AST 2011

Workshop on Aircraft System Technology

PARAMETER OPTIMIZATION FOR AUTOMATED SIGNAL ANALYSIS FOR CONDITION MONITORING OF AIRCRAFT SYSTEMS

Mike Gerdes*[1], Dieter Scholz[1]

[1]Aero - Aircraft Design and Systems Group, Hamburg University of Applied Sciences
Berliner Tor 9, 20099 Hamburg, Germany

mike.gerdes@haw-hamburg.de

Abstract

In the PAHMIR (Preventive Aircraft and Health Monitoring) project pattern recognition and signal analysis is used to support and simplify the monitoring of complex aircraft systems. 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 was developed that uses global heuristic search and optimization to find a good parameter set for the signal analysis. 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. Genetic optimization showed the best performance.

1 INTRODUCTION

An aircraft consists of many complex systems, which together define the state of the aircraft. Many systems are difficult to monitor or they give only little information to the aircraft maintenance systems. In [1] it was analyzed how much money could be saved, if a faulty system is replaced before it fails. Faults leading to a delay, can be prevented. It is either possible to replace the system 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 behavior. Ideally, a computer should be able to compute the condition of a system to reduce the amount of humans involved in the process. One approach is to let the monitoring system learn the behavior of a monitored system. Future faults can be extrapolated based on the condition of a system in the present and the past.

In the PAHMIR (Preventive Aircraft Health Monitoring for Integrated Reconfiguration) project systems are monitored by watching the inputs and outputs of a system and then forecast an error based on these values. Focus of the monitoring is on sound and vibration. The monitoring data is preprocessed and then automatically analyzed by a pattern recognition algorithm. Pattern recognition is used to find patterns in the data that represent a certain condition of the system. In [2] signal analysis and machine learning are used to detect the condition of an experimental setup. The concept is half automated and does need much fine tuning by hand. This is because the process depends on several different parameters. Each of these parameters need to be adapted to the data. Goal of this paper is to automate the selection of an optimal parameter set for the preprocessing to generate data, which yields the best results when it is analyzed by the pattern recognition algorithm. These parameters include:

- Signal transformation from the time domain into the frequency domain
- Noise reduction
- Grouping of frequencies
- Calculation of the maximum and mean frequency power of every frequency group
- Calculation of the number of peaks of every group
- Transformation of the frequency groups back into the time domain
- Calculation of the maximum and mean amplitudes
- Calculation of the maximum and mean values of the complete signal

The process for the condition monitoring in PAHMIR is shown in Figure 1. Input data is recorded and then is sent into a preprocessing step, where noise is removed and additional information is extracted from the data. The data is sent to a pattern recognition algorithm that tries to categorize the data after it is prepared. The performance of the pattern recognition algorithm is influenced by the number of available samples to learn from, the data preprocessing and the algorithm itself.

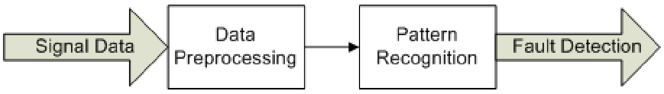

Figure 1: The PAHMIR condition monitoring process

1.1 Signal Data

The proposed concept can work with any kind of input data, however it is assumed that the data is a discrete signal with more than one data point. A sampling frequency of higher than 1 kHz is required. If a lower frequency is used, then the preprocessing needs to be adapted to that frequency. The signal source does not matter, it can be sound, vibration, temperature power consumption, weight or magnetic flow data as long as it is a one dimensional time series source. If more than one data source is used or a data sample does have more than one dimension, then the preprocessing algorithm also needs to be adapted or the data needs to be transformed. The simplest way is to have one preprocessing step for every dimension of the data and then concatenate the preprocessed data before giving it to the pattern recognition.

1.2 Preprocessing

Noise and the amount of data are reduced and additional information is added during preprocessing to the data. First, the data is transformed into the frequency domain, where the noise is reduced. Then, frequencies are grouped. It is possible that the frequency span of the groups overlap each other. E.g. if the frequencies 1 to 50 belong to one group and have an overlap of 50 %, then the second group contains the frequencies from 26 to 75 and the third group contains the frequencies from 51 to 100. Mean and maximum power are 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 preprocessing and the possible values.

Table 1: Preprocessing parameters

Parameter Name	Possible Values	Unit
Block width	0–1000	Hz
Block overlap	0–50	%
Noise reduction	0–5	–
Calculate the mean amplitude for each block	true or false	–
Calculate the maximum amplitude for each block	true or false	–
Calculate the mean frequency power for each block	true or false	–
Calculate the maximum frequency power for each block	true or false	–
Calculation the number of peaks for each block	true or false	–
Minimum Value of a peak	0–5	–
Calculate the overall mean and maximum values	true or false	–

1.3 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. The original concept of PAHMIR uses Decision Trees [2]. In this paper two other algorithms (Bayesian Network and Support Vector Machine) are compared with Decision Trees (see Section 2.1).

2 METHODS

The basic concept of the PAHMIR condition monitoring process was modified in two ways. First different pattern recognition algorithms were tested and second an optimization process was developed to find a good parameter set for the data preprocessing.

2.1 Pattern Recognition

The influence of different pattern recognition algorithms was analyzed in experiments. Only algorithms that can evaluate a data sample fast, can be understood and checked by a human and have low hardware requirements can be used for the pattern recognition in PAHMIR. The restrictions for algorithms are given by the aircraft environment. Three algorithms were chosen for comparison: Decision Tree, Bayesian Network and Support Vector Machine.

Decision Trees

Decision Trees are very easy to understand. They are unidirectional trees. The leafs are the categorizes of the data, while the nodes are if-then decisions. Decision trees use the concept of information gain for learning and finding the attribute in the data that divides the data so that the most information is gained. [3] explains decision trees in further detail.

Bayesian Network

Bayesian networks are trees like decision trees. A Bayesian network does not use information gain to construct the tree, instead Bayesian networks use conditional probability. Conditional probability means the probability that an attribute has a certain value, if the value of another related attribute is known. Nodes of the network contain the probabilities of choosing the next node based on knowledge of the stare of some other nodes. Bayesian networks are explains in further detail in [3] .

Support Vector Machines

Support Vector Machines or SVMs are a complete numerical approach. The goal is to separate the data samples by a linear function into classes. If it is not possible to separate the data into classes with a "straight line" then the data set is transformed into a higher dimensional space until it is possible to separate the data samples. A special kind of function is used to perform the transformation. The data is transformed by weighted dot products. The weight vector is called support vector. [3] gives more details on SVMs.

2.2 Optimization Concept

The preprocessing and the corresponding parameters influence the accuracy of the pattern recognition strongly. Tests of the learning function showed that the solution space 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.

An optimization step is included in the process shown in Figure 1. The optimization takes part during the learning step, which is needed before any data can be evaluated. Figure 2 shows the modified PAHMIR condition monitoring process. First a sample data set is preprocessed with a random parameter set. The preprocessed data is then fed into a pattern recognition algorithm that searches for patterns. The resulting algorithm is then tested and yields an accuracy percentage. Then the optimization loop is entered and a new parameter set is chosen, based on an optimization algorithm (Greedy Search, Simulated Annealing or Genetic Algorithm). After the new parameter set is chosen the loop starts again. The output is a parameter set, which is used for the data preprocessing in the PAHMIR condition monitoring process (Figure 1). All three algorithms are adapted to the problem and start from a given data base of parameter sets. The parameter range of the mutations (for Simulated Annealing and the Genetic Algorithm) is also adapted and all algorithms work with the same in and output to be able to chain them.

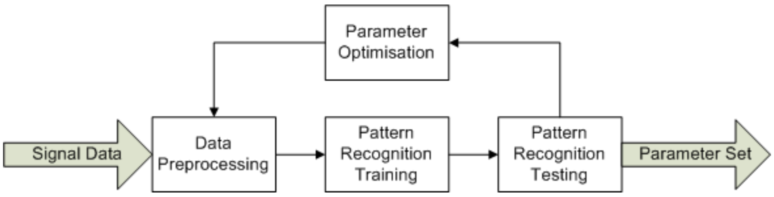

Figure 2: PAHMIR condition monitoring process with optimization

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.

Greedy Search

Greedy search is the simplest optimization algorithm of the three that were analyzed. Greedy search starts at a random starting point in the parameter space and compares the value of that point with the value of a random neighbor. If the neighbor performs better, then the neighbor is chosen, otherwise the current parameter set stays at the starting point. In [3] is a more detailed explanation of the Greedy Search algorithm.

In this paper the greedy search algorithm is implemented without any modifications. A neighbor is defined by a block size of 1 Hz to 50 Hz and a block overlay that varied by up to 10 % of the starting point. All other values can vary by one point and have a 50 % probability of changing. Greedy Search stops, if the best value does not change for 30 steps.

Simulated Annealing

Simulated annealing is a more complex variant of greedy search. In simulated annealing the algorithm may choose a worse neighbor depending on a probability, which decreases with the difference between the performance of the neighbor and the number of performed steps. With simulated annealing it is possible to move away from a local maximum and not to get struck. There is a more detailed explanation of the Simulated Annealing algorithm in [3].

A new neighbor is found by varying the block size up to 500 Hz and the block overlay up to 20 %. The range of both values decreases with the number of performed function evaluation. All other values vary by one point and have a probability to do so of 50 %. The function that controls when a worse neighbor is selected is a linear function. A neighbor performance value may be up to 20 % worse than the current point. This value decreases linearly over time until it reaches 0. Simulated Annealing stops, if 480 optimization steps are executed. The best value is returned and not the current value.

Genetic Algorithm

The genetic algorithm is a more complex simulated annealing. It evaluates different points in parallel, chooses the best and creates new variations of those by combining and changing the parameter sets. With the genetic algorithm it is possible to search a wider area in the problem space and a higher chance to find the global maximum. The algorithm performs multiple evaluations in parallel thus it is slower than the other two algorithms. In [3] is a more detailed explanation of the Genetic Algorithm.

The genetic algorithm used the same mutations as the Simulated Annealing algorithm. Reduction of the mutation variance over time was used to force a better convergence. New children were created by taking the block width of one member, the block overlay of another, one part of the other remaining parameters from a third parent and the rest parameters form a fourth parent. The first third of the population was left unchanged, the rest of the population can mutate. Genetic evolution stopped, if 20 generations with 24 members have been evaluated.

3 EXPERIMENTS

Four different experiment setups were used to evaluate the concept, optimization algorithms and pattern recognition algorithms. First the influence of different input data sizes, then the influence of choosing different starting points was tested. Third different optimization algorithms (Greedy Search, Simulated Annealing and Genetic Algorithm) and combinations of them were tested with the sample data. Then different pattern recognition algorithms (Decision Trees, Bayesian Networks and SVMs) were tested.

Data which was used for the experiments was generated in a test rig (see Figure 3. The test rig simulates a part of the air recirculation system of an A340-600 aircraft. Valves control the two inlets and the outlet. For the experiment data different valve conditions are chosen and then the vibration of the recirculation fan was recorded. One sample was recorded every 10 seconds. Every condition was monitored for 4 minutes, which results in 24 samples per condition. In total 25 conditions were recorded. All experiments used a 10-fold cross-validation to check the performance of the calculated pattern recognition.

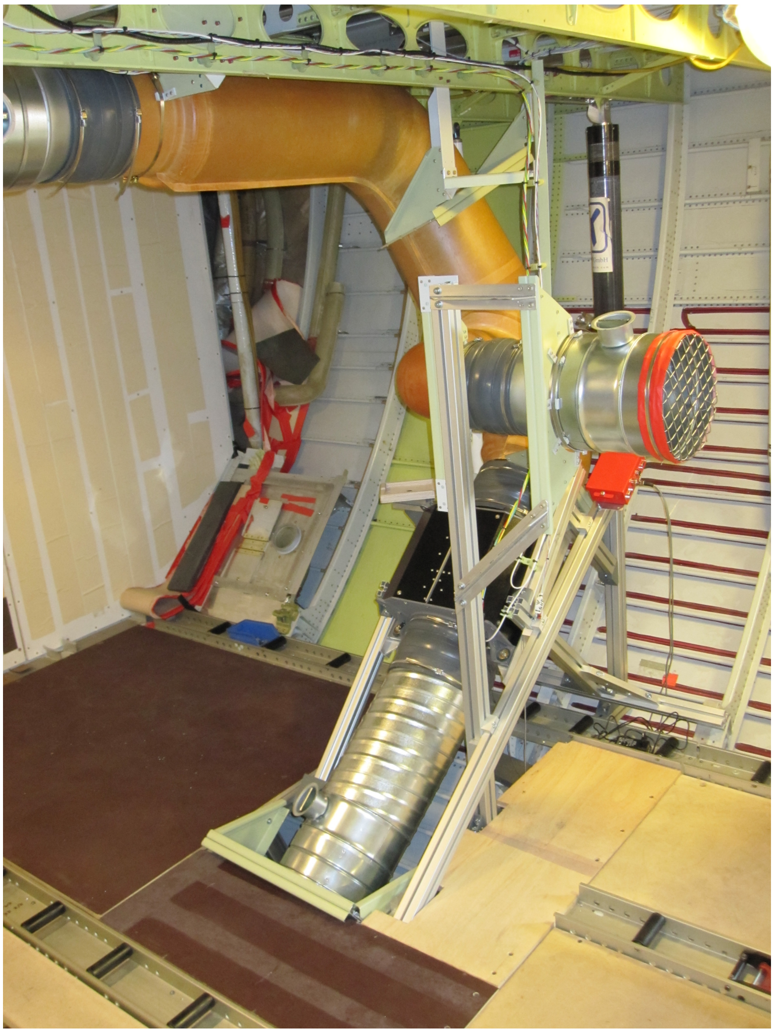

Figure 3: Test rig for recording experiment data

3.1 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 samples (5, 10, 15, 20, 24) per class. 24 samples was 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 setup was used (see subsection 2.2).

3.2 Starting Points

An experiment was performed to show the effect of starting at different points. Genetic algorithm was used with 20 samples per class. The algorithm was evaluated ten times with different randomly selected starting populations.

3.3 Optimization Algorithm

The three different optimization algorithms (Greedy Search, Simulated Annealing, and Genetic Algorithm) were tested alone with different parameter sets. Then Simulated Annealing and Genetic Algorithms were chained so that one produced starting points for the other one. The idea was to use one algorithm to find a good starting point and that the following algorithm can use that point to perform even better, than it would normally 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

3.4 Pattern Recognition

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

4 RESULTS

This section shows the results of the experiments as they were described in the previous section.

4.1 Number of Samples

The pattern recognition accuracy between two sample data bases with a different number of samples varies a lot (Table 2). As it is visible the accuracy of the smaller sample base and the full sample base are very similar. The difference is only for the 5 data sample base significantly. 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.

Table 2: Evaluation of different data sample sizes

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

4.2 Starting Points

The pattern recognition accuracy depends on the selected starting points (Table 3). Calculation times depend on the selected parameters and vary. A maximum accuracy of 99.4 % was reached in the experiments. This value is higher than the value in Table 2 where the best value was 98 %. The random selected starting points and the randomness of the optimization algorithm are the cause of this effect. With a larger population and more generations it is possible to reduce that effect and get a better convergence. Still all starting points reached a very good accuracy.

Table 3: Evaluation of the influence of different starting points

Starting Population Number	Correctly Classified Samples	Calculation Time
1	98.6 %	3967 s
2	98.1 %	5596 s
3	98.3 %	5653 s
4	98.6 %	4643 s
5	99.4 %	4492 s
6	98.8 %	4352 s
7	98.6 %	4403 s
8	98.6 %	4638 s
9	98.9 %	4850 s
10	98.9 %	4568 s

4.3 Optimization Algorithm

The selection of the algorithm greatly influences the performance of the optimization as this section shows. Greedy Search and Simulated Annealing are more random and converge slower.

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 s.

Greedy Search

The best result of the Greedy Search algorithm was 97.7 % and the calculation time was only 1250 s. 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 s. This is unsurprisingly due to the fact that both algorithms evaluated the function 480 times. 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 s 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. However none of the experiments that were performed delivered the global maximum.

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.

4.4 Pattern Recognition

Table 4 shows accuracy of the different pattern recognition algorithms. To be able to use the Bayesian Network algorithm all values need to be discredited (an attribute can only have a pre-defined value). If numerical values are used, then Weka needs to discretize the attribute values automatically, which results in an “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).

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 attributes, however the Decision Tree algorithm seems to suffer and performs weak.

Table 4: Evaluation of different pattern recognition algorithms with optimization

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

In Table 5 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 s
SVM	98.9 %	1860 s
Bayesian Network	98.9 %	193 s

While Bayesian Networks deliver good results, they are not the best. The Table 5 also shows that the calculation time depends on the number of the blocks. 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.

5 CONCLUSIONS

The results show that an optimization can increase the performance of the signal analysis and pattern recognition. However the increase is less than 5 %. Still it is possible to push the accuracy up to 99.4 %. Even the short searches with a small population showed a good performance compared to choosing parameters by hand, which is nearly equal to chose a random parameter set. One goal of the project to reduce the amount of expert knowledge to use the system is achieved. 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.

6 REFERENCES

References

[1] GERDES, Mike; SCHOLZ, Dieter; RANDERATH, Bernhard: Reducing Delays Caused by Unscheduled Maintenance and Cabin Reconfiguration. In: ESTORF, Otto von; THIELECKE, Frank (Hrsg.): 2nd International Workshop on Aircraft System Technologies, AST 2009 (TUHH, Hamburg, 26./27. März 2009). Aachen : Shaker, 2009, S. 109 - 119

[2] GERDES, Mike; SCHOLZ, Dieter: Feature Extraction and Sensor Optimization for Condition Monitoring of Recirculation Fans and Filters. In: DGLR: Deutscher Luft- und Raumfahrtkongress 2009 : Tagungsband - Ausgewählte Manuskripte (DLRK, Aachen, 01.-04. September 2009). - ISBN: 978-3-932182-63-4. Download: http://PAHMIR.ProfScholz.de

[3] RUSSEL, Stuart; NORVIG, Peter: Artificial Intelligence: A Modern Approach. New Jersey : Pearson Education, 2003.