Static and dynamic novelty detection methods for jet engine health monitoring

By Paul Hayton\textsuperscript{1}, Simukai Utete\textsuperscript{1}, Dennis King\textsuperscript{2}, Steve King\textsuperscript{2}, Paul Anuzis\textsuperscript{2} and Lionel Tarassenko\textsuperscript{1,*}

\textsuperscript{1}Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, UK
\textsuperscript{2}Rolls-Royce Civil Aero-Engines, Derby DE24 8BJ, UK

Novelty detection requires models of normality to be learnt from training data known to be normal. The first model considered in this paper is a static model trained to detect novel events associated with changes in the vibration spectra recorded from a jet engine. We describe how the distribution of energy across the harmonics of a rotating shaft can be learnt by a support vector machine model of normality. The second model is a dynamic model partially learnt from data using an expectation–maximization-based method. This model uses a Kalman filter to fuse performance data in order to characterize normal engine behaviour. Deviations from normal operation are detected using the normalized innovations squared from the Kalman filter.

Keywords: novelty detection; health monitoring; support vector machines; Kalman filtering; expectation–maximization

1. Introduction

The novelty detection paradigm is ideally suited to monitor the health (or condition) of safety-critical systems such as jet engines, for which abnormal events occur very rarely. Jet engines are now routinely monitored using vibration and performance sensors, both during their pass-off tests and during flight. In any database of vibration and performance data, however, there will be very few examples of abnormal behaviour. Nevertheless, the engine being monitored might unexpectedly display an abnormal type of behaviour which has not been seen before, and it is essential that such an occurrence should be detected as early as possible. With novelty detection, a model of normality is learnt from training data known to be normal and abnormalities are subsequently identified in test data by testing for novelty against the previously learnt model.

There are many examples of model-based fault detection algorithms in the literature (e.g. Kadirkamanathan \textit{et al.} 2002; Korbicz \& Janczak 2002; Venkatasubramanian \textit{et al.} 2003). A fault is a known example of an anomalous condition for the plant being monitored. A novel event, in contrast, is a \textit{previously unseen} example of an anomalous condition. In this paper, we concentrate on model-based novelty detection using two different types of models.

* Author for correspondence (tarassenko@eng.ox.ac.uk).

One contribution of 15 to a Theme Issue ‘Structural health monitoring’.
The first distinction to be drawn between the different types of models available to us relates to whether they are *physical* or *statistical* models. A thermodynamic model is an example of a physical model, whereby the steady-state aerothermodynamic performance of a jet engine can be simulated by an assembly of thermodynamic models that are solved numerically subject to continuity, momentum and energy constraints. This paper is concerned with statistical models, for which the parameters of the models are *learnt* from large numbers of examples of engine data acquired during normal operation (training data). These statistical models may either be *static* or *dynamic* models.

Static models are sometimes divided into statistical and neural network models (Markou & Singh 2003), but we prefer to view the latter within the framework of statistical pattern recognition; therefore we do not make such a distinction. In §2, we describe the training of a static neural network model to detect novel events arising out of changes in the vibration spectra recorded from a jet engine. Dynamic models explicitly encode temporal information which may be essential to identify certain types of novel events. In §4, we investigate a dynamic model for the same jet engine in which different types of performance parameters are fused in the learnt model of normality.

2. Static model of normality

(a) Introduction

Vibration information has traditionally been the main source of information for identifying abnormal behaviour in a jet engine. Jet engines have a number of rigorous pass-off tests before they can be delivered to the customer. The main test is a vibration test over the full range of operating speeds. Vibration gauges are attached to the casing of the engine (see figure 14 in appendix A for a block diagram of a typical jet engine) and the speed of each shaft is measured using a tachometer. The engine on the test bed is slowly accelerated from idle to full speed and then gradually decelerated back to idle. As the engine accelerates, the rotation frequency of the two (or three) shafts increases and so does the frequency of the vibrations caused by the shafts. A *tracked order* is the amplitude of the vibration signal in a narrow frequency band centred on a harmonic of the rotation frequency of a shaft, measured as a function of engine speed. It tracks the frequency response of the engine to the energy injected by the rotating shaft. Most of the energy in the vibration spectrum is concentrated in the fundamental tracked orders and their main harmonics. These, therefore, constitute the ‘vibration signature’ of the jet engine under test. It is very important to detect departures from the normal or expected values of these tracked orders as they will indicate an abnormal pattern of vibration.

In our previous work (Nairac et al. 1999), we investigated the vibration spectra of a two-shaft jet engine, the Rolls-Royce Pegasus. In the available database, there were vibration spectra recorded from 52 normal engines (the training data) and from 33 engines with one or more unusual vibration feature (the test data). The shape of the tracked orders with respect to speed was encoded as a low-dimensional vector by calculating a weighted average of the vibration amplitude over six different speed ranges (giving an 18-dimensional vector for three tracked orders). With so few engines available, the $K$-means clustering algorithm was
used to construct a very simple model of normality, following component-wise normalization of the 18-dimensional vectors. The distribution of the training data in the transformed space was approximated by a small number ($K=4$) of spherical clusters representing the different types of vibration patterns, according to the ‘smoothness’ of the engine.

The novelty of the vibration signature for a test engine was assessed as the shortest distance to one of the kernel centres in the clustering model of normality (each distance being normalized by the width associated with that kernel). If the test vector was sufficiently far from all cluster centres, then it was clearly in a region of space with very few training patterns; hence, it was deemed to be novel. The novelty threshold was chosen so as to accept all training patterns as normal. When cumulative distributions of novelty scores were plotted both for normal (training) and test engines, there was little overlap found between the two distributions (Nairac et al. 1999). A significant shortcoming of the method, however, is its inability to rank engines according to novelty, since the shortest normalized distance is evaluated with respect to different cluster centres for different engines.

In the section that follows, we describe the construction of a support vector machine (SVM) model of normality to characterize the distribution of energy across the vibration spectra acquired from a three-shaft engine, the Rolls-Royce Trent 500. The SVM paradigm provides a geometrical description of normality in an appropriate space, a direct indication of the patterns on the boundary of normality (the support vectors) and, perhaps most importantly, a ranking of ‘abnormality’ according to the distance to the separating hyperplane in feature space.

(b) Support vector machine for novelty detection

The SVM approach to novelty detection (Hayton et al. 2001) is sometimes described as the one-class SVM problem (Schölkopf et al. 2000b; Rätsch et al. 2002). The one-class SVM problem is easier than density estimation as it deals with estimating regions of high probability for the training data. The normal patterns from which the training set is assembled are known as ‘positive data’ and the abnormal patterns which may occur in the test set represent the ‘negative data’. The model is learnt from the positive data only and this description is then used to assess whether a test pattern is likely to have been generated by the same process as the training set (normal or positive data). Geometrically, this means finding some hyperplane $w \in F$ that separates the normal training data from the origin at some threshold $r$. One estimates a function $f(x) = (w \cdot \Phi(x))$ and decides that a pattern $x$ belongs to the normal class whenever $f(x) \geq r$.

Suppose we are given a set of ‘normal’ data points $X = \{x_1, \ldots, x_t\}$. Our goal is to construct a real-valued function which, given a previously unseen test point $x$, characterizes the ‘$X$-ness’ of the point $x$, i.e. which takes on large values for points similar to those in $X$. The algorithm that we present here will return such a function, along with a threshold value, such that a prespecified fraction of $X$ will lead to function values above threshold. In this sense, we are estimating a region which captures a certain probability mass.

This approach employs two concepts from SVM theory (Vapnik 1995) that are essential for good generalization performance in high-dimensional spaces: (i) maximizing a margin and (ii) non-linearly mapping the data into some feature
space \( F \) endowed with a dot product. The latter need not be the case for the input domain \( \chi \) which may be a general set. The connection between the input domain and the feature space is established by a feature map \( \Phi: \chi \to F \), i.e. a map such that some simple kernel (Boser et al. 1992; Vapnik 1995)

\[
k(x, y) = (\Phi(x) \cdot \Phi(y)),
\]

such as the Gaussian

\[
k(x, y) = e^{-\|x-y\|^2/c},
\]

provides a dot product in the image of \( F \). In practice, we need not necessarily worry about \( \Phi \), as long as a given \( k \) satisfies certain positivity conditions (Vapnik 1995).

As \( F \) is a dot product space, we can use tools of linear algebra and geometry to construct algorithms in \( F \), even if the input domain \( \chi \) is discrete. Below, we derive our results in \( F \), using the following shorthand notation:

\[
x_i = \Phi(x_i),
\]

\[
X = \{x_1, \ldots, x_\ell\}.
\]

Indices \( i \) and \( j \) are understood to range over \( 1, \ldots, \ell \) (in compact notation: \( i, j \in [\ell] \)); similarly, \( n, p \in [\ell] \). Boldface Greek letters denote \( \ell \)-dimensional vectors whose components are labelled using normal face typeset.

Using an algorithm proposed for the estimation of a distribution’s support (Schölkopf et al. 2000b), we seek to separate \( X \) from the origin with a large margin hyperplane committing few training errors. Projections on the normal vector of the hyperplane then characterize the \( X \)-ness of test points, and the area where the decision function takes the value 1 can serve as an approximation of the support of \( X \).

The decision function is found by minimizing a weighted sum of a support vector type regularizer and an empirical error term depending on an overall margin variable \( \rho \) and individual errors \( \xi_i \),

\[
\min_{w \in F, \xi \in \mathbb{R}^\ell, \rho \in \mathbb{R}} \frac{1}{2} \|w\|^2 + \frac{1}{\nu \ell} \sum_i \xi_i - \rho,
\]

subject to \( (w \cdot x_i) \geq \rho - \xi_i, \quad \xi_i \geq 0 \).

The precise meaning of the parameter \( \nu \) governing the trade-off between the regularizer and the training error will become clear later. Since non-zero slack variables \( \xi_i \) are penalized in the objective function, we can expect that if \( w \) and \( \rho \) solve this problem, then the decision function

\[
f(x) = \text{sgn}((w \cdot x) - \rho),
\]

will be positive for many examples \( x_i \) contained in \( X \), while the support vector type regularization term \( \|w\| \) will still be small.

We next compute a dual form of this optimization problem. The details of the calculation, which uses standard techniques of constrained optimization, can be found in Schölkopf et al. (2000a). We introduce a Lagrangian and set the derivatives with respect to \( w \) equal to zero, yielding in particular

\[
w = \sum_i \alpha_i x_i.
\]
All patterns \( \{ \mathbf{x}_i, i \in [l], \alpha_i > 0 \} \) are called support vectors. The expansion (2.8) turns the decision function (2.7) into a form which depends only on the dot products, \( f(\mathbf{x}) = \text{sgn}((\sum_i \alpha_i \mathbf{x}_i \cdot \mathbf{x}) - \rho) \).

By multiplying out the dot products, we obtain a form that can be written as a nonlinear decision function on the input domain \( \chi \) in terms of a kernel (2.1) (cf. (2.3)).

A short calculation yields

\[
f(\mathbf{x}) = \text{sgn} \left( \sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) - \rho \right).
\]

In the argument of the \( \text{sgn} \), only the first two terms depend on \( \mathbf{x} \); therefore, we may absorb the next terms in the constant \( \rho \), which we have not fixed yet. To compute \( \rho \) in the final form of the decision function

\[
f(\mathbf{x}) = \text{sgn} \left( \sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) - \rho \right),
\]

we employ the Karush–Kuhn–Tucker (KKT) conditions of the optimization problem (e.g. Vapnik 1995). They state that for points \( \mathbf{x}_i \) where \( 0 < \alpha_i < 1/(v\ell) \), the inequality constraint (2.6) becomes equalities (note that in general, \( \alpha_i \in [0, 1/(v\ell)] \)), and the argument of the \( \text{sgn} \) in the decision function should equal 0, i.e. the corresponding \( \mathbf{x}_i \) sits exactly on the hyperplane of separation.

The KKT conditions also imply that only those points \( \mathbf{x}_i \) can have a non-zero \( \alpha_i \) for which the first inequality constraint in (2.6) is precisely met; therefore, the support vectors \( \mathbf{X}_i \) with \( \alpha_i > 0 \) will often form but a small subset of \( \mathbf{X} \).

Substituting (2.8) (the derivative of the Lagrangian by \( w \)) and the corresponding conditions for \( \xi \) and \( \rho \) into the Lagrangian, we can eliminate the primal variables to get the dual problem. A short calculation shows that it consists of minimizing the quadratic form

\[
W(\alpha) = \frac{1}{2} \sum_{ij} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j),
\]

subject to the constraints

\[
0 \leq \alpha_i \leq 1/(v\ell), \quad \sum_i \alpha_i = 1.
\]

This convex quadratic program can be solved with standard quadratic programming tools. Alternatively, one can employ the sequential minimal optimization algorithm described in Schölkopf et al. (1999), which was found to approximately scale quadratically with the training set size.

3. Application of support vector machine to jet engine vibration signatures

The SVM algorithm for novelty detection is applied to the distribution of vibration energies across the vibration harmonics for the three shafts of the Rolls-Royce Trent 500. Typically, in a jet engine, the amplitude of vibration is a measure of how well balanced an engine shaft is and does not necessarily give an indication of the health of an engine. However, interactions between components of the engine give rise to harmonics (known as ‘multiple orders’) or sub-harmonics (known as ‘fractional orders’) of the shaft speed which can indicate a developing fault.
The approach taken is to model the distribution of energy associated with each shaft over the harmonics relative to the fundamental vibration amplitude. A vector representing the vibration energy for a shaft is therefore generated,

\[ \tilde{v} = \begin{pmatrix} \text{speed} \\ 0.5s/1s \\ 1.5s/1s \\ 2s/1s \\ 3s/1s \end{pmatrix}, \]

where \textit{speed} is the shaft speed as a percentage of the full shaft speed and \textit{s} represents the shaft frequency so that \(0.5s\) is the amplitude of vibration at half the shaft vibration frequency (a half-order). By dividing the vibration amplitude for each fractional or multiple tracked order by the fundamental amplitude, we are removing the dependency on absolute vibration amplitude.

\[(a)\] Data extraction

A number of Trent 500 engines were monitored for several months during the Engine Development Programme and the vibration data were recorded from each of the three shafts. As vibration spectra are extracted by the system every 0.2 s (King et al. 2002), this represents a considerable amount of data. To generate a set of training vectors of manageable size, only vectors which differ from the previous accepted vector significantly are added to the training set. When a jet engine maintains a constant condition for long periods of time, only one training vector is generated but as the engine changes state (for example, during acceleration), many vectors are generated.

Using the training dataset, the SVM algorithm finds the hyperplane that separates the normal data from the origin in feature space with the largest margin. The number of support vectors gives an indication of how well the algorithm is generalizing (if all data points were support vectors, the algorithm would have memorized the data). A Gaussian kernel was used with a width \(c=40.0\) in equation (2.2). This value was chosen by starting with a small kernel width (so that the algorithm memorizes the data), increasing the width and stopping when similar results are obtained both on the training set and another set of data kept apart for validation. The number of support vectors generated depends both on the similarity criterion and the number of training patterns. With of the order of \(10^3\) training patterns generated from 54 engine runs (four different engines), the number of support vectors varied from a minimum of 7 (for the low-pressure (LP) shaft) to a maximum of 36 (for the intermediate-pressure (IP) shaft).

\[(b)\] Data visualization

To illustrate the effectiveness of the trained SVM model, the data were visualized in two dimensions, using the Neuroscale algorithm (Tipping & Lowe 1998). The mapping of a dataset onto a two-dimensional space for visualization purposes is known in the pattern recognition literature as multi-dimensional scaling. A well-known example of such a dimensionality-reduction mapping is
Sammon’s (1969) mapping, which seeks to find a configuration of image points in the two-dimensional visualization space, such that the distances $d_{ij}$ between image points are as close as possible to the corresponding distances $\delta_{ij}$, in the high-dimensional input space. Since it is not possible to find a configuration for which $d_{ij} = \delta_{ij}$, Sammon’s mapping uses the following sum-of-squared-error criterion for assessing the suitability of a configuration with respect to the others:

$$E_S^2 = \frac{1}{\sum_{i<j} \delta_{ij}} \sum_{i<j} \left( \frac{d_{ij} - \delta_{ij}}{\delta_{ij}} \right)^2.$$ 

The Neuroscale algorithm is an extension of Sammon’s mapping, in which the mapping from the high-dimensional input space to the two-dimensional visualization space is parameterized using a Radial Basis Function neural network (Tipping & Lowe 1998).

The data from two engine tests where events of interest occurred were visualized alongside the training data for two of the engine shaft models. The first event was a simultaneous increase in two of the vibration modes of the IP shaft of one of the development engines.

The Neuroscale visualization plot shows the training data as the densely packed grey circles and the points from the test data sequence as black crosses. As can be seen from figure 1, the latter indicates a significant excursion away from the region of normality in the Neuroscale visualization plot.

Figure 2 shows that running the SVM model of normality over the engine test data for the IP shaft identifies the novel vibration pattern associated with abnormal increases in amplitude (relative to the others) for the fundamental, 1i, and half-order, 0.5i. With the novelty threshold used (previously determined empirically on past data), the period of novelty lasts for 4 s, as denoted by the two vertical lines.
The second use of the SVM model of normality is for an abnormal pattern of vibration for the LP shaft as the engine decelerated by 7%, 2 min after suffering a foreign object damage event. Again, the Neuroscale visualization plot (figure 3) shows that the test sequence data vectors (black crosses) are some distance away from the contractual limit (C).

Figure 2. An example vibration pattern identified as novel (between the two vertical lines) by the SVM model of normality for the IP shaft. The 1i and 0.5i tracked orders show a large increase in vibration level during the interval marked by the two vertical lines.

Figure 3. Neuroscale visualization plot for the five-dimensional vectors of speed and tracked orders for the LP shaft.

The second use of the SVM model of normality is for an abnormal pattern of vibration for the LP shaft as the engine decelerated by 7%, 2 min after suffering a foreign object damage event. Again, the Neuroscale visualization plot (figure 3) shows that the test sequence data vectors (black crosses) are some distance away from the contractual limit (C).
from the (normal) training data shown as small grey circles, although the black crosses are closer to the latter than in the IP shaft example. The short excursion away from normality causes the SVM model to identify novelty for approximately 1 s, during the period indicated by the two vertical lines of figure 4.

4. Dynamic model

(a) Introduction

A state-variable model is a dynamic model (usually linear), for which the interrelation between engine ‘states’, inputs and outputs is described with a set of (linear) differential equations. State-space descriptions provide a mathematically rigorous tool for system modelling and residual generation, which may be used in fault detection (Chen & Patton 1998) or, as in this paper, novelty detection. Residuals (the difference between the observations and predicted observations from the model) need to be processed in order to identify the novel event reliably and robustly. By assembling a set of linear models for a range of power conditions, a piece-wise linear state-space model can be constructed. In this paper, we present results from a linear dynamic model used at a given power condition (greater than 70% of maximum engine power).

The Kalman filter is a recursive linear algorithm for estimating system states (Gelb 1974). This section describes the use of a Kalman filter as a linear dynamic model capable of fusing performance data to detect novel events. A further aspect of this work, in common with §2 on static models, is the emphasis on learning from
data: some of the parameters of the linear dynamic model are learnt, using an expectation–maximization (EM)-based method described in Ghahramani & Hinton (1996), during a prior training phase with normal data only.

(b) Model description

It is assumed that the system to be monitored can be described in its fault-free condition by a linear, discrete-time, dynamic model described by equations (4.1) and (4.2). Note that the variables which are to be related in the model must be ones for which a linear relationship can be extracted.

The state equation of the system is given by
\[ x(k + 1) = A(k)x(k) + w(k), \]  
where \( x(k) \) is the state at time-step \( k \); \( A(k) \) is the process model at time-step \( k \); and \( w(k) \) is a zero-mean Gaussian noise process, with covariance matrix \( Q \).

\( x(k) \) is a hidden state—there is no direct access to it. However, observations \( y(k) \) are made, which are assumed to be described by a measurement equation relating the observations to the state.

The measurement equation is
\[ y(k) = C(k)x(k) + v(k), \]  
where \( y(k) \) is the observation vector; \( C(k) \) is the observation model at time-step \( k \); and \( v(k) \) is a zero-mean Gaussian noise process with covariance matrix \( R \).

The model matrices \( A \), \( C \), \( Q \) and \( R \) are set during a training phase under normal operating conditions (see below). During monitoring, excursions away from the trained model can be identified using the Kalman filter innovations,
\[ v(k) = y(k) - C(k)\hat{x}(k|k-1), \]  
where \( \hat{x}(k|k-1) \) is the estimate of the state at time-step \( k \), given the knowledge at time-step \( k-1 \). The Kalman filter runs through a cycle of prediction and correction. The filter’s estimate of state and the error covariance are updated when a measurement is obtained.

The Kalman filter prediction
\[ \hat{x}(k|k-1) = A(k)\hat{x}(k-1|k-1), \]  
has associated uncertainty, given by the state prediction covariance
\[ P(k|k-1) = A(k)P(k-1|k-1)A^T(k) + Q(k). \]  

The Kalman gain is given by
\[ K(k) = P(k|k-1)C^T(k)S^{-1}(k), \]  
where \( S(k) \) is the innovation covariance.

The innovation covariance \( S(k) \) is
\[ S(k) = C(k)P(k|k-1)C^T(k) + R(k). \]  

The state estimate is
\[ \hat{x}(k|k) = \hat{x}(k|k-1) + K(k)v(k). \]  

The associated state covariance \( P(k) \) is given by
\[ P(k) = (I - K(k)C(k))P(k|k-1). \]
5. Novelty detection using dynamic model

In the sections that follow, we present an approach for learning two of the four model matrices $A$, $C$, $Q$ and $R$ from data. The models are speed-based performance models of the engine, in which the relationships are learnt from training sets of normal data. Changes in test data are then detected by monitoring the normalized innovations squared (NIS) from the Kalman filter, where NIS is defined as follows:

$$\text{NIS}(k) = r^T(k)S^{-1}(k)r(k).$$  (5.1)

The innovations should be zero mean and white, with covariance consistent with that calculated by the filter (Bar-Shalom & Li 1993).

Whenever there is a departure from the normal behaviour captured in the learnt model, there is a rise in NIS. This can be caused by gradual component deterioration or by unexpected events which affect the relationship between the performance parameters, or observations $y$, measured on the engine. To illustrate this, we consider one such event, which occurred in a development Trent 500 engine on a test bed. A problem with the radial driveshaft caused the three shaft speeds to undergo a sudden change from normal behaviour. In the main event region, the speeds diverge for approximately 10 s: the high-pressure (HP) shaft speed increases, the LP shaft speed remains approximately constant and the IP shaft speed decreases. The engine is then shut down.

(a) Expectation–maximization-based learning method

The learning of the model of normality uses the EM algorithm within the framework described by Ghahramani & Hinton (1996). For a sequence of observation vectors $y$ and state vectors $x$, the joint log probability, given a Gaussian initial state density, is shown to be expressible as a sum of quadratic terms. The EM algorithm consists of successive expectation and maximization steps. In the expectation stage, the expected log likelihood is computed. The maximization step involves estimating the system parameters by taking partial derivatives of the expected log likelihood and equating them to zero. This method is further described in Roweis & Ghahramani (1999), which also presents algorithms for determining the parameters. Software implementing these algorithms can be found in a well-known Kalman filtering toolbox (Murphy 1998–2002). In the sections that follow, we show how these algorithms can be used to learn performance-based models for the Trent 500 from normal data. Learning of the model matrices from data does not give a unique state-space representation and so the learning process has to be guided by imposing strong prior constraints.

(b) Training and test data

Training data sequences are taken from 3 days of engine running: from runs several days before the event and from running on the event day, but prior to the event. The test data sequences are taken from the event day—from a later sequence than any of the training data, but prior to the main event, and from the main event itself. The data are subsampled, so that one in every five points is extracted. This corresponds to a data point per second. The subsampled data are then averaged over a five-point sliding window.
In the engine datasets, there are long periods of running at constant speed. Accelerations and decelerations, by contrast, tend to cover a relatively small number of data points. The data points are required to be sequential as their time history is important. The datasets used for training models are balanced so that the different types of operating regime are given as equal a weighting as possible. In this paper, we also concentrate on models of normality for smooth acceleration and deceleration.

The training data are shown in the eight sequences of figure 5. In both training and testing, the shaft speed data used are in the region where the LP speed is between 70 and 90% of maximum LP speed. Within this speed sub-range, the assumption of model linearity is taken to be valid. Figure 6 presents the two sequences of test data; figure 6b covering the main event. Although the three shafts are separate systems, the shaft speeds should change in tandem. This condition is violated as a result of the radial driveshaft problem in the event sequence of figure 6, between t=30 and 50, when (i) the LP shaft speed does not change, (ii) the IP shaft speed decreases, and (iii) the HP shaft speed increases.

Initial tests are carried out with a dynamic model of normality based solely on a three-speed observation vector. In order to make learning possible, we introduce the constraint that the measurement matrix $C$ should be equal to the identity matrix. Hence, equation (4.2) becomes

$$y(k) = x(k) + v(k). \tag{5.2}$$

The observation vector is therefore the state corrupted by noise. The measurement noise covariance matrix $R$ is assumed to be diagonal and set using engine measurement uncertainty data sheet values provided by Rolls-Royce,

$$R = \begin{bmatrix} 0.1538 & 0.0 & 0.0 \\ 0.0 & 0.0989 & 0.0 \\ 0.0 & 0.0 & 0.0902 \end{bmatrix}.$$  

The $C$ and $R$ values remain fixed throughout (no learning). The state transition matrix, $A$, the process noise covariance, $Q$, and the initial state covariance, $P(0)$, are learned using the EM-based algorithm of Ghahramani & Hinton (1996). The state transition matrix is initialized to $I$ and the process noise covariance to $0.1I$. The initial state is taken from an average over five points for one of the training sequences, and the initial state covariance is set to $I$.

The learning curve for the system is shown in figure 7. After 100 EM iterations, the state transition matrix is found to be

$$A = \begin{bmatrix} 0.9661 & 0.06601 & -0.03673 \\ -0.02079 & 1.041 & -0.02265 \\ -0.01780 & 0.03546 & 0.9799 \end{bmatrix}.$$  

Thus, learning does not change the state transition matrix, $A$, significantly as it remains approximately equal to $I$. The process noise covariance matrix becomes

$$Q = \begin{bmatrix} 0.01878 & 0.01166 & 0.01011 \\ 0.01166 & 0.007916 & 0.006619 \\ 0.01011 & 0.006619 & 0.005852 \end{bmatrix}.$$
Figure 5. Training data. The figure shows shaft speed data (acceleration and deceleration manoeuvres) taken from several days prior to the event (a, four plots) and also from the event day (b, four plots), but from sequences occurring well before the event. n1v, LP shaft speed; n2v, IP shaft speed; n3v, HP shaft speed (all expressed as a percentage of the maximum speed for that shaft).
Figure 6. Test data. The figure shows shaft speed data taken from the event day. (a) Test data recorded before the event. (b) Data acquired just before and at the time of the event ($t=30$).

Figure 7. The log likelihood values over 100 EM iterations, for a system with a three-speed observation vector, are shown.
The learned initial state uncertainty is
\[
P(0) = \begin{bmatrix} 20.36 & 14.77 & 11.10 \\ 14.77 & 11.71 & 7.796 \\ 11.10 & 7.796 & 6.190 \end{bmatrix}.
\]

Once the three-speed model has been learned, it is applied to the training sequences of figure 5 to check the range of NIS values obtained during the (normal) changes of shaft speed associated with acceleration and deceleration manoeuvres.

Figure 8 shows the changes in NIS for the training data sequences taken from an engine for several days prior to the event and figure 9 shows the changes in NIS for training data sequences recorded during the morning of the event day. (Note that figures 8 and 9 show the time of day on the horizontal axis.) With each sequence, there is a transient rise in NIS when the acceleration (or deceleration) manoeuvre occurs, but never above a value of 3.0. On the basis of these results, it would be possible to set a threshold of 8.0 for the detection of a novel event (approximately 2.5 times the maximum value of NIS on the training data).

When the test data of figure 10 are run through the same three-speed model, figure 10a gives a maximum NIS of 0.6, but the diverging speeds from \( t=17:30:03 \) in the event sequence cause the NIS value to rise first to 8, then to 10 at \( t=17:30:33 \) and eventually to 15, just before the automatic shutdown occurs. With the novelty threshold set at NIS=8.0, the event would have been detected at \( t=17:30:16 \).

(c) Speeds, pressure and temperature

The model of normality is now extended to five performance parameters by incorporating the high-pressure compressor (HPC) exit pressure, \( p_{30} \) (\( p_{30} \) is one of the \( P_3 \) set of pressure readings; see figure 14 in appendix A), and the turbine gas temperature, \( tgt \), into the observation vector, in addition to the three shaft speeds.

As with the three-speed model, \( C \) is set to \( I \). \( R \) is now a diagonal matrix, with two extra values for the measurement uncertainties for \( p_{30} \) and \( tgt \),
\[
R = \begin{bmatrix} 1.1538 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0989 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0902 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.7300 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 2.000 \end{bmatrix}.
\]

The state transition matrix, \( A \), and the process noise covariance, \( Q \), are again learnt, after being set to initial values of \( I \) and 0.1\( I \), respectively. The initial state is taken from an average over five points for one of the training sequences and the initial state covariance \( P(0) \) is set to \( I \). \( A \), \( Q \) and \( P(0) \) are all learned, as with the three-speed model.
After 100 EM iterations, the learned value of $A$ is

$$A = \begin{bmatrix} 0.7467 & -0.1705 & 0.4583 & 0.02863 & -0.01574 \\ -0.1537 & 0.8918 & 0.2847 & 0.01771 & -0.00982 \\ -0.1232 & -0.08763 & 1.232 & 0.01448 & -0.00819 \\ -2.807 & -1.8870 & 5.110 & 1.320 & -0.1781 \\ -3.558 & -2.6820 & 6.827 & 0.4233 & 0.7631 \end{bmatrix}.$$ 

The learned process noise covariance is

$$Q = \begin{bmatrix} 0.00148 & 0.000414 & 0.000396 & 0.01429 & -0.00542 \\ 0.000411 & 0.000661 & 0.000473 & 0.00846 & -0.00147 \\ 0.000394 & 0.000473 & 0.000600 & 0.00725 & -0.000203 \\ 0.0143 & 0.00846 & 0.00725 & 0.2048 & -0.0624 \\ -0.00542 & -0.00148 & -0.000207 & -0.0624 & 0.0669 \end{bmatrix}.$$
Figure 9. (a–d) The change in NIS for training data sequences taken from the day of the event, but well before the event. This test on training data is included as a check.

Figure 10. The changes in NIS for the two test sequences (three-dimensional observation model), (a) prior to the radial driveshaft event and (b) at the time of the event. Note the two different scales used for the NIS axis.
The learned initial state covariance is

\[ P(0) = \begin{bmatrix} 20.75 & 14.88 & 10.83 & 264.1 & 309.6 \\ 14.88 & 11.77 & 7.56 & 198.3 & 216.4 \\ 10.83 & 7.56 & 5.79 & 136.3 & 165.4 \\ 264.1 & 198.3 & 136.3 & 3436 & 3898 \\ 309.6 & 216.4 & 165.4 & 3898 & 4728 \end{bmatrix}. \]

As before, once the five-dimensional observation model has been learned, it is applied to the training sequences of figure 5 to check the range of NIS values on the training data. Figure 11 shows the three speeds, p30, tgt and NIS for the first four of the eight training sequences (taken from engine runs which took place several days before the event).

Figure 12 shows the three speeds, p30, tgt and NIS values for the four training sequences taken from the event day, but well before the occurrence of the event. These figures reveal that the largest value of NIS on the training data for the five-dimensional observation model is just below 15.0, which occurs for the fourth of the training sequences.

Figure 11. (a–d) The change in NIS for the first four training sequences (several days before the event) using the five-dimensional model (speeds plus tgt and p30).
the eight sequences. NIS values are higher than for the three-speed model as a result of the increased dimensionality of this model. Based on these results, it would be possible to suggest a novelty detection threshold set at 37.0 for the five-dimensional observation model (approximately 2.5 times the maximum value of NIS on the training data).

Figure 13 shows the three speeds, p30, tgt and NIS for the two test sequences. With the first test sequence (figure 13a), NIS only rises to 4.0. When the model is applied to the event sequence which includes the diverging speeds (figure 13b), the NIS value rises rapidly to 60.0 before subsequently decreasing. It is clear that the five-dimensional observation model, which includes more information about the engine than just the three shaft speeds, would have allowed an earlier detection of the novel event (at $t=17:30:01$, if a novelty threshold of 37.0 had been used).

The limits for normal NIS values could be set in a training phase by working out time-averaged values over data windows of several tens of seconds. These statistically significant values could then be used for comparison in the testing phase. However, in testing, point-by-point values of NIS (i.e. every second) would still be used in preference to averages owing to the need to detect sudden changes.

Figure 12. (a–d) The change in NIS for the last four training sequences (same days as the event but well before its occurrence) using the five-dimensional model (speeds plus tgt and p30).
Novelty can be used to detect anomalous behaviour in a jet engine. Section 2 showed how a static model of normality can be trained to learn the normal distribution of energy among main vibration tracked orders. A five-dimensional vector was constructed for the vibration amplitudes of the main harmonics (0.5, 1.5, 2 and 3 times the fundamental) at each speed, normalized with respect to the amplitude of the fundamental. Using a training dataset of normal five-dimensional vectors, the SVM algorithm found the hyperplane that separates the normal data from the origin in feature space with the largest margin. Training the SVM model consists of minimizing a quadratic subject to a set of constraints using standard quadratic programming tools, as previously described in the SVM literature. The number of support vectors was determined empirically using a validation set and the SVM model was then used to highlight two events which gave rise to abnormal vibration patterns. Although the SVM model can be used to rank novelty, with these two examples it was simply used with a ‘novelty threshold’ determined empirically from analysis of past data.

For the dynamic model of §4, based on three or five performance parameters, the method of Ghahramani & Hinton (1996) and Roweis & Ghahramani (1999) was used to learn the state transition matrix, the process noise covariance matrix and the initial state uncertainty. Linear behaviour within a restricted engine speed range was assumed to be a sufficiently accurate approximation to the behaviour of the engine during smooth acceleration and deceleration manoeuvres. In order to make it possible to learn the model parameters from training data, a number of further assumptions were introduced. The main one of these was that the observation vector is simply the state corrupted by noise, i.e. the measurement matrix \( C \) is the identity matrix.

A three-speed model of normality was able to identify as novel an event which gave rise to diverging speeds prior to automatic shutdown of the engine on the test bed. When two further parameters (HPC exit pressure and turbine gas

6. Conclusions
temperature) were added and a new five-dimensional model learnt from the same training sequences, the level of discrimination between the novelty index (NIS) on the anomalous data and the normal training data increased. This is not surprising as more information is being provided, but increased dimensionality can make the learning of the model from data more difficult.

A complete description of the engine’s behaviour for novelty detection purposes can be provided through the use of multiple models. It is then possible to switch between different dynamic models according to speed range and the rate of change of speed, depending on the type of acceleration or deceleration manoeuvre (smooth, rapid or ‘slam’).

In the continuing application of this work on the Rolls-Royce Trent family of jet engines (Bailey et al. 2004), it is planned that novel events will be identified during flight (on the rare occasions on which they occur) and notified to the maintenance engineering crew on the ground once the aircraft has landed.

The authors wish to thank Dr Visakan Kadirkamanathan for his helpful discussions on learning dynamic models from data.

Appendix A

Figure 14. Temperature and pressure notation of a typical turbo-jet engine. Picture courtesy of Rolls-Royce plc (1986). Picture reproduced with the permission of Rolls-Royce plc.
References


