Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2007)

Gaussian Process Latent Variable Models for Fault Detection

Luka Eciolaza[1], M. Alkarouri[1], N. D. Lawrence[2], V. Kadirkamanathan[1], and P.J. Fleming[1]

[1] *Rolls-Royce Supported University Technology Centre in Control and Systems Engineering, Dept of Automatic Control and Systems Engineering, The University of Sheffield, Mappin Street, Sheffield, S1 3JD, UK.*

[2] *Dept of Computer Science, The University of Sheffield, Regent Court, 211 Portobello Street, Sheffield, S1 4DP, UK.*

Abstract— **The Gaussian process latent variable model (GPLVM) is a novel unsupervised approach to nonlinear low dimensional embedding proposed by Lawrence (2005). This paper presents the development of a framework for the implementation of the GPLVM for fault detection. A series of experiments have been carried out comparing and combining the GPLVM to the conventional and widely used linear dimension reduction technique of Principal Component Analysis (PCA). The inclusion of the GPLVM for the visualisation and data analysis, led to a considerable improvement in the classification results.**

Index Terms—**Fault detection, Dimensionality reduction, Principal Component Analysis.**

I. INTRODUCTION

An increasingly massive amount of data is being generated by the use of modern technologies in a multitude of domains, such as engineering. These modern technologies and the growing concern of the health monitoring and product condition monitoring concepts lead to large scale, highly complex, multivariate systems. Early and accurate fault detection and diagnosis for these systems can minimize down time, and reduce manufacturing costs. Industrial products are becoming more heavily instrumented, resulting in more data becoming available for use in detecting and diagnosing faults. It is therefore critical to many businesses to have adequate means in place for transforming the vast volumes of data into information relevant for decision-making.

By projecting the data into a lower-dimensional space that accurately characterizes the state of the analysed system, dimensionality reduction techniques can greatly simplify and improve health monitoring and fault detection procedures. In this paper, the Gaussian process latent variable model (GPLVM) is used for fault detection purposes and it is compared against the Principal component analysis (PCA), technique that has been studied and implemented by several academic and industrial engineers for fault detection [1].

The GPLVM [2], [3], [4] is a novel unsupervised approach to nonlinear low dimensional embedding. It has been tested in a number of applications with satisfactory results, as in [5]. This paper presents the results of using it in a fault detection environment. The different experiments carried out and explained throughout the paper show considerable improvement over PCA for the obtained visualisation and classification results.

The layout of the paper is as follows: Section 2 describes briefly the main characteristics of the original data used for the analysis and the pre-processing stage realised to it. Section 3 outlines the theory behind the two techniques used for dimensionality reduction. Section 4 shows the experiments realised to compare and combine the two used techniques, and gives the results obtained. The paper concludes with a discussion in Section 5.

II. ORIGINAL DATA FORMAT AND THE PREPROCESSING OF IT

The main purpose of the work presented on the paper is to show the fault detection capabilities of the novel dimensionality reduction technique GPLVM. For this reason and due to confidentiality requirements we will not focus on giving detailed information of the nature of the data analysed. However, for the better understanding of the work carried out some key facets must be explained. The data analysed within this paper is from an extremely reliable product type where operational failure events are rare, and so for this particular analysis, an artificially balanced dataset with good and bad product tests has been used, with 400 tests in total from 200 product serial numbers.

An example of the product test type used for this analysis is shown in Fig. 1. This type of test is designed to monitor the performance characteristics of the product and it consists of running the product through seven predefined steady-state stages over the whole range of operating conditions the product can provide.

For the analysis reported in the paper, information from twelve parameters of the tests have been used. The nature of the variables used was multifaceted, going from vibration information of high dynamics to various temperatures and pressures where the dynamics are slower.

A. Data Pre-processing:

The authors gratefully acknowledge financial support of the DTI, UK, for the project BROADEN and contributions from Rolls-Royce plc.

1-4244-0705-2/07/$20.00 ©2007 IEEE

The whole dataset used for the analysis goes through a pre-processing stage in order to get it ready for the posterior analysis. Our aim is the transformation of the measurement signals into a set of multi-dimensional features retaining as much relevant information as possible. The pre-processing stage of the data needs various steps:

Data Segmentation: As mentioned before the analysed tests are divided into seven well defined steady state stages where the performance of the products through their power ranges are monitored. This test is one manoeuvre out of a number of manoeuvres applied to the product for testing different characteristics, so in one first step of the segmentation, the particular selected manoeuvre where the performance characteristics of the product are analysed must be identified. Once these manoeuvres are identified, they are segmented into different stages depending on the variation of the power indicator signal. This means that after the two steps of the segmentation, seven independent portions of time-series data will be identified from each test.

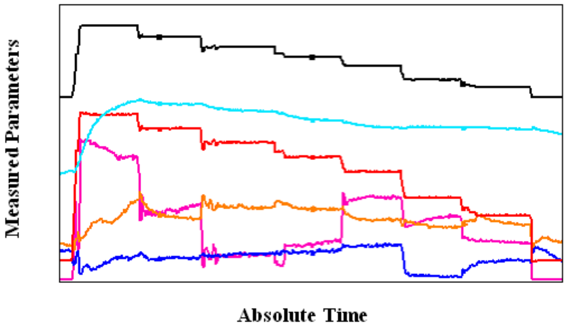


Fig. 1. Example of a data file used for the analysis

Feature extraction: From each of the 7 segments of the tests, and for each one of the 12 measured parameters, various standard statistical features for time-series classification are extracted. Those features are the mean value,

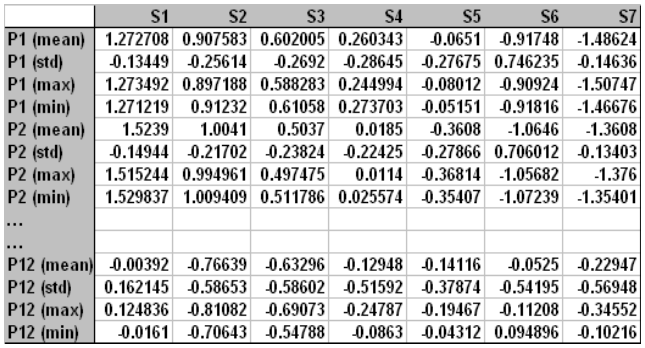
$$\mu = \frac{\sum_{t=1}^{n} y(t)}{n}, \tag{1}$$

standard deviation,

$$\sigma = \sqrt{\frac{\sum_{t=1}^{n} (y(t)-\mu)^2}{n}}, \tag{2}$$

the maximum value, and minimum value. These features are selected because they are simple and easy to implement and although more elaborated features may lead to better results, statistical features are a starting point for the evaluation of the method, and the improved methods of feature selection should be addressed in future work.

Considering there are 7 segments with 12 parameters recorded in each segment and 4 features extracted out of each one of the recorded parameter, as a result and for each test we obtain a summary table of 7 columns by 48 rows as in Fig. 2.



	S1	S2	S3	S4	S5	S6	S7
P1 (mean)	1.272708	0.907583	0.602005	0.260343	-0.0651	-0.91748	-1.48624
P1 (std)	-0.13449	-0.25614	-0.2692	-0.28645	-0.27675	0.746235	-0.14636
P1 (max)	1.273492	0.897188	0.588283	0.244994	-0.08012	-0.90924	-1.50747
P1 (min)	1.271219	0.91232	0.61058	0.273703	-0.05151	-0.91816	-1.46676
P2 (mean)	1.5239	1.0041	0.5037	0.0185	-0.3608	-1.0646	-1.3608
P2 (std)	-0.14944	-0.21702	-0.23824	-0.22425	-0.27866	0.706012	-0.13403
P2 (max)	1.515244	0.994961	0.497475	0.0114	-0.36814	-1.05682	-1.376
P2 (min)	1.529837	1.009409	0.511786	0.025574	-0.35407	-1.07239	-1.35401
...							
...							
P12 (mean)	-0.00392	-0.76639	-0.63296	-0.12948	-0.14116	-0.0525	-0.22947
P12 (std)	0.162145	-0.58653	-0.58602	-0.51592	-0.37874	-0.54195	-0.56948
P12 (max)	0.124836	-0.81082	-0.69073	-0.24787	-0.19467	-0.11208	-0.34552
P12 (min)	-0.0161	-0.70643	-0.54788	-0.0863	-0.04312	0.094896	-0.10216

Fig. 2. Example of summary table with extracted normalised features from each test. (S1… S7) refers to the seven segments of each test. (P1 … P12) refers to the 12 measured parameters in each test.

Normalisation: The value ranges of the parameters vary a lot depending on their nature. Considering that the methods used for this work will aim to correlate the information extracted from these parameters across all the tests, and in order to avoid the influence of this variation of values between parameters, a normalisation process of each extracted feature independently across all the tests has to be done. The translation

$$y_{norm} = \frac{y-\mu}{\sigma}, \tag{3}$$

is used for the normalisation and as a result the features across the tests will be normally distributed with a mean of zero and a standard deviation of 1.

III. METHODS OF ANALYSIS

It is the first time where the GPLVM have been used for fault detection purposes within a complex system of an industrial environment. This paper presents the work done to evaluate its performance, comparing it with the conventional and widely used PCA. Both of these techniques are dimensionality reduction techniques originally introduced as a way to overcome the curse of the dimensionality when dealing with vector data in high-dimensional spaces and as a modelling tool for such data.

Dimensionality reduction is defined as the search for a low dimensional manifold that embeds the high-dimensional data. A manifold (a coordinate system) that will allow projection of the data vectors on it and obtain a low-dimensional, compact representation of the data.

The two compared methods are described in this section. Throughout the rest of the paper, vectors in the measured data space will be denoted by **{y}** and the dimension of the space by *d*. Vectors in the reduced (or latent) space will be denoted by **{x}** and the dimensions of that space will be *q*.

A. Principal Component Analysis (PCA)

This method is perhaps the most widely used technique for obtaining a lower dimensional representation of a dataset, probably due to its conceptual simplicity. The relatively efficient algorithm seeks orthogonal linear projections of the data with maximum variance, looking for a linear embedding of the data which is optimal under linear reconstruction for a quadratic loss [6].

Only the briefest description will be given here. The principal components algorithm seeks to project by a linear transformation, the data into a new *d*-dimensional set of Cartesian coordinates $(x_1, x_2, \ldots, x_n)$. The new coordinates have the following property: x_1 is the linear combination of the original y_i with maximal variance, x_2 is the linear combination which explains most of the remaining variance and so on. It should be clear, that if the *d*-coordinates are actually a linear combination of *q*<*d* variables, the first *q* principal components will completely characterize the data and the remaining *d–q* will be zero. In practice, due to measurement uncertainty, the principal components will all be non-zero and the user should select the number of significant components for retention.

The required computation steps are as follows: Given data $\{y\}_i = \{y_{1i}, y_{2i}, \ldots, y_{ni}\}$, *i*=1,…,*N*, form the covariance matrix C as,

$$C = \frac{1}{N}\sum_{i=1}^{N}(\{y\}_i - \{\overline{y}\})(\{y\}_i - \{\overline{y}\})^T , \quad (4)$$

from the sample vectors, where $\{\overline{y}\}$ is the vector of means of the y data, and then perform an eigenvalue decomposition of it using,

$$C = V \Lambda V^T , \quad (5)$$

where the diagonal matrix Λ contains the non-negative real eigenvalues.

The transformation from data space to the coordinate system defined by the principal components is computed as,

$$\{x\}_i = V^T(\{y\}_i - \{\overline{y}\}) . \quad (6)$$

Considered as a means of dimension reduction then, PCA works by discarding those linear combinations of the data that contribute least to the overall variance or range of the dataset.

B. Gaussian Process Latent Variable Models (GPLVM)

The GP-LVM [2], [3] is a fully probabilistic, non-linear, latent variable model that generalizes principal component analysis. The model was inspired by the observation that a particular probabilistic interpretation of PCA is a product of Gaussian process models each with a linear covariance function. Through consideration of non-linear covariance functions a non-linear latent variable model can be constructed.

The probabilistic approach to dimensionality reduction is to formulate a latent variable model, where the latent dimension, *q*, is lower than the data dimension, *d*. The latent space is then governed by a prior distribution *p*(X), where the points in latent space are given by $X = [x_1 \ldots x_N]^T$. The latent variable is related to the observation space through a probabilistic mapping,

$$y_{ni} = f_i(x_n) + e_n$$

where y_{ni} is the *i*th feature of the *n*th data point and '*e*' is a noise term that is typically taken to be Gaussian.

The GPLVM places the prior distribution over the mappings rather than the latent variables as in the probabilistic PCA (PPCA) proposed in [7]. The mappings may then be marginalised and the marginal likelihood,

$$p(Y \mid X) = \prod_{i=1}^{d}\prod_{n=1}^{N} p(y_{in} \mid f_{in})\, p(f \mid X) ,$$

optimised with respect to the latent variables.

This technique provides a smooth probabilistic mapping from latent to data space. That means that while most approaches to non-linear dimensionality methods focus on preserving local distances in data space, the GPLVM focuses on exactly the opposite, keeping things apart in latent space that are far apart in data space.

However, as shown in [4], the GPLVM can be generalized, through back constraints, to additionally preserve local distances. In the back-constrained GPLVM, the likelihood is optimised with the constraint of local distance preservation. This constraint is implemented learning the mapping from the data space to the latent space. This means that there will be two models working simultaneously: a dissimilarity preserving, probabilistic GPLVM mapping from latent to data space, and a local distance preserving mapping from data to latent space.

In the experiments realised on this work the back-constrained GPLVM was used, considering important local distance preservation from data to latent space.

(See http://www.dcs.shef.ac.uk/~neil/gplvm for the MATLAB code of the GPLVM)

IV. FRAMEWORK FOR EXPERIMENTS

Having explained the main characteristics of the data used for the analysis, the pre-processing stage and the basics of the two different dimensionality reduction techniques compared in this work, the various experiments carried out, and summarised in Fig. 3, will be introduced in this section.

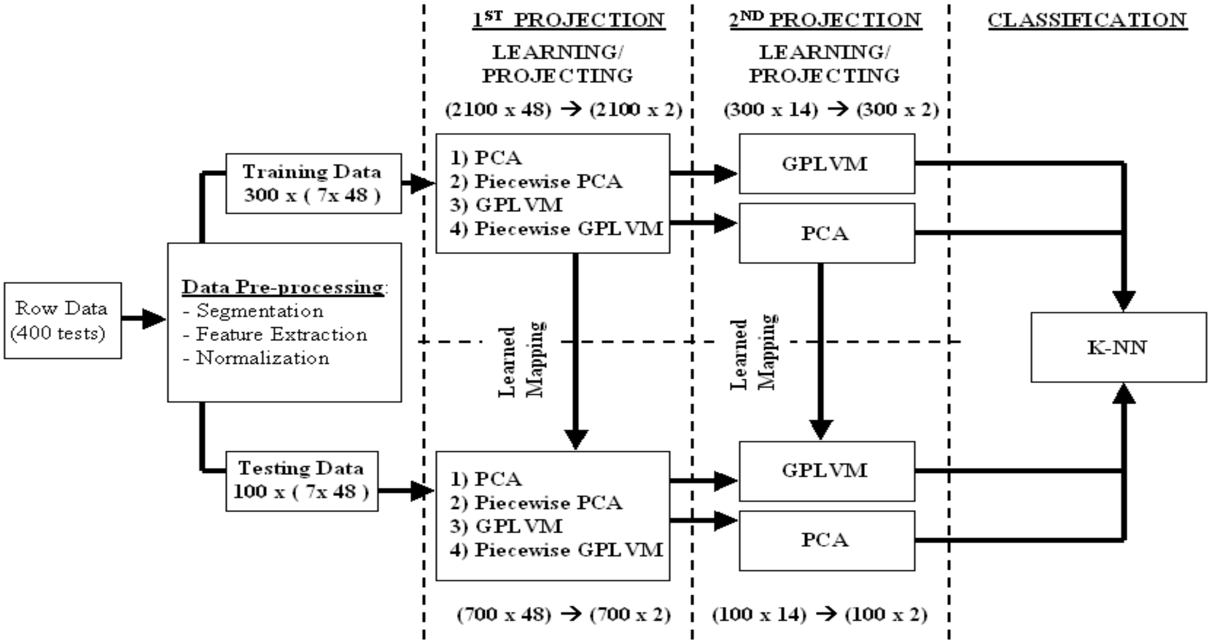


Fig. 3. Framework designed for the comparison and combination of the dimension reduction techniques, GPLVM and PCA, in fault detection applications.

To begin with, the whole dataset is divided into 2 different sets, one the training data (300 tests) and the other one the testing data (100 tests). The training data is needed to characterise the behaviour of the product family and learn the conversion rules for the correspondent dimension reductions. Fig. 3 shows the framework used for the experiments. The objective of the dimensionality reduction techniques is to project the summary data table, with the extracted features, of the original data files down to a bi-dimensional space where its performance state will be summarised into two coordinates, and then the K-Nearest Neighbour (K-NN) classification algorithm is applied to evaluate the fault detection capabilities of each technique for this particular dataset.

After the pre-processing stage and before the data of a particular data file gets reduced to 2 dimensions, there are two independent projection stages.

A. 1st Projection Stage

The original data files have seven pre-defined steady state segments and from each one a full feature column is extracted (Fig. 2). In this first projection stage all of the seven segments are considered as independent data measurements and they are projected individually down to 2 dimensions. An example of the projection obtained is shown in Fig. 4 where it is clear the division of the data into 7 classes, each one representing one of the independent segments of the original data files. This stage is part of the product data analysis and apart from representing a powerful visualisation functionality where the tests are presented as curves of 7 points, information about most influential segments and variables for the projections can be established.

Four different approaches have been considered for this first projection stage. These approaches determine the four experiments realised and analysed for the comparison of the two used reduction techniques. In all of them, from an initial 2100x48 matrix available for training data (300 tests each with 7 segments), we get a new 2100x2 matrix.

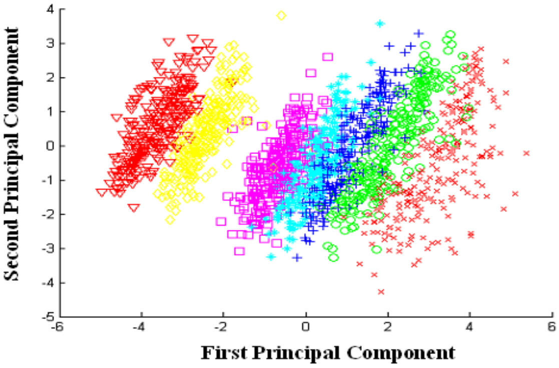


Fig. 4. Example of the result obtained after the first projection stage. The figure shows the projection obtained applying the PCA to the data. There are 2100 points divided in 7 classes.

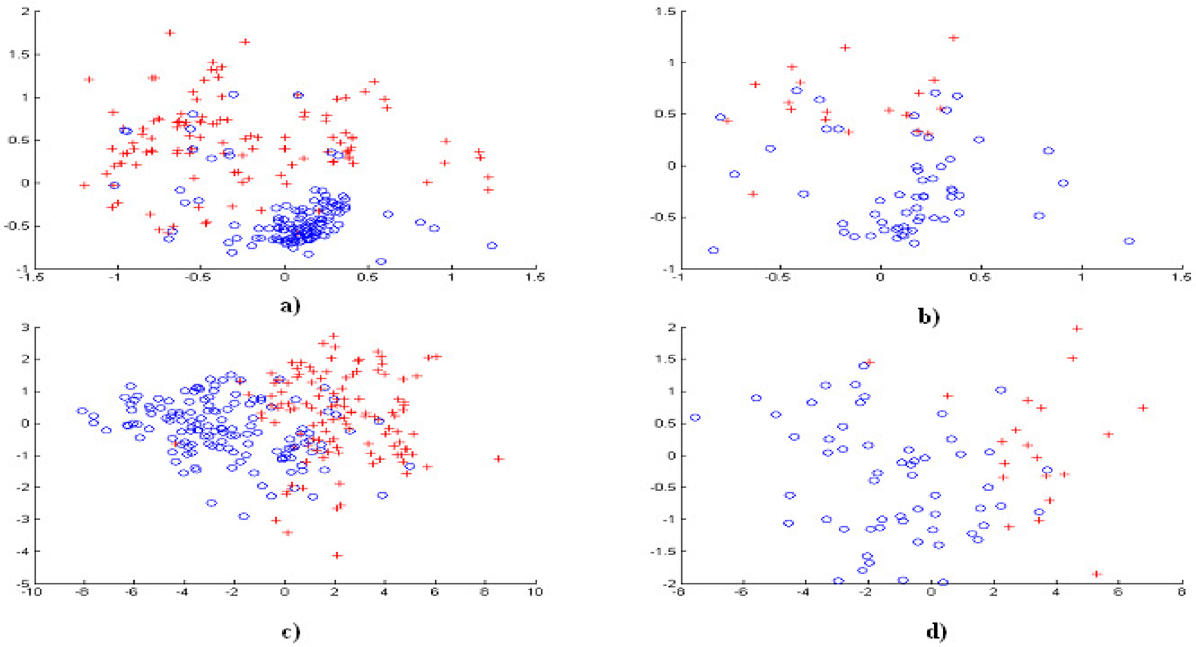


Fig. 5. (a, b, c, d) are some example figures obtained in the second projection stage of the analysis. (a) represents the projection of the training dataset using GPLVM for both projection stages. (b) shows the testing data projected into the latent space with the mappings learned from the reduction realised for (a). (c) represents the projection of the training dataset using PCA for both projection stages. (d) shows the testing data projected into the latent space with the mappings learned from the reduction realised for (c).

PCA: Linear PCA is applied to the data available after the pre-processing. Fig. 4 shows the projection of the data obtained from it.

Piecewise PCA: In this approach, an independent PCA is applied for each one of the seven steady state segments of the original files. E.g.: First segment information from the 300 training sets (300x48) are reduced to a 300x2 matrix, and so on. Fig. 4 shows that the first principal component information, applying a simple PCA to the whole data, is equivalent to saying which of the seven segments within the test the point belongs to. With this new approach more meaningful principal components of each segment independently are intended.

GPLVM: The GPLVM is applied to the data available after the pre-processing.

Piecewise GPLVM: Likewise the Piecewise PCA, in this approach, an independent GPLVM is applied for each one of the seven steady state segments of the original files.

B. 2^{nd} Projection Stage

After the first projection stage the original data files are reduced to (7x2) matrices. In the second projection stage, these (7x2) matrices will be grouped as (1x14) vectors, and they will be reduced to a single point (two coordinates). Examples of projections obtained in this second stage are shown in Fig. 5a, and 5c, where each point represents one product test. Effectively, this is the stage used for the comparison of the GPLVM and the PCA. Both techniques will have the same data input, which comes from the first projection stage, and it represents the ideal framework to compare and evaluate the quality of the dimension reduction capabilities of both techniques. The analysed dataset has two known classes. The technique with best fault detection capabilities will be the one that separates best the two classes, so the one that after applying some classification algorithm gives the best result.

For the projection of testing data the mappings learned from the training data are used. With the PCA technique, the projection of the testing data is a straightforward task where (6) needs to be applied with the eigenvectors obtained from training data. For the GPLVM the projection of test data is more complicated because the mapping learned from the training data is from latent (projection) to data space. However, the use of back constraints in order to preserve local distances in data space means that a mapping from data to latent space is also learned and so the additional advantage of using back constrained GPLVM is the straightforward task of projecting the testing data. . Fig. 5b and 5d are some examples of the testing data projected with the obtained conversion rules, in this case the conversion rules obtained from Fig. 5a and 5c respectively.

TABLE I
EXPERIMENTAL RESULTS

Experiment Number	First Projection Method	Second Projection Method	Classification Results for Training Data (%)	Classification Results for Testing Data (%)
1)	PCA	GPLVM	**90.3**	**84.9**
		PCA	90	75.6
2)	Piecewise PCA	GPLVM	90.7	75.6
		PCA	90	70.9
3)	GPLVM	GPLVM	92	76.7
		PCA	88.7	79.1
4)	Piecewise GPLVM	GPLVM	91.3	75.6
		PCA	**88**	**86.1**

The results obtained from the different experiments are shown in Table I, where the classification results for two classes is presented. The K-Nearest Neighbour algorithm is used for this classification.

V. DISCUSSION

This paper presents the results of the analysis comparing the GPLVM against PCA for fault detection purposes. The reading of these results show the clear benefit of including the GPLVM within the analysis framework developed. Any dataset obtained from an industrial process, will always have some kind of non-linearity in it because of the elements included (sensors, etc) that not being perfect always introduce little errors and so nonlinearities e.g.: the hysteresis effect. With the inclusion of the GPLVM for the analysis and fault detection of this industrial processes, the nonlinearities are considered and as a result the final result improves. Taking into account experiments 1 and 2, where variations of PCA are applied for the first projection stage. This stage could be considered as part of the linear pre-processing stage. In the second projection stage, the GPLVM improves the classification results obtained with respect to the PCA, and for testing data in particular this improvement is considerable. It definitely shows the added value of using the GPLVM.

Taking into account experiments 3 and 4 where variations of GPLVM were used in the first stage, the use of PCA for the second projection gives better results than the GPLVM. It seems that the nonlinearities of the dataset are modelled in the first projection stage and a linear technique for the second stage gives better results. However, the really interesting thing in the 3 and 4 experiments is that if PCA performance is compared to PCA performance in the 1 and 2 experiments, the results in 3 and 4 have improved dramatically, again showing the added benefit of using the GPLVM.

Out of the 4 experiments presented within the paper, the best results are obtained with a combination of reduction techniques. (PCA + GPLVM) and (Piecewise GPLVM + PCA) are the models with better results.

It is clear that using the GPLVM at some stage of the data analysis improves considerably the final results, obtaining better separation in the projections and so better fault detection results comparing to PCA.

REFERENCES

[1] L.H. Chiang, E.L. Russell, and R.D. Braatz, “Fault Detection and Diagnosis in Industrial Systems”, London: Springer-Verlag, 2001.

[2] N.D. Lawrence, “Gaussian process models for visualisation of high dimensional data”, Advances in Neural Information Processing Systems, pp. 329-336. Cambridge, MA: MIT Press, 2004.

[3] N.D. Lawrence, “Probabilistic non-linear principal component analysis with Gaussian process latent variable models”, Journal of Machine Learning Research, 6, (pp 1783-1816), 2005.

[4] N.D. Lawrence, J. Quiñonero-Candela “Local Distance Preservation in the GPLVM through Back Constraints”, Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, Pa, 2006.

[5] K. Grochow, S. Martin, A. Hertzmann, and Z. Popovi’c, “Style-Based Inverse Kinematics”, ACM Transactions on Graphics 23, 3, 2004.

[6] I.T. Jolliffe, “Principal Component Analysis”, New York: Springer-Verlag, 1998.

[7] M.E. Tipping, C.M. Bishop, “Probabilistic principal component analysis”, Journal of the Royal Statistical Society, B, 6, 611-622, 1999.