

# **Condition Monitoring and Management from Acoustic Emissions**

Niels Henrik Pontoppidan

Kongens Lyngby 2005  
IMM-PHD-2005-147

Technical University of Denmark  
Informatics and Mathematical Modelling  
Building 321, DK-2800 Kongens Lyngby, Denmark  
Phone +45 45253351, Fax +45 45882673  
[reception@imm.dtu.dk](mailto:reception@imm.dtu.dk)  
[www.imm.dtu.dk](http://www.imm.dtu.dk)

IMM-PHD: ISSN 0909-3192

# Summary

---

In the following, I will use technical terms without explanation as it gives the freedom to describe the project in a shorter form for those who already know.

The thesis is about condition monitoring of large diesel engines from acoustic emission signals. The experiments have been focused on a specific and severe fault called scuffing. The fault is generally assumed to arise from increased interaction between the piston and liner. For generating experimental data destructive tests with no lubrication, oil has been carried out. Focus has been on modeling the normal condition and detecting the increased interaction due to the lack of lubrication as a deviation from the normal.

Linear instantaneous blind source separation is capable of picking out the relevant hidden signals. Those hidden signals and the estimated noise level can be used to model the normal-condition, and faults can be detected as outliers in that model. Among the investigated methods the *Mean field independent component analysis* with diagonal noise covariance matrix models is best at modeling the observed signals. Nevertheless, this does not imply that this is the best model to detect the outliers.

Another contribution of this work is the analysis of the angular position changes of the engine related events such as fuel injection and valve openings, caused by operational load changes. With inspiration from speech recognition and voice effects the angular timing changes have been inverted with the *event alignment* framework. With the event alignment framework it is shown that non-stationary condition monitoring can be achieved.



# Resumé

---

Emnet i denne PhD afhandling er tilstandsovervågning ved brug af ultralyd i store diesel motorer, der bruges til skibe. Målet har været at kunne detektere en specifik og alvorlig fejl kaldet: Scuffing. Idet menes at fejlen opstår ved kontakt mellem cylindervæggen og stemplet er følgende eksperiment udført: Ved afbrydelse af smøreolien til cylinderen er det forsøgt at fremprovokere Scuffing. Efterfølgende er det forsøgt, at lade algoritmer trænet på det normale lydbillede, at detektere det ændrede lydbillede som følge af den manglende smøring.

Lineær instantan blind signal separation kan finde de relevante skjulte signaler. Disse skjulte signaler kan bruges til at modellere normaltstanden sammen med det estimerede støj niveau. Fejl kan følgelig detekteres som afvigere fra denne model. Blandt de undersøgte metoder er *Mean field independent components analysis*, med diagonal støj kovarians matrice, den bedste til at modellere de observerede signaler. Men det vises også at det ikke nødvendigvis medfører at dette er den bedste metode til fejl-detektion.

Vinkelforskydninger i motorens lydbillede, eksempelvis indsprøjtning og ventil åbning, forårsaget af de operationelle tilstandsændringer er blevet analyseret og modelleret med signal behandling inspireret af tale genkendelse og lydeffekter til musik. Denne metode kaldet *event alignment* muliggør tilstandsovervågning med skiftende operationelle tilstande, dvs. under skiftende belastninger.



# Preface

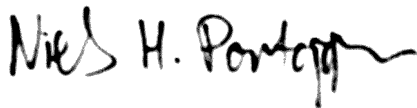
---

This thesis was prepared at Informatics Mathematical Modelling, the Technical University of Denmark in partial fulfillment of the requirements for acquiring the Ph.D. degree in engineering.

The thesis deals with various aspects of mathematical modeling of the engine condition with acoustic emission signals. The two main topics are application of generative linear models for condition monitoring and non-stationarity through event alignment. It is based on the topics and research in relation to the enclosed research papers written during the period 2002–2005, and elsewhere published.

The thesis was defended on October 6, 2005 at DTU. The review committee consisted of: Professor Lars Kai Hansen, DTU (Chairman), Professor Fredrik Gustafsson, Linköping University, Sweden, and Dr. John Alexander Steel, Heriot-Watt University, Edinburgh, Scotland. The Ph.D. degree in engineering was awarded on November 18, 2005.

Lyngby, January 2006

A handwritten signature in black ink, reading "Niels H. Pontoppidan". The signature is written in a cursive, flowing style with a long horizontal stroke at the end.

Niels Henrik Pontoppidan





# Enclosed research papers

---

## Journal papers

[[Appendix A](#)] N. H. Pontoppidan and S. Sigurdsson. Independent components in acoustic emission energy signals from large diesel engines. *Submitted to International Journal of COMADEM*, 2005. URL <http://www2.imm.dtu.dk/pubdb/p.php?id=3885>

[[Appendix B](#)] N. H. Pontoppidan, S. Sigurdsson, and J. Larsen. Condition monitoring with mean field independent components analysis. *Mechanical Systems and Signal Processing*, 19(6):1337–1347, nov 2005b. URL <http://dx.doi.org/10.1016/j.ymssp.2005.07.005>. Special Issue: Blind Source Separation

## Conference papers

[[Appendix C](#)] N. H. Pontoppidan, J. Larsen, and S. Sigurdsson. Non-stationary condition monitoring of large diesel engines with the AEWATT toolbox. In [Pusey et al. \[2005\]](#). URL <http://www.imm.dtu.dk/pubdb/p.php?3351>. In *Proceedings of MFPT59*.

[[Appendix D](#)] Runar Unnthorsson, Niels Henrik Pontoppidan, and Magnus Thor Jonsson. Extracting information from conventional AE features for fatigue onset damage detection in carbon fiber composites. In [Pusey et al. \[2005\]](#). URL <http://www.imm.dtu.dk/pubdb/p.php?3289>. In *Proceedings of MFPT59*.

- [**Appendix E**] N. H. Pontoppidan and J. Larsen. Non-stationary condition monitoring through event alignment. In *IEEE Workshop on Machine Learning for Signal Processing*, pages 499–508, Piscataway, New Jersey, September 2004. IEEE Press. URL <http://isp.imm.dtu.dk/mlsp2004>
- [**Appendix F**] Niels Henrik Pontoppidan and Ryan Douglas. Event alignment, warping between running speeds. In [Rao et al. \[2004\]](#), pages 621–628. ISBN 0-954 1307-1-5. URL <http://www.imm.dtu.dk/pubdb/p.php?3111>. In *Proceedings of COMADEM 2004*.
- [**Appendix G**] N. H. Pontoppidan and J. Larsen. Unsupervised condition change detection in large diesel engines. In C. Molina, T. Adali, J. Larsen, M. Van Hulle, S. Douglas, and Jean Rouat, editors, *2003 IEEE Workshop on Neural Networks for Signal Processing*, pages 565–574, Piscataway, New Jersey, September 2003. IEEE Press. URL <http://isp.imm.dtu.dk/nsp2003>
- [**Appendix H**] N. H. Pontoppidan, J. Larsen, and T. Fog. Independent component analysis for detection of condition changes in large diesels. In [Shrivastav et al. \[2003\]](#). ISBN 91-7636-376-7. URL <http://www.imm.dtu.dk/pubdb/p.php?2400>. In *Proceedings of COMADEM 2003*.

## Various material (not enclosed)

- N.H. Pontoppidan, S. Sigurdsson, and J. Larsen. AEWATTtoolbox for MATLAB. only available through licensing, 2005c
- AEWATT Project Consortium. Deliverable 10, Detection and decision making methods for automated condition monitoring and management of machines. Technical report, July 2005
- AEWATT Project Consortium. Mid Term Assessment Report. Technical report, June 2004b
- AEWATT Project Consortium. Deliverable 8, Data Acquisition Strategy and Signal Preprocessing. Technical report, January 2004a
- AEWATT Project Consortium. Deliverable 2, AE propagation and signal/event correlation. Technical report, 2003a

# Acknowledgements

---

First, I wish to thank my supervisor Jan for the opportunity, collaboration and belief. Secondly, I would also like to thank my co-supervisor Torben at MAN B&W for initiating the AEWATT project. Moreover, I wish to thank the other professors at the signal processing group: Lars and Ole. The guidance and confidence throughout the last 6 years has been highly appreciated. Also thanks to all other members of the group for many lively discussions; Ulla and Mogens for keeping the various parts of the research group in good shape; Tue for keeping a good spirit in B321R123; My old partner in crime Mads; Siggie for collaboration and deeply appreciated proofreading and comments on the thesis.

Also thanks to: the AEWATTT project participants at MAN B&W (Copenhagen), Heriot-Watt University (Edinburgh), Envirocoustics (Athens) and the Greek Public Power Corporation (Athens); Runar Unnthorson at the University of Iceland. I will also take the opportunity to thank the people that I have had valuable discussions with (though especially Mark Goodmann and David Mba) at the COMADEM, MFPT, NNSP and MLSP conferences, as well as the committees organizing those venues.

I wish to thank my whole family especially my parents Ulla & Claus, my brother Peter, and grand mother Märta. Whilst writing this, my thoughts go to my late grandparents: Nils and Inge & Johan Henrik.

A parallel track has been with my wife Christina. During this Ph.D., we have accomplished a lot: Wedding, building a new house, and our lovely children Vibeke and Johan Henrik. This is my true stronghold – tak skat!



# Contents

---

Summary	i
Resumé	iii
Preface	v
Enclosed research papers	vii
Acknowledgements	ix
<b>1 Condition Monitoring and Management from Acoustic Emissions</b>	<b>1</b>
1.1 Setting the stage . . . . .	2
1.1.1 Maintenance strategies . . . . .	2
1.1.2 Monitoring strategies . . . . .	3
1.2 Condition monitoring of large diesel engines . . . . .	5
1.2.1 Modeling and classification . . . . .	5
1.3 This thesis . . . . .	6
1.3.1 System overview . . . . .	7
<b>2 Acquisition and pre-processing</b>	<b>11</b>
2.1 Experimental data . . . . .	11
2.2 Acoustic emission signals . . . . .	12
2.3 Acquisition . . . . .	14
2.4 Crank angle conversion . . . . .	16
2.4.1 Calculating Crank Samples . . . . .	16
<b>3 Event alignment</b>	<b>25</b>
3.1 Time alignment . . . . .	27

3.1.1	Automatic warp paths . . . . .	29
3.1.1.1	Warp path constraints . . . . .	31
3.1.1.2	Limits of dynamic time warping . . . . .	31
3.1.2	Landmarks . . . . .	32
3.1.3	Frequency preserving time stretching . . . . .	34
3.1.4	Spline interpolation . . . . .	35
3.1.4.1	Inverse warp paths . . . . .	35
3.2	Amplitude alignment . . . . .	37
3.3	Modeling the continuous warp functional . . . . .	39
3.4	Downsampling - a crude approach to removing load changes . . . . .	40
<b>4</b>	<b>Condition modeling</b>	<b>43</b>
4.1	Properties: Independent, orthogonal and uncorrelated . . . . .	45
4.2	Mean field independent component analysis . . . . .	47
4.2.1	Priors . . . . .	50
4.2.1.1	Gamma source distribution . . . . .	51
4.2.1.2	Positively constrained Gaussian source distribution . . . . .	51
4.2.2	The transposed problem . . . . .	52
4.2.3	Covariance structure . . . . .	57
4.3	Principal component analysis . . . . .	57
4.3.1	Positive PCA . . . . .	59
4.4	Information maximization independent component analysis . . . . .	60
4.5	Unsupervised Gaussian mixtures . . . . .	61
4.6	Simpler methods . . . . .	62
4.7	Regions of acceptance on synthetic data . . . . .	63
<b>5</b>	<b>Performance measures and model selection</b>	<b>67</b>
5.1	Learning paradigms . . . . .	69
5.2	Generalization error . . . . .	69
5.2.1	Learning curves . . . . .	70
5.3	Penalty methods . . . . .	71
5.3.1	Bayesian information criterion . . . . .	72
5.3.2	Model selection with Ill-posed principal component analysis . . . . .	73
5.4	Supervised classification performance measures . . . . .	76
5.4.1	Receiver operator characteristics . . . . .	76
5.4.2	Threshold optimization . . . . .	78
5.4.2.1	Newman-Pearson criterion . . . . .	79
5.4.2.2	Maximal separation . . . . .	80
5.4.2.3	Minimal distance . . . . .	80
5.4.3	Area under ROC curve . . . . .	81
5.4.4	Learning curves for receiver operator characteristics curve (ROC) statistics . . . . .	81
5.5	Unsupervised classification . . . . .	82

---

5.5.1 Hypothesis testing . . . . .	84
<b>6 Discussion and conclusion</b>	<b>87</b>
<b>A Independent component analysis in large diesel engines</b>	<b>93</b>
<b>B Condition monitoring with Mean field independent components analysis</b>	<b>109</b>
<b>C Non-stationary condition monitoring of large diesel engines with the AEWATT toolbox</b>	<b>123</b>
<b>D Extracting information from conventional AE features for fatigue onset damage detection in carbon fiber Composites</b>	<b>135</b>
<b>E Non-stationary condition monitoring through event alignment</b>	<b>147</b>
<b>F Event alignment, warping between running speeds</b>	<b>159</b>
<b>G Unsupervised condition change detection in large diesel engines</b>	<b>169</b>
<b>H Independent component analysis for detection of condition changes in large diesels</b>	<b>181</b>
<b>I Calculations with Mean field independent component analysis</b>	<b>193</b>
<b>J Abbreviations</b>	<b>195</b>





## CHAPTER 1

# Condition Monitoring and Management from Acoustic Emissions

---

Condition monitoring is a truly multidisciplinary field that grasps much wider than the signal analysis methods investigated in this thesis. Condition monitoring also include strategic, economical, mechanical, as well as social science aspects. This chapter introduces the general condition monitoring task and puts perspective on the expectations to such systems.

The remainder of the thesis more or less follows the information flow of the algorithms, i.e., beginning close to sensors with preprocessing and ending with the condition outputs. However, in order to understand and describe the steps in this chain, everything has to be taken into consideration. E.g. the performance evaluation of preprocessing methods include knowledge on how the signals are processed afterwards.

## 1.1 Setting the stage

Condition monitoring is a well-known task to human beings - we scan the environment for changes continuously, even while performing other tasks. We are alerted by unexpected sounds, movements, and even patterns in our environment. We compare what we observe to the knowledge of how it appeared 2 minutes, hours, or days ago. Abrupt changes such as unexpected sounds from your child, car, bicycle, and CD player alerts you. Traditionally specialized helpers such as dogs barking at approaching people, canaries in coalmines, concealed wires connected to bells have been used. With gradual changes, the best helper is sometimes just a pair of fresh eyes/ears that are not been accustomed to the slow drift. In the mechanical world the condition monitoring task have been performed by the skilled people that operate the machinery on a regular basis. The engine operator will gradually learn how the engine sounds in different operational settings. In addition, we are pursuing this capability with the signal processing and learning framework.

### 1.1.1 Maintenance strategies

A main theme of the Condition Monitoring and Diagnostic Engineering Management (COMADEM) conferences that I attended in 2003 and 2004, was profitability of condition based monitoring. Not all parts and faults are worthy of a condition monitoring system; the simplest example is that you don't need a red light to tell you that a light bulb will break in 5 minutes - at least not at home. However when running an airport, you need a system that monitors the percentage and spatial distribution of broken bulbs in the runway system due to requirements given by the International Air Transport Association. Essentially the necessary monitoring level does not depend on the type of the part, but on the impact of its failure.

Although the management strategies are not a part of this thesis, I will outline my understanding of four such strategies. I will differ between *failure based* maintenance, *scheduled based* maintenance, *condition based* maintenance and *Prescription based Health Management*:

- No maintenance at all. The *use until destroyed* strategy is not that relevant with large diesel engines.
- a With a failure based maintenance strategy the machinery is operating until the fault occurs. Then the fault is fixed and the machinery goes into operation again, e.g., change the bulb and turn the light on again.

- b With a schedule based maintenance strategy the different parts are inspected or even replaced after a specific number running hours, e.g., a bearing is replaced after 3000 running hours even if it looks normal. The replacement times differs from part to part, and is generally set very conservative such that the number of components that do fail within the specified time interval is very small. This strategy is for instance used within aviation industry, ship propulsion, as well as cars: as the oil should be changed every 15.000 km or year whatever comes first.
- c With a condition based maintenance strategy, parts are replaced and repaired when a fault is expected to happen within a near future based on the health of the machine.
- d With a prescription based health management system, the usage pattern and expected usage pattern is taken into consideration when scheduling maintenance. It is as computer war-games, e.g., each unit has an associated health bar, and the commanders (you + computer AI) are considering: Can this mission be fulfilled with that vehicle? Moreover, is it still usable afterwards?

The optimal strategy depends on a wide range of diverse and coupled parameters, where I guess economic and safety issues has the greatest impact when dealing with ship propulsion condition monitoring (CM). The common belief is that increasing the level of CM constitutes an economic improvement, the first by repairing, later preventing many failures with the scheduled replacement of selected parts, that furthermore prevent the additional and often more severe faults caused by the original fault. The third improvement is achieved by improving the availability of the machinery, since maintenance is only scheduled when actually required. However these improvements have an associated cost: Strategy **b** require that the machine is taken out of service on a regular scale, and before that the proper replacement intervals should be determined. Strategy **c** require that the condition can be monitored in a reliable way, which most likely requires sensors, acquisition boards and specialized signal processing as well as knowledge on how the condition evolves. With strategy **d** the current condition and the expected usage are combined to give allow for trade off between risk of failure, success without recovery etc. Compared to the other strategies this also require analysis of wear as function of usage.

### 1.1.2 Monitoring strategies

The core of this thesis is “How to turn the observed acoustic emission energy (AEE) time series into a condition monitoring output”. Among the others we need to select the type of output, again they are order by increased “complexity”

1. Fault alarm, after the fault occurred
2. Late warning, just before it faults
3. Early warning, possibly with some failure time horizon
4. Wear level continuously monitored, more accurate failure time horizon

As with the maintenance strategies, systems that are more sophisticated are required to reach a higher performance level. Obviously, the longer time before the breakdown, the smaller are the deviations we are looking for and thus the harder we must work with features and algorithms find them.

Another issue on whether the monitoring system should model the normal condition and/or specific faults. In this thesis, only the normal condition is modeled and deviations from the normal model are therefore just faults. In a previous EC project Fog used neural networks trained on some specific faults to classify other instances of those specific faults [Fog, 1998]. This relates to the difference between unsupervised and supervised learning. On whether the model is given examples *with or without* associated labels during training. The labels tell the system that this is how the normal ones look like; and this is how the faulty ones looks like - and then we ask what this is? Without the labels the system is not told, but still expected to be discriminate between classes. It is like giving apples and oranges to a child. If you first tell this is an apple, this is an orange the child should get the idea. If you do not tell, the child might just say: fruit! This thesis deals with giving the child apples with no labels, and then expect it to say “not apple” when given an orange. But in this case we actually we don’t know if the child is given both green and red apples to begin with. . .

As experiments conducted within the AEWATT project, have not been reproduced and scuffing has not been encountered, it is unlikely that an accurate failure time horizon statistics can be achieved with those available data sets. Simply our knowledge on how the fault emerges in the engine is not good enough. Further, we cannot be sure if the way the fault is induced is also the way the fault appears outside the laboratory.

In this thesis only the normal condition is considered for modeling, thus faults will just be labeled as faults. It is possible to add new models and slowly build a supervised system. This requires that the relevant data is saved when a fault occur, and that this package forwarded to experts for diagnosing that could lead to new models that diagnose this fault. If storage permits, also scheduled acquisitions could be considered.

## 1.2 Condition monitoring of large diesel engines

At the beginning of the project, it seemed that everybody used the same methods: feature extraction with principal component analysis (PCA) followed by different neural network structures trained in a supervised manner to classify normal condition against a few other conditions. Fog [1998] and Ypma [2001] describe numerous ways to extract features, and how to build and train pattern recognizer's with good generalization. This includes simple regularization schemes, complicated resampling methods as bootstrapping, and adaptive structures. The group at Sheffield University used the traditional cylinder pressure and vibration in similar supervised classification setups [Chandroth and Sharkey, 1999, Chandroth et al., 1999a,b]. Another way to increase generalization is using ensembles. Sharkey et al. [2000] created an ensemble by combining neural networks into a majority voting system. Even though their ensembles are created by random combination, the general idea is to combine precise *and* diverse classifiers, here neural networks in a controlled way. The need for diversity is apparent as additional information is gained from multiple but dependent votes, e.g., they make the same false alarms and do not detect the same defects. The diversity can be obtained by (a) using information from different types of sensors and (b) reusing the data to create neural networks that differ in various ways: bagging, boosting, resampling, etc. Precision can be obtained by applying regularization etc.

Neill et al. [1998] showed that acoustic emission (AE) is superior to pressure- and vibration information wrt. signal to noise ratio, and that the AE is sufficient in a more realistic industrial like setting. This has also been reported by Fog et al. [1999]. The reason is that distance damping of the stress waves increases with frequency, thus with higher frequency AE signals the damping of surrounding noise sources is increased compared to the vibration signal.

### 1.2.1 Modeling and classification

Previous experiments using artificial neural networks (ANN) for condition monitoring of large diesel engines by Chandroth et al. [1999a,b], Fog et al. [1999], Neill et al. [1998], and Sharkey et al. [2000] have been *supervised*, i.e., based on training with known labels. An expert produced the labels and/or fault was induced. In addition, the faults were induced several times to achieve good statistics, which is an obstacle when the interesting parts are the liner and piston on large diesel engines.

The observations presented to algorithms during the learning procedure (called

the training set) is a sampling of the true distribution of the machinery's states, hence we are not guaranteed to obtain the full distribution. Further, the observations include a sampling of the noise distribution at the same position.

Consider a quite flexible model trained on relatively few observations with some noise. If the model is able to adapt very well to the training set it is also possible that it has adapted very well to the observed noise. If it also models the noise, then this will not adapt that well to other samples from the same process. This is called overfitting and is normally reduced by constraining the learning process, so that it cannot adapt fully to the training set, either by resampling (bootstrapping), regularization (weight decay), or optimization of architecture (pruning).

Generalization can also be achieved by combining networks in ensembles. [Sharkey et al. \[2000\]](#) have tried to generate ensembles that differ by either, a) combining networks based on different sensors or feature sets. b) Randomizing the initial conditions (useful when you only have a limited number of examples). c) Varying the architecture (e.g. pruning, regularization), d) Exposing the different networks to different examples (resampling). Their conclusions was that the best ensemble consisted of combinations of all of the above, i.e., both different sensors, data resampling and different initial conditions – and by combining the ensembles randomly. Combining the outputs of ensembles was further investigated with a more theoretic setup by [Whitaker and Kuncheva \[2000\]](#) wrt. accuracy and diversity among the ensemble members. Simply ensembles are not bound to work - for instance an ensemble of “football-experts” at our department predicted that France would win the football world cup 2002. Obviously, a strong bias towards Denmark was among the causes leading to increased uncertainty.

## 1.3 This thesis

This thesis investigates some digital signal processing methods for the application of a condition monitoring system aimed at large marine propulsion engines. It deals with the signal processing that allows for upgrading the current restrictive schedule maintenance strategy to a condition based strategy.

Besides, from the benefit of a condition-based strategy, the system described in this thesis could also contribute to decreasing the environmental cost. The additives in the lube oil contribute considerably to the pollution. AEE is generated by friction, so the system considered can and do reveal the friction between piston and liner as seen later in [Figure 4.3](#). The condition output can be used to

only inject the necessary amount of lube oil and thereby reduce the pollution.

Due to my background in speech and sound acoustics the applied methods draw on knowledge and applications developed in that field, e.g., the idea that we can hear and separate the sources and that we learn how an engine should sound like. The framework has two independent contributions to the area of condition monitoring: *a)* the application of blind source separation that incorporates knowledge about the domain of observations, and *b)* the application of event alignment, which is a time stretch method that model known operational changes and allows for non-stationary condition monitoring.

For the application of blind source separation methods, it was initially the idea to put a sensor array on the cylinder. The array ended up being two sensors due to the cost of each sensor, thus changing the scope of blind source separation to separating hidden signals in repetitions instead in spatial distributed channels. A combination would be possible by first separating the hidden sources from the time synchronized channels, followed by separating the “additional” hidden signals from the repetitions. This would require some grouping of the first set of hidden signals, due to inherent permutation of estimated channels. The additional knowledge that is applied in the blind separation problem is the non-negativity of the observation signals. This knowledge can be implemented as a constraint or as a source prior, that ensure that the estimated sources are also non-negative.

The application of time stretch methods is based on the concept of time quantization and time stretching from digital signal processing for music production. Such methods have previously also been applied to align spoken words for speech recognition applications. The idea is that the engine cycle can be considered as a musical measure with a repeated beat: `dnk tssh dnk tssh`. The angular position of events is analogue to the position in the measure. Changing the tempo of a rhythm require proportional scaling of all inter event time differences. Those that do not scale proportional to the tempo change their rhythmical/angular position, and the event alignment framework can compensate for that.

### 1.3.1 System overview

Figure 1.1 outlines a simplified version of the system considered in this thesis. The simple form corresponds to the way new examples would face the system, but omits all the information feedback on model sizes, classification thresholds etc. In general this thesis deals with the three last boxes, as the choice of sensors and signal conditioning was done by project partners.

To prepare for the following chapters, a quick introduction to the 5 boxes in [Figure 1.1](#) follows.

**Sensors, [chapter 2](#)**

The sensors acquire ultrasonic stress waves on the outside of the cylinder liner/cover. These stress waves are generated by micro cracking in the material. The frequency range is 100 kHz - 1.25 MHz

**Signal conditioning, [chapter 2](#)**

The bandwidth of the signals are reduced to 10 kHz by root mean squaring. The new signal is not the actual waveform but the energy envelope signal.

**Preprocessing, [chapter 2](#) and [3](#)**

The signals are transformed from time to crank angle domain. The timing of engine related events can be aligned.

**Modeling, [chapter 4](#)**

A model is trained on preprocessed normal condition examples. The normal and faulty examples separates in a one-dimensional feature when the model is applied to examples. E.g., the model knows the normal pattern and as the faulty pattern gradually emerges, the deviation measure gradually increases.

**Detection, [chapter 5](#)**

The property of the model output should be that the values for normal and faulty examples should be separable. The simplest classifier, which is the one considered here, the detecting the fault is a matter of detecting that the one dimensional measure has crossed the threshold.

The more detailed [Figure 1.2](#) also outline the information flow in the opposite direction. Examples processed by some parameters are propagated through the whole system, and the best set of parameters are selected and sent back to the respective processing blocks (the red arrows). In that figure only two parameters are considered, the size of the models and the classification threshold. The same underlying approach extends to selecting model families, feature extraction methods and even sensors types where decisions based on the whole application are sent back to the blocks where it belong.

It should also be noted that some methods can be used in different ways in different blocks, for instance can [PCA](#) be applied as a feature extraction prior to other models that use the reduced feature subspace as their input. Or [PCA](#) can be used as it is mainly used here as a generative model that describe the observed data.



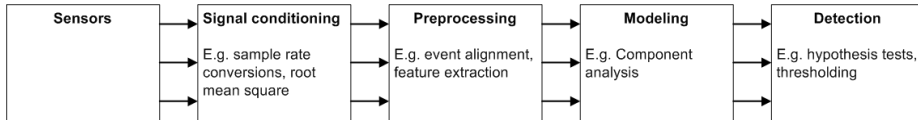


Figure 1.1: The simplified information flow gives a general system overview. Sensors are attached on the monitored specimen, the signals obtained are conditioned before preprocessing, and finally detection based on the output of the modeling. This is only the general structure, as some methods like the component analysis methods can do both preprocessing and modeling at the same time.

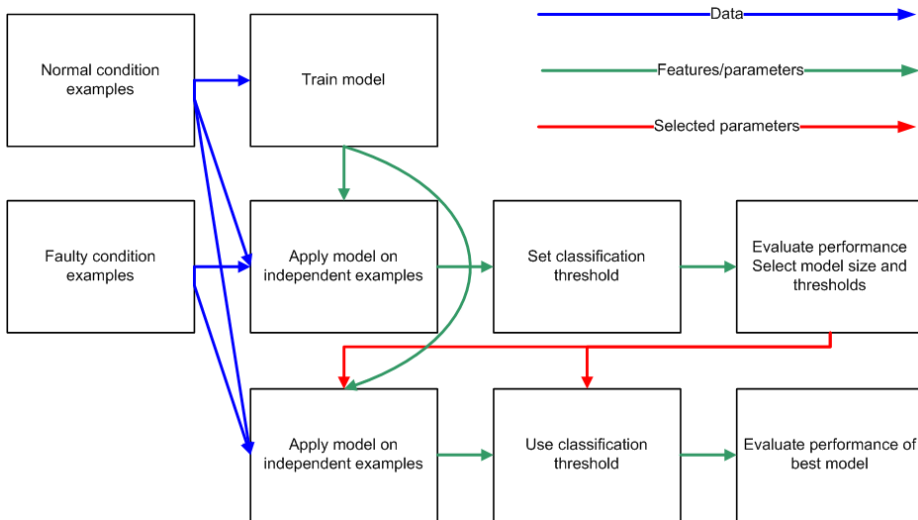


Figure 1.2: A slightly more complicated information flow that show how some parameters are obtained using independent training sets. The three first blocks of Figure 1.1 have been omitted for simplicity. The two example pools contain normal and faulty examples. Two parameters, the size of the model and a classification threshold, in the models are optimized by measuring the performance. The model is only trained on the normal condition. Only the last row, the performance evaluators know the true labels of the examples, the rest of the model only see the observed signals. When it comes to applying the optimized model to new examples we are back to the simpler straightforward system in Figure 1.1.



## CHAPTER 2

# Acquisition and pre-processing

---

This chapter describe the data acquisition setup, and the acoustic emission (AE) signals that are acquired with the system from the engine. Further I outline the preprocessing that makes the AE signals computationally usable as input signals to the condition monitoring system.

## 2.1 Experimental data

The experimental data used for illustrative purposes here is a destructive test due to MAN B&W Diesel A/S. The data set consist of two three load conditions, 25%, 50% and 75%. The fault condition is induced by obstructing the application of lube oil inside the monitored cylinder while the engine was at 25% load. The other cylinders got lube oil during the whole the experiment, and since the cylinders are connected through the bottom oil “bathtub” some oil was sucked up in each cycle with the fresh air.

It was quickly discovered that the operational condition changes was problematic. Initially only the period referred to as Experiment 1 in [Figure 2.1](#) was considered, since only one change occurs in this period - the shutdown of the lube oil system. The period labeled Experiment 2 refers to an additional faulty

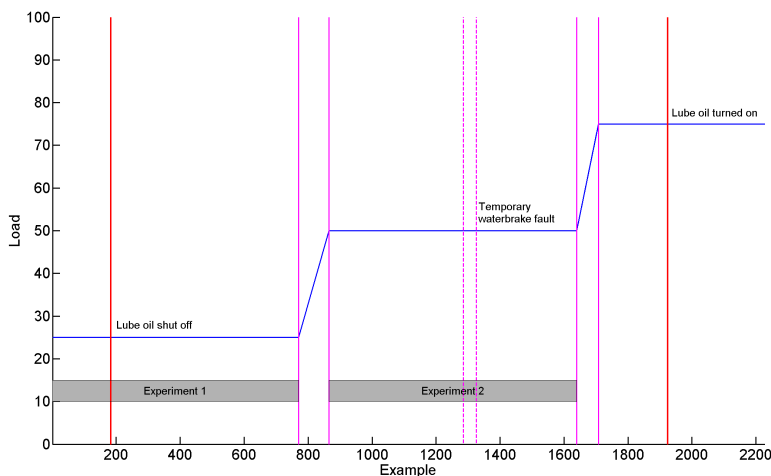


Figure 2.1: Time line of main experiment conducted at MAN B&W Diesel

condition occurs while the lube oil system was turned off. This unintended fault occurred in the attached water brake, which controls the loading of the engine. Luckily, it was temporary during an otherwise stable condition and it revealed interesting properties on model and sensor selection.

## 2.2 Acoustic emission signals

From *ASTM E 610-8* [[appliedinspection.com](http://appliedinspection.com)]

**Acoustic Emission** the class of phenomena whereby transient elastic waves are generated by the rapid release of energy from a localized source or sources within a material, or the transient elastic wave(s) so generated.

**Acoustic Emission** is the recommended term for general use. Other terms that have been used in AE literature include (1) stress wave emission, (2) micro-seismic activity, and (3) emission or acoustic emission with other qualifying modifiers

The AE signals encountered on large diesel engines are mostly stress waves living on the surface of specimens. They are generated due to rupture of internal micro-bindings in the material. In popular words, it is the “oh-no’s” of the internal bindings that we observe. Mechanical events that generate AE are crack

formation, friction, impact. In addition, fluid and gas flows generate **AE**. The two type of sources separate in the frequency domain, such that needle impact and fuel injection flow could be separated in the raw **AE** signals [Douglas et al., 2004].

**AE** has been reported to be superior to vibration data acquired by Neill et al. [1998]. For condition monitoring of large diesel engines the **AE** signals have the nice property that the spatial damping is considerably larger than with vibration data (in the range up to 20 kHz), and thus have a better signal to noise ratio. This also means that the **AE** signals are far more localized, e.g., appearing virtually only on the cylinder where they are generated. However, the damping is also so strong that the distance between the sensor and source should be minimized. In addition, material interfaces along the signal path should be taken into consideration, thus the different damping of the different sources is important when considering the sensor locations. All those considerations was taken into account when the number and position of sensors as was decided by project partners and reported in the “Specification of preliminary sensor array” [AEWATT Project Consortium, 2003b].

Considerably work on condition monitoring has taken place on smaller and simpler structures than the large diesel engine, e.g.: Bearings [Mba, 2005, Neill et al., 1998], pumps [Ypma, 2001], gear boxes [Randall, 1987, Tan et al., 2005], compressors [Elhaj et al., 2003]. Initially condition monitoring (**CM**) was carried out using vibration analysis and then in recent years the use of **AE** has gained attention. The use of **AE** originates from analysis of relations between applied forces and **AE** level for simple structures as beams, rods and cones conducted by Kaiser in the 1950’s. Kaiser also revealed the property that the **AE** remembers the force that was applied to it, since it takes a stronger force to generate **AE** next time, this is called the Kaiser effect [appliedinspection.com]. The increased use of **AE** follows the greater availability of reasonably priced equipment that can handle and capture the very broadband **AE** signals. Less than a decade ago acquisition of **AE** signals for longer periods was problematic [Reuben, 1998]. Further, virtually all theory and knowledge from vibration monitoring can be applied to **AE** since the two signals are caused by the same events, thus many of phenomenon’s that has been used for monitoring also appear in the **AE** but with less noise – and noise has always been the large problem with vibration.

With more sensors and/or a much simpler geometry of the specimen, as opposed to the complex structure of a 3 storage high diesel engine, additional information can be inferred from the **AE** signals. Depending on the dispersive properties of the material the arrival time of high and low frequency components differ as a function of the traveled distance. Further the ratio between high and low frequency peaks reveal the type and location of faults in composites [Dunegan, 2000].

Another well-studied field is order analysis. Based on the geometry of bearings and gearwheels the frequency where a specific fault will appear can be calculated [Randall and Antoni, 2003], moreover, the paper introduces a new way of separating the pseudo stationary parts and the noise. In order to do so, a good estimate of the spectrum has to be acquired by averaging over a few [Randall, 1987], or many cycles depending on the noise type and stationarity of the signals. This has not been considered here, as averaging over say 20 cycles would give 9 examples of the normal condition when only 180 normal examples are available.

As such the AE is very much like *Vibration* and in many cases events generate both, e.g., impacts and rubs both leads to small movements of the structure (vibration) and micro cracking inside the material (AE), and the magnitude of signals are functions of the amplitude of forces and the wear.

## 2.3 Acquisition

The frequency range of the raw AE begin at 100 kHz and goes up. The acquisition system for raw AE signals at MAN operate at 2.5 MHz. For the signal processing techniques considered in this thesis, 2.5 MHz sampling rate is rather high. So the bandwidth is lowered considerably through analogue root mean square (RMS) processing to 20 kHz. That frequency was determined from the maximal capabilities of the combination of data acquisition board and computer at MAN B&W.

The MAN RMS systems has a time constant of  $120\mu s$  corresponding to a cut-off frequency around 8.3 kHz, which is just below than the Nyquist frequency 10 kHz upper frequency limit @ 20 kHz sampling.

RMS pre-processing turns the signals into acoustic emission energy (AEE) signals, containing the energy in the  $120\mu s$  period (overlapping is  $70\mu s$ ). The squaring of the signal corresponds to a convolution in the frequency domain and the lowpass filtering is just an averaging process, thus with the RMS some time resolution as well as frequency information is lost, however the energy is not lost – the resolution is just decreased.

The data rate is not constant throughout a process. In speech and music the information rate is lower than the necessary sampling rate, e.g., my word rate is not 4 kHz, but since I want to communicate different messages bandwidth are used to code such messages. In addition, after transmission the decoding reduces the data stream to a sequence of that word at that time. Is this important -

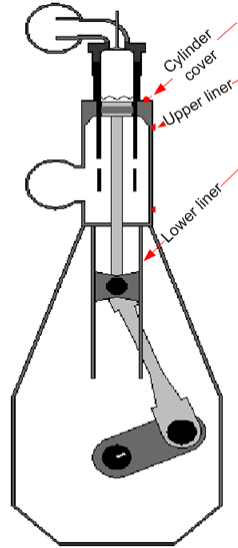


Figure 2.2: Sensor positions: Cylinder cover, upper liner, lower (on the upper liner). Engine sketch due to Ryan Douglas, [HWU](#).

yes because for the condition monitoring process it might be sufficient with the “word-rate” rather than the full sampling rate and what we loose is the ability to discriminate between different messages. If the frequency is used to, code messages filter banks and pattern recognition might be applied as a decoder step. Automatic music transcription is an attempt at such decoding, and a more successful example is codebooks used with auto regressive processes estimation and transmission in GSM telecommunication.

The link to the diesel engine signals is obvious. The AE stress waves appear in the ultrasound domain; however the process that provokes the micro cracking is related to the rotational speed of the engine which is 4-6 decades below. Therefore, we seek information in a scale that reveals the individual engine related events but not the individual cracks.

Within the AE-WATT project, the sensors have been placed on the cylinder cover, high and low on the upper part of the liner. The sensors can be placed on the cylinder cover and the upper part as [Figure 2.2](#) show.

## 2.4 Crank angle conversion

Chandroth and Sharkey [1999] state that all engine cycles regardless of running speed have the same number of samples in the crank angle domain. What they do not discuss is how to convert the signals from the time into the crank angle domain. Sampling in the crank angle domain corresponds to sampling with a constant angle displacement, e.g., at  $1^\circ$ ,  $2^\circ$  and so on. In order to drive such sampling a trigger signal is necessary – which can be generated using a light source, photo-resistor, and “checkerboard tape” on the circumference of a flywheel connected to the crankshaft. Such a system is sketched [Figure 2.4](#).

In data sets available to me, all signals (including Top dead center, crank pulse and AEE) have been sampled synchronously, i.e., we have the pulse signals on the same timescale as the AEE signals as shown in [Figure 2.5](#). The conversion to the crank domain, is a matter of detecting the flanks in the crank pulse square wave and detecting the beginning of each cycle from the Top Dead Center signal. The remaining question is how to calculate the new samples. In our case, we have had 1024 and 2048 crank angle pulses pr. revolution. With a rotational speed of 60-120 revolutions per minute (rpm) it corresponds to varying the sample rate (in time domain) between 1024 and 4096 Hz, somewhat lower than the original 20 kHz. Therefore, the conversion is also a downsampling process. The top dead center (TDC) marker is aligned for one particular cylinder, so for the remaining cylinders the TDC the phase shift should be calculated depending on number of cylinders and fire sequence. The fuel injection takes place around TDC(both before and after) so splitting the signals into individual cycles is not convenient, instead the splitting takes place at bottom dead center (BDC) ( $180^\circ$  out of phase) where less activity takes place. This means that each *engine cycle example* consist of first the blow out of exhaust gases followed by injection, combustion and expansion phase. This is also how [HWU](#) split the signals. The path from the engine to the observation matrix is depicted in [Figure 2.3](#).

### 2.4.1 Calculating Crank Samples

During the work with the event alignment, I became suspicious when the peak amplitude of the combustion peaks dropped as a function of the load. Intuitively it should rise as most engines emit more noise when the loading increases. I approached the mechanical engineers at [HWU](#) with this question and their answer was *does the total energy in the injection period also drop?* This turned my attention towards the problem of calculating the crank samples properly. An analogue [RMS](#) module processed signals between sensors and acquisition system, thus the signals were already non-negative – which in the following discussion



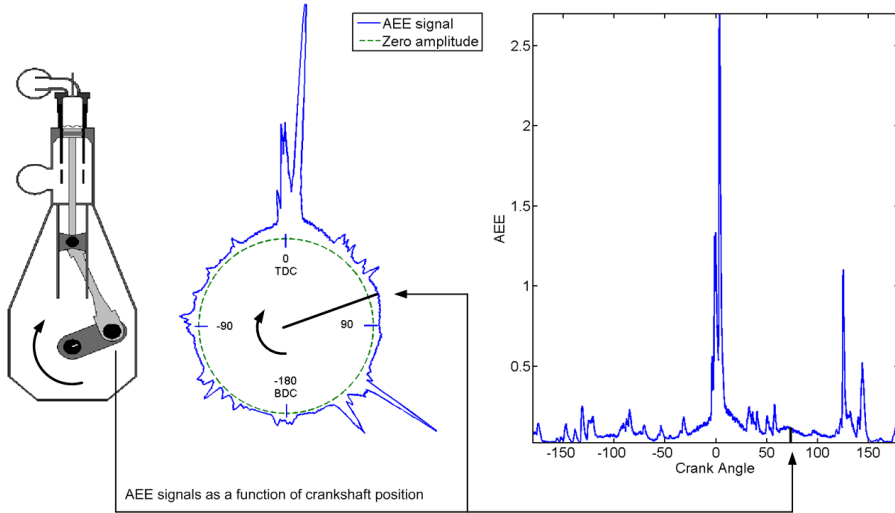
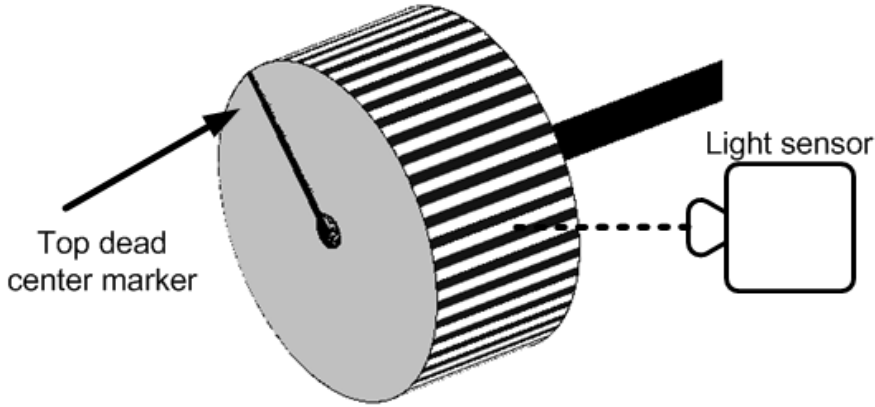


Figure 2.3: Simplified crank angle domain setup. The sampling of the data is synchronized with crankshaft position and the AE signal can be shown as in the middle radar like plot. Each cycle going from  $-180^\circ$  to  $180^\circ$  is considered a single example with 1024/2048 features depending on the resolution of the angle encoder.

is a key property. Even though it turned out that the initial approach based on a re-computation of the RMS (called RMS for the remainder of this section) altered the energy ranking of the signals (as a function of load), it did not influence the performance our experiments since each load was processed independently. Further when several loads was considered, amplitude mismatches was taken care of by the event alignment procedure (section 3.2). However, such nice recoveries should not prevent us from doing it right.

The question is whether the conversion is a domain transformation or a resampling process. Initially I believed the second. The difference between the two approaches is displayed in Figure 2.6 that show that the RSS changes the amplitude of the signals (due to the compression of the domain). Let us look at the simple math.  $x[n]$  is the AEE signal in the time-domain. The signal  $CRK[c]$  holds the indices of the rising (or falling) edges in the crank pulse signal, i.e.  $CRK[1] = 3$  tells that the first crank pulse goes high for  $n = 3$  as seen in Figure 2.5. The index  $c$  goes from 1 to the number of points per revolution (ppr) in the acquisition system (here 2048 or 1024). It should also be noted that using the time-domain samples between each crank pulse, violates the Nyquist-criterion during the downsampling process (as seen in Figure 2.9). On the other hand, fulfilling the Nyquist-criterion implies smearing in the angular domain,



Disc attached to crankshaft

Figure 2.4: A disc with interleaved black and white squares on its circumference generate a square wave signal similar to upper signal in Figure 2.5 - this is the Crank Pulse Signal. Another such signal is the Top Dead Center pulse that emits a pulse when the piston is at its uppermost position - this signal can be used to segment the signal into cycles.

and only the samples between the crank pulses where used.

ROOT-MEAN-SQUARE (RMS)

$$rms[c] = \sqrt{\frac{1}{CRK[c+1] - CRK[c] - 1} \sum_{n=CRK[c]}^{CRK[c+1]-1} x[n]^2} \quad (2.1)$$

ROOT-SUM-SQUARE (RSS)

$$rss[c] = \sqrt{\sum_{n=CRK[c]}^{CRK[c+1]-1} x[n]^2} \quad (2.2)$$

LOAD	NUMBER OF TIME SAMPLES BETWEEN CRANK SAMPLES	FILE NUMBERS
25 %	7-8	001 - 150
50 %	5-7	150 - 275
75 %	4-6	275 - 344

Table 2.1: Number of time samples that are used to calculate the crank sample values

Calculating the total energy from the crank samples reveals the difference

$$E_{total} = \sqrt{\sum_{n=CRK[1]}^{CRK[ppr+1]-1} x[n]^2} \quad (2.3)$$

$$E_{rms} = \sqrt{\sum_{c=1}^{ppr} rms[c]^2} \quad (2.4)$$

$$= \sqrt{\frac{1}{CRK[2] - CRK[1] - 1} \sum_{n=CRK[1]}^{CRK[2]-1} x[n]^2 + \sum_{c=2}^{ppr} rms[c]^2} \quad (2.5)$$

$$E_{rss} = \sqrt{\sum_{c=1}^{ppr} rss[c]^2} \quad (2.6)$$

$$= \sqrt{\sum_{n=CRK[1]}^{CRK[2]-1} x[n]^2 + \sum_{c=2}^{ppr} rms[c]^2} \quad (2.7)$$

Figure 2.7 show that the energy in the original RMS signal (labeled true) is lower after file number 300 than in the beginning. This is not the case with the true and the RSS signals. Thus, the RMS conversion alters the energy ranking of the examples. In the end, I have settled on the Root-Sum-Square of time samples between two crank pulses, as this conserve the total energy in a cycle, such that two cycles with different running speeds can be compared. In addition, I settled on using samples between two crank pulses to calculate the crank samples, i.e., prioritizing energy location at the expense of some aliasing. However, neither the RSS nor RMS conversion can be used with the raw signals, as both methods assume that the signals are non-negative to begin with.

The conclusion is that the conversion to crank angle domain, is not a resam-

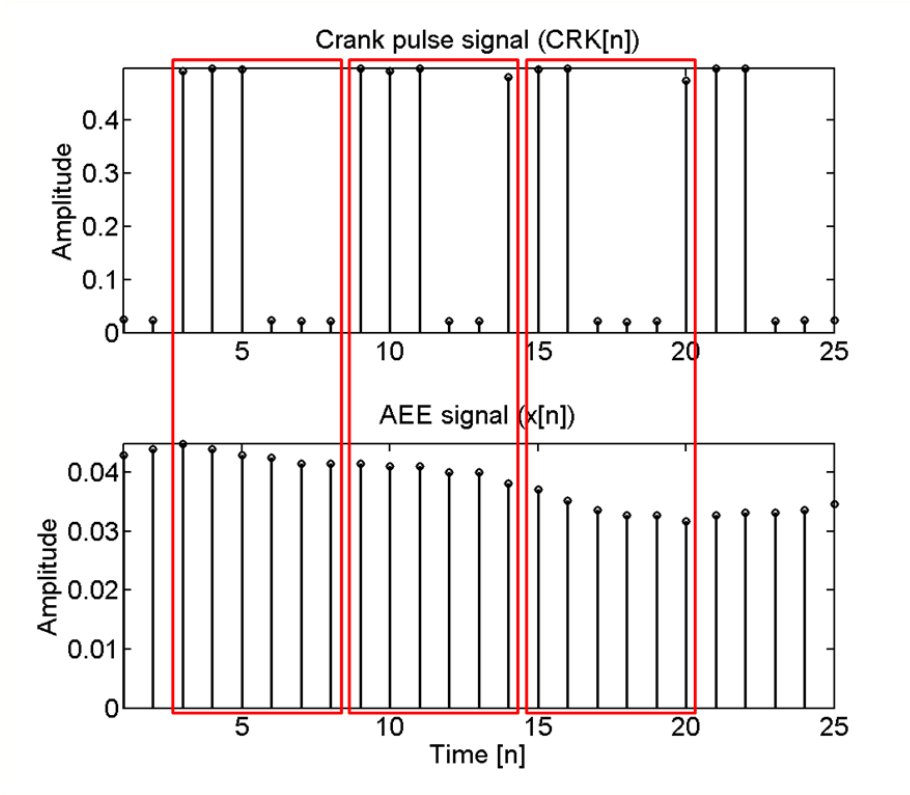


Figure 2.5: Simultaneous acquisition of crank pulse signals AEE signals. The crank angle sampling is based on localizing the rising edges of the crank pulse signal. The time between successive rising pulses as a function of the load is given in Table 2.1

pling/interpolation process but a transformation. Therefore, one should not normalize with the (square root of) number of samples between each crank pulse. However, this approach is only valid for already non-negative signals – the open question remains: how to convert signals that contain both positive and negative values. Perhaps multiplication of the interpolated value with the number of (time) samples between the two crank trigger pulse was possible.

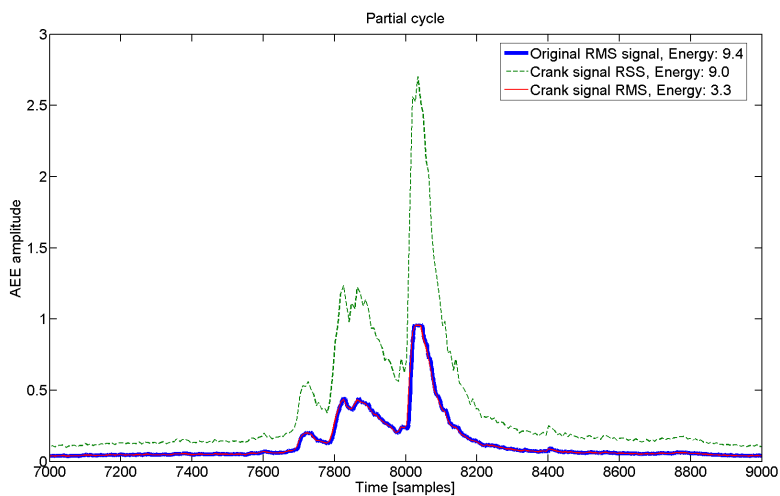


Figure 2.6: Comparison of crank angle conversion schemes, RSS and RMS. The Crank signal RSS converted using Equation 2.2 differ from the original RMS signal in the time domain. The Crank signal RMS converted using Equation 2.1 does not differ from the original RMS signal, although it is sampled at a lower rate. However, this is not the whole picture, so Figure 2.7 show the total energy in each cycle instead.

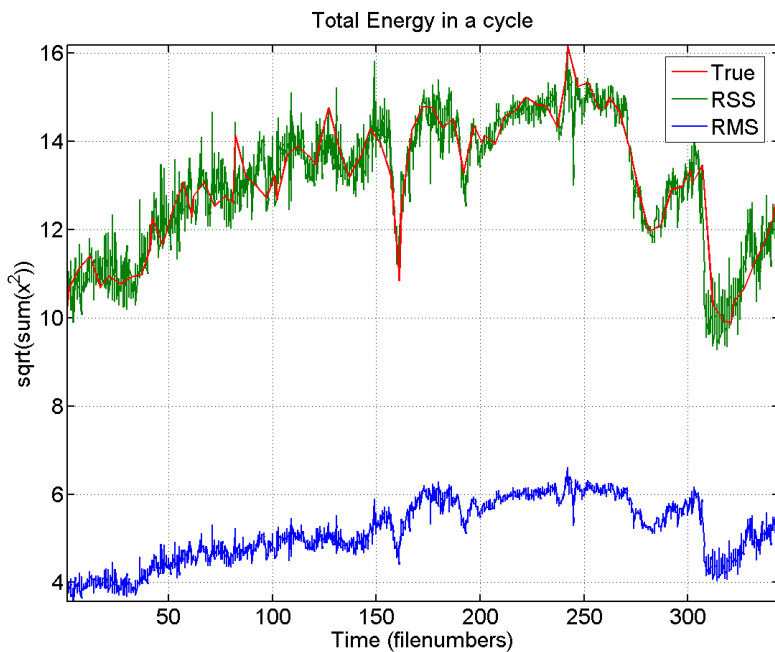


Figure 2.7: Engine cycle energy. This figure shows that summing the squared time sample values instead of averaging preserves the amount of energy in the cycle, such that when comparing the cycle energy, the RSS is similar to the time domain energy while RMS is not. Notice that the time domain energy (labeled true) was only calculated for a subset of the time domain files.

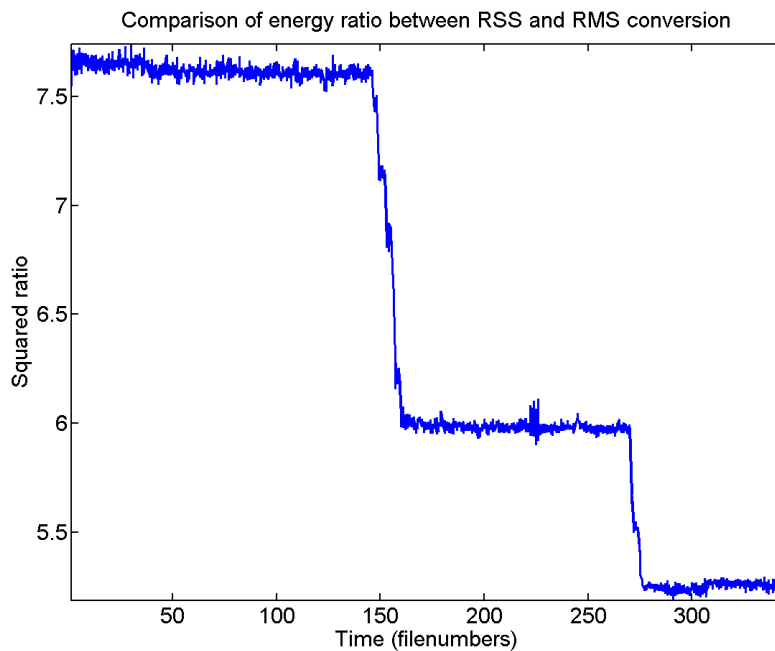


Figure 2.8: The squared ratio of energy in cycles converted with sum. The squared ratio corresponds to the number of samples between each crank pulse as tabulated in Table 2.1, the numbers in the table are 7-8, 5-7 and 4-6 for the three periods

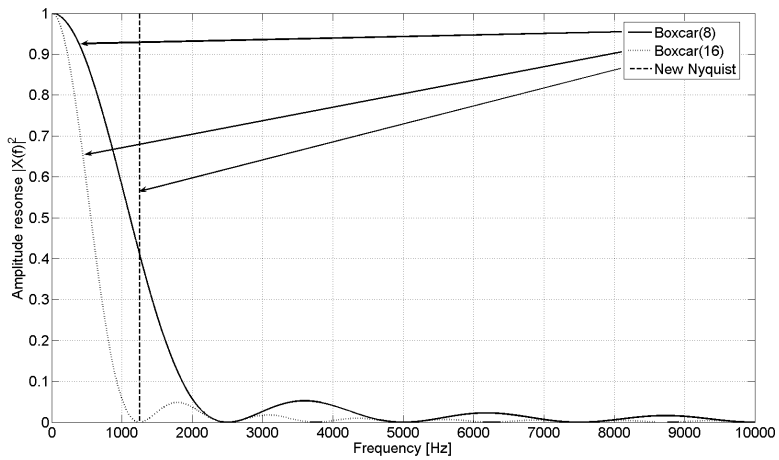


Figure 2.9: Using the sum or average of the 8 samples between two crank pulses is not enough low-pass filtering for 8 times downsampling, since the first zero is at 2500 Hz, when it should have been at 1250 Hz. Proper filtering is obtained using 16 samples, i.e., the samples between three crank pulses – however this smears the location of energy.



## Event alignment

---

Many publications on condition monitoring have been restricted to stationary conditions, i.e., detecting a fault while nothing else changes. Under marine operation, the settings are changing from time to time, due to navigation and/or water current flow. In both cases the amount of power that the engine has to deliver changes. Additional power can be produced with additional fuel (bigger explosion) or quicker rotation (additional explosions). In both cases, the angular timing of events can be optimized for combustion performance. The timing can be changed with mechanical devices [Jensen, 1994] or electronically as in the *Intelligent Engine* developed by MAN B&W. Such movement is observed in Figure 3.1 just after 0 degrees. The peak is delayed around example 800 and further around example 1600, both places where the load changes. The event at 130 degrees does not move, showing that the timing changes are not constant, but changing as a function of angular position and the applied load.

From a condition monitoring, point of view an alarm generated from such timing change is false and should be avoided, so the condition monitoring system has to be invariant wrt. such changes. Where and how this invariance should be build into the system depend on the application. Basically three ideas has been considered:

1. Train the different models for different settings. Each operational setting has its own model. This would be reasonable if only a relatively small

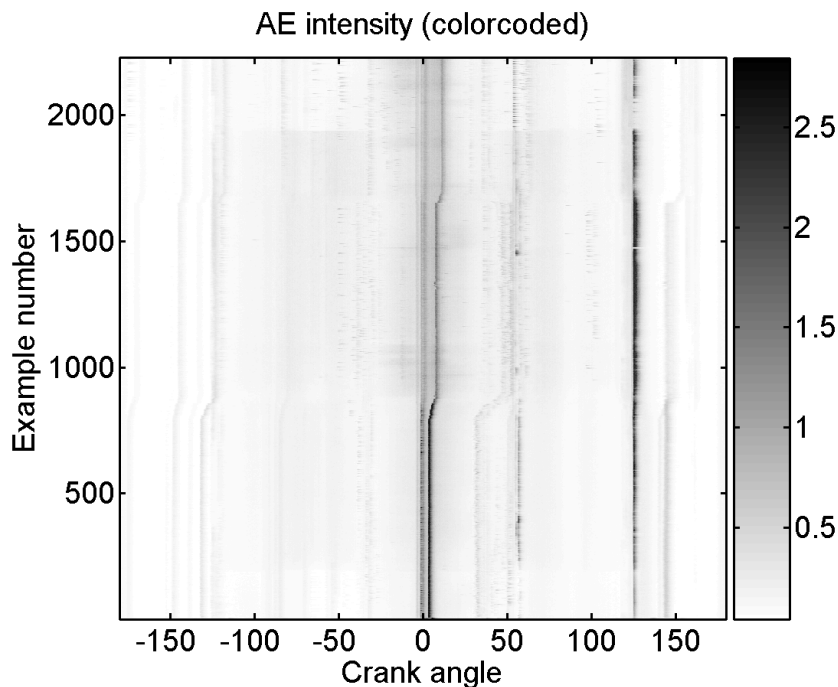


Figure 3.1: The AE intensity for 25%, 50% and 75% load, the timing changes in the signals are very pronounced just after  $0^\circ$ , i.e., in the combustion period. Around  $135^\circ$  the piston passes the scavenge air holes; these holes are not movable so the event is fixed in angular position. The load changes from 25% to 50% around example 800 and from 50% to 75% around 1600.

subset of operational settings was widely used.

2. Train a single model on data from different settings. If a relatively small subset of operational settings was used. Samples from all of them would be combined into a model used all the time.
3. Train a single model on data from a single setting, and formulate a warp model for the other settings with *event alignment*.

In a recent master thesis project, vibration signals were used for condition monitoring (CM) of windmills in operation. As a preprocessing step, the observed signals were grouped in power intervals, and new observations compared to the models trained with that power setting [Jørgensen, 2003]. With data from three load settings, all obtained from the test bed engine in Copenhagen, the two first methods were outperformed by event alignment [Pontoppidan et al.

[2005a] and Appendix C].

The event alignment as such is a pre-processing step that remove the variations due to known timing changes in the acoustic emission (AE) signals. In image and speech processing it is known as warping, and in other fields of research such methods are known as *functional data analysis*, *signal matching*, and *data registration*. Recently a similar method was applied to rail track condition data obtained with a measurement vehicle [van de Touw and Veevers, 2003]. There the observed changes were due to calibration errors and to the fact that the locomotive could not maintain the same speed profile from measurement to measurement.

The basic idea is that a warp model can be used to transform signals from one setting into another setting; and when applied to deviations the transformation should fail to transform them into the reference condition. Graphically the warping moves and scales a volume (an ellipsis in Figure 3.2) such that it matches the reference volume. When applied to examples outside that volume (faulty examples) the warping should miss the reference volume. In Figure 3.2 the crosses outside the upper left circle should be warped to positions outside the upper right circle.

The timing of the different engine events, e.g., injection and valve operations, are a part of the engine layout, the term for the control of the engine based on parameters such as load, running speed and usage. The visible timing changes in Figure 3.1 are the result of changing load changes under operation on the *propeller curve* (a particular engine layout). This layout defines the running speed and the timing of events as a function of the load under the normal setting when run as a marine engine.

When the engine is run as a power plant, i.e., attached to a power generator like the engines at Kos Island Power Plant, the running speed is kept constant even though the load is changing. This layout is the *generator curve* [Jensen, 1994]. Some other operational layouts are mentioned in Table 3.1. In the AEWATT project experiments have been conducted with the propeller and generator curves, and the development and research of the event alignment procedure is based on the properties of the propeller curve.

## 3.1 Time alignment

In Figure 3.1 we observe that some events are moving as a function of the load while others stay at the same angular position regardless of load changes. Let

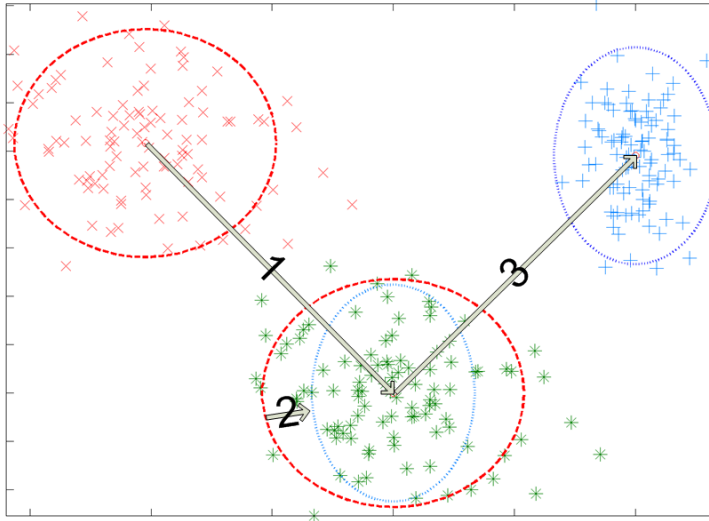


Figure 3.2: Basic idea of Gaussian warping, the upper left cluster is moved (1), stretched (2), and moved again (3) to match the area of the upper right cluster. Examples outside the dashed circle at the original location end up outside the dotted circle at the end location. Converted into normal and faulty examples the faulty examples outside the acceptance original region are not moved into the final acceptance region.

Propeller curve	is for normal marine operation
Generator curve	is for constant running speed independent of load
NOx curve	is optimized for reduced NOx emission
Vibration curve	is for harbor navigation, where running speed is close to structural eigenfrequency

Table 3.1: Some engine layouts, see further Jensen [1994]. Data from propeller and generator curve have been acquired during the AEWATT project

us assume that these changes can be inverted by distorting the time axis

$$y(t) = x(w(t)) \quad (3.1)$$

where  $w(t)$  is the time warp function that inverts the timing changes. Now the question is simply how the warping function should be obtained. Assuming that the sequence of events is constant, e.g., the same events are observed in the same

order regardless of load, we can think of some properties that the warp function must fulfill.

- Monotonically increasing, the warp should not imply that something is happening in reverse order, i.e., combustion before injection
- Continuous, such that events are not skipped

Additionally properties that regulate how much the warp function can deviate from  $w(t) = t$  as well as limits on the local advancement speed  $\frac{dw(t)}{dt}$  are considered in the following sections.

If the sequence of events were not constant, the event alignment problem would have to be addressed in a different manner by separating the events before individual event alignment. In the simplest case, the two events could be spectrally separable, such that a different warp function could be applied to the different spectral components, but it has not been necessary and thus not investigated. If the events were not separable in frequency, the traditional use of blind source separation on simultaneous recorded channels could be used to split the events prior to individual event alignment.

### 3.1.1 Automatic warp paths

The dynamic time warping (DTW) was developed in the context of speech recognition, solving the problem that the length of each phoneme is varying from observation to observation. When matching a sound against a reference, the length of each phoneme could be adjusted to follow the reference [Ellis]. This is pretty close to the usage here. However due to the increased similarity from cycle to cycle compared to repetitions of phonemes, the warp should be learned for a group instead of for an example. Thus, the warp path should be obtained from and applied to observations from the same condition, say 50% load.

The warp paths are obtained by splitting the data into overlapping frames of equal length from which the windowed short time fourier transform (STFT) is normally calculated. Say we have  $F$  frames and two signals of same length we now compare each frame in signal 1 with each frame in signal 2 – and obtain a  $F \times F$  similarity matrix. The warp path is the optimal path from position (1,1) to (F,F). The comparison is the standard angular difference between vectors. Visualizing the similarity matrix as a landscape we select the route with the lowest sum of heights, with respect to the constraint that we are not allowed

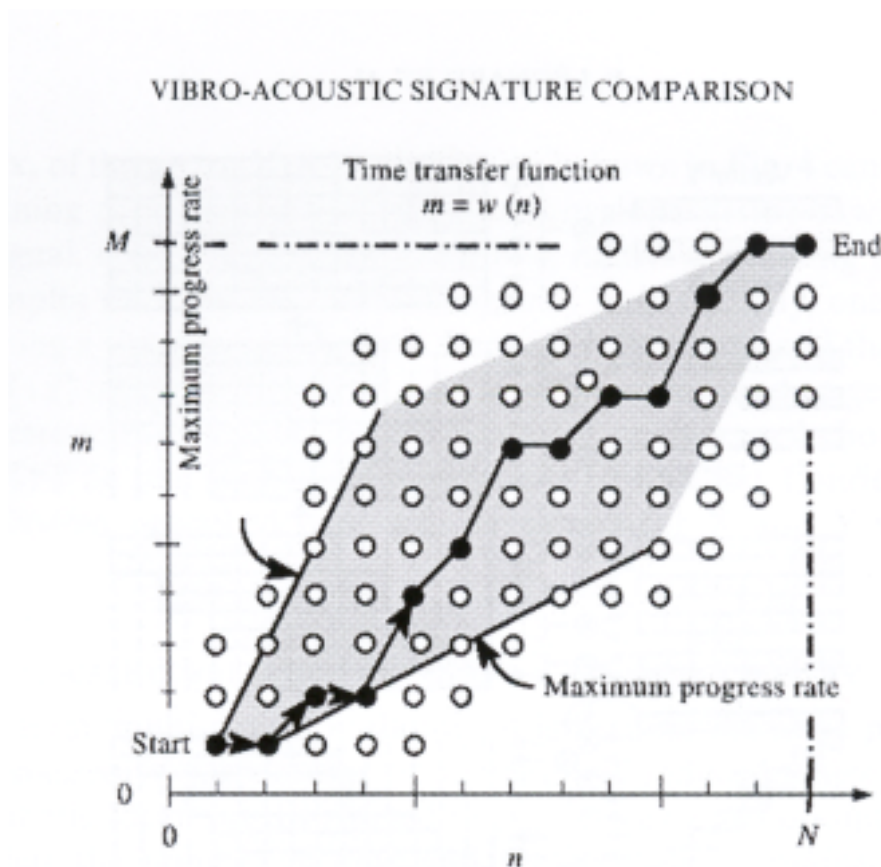


Figure 3.3: Warp path obtained with Itakura-parallelgram (from Leonard et al. [2000]). The warp path is non-decreasing, i.e., obviously the warp function should not go back only forward in time.

moving back in any direction, i.e. monotonically increasing paths. Due to the *sum* the height differences on the path do not matter. The warp path is now a new sequence (with repetitions) of the original frames that matches the reference.

Looking on the similarity measures (the gray scale coded images) in Figure 3.4 and 3.5 we see that the peaks/landmarks in the signals are easily identified as “white” passes at the dark mountains, thus the similarity measure correctly identify that the peaks should be aligned. Away from those passes the landscape seems flat and as Figure 3.4 and 3.5 show it is here the DTW fails to find a reasonable path. In this context the simplest warp path is  $w(t) = t$ , and

complexity is due to curvature and slopes far from 1, i.e., vertical and horizontal local warp paths.

### 3.1.1.1 Warp path constraints

The first set of constraints applied to the warp paths are local constraints, the ones that tell which steps are possible. In the simplest case, only three steps are allowed: (1,1), (0,1), and (1,0). These steps do not allow for skipping frames, but two frames can be put on top of each other (that is played back simultaneously) if (0,1) is chosen. Also the steps (1,2) and (2,1), that make frame skipping possible, can be allowed (they are in [Figure 3.3](#)), but are not available in the [DTW](#) code due to [Ellis](#) and have not been considered in this thesis.

The second set of constraints are more global and somewhat heuristic trying to keep the warp paths within *reasonable bounds*, by putting limits on how far the warp path can move away from the simplest warp path. The Itakura-Parallelogram [[Leonard et al., 2000](#)] define a minimal and maximal progression rate both with respect to the start and end point as seen in [Figure 3.3](#). From below the warp path is first constrained by the minimal progression rate and later by the maximal progression towards the end, and in an opposite manner by the upper bounds. The Sakoe-Chiba band on the other hand specify a narrow or wide straight *highway* from start to end. In [Figure 3.4](#) and [3.5](#) the Itakura-Parallelogram are applied to the left side and Sakoe-Chiba band to the right side plots. The obtained paths are a function of both local and global constraints, however the constraints cannot save the [DTW](#) curves from being too complex.

### 3.1.1.2 Limits of dynamic time warping

Finding the warp paths from the available data is problematic, since we observe both time and amplitude changes in the signals as a function of the load. This leads the [DTW](#)-algorithm [[Ellis](#)] to propose warp paths that in general are more complex than necessary, as they additionally try to solve the amplitude problem as well, even though the angular vector comparison should take those differences out.

Also, when no time warping is not necessary the [DTW](#) proposes a warp function that is not  $w(t) = t$  as shown in [Figure 3.4](#). Moreover, when time warping is needed the complexity of the warp paths (compared to the simplest warp path  $w(t) = t$ ) is too high. For instance [Figure 3.5](#) have very steep curves at the end (100-120), due to the differences in the peak tails in [Figure 3.7](#)

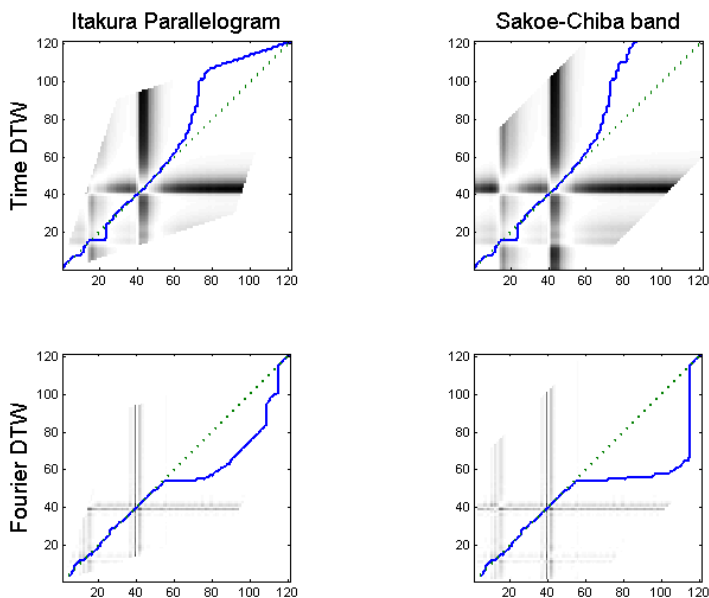


Figure 3.4: Warp paths obtained with DTW from 25% load to 25% load, thus no warping is necessary. The time DTW is based on aligning time samples, while the Fourier DTW is based on aligning STFT of frames

### 3.1.2 Landmarks

From propeller curve acoustic emission energy (AEE) signals from the Man B&W test bed engine, I defined a set of landmarks that should align the engine events and thus the signals in angular domain. My landmarks also shown in Figure 3.6 are solely based on the peaks in the signal, i.e., not using any mechanical engine knowledge at all. All peaks got a landmark, regardless of the origination of the peak. This ensured that both moving events were aligned and stable events kept aligned. The landmarks have been compared with a similar analysis of the same date performed by Heriot-Watt University, UK (HWU) and reported in AEWATT Deliverable 2 [AEWATT Project Consortium, 2003a]. In their analysis, they focused on understanding and labeling of the events according to their mechanical origin. Figure 3.6 show the consistency between their findings and my landmarks, where the labels originate from their tables.

The AEE signals obtained from the engine are very similar across load settings; the sequence of events seems constant. Therefore, the warping should be a



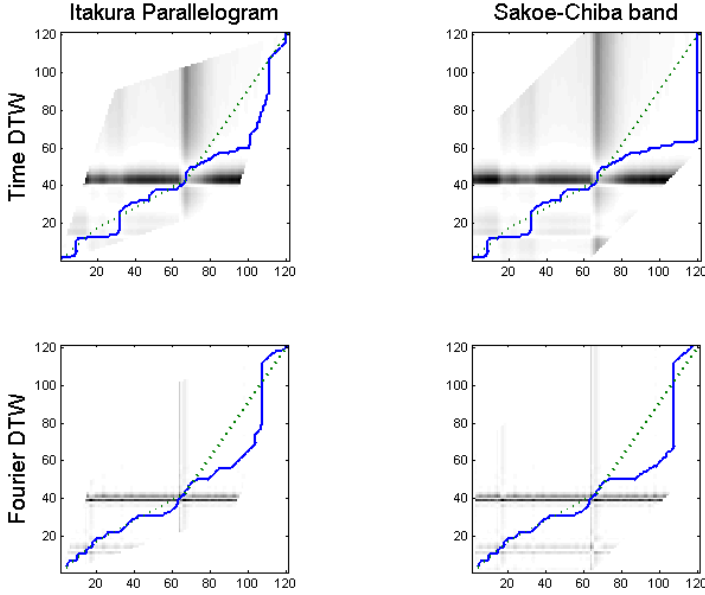


Figure 3.5: Warp paths obtained with DTW from 50% load to 25% load. For the STFT (denoted Fourier) warp paths, optimal window length and overlap was selected (one for all load settings). The time warp paths use just a single point. For signal the four warp paths are consistent, still for the single sample based methods. The dotted lines in the plot indicate piecewise linear interpolation between the landmarks shown in Figure 3.7

matter of stretching the duration and spacing of and in between the events. In my model, each peak in the signal is an engine event. The notion landmarks originates from image warping and are obviously objects that we can identify independently in all observations. The landmarks provide information that can be used to identify the underlying true timing map, and are the time indices that describe the start, peak and end of each peak. Given a reference signal with  $N$  landmarks at times:  $\{L_1\}_n$  the warp function  $w(t)$  is the one that transform the other set of landmarks  $\{L_2\}_n$  into the reference.

$$w(\{L_2\}_n) = \{L_1\}_n, \quad n = 1, \dots, N \quad (3.2)$$

$$\{L_{25\%}\}_n = \{1, 38, 47, \dots, 1845, 1858, 1890, 2048\} \quad (3.3)$$

$$\{L_{50\%}\}_n = \{1, 60, 76, \dots, 1847, 1861, 1920, 2048\} \quad (3.4)$$

Where each landmark is the crank angle sample number, starting with 1 at  $-180^\circ$  and ending with 2048 at  $(179.8)^\circ$ . A warp from mapping events posi-

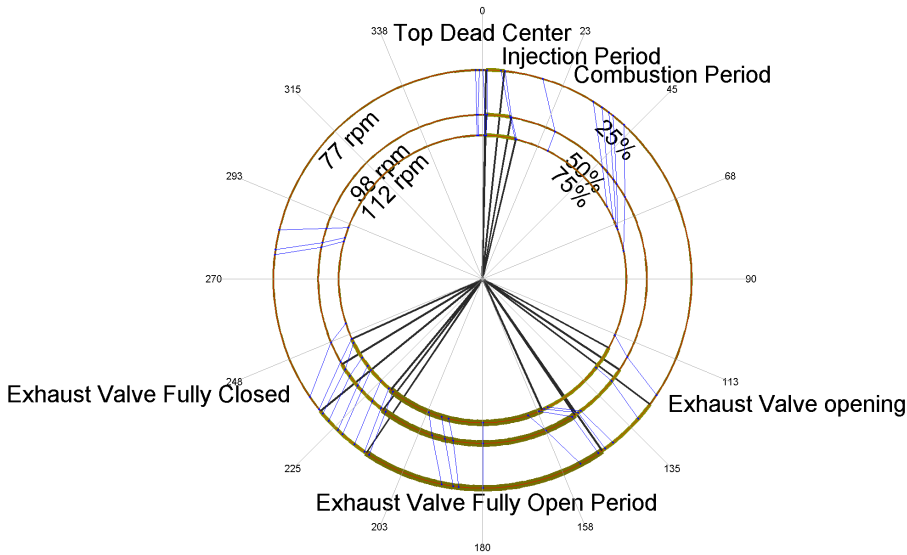


Figure 3.6: The landmarks in *radar view* for propeller curve data from Man B&W test bed. The thin blue lines indicate the landmarks that I obtained by hand, the thicker (and fewer) arcs are based on the tables describing event positions in AEWATT Deliverable 2 [AEWATT Project Consortium, 2003a].

tioned as at load 50% to 25% would be  $w(\{L_{50\%}\}_n) = \{L_{25\%}\}_n$ .

### 3.1.3 Frequency preserving time stretching

One thing is obtaining the warp path, but it also has to be applied to the signal. The *Phase Vocoder* that allows signals to be stretched in time without moving the frequency components was first described by Flanagan and Golden [1966]. Like the DTW it works with the STFT of the signals. The concept is simple – by changing the number of samples between each (overlapping) signal frame in the STFT the overall duration of the whole signal can be changed without changing the time/frequency content in the individual frames. The problem with applying the phase vocoder on condition monitoring data is that while it preserves the overall frequency content; important peaks can either be repeated or removed, and those dropped/spurious peaks would generate an alarm. With that in mind overall frequency preservation as a requirement was

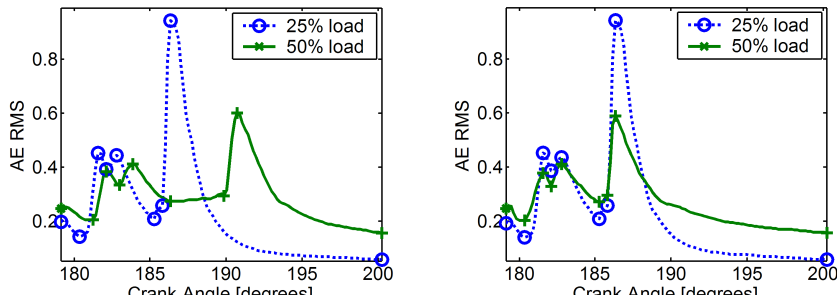


Figure 3.7: Landmarks and signals for two load settings. Notice the  $180^\circ$  degree phase shift on the axis ( $180 = \text{TDC}$ ). The left figure show the signal with landmark indicators before aligning. The time warp that aligns the landmarks also aligns the peaks in the signals and this result is shown to the right.

dropped. Moreover, the distance between some of the engine events is so small that good spectral measures of the individual peaks are not available.

### 3.1.4 Spline interpolation

A set of sequential landmarks identify some points on the warp path, but does not tell what happens in between the landmarks. Two similar methods have been investigated, piecewise linear interpolation and cubic spline interpolation. The piecewise linear interpolation merely connects the landmarks with straight lines, thus at the landmarks the slope of the warp path is discontinuous. With cubic splines, the second derivative, the curvature, is continuous [Shampine et al., 1997], and the abrupt slope changes at the landmarks are removed. However, the cost is a more wiggly warp path as seen in Figure 3.8 and Figure 3.9. Moreover the cubic splines do not guarantee that a set of monotonically increasing landmarks result in a monotonically increasing warp path. In Figure 3.8 the cubic spline proposes a warp path that *goes back in time* around  $5^\circ$ , and thus violates the monotonically increasing requirement.

#### 3.1.4.1 Inverse warp paths

Now the warp paths obtained from two landmarks sequences is compared to the warp path obtained when exchanging the two sequences. The two warp paths should ideally be each others inverse – actually that is the whole idea with the landmarks that they should invert the timing changes that have been applied

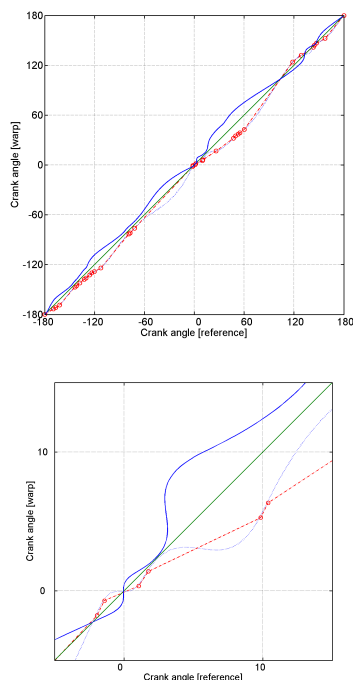


Figure 3.8: The circles indicate the landmarks. The two dotted lines indicate the piecewise linear and cubic spline interpolations of the landmarks. The solid wiggly line is the actual warp path based on cubic spline interpolation. It is easily seen that the warp path is just the landmark interpolation mirrored in the straight “no warp” line. The figure to the right is a zoom in on the injection period one of the most difficult periods. The landmarks are close and the slopes are changing. As Figure 3.1 also show the events under the 50% load condition happen after the 25% load.

to the AE signals. With the linear interpolation between landmarks this holds, but not for the cubic splines. However since the difference is rather between the two large it is worthwhile to consider both warp paths and select the best; and obviously invert the warp path if necessary. Thus if one of them violates the requirements we can use the other one, if both of them are valid we can choose the one that is closest in some measure, say mean square, to the linear warp path. In Figure 3.9 the green warp path - obtained from the inverse landmarks and then inverted is considerably better than the warp path obtained directly.

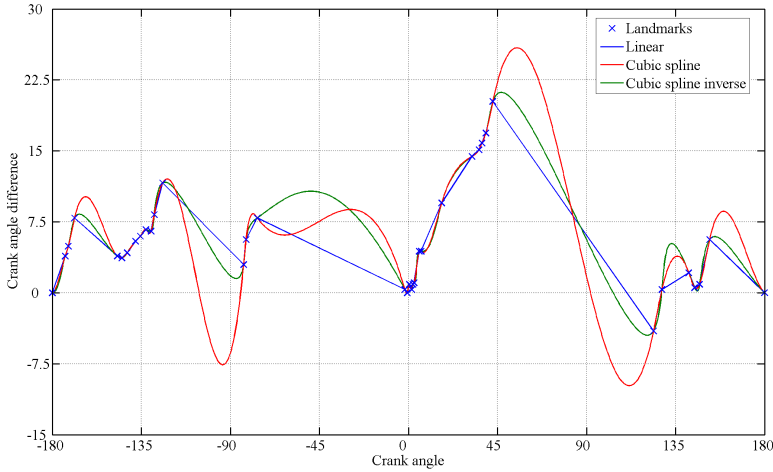


Figure 3.9: Warp paths, inverse warp paths

## 3.2 Amplitude alignment

Not all of the variation can be explained by the time stretching also an amplitude mismatch as seen in [Figure 3.7](#) has to be corrected.

Based on some independent samples of the reference (denoted  $x$ ) and warp (denoted  $y$ ) condition, an amplitude alignment function can be estimated. Three simple methods have been considered: Addition [Equation 3.5](#), multiplication [Equation 3.6](#) and a combination [Equation 3.7](#). As both mean and variance differences were observed between loads the combined method was selected. The mean and variance of the two set of samples are estimated and denoted  $\mu_{x,y}$  and  $\sigma_{x,y}^2$ . All estimates are obtained from independent sets, i.e., learning the whole event alignment requires three independent training sets from each condition: landmarks, mean, and variance.

$$\tilde{y}(t) = y(w(t)) - \mu_y(w(t)) + \mu_x(t) \quad (3.5)$$

$$\tilde{y}(t) = y(w(t)) \frac{\mu_x(t)}{\mu_y(w(t))} \quad (3.6)$$

$$\tilde{y}(t) = (y(w(t)) - \mu_y(w(t))) \frac{\sigma_x^2(t)}{\sigma_y^2(w(t))} + \mu_x(t) \quad (3.7)$$

If the variance in the reference condition is larger than the variance in the

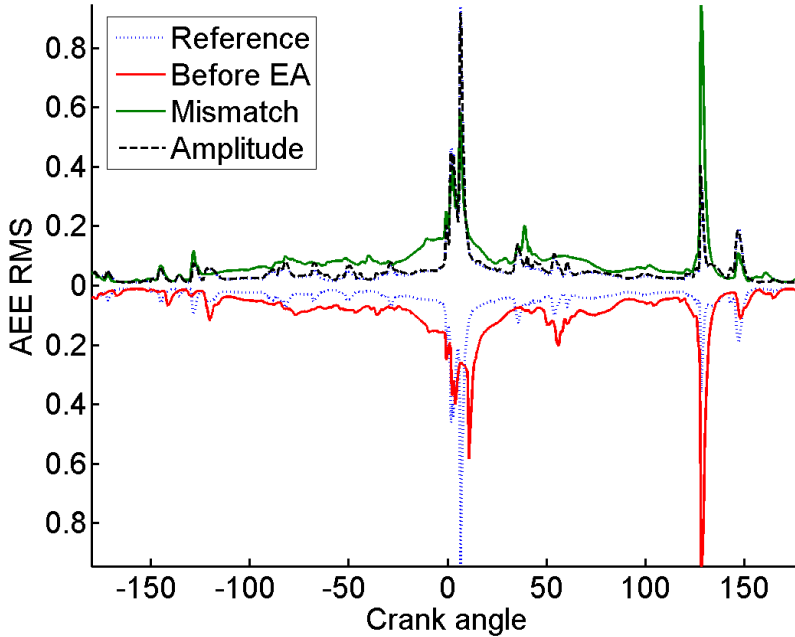


Figure 3.10: Amplitude mismatch after time alignment. The figures show the mean of a group of signals taken through the steps in the event alignment. For clarity, two of the signals have been vertically flipped. We begin with the red signal that we want to align with the dotted blue signal. The first step is the time alignment that results in the green signal. Finally the amplitude alignment is applied which puts dashed black signal on top of the reference signal.

warp condition, an amplification of the variance could cause the amplification of measurement noise. To prevent this, the ratio  $\frac{\sigma_x^2(t)}{\sigma_y^2(w(t))}$  is constrained by a limit of 1.

However, by not scaling the variance up when necessary, it also makes the event aligned examples *supernormal* since they are much closer to the load mean signal as seen in Figure 3.11. Thus, the event alignment warp would make faulty examples more like the normal and possibly prevent detection. It could also lead to serious overfitting and false alarms if such examples are used to train models or learn rejection thresholds, due to artificial smaller mode variance.

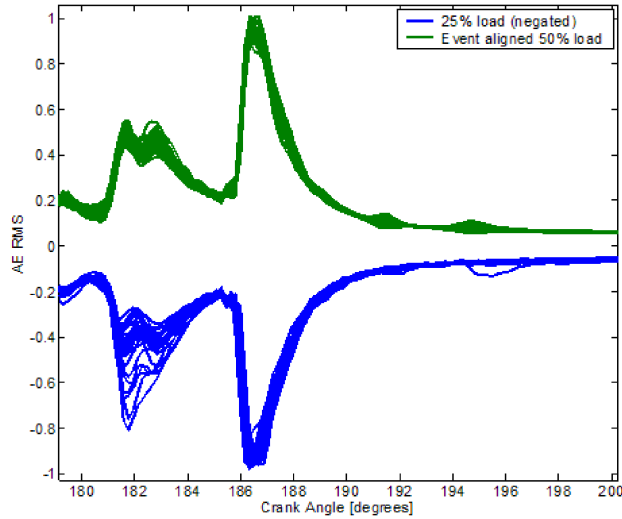


Figure 3.11: Variation in the AEE signal in the injection period ( $180^\circ$  degree phase shift) after event alignment is applied. The variance in the signals at reference load (25%) is higher than in the event aligned 50% signals. This is due to the constraints on the variance scaling that protects against amplification of measurement noise, but also introduce the risk of making the event-aligned signals *super-normal*.

### 3.3 Modeling the continuous warp functional

In the event alignment publications [Pontoppidan and Larsen, 2004, Pontoppidan and Douglas, 2004], it was demonstrated how event alignment could be used for non-stationary change detection. The analysis with three different load settings on the propeller curve indicated that it would be possible to formulate a warp model that was continuous in the load parameter. The event alignment framework considered in this chapter, require one warp function for each load. A functional behavior was revealed by a visualization that preserves angular and time information by displaying the data on circles where the circumference is proportional to the cycle time. Figure 3.6 shows that relatively smooth non-linear functions of load and angular position can be formulated in this domain. From such figures one can also identify the events with constant time length and other functions of dependence, e.g., square, log etc...). From an engine developer/manufacturers point of view, this analysis might not be necessary. The adjustment of the events are computed by the engine control system, so the information must reside somewhere in the system and the design. However, information has not been available.

Since [Figure 3.6](#) indicated that a functional behavior was able to explain the movement of the events, i.e., describing the placement of the landmarks as a function of the load, Dr. S. Sigurdsson is currently investigating if it is possible for artificial neural networks ([ANN](#)) to learn this function. The warp path is approximated by a positive mixture of tanh-functions. The tanh-function is monotonically increasing, and so is a warp path consisting of tanh's with positive weights.

$$w(n, \lambda) = \sum_{k=1}^K a_k \tanh(b_k n + d_k \lambda + c_k), \quad a_k, b_k \geq 0 \forall k \quad (3.8)$$

where  $n$  is the crank angle position,  $\lambda$  the load, and  $a_k, b_k, c_k$ , and  $d_k$  the parameters of the [ANN](#). Keeping  $a_k, b_k$  non-negative fulfils the monotonic requirement, while  $d_k$  model the load dependency and  $c_k$  is just a translation.

The interpolation between known loads comes from considering the training inputs with targets  $(20, \{L_{20}\}_n), (40, \{L_{40}\}_n)$ , so that a functional behavior for  $\{L_\lambda\}_n$  can be learnt. Moreover if this is successful additional input such as engine layout curve (propeller, generator, vibration, NOX-curve) can be considered.

### 3.4 Downsampling - a crude approach to removing load changes

Finally as a simple alternative, it was investigated if the timing changes in the [AEE](#) signals could be removed with downsampling. Comparing the upper and lower panel of [Figure 3.12](#) the timing changes in the upper panel was been removed. While the downsampling removed the timing changes it did not remove the amplitude changes, thus some load dependent processing would still be necessary. Alternatively the [AEE](#) signals in the period could just be summed to preserve energy. Still timing information for proper selection of intervals would be necessary. With this setup the landmarks would just be used like crank pulses in the crank angle conversion ([section 2.4](#), i.e., that the observed crank samples between each landmark was summed. This could be seen as a more advanced usage of *domain knowledge*, compared to the work by [Chandroth and Sharkey \[1999\]](#), where domain knowledge is applied by only considering the most *important* part of the cycle.

A simple test was conducted where the necessary downsampling factor for the whole cycle was roughly estimated from the obtained landmarks to around 50. In the example in [Figure 3.12](#) the downsampling factor is 96. Recall downsampling



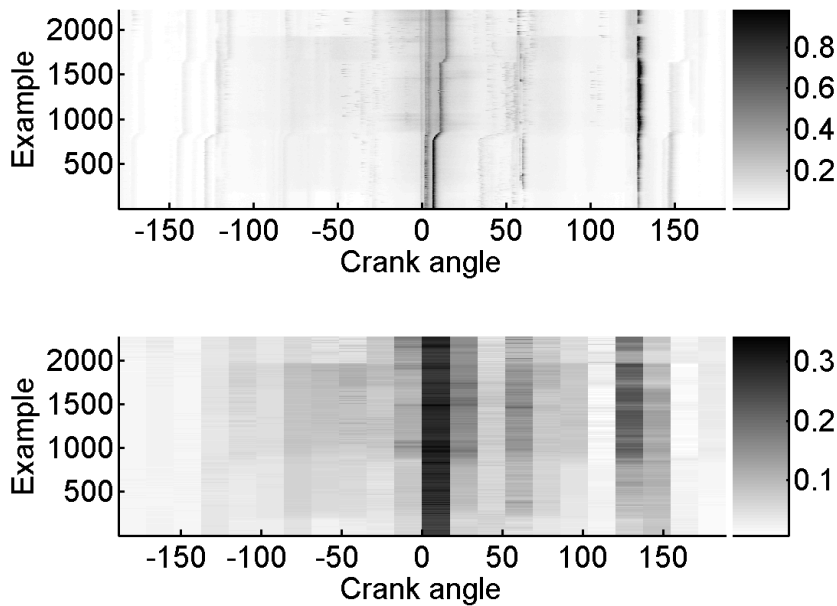


Figure 3.12: The timing differences has been downsampled away (factor 96).

with factor 96 results in lowpass filtering and selection of every 96'th sample. The factor is higher as all events needs to be in the same 96 sample long block regardless of load. An event that is moving from one block to another would result in timing changes. However, some faults were removed in this way. The faulty water brake resulted in unstable timing of the events [Pontoppidan and Larsen, 2003], and downsampling with factor 8 lead to decreased fault detection of that fault [Pontoppidan and Larsen, 2004].



# Condition modeling

---

In this chapter, I outline the various methods that I have used to model the engine condition in observed data. The methods share some important properties, namely being generative and capable of data reduction. The two main equations in this chapter are

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \boldsymbol{\nu}, \quad \boldsymbol{\nu} \sim N(0, \boldsymbol{\Sigma}) \quad (4.1)$$

$$\mathbf{X} = \mathbf{A}\mathbf{S} + \boldsymbol{\Gamma}, \quad (4.2)$$

where  $\mathbf{x}$  is the observation vector of size  $d \times 1$ ,  $\mathbf{A}$  the mixing matrix of size  $d \times k$ ,  $\mathbf{s}$  the source signal of size  $k \times 1$  and  $\boldsymbol{\nu}$  the additive noise is also  $d \times 1$ .  $d$  is the number of features and  $k$  the number of components, and  $k \ll d$ . The observation matrix  $\mathbf{X}$  is generated by stacking several realizations of the observation vectors. Here the different realizations comes from different engine cycles acquired with the same sensor, i.e., they are not simultaneously recorded as in the classical blind source separation problems [Bell and Sejnowski, 1995, Molgedey and Schuster, 1994]. Similarly the source matrix  $\mathbf{S}$  and the noise matrix  $\boldsymbol{\Gamma}$  comes from stacking the  $N$  source vectors and noise vectors.

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \quad (4.3)$$

$$\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_N\} \quad (4.4)$$

$$\boldsymbol{\Gamma} = \{\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \dots, \boldsymbol{\nu}_N\} \quad (4.5)$$

The assumption that the noise is Gaussian with zero mean does not hold completely, as the sensor noise and the signal is added and squared in the root mean square (RMS). Being uncorrelated they add as energy signals, i.e.  $rms = \sqrt{s^2 + 2ns + n^2}$ , having  $s, n$  as signal and noise respectively. Since the signal and noise are uncorrelated the mean of the  $2ns$  is zero, and further since the overall noise level is low compared to the signals the non-zero mean can be neglected.

Equation 4.1 describe how the  $k$  hidden signals in  $\mathbf{A}$  are weighted by the coefficients in  $\mathbf{s}$  to generate the observed signal  $\mathbf{x}$ . In other words the  $\mathbf{A}$  matrix contain those signal parts that the observed signals can be made up from - it acts like a basis for the normal condition. The idea is to learn this basis set from a collection of normal condition data, making the model capable of generating the different modes in the observed training data. By applying the component analysis methods the orthogonal/independent directions in the observed data should result in a basis, i.e., columns in the mixing matrix, that contains signatures with the descriptive quality like source 3 (the third row of  $\mathbf{S}$ ) model the amplitude of the injector event signal in column 3 of the mixing matrix. As Figure 4.1 show, such clear descriptive quality is not always encountered, since the columns of the mixing matrix seem to model parts of all events in the cycle.

In the introduction of the chapter, data reduction was stated as a property of the models. Consider a group of observations as in Equation 4.2, and assume that  $k < N \wedge k < d$ , then the  $Nd$  values of the observation matrix is modeled by the much smaller  $k(d + N)$  values, e.g., each example  $\mathbf{x}$  is modeled by the  $k$  source values multiplied on the  $k$  core signals in the mixing matrix.

In addition, noise model assumptions can be made. In this thesis, two assumptions have been considered. Either  $\mathbf{\Sigma} = \sigma^2 \mathbf{I}$ , i.e., independent and identically distributed (iid), which assume a constant noise level throughout the engine cycle. Alternatively, the more advanced  $\mathbf{\Sigma} = \sigma^2 \mathbf{I}$  where the noise level is assumed to be varying through out the engine cycle (see subsection 4.2.3).

Solely examples acquired under normal conditions have been modeled, but the same procedure could be repeated on other important faulty conditions, as this would enable the identification of these faults. This way the simple normal/faulty condition monitoring system could be expanded into a more full diagnosis system.

The models describe signals originating from one sensor by already observed signals from that sensor. Conceptually the model knows how that engine normally sounds with that sensor in that position. The knowledge of the normal sound pattern is used to output the negative log-likelihood (NLL) for each example, given the normal condition model and possibly the expected noise level. The

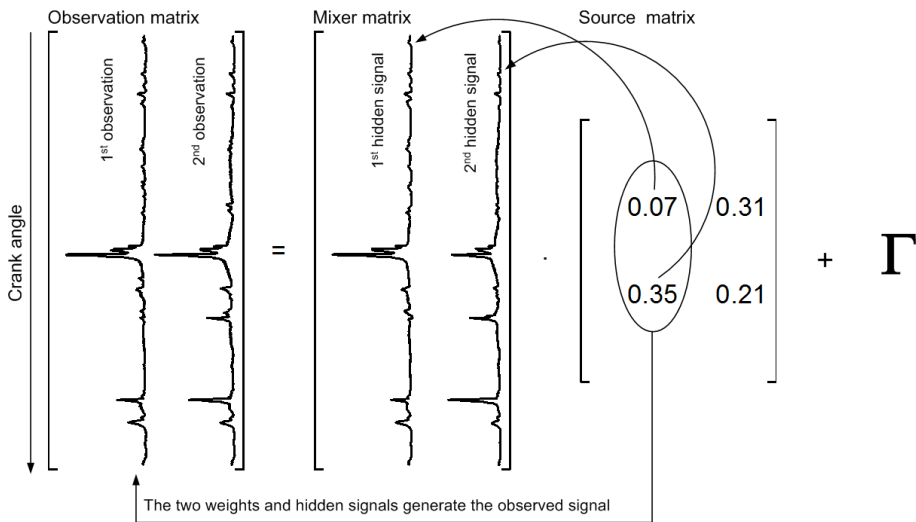


Figure 4.1: Graphical explanation of matrix setup. Here an observation matrix with 2 examples is expressed by the weighted sum of the two columns of the mixing matrix. The elements in the source matrix are the gains or activations of the core signals found in observed data using the component analysis methods.

identification of a fault is performed by monitoring the [NLL](#) against a threshold. Depending on the amount of data this threshold can be established by comparing the [NLL](#) of known normal and faulty examples (supervised) or when only normal examples are available by selecting an inherent rejection rate of the normal examples. In chapter 5 the handling of the [NLL](#) values and thresholds are described further.

## 4.1 Properties: Independent, orthogonal and uncorrelated

Prior to the description of the actual algorithms, some properties of variables are considered. Columns vectors  $\mathbf{x}$  and  $\mathbf{y}$  are either orthogonal, uncorrelated or statistically independent if one or more of the corresponding equations are

fulfilled

$$\text{Orthogonal if :} \quad \mathbf{x}^\top \mathbf{y} = 0 \quad (4.6)$$

$$\text{Uncorrelated if :} \quad (\mathbf{x} - \mu_{\mathbf{x}})^\top (\mathbf{y} - \mu_{\mathbf{y}}) = 0 \quad (4.7)$$

$$\text{Statistically independent if :} \quad \int \int p(\mathbf{x}, \mathbf{y}) \log \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x})p(\mathbf{y})} d\mathbf{x} d\mathbf{y} = 0 \quad (4.8)$$

Statistical independence is that the joint distribution can be factorized by the marginal distributions:  $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y})$ , making the numerator and denominator in Equation 4.8 equal.

Another definition of independent is

$$E\{g_1(p(y_i))g_2(p(y_j))\} - E\{g_1(p(y_i))\}E\{g_2(p(y_j))\} = 0, \quad i \neq j \quad (4.9)$$

However, it is not very convenient as it requires trying all measurable functions  $g_1$  and  $g_2$  [Hyvärinen, 1999]. Nevertheless, it shows what statistical independence is really about - that there should be absolutely no way of linking the observations in  $y_i$  with those in  $y_j$ .

For joint Gaussian distributions independent and uncorrelated is equivalent [Hyvärinen, 1999]. Since the covariance matrix becomes diagonal and the distribution factorizes in the two marginal distributions. For zero mean signals uncorrelated and orthogonal is equivalent; and joined these two properties imply that zero mean orthogonal Gaussian variables are independent. However, this is not good as the independent components analysis (ICA) algorithms require that none/or at most one source is Gaussian to recover the mixing matrix directions. This is due to the *summation* property of the alpha-stable distributions where Cauchy and Gaussian is the most common. The Cauchy distribution, which has heavier tails than the Gaussian, was considered as source prior for monaural ICA in my Masters Thesis [Dyrholm and Pontoppidan, 2002]. The summation property is that adding two independently drawn numbers from a Gaussian (or Cauchy) results in a sum that follows a Gaussian (or Cauchy) distribution with changed parameters [Conradsen, 1995]. Another way of looking at the summation property is that, convolving two alpha-stable distributions of the same family results in a new distribution of that family, i.e., it only changes parameter values. In contrast, adding two uniform stochastic variables results in a triangular distribution and thus the family changes. For deeper insight on alpha-stable distributions see Kidmose [2001].

In order to investigate what statistical independence and orthogonality constraints imply for the source separation and identification problem a small experiment with synthetic data is conducted. Figure 4.2 shows the results of a little experiment with two classes of data, where ICA is capable of discriminating the

two classes while principal component analysis (PCA) is not. PCA is crippled by its requirement that the estimated sources should be orthogonal, which also imply that the direction classes in data should be orthogonal. The ICA only requires the direction of the classes to be linearly independent. So while this is an example where ICA works and PCA does not, there is no guarantee that the PCA will not work on data available, as this is solely governed by the properties of the data. The example did reflect one assumption of how the acoustic emission energy (AEE) signals in normal and faulty examples differ. Simply we might have a strong event as combustion where the increased wear between piston and liner “drown” in the combustion mode variance. However, at another angular position with no prominent events the increased wear will show up. In the two dimensional case where all points are in one quadrant (not as Figure 4.2) subtraction of the mean (centering before PCA) will save the day for PCA, as the normal and faulty end up on each side of the origo. However when the mean is already zero as in Figure 4.2 the centering does not help PCA.

The ability to distinguish between the two synthetic classes can also be transferred to real data acquired on the MAN test bed engine. The whole data set outlined in Figure 2.1 basically have 5 different conditions: 25% load with and without oil, 50% load without oil and 75% load without and with oil. The temporary fault in the waterbrake at 50% load is ignored, since it is not as stable as the other faults. The mean field independent component analysis (MFICA) and unsupervised Gaussian mixtures (UGM) methods are capable of grouping the examples based on load and oil level without any prior knowledge on the form and time of the changes, essentially like a human expert. Figure 4.8 show the estimated directions describing the individual loads and the *oil on/off* (as # 3) core signals. Especially the MFICA provide an *oil on/off* source that is gradually increasing and drops to the baseline after the lube oil is restored. With sufficient examples, a trigger threshold on the independent component could be established, which would imply the specific fault (modeled by the independent direction) had occurred. This is a standardized way to achieve fault detection with other models, e.g., Gaussians [Basseville and Nikiforov, 1993] and State Space Models [Gustafsson, 2001]. Unfortunately, the setup is not economically feasible for detecting increased liner-piston interaction or scuffing, as destroying numerous liners and pistons for large diesel engines is too costly.

## 4.2 Mean field independent component analysis

In the mean field independent component analysis (MFICA), the method for estimating the sources is derived from physics. The posterior distribution of the sources is approximated by a Gaussian - and the “optimal” estimate is the mean

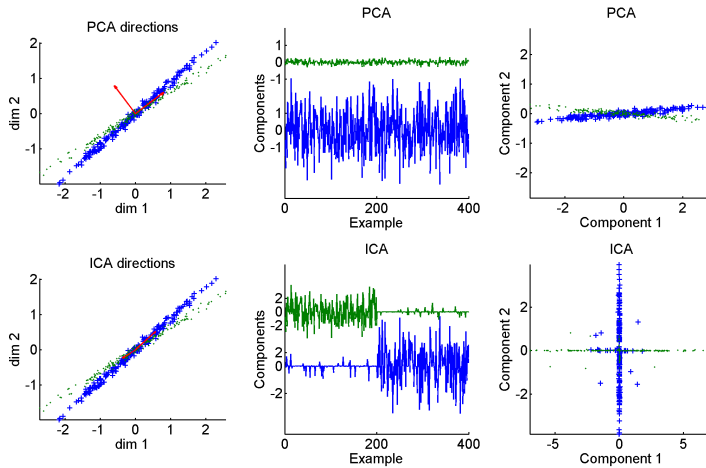


Figure 4.2: Two linearly independent classes are separated using PCA and ICA. The left figures show the data and the estimated directions. The middle figures show the estimated sources as time series (the two zeros indicate the zero level for each component). The scatter plots of the estimated components to the right show the ICA separation of the two directions in the observed data into two statistically independent components, while the PCA has just rotated the observed signals. The ICA scatter plots of the estimated components to the right display the independence as most examples are very close to one of the axes, i.e.,  $p(x, y) = p(x)p(y)$ .

of that Gaussian. The MFICA algorithm due to Højten-Sørensen et al. [2002] is implemented in a Matlab toolbox developed at DTU [Kolenda et al., 2002].

The MFICA allows for specifying the source distribution and constraints on the mixing matrix. When handling the non-negative AEE signals, it makes sense to specify an instantaneous non-negative mixing of non-negative signals, i.e., having non-negative elements in both source and mixing matrix. If the raw acoustic emission (AE) signals were used the non-negativity constraint should not be applied, since those signals take on both negative and positive values. In the toolbox the setting `prior.method='positive'` sets up a non-negative constraint on the elements of the mixing matrix using a Lagrange multiplier and the source prior becomes an exponential distribution (with scaling  $\beta = 1$ ).

For Gaussian observation noise the mean field approximation to the posterior distribution for component  $k \in [1, K]$  is the prior (Equation 4.10) times



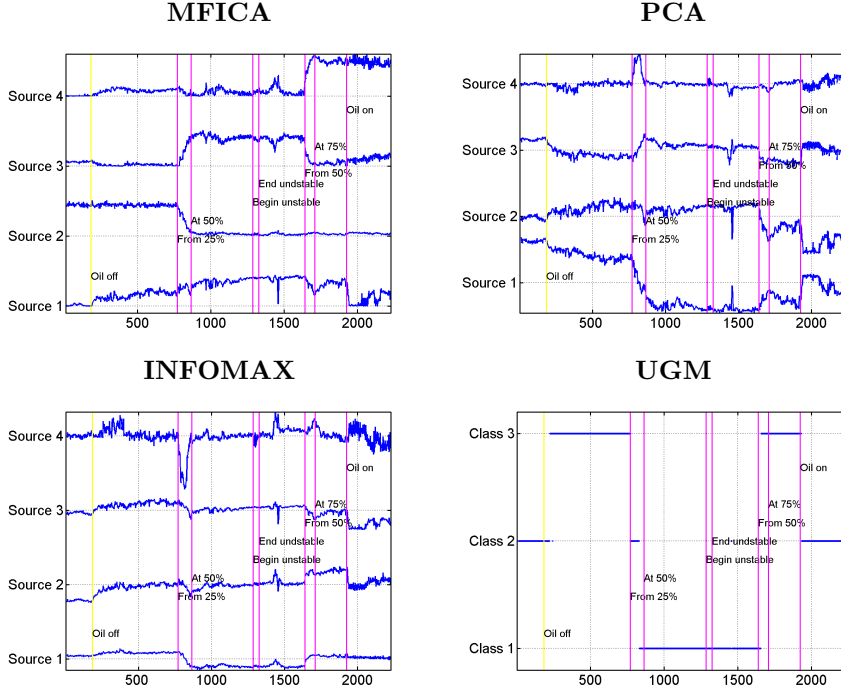


Figure 4.3: MFICA, PCA, INFOMAX and UGM applied to the data set with 3 loads and lube oil on/off outlined in Figure 2.1. The vertical lines indicate the condition changes, e.g., oil and load changes. For MFICA, PCA, and INFOMAX the four sources are plotted as time series (rows of  $\mathbf{S}$ ). The UGM shows the classification of the same examples into three classes. The UGM and INFOMAX are applied to the 4-dimensional PCA subspace. Especially the MFICA result is good, as the oil off period is modeled by source 1 that is increasing while the oil is off, and drops to zero as the oil is put back on. In addition, the UGM with 3 classes, detect the oil change and put the first and last examples in the same class. The PCA and INFOMAX components also change at the oil changes, but not in as clear and systematic way as with the MFICA and UGM.

a Gaussian with mean field parameters  $\gamma_k$  and  $\lambda_k$ .

$$P_q(s_k) = \begin{cases} \frac{1}{\beta} \exp -\frac{1}{\beta} s_k & , s_k > 0 \\ 0 & , s_k \leq 0 \end{cases} \quad (4.10)$$

$$P(s_k) \propto \exp \{-\lambda_k s_k^2 - \gamma_k s_k\} \quad (4.11)$$

$$\log \int P(s_k) P_q(s_k) ds = \log \left( -\frac{\sqrt{\pi} \beta \exp \frac{(\beta - \gamma_k)^2}{2\lambda_k} \left( \operatorname{erf} \left( \frac{\beta - \gamma_k}{\sqrt{2\lambda_k}} \right) + 1 \right)}{\sqrt{2\lambda_k}} \right) \quad (4.12)$$

The first and second derivatives wrt. the mean field parameter  $\gamma$  give the Equation 44 and 45 in Højten-Sørensen et al. [2002], that are used in the mean field algorithm [Kolenda et al., 2002].

In order to understand what we get from the MFICA, recall the two-dimensional example in Figure 4.2, and remember that the size of the matrices in Equation 4.2 are  $\mathbf{X}$ :  $d \times N$ ,  $\mathbf{A}$ :  $d \times k$ , and  $\mathbf{S}$ :  $k \times N$ . What we obtain is that the statistically independent components, the rows of  $\mathbf{S}$  can be interpreted as activations (coordinates) of the linearly independent directions in the observed data, i.e., the columns of  $\mathbf{A}$ .

Since the mixing matrix is not partial of an unitary matrix as with PCA, and the rows of the source matrix should follow specific distributions, the application of the model to new examples require iterative solutions similar to the training procedure, whilst keeping all but the source values to the values obtained from the training process.

The current available implementation of the MFICA algorithm uses an expectation Maximization (EM) algorithm to update and learn its parameters. The EM algorithm is a “coordinate” descent algorithm, in the way that the set of parameters are split in two groups. The algorithm switches between optimizing the parameters of the two groups with the other group of fixed parameters fixed to their previously optimized values. It can be proved [Bishop, 1995] that none of the steps will increase the bound. However, it also known that the convergence speed is slower compared to some other methods that also converge [Olsson et al., 2005].

### 4.2.1 Priors

When investigating the histograms of the estimated sources I noticed that, with small number of components the source estimates seemed to follow a gamma-distribution instead of the exponential prior-distribution. Obviously selecting an exponential source distribution does not necessarily result in exponential distributed sources estimates as they depend more on the observed data than the prior if enough information is available. However, the prior in this case influences the source estimates since they are not negative.

In other settings, with independent prior distributions having the maximal density at 0, the estimated sources normally lie on or very close to the axes, see for instance Figure 4.2 (lower to the right). But here the source estimates are out in the field, so to speak, which comply quite well with independent gamma-distributions with shape parameter  $k > 1$ . I guess the reason that the source

estimates are not one the axes, is that on the axis tells that the corresponding hidden signal is not used, and for a proper fit to the observed data, with few components, the algorithm needs virtually all hidden signals to model the observed data. While it is a speculative conclusion to draw, this could improve classification as the exponential source prior tells the algorithm to select low value sources if it is not well justified by data. This could help when applying the model to new examples, since the model is pushed towards modeling new examples as noise. However this is not investigated, and deliberately selecting a wrong prior is not that good Bayesian style.

#### 4.2.1.1 Gamma source distribution

Eventually the gamma distribution was considered as a source prior candidate within the mean field framework. The paper by [Højten-Sørensen et al. \[2002\]](#) gives the calculations that needs to be carried out in order to have the update formulas in the [EM](#) algorithm. Therefore, the straightforward idea was: Would classification improve, if the independent components (the sources) followed a gamma distribution.

The calculations in [Appendix I](#) deals with the Expectation step of the [EM](#) algorithm, i.e., how to get to the update formulas with source mean and covariance parameters that define the source signals, whilst keeping the noise variance and the mixing matrix fixed as described in the [EM](#) algorithm.

Unfortunately, the analytic result (given in [Equation I.2](#)) is not applicable since the Laguerre terms results in numeric overflow. It is not known whether this is due to missing normalization terms, bad approximations somewhere, or the results are just small differences between (ridiculously) huge numbers. However, it is observed that the analytic simplification that happens for  $\beta = 1$ , where the Gamma distribution becomes an exponential distribution leads to numerically stable update formulas given in [Equation 4.12](#). This indicates that it is not the full, but the temporary results that are overflowing, which does not help as the result we are looking is most likely buried way under the noise level of a sum of four  $10^{100000}$  numbers.

#### 4.2.1.2 Positively constrained Gaussian source distribution

In an attempt to verify that it is the temporary results in the computation with the gamma distribution, the positively constrained Gaussian distribution is considered. The computations with this prior, where the analytic calculations

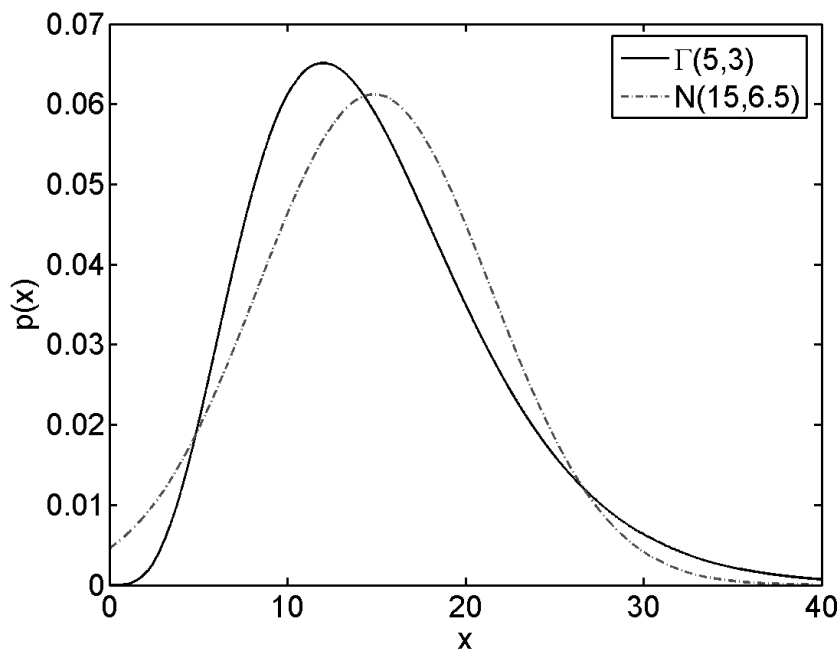


Figure 4.4: The similarity of a certain Gamma distribution and a normal distribution with a positive mean value

giving the  $f, df$  terms have already been done in [Højten-Sørensen et al. \[2002\]](#). The mean and variance was fixed to mimic a gamma distribution as shown in [Figure 4.4](#). Due to the current limitations in the implementation of the MFICA algorithm, this mean and variance are not optimized, and further selecting a lower mean value result in that the optimization diverge and stops. As shown in [Figure 4.5](#) the NLL on independent test examples obtained with this source prior is not as stable as with the exponential distribution.

### 4.2.2 The transposed problem

The setup given in [Equation 4.2](#) does not result in statistically independent AEE components. Instead statistically independent activations of linearly independent AEE signals are obtained. With this setup, the size of the mixing matrix does not depend on the number of examples, moreover the likelihood that is obtained from the MFICA algorithm is calculated pr. column in  $\mathbf{S}$ , i.e. pr. example in the normal setup.

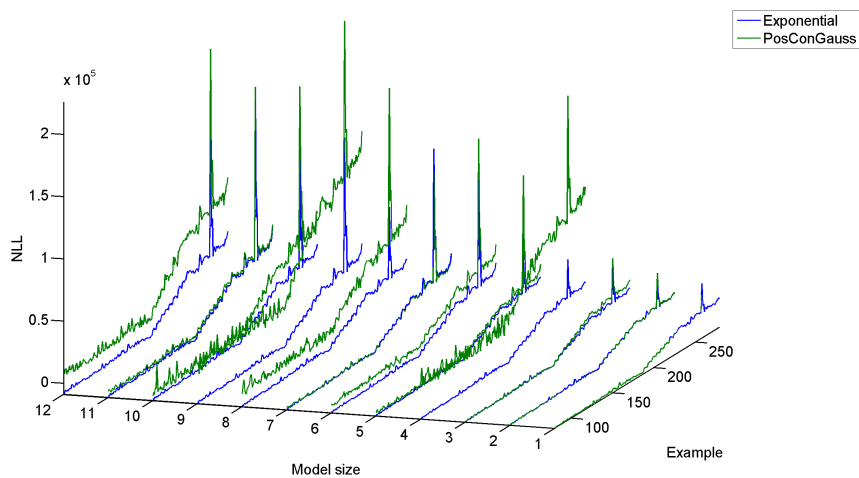


Figure 4.5: Negative Log Likelihood with Positively constrained Gaussian source distribution. The NLL is obtained from an independent test set, and is generally better (lower) with the exponential source prior than with the positively constrained Gaussian.

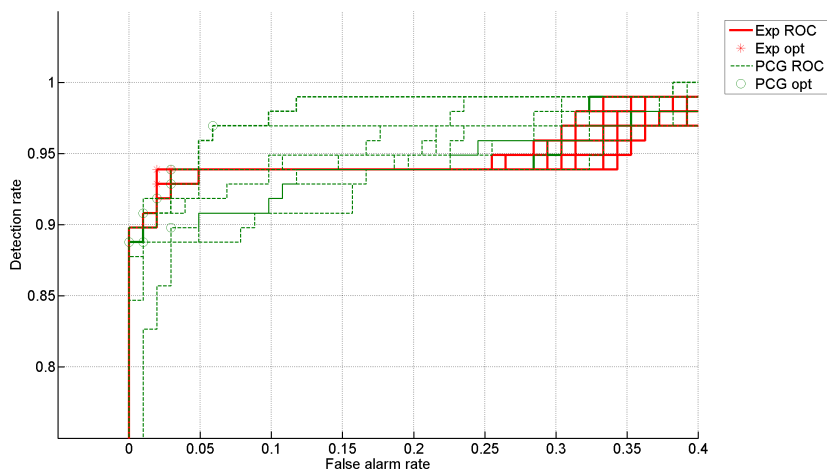
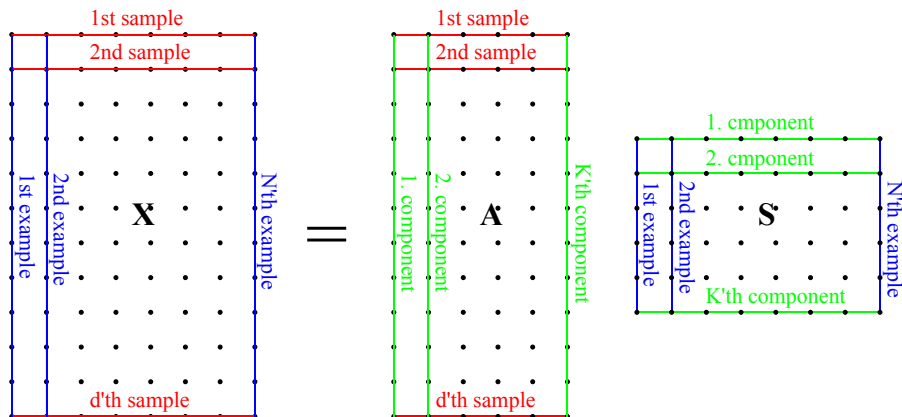


Figure 4.6: ROC curves obtained on an independent test set. Also the ROC show that the results with the exponential distribution are more stable for varying model sizes. The better performance seen here for the exponential distribution is not significant.

Normal:



Transposed:

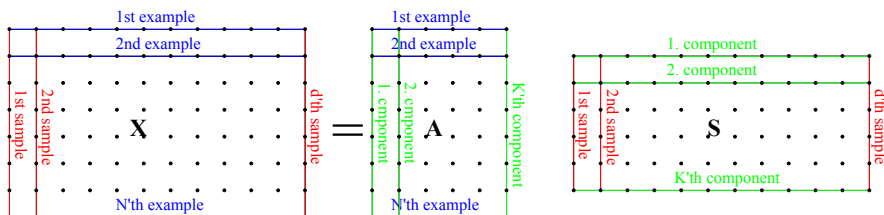


Figure 4.7: ICA in its two transposed forms

Still it is interesting to search for the statistically independent **AEE** signals, so the transposed problem is considered. Now  $\mathbf{X}$  is  $N \times d$ ,  $\mathbf{A}$ :  $N \times k$  and  $\mathbf{S}$ :  $k \times d$ , the independent components (the rows of  $\mathbf{S}$ ) are **AEE** signals and the rows of  $\mathbf{A}$  are the mixing of those components for each example. In the general case in this thesis with many more features than examples and components, i.e.,  $d \gg N > K$ , the proportionality of the matrices change considerably as in **Figure 4.7**. In the normal case the observation and mixing matrix are tall and thin  $d$  to  $K$  which is in the order of hundreds, while the source matrix is broad and low,  $K$  to  $N$  which is in the order 1/5 to 1/40. For the transposed case the two ratios are just exchanged such that the mixing matrix is still tall thin, but

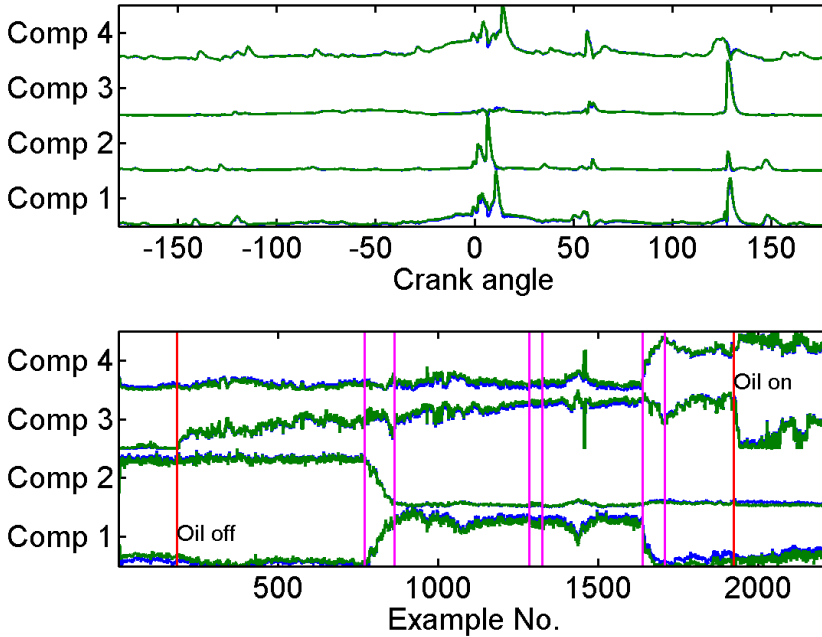


Figure 4.8: Applying MFICA to the transposed problem. Besides from the two inherent ambiguities with ICA scaling and permutation, the result is more or less an exchange of the mixing and source matrix, i.e.,  $\mathbf{X} = \mathbf{A}\mathbf{S}$  and  $\mathbf{X}^\top = \mathbf{S}^\top\mathbf{A}^\top$ . The scaling and permutation has been fixed manually. In the upper figure the columns of the normal mixing matrix is shown together with the independent components of the transposed problem. Moreover, the lower figure shows the independent components of the normal problem and the mixing matrix of the transposed problem.

with the lower ratio of 1 to 5 – 40, and now the source matrix become very long and low. The size differences are easily seen in Figure 4.7, the figures assume that  $d \gg N > K$ .

Does the transpose influence the results? First, the almost square problem with 2048 features (the crank angle samples pr. cycle) and 2227 examples is considered assuming 4 components. This is a repetition of the experiment conducted earlier, which revealed the three load classes + the additional oil on/off class. Figure 4.8 show the result of applying the MFICA to the transposed observation matrix compared to the experiment shown in Figure 4.3. ICA has an inherent scaling and permutation ambiguity, which has been manually fixed by normalizing the numerical range to  $\pm 1$  and alignment of the components.

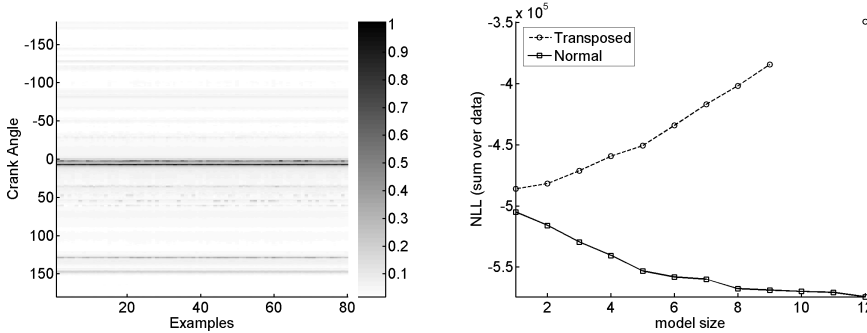


Figure 4.9: Comparing the signals (to the left) and **NLL** values (to the right) obtained with the normal and transposed form. While there is no difference in the recovered signals on an independent training set, the mean of the **NLL** values differ, with the normal form as the best.

Besides from these ambiguities the two versions give nearly transposed results, where the mixing and source matrices are just exchanged and transposed.

$$\mathbf{X} = \mathbf{A}_n = \mathbf{S}_n \quad (4.13)$$

$$\mathbf{X}^\top = \mathbf{A}_t \mathbf{S}_t \quad (4.14)$$

$$= \mathbf{A}_n^\top \mathbf{S}_n^\top \quad (4.15)$$

The noise levels and mean square residuals are on the same scale  $\sim 3 \cdot 10^{-4}$ . In addition, the transposed problem has a slightly better **NLL**.

With 80 examples in the training set and 80 in the test set as in Figure 4.9, the performance of the normal problem is much better than the transposed. The transposed model requires that the size of the new observation matrix is equal to observation matrix used to train the model (otherwise, the matrix setup does not hold). The **MFICA** algorithm runs faster for the normal problem, just as the optimization diverges when solving the transposed problem. This is for instance why there is no **NLL** with 10 and 11 components for the transposed setup in Figure 4.9.

For the normal problem the size of  $\mathbf{A}$  does not involve  $N$ , so that particular matrix setup ensure that a model obtained from a training set of some  $N$  examples can be used directly on observation matrices with  $M \neq N$  examples.



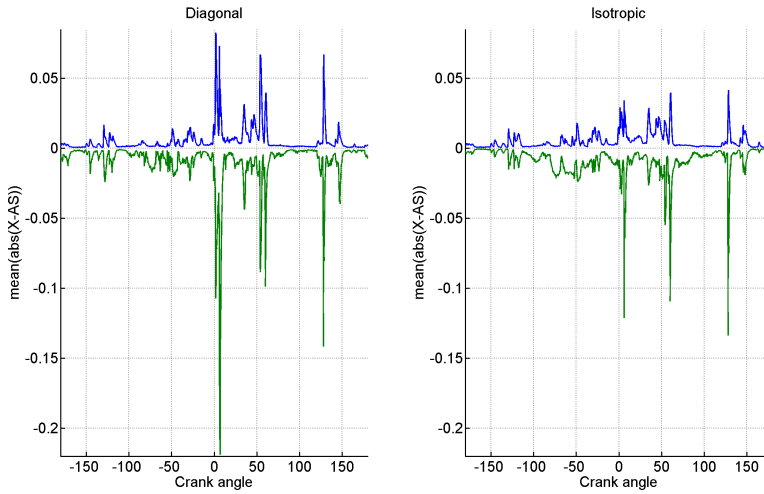


Figure 4.10: Comparing the mean residuals of normal and faulty test examples

### 4.2.3 Covariance structure

In the MFICA algorithm the structure of the covariance is assumed to be either isotropic (same noise variance for all  $d$  samples), or diagonal (independent noise variance for all  $d$  samples). While the later give  $d - 1$  additional parameters it also allows for modeling that the noise variance is not constant through out a cycle, but varying as a function of angular position. Moreover, as Figure 4.10 indicate the obtained residuals with both noise models vary considerably with the angular position. The diagonal noise variance leads directly to that some of the peaks in the residual becomes higher (also seen in Figure 4.10), and further this happens with both the normal and faulty test examples from experiment 1. In Figure 4.11 we see that the covariance structure has influence on the ability to detect faults, the separation of normal and faulty examples is better with the diagonal than both with the isotropic noise and for the PCA.

## 4.3 Principal component analysis

The PCA is a simple (at least when compared to ICA) and yet powerful method that finds the orthogonal directions in the observations with the highest vari-

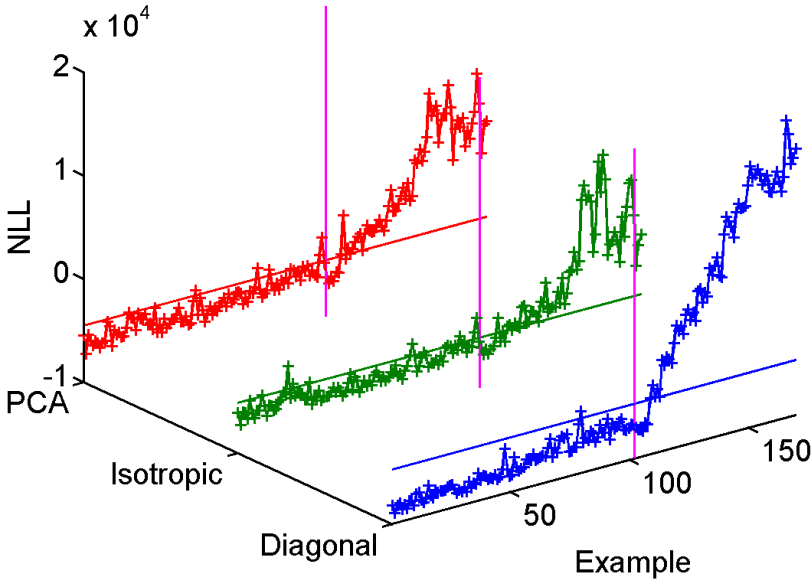


Figure 4.11: Comparing NLL values obtained with isotropic and diagonal noise covariance. The example numbers are arbitrary. The diagonal covariance allows for better performance as there is fewer of the normal examples that exceed the threshold.

ance. If a singular value decomposition (SVD) is applied to the observed data, i.e.,  $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ , then the mixing matrix  $\mathbf{A} = \mathbf{U}$  and the source matrix  $\mathbf{S} = \mathbf{D}\mathbf{V}^\top$  in Equation 4.2. The directions found in observed data with PCA are orthogonal/uncorrelated, which is a harsher but numerically simpler constraint than the linear independence required by ICA. Additionally, the PCA mixing matrix in its full size is a unitary matrix, such that  $\mathbf{A}^{-1} = \mathbf{A}^\top$ , for the economically sized SVD the pseudo inverse inherits that property so that:  $\mathbf{A}^\dagger = \mathbf{A}^\top$ . This makes the PCA models easily applicable to new examples as  $\mathbf{s}_{new} = \mathbf{A}^\top \mathbf{x}_{new}$ . When only a subset of the principal directions is used, the noise  $\boldsymbol{\nu}$  in Equation 4.2 is modeled by the discarded subspace.

The further difference between independent, orthogonal and uncorrelated is outlined in section 4.1 and some implications are encountered for the constrained PCA in subsection 4.3.1.

There is no likelihood model connected to the SVD algorithm, however since

the PCA is based on ranked subspaces, the discarded subspaces will span the residual, and a likelihood model of that can easily be obtained. The likelihood model given in Hansen and Larsen [1996] is adapted here. The noise is modeled by a white Gaussian noise source with variance equal to the sum of the variance in the remaining subspaces.

### 4.3.1 Positive PCA

Since the observed data is already positive, it could be interesting to try variations of PCA where the elements of the matrices are assumed non-negative. If we can specify PCA as a (constrained) optimization problem we could add the non-negativity constraint to obtain a special case of the PCA algorithm. First we need to analyze PCA to find those normal constraints.

The principal components are found along those pair wise orthogonal directions (columns in  $\mathbf{W}$ ) that explains most variance in the observed data  $\mathbf{X}$ . If we want to pursue that as an optimization problem, then

$$\mathbf{W}_{pca} = \arg_{\mathbf{W}} \max \{ \mathbf{W} \mathbf{X} \mathbf{X}^{\top} \mathbf{W}^{\top} \}, \quad s.t. \mathbf{W}^{\top} \mathbf{W} = \mathbf{I} \quad (4.16)$$

Without additional constraints  $\mathbf{W}_{pca}$  is equal to the eigenvectors of the sample covariance matrix  $\mathbf{X} \mathbf{X}^{\top}$ .

In positive PCA we also require that all elements in  $\mathbf{W}$  are non-negative, and this is problematic! Imagine some points in a three-dimensional space  $(x, y, z)$ . Having selected the first positive principal direction as  $(1/2, 0, 3/4)$ , now the second positive principal has to be  $(0, 1, 0)$  since non-zero elements in  $x$  or  $z$  dimension would break the orthogonality.

In general the non-negativity along with the orthogonality imply that only one principal direction can have energy in a feature, thus only one element in each row of  $\mathbf{W}$  can be non-zero. Therefore, while it is possible to specify and find a set of positive principal components, the combination of the two constraints is indeed problematic. The orthogonality constraint prevents the principal components from sharing dimensions, and prevent reasonable situations where the variance of one feature is connected to two other non-connected features cannot be modeled. A positive PCA algorithm was implemented in Matlab using the constrained minimizer `fmincon` with the given constraints but showed very small usability due to the above issues.

The non-negative matrix factorization (NMF) due to Lee and Seung [1999] estimates something that is similar to “positive” PCA, but it is not positive PCA as

such since it does not obey the orthogonality constraint. It has been much more successful than its “correct cousin” and achieved much attention. Given  $\mathbf{X}$ , the NMF estimates  $\mathbf{W}\mathbf{L}\mathbf{H} = \mathbf{X}$  where the sizes of the three matrices are equal to PCA, and the columns of  $\mathbf{W}$  and rows of  $\mathbf{H}$  have unit length and the elements are non-negative - however the columns are not orthogonal. Besides from not having an associated likelihood function, Donoho and Stodden [2003] have shown that NMF has problems when some parts are repeated throughout all examples, which is problematic with the highly repetitive engine cycles. They referred to problems with the repeated torso in matchstick swimmer figures. Hansen et al. [2005] states the repetition makes the components non-unique. The data that Lee and Seung [1999] initially presented NMF with was handwritten digits 0-9. Moreover, a subdivision of the parts in the digits 0-9 does not result in one part that is always on, especially not with handwritten digits. In Højten-Sørensen et al. [2002] the parts of handwritten examples of the digit 3 is extracted using the MFICA algorithm with same mixing matrix constraint and source prior as used here with the diesel engine signals.

## 4.4 Information maximization independent component analysis

The information maximization independent component analysis (INFOMAX) algorithm due to Bell and Sejnowski [1995] was not the first blind source separation algorithm, as similar methods had already been developed by Molgedey and Schuster [1994] and by Cardoso, Jutten, Herault, and Comon. The INFOMAX algorithm is based on estimating a transformation of the observed data that minimizes the redundancy between its outputs, which is equivalent to maximizing amount of information in the outputs. When the INFOMAX algorithm recovers the sources from the observed data it uses the inverse of the mixing matrix, and consequently it requires a quadratic mixing matrix, thus the number of sources and observations has to be equal. PCA is widely used to reduce the dimensionality before INFOMAX, and in this way the true mixing matrix is factorized into a non-square feature reduction part estimated with PCA ( $\mathbf{U}$ ) and a square part ( $\mathbf{A}$ ) obtained with INFOMAX. This result in slight modifications of Equation 4.1:

$$\mathbf{x} = \mathbf{U}\mathbf{A}\mathbf{s} + \boldsymbol{\nu}, \quad \boldsymbol{\nu} \sim N(0, \boldsymbol{\Sigma}) \quad (4.17)$$

where  $\mathbf{U}$  is the  $k$  component mixing matrix obtained with PCA, size  $d \times k$ ,  $\mathbf{A}$  is  $k \times k$  is the quadratic mixing matrix obtained by the INFOMAX algorithm,  $\mathbf{s}$  is the INFOMAX source matrix,  $k \times 1$ . Although the INFOMAX is noise-free, the observation noise is modeled the remaining components of the PCA. Further,

the noise spanned by the remaining components is necessary when comparing models of different sizes, e.g., when using algebraic penalty functions as Bayesian information criterion (BIC).

The quadratic constraint originates from the need of the inverted mixing matrix  $\mathbf{A}^{-1}$ , has been lifted and the framework extended to the non-quadratic case by among others [Lee et al., 1998, Lewicki and Sejnowski, 2000].

## 4.5 Unsupervised Gaussian mixtures

The UGM model [Larsen et al., 2002, 2001] is very different from the component analysis methods mainly considered in this thesis. Instead of finding linearly independent directions in the observed data, it looks for clusters of examples in the feature domain. Each cluster is represented by a mean value vector and a covariance matrix, the mean of the clusters is used to generate the signature AEE signals, and in that sense the UGM can also be seen as a generative model. In most cases the UGM is used after that PCA (or possibly ICA) has reduced the number of features (similar to the use with INFOMAX), due to computational performance. Initially it was tested if the unsupervised Gaussian mixture model could discriminate between the different conditions - it could as shown in the beginning of this chapter.

The UGM has previously been reported to be able to discriminate between different classes of mail - including spam [Larsen et al., 2002]. Moreover, it is capable of handling data sets with both labeled and unlabeled examples, i.e., a hybrid unsupervised-supervised setup. In this way, the labels on the possible few examples act as guidelines for the automatic discrimination of the remaining examples. In popular words, the algorithm group the examples based on their position, and then obtain the labels from labeled examples in the group. Furthermore if two classes are somewhat overlapping the labeled examples can be used to select a more optimal threshold, by balancing the width of the two clusters to the actual position of the examples with labels. Another important aspect of the UGM is the hierarchical clustering principle also reported in Larsen et al. [2002, 2001]. Initially, the two clusters with most overlapping probability densities are linked together, now they can be thought of as being one. Next, the two second-most overlapping clusters (regarding the new cluster as one) are linked. Besides, from allowing more advanced decision boundaries between classes, the hierarchical clustering also acts as taxonomy, since an example can be traced from its highest level to the very narrow description. E.g., {spam, money, Nigerian Scam} and {not spam, paper call, conference}.

With the available engine data, that assumingly consist of 5 classes (25%  $\pm$  oil, 50% - oil and 75%  $\pm$  oil), the hierarchical clustering could initially connect the classes based on the load or the oil setting. In the experimental results shown in [Figure 4.3](#), the UGM is the only model besides from MFICA that put the two periods with lube oil into the same class.

The Gaussian mixture model could also be used as more advanced classification scheme compared to the one-dimensional threshold considered in this thesis. In that experiment only the principal components where considered as inputs, but also the [PCA-NLL](#) (or from ICA) could be used. With the relatively good performance achieved with the simple thresholds, it has not seemed beneficial to go further with the [UGM](#) in that sense.

## 4.6 Simpler methods

For the detection of increased friction between liner and piston, the mean value ought to be a good measure. If the friction increases the Overall level increases, then the mean increases. [Goodman \[2004\]](#) describe how the baseline level can be used to dose the lube oil for a bearing. If the level is 8 dB above the established baseline, the bearing needs lube oil. When the level is back at the baseline, no more lube oil should be applied. Over-lubrication can cause to increased pressure, heat, stresses etc. all leading to failures [[Goodman, 2004](#)]. A similar idea was proposed by MAN B&W as a means to decrease pollution by minimizing the amount of lube oil based on the condition monitoring output. Most likely the effect of over-lubrication is not quite as severe with the large diesel engine as with the smaller bearings.

The classification performance reported [Table 4.1](#) was calculated from resampled data sets, consisting of several loads where with both normal and faulty data. The fault was induced by shutting down the lube oil supply to the cylinder. The thresholds and the models were only obtained from *normal examples* without labels, and then applied to other examples with and without lube oil. In the stationary case, where each load is modeled independently, the mean-value model is performing better than the PCA and ICA models. This is also the case when the observed signals are preprocessed with the full event alignment, i.e., when the variations due to load changes are removed. However, when all load settings are used to train a single model the performance of the mean-value classifier drops considerably.

When deciding on using PCA/ICA or mean-value one has to chose between an approach where you model the load changes followed by a simple model (the

	Load dependent	All loads	Time alignment	Event alignment
MEAN	0.98	0.71	0.71	0.98
PCA	0.97	0.89	0.95	0.96
MFICA	0.95	0.90	0.94	0.95

Table 4.1: AUC performance measures reported in Pontoppidan et al. [2005a]. The first column reports the performance when handling different loads independently, i.e., one model/threshold for each load. In the second column, the load is not used, so it is one big model for all loads. In the third column, the timing of the events has been aligned, and in the fourth column, the amplitudes have been adjusted too. With the event alignment, the performance is very close to handling the loads independently.

mean value), or apply a more advanced model to the bulk of data. In this case, the best performance is obtained using some modeling of the load setting (either different models or event alignment) and then a simple mean value model before classification. The next question is how does this work with other types of faults? It depends on the fault type. If the new fault type does not give rise to an overall increase, the mean-value will not work. The faulty water brake, labeled “Experiment 2” in Figure 2.1, is an example of a fault that the mean-value model cannot detect.

## 4.7 Regions of acceptance on synthetic data

In order to test some assumptions on the behavior of the PCA and MFICA models a synthetic data set with following parameters is created  $d = 2048$ ,  $K = 2$ ,  $N = 40$ ,  $\sigma = 1/100$ . The two columns of the mixing matrix are linearly independent and non-negative. The two sources are drawn from statistically independent gamma distributions and finally iid Gaussian with variance  $10^{-4}$  is added. This constitutes the training set, which are used to train one MFICA and PCA model with 2 components each. With those two models 400 examples are generated in the same way, and the 95% percentile of the NLL with the two models is calculated. An example is accepted if its NLL is not exceeding the 95% percentile.

With the two models and rejection thresholds, it was tested which synthetic combinations in the parameter space  $s_1, s_2, \sigma^2$  where 6 out of 10 examples are accepted. A cube with a  $70 \times 70$  linear spaced grid from  $(0, 0)$  to  $(2, 2)$ , and 70 values of noise variance on a logarithmic scale from  $10^{-8}$  to  $4 \cdot 10^{-4}$  is considered. At each of the 3.5 million points, ten example signals are generated. The point belongs to the acceptance region if at least 6 of the 10 signals are accepted. The regions of acceptance as functions of the sources and noise variance are

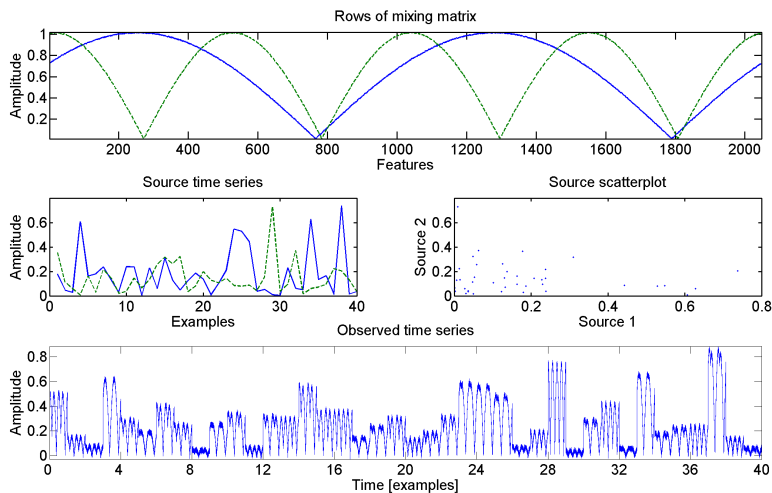


Figure 4.12: Synthetic data set used to train one PCA and MFICA with 2 components each. The first row displays the two columns of the mixing matrix, the second row show the 40 source examples, and a scatter plot of the two independent sources. The last row show the 40 observations (each consisting of 2048 samples) as one long time series.

displayed in figure [Figure 4.13](#).

The characteristics of the acceptance regions in [Table 4.2](#) are as expected. When the noise level is below the model noise level larger deviations in the source locations are accepted, as the model allows observation noise with variance similar to the training data. As the noise level, increases the acceptance region diminishes, and finally vanishes as the noise level exceed the model noise level by some factor.

Test set noise variance	Acceptance region
below model level	larger than training data
similar to model level	similar to training data
above model level	lesser than training data, vanishing

Table 4.2: Characteristics of acceptance regions as a function of noise level. The acceptance regions for noise level equal to the model noise level is shown in [Figure 4.14](#), all three regions are seen in [Figure 4.13](#).

The experiment show that two scenarios, that could indicate faults, leads to higher NLL values. When the sources move away from their normal area, it resemble faults where a known source becomes louder and louder. The other



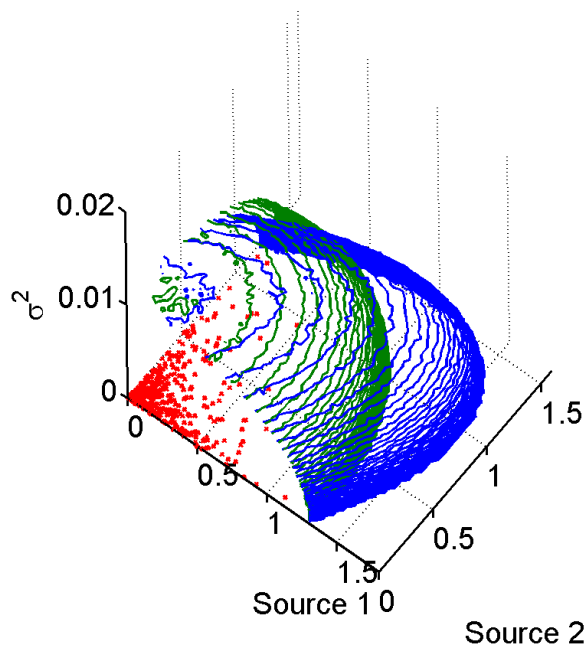


Figure 4.13: Acceptance regions for PCA (green) and MFICA (blue) as a function of the sources and noise level  $\sigma^2$ . The two shells are the contours of accepting at least 6 out of the 10 realizations. The points in the  $x, y$  plane are the source locations used to generate the 95% percentile classification boundary on the [NLL](#) (the added noise had variance  $10^{-4}$ ). The figure show that both methods reject examples that are noisier and those with source locations away from the *normal* area

scenario is when the noise level increases, possibly due to an almost constant friction source, or because a peak appears where nothing used to happen.

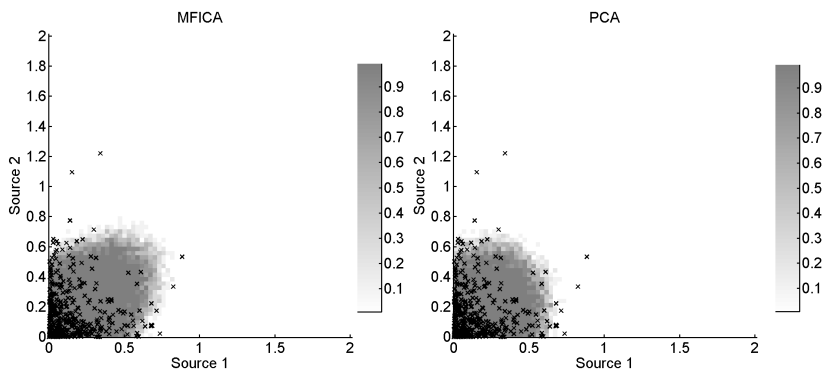


Figure 4.14: Acceptance region for data with same noise variance as the training data (a slice of Figure 4.13). The acceptance region for the MFICA looks like a rotated ellipsis while the acceptance region for the PCA is circular. The Elliptic form for the MFICA is suited at accepting the occasional points with both sources active.

## CHAPTER 5

# Performance measures and model selection

---

Given some data, we can come up with all kinds of models explaining it: Simple, complex, small, large, correct, good, bad etc. The question is: how do we robustly find the model that generalizes well, and what is the best model anyway.

The last question is the easy one: The right model is the true model that generated data, and we do not normally know it. In most cases, we will have to settle on the best model, and that question can be partially answered, as we can choose the best model among the available.

Model selection is making decisions on several levels, e.g.:

- Input parameters, e.g. Original, derived, time-domain, spectral, wavelets, crank-angle domain
- Model families, e.g. [ICA](#), [PCA](#), [UGM](#), [ANN](#)
- Set sizes, e.g. Number of components, number of clusters, and number of training examples
- Noise models, e.g. Gaussian, isotropic, diagonal, free, no-noise
- Parameter values, including hyper parameters

The decisions on each level do not have to be of the type: select the best. Additionally, selecting the best group is much more complicated. Actually, it is a complicated to select the best overall team, i.e., combining the decisions from the individual levels. Further the decision can be constrained such that trade-offs between performance and say computational complexity, memory consumption has to be taken into accounts.

An easily understood aspect of model selection is tuning the order of an interpolating polynomial. Let us assume that a polynomial of some order is a reasonable approximation. Polynomials of lesser order do not capture the full structure of the data; this misfit can be labeled as *bias*. Polynomials of higher order do capture the structure of the data, but also the noise, the misfit due to the noise can be labeled *variance*. If we examine the residual of some other points drawn from the same model, we will first encounter the *bias* regime where the residual decreases as the order increases. At some point the residuals will rise again - this is the *variance* regime. Selecting the optimal order is a trade-off between the bias and variance, and is the setting that gives the lowest residual on some other points from the same model. The effect that the model also learns the noise is called overfitting. A very similar example is given on page 12 in Bishop [1995]: The true signal is one period of a sine with added white Gaussian noise. It is best represented by a 3<sup>rd</sup> degree polynomial. Those with higher order than 3, capture too much of the noise.

Over-fitting also has some easily understood properties in condition monitoring. When trying to detect deviations from the normal condition, an overfitted model requires the observed data to be *exactly* like the training data, i.e., with identical noise. Otherwise, we get false alarms. In applications where the fault observations are also available for training (supervised systems) the similar problem exist; simply to recognize a fault we need exactly the same noise signals as in the training data, i.e., it will perform poorly on test data. Overfitting is poison for the condition monitoring system in a real world setup; because if we look upon the end users as being probabilistic learning machines, numerous false alarms will automatically lower their confidence in the system - to the extend where it is ignored, as in the tale of *Peter and the Wolf*. At MFPT'59 Galpin et al. [2005] presented examples on systems in service, that had been alarming for 10-15 months prior to breakdown - without end user interaction! Although unjustified ignorance is most likely to blame in this example, it show the existence of prior *anti-belief* working against such systems.

In the following sections, some methods for selecting the models are considered. First, some based on the residual errors, followed by methods based on classification performance. Although I have not added new concepts, models in this field, it is very important and necessary tool for reliable condition monitoring.

## 5.1 Learning paradigms

How do we learn? How do models learn? It is commonly stated that one should learn from one's faults. That is not particularly wrong for models either, since many parameter update formulas differentiate the mean square error with respect to model parameters to find hopefully global minima's. In addition, the models learn the parameters given the examples. Learning also comes from knowledge transfer, that you tell me that this is signal component is the fuel injection process. When the component changes I might also learn to differ between the normal and faulty sound profiles, possibly by someone telling me the difference. This is the difference between the unsupervised and supervised learning paradigms. In unsupervised setups, the model has to figure out the underlying group structure on its own. As the example given earlier, a bowl of fruit can be labeled as fruit or divided into apples and bananas. If we on the other hand tell the model that this is an apple and this is a banana, the model would adopt that classification. If the model is not told on which detail level we want the answers - it does not know. An interesting hybrid is pursued in [Larsen et al. \[2001\]](#), there only a fragment of the examples used for training are given labels, i.e., a mixture of unsupervised and supervised learning. Besides telling the algorithm, which detail level it should use, it also allows for cost savings since the labels are often hand labeled by experts and thus costly.

The challenge is that we are learning models from examples, and while we want to squeeze so much information on the true distributions or functions out of each example as possible, we don't want to learn the example it self. The learning paradigm is deeply connected to how the learning is evaluated. If the models are evaluated on the same examples as they were trained, there is a high risk of ranking an overfitted model above models that generalize well on independent test examples, and therefore we use the test sets.

## 5.2 Generalization error

Given a set of trained models and a test set independent of the training set, we will select the model that performs best on the test set. For a given model-family of increasing complexity, the ability to explain data will increase, i.e., the models underfit but become increasingly better, until the optimal complexity is reached. For models with greater complexity than this, the ability to explain the test decreases again, since the trained model has over-fitted to the training examples. [Figure 5.1](#) show this phenomenon, which is often called the bias-variance trade off [[Bishop, 1995](#)]

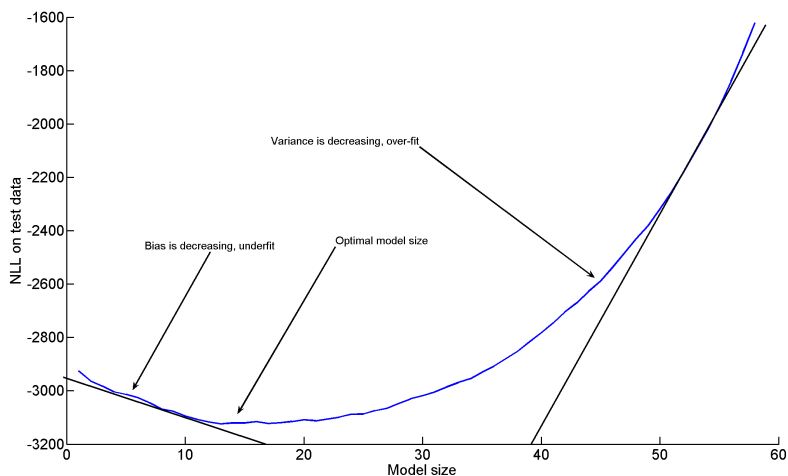
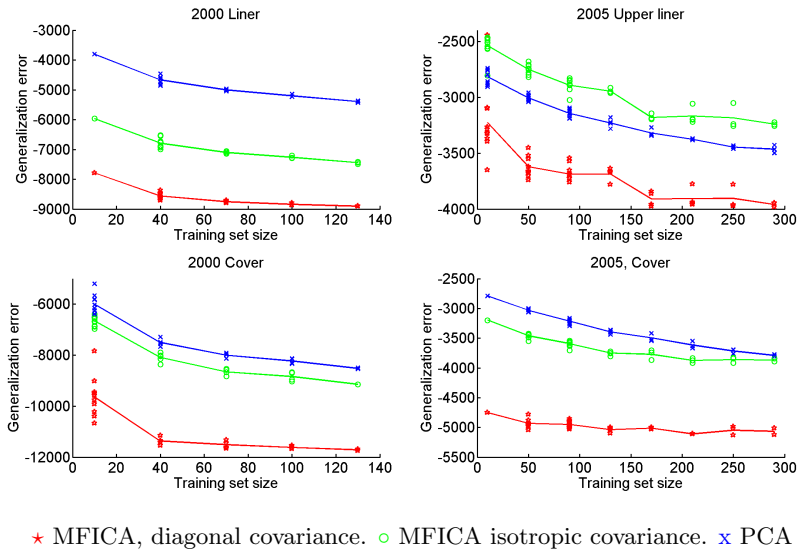


Figure 5.1: Generalization error, clear indication of the optimal model size - and the bias-variance trade off

### 5.2.1 Learning curves

Learning curves is a powerful concept that allows us to investigate the influence of the training set size on the performance. It is similar to model size estimation, where the performance as a function of training set size is investigated instead of the model size. Sometimes the learning curves possible reveal that with small training set size the simplest model win, and then when the training set size is increased a more complex model win. What happens is that the more complex model needs more data to learn its parameter values correctly. At some point (unless other things happen), the learning curve should level out, since the model does not gain additional information from the extra examples. Unfortunately, we have so few examples that it does not really happen. In the year 2000 data, there are 200 examples with normal conditions, whereas the 2005 data have 400 examples with normal conditions. With the old data, the learning curves have not really leveled out yet, although the improvements do decrease. It is best seen in the PCA curves as the MFICA algorithm needs excessive computational time to converge to *eps* tolerance. I hope that the convergence speed of the MFICA algorithm is improved in the future, such that better learning curves can be obtained. The ranking of the learning curves is clear, MFICA with diagonal covariance structure is consistently better than the MFICA with isotropic covariance and PCA. In 3 out of the 4 cases the MFICA



★ MFICA, diagonal covariance. ○ MFICA isotropic covariance. × PCA

Figure 5.2: Learning curves for upper Liner and Cylinder cover signals in old and new dataset. The learning curves reveal the ranking of the methods when considering the modeling of the normal condition. The normal condition is best modeled by the mixing and source matrices obtained with MFICA and diagonal covariance. In the old data set (to the left), the MFICA with isotropic covariance is better than the PCA. However, with the newly obtained examples the PCA is better for the liner channel.

with isotropic covariance is better than the PCA, but not for the upper liner signal in the new data set. As a comparison the learning curves for classification performance is given in [subsection 5.4.4](#) – and they tell a different story.

## 5.3 Penalty methods

Estimating the generalization error from the training error by adding a penalty term that grows with the complexity of the model has also been heavily investigated. Among the best-established methods is the Laplacian [Minka, 2001], and Bayesian information criterion (BIC) [Schwarz, 1978]. Also the Akaike’s information criterion (AIC) due to Akaike should be mentioned here.

The idea is to approximate the generalization error by adding a term to the log likelihood that scales with some function of the model complexity. The number of parameters might not be equal to the number of elements. For principal com-

Method	Penalty (pr. set of size $N$ )	Normalized penalty (pr. example)
AIC	$\dim(\Theta)$	$\dim(\Theta)/N$
BIC	$\dim(\Theta) \log N/2$	$\dim(\Theta) \log N/(2N)$

Table 5.1: Penalties for AIC and BIC where  $N$  is the number of training examples,  $\Theta$  is the full set of parameters in the model. The penalty is added to the negative log likelihood and subtracted from the (positive) log likelihood.

ponent analysis (PCA) it is not number of elements in the estimated covariance matrix but the number of free parameters – when respecting that the covariance matrix is symmetric [Hansen and Larsen, 1996], eqv. to that the transformation matrix lives on a Steifel manifold [Minka, 2001].

### 5.3.1 Bayesian information criterion

The Bayesian information criterion (BIC) approximates the Laplacian approximation, where only terms that grow with  $N$  are kept [Minka, 2001]. The BIC can be used with models that are associated with a likelihood function that can be compared across model sizes. In this thesis framework, that is PCA, unsupervised Gaussian mixtures (UGM) and independent components analysis (ICA) models: information maximization independent component analysis (INFOMAX) Molgedey-Schuster’s ICA, and mean field independent component analysis (MFICA). BIC has previously been successfully applied to model selection with ICA [Hansen et al., 2001], and with PCA [Hansen and Larsen, 1996]. Further, Minka [2001] show that BIC is a suboptimal estimator of the model size with PCA, when the number of examples is not much larger than the number of features. In the following section I show that BIC with very few training examples suffer from a gross underestimate of the noise variance. The PCA model becomes *over confident*, such that the stepwise improvement in the negative log likelihood as model size increases, outperforms the BIC penalty and forces to it to select too large models.

As Table 5.1 show the AIC is similar to BIC but generally favors larger models as the penalty is smaller than BIC when the number of parameters  $\dim(\Theta)$  exceeds  $e^2 \sim 8$ .



### 5.3.2 Model selection with Ill-posed principal component analysis

In this section I will repeat the ill-posed experiment conducted in [Minka \[2001\]](#), where the number of examples ( $N = 20$ ) is much less than the number of features  $d = 1024$ . Further I will expand it by also allowing the [BIC](#) to be used with an estimate of the noise level obtained from an independent test set. The experimental data are the usual normal condition acoustic emission energy ([AEE](#)) from the MAN test bed with 25% load. This expanded experiment was conducted after some debate, on how to calculate the noise level estimate in [Equation 5.1](#) with  $K$  components.  $\lambda_n$  contains the  $N$  squared diagonal of the SVD eigenvalue matrix divided by the number of features.

$$\hat{\sigma}_K^2 = \frac{1}{M - k} \sum_{k+1}^N \lambda_k \quad (5.1)$$

In [Hansen and Larsen \[1996\]](#)  $M$  is the number of observed features  $d$  and in [Sigurdsson \[2003\]](#)  $M$  is the number of examples  $N$  - and  $M = d$  is correct! The use of  $\hat{\sigma}_{K,N}^2$  results in a BIC curve with a local minima, before it increases and eventually drops again. The last drop is due to the last very small eigenvalues. With  $\hat{\sigma}_{K,d}^2$  (the lower curve) the BIC curve never increases as the fit gets so good that the algebraic penalty cannot balance that out. With the more accurate noise level estimate obtained from an independent test set, the expected BIC curve with a global minimum is obtained - we will return to that shortly. The negative log-likelihood ([NLL](#)) for a test set (i.e., generalization error) using the three noise level estimates is shown as lower left figure in [Figure 5.3](#). The estimates using the test error and the overestimated noise level leads to wrong conclusions, only using the underestimated noise level to calculate the [NLL](#) leads to model size comparable to the other estimates. Simply the generalization error show how the estimates from the training set works on the independent test set, and if additional information from another independent test set is provided, i.e. better noise level estimate, then we are not able to detect the overfitting any longer.

With additional training examples the problem persists and in [Figure 5.4](#) with 80 training examples the BIC selects to few components with the overestimated noise level, too many with the underestimated noise level. The closest estimate to the generalization error is the BIC with the noise level estimate from the independent test set. However if an independent test set is necessary for obtaining a proper noise level estimate in order to select the model size, the generalization error which is not an approximation and also uses a test set should be preferred. In either of the cases does the more accurate Laplacian model selection scheme [\[Minka, 2001\]](#) give better estimates than the generalization error and the *test*

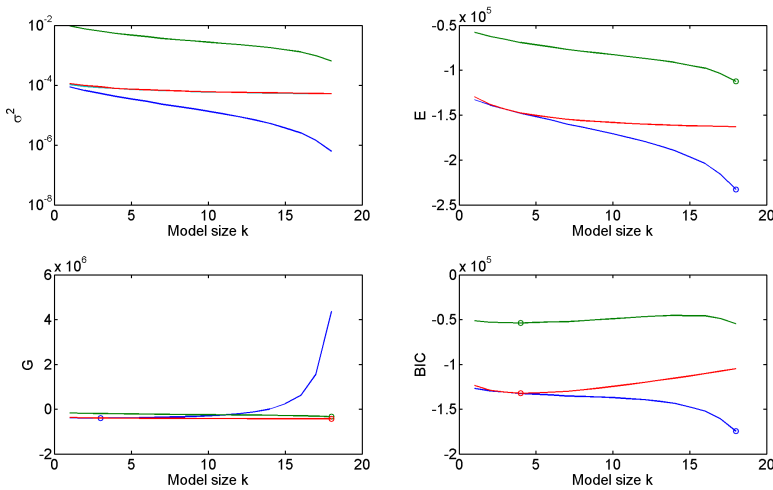


Figure 5.3: Model selection with fewer examples than features ( $d = 2048, N = 20$ ). The selected model sizes are given below. In all figures green curves are using the overestimate  $\sigma_N^2$ , blue using the underestimate  $\sigma_d^2$  and red the test estimate  $\sigma_{test}^2$  of the model noise variance. In the upper left figure  $\hat{\sigma}^2$  for  $M = \{d, N\}$  as well as the mean square test and train error as a function of the model size  $k$  is shown. The barely noticeable cyan curve is another test estimate of the noise variance. The underestimate  $\hat{\sigma}_d^2$  follows the training error (as expected), whilst none of the estimates follow the test error (in the middle). The upper right figure repeat the same curvature for the **NLL** using the two noise level estimates from the training set as well as the test noise level estimate. The **NLL** for a test set is shown in the lower left figure show that the generalization error only increases as it should when the using the underestimated noise level. Finally, the lower right figure display the BIC curves. The estimated model sizes are given in [Table 5.2](#)

*BIC*. The example carried out in [Figure 5.4](#) show that the breakdown of the **BIC** with PCA can sometimes be prevented by selecting the model size as the first local minimum. However, this is not always working as seen in [Figure 5.3](#). Thus, we already knew **BIC** is not an appropriate model selection scheme for PCA, when the number of examples is much less than the number of features.

**Model size estimates**

Noise level	Estimator	BIC	Generalization error
underestimated	$\hat{\sigma}_d^2$	18	3
accurate	$\hat{\sigma}_{test}^2$	4	18
overestimated	$\hat{\sigma}_N^2$	4	18

The `laplace_pca` code due to [Minka](#) selects 7 components.

Table 5.2: Model size estimates for  $d = 2048, N = 20$ .

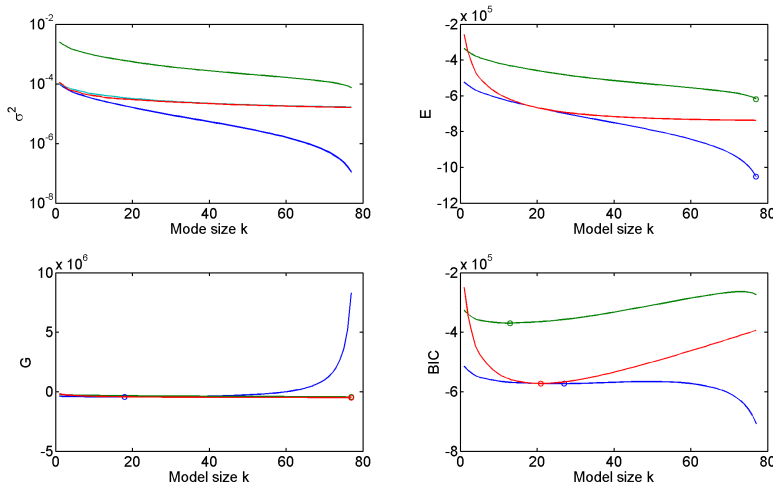


Figure 5.4: Same experiment as in Figure 5.3 but with more training examples:  $d = 2048, N = 80$ . In all figures green curves are using the overestimate  $\hat{\sigma}_N^2$ , blue using the underestimate  $\hat{\sigma}_d^2$  and red the test estimate  $\hat{\sigma}_{test}^2$  of the model noise variance. Still a better noise estimate is needed for the BIC to select a reasonable model size while. The BIC using the noise estimate from the training set as a local minima at  $k = 27$ , after a small increase it drops as the estimated noise variance approaches zero and the model collapses. The estimated model sizes are given in Table 5.3.

**Model size estimates**

Noise level	Estimator	BIC	Generalization error
underestimated	$\hat{\sigma}_d^2$	27/78	18
accurate	$\hat{\sigma}_{test}^2$	21	77
overestimated	$\hat{\sigma}_N^2$	13	77

The `laplace_pca` code due to [Minka](#) selects 32 components .

Table 5.3: Model size estimates for  $d = 2048, N = 80$ .

## 5.4 Supervised classification performance measures

Instead of measuring the residual on an independent test set, the classification performance, e.g., the false alarm and detection rate on an independent test set is considered. Since both normal and faulty examples are available and the models are given the distributions of the normal and faulty examples, the classification is supervised.

If one uses the log likelihood from generative model with a noise assumption, there is a close link between the residual through the log likelihood to the classification performance. Conceptually we are looking for models that have a low false alarm rate and high detection rate. Supervised classification is very well studied, the literature is rich and the theoretic results on separating two types with some specified distributions, can be obtained from sections on *hypothesis testing* or *test theory* from a standard statistics book, e.g. [Conradsen \[1995\]](#).

Hypothesis testing is a framework that addresses the two types of error in the two-class confusion matrix (anti-diagonal in [Table 5.4](#)). The rejection of a true hypothesis is called the *Type I error*, while the acceptance of a false hypothesis is called *Type II error*. Often the distributions of faulty and normal examples are overlapping, thus all thresholds will result in that some of the normal will be labeled as faulty (Type II) and vice versa (Type I). With some assumptions on the distribution type and parameters on the two classes, we can predict the number of false alarms and missed alarms as a function of the threshold. That also allow us to optimize the threshold wrt. our specific needs, i.e., if one of the types of error has greater economic, safety or environmental cost. In marine transportation and especially in aviation, the operation time threshold is put on the safe side. That means that after so many hours the component is considered faulty, thus the probability of a missed alarm is very low and consequently the false alarm rate, causing the replacement of healthy components, is very high.

### 5.4.1 Receiver operator characteristics

With both normal and faulty examples available, we can establish a relation between false alarm and detection rate. The receiver operator characteristics curve (ROC) displays the coupling of the false alarm rate and the detection rate as a function of the threshold and model for the two-class problem (normal/faulty). The worst class of classification models and thresholds are the ones where the false alarm and detection rate is equal, i.e., from reject to accept all

	Hypothesis is true E.g. cases with fault	Hypothesis is false E.g. cases with no fault
Accept hypothesis “Faulty”	$a$ , TruePositive (TP)	$b$ , FalsePositive (FP) Type II error False alarm
Reject hypothesis “Normal”	$c$ , FalseNegative (FN) Type I error Missed alarm	$d$ , TrueNegative (TN)

Table 5.4: Two class confusion matrix. The elements  $C_{ij}$  in the confusion matrix represent the number (or probability, then accuracy formula Equation 5.4 is invalid) of attaching label  $i$  to an element belonging to class  $j$ . The  $a, b, c, d$  indicate the variable names that is commonly used with calculations with the confusion matrix for the two class problem.

modes. While the end points correspond to very badly selected thresholds, the ones in between would be the result of identical distributions for the normal and faulty examples. In Figure 5.5 this is the straight line between lower left corner (marked  $\circ$ ) and upper right corner (marked  $\star$ ).

The perfectly selected model and threshold discriminates accurately between normal and faulty examples thus the false alarm rate is 0 and the detection rate is 1, indicated by the  $\square$  in the upper left corner of Figure 5.5. If the ROC is below the straight line that resembles pure guessing the reversed classifier should be used, simply the rejected examples should be accepted and vice versa. However if it was the output of some modeling based on assumptions - better check those assumptions!

Generally, we will take a collection of examples with known classification, and test whether the classifier outputs the known labels, those results can be reported in the Confusion matrix (see Table 5.4). The elements of the confusion matrix  $C_{ij}$  holds the number of examples classified as class  $i$ , but really belonging to class  $j$ , e.g.  $C_{f,n} = 7$  normal examples where labeled faulty. The From the confusion matrix the some relevant metrics can be calculated.

$$\text{Detection probability: } PD = \frac{a}{a+c} \quad (5.2)$$

$$\text{False alarm probability: } PFA = \frac{b}{b+d} \quad (5.3)$$

$$\text{Accuracy: (does not hold for probability rates)} \quad A = \frac{a+d}{a+b+c+d} \quad (5.4)$$

Given a confusion matrix (size 2 by 2) obtained from two independent classifiers opposed to a single classifier vs. ground truth, the  $C_{ij}$  element of the confusion

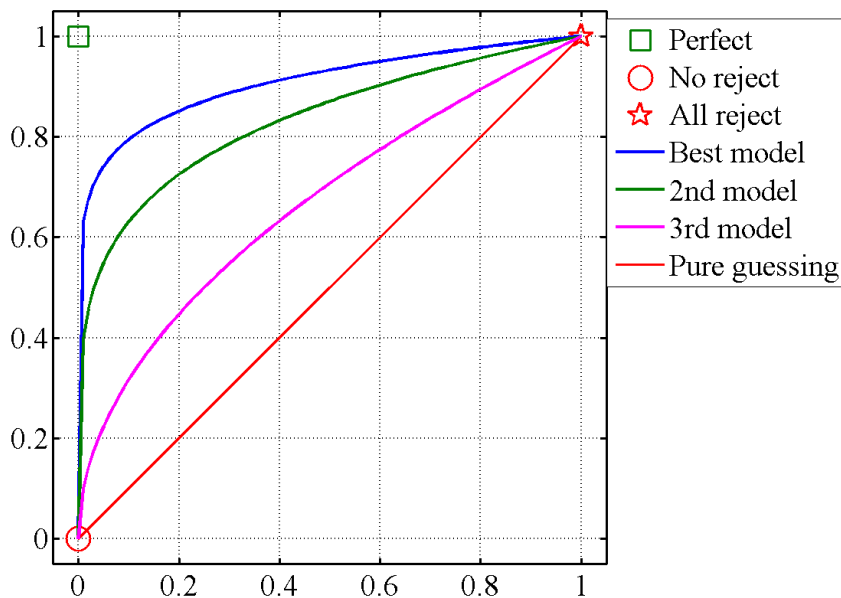


Figure 5.5: ROC curve sketch. The curves describe the coupling of false alarm rate and detection rate for 4 models. The blue is the best one, while the red is as the random guess. If the curves were mirrored in the diagonal it would indicate that the dual classifier should be used, i.e., accept the rejected and vice versa.

matrix holds the number of examples that classifier 1 put in class  $i$  and classifier 2 put in class  $j$ . In such matrix, the measures related to diversity and consensus can be calculated from the probabilities  $a$  to  $d$  [Whitaker and Kuncheva, 2000]. Further, Equation 5.4 only holds for probabilities if the number of faulty and normal examples are equal, otherwise the class with the fewest examples are weighted too much in that measure. With the detection and false alarm probability, the number of examples in nominator and denominator cancel each other

$$\frac{b}{b+d} = \frac{b/N_f}{b/N_f + d/N_f}.$$

## 5.4.2 Threshold optimization

In the following  $F(\tau)$  and  $D(\tau)$  denotes the false alarm and detection rate as functions of the threshold  $\tau$ . The threshold is the value that we use to

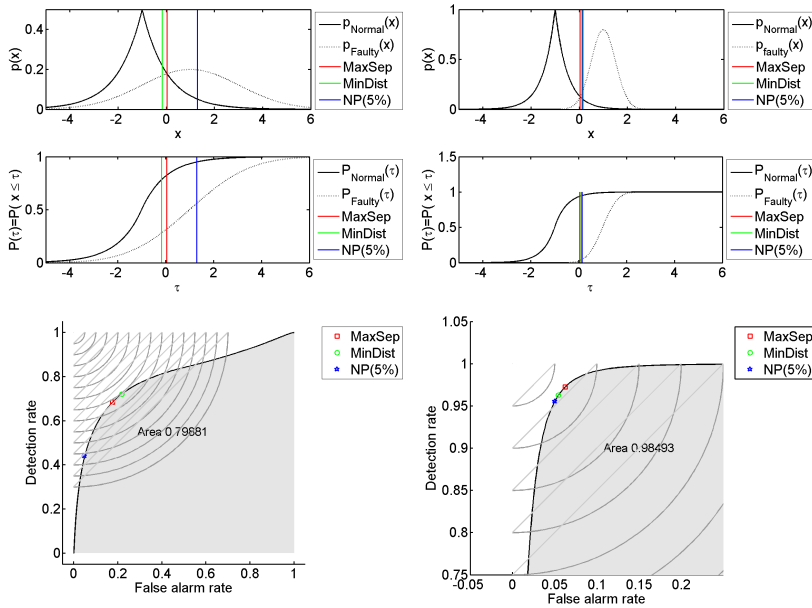


Figure 5.6: Two examples separation between two classes. The two columns differ on the overlap between the two classes. The upper panels show the densities, and the middle panels the cumulated densities, whilst the lower panels show the ROCs curves. The vertical bars indicate the thresholds found with the tree types of threshold selection methods: *Newman-Pearson*, *Maximal separation* and *Minimal distance*. As the separation of the two classes is increased in the right panels, the three threshold become almost identical (the right ROC is zoomed!). The straight lines are the contours of the maximal separation measure, while the arcs are the contours of the minimal distance measure. Notice that the thresholds might be close to the points where the two distributions intersect, but not exactly on.

discriminate between normal and faulty specimens. Here it is assumed that normal values have lower values than the faulty ones.

#### 5.4.2.1 Newman-Pearson criterion

The Newman-Pearson criterion states that we should select the model with the highest detection rate, whilst the false alarm rate is not exceeding  $\alpha$  [Scott and Silverman, 2004], as a constrained optimization problem it is

$$\tau_{NP} = \arg_{\tau} \max D(\tau) \text{ such that } F(\tau) \leq \alpha \quad (5.5)$$

The Newman-Pearson criterion does not take into account how the false alarm rate and detection rate evolve as a function of the threshold. Imagine that  $D(\tau)$  is almost constant, but never the less increasing, while  $F(\tau)$  increases (more than the detection threshold) from 0 to  $\alpha$ . The Newman-Pearson criterion chooses the point  $(\alpha, \beta)$ , then virtually identical detection could have been achieved with a lower false alarm rate, thus the overall performance has dropped. The two following sections cast threshold selection as an optimization problem with some additional measures, such that an optimal threshold in that respect can be obtained.

#### 5.4.2.2 Maximal separation

The threshold that maximizes the separation between two classes, is easily identified in the middle panel of [Figure 5.6](#), as the point where the cumulated density function (CDF) of the normal and faulty examples (solid and dotted curves) are furthest apart, i.e., the distance between  $D(\tau)$  and  $F(\tau)$  is maximal. This is also equivalent to minimizing the total amount of faults, which is  $1 - (D(\tau) - F(\tau))$ .

$$\tau_{MaxSep} = \arg_{\tau} \max\{D(\tau) - F(\tau)\} \quad (5.6)$$

The contours of this measure are straight lines parallel with the pure guessing line.

#### 5.4.2.3 Minimal distance

One issue with the *Maximal separation* optimization is that it does not select thresholds that are near the optimal classification (0,1). Anywhere on the contour lines is equally good.

$$\tau_{MinDist} = \arg_{\tau} \min \sqrt{F(\tau)^2 + (1 - D(\tau))^2} \quad (5.7)$$

The arcs in the lower panels of [Figure 5.6](#) indicate the equidistant points on the ROC wrt. the optimal classification point. This measure favors thresholds that that in general has equal number of false and missed alarms, and allows slightly more faults than the *max-sep* thresholds.

In the end the methods could be combined, allowing the *customer* to get the optimal performance within the specified requirements. It has not been tested but it would look like e.g.:

$$\tau_{CMB} = \arg_{\tau} \max\{D(\tau) - F(\tau)\} \text{ such that } F(\tau) \leq \alpha \quad (5.8)$$



It is also seen in [Figure 5.6](#) that the classification method must be quite good in, before it is allowed to do something “better” than the Newman-Pearson criterion in [Equation 5.8](#). However thresholds obtained with the maximal separation on the engine data have had lower false alarm rate than 5%, so it would be relevant to consider this combination.

### 5.4.3 Area under ROC curve

The area under the ROC (AUC) is also a direct measure of performance, since it is equivalent to the probability of identifying the faulty example when given one normal and one faulty [[Cortes and Mohri, 2004](#), [Hanley and McNeil, 1982, 1983](#)]. In [Figure 5.6](#) the distributions in the right column are more separated than those in the left column are, and accordingly the area under the ROC curve is higher (0.98 vs. 0.8). In order to see the variation in the AUC measures 200 points were drawn from each of the two densities used in the left column of [Figure 5.6](#). From those 200+200 examples the ROC was calculated, and for the 200 pairs the number of times where the faulty example had a higher value than the normal example was counted. The experiment was repeated 400 times, the results reported in [Figure 5.7](#), and showed that 95% of the measures are in the interval  $0.8 \pm 0.05$ . Further this allows for evaluation of the confidence intervals given in [Cortes and Mohri \[2004\]](#) with real data, that shows the confidence interval obtained with their method underestimates the variance in the AUC slightly, since the histogram of the ranking measures has slightly heavier tails.

### 5.4.4 Learning curves for ROC statistics

In [subsection 5.2.1](#) the influence on the ability to model the normal condition was investigated, here the ability to discriminate in a semi-supervised setup is considered. Only normal condition data has been used to train the models, and both known normal and faulty examples have been used to select the threshold that provides maximal separation between normal and faulty examples. Compared to the generalization error learning curves, the learning curve in [Figure 5.8](#) some interesting observations can be made. First on the liner signals the additional training examples cannot improve on something that is already virtually perfect, what we see is just more or less random noise. Moreover, we can see that this classification noise is slightly larger in the new data set, which comply nicely with the fact that this fault is only reduced oil and not oil shutdown. On the signals acquired on the cylinder cover, the models benefit from additional examples in the training set and better performance is achieved with the diagonal MFICA, followed by PCA. It should also be noted that in the mean, the

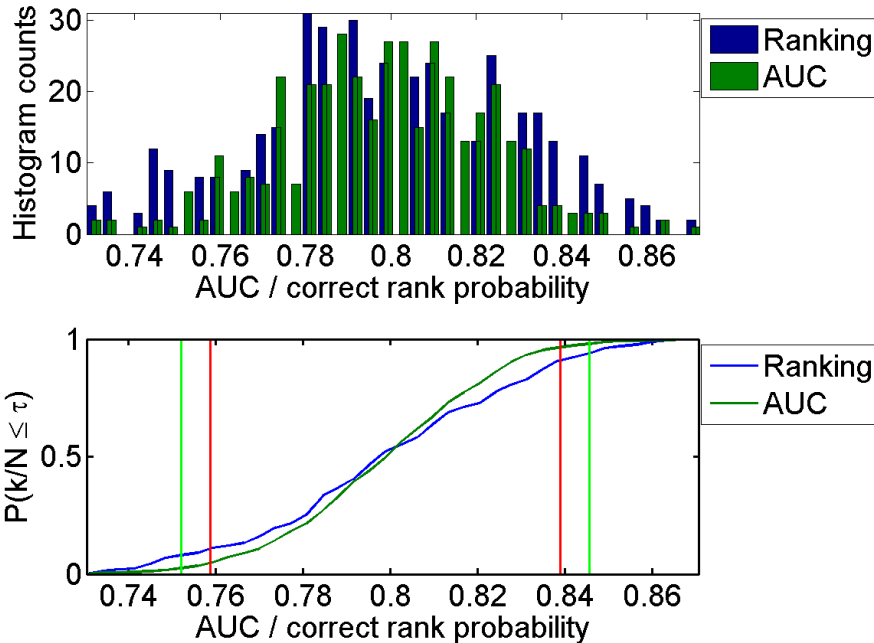
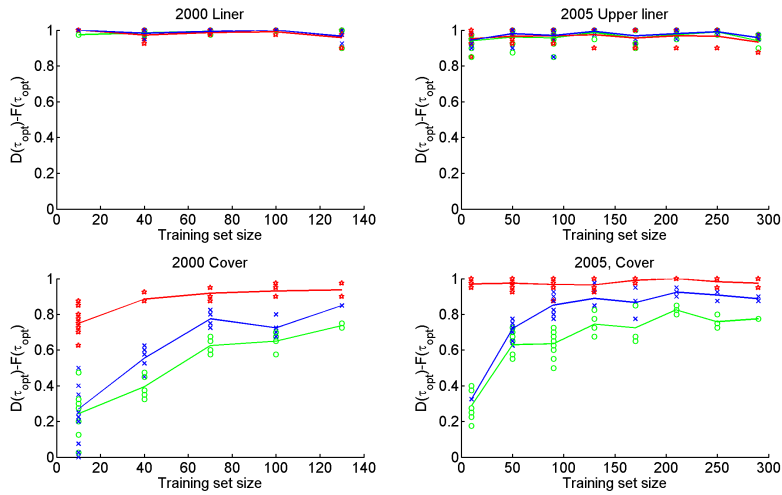


Figure 5.7: Confidence intervals on area under ROC curve. The vertical green lines indicate the 95% spread in ranking performance. The vertical red lines indicate the 95% confidence interval due to Cortes and Mohri [2004]. The (darker) green histogram and curve describe the density and cumulated density for the 400 calculations of the Area under the ROC. The blue histogram and curve describe the density and cumulated density for the ranking error rate calculated on the same data set. The Ranking error rate has slightly heavier tails and thus larger variance compared to the AUC.

ranking of the methods on the liner is PCA, isotropic and diagonal MFICA. However, that is only in the mean.

## 5.5 Unsupervised classification

If only normal condition examples are available, the model selection is solely based on false alarm rates, as the detection rate require access to faulty examples. Conceptually, this is similar to measuring the generalization error on an independent test set with normal examples.



★ MFICA, diagonal covariance. ○ MFICA isotropic covariance. × PCA

Figure 5.8: Learning curve with the maximal separation ROC measure.

Moreover, with an unsupervised system, all fault conditions are classified wrt. the same class, the normal condition class. Examples that are not normal are *faulty* and not identified as being either: Injection valve failure, increased piston/liner wear etc. Due to the nature of the data available in the AEWATT project consortium, unsupervised classification is of primary interest. It was believed that current methodology would not allow models to be transferred from engine to engine. For instance, the engines manufactured by MAN (or under licenses) are virtually unique, even though they might have same cylinder diameter, number of cylinders etc. Thus even two engines of the same type the acoustic emission (AE) signals would presumably not be identical, and furthermore Frances et al. [2003] have reported considerable variance from cylinders on the same engine. Therefore, an individual model is required for all combinations of engine, cylinder, and conditions. Thus for a real supervised setup all the faults that we want to identify should be induced on all engines of interest.

So without any faulty data examples, we will resort to train the models on training examples, obtain the NLL values from another set of training examples. Finally, use the say 95% or 99% percentile from yet another test set as the rejection threshold; with that model, we virtually set the sail! With such an approach, classification accuracy on known faulty and normal examples of 97% was achieved Table 4.1. Here we should also apply the existing knowledge on combining classification outputs, e.g., majority voting systems using PCA

and MFICA and data resampling. Since many have gone in that direction an alternative, approach is outlined in the next section.

### 5.5.1 Hypothesis testing

When a threshold on the **NLL** value is determined from the cumulated density function (or also from a **ROC**), we also know the percentile that we select also define the inherent false alarm rate with that threshold, i.e. we know that the  $x$  % of the normal examples exceed the threshold. Therefore, we add another modeling-layer, a binomial hypothesis test [Conradsen, 1995]. With this test, a new threshold, on the number of threshold crossings in a given window, can be calculated. For a window of 78 examples (as reported in Pontoppidan et al. [2005b]) and a 5% false alarm rate we would expect 4 false alarms. Setting the counting threshold to 10, meaning that 10 examples out of the 78 have to exceed the **NLL** threshold to generate an alarm on the next level, lowers the false alarm rate to 1% as seen in Figure 5.9. Obviously, setting the alarm threshold higher causes additional false alarms. Is this achieved without costs? No, this way we move the detection threshold towards the faulty examples; but we do not know how close or far they are from the normal examples. With an inherent false alarm rate of 5% we know that the **NLL** threshold is near the normal condition, and by allowing some false alarms the decision boundary becomes more elastic.

Also and this is important and has been seen that with signals from the test bed in Copenhagen, when the faulty occurs we are not in doubt, the alarm rate easily exceeds to 10 alarms in the 78 example window as seen in Figure 5.10. Further we loose the ability to detect small deviations, e.g., if the overall rejection rate rises to 6%. If we want to detect those slowly drifts we should also consider longer windows (in parallel to the short), as they estimate current rejection rate more accurate, and thus smaller deviations can be detected. Overall using the binomial hypothesis test allows for minimizing the false alarms, mostly at the expense of delayed detection of faults.

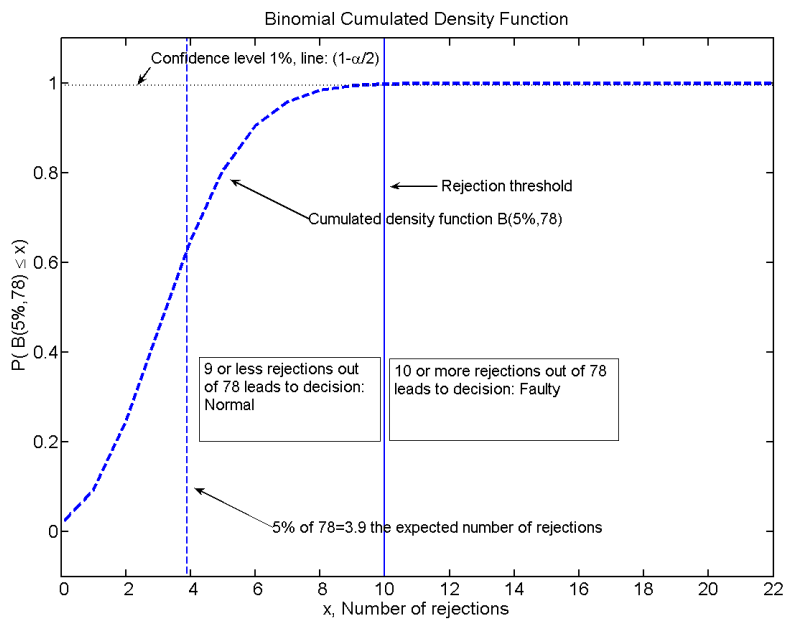
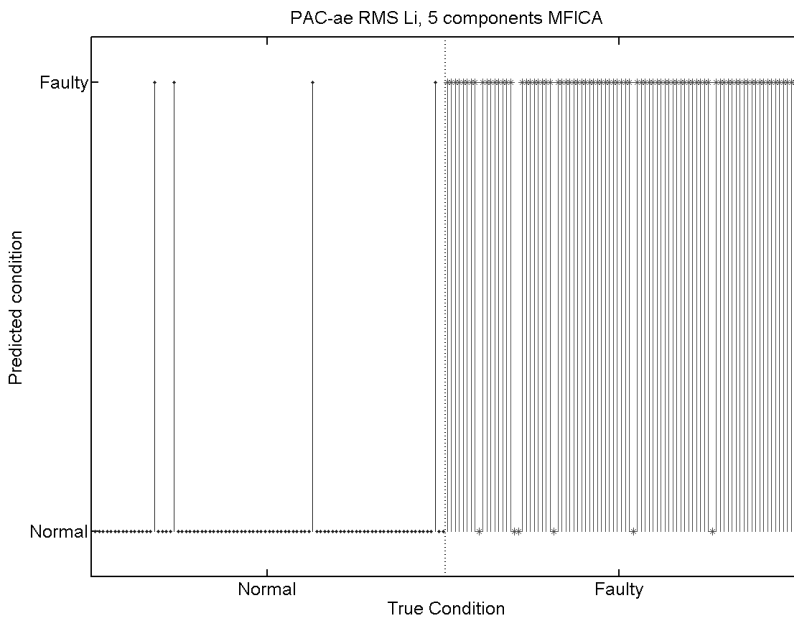


Figure 5.9: Cumulated binomial density for hit rate 5% and 78 examples in each window



*Figure 5.10:* Rejection rates in normal and faulty examples acquired on the Copenhagen test bed engine, processed with 5 components MFICA. The rejection rate for the faulty examples clearly exceeds the threshold of 10 examples.

## Discussion and conclusion

---

First, let us sum up the important conclusions from the previous chapters. The way the crank conversion takes place is important – is it an interpolation or a transformation. For the [RMS](#) signals I have settled on that it is a transformation, and ended up with summing the square of the time [RMS](#) signals between neighboring crank pulses, as this preserves the energy ranking of cycles. Since the crank conversion does not align all engine events on the same angular positions, a method that aligns them is developed. The method addresses both angular and amplitude changes, and provides a basis for non-stationary condition monitoring.

For modeling it is shown that [ICA](#) is superior to [PCA](#), both when comparing what the methods extract and how on good they model the observed signals. However, it is also shown that this does not necessarily imply that leads to superior classification performance – albeit this is in the case where perfect classification is already achieved. However, for cylinder the cover signals, that are less suited for the detection of the interaction component due to the structural damping, the better modeling also results in better classification as shown in the learning curves.

While it is apparent that simpler models than the [PCA](#) and [ICA](#) can detect the increased interaction, it is also my belief that the fault induced in the old data set is too easy to detect, and simply the deviation from the normal condition

is too large. Instead focus has been on modeling the normal condition and it's known changes due to operational condition changes, i.e., no assumptions on the size of the error has been made. This comes with the prices that the model does not know at which error level it blows up.

While the research have resulted in methods providing non-stationary condition monitoring system of large diesel engines, long term testing of the methods is still necessary in order to demonstrate a proof of concept. I think the remaining issues are more related to the false alarm rate than the detection of faults. With the limited data available we still do not know the true variability of the AEE signals. Obviously, this also influences the strength of ranking among approaches, since we do not know where we really are on the learning curves when considering the unknown long-term variations.

Since the new destructive experiment was carried out so late, it did not really influence the research. On the other hand, it provides a truly independent test case. Unfortunately there are some dramatic changes in the new AEE signals, when compared to the old data set, which is the data set considered in the thesis. The AEE signals are much noisier, possibly due to a) changes in the engine or b) that the new sensors are more sensitive to noise. In the period between the two experiments the injection valves have been updated, the engine control programs have been updated, such that the engine is delivering more power today than then. Determining whether the changes are due to engine changes or acquisition changes is going to be very important, as the quality of the data acquisition is one of the most important factors contributing the overall performance.

What we have seen is that the landmarks defined from the peaks in the AEE signals are much more fluctuating and do not have the same smooth structure as a function of load as before. One of the reasons for this could be the increased crosstalk between cylinders. This means that the timing changes occurring on the other cylinders, i.e., 90, 180 and 270 degrees out of phase, mix with the timing changes on the cylinder in question. This leads to situations where simultaneously occurring events (in angular domain) are pulling the apparent landmark in opposite directions, and possibly change the sequence of the engine events. Imagine, the engine control program delays an engine event a few degrees such that it passes a fixed event heard through cross talk - indeed possible. The additional crosstalk could arise from a couple of things. Either that the sensors is just more sensitive, thus picking up more signal. Alternatively, the sensor location compared to the old location provides lower damping wrt. other cylinders. Recall the damping of the AEE signals is increasing with frequency, thus lowering the high pass cutoff also lowers the damping wrt. crosstalk, thus the lowered high pass cut off<sup>1</sup> in the sensor picks up crosstalk that always were

---

<sup>1</sup>due to broader frequency range of the sensor it self – not to confuse with the pre-amplifier



---

there. The new data set raises an important question on how to continue with condition monitoring of large diesel engines. The question remains: Are we going to focus on detecting increased wear or continue to detect deviations from the normal condition?

If the modeling the normal condition option is selected, the level of sophistication should also be determined. Should it be a model localized in time, that continuously learns and updates the current condition, and consequently results in “false” alarms when operational changes occur. The false alarms could be minimized by comparing with ongoing operational changes - we’ll know that the probability of a false alarm given a operational change is high, and thus with that knowledge in mind, the probability, that the alarm was caused by a fault, is much less, i.e., *explaining away* theory [MacKay, 2003].

On the other hand, should it continuation of the path set out in this thesis, where the different operational modes of the normal condition are modeled. This way the condition monitoring system become invariant to the known operational changes, and the false alarms due to those changes are removed. This path requires more work than the first, as we need to learn those changes, which requires an investigation on how events move as a function of the load. Further how this affects the amplitude. This investigation is necessary for each engine layout considered. On the other hand, this information ought to be available in house for an engine manufacturer. I will also argue that a better understanding on how the AEE signals change on larger time scales is necessary.

With plenty resources possibly the supervised path could also be considered. Repeated experiments of the *path to scuffing* could allow for trending and possibly failure time horizons. Nevertheless, it requires that scuffing is achieved, and that the experiment is repeated a sufficient number of times. Still the variation of the normal condition, when moving to another engine or engine type should be investigated.

What unfortunately remains a question is how scuffing looks like? How does the engine behave prior to scuffing? Oil manufactures claim that wrong or no oil eventually leads to scuffing. In addition, it makes sense that the problems with lubrication leads to increased interaction, thus causing wear and damage where the piston and liner interacts. However, it remains uncertain if there are there other similar small faults that lead to this fault.

During the new destructive experiment, the AEWATTtoolbox was running *online*, processing measurements as they appeared. So the system presented in subsection 1.3.1 have been developed. The experiment revealed variations in

the AEE with constant load, which we had not observed before. The changes are just small angular movements of some of the events, and those changes must be addressed before the system can be applied online. Two methods can be proposed: One is to acquire normal condition data over a longer period, so the models can learn the true variation of the normal condition. The other method require the same amount of observations, but instead the variation could be handled by the allowing the event alignment procedure to adjust the locations of the landmarks (within bounds observed from the new large data set). Since it allows for  $\pm 1$  sample movements a simple test was conducted by considering a MFICA model with tri-Diagonal noise covariance matrix, (diagonal copy in the two sidebands). With that setup NLL of normal day 3, data approached the normal day 1 data, while preserving the distance (still in NLL values) to the faulty examples. While the test provided easily demonstrated what is necessary the approach is computationally costly and thus not interesting.

Even though it is just given as an example in the beginning of [chapter 4](#), the separation of signal components and automatic grouping the whole data set with the MFICA algorithm is important. Such expert like grouping can be referred to as cognitive components analysis and ICA has recently been reported to be able at that in other settings [[Feng et al., 2005](#), [Hansen et al., 2005](#)]. When implementing future condition monitoring systems, the usage of such clear components that follow the expected behavior of the engine could make the systems appear less *black box* in the eyes of the end users. They would know what the system is “looking” for.

The friction component obtained in that experiment is interesting on its own. The experiment also demonstrates the superiority of the MFICA to the INFO-MAX and PCA in such settings. As experiment with the other methods show: all models can detect that there are changes at same points; but only the MFICA result in four components that we can attach to the friction, 25%, 50% and 75% load. The experiment shows the strength of the MFICA algorithm, that it can separate the signatures that are independent of load and those that change with the load.

Looking forward I foresee the combination of some of the approaches. As we have become more confident with the properties of the large diesel engines, it is clear that the event alignment with one reference load is going to be problematic. During the design of the engine layout, a few points on the propeller curve are selected and the timing and etc. is optimized in those points. Accordingly, the event alignment could use those points as references and transform the signals into nearest optimization point. That would limit how much the warp should move the signals. Moreover, the condition modeling should benefit from the multiple loads as the MFICA in general work better when the signal parts are independent. This is clearly seen in the two examples given in the beginning of

chapter 4, where the model is able to separate parts out that depend on having a specific load and those that are fully independent of the load. Also for the future directions, inspiration from Air Canada should be considered. For several years, Air Canada been making a single recording of some hundred parameters during every take off and landing on their Airbus A320 fleet. By applying text mining on the maintenance logs, they were able to select the times where a system should have raised an alarm. From engine data collected the following year that kind of faults where foreseen with fair success, when compared to the actual replacement of parts reported in the maintenance logs [Letourneau et al., 2005]. Their success should be transferable to the marine propulsion application, and hints towards how the current acquisition system and proposed framework for condition monitoring could be integrated in a continuously updated health management system.



APPENDIX A

# Independent component analysis in large diesel engines

---

N. H. Pontoppidan and S. Sigurdsson. Independent components in acoustic emission energy signals from large diesel engines. *Submitted to International Journal of COMADEM*, 2005. URL <http://www2.imm.dtu.dk/pubdb/p.php?id=3885>



# Independent components in acoustic emission energy signals from large diesel engines

Niels Henrik Pontoppidan<sup>1</sup> and Sigurdur Sigurdsson<sup>2</sup>  
Informatics and Mathematical Modeling  
Richard Petersens Plads  
Technical University of Denmark Building 321  
2800 Lyngby Denmark

Submitted to International Journal of COMADEM

<sup>1</sup>corresponding author, [nhp@imm.dtu.dk](mailto:nhp@imm.dtu.dk)

<sup>2</sup>[siggi@imm.dtu.dk](mailto:siggi@imm.dtu.dk)

## **Abstract**

This paper analyses acoustic emission energy signals acquired under mixed load conditions with one induced fault. With Mean field independent components analysis is applied to an observation matrix build from successive acoustic emission energy revolution signals. The paper presents novel results that provide remarkable automatic grouping of the observed signals equivalent to the grouping obtained by human experts. It is assumed that the observed signals are a non-negative mixture of the hidden (non-observable) non-negative acoustic energy source signals. The mean field independent component analysis incorporates those constraints and the estimates of the hidden signals are meaningful compared to the known conditions and changes in the experiment. Most important is the estimate of the load independent wear profile due to the induced fault. The strength of this signature increases as the load progress and disappear as the induced fault is removed – this result has not been achieved with classical independent components analysis or principal components analysis.



# 1 Introduction

In the last two decades blind source separation by independent components analysis (ICA) have gained a lot of attention. ICA has been reported to separate speakers in mixtures [1], spotting topics in chat rooms [6], finding activation patterns in functional neuroimages [9] just to mention a few applications. Recently ICA was reported to provide cognitive groupings from observed signals without any prior knowledge of the true groupings. There Cognitive component analysis (COCA) is defined as the process of unsupervised grouping of data such that the ensuing group structure is well-aligned with that resulting from human cognitive activity [4, 2]. In this paper I show how similar results can be obtained from applying the mean field independent components analysis (MFICA) algorithm, due to Højen-Sørensen et al. [5], to acoustic emission (AE) energy signals obtained from a large diesel engine. The experiments show that the MFICA algorithm is capable of extracting a signal profile describing an induced fault and its development, which is not the case for the Information maximization ICA [1] and Principal component analysis. In this paper no performance numbers are given, instead the raw output of the ICA algorithms are provided as they speak for themselves.

## 2 Experimental data

Acoustic emission signals were acquired from the two stroke MAN B&W test bed engine in Copenhagen. The signals were sampled at 20 KHz after analogue RMS filtering ( $\tau = 120\mu s$ ) had been applied. Also the Top Dead Center and angle encoder signals were obtained, and the AE RMS signals were segmented into single revolutions (at bottom dead center) before domain was changed to crank angle. This results in signals with 2048 points pr. revolution as seen in

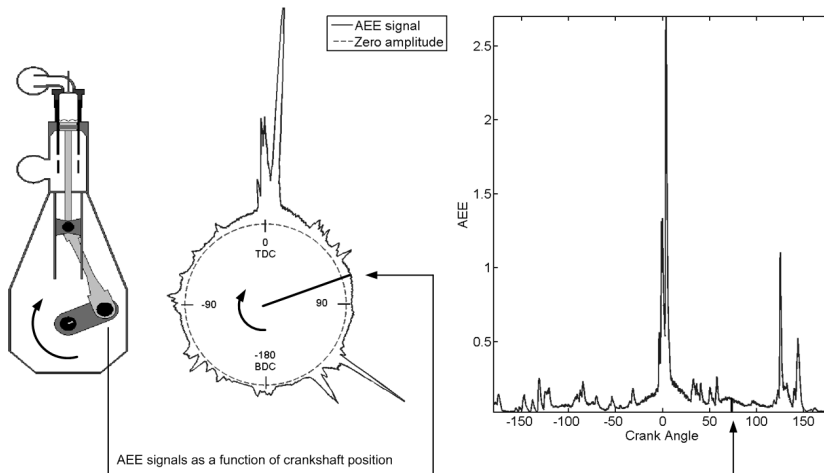


Figure 1: Signals sampled in crank angle domain

Figure 1. In this domain the engine related events are more or less appear at the same position in every cycle. At COMADEM 2003 signal processing removing those changes were introduced by the author [11], but that approach is not considered in this paper. During the experiment outlined in Figure 2 the operational conditions were changed by increasing the load on the propeller curve. Also after 180 revolutions at the lowest load the lubrication for the monitored cylinder was shut off, and in the end the lubrication oil was restored. Even though actual scuffing did not occur contact marks inside the cylinder liner was observed by inspection afterwards [3].

### 3 Modeling

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \boldsymbol{\nu}, \quad \boldsymbol{\nu} \sim N(0, \sigma^2 \mathbf{I}) \quad (1)$$

$$\mathbf{X} = \mathbf{A}\mathbf{S} + \boldsymbol{\Gamma}, \quad (2)$$

where  $\mathbf{x}$  is the observation vector of size  $d \times 1$ ,  $\mathbf{A}$  the mixing matrix of size  $d \times k$ ,  $\mathbf{s}$  the source signal of size  $k \times 1$  and  $\boldsymbol{\nu}$  is the additive (independent and

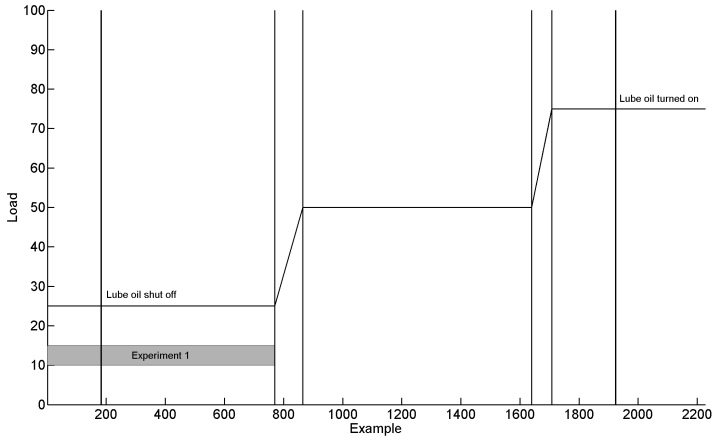


Figure 2: Time line for destructive experiment carried out with MAN B&W's test bed engine

identically distributed) i.i.d. Gaussian noise with variance  $\sigma^2$  also of size  $d \times 1$ .  $d$  is the number of features and  $k$  the number of components, and  $k \ll d$ . The noise is assumed to be i.i.d. Gaussian even though the RMS conditioning turns an uncorrelated zero mean additive noise component into a strictly non-negative noise component. However such noise model is not currently available with the MFICA algorithm. The MFICA algorithm differs from other ICA algorithms by allowing a broad range of source priors and mixing matrix constraints. For more information on the MFICA algorithm refer to [5] and [7].

The observation matrix  $\mathbf{X}$  is generated by stacking several realizations of the observation vectors. Here the different realizations come from different engine revolutions acquired with the same sensor. Simultaneously should be understood as at same angular position in this setup, and not as simultaneously recorded as the case in the classical blind source separation problems [1, 8]. Similarly the source matrix  $\mathbf{S}$  and the noise matrix  $\mathbf{\Gamma}$  comes from stacking the

$N$  source vectors and noise vectors.

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \quad (3)$$

$$\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_N\} \quad (4)$$

$$\mathbf{\Gamma} = \{\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \dots, \boldsymbol{\nu}_N\} \quad (5)$$

Equation 1 describe how the  $k$  hidden signals in  $\mathbf{A}$  are weighted by the coefficients in  $\mathbf{s}$  to generate the observed signal  $\mathbf{x}$ . In other words the  $\mathbf{A}$  matrix contain those signal parts that the observed signals can be made up from - it acts like a basis for the normal condition. The idea is to learn this basis set from a collection of normal condition data, making the model capable of generating the different modes in the observed training data. By applying the component analysis methods the orthogonal/independent directions in the observed data should result in a basis, i.e., columns in the mixing matrix, that contains signatures with the descriptive quality like source 3 (the third row of  $\mathbf{S}$ ) model the amplitude of the injector event signal in column 3 of the mixing matrix.

In Figure 3 the modeling of a normal and a faulty example (both at 25% load) is given. The source matrix reveals that the 2nd hidden signal models the normal condition part, while the 1st hidden signal models the additional part arising from the fault condition. However the two hidden signals are quite similar. The mixing matrix with the independent directions was estimated from 25% normal and faulty examples. We will later see much more difference between the hidden signals when two additional loads

### 3.1 Principal Components Analysis

The Principal components are obtained from the Singular Value Decomposition of the observation matrix  $\mathbf{X} = \mathbf{UDV}^\top$ . The 4 component mixing matrix is

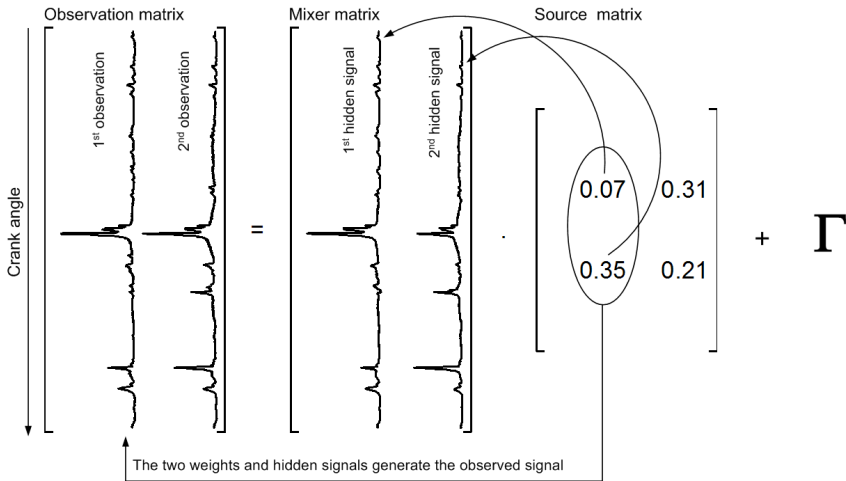


Figure 3: Data matrix setup. The first example is normal and the second faulty. The mixing matrix was obtained from observations at 25% load only

estimated as the four first columns of the left hand side matrix  $\mathbf{U}$ . The four source components are estimated as four first columns of the left hand side matrix  $\mathbf{V}$  weighted by the four largest singular values (and transposed). The method and matrix setup is further described in [10].

### 3.2 Information maximization ICA

The Information maximization ICA (IMICA) due to [1] require that the mixing matrix is  $\mathbf{A}$  square as the source estimates are obtained from  $\hat{\mathbf{S}} = \mathbf{A}^{-1}\mathbf{X}$ . This implies that the number of sources and observations should be equal, in this case 2227 sources! Often PCA is used reduce the dimensionality, such that  $\hat{\mathbf{S}} = \mathbf{A}^{-1}\mathbf{U}\mathbf{X}$  so actually the input to the IMICA is the 4 principal components shown in Figure 7. The method and matrix setup is further described in [10].

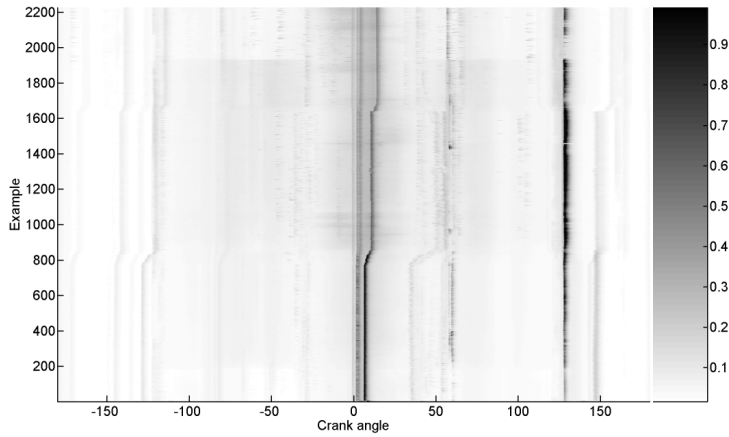


Figure 4: The full data set. The amplitude is color coded, i.e., the stronger the signal the darker the color.

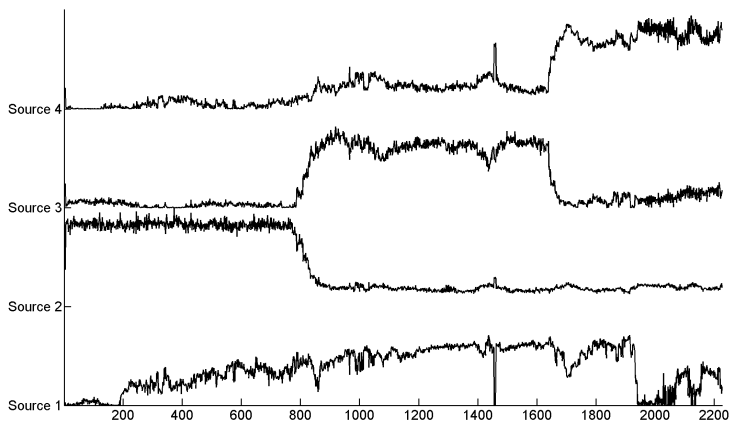


Figure 5: The independent components. Source 1 models the increased wear due to the removed oil. Source 2, 3, and 4 model the 25%, 50% and 75% load respectively. Changes in the source signals comply with the occurrence of operational changes given in Figure 2

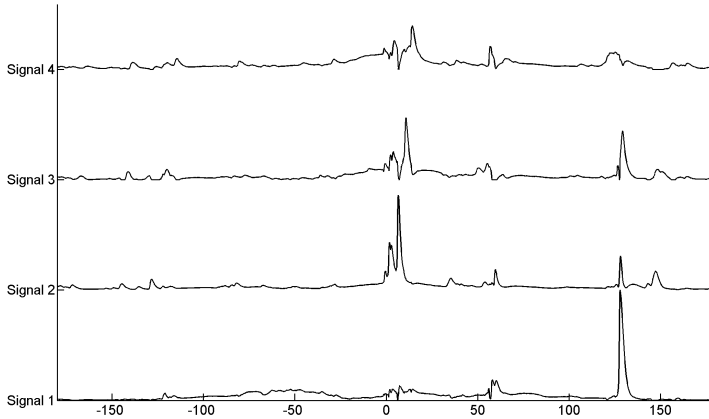


Figure 6: The hidden signals (columns of the mixing matrix  $\mathbf{A}$ ). The first one picks up the increased friction profile while the remaining model the normal condition at 25%, 50% and 75% load

## 4 Finding the increased wear signature

Now we consider the full data set shown in Figure 4 and apply the MFICA algorithm to estimate the hidden signals and the independent activations of those hidden signals. The only knowledge that the algorithm is given is that it is non-negative mixing of four independent non-negative sources. No information is given on the operational changes and the induced faulty - thus the separation is unsupervised.

The results in Figure 5 are impressive: Source 1 model the wear due to increased friction between piston and liner. It suddenly appears just after the oil was shut down, increases throughout the experiment until the lube oil system is restored. The remaining sources model the load changes, with only slight problems of separating the 50% and 75% loads fully. It is fair to conclude that the MFICA resulted in a highly informative clustering of the observed signals, directly in line how we group the observations, and thus an example of the

powerful cognitive properties of the independent components as reported in [4].

Also the hidden signals shown in Figure 5 are remarkable. The first signal clearly picks up the more or less constant noise from the increased friction; it is lower in the beginning and in the end possibly due to the fact that the cylinder sucked up oil from the outer cylinders from the bottom tub. The signal also contains the quite severe component that is generated when the piston passes the scavenge air holes in the downstroke. The remaining components model the changes in the normal condition signals as a function of the load, e.g., the movement of the peaks in the injection period right after TDC. The hidden signals shown in Figure 3 were obtained from normal and faulty examples at 25% load. When comparing those to the ones obtained with the additional examples from 50% and 75% load, the MFICA algorithm was able to provide a much better estimate of the signal component modeling the increased friction between piston and liner. With the multiple loads the independence of the increased friction signal and the normal engine events become more apparent for the algorithm.

For comparison the source estimates using PCA are IMICA are shown as an reference. As Figure 7 and Figure 8 clearly the methods capture the changes, i.e., the sources change when the condition changes. However, the result is not comparable to the cognitive grouping provided by the MFICA, we would also expect that the hidden signals obtained with those two methods contain parts from all conditions, e.g., not like the hidden signals in Figure 6.

## 5 Conclusion

This paper provides new insight on the use of independent components analysis for condition monitoring. It has been a goal throughout the whole AEWATT



project to find a signal component that picks up the increased friction between piston and liner regardless of the operational condition. The accurate grouping of the examples obtained without telling the algorithm what to look for was remarkable and fully aligned with the experimental setup. We believe that this provides new and promising opportunities in field of condition monitoring.

## Acknowledgements

The European Commission is acknowledged for the funding of this work through the 5th framework AEWATT project. Experimental data was generously provided by project partners at MAN B&W. The engine sketch in Figure 1 is due to Ryan Douglas, Heriot-Watt University. The presentation and discussion on cognitive components in our local journal club provided inspiration for this work.

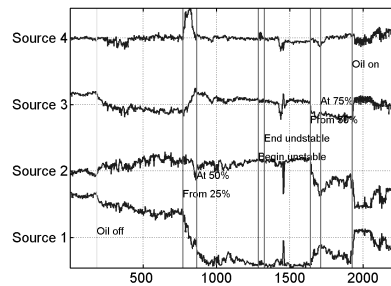


Figure 7: Source estimates using principal components analysis of the whole data set.

## References

- [1] A. Bell and T. Sejnowski. An information-maximisation approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159,

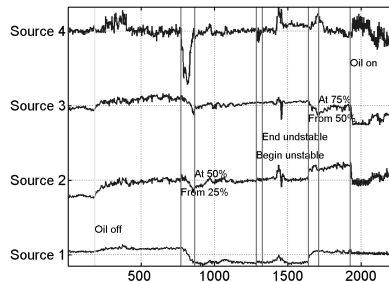


Figure 8: Source estimates using Information maximization ICA on the whole data set.

1995.

- [2] L. Feng, L. K. Hansen, and J. Larsen. On low level cognitive components of speech. In Honkela et al., editor, *AKKR'05 International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning*, Helsinki, Finland, jun 2005. Pattern Recognition Society of Finland.
- [3] Torben Fog. Scuffing Experiment, Experiment Log. 2000.
- [4] L. K. Hansen, P. Ahrendt, and J. Larsen. Towards cognitive component analysis. In Finnish Cognitive Linguistics Society Pattern Recognition Society of Finland, Finnish Artificial Intelligence Society, editor, *AKRR'05 -International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning*. Pattern Recognition Society of Finland, Finnish Artificial Intelligence Society, Finnish Cognitive Linguistics Society, jun 2005.
- [5] P. A. Højen-Sørensen, O. Winther, and L. K. Hansen. Mean field approaches to independent component analysis. *Neural Computation*, 14:889–918, 2002.
- [6] T. Kolenda, L. K. Hansen, and S. Sigurdsson. Independent components in text. In *Advances in Independent Component Analysis*, pages 229–250. Springer-Verlag, 2000.

- [7] T. Kolenda, S. Sigurdsson, O. Winther, L. K. Hansen, and J. Larsen. DTU:Toolbox. Internet, 2002. <http://isp.imm.dtu.dk/toolbox/>.
- [8] L. Molgedey and H.G. Schuster. Separation of a mixture of independent signals using time delayed correlations. *Phys. Rev. Lett.*, 72(23):3634–3637, 1994.
- [9] K. Petersen, L. K. Hansen, T. Kolenda, and E. Rostrup. On the independent components of functional neuroimages. In *Third International Conference on Independent Component Analysis and Blind Source Separation*, pages 615–620, 2000.
- [10] N. H. Pontoppidan, J. Larsen, and T. Fog. Independent component analysis for detection of condition changes in large diesels. In Om P. Shrivastav, Bassim Al-Najjar, and Raj B.K.N. Rao, editors, *COMADEM 2003*. COMADEM International, 2003.
- [11] Niels Henrik Pontoppidan and Ryan Douglas. Event alignment, warping between running speeds. In Raj B.K.N. Rao, Barry E. Jones, and Roger I. Grosvenor, editors, *COMADEM 2004*, pages 621–628, Birmingham, UK, aug 2004. COMADEM International.



## APPENDIX B

# Condition monitoring with Mean field independent components analysis

---

N. H. Pontoppidan, S. Sigurdsson, and J. Larsen. Condition monitoring with mean field independent components analysis. *Mechanical Systems and Signal Processing*, 19(6):1337–1347, nov 2005b. URL <http://dx.doi.org/10.1016/j.ymsp.2005.07.005>. Special Issue: Blind Source Separation



# Condition Monitoring with Mean Field Independent Components Analysis

Niels Henrik Pontoppidan, Sigurdur Sigurdsson and Jan Larsen

*Informatics and Mathematical Modelling - IMM  
Technical University of Denmark  
Richard Petersens Plads - Building 321  
DK-2800 Kongens Lyngby - Denmark  
[www.imm.dtu.dk](http://www.imm.dtu.dk)*

---

## Abstract

We discuss condition monitoring based on mean field independent components analysis of acoustic emission energy signals. Within this framework it is possible to formulate a generative model that explains the sources, their mixing and also the noise statistics of the observed signals. By using a novelty approach we may detect unseen faulty signals as indeed faulty with high precision, even though the model learns only from normal signals. This is done by evaluating the likelihood that the model generated the signals and adapting a simple threshold for decision. Acoustic emission energy signals from a large diesel engine is used to demonstrate this approach. The results show that mean field independent components analysis gives a better detection of fault compared to principal components analysis, while at the same time selecting a more compact model.

*Key words:* Mean Field Independent Components Analysis, Condition Monitoring, Unsupervised learning

---

## 1 Introduction

In this paper we apply *mean field independent component analysis* (MFICA) to condition monitoring of a large two-stroke diesel engine. The setup is as follows: from a collection of examples gathered under normal running conditions we learn the underlying sources of the signal and how they are mixed

---

*Email addresses:* [nhp@imm.dtu.dk](mailto:nhp@imm.dtu.dk) (Niels Henrik Pontoppidan),  
[siggi@imm.dtu.dk](mailto:siggi@imm.dtu.dk) (Sigurdur Sigurdsson), [jl@imm.dtu.dk](mailto:jl@imm.dtu.dk) (Jan Larsen).

together. Given a new collection of observed signals we test if the obtained model explains the data equally well as it did with the known normal ones; if not the condition is faulty.

The interesting result is that condition monitoring can be achieved with a single feature by a simple threshold comparison, and further where model and threshold is derived from normal condition data only. Still we obtain a system capable of detecting several types of faults, however the different faults are not directly identified by the system. This setup is highly relevant as we are facing a problem where specific modeling of specific faults is not economically feasible.

We will introduce MFICA and apply it to a collection of labeled normal and faulty examples and show how the two classes separate by looking at the underlying hidden signals and the independent components. We compare this to a setup where only normal examples are available for training. This comparison show that we can separate the two classes with a model build from normal examples only, while obtaining good detection of faulty examples. Whith this in mind we outline our unsupervised method and its results.

### *1.1 Data setup*

We are working with real Acoustic Emission energy (AEE) signals acquired on the liner on the two-stroke test bed engine at MAN B&W Diesel in Copenhagen. The AEE signals are sampled in the crank angular domain and partitioned such that a single example represents the AEE during a single engine cycle. The engine cycle for a two-stroke engine is one revolution of the camshaft, so each sample corresponds to the AEE at a certain angular position of the camshaft and piston. Figure 1 show the engine with piston and camshaft (left), a AEE “radar-plot” where the AEE amplitude is shown as distance from the center (middle), and finally the AEE signal in the linear crank angle domain (right). Each observation consist of 2048 AEE samples in the crank angle domain from  $-180^\circ$  to  $180^\circ$ , i.e. one revolution of the camshaft. The peaks in the AEE signals are the results of engine related events, e.g., the peak around  $0^\circ$  is the combustion and fuel-injection operation.

In the crank angular domain all observations signals have same length regardless of running speed[1]. However, many engine designs, including the MAN B&W Diesel test bed engine in Copenhagen, optimize performance by moving the angular position of certain events as a function of load and speed settings. For instance by advancing the fuel injection in order to inject more fuel. In this paper we will focus on a single running condition, and refer to our ongoing research on event alignment[2] for handling the non-stationary case under



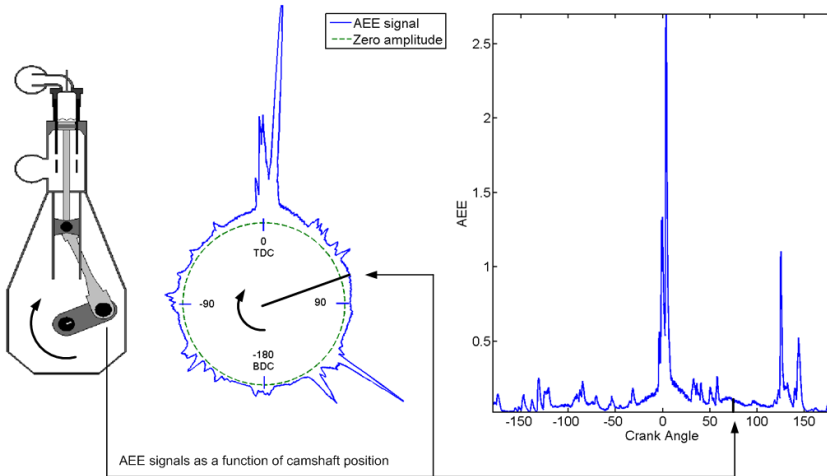


Fig. 1. Crank Angle domain sampling of acoustic emission energy signals. The two arrows point at the data points as a position of the camshaft wrt.  $0^\circ$  aka. Top Dead Center (TDC) where the combustion takes place.

multiple running conditions.

We denote an observation of the AEE signal in an engine cycle by  $\mathbf{x}$  being a  $d \times 1$  vector of non-negative elements. From a set of from  $N$  such vector cycles we build the training matrix  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$  as seen in Figure 2 to the left. In the blind source separation (BSS) framework we assume that the training matrix is generated by a linear mixing of  $K$  underlying non-negative AEE signals plus white Gaussian noise given by

$$\mathbf{X} = \mathbf{A}\mathbf{S} + \mathbf{\Gamma}, \quad (1)$$

where  $\mathbf{A}$  is a  $d \times K$  mixing matrix,  $\mathbf{S}$  is  $K \times N$  source matrix and  $\mathbf{\Gamma}$  is  $d \times N$  noise matrix.

As Figure 2 show, the columns of  $\mathbf{A}$  contain the underlying hidden AEE signals. A column in  $\mathbf{S}$  contain the gain factors for each of the hidden AEE signals in the mixing matrix that are used to generate an observation, e.g.  $\mathbf{x}_{1:d,1} = \mathbf{A}_{1:d,1} \cdot \mathbf{S}_{1,1} + \mathbf{A}_{1:d,2} \cdot \mathbf{S}_{2,1} + \mathbf{\Gamma}_{1:d,1}$ .

## 2 Mean field independent components analysis

In a condition monitoring framework using the MFICA, the columns of the mixing matrix  $\mathbf{A}$  may be interpreted as underlying AEE signals, generated by specific events. For instance, these sources could be the results of specific

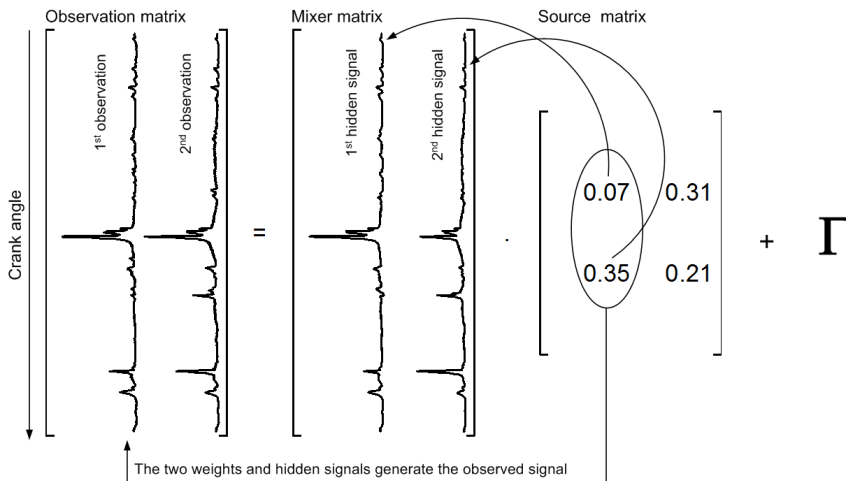


Fig. 2.  $\mathbf{X} = \mathbf{A}\mathbf{S} + \mathbf{\Gamma}$ , Matrix setup for blind source separation with two observations and components.  $\mathbf{X}$ ,  $\mathbf{A}$ ,  $\mathbf{S}$  and  $\mathbf{\Gamma}$  are observation, mixing, source and noise matrix respectively. Each observation signal  $\mathbf{x}$  is generated by mixing the columns in the mixing matrix  $\mathbf{A}$  weighted by the corresponding set of gain factors (as a column) in the source matrix  $\mathbf{S}$  by the corresponding column in the Gaussian noise matrix  $\mathbf{\Gamma}$ .

engine impacts and scratching. As the AEE signals are inherently nonnegative, it is appropriate to assume that the various energy sources are additive and no source is able to extracting energy from the system. This implies that both the elements of the mixing matrix and source matrix are nonnegative.

## 2.1 Training

Recently, the ICA was extended with a Bayesian framework using an advanced mean field training [3], making it possible to incorporate constraints on the source and mixing matrix. The MFICA accomplishes this by defining an appropriate prior distribution over the sources. Given the noise model in Equation 1, the likelihood for the parameters and sources of the MFICA may be written as

$$p(\mathbf{X}|\mathbf{A}, \mathbf{\Sigma}, \mathbf{S}) = (\det(2\pi\mathbf{\Sigma}))^{-N/2} \exp\left(-\frac{1}{2}\text{Tr}\{(\mathbf{X} - \mathbf{A}\mathbf{S})^\top \mathbf{\Sigma}^{-1}(\mathbf{X} - \mathbf{A}\mathbf{S})\}\right) \quad (2)$$

where noise has zero mean and  $\mathbf{\Sigma}$  is the noise covariance matrix. The aim of MFICA is to estimate the unknown quantities, the sources  $\mathbf{S}$ , the mixing matrix  $\mathbf{A}$  and the noise covariance  $\mathbf{\Sigma}$  from the observed data. For the condition monitoring problem we need to characterize the unknown parameters, the noise is simplified to an isotropic Gaussian distribution where  $\mathbf{\Sigma} = \sigma^2\mathbf{I}$ , the

sources are assumed exponential distributed, given by the prior distribution  $p(\mathbf{S}) = \eta \exp(-\eta\mathbf{S})$  where  $\eta > 0$ , and the mixing matrix is assured nonnegative elements by combining Lagrange multipliers to the mean field training. The parameter estimation is done in a Bayesian manner, by integrating out the hidden variable  $\mathbf{S}$ , i.e.,

$$p(\mathbf{X}|\mathbf{A}, \mathbf{\Sigma}) = \int p(\mathbf{X}|\mathbf{A}, \mathbf{\Sigma}, \mathbf{S})p(\mathbf{S}) d\mathbf{S} \quad (3)$$

and using this new likelihood to optimize the parameters. Unfortunately, this integral is intractable to solve analytically. Instead, equation (3) is approximated using the so-called adaptive Thouless-Anderson-Palmer mean field approach [4]. For details on the MFICA we refer to [3] and also to the freely available Matlab toolbox [5].

While the parameters may be estimated with MFICA, we still need to determine the number of components  $K$ . This corresponds to a model selection problem where we are interested in finding a model that fits the data well in the face of limited data, i.e., has good generalization capabilities on unseen data. If the  $K$  is selected too small compared to the optimal  $K$ , we get a too simple model that does not capture the underlying function generating data, i.e., the sources and mixing matrix. On the other hand, selecting a  $K$  that is too large gives a too complex model that fits to the additive noise. Various methods for model selection have been proposed, e.g., empirically with cross-validation resampling schemes [6] that we use here and algebraically, e.g. using Bayesian information criterion [7].

## 2.2 Using the trained model on a new example

Given a new example  $\mathbf{x}$  and a model defined by  $\mathbf{A}$  and  $\mathbf{\Sigma}$  we redo a partial BSS problem, i.e., solve for  $\mathbf{s}$  in  $\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{\Sigma}$  whilst keeping  $\mathbf{A}$  and  $\mathbf{\Sigma}$  fixed. Due the constraints on  $\mathbf{s}$  the solution is obtained using the same mean field optimizer that was used for the training. The output of this optimization is the components  $\mathbf{s}$  and the corresponding negative log likelihood (NLL)  $-\log p(\mathbf{x}|\mathbf{A}, \mathbf{s}, \mathbf{\Sigma})$ . We have previously showed[8] how the log likelihood (NB not negative) dropped significantly just after a condition change, and further how it regained its usual level after a temporary fault in the water brake disappeared. Effectively for classification purpose we can reduce the dimensionality from 2048 (original data dimension) to NLL with MFICA.

### 2.3 Classification with the negative log likelihood

The MFICA does a good job on separating the normal and faulty examples in the NLL-domain. A simple but effective classification method is to set an NLL threshold to separate the faulty examples from the normal. This is related to novelty detection [9]. The threshold is obtained from an empirical cumulated density function build from another set of known normal examples. From the the cumulated density function we can derive a threshold with an inherent rejection rate of say 5%. Even with an inherent rejection rate of 0% we could still face false alarms, however both settings can be taken care of with binomial hypothesis testing against the inherent rejection rate (see further [10]).

## 3 Comparing two and one class training

In this section we will compare the results obtained when solving two different BSS problems. We take 140 known normal and 140 known faulty examples and split them into two sets of 40 (for training) and 100 (for testing) examples. We will call the first problem *two class* as the observation matrix is build from the 80 labeled training examples. In the second problem, the *one class*, we will only use the 40 normal examples. We will solve the two class problem assuming two independent components and compare this result with the result that we obtain when solving the one class problem assuming only a single component. It is expected that one of the columns in the two class mixing matrix should resemble the column in the one class mixing matrix. Further the the classes should separate in component domain. Figure 3 show the two columns of the two class mixing matrix to the right and further the panel to the left show that 200 labeled test points separate in component domain. The two columns of the two class mixing matrix are repeated in Figure 4 to the right, where we see that the second column resembles the column of the one class mixing matrix shown above the two. Furthermore the left panels of Figure 4 show that the 200 labeled test points also separate in the single component domain – although not as good as in the two class case.

## 4 Unsupervised condition monitoring

Due to economic figures the supervised setup is not an option for our application. It is simply to expensive to conduct the experiments for a wide range of faults and therefore we aim for a unsupervised setup where the system consist of a model of the normal condition so that derivations from the normal condition can be detected. The setup can later be turned into a semi supervised

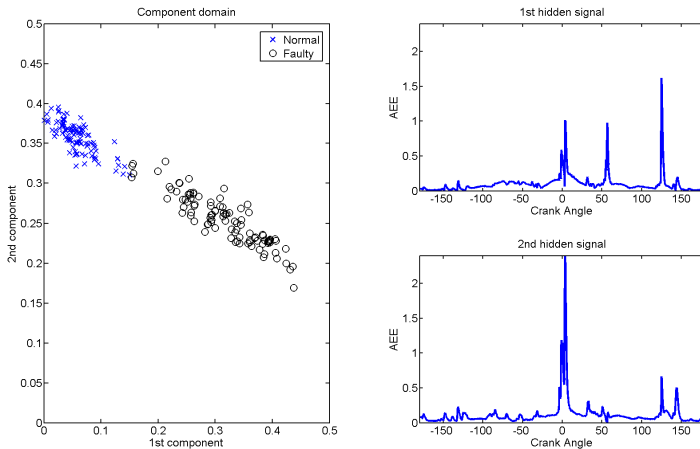


Fig. 3. The normal and faulty examples separate in source domain, the faulty examples tend to contain more of the 1st hidden signal. Further we add that the 2nd hidden signal looks very normal when we compare to the single component obtained from normal examples only in Figure 4.

setup by having additional models for specific faults in parallel to the normal model.

In the one class example we assumed one independent component, but as the following results show, more than one component can give a better classification performance. This could very well be controlled by the variation and number of independent modes in the normal condition. We have recently followed that idea, and trained a unsupervised model on a collection of examples acquired under multiple normal load settings. However it turned out that the performance was inferior to our event alignment method[10].

#### 4.1 Experimental setup

The experiment was conducted by acquiring the AEE signals before and after a fault condition was induced by closing lubricating oil system attached to the monitored cylinder. This resulted in increased friction and wear that possibly could lead to a severe fault called scuffing. In this paper we only used the first two hours of data, so what we detect is an early warning. Visual inspection after 6 hours of running without lubricating oil revealed contact marks on the upper rings inside the cylinder.

- 70 repetitions with resampling of both training and evaluation examples
- 20 examples in training matrix

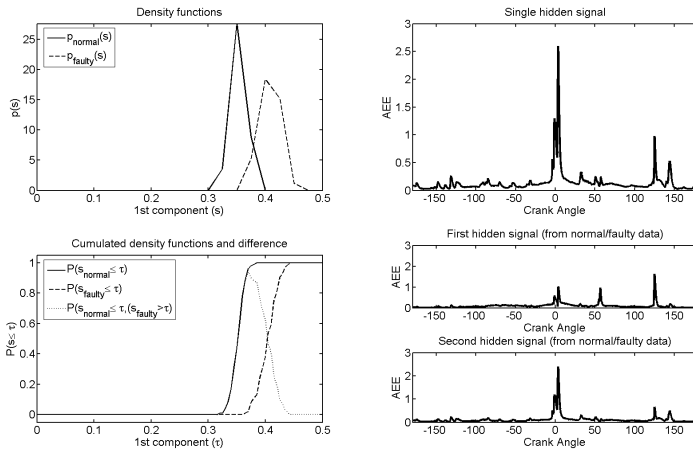


Fig. 4. The observation matrix consist of normal examples only, this is one class, single component MFICA. The two panels to the right show the empirical densities and cumulated densities for the sources of normal and faulty data. The lower panel show that  $s \leq 0.36$  for 95% of the normal examples whilst  $s > 0.36$  for virtually all faulty examples. To three panels to the right show the single hidden signal from the “unsupervised” mixing matrix, that can be compared to the two hidden signals obtained from the two class setup in Figure 3. Clearly the second hidden signal resembles the single hidden signal better than the first thereby being the “normal” hidden signal.

- Testing for 1-12 components
- Set rejection rate 5%
- NLL threshold learning set with 70 normal examples
- Evaluation set containing 70 examples with known labels, whereas 30-40 are normal

## 4.2 Results

We will compare the performance of MFICA to a very similar Principal Components Analysis method, described in [11], that does not obey the non-negativity constraints on  $\mathbf{A}$  and  $\mathbf{S}$ . For each of the 67 experiments we compare the best MFICA and PCA model. The model with lowest false alarm rate and highest detection rate (as a squared distance from the optimal Receiver-Operator-Characteristics point) is the best model. Of the 70 experiments MFICA is better than PCA in 47 ( $\sim 67\%$ ), in 18 experiments ( $\sim 26\%$ ) the two methods have equal performance. PCA is only better than MFICA in 5 of the experiments ( $\sim 7\%$ ). If we compare the number of components in the “best” models (for MFICA the 65 experiments and PCA the 23 experiments), we see

MFICA performs better	Equal performance	PCA performs better
47 times	18 times	5 times

Table 1

Performance statistics on the 70 runs

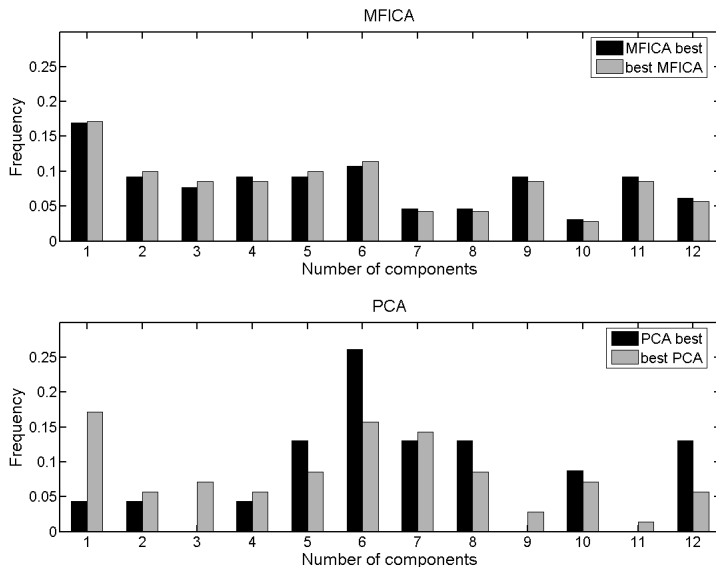


Fig. 5. Normalized histograms for best number of components. The histograms called *best MFICA/PCA* are from all experiments, whereas the *MFICA/PCA best* are build from the experiments where MFICA or PCA is at least as good as the other model, i.e. *MFICA best* is build from 65 examples and *PCA best* from 23.

in Figure 5 that the *PCA best* histogram is peaked around 6 components. The *MFICA best* histogram is flatter and with a peak at only one component.

In ?? we show the improvement in the ROC domain for the 47 experiments where MFICA is better than PCA. In 38 of those the improved false alarm rate is achieved without decreasing the detection rate. The mean improvement is  $-0.045$  (from 0.0752 to 0.03) on the false alarms and  $-0.0057$  (from 1 to 0.9943) on the detection rate. So although PCA already gives good classification performance, MFICA is capable of improving on that.

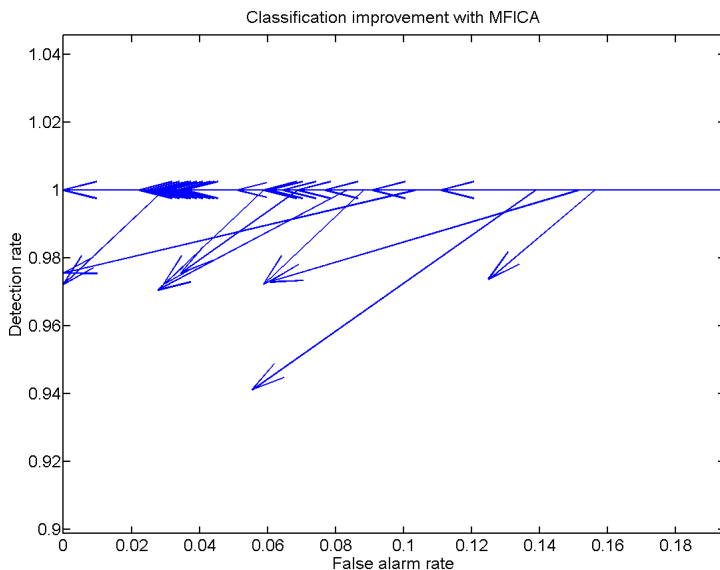


Fig. 6. Improvement with MFICA. The arrows point from the PCA to the MFICA ROC point for the experiments where MFICA is better. Only 9 of the 47 arrows point downwards, the rest show reduced false alarm rates without decreased detection rate

## 5 Conclusion

We have described how the advanced blind source separation technique, Mean field independent components analysis, can be applied to a realistic condition monitoring problem. The experiments show how this method performs better than a similar method based using Principal Components Analysis. We have planned improved experiments where the lubricating oil level is reduced over time, to induce even more subtle faults.

### *Acknowledgements*

The authors are funded by the European Commission through the AE-WATT project, grant GRD2-2001-50014. Experimental data was kindly provided by project partner MAN B&W Diesel A/S. The authors recognize the fruitful collaboration with the Department of Mechanical and Chemical Engineering at Heriot-Watt University. The engine sketch to the left in Figure 1 is due to Ryan Douglas, Heriot-Watt University.



## References

- [1] G. Chandroth, A. Sharkey, Utilising the rotational motion of machinery in a high resolution data acquisition system, in: Proc of Computers and ships- from ship design and build, through automation and management and on to ship support, 1999.
- [2] N. H. Pontoppidan, J. Larsen, Non-stationary condition monitoring through event alignment, in: IEEE Workshop on Machine Learning for Signal Processing, IEEE Press, Piscataway, New Jersey, 2004, pp. 499–508.  
URL <http://www.imm.dtu.dk/pubdb/p.php?3131>
- [3] P. Højen-Sørensen, O. Winther, L. Hansen, Mean field approaches to independent component analysis, *Neural Computation* 14 (2002) 889–918.  
URL <http://www.imm.dtu.dk/pubdb/p.php?611>
- [4] M. Opper, O. Winther, Tractable approximations for probabilistic models: The adaptive thoulless-anderson-palmer mean field approach, *Phys. Rev. Lett.* 86 (2001) 3695–3699.  
URL <http://www.imm.dtu.dk/pubdb/p.php?614>
- [5] T. Kolenda, S. Sigurdsson, O. Winther, L. Hansen, J. Larsen, Dtu:toolbox, Internet (2002).  
URL <http://mole.imm.dtu.dk/toolbox/>
- [6] B. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge, 1996.
- [7] G. Schwarz, Estimating the Dimension of a Model, *The Annals of Statistics* 6 (1978) 461–464.
- [8] N. H. Pontoppidan, J. Larsen, T. Fog, Independent component analysis for detection of condition changes in large diesels, in: O. P. Shrivastav, B. Al-Najjar, R. B. Rao (Eds.), *COMADEM 2003*, COMADEM International, 2003.  
URL <http://www.imm.dtu.dk/pubdb/p.php?2400>
- [9] C. Bishop, Novelty detection and neural network validation, in: *IEE Proceedings - Vision Image and Signal Processing*, Vol. 141, 1994, pp. 217–222.
- [10] N. H. Pontoppidan, J. Larsen, S. Sigurdsson, Non-stationary condition monitoring of large diesel engines with the AEWATT toolbox, in: *Essential Technologies for succesful prognostics*, Society for Machinery Failure Prevention Technology, 2005.  
URL <http://www.imm.dtu.dk/pubdb/p.php?3351>
- [11] N. H. Pontoppidan, J. Larsen, Unsupervised condition change detection in large diesel engines, in: C. Molina, T. Adali, J. Larsen, M. Van Hulle, S. Douglas, J. Rouat (Eds.), *2003 IEEE Workshop on Neural Networks for Signal Processing*, IEEE Press, Piscataway, New Jersey, 2003, pp. 565–574.  
URL <http://www.imm.dtu.dk/pubdb/p.php?2465>



## APPENDIX C

# Non-stationary condition monitoring of large diesel engines with the AEWATT toolbox

---

N. H. Pontoppidan, J. Larsen, and S. Sigurdsson. Non-stationary condition monitoring of large diesel engines with the AEWATT toolbox. In [Pusey et al. \[2005\]](#). URL <http://www.imm.dtu.dk/pubdb/p.php?3351>



## 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 recalculate 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 ( $\mathbf{x}$ :  $D \times 1$ ,  $\mathbf{X}$ :  $D \times N$ ).

$$\mathbf{X} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \dots \quad \mathbf{x}_N] \quad (1)$$

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.

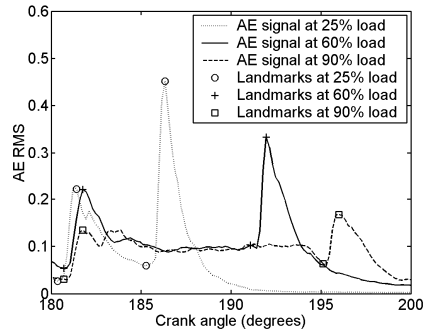


Figure 1: The AE RMS signals during the injection period with different load setting. The markers are the landmarks, indicating the time positions that should be aligned.

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.

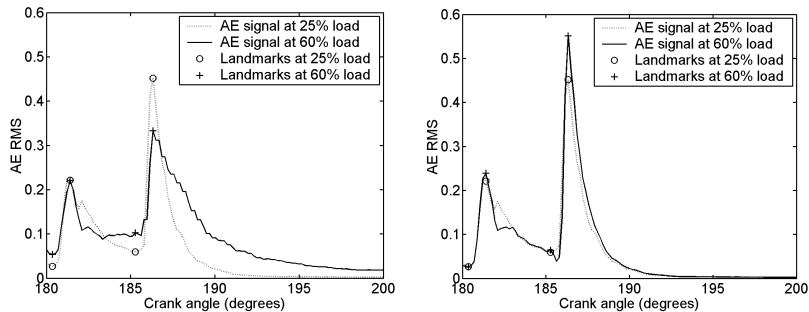


Figure 2: The left panel shows the AE signals at 25% and 60% load from Figure 1 after event aligning the signal at 60% load with the signal at 25% load, using a piece-wise linear spline-interpolation. The right panel shows the results after event and amplitude alignment.

**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  $\varepsilon$  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.  $K \ll D$ . A common way of extracting features is to estimate the source signals, 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  $\varepsilon$  as isotropic Gaussian noise. The left panel of Figure 3 illustrates the use of PCA for feature extraction with two dimensional toy patterns.

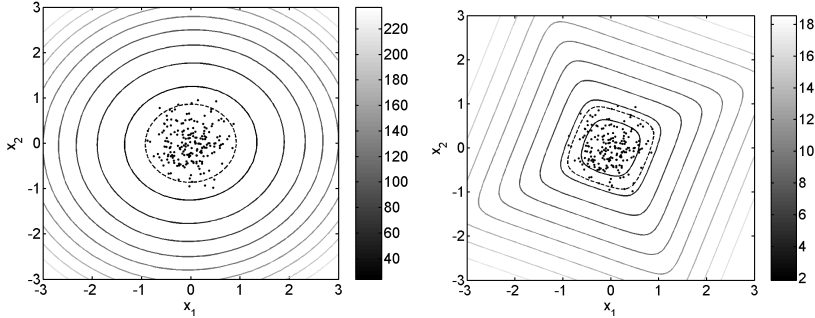


Figure 3: The left panel illustrates the negative log-likelihood results in 2-D input space for the PCA and the right panel for the ICA. The dots are the measured normal patterns, the solid lines are contours of the negative log-likelihood surface and the dashed line is a threshold found by computing 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.

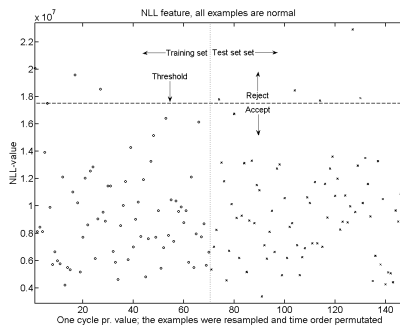


Figure 4: Negative Log-likelihood for training and test set of known normal examples (2 components MFICA)

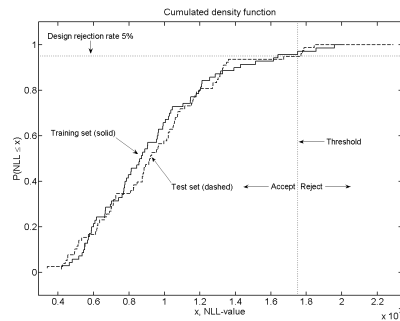


Figure 5: Empirical cumulated density function of NLL values in Figure 4. The two CDF's are very similar.

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) were 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.

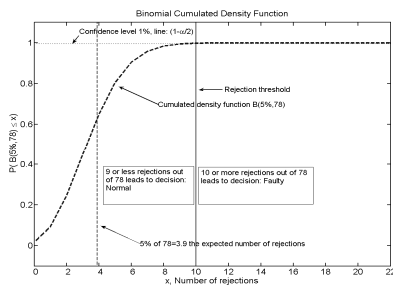


Figure 6: Binomial hypothesis testing with Kos engine data. Target rejection rate 5%, confidence level 1% on 78 examples give new threshold 10 (10/78=13%)

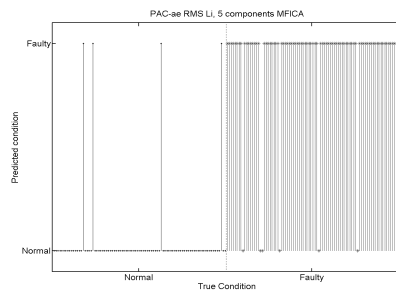


Figure 7: Binomial sequence using MFICA with 5 components on data from Copenhagen test bed. The number of rejected examples rises at the condition change (NB examples not in time order).

**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
<b>MEAN</b>	0.977	0.709	0.714	0.980
<b>PCA</b>	0.966	0.894	0.946	0.957
<b>MFICA</b>	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/awatt>). 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 ICASSP2001, 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



## APPENDIX D

# Extracting information from conventional AE features for fatigue onset damage detection in carbon fiber Composites

---

Runar Unnthorsson, Niels Henrik Pontoppidan, and Magnus Thor Jonsson. Extracting information from conventional AE features for fatigue onset damage detection in carbon fiber composites. In Pusey et al. [2005]. URL <http://www.imm.dtu.dk/pubdb/p.php?3289>





# EXTRACTING INFORMATION FROM CONVENTIONAL AE FEATURES FOR ONSET DAMAGE DETECTION IN CARBON FIBER COMPOSITES

Runar Unnthorsson<sup>a</sup> and Niels Henrik Pontoppidan<sup>b</sup> and Magnus Thor Jonsson<sup>a</sup>

<sup>a</sup> Department of Mechanical Engineering, University of Iceland  
Hjarðarhaga 2-6, IS-107 Reykjavik, Iceland

<sup>b</sup> Informatics and Mathematical Modeling, Technical University of Denmark  
Richard Petersens Plads 321, 2800 Lyngby, Denmark  
Corresponding author Runar Unnthorsson, runson@hi.is

**Abstract:** We have analyzed simple data fusion and preprocessing methods on Acoustic Emission measurements of prosthetic feet made of carbon fiber reinforced composites. This paper presents the initial research steps; aiming at reducing the time spent on the fatigue test. With a simple single feature probabilistic scheme we have showed that these methods can lead to increased classification performance. We conclude that: the derived features of the TTL count leads to increased classification under supervised conditions. The probabilistic classification scheme was founded on the histogram, however different approaches can readily be investigated using the improved features, possibly improving the performance using multiple feature classifiers, e.g., Voting systems; Support Vector Machines and Gaussian Mixtures.

**Key Words:** Acoustic Emission; Carbon fibres; Data fusion; Fatigue testing; Probabilistic classification; Supervised learning

**Introduction:** During the design phase of a prosthetic carbon fiber foot at Össur hf., prototype models are built, evaluated, and improved to meet the design criteria. For evaluation several testing methods are used; visual tests, stiffness measurements and fatigue testing. Feet are mainly subjected to dynamic loading, which sometimes requires low loads to initiate and propagate faults [16]. Because of this fatigue testing is extremely important part of the design process. Fatigue tests of composites can take very long time, up to several weeks. It is therefore valuable for an engineer, that wants to test a prototype, to be able to see if it will fail early during the test. Considerable time can be saved by this. By shortening the test time, feet can be; designed and developed in less time and at a lower cost than previously possible. During the fatigue testing of an prosthetic feet, the stiffness is monitored and visual tests are also performed regularly. However, visual tests can only detect faults that extend to the surface of a foot or affect it in some way. In order to be able to predict the fatigue strength of a foot undergoing fatigue testing, other NDT methods can be used for obtaining data.

When microstructural changes occur in composites, energy is released and transient stress waves are generated. These stress waves are called called Acoustic Emissions (AE) [16]. The stress waves travel through the composite and when they reach the surface it will vibrate. The small surface displacements generated by the vibration of the composite can be detected by using appropriate sensors. According to Duesling [3] piezoelectric sensors are most popular. AE can be generated by several types of damage that occurs in the material, e.g. fibre breakage, matrix cracking, delamination etc. Because microstructural damage generates AE the method has the potential of detecting damage in its early stages, much earlier than possible by monitoring deflection or performing visual inspection. Therefore a microstructural damage that starts to formate in the last cycles of a fatigue test can be detected, a damage that would pass undetected by the other two previously mentioned methods.

AE signals are not only generated by damage, they can also come from other sources, such as; the testing machine [16], electrical disturbances [16, 5], friction due to rubbing of parts [12] and the friction generated by opening and closing of matrix cracks [13]. According to Hamstad [7] AE is also generated under loading because of the different material properties of the fibers and the matrix.

Friction generated AE can provide important information about the condition of an composite [1]. Some researchers have attempted to filter the AE signal generated by friction from the total signal in order to better detect AE from damage, but this can be difficult [15]. The fact that a damage only emits AE once but friction many times suggests that approaches based on friction will be more robust.

An AE parameter analysis is the conventional way of performing an AE monitoring. The results from using standard AE parameters, especially amplitude, are contradicting in the literature [5, 2, 8, 3]. Amplitude suffers from attenuation. The main reasons for the attenuation are; geometric spreading, dispersion, internal friction and scattering [11]. Also, Prosser et al. [10] reported that the same type of damage doesn't always produce AE with the same amplitude. This indicates that amplitude is not a very good feature. Godin et al. [6] claimed that conventional AE analysis cannot distinguish between different AE sources and suggested that more advanced methods, such as multivariate analysis and classifiers, should be used. Similar comments were made by Tsamtsakis et al. [15]. They suggested that new parameters, like force or displacement, should be added.

This paper reports the results of using several methods to extract information from conventional AE features. The features were obtained using a commercial AE acquisition system (AESmart 2000 from DECI Inc) and consist of a fixed set of standard AE features. The emphasis was put on generating features which could be used for fast and reliable detection of damage onset. It was also considered important for the detection method to be robust. Robustness was believed to be obtained by basing the features on friction based AE.

The outline of the remaining of the paper is as follows: First the setup of the acquisition system is explained then the methodology is explained. It starts with explaining how new features are derived from the available feature set by both applying sensor fusion and using moving window second order moment. The section ends with a description a simple classification system based on the assumption that faulty specimens have inhibit greater AE

activity than normal ones, thus threshold on the features can be used for classification. We compare classification properties of the different features by using receiver-operator characteristics (ROC) curves. The features are compared and discussed – identifying the useful features. Finally the concluding remarks are made and future work is outlined.

**System Setup:** The fatigue test specimens were two types of prosthetic feet differing slightly in stiffness and size. Foot no. 1 is stiffer and smaller than foot no 2. The construction of both feet is the same, they are made from unidirectional carbon fiber reinforced epoxy and woven mats are used for the top and bottom layers for nicer look. The test was performed according to ISO 10328 specifications, with a “Foot/Limb” test system at Össur’s testing facilities. During the fatigue test, two actuators were used to flex the foot for  $2 \times 10^6$  cycles at 1.5 Hz. The maximum load for each actuator was kept constant and the deflection was monitored. No change in maximum deflection was measured, which means that no stiffness reduction was observed during the tests. For data collection SE9125-MI data transducers and the AESmart 2000 AE acquisition system, from DECI Inc., was used.

According to Dunegan [4] the system splits the AE signal into low frequency (LF) and high frequency (HF) signals. The LF signal contains frequencies between 20 to 60 kHz and the HF signal contains frequencies between 100 to 500 kHz. Data is only collected from one sensor at a time, each time the system switches between the sensors it stores the data in an Excel file. The data that is stored is not the actual AE signal but instead the following features that the system extracts from this signal; time of the recorded event, HF counts above threshold (TTL counts), HF peak amplitude, LF peak amplitude, ratio of the HF/LF peak amplitudes, event count, time difference between HF and LF signals. Other features are also stored by the system but, they are irrelevant to the test.

The system was set up to switch between the two sensors, i.e. one on each foot, using 6 second dwell time. The monitoring was performed twice a day for 4 to 6 hours each time. The gain for both the LF and HF gain was set to 60 dB. The threshold setting for the LF and HF was set to 200 mV. The time interval for counting the TTL counts was set to 1 ms.

**Data preprocessing:** During data acquisition each foot was monitored for periods of six seconds, before the system switched started monitoring the other foot. Each entry in the data file is the result of the following events

1. AE signal crosses the trigger threshold
2. Acquisition board computes the set of features from a 1 ms analysis window
3. Acquisition board goes back to detecting threshold crossing in AE signal. Further the system might also change foot if the 6 s period has ended

Moving window second order moment: We compute the second order moment over a fixed number of consecutive data entries, move the window and repeat. This gives the moving window second order moment that peaks when the local mean value change, and also when

the local variance increase. So this is just a simple “amplification” of changes in a time series. Besides from giving a different measure (variance instead of values) this method also performs filtering since a number of consecutive samples are used for each value.

$$\tilde{\mu}(n) = \frac{1}{N} \sum_{n'=0}^{N-1} x(n - n') \quad (1)$$

$$\tilde{\sigma}^2(n) = \frac{1}{N-1} \sum_{n'=0}^{N-1} (x(n - n') - \hat{\mu}(n))^2 \quad (2)$$

Time normalization: Each data entry comes from a 1 ms analysis window triggered by a threshold crossing. So we can imagine several regimes (explaining the data entries in the 6 s periods where the system monitors each foot).

1. few data entries with a few high readings – occasional low activity
2. many data entries with a few high readings – steady low activity
3. few data entries with many high readings – occasional high activity
4. many data entries with many high readings – steady high activity

This means that if we omit the time information and look at the feature values, we cannot differ between the occasional and steady activity. Instead we fuse the features in each 6 s period. We deliberately use the term fuse, even though most features are just summed up during the period. But for the time entry we take the earliest value, for the event count we take the maximum, and for the ratio we recalculate; since the original ratio computation was constrained. Another benefit of time normalization is reduction of data size as several events are combined into one.

We can also apply the moving window second order moment to the time normalized data in order to further enhance changes.

Energy normalization: In the LF/HF peak amplitude we have the peak amplitudes, we already have the number of TTL counts in a 6 s period. Multiplying those two quantities and sum over the events in the period gives an upper bound estimate of the “energy”.

$$E = \sum_i TTL_i P_i \quad (3)$$

Filtering: One way of decreasing feature variance is low-pass filtering, essentially this forces the classification system to evaluate the feature value for consecutive examples. This smears out occasional high values, but also the sudden steps in the feature, thus this delays the significant changes. Further notice that this is related to the post classification processing described last in the following section.

**Feature processing**: We adopt a very simple setup for supervised single feature classification of the specimen condition. Assuming that a damaged specimen generate a “loud” AE

signal, it will also generate many TTL counts with high peak amplitudes and many event counts. Thus as a rule of thumb: The feature value in a faulty specimen should be larger than the normal. In order to compute the feature thresholds, we compute the histograms of each feature on two sets of data, a normal and a faulty. Histograms are not directly well suited for classification tasks, so instead we use the normalized cumulated sum of the histogram, corresponding to a sample of the true cumulated density function (CDF). The CDF gives the probability that the feature is less or equal to  $\tau$ .

$$P(x \leq \tau) = \int_{x=-\infty}^{\tau} p(x) \quad (4)$$

A good feature for classification will behave such that  $P_{\text{normal}}(x \leq \tau) \gg P_{\text{faulty}}(x \leq \tau)$  is fulfilled, not everywhere but in an interval. Actually  $P_{\text{normal}}(x \leq \tau) = X, X \in [0; 1]$  says that  $X$  of the normal examples have a feature value  $x$  less or equal to  $\tau$ . Obviously this means that thresholding with this value of  $\tau$  corresponds to  $1 - X$  false alarms, and that  $1 - P_{\text{faulty}}(x \leq \tau)$  of the faulty specimens is correctly detected (see further [14]). Varying  $\tau$  from the smallest to the largest observed value whilst tabulating  $1 - P_{\text{normal}}(x \leq \tau)$  and  $1 - P_{\text{faulty}}(x \leq \tau)$  gives the Receiver Operator Characteristics (ROC) shown in Figure 1.

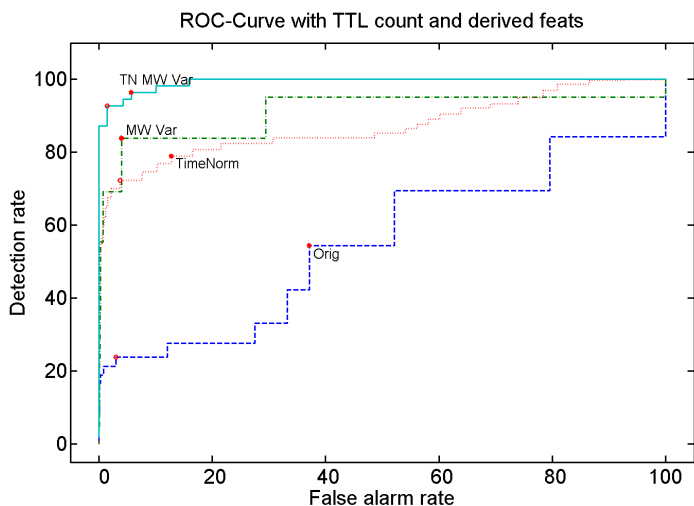


Figure 1: ROC-curves for the TTL count and the three derived features. The o's and stars on the curves indicate the performance of the two optimal thresholds. Notice the consensus between the different optimality criterions wrt. ranking of the features – also the Neyman-Pearson criterion at 5% agrees. The figure also show our conclusion, that the time normalization (data fusion) and moving window variance (processing) increase the performance of the system.

$$\text{Detection rate } D(\tau) = 1 - P_{\text{faulty}}(x \leq \tau) \quad (5)$$

$$\text{False alarm rate } F(\tau) = 1 - P_{\text{normal}}(x \leq \tau) \quad (6)$$

This approach with a single dimension classifier is very simple, with multiple dimensions, like using several of the obtained features, either pattern recognition or voting systems should be explored.

We adopt two rules for obtaining the optimal threshold:

1. Maximize the difference between the detection rate and false alarm rate
2. Minimize the absolute distance to the optimal point (100% detection, 0% false alarm)

Also applicable is the Neyman-Pearson Criterion[14], where the best classifier/threshold is the one with the highest detection rate given a set false alarm rate, say 5%.

Having decided on a threshold we also have knowledge of the false alarm rate of the classifier under normal conditions. In many cases occasional false alarms will occur, and we could filter out with a moving average low pass filter. The length of the filter and the significant number of alarms for the filtered signal can be obtained through the Binomial hypothesis testing (this is further described for a unsupervised setting in [9]).

In short we can vary the threshold in order to balance the rate of false alarms versus detection, however it is a trade-off. Direct improvement is only available through a better classifier. Another available trade-off is between classification accuracy and delay. If we know that the false alarm rate is 5 out of 100, we could wait until 12 out of 100 was classified as faulty; however this will introduce a delay since we need to gather enough fault-classifications.

**Results and Discussion:** Table 1 shows the statistical parameters for the ROC curves corresponding to the original features, i.e. those provided by the acquisition system.

Feature	false 1	detect 1	false 2	detect 2	Area
TTL count	3	24	37	54	0.55
HF amp	3	10	18	15	0.15
LF amp	0	2	25	8	0.07

Table 1: Original data

None of these features are able to discriminate well between normal and faulty signals under the assumption that faulty signals are louder. The LF amp with the assumption that normal signals are loud works quite well (area under ROC-curve 0.93), this could fit the “steady low amplitude” regime described under *time normalization*.

Applying a *moving window variance* on the original features the classification performances is considerably improved. Now take a look at the features generated by taking a moving window variance of the original data, shown in Table 2, then we have features with considerably better detection performance.

This increase in performance is in some cases followed by increased false detection rate. However, these features are generally better suited for classifying as indicated by the much larger area under the ROC curves. The performance of the time normalized data is shown in Table 3.

Feature	false 1	detect 1	false 2	detect 2	Area
TTL count	4	84	4	84	0.91
HF amp	16	87	16	87	0.84
LF amp	36	79	36	79	0.67

Table 2: Moving Window Variance of Original data

Feature	false 1	detect 1	false 2	detect 2	Area
TTL count	4	72	13	79	0.87
HF amp	11	49	22	56	0.60
LF amp	3	8	48	45	0.31
Event count	43	91	26	73	0.78

Table 3: Time Normalized data

These features result in better classification than with the original features (Table 1), however not as good as with the moving window variance. However the combination, by applying moving window variance on those features we hoped for similar improvement as was observed for the original data. Table 4 lists the ROC performance of the new set of features.

Feature	false 1	detect 1	false 2	detect 2	Area
TTL count	1	93	6	96	0.99
HF amp	13	93	13	93	0.92
LF amp	23	84	23	84	0.81
Event count	83	100	54	55	0.52

Table 4: Moving Window Variance of Time Normalized data

According to the Moving window variance of the normalized TTL counts is definitely a good feature.

Feature	false 1	detect 1	false 2	detect 2	Area
TTLxHF	13	56	13	56	0.54
TTLxLF	5	19	40	35	0.29
TTLx(HF+LF)	5	29	42	51	0.43

Table 5: Energy Normalized data

In order to see if the classification performance of the feature created by applying moving window variance on the normalized TTL counts could be improved further, a filtering was applied. As Table 6 shows that the filtering improved the false alarm rate compared to Table 4, but the area below the curve is slightly decreased as the detection rate is also decreased. Figure 2 shows the time series of both features derived from the TTL count as well as their respective classification. The various preprocessing steps reduce the local variance of the classification.

Feature	false 1	detect 1	false 2	detect 2	Area
NTTL variance	0	95	0	95	0.98

Table 6: Filtered moving window variance of the normalized TTL counts

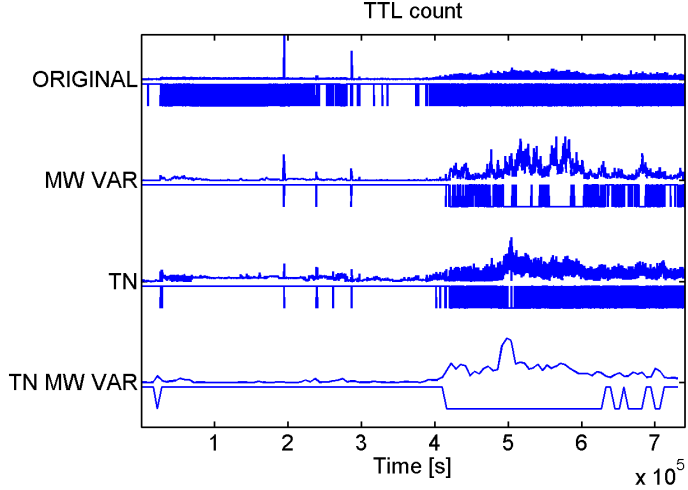


Figure 2: The time series of the TTL count and three derived features. Each entry in the features has been classified; this binary output is plotted just below the feature (the lower value is faulty, upper is normal). It is believed that the specimen turns faulty after  $4 \cdot 10^5$  s. The uncertainty using the original TTL count is clearly visible during under both normal and faulty conditions, as the classification switches all the time.

**Conclusion:** We have analyzed simple data fusion and preprocessing methods on Acoustic Emission measurements of prosthetic feet made of carbon fiber reinforced composites. This paper presents the initial research steps; aiming at reducing the time spent on the fatigue test. With a simple single feature probabilistic scheme we have showed that these methods can lead to increased classification performance. We conclude that: the derived features of the TTL count leads to improved classification and damage detection in a supervised setup.

The probabilistic classification scheme was founded on the histogram, however different approaches can readily be investigated using the improved features, possibly improving the performance using multiple features classifiers, e.g., Support Vector Machines and Gaussian Mixtures.

**Acknowledgements:** The authors gratefully acknowledge Össur hf. ([www.ossur.com](http://www.ossur.com)) for providing samples and access to their testing facility. The research of Runar Unnthorson was supported by a research grant from RANNIS, the Icelandic Research Council ([www.rann.is](http://www.rann.is)). His visit at DTU was supported by grants from the Helga Kristjánsdóttir



and Sigurliði Kristjánsson Memorial Fund and Landsvirkjun's ([www.lv.is](http://www.lv.is)). Niels H. Pontoppidan was supported by the European Commission through Growth Project AE-WATT ([mole.imm.dtu.dk/aewatt/](http://mole.imm.dtu.dk/aewatt/)).

## References:

- [1] J. Awerbuch and S. Ghaffari. Monitoring progression of matrix splitting during fatigue loading through acoustic emission in notched unidirectional graphite/epoxy composite. *Journal of Reinforced Plastics and Composites*, 7(3):245–64, 1988.
- [2] S. Barre and M.L. Benzeggagh. On the use of acoustic emission to investigate damage mechanism in glass-fibre-reinforced polypropylene. *Composites Science and Technology*, 52(3):369–376, 1994.
- [3] L.A. Duesing. Acoustic emission testing of composite materials. *Reliability and Maintainability Symposium, 1989. Proceedings., Annual*, pages 128–134, 1989.
- [4] H.L. Dunegan. An alternative to pencil lead breaks for simulation of acoustic emission signal sources, August 2000.
- [5] M. Giordano, A. Calabro, C. Esposito, A. D'Amore, and L. Nicolais. An acoustic-emission characterization of the failure modes in polymer-composite materials. *Composites Science and Technology*, 58(12):1923–1928, 1998.
- [6] N. Godin, S. Huguet, R. Gaertner, and L. Salmon. Clustering of acoustic emission signals collected during tensile tests on unidirectional glass/polyester composite using supervised and unsupervised classifiers. *NDT and E International*, 37(4):253–264, 2004.
- [7] Marvin A. Hamstad. Acoustic Emission Primer, Lecture delivered by Dr. Hamstad at the acoustic emission working group meeting on august 4, 2003 (aewg-46). internet, 2004. [http://www.engr.du.edu/profile/frame\\_bio.html](http://www.engr.du.edu/profile/frame_bio.html).
- [8] K. Ono. Acoustic emission behavior of flawed unidirectional carbon fiber-epoxy composites. *Journal of Reinforced Plastics and Composites*, 7(1):90–105, 1988.
- [9] Niels Henrik Pontoppidan, Jan Larsen, and Sigurdur Sigurdsson. Non-stationary Condition Monitoring with the AEWATT Toolbox. Submitted to MFPT 59, 2005.
- [10] W.H. Prosser, K.E. Jackson, S. Kellas, B.T. Smith, J. McKeon, and A. Friedman. Advanced waveform-based acoustic emission detection of matrix cracking in composites. *NDT and E International*, 30(2):108, 1997.
- [11] William H. Prosser. Applications of advanced waveform-based ae techniques for testing composite materials. *Proceedings of the SPIE conference on Nondestructive Evaluation Techniques for Aging Infrastructure and Manufacturing: Materials and Composites*, pages 146–153, 1996.
- [12] S.R. Ravishankar and C.R.L. Murthy. Characteristics of ae signals obtained during drilling composite laminates. *NDT and E International*, 33(5):341–348, 2000.
- [13] N. Sato and T. Kurauchi. Interpretation of acoustic emission signal from composite materials and its application to design of automotive composite components. *Research in Nondestructive Evaluation*, 9(3):119–136, 1997.

- [14] Clayton Scott and Jeffrey Silverman. The Newman-Pearson Criterion. internet, 2004.  
<http://cnx.rice.edu/content/m11548/latest>.
- [15] D. Tsamtsakis, M. Wevers, and P. De Meester. Acoustic emission from cfrp laminates during fatigue loading. *Journal of Reinforced Plastics and Composites*, 17(13):1185–1201, 1998.
- [16] M. Wevers. Listening to the sound of materials: acoustic emission for the analysis of material behaviour. *NDT and E International*, 30(2):99–106, 1997.

## APPENDIX E

# Non-stationary condition monitoring through event alignment

---

N. H. Pontoppidan and J. Larsen. Non-stationary condition monitoring through event alignment. In *IEEE Workshop on Machine Learning for Signal Processing*, pages 499–508, Piscataway, New Jersey, September 2004. IEEE Press. URL <http://isp.imm.dtu.dk/mlsp2004>



# NON-STATIONARY CONDITION MONITORING THROUGH EVENT ALIGNMENT

Niels Henrik Pontoppidan and Jan Larsen  
Informatics and Mathematical Modeling, Building 321  
Technical University of Denmark, DK-2800 Lyngby, Denmark  
Phone: +45 4525 3899,3923 – Fax: +45 4587 2599  
E-mail: nhp,jl@imm.dtu.dk – Web: isp.imm.dtu.dk

**Abstract.** We present an event alignment framework which enables change detection in non-stationary signals. Classical condition monitoring frameworks have been restrained to laboratory settings with stationary operating conditions, which are not resembling real world operation. In this paper we apply the technique for non-stationary condition monitoring of large diesel engines based on acoustical emission sensor signals. The performance of the event alignment is analyzed in an unsupervised probabilistic detection framework based on outlier detection with either Principal Component Analysis or Gaussian Processes modeling. We are especially interested in the true performance of the condition monitoring performance with mixed aligned and unaligned data, e.g. detection of fault condition of unaligned examples versus false alarms of aligned normal condition data. Further, we expect that the non-stationary model can be used for wear trending due to longer and continuous monitoring across operating condition changes.

## INTRODUCTION

We pursue Condition Monitoring (CM) systems which are capable of detecting faults in large diesel engines used for propulsion and power generation. Such operation involves frequent changes in either load or speed. The current problem is that those trivial changes result in false alarms that cannot be separated from alarms originating from real faults.

MAN B&W Diesel has conducted experiments simulating realistic marine operation with multiple loads. Faults resembling scuffing was induced by means of shutting the lubricating oil system off after the engine was started. Scuffing is a severe fault that with time evolves into damaging contact between cylinder piston and liner. In the acquired data set we have have identi-

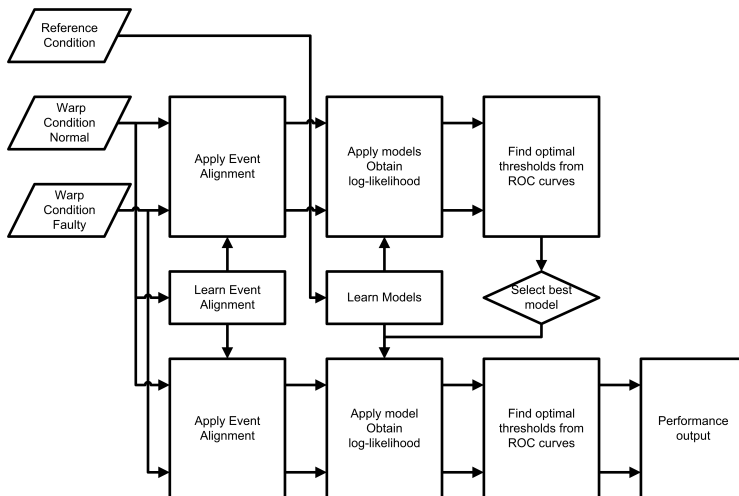


Figure 1: System overview. The figure outlines the flow of information during the experiments.

fied a stable functional dependency between signals from different operational conditions. We propose a novel method that builds invariance into the Condition Monitoring System (CMS) by inverting those functional changes prior to outlier detection. The traditional approach of sampling of data in the crank angular domain [1] is not enough to remove those changes. Further the available Dynamic Time Warp algorithm [3, 9] was discarded since repeated time frames produced fault-like signals.

We have previously cast unsupervised condition monitoring as an outlier detection problem with generative models [10]. The generative models allows for localization of large deviations that indicates the origin of the fault. With this setup we have successfully detected induced scuffing (piston rubbing against the liner) and externally generated faults under stationary conditions. However, we were not able to distinguish between alarms due to faults or operational changes under non-stationary conditions. Also, other recent monitoring applications [2, 5] have been limited to fixed operating conditions.

## MODELING

The following section presents the data setup and the describe the use of Principal Component Analysis and Gaussian Processes for modeling in a condition monitoring framework.

**Data setup.** Data was acquired on MAN B&W Diesel’s two-stroke test bed engine, under controlled varying conditions. The acquired ultrasonic acoustic emission signals were preprocessed by short time root mean square and converted from time domain into the crank angular domain using a crank

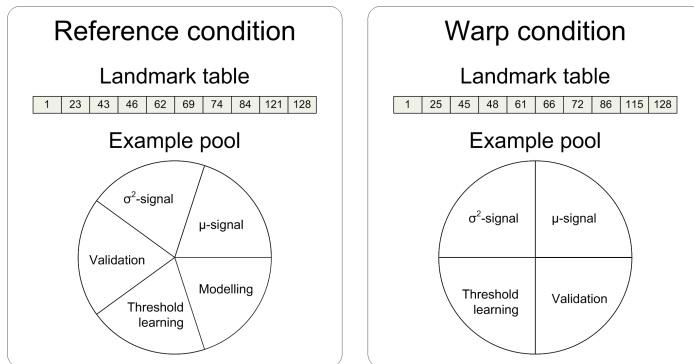


Figure 2: Data partitioning for *Event alignment* and *Change detection*. Different sizes of partions are allowed. We use different sets of data to learn the warping parameters ( $\mu$  and  $\sigma^2$ -signals), rejection threshold. Further we also learn the model parameters from a set of data from the reference condition. Finally we use unused examples for validation of the performance.

angle tachometer. Further the signals were downsampled to enable Gaussian Process modeling. The properties of each RMS AE signal,  $\mathbf{x}$ , is:  $d$  non-negative elements sampled at specific angular positions at constant angular sample rate regardless of engine operating conditions. Figure 1 show the system overview explained in this section. We center data (denoted  $\tilde{\mathbf{x}}$ ) before modeling by subtracting the mean obtained across a subset of Normal Condition (NC) examples. Throughout the experiments we estimate parameters in a step-wise manner using resampling. As shown in Figure 2 a specific data subset used to learn a parameter is only used one time. Two modeling schemes for the analysis of the performance of the event alignment are deployed. The approach is a mixture of supervised and unsupervised learning. Unsupervised modeling is used to model NC data by training a set of parameters  $\theta$  in Principal Component Analysis (PCA) and Gaussian Process (GP) models described below. The log-likelihood of the NC model is used as a measure of how much an example belongs to a model, and the log-likelihood density of NC and Faulty Condition (FC) examples in general separates. A rejection threshold is obtained in a supervised manner by finding selecting an optimal point on the Receiver Operator Characteristics Curve from a set of labeled NC and FC examples.

**Principal Component Analysis Model.** From a set of  $N$  centered normal examples  $\tilde{\mathbf{x}}$  (size  $d \times 1$ ) we build the training matrix  $\mathbf{X}_{\mathcal{T}}$  by stacking (size  $d \times N$ ).  $\mathbf{X}_{\mathcal{T}} = [\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots, \tilde{\mathbf{x}}_N]$ . From this training matrix we estimate a principal component matrix **PC** (size  $N \times N$ ), and a projection matrix  $\mathbf{U}$  (size  $d \times N$ ) through the Singular Value Decomposition (SVD)  $\mathbf{X}_{\mathcal{T}} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^{\top}$ . The number of principal components  $k$  is controlled by using the first  $k$  columns of  $\mathbf{U}$  and ( $k$ ) rows of **PC**.

$$\mathbf{PC} = \mathbf{\Lambda}\mathbf{V}^{\top}$$

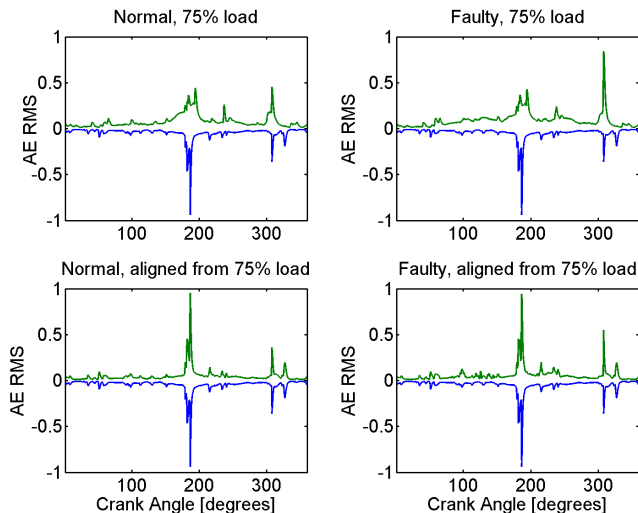


Figure 3: Application of the event alignment. The normal condition signal is displayed with negative sign. Normal and faulty condition 75% load data were event aligned using the warp for normal condition 75% load data. Since the faulty condition 75% load does not comply with this model, the aligned faulty condition examples display deviation around 100-150 degrees.

When applying the PCA model to new examples we multiply with the first  $k$  transposed columns of  $\mathbf{U}$  from the left and obtain  $\mathbf{s}_k$  plus the noise  $\epsilon$ , as the remaining  $d - k$  components that span a Gaussian noise space [8, 10].

$$\mathbf{s}_k = \mathbf{U}_k^\top \tilde{\mathbf{x}} \quad (1)$$

$$\tilde{\mathbf{x}} = \mathbf{U}_k \mathbf{s}_k + \epsilon \quad (2)$$

It follows directly from the properties of the SVD and (2) that the principal components of NC examples follow a multivariate zero mean Gaussian with covariance  $\mathbf{\Lambda}_k$  (using the first  $k$  columns and rows of  $\mathbf{\Lambda}$ ). Let  $\boldsymbol{\theta}$  denote all estimated parameters [8, 10], then  $p(\mathbf{x}|\boldsymbol{\theta}, k) = p(\mathbf{s}_k|\boldsymbol{\theta})p(\epsilon|\boldsymbol{\theta})$  and the log-likelihood is  $\mathcal{L} = \log p(\mathbf{s}_k|\boldsymbol{\theta}) + \log p(\epsilon|\boldsymbol{\theta})$

**Gaussian Process Model.** As an alternative to the PCA subspace model we can perform modeling directly in the observed domain. Through Gaussian Process (GP) modeling we obtain a measure of how much an example deviates from the reference condition. From Gibbs and MacKay [7] we have (with interchanged  $\mathbf{t}$  and  $\mathbf{x}$  relative to Gibbs MacKay notation)

$$Q(t_i, t_j) = \theta_2^2 \exp\left(-\frac{(t_i - t_j)^2}{2\theta_1^2}\right) + \theta_3^2 \delta(t_i, t_j) \quad (3)$$

$$p(\tilde{\mathbf{x}}|\mathbf{Q}, \mathbf{t}) = \frac{1}{Z} \exp\left\{-\frac{1}{2}\tilde{\mathbf{x}}^\top \mathbf{Q}^{-1}\tilde{\mathbf{x}}\right\}, \quad (4)$$



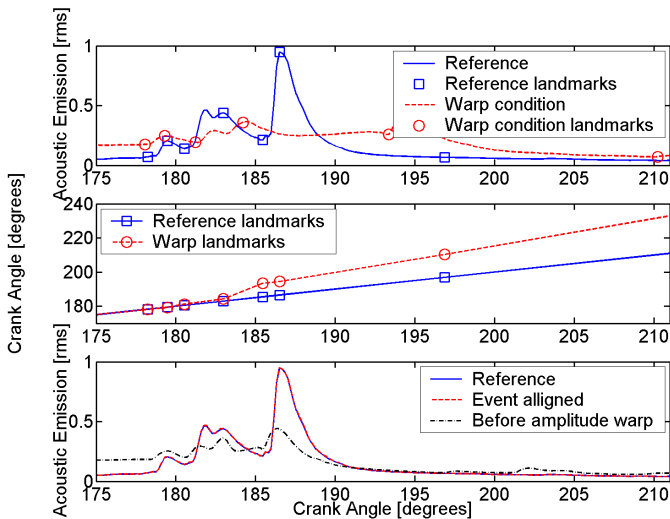


Figure 4: Examples and landmarks during injection period. The upper figure displays the mean signals and landmarks for the two conditions. The middle figure displays the landmarks. In the beginning they are almost equal and the warp condition evolves slower than the reference condition. The lower figure displays the mean signal before amplitude warp, and the mean signal of the fully event aligned. The mean signals are shown before, during and after event alignment. Notice that the event alignment results in equal mean signals.

where  $\mathbf{t}$  is the vector of crank positions, and  $\tilde{\mathbf{x}}$  is the corresponding observed centered values. The covariance matrix  $\mathbf{Q}$  is a function of the index vector  $\mathbf{t}$  and the parameters  $\boldsymbol{\theta}$ . The last term of Equation 3 is the noise part. The negative log-likelihood for the example  $\tilde{\mathbf{x}}$  given the parameters  $\boldsymbol{\theta}$  defining the covariance matrix  $\mathbf{Q}$  is

$$\mathcal{L} = \frac{1}{2} \log |\mathbf{Q}| + \frac{1}{2} \tilde{\mathbf{x}}^\top \mathbf{Q}^{-1} \tilde{\mathbf{x}} \quad (5)$$

For each training example  $\tilde{\mathbf{x}}_{n_\theta}, n_\theta \in N_\theta$  we train an independent GP with parameters  $\{\boldsymbol{\theta}_{n_\theta}\}$  through minimization of  $\mathcal{L}$  using `minimize.m` [11]. Finally we perform average over the parameters obtained from different training examples to obtain the final model parameters.

$$\hat{\boldsymbol{\theta}}_{n_\theta} = \arg \min_{\boldsymbol{\theta}} \left\{ \frac{1}{2} \log |\mathbf{Q}| + \frac{1}{2} \tilde{\mathbf{x}}_{n_\theta}^\top \mathbf{Q}^{-1} \tilde{\mathbf{x}}_{n_\theta} \right\} \quad (6)$$

$$\hat{\boldsymbol{\theta}} = \log \left( \frac{1}{\text{card}(N_\theta)} \sum_{n_\theta \in N_\theta} \exp(\hat{\boldsymbol{\theta}}_{n_\theta}) \right) \quad (7)$$

In order to ensure positive parameters without enforcing constraints `minimize.m` uses reparameterization, hence, the averaging takes place in the natural parameter space and explains the `exp` and `log` in (7). The original implementation of the Gaussian Processes was due to Carl Rasmussen [11], but the we

have customized the input/output structure to fulfill our needs, e.g., allowing the training and use of the  $\mathbf{Q}$  matrix.

## DETECTION

Outlier detection with log-likelihood is based that NC and FC examples separate in log-likelihood space. For instance we expect that the number and/or characteristics of the underlying hidden sources are changed when entering the FC, thus examples acquired from a FC are poorly described by a model trained on NC examples. We expect that combinations of increased noise or increased strength of certain acoustical sources results in a lower log-likelihood value. From a set of labeled preprocessed examples we build the accumulated densities for the features  $p(\mathcal{L}_{\text{NC}} \leq \tau)$  and  $p(\mathcal{L}_{\text{FC}} \leq \tau)$ . Each value of  $\tau$  corresponds to a *true detection / false alarm* ratio, and we choose the optimal rejection threshold  $\hat{\tau}$  that is closest in distance to 100% detection and 0% false alarms. The threshold can also be obtained in other manners, e.g., selecting the threshold that detects most faults with a constant false alarm rate.

## EVENT ALIGNMENT (WARPING)

We present *event alignment* as a novel tool for non-stationary Condition Monitoring (CM) of large marine diesel engines. The tool is necessary since the current CMSs are not invariant to certain known operational changes - in particular load/speed changes. Given two different NC, the event alignment transforms examples from one condition into the other condition, thus facilitating a CMS trained on the reference system to correctly detect deviations under both conditions. The result is that the CMS becomes invariant to changes between the two NC's. With more NC's we expect that interpolation between a few warps is possible. Non-stationary condition monitoring is important when considering diesel engines since the operating conditions change frequently. Under normal marine conditions Frances *et al.* [6] have observed large variability. In our data sets we have found that such variability is largely described by the changing operation conditions, indicating that unwanted false alarms could be suppressed by adopting to the changes invoked by the operating conditions. It should also be noted that application of the same model on a continuous flow of data could allow for trending of wear, that is not necessarily possible with multiple models, as models might focus on different properties of the condition modes.

Obviously one should take care that examples which do not originate from the warp condition are not transformed into the reference condition. For instance the event alignment should try to preserve the same variations as in the reference condition, as this prevents the event aligned examples of becoming super-normal. Furthermore it prevents examples from other

conditions of being transformed into the reference condition. Overfitting with event alignment is still an open issue which needs further research.

Dynamic Time Warp based on phase vocoder techniques [4] described by Ellis [3] and Keough [9] performs the time-warp while keeping the frequency content unchanged. The phase vocoder is based on short time Fourier transformation and accomplishes the time-warp by interchanging the number of samples between overlapping time frames at playback time, e.g., moving the overlapping windows further apart in order to stretch the signal. Dynamic Time Warping is uninteresting for CM in the time domain, as it repeats or drops time frames if necessary, possibly duplicating peaks or removing fault signatures. Instead we decide on time-warps that keep the waveform structure or envelope unaltered, e.g. spline interpolation. Also piece-wise linear interpolation was tried, but in the present case the cubic splines provided better results.

### Event alignment model

The event alignment consists of two non-linear warps, the first performs time-alignment and the second performs amplitude mapping. Definitions (also see Figure 2)

$\mathbf{L}$  The landmark vector defines the angular position of the important events. Most events are described by three landmarks: begin, peak and end.

$\boldsymbol{\mu}$  the vector containing the mean signal (across examples for each angular position)

$\boldsymbol{\sigma}^2$  the vector containing the variation around  $\boldsymbol{\mu}$ .

The landmarks were picked by hand, and are very specific for the application. Even changing a sensor position would change the landmarks. Thus, automatic identification of landmarks is to be addressed in future studies. The event alignment transforms warp condition examples  $\mathbf{x}_W$  described by  $\boldsymbol{\mu}_W, \boldsymbol{\sigma}_W^2$  and landmarks  $\mathbf{L}_W$  into aligned examples  $\mathbf{x}_A$  resembling the reference condition  $\boldsymbol{\mu}_R, \boldsymbol{\sigma}_R^2$  and  $\mathbf{L}_R$ .  $f(\cdot)$  is an interpolating function that performs the time alignment of events based on the two set of landmarks. The vector  $\mathbf{g}$  is a sample-wise constrained re-scaling factor that accounts for compression of variance when the variance in the warp condition is larger than in the reference condition.

$$\mathbf{x}_A = (f(\mathbf{x}_W, \mathbf{L}_R, \mathbf{L}_W) - \boldsymbol{\mu}_W) .* \mathbf{g} + \boldsymbol{\mu}_R, \quad (8)$$

$$\boldsymbol{\mu}_W = \langle f(\mathbf{x}_W, \mathbf{L}_R, \mathbf{L}_W) \rangle \quad (9)$$

$$g_i = \begin{cases} 1 & , \sigma_{iR} > \sigma_{Wi} \\ \sigma_{Ri}/\sigma_{Wi} & , \sigma_{Ri} > \sigma_{Wi} \end{cases}, \quad i = 1, 2, \dots, d \quad (10)$$

where  $.*$  denoting Hadamard matrix multiplication. The constraint prevents amplification of measurement noise. Unconstrained re-scaling can lead to negative values that do not correspond to the non-negative RMS signals.

In some cases this constraint leads to overfitting, as the aligned examples become “more” normal than the un-aligned examples. In the following section we encounter this problem in experiment 5 for Gaussian Process modeling.

## EXPERIMENTS AND RESULTS

We create pseudo-realistic data sets in order to compensate for lack of data by resampling of examples within periods of stable conditions. Examples are resampled by drawing random examples from pools of data and only used once. That is, examples used to learn the model, warp or threshold are not used during performance evaluation. Resampling of examples facilitates evaluation and analysis of the models at the expense that condition changes become more abrupt, thus analysis of alarm time and trending is not possible.

We measure the performance of event alignment on its ability to correctly separate FC and NC examples during changing operational conditions.

All experiments (see Table 1) were conducted using a model trained on random examples drawn from the 25% load NC. Table 2 reports the obtained detection rates using the two different modeling schemes. For PCA only the performance with the optimal number of components is reported. Experiment 1 shows the performance of the stationary system on stationary data. Applying the stationary system to non-stationary data would label all normal conditions as faulty since the CMS cannot discriminate between normal variations and true faults, thus the resembling the non-stationary conditions is indeed promising.

Experiment 2 and 3 demonstrate the ability to align other NC with the reference condition while the event alignment of FC examples using the same model are correctly labeled as faulty. In experiment 2 we obtain the same performance as the stationary system, but in experiment 3 the performance is degraded. This is due to the downsampling of examples. The original AE RMS vectors had  $d = 2048$  samples per revolution, and since the training of the Gaussian Process model involves inversion of  $d \times d$  square matrix, all examples have been downsampled with a factor 8. The fault leads to unstable timing of events and the downsampling smears out these changes. We notice that the PCA preprocessing suffers more from downsampling than the GP model, however, without any downsampling, PCA also yields 80-90% detection rate and 15-20% false alarm rate (similar to the GP with downsampled data). As expected, the overall performance is reduced in comparison with the stationary experiment.

Experiment 4 demonstrates how the non-stationary system is able to discriminate between aligned NC data and un-aligned FC data. The result is similar to that of the stationary system.

In experiment 5 we test the CMS w.r.t. overfitting. We cheat the system and take examples warped into the normal condition as normal and un-warped NC data as “faulty”. The overfitting in the event alignment, i.e., the examples are warped into being super-normal, is detected with the GP

Exp	Normal data	Faulty data
1	25 % load, lube oil on	25 % load, lube oil off
2	75 % load, lube oil on	75 % load, lube oil off
3	50 % load, lube oil "on"	50% load, unstable speed
4	75 % load, lube oil on	25 % load, lube oil off
5	50 % load, lube oil "on"	25 % load, lube oil on
6	Mixed loads, lube oil on	Mixed loads, lube oil off

TABLE 1: List of experiments. During experiment 3 and 5 examples acquired without lube oil was warped into the reference condition with lube oil.

Exp	PCA	Detec/False	# Comp	GP	Detec/False
1		95/ 5	2		95/ 5
2		95/ 5	2		95/ 5
3		60-65/20-30	3		80-90/15-20
4		95/ 5	2		95/ 5
5		50-60/65-80	36		80-95/ 0-15
6		95/ 5	2		95/ 5

TABLE 2: Condition Monitoring Performance. Detec/False denotes detection versus false alarms rate in percentage.

modeling, that incorrectly label 80-95% of the NC examples as FC. However, using PCA modeling the overfitting disappears, even though the PCA uses much more components than usually, indicating that it is looking for very small changes.

In experiment 6 we collect both aligned and un-aligned examples from experiment 1,2 and 4 in order to demonstrate that the system is capable of performing non-stationary condition monitoring with the same performance as in the individual experiments. This demonstrates that the obtained optimal rejection thresholds are stable in the three experiments 1,2, and 4.

## CONCLUSION

The experiments show that non-stationary condition monitoring is indeed possible. It is important to notice that the event alignment does not decrease the overall condition monitoring performance as the results obtained in experiment 2 and 4 are equal to the stationary results in experiment 1. Furthermore, the performance obtained using both mixed aligned and original data in experiment 6 is the same as in the individual experiments, indicating that the optimal rejection thresholds are fairly constant even with several warp conditions each having its own set of event alignment parameters. The conclusion is that non-stationary CM indeed can be obtained by extending a stationary CMS with event alignment.

Future work will concentrate on refining the method to handle a larger range operation conditions, automatic detection of landmarks, and further

investigations related to overfitting. In addition, we will evaluate whether the framework will allow for wear trending, which of course calls for new experiments involving much larger time scales. We will also pursue fast Gaussian Processes in order to avoid signal downsampling.

## ACKNOWLEDGEMENTS

The work is supported by EU Competitive and Sustainable Growth Programme GRD2-2001-50014 through the AE-WATT project. Data was provided by MAN B&W Diesel A/S. Discussions with Sigurdur Sigurdsson and Mads Dyrholm as well as project partners at Heriot-Watt University were very much appreciated.

## REFERENCES

- [1] G. Chandroth and A. Sharkey, "Utilising the rotational motion of machinery in a high resolution data acquisition system," in **Proc of Computers and ships**, May 1999.
- [2] G. Chandroth, A. Sharkey and N. Sharkey, "Vibration signatures, wavelets and principal components analysis in diesel engine diagnostics," in **Marine Technology Odra 99**, Oct. 1999.
- [3] D. Ellis, "Dynamic Time Warp (DTW) in Matlab," Internet, <http://labrosa.ee.columbia.edu/matlab/dtw/>.
- [4] J. Flanagan and R. Golden, "Phase Vocoder," **Bell System Technical Journal**, pp. 1493–1509, Nov. 1966.
- [5] T. Fog, L. Hansen and J. Larsen et al., "On Condition Monitoring of Exhaust Valves in Marine Diesel Engines," in Y. H. Hu, J. Larsen, E. Wilson and S. Douglas (eds.), **Proceedings of the NNSP IX**, IEEE, Piscataway, New Jersey, 1999, pp. 225–234.
- [6] A. Frances, J. Gill, J. Reuben and J. Steel, "A Study of the Variability of Acoustic Emission Signals from a Medium Size Marine Diesel Engine under Service Conditions," in O. P. Shrivastav, B. Al-Najjar and R. B. Rao (eds.), **COMADEM 2003**, COMADEM International, Växjö, 2003, pp. 503–512.
- [7] M. N. Gibbs and D. J. C. MacKay, "Efficient implementation of Gaussian Processes for Interpolation," <http://www.inference.phy.cam.ac.uk/mng10/GP/gpros.ps.gz>, 1997.
- [8] L. Hansen and J. Larsen, "Unsupervised Learning and Generalization," in **Proceedings of the IEEE Int. Conf. on Neural Networks 1996**, 1996, vol. 1, pp. 25–30.
- [9] E. Keogh, "Exact Indexing of Dynamic Time Warping," in **Proceedings of the 28th VLDB Conference, Hong Kong, China**, 2002.
- [10] N. Pontoppidan and J. Larsen, "Unsupervised Condition Change Detection In Large Diesel Engines," in C. Molina, T. Adali, J. Larsen, M. Van Hulle, S. Douglas and J. Rouat (eds.), **2003 IEEE NNSP Workshop**, Piscataway, New Jersey: IEEE Press, Sept. 2003, pp. 565–574.
- [11] C. E. Rasmussen, "minimize.m, gps00.m," <http://www.kyb.tuebingen.mpg.de/~carl>, 2001,2003.

APPENDIX F

# Event alignment, warping between running speeds

---

Niels Henrik Pontoppidan and Ryan Douglas. Event alignment, warping between running speeds. In [Rao et al. \[2004\]](#), pages 621–628. ISBN 0-954 1307-1-5. URL <http://www.imm.dtu.dk/pubdb/p.php?3111>





## EVENT ALIGNMENT, WARPING BETWEEN RUNNING SPEEDS

Niels Henrik Pontoppidan<sup>1</sup> and Ryan Douglas<sup>2</sup>

<sup>1</sup>Informatics and Mathematical Modelling, Richard Petersens Plads 321, Technical University of Denmark, 2800 Lyngby, Denmark. [nhp@imm.dtu.dk](mailto:nhp@imm.dtu.dk), Tel +45 4525 3899, Fax +45 4587 2599

<sup>2</sup>Dept. of Mechanical & Chemical Engineering, Heriot-Watt University, Riccarton, EH14 4AS, Edinburgh, United Kingdom. [r.douglas@hw.ac.uk](mailto:r.douglas@hw.ac.uk), Tel +44 131 451 4569, Fax +44 131 451 3129

### ABSTRACT

We are pursuing a system that monitors the engine condition under multiple load settings, i.e. under non-stationary operating conditions. We have obtained data from the electronically controlled 2-stroke engine at MAN B&W Research Copenhagen. The running speed when data acquired under simulated marine conditions (different load settings on the propeller curve) was in the range from 60 to 120 rotations per minute; furthermore the running speed was stable within periods of fixed load.

Electronically controlled engines can change the angular timing of certain events, such as fuel injection in order to optimize its performance. However this behaviour inhibits our framework presented in COMADEM 2003 from detecting condition changes across those load changes.

This paper evaluates different methods that align acoustic emission signals observed under different load settings. We evaluate the methods on data from the fuel injection period where the largest deviations in timing occur.

The idea is that we, given aligned data, can use the already developed component analysis framework for non-stationary monitoring of condition changes. It should further be noticed that the proposed warp framework also enables alignment across cylinders and engines.

### KEYWORDS

Event alignment, signal processing, non-stationary condition monitoring, acoustic emission.

### INTRODUCTION

We have obtained acoustic emission (AE) RMS signals from the cylinder liner and cover of the electronically controlled 2-stroke at MAN B&W Research Copenhagen. During the acquisition the running speed was in the range 60-120 rotations per minute. Further the running speed was virtually constant during periods of constant load settings.

Up to now research has mainly focused on condition monitoring under fixed operational conditions, see further [1], [2] and [3]. We are currently pursuing non-stationary condition monitoring, i.e. condition monitoring under different load settings that should resemble realistic marine conditions. Electronically controlled engines can change the angular timing of certain events, such as fuel injection in order to optimize its performance. However this behaviour inhibits our framework presented in COMADEM 2003 [1] from detecting condition changes across those load changes. The result is a false alarm triggered by the condition change. Also mechanically controlled engines display such

variations[4], due to the fact that some events have fixed length in time and some in angular “time”, thus it is not sufficient to use the crank angular domain as described in [5] to overcome this problem. Joint research in the AE-WATT project has revealed a stable functional dependence in the observed AE signals w.r.t. running speed/load, which this paper exploits in order to compare AE signals observed under different load settings. We expect to add this novel tool to our component analysis framework [1] enabling non-stationary condition monitoring.

**Timing changes during injection period**

The three events depicted in **Figure 1** are believed to arise from mechanical interaction between the injector spindles and their respective stops within the injector, with fuel delivery occurring between the region encompassing the first and second peaks and the last peak. The process is partly mechanically controlled by pre-set spring pressures and partly electronically controlled since the fuel flow to the injector is electronically governed.

In order to meet an increased load the engine response is to inject more fuel. This is achieved by prolonging the fuel delivery period with consequential retarded closure of the injector. Since the AE directly reflects the mechanical operations within the injector the increased fuel injection duration is readily identifiable.

Just as the engine changes the timing of the events, we are going to undo those changes. Figure 1 shows the mean injection period signals at three different loads on the propeller curve. All loads have been annotated with a set of event landmarks. The following sections describe the applied method and the result is the alignment of the 50% and 25% load data.

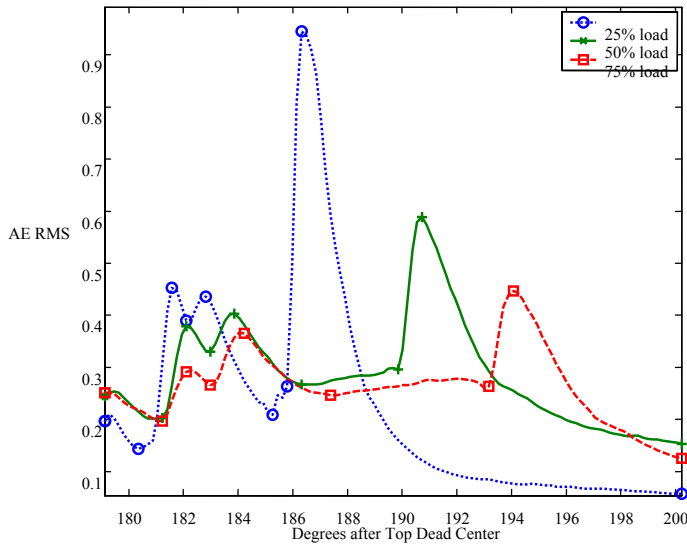


Figure 1: Mean Acoustic emission signals during injection period with different load settings. The markers show the time position of the landmarks that should be aligned. Notice that the TDC refererrs to another cylinder 180° degrees out of phase.

## METHODOLOGY

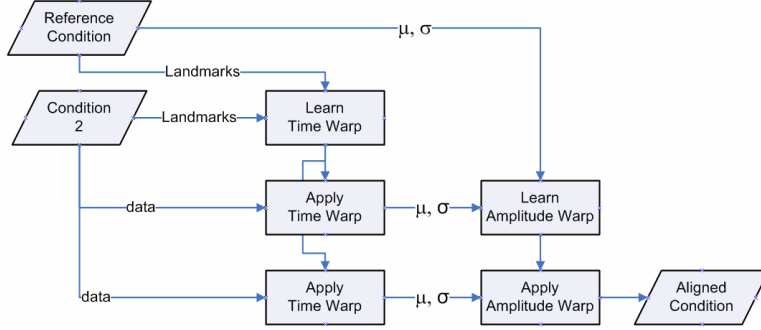


Figure 2: Outline of event alignment algorithm

Equation 1 and 2 define the warping of the observed signal  $x_2[n]$  into the aligned signal  $x_A[n]$ . The first step is applying the time-warp function  $f()$ , i.e., a function that aligns the landmarks and events of the two conditions in time. This possibly leaves amplitude mismatch which is resolved by subtracting the “other condition” mean  $\mu_2[n]$ , followed by compression of variance  $g[n]$ , ending with addition of the reference mean  $\mu_R[n]$  (see step 1-3 in figure 2).

$$x_A[n] = (f(x_2[n]) - \mu_2[n])g[n] + \mu_R[n] \quad (1)$$

$$g[n] = \begin{cases} 1 & \sigma_2[n] > \sigma_R[n] \\ \sigma_2[n] / \sigma_E[n] & \sigma_2[n] \leq \sigma_R[n] \end{cases} \quad (2)$$

We use data to learn the parameters of the event alignment. In order to ensure generability of the algorithm we obtain individual subsets for the learning of each function, i.e. we randomly select some examples that we learn the respective landmarks from, another set for the respective mean-signals and yet another set for the variance.

### Warp path

The function  $f()$  describes the warp-path[6], i.e. a time-stretching function. An example of a warp path is shown in Figure 3. The local slopes correspond to the necessary local (reciprocal) time-stretching. Depending on how the warp path is obtained a set of constraints can be defined, e.g. not allowing negative slope etc. The dark rhomb in the figure is the Itakura-parallelogram[7], which is one of normally applied constraints. We have applied another constraint namely the landmarks, which we obtain from analysing the engine. Simply if  $f()$  aligns the landmarks it also aligns the signals..

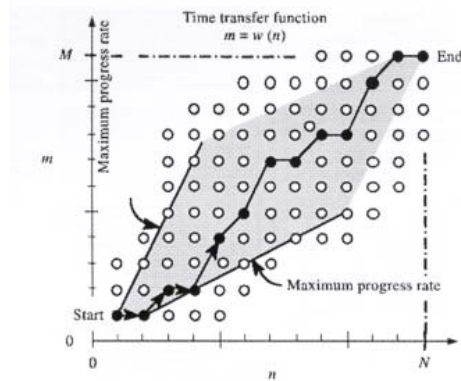


Figure 3: A warp path. Figure due to Leonard et al. [6]. The local slopes correspond to the necessary local (reciprocal) time-stretching.

#### *Dynamic Time Warping based on Phase Vocoder Techniques*

Dynamic Time Warping (DTW) has successfully been applied to alignment of speech segments [8]. In DTW the actual time alignment is performed using Phase Vocoders [9]. The Phase Vocoder (PV) alters the time duration of a sequence whilst keeping the frequency information literally unchanged[10], i.e., playing speech at a faster rate without the well known chip-monk effect. However for alignment of signals in a component analysis based framework as ours, the artefact of spurious peaks is problematic. What happens is that the PV in some cases repeats or skips frames of observed signal, possibly removing or repeating the, for us important events. Thus using DTW for the functional form of  $f()$  in Eqn. (1) was abandoned.

#### *Spline interpolation in time domain*

By allowing changes in the frequency content can use spline-interpolation in the time-domain. We have tested 2 types of splines, piecewise linear (1<sup>st</sup> order) and cubic (3<sup>rd</sup> order) splines. In many cases the cubic interpolation is better, as the derivatives of the warp-path are continuous. This means that the time-stretch at the landmarks is smoother. Sometimes, especially if landmarks are close to each other, cubic interpolation can lead to negative slope. This is an issue that we will have to investigate further, most likely ending up with a constrained regression scheme.

#### *Amplitude warp*

The function  $g[n]$  is only allowed to compress variance, since we cannot determine the source of the observed variance. Is the observed variance due to mode variation or measurement noise? Indeed amplification of measurement noise would be wrong. In experiments with unconstrained  $g[n]$  we observed that amplification of measurement noise lead to negative values – remember the observed signals are non-negative RMS signals. On the other side the constraint also keep the variance after alignment lower or equal to the variance in the un-aligned data, thus the aligned examples seem more “normal” than the un-aligned; this is called over-fitting an important issue that we will investigate further.

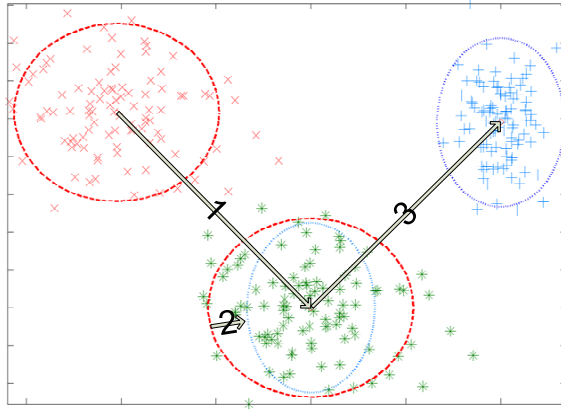


Figure 4: Example of amplitude warp. Samples from a two-dimensional i.i.d. Gaussian are translated, scaled and translated, i.e., removing mode mean, re-scaling variance and adding reference mean.

**Example of event alignment**

Figure 5, Figure 6 and Figure 7 show how 50% load data is event alignment into resembling 25% load data. Figure 5 show the data after time-stretching, as expected the landmarks, peaks and valleys are aligned but we notice the prominent amplitude mismatch. Figure 6 shows the data means after amplitude warp – they are identical. Notice that this is even though another set of examples was used to learn the parameters as Figure 2 indicate. Figure 7 displays the result of applying the event alignment to a set of 50% load examples, again another set of examples was used to learn the parameters.

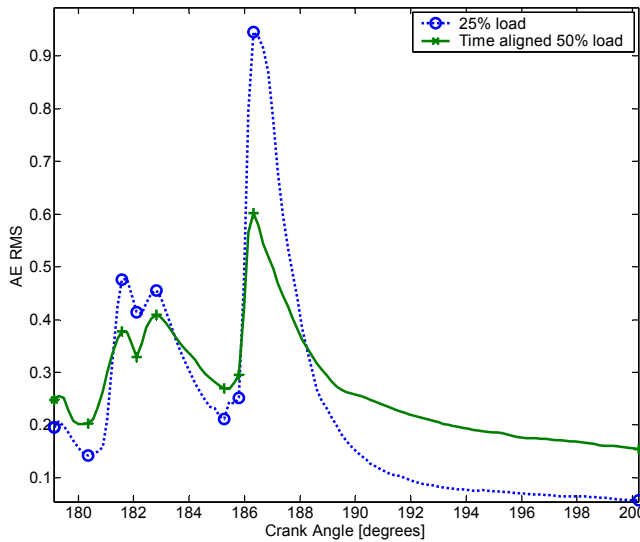


Figure 5: Amplitude mismatch after time warp. The two displayed signals are meaned over 30 cycles.

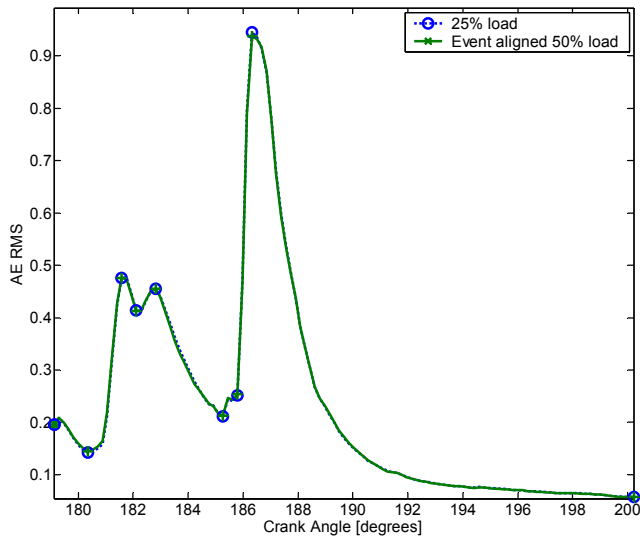


Figure 6: The event alignment provides a perfect match of the mean signals of 25% load data and aligned 50% data.

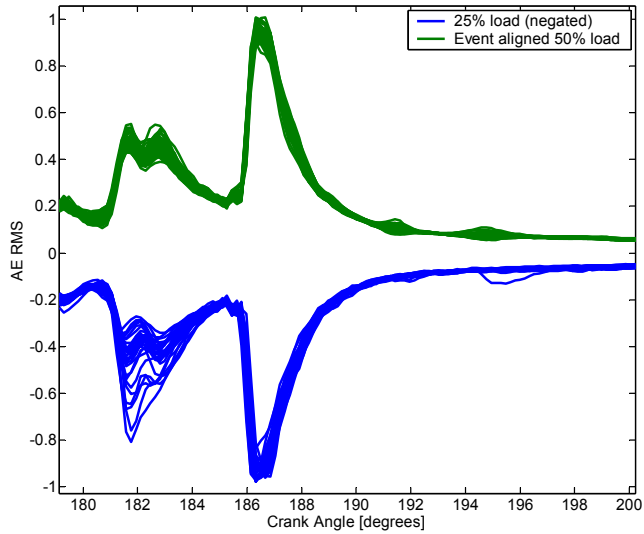


Figure 7: Examples of negated 25% load data and 50% load data. Notice the lesser amount of variance around 182° in the aligned data.

## CONCLUSION

We have demonstrated how knowledge of engine events can be used to turn data acquired under one operational condition into resembling another also known condition. We believe that this approach enables condition monitoring across known condition changes and thus enables non-stationary condition monitoring. Non-stationarity is a key component in our research for reliable condition monitoring under marine conditions, and we will continue this research and conduct the necessary experiments with full cycle data that demonstrate the non-stationary behaviour of the whole condition monitoring system. Another line of work is automatic identification of events where our research indicates that other sensor positions, namely close to the injector could provide better resolution w.r.t. events.

## ACKNOWLEDGEMENTS

The work is supported by EU Competitive and Sustainable Growth Programme GRD2-2001-50014 – the AE-WATT project. Data was provided by MAN B&W Diesel A/S. Discussions with other project partners was very appreciated.

## REFERENCES

- [1] Pontoppidan, N. H., Larsen, J., Fog, T., “Independent component analysis for detection of condition changes in large diesels” in COMADEM 2003, no. 16, pp. 493-502, 2003.
- [2] G. Chandroth, A. Sharkey and N. Sharkey, “Vibration signatures, wavelets and principal components analysis in diesel engine diagnostics” in Marine Technology Odra 99, Oct. 1999.
- [3] T. Fog, L. Hansen, J. Larsen, H. Hansen, L. Madsen, P. Sørensen, E. Hansen and P. Pedersen, “On Condition Monitoring of Exhaust Valves in Marine Diesel Engines,” in Y. H. Hu, J. Larsen, E. Wilson and S. Douglas (eds.), Proceedings of the IEEE Workshop on Neural Networks for Signal Processing IX, IEEE, Piscataway, New Jersey, 1999, pp. 225–234.
- [4] A. Frances, J. Gill, J. Reuben and J. Steel, “A Study of the Variability of Acoustic Emission Signals from a Medium Size Marine Diesel Engine under Service Conditions,” in O. P. Shrivastav, B. Al-Najjar and R. B. Rao (eds.), COMADEM 2003, COMADEM International, 2003, pp. 503–512.
- [5] G. Chandroth and A. Sharkey. Utilising the rotational motion of machinery in a high resolution data acquisition system. In Proc of Computers and ships- from ship design and build, through automation and management and on to ship support, May 1999.
- [6] Francois Leonard, Marc Foata, and Jean-Yves Paquin. Vibro-acoustic signature comparison and time-warping correction with multi-scale correlation. Mechanical Systems and Signal Processing, 14(3):443–458, 2000.
- [7] E. Keogh, “Exact Indexing of Dynamic Time Warping,” in Proceedings of the 28th VLDB Conference, Hong Kong, China, 2002.
- [8] D. Ellis, “Dynamic Time Warp (DTW) in Matlab”, <http://labrosa.ee.columbia.edu/matlab/dtw>.
- [9] J. Laroche, “Time and Pitch scale modification of audio signals” in “Applications of digital signal preprocessing to audio and acoustics”, Mark Kahrs and Karlheinz Brandenburg (eds.), Kluwer Academic Publishers, March 1998.
- [10] J. Flanagan and R. Golden, “Phase Vocoder”, Bell System Technical Journal, pp. 1493-1509, Nov. 1966.





## APPENDIX G

# Unsupervised condition change detection in large diesel engines

---

N. H. Pontoppidan and J. Larsen. Unsupervised condition change detection in large diesel engines. In C. Molina, T. Adali, J. Larsen, M. Van Hulle, S. Douglas, and Jean Rouat, editors, *2003 IEEE Workshop on Neural Networks for Signal Processing*, pages 565–574, Piscataway, New Jersey, September 2003. IEEE Press. URL <http://isp.imm.dtu.dk/nmsp2003>



# UNSUPERVISED CONDITION CHANGE DETECTION IN LARGE DIESEL ENGINES

Niels Henrik Pontoppidan and Jan Larsen

Informatics and Mathematical Modelling, Technical University of Denmark

Richard Petersens Plads, Building 321, DK-2800 Lyngby, Denmark

Phone: +45 45253899, Fax: +45 45872599

E-mail: nhp,jl@imm.dtu.dk, Web: isp.imm.dtu.dk

**Abstract.** This paper presents a new method for unsupervised change detection which combines independent component modeling and probabilistic outlier detection. The method further provides a compact data representation, which is amenable to interpretation, i.e., the detected condition changes can be investigated further. The method is successfully applied to unsupervised condition change detection in large diesel engines from acoustical emission sensor signal and compared to more classical techniques based on principal component analysis and Gaussian mixture models.

## INTRODUCTION

Identification of engine conditions and faults is important for automatic monitoring of critical failures in large marine diesel engines and stationary power plants. The possibility of early detecting small defects prior to evolving into serious breakdowns often reduce the costs for repair significantly. While the long term objective is to classify engine conditions into known fault types, this work focuses merely on the detection of condition changes.

The literature suggests that monitoring based on acoustical emission (AE) offers advantages over sensor techniques such as pressure and vibration [19, 20]. The signal-to-noise ratio is typically better for AE sensor signals, and further a system based on AE is more suitable from an operational point of view. Previous work on adaptive signal processing and machine learning [4, 5, 7, 6, 13, 21, 22] has mainly focused on supervised learning from sensor data and known faults. This paper focuses on unsupervised learning for significant detection of changes in measured AE signals, that is, modeling the probability density of the AE signal. Since AE data are abundant we focus on models, which also offers compact data representation, such as the Independent Component Analysis (ICA), Principal Component Analysis (PCA) and Unsupervised Gaussian Mixture (UGM) models in combination with PCA. The probability density associated with the trained ICA, PCA

and UGM models [9, 18, 16, 12, 15] can be used to identify events which do not conform to the model assumptions [15, 3] and thus represent a significant change in engine condition.

The next section presents the modeling framework and a novel change detection algorithm based on ICA and/or PCA models. The results of a comparative analysis using Bayesian Information Criterion (BIC) and Receiver Operator Characteristics (ROC) is followed by the description of data acquisition, experimental setup, ending with the concluding remarks.

## MODELING FRAMEWORK

Feature vectors from  $N$  examples (revolutions) are assembled into a training data matrix  $\mathbf{X}_{\mathcal{T}}$  of size  $d \times N$   $\mathbf{X}_{\mathcal{T}} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$ . Unsupervised modeling considers modeling the probability density  $p(\mathbf{x}, \boldsymbol{\theta})$ , where  $\boldsymbol{\theta}$  is a parameter vector. The model parameters estimated from available training data  $\mathbf{X}_{\mathcal{T}}$  are denoted by  $\hat{\boldsymbol{\theta}}$ .

### Principal Component Analysis Model (PCA)

Since  $d$  is typically larger than  $N$  we will invoke the PCA model [9, 18], which considers a  $K$  dimensional  $K \ll d$  signal space with rank  $K \leq \min(d, N)$  covariance, and an additive isotropic noise,  $\tilde{\mathbf{x}} = \mathbf{s} + \mathbf{v}$ ,  $p(\tilde{\mathbf{x}}|\boldsymbol{\theta}) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\tilde{\mathbf{x}}})$ , where  $\tilde{\mathbf{x}} = \mathbf{x} - E\{\mathbf{x}\}$  and  $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\tilde{\mathbf{x}}})$  is the zero mean Gaussian distribution with covariance matrix  $\boldsymbol{\Sigma}_{\tilde{\mathbf{x}}} = \boldsymbol{\Sigma}_{\mathbf{s}} + \sigma_{\varepsilon}^2 \mathbf{I}$ ,  $\boldsymbol{\Sigma}_{\mathbf{s}}$  has rank  $K$ .

The model is estimated from data using a singular value decomposition of centered data  $\tilde{\mathbf{X}}_{\mathcal{T}}$ ,  $\tilde{\mathbf{x}}_n = \mathbf{x}_n - \hat{\boldsymbol{\mu}}_{\mathbf{x}}$  and  $\hat{\boldsymbol{\mu}}_{\mathbf{x}} = N^{-1} \sum_{n=1}^N \mathbf{x}_n$ . Assuming  $d \geq N$ , the SVD is given as  $\tilde{\mathbf{X}}_{\mathcal{T}} = \mathbf{U} \mathbf{D} \mathbf{V}^{\top}$ , where  $\mathbf{U}$  is  $d \times N$  and  $\mathbf{V}$   $N \times N$  are left and right eigenvectors and  $\mathbf{D}$  is the  $N \times N$  diagonal matrix of decreasing singular values. Define  $\tilde{\mathbf{U}}$  as the first  $K$  columns of  $\mathbf{U}$ , then for specific choice  $K$

$$\hat{\boldsymbol{\Sigma}}_{\mathbf{s}} = \frac{\tilde{\mathbf{U}} \text{diag}(D_1^2 - \hat{\sigma}_{\varepsilon}^2, \dots, D_K^2 - \hat{\sigma}_{\varepsilon}^2) \tilde{\mathbf{U}}^{\top}}{N}, \quad \hat{\sigma}_{\varepsilon}^2 = \frac{1}{N(d-K)} \sum_{i=K+1}^N D_i^2. \quad (1)$$

In order to estimate the optimal model complexity,  $K_{\text{opt}}$ , we use the Bayesian information criterion (BIC) [17, 10, 18]. BIC is an estimate of model evidence given by

$$p(\tilde{\mathbf{X}}|K) \approx p(\tilde{\mathbf{X}}|\hat{\boldsymbol{\theta}}) \cdot p(\hat{\boldsymbol{\theta}}) \cdot (2\pi/N)^{\dim(\boldsymbol{\theta})/2} \quad (2)$$

Here  $p(\tilde{\mathbf{X}}|\hat{\boldsymbol{\theta}})$  is the likelihood on training data and  $p(\hat{\boldsymbol{\theta}})$  is the prior on parameters. When no explicit prior is available, we use an improper uniform prior. In the case of the PCA model the parameter vector is  $\boldsymbol{\theta} = (\boldsymbol{\mu}, \tilde{\mathbf{U}}, \sigma_{\varepsilon}^2, D_1, \dots, D_K)$ . That is,  $\dim(\boldsymbol{\theta}) = d + K(2d - K + 1)/2 + 1 + K$ .

The PCA model can also be written as

$$p(\mathbf{x}|\boldsymbol{\theta}) = p(\mathbf{y}|\boldsymbol{\theta}) \cdot p(\boldsymbol{\varepsilon}|\boldsymbol{\theta}) \quad (3)$$

where  $\mathbf{y} = \tilde{\mathbf{U}}^\top \mathbf{x}$  is  $K$  dimensional signal space and  $\boldsymbol{\varepsilon} = U_\varepsilon^\top \mathbf{x}$  is the  $d - K$  dimensional noise space with diagonal covariance structure  $\sigma_\varepsilon^2 \mathbf{I}$ . Here  $U_\varepsilon$  are the last  $d - K$  columns of  $\mathbf{U}$ . Under the model,  $\tilde{\mathbf{x}}$  is estimated by  $\tilde{\mathbf{U}}\mathbf{y}$ . That is, the columns of  $\tilde{\mathbf{U}}$  can be interpreted as  $K$  AE signatures which describe the  $\mathcal{H}_0$  condition. The principal components (sources)  $\mathbf{y}$  express the strength of each signature.

### Noise Free Independent Component Analysis Model (ICA-BS)

Assume the noise free ICA model [16]  $\tilde{\mathbf{x}} = \mathbf{A}\mathbf{s}$ , where  $\mathbf{A}$  is a  $d \times K$  mixing matrix and  $\mathbf{s}$  the  $K$  dimensional source vector with statistically independent components. The non-quadratic noise free ICA can be performed in two steps by decomposing  $\mathbf{A} = \tilde{\mathbf{U}}\boldsymbol{\Phi}$ , where  $\tilde{\mathbf{U}}$  is  $d \times K$  projection matrix onto the  $K$  subspace spanned by the sources, and  $\boldsymbol{\Phi}$  is  $K \times K$  mixing matrix. If the source space is  $K$ -dimensional and second order moments of  $\tilde{\mathbf{x}}$  exist, then the projection matrix can be obtained from an SVD projection as described in the previous subsection.

We will use the Infomax algorithm [2] with classical  $\tanh(\cdot)$  nonlinearity<sup>1</sup>. The deployed implementation of the algorithm can be obtained from ICA-ML *DTU:toolbox* [14].

For model selection we will use BIC Eq. (2) with the assumption of independent signal and noise spaces as in Eq. (3), i.e.  $p(\mathbf{x}|\boldsymbol{\theta}) = p(\mathbf{y}|\boldsymbol{\Phi}) \cdot p(\boldsymbol{\varepsilon}|\sigma_\varepsilon^2)$ , where  $p(\mathbf{y}|\boldsymbol{\Phi})$  is the Infomax likelihood [16]

$$p(\mathbf{y}|\boldsymbol{\Phi}) = |\det(\boldsymbol{\Phi})|^{-1} \cdot p_s(\boldsymbol{\Phi}^{-1}\tilde{\mathbf{U}}^\top \mathbf{x}) \quad (4)$$

with  $p_s(\mathbf{s}) = \prod_i 1/\pi \cosh(s_i)$ . The noise likelihood function is [10, Eq. (12)]

$$p(\boldsymbol{\varepsilon}|\hat{\sigma}_\varepsilon^2) = (2\pi\hat{\sigma}_\varepsilon^2)^{-N(d-K)/2} \cdot \exp(-N(d-K)/2) \quad (5)$$

Since  $\boldsymbol{\theta} = (\boldsymbol{\mu}, \tilde{\mathbf{U}}, \sigma_\varepsilon^2, \boldsymbol{\Phi})$ , the total number of parameters are  $\dim(\boldsymbol{\theta}) = d + K(2d - K + 1)/2 + 1 + K^2$ .

### Noisy Independent Component Analysis Model (ICA-MF)

An advanced Bayesian ICA using mean field training [12] enables the training of an ICA model with noise,  $\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{e}$ , under flexible source distributions and possible priors on the mixing matrix. The noise is assumed Gaussian, independent of the sources, and with diagonal covariance matrix. The pre-processing SVD projection step is not exact in the case of noise, i.e., the estimation procedure estimates the  $d \times K$  mixing matrix  $\mathbf{A}$  directly.

<sup>1</sup>Corresponding to identical source priors  $p_{s_i}(s_i) = 1/\pi \cosh(s_i)$ .

As described above, the columns of  $\mathbf{A}$  correspond to AE RMS signatures associated with individual sources, which consequently are non-negative. We therefore invoke a non-negativity prior constraint on the mixing matrix. The activation of these signatures should also be non negative, i.e., source should be non-negative and consequently we use an exponential prior source distribution. The noisy ICA model is estimated using the the ICA-MF *DTU:toolbox* code [14].

The number of sources is also in this case estimated using BIC, Eq. (2).

### Unsupervised Gaussian Mixture Model (UGM)

For comparison we also consider the Gaussian mixture model with SVD signal space preprocessing [11]. Thus as in Eq. (3) we assume  $p(\mathbf{x}|\boldsymbol{\theta}) = p(\mathbf{y}|\boldsymbol{\theta}) \cdot p(\boldsymbol{\varepsilon}|\sigma_{\boldsymbol{\varepsilon}}^2)$ , where  $p(\mathbf{y}|\boldsymbol{\theta})$  is the Gaussian mixture density  $p(\mathbf{y}|\boldsymbol{\theta}) = \sum_{c=1}^C P(c)p(\mathbf{x}|c, \boldsymbol{\theta}_c)$ , with  $p(\mathbf{x}|k, \boldsymbol{\theta}_c) = \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ , and  $\boldsymbol{\theta} = \{P(c), \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\}$  consists of mixing proportions  $P(c)$  as well as means  $\boldsymbol{\mu}_c$  and covariances  $\boldsymbol{\Sigma}_c$  of the Gaussian components. The model is estimated using the generalizable Gaussian mixture algorithm [15] and the subspace dimension  $K$  and number of components  $C$  are selecting using the BIC criterion, Eq. (2).

### Novelty detection

A general treatment of change detection is presented in e.g., [1, 8]. Here we suggest to deploy the novelty detection method proposed in [15, 3], which makes it possible to evaluate whether new examples conform to the trained model  $p(\mathbf{x}, \hat{\boldsymbol{\theta}})$ . A test sample  $\mathbf{x}$  conforming with the trained model will have high log-likelihood whereas a sample from another condition will have low log-likelihood value. In order to perform a formal comparison, we consider the cumulative density of the log-likelihood values.

$$Q(t) = \text{Prob}(\log p(\mathbf{x}|\hat{\boldsymbol{\theta}}) < t) \quad (6)$$

$Q(t)$  can be interpreted as the empirical estimate of the probability that the example  $\mathbf{x}$  (with log likelihood  $t$ ) belongs to the model given by the parameters  $\hat{\boldsymbol{\theta}}$ , i.e. the model that generated the training set.<sup>2</sup>Using a threshold, e.g.,  $t_{min} = 5\%$ , new examples where  $Q(t) < t_{min}$  are rejected under  $\mathcal{H}_0$  at a 5% significance level. See further figure 1.

## EXPERIMENTAL RESULTS

We consider data from three experiments described in the following section, however we only show results from the first experiment, in which the lubri-

<sup>2</sup>For a Gaussian density  $Q(t)$  is  $\chi^2$  distributed. In general, we can only compute this from samples, e.g., by generating an arbitrarily large sample from the generative model  $p(\mathbf{x}|\hat{\boldsymbol{\theta}})$ .

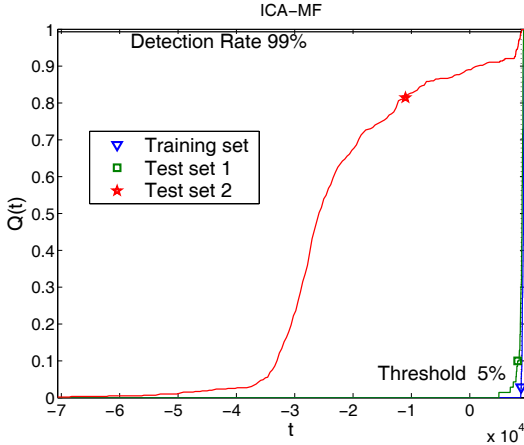


Figure 1: Cumulative log-likelihood density,  $Q(t)$ , from experiment 1 using ICA-MF with 6 components for *training set*, *test set 1* and *test set 2*. *training set* and *test set 1* are very close which means that if we use a 5% threshold on *training set* curve only very few example will be falsely detected. On the other hand, 99% of the examples in *test set 2* will be detected as novel, i.e., as a new condition.

cating oil is shut off. The other experiments gives similar performance results besides from changes in the optimal number of components.

The  $Q(t)$  function of the trained models are computed from the training set. Choosing a specific threshold  $t_{\min}$  then the false alarm rate can be estimated as the fraction of examples in *test set 1* (belonging to  $\mathcal{H}_0$ ) for which log-likelihood  $\log p(\mathbf{x}|\hat{\theta}) < t_{\min}$ . Similarly the true detection probability is estimated as the fraction of examples on *test set 2* (belonging to  $\mathcal{H}_1$ ) for which log-likelihood smaller than  $t_{\min}$ . By varying  $t_{\min}$  the so-called receiver operation characteristics (ROC) curves are formed, which is shown in figure 3. Larger area under the ROC curve implies higher true detection for a given false alarm. Clearly ICA-MF shows best true performance. In order to interpret the nature of the changed condition we can evaluate the difference between a test feature vector  $\mathbf{x}$  and its estimate under the model. For ICA-MF we first estimate the source  $\hat{\mathbf{s}}$  and then compute the estimate under the model  $\hat{\mathbf{x}} = \hat{\mathbf{A}}\hat{\mathbf{s}}$ . The interpretation is shown in figure 5.

## EXPERIMENTAL SETUP

The data set consists of acoustical emission (AE) root-mean squared (RMS) signals acquired with four AE sensors. In this work we will use a single sensor placed on the liner (cylinder casing). Data was recorded for 10 seconds followed by a pause of 60 seconds as a simple compression scheme. Data was originally sampled at 2.5 MHz using the RMS time constant  $50 \mu\text{s}$ . The signal is resampled into crank angle domain using a crank encoder. This par-

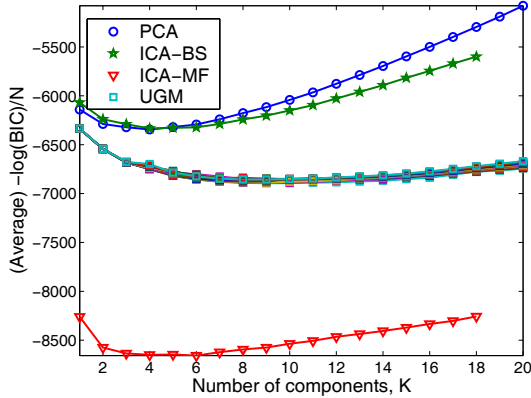


Figure 2: Model selection using the BIC criterion Eq. (2). The curves display  $-\log p(\tilde{\mathbf{X}}|K)/N$ . ICA-MF averaged over 4 runs and UGM over 20 runs. Further UGM curves for 2-16 clusters are plotted. PCA and ICA-BS achieved minimum BIC value for  $K = 4$ , ICA-MF for  $K = 6$ , and UGM for the combination  $K = 9$  and  $C = 2$ . The significantly lower BIC of the ICA-MF model indicates this model is preferable, possibly due to a more advanced noise model.

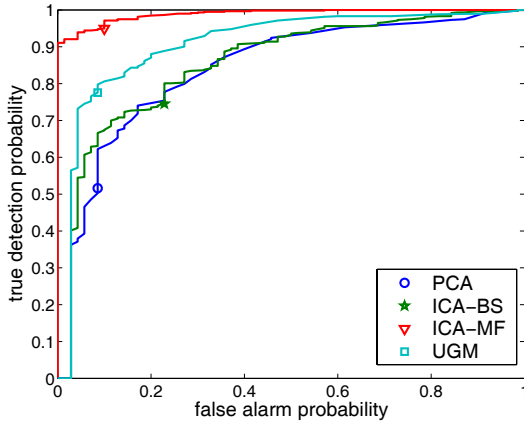


Figure 3: ROC curves from experiment 1 (shutting off lubrication) shows probability of false alarm versus true detection. Clearly the noisy ICA model (ICA-MF) provides best performance.

tially compensates for variations in rotation speed and establish the relation between AE signal expression and mechanical events during the combustion cycle. Define the  $d = 2048$  dimensional feature vector

$$\mathbf{x}_n = [x_n(1), \dots, x_n(i), \dots, x_n(2048)]^\top, \quad (7)$$

where  $x_n(i)$  is the RMS AE signal for cycle  $n$  at angle  $(i - 1) \cdot 360/2048^\circ$ .  $0^\circ$  corresponds to the top position as indicated by top pulse signal. 21 other



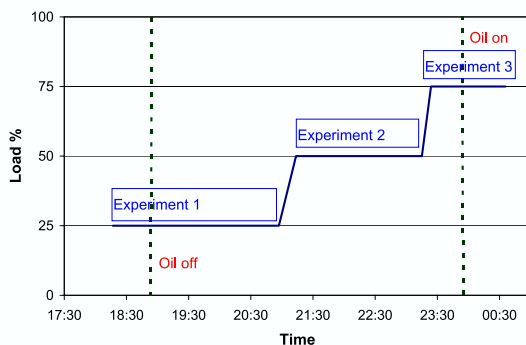


Figure 4: Time line of experiment. The curve shows the increasing load as function of time. The numbered boxes refer to the three experiments described in the experimental setup section. The two vertical lines indicate when lubricating was turned off and on.

channels (including top and crank-pulses) were acquired from the cylinder at MAN B&W Diesel’s Research Engine<sup>3</sup> in Copenhagen. These additional signals can be used to interpret the results of AE signal analysis.

During the experiment, the engine load was changed from 25% to 75%. In the middle of the 25% load period the cylinder lubrication was turned off, and in the middle of the 75% load period lubrication was re-established. Figure 4 shows the actual timing of these events.

From the entire data set we have selected periods where the engine displays non-trivial abrupt condition changes. Thus we are not interested in detecting that the load changes but e.g., that lubrication is turned on or off.

Knowledge about condition changes is obtained from manual annotations by MAN B&W and from additional 21 sensor channels. This information is not directly passed to the algorithms, but is used in order to design relevant data periods and for performance evaluation.

We consider three experiments indicated in Figure 4.

**Experiment 1: Shutting Off Lubrication** After turning on the engine, the load stabilized at 25% on the propeller curve. After a while the lubrication to the cylinder is turned off. The objective is to detect this operation condition change.

**Experiment 2: Unstable Revolution Speed** The engine is running at 50% load with the lubrication system turned off. Inspection of the revolution speed obtained from timing signal indicates that the engine condition undergoes sudden changes in the middle of this period, which is probably caused by engine load fluctuations. We aim to detect the start and end of this period.

**Experiment 3: Re-establishment of lubrication** The engine is running at 75% load without lubrication. After 30 minutes lubrication is re-

<sup>3</sup>Test bed, 4 cylinders, 500 mm bore, 10.000 BHP.

established, possibly lowering the wear rate. We aim to detect this change.

In order to validate the performance of the detection we consider a null-hypothesis  $\mathcal{H}_0$ , the current normal condition, and a new condition,  $\mathcal{H}_1$ . The data from each experiment are divided into:

**Training set** contains examples from the current engine operation condition  $\mathcal{H}_0$ .

**Test set 1** contains examples from current condition,  $\mathcal{H}_0$ , and is used for model validation.

**Test set 2** contains examples that we based on annotations believe come from the the new condition,  $\mathcal{H}_1$ .

## CONCLUSION

This paper presented a novel probabilistic change detection framework based on independent component analysis (ICA) modeling. The method was successfully applied to unsupervised condition change detection in large diesel engines using acoustical emission sensors. The overdetermined noisy ICA model using mean-field Bayesian learning showed best performance.

**Acknowledgment.** The work is supported by EU Competitive and Sustainable Growth Programme GRD2-2001-50014 through the AE-WATT project <http://isp.imm.dtu.dk/awatt>. Torben Fog Man B&W is acknowledged for fruitful discussions and data support. Further we would like to thank Kaare B. Petersen and Lars Kai Hansen for critical comments.

## REFERENCES

- [1] M. Basseville and I. V. Nikiforov, **Detection of Abrupt Changes: Theory and Application**, Prentice-Hall, 1993.
- [2] A. Bell and T. J. Sejnowski, "An Information-Maximization Approach to Blind Separation and Blind Deconvolution," **Neural Computation**, vol. 7, pp. 1129–1159, 1995.
- [3] C. M. Bishop, "Novelty Detection and Neural Network Validation," **IEEE Proceedings - Vision Image and Signal Processing**, vol. 141, no. 4, pp. 217–222, 1994.
- [4] G. Chandroth, A. J. C. Sharkey and N. E. Sharkey, "Cylinder Pressures and Vibration in Internal Combustion Engine Condition Monitoring," in **Comadem 99**, July 1999.
- [5] G. Chandroth, A. J. C. Sharkey and N. E. Sharkey, "Vibration signatures, wavelets and principal components analysis in diesel engine diagnostics," in **Marine Technology Odra 99**, October 1999.
- [6] T. Fog, **Condition Monitoring And Fault Diagnosis in Marine Diesel Engines**, Ph.D. thesis, Technical University of Denmark, 1998, IMM-PHD-1998-52.

- [7] T. L. Fog, L. K. Hansen, J. Larsen, H. S. Hansen, L. B. Madsen, P. Srensen, E. R. Hansen and P. S. Pedersen, "On Condition Monitoring of Exhaust Valves in Marine Diesel Engines," in Y. H. Hu, J. Larsen, E. Wilson and S. Douglas (eds.), **Proceedings of the IEEE Workshop on Neural Networks for Signal Processing IX**, IEEE, Piscataway, New Jersey, 1999, pp. 225–234.
- [8] F. Gustafsson, **Adaptive Filtering and Change Detection**, John Wiley and Sons LTD, 2000, ISBN: 0 471 49287 6.
- [9] L. Hansen and J. Larsen, "Unsupervised Learning and Generalization," in **Proceedings of the IEEE Int. Conf. on Neural Networks 1996**, 1996, vol. 1, pp. 25–30.
- [10] L. K. Hansen, J. Larsen and T. Kolenda, "Blind Detection of Independent Dynamic Components," In **proc. IEEE ICASSP'2001**, vol. 5, pp. 3197–3200, 2001.
- [11] L. K. Hansen, S. Sigurdsson, T. Kolenda, F. . Nielsen, U. Kjems and J. Larsen, "Modeling Text with Generalizable Gaussian Mixtures," in **IEEE ICASSP'2000, Istanbul, Turkey**, June 2000, vol. VI, pp. 3494–3497.
- [12] P. Højén-Sørensen, O. Winther and L. K. Hansen, "Mean Field Approaches to Independent Component Analysis," **Neural Computation**, vol. 14, pp. 889–918, 2002.
- [13] M. Kaveh, A. H. Tewfik, K. M. Buckley et al., **Integrated Predictive Diagnostic Tools**, MultiUniversity Center for Integrated Diagnostics, chap. 4.b: Signal processing for the detection and classification of acoustic emissions in integrated diagnostics, 2000.
- [14] T. Kolenda, S. Sigurdsson, O. Winther, L. K. Hansen and J. Larsen, "DTU:Toolbox," Internet, 2002, <http://isp.imm.dtu.dk/toolbox/>.
- [15] J. Larsen, L. K. Hansen, A. Szymkowiak, T. Christiansen and T. Kolenda, "Webmining: Learning from the World Wide Web," **Computational Statistics and Data Analysis**, vol. 38, pp. 517–532, 2002.
- [16] T.-W. Lee, **Independent Component Analysis: Theory and Applications**, Kluwer Academic Publishers, September 1998, ISBN: 0 7923 8261 7.
- [17] D. J. C. MacKay, "Bayesian Model Comparison and Backprop Nets," in **Advanced of Neural Information Processing Systems 4**, 1992, pp. 839–846.
- [18] T. P. Minka, "Automatic choice of dimensionality for PCA," in T. Leen, T. Dietterich and V. Tresp (eds.), **Advances in Neural Information Processing Systems 13**, MIT Press, 2001, pp. 598–604.
- [19] G. D. Neill, S. Benzie, J. D. Gill, P. M. Sandford, E. R. Brown, J. A. Steel and R. L. Reuben, "The relative merits of acoustic emission and acceleration monitoring for detection of bearing faults," in **COMADEM**, December 1998, pp. 651–661.
- [20] R. L. Reuben, "The Role of Acoustic Emission in Industrial Condition Monitoring," **International Journal of COMADEM**, vol. 1, no. 4, pp. 35–46, Oct. 1998, ISSN 1363-7681.
- [21] L. M. Rogers, "Use of acoustic emission methods for crack growth detection in offshore and other structures," **Trans IMarE**, vol. 110, no. part 3, pp. 171–180, 1998.
- [22] A. Ypma, **Learning methods for machine vibration analysis and health monitoring**, Ph.D. thesis, Dept. of Applied Physics, Delft University of Technology, Nov. 2001.

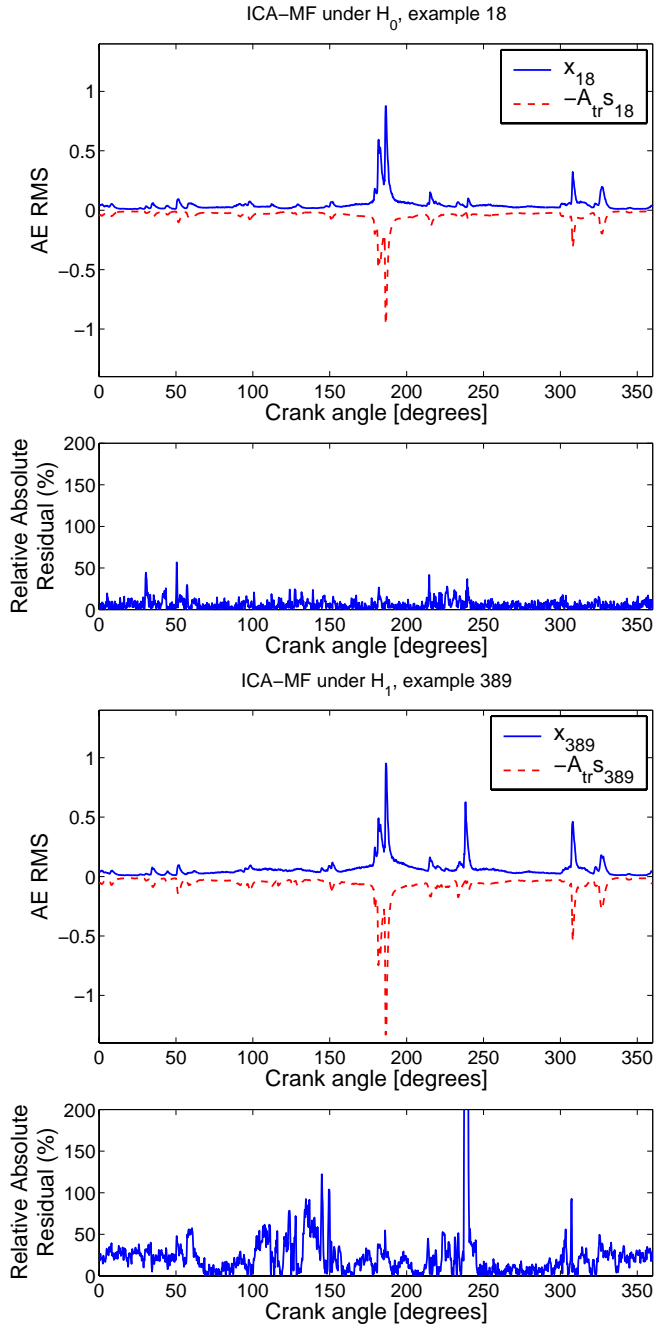


Figure 5: Interpretation of examples which ICA-MF detected as belonging to  $\mathcal{H}_0$  or  $\mathcal{H}_1$ . The upper panel shows  $\mathbf{x}$  and the (negative) estimate  $\hat{\mathbf{x}} = \mathbf{A}\mathbf{s}$ , and the lower panel the relative error  $100\% \cdot |(\mathbf{x} - \hat{\mathbf{x}})/\hat{\mathbf{x}}|$ . Under  $\mathcal{H}_0$  the relative error typically is around 10% while the example under  $\mathcal{H}_1$  possesses very high error around 1500% for crank angle position close to  $240^\circ$ . Knowledge about the engine combustion cycle at crank angle position  $240^\circ$ , can then be used to identify the nature and impact of detected condition change.

APPENDIX H

# Independent component analysis for detection of condition changes in large diesels

---

N. H. Pontoppidan, J. Larsen, and T. Fog. Independent component analysis for detection of condition changes in large diesels. In [Shrivastav et al. \[2003\]](#). ISBN 91-7636-376-7. URL <http://www.imm.dtu.dk/pubdb/p.php?2400>



# INDEPENDENT COMPONENT ANALYSIS FOR DETECTION OF CONDITION CHANGES IN LARGE DIESEL ENGINES

Niels Henrik Pontoppidan<sup>1</sup>, Jan Larsen<sup>1</sup> and Torben Fog<sup>2</sup>

<sup>1</sup> Informatics and Mathematical Modelling, Technical University of Denmark,  
Richard Petersens Plads 321, 2800 Kgs. Lyngby, Denmark  
{nhp,jl}@imm.dtu.dk, Tel +45 4525 3351, Fax +45 4587 2599

<sup>2</sup> MAN B&W Diesel A/S  
Teglhølmegade 41, 2450 København S, Denmark  
Torben.Fog@manbw.dk, Tel +45 3385 1759, Fax +45 2285 1030

## ABSTRACT

Automatic detection and classification of operation conditions in large diesel engines is of significant importance. This paper investigates an independent component analysis (ICA) framework for unsupervised detection of changes in and possibly classification of operation conditions such as lubrication changes and increased wear based on acoustical emission (AE) sensor signals. The probabilistic formulation of ICA enables a statistical detection of novel events which do not conform to the current ICA model, thus indicating significant changes in operation conditions. Novelty of an observation is measured through the likelihood that the model has produced that observation. Evaluation of likelihood ratios allows the framework to also handle multiple models, thus enabling classification of operation conditions; furthermore the likelihood also serves as a link to traditional change detection. The framework is evaluated on measured AE signals in an experiment where the operational condition varies. In particular, we compare the performance of mean field ICA, information-maximization ICA, and Principal Component Analysis. For detection of changes the performance is also compared to standard methods, e.g. mean value step detection.

## INTRODUCTION

Identification of engine conditions and faults is important for automatic monitoring of critical failures in large marine diesel engines and stationary power plants. Early detection of small defects prior to evolving into serious breakdowns often reduces the costs for repair significantly. The literature suggests that monitoring based on acoustical emission (AE) offer advantages over sensor techniques such as pressure and vibration [1, 2]. The signal-to-noise ratio is typically better for AE sensor signals, and further a system based on AE is more feasible from an operational point of view. Previous work on adaptive signal processing and machine learning [3, 4, 5, 6, 7, 8, 9] has mainly been focusing on supervised learning from sensor data and known faults. This paper focuses on unsupervised learning for significant detection of changes in measured AE signals, that is, modelling the probability density of the AE signal. Since we measure many samples of the AE signal we suggest a model, which also offers a compact data representation, such as the Independent Component Analysis (ICA) and Principal

Component Analysis (PCA) models. The probability density associated with the trained ICA and PCA models [10, 11, 12, 13] can be used to identify events which do not conform to the model assumptions [14, 15] and thus represent a significant change in engine condition.

In section 2 the data acquisition and experimental setup is described. Section 3 presents the modelling framework and a novel change detection algorithm based on ICA or PCA models. A comparative analysis and discussion using the suggested method is provided in section 4, and finally, section 5 state the conclusions.

Throughout vectors and matrices are identified by lowercase bold and uppercase bold letters respectively, i.e. the vector  $\mathbf{x}$  and matrix  $\mathbf{X}$ .

## EXPERIMENTAL SETUP

The data set consist of two acoustic emission (AE) energy (RMS) signals  $y_1(t)$ ,  $y_2(t)$  acquired at 20 kHz with two very sensitive Physical Acoustics Corporation sensors placed on the cylinder liner and cover, respectively. The signals are resampled into the crank angle domain to provide 2048 samples per engine revolution. Further the two signals are stacked into the  $d=4096$  dimensional feature (row) vector  $\mathbf{x}$ .

$$\mathbf{x} = [y_1(1), \dots, y_1(2048), y_2(1), \dots, y_2(2048)]^T \quad (1)$$

In addition, 21 other channels, including top and crank-pulses were acquired from the cylinder at MAN B&W Diesel's Research Engine<sup>1</sup> in Copenhagen.

For each experiment, we consider three data sets:

- A Training set containing stationary examples under the current engine operation condition  $H_0$ .
- Test set 1 containing examples under the same condition as in the training set,  $H_0$ , which is used for model validation.
- Test set 2 containing examples that are investigated for changed in engine condition,  $H_1$ .

Thus we are able to check against false rejection of  $H_0$  and to some extent also false accept of  $H_1$ . During the experiment, the engine load was changed from 25% to 75%. In the middle of the 25% load period the cylinder lubrication was turned off, and in the middle of the 75% load period this system was turned on again.

### *Experiment 1: Shutting Off Lubrication*

Initially the engine is stabilized at 25% load. After a while the lubrication to the cylinder is turned off. The objective is to detect this change in operation condition shortly after it occurred.

### *Experiment 2: Unstable Revolution Speed*

The engine is running at 50% load and the lubrication system is turned off. Inspection of the revolution speed obtained from timing signal indicated that the engine undergo some sudden changes in the middle of this period. This is possibly caused by engine load fluctuations. We aim to detect the start and end of this period.

### *Experiment 3: Increased wear and re-establishment of lubrication*

---

<sup>1</sup>Test bed, 4 cylinders, 500 mm. bore, 10.000 BHP.



The engine is running at 75 % load without lubrication. After 30 minutes lubrication is re-established, lowering the wear rate. We aim to detect this change of AE activity.

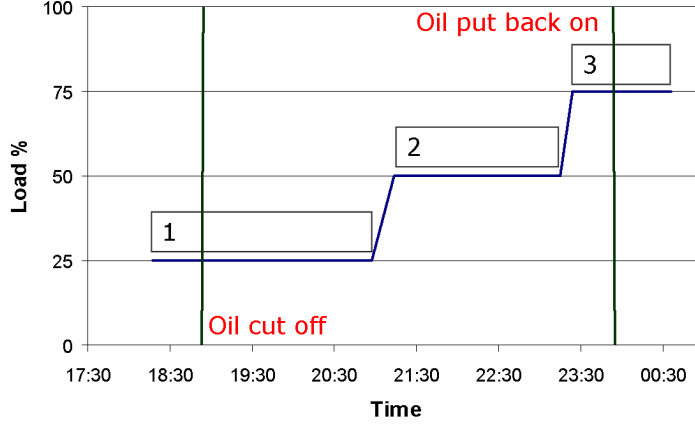


Figure 1, Time line of experiment. The stair like curve shows the increasing load as function of time. The numbered boxes refer to the three experiments described in the previous sections. The two vertical lines indicate when the lubricating system was turned off and on.

## MODELING FRAMEWORK

### Novelty detection

A general treatment of change detection is presented in e.g., [16, 17] here we deploy the novelty detection method proposed in [14, 15] which makes it possible to evaluate whether new examples conform to the model trained on the training set  $\mathcal{T}$ . The novelty detection is based on input density  $p(\mathbf{x} | \mathcal{T})$  of the trained model. Consider the cumulative distribution of density values over the training set for all thresholds  $t$ . By selecting a low threshold  $Q_{\min}$  identifying the corresponding  $t_{\min} = \arg \min_t Q(t) \geq Q_{\min}$ , novel events are detected as those where  $Q(t)$  is less than  $t_{\min}$ , see further figure 2.

$$Q(t) = P(\mathbf{x} \in R), R = \{\mathbf{x} : p(\mathbf{x} | \mathcal{T}) < t\} \quad (2)$$

So  $Q(t)$  is the probability that the example  $\mathbf{x}$  is under the same condition  $H_0$  as examples in the training set. The presented method assumes that examples,  $\mathbf{x}$ , in the training set, i.e. drawn from the normal condition model  $H_0$ , share underlying hidden sources, and that we are able to identify those (or linear combinations) correctly. As usual, we are faced with the problem of over fitting, where too many sources allow the model to adapt to the noise in the training examples, and too few sources prohibits the model in learning the different variations. With test set 1 we are able to detect over fitting, as it contains examples that should be accepted as  $H_0$ , like the training set.

For an example  $\mathbf{x}$  in the training set with mixing matrix  $\mathbf{A}$  and corresponding source vector  $\mathbf{s}$  the log-likelihood<sup>2</sup> is given by

$$\log p(\mathbf{x} | \mathcal{T}) \log p(\mathbf{x} | \mathbf{A}, \Sigma_e, \Sigma_s) = \log \int p(\mathbf{x} | \mathbf{A}, \Sigma_e, \Sigma_s, \mathbf{s}) p(\mathbf{s}) d\mathbf{s} \quad (3)$$

<sup>2</sup> The probability density of  $\mathbf{x}$  given the estimated model parameters.

Where  $\Sigma$  is the covariance of the residuals from the training set, and  $\Sigma_s$  is the covariance of the sources estimated from the training set.

Define  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  as the set of  $N$  examples<sup>3</sup> and the number of used sources/components,  $K$ .

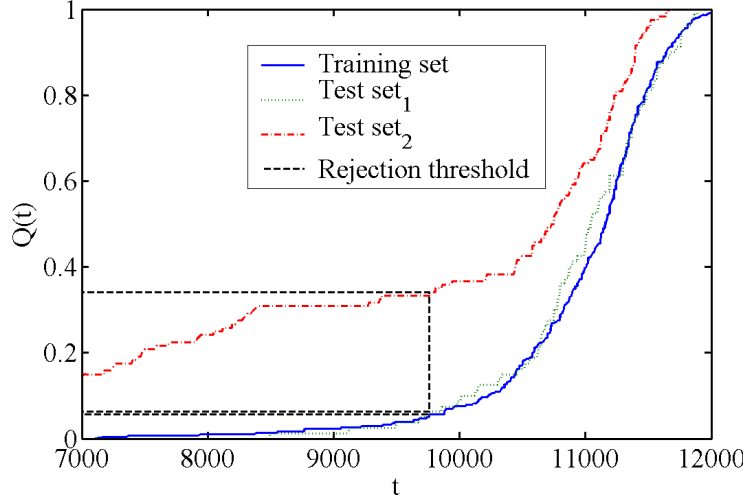


Figure 2, Cumulated log-likelihood densities,  $Q(t)$ , from experiment 2 using ICA with 2 components. The solid (and smooth) line shows the cumulated density for the training examples. The dotted line show the cumulated density for test set 1 and is close to the training set. The dash-dotted line show the cumulated density for test set 2 and is above the training set curve, showing that many of these examples are rejected. The vertical (dashed) line show the threshold  $t_{\min}$  together with the corresponding (horizontal) lines at the different rejection levels.

### 3.2. PCA

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V} \quad (4)$$

Where  $\mathbf{X}$  is  $d \times N$ ,  $\mathbf{U}$  is  $d \times d$ ,  $\mathbf{D}$  is  $d \times N$ , and  $\mathbf{V}$  is  $N \times N$ . We identify the mixing matrix as the  $K$  first columns of  $\mathbf{U}$ ,  $\mathbf{A} = \mathbf{U}_K$ , and the source matrix  $\mathbf{S}_K$  as the first  $K$  principal components  $\mathbf{D}_K \mathbf{V}_K$ . Given a new example  $\mathbf{x}$  we get the corresponding source  $\mathbf{s} = \mathbf{A}\mathbf{x}$ , and the residual  $\boldsymbol{\varepsilon} = \mathbf{x} - \mathbf{A}\mathbf{s}$ . We assume that the residual is Gaussian with diagonal covariance, and that the source distribution  $p(\mathbf{s})$  can be approximated by a Gaussian with zero mean and known diagonal covariance given by  $\mathbf{D}_K$ . Under these assumptions Eqn. (3) is analytically tractable [10, 11] and is given by

$$\log p(\mathbf{x} | \mathbf{A}, \boldsymbol{\Sigma}_\varepsilon, \boldsymbol{\Sigma}_s) = \frac{1}{2} (\log |\boldsymbol{\Sigma}| - \log |\boldsymbol{\Sigma}_\varepsilon| - \log |\boldsymbol{\Sigma}_s|) - \frac{1}{2} \mathbf{x}^T (\boldsymbol{\Sigma}_\varepsilon^{-1} + \boldsymbol{\Sigma}_\varepsilon^{-1} \mathbf{A} \boldsymbol{\Sigma}_s \mathbf{A}^T) \mathbf{x} - \frac{1}{2} \log 2\pi \quad (5)$$

Where

$$\boldsymbol{\Sigma} = (\mathbf{A}^T \boldsymbol{\Sigma}_\varepsilon^{-1} \mathbf{A} + \boldsymbol{\Sigma}_s^{-1})^{-1} \quad (6)$$

### Information-maximization ICA (IM ICA)

<sup>3</sup> Each example corresponds to one revolution of the engine, for which two AE signature waveforms of 2048 samples are acquired.

Using PCA as a pre-processing dimensionality reduction step onto  $K$  dimensions, we can the apply Infomax ICA with square mixing matrix [18, 12].

$$\mathbf{X} = \mathbf{U}_K \mathbf{A} \mathbf{S} \quad (7)$$

Where  $\mathbf{X}$  is  $d \times N$ ,  $\mathbf{U}$  is  $d \times K$ ,  $\mathbf{A}$  is  $K \times K$  and  $\mathbf{S}$  is  $K \times N$ . We use the ICA-ML DTU:toolbox [19] for training

### *Mean Field ICA with Positive Constraints on Source and Mixing Matrices (MF ICA)*

An advanced Bayesian ICA using mean field training [13] enables the possibility of avoiding PCA as a pre-processing step as well as imposing priors on the sources and mixing matrix.

The AE signals is the observable result of an additive process combining energy from various sources in the cylinder. If we want to model this, both source- and mixer matrix must only contain non-negative elements. We estimate a positive source matrix  $\mathbf{S}$ , where the elements of each column are exponential distributed, and a mixer matrix  $\mathbf{A}$  having non-negative elements using the ICA-ADATAP DTU:toolbox [19].

$$\mathbf{X} = \mathbf{A} \mathbf{S} \quad (8)$$

$\mathbf{X}$  is  $d \times N$ ,  $\mathbf{A}$  is  $d \times K$  and  $\mathbf{S}$  is  $K \times N$ . Given a new example and the trained model the code provides estimates of sources and the associated log-likelihood. The returned log-likelihood (as well as the sources) is a mean field approximation to **Eqn. (3)** obtained by minimizing a Kullback-Leibler divergence [13].

## RESULTS

We have selected the number of components that reject the expected number of examples from test set 2 while still accepting examples from test set 1. The following tables show these results and the obtained performance. Figures 3-6 show  $Q(t)$  for individual examples in different experiments using the algorithms. Looking at these figures, the condition changes are easily spotted. Generally mean field ICA and PCA works best, which is due to the fact that their log-likelihood also depends on the noise. These algorithms are both able to detect that the sources are changing and/or the yielding those examples is evaluated using both the sources and the residuals.

	PCA	IM ICA	MF ICA
Test set 1	10 %	13 %	10 %
Test set 2	93 %	80 %	89 %
No. of components	3	27	3

Table 1, Experiment 1: detecting oil off. The expected rejection rate of test set 2 is 93%.

	PCA	IM ICA	MF ICA
Test set 1	6 %	4 %	5 %
Test set 2	35 %	31 %	33 %
No. of components	5	12	2

Table 2, Experiment 2: Detecting temporary external condition change. The expected rejection rate of test set 2 is 34 %.

	PCA	IM ICA	MF ICA
Test set 1	13 %	5 %	10 %
Test set 2	98 %	94 %	98 %
No. of components	97	12	1

Table 3, Experiment 3: Detecting oil on. The expected rejection rate of test set 2 is 100%

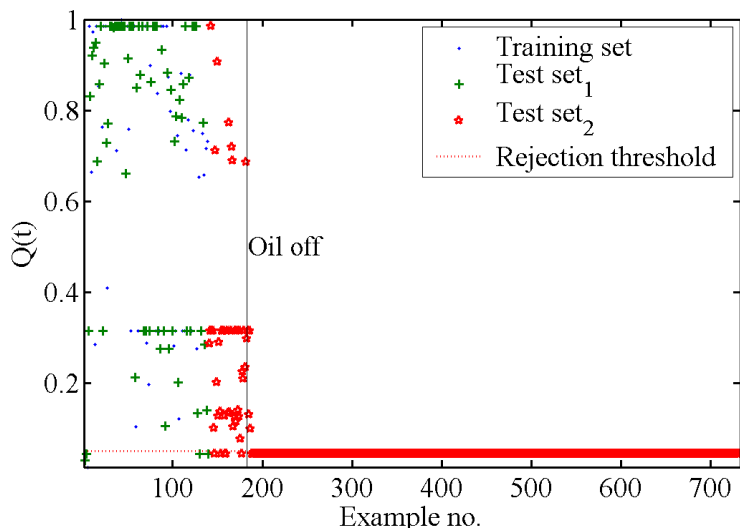


Figure 3,  $Q(t)$  for each example using PCA in experiment 1. The probability of coming from the normal condition clearly drops after the oil was cut off.

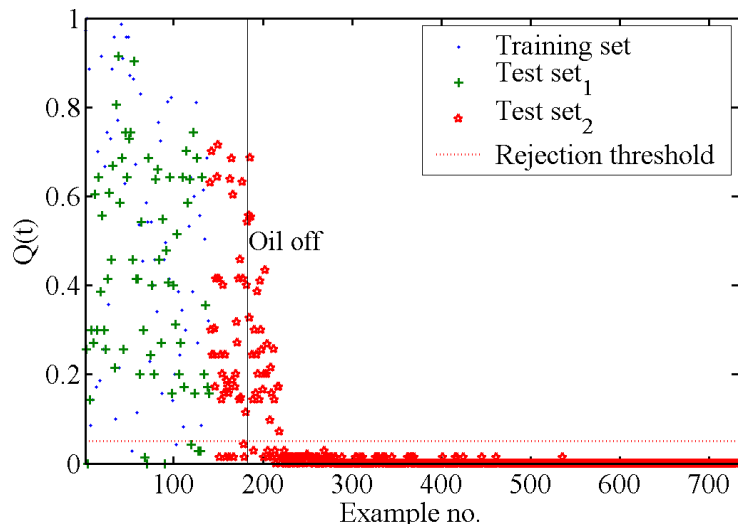


Figure 4,  $Q(t)$  for each example using mean field ICA in experiment 1. The probability of coming from the normal condition clearly drops after the oil was cut off.

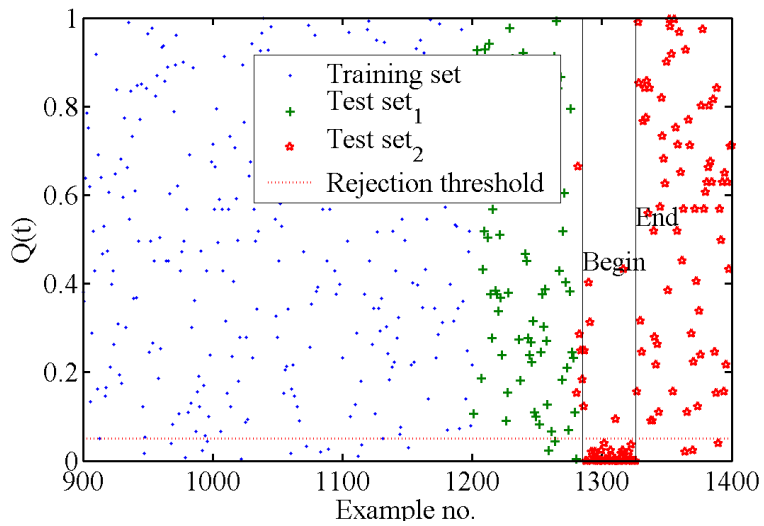


Figure 5,  $Q(t)$  for each example using Infomax ICA in experiment 2. The probability of coming from the normal condition clearly drops and returns thus indicating that the engine return to the previous condition.

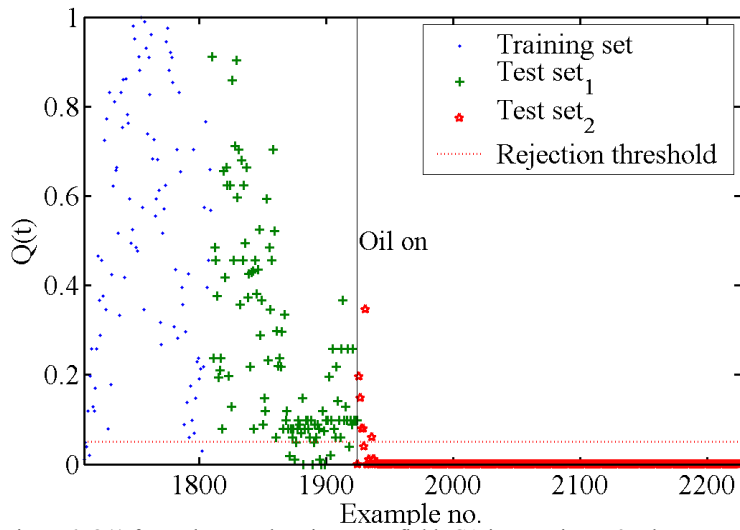


Figure 6,  $Q(t)$  for each example using mean field ICA in experiment 3. The probability of coming from the normal condition clearly drops after the oil was put back.

### *Simpler schemes*

With ICA and PCA we are able to detect the changes. Experiments with the CUSUM algorithm [16] and Bayesian step detection [20] using means of the revolutions<sup>4</sup>, show that simpler schemes also detect some of these condition changes. In settings where the amount of data is too large, these simpler schemes can be used to pre-select time-windows for further analysis.

### **CONCLUSION**

We have demonstrated the ability to detect changes in the operating parameters, including some parameters that were not monitored, for instance an external parameter. Furthermore we detect transitory condition changes, where the engine quickly returns to the previous condition. In future we'll exploit this fact and extend the method aiming for classification, based on likelihood ratios. This should fix the apparent problems with only detecting changes namely, indication of causes as well as verification of fixes – when the condition returns after repair. Given vast amounts of data, segmented by simpler schemes our extended method should also be able to classify and group the segments.

### *Acknowledgements*

This work is supported by EU Competitive and Sustainable Growth Programme GRD2-2001-50014 – acronym AE-WATT (<http://isp.imm.dtu.dk/aewatt>).

### **REFERENCES**

- 1 R. L. Reuben, "The role of acoustic emission in industrial condition monitoring," International Journal of COMA-DEM, vol. 1, no. 4, pp. 35–46, Oct. 1998, ISSN 1363-7681.
- 2 G. D. Neill, S. Benzie, J. D. Gill, P. M. Sandford, E. R. Brown, J. A. Steel, and R. L. Reuben, The relative merits of acoustic emission and acceleration monitoring for detection of bearing faults, COMADEM, 1998.
- 3 G. Chandroth, A. J. C. Sharkey, and N. E. Sharkey, "Cylinder pressures and vibration in internal combustion engine condition monitoring," in Comadem 99, July 1999.
- 4 G. Chandroth, A. J. C. Sharkey, and N. E. Sharkey, "Vibration signatures, wavelets and principal components analysis in diesel engine diagnostics," in Marine Technology ODRA 99, October 1999.
- 5 T. L. Fog, L. K. Hansen, J. Larsen, H. S. Hansen, L. B. Madsen, P. Sørensen, E. R. Hansen, and P. S. Pedersen, "On condition monitoring of exhaust valves in marine diesel engines," in Proceedings of the IEEE Workshop on Neural Networks for Signal Processing IX, Y. H. Hu, J. Larsen, E. Wilson, and S. Douglas, Eds., Piscataway, New Jersey, 1999, IEEE, pp. 225–234.
- 6 Torben Fog, Condition Monitoring And Fault Diagnosis in Marine Diesel Engines, Ph.D. thesis, Technical University of Denmark, 1998, IMM-PHD-1998-52.
- 7 Mos Kaveh, Ahmed H. Tewfik, Kevin M. Buckley, et al., Integrated Predictive Diagnostic Tools, chapter 4.b: Signal processing for the detection and classification of acoustic emissions in integrated diagnostics, MultiUniversity Center for Integrated Diagnostics, 2000.
- 8 L.M. Rogers, "Use of acoustic emission methods for crack growth detection in offshore and other structures," Trans IMarE, vol. 110, no. part 3, pp. 171–180, 1998.
- 9 Alexander Ypma, Learning methods for machine vibration analysis and health monitoring, Ph.D. thesis, Dept. of Applied Physics, Delft University of Technology, Nov. 2001.

---

<sup>4</sup>  $\frac{1}{2048} \sum_{n=1}^{2048} y_1(n)$

- 10 L.K. Hansen and J. Larsen, "Unsupervised learning and generalization," in Proceedings of the IEEE Int. Conf. on Neural Networks 1996, 1996, vol. 1, pp. 25–30.
- 11 Thomas P. Minka, "Automatic choice of dimensionality for pca," in Advances in Neural Information Processing Systems 13 , T.K. Leen, T.G. Dietterich, and V. Tresp, Eds. 2001, pp. 598–604, MIT Press.
- 12 Te-Won Lee, Independent Component Analysis: Theory and Applications, Kluwer Academic Publishers, September 1998, ISBN: 0 7923 8261 7.
- 13 Pedro Højen-Sørensen, Ole Winther, and Lars Kai Hansen, "Mean field approaches to independent component analysis," Neural Computation, vol. 14, pp. 889–918, 2002.
- 14 Jan Larsen, Lars Kai Hansen, Anna Szymkowiak, Torben Christiansen, and Thomas Kolenda, "Webmining: Learning from the World Wide Web," Computational Statistics and Data Analysis, vol. 38, pp. 517–532, 2002.
- 15 C.M. Bishop, "Novelty detection and neural network validation," IEE Proceedings - Vision Image and Signal Processing, vol. 141, no. 4, pp. 217–222, 1994.
- 16 M. Basseville and I.V. Nikiforov, Detection of Abrupt Changes: Theory and Application, Prentice-Hall, 1993.
- 17 Fredrik Gustafsson, Adaptive Filtering and Change Detection, John Wiley and Sons LTD, 2000, ISBN: 0 471 49287 6.
- 18 Bell and T.J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," Neural Computation, vol. 7, pp. 1129–1159, 1995.
- 19 T. Kolenda, S. Sigurdsson, O. Winther, L.K. Hansen, and J. Larsen, "DTU:toolbox," Internet, 2002, <http://isp.imm.dtu.dk/toolbox/>.
- 20 J. J. K. O. Ruanaidh and W.J. Fitzgerald, "Numerical Bayesian Methods Applied to Signal Processing", Springer Verlag, 1996.





## APPENDIX I

# Calculations with Mean field independent component analysis

---

This appendix just serves to capture the calculations that lead to the analytic, nevertheless useless result with Gamma distribution as source prior. For a single component  $s$  that follows a gamma distribution with parameters  $k$ , and  $\beta$ , and the mean field approximation is a normal distribution with parameters  $\lambda$  and  $\gamma$ :

$$P_q(s) = \begin{cases} \frac{S^{k-1} \exp -\frac{s}{\beta}}{\Gamma k \beta^k} & , s > 0 \\ 0 & , s \leq 0 \end{cases} \quad (\text{I.1})$$

$$\log \int P(s)P_q(s)ds = \log \frac{2^{\frac{k}{2}-2}\beta^k \lambda^{-\frac{k}{2}}}{\Gamma(k)\sqrt{\pi}} \left( \right. \quad (I.2)$$

$$\frac{(\kappa + 1)}{\sin(\frac{1}{2}\pi k)\Gamma(\frac{3-k}{2})} \text{LagL}\left(-\frac{k}{2}, \frac{1}{2}, \kappa\right)$$

$$- \frac{(1 - \beta\gamma)^2}{\beta^2 \lambda \sin(\frac{1}{2}\pi k)\Gamma(\frac{3-k}{2})} \text{LagL}\left(-\frac{k}{2}, \frac{3}{2}, \kappa\right)$$

$$- \frac{(1 - \beta\gamma)(1 - k)(\kappa + k - 1)}{\sqrt{2}\beta\sqrt{\lambda} \cos(\frac{1}{2}\pi k)\Gamma(\frac{4-k}{2})} \text{LagL}\left(\frac{1-k}{2}, \frac{1}{2}, \kappa\right)$$

$$+ \frac{(1 - \beta\gamma)^3(1 - k)}{\sqrt{2}\beta^3 \lambda^{\frac{3}{2}}(k - 1) \cos(\frac{1}{2}\pi k)\Gamma(\frac{4-k}{2})} \text{LagL}\left(\frac{1-k}{2}, \frac{3}{2}, \kappa\right) \left. \right)$$

$$\kappa = \frac{(1 - \beta\gamma)^2 \pi^2}{2\beta^2 \lambda} \quad (I.3)$$

, where  $\text{LagL}(n, a, x)$  denotes the generalized Laguerre function which is a polynomial function of order  $n$ , solution to a certain differential equation. For a constant  $a$  the Laguerre polynomials  $\text{LagL}(\{i, j\}, a, x) = f_{\{i, j\}}(x)$  are orthogonal on the interval  $[0, \infty[$  with respect to the weight function  $w(x)$  [Weisstein, Wolfram Research, 2004]

$$\int_0^\infty w(x) f_i(x) f_j(x) dx = 0, \quad i, j \in \mathcal{N} \quad (I.4)$$

$$w(x) = x^a e^{-x} \quad (I.5)$$

When  $n$  is not a non-negative integer, the analytic extension of the Laguerre polynomials is a polynomial of infinite order, where the coefficients are given by functions of the Gamma-function.

$$\text{LagL}(n, a, x) = \frac{\text{Pochhammer}(a + 1, n)}{n!} {}_1F_1(-n, a + 1, x) \quad (I.6)$$

$$\text{Pochhammer}(x, n) = \frac{\Gamma(x + n)}{\Gamma(x)} \quad (I.7)$$

$${}_1F_1(a, b, x) = \sum_{k=0}^\infty \frac{\text{Pochhammer}(a, k)}{\text{Pochhammer}(b, k)} \frac{x^k}{k!} \quad (I.8)$$

$$\text{LagL}(n, a, x) = \frac{\Gamma(a + 1 + n)}{\Gamma(a + 1)n!} {}_1F_1(-n, a + 1, x) \quad (I.9)$$

$$= \frac{\Gamma(a + 1 + n)}{\Gamma(a + 1)n!} \sum_{k=0}^\infty \frac{\Gamma(-n + k)\Gamma(a + 1)}{\Gamma(-n)\Gamma(a + 1 + k)} \frac{x^k}{k!} \quad (I.10)$$

$$= \frac{\Gamma(a + 1 + n)}{\Gamma(-n)n!} \sum_{k=0}^\infty \frac{\Gamma(-n + k)}{\Gamma(a + 1 + k)} \frac{x^k}{k!} \quad (I.11)$$

# Abbreviations

---

<b>AE</b>	acoustic emission
<b>AEE</b>	acoustic emission energy
<b>AIC</b>	Akaike's information criterion
<b>ATDC</b>	after top dead center
<b>ANN</b>	artificial neural networks
<b>AUC</b>	area under the receiver operator characteristics curve ( <a href="#">ROC</a> ) and is equivalent to the probability of correctly selecting the faulty specimen when given a normal and a faulty specimen
<b>BDC</b>	bottom dead center is the opposite position of the piston to top dead center ( <a href="#">TDC</a> ). Often used as split points when observed signals are segmented into cycles
<b>BIC</b>	Bayesian information criterion
<b>COMADEM</b>	Condition Monitoring and Diagnostic Engineering Management
<b>CDF</b>	cumulated density function
<b>CM</b>	condition monitoring
<b>DTU</b>	Technical University of Denmark

---

<b>DTW</b>	dynamic time warping
<b>EM</b>	expectation Maximization
<b>HWU</b>	Heriot-Watt University, UK
<b>ICA</b>	independent components analysis
<b>iid</b>	independent and identically distributed
<b>INFOMAX</b>	information maximization independent component analysis
<b>MFICA</b>	mean field independent component analysis
<b>NLL</b>	negative log-likelihood
<b>NMF</b>	non-negative matrix factorization
<b>PCA</b>	principal component analysis
<b>ppr</b>	points per revolution
<b>RMS</b>	root mean square
<b>ROC</b>	receiver operator characteristics curve
<b>rpm</b>	revolutions per minute
<b>STFT</b>	short time fourier transform
<b>SVD</b>	singular value decomposition
<b>TDC</b>	top dead center is the point where piston is at its top position and the vertical speed is zero. Often used as a reference point. With angular encoding a marker is placed at <b>TDC</b> for a given cylinder so that all positions are with respect to this <b>TDC</b> – denoted after top dead center ( <b>ATDC</b> ).
<b>UGM</b>	unsupervised Gaussian mixtures

# Bibliography

---

- AEWATT Project Consortium. Deliverable 10, Detection and decision making methods for automated condition monitoring and management of machines. Technical report, July 2005.
- AEWATT Project Consortium. Deliverable 2, AE propagation and signal/event correlation. Technical report, 2003a.
- AEWATT Project Consortium. Deliverable 3, specification of preliminary sensor array. Technical report, 2003b.
- AEWATT Project Consortium. Deliverable 8, Data Acquisition Strategy and Signal Preprocessing. Technical report, January 2004a.
- AEWATT Project Consortium. Mid Term Assessment Report. Technical report, June 2004b.
- appliedinspection.com. Acoustic Emissions Terms and Definitions. Internet. <http://www.appliedinspection.com/aeterms.htm>.
- Michèle Basseville and Igor V. Nikiforov. *Detection of Abrupt Changes: Theory and Application*. Prentice Hall, 1993.
- A. Bell and T. Sejnowski. An information-maximisation approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995. URL <http://citeseer.nj.nec.com/48796.html>.
- Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- G. Chandroth and A. Sharkey. Utilising the rotational motion of machinery in a high resolution data acquisition system. In *Proc of Computers and ships-*

- from ship design and build, through automation and management and on to ship support*, May 1999.
- G. Chandroth, A.J.C. Sharkey, and N.E. Sharkey. Cylinder pressures and vibration in internal combustion engine condition monitoring. In *Comadem 99*, July 1999a.
- G. Chandroth, A.J.C. Sharkey, and N.E. Sharkey. Vibration signatures, wavelets and principal components analysis in diesel engine diagnostics. In *Marine Technology ODRA 99*, October 1999b.
- Knut Conradsen. *En introduktion til statistik*. IMM, 6. edition, 1995.
- Corinna Cortes and Mehryar Mohri. Confidence intervals for the area under the ROC curve. In *NIPS'04*, December 2004.
- David Donoho and Victoria Stodden. When Does Non-Negative Matrix Factorization Give a Correct Decomposition into Parts? In *NIPS2003*, 2003.
- R.M. Douglas, P. Nivesrangsan, A.I.F. Robertson, E.R. Brown, J.A. Steel, R.L. Reuben, and T.L. Fog. Acoustic Emission as a Tool to Reveal Diesel Injector Performance. In [Rao et al. \[2004\]](#), pages 316–324. ISBN 0-954 1307-1-5.
- H.L. Dunegan. A New Acoustic Emission Technique for Detecting and Locating Growing Cracks in Complex Structures. Technical Report 5, DECI Inc., San Juan Capistrano, CA, US, May 2000.
- Mads Dyrholm and Niels Henrik Pontoppidan. Blind signalseparation. Master's thesis, Informatics and Mathematical Modelling, Technical University of Denmark, January 2002. Not published and not available.
- A. Elhaj, F. Gu, A.D. Ball, and J.T. Wright. A study of instantaneous angular speed of a reciprocating compressor for condition monitoring. In [Shrivastav et al. \[2003\]](#), pages 271–279. ISBN 91-7636-376-7.
- Dan Ellis. Dynamic time warp (dtw) in matlab. Internet. URL <http://labrosa.ee.columbia.edu/matlab/dtw/>.
- L. Feng, L. K. Hansen, and J. Larsen. On low level cognitive components of speech. In Honkela et al., editor, *AKKR'05 International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning*, Helsinki, Finland, jun 2005. Pattern Recognition Society of Finland. URL <http://www2.imm.dtu.dk/pubdb/p.php?3664>.
- J.L. Flanagan and R.M. Golden. Phase vocoder. *Bell System Technical Journal*, pages 1493–1509, November 1966. URL <http://www.ee.columbia.edu/~dpwe/e6820/papers/FlanG66.pdf>.

- T. Fog, L.K. Hansen, J. Larsen, H.S. Hansen, L.B. Madsen, P. Sørensen, E.R. Hansen, and P.S. Pedersen. On condition monitoring of exhaust valves in marine diesel engines. In Y. H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing IX*, pages 225–234, Piscataway, New Jersey, 1999. IEEE.
- Torben Fog. *Condition Monitoring And Fault Diagnosis in Marine Diesel Engines*. PhD thesis, Technical University of Denmark, 1998. IMM-PHD-1998-52.
- A.K. Frances, J.D. Gill, J.L. Reuben, and J.A. Steel. A Study of the Variability of Acoustic Emission Signals from a Medium Size Marine Diesel Engine under Service Conditions. In [Shrivastav et al. \[2003\]](#), pages 503–512. ISBN 91-7636-376-7.
- Lindsay K. Galpin, Fraser D. Flemming, and Martin Briddon. Condition based maintenance: A support pillar to reliability centred maintenance. In [Pusey et al. \[2005\]](#).
- Mark A. Goodman. *Ultrasonic Tips On Monitoring Bearings For Proper Lubrication and Wear*, 2004.
- Fredrik Gustafsson. Stochastic observability and fault diagnosis of additive changes in state space models. In *Proceedings of IEEE International Conference on Signal and Speech Processing*, Apr 2001.
- James A. Hanley and Barbara J. McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, April 1982.
- James A. Hanley and Barbara J. McNeil. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology*, 148(3):839–843, September 1983.
- L. K. Hansen, P. Ahrendt, and J. Larsen. Towards cognitive component analysis. In Finnish Cognitive Linguistics Society Pattern Recognition Society of Finland, Finnish Artificial Intelligence Society, editor, *AKRR'05 -International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning*. Pattern Recognition Society of Finland, Finnish Artificial Intelligence Society, Finnish Cognitive Linguistics Society, jun 2005. URL <http://www2.imm.dtu.dk/pubdb/p.php?3591>.
- Lars Kai Hansen, Jan Larsen, and Thomas Kolenda. Blind detection of independent dynamic components. In *Proceedings of ICASSP'2001*, volume 5, May 2001.

- L.K. Hansen and J. Larsen. Unsupervised learning and generalization. In *Proceedings of the IEEE Int. Conf. on Neural Networks 1996*, volume 1, pages 25–30, 1996.
- P. A. Højen-Sørensen, O. Winther, and L. K. Hansen. Mean field approaches to independent component analysis. *Neural Computation*, 14:889–918, 2002. URL <http://www.imm.dtu.dk/pubdb/p.php?611>.
- Aapo Hyvärinen. Survey on Independent Component Analysis. *Neural Computing Surveys*, 2:94–128, 1999. URL <http://www.cis.hut.fi/aapo/papers/NCS99web/NCS99web.html>.
- Peter Storegård Jensen. *Skibsdieselmotorer: Konstruktion, drift, vedligehold*. G.E.C. Gads Forlag, 1994.
- Christian Holm Jørgensen. Analysis for vibration signals. Master’s thesis, Technical University of Denmark, 2003.
- P. Kidmose. *Blind separation of heavy tail signals*. PhD thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2001. URL <http://www2.imm.dtu.dk/pubdb/p.php?794>.
- T. Kolenda, S. Sigurdsson, O. Winther, L. K. Hansen, and J. Larsen. DTU:Toolbox. Internet, 2002. <http://isp.imm.dtu.dk/toolbox/>.
- J. Larsen, A.S. Have, and L.K. Hansen. Probabilistic hierarchical clustering with labeled and unlabeled data. *International Journal of Knowledge-Based Intelligent Engineering Systems*, 6(1):56–62, 2002.
- Jan Larsen, Lars Kai Hansen, Anna Szymkowiak, Torben Christiansen, and Thomas Kolenda. Webmining: Learning from the world wide web. *Computational Statistics and Data Analysis*, 2001. URL <http://citeseer.nj.nec.com/article/larsen01webmining.html>.
- DD. Lee and HS. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, (401):788–791, 1999.
- T. Lee, M. Lewicki, M. Girolami, and T. Sejnowski. Blind source separation of more sources than mixtures using overcomplete representations, 1998.
- Francois Leonard, Marc Foata, and Jean-Yves Paquin. Vibro-acoustic signature comparison and time-warping correction with multi-scale correlation. *Mechanical Systems and Signal Processing*, 14(3):443–458, 2000. ISSN 08883270.
- S. Letourneau, C. Yang, C. Drummond, E. Scarlett, J. Valdes, and M. Zaluski. A Domain Independent Data Mining Methodology for Prognostics. In [Pusey et al. \[2005\]](#).



- Michael S. Lewicki and Terrence J. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12(2):337–365, 2000.
- David J.C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003. ISBN 0521642981.
- David Mba. The use of acoustic emission for estimation of bearing defect size. In Pusey et al. [2005], pages 583–592.
- Tom Minka. `laplace_pca`. Internet. [http://www.stat.cmu.edu/~minka/software/pca/laplace\\_pca.m](http://www.stat.cmu.edu/~minka/software/pca/laplace_pca.m).
- T.P. Minka. Automatic choice of dimensionality for pca. In T.K. Leen, T.G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference*, volume 13, pages 598–604, 2001.
- L. Molgedey and H.G. Schuster. Separation of a mixture of independent signals using time delayed correlations. *Phys. Rev. Lett.*, 72(23):3634–3637, 1994.
- G.D. Neill, S. Benzie, J.D. Gill, P.M. Sandford, E.R. Brown, J.A. Steel, and R.L. Reuben. *The relative merits of acoustic emission and acceleration monitoring for detection of bearing faults*. COMADEM, 1998.
- R. K. Olsson, K. B. Petersen, and T. Lehn-Schiøler. State-space models - from the EM algorithm to a gradient approach. *Neural Computation - submitted*, 2005. URL <http://www2.imm.dtu.dk/pubdb/p.php?3617>. submitted.
- N. H. Pontoppidan and J. Larsen. Unsupervised condition change detection in large diesel engines. In C. Molina, T. Adali, J. Larsen, M. Van Hulle, S. Douglas, and Jean Rouat, editors, *2003 IEEE Workshop on Neural Networks for Signal Processing*, pages 565–574, Piscataway, New Jersey, September 2003. IEEE Press. URL <http://isp.imm.dtu.dk/nns2003>.
- N. H. Pontoppidan and J. Larsen. Non-stationary condition monitoring through event alignment. In *IEEE Workshop on Machine Learning for Signal Processing*, pages 499–508, Piscataway, New Jersey, September 2004. IEEE Press. URL <http://isp.imm.dtu.dk/mlsp2004>.
- N. H. Pontoppidan and S. Sigurdsson. Independent components in acoustic emission energy signals from large diesel engines. *Submitted to International Journal of COMADEM*, 2005. URL <http://www2.imm.dtu.dk/pubdb/p.php?id=3885>.
- N. H. Pontoppidan, J. Larsen, and T. Fog. Independent component analysis for detection of condition changes in large diesels. In Shrivastav et al. [2003]. ISBN 91-7636-376-7. URL <http://www.imm.dtu.dk/pubdb/p.php?2400>.

- N. H. Pontoppidan, J. Larsen, and S. Sigurdsson. Non-stationary condition monitoring of large diesel engines with the AEWATT toolbox. In Pusey et al. [2005]. URL <http://www.imm.dtu.dk/pubdb/p.php?3351>.
- N. H. Pontoppidan, S. Sigurdsson, and J. Larsen. Condition monitoring with mean field independent components analysis. *Mechanical Systems and Signal Processing*, 19(6):1337–1347, nov 2005b. URL <http://dx.doi.org/10.1016/j.ymsp.2005.07.005>. Special Issue: Blind Source Separation.
- N.H. Pontoppidan, S. Sigurdsson, and J. Larsen. AEWATTtoolbox for MATLAB. only available through licensing, 2005c.
- Niels Henrik Pontoppidan and Ryan Douglas. Event alignment, warping between running speeds. In Rao et al. [2004], pages 621–628. ISBN 0-954 1307-1-5. URL <http://www.imm.dtu.dk/pubdb/p.php?3111>.
- Henry C. Pusey, Sallie C. Pusey, and W.R. Hobbs, editors. *Essential Technologies for succesful prognostics*, Proceedings of 59th meeting of the Society for Machinery Failure Prevention Technology, Winchester, VA 22601, April 2005. Society for Machinery Failure Prevention Technology (A Divison of the Vibration Institute).
- R.B. Randall. *Frequency Analysis*. Bruel & Kjaer, Naerum, Denmark, 3 edition, 1987. ISBN 87 87355 07 8.
- R.B. Randall and J. Antoni. A New FFT-Based Method of Separating Discrete-Frequency and Random Signals. In Shrivastav et al. [2003], pages 415–423. ISBN 91-7636-376-7.
- Raj B.K.N. Rao, Barry E. Jones, and Roger I. Grosvenor, editors. *Condition Monitoring and Diagnostic Engineering Management, Proceedings of the 17th International Congress*, Birmingham, UK, 2004. COMADEM International, COMADEM International. ISBN 0-954 1307-1-5.
- R.L. Reuben. The role of acoustic emission in industrial condition monitoring. *International Journal of COMADEM*, 1(4):35–46, October 1998. ISSN 1363-7681.
- G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6: 461–464, 1978.
- Clayton Scott and Jeffrey Silverman. The Newman-Pearson Criterion. internet, 2004. <http://cnx.rice.edu/content/m11548/latest>.
- L.F. Shampine, R.C. Jr. Allen, and S. Pruess. *Fundamentals of Numerical Computing*. John Wiley & Sons, Inc., 1997. ISBN 0-471-16363-5.

- A.J.C. Sharkey, N.E. Sharkey, U. Gerecke, and G.O. Chandroth. The “test and select” approach to ensemble combination. In *Lecture Notes in Computer Science*, volume 1857 of *Proceedings of First International Workshop on Multiple Classifier Systems*, pages 30–44, 2000. ISBN 3-540-67704-6.
- Om P. Shrivastav, Bassim Al-Najjar, and Raj B.K.N. Rao, editors. *Condition Monitoring and Diagnostic Engineering Management, Proceedings of the 16th International Congress*, 2003. COMADEM International, Växjö University Press. ISBN 91-7636-376-7.
- Sigurdur Sigurdsson. *A Probabilistic Framework for Detection of Skin Cancer by Raman Spectra*. PhD thesis, Technical University of Denmark, Informatics and Mathematical Modelling, 2003.
- C.K. Tan, P. Irving, and David Mba. Prognostic Potential of the Acoustics Emission Technique for Spur Gears. In Pusey et al. [2005], pages 67–76.
- Runar Unnthorsson, Niels Henrik Pontoppidan, and Magnus Thor Jonsson. Extracting information from conventional AE features for fatigue onset damage detection in carbon fiber composites. In Pusey et al. [2005]. URL <http://www.imm.dtu.dk/pubdb/p.php?3289>.
- J. van de Touw and A. Veevers. Signal Matching of Rail Track Monitoring Data. *International Journal of COMADEM*, 6(4):33–41, October 2003.
- Eric W. Weisstein. Laguerre Polynomial. From MathWorld – A Wolfram Web Resource. URL <http://mathworld.wolfram.com/LaguerrePolynomial.html>.
- C.J. Whitaker and L.I. Kuncheva. Examining the relationship between majority vote accuracy and diversity in bagging and boosting, 2000. URL <http://citeseer.nj.nec.com/507415.html>.
- Wolfram Research. Mathematica Documentation: LaguerreL. Internet, November 2004. URL <http://documents.wolfram.com/mathematica/functions/LaguerreL>.
- Alexander Ypma. *Learning methods for machine vibration analysis and health monitoring*. PhD thesis, Dept. of Applied Physics, Delft University of Technology, November 2001.