14th IMEKO TC10 Workshop Technical Diagnostics
New Perspectives in Measurements, Tools and Techniques
for system's reliability, maintainability and safety
Milan, Italy, June 27-28, 2016

Automated Parameter Optimization for Feature Extraction for Condition Monitoring

Mike Gerdes[1], Diego Galar[1,2], Dieter Scholz[3]

[1] Luleå University of Technology, Division of Operation and Maintenance Engineering, SE-971 87 Luleå, Sweden, email: diego.galar@ltu.se
[2] Luleå University of Technology, Division of Operation and Maintenance Engineering, SE-971 87 Luleå, Sweden, email: diego.galar@ltu.se
[2] Hamburg University of Applied Sciences, Aero – Aircraft Design and Systems Group, Berlinter Tor 9, D-20099 Hamburg, Germany, email: info@profscholz.de

***Abstract* - Pattern recognition and signal analysis can be used to support and simplify the monitoring of complex aircraft systems. For this purpose, information must be extracted from the gathered data in a proper way. The parameters of the signal analysis need to be chosen specifically for the monitored system to get the best pattern recognition accuracy. An optimization process to find a good parameter set for the signal analysis has been developed by the means of global heuristic search and optimization. The computed parameters deliver slightly (one to three percent) better results than the ones found by hand. In addition it is shown that not a full set of data samples is needed. It is also concluded that genetic optimization shows the best performance.**

I. INTRODUCTION

An aircraft consists of many complex systems, which together define the state of the aircraft. Many systems are difficult to monitor or provide little information to the aircraft maintenance systems. In Gerdes et al. [1] potential savings were analyzed, if a faulty system is replaced before it fails. Faults leading to a delay can be often prevented (in 20% of the cases without additional sensors and about 80% with additional sensors). For the air conditioning system these faults include material weaknesses or broken valves. However to prevent these faults a new maintenance and monitoring strategy is needed. This strategy is condition based maintenance (CBM). It is either possible to replace the system/component/part on a fixed interval like it is commonly done in aircraft maintenance or to monitor the condition of the system and predict when the system will fail. The condition monitoring approach needs a deep understanding of the system and its behaviour. Ideally, a computer should be able to compute the condition of a system to reduce the amount of humans involved in the process.

CBM differs strongly from the traditional aircraft maintenance strategy (Reliability Centred Maintenance). Condition-based maintenance (CBM) is based on condition monitoring and aims at preforming maintenance based on the system condition and trend of the system condition. CBM can be used as a way to realize RCM [2]. The focus of CBM was traditionally on diagnosis, but with recent developments fault prognosis gets more in the focus [3]. Introducing new maintenance concepts into the civil aircraft environment is a difficult and complex task due to the focus on safety. A hybrid approach that uses the established maintenance plans as the first priority and condition based maintenance as the second priority might be a method to introduce condition monitoring into the aircraft [4]. Aerospace regulations also require that any decisions on maintenance, safety and flightworthiness to be auditable and data patterns to relate to known information [4].

The goal of this paper is to automate the selection of a good parameter set for the feature extraction to generate features, which yields the best results when analysed by the pattern recognition algorithm in order to get better classification results for the condition monitoring. Another goal coherent with the big data issues is to use the automation to reduce the need of human configuration of the system since data collected are so huge that the role of humans in data cleaning and preprocessing is consuming too many resources in the industry [5].

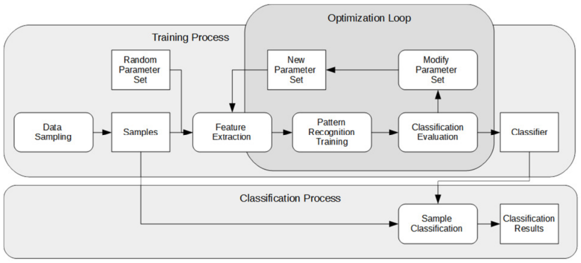
II. PROPOSED METHOD

The proposed method is based on condition monitoring with machine learning. The basic condition monitoring process consists of data sampling, feature extraction and pattern learning followed by classification of new data samples. An optimization loop for the pattern learning is added to enhance the performance of the learning and the resulting classifier. Goal of the optimization is to improve the feature extraction to improve the performance of the pattern recognition. Traditional approaches first extract features and then use these features for pattern recognition [6]. In contrast here the feature extraction depends on the results of the pattern recognition and is not independent. The feature extraction is controlled by a set of parameters which modify and enable or disable feature extraction operations.

The feature extraction and the corresponding parameters influence the accuracy of the pattern recognition strongly. Experiments [7] showed that the solution space for an optimal parameter set is not linear and does have many local minima. Such a problem is difficult to optimize with traditional methods, because the solution space is very large. An automated parameter configuration is needed to find an optimal parameter set that will improve the performance of the pattern recognition. Heuristic search methods, which search for a minimum or maximum, can help to find a good solution to the optimization problem. The goal of the optimization is to maximize the percentage of the correctly classified data samples. .

Traditionally the pattern recognition algorithms are optimized by modifying the underlying algorithm. This optimization concept doesn't touch the optimization algorithm. It optimizes the input data so that a high accuracy is gained. As a side effect the chosen parameters show which signal processing steps are important and which are not needed for a successful classification.

Fig. 1. Condition Monitoring Process with Parameter Optimization



The complete process is shown in Fig. 1. The optimization takes part during the learning step, which is needed before any new data can be classified. The process starts by generating a random parameter set for the feature extraction. The calculated features are then fed into a pattern recognition algorithm that searches for patterns and learns these patterns..

The resulting pattern recognition is then tested with a test set and yields classification accuracy. Then the optimization loop is entered and a new parameter set is generated, based on an optimization algorithm (Greedy Search, Simulated Annealing or Genetic Algorithm). After the new parameter set is generated the loop starts again with the feature extraction for the training data. The output of the process is a parameter set and a pattern classifier, which are used for the feature extraction and classification in the condition monitoring process.

A. Feature Extraction

The feature extraction starts with noise reduction. First, the data is transformed into the frequency domain, where the noise is reduced. Then, frequencies are grouped. Mean and maximum power is calculated for every frequency group, as well as the number of peaks. Then each group is transformed back into the time domain, where the mean and maximum amplitudes are calculated. The mean and maximum frequency power and mean and maximum amplitude of the complete signal is calculated as a last step. Table 1 shows the parameters of the feature extraction and the possible values.

Block Width defines how many frequencies are grouped in the frequency domain to form a block for detailed feature extraction.

The **Noise Reduction Factor** defines how much noise will be reduced. The noise reduction in this concept removes all frequencies in the frequency which power is below *noise reduction factor* times *mean power.*

Peak Border controls what frequencies are defined as peaks. Any frequency which power is greater or equal than the *Peak Border* times the *Mean Power* is defined as a peak.

The **confidence factor** controls how much tree pruning is done and is a parameter of the J48 algorithm of the WEKA software [8]. A confidence factor of greater than 0.5 means that no tree pruning is done. The lower the confidence factor is the more pruning is done.

Table 1. Feature Extraction Parameters

Parameter	Possible Values	Default Value
Block Width	5/50/100/200	100
Noise Reduction Factor	0/1/2/5	1
Maximum Amplitude	Yes/No	Yes
Mean Amplitude	Yes/No	Yes
Maximum Power	Yes/No	Yes
Maximum Frequency	Yes/No	Yes
Mean Power	Yes/No	Yes
Number of Peaks	Yes/No	Yes
Peak Border	1/2/5	2
Global Maximum Amplitude	Yes/No	Yes
Global Mean Amplitude	Yes/No	Yes
Global Maximum Power	Yes/No	Yes
Global Mean Power	Yes/No	Yes
Global Number of Peaks	Yes/No	Yes
Confidence Factor	0.0001/0.001/0.01/0.1/1	0.001

All other parameters are Boolean parameters which control if a given feature is calculated or not. Elementary feature extraction operations can be executed in any order and allow the creation of a set of feature extraction operations the can be different for each problem [9]. This makes elementary extraction operations also good for machine learning. The used operators are fast to compute and can be used for online monitoring..

The used feature extraction operations are determined by the parameters in Table 1. The values for the parameters are randomly generated or generated during the optimization using a search algorithm, which is shown in the next section.

B. Pattern Recognition Training

Pattern recognition belongs to the area of artificial intelligence. It is used to find patterns in data that allows the algorithm to categorize that data. First the algorithm has to "learn" or find the patterns in the data and construct a function or algorithm that represents that data. After that, new data samples can use the function or algorithm to categorize the new data based on the experience of the old data. This method uses supervised learning, where each data sample belongs to a known predefined class. Possible algorithms are Decision Trees, Support Vector Machines or Bayesian Networks. Artificial Neural Networks were not included in the evaluation, because they are a not deterministic pattern recognition algorithm. Deterministic learning is an important factor when the aircraft environment is considered, which the main driver for this research was.

C. Optimization Loop

The basic pattern learning process as explained above is improved with an optimization loop, which shall improve the classification accuracy by selecting an optimal process parameter set for the feature extraction step. The optimization can use different algorithms to find the optimum. Local search methods are recommended due to large size of the solution space.

III. VALIDATION

The data which was used for the experiments was generated on a test rig (see Fig. 2) by Airbus in Hamburg. The test rig simulates a part of the air circulation system of an A340-600 aircraft. Valves control the airflow of the two inlets and the outlet at the bottom. For the experiment different valve positions were chosen and the vibration of the recirculation fan was recorded. A one second long sample was recorded every ten seconds. The outlet valve was never fully closed to prevent permanent damage of the equipment. In total 25 conditions were recorded (see Table 2).

Table 2. Valve positions for the experiment (0° is fully open and 90° is fully closed)

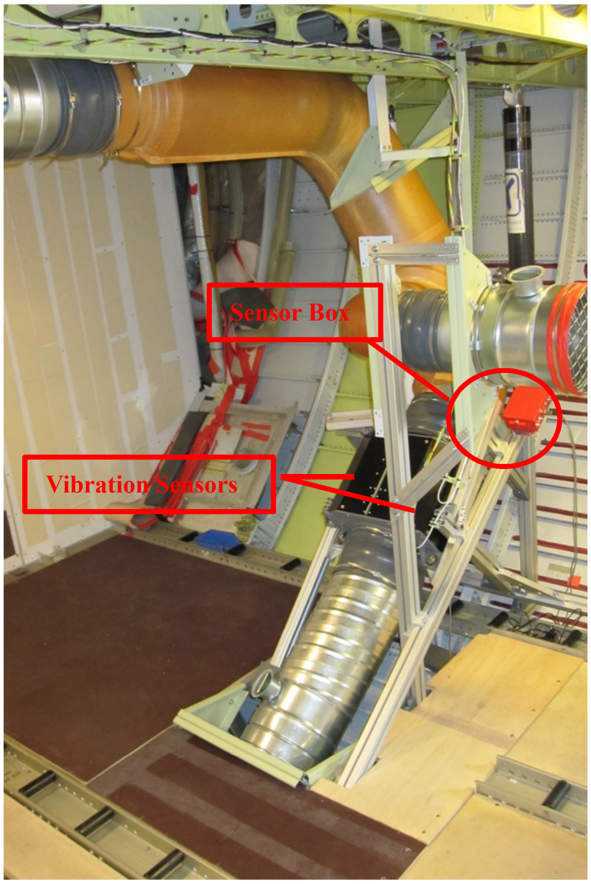
Inlet Valve 1 Position	Inlet Valve 2 Position	Outlet Valve Position
0°	0°	0°
0°	0°	45°
0°	30°	0°
0°	45°	0°
0°	45°	45°
0°	60°	0°
0°	90°	0°
0°	90°	45°
45°	0°	0°
45°	0°	45°
45°	30°	0°
45°	45°	0°
45°	45°	45°
45°	60°	0°
45°	90°	0°
45°	90°	45°
60°	60°	0°
90°	0°	0°
90°	0°	45°
90°	30°	0°
90°	45°	0°
90°	45°	45°
90°	60°	0°
90°	90°	0°
90°	90°	45°

Every condition was monitored for four minutes, which results in 24 samples per condition. In total 600 data samples were recorded. The data was recorded using an autonomous recording box by Airbus. Two vibration sensors were attached to the test rig (see Fig. 2). Vibration and sound data was sampled with a rate of 44 kHz. Data is saved in a raw wave format with two channels onto a SD card and then transferred onto a PC.

All experiments used a 10-fold cross-validation to check the performance of the calculated pattern recognition. The software WEKA [8] was used for the experiments.

Three different experiment setups were used to evaluate the concept with the mentioned data set form test rig above, optimization algorithms and pattern recognition algorithms. The same data set was used for all three experiments. Each experiment modified the optimization algorithm, the learning algorithm or the number of uses data sets for training.

Fig. 2. Airbus Test Rig for Data Recording



D. *Experiment 1: Sample Data*

In the first experiment the influence of the number of samples on the calculation time and the pattern recognition accuracy was evaluated. The optimization and learning process was tested five times with an increasing number of training samples (5, 10, 15, 20, and 24) per class. 24 samples were the maximum possible number of samples per class (all recorded samples). The best solution that was found with the reduced sample set was used to classify the full sample set. This was done to show if it was possible to train the algorithm with a reduced data set and gain the full classification accuracy. The genetic algorithm and SVM were used.

E. *Experiment 2: Optimization Algorithm*

Three different optimization algorithms (Greedy Search, Simulated Annealing, and Genetic Algorithm) were tested alone with different parameter sets. As a second test Simulated Annealing and Genetic Evolution were chained so that one produced starting points for the other one. The idea is to use one algorithm to find a good starting point so that the following algorithm can use that point to perform even better, than it would normally do alone. The single algorithm experiments and the chained experiments used the same number of function evaluations to be comparable. All algorithms started at the same starting point. The genetic algorithm generated additional random starting points up to the needed population size.

- Greedy Search
- Simulated Annealing
- Genetic Algorithm
- Simulated Annealing and Genetic Algorithm
- Genetic Algorithm and Simulated Annealing

The abort criteria for the different learning algorithms were:

- Greedy Search stops, if the best value does not change for 30 steps.
- Annealing stops, if 480 optimization steps are executed.
- Genetic evolution stops, if 20 generations with 24 individuals have been evaluated (for a total of 480 evaluated individuals).

F. *Experiment 3: Pattern Recognition*

In this experiment the performance of the three algorithms for pattern recognition (Decision Tree, Bayesian Network and Support Vector Machine) were compared using a Genetic Algorithm for optimization. The run time of the algorithms was measured beside the percentage of correctly classified samples.

IV. RESULTS AND DISCUSSION

The results of the experiments as they were described in the previous section are shown below. The calculations were done using an older Intel dual core processor. They should not be viewed as absolute times. Calculation time is just to compare the relative speed of the algorithms.

G. *Experiment 1: Number of Samples*

The pattern recognition accuracy between two sample data bases with a different number of samples varies significantly (Table 3). As it is visible, the classification accuracy of the method with the smaller sample base and the full sample base are very similar. The difference is only for the 5 data sample base significantly. This means that the data samples contain enough significant data so that only 10 samples are needed for the training to get a good classification result. This observation can be used to reduce the training time a lot, if only half of the available data samples are taken. Another advantage of this approach is that the other half of the data samples can be used to verify the classification results as a testing data set. It is worthy to note that this indicates a high resilience of the input data to noise and that only a few data samples are enough to ensure a good classification of the new data samples. The results also show that the used algorithms are good at generalizing.

Table 3. Evaluation of different data sample sizes

Data Samples per Class	Correctly Classified	With 24 Samples for Testing	Calculation Time
5	90 %	96 %	1166 seconds
10	96 %	97 %	2767 seconds
15	97 %	96 %	3572 seconds
20	98 %	96 %	6182 seconds
24	98 %	98 %	6700 seconds

H. Experiment 2: Optimization Algorithm

The selection of the algorithm greatly influences the performance of the optimization as this section shows.

No Optimization

A calculation without an optimization step was done to be able to judge and evaluate the results of the different optimization algorithms.

24 random parameter sets were generated, evaluated and the best parameter set was selected. This resulted in an accuracy of 97.5 % and took 517 seconds.

Greedy Search

The best result of the Greedy Search algorithm was 97.7 % and the calculation time was only 1250 seconds. This is as expected. Greedy Search is a really fast algorithm but it also can get stuck in a local maximum easily. To get better results the algorithm needs to be executed more than one time, which negates the speed advantage.

Simulated Annealing

Simulated Annealing had about the same speed as the Genetic Algorithm of about 5605 seconds. This is unsurprisingly due to the fact that both algorithms evaluated the function 480 times (the same number of iterations as for the genetic algorithm). Simulated Annealing achieved an accuracy of 97.7 %, which is similar to the Greedy Search algorithm and a bit worse than the Genetic Algorithm. The problem space contains many local maxima and is very huge. Simulated Annealing does not get trapped in a local maximum as fast as Greedy Search, but can also fall in that trap if the problem space contains very many local maxima.

Genetic Algorithm

The Genetic Algorithm had the highest accuracy with 98 %. It needed 5418 seconds to finish. The Genetic Algorithm delivers the similar results as simulated annealing. It searches at multiple places at once and then chooses the best ones to continue.

Simulated Annealing and Genetic Algorithm

Using Simulated Annealing to create a base population works quite well, however the results are not better than using the Genetic Algorithm alone (98.6 %) and the calculation time was twice as long.

Genetic Algorithm and Simulated Annealing

The idea to use the best parameter set of the Genetic Algorithm as a starting point for Simulated Annealing works also well and results in an accuracy of 98.3 %. The calculation time again was twice as long as for a single algorithm.

I. Experiment 3: Pattern Recognition

Table 4 shows accuracy of the different pattern recognition algorithms with genetic algorithm optimization. To be able to use the Bayesian Network algorithm all values need to be discretized (a feature can only have a pre-defined value). If numerical values are used, then Weka [8] needs to discretize the feature values automatically, which results in a "No memory left" error. To limit the amount of needed memory and make the calculation feasible the maximum number of blocks was limited to 15, 10 data samples per class and the bandwidth of the input data was only 7.5 kHz (half the bandwidth of the original data samples).

Table 4. Evaluation of different pattern recognition algorithms with optimization

Pattern Recognition Algorithm	Correctly Classified Samples	Calculation Time
Decision Trees	94.4 %	1381 seconds
SVM	99.4 %	12312 seconds
Bayesian Network	99.4 %	2791 seconds

It is visible in Table 4 that the SVM performs the best and Decision Trees perform worst. The Bayesian Network algorithm works well because of the reduced amount of features; however the Decision Tree algorithm seems to suffer from the reduced number of features and performs weak.

In Table 1 are the three different algorithms tested with the same parameter set and without optimization. It is visible that SVM delivers again the best results. There is a minimal optimization included in the calculation. 24 random parameter sets were generated (the same as the starting parameter set for Table 4) and then the parameter set with the best performance was used.

Table 5. Evaluation of different pattern recognition algorithms without optimization

Pattern Recognition Algorithm	Correctly Classified Samples	Calculation Time
Decision Trees	91.7 %	71 seconds
SVM	98.9 %	1860 seconds
Bayesian Network	98.9 %	193 seconds

While Bayesian Networks deliver good results, they are not the best. Table 5 also shows that the calculation time depends on the number of the blocks and thus the total number of features for the training. If that number is restricted, then all algorithms perform significantly faster. The optimization process doesn't give a significant improvement in this setup. This is due to the fact of the solution space was much smaller and that the random starting points where good.

V. CONCLUSIONS

Adding an optimization loop to for improving the feature extraction parameters and thus to improve the classification accuracy showed good results. The classification accuracy of all tested learning algorithms improved just by having a better feature set and without making any changes at the learning algorithms. The results show that an optimization can increase the performance of the signal analysis and pattern recognition. However the increase is less than 5 %, which is also due to the high noise resilient input data. Still it is possible to push the accuracy up to 99.4 %. Genetic algorithms performed well even with the short searches with a small population. All algorithms showed a good performance compared to choosing parameters by hand, which is nearly equal to choose a random parameter set. One goal of the research was to reduce the amount of expert knowledge to use the system is achieved. By using automatic parameter optimization no expert is needed to adept the feature extraction parameters for the problem instead the algorithms adopts itself to the given data. The concept works well if a significant number of data samples are available.

Another advantage of the concept is that it can be parallelized without much work, if a genetic algorithm is used. The members of the population can be spread over the available processor. With parallelization it is possible to reduce the computation time a lot and a much larger space can be searched in the same time.

The research results are based on the data from the Airbus test rig, which does not represent real world data. Real world data for aircraft components is difficult to get because of safety restrictions, which makes it really difficult to install measuring equipment in a non-test flight airplane. Also the number of tested algorithms is quite restricted due to the fact that only deterministic pattern recognition methods were considered. However it is possible to use different algorithms and methods for pattern recognition and optimization.

Future work will include testing the concepts for real world data and use of the method for condition monitoring and trending.

VI. CITATIONS AND REFERENCES

[1] M. Gerdes, D. Scholz und B. Randerath, „Reducing Delays Caused by Unscheduled Maintenance and Cabin Reconfiguration,“ in 2nd International Workshop on Aircraft System Technologies, AST 2009 (TUHH, Hamburg, 26./27. March 2009), 2009.

[2] G. Niu und M. Pecht, „A framework for cost-effective and accurate maintenance combining CBM RCM and data fusion,“ in Reliability, Maintainability and Safety, 2009. ICRMS 2009. 8th International Conference on, 2009.

[3] A. K. Jardine, D. Lin und D. Banjevic, „A review on machinery diagnostics and prognostics implementing condition-based maintenance,“ Mechanical systems and signal processing, Bd. 20, Nr. 7, pp. 1483-1510, 2006.

[4] P. Phillips und D. Diston, „A knowledge driven approach to aerospace condition monitoring,“ Knowledge-Based Systems , Bd. 24, Nr. 6, pp. 915-927, 2011.

[5] H. V. Jagadish, J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan und C. Shahabi, „Big Data and Its Technical Challenges,“ Communications of the ACM, Bd. 57, Nr. 7, pp. 86-94, July 2014.

[6] M. H. Nguyen und F. De la Torre, „Optimal feature selection for support vector machines,“ Pattern recognition, Bd. 43, Nr. 3, pp. 584-591, 2010.

[7] M. Gerdes und D. Scholz, „Feature Extraction and Sensor Optimization for Condition Monitoring of Recirculation Fans and Filters,“ in Deutscher Luft- und Raumfahrtkongress 2009 : Tagungsband - Manuskripte (DLRK, Aachen, 08.-10. September 2009), 2009.

[8] University of Waikato, WEKA, 2009.

[9] I. Mierswa und K. Morik, „Automatic feature extraction for classifying audio data,“ Machine learning, Bd. 58, Nr. 2-3, pp. 127-149, 2005.