in the space of the signals in Figure 10.

In total, 12 potential fault locations are detected by the algorithm. The analytics calculated over these locations are shown in Table 1. Of the 12 potential events that are detected, four are related to the synthetic faults injected into the system. The analytics relating to the true synthetic faults are highlighted in bold.

4.2.2. Discussion of synthetic fault results

The G-UMM training in the presence of fault events assigns clusters in the same operating regions as for the healthy data

Table 1. Generated analytics for the synthetic fault data set

	Length	CUSUM	Threshold Alert
1	1	0.0282	✗
2	**2**	**0.0966**	✗
3	**7**	**1.1730**	✓
4	**13**	**1.5117**	✓
5	**17**	**7.0798**	✓
6	1	0.0548	✗
7	1	0.0586	✗
8	1	0.0508	✗
9	2	0.1037	✗
10	1	0.0008	✗
11	1	0.0358	✗
12	1	0.0265	✗

set, see Figure 9 in comparison with Figure 6. As before the algorithm assigns the vast majority of the data to a cluster in a small region of the signal space where the actuator is approximately stationary.

This can be seen as the 'elbow' of the data distribution in Figure 9. There are two other branches of the data distribution which can be attributed to the behaviour of the VSV when it is moving in positive and negate directions. These two areas contain significantly fewer data points in comparison. The implication of this is that many clustering/classification algorithms see the 'elbow' of the data distribution as containing all the useful information and will see the remaining data as unimportant for describing the data. For example, the SVM is an alternative and very popular method for the classification/clustering step. Experimentation has shown that due to the described problem, the SVM must assign an extremely large amount of support vectors (essentially all of the data that is not in the 'elbow') causing the SVM to predict the outliers extremely poorly. In comparison, the G-UMM is able to put a distribution over the main mass of the data set and further distributions over the positive and negative VSV velocity areas. This allows the whole data set to be characterised well. The uniform distribution then facilitates the detection of outliers. A further advantage is the simplicity (both mathematically and computationally) of the algorithm.

A threshold on the length analytic is chosen as $\mathcal{T}_L = 1$ such that potential faults of length 1 are not considered. The choice helps to reject any outliers that have been misclassified by the G-UMM HMM algorithm and which might have a large CUSUM value despite having a short length. A threshold value for the CUSUM analytic is chosen based on the CUSUM values given in Table 1. The threshold, $\mathcal{T}_{CS}$, is chosen by observation as $\mathcal{T}_{CS} = 0.5$, such that any CUSUM value observed above this is considered to be a true fault event caused by sticking in the actuator. As may be expected, more than the four synthetic sticking events are detected. Of the four synthetic faults, all but the shortest ($1s$ stick) have a CUSUM of magnitude greater than the threshold of $\mathcal{T}_{CS} = 0.5$, simply differentiating them from those that are
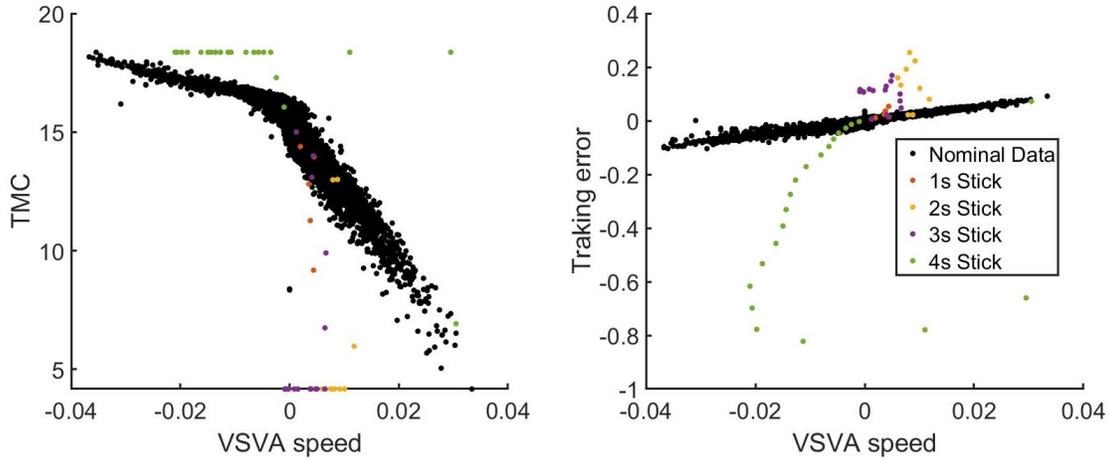
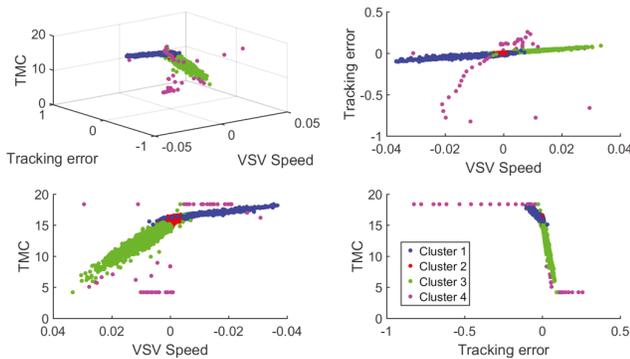Figure 8. Visualising synthetic faults in the space of the parameters

Figure 9. G-UMM Classification of outliers in faulty data. The data distribution has been classified into three areas; Positive velocity (Green dots), negative velocity (Blue dots) and near zero velocity (Red dots). Outliers have been assigned to the uniform distribution (Pink dots).
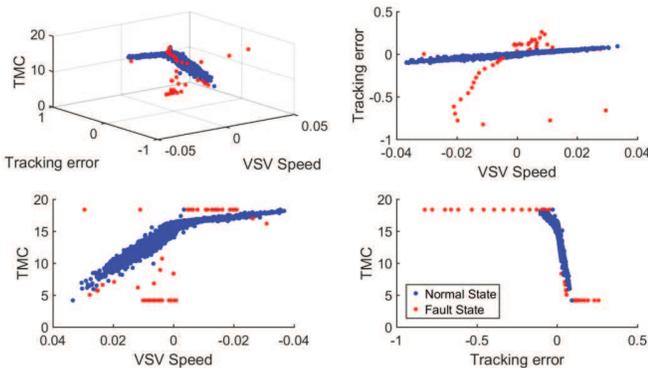

Figure 10. Fault detection using HMMs: Detected faults in the space of the input signals.

part of normal operation. The origin of these potential false positives is the large probability that the producing data point is in the outlier cluster. If the outlier has negligible proba-

bility of belonging to the normal distribution of the data, the HMM may misclassify it as faulty. The length of the sticking events is significantly longer for the synthetic faults than for the falsely identified data points. Three of the four synthetic sticking events can be detected wi th no false positives, as indicated in Table 1. As expected, for the $1s$ synthetic sticking event, the tracking error displayed over the duration of the fault, and hence the CUSUM, is less than the threshold $\mathcal{T}_{CS}$ as well as at another detected event in the data that is part of normal operation. This event is hence not differentiable from behaviour considered as healthy. Investigating the competing section of data, it is found that a significant tracking error and large current is found at this point, indicating that the control architecture is struggling to match the demanded position.

The length analytics does not match the length of the generated faults. This is because at the start of the faults the tracking error is indistinguishable from the noise in the healthy signal and so the start point cannot be detected accurately. Overall the method is found to work well for this synthetic case study. The method detects a suitable level of potential fault events from which a baseline level of health can be established. Within these all of the synthetic faults under test are located, although as expected the lowest level is not distinguishable from the base line.

4.3. Healthy data set study

The method is now employed over a data set containing 14 real test flights that are consecutive in time. The average number of training data points across all data sets is 36663. The data set is not expected to show any sticking behaviour and so is considered as normal for the purpose of this study.

4.4. Healthy data set study results

The G-UMM is trained individually for each flight, using only the data from that flight data set, parameters are initialised as

before. The HMM is then applied with matrices set as before.

The results show a reasonably uniform level of magnitude in each of the analytics as can be seen in Figure 11. The number of detected events fluctuates between 2 and 6 with flight 14 detecting a largest amount (6 events). The average length of the sticking events is low, with no events with length greater than 4. Recalling that the sampling frequency is 4Hz for this data set, there are therefore no detected events lasting longer than one second. The CUSUM analytic is also low for all flights, with the maximum being less than 0.4. This gives us confidence in the choice of threshold chosen in Section 4.2 since, as expected, all of the detected event are considered as part of normal system behaviour based on the threshold.
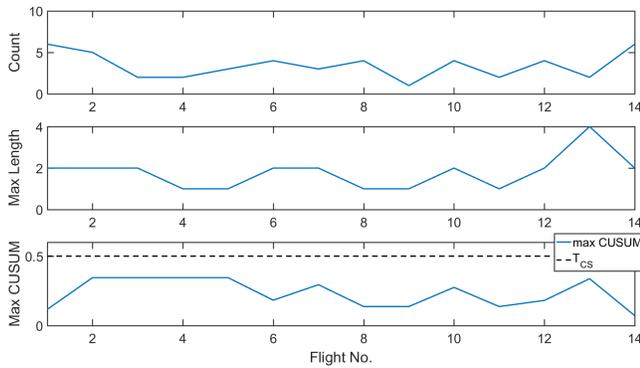

Figure 11. Analytics for the fault detection applied scheme applied to a set of 14 healthy flights.

4.4.1. Discussion of healthy data set results

In comparison to the synthetic faults in the previous example the baseline level detected events for the healthy flights is significantly less than all but the shortest (1s) synthetic fault. This result gives us confidence in the ability to detect true sticking events using a simple threshold on the CUSUM. Note that the $1s$ synthetic fault in the previous example is comparable to tracking delays found during normal operation.

4.5. Real sticking event results

The method is applied to the task of detecting a real sticking event, pictured in Figure 5. G-UMM and HMM parameters/matrices are initialised as before. The threshold on the CUSUM is set as determined in Section 4.2.2, namely $\mathcal{T}_{CS} = 0.5$. The generated analytics are given in Table 2. The true sticking event has been detected by the algorithm, see Figure 12, and has been assigned a large CUSUM value of 4.7161, see Table 2. All other potential events are assigned a CUSUM value much less than the threshold and are hence rejected.
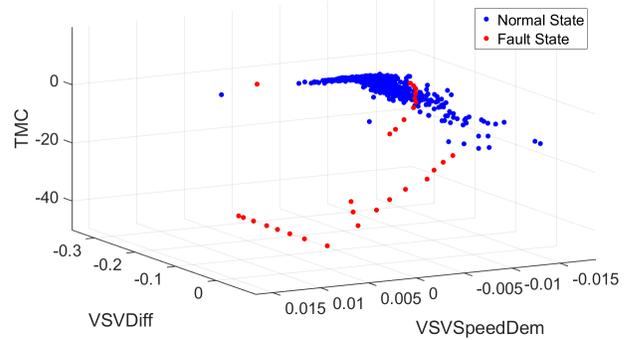

Figure 12. Detected faults for the reall sticking event.

Table 2. Generated analytics for the real sticking event.

	Length	CUSUM	Threshold Alert
1	1	0.0232	✗
2	1	0.0212	✗
3	1	0.0244	✗
4	5	0.0907	✗
5	1	0.0111	✗
6	1	0.0233	✗
7	1	0.0202	✗
8	1	0.0197	✗
9	1	0.0019	✗
10	30	4.7161	✓

4.5.1. Discussion of real sticking event results

The algorithm is successful in detecting the real sticking event in the data. The assigned length and CUSUM analyics are significantly larger than the respective thresholds. The true sticking event. as well as the potential events that are detected by the algorithm are of comparable magnitude to those calculated in the synthetic data set study, see Table 1. The result gives further confidence in the algorithms ability to detect transient sticking.

5. CONCLUSION

In this work a new fault detection scheme has been developed for the detection of transient faults in time series data. The scheme characterises the data using a G-UMM which in turn provides the observed classes a HMM that has the ability to predict the hidden state of the system, either healthy or faulty. The G-UMM HMM fault detection scheme is shown to be well suited to the task of condition monitoring in the case of the detection of transient faults by application to a real world case study. The case study considers the detection of transient faults in the VSV of a GTE. Results are given for synthetic faults, normal healthy data and a real fault event.

The results of the synthetic data study show that the developed method performs well at detecting transient faults of the type investigated in this study. The data is relatively low frequency (4Hz), limiting the detectability of faults over a very

short time scale. Given a higher sampling frequency it is expected that much shorter time scale transients will be able to be detected. The CUSUM of the synthetic faults given in Tables 1 and 2 indicate that the detected transient faults are easily differentiable from the base level of detection. This can be seen by comparison of both the potential false positives, also shown in Tables 1 and 2, and the study performed over multiple real data sets, shown in Figure 11. For the case of a real fault, the fault event is detected by the proposed scheme.

The developed method relies on the assumption that the data is representative of the healthy behaviour of the system. However, it is possible that anomalous data is observed at normal operation (due to, for example, unusual changes in operating condition or large unmeasured disturbances), although such examples are not observed in the given case study. In this case post processing, via pattern recognition algorithms or expert knowledge, is required to determine the cause of the anomaly. The diagnosis should be fed back in to the process in order to inform future decisions, and indicates a future direction of this work.

The developed condition monitoring approach therefore achieves a step change in capability for the presented VSVA case study. The new approach provides capability in both early detection of developing faults as well as in the ability to generate interpretable analytics to quantify the health of the system. The approach can be readily applied to a wider range of applications in which normal operating behaviour can be well defined by a G-UMM.

ACKNOWLEDGMENT

REFERENCES

Arthur, D., & Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual acm-siam symposium on discrete algorithms* (pp. 1027–1035).

Baldi, P., & Brunak, S. (2001). *Bioinformatics: the machine learning approach* (Vol. 3) (No. 3). MIT Press.

Barnett, V., & Lewis, T. (1974). *Outliers in statistical data*. Wiley.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Blömer, J., & Bujna, K. (2016). Adaptive seeding for gaussian mixture models. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 296–308).

Chandola, V., Banerjee, A., & Kumar, V. (2009, jul). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, *41*(September), 1–58.

Chen, J., & Patton, R. J. (1999). *Robust Model-Based Fault Diagnosis for Dynamic Systems* (Vol. 3). Boston, MA: Springer US.

Coretto, P., & Hennig, C. (2009). Identifiability of Gaussian / uniform mixtures. , 1–24.

Dempster, A., Laird, N., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B Methodological*, *39*(1), 1–38.

Ding, S. X. (2013). *Model-Based Fault Diagnosis Techniques* (Vol. 1). London: Springer London.

Fan, C., Xiao, F., Zhao, Y., & Wang, J. (2018). Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data. *Applied Energy*, *211*, 1123 - 1135.

Ge, Z., Song, Z., & Gao, F. (2013, mar). Review of Recent Research on Data-Based Process Monitoring. *Industrial & Engineering Chemistry Research*, *52*(10), 3543–3562.

Görnitz, N., Braun, M., & Kloft, M. (2015). Hidden markov anomaly detection. In *International conference on machine learning* (pp. 1833–1842).

Görnitz, N., Lima, L. A., Müller, K.-R., Kloft, M., & Nakajima, S. (2017). Support vector data descriptions and k-means clustering: One class? *IEEE transactions on neural networks and learning systems*.

Isermann, R. (2011). *Fault-Diagnosis Applications* (Vol. 5) (No. 5). Berlin, Heidelberg: Springer Berlin Heidelberg.

Jena, D., & Panigrahi, S. (2014). Motor bike piston-bore fault identification from engine noise signature analysis. *Applied Acoustics*, *76*, 35–47.

Kim, D., & Seo, B. (2014). Assessment of the number of components in gaussian mixture models in the presence of multiple local maximizers. *Journal of Multivariate Analysis*, *125*, 100 - 120.

Li, J., Pedrycz, W., & Jamal, I. (2017). Multivariate time series anomaly detection: A framework of hidden markov models. *Applied Soft Computing*, *60*, 229–240.

Liao, L., Jin, W., & Pavel, R. (2016, nov). Enhanced Restricted Boltzmann Machine with Prognosability Regularization for Prognostics and Health Assessment. *IEEE Transactions on Industrial Electronics*, *63*(11), 7076–7083.

Liu, Q., Dong, M., & Peng, Y. (2012, oct). A novel method for online health prognosis of equipment based on hidden semi-Markov model using sequential Monte Carlo methods. *Mechanical Systems and Signal Processing*, *32*, 331–348.

McKenzie, P., & Alder, M. (1994). Selecting the optimal number of components for a Gaussian mixture model. In *Proceedings of 1994 ieee international symposium on information theory* (p. 393). IEEE.

McLachlan, G., & Peel, D. (2004). *Finite mixture models*. John Wiley & Sons.

Mt, A. S. (2005). Robust Statistics. (1986), 1–11.

Nasios, N., & Bors, A. G. (2006). Variational Learning for Gaussian Mixture Models. , *36*(4).

Peel, D., & McLachlan, G. J. (2000). Robust mixture modeling using the t distribution. *Journal Statistics and Computing*, *10*, 339–348.

Pimentel, M. A. F., Clifton, D. A., Clifton, L., & Tarassenko, L. (2014). *A review of novelty detection* (Vol. 99).

Ping Li, & Kadirkamanathan, V. (2001). Particle filtering based likelihood ratio approach to fault diagnosis in nonlinear stochastic systems. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, *31*(3), 337–343.

Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, *77*(2), 257–286.

Redner, R. A., Walker, H. F., Mathematics, A., & Review, S. (1984). Mixture Densities, Maximum Likelihood and the Em Algorithm. *SIAM Review*, *26*(2), 195–239.

Smyth, P. (1994, jan). Hidden Markov models for fault detection in dynamic systems. *Pattern Recognition*, *27*(1), 149–164.

Suh, S., Chae, D. H., Kang, H.-G., & Choi, S. (2016). Echo-state conditional variational autoencoder for anomaly detection. In *Neural networks (ijcnn), 2016 international joint conference on* (pp. 1015–1022).

Vincent, P., & Bengio, Y. (2003). Manifold parzen windows. In *Advances in neural information processing systems* (pp. 849–856).

Viterbi, A. J. (1967, apr). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, *13*(2), 260–269.

Xu, L., & Jordan, M. I. (1996, jan). On Convergence Properties of the EM Algorithm for Gaussian Mixtures. *Neural Computation - NECO*, *8*(1), 129–151.

Yan, W., & Yu, L. (2015). On Accurate and Reliable Anomaly Detection for Gas Turbine Combustors : A Deep Learning Approach. In *Phm conference* (pp. 1–8).

BIOGRAPHIES

William R. Jacobs received the M.Phys. degree in physics with mathematics from the University of Sheffield, Sheffield, UK, in 2010 and the PhD degree in nonlinear system identification from the University of Sheffield in 2016. He is currently working as a Research Associate in the Rolls Royce University Technology Centre, Department of Automatic Control and Systems Engineering at the University of Sheffield. His research interests include nonlinear

system identification, fault detection and health management.

Huw L. Edwards received his Masters in Electrical and electronic engineering from the university of Bristol, UK in 2006. He has worked at Rolls-Royce for 10 years in a wide range of fields, initially focused in electrical engineering blue sky concepts including electrically powered aircraft and electric engine accessories. His recent focus has been in measurement systems and predictive health monitoring analytics for gas turbine engines. He is a Member and chartered engineer with the IET.

Ping Li earned his PhD in control engineering from The University of Sheffield, UK in 2002 and is currently a Research Associate with the Department of Automatic Control and Systems Engineering in the same university. He also earned a BEng degree in control engineering and an MSc degrees in inertial technology and navigation both from Northwestern Polytechnical University (NWPU), Xian, China. He held Postdoctoral Research Associate positions in Loughborough University and University of Leicester, UK respectively before he took his current position. He has worked extensively in the areas of model-based fault diagnosis and modelling and system identification. His research interests include fault detection and isolation, optimal filtering, estimation and system identification, and their applications to condition monitoring and control.

Visakan Kadirkamanathan received the B.A. and Ph.D. degrees in Electrical and Information Engineering from the University of Cambridge, UK. He held Research Associate positions at the University of Surrey and the University of Cambridge before joining the Department of Automatic Control and Systems Engineering, The University of Sheffield, as a Lecturer in 1993. He is currently the Head of the Department and a Professor of Signal and Information Processing, affiliated to the Centre for Signal Processing and Complex Systems. His research interests include nonlinear signal processing, system identification, intelligent control and fault diagnosis with applications in systems biology, aerospace systems and wireless communication. He has coauthored a book on intelligent control and has published more than 140 papers in refereed journals and proceedings of international conferences. Professor Kadirkamanathan is the Co-Editor of the International Journal of Systems Science and has served as an Associate Editor for the IEEE Transactions on Neural Networks and the IEEE Transactions on Systems, Man, and Cybernetics, Part B. He was also the Conference Chair of the 4th IAPR International Conference on Pattern

Recognition in Bioinformatics (PRIB 2009) held in 2009 at Sheffield, UK.

Andrew R Mills, CEng, PhD, is a Senior Researcher and Research Programme Manager at the University of Sheffield. He has worked in the defence industry on aerospace and automotive applications before his current post within the University Technology Centre supported by Rolls-Royce. Research interests are in a broad range of aspects related to control and health management systems including: signal processing algorithms, instrumentation, advanced processing technologies and model based design.