Preprint, accepted for MFPT59, Virginia Beach, USA, April 2005.
www.mfpt.org

NON-STATIONARY CONDITION MONITORING WITH THE AEWATT TOOLBOX

Niels Henrik Pontoppidan, Jan Larsen and Sigurdur Sigurdsson

Informatics and Mathematical Modeling, Technical University of Denmark
Richard Petersens Plads 321, 2800 Lyngby, Denmark
Email: {nhp, jl, siggi}@imm.dtu.dk

Abstract: We are developing a specialized toolbox for non-stationary condition monitoring of large 2-stroke diesel engines based on acoustic emission measurements. The main contribution of this toolbox has so far been the utilization of adaptive linear models such as Principal and Independent Component Analysis, as combined modeling and feature reduction methods. These models describe the, say 1024 or 2048, acoustic emission samples per engine revolution, i.e. data are in the crank angle domain. In this framework we have applied unsupervised learning using only one feature – the log-likelihood of an example given the trained linear model. The setup is semi unsupervised, as model parameters are learnt from normal condition data only, thus the system is not directly capable of error identification. However, it should be noticed that the adaptive linear models allow for some diagnosis based on the angular location of residual energy. Also, the framework can be extended, for instance by post modeling of repeated faults. Furthermore, we have investigated the problem of non-stationary condition monitoring when operational changes induce angular timing changes in the observed signals. Our contribution, the inversion of those angular timing changes called "event alignment", has allowed for condition monitoring across operation load settings, successfully enabling a single model to be used with realistic data under varying operational conditions.

Key Words: Component analysis; Condition monitoring; Event alignment; Non-stationarity; Unsupervised learning

Introduction: We are working on condition monitoring with acoustic emission measurements from large 2-stroke diesel engines used for ship propulsion and power generation. The acoustic emission allows for non-intrusive monitoring as the sensors can be attached on the outside of the cylinder. The AEWATT toolbox is developed for the detection of increased friction between piston and liner, a problem that eventually lead to a severe fault: Scuffing. In recent publications we have suggested and analyzed a collection of algorithms capable of non-stationary condition monitoring. In [9] and [10] we outlined the use of adaptive linear models for stationary condition monitoring and in [11] and [12] we added the event alignment that adds the non-stationarity to the system. The data is non-stationary as the engine control optimizes performance by advancing and delaying events, e.g. prolonging fuel injection time when the load increases. In this paper we apply the event alignment algorithm to experimental data, not used for the development of the system, and show that we obtain the same performance using event alignment as if we had handled each load setting with an independent model. One experiment was

conducted by MAN B&W Diesel A/S on their test bed engine in Copenhagen – experimental data used for development of the toolbox was acquired on this engine as well. Additionally we have obtained normal condition data from an in service engine used for power generation by Public Power Corporation on Kos island, Greece.

Data setup and pre-processing: Although it is not directly a part of the AEWATT toolbox we will describe the data acquisition setup as design choices in the toolbox are based on the properties of the acquisition. The engine is equipped with acoustic emission sensors (ultrasonic, 100 kHz – 1 MHz). Further the engine is equipped with tachometer that allows for sampling in the crank angular domain with a resolution of 1024 / 2048 samples per revolution (ppr) depending on the actual system (several systems have been used). Also, the sample-rate is considerably lowered from 2MHz to 20 kHz, by use of analogue root mean squaring, thus the data becomes non-negative.

The crank angle sampling is performed indirectly using two 20 kHz signals containing the *top dead center* and *crank* pulses. The flanks in these two signals indicate when a new cycle begins and when a new crank sample should be calculated. With a running speed of 1-2 Hz and 2048 ppr, the new sample rate does not exceed 4 kHz thus the conversion from time to crank angular domain is also a downsampling. The toolbox default is to re-calculate the RMS; taking the square root of the mean of the squared values between two crank pulses. When the conversion has taken place, each engine cycle is represented as vector of length D with non-negative elements. Stacking N such cycles gives an observation matrix $\mathbf{X}$ that is later used for training of the linear models ($\mathbf{x}$: Dx1, $\mathbf{X}$: DxN).

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_N \end{bmatrix} \qquad (1)$$

As the experiments are very expensive to conduct, we are faced with limited data, so in order to test our models and hypotheses, data resampling is utilized, meaning that $\mathbf{X}$ does not have to be constructed from N consecutive cycles. The data obtained from the test bed engine in Copenhagen has different known load settings – contrary to the data acquired on the Kos engine where such control information is not available. The engine at Kos was monitored by acquiring a few cycles every hour for 9 hours – and in this context we regard those data to be acquired under a stationary normal condition.

Non-stationary data alignment: After data preprocessing we have transformed the data into the crank angle domain, where each pattern is a single engine cycle, showing different events occurring. These patterns have usually high dimensionality, e.g. the AE RMS signals used for the experiments have 1024 and 2048 dimensions, depending on the angle encoder resolution. Under different running conditions with engine load changes, these patterns become highly non-stationary as both the timing and amplitude of different events changes dramatically. This makes it impossible to directly compare the patterns in the crank angle domain. Thus, the patterns need to be alignment prior to feature extraction and detection. Figure 1 illustrates the event time changes during an injection period of a diesel engine. AE RMS signals at 25%, 60% and 90% are shown. The points indicate the landmarks, indicating the time positions that should be aligned. Note that the individual events are in the same order, regardless of load.

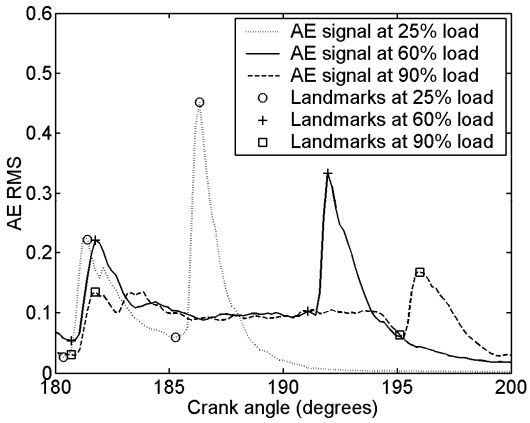

Figure 1: The AE RMS signals during the injection period with different load setting. The markers are the landmarks, indicating the time positions that should be aligned.

Automatically aligning the events of the patterns by means of, e.g., dynamic time warping, have shown poor results, as the patterns are very complex. Instead, we have relied on manually constructing landmarks from the data and used spline-interpolation to align the events [11][12]. All patterns are aligned to a selected reference patterns. Currently, the AEWATT toolbox implements first (piece-wise linear) and third order (cubic) spline-interpolation for event alignment. The left panel in Figure 2 illustrates the event alignment of a single AE RMS pattern using the piece-wise linear alignment.

On top of the event alignment, an amplitude alignment should take place. A scaling of the amplitude of the data has been shown to work well. The scaling is the ratio between the reference pattern and the average pattern at a constant load. The reference pattern is the average of the AE RMS patterns at 25% load. The right panel in Figure 2 shows the results after amplitude and event alignment.

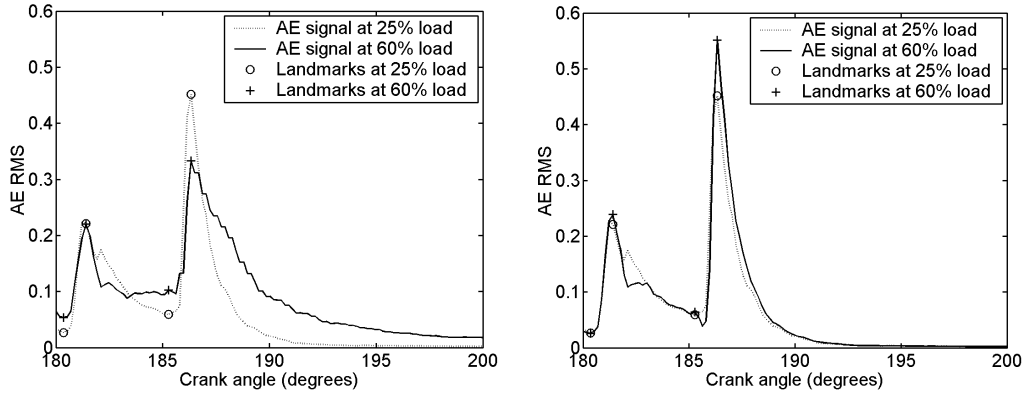

Figure 2: The left panel shows the AE signals at 25% and 60% load from Figure 1 after event aligning the signal at 60% load with the signal at 25% load, using a piece-wise linear spline-interpolation. The right panel shows the results after event and amplitude alignment.

Feature extraction: Feature extraction aims at extracting relevant information from the measured/preprocessed data. This is extremely important when the size of the measured data is large. The patterns of the AE data considered here have 1024 and 2048 dimensions and condition monitoring of such large and complex signals is very difficult. By extracting the relevant information, the dimension may be reduced by orders of magnitude or even to a single feature.

Simple single feature can easily be extracted from the patterns, e.g., empirical average or maximum value. The problem with these types of features is that they do not take into account more general changes in the patterns, e.g., changes that do not influence the average or maximum value. For instance, a fault that would cause a time shift in the patterns will not be detected with these simple features. It is important that feature extraction methods detect such changes, to be able to cover a wider range of engine faults.

An important property of a feature extraction method is to be able to learn the difference between normal and faulty conditions using only normal patterns. It is extremely time consuming and expensive to induce all possible faults in an engine to obtain faulty measurements. Feature extraction methods that only learn from normal data may be considered as *semi unsupervised learning* models and have great advantage compared to models applying *supervised learning*.

In the AEWATT toolbox we have focused on feature extraction based on linear transformation or components analysis of the measured patterns, using only normal patterns. We assume that each pattern $\mathbf{x}$ with size $D \times 1$ is generated from the noise model $\mathbf{x} = \mathbf{A}\mathbf{s} + \varepsilon$, where $\mathbf{s}$ is a $K \times 1$ source signal, $\mathbf{A}$ is a $D \times K$ mixing matrix and ε is a $D \times 1$ additive noise variable. Further, we assume that the dimension of the source signal is much lower than the measured data $\mathbf{x}$, i.e. $K << D$. A common way of extracting features is to estimate the source signal $\mathbf{s}$, thus reducing the dimensionality from D to K dimensions.

There exist a number of methods for estimating the system $\mathbf{x} = \mathbf{A}\mathbf{s} + \varepsilon$. Here we will mention two data adaptive methods in the AEWATT toolbox, *principal components analysis* (PCA) and *independent components analysis* (ICA). Both models can be formulated in a probabilistic way, which opens for the possibility to apply a very effective way of extracting features, by using the so-called *likelihood* function. The negative log-likelihood values for patterns that are similar to the training set patterns, which are normal, will have lower values compared to patterns that are different, e.g. faulty patterns.

The main goal of PCA is to retain as much variance of the original data as possible. Moreover, the columns of $\mathbf{A}$ are constrained to be orthogonal. This may be done by applying singular value decomposition, given by $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, where $\mathbf{X}$ is a $D \times N$ matrix of N measured patterns, $\mathbf{U}$ is a $D \times N$ orthonormal matrix, $\mathbf{V}$ is a $N \times N$ orthonormal matrix and $\mathbf{D}$ is an $N \times N$ diagonal matrix of singular values, where the elements, in

ascending order, corresponding to the standard deviation of the data. The matrix $\mathbf{A}$ is then estimated by retaining the first K columns of $\mathbf{U}$. Previously, the PCA has been modeled in a probabilistic framework [4], by assuming $\mathbf{x}$ to be a multivariate Gaussian variable with ε as isotropic Gaussian noise. The left panel of Figure 3 illustrates the use of PCA for feature extraction with two dimensional toy patterns.

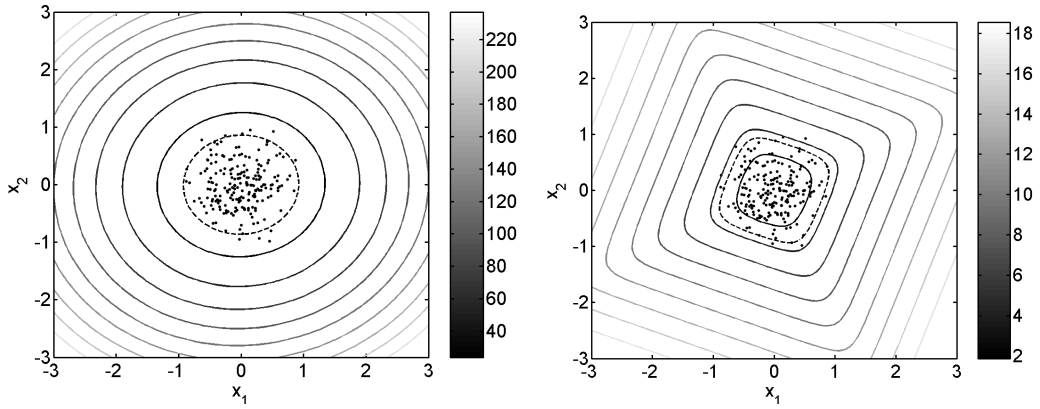

Figure 3: The left panel illustrates the negative log-likelihood results in 2-D input space for the PCA and the right panel for the ICA. The dots are the measured normal patterns, the solid lines are contours of the negative log-likelihood surface and the dashed line is a threshold found by computing the 5% fractal of the normal patterns. Measured patterns that lie outside the area marked with the dashed lines would be detected as faulty. Note the difference between the negative log-likelihood contours of these two methods. The PCA is optimal if the patterns are Gaussian distributed, while the ICA is better at arbitrary distributions.

The ICA method has recently gained popularity in data analysis. The method assumes that the source signals $\mathbf{s}$ are statistically independent. ICA was introduced as information maximization [1] and separation [8], and has recently been extended in a Bayesian framework [5] that allows for specification of certain prior assumptions. For instance, the AE RMS signals are positive and may be considered as positive addition of positive sources, which constrains the elements of $\mathbf{A}$ to be non-negative. This is possible to obtain with the Mean Field ICA algorithm developed at DTU [5][7], which is incorporated into the AEWATT toolbox. As with the PCA, the likelihood function of the ICA acts as a feature extraction for the data. The right panel of Figure 3 illustrates the use of ICA for feature extraction with two dimensional toy patterns.

For both PCA and ICA we need to estimate the number of components K. With too few components the underlying structure of the data is not captured, while too many components will result in a noisy estimate. An optimal number of components keeps the most important components, while disregarding the less important and noisy components. Selecting the optimal number of components is not trivial and many different methods exist. In the toolbox we have focused on the Bayesian Information criterion [4] and well established partitioning schemes, e.g. cross-validation [3], [13].

Note that it is possible to consider both the PCA and ICA as dimension reduction methods. By estimating the source signals $\mathbf{s}$, the dimension of the data is reduced from D to K, thus obtaining K features. Further modeling may then be done with, e.g., density models like the Gaussian mixture model, which is also included in the toolbox. Although we have not experienced increased performance applying this scheme to AE signals, it may be beneficial for other applications.

Classification: We have adopted a semi unsupervised, single feature, fault detection based on single cycle examples. Its semi unsupervised since parameters are learned from normal condition data only. The combined feature extraction and modeling schemes are trained using a subset of the known normal examples. From another subset of known normal examples we build the histogram and cumulated density function for our selected feature: the *negative log-likelihood* (NLL). Thus no known faulty examples are used during model training and threshold estimation. As we expect that normal examples have lower NLL-value than faulty, so the classification boundary becomes a maximal acceptable NLL-value, e.g. a rejection level. A very similar approach was successfully applied to (truly) unsupervised classification of emails by Szymkowiak et al. [13]. We enforce a *tight* boundary by aiming for a rejection rate of normal examples in a test set of say 5%. Unfortunately, this also corresponds to a (design) false alarm rate of 5%. Figure 4 display two NLL-feature time series obtained with data from the Kos engine. For this example 4 of 70 examples in the training set and 5 of 78 in the test set were rejected. The cumulated density functions in Figure 5 show that it is impossible to select a threshold that separates the training and test set; this is good as both sets contain normal condition data. This also shows how the rejection rate is converted to a false alarm rate. One well known way of reducing the false alarm rate is through multiple independent classifiers combined with majority vote system.

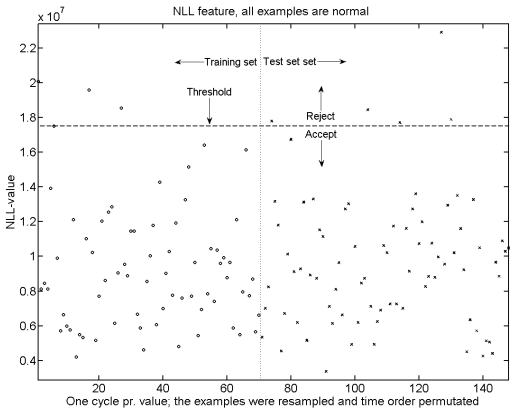

Figure 4: Negative Log-likelihood for training and test set of known normal examples (2 components MFICA)

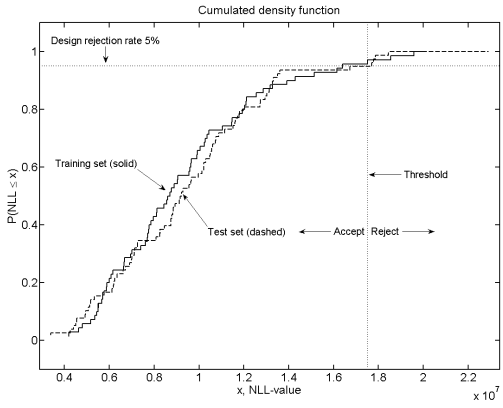

Figure 5: Empirical cumulated density function of NLL values in Figure 4. The two CDF's are very similar.

Binomial hypothesis test: A 5% false alarm rate is way too high for condition monitoring of large diesel engines. Even with a 0% rejection level on an independent validation set, we could still encounter false alarms since the model and threshold is learned from only a

subset of data. The intuitive way to deal with the false alarms is to monitor the rejection rate and only react when the rate of alarms gets high enough. This corresponds to *binomial hypothesis testing* [1]. We can use a binomial hypothesis test to address the inherent false alarms, as well as reduce the false alarm rate by considering a number of successive engine cycles as a whole. We treat the normal/faulty classification of each engine cycle as an independent binomial experiment (normal=0, faulty=1). If the observed binomial sequence can be accepted, with a given confidence level, as being drawn from a binomial process with hit rate equal to the set rejection rate, the classification of the set as a whole is *normal*.

From an engine producing power at Kos we have acquired a few cycles every hour under assumed normal conditions. PCA and MFICA models were trained on a subset of examples (not in time order) and the target rejection rate was 5%. The critical value for the binomial hypothesis test was calculated with confidence level of 0.01 on 78 normal examples to be 10 [2]. 80 such resampled data sets (with 78 examples) where accepted as being normal regardless of model and number of components. The binomial cumulated density function for B(5%,78) is shown in Figure 6. To the right in Figure 7 two binomial sequences generated from the test bed engine at MAN B&W in Copenhagen show that the rejection rate under faulty conditions is much larger than under normal conditions. Actually, they were close to 1 and therefore would this sequence as a whole be classified as faulty.

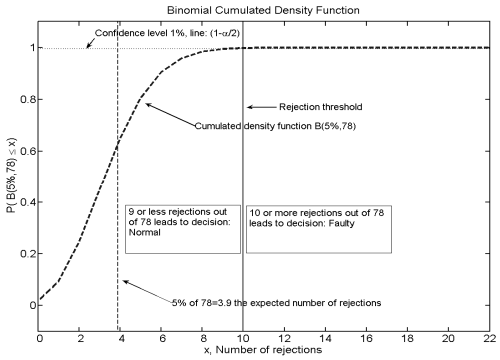

Figure 6: Binomial hypothesis testing with Kos engine data. Target rejection rate 5%, confidence level 1% on 78 examples give new threshold 10 (10/78=13%)

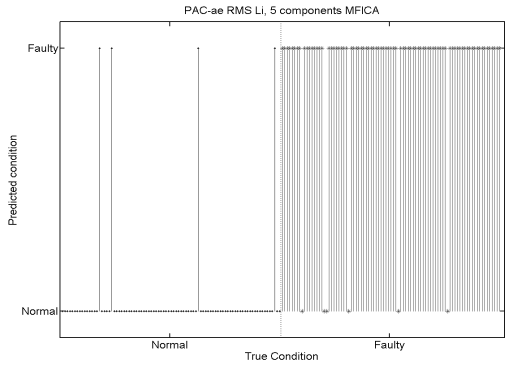

Figure 7: Binomial sequence using MFICA with 5 components on data from Copenhagen test bed. The number of rejected examples rises at the condition change (NB examples not in time order).

Alignment validation: The event alignment is validated using the area under the receiver operation curve (AUC) performance metric. By using the AUC it is possible to evaluate the quality of the feature extraction without making any assumptions on the classification system. The following section will look into the performance of the classification. The AUC may be considered as the probability that the feature value of a faulty pattern is higher then the value of a normal pattern. This gives the highest AUC value as 1, indicating that the two classes may be completely separated using correct threshold for classification. A feature that does not discriminate between classes has the

AUC value of 0.5, indicating that the feature values for the classes are completely overlapping and give random results.

The data used to validate the event alignment is composed of AE RMS signals where scuffing is induced at three different loads, obtained from the cylinder liner of the electronically controlled 2-stroke engine at MAN B&W Research Copenhagen. The loads are at 25%, 60% and 90%. Due to load difference, the data is highly non-stationary. The data is preprocessed by converting the AE RMS signal to angle domain, giving patterns having 2048 dimensions. We considered three methods for feature extraction; the empirical average (MEAN), PCA and MFICA. Note that MEAN is very good feature for detecting scuffing, while has shown poor performance at detecting other faults. We apply 4 different preprocessing schemes; (1) weighted average of models trained on data at each load, (2) models trained on all data without alignment, (3) models trained on all data with event alignment, (4) models trained on all data with event and amplitude alignment. Note that preprocessing (1) corresponds to a stationary condition at each load. The results are shown in Table 1.

	(1) Average of stationary	**(2) No event alignment**	**(3) Event alignment**	**(4) Event/amplitude alignment**
MEAN	0.977	0.709	0.714	0.980
PCA	0.966	0.894	0.946	0.957
MFICA	0.954	0.899	0.940	0.947

Table 1: The AUC for three methods; empirical average (MEAN), PCA and MFICA, using different data preprocessing; (1) weighted average of models trained on data at each load, (2) trained on all data without alignment, (3) trained on all data with event alignment, (4) trained on all data with event and amplitude alignment. The results show that it is possible to obtain the similar performance using event/amplitude alignment as training models at each load.

The results show that by applying event/amplitude alignment it is possible to obtain similar performance as training models at each load. Without event alignment the performance of all methods decreases significantly. Using only event alignment improves the performance of PCA and MFICA dramatically, obtaining almost the same performance as stationary modeling. This shows that event alignment is the most important alignment for the advanced methods. On the other hand, event and amplitude alignment is necessary to improve the performance of simple methods on non-stationary data, while the improvement is marginal for PCA and ICA.

Summary: We have demonstrated some key components for non-stationary condition monitoring with the AEWATT toolbox and showed how these components can be utilized to decrease the number of false alarms significantly. Especially, the results obtained with event alignment are promising, as we cannot learn PCA and MFICA parameters for all possible load settings. Furthermore, we are currently investigating interpolation between known load models within the event alignment framework.

Acknowledgements: This work is supported by EU Competitive and Sustainable Growth Programme GRD2-2001-50014 – acronym AE-WATT (http://isp.imm.dtu.dk/aewatt).
The experimental data was provided by AEWATT project partners: MAN B&W Diesel A/S (Denmark), Heriot-Watt University (Scotland, UK.), Envirocoustics (Greece) and Public Power Generation (Greece).

References:

[1] Bell, A. and Sejnowski, T., *An information-maximisation approach to blind separation and blind deconvolution*, Neural Computation, Volume 7, number 6, pp. 1129-1159, 1995
[2] Conradsen, K., *En introduktion til statistik*, IMM-DTU, 6. edition, 1995
[3] Efron, B. and Tibshirani, R.J., *An Introduction to the Bootstrap*, Chapman & Hall, Monographs on Statistics and Applied Probability, 1993
[4] Hansen, L.K. and Larsen, J., *Unsupervised Learning and Generalization*, Proceedings of the 1996 IEEE International Conference on Neural Networks, vol.1, pp. 25-30, 1996
[5] Hansen, L.K., Larsen, J., and Kolenda, T., *Blind detection of independent dynamic components*, In Proceedings of ICASSP'2001, vol. 5, pp. 3197-3200, 2001
[6] Højen-Sørensen, P.A.d.F.R. and Winther, O. and Hansen, L.K., *Mean Field Approaches to Independent Component Analysis*, Neural Computation, 2001
[7] Kolenda, T., Sigurdsson, S., Winther, O., Hansen, L.K. and Larsen, J., *DTU:Toolbox*, Internet: http://mole.imm.dtu.dk/toolbox/, IMM-DTU, 2002
[8] Molgedey, L. and Schuster, H.G., *Separation of a mixture of independent signals using time delayed correlations*, Physical Review Letters, 1994
[9] Pontoppidan, N.H., Larsen, J., Fog, T., *Independent component analysis for detection of condition changes in large diesels*, COMADEM 2003, no. 16, pp. 493-502, 2003
[10] Pontoppidan, N.H. and Larsen, J., *Unsupervised Condition Change Detection In Large Diesel Engines*, 2003 IEEE Workshop on Neural Networks for Signal Processing, pp. 565-574, 2003
[11] Pontoppidan, N.H. and Douglas, R., *Event alignment, warping between running speeds*, COMADEM 2004, no. 17, pp. 621-628, 2004
[12] Pontoppidan, N.H. and Larsen, J., *Non-stationary condition monitoring through event alignment*, IEEE Workshop on Machine Learning for Signal Processing, pp. 499-508, 2004
[13] Ripley, B.D., *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge, 1996
[14] Szymkowiak, A. and Larsen, J. and Hansen, L.K., *Hierarchical Clustering for Datamining*, Proceedings of KES-2001 Fifth International Conference on Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies, pp 261-265, 2001