

# **A data mining approach to reveal patterns in aircraft engine and operational data**

Srini Sundaram, Iain G.D.Strachan and David A. Clifton  
Oxford BioSignals Ltd.  
174, Milton Park, Abingdon, Oxfordshire, OX14 4SE  
+44 (0) 1235 433574  
[srini.sundaram@oxford-biosignals.com](mailto:srini.sundaram@oxford-biosignals.com)

Steve King, John Palmer,  
Rolls-Royce plc, EHM Global Capability Group, PO Box 31,  
Derby, DE24 8BJ.

## **Abstract**

Modern day aircraft engines are embedded with sensors acquiring vast amounts of data related to engine performance such as pressure, temperature, efficiency, and speed during takeoff, climb and cruise stages of flight. These performance data are stored in performance reports. In addition, the maintenance actions across the fleet, service history and observations during flight are logged into service reports and pilot logs respectively. A typical fleet-wide performance dataset can be very large. Efficient operation and maintenance of a fleet requires proper use and analysis of the available data, searching for potentially useful trends within large datasets. Such trends may allow a fleet specialist to establish engine behaviour profiles under different conditions, and may provide indication of abnormalities in engine condition, or of hazardous events. Currently, fleet specialists use various data analysis tools to identify and analyse abnormal behaviour. Data is typically obtained from multiple sources, making this a complex challenge. To address this challenge, this paper introduces a data mining tool capable of assisting fleet specialists by searching for useful patterns in large datasets, generating reliable, timely alerts when “abnormal” patterns are identified.

## **1. Introduction**

Fleet management requires decision support tools to provide indication of potentially abnormal events during engine operation. The information is usually maintained in performance or service reports containing data from a fleet of engines. These datasets are typically multivariate and examples of events are rare compared to the quantity of available data.

This paper describes the application of two techniques to the analysis of large fleet-wide datasets:

Visualisation techniques are used to project such high-dimensional data into two dimensions for visual inspection providing knowledge about structure in the high-dimensional dataset, which can inform the process of constructing a model of normal system behaviour<sup>(1)</sup> as required for automated novelty detection.

Prediction techniques can estimate the correlation between a selection of “input” parameters in the dataset, if such a correlation exists, and predict the expected distribution of a selected output parameter as a function of those inputs. The observed data is compared with the predicted distribution, which can indicate abnormal system behaviour if the prediction error is above some threshold.

## 2. Visualisation and Prediction

The following sections describe three techniques that have been investigated for use in analysing an exemplar dataset. These are Sammon’s mapping, NeuroScale, and Gaussian Processes. The first two techniques are visualisation techniques that represent multidimensional data in two dimensions, whereas the latter is a probabilistic model that can provide prediction.

### 2.1 Sparse Approximated Sammon Stress (SASS)

Sammon’s mapping is a non-linear technique that creates a two-dimensional configuration of  $n$ -dimensional ( $n > 2$ ) points by preserving, in the visualisation, the interpoint distances<sup>(2)</sup> in the original high-dimensional data space. In this paper, Sammon maps were computed using a modified form of the Sammon Stress metric that involves random sub-sampling of the complete set of inter-point distance pairs. The standard Sammon’s mapping algorithm minimises an objective function known as the *Sammon Stress*,  $E$ :

$$E = \frac{1}{\sum_i \sum_{j>i} d_{ij}^*} \sum_i \sum_{j>i} \frac{(d_{ij}^* - d_{ij})^2}{d_{ij}^*} \quad (1)$$

where  $d_{ij}^*$  is the Euclidean distance between vectors  $i$  and  $j$  in data space, and  $d_{ij}$  is the Euclidean distance between corresponding vectors in the visualisation space. The objective function is minimised using a gradient descent technique that adjusts the positions of the vectors in visualisation space until  $E$  reaches some minimum, or a maximum number of iterations is reached.

This technique has significant limitations for large datasets in that it requires  $O(N^2)$  storage locations for  $N$  vectors, while the computational complexity also scales quadratically with  $N$ . The original version of Sammon’s mapping suggested a practical limit of around 200 vectors. This has increased to a few thousand, given contemporary computing resources.

The new metric is called **Sparse Approximated Sammon Stress** (SASS), and overcomes this problem by random subsampling of the set of interpoint distances. In practice it has been found that, even with  $N > 10^4$ , randomly sampling the distances from each vector to around 100 others is sufficient to approximate the Sammon stress and provide a visualisation that closely matches that obtained from conventional use of Sammon’s mapping using all vectors. Formally, if we define  $S$  to be the sparse subset

of the index pairs  $(i, j)$ , for which the Euclidean distance is calculated, then we define the objective function to be:

$$E_{SASS} = \frac{1}{\sum_{i,j \in S} d_{ij}^*} \sum_{i,j \in S} \frac{(d_{ij}^* - d_{ij})^2}{d_{ij}^*} \quad (2)$$

The two sets of distances  $d_{ij}, d_{ij}^* : (i, j) \in S$  may be stored as sparse matrices. Given that only 100 distances for each vector need to be stored for each of  $N = 10^4$  data points, this represents an increase in speed of a factor of  $50^1$ , and requires considerably less storage – enabling larger datasets to be visualised. For the results of the analysis described in this paper, considerable further increases in speed were obtained by replacing the gradient descent procedure originally used by Sammon with a standard optimization algorithm (Scaled Conjugate Gradients).

## 2.2 NeuroScale Algorithm

The *NeuroScale* neural network<sup>(2)</sup> allows the visualisation of high-dimensional vectors by mapping them to lower numbers of dimensions (typically two, for visual inspection)<sup>(3)</sup>. Feature vectors are extracted from the data that capture the difference between normal and abnormal operation. For every high-dimensional feature vector, the *NeuroScale* network provides a corresponding pair of  $(i, j)$  co-ordinates. This is a projection from  $D > 2$  dimensions to  $D' = 2$ .

The training algorithm of the *NeuroScale* network attempts to preserve the inter-pattern distances of high-dimensional vectors after projection into 2-dimensional space by minimizing the Sammon Stress metric,  $E$ , as defined in equation (2).

In contrast to Sammon’s mapping, which minimises the objective function by adjusting the locations of the visualisation vectors, the *NeuroScale* algorithm adjusts the output weights of a Radial Basis Function network in order to reduce the value of  $E$ .

Thus,  $n$ -dimensional feature vectors which are similar (i.e., close together in the original high-dimensional data space) should be kept close together after projection into 2-dimensional space. Conversely,  $n$ -dimensional vectors that are significantly different from one another (i.e., far apart in high-dimensional space) should remain well-separated after projection into 2-dimensional space. The objective is to allow clusters of feature vectors corresponding to “normal” behaviour to be evident, with feature vectors corresponding to “abnormal” behaviour to be far removed from them (and thus detectable by some later analysis technique).

SASS, in comparison to *NeuroScale*, provides considerable savings in memory requirements, and there may also be an advantage in the proposed SASS algorithm in

---

<sup>1</sup> The total number of interpoint distances to calculate is given by  $n(n-1)/2$ , giving 49,999,500 distances for the full set, as opposed to  $10,000 \times 100 = 1,000,000$  for the sparse approximation

<sup>2</sup> A Radial Basis Function neural network.

that it will be less sensitive to outliers. This is because the SASS algorithm can place the vector in the visualisation corresponding to such an outlier to be far from the rest of “normal” data, without affecting the relationships between the other points. By contrast, in order to reduce the stress measure when encountering datasets including such outliers with NeuroScale, the algorithm must adjust the output layer weights of an RBF neural network – a change that is likely to affect all the visualisation vectors.

### 2.3 Gaussian Processes

A Gaussian Process (GP) is a stochastic process that can be explained using Bayesian linear regression framework. Consider a linear regression model using set of basis functions  $\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \phi_3(\mathbf{x}) \dots \phi_M(\mathbf{x})$  and training dataset  $\mathbf{x} = x_1, x_2, \dots, x_N$ . The model output is given by the linear combination of basis functions and random variables.

$$Y(x) = \sum_{j=1}^M W_j \phi_j(\mathbf{x}) \quad (3)$$

where  $\mathbf{W}$  represent the set of random variables. To extend the above to Bayesian technique, standard approach is to use a prior distribution based on the knowledge about the data and once the data is observed, the posterior distribution is formed using Bayes theorem. If  $\mathbf{W}$  is assumed to have a Gaussian distribution with zero mean with covariance  $\Sigma$ , the process  $Y(x)$  is also Gaussian with mean and covariance functions given as follows.

$$E_{\mathbf{w}}[Y(\mathbf{x})] = 0 \quad (4)$$

$$E_{\mathbf{w}}[Y(\mathbf{x})Y(\mathbf{x}')] = \Sigma \phi(\mathbf{x}) \phi^T(\mathbf{x}') \quad (5)$$

In the above equation (5), the covariance of  $Y(\mathbf{x})$  and  $Y(\mathbf{x}')$  is defined by a function  $K(\mathbf{x}, \mathbf{x}')$  and it characterises the correlations between different vectors in the process. The usual choice of a function is that it is stationary such that the covariance is only affected by the distance between  $\mathbf{x}$  and  $\mathbf{x}'$  and not their location.

$$K(\mathbf{x}, \mathbf{x}') = K(\mathbf{x} - \mathbf{x}') \quad (6)$$

Gaussian Process can be viewed as Bayesian linear regression over infinite number of basis functions. The eigenfunctions of the covariance function form the set of basis functions. Mercer's Theorem<sup>(5)</sup> states that covariance function can be expressed as weighted sum of eigenfunctions

$$K(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{x}) \phi_i(\mathbf{x}') \quad (7)$$

The above expression is equivalent to the representation of matrix  $K$  using *Singular Value Decomposition* as matrix of eigenvectors and eigenvalues. Thus, if eigenfunctions are chosen as basis functions and the prior over the weights is chosen to be diagonal matrix of eigenvalues, then the GP becomes Bayesian linear regression. In this paper,

two classes of covariance functions are considered for the analysis: the Squared Exponential (SE) and Rational Quadratic (RQ) functions, described below.

### Squared Exponential

The SE covariance function is widely used and is defined to be:

$$K_{SE}(r) = v_0 \exp\left(-\frac{1}{2} \sum_{i=1}^d a_i (x_i - x_i')^2\right) + b \quad (8)$$

where  $b$  denotes the bias that controls the vertical offset of the GP. The SE covariance function is stationary (a function of  $x - x'$  that is invariant to translations) and isotropic (a function of  $x - x'$  that is invariant to translation and rotation). SE covariance function is infinitely mean-square differentiable, the sample functions obtained from the prior with SE function have mean-square derivatives of all orders, thus the Gaussian process is very smooth<sup>(7)</sup>. The hyper parameter  $v_0$  represents a positive pre-factor applied to the exponential term in equation (8) and the term  $a_i$  represents weights on the inputs in the covariance function, allowing different distance measures to be used for each dimension.

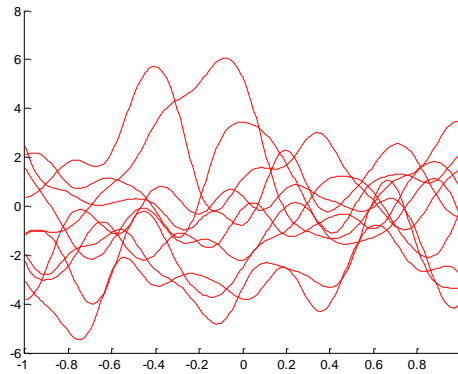


Figure 1 – GP Prior sampling with SE covariance function

### Rational Quadratic

The RQ covariance function is given by

$$K_{RQ}(r) = v_0 \left(1 + \sum_{i=1}^d a_i (x_i - x_i')^2\right)^{-\nu} + b \quad (9)$$

The parameters  $v_0, a, b$  play the same role as defined in the SE covariance function. RQ can be considered as a scaled mixture of squared exponential functions based on theory that sum of the covariance function is also a covariance function. The parameter  $\nu$  controls the rate of decay of the covariance function. When  $\nu \rightarrow \infty$ , the RQ covariance function becomes SE function resulting in infinitely mean-square differentiable sample functions and processes are smooth. In comparison with the other family of covariance functions, Matern, RQ covariance function results in mean square differentiable processes for any finite value of  $\nu$ , whereas in Matern, a finite  $\nu$  (E.g.  $\nu = 1/2$  and  $\nu = 3/2$ ) results in rougher process<sup>(7)</sup>.

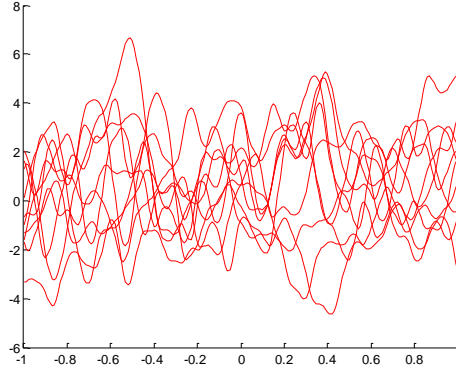


Figure 2 – GP Prior sampling with RQ covariance function

### Training and Prediction Framework

Once the covariance function is fixed, the dataset log likelihood and the partial derivatives of log likelihood with respect to the parameters  $\boldsymbol{\theta} = \{v_0, b, a_i, \sigma^2\}$  (squared exponential) or  $\boldsymbol{\theta} = \{v_0, b, a_i, \nu, \sigma^2\}$  (rational quadratic) are calculated where  $\sigma^2$  represent the Gaussian noise of variance  $\sigma^2$ , added to the covariance parameter set  $\boldsymbol{\theta}$ . The log likelihood is given by

$$L = p\left(\frac{\mathbf{t}}{\boldsymbol{\theta}}\right) = -\frac{1}{2} \log \det K - \frac{1}{2} \mathbf{t}^T K^{-1} \mathbf{t} \quad (10)$$

Once the above log likelihood and partial derivatives are calculated, the optimised parameters are found using scaled conjugate gradient algorithm.

Using the trained model, prediction on a new set of data may be performed. In equation (10), the covariance matrix term  $K^{-1}$  depends on the training data alone. To calculate prediction  $T^*$  for a new set of test data  $\mathbf{x}^*$ , it is necessary to compute conditional distribution  $p(T^* | T_1, T_2, \dots, T_N)$ . As the trained model is Gaussian, this distribution is also Gaussian. If  $\mathbf{K}$  denotes the covariance matrix of the training data,  $\mathbf{k}$  denote the  $N \times 1$  covariance between the training data and  $T^*$  and  $k^*$  denotes the variance of  $T^*$ , the predicted mean and variance given at  $\mathbf{x}^*$  as follows <sup>(5)</sup>:

$$E[T^*] = \mathbf{k}^T K^{-1} \mathbf{t} \quad (11)$$

$$\text{var}[T^*] = k^* - \mathbf{k}^T K^{-1} \mathbf{k} \quad (12)$$

From the above equations (11) and (12), GP can be used for prediction by defining covariance function without actually defining basis functions in equation (3), assuming GP with zero mean.

## **Automatic Relevance Determination**

In the GP Bayesian framework, a Gaussian prior distribution is assumed and each input variable is associated with a separate hyper parameter. During Bayesian learning, these hyperparameters are optimised using a scaled conjugate gradient algorithm. After training, the hyperparameters can be used for relevance determination<sup>(5)</sup>. As the hyperparameters represent inverse variance (1/variance) of the prior distribution, larger values of a hyperparameter mean that the weights  $a_i$  in equation (10) and equation (11) are near zero and the corresponding input becomes statistically less important to the target.

## **3. Experimental Data and Results**

The section presents the results obtained using the three different analytical techniques described above to an exemplar dataset of engine data. It is shown that the techniques are not only able to detect “abnormal” trends in the data, but also give some descriptive information about the engine behaviour.

### ***3.1. Engine Data***

Nine different datasets, each containing data from four engines, were obtained during the period 2006-2007. Each dataset contains engine parameters such as pressure, temperatures, speed, etc. The parameters were measured at fixed intervals during the flights.

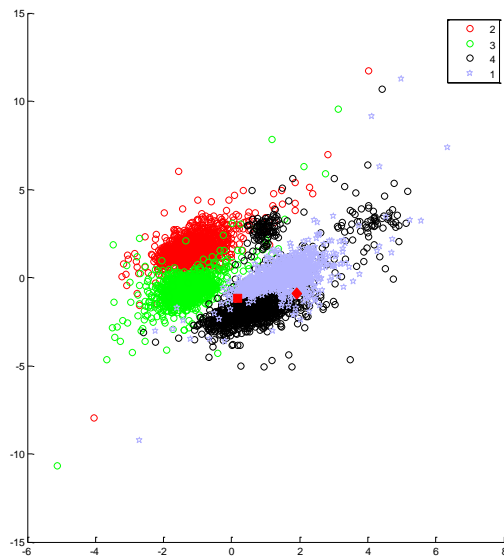
### ***3.2. Data Preprocessing***

Features vector consist of performance parameters such as pressure, temperature and derived parameters are extracted from the data. The extracted features were then combined in a (high-dimensional) feature vector for further analysis.

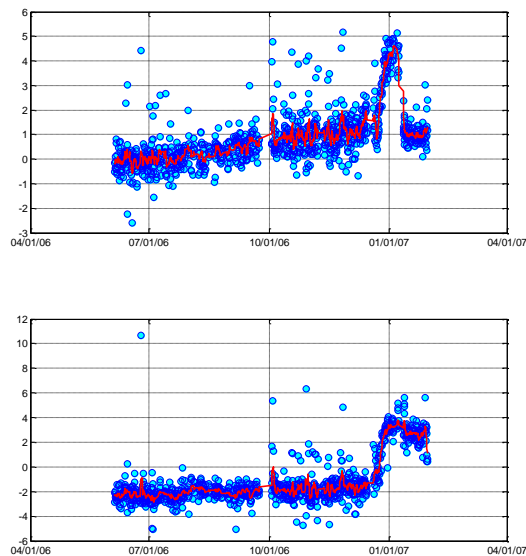
Once the set of feature vectors was obtained, additional pre-processing of the data was performed by removing invalid measurements due to acquisition errors. Secondly, normalisation was performed to ensure that all parameters varied over a similar range, such that variables which have a large dynamic range do not dominate the analysis of variables that vary over a smaller dynamic range. In this paper, a component-wise normalisation (i.e., a zero-mean, unit-variance transformation) was applied, where each feature is standardised with respect to its own mean and variance.

### ***3.3. Visualisation using Sammon’s mapping***

For Sammon’s mapping, a set of five-dimensional feature vectors was extracted from exemplar flight data. In Figure 3, the visualisation of feature vectors using Sammon’s mapping for the exemplar dataset 1 is shown. Figure 4 shows the two coordinates of the Sammon projection, through time. It shows a clear step change event.



**Figure 3 Event Detection in Flight Dataset 2- Sammon Projection**  
 (Each engine plotted in different colours, a red square and diamond mark the first and last data points respectively for one of the engines)



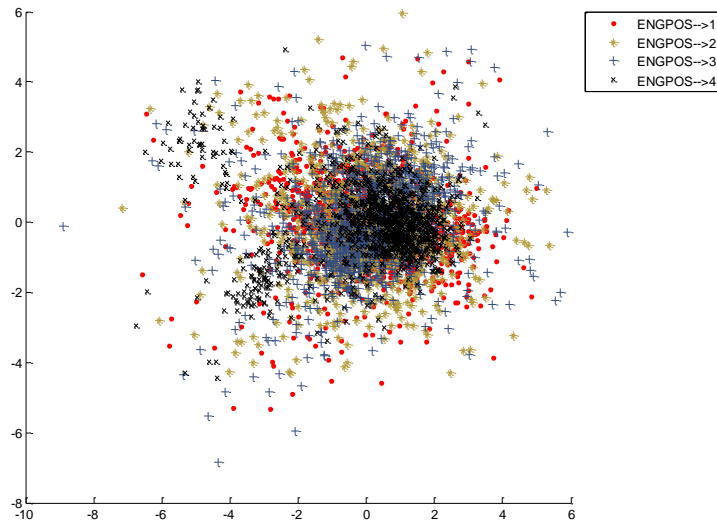
**Figure 4 Step Changes Detection in Flight Dataset 2 - Sammon Visualisation through time**

### 3.4. Visualisation using NeuroScale

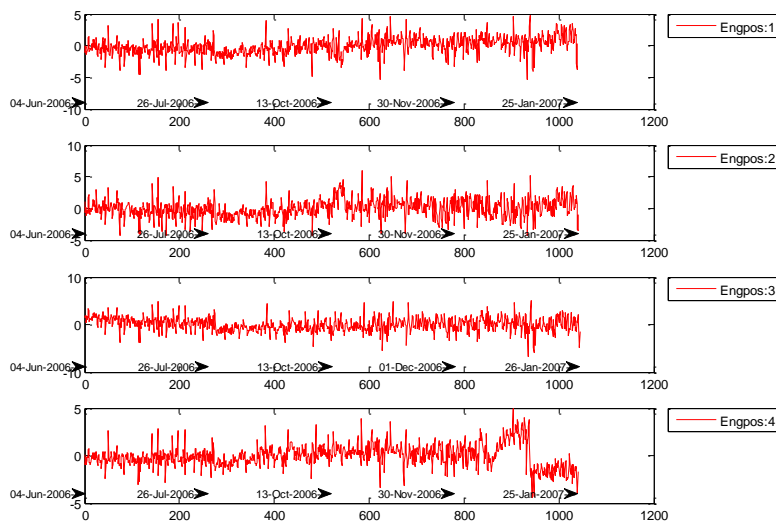
When applying the NeuroScale technique, a set of five-dimensional feature vectors was extracted from exemplar flight data, in a similar fashion to that used with Sammon's

mapping, described above. Figure 5 shows the visualisation of features derived from the engine performance parameters obtained using NeuroScale.

50 kernels were chosen for the number of hidden nodes in an RBF network, which was experimentally shown to provide suitable visualisation. The basis functions of the RBF form a spherical Gaussian mixture model trained using the EM algorithm. The width of each kernel is set to the empty matrix for non-Gaussian activation functions. The network is trained using the shadow targets algorithm<sup>(8)</sup>. Figure 6 shows the NeuroScale  $x$ -coordinates over time, respectively, in which Engine 4 may be seen to exhibit step-change behaviour between 01/12/2006 and 25/01/2007.



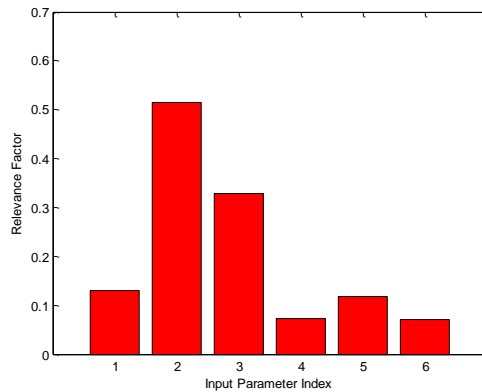
**Figure 5 Event Detection in Flight Dataset 2 - Neuroscale visualisation**



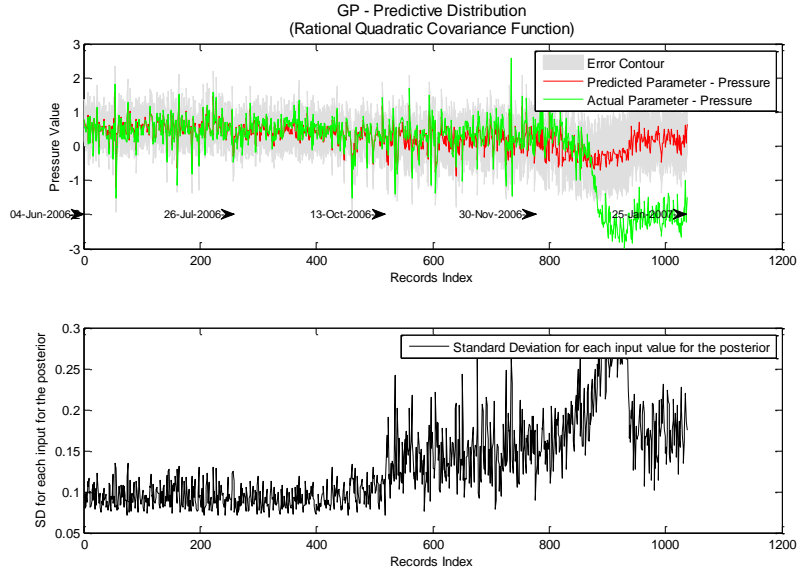
**Figure 6 - Neuroscale visualisation through time**

### 3.5. Gaussian Processes

The dataset described in section 3.1 was analysed using the GP technique to detect abnormal trends or events in the exemplar dataset. The GP prior is chosen as Gaussian with zero mean. Two types of covariance functions were used for analysis as described in Section 2.3. The hyperparameters (input weights  $a_i$ , bias  $b$ , noise variance  $\sigma^2$ , positive pre-factor  $\nu_0$ ) for the GP model are initialised randomly in the range [0 1]. For each engine, the first 50% of the available data were taken for training of the GP, and the hyperparameters were trained using a Scaled Conjugate Gradient Optimisation algorithm <sup>(4)</sup>. The remainder of the data is used for testing. Figure 8 shows the GP predictive performance for Dataset 2. Step change is (potentially indicative of engine deterioration or other interesting behaviour) observed in the data compared to GP predicted output between 01/12/2006 and 25/01/2007. At the end of training, the weights  $a_i$  denote the weights allowances corresponding to the inputs parameters towards predicting output. The relevance vector is calculated by taking the exponential of the weights. From experiments, it is read as follows: 0.11137, 0.45223, 0.04098, 0.07456, 0.11106 and 0.04894 for six parameters respectively. This indicates the parameter 2 is statistically more relevant to the target followed by parameters 1, 5, 4 and then parameters 6 and 3.



**Figure 7 Automatic Relevance Determination – SE GP Process with 6 input parameters**



**Figure 8 Flight Dataset 2 – Engine 4 - Step Change Event – GP Predictive performance**  
 (Note: Error contour (the shaded area) represents the pointwise  $\mu \pm 2\sigma$  for each input value for the posterior)

One of the challenges of using GPs for prediction is the complexity involved in computing the inverse covariance function. For a dataset comprising  $N$  patterns, the complexity is  $O(N^3)$ . To use GPs for the analysis of fleets of engines requires the use of much larger datasets, and thus complexity reduction techniques will be investigated as future work. These techniques could include subdividing the large datasets into a number of smaller subsets and using a committee of GPs or a mixture of experts<sup>(9)</sup>. Finally, the two covariance functions described in this paper result in processes that are infinitely differentiable and smooth. Other covariance functions such as the “Matern” class<sup>(6)</sup> that allow rougher processes will be investigated for their suitability in future case studies.

## 5. Conclusions

Performance datasets from different flights containing engines with examples of events or trends were investigated using a modified version of Sammon’s mapping, the NeuroScale method, and the Gaussian Process technique. Results demonstrate that the two visualisation techniques help the user to identify abnormalities by dimensionality reduction. The GP gives the expected values of a feature vector and the observed values of the feature vector are compared with the predicted values for abnormalities.

This paper introduces a modified Sammon’s mapping algorithm by random sub-sampling of the complete set of inter-vectors distances and overcomes the perennial problem associated with Sammon maps of requiring  $O(N^2)$  storage and computation. Secondly, using Gaussian Process for prediction, a confidence measure is associated with the predictive distribution. In abnormal cases, when the observed engine data begins to deviate from the training data distribution (of assumedly normal data), predictive variance increases. This confidence measure provides a graphical method of

communicating results of analysis to users. Finally, in contrast to conventional methods in which input parameters are chosen in an *ad hoc* manner, this paper has performed Automatic Relevance Determination (ARD) to identify the effect of each input on the predictive output.

## Acknowledgements

The authors acknowledge the support of Oxford BioSignals Ltd. and Rolls-Royce PLC. The authors wish to thank Paul Flint and Lionel Tarassenko for valuable discussions.

## References

1. L. Tarassenko, D.A. Clifton, P.R. Bannister, S. King, and D. King: ‘Novelty Detection’, in K. Worden, et al. (eds): Encyclopedia of Structural Health Monitoring. John Wiley and Sons, New York, 2008.
2. J.W. Sammon, 'A Non-linear Mapping for Data Structure Analysis', IEEE Transactions on Computers 18(5), pp 401–409, 1969.
3. D. Lowe and M.E. Tipping, ‘Feed-Forward Neural Networks and Topographic Mappings for Exploratory Data Analysis’, Neural Computing and Applications, Vol. 4, pp 83-95, 1996.
4. C.M. Bishop, ‘Pattern Recognition and Machine Learning’, Springer-Verlag, Berlin, 2006.
5. I. Nabney, 'Netlab: Algorithms for Pattern Recognition', Springer-Verlag, Berlin, 2002.
6. D. J. C. MacKay, ‘Introduction to Gaussian processes’, In C. M. Bishop, editor, Neural Networks and Machine Learning, volume 168 of NATO ASI Series, pp 133-165, Springer, Berlin, 1998.
7. Carl Edward Rasmussen and Chris Williams, "Gaussian Processes for Machine Learning”, chapter 4, the MIT Press, 2006
8. D. Lowe and M. E. Tipping. Neuroscale: Novel topographic feature extraction with radial basis function networks. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, Advances in Neural Information Processing Systems 9. Cambridge, MA: MIT Press, 1997.
9. R. A. Jacobs, M. I. Jordan, S. Nowlan, and G. E. Hinton, “Adaptive mixtures of local experts”, Neural Computation, 3, pp 1-12, 1991.