DETECTION OF PLANT-WIDE DISTURBANCES USING A SPECTRAL CLASSIFICATION TREE

Nina F. Thornhill* and Hallgeir Melbø+

*Department of Electronic and Electrical Engineering, UCL, UK
+ABB Corporate Research Centre, Billingstad, Norway

Abstract: This article demonstrates the use of agglomerative hierarchical clustering to detect the structure within a data set. When combined with spectral principal component analysis to capture the main spectral features of a data set it allows visualization of the structure of a model with an optimum number of principal components. The paper presents the theory and methods for construction of the tree and gives an example using industrial data. *Copyright © 2006 IFAC*.

Keywords: Clustering; fault detection; hierarchical classification; performance analysis; plant-wide; principal component analysis; process operation; process monitoring.

1. INTRODUCTION

Large data bases are being accumulated by companies operating oil, gas and chemical processes. When these data are used in a plant audit, the aim is to find groups of measurements having similar characteristics so that the propagation paths of disturbances can be tracked through a process. With large data sets, however, it becomes challenging to present the results of a multivariate analysis. While principal component analysis (PCA) might reduce several hundred measurements to, say, ten principal components there then remains an issue of presentation of the ten-dimensional model to the analyst. This paper demonstrates a method for visualization of the clusters in a high-dimensional spectral PCA model by means of a hierarchical classification tree. The key elements in the procedure are an agglomerative hierarchical clustering algorithm and a recursive algorithm to create the tree.

Multivariate analysis using spectral methods has recently been reported for the case of dynamic disturbances (Thornhill *et. al.*, 2002; Xia and Howell, 2005). The benefits for plant audit application are that the power spectra are insensitive to time delays or phase lags between different measurement points and therefore bypass the need for time shifting and other methods needed for correlation-based analysis in the time domain.

The next section of the paper gives a review of related work while Section 3 gives the formulation of spectral multivariate data analysis. A distance measure for clustering analysis is also discussed in Section 3 together with the automated algorithm for creation of the hierarchical classification tree. An industrial data set is then analyzed to illustrate the concepts showing the clustering patterns present before and after a plant shutdown in which maintenance was carried out. The paper ends with a conclusion section.

2. BACKGROUND AND CONTEXT

2.1 PCA for cluster analysis

Descriptions of principal component analysis may be found from many sources, for example Chatfield and Collins (1980) and Wold *et. al.* (1987). In analytical chemistry, near infrared (NIR) and nuclear magnetic resonance (NMR) spectroscopy data are routinely analysed by PCA (Alam and Alam, 2005; Ozaki et. al., 2001) and Seasholtz (1999) described the industrial application of multivariate calibration in NIR and NMR spectroscopy at Dow Chemical Company. Principal component analysis has proved useful in other diverse areas such as paint colour analysis (Tzeng and Berns, 2005), in the analysis of the relationship between the crispness of apples and recorded chewing sounds (De Belie *et al*, 2000) and in water quality analysis (e.g. Brodnjak-Voncina *et al*, 2002). Industrial uses include principal component analysis for monitoring of machinery and process equipment (Wu *et. al*. 1999; Malhi and Gao, 2004; Flaten *et al*., 2005). All these applications have the common aim to discover structure within the data set, to ascertain the items within the data set that belong together and to relate the results to underlying mechanisms. They are normally run off-line.

A prevalent area in process monitoring [Wise *et al*., 1990; Kresta *et. al*., 1991; Wise and Gallagher, 1996; Qin, 2003] is on-line multivariate statistical process control in which new measurements are projected into a PCA calibration model that was developed during normal operation. Multivariate warning and alarm limits are set which test whether a new set of measurements is within the normal bounds captured by the calibration model [Jackson and Mudholkar, 1979; Martin and Morris, 1996].

The work in this paper is aimed towards the first type of application and concerns the visualization of structures in a data set and ascertains the items that belong together, where the *items* are the power spectra of time trends at a given measurement point.

2.2 Visualization of high dimension PCA models

The visualization of a high dimension multivariate PCA model has previously been examined by Wang *et. al*., (2004) for the purposes of a multivariate statistical process monitoring. They used parallel coordinates to display multiple dimensions of the score space. Each day of running in a data set was represented by one piecewise linear trend in the parallel coordinate plot, and these trends were overlaid on top of one another. Although it was not possible to see any structure within the plot it was possible to identify abnormal days of running by inspection of outliers in the parallel coordinates plot.

2.3 Hierarchical classification

Gordon (1987) gave a comprehensive review of hierarchical classification, distinguishing between agglomerative and divisive methods. Agglomerative hierarchical clustering is an unsupervised algorithm for building up groups of similar items from a population of individual items. The basic algorithm (Duda et. al., 2000) starts with *N* clusters each containing one item and proceeds as follows:

```
repeat
        find the pair of nearest clusters
        merge them into one cluster
until there is one cluster containing N items
```

The results of agglomerative hierarchical clustering may be visualized in a classification tree in which the items of interest are the leaves on the tree and are joined into the main tree and eventually to the root of the tree by branches. Industrial applications of clustering and/or classification trees have included methods for office buildings to detect days of the week with similar profiles of energy use (Seem, 2005), the presentation of results from an end-point detection method in a crystallization process (Norris *et. al*., 1997) and from analysis of illegal adulteration of gasoline with organic solvents (Wiedemann *et. al*., 2005). The method presented in this paper uses agglomerative classification in the score space of all significant principal components.

Classification trees are also used in divisive classification in which a large group of items is recursively split into subcategories. In the area of process analysis, divisive classification has been combined with PCA for detection of key factors that affect process performance in a blast furnace and a hot stove system generating hot air for the blast furnace (Lee *et. al*, 2004). Clusters of items appearing in the score space of the first two principal components were identified and then further divided into sub-clusters using PCA recursively.

3. METHODS

3.1 Spectral PCA

In spectral principal component analysis (PCA) (Thornhill *et. al*., 2002) the rows of the data matrix $\mathbf{X}$ are normalized power spectra $P(f)$:

$$\mathbf{X} = \begin{pmatrix} x_1(f_1) & \ldots & x_1(f_N) \\ .. & .. & .. \\ x_m(f_1) & \ldots & x_m(f_N) \end{pmatrix} \begin{matrix} N \text{ frequency channels } \rightarrow \\ m \text{ measurements } \downarrow \end{matrix}$$

A PCA decomposition reconstructs the $\mathbf{X}$ matrix as a sum over *p* orthonormal basis functions $\mathbf{w}'_1$ to $\mathbf{w}'_p$ which are spectrum-like functions each having N frequency channels arranged as a row vector:

$$\mathbf{X} = \begin{pmatrix} t_{1,1} \\ \ldots \\ t_{m,1} \end{pmatrix} \mathbf{w}'_1 + \begin{pmatrix} t_{1,2} \\ \ldots \\ t_{m,2} \end{pmatrix} \mathbf{w}'_2 + \ldots + \begin{pmatrix} t_{1,p} \\ \ldots \\ t_{m,p} \end{pmatrix} \mathbf{w}'_p + \mathbf{E}$$

The *i*'th spectrum in $\mathbf{X}$ maps to a spot having the co-ordinates $t_{i,1}$ to $t_{i,p}$ in a *p*-dimensional space. The $t_{i,1}$ to $t_{i,p}$ are called scores and represent the weightings of the basis functions needed to approximately reconstruct the spectrum in the *i*'th row of the data matrix. Similar spectra have similar *t*-coordinates and form clusters in the score space.

The key to finding meaningful clusters is the choice of distance measure. In process performance analysis the angular measure discussed in Duda *et. al*., (2000) is often more suitable than Euclidian distances. The reason for this observation is that the PCA clusters frequently take the form of plumes radiating from the origin. Raich and Cinar (1997) also observed plumes in their analysis of simulated faults in the Tennessee Eastman benchmark model.

Let he vector $\mathbf{t}'_i = \left(t_{i,1},\ t_{i,2}, \ldots\ t_{i,p}\right)$ be the *i*'th row of matrix $\mathbf{T}_p$ in $\mathbf{X} = \mathbf{T}_p \mathbf{W}_p^{\tau} + \mathbf{E}$ in a *p* principal component model. A measure for membership of a plume is that the direction of vector $\mathbf{t}'_i$ in the multidimensional score plot lies within the same solid angle as those of other $\mathbf{t}'$ – vectors belonging the plume. The angle between $\mathbf{t}'_i$ and $\mathbf{t}'_j$ may be determined through calculation of the scalar product:

$$\cos\left(\theta_{i,j}\right) = \frac{\mathbf{t}'_i \bullet \mathbf{t}'_j}{\left|\mathbf{t}'_i\right|\left|\mathbf{t}'_j\right|}$$

where:

$$\mathbf{t}'_i \bullet \mathbf{t}'_j = \sum_{k=1}^{p} t_{i,k}\, t_{j,k} \quad \text{and} \quad \left|\mathbf{t}'_i\right| = \sqrt{\sum_{k=1}^{p} t_{i,k}^{\;2}}$$

3.2 Clustering

A matrix $\mathbf{A}$, whose elements are $\theta_{i,j}$, is to be analyzed to find high-dimensional plumes in the PCA score plot. Two items in the score plot whose $\mathbf{t}'$ – vectors point in similar directions give a small value of $\theta_{i,j}$. The agglomerative hierarchical clustering algorithm is based on Chatfield and Collins (1980):

Algorithm: Agglomerative classification

Step 1: The starting point is the matrix of angular distances with elements $\theta_{i,j}$. A text vector of row and column headings is also defined which initially is $(1\ \ 2\ \ 3\ \ 4\ \ 5\ \ ....)$ to keep track of the items in the data set. For an process performance analysis application the items are the *N* plant profiles in the data set, for a process audit the items are the *m* tags.

Step 2: At the *k*'th iteration, the smallest non-zero value $\theta_{i,j}$ in the matrix is identified. Its row and column indexes *i* and *j* indicate the smallest angular separation and these are clustered together.

Step 3: A smaller matrix $\mathbf{A_k}$ is then generated from the original. It does not have rows and columns for the two similar items identified at step 2. Instead, it has one row and column that give the distances of all the other items from the cluster. The distances are $\min\{\theta_{i,n},\theta_{j,n}\}$, i.e. the angular distance between the *n*'th item and whichever member of the cluster was closer. For instance, if $\theta_{9,15}$ is the smallest angular separation in the matrix then rows 9 and 15 would be deleted and replaced by a new single row, and likewise for columns 9 and 15.

Step 4: The row and column headings are redefined. The heading for the new row created at step 3 indicates the items that have been combined. For instance, if the smallest angular separation at Step 3 had been $\theta_{9,15}$ then the new heading would be (9 15).

Step 5: The results of the *k*'th step are written to a report showing the cluster size defined as the maximum distance between items in the cluster, the row heading for the cluster formed at iteration *k*, and the two sub-clusters within it.

Step 6: Steps 2 to 5 are repeated until all the items have been clustered. At any stage, the outcome of the next step is either another item added to a cluster already identified or the combining of two items to start a new cluster.

A feature of the agglomerative hierarchical classification procedure presented here is that it provides a text-based report which enables the detection of significant clusters as well as automated generation of the hierarchical tree plot.

3.3 Dealing with noise

There is an assumption underlying a process audit which is that any tag whose power spectrum has spectral features is being upset by unwanted dynamics which could be reduced by control action. The assumption is justified in a control systems study where the idea is that nothing but random noise should be present. The spectrum of random noise is broad band and in theory it is flat. Such a spectrum maps to the origin in spectral PCA (i.e. the elements in $\mathbf{t}'_i = \left(t_{i,1},\ t_{i,2},\ ...\ t_{i,p}\right)$ are close to zero) because the p $\mathbf{w}'$ – vectors of the model reflect the spectral features in the data set. The broadband noise is captured by the remaining $m-p$ components and appears in the $\mathbf{E}$ matrix. Tags with small values of $\|\mathbf{t}'\|$ are therefore excluded from the hierarchical tree. In the case study presented below, the tags with the 90% longest $\mathbf{t}'$ – vectors were plotted. Therefore, out of 60 tags, 54 appear in the tree and the excluded ones are classified as broad-band noise.

3.4 Plotting of the hierarchical tree

The graphical representation of the hierarchical tree can be extracted from the report generated by the algorithm of Section 3.2. It utilises an algorithm which starts at the top and systematically searches down the left and then the right branches and sub-branches to parse the structure of the tree. The algorithm is recursive meaning it calls itself over and over again in a nested way until it reaches a leaf of the tree. The end result is a set of x– and y– coordinates tracing the path that joins each individual item on the horizontal axis to the master node at the top of the tree.

Algorithm: Path Search

```
At the current node,
Step 1: Search left if the next node to the left is not done
    find description of the next node to the left
    if the next node to the left is a leaf of the tree
        set label equal to the item number
        mark the path to that leaf as done
        return (back to the next highest level of recursion)
    else if the next node to the left is not done yet
        call Path Search (recursive call)
        build the path by adding the y-coordinate of the
        node to the path (the path starts empty)
    else
        mark the left node to the left as done.
Step 2: Search right if  the next node to the right is not done:
    find the next node to the right
    If the next node to the right is a leaf of the tree
        set label equal to the item number
        mark the path to that leaf as done
        return (back to the next highest level of recursion)
```

else if the next node to the right is not done yet
call Path Search (recursive call)
build the path by adding the y-coordinate of the node to the path (the path starts empty)
else
mark the node to the right as *done*
mark the current node as *done*.

Step 3: Plot paths for each leaf as a stairs plot to construct the tree from the leaf the tree

The result is a set of paths, one for each leaf of the tree. These paths may be plotted as stair plots to construct the tree.

Some nodes in the classification tree have more than two sub-branches, for example Tags 2, 9 and a subcluster including 21, 4, and others are joined by a single horizontal line at 7.6 degrees in the middle of Figure 3. The reason is that the overall maximum distance between items in the cluster does not always grow when a new item is added. Items join growing clusters in turn according to their distance from the nearest item that is already in the cluster. However their inclusion does not necessarily make the overall size of the cluster larger because items already in the cluster may be further apart from each other than they are from the new item.

4. CASE STUDY

4.1 Data sets

The aim of the case study is to use the hierarchical tree derived from spectral PCA to aid and evaluate the maintenance activity. The mean centred time trends of the data set for the case study are shown in Figure 1 and the spectra are in Figure 2. There are two panels for each because the data shown are from before and after a maintenance shutdown. In fact, part of the value of the study is in the presentation of these high density plots. Each represents one day of running and shows all tags. This is not a standard display on an operator's panel. The hierarchical tree aids the detection of tags with similar power spectra and the high density plots allow the engineer to visually confirm the findings.

The scaling used in the time trends is referenced to the *before* case. For instance, time trend 5 in the *before* panel has been scaled to unit standard deviation and the time trend 5 in the after panel is scaled with the same factor. The large deviation in Tag 4 in the *after* panel arises because that time trend moved more than the time trend of Tag 4 before maintenance.

The power spectra in the plots are scaled to the same maximum peak height for visualization purposes. In the spectral PCA computations, however, all spectra are scaled to unit power.

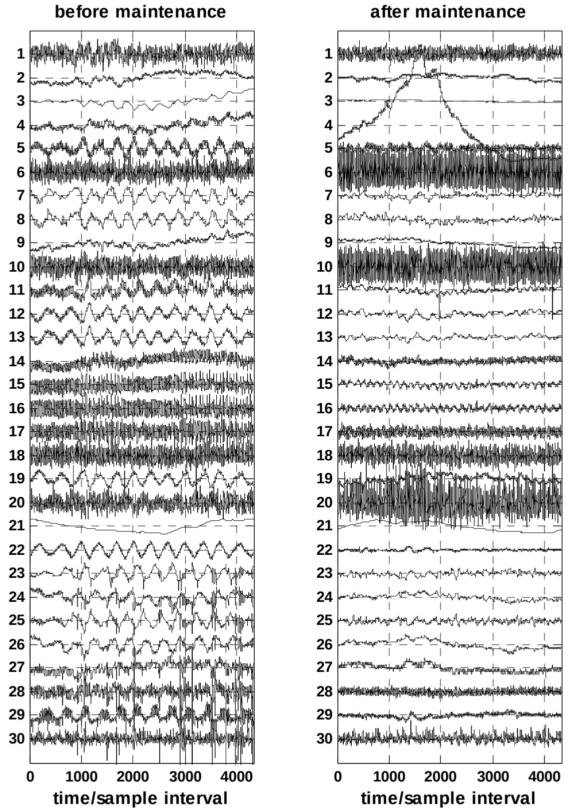

Figure 1. Time trends before and after maintenance

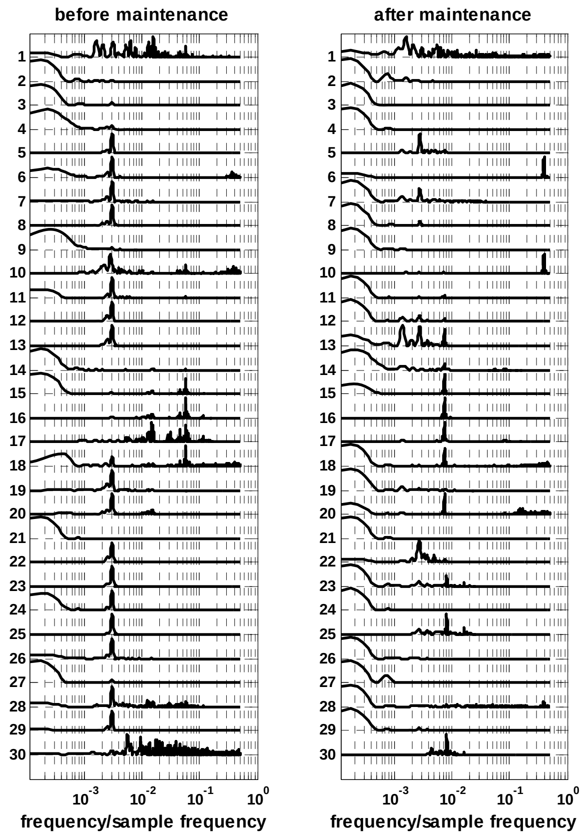

Figure 2. Power spectra before and after maintenance

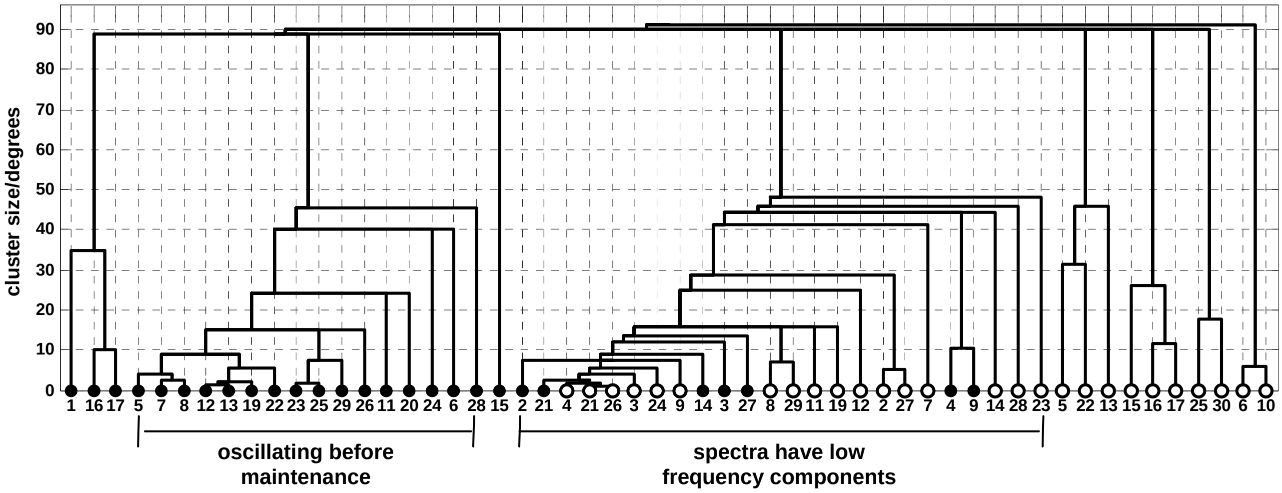

Figure 3. Hierarchical classification tree. Tags with black spots are from before maintenance data set, white spots are after maintenance.

4.2 Results

All 60 spectra (30 from before and 30 from after maintenance) were analysed together using spectral PCA and presented in Figure 3 as a hierarchical classification tree. The analysis needed 11 principal components to capture 99.5% of the variability in the data set. As discussed earlier, 54 of the 60 tags are represented in the tree, the remaining tags have broad band spectra similar to random noise and mapped close to the origin in the Spectral PCA score plot.

In Figure 3, each spot on the horizontal axis represents a complete spectrum from Figure 2. Black spots are the spectra before maintenance and the white are the spectra after maintenance. The numbers below indicate which tag generated the spectrum.

Clusters in the tree share a common branch into the main part of the tree and they represent tags which have similar power spectra and hence similar dynamic features in their time trends. Clusters are clearly visible in the tree; a cluster is a group of tags such as 25 and 30 on the extreme right, or the large group labelled as *spectra have low frequency components*. The *y*-axis shows the cluster size as the maximum angular separation between any two items within the cluster. Some sub-clusters also exist, such as tags 5, 7 and 8 which form a distinct sub-group in the cluster labelled *oscillating before maintenance*.

Here, the clusters have been identified by inspection, however they can also be detected automatically when the length of the branch joining a cluster to the main tree exceeds a pre-set fraction of the cluster size (both measured on the *y*-axis).

Oscillating before maintenance group: A group of tags that were oscillating before maintenance had a strong spectral peak at about 0.0028 on the normalized frequency axis (350 samples per cycle). They are Tags 5, 7, 8, 11-13, 19, and 22-29 which appear as a cluster. The tree shows tags 20 and 6 were also participating in the same oscillation before maintenance. It is not easy to tell that 20 was oscillating from its time trend, but spectral PCA detects the oscillation within the noise.

There are no tags from the after-maintenance data set in this group which demonstrates that maintenance successfully addressed the plant-wide disturbance.

Tags 15, 16 and 17: Tags 16 and 17 are clustered together in the before-maintenance data set, showing their spectra were more similar to each other than to any other spectra in the combined data set. The tree shows that Tag 1 is similar also. Although Tag 1 has spectral content across a broad range it also shares a prominent spectral peak with Tags 16 and 17. After maintenance, tags 16 and 17 lie in a different cluster and are joined by 15 showing that the dynamic behaviour of Tags 16 and 17 was changed by the maintenance activity The spectrum of tag 15 has some low frequency content which is stronger in the before maintenance data set. That is why Tag 15 did not join the {16, 17} cluster before maintenance. Tag 1 from the after maintenance data set is not in the tree because its spectrum was similar to random noise.

Low frequency components group: A cluster in the middle of the tree contains numerous tags both from before and after maintenance. Their spectra have low frequency components in common, and the time trends show that they all have slow drifting non-stationery behaviour. There are more tags in this group after maintenance than before. Some of them such as 7, 8, 11, 12, 19, 24, 27 and 29, migrated into this group once the main oscillating disturbance was removed. Many are indicators and are responding to long term drifts in ambient or operating conditions.

Small clusters: After maintenance there are several small clusters. These are:

- 5, 13 and 22: They have a broad peak at about 3×10^{-3} on the normalized frequency axis. No other tags share this spectral feature. The frequency of this spectral feature is very similar to that of the main oscillation in the before-maintenance data set. The classification tree shows, however, that it is a new frequency because the 5, 13 and 22 cluster form the after-maintenance data set is not connected to the cluster labelled *oscillating before maintenance*.

- 25 and 30: They have some spectral content at about 8×10^{-3} on the normalized frequency axis.
- 6 and 10: These tags have a high frequency spectral feature at 4×10^{-1} on the frequency axis. In fact, an inspection of the spectra shows that this feature was present before maintenance but was dominated by the main oscillation at 0.0028 on the frequency axis. It is more prominent after maintenance because the interference of the main oscillation has been removed.

Tags not in the tree: The tags excluded are 10, 18 and 30 in the before maintenance set and 1, 18 and 20 in the after maintenance set. Figure 1 shows that their time trends do not have any distinctive dynamic features, just noise. The benefit of the exclusion of tags with small $\mathbf{t}'$ – vectors from the tree is shown by considering Tags 6 and 10 in the after maintenance data set. They could be mistaken as random by visual inspection, however the spectral analysis shows that they have a distinctive high frequency peak and they therefore appear in the tree.

5. CONCLUSIONS

The paper has presented a hierarchical classification tree as a mean of visualization of the structure within a principal component model of arbitrary dimensions. Each item in the tree represents the power spectrum from one measurement point in the process and the vertical axis is an angle measure that indicates how similar the spectra are to one another. An industrial case study showed that the tree is useful in combination with high density plots of time trends and spectra for interpreting and understanding the impact of the maintenance activity.

6. ACKNOWLEDGMENTS

We are very grateful to John Cox and Michael Paulonis of the Eastman Chemical Company for making the data sets available for the case study. The first author gratefully acknowledges the support of the Royal Academy of Engineering (Global Research Award) and of ABB Corporate Research.

7. REFERENCES

Alam. T.M., and M.K. Alam (2005) Chemometric analysis of NMR spectroscopy data: A review, *Annual Reports on NMR Spectroscopy*, 54, 41-80.

Brodnjak-Voncina, D., D. Dobcnik, M. Novic, and J. Zupan (2002). Chemometrics characterisation of the quality of river water, *Analytica Chimica Acta*, 462,87-100.

Chatfield, C., and A.J. Collins (1980). *Introduction to multivariate analysis*, Chapman and Hall, London, UK.

De Belie, N., V. De Smeldt and J. De Baerdemaeker (2000). Principal component analysis of chewing sounds to detect differences in apple crispness, *Postharvest Biol. Technol.*, 18., 109-119.

Duda, R.O., P.E. Hart and D.G. Stork (2000). *Pattern Classification* (*2nd Edition*), Wiley-Interscience.

Flaten, G.R., R. Belchamber, M. Collins A.D. Walmsley (2005). Caterpillar - an adaptive algorithm for detecting process changes from acoustic emission signals, *Analytica Chimica Acta*, 544, 280-291.

Gordon, A.D., (1987). A review of hierarchical classification, *Journal of the Royal Statistical Society*, 150, 119-137.

Jackson, J.E., and G.S. Mudholkar (1979). Control procedures for residuals associated with principal components analysis, *Technometrics*, 21, 341-349.

Kresta, J.V., J.F. MacGregor and T.E. Marlin (1991). Multivariate statistical monitoring of process operating performance, *Canadian Journal of Chemical Engineering*, 69, 35-47.

Lee, Y-H., K.G. Min, C. Han, K.S. Chang T.H. Choi (2004). Process improvement methodology based on multivariate statistical analysis methods, *Control Engineering Practice*, 12, 945–961.

Malhi. A., and R.X. Gao (2004). PCA-based feature selection scheme for machine defect classification, *IEEE Transactions on Instrumentation and Measurement*, 53, 1517-1525.

Martin, E.B., and A.J. Morris (1996). Non-parametric confidence bounds for process performance monitoring charts, *Journal of Process Control*, 6, 349-358.

Norris, T., P.K. Aldridge and S.S. Sekulic (1997). Determination of end-points for polymorph conversions of crystalline organic compounds using on-line near-infrared spectroscopy, *Analyst*, 122, 549–552.

Ozaki, Y., S. Sasic and H.H. Jiang (2001). How can we unravel complicated near infrared spectra? Recent progress in spectral analysis methods for resolution enhancement and band assignments in the near infrared region, *Journal of Near Infrared Spectroscopy*, 9, 63-95.

Qin S.J. (2003). Statistical process monitoring: basics and beyond, *Journal of Chemometrics* 17, 480-502.

Raich, A., and A. Çinar (1997). Diagnosis of process disturbances by statistical distance and angle measures, *Computers & Chemical Engineering*, 21, 661-673.

Seasholtz, M.B., (1999). Making money with chemometrics, *Chemometrics and Intelligent Laboratory Systems*, 45, 55-64.

Seem, J.E., 2005, Pattern recognition algorithm for determining days of the week with similar energy consumption profiles, *Energy and Buildings*, 37, 127–139.

Thornhill, N.F., S.L. Shah, B. Huang, and A. Vishnubhotla, (2002). Spectral principal component analysis of dynamic process data, *Control Engineering Practice*, 10, 833-846.

Tzeng, D.Y., and R.S. Berns (2005). A review of principal component analysis and its applications to color technology, *Color Research and Application*, 30, 84-98.

Wang, X.Z., S. Medasani, F. Marhoon and H. Albazzaz (2004). Multidimensional visualization of principal component scores for process historical data analysis, *Industrial Engineering and Chemistry Research*, 43, 7036-7048.

Wiedemann, L.S.M., L.A. d'Avila and D.A. Azevedo (2005). Adulteration detection of Brazilian gasoline samples by statistical analysis, *Fuel*, 84, 467–473.

Wise, B.M., and N.B. Gallagher (1996). The process chemometrics approach to process monitoring and fault detection, *Journal of Process Control*, 6, 329-348

Wise, B.M., N.L. Ricker, D.F. Veltkamp and B.R. Kowalski (1990). A theoretical basis for the use of principal components models for monitoring multivariate processes, *Process Control and Quality*, 1, 41-51

Wold, S., K. Esbensen and P. Geladi (1987). Principal Component Analysis, *Chemometrics and Intelligent Laboratory Systems*, 2, 37-52.

Wu, H.D., M. Siegel and P. Khosla (1999). Vehicle sound signature recognition by frequency vector principal component analysis, *IEEE* Transactions on Instrumentation and Measurement, 48 1005-1009

Xia, C. and J. Howell, (2005). Isolating multiple sources of plant-wide oscillations via spectral independent component analysis. *Control Engineering Practice*. **13**, 1027-1035.