# CHAPTER 2

# THESIS ASSUMPTIONS

Automatic condition monitoring of diesel engines can be regarded as a task, which can be divided into sub-tasks. Not all sub-tasks are to be described and investigated in details in this thesis. The aim of this chapter is to discus the main task, automatic condition monitoring on diesel engines, and the sub-tasks. The content of the discussion is to provide several proposals on how to solve the different tasks, and to bring reasonable arguments on why one proposal is better than another. At the end of the chapter an overview over the selected proposals is given, and this overview is then regarded as the final specification on how the automatic condition monitoring system has to be implemented during the project.

## 2.1 MAN B&W Diesel's Research Engine, Copenhagen

The test set applied in this thesis is from a test run on the MAN B&W Diesel's Research Engine, Copenhagen. This is a four cylinder, 500 mm bore, 10,000 BHP diesel engine [24]. four changes are induced in the app. seven hours long test. The engine starts with a load at 25%, and after a while the oil lubrication system is shut off. After a certain time the load is changed from 25% to 50%, and then the load is increased to 75%. Next, the oil lubrication system is turned on again. This is the four induced changes, however, between the two last load enhancements something happens with the engine[1]. This change will be addressed as the unstable region, and will be regarded as an engine condition change together with the other engine condition changes.

## 2.2 Further discussion on sensors

In chapter 1 a short description of the sensors was given. Though the focus is not on the sensors in this thesis, it is worth giving more information about the topic, since some interesting considerations must be made. There is an inconsistency between the two statements: the acoustic emissions are "… waves with ultrasonic frequencies, i.e. from app. 100 kHz to 1 MHz", and that the sensor signals are sampled "… at a 20 kHz sample frequency". The sample theorem, [23] states that the sample frequency must be at least twice the maximum frequency of the signal, which has to be sampled. If the maximum frequency is 1 MHz, then the sample frequency should be at least 2 MHz. Chandroth et al., [7], recommends the sample freguency to be 3-4 times more than the highest frequency in the analogue signal. The argument is that the anti-aliasing filter is unable to exactly cut off all frequencies higher than the cut off frequency.

Fortunately, everything turns out to be good, because the analog AE sensor signals do not consist of the "raw" data, but the Root Mean Square (RMS)-value of the "raw" data. The AE sensor signals are measured in volts and from [22] the RMS-value of a voltage signal is given by,

[1] The technicians think that it probably is caused by engine load fluctuations.

$$V_{RMS} = \sqrt{\frac{1}{T}\int_{t_0}^{t_0+T} v(t)^2\, dt}\ , \qquad (2.1)$$

where $v(t)$ is the voltage signal (corresponds to the "raw" AE signal) and $V_{RMS}$ is the RMS-value of $v(t)$ in the time period $T$. The name Root Mean Square stems from the procedure that the voltage signal is first squared, then the mean value is calculated and finally the square root is taken. The AE-RMS signals are created by an analog approach, not a digital one. Figure 2.1 shows an AE-RMS signal from a single cycle.

Applying RMS signals instead of the "raw" data reduces the amount of input data significantly, since the sample frequency is significant lower (20 kHz), but there is a risk of removing significant information about the AE's. Therefore it could be interesting to apply the raw data to investigate how much information is removed when RMS signals are used. It could also be interesting to determine how the relation is between the engine condition and the AE spectrum, since different research projects have shown that AE's are generated when engine condition changes occur, e.g. evolution of small cracks [7], [10], [11], [12], [15], [16], [17], [18], [20], [21]. Perhaps some AE's related to a specific engine condition change are likely to exist in one frequency band, and other AE's related to other engine condition changes are likely to exist in other frequency bands.

### *2.2.1 Number and location of sensors*

Other considerations about the sensors are the number and the location of sensors on the engine. A compromise must be found when determining the number of sensors, since too few sensors are not capable of sensing all significant AE's. But on the contrary, too many sensors will cause a risk of providing too many data points, thereby increasing the input dimensionality, which in the end will suffer from the curse of dimensionality mentioned in chapter 1.

Since AE's declines dramatically with the distance from their generation point, it is necessary to investigate how near the generation point the sensors must be placed in order to sense the AE's. Not only the engine condition changes cause generation of AE's. Also events related to normal engine behavior generate AE's, and this makes it more difficult to implement the automatic condition monitoring system. However, there is a chance of isolating the wanted AE's from the not-wanted AE's, because the AE's only exists in a small area due to the dramatically declining with distance.

The "declining" property can also be used to detect the generation point in the sense that if more than one sensors are placed relatively near to each other, the nearer a sensor is to the generation point, the "clearer" is the AE. So if one sensor senses the AE very good, and the others do not, then the conclusion must be, that the AE was generated near the sensor that sensed the AE best. However, source location, i.e. location of the AE generation point, is not an easy task, since the AE propagation depends mainly on the material through which the AE propagates, [15], [17].

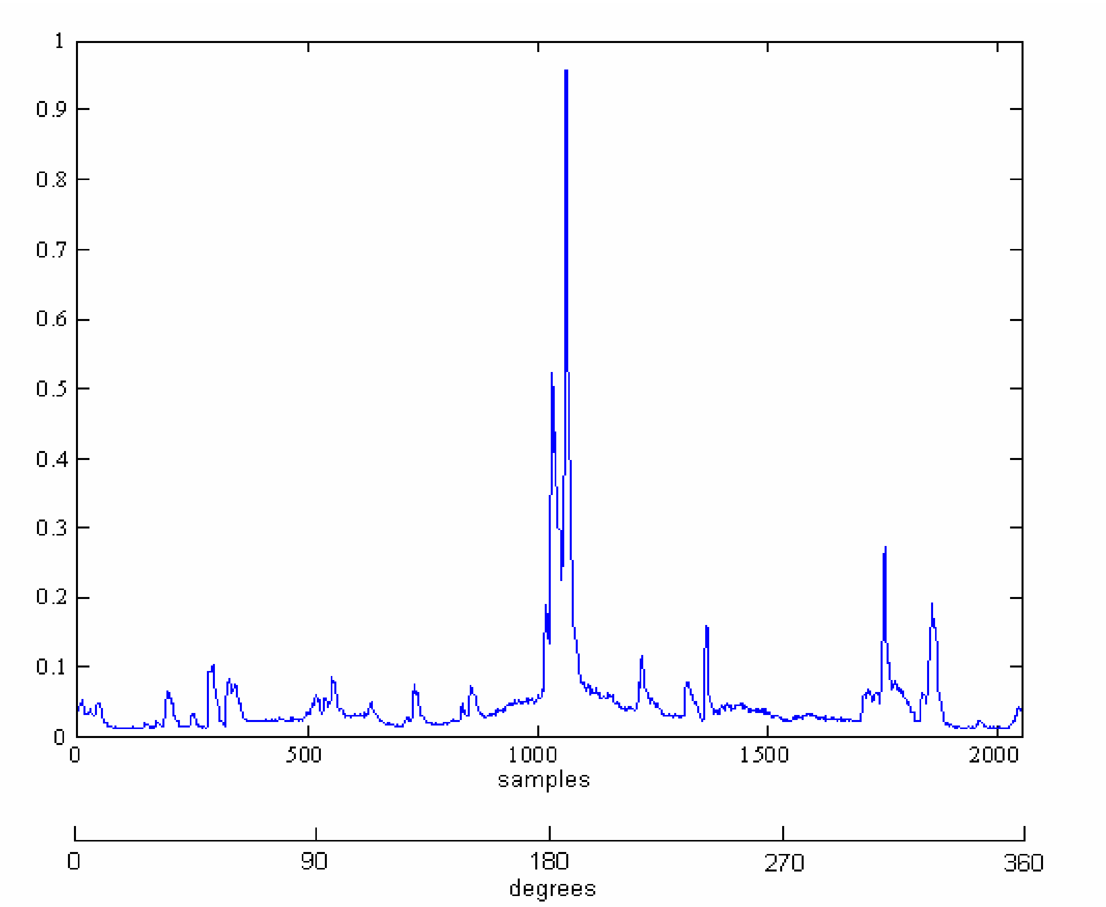


*Figure 2.2 Plot of a single AE-RMS cycle. The x-axis is given in both samples and degrees.*

This thesis will not consider anything about the sensors and AE's, i.e. choice of sensor type, number of sensors, placement of sensors, AE studies, etc. The only thing in the thesis, which is related to the topic is that the implementation of an automatic condition monitoring system is based on a test, where four AE-RMS are provided and crank angle domain coded so that each sensor provides 2,048 samples per engine cycle.

## 2.3 Pre-processing and feature extraction

The purpose with pre-processing and feature extraction is to prepare or modify the input data so it can be processed more efficiently. The litterature, [8], [9], [11], [13], [15], [16], [20] suggests a few methods, but unfortunately some of them are frequency based, and this is not appropriate for this work, since only RMS signals are provided. The A/D conversion of the analog sensor signals can be regarded as a pre-processing step, since the input data is modified to make it for processing in the PC. However, the A/D conversion is not enough, since the dimension of the digital sensor signals is too high, as discussed in the previous sections. The high dimension is just one obstacle among others. Also noise in the sensor signals can be present and give rise to trouble, so this must also be considered and solved.

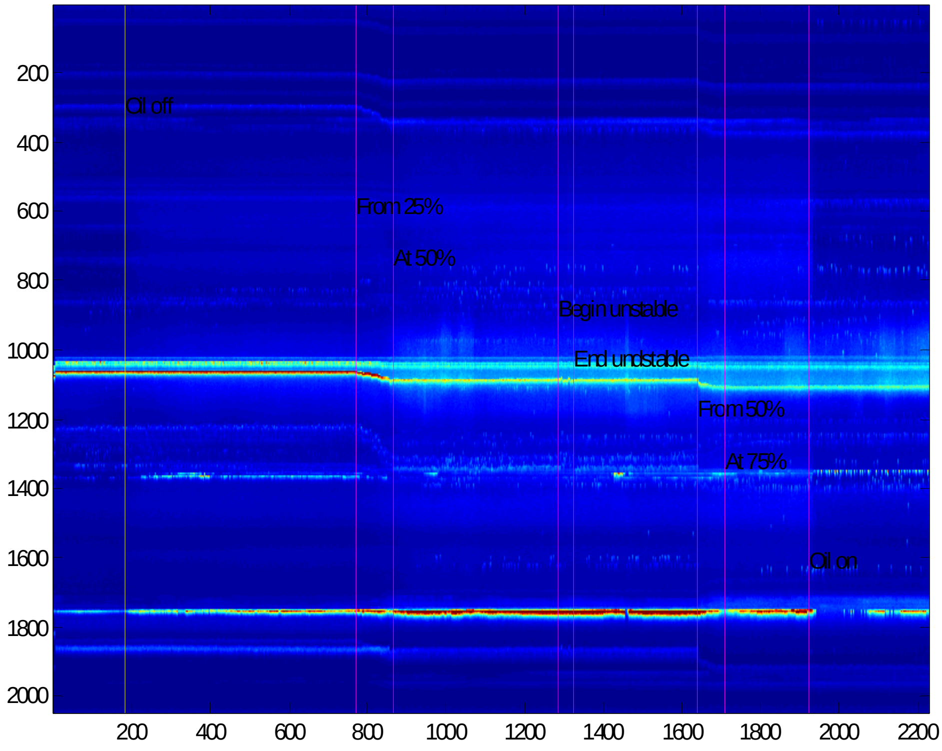


*Figure 2.2 Image of all 2,227 cycles from the test applied in the thesis. The cycles are on the x-axis and the cycle samples are on the y-axis. The more red a sample is, the larger is its magnitude. The more blue a sample is, the smaller is its magnitude. The vertical lines show the cycles where different experiments (engine condition changes) are induced.*

### *2.3.1 Noise*

The most obvious noise source is background noise coming from the diesel engine. It is common knowledge that engines make a lot of noise, which can be heard by the human ear. This type of noise is usually low noise, and by applying a high-pass filter on the raw data, all the AE's will not be filtered away, since they consist of high frequencies, but the engine background (low) noise will be. Other noise sources are reflections from AE's propagating through materials, and interference between different AE's. These two noise sources depend strongly on the geometry of the material through which the AE's propagate, and of the material itself. The literature states that it s almost impossible to model the propagation of the AE's and the interference between different AE's due to the dependency of material and material geometry, [15], [17].

Noise is not considered a primary objective in this thesis. This is mainly because the data are RMS-values and not raw data. Thus studying the frequency band of the AE's is not feasible. This is also the case with a study of the AE propagation through the engine. A simple noise test is performed a selected experiments. The idea is to apply white additive Gaussian noise on the AE-RMS signals with different Signal to Noise Ratios (SNR).

*2.3.2 The mean value and the variance as features*

If the normal condition of the engine is changed, e.g. if a small crack is under evolution, then it is reasonable to believe that the number of AE's and the magnitude of the AE's increase in the new condition. This will give a rise in the RMS-value of the sensor signal. In chapter 3 it is shown that this also give a rise in the mean value of a *cycle* sensor signal, i.e. the mean value of the 2,048 samples per engine revolution. But one has to be careful with a large input dimension reduction (2,048 to 1), since the risk of removing to much signal information is high.

In chapter 3 it is also shown that the variance of a cycle signal can be used as a feature. An explanation on this could be that certain events happen at different positions of the engine cycle. E.g. the combustion of the diesel takes place in one area of the cycle and the exhaustion takes place in another area of the cycle. If a small crack is under evolution in the combustion chamber, then it is likely that the number and the magnitude of the AE's generated by the small crack will be increased, thereby causing the AE-RMS signal to increase at this position of the engine cycle. If everything else works okay, then the variance of the cycle must increase.

*2.3.3 Principal Component Analysis*

There exist a number of approaches to provide features than retrieving the mean value and the variance of the cycles. One of them is Principal Component Analysis (PCA). This is a classical method, which reduces the input dimension by projecting the input data onto a coordinate system with smaller dimension. PCA is more complex than estimating the mean value and the variance of the cycles, but it makes it easier to control the amount of information in the input data that is removed in order to reduce the input dimension. It has been chosen to perform PCA in this work, since the two simple approaches (estimating the mean value and the variance) turned out to be incapable of detecting a very important change in the engine condition[1].

Only the three mentioned pre-processing and feature extraction methods are applied in this thesis. It is likely that other methods will provide better features, but this is not regarded as the primary objective. The motivation behind this choice is that focus is on how to detect and estimate changes in a diesel engine by looking at the changes in one or more feature signals. If an automatic condition monitoring system has been implemented and a better pre-processing and feature extraction method is applied, thus providing better feature signals, then the system will perform better. Therefore the primary objective is to implement a system that is capable of detecting and estimating changes in a given feature signal and secondary.

## 2.4 Segmentation

Figure 2.2 shows the changes in the engine as a function of time. The test providing the image is a controlled test, i.e. the technicians induce the engine condition changes at specific time instants, thereby making them able to segment the data precisely and the vertical lines in the figure show this. The primary objective with the automatic segmentation task is to segment the data in the sense that these segments are quite similar to the segments given by the technician.

[1] The unstable region – see chapter 3.

A lot of considerations have to be done when solving segmentation tasks. First of all, if the engine leaves its normal condition, should the automatic condition monitoring system also determine when the engine enters a new condition? This might be a good idea since not all engine condition changes are related to engine errors. E.g. when the load is increased, the engine condition changes, which can be seen in figure 2.2. If the system is not reset to the new condition, then the monitoring task will probably fail, since the features of the input data might have undergone some drastic changes. Therefore it is necessary to detect when the engine enters the new condition. And since the engine is likely to change several times in a day, during normal engine behavior, a need for an automatic detection of the new condition is preferred in stead of having a human being detecting the new conditions.

Another consideration about the segmentation task is whether engine condition changes related to normal engine behavior must be detected or not. As mentioned in the previous section, several normal behavior changes occur, and if the crew are to be alerted every time e.g. the load is changed, then the number of alarms is unnecessarily high. It is a very good idea to keep the number of alarms on a true level, i.e. an alarm must only be related to a critical engine condition change. This way the crew will not become “immune” to the alarms, and they will probably respond fast when an alarm is set, since they know it is critical.

One approach to keep the number of alarms related to normal engine condition changes down, is to use information from the ship control panel, and combine it with the automatic condition monitoring system. When the load is changed, information is sent to the condition system, and when a change in the engine condition is detected, the system knows that this change is likely to correspond to the load change. Therefore, probably no alarm is set.

When considering the segmentation task it is necessary to determine how fast the change detection system should be. When do the crew have to be alerted? At the exact moment when the alarm occurs or 10 minutes after? Or a day after? This is very important, since very complex and very simple change detection systems can be developed. If there is a lot of time to assess whether an engine condition change is present or not and to locate the change point(s), then perhaps a complex system will be preferred. On the other hand, if the amount of time is short, then a simple system is needed, since they are usually fast compared to the complex systems.

*2.4.1 Splitting the segmentation task in parts*

Basseville et al. [1] proposes a change detection system, which consists of three parts, an on-line change detection algorithm, an off-line hypothesis test and an off-line change point estimation. The system proposal is shown in figure 2.3. First of all, feature signals are generated by a chosen method. Then a fast, but not necessarily precise, on-line change detection is performed. On-line means in this case that an assessment on whether a change has occurred or not is given for every cycle. The on-line change detection does not have to set an alarm exactly at the time of the engine condition change – its primary objective is just to detect that “perhaps something is wrong”.

When the on-line change detection part has detected a change, then an off-line hypothesis test is created. By means of the alarm(s) given by the on-line change detection part, a window of cycles is selected. In this window there might be a change – this is what the on-line change

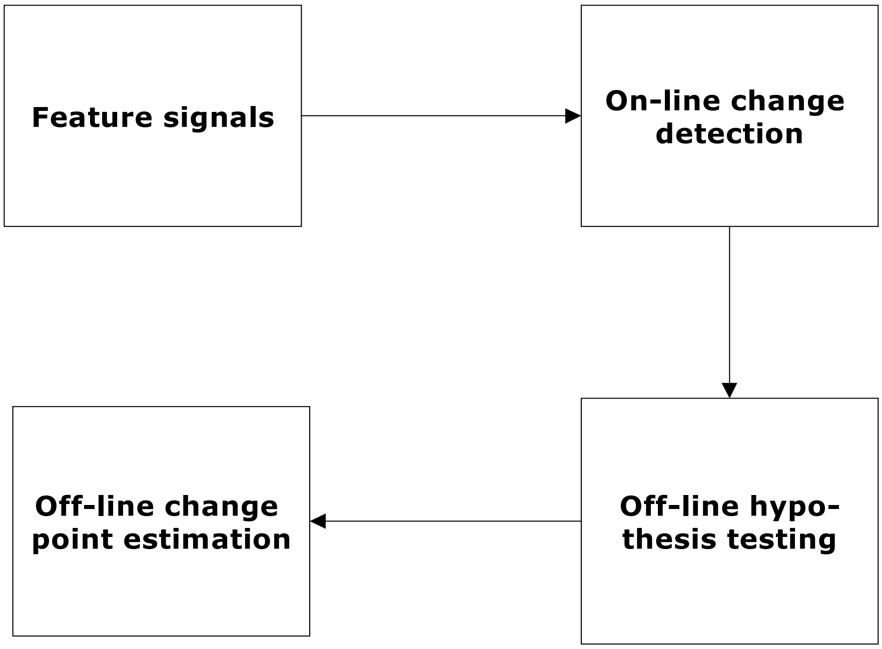


*Figure 2.3: Proposal on a change detection system.*

detection has concluded. Next, the hypothesis: “A change has occurred” is tested. The outcome of the test is that the window of cycles includes a change or not. The idea of an off-line hypothesis test part is that it should be more complex than the simple and fast on-line change detection in order to make it more reliable whether a change has occurred or not.

If the off-line hypothesis test is a success, then it could be interesting to determine the exact time of engine condition change. This will most likely make the classification task easier, but it will also give the crew better opportunities when they act on the engine condition change. Thus a final off-line part is activated with the goal of estimating the change point more precise than the on-line part did. Again this off-line part is usually more complex than the on-line part, so this is the basic argument of choosing this type of change detection system.

One could also abandon the idea of splitting the change detection system in parts and just have a single part, which handles all the change detection tasks. However, this can be very complex, since then you must have a system that not only detects the change point, but also estimates the change point very precisely and of course gives very reliable alarms. However, if time is not a problem, this could be a proper solution.

In this thesis the system shown in figure 2.3 is chosen. The reason is that it is intended to make the system able to alert the crew very fast, but also to have a very efficient system that adapts the calculation power to the situation. I.e., when no change occurs, only few calculations are done, and when a change has occurred, many calculations are done in order to make reliable change detection. However, the choice of a system depends on the criteria given by the client, that is the shipping company. Since no criteria are given here, there is a free choice.

## 2.5 Evaluating change detection systems

A very relevant question is how to evaluate a change detection system and compare it with other change detection systems. There are several properties, which can be investigated, and these are described in this section. In the following a decision function y($k$) of a feature signal is given, where $k$ is the index of the observations, i.e. cycles. The alarm time denoted $t_a$ is the sample at which the change in condition occurs. The alarm time given by a change detection system is denoted as $t_{a,sys}$. Definitions of four types of alarms is given:

- *False alarms* are all the alarms given by a system at time indices when no change has occurred ( $t_{a,sys} < t_a$ ).

- *True alarms* are all the alarms given by a system at time indices from when a change has occurred to a few samples ($t_{a\_True}$) after ($t_a <= t_{a,sys} < t_a + t_{a\_True}$). The size of $t_{a\_True}$ depends on the situation. One could define the true alarms for $t_{a\_True} = 0$, but this is unrealistic since a large number of change detection algorithms use a number of samples to state whether an alarm must be set or not.

- *Late alarms* are all the alarms given by a system at time indices from $t_{a\_True}$ samples *after* the change has occurred until the last, but one sample of the feature signal.

- *Unseen alarms* are all the alarms given by a system at time indices equal to the last sample of the feature signal. One could argue that some late alarms close to the last sample of the feature signal also should be regarded as unseen alarms, but again this is dependent on the situation

When several tests are performed, the number of false, true, late and unseen alarms is incremented every time the system detects a change.

In figure 2.4, which shows the definitions and notations mentioned above, the length of the decision function is denoted $N$. In the rest of the report the feature signals are mostly denoted by $y$, but in this section the notation is applied for a decision function. This should not confuse the reader, since it is clear when there is talking about a feature signal or a decision function. A feature signal is the signal provided by the mentioned pre-processing and feature extraction task, which reduces the dimensionality of the input data set. A decision function is a function, which is created from the feature signal, and on which a specific threshold is defined. If the decision function at some time instant exceeds the threshold, an alarm is set. In some cases the feature signals also perform as decision functions, but this is not the case in this thesis.

### *2.5.1 Receiver operating characteristic curves*

Another method to evaluate change detection systems is based on Receiver Operating Characteristic curves, or simply ROC curves. This is a classical method and the one used in [24]. The idea is to sweep the threshold from the minimum to the maximum of the threshold and detecting the first time instant (or cycle) $t_{a,sys}$ where the decision function exceeds the threshold. The decision function is divided in two regions, one *before* $t_a$ (the false region) and one *after* $t_a$ (the true region). Then two rates are defined:

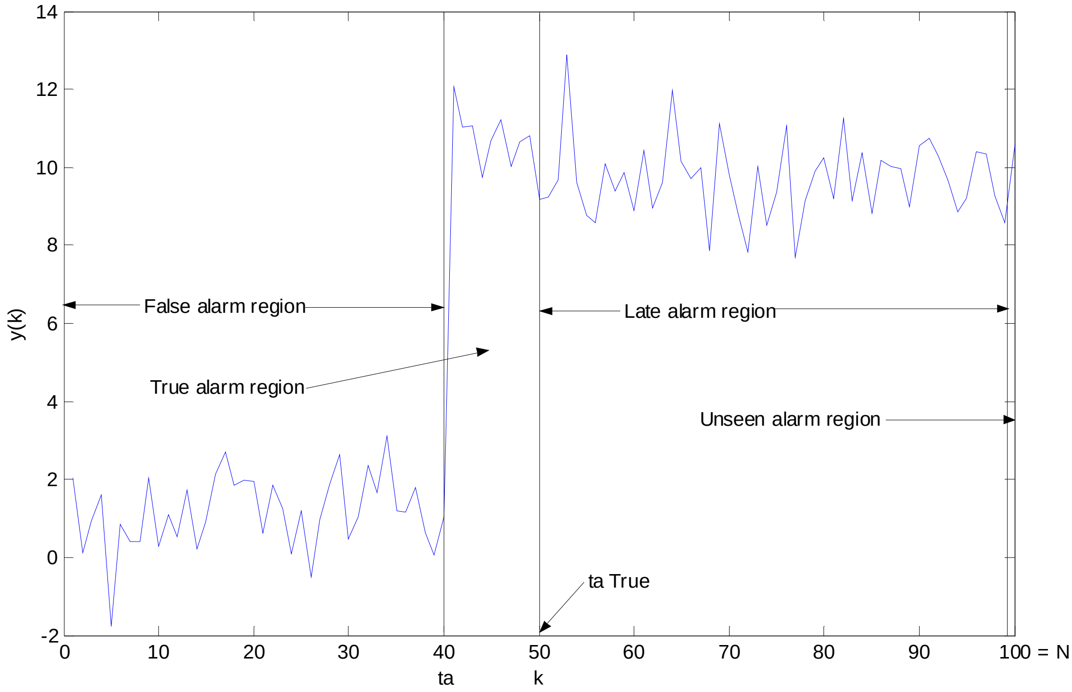


*Figure 2.4: Definition of alarms and signal notations.*

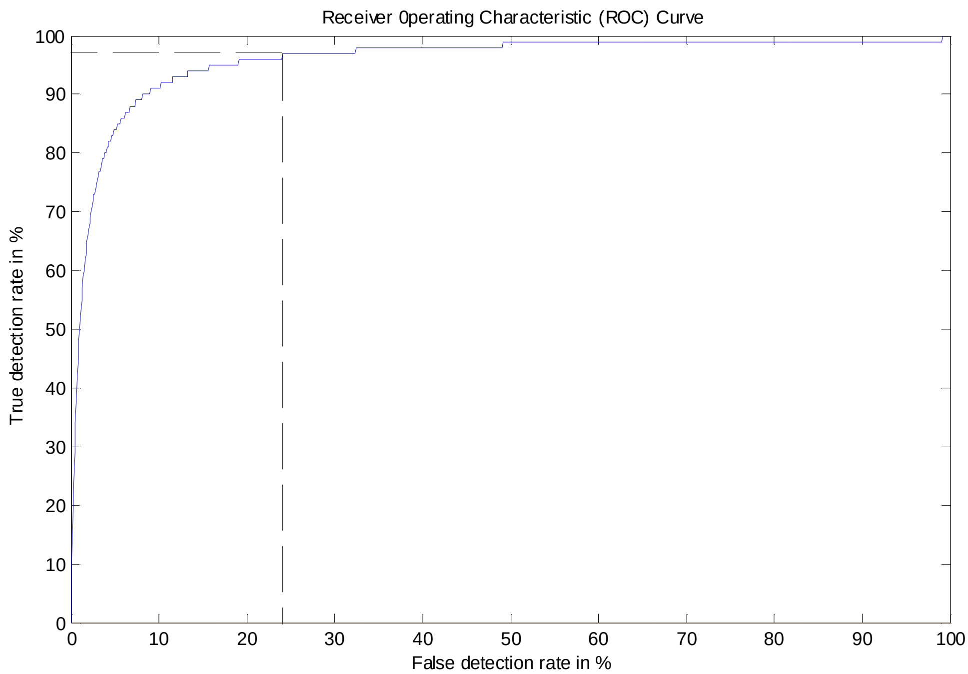


*Figure 2.5: Example of a ROC curve. At the chosen threshold the false detection rate is 24% and the true detection rate is app. 98%.*

- True detection rate: The number of cycles in the *true* region after $t_{a,sys}$ divided by the total number of cycles in the *true* region.
- False detection rate: The number of cycles in the *false* region after $t_{a,sys}$ divided by the total number of cycles in the *false* region.

An example of a ROC curve is given in figure 2.5. The ideal change detection system will be placed in the upper left corner with the coordinates (0%; 100%). This corresponds to the true change point $t_a$ has been detected exactly, i.e. $t_{a,sys} = t_a$. The worst performance of a change detection system is given in the lower left corner (0%; 0%), where no alarm is set by the system, i.e. $t_{a,sys} = N$. This means that the system is not capable of detecting the change. In other words: Unseen alarms are present because the threshold is never exceeded. There is also a second worst system performance corresponding to the upper right corner. Here the threshold is always exceeded causing the system to give alarms constantly. The fourth and last corner in the ROC plot corresponds to an unrealistic situation, where the true detection rate is 0% and the false detection rate is 100%. Optimization of the system by means of a ROC curve is done, by maximizing the area under the ROC curve.

The ROC curves are based on the *confusion matrix*, which is shown in figure 2.6. The confusion matrix elements are here binary, but they can easily be converted to e.g. ratios (%) as is the case in figure 2.5. Consider the event *True* that a change has occurred, and the event *False*, no change has occurred. Then the four elements are defined as:

- *a*: When a change occurs, the system detects the change, and when no change occurs, the system detects no change.
- *b*: When a change occurs, the system detects the change, and when no change occurs, the system detects a change.
- *c*: When a change occurs, the system does not detect the change, and when no change occurs, the system detects no change.
- *d*: When a change occurs, the system does not detect the change, and when no change occurs, the system detects a change.

In this work, ROC curves are used as the method to evaluate the change detection system, since it seems to be the method applied in reality. In this manner the change detection system can be compared with other change detection systems.

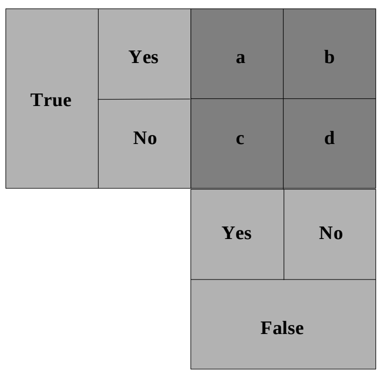


*Figure 2.6: The confusion matrix.*

# 2.6 Re-sampling

*2.6.1 Enlarging the AE-RMS data set*

When a method or an algorithm has to be evaluated it is standard procedure to set up a test and repeat it a proper number of times. Then an outcome could be: "In 90% of the cases the method acts as expected", or "The algorithm detects only 50% of the change points." In this thesis only a single test has been performed as described in section 2.1. If nothing further is done, it will not end up with outcomes like the before mentioned. This could be a problem since you would have no idea how the methods work in general.

One solution is to repeat the test 10, 20 or 100 times[1], thus statistics can be applied to evaluate the methods, but this is a rather time consuming solution because a single test last app. seven hours. Therefore another approach, re-sampling, was investigated. The basic principle is to create a synthetic data set from the original data set by choosing samples from the original data set in different ways. Five methods have been developed and investigated and they are described in the following sections.

*2.6.2 The original data set, experiment no. 1*

The starting point of all five re-sampling methods is experiment no. 1 acquired from AE sensor no. 1. Two types of plots are used to explain the re-sampling methods. The first is the time plot of the AE-RMS signal and the second is the feature signal created by calculating the mean of the cycles in the AE-RMS signal. Figure 2.7-9 show scaled images of the cycle evolution during time. Figure 2.7 is the full AE-RMS signal. Figure 2.8 and 2.9 are zoomed images in the intervals [0; 0.2] and [0.2; 1], respectively.

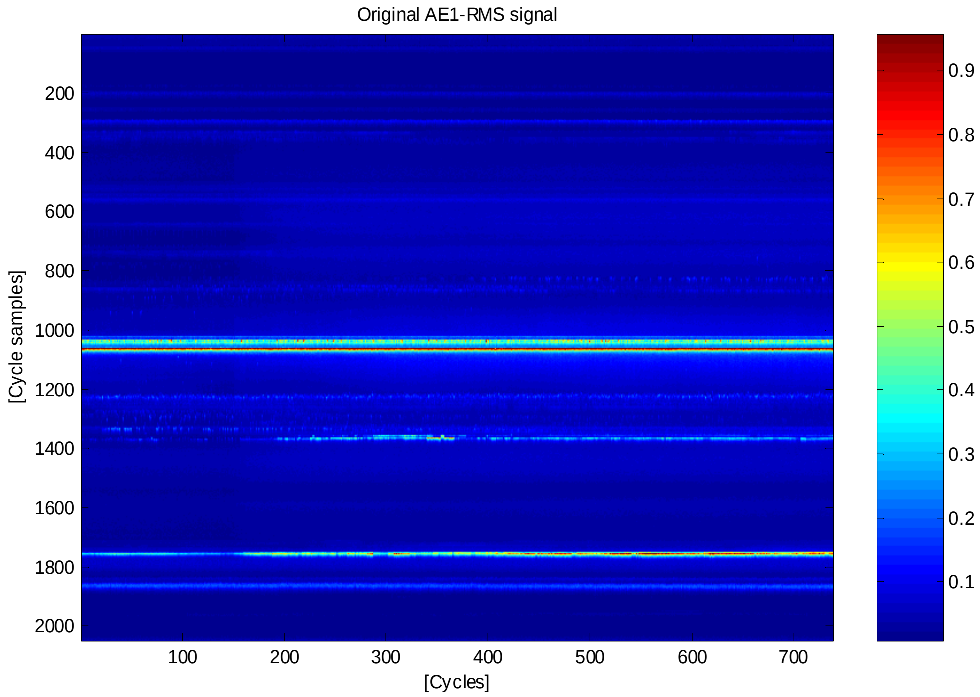


*Figure 2.7: Scaled image of the full AE-RMS signal for experiment no. 1.*

[1] Central limit theorem states that a test must be repeated more than 30 times in order to use statistics.

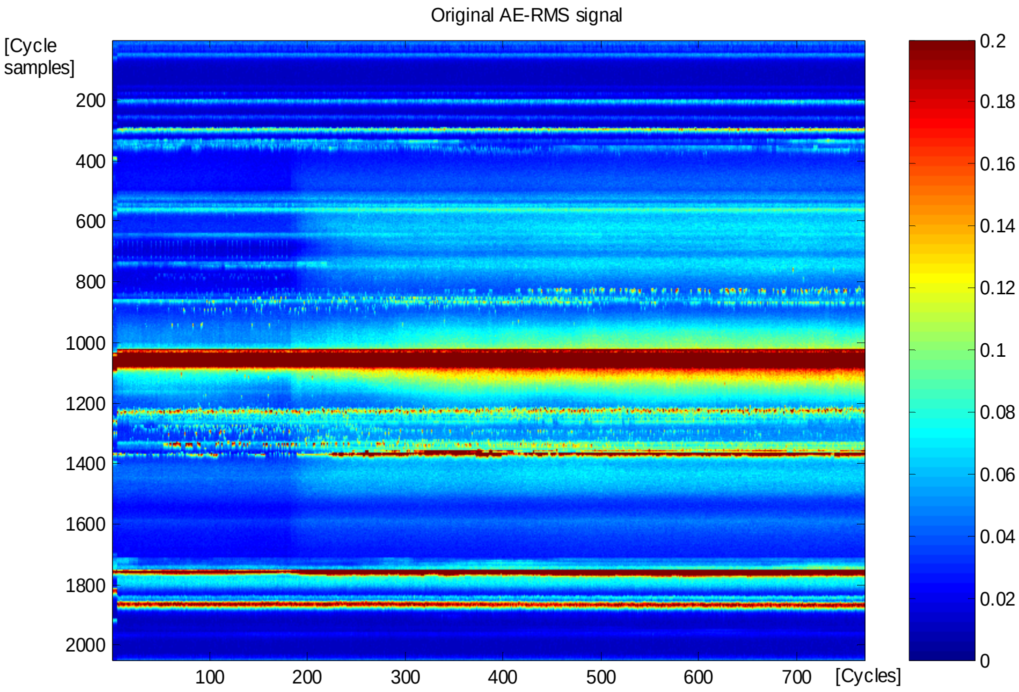


*Figure 2.8: Scaled image of the AE-RMS signal, sensor no. 1 for experiment no. 1 in the interval [0; 0.2].*

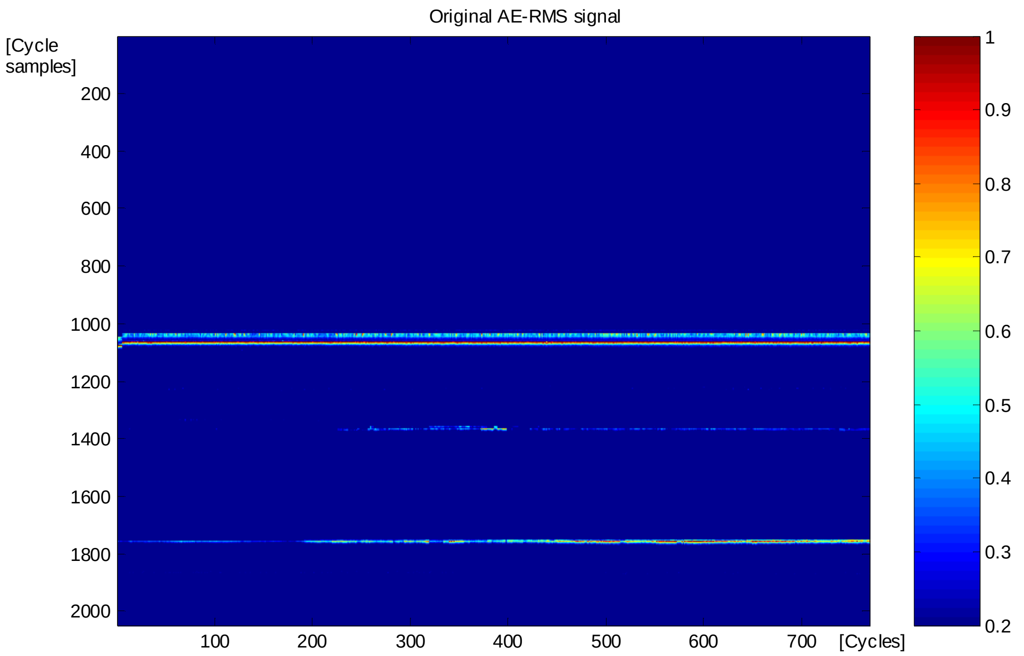


*Figure 2.9: Scaled image of the AE-RMS signal for experiment no. 1 in the interval [0.2; 1]*

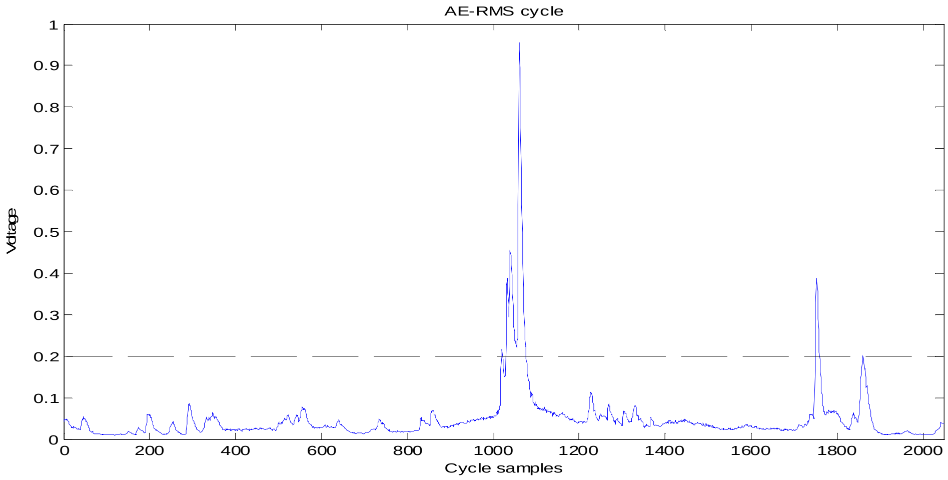


*Figure 2.10: The small peaks are separated from the large peaks by the limit at 0.2*

The reason behind the zoom plots is that at first sight the intuitive idea is, that the peaks in the AE-RMS signal play the greatest role due to the change detection issue - they are the largest, and thereby the most visible, in the AE-RMS cycle time plot. In fact it is possible to detect all five changes in the full test by visual inspection of the largest peaks only. The small peaks are very small compared to the large peaks, so they must be separated from each other. Looking at a single cycle, measuring the interval of the small peaks and the interval of the large peaks does this. Figure 2.10 shows this procedure. Notice that the cycle samples are not given in degrees, but in 2,048 samples per cycle. In appendix A, figures similar to figure 2.7-9 are given for the remaining sensors, but with other limits, which are appropriate for these sensors.

### *2.6.3 Re-sampling, method no. 1*

This method is the simplest one. The first change point is denoted $cp_1$ and the second change point $cp_2$. The re-sampled and the original AE-RMS signals are divided into two regions, one from cycle 0 to cycle $cp_1-1$ and one from cycle $cp_1$ to the end of the experiment, which is the cycle where the load is increased from 25% to 50%. Next, cycles are chosen randomly from the original AE-RMS signal in the first region and put in the first region of the re-sampled AE-RMS signal one by one until this region is filled with new cycles. For region two the same procedure is done. The pseudo code is shown below.

*Pseudo code for re-sampling method no. 1*

```
% Region 1:
for i=0:cp1-1
    % Calculate index of cycle in region 1 of original data:
    index = random number;
    % Put this cycle in the re-sampled data:
    RESAMP_DATA(i) = ORIG_DATA(index);
End

% Region 2:
for i=cp1:end of experiment
    % Calculate index of cycle in region 2 of original data:
    index = random number;
    % Put this cycle in the re-sampled data:
    RESAMP_DATA(i) = ORIG_DATA(index);
end
```

An important detail about how the cycles are chosen randomly is worth mentioning here. The *rand*() function in Matlab returns pseudo random numbers in the interval [0 1]. *rand*() uses different generators to produce sequences of pseudo random numbers, and these generators can be chosen by selecting a state of *rand*(). Every time *rand*() is reset to a specific state it will generate the same number of pseudo random numbers. Thus, when a new engine test is simulated, a new state of *rand*() is chosen and the new cycles are selected by means of this sequence of pseudo random numbers. Whenever you want to work with this particular re-sampled test, you reset *rand*() to the corresponding test and re-sample with the same sequence of pseudo random numbers. Clearly this approach has a great advantage in that when a new method must be tested, then you can test and compare the methods on the exact same signals.

Figure 2.11 shows the re-sampled AE-RMS signal for sensor no. 1, experiment no. 1 when re-sampling method no. 1 is used. The engine condition change related to the oil lubrication being shut off is far more abrupt than in the original case. This is also seen in the following figure, which is a subplot of the original feature signal and the re-sampled feature signal.

The DMD change detection algorithm is applied on the feature signals in figure 2.12. If we look at the original feature signal, then it is clear, that the signal can be split into three parts, the normal condition $H_0$, the drift from the normal condition to the new condition $H_0 \rightarrow H_1$, and the new condition $H_1$. The cycle at which the oil is turned off is called change point 1, $cp_1$, and the cycle where the new condition seems to be steady is defined as change point 2, $cp_2$. From the same figure it is also clear that $H_0$ seems to have constant empirical deviation, but slightly variable empirical mean. In $H_1$ the same observations are done, though one could

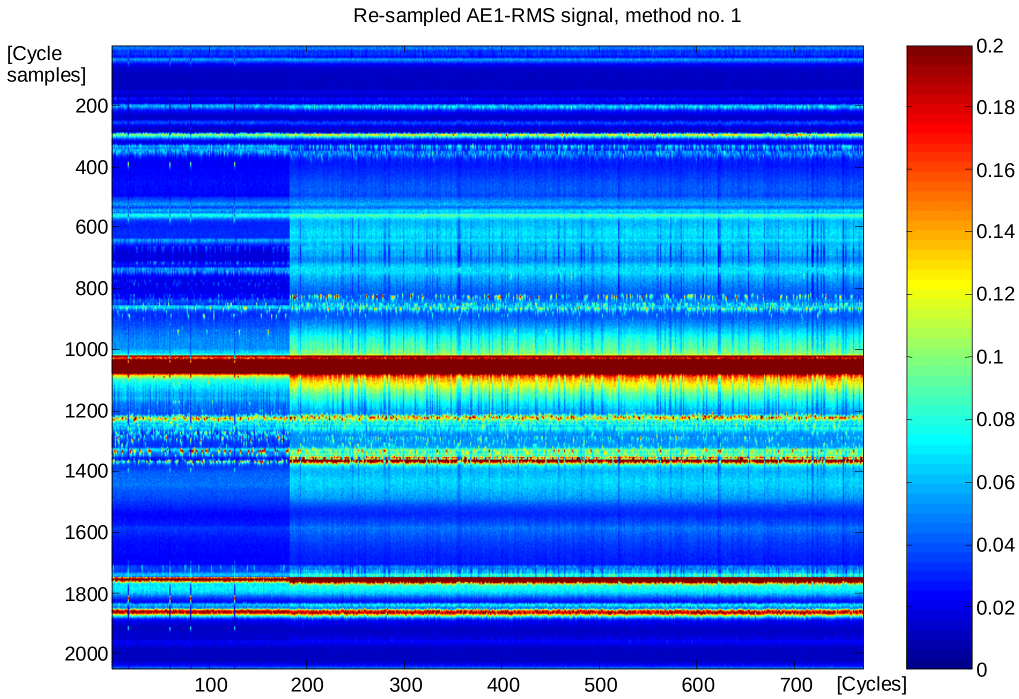


*Figure 2.11: Time plot of the re-sampled AE-RMS cycles of experiment no. 1.*

argue that the empirical mean increases, but this does not play an important role in this discussion.

The drift is relatively steep in the beginning but fades out after about 50 cycles. By knowledge of the DMD change detection algorithm it is certain that the algorithm is capable of detecting the change in the feature signal related to the event of the oil being turned off[1]. However, it will set an alarm several samples after $cp_1$ because of the long drift. Inspection of the re-sampled feature signal in figure 2.12 a) reveals at least three significant observations:

- The drift $H_0 \rightarrow H_1$ is more abrupt.
- The two conditions $H_0$ and $H_1$ are steadier.
- The empirical deviation of condition $H_1$ is larger.

The DMD change detection algorithm has less difficulty detecting the change in the re-sampled feature than the change in the original feature signal because of observation no. 1 and no. 2. The ratio is poorer in the re-sampled feature signal, but this will have no practical influence on the change detection performed by the DMD algorithm, since the ratio is still good enough. Thus it is postulated that if re-sampling method no. 1 is used to enlarge the AE-RMS data set, the re-sampled data set is too far away from the original data and easier to detect, so it will cause wrong results.

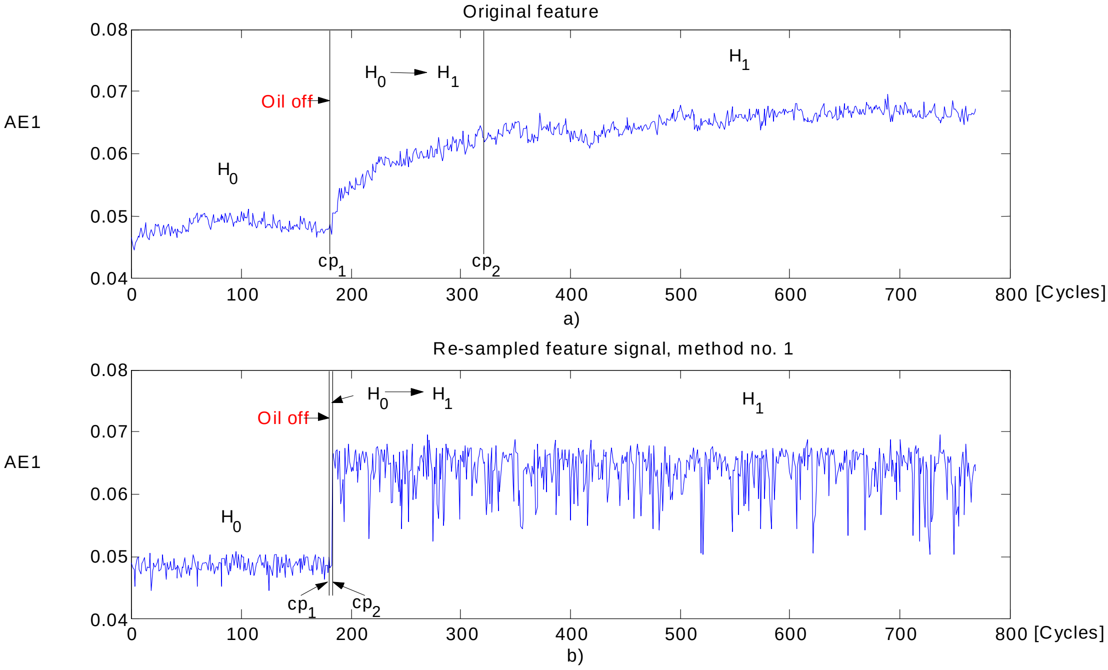


*Figure 2.12: Time plot of a) the original vs. b) the re-sampled AE-RMS feature signals of experiment no. 1 using re-sampling method no. 1. The feature signal is the mean value of sensor no. 1.*

[1] This is because the ratio of $H_0$ and $H_1$ is good.

*2.6.4 Re-sampling, method no. 2*

One way to make the engine condition change less abrupt than in the first re-sampling method is to move the beginning of the new condition to a point, where the engine condition seems to be steady. This is reasonable since a lot of cycles from the drift are being used in the new re-sampled condition, thereby increasing the deviation of the new condition. This method is implemented as the second re-sampling method and shown in the figure 2.13-14.

From these figures it is clear that the deviation of the new condition indeed is decreased to a more realistic level according to the original data. However, the change is still abrupt since it has been shifted to the time where the new condition is assumed to be steady. Another important thing is that the true change and drift remain the same, thus the different tests are run on the same change and drift. This is definitely not an adequate solution. Therefore other solutions must be investigated.

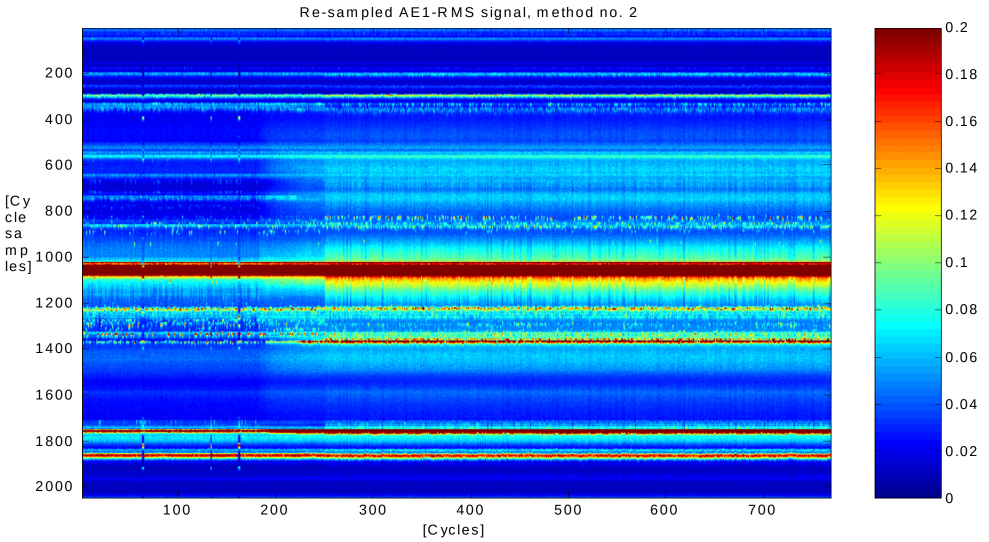


*Figure 2.13: Time plot of the re-sampled AE-RMS cycles of experiment no. 1 applying re-sampling method no. 2.*

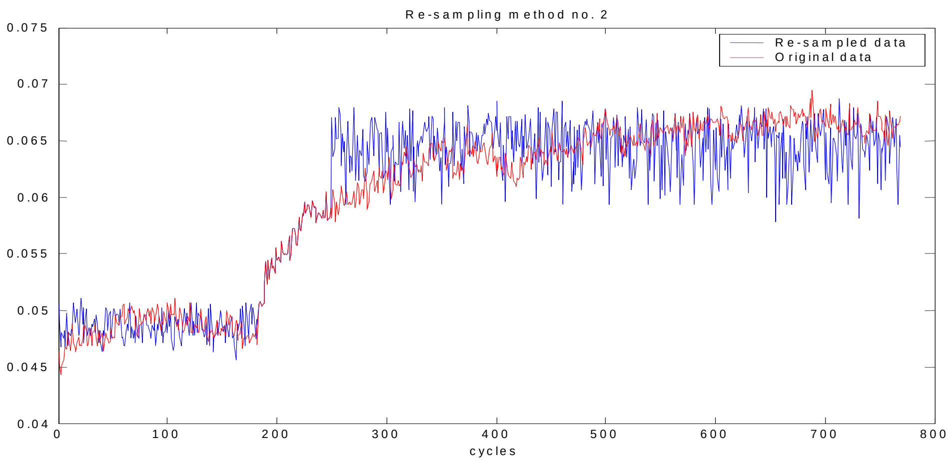


*Figure 2.14: Re-sampled feature signal method no. 2. The change is shifted, but remains abrupt. The true change is not re-sampled.*

*2.6.5 Re-sampling, method no. 3*

The reason why the two previous re-sampling methods did not work is that the original data has a non-abrupt change, and that the different condition can not be assumed to be steady. Though there is a distinct deviation in the conditions it also seems that the cycles are dependent on the recent cycles. This observation is investigated in the third re-sampling method in which the experiment is again divided into two regions, from the beginning of the experiment to the first change point, and from the first change point to the end of the experiment. Next, a window of size *N* is at first swept over the normal condition and then at the new condition. The new re-sampled cycles are then found by choosing randomly among the *N* recent cycles.

This windowing re-sample method is shown in figures 2.15-16. The conclusion is that the re-sampled signals look more like the original ones, than they did in the previous methods. However, there is an error in the code causing the *N* last cycles of the two conditions not to be re-sampled. This is an error that could be handled very easily, but nothing was done about it since another observation reveals that the method is not good enough[1]. The re-sampled signal is too similar to the original one since it almost exactly follows the original one. One way to overcome this problem is to increase the size of the window. Doing this will on the other hand increase the empirical deviation of the conditions and smooth the conditions.

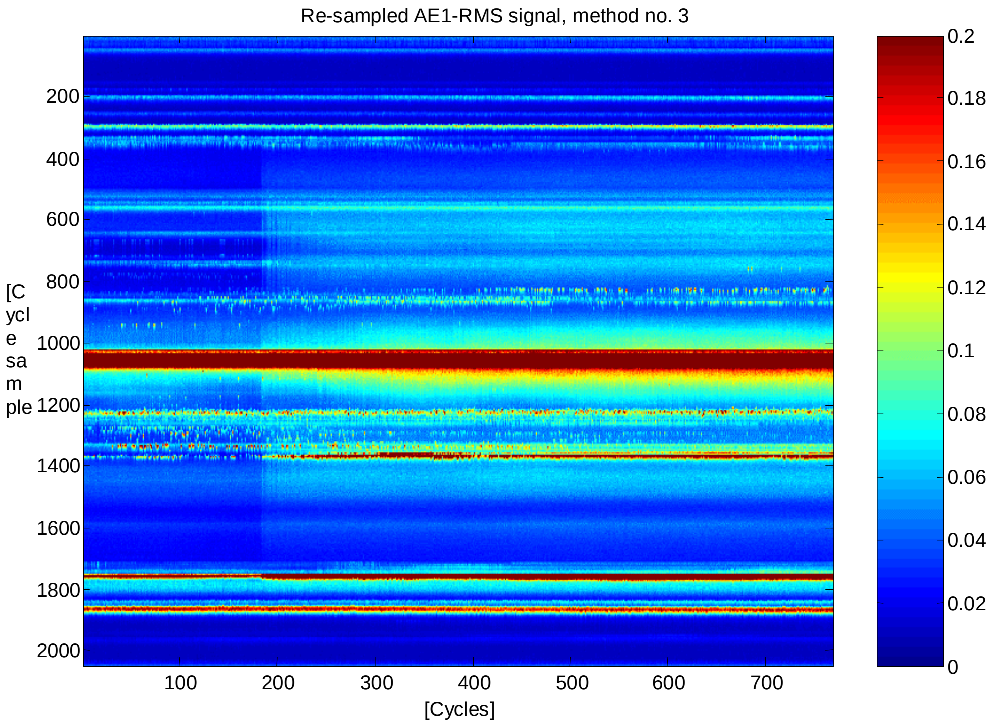


*Figure 2.15: Time plot of the re-sampled AE-RMS cycles of experiment no. 1 applying re-sampling method no. 3.*

[1] During the end of the project it was observed that the re-sampled cycle was chosen not among the *N recently* cycles but the *N next* cycles.

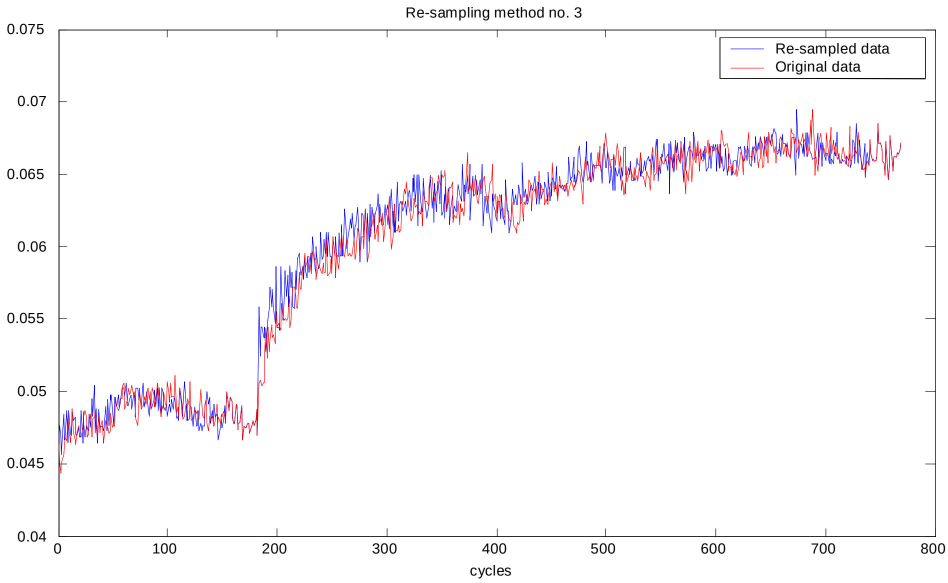


*Figure 2.16: Re-sampled feature signal method no. 3.*

### *2.6.6 Re-sampling, method no. 4*

The fourth re-sampling method is almost the same as the third method, and shown in figure 2.17-18. The difference is that the experiment is not divided in two regions. The window is moved over the whole experiment, including the change point and the drift. There is still an almost too good similarity between the original data and the re-sampled, but this was expected due to the almost identical methods. One noticeable difference is worth mentioning. The re-sampled drift is less abrupt and has an increased empirical deviation. This will indeed make the change detection task more difficult, but again – the similarity between the original and the re-sampled data in the two conditions is too high to give useful results.

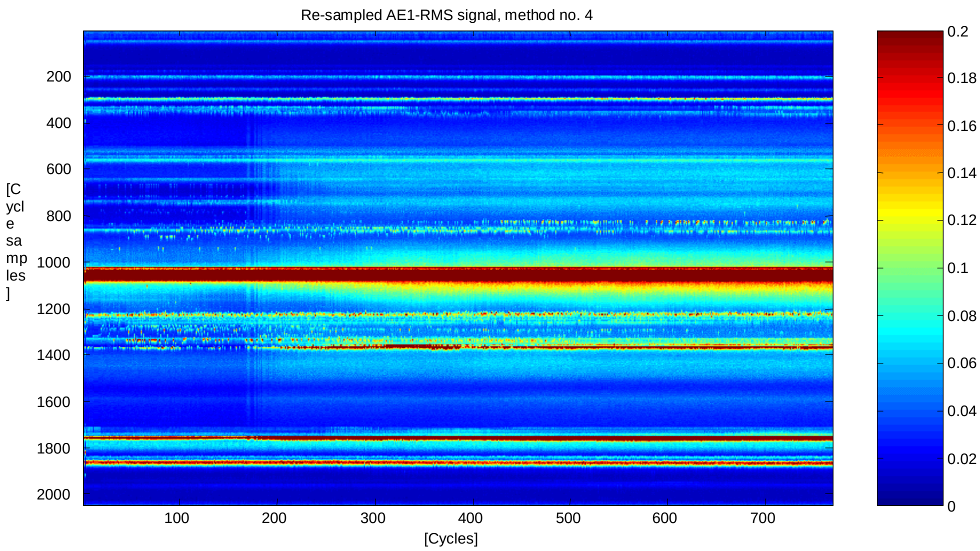


*Figure 2.17: Time plot of the re-sampled AE-RMS cycles of experiment no. 1 applying re-sampling method no. 4.*

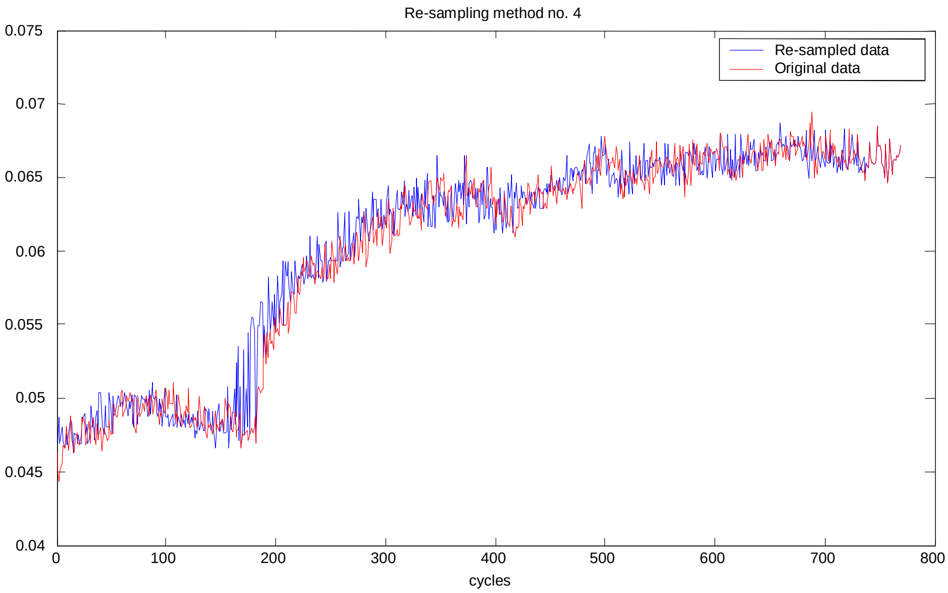


*Figure 2.18: Re-sampled feature signal method no. 4.*

*2.6.7 Re-sampling, method no. 5*

The fifth and final re-sampling method is also the re-sampled method applied in the statistical tests. It differs significantly from the other methods by using information from both conditions to create a new cycle[1]. As mentioned before, the conditions are not steady and the cycles are slightly dependent on the recent cycles. In this method this dependency is implemented by taking the empirical mean value of the feature signal. Then the empirical mean value is scaled in order to put it in the interval [0; 1]. Now this “probability”-signal is used to create the new re-sampled cycles by the equation,

$$cycle_{re}(c) = (1 - \alpha(c)) \cdot rand_{H0} + \alpha(c) \cdot rand_{H1}, \tag{2.2}$$

where $cycle_{re}$ is the re-sampled cycle, $c$ is index of the cycles, $\alpha(c)$ is the probability-signal and $rand_{H0}$ and $rand_{H1}$ are randomly chosen cycles in the normal condition and the new condition, respectively.

The results of this re-sampling method are given in the figures 2.19-21. It is observed that the empirical deviations are again increased, but not as smoothened as in the past methods. Also the fluctuations in the conditions are present, but they are almost buried in the increased deviation. The drift is not abrupt, but shifted a few cycles to the right.

In this method, other probability signals (or mixer functions) can be used, but this is not done here. However, one method must be chosen, and this last method was chosen among all described methods, due to its properties described above.

[1] Method no. 4 exploits in fact also information from both conditions, but this amount is small compared to re-sampling method no. 5, since it was only applied in the region around the drift.

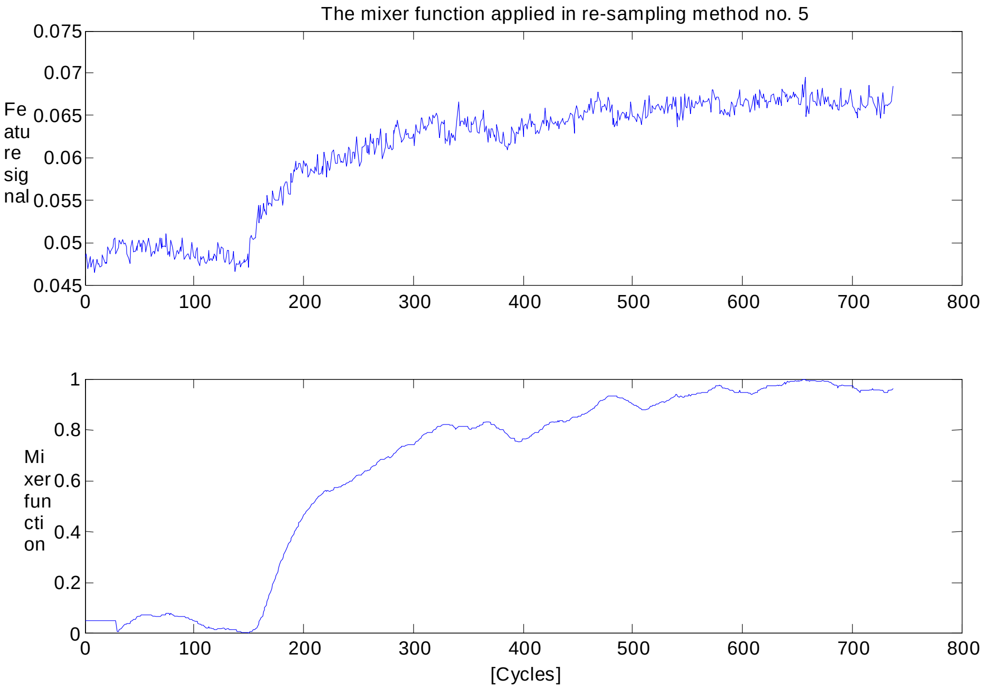


*Figure 2.19: The feature signal and the scaled mixer function.*

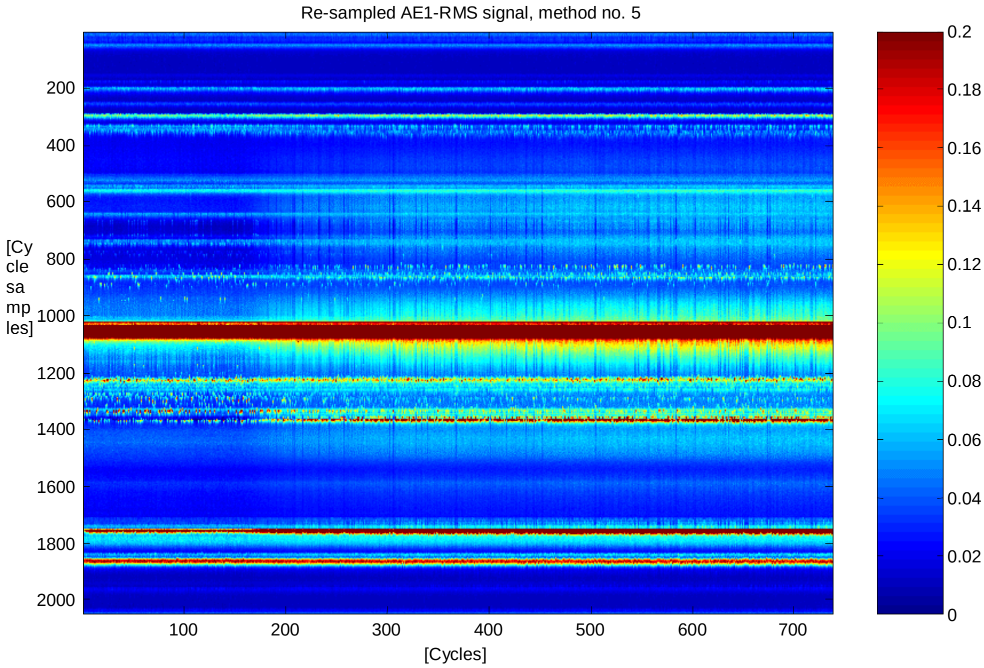


*Figure 2.20: Time plot of the re-sampled AE-RMS cycles of experiment no. 1 applying re-sampling method no. 5.*

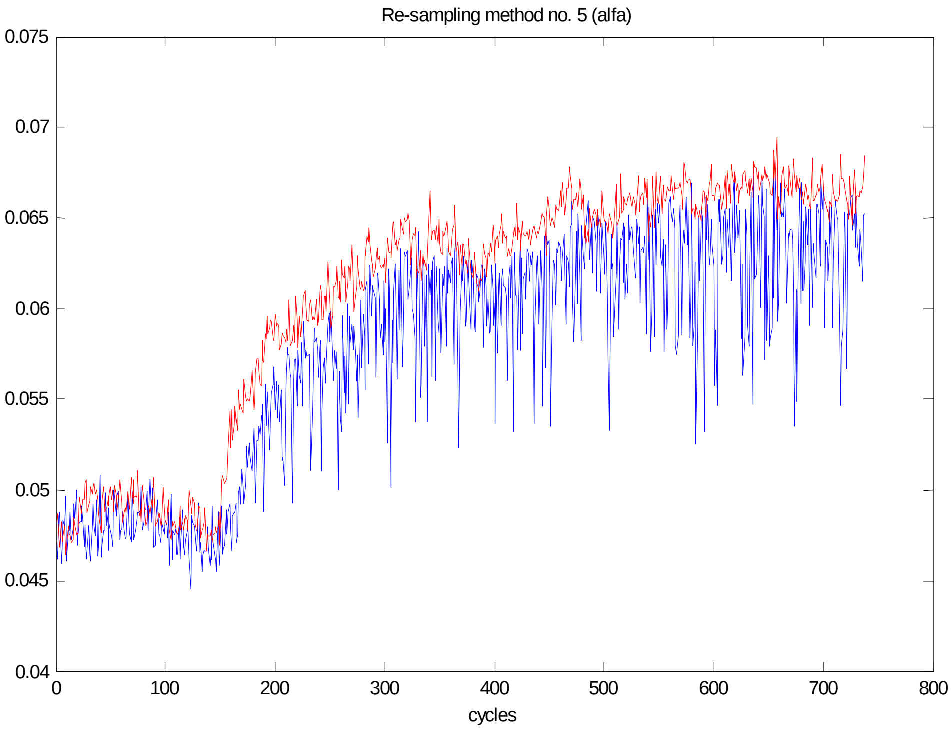


*Figure 2.21: Re-sampled feature signal method no. 5.*

*2.6.8 Project specification*

At this point it has been discussed how the described problems could be solved. Several issues have been discussed, but unfortunately not all can be investigated in this project. Therefore, choices must be made about how the automatic condition monitoring system should be implemented in this project. In the following, a short specification of the project is given.

- Four sensors are applied.

- Only the crank angle coded AE-RMS data is used. Thus the data set consists of four sensors providing 2,227 cycles with 2,048 samples per cycle.

- No noise reducing is done on the data.

- Re-sampling method no. 5 is applied in order to enlarge the single data set so statistics can be used on the system.

- Feature signals are generated in three ways: The mean value of the cycles, the standard deviation of the cycles, and PCA.

- A noise test is performed on the system. White additive Gaussian noise is added to the feature signals at selected Signal to Noise Ratios (SNR's).

- The segmentation task is split in three sub-tasks:
    - An on-line change detection algorithm, which at each cycle assess whether or not the engine has left its normal condition into a new condition. Thus, two alarms must be set, one when the engine has left the normal condition, and one when it has entered the new condition.
    - An off-line hypothesis test, which selects a window of cycles in which the two alarm times given by the on-line algorithm are present, and gives a reliable statement on the hypothesis: "The engine has changed condition".
    - An off-line change point estimation, which estimates the change points in the window of cycles, where the two alarm times are present, more precisely than the on-line algorithm. This is only done if the off-line hypothesis test is a success, i.e. it is very probable that a change has occurred.
- Classification is not considered at all.

The short project specification is shown as a flow diagram in figure 2.22 below.

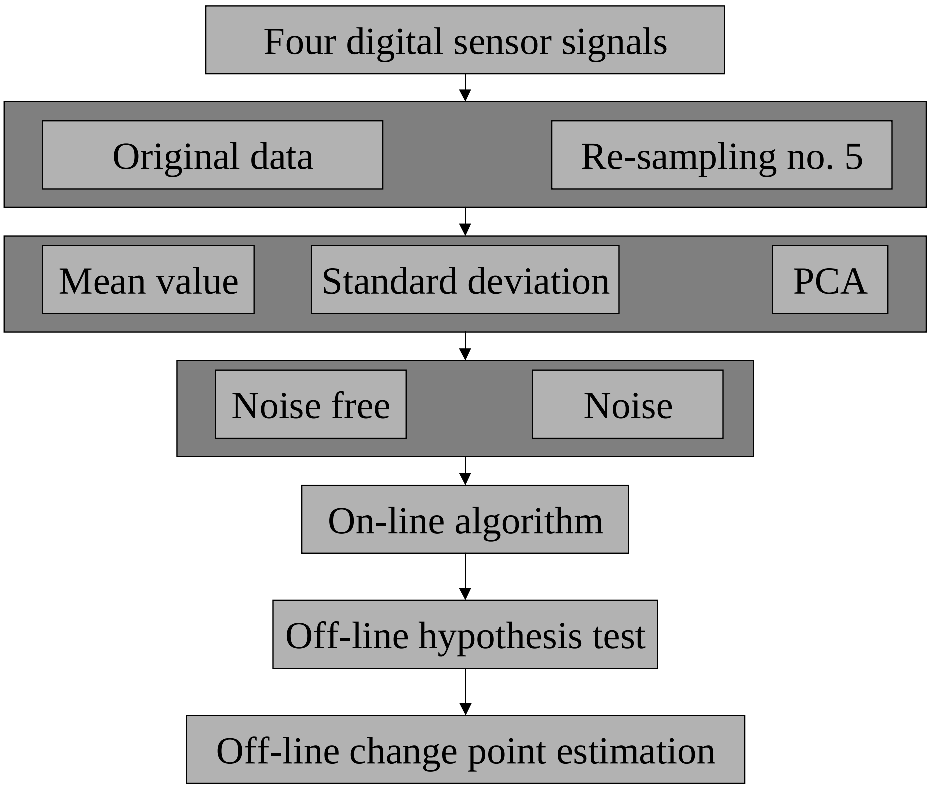


*Figure 2.22: Flow diagram of the short project specification.*