CHALMERS

Configurable on-board vehicle data logging with Principal Component Analysis

Master of Science Thesis

NADIA HOLTRYD

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden, 2013

Acknowledgements

The thesis project was performed at Volvo Group Trucks Technology in the summer and autumn of 2013.

I would like to thank my supervisor Wilhelm Wiberg, Volvo GTT, for coming up with the original idea for the thesis and providing guidance and inspiration along the way. I would also like to thank my examiner Olaf Landsiedal, assistant professor at Chalmers University of Technology.

Moreover, I want to express my gratitude towards all members of the Embedded Software Development team at Volvo GTT who hosted me and were always nice and helpful.

Nadia Holtryd
Göteborg, 2013

Abstract

Modern trucks are complex electro-mechanical systems with dozens of networked electronic control units (ECUs). The operation of these systems is monitored through diagnostics and logging. Stakeholders are interested in diverse sets of the collected information. During product development, diagnostic information helps to improve the product, while traffic safety research can benefit from, for example, accident statistics. Recording and storing large amounts of data for stakeholders to analyze later is often not possible in desired scales due to on-board hardware constraints. Similarly, analysis of these data volumes can take a considerable effort.

The solution proposed in this thesis uses statistics to implement an on-board logging unit (LU) that has low resource usage, is easily changed, and returns valuable and compressed information. The system consists of a configurable on-board unit that uses a multivariate statistical method called Principal Component Analysis (PCA). The PCA technique can create a compressed abstraction of the raw data, which highlights patterns and relations. The LU can send this data to an off-board data facility, where further analysis is possible. On-board compression allows the system to respond to a growing demand of diagnostic data without overtaxing the on-board resources. Stakeholders can dynamically upload new configurations to the vehicle to enable quick and frequent changes, which ensures that the currently most important information is gathered.

This thesis work investigates the solution through development and test of a prototype unit. The developed software runs in a simulated environment with log files from real vehicles. The method gives versatile analysis abilities for linearly related signals. In tests is the method shown to be able to distinguish between different driving scenarios. Tests regarding monitoring tire pressure and brake performance investigate the sensitivity of the PCA method. In the case of tire pressure monitoring a radius change of 9.5 mm is detectable, which shows ten times higher detection sensitivity compared to the unprocessed signals. In the case of brake monitoring is it possible to detect a difference between braking on flat and sloping ground with 5% inclination. The developed PCA implementation managed to capture large amount of variance in the compressed datasets, a compression to 15-30% of the original data contained 70-90% of the variance. Calibration is needed for the method to perform well, which could possibly be time consuming. Through the performed tests is the applicability of PCA in the vehicular setting shown to be good

The possibility to make better decisions based on available information is beneficial in all aspects related to vehicles. The possibilities for lower fatality in traffic accidents and better resource usage of both fuel and vehicle parts, makes the search for better vehicle data logging solutions important both with regard to resource management and safety requirements.

Keywords: vehicle sensor data, PCA, data aggregation, embedded systems

Contents

Abbreviations

CAN	Controller Area Network
CDM	Compressed Data Model
DTC	Diagnostic Trouble Code
ECU	Electronic Control Unit
EDR	Event Data Recorder
GPS	Global Positioning System
HW	Hardware
LU	Logging Unit
NMSE	Normalized Mean Square Error
NN	Neural Network
OBD	On-Board Diagnostics
PC	Principal Component
PCA	Principal Component Analysis
PDA	Personal Digital Assistant
RPM	Revolutions Per Minute
SOM	Self-Organizing Maps
SWC	Software Component
TPMS	Tire Pressure Monitoring System

1 Introduction

Solving the transportation needs of the world is a huge problem with widespread impacts on the environment and on human lives. One way of making the transportation solutions more effective is to use a specially tailored vehicle for the intended usage. The automotive industry is increasingly focused on specialization and customization. The goal is to optimize product features with regard to transport effectiveness, safety, quality, and environmental impact leading to a minimized lifetime cost for the vehicle [1]. One piece in the puzzle is to have better understanding of how the vehicles actually are used, through the sensor data available in the vehicles.

Modern trucks are complex electro-mechanical systems with dozens of networked Electronic Control Units (ECUs). Sensors are placed throughout the system. This connectivity makes it possible to access sensor signals, which give a view of the state of the vehicle and its components and subsystems. The available information is of interest to a number of different stakeholders, as for example development, fleet owners, and traffic accident researchers. The stakeholders want to optimize the vehicle and its functionality, which leads to a need for various types of information, as for example driver behavior and vehicle usage [2] [3]. Developers want to know how the vehicle is performing, to get errors, faults, and abnormal behavior, in order to improve the vehicle and understand when and why components fail. Fleet owners want to be able to plan for maintenance to minimize the risk of vehicles breaking down and missing scheduled runs [4] [5] [6]. Drivers, fleet owners, automotive companies, and insurance companies are all interested in on-board information from the vehicles [7].

There is a nearly endless amount of available information on-board the vehicles and the demand for more useful information to the stakeholders is growing. Today, there are strict constraints for an on-board ECU. There are limits for processing capability, memory, and power consumption [8]. Because of the limited resources needs stakeholders make a stringent selection of which information to record. Commonly, stakeholders cannot update their selection continuously under the vehicles lifetime since the conditions for logging are not configurable. At service-centers is the recorded information in the vehicles uploaded to off-board data facilities. The sending of information is slow and occurs infrequently. The vehicles can also send information through telematics, but the capability is limited because of the cost for sending large amounts of data. Another consideration is that the large amounts of information at off-board data facilities needs processing to become useful which highlights the need for the “right” information to be recorded [9].

1.1 Project proposal and objective

The proposal in this work is to have a configurable on-board unit that uses a multivariate statistical method called Principal Component Analysis (PCA). PCA can compress the data, while keeping the valuable information content. The logging unit can send the compressed data to off-board data facilities where further analysis is possible. The compression of information can solve the problem of growing amount of information needed while not overtaxing the limited on-board resources. The stakeholders can dynamically download configurations to the vehicle, to enable quick and frequent changes. This can ensure that the LU records the most important information at each moment. By using the PCA method, additional important information becomes available, when analyzing relations between signals. The extracted information can support other types of analysis regarding the vehicle than was possible before [10, 11].

The key questions are what advantages the proposed approach can have for the vehicular setting, considering usefulness of the PCA, configurability, and resource demand. The first question with respect to the PCA method is its suitability for the intended application. In this work is the PCA methods usefulness evaluated with regard to efficient compressions and ability to highlight patterns and relations in the dataset.

This project investigates the found logging solution through design and implementation of a prototype. The implemented LU tests the applicability of the method by using it to analyzing different vehicle usage scenarios. Tests, monitoring changes in the tire pressure and brake performance, investigated and determined the sensitivity of the method to notice change in the relation of signals. The developed PCA implementation manages to capture a large amount of the variance in the compressed datasets, in shown tests compression to 15-30% of the original dataset captures 70-90% of the variance. Moreover, the method showed good capability of detecting small changes. In the case of monitoring tire pressure, ten times higher detection sensitivity compared to the unprocessed signals was confirmed. In the case of case of monitoring brake performance, it was possible to detect if the brake event had occurred on flat or sloping ground with 5% inclination. The signals used in this study are highly correlated and linearly interrelated, which made them well suited for the PCA analysis. Scenarios where signals do not exhibit a linear relation to each other are not suitable for the analysis. Neither are situations where the behavior of interest is not distinguishable from other events.

1.2 Outline

In this chapter (Chapter 1), is the topic of the thesis and the research objective introduced. Chapter 2 gives a background to the topic of vehicle logging and data analysis, presents the PCA method, and describes the related work. Next, Chapter 3 describes the design and implementation of the method followed by evaluation of developed PCA unit. Chapter 4 presents and discusses the test results. The results are an outcome of three different test scenarios; a) vehicle usage and driver behavior tests, b) tire pressure detection test and c) brake performance test. In each section is the summary of test results presented. Chapter 5 summarizes the thesis results and evaluates the goal attainment. Finally, Chapter 6 presents the conclusions regarding the thesis work and make suggestions for future research directions within the area.

2 Background

This chapter describes the background and related work of this thesis. Section 2.1 describes logging in the vehicular setting and the different approaches taken in this area. The next section, Section 2.2, describes data analysis and presents statistical methods, followed by a description of the PCA method. Finally, Section 2.3 describes the related work with regard to both released products and research solutions.

2.1 Vehicle logging

The modern truck is a complex mechatronic system containing several distributed and networked ECUs. The different ECUs are specialized to handle specific subsystems and their functionality, as for example an Engine Control Module or Gearbox Control Module. Each ECU receives and transmits messages on the different networks using for example the CAN protocol [12]. The ECU, controlling the specific subsystem, is connected to sensors and actuators, which together provide the functionality. In the event of faults Diagnostic Trouble Codes (DTC) are set. The On-board Diagnostics (OBD) connection is the legally defined standard for the connection through which DTCs can be read at service-centers [13] [14].

Figure 1. In-vehicle electrical system [15].

One common approach to logging is to study events. A basic assumption of the event-centered approach is that experts on the system can define events of interest and that it is possible through signal values to identify when these events occur. For selecting which information to collect the same techniques can be used for different logging objectives. In the traditional approach, what varies between different logs is; a) what trigger conditions are used for the event, b) which information is saved, and c) how this information is saved [9].

In a traditional logging approach, there are in principal four different ways to save information; *event records, histograms*, *sensor signal values* and *counters* [9].

Events of interest activate *event records*, which records the current time and duration of the event. In addition, a LU can record other sensor values or counters that are of interest to the specific event. The event activation is for example one or more sensor values above or below a threshold. A single log has the capability of saving up to a certain number of events to flash memory, but after that the oldest record is overwritten (circular overwriting). Filtered event records are a special case of event records where the event activation and deactivation conditions need to be fulfilled for a specific time to make the event a recorded event. This is to avoid saving noise as a real event [9].

Histograms record information about probability of two or more variables being in certain states. The histogram is an estimation of the probability distribution of continuous variables. It records in which interval sensor values currently are in, in two or more dimensions. For example, a log of the engine can have histograms where engine speed is related to engine torque, coolant temperature and vehicle speed. For each of the histograms, engine speed is on one axis and the other variable on the other. These histograms give information about how long time the engine is used in different settings [9].

Sensor signal values are recorded each time the log is executed. The most common values are fuel consumption, distance, and time. The fuel consumption and distance are calculated as the change between two consecutive readings i.e. the last time the log was active and the current value [9].

Counters are usually an occurrence counter that count how many times a special event have transpired.

The resource usage is an important aspect of a log since it is run onboard the vehicles. The CPU load is small because the logs are seldom run compared to other tasks and have low computational complexity. However, memory demand can be large [9].

2.1.1 Objectives and approaches

Commonly, logging has the purpose of studying the vehicles health status and how the vehicle is used [9]. Since modern trucks are built for a certain usage, is it interesting to see how well they handle the wear when used as intended. In this area, experts of the system define intervals and circumstances that are unusual, can indicate fault etc. The sensor values can for example be how long the truck has run, current speed, road quality etc. Development can use this type of information, to tailor the system to a specific use case. Information recorded for the purpose of creating a better product is called post-development customer data [16] [17]. By logging relevant information can validation be made that developed functionality is used in the intended way and is of use for the customer. A special area of this type of logging is in test vehicles where the limited resource demands, of for example storage, characteristic of the mass produced truck are not applicable. In this setting data collection devices focus on collecting all data on the internal networks. The acquisition of data is activated when an exceptional behavior occurs, which is observed by the test driver or the system. One common solution for the collection is to use a rolling buffer. All sensor values are stored in this device at certain times. The last buffered time series will be recorded when a fault occurs. [3]. The solution with a rolling buffer is also useful for crash logs, where it is interesting to have knowledge of the state of the system leading to the crash [18].

In diagnostics the objective is to identify and trace faults. The events that are of interest are when faults occur. The future in this area is to be able to predict faults before they occur, to avoid costly unplanned stops [4] [5] [6]. Service shops can replace parts if the probability that they will break is great since the cost of a new part can be much less than that of the unplanned stop.

Logging solutions for accidents and crashes need to take into account that the LU must survive the crash to be useful. Research has shown that this is in many cases not true, the crash logs are destroyed or incomplete after the crash [19]. In this area, one possibility is that vehicles contact the correct authorities automatically and given information directly about the accident scene [7]. The capability of vehicles to call for assistance in the event of an accident is a service sold by many companies, but not the capability to send relevant information regarding the crash [20] [21] [22].

Event Data Recorder (EDR) is an LU that handle information about crashes called. There exists an IEEE standard for Motor Vehicle Event Data Recorders where the goal is that comparable data is generated by all vehicles [23]. Insurance companies, the police, automotive companies, and traffic accident researchers use the information. The use of vehicle data in this setting is still under considerable debate because of the sensitive personal information contained in the EDR. In the EU there have been projects considering a mandatory introduction of accident data recording devices [24]. In the US, the National Highway Traffic Safety Administration (NHTSA) has proposed a rule to require all automakers of light passenger cars to install EDRs in the vehicles by 2014 [25] [26]. Laws govern logging in many other areas as well, such as faults, emissions, and driver rest times.

Research in traffic accidents and driver behavior are two areas where data from vehicles are vital. Many times data is special logging solutions used in test vehicles to collect information, but data from ordinary usage is often more useful since drivers often change their behavior when they know that they are observed [27]. Studies on automatically classifying drivers are common, since bad drivers are a risk in the road traffic.

Many of the discussed areas use specialized logging solutions, to avoid the strict resource limitations for logging solutions incorporated in the vehicular system. Many of these solutions use the OBD connection to access sensor values and PDAs for the calculations [28].

More sophisticated solutions to logging are also emerging, especially with the goal of automatic diagnostics [29] [30]. One idea is to use a model-based solution, which is becoming more and more popular in different areas of monitoring, from chemical processes to nuclear power plants. The model-based solution is to use the data itself to build models. These are often a low dimensional summary of the datasets characteristics and relations. The technique finds the models that represent normal behavior dynamically from the system or group of systems. With these models, it is possible to see if the current behavior of the system corresponds to the model or if the system has begun to exhibit a new behavior by checking agreement of new datasets with the found model [31, 32, 33, 34]. In automatic diagnostics and fault detection, the goal is self-monitoring vehicles, which would make it possible to have predictive maintenance. Through using predictive maintenance, great benefits are possible with regard to lifetime costs and resource consumption [35] [36] [37].

Related work (Section 2.3) presents some of the ideas and prototypes for more sophisticated logging systems. At the heart of a more advanced logging system is the problem of analyzing data.

2.2 Data analysis

There exist many different data analysis methods. The methods make different assumptions concerning the dataset and are applicable to different problem settings. What they have in common is the goal of finding relations and patterns in the data through a good representation, i.e. to represent the data in a way that highlights its characteristics. Through a more low dimensional representation, relations or patterns are easier to observe [38] [39].

2.2.1 The problem of pattern recognition

In data analysis, it is important to differentiate between supervised and unsupervised classification [39]. In supervised classification, the problem is to label a newly arrived pattern from an already known set of different labels. Unsupervised classification is exploratory, the labels obtained are solely dependent on the data and there is no earlier knowledge of labels [38] [39].

There are three main steps in unsupervised classification i.e., clustering, see Figure 2. The first step is pattern representation, which can include feature selection or abstraction. The goal is to find a representation of the data that make the pattern of interest in the data more distinct and easier to analyze. To simplify the analysis, an effort to identify the most effective subset, for the intended analysis, is possible. This activity is called feature selection. This first step is often the hardest and most important for getting good results from the analysis. The second step is to measure similarity between the objects. A distance function can find the dissimilarity between two patterns, as for example Euclidean distance. Grouping the patterns, performing the third step of the algorithm, is performed differently depending on the algorithm [38] [39].

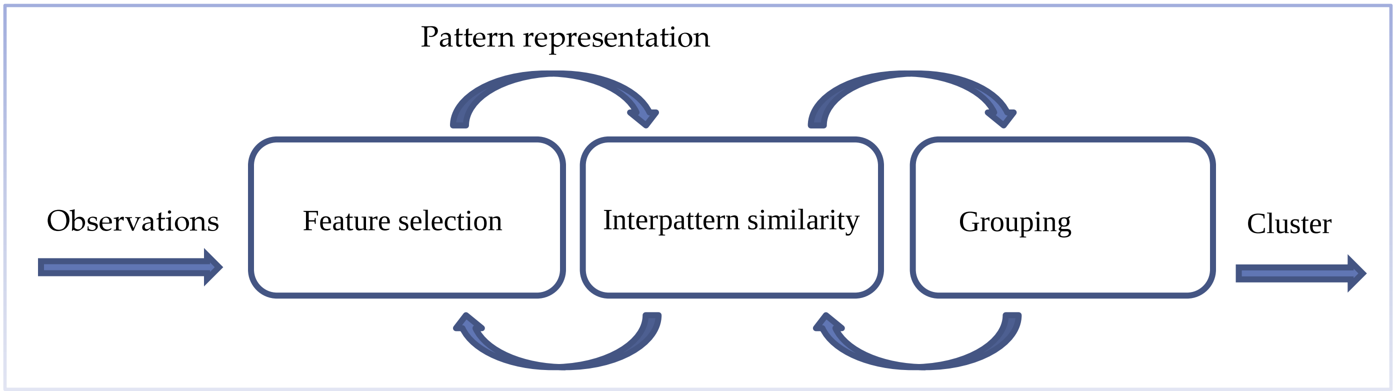

Figure 2. The three general steps in clustering; feature selection, interpattern similarity comparison, and grouping.

K-means clustering is one of the simplest unsupervised learning algorithms. The main advantage of that method is its simplicity, which makes it one of the most commonly, used clustering algorithms.

The method of K-means clustering has the following steps:

1. Choose an initial solution of k centers, centroids.
2. Assign each object (point) to its closest centroid.
3. Recalculate the position of the centroid according to the mean of the assigned objects.
4. Repeat steps 2 and 3 until the position of the centroids no longer change.

One of the problems with the method is that the result is heavily dependent on the initial assignment of centroids. Another problem is that the number of clusters has to be known before the method is used and executing the algorithm with the wrong number of clusters produces poor results [40].

Clustering tendency is the analysis of data to determine if and to which degree the data exhibits clustering. All data can be clustered but only data that have a good cluster tendency has a meaningful result as clusters. It is nearly as hard to determine clustering tendency as to determine good clustering [41].

Hopkins method is a test to see how clustered a dataset is compared to a Gaussian distribution. The test uses Euclidean distance to measure the distance between data points. The idea of the method is to compare the analyzed dataset to a Gaussian distributed dataset, through comparing the distance between the points. U is the inter pattern distance between the real data points and w is the distance between the artificial data point in a randomly selected pattern while d is the dimension, see Equation 1.

$$H = \frac{\sum_{j=1}^{m} u_j^d}{\sum_{j=1}^{m} u_j^d + \sum_{j=1}^{m} w_j^d} \qquad (1)$$

The null hypothesis is that H should be above 0.5 if the data exhibits a clustering better than a random distribution [42].

2.2.2 Statistical methods

This section presents some statistical methods, which is used throughout the report.

Variance is a measure of dispersion of a random variable in a distribution. If X is a random variable with mean μ then variance is calculated according to Equation 2 [43] [44].

$$Var(X) = \sigma^2 = \mathrm{E}[(\mathrm{X} - \mu_X)^2] \tag{2}$$

Covariance shows how different datasets vary from the mean with respect to each other [44]. A positive covariance means that the datasets increase together i.e. if one of them increases the other will as well. A negative covariance indicates that if one of the variables increases the other will decrease. If the covariance is zero the datasets are linearly independent of each other.

If X and Y are random variable with means μ_X and μ_Y then their covariance is calculated according to Equation 3 and 4 [43] [44].

$$Cov(X, Y) = \mathrm{E}[(\mathrm{X} - \mu_X)(\mathrm{Y} - \mu_Y)] \tag{3}$$

$$or$$

$$Cov(X, Y) = \frac{\sum_{i-1}^{n}(X_i - \mu_X)(Y_i - \mu_Y)}{(n-1)} \tag{4}$$

Equation 5 describes how a covariance matrix is calculated, in the case of three dimensions X, Y and Z. The matrix will be symmetric, since covariance is symmetric [43] [44].

$$\begin{matrix} Cov(X,X) & Cov(X,Y) & Cov(X,Z) \\ Cov(Y,X) & Cov(Y,Y) & Cov(Y,Z) \\ Cov(Z,X) & Cov(Z,Y) & Cov(Z,Z) \end{matrix} \tag{5}$$

Correlation is closely related to covariance. The equation for calculating the correlation is shown below, Equation 6. The correlation between two variables always ranges from -1 and 1.

$$Cor(X, Y) = \frac{Cov(X,Y)}{\sqrt{(VarX)\,(VarY)}} \tag{6}$$

For example if two variables have perfect positive correlation i.e. correlation of 1, then they lie on a straight line with positive slope [43] [44].

Confidence intervals are a type of estimate to indicate the reliability of an approximation of an observed population parameter. The confidence interval depends on which distribution the population has [43] [44].

2.2.3 Principal Component Analysis

PCA is a statistical technique for multivariate analysis. The method identifies patterns in data and represents the data in a way that highlights similarities and differences. The central idea is to reduce the number of dimensions of the data while preserving as much as possible of the variations in the original dataset. PCA has four goals. The first goal is to extract the most information from the data. The second goal is to compress the data by only keeping the most characterizing information. The third goal is to simplify the description of the data and the fourth goal is to enable analysis of the structure of the observations. The analysis enables conclusions to be drawn regarding the used variables and their relations [45] [11] [10].

The analysis is performed through transforming the data to a new set of variables, called the Principal Components (PCs). The PCs are uncorrelated and ordered so that the first few PCs retain most of the variations of the total dataset. The first PC describes the dimension in which the data has the biggest variation (variance) and the second component describes the dimension in which it has the second largest variation (variance) [45] [11] [10]. Figure 3 shows samples from two linearly related variables. The first PC will be in the direction of the most dominant relation in the dataset, as is shown in the figure. The second PC is in the second largest variation but orthogonal to the first.

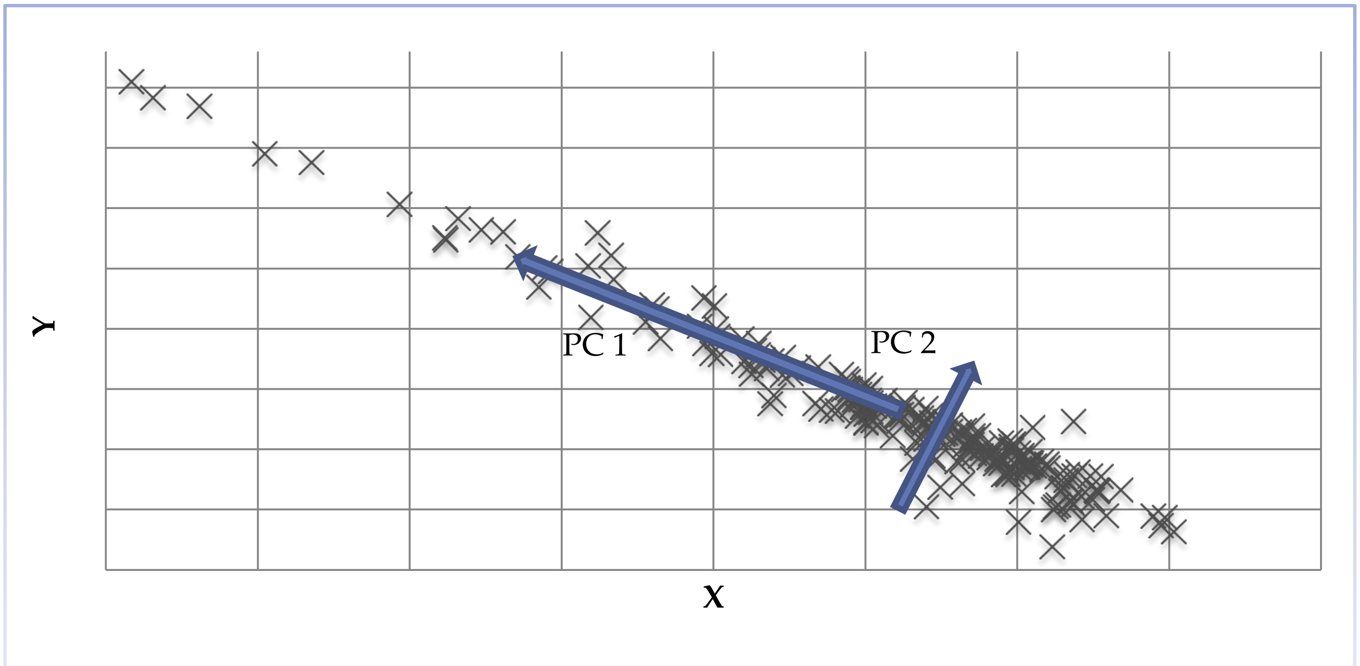

Figure 3. Plot of samples from two closely related variables. Note that PC1 is in the direction of largest variance in the data set while PC2 is in the direction of second largest variance, PC2 is also orthogonal and independent of PC1.

At the heart of the PCA is the calculation of *eigenvectors* and *eigenvalues*. An eigenvector (v) is a non-zero vector which when multiplied by a square matrix (A) yields a constant multiple of that matrix [43] see Equation 7. The constant multiple is the eigenvalue (λ) [11].

$$A * v = \lambda * v \qquad (7)$$

PCA is the decomposition of the data matrix X, with m samples and n variables. The parts of the decomposition are the sum of the outer product of vectors t_i and p_i in addition to the residual matrix E, see Equation 8. The first part, $T_k {P_k}^T$, describes the system variations and the second part the residual matrix E contains the unwanted information e.g. noise [32] [31]. The T used in the different equations is the transpose symbol.

$$X = t_1 {p_1}^T + t_2 {p_2}^T + \cdots + t_k {p_k}^T + E = T_k {P_k}^T + E \qquad (8)$$

The definition of the covariance matrix of X, with m rows and n columns is in Equation 9.

$$Cov(X) = \frac{X^T X}{m-1} \qquad (9)$$

The vectors p_i are orthonormal and the vectors t_i are orthogonal which leads to Equation 10 and 11.

$$if\ i = j\ then\ \ {p_i}^T p_j = 1 \qquad (10)$$

$$if\ i \neq j\ then\ \ {p_i}^T p_j = 0\ and\ \ {t_i}^T t_j = 0 \qquad (11)$$

The original data X multiplied with the transformation vector p_i results in the t_i vector, see Equation 12. The t_i vector is the *score*, which is the result from the analysis.

$$X p_i = t_i \qquad (12)$$

The score vectors contain the information of how the observed data, the samples, are related to each other. The transformation vectors p_i, *loadings*, are the eigenvectors of the covariance matrix of X. These loadings contain the information of how the variables are related to each other. For each eigenvector p_i , Equation 7, holds. Compare Equation 13 to Equation 7 to see the similarity to the equation for solving ordinary Eigen problems.

$$Cov(X) p_i = \lambda_i p_i, where\ \lambda\ is\ the\ eigenvalue\ associated\ with\ the\ eigenvector\ p_i \qquad (13)$$

The eigenvalues are a measure of the variance described by the corresponding score and loadings pair. It can be proven that this pair captures the greatest amount of variation in the dataset, i.e. the pair gives results with a linear factor from the optimal solution [32] [31].

2.2.3.1 Choosing the correct number of PCs

How many PCs to use is a very important consideration for the analysis. The problem is open but there are guidelines. The number of components k, has to be less than or equal to the dimensions of the data matrix X, see Equation 14.

$$k \leq \min\{m, n\}\, where\ m = number\ of\ samples\ and\ \ n = number\ of\ variables \quad (14)$$

There are a number of different criteria for choosing the correct number of PCs, depending on the application. First guidelines for methods that focus on capturing the main characteristic in the dataset is considered .One indication of how many PCs to keep is to look at the eigenvalues. The eigenvalue associated with a certain component is the variance captured by that component. Through this method a decision to be satisfied with capturing 80% or 90% of the variance in the dataset, would lead to the associated number of PCs. Another method is to look for a sharp turn in the curve of the eigenvalues; this method is called the *elbow, knee,* or *scree* test. The components after the knee are not as important and can be omitted. This gives a good indication of how many PCs is needed. A third common method is to keep only those components whose eigenvalue is larger than the average. This is similar to keeping only those eigenvectors whose eigenvalues that are larger than one for PCA using the correlation matrix. Since the described methods are based on heuristics is it recommended to use more than one method for deciding the number of PCs [32] [31].

However, in some applications the objective is not to capture the main characteristic but to detect other patterns in the data. In these cases all PCs can be of interest since they each describe different patterns in the data. The least PC would for example describe the least significant relation. This is the relation most likely to contain outliers, observations that are atypical from the main part of the data [11].

2.2.3.2 Scaling

The scaling of the data is an important step of the analysis method, which can greatly affect the results. One common scaling method is called auto-scaling. Auto-scaling is good to use if the different variables use different units for measurement. The method first normalizes the data so that the variance of the different samples is equal, through dividing each sample with the standard deviation for that variable. The second step is to mean centralize the data so that the data is centered around zero. After auto-scaling is completed the data varies between 1 and -1. In some applications it is more effective to only use mean centralization on the data. This is when the spread of the data from a variable is of interest for the analysis and the different variables have the same unit [32] [31]. Section 3.2.2 discusses scaling is further.

2.2.3.3 Validation of model fitness

One simple method for creating a model of the data is to use linear regression. Based on the available data, an estimation is made by use of the *least squares method* to determine the best fitness. The least squares method gives the curve that minimizes the sum of squares of the vertical distances from each point to the curve. Equation 15 describes the relation between the variables X and Y by a linear equation.

$$Y = \beta_0 + \beta_1 X + \varepsilon \qquad (15)$$

In the equation β_0 and $\beta_1$are constants called model regression coefficients and ε is the error or random disturbance. β_0 is the crossing point with the Y-axis for the linear approximation and β_1 is the slope of same approximation. If the parameters are assumed to have a normal distribution a confidence interval can be given for the values to indicate the certainty of the model [46].

Two related statistics commonly used to measure the model fitness for PCA is Hotteling's T^2 statistic and the Q statistic [32] [31].

Hotteling's T^2 statistic checks the variations within the model. The T^2 statistic is the sum of normalized squared scores, see Equation 16 where t_i refers to the i^{th} row of $T_k$which is the matrix of k scores vectors from the PCA model and λ^{-1} is a diagonal matrix containing the inverse eigenvalues corresponding to the k eigenvectors [32] [31].

$$T_i^2 = t_i \lambda^{-1} t_i^T = x_i P \lambda^{-1} P^T x_i^T \qquad (16)$$

The Q statistic determines if the sample shows an unusual variation outside of the model. It compares the difference between the data point and its projection in the model, see Equation 17.

$$Q_i = e_i e_i^T = x_i \left(I - P_k P_k^T \right) x_i^T \qquad (17)$$

where Q is the sum of squares for each sample of E, for the i^{th} sample in X (x_i), e_i is the i^{th} row of E, P_k is the matrix of the first k loadings vectors retained in the PCA model, and I is the identity matrix. The Q statistic is a measure of the amount of variation in each sample not captured by k PCs. Statistical limits can be developed for Q and T^2 which helps determine the model fitness [32] [31].

2.3 Related work

In the field of vehicle data acquisition and analysis there exists both finished products and research ideas and prototypes.

2.3.1 Products

There are several telematics products based on vehicle data. Volvo Trucks has a fleet and vehicle management system called Dynafleet, where available information are, for example, vehicle position, driver rest times, fuel consumption and service intervals [22]. General Motors has a system called OnStar, which has approximately the same services [21]. Part of this system is vehicle diagnostics, which uses the DTCs for the different systems. For diagnostics, Volvo has a system called Remote Diagnostics, where the basic idea is that DTCs are communicated automatically to a service spot. The service spot can prepare for the service by, for example, order the parts that need to be replaced. Both companies also have preventive/recommended maintenance, where the maintenance intervals are automatically updated depending on the usage of the vehicle. Several other companies have similar products. The work in this project focuses on the on-board analysis in contrast to these products. The aim in this thesis is also to extend the available information from the vehicle.

There exist several products with logging capabilities, one of these focusing on advanced analysis is MineFleet [47] [48]. The information available through the OBD II connection is used and PDAs in the vehicle store and/or analyze the information. In contrast to these products the idea in this paper is to focus on extending the analysis capabilities within the existing electrical system since only a limited number of signals are available through the OBD connection. Many of the ideas used for PDA based approaches are not applicable in this work since the resource constraints are much less strict when using PDAs. This thesis focuses on investigating onboard analysis within the electrical system.

2.3.2 Research and development

This section presents projects using PCA in different settings. Next section describes different research solutions to vehicle data acquisition and analysis.

2.3.2.1 PCA in monitoring applications

Many projects use PCA for monitoring in different applications. The project described in [31] presents multivariate statistical process control tools for semiconductor etch processes. The tools are intended to determine if the etch process is operating within normal parameters or not. The application of these tools is complicated due to the large amount of normal systematic variation, which often is much larger than the faults variation. In the project is Q and T^2 statistics used to determine if new values are within the created PCA model.

The project described in [32] uses PCA to monitor temperature sensors in a nuclear reactor. The paper presents how a PCA model is created that maps the temperature variables into lower dimensional space and tracks their behavior using Q and T^2 statistics. In tests, a drift is imposed on the sensors and this disturbance is detected using the model. The drift information is contained in higher order PCA components. Specifically it is shown that the drift can be traced to the faulty sensor.

In [33] diagnosis of chemical process and sensor faults using PCA is described. The method is based on the partition that PCA does in the measurement space between normal and faulty subspace. Through the use of a direction vector, which describes the behavior of the fault or normal behavior, fault reconstruction can be accomplished.

In contrast to these three papers this project focuses on using PCA in the automotive domain. Since the behavior of the system and its sensors are of great importance for the results of the analysis, the choice of the application domain is of great importance. Next section describes suggested solutions for the area of vehicle data acquisition and analysis.

2.3.2.2 Vehicle data analysis solutions

Schweppe et. al. propose a flexible solution, that trades more on-board processing power against less memory, for data gathering in vehicles [8]. The solution makes use of a processing graph with operator nodes and connecting data streams which shall enable a detailed preprocessing where decisions about the importance of the data can be made. In the solution raw data is saved. The similarity with this project is that here is also a configurable solution considered which can change the timing and selection of data to be recorded. The difference to this project is that this project uses a compressed representation of the data. A compressed representation is used since that would enable more data to be saved. The use of compression also gives the possibility to send the data through telematics more cost effectively.

The VEhicle DAta Stream Mining (VEDAS) system was developed for monitoring and mining vehicle data streams in real-time [28]. The researchers in the VEDAS project are also involved in the product MineFleet, which is described in Section 2.3.1. The VEDAS system uses on-board PDA-based distributed data stream mining system and other remote modules connected through wireless networks. The system can monitor the data from the vehicle's on-board diagnostic system (through the OBD-II interface) and the Global Positioning System (GPS). The system analyzes the collected data both locally and in a central module. The first analysis aims to reduce the dimensions of the data through incremental PCA, Fourier transform, or online linear segmentation. The central module is a server running on a desktop with database access. The modules main functionality is to visualize the data and provide an interface to control the data mining operations on-board the vehicles. The approach is different from the work presented in this report since the system can only sample signals through the OBD-II interface. Only using signals through the OBD connection severely limits the logging capabilities since only a small selection of the total amount of signals is available. The other important difference is the use of PDA's instead of embedded ECUs for the analysis as is done in this project. The use of PDAs instead of the onboard resources changes the problem setting since the resource constraints are not as strict.

The same research team has suggested methods for data logging in [5] [6] [35] [36] [37] [49] [50] [51] [52] where ideas from data mining and analysis are applied. One of the differences is that these articles focus on automatic diagnostics and fault detection while this work aims at a data abstraction, which system experts can understand. The assumption in this thesis work is that useful analysis results is easier to created with the use of a configurable LU than with an autonomous self-learning system. Since the results of the analysis can be quickly understood and fine-tuned with vehicle system knowledge. Next, follows presentation of some of the articles in more detail.

In [5] a self-organized method is described where PCA is used. The method is tested on simulated vehicle data and a dataset from computer hard drives. In contrast to this work were the PCA method is evaluated with real data from vehicles. The use of real vehicle data enables evaluation of the method for the intended application. Since the method is dependent on the application setting is it important to investigate the methods fitness with actual vehicle data, which is possible in this project. The PCA implementation proposed in this work is also configurable to enable different types of analysis.

In [49] a telematics based method is described where Self-Organizing Maps (SOM) is used. The idea is to use low-dimensional representations of sensor values in the subsystems of the vehicles and use these representations for comparison. If the comparison finds the representation to be deviating from similar systems in a fleet, is the system possibly faulty. Instead of using SOMs, as analysis technique is in this project PCA used, this is among other reasons because it is easier to see the relation back to the signals with PCA.

The approach described in [50] uses a method called COSMO (COnsensus Self-organized MOdeling) which aims to “reduce development costs of fault detection software and create vehicle individual service planning”. In the method is models found through comparing Normalized Mean Square Error (NMSE). The null hypothesis is that the model variables (signals) follow a multivariate Gaussian distribution. Data from real vehicles is used to test the method. The weakness with this method is that the number of different combinations that need to be tested is large. In this project, PCA is used and self-organizing search for models is not considered since the models that PCA creates can be understood and used by experts of the vehicular system. PCA gives possibilities of understanding the relation that the compressed dataset describes.

The project presented in [51] describes an idea where on-board embedded agents searches after interesting relationships, that is described by different models as for example linear models or self-organizing maps. The suggested comparison method depends on which kind of model is suitable. For example are correlation coefficients or fitness error used in the case of linear models and distortion measure when clustering is considered. The search agents can send interesting relationships that they have found to a back-office, where these relationships can be compared against other systems. The idea of the method is to create a collaboratively self-learning group of systems. The idea is further explored in [52] where a method is described for searching for non-random nonlinear relationships. The method is demonstrated on synthetic and real datasets with one of the conclusions being that exploring relationships on real data is much harder since almost all real signals are related to some degree. However, in this work PCA is tested on real datasets from vehicles which give the possibility to test under the conditions for the application setting.

In [6] a classification method using Intelligent Sensors (agents) to do a similarity analysis is presented. Sequential clustering is used to create the Compressed Data Models (CDMs) that will be incrementally classified with the similarity analysis. Compressed data models are compared for similarity with other models existing in a "knowledge base"; if the new model is sufficiently different a new class is created and added. The incremental clustering is based on supervised clustering, where the idea is to find the largest density in the data space first i.e. the most important cluster. The method is tested with synthetic data. One problem area was to define the threshold for the classification. In contrast to using supervised clustering is the idea in this work to use PCA to create abstractions of the data. The difference to these works is that the authors aim for a self-organizing and learning system while in this project the aim is to investigate a configurable on-board approach. With the compressed dataset in this project it should be easy to understand the relation to the original dataset. The goal is also that the analysis should be lightweight enough to be done on-board.

In addition, other researchers is active in the area, an example with machine learning comes next. A neural network (NN) based fault diagnostics is described in [53], which is tested on a coolant system. The back propagating NN is trained on data from healthy and faulty conditions. An accurate diagnostic function is created. In contrast to this work is the method used for analysis PCA, which gives a better chance of understanding what is wrong. The relations between the signals can be understood and then also the models created with the signals can be understandable. The work presented here is focusing on on-board analysis.

In summary, there exists several products regarding vehicular logging but none of them uses a sophisticated on-board analysis method. PCA is used with good results in many other monitoring applications but have not been thoroughly investigated for on-board vehicle logging. Researchers evaluate several other analysis techniques for the vehicular setting. The results from these other analysis methods are not as easily understood as the PCA proposed in this work. Many of these approaches do not take into consideration on-board logging as is done here. Another difference is the proposed configurability and compression of the data that are important features of the LU proposed in this thesis.

3 Design and implementation

This chapter describes the design and implementation of the LU. Section 3.1 describes the complete data collection and analysis system. Next, Section 3.2 presents the developed PCA unit. Furthermore, justifications for the different decisions with regard to the design are described.

The subject of vehicle data acquisition and analysis has four main challenges. Addressing them is the foundation of the proposed design. Firstly, large amount of real-time information is available from the vehicles sensors. Secondly, the storage capacity is limited on-board the vehicles. Thirdly, the different stakeholders have diverse needs. Finally, the definition of what is valuable information will likely change over time.

The solution is to use automated analysis of the data to decrease the needed storage and to increase the information value of the data. Stakeholders are interested in the conclusions that can be drawn from the information, not in the data itself. The needs of this group will likely develop over time and new stakeholders are likely to emerge, as can be seen in Section 2.1. To meet the diverse needs of the stakeholders, the solution is to have a highly configurable LU. The high configurability can also be useful for the future since as the vehicles functionality evolves over time, so will the desired type of information.

The developed solution is a highly configurable software unit that handles analysis on-line in an ECU. The LU uses PCA to analyze and decrease the dimensions of the data. The high configurability solves the problem of stakeholders’ diverse and changing needs.

3.1 System overview

High configurability in combination with the possibility to download new configurations and upload data, changes the way logging can be used. The design of the LU makes it possible to use an interactive method of working with logging. The different stakeholders can flexibly choose the information they think is interesting. What is recorded can be tailored in order to monitor the usage of different vehicles, or fleets of vehicles, depending on the current need. Only the configuration needs to change; the components for logging are the same. One example is that the usage of a new functionality can be more closely monitored after it has been newly introduced by adding a configuration tailored to the new functionality. This information can be used to determine if it is working as intended.

Figure 4 describes workflow around the vehicle data acquisition and analysis system, the different parts that can be seen in the figure are:

1. The stakeholders have diverse needs and through the configuration can the stakeholders specify which information is of value to them. The configuration is sent to the on-board LU. The same configuration can be used on one or more vehicles, or fleets of vehicles. Here, logging can be tailored to specific needs and both long term and short-term logging can be configured.

2. The configuration is downloaded to the vehicle through the telematics gateway.

3. The on-board LU performs logging according to the different configurations. Sensor signal values are received and their information content is analyzed and saved. Compression is performed on the data. The chosen datasets are saved or directly uploaded depending on the nature of the information. A model of selected signals relations is recorded in the LU and a new model version needs to be created first when new signal values differ from the model. The use of models as well as the traditional approach (described in section 2.1.1 and 2.3) to distinguish interesting data can decrease the amount of data that needs to be saved. The use of models can also enable new functionality.

4. The compressed datasets are sent to an off-board logging data facility.

5. In the off-board facility more advanced analysis can be performed and long term storage is made available. The stakeholder that have requested the information can analyze the results and perform new logging accordingly if further information is needed or if the information needs have changed. Based on the data analysis models can also be found, which can be used in the on-line unit.

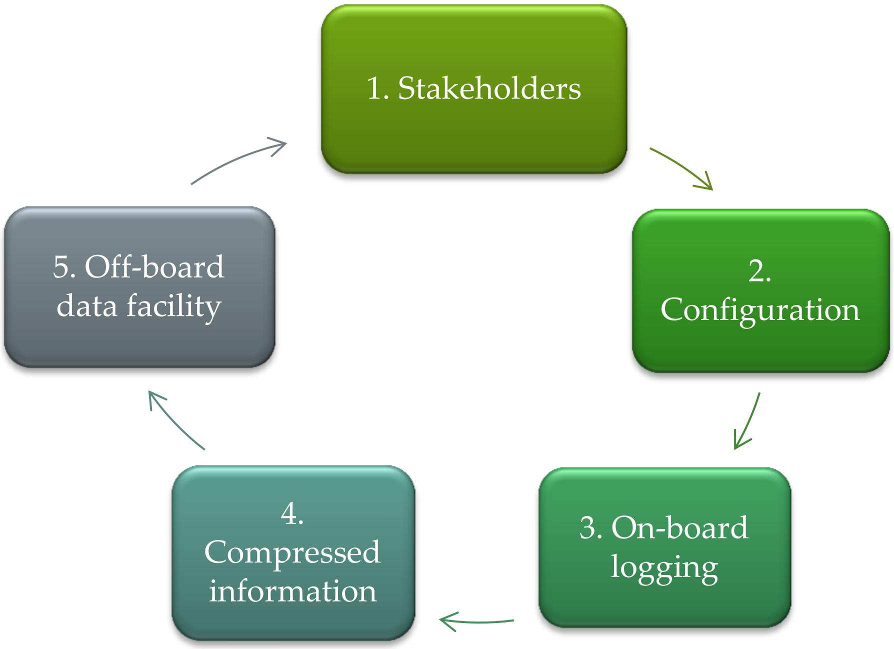

Figure 4. The workflow regarding the future vehicle data acquisition and analysis system.

To be able to evaluate the proposed idea, the analysis method has been implemented. The developed prototype has focused on one step in the proposed data collection system, the on-board logging (3). The objective has been to reduce the massive amount of data to a meaningful representation, which enables later analysis. The next section provides a closer description of the design and implementation of the developed PCA unit.

3.2 Developed PCA unit

The solution is highly configurable to enable continuous changes to be made regarding what is recorded and how it is performed. The unit can contain different logging methods so that the information requester can chose the correct method according to their needs. One of the methods in such a logging approach can be the PCA. PCA uses some of the common statistical tools for classifying data such as mean, variance, covariance, and correlation. Many of these can also be used as standalone methods for classifying data, to enable more functionality to the user.

PCA is chosen for the prototype because the method exemplifies a category of analysis methods. If the data has linear relations and is correlated, as data often is in vehicular systems, the method will give a compression that maintains a high amount of the information in the original dataset. It is a well-known technique that has been shown to give good results in other monitoring situations [31] [32] [33]. The described solution saves a compact summary of the data, which is derived by applying ideas from statistics to enable an analysis while preserving its characteristics.

The areas of interest with regard to the PCA method were how suitable the method was to use in this setting and what benefits and capabilities it has. Further topics were which quality the PCA compression can give and which analysis capabilities are possible with the use of the configuration. Furthermore, questions about how complicated the analysis was to implement and if it was possible to make a simplified version existed. Another important subject was regarding if and in that case how well the current system would support a configurable unit. Moreover, the resource demands of the unit were another concern. All these topics are most easily explored by developing a prototype.

The proposed LU can be placed both in a gateway node and distributed in the different subsystems. Logging in the gateway node, which has access to many of the data buses, can focus on general information. However, logging in the subsystems can focus on that system, because more detailed information will be available locally. Figure 5 shows an overview of the unit, where signals containing sensor values are the input and the compressed data are the output.

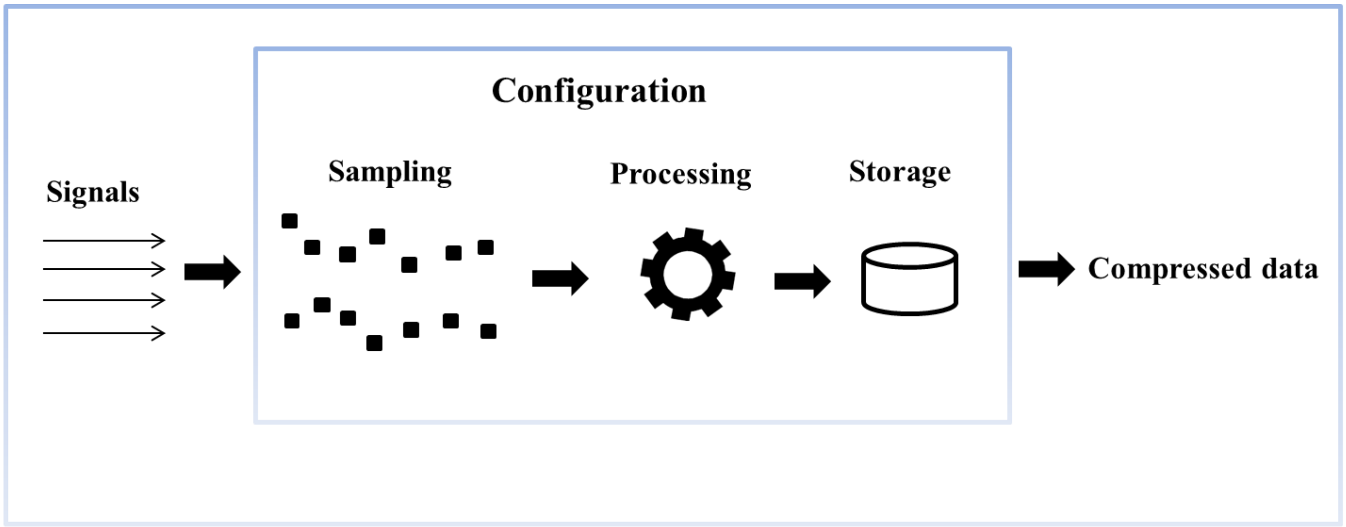

Figure 5. Overview of the PCA component, input is raw signal values and output is the compressed dataset.

3.2.1 Handling of data streams

One important question is how to divide the stream of sensor values that the LU receives. Since the usage of the vehicle might change over time, will the dataset exhibit concept drift, i.e. a model based on historical data does not correctly reflect the current system. When receiving a stream of data there are different methods to determine which data to use for calculation, see Figure 6. One method is to divide the stream into chunks of appropriate size that the systems can handle and ignore the fact that the input is a stream. When one chunk has been processed, the next can be handled. Another approach is to have a continuously updated dataset of a predefined size, which is the chunk that is used for calculation. When new data arrives, the dataset is updated with those values and the oldest values are excluded from the set [39]. However, regardless of if a continuously updated dataset is used or if static chunks are used and discarded, a time window of the system will be the basis of the calculations.

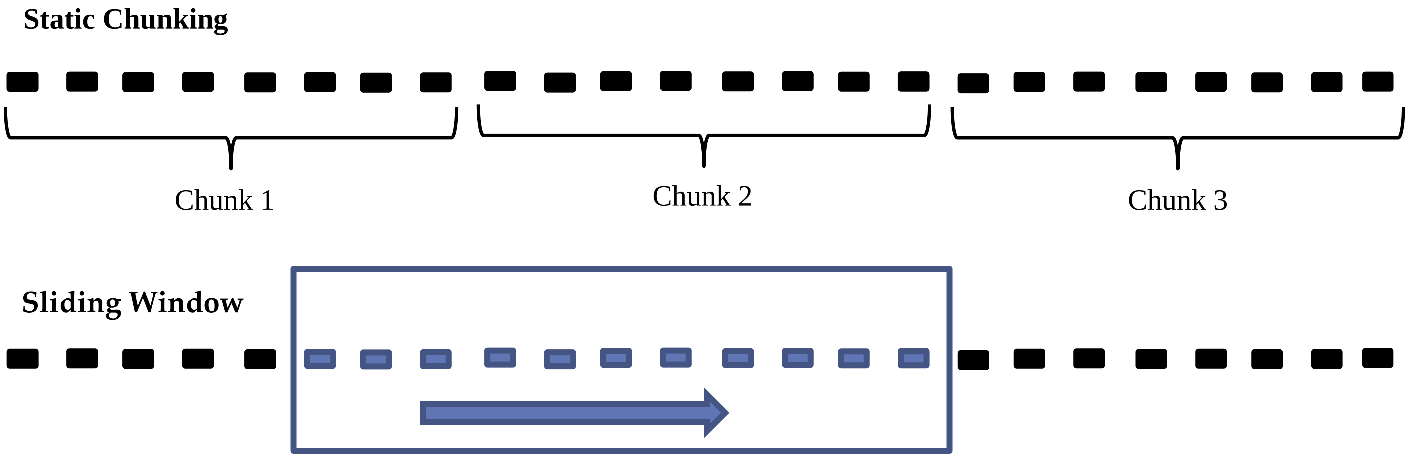

Figure 6. Two different methods for handling streamed data; static chunking and sliding window.

The data in the vehicle will usually be non-stationary with regard to mean and standard deviation, since that is normal in the vehicles usage. Both the mean and deviation will vary depending on the time window that is currently observed. This means that very different behavior can be observed since the mean and standard deviation is used for scaling the dataset. Section 2.2.3.2 describes more details about scaling while Section 4.1 discusses further the effect of the size of the observed time window. The chosen method for handling the streams in the developed LU is to take in a predefined amount of data and do the analysis on this chunk. The size of the chunk is configurable as well as the sampling frequency. Since it is not known how to handle the streams in an optimal way, is it important to be able to change this functionality.

3.2.2 Configurability and design

Configurability has been an important aspect of the component. There are three main configurable settings with regard to the input data to the LU. Firstly, which signals are monitored and used for sampling. The LU has access to many different signals but all of them are not of interest to analyze together. Secondly, the sampling frequency selected for these signals is also configurable. How often the value of a signal is sampled greatly affects which behavior will be seen in the sampled data, since the raw signal can be continuous while the samples are discrete. Thirdly, the number of samples to be used in the analysis is configurable. The number of samples together with the sampling frequency will determine what behavior is seen within the observed time window. The idea is that through changing the settings different behaviors can be observed. Using the remotely downloadable configuration, can the logging be easily changed and customized for a fleet, a group of trucks, or a single vehicle.

Figure 7 shows the logically division of the LU into three parts. The first part, the wrapper, handles reading and chunking of the data stream i.e. to read the selected sensor signals. It also handles how often the values are read by the choice of sampling frequency. The Wrapper also handles writing to flash memory.

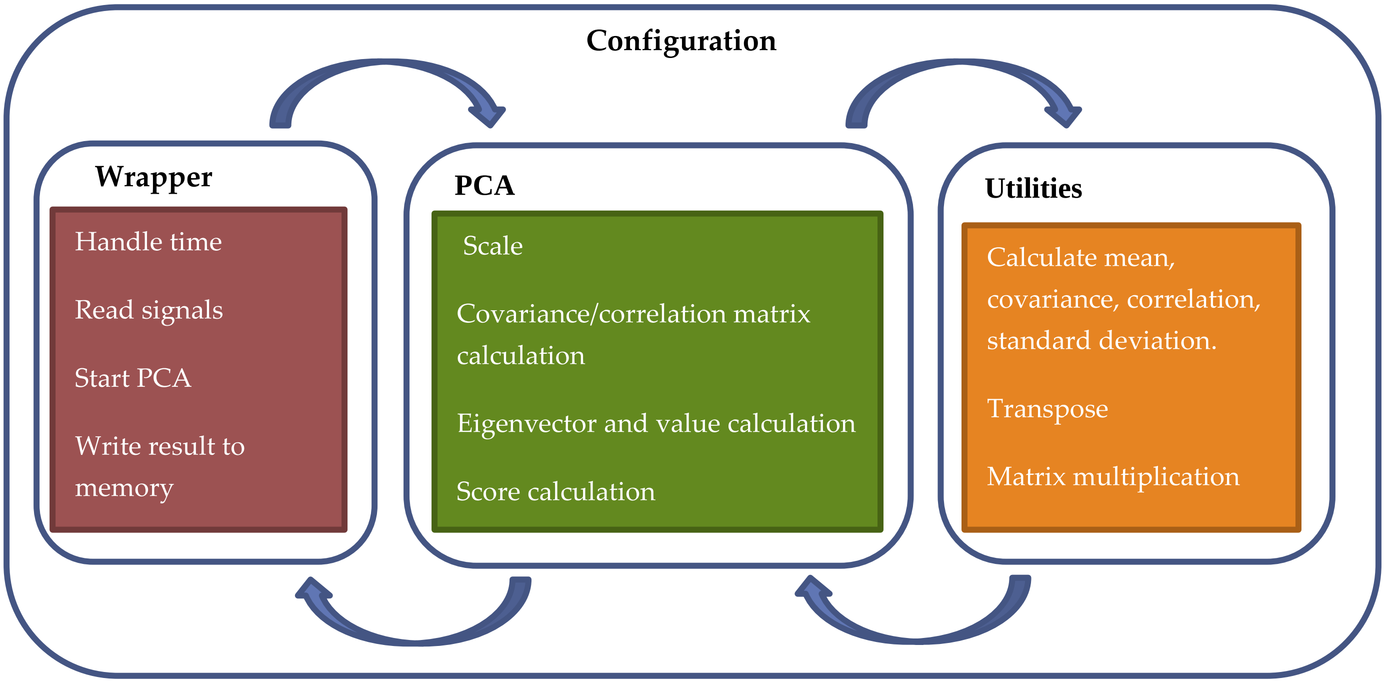

Figure 7. The three logical parts of the PCA Unit; Wrapper, PCA and Utilities. All parts use the configuration.

The second part performs the PCA. The implemented PCA method has the following steps:

1. Scaling of the raw data is the first step of the method. The data is auto-scaled, i.e. the data is scaled to unit variance and zero mean, see Figure 8. After the scaling the data is transposed to enable easier calculations. Two different methods are used for scaling; one is dynamic and the other is static. In the dynamic setting the data is scaled according to the mean and standard deviation in given dataset. The static version is used for comparisons, between different executions, and scales the different datasets with the same mean and standard deviation.
2. A covariance or correlation matrix is created from the raw data. The matrix will have as many rows and columns as the data have dimensions i.e. signals.
3. The eigenvalues and vectors are calculated from the covariance or correlation matrix.
4. According to the chosen number of PCs in the configuration, the eigenvector/s are selected and transposed. The first PC has the largest eigenvalue, so through sorting the values the correct components are found.
5. The eigenvector or matrix of eigenvectors if more than one PC is needed is multiplied with the normalized data. The resulting matrix or list is the original data, now in terms of its eigenvalues. This data, the score, is the result of the analysis.

The PCA uses methods from the third part, Utilities. The utilities part contains methods for different steps of the analysis, as calculating mean, covariance etc. and performing auto-scaling. The PCA part contains different versions of the analysis method, using for example different scaling methods or number of PCs. The configuration is used in all the different parts of the system since it specifies the current setting.

The scaling method used is auto-scaling since this method makes it possible to compare data with different units. Figure 8 shows the effect on the data from the scaling method.

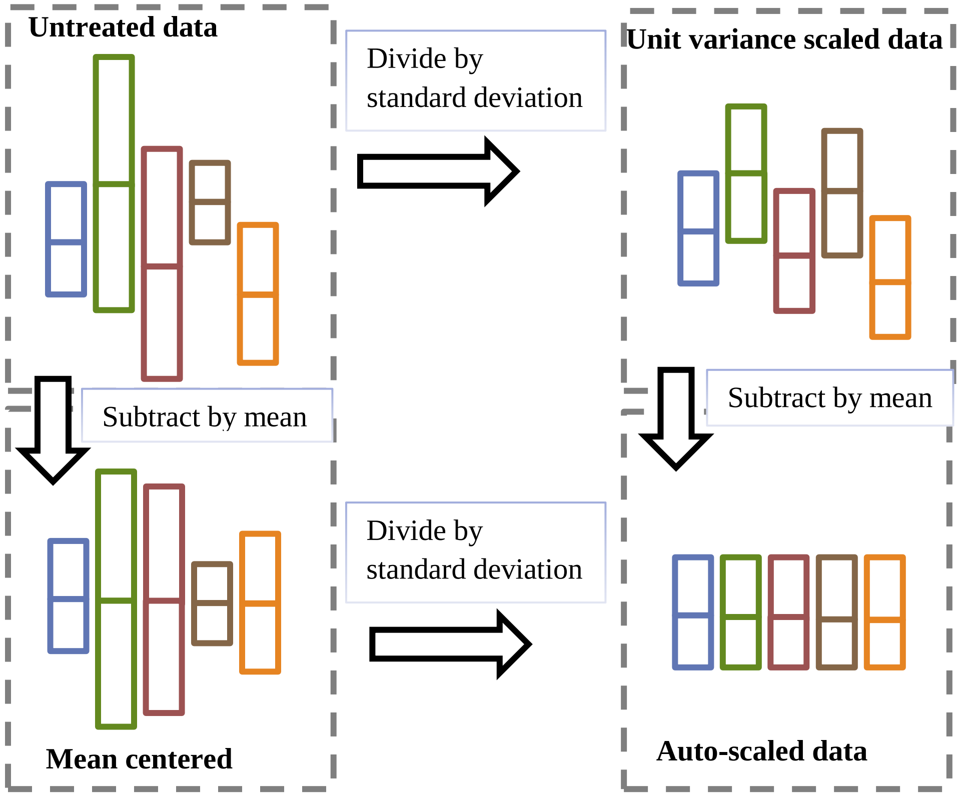
Untreated data
Divide by standard deviation
Unit variance scaled data
Subtract by mean
Subtract by mean
Divide by standard deviation
Mean centered
Auto-scaled data

Figure 8. Overview of different scaling methods impact on data. The methods presented are mean centering and unit variance scaling. Also shown is the combination of these two, auto-scaling.

3.2.3 Implementation

The implementation is done using Autosar [54] as an atomic software component (SWC). Tools used for development of the code are the actual production code tool chain for developing embedded software at Volvo GTT. It was important that the developed LU should be as close as possible to real production code to enable conclusions to be drawn regarding the feasibility of the implementation of the method in this setting. The complete LU and PCA method is implemented in the programming language C [55]. The implemented functionality has been unit tested with the help of the CUnit framework [56]. Unit testing is the method for testing individual pieces of the source code and has been used as a white box testing method [57].

The PCA implementation, being the subject of present work, is based on the descriptions found in [11] [31] [32] [45]. The statistical tools used in the algorithm, as covariance, are implemented from the definitions found in [44]. The calculation of eigenvalues and vectors cannot be done exactly for matrices with more than three dimensions. An approximation for a quadratic matrix has been implemented that first does a Householder reduction to tridiagonal form and uses a QL algorithm for solving the eigenvalues and vectors from the symmetric tridiagonal matrix. This method is chosen because the matrix used to find eigenvectors and their values will always be symmetric and quadratic since it is the covariance or correlation matrix. Another important observation is that the values will always be real, which means that we do not need a solver for imaginary numbers. The QL algorithm is a well-known Eigen solver. The implemented Eigen solver is based on two different algorithms, that both use Householder reduction and QL. The first one is from *tred* and *tql2* in [58] and the corresponding FORTRAN implementation in EISPACK and the Java implementation in JAMA. The second one is derived from the description in [59]. Householder reduction is a method for simplifying a matrix to tridiagonal form. Comparison of the results from the two implementations has been used to verify the correctness. The method uses orthogonal transformations and to reduce a $n\ x\ n$ symmetric matrix to tridiagonal form needs $n - 2$ orthogonal transformations. To find the eigenvalues the roots of the characteristic polynomial need to be found, as can be seen in Section 2.2.3, Equation 7.A QL algorithm is based on the observation that any real matrix (A) can be decomposed to the form:

$$A = Q * L \qquad (16)$$

Where Q is orthogonal and L is a lower triangular matrix. From this we can get Equation 17 and 18 where A' is an orthogonal transformation of A.

$$A' = R * L \qquad (17)$$

$$A' = Q^T * A * Q \qquad (18)$$

The QL algorithm consists of a sequence of orthogonal transformations. The workload for a tridiagonal matrix is $O(n)$ which is highly efficient compared to a general matrix where the workload would be $O(n^3)$. However, since the eigenvectors are also required for the PCA method the workload will be higher, about $3n^3$ [58].

The component is built into a simulated ECU (.dll) that is executed in CANoe [60] with other simulated ECUs. The CANoe tool enables simulation of an entire electrical system with ECUs connected by a CAN network. Log files from vehicles can be used as input. When the tests are executed CANoe will write data to a simulated flash memory. This simulated memory, a textual file, contains the memory structures that have been specified. The data is collected and transformed to readable form and plotted in graphs. This is done in the programming language Python [61]. The complete implementation is ready for being executed in hardware (HW) but this has not been done in the project because of time limitations.

4 Results and evaluation

The main objective of the evaluation is to see if the implemented PCA method performs well and is able to generate useful results in the automotive setting. The tests have been performed to investigate if PCA can give the benefits described in Section 2.2.3. Namely, a) how well PCA extracts the most important data and how good compression that is possible while keeping only this information and b) what kind of results is possible using the simplified description of the data to analyze the structure and relations between signals. Presented results show that the method can be used for compression as well as analysis with good results. Additionally, tests are presented that show how calibration through the configuration affects the outcome.

The configurability of the implemented LU has been a precondition for the evaluation and the performed tests are all done with the same implementation by only changing the configuration. The diversity of the analysis that is possible is the result of the configurability. Similarly, evaluation of configurability is performed through all the different test cases, since the different configurations give the ability to see diverse behaviors.

Log files have been collected from different vehicle test scenarios. The scenarios are designed to generate data of interest to aid in the evaluation of the developed LU. For the experimental evaluation, a subset of sensor signals was used. The signals used in the tests were chosen on the basis that they were available, easy to affect by normal usage, and easy to understand. Moreover, the signals were close to continuous and had a fast visible reaction. The chosen signals are related to the general vehicle usage and not a specific subsystem. The first type of test is related to general usage of the vehicle (Section 4.1) while the second one analyzes signals to determine changes of the tire pressure (Section 4.2) and the third tests differences between brake scenarios (Section 4.3). The first section will also exemplify the different configurations that are possible with the LU and how the choice of these parameters affects the results of the analysis (Section 4.1.1).

The first test scenario is designed to show how well the PCA method can compress data in the vehicular setting. The first tests are concerned with distinguishing between different general vehicle usages scenarios. The idea here is to see how hard it is to detect which scenario that is executed and how well this difference can be captured by the PCs. The possibility to be able to differentiate between different driving scenarios, through a compressed model, can be useful since less data needs to be recorded despite that the same information is available as if all the signals values had been recorded. This will give an indication of the compression ability of the method.

The second test scenario is designed to give results regarding the sensitivity of the method. The main question to be answered is how small changes can be detected. This is done through observing two closely related signals, vehicle speed and engine speed, which relation can be affected by a configurable parameter. The parameter exists to fine-tune the calculation of vehicle speed to the current wheel circumference, which is affected for instance by wheel dimensions and rear axle ratio. The functionality of the existing system can be used as a tire pressure monitoring system since the tire pressure has impact on the wheel radius and wheel circumference. The resulting sensitivity is related to the requirements that exist on actual tire pressure monitoring systems to give a real life comparison.

The third scenario is chosen to see the capability of the PCA method to detect small changes in a behavior concerning the relation of multiple signals. This scenario is related to the usage of the vehicle, but instead of investigating the general dissimilarity, between different usage scenarios, a specific functionality is in focus. The idea is that it could be easier to detect a true divergent behavior from normal variations, if the relation instead of the individual signals is monitored. The brake functionality is selected for the tests, since the braking behavior is affected by many factors as load, road inclination, speed etc. which makes it complex to define an interval for normal behavior.

4.1 Scenario 1: Monitoring vehicle usage

This chapter presents results regarding calibration of the PCA method and the vehicle usage detection. The calibration also shows results regarding the applicability of PCA in the vehicular setting.

Log files have been collected in four different driving scenarios; city-, highway- , low gear- and uphill-downhill driving. Two of the scenarios, highway and uphill-downhill, have also been done with a loaded swap body. The scenarios have been chosen because they are all part of normal vehicle usage.

1. City driving is characterized by low speed, around 50 km/h and a large number of decelerations and accelerations. The route contained a lot of roundabouts and turns.
2. The highway driving scenario is driving to a highway and then smooth driving with little speed change. This has been done both with and without load.
3. The driving scenario with low gear is also highway driving, but this time with high rpm (Revolutions Per Minute) which means that a lower gear has been chosen for the same speed.
4. The uphill-downhill scenario is done by driving up and down the same hill; it contains deceleration, acceleration, and turning. The same route is driven both with and without load.

4.1.1 Calibration

This section presents results regarding the suitability of using PCA in the vehicular systems. The suitability is shown by first investigating the relations between the signals and continues with investigating what results this gives for compression (Section 4.1.1.1). This gives a basis for the discussion regarding the calibration of the PCA method. In the subsequent sections the effect of sampling frequency (Section 4.1.1.2) and scaling (Section 4.1.1.3) on the observed behavior is investigated.

4.1.1.1 Compression and correlation

The first area to investigate was if the PCA methods requirements concerning the data hold for the vehicular setting. The main requirements are that the sensor data are linearly related and correlated. This is a very important characteristic that the data need to exhibit for PCA to be able to work in this setting, where the goal is to be able to compress the dimensions of the dataset with a high degree of variance in the resulting compression. The reason is that if the correlation between signals is zero then the values in the correlation matrix will be zero. The eigenvalues and vectors are calculated from the correlation matrix, so valuable result is not reached in this case. Section 2.2.3 and 3.2.2 describes more details about the PCA method.

The signals selected for the experiments are linearly related, which Figure 9 and 10 shows since they display the correlation of the signals. How closely correlated the signals were, differed between the different log files and depending on the sampling frequency. This difference is observed when comparing Figure 9 and 10. Figure 9 shows the correlation between the different signals from the log file uphill-downhill driving without load. A correlation of one indicates that the signals are perfectly linearly related, i.e. they form a straight line (see Section 2.2.2, Statistical methods). A correlation of minus one indicates a negative linear relation while zero indicates no linear relation. Note that correlation is symmetric i.e. the correlation of variables X and Y is the same as the correlation of Y and X. Figure 9 shows that vehicle speed, engine speed, and gear are strongly positively related to each other while they have a negative relation to lateral acceleration and steering wheel angle. This is a plausible result since steep steering is usually done when the speed is low. Other signals that are strongly related to vehicle and engine speed are engine torque and accelerator pedal position.

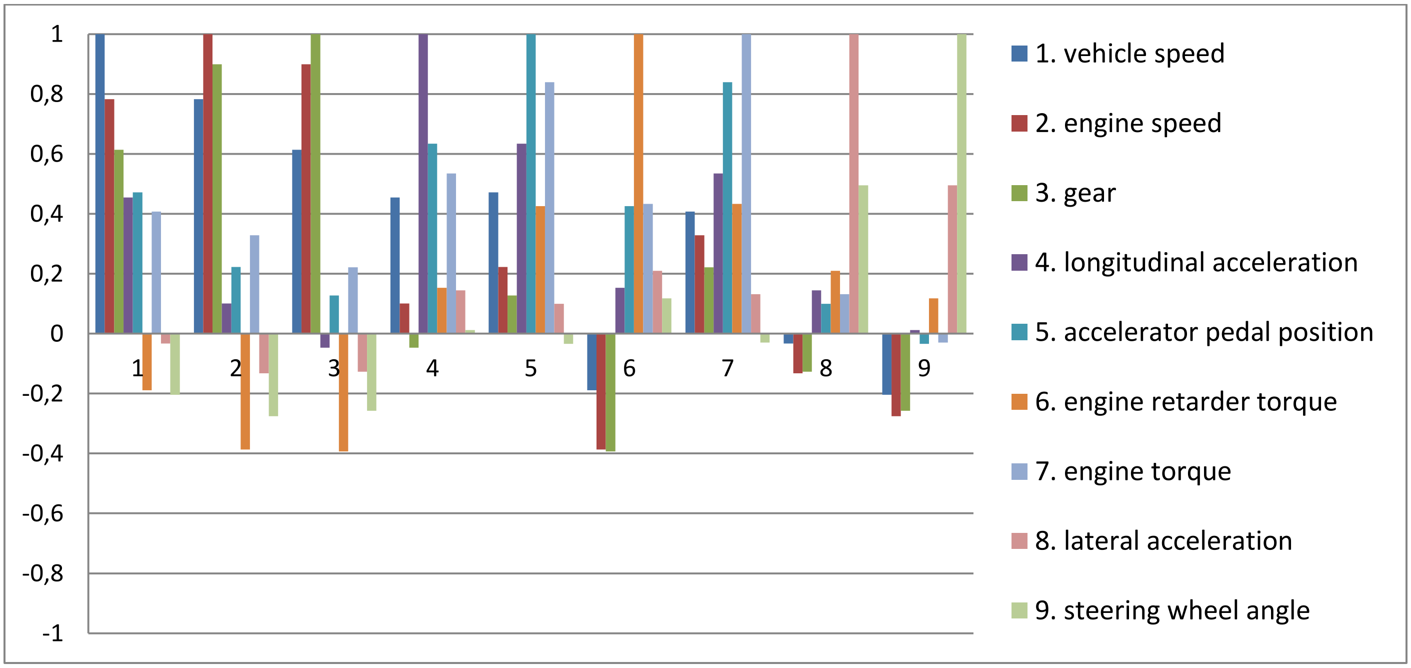

Figure 9. Correlations for 9 different signals from the log file Uphill-downhill without load. The correlation of 1 that all signals have, is the signals relation with itself. Note that the signals vehicle speed (1), engine speed (2), and gear (3) are closely related and have a negative correlation to engine retarder (6), lateral acceleration (8), and steering wheel angle (9).

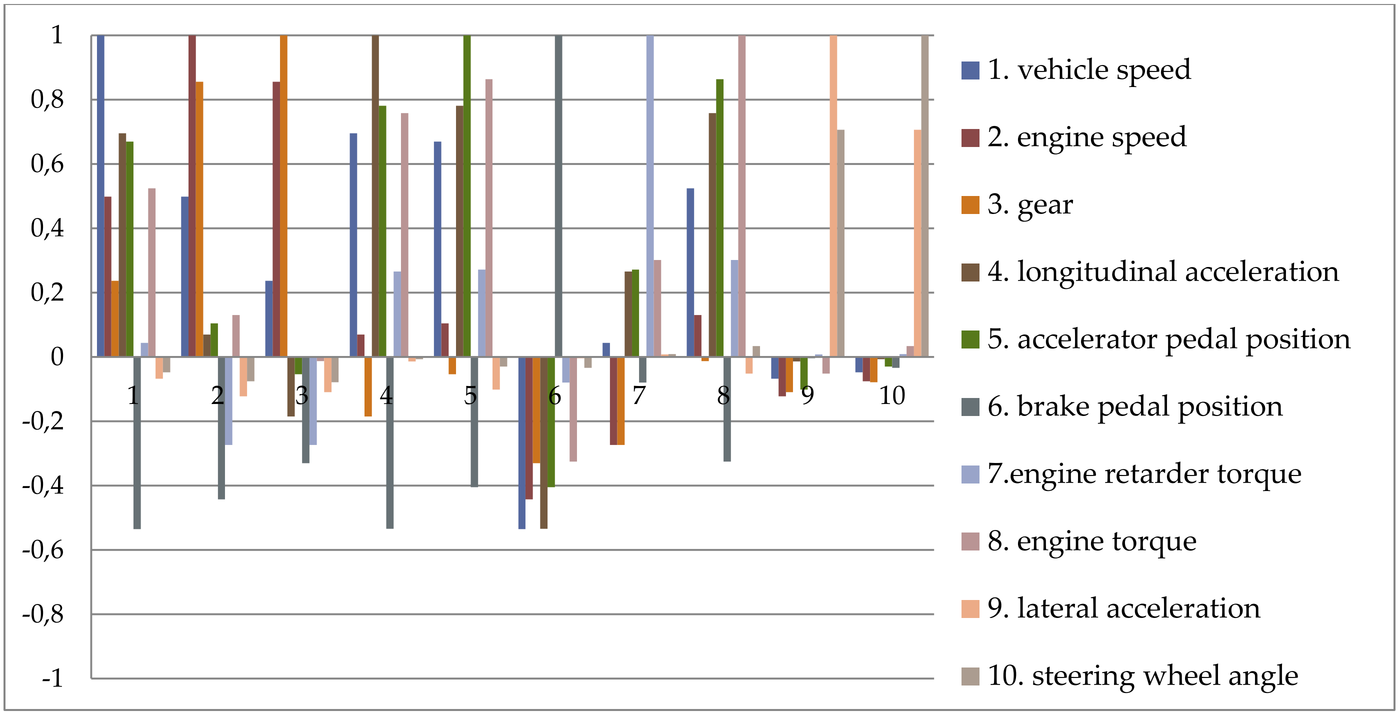

Figure 40. Correlations for 10 different signals for the log file city driving. The correlation of 1 that all signals have, is the signals relation with itself. Note that in comparison with the Figure 9 is the correlations different, but the same main relations are still the same. See for example the signals vehicle speed (1), engine speed (2) and gear (3) where the correlation is still strong but vehicle speed is less closely correlated to the other two.

Figure 10 displays the correlation between the signals for the log file city-driving. In the figure it is clear that the correlation between lateral acceleration (9) and steering wheel angle (10) are more closely related in this scenario than in the uphill-downhill without load. The brake pedal position (6) is negatively correlated to all the other signals. Moreover, vehicle speed (1) is less closely related to gear (3) and engine speed (2) due to many different gears used compared to the uphill-downhill scenario. It is worth noting that the correlation between the different signals differs depending on the driving scenario. This is important since it is this difference in correlation, which enables the distinction between the different vehicle usages. It is also the assumption that the correlation between signals will be different between a faulty and non-faulty system that can give diagnostic capabilities when using the PCA method.

The question of how many PCs are needed to capture a large part of the variance is very important if PCA is to be used for compression. The reason for the importance of the variance is because it can be used as a measure of how much of the characteristic of the dataset is captured by the compression. The variance captured by each PC corresponds to the eigenvalue associated with that eigenvector, see 2.2.3. As can be seen in Table 1, using the first component captures 70-90% of the variance in most cases. Table 1 displays the achieved compression rates. The amount of loss in the compression is equal to the variance that is not captured by the first PC.

Log file	Variance captured by PC1, PC2, PC3	Number of signals	Sampling periodicity (s)	Achieved compression (using score on PC1)
Highway driving with load	97, 3, 0	3	9	33.3%
Highway driving low gear	78.2, 17.5, 4.3	6	1	16.7%
City driving	82, 15.8, 2.2	6	1	16.7%
Uphill without load	93.9, 4.1, 1	7	0.25	14.3%
Uphill-downhill without load	93.8, 4.9, 0.9	7	3	14.3%
Downhill with load	90.5, 4.7, 0.03	7	0.25	14.3%

Table 1. Executions of some log files and the amount of variance capture by their first three components. As can be seen, PC 1 captures between 70-90% of the variance in the datasets. The achieved compression is the amount of variance captured with PC 1, i.e. 97% for the first log file, so in the compressed dataset 3% of the variance will be lost.

The amount of variance captured with a specific compression is the amount of variance captured by the first component. It depends on which signals are used and how closely they are related in the specific log file. The correlation of signals affects the variance captured. It is easy to trace the relation back to the correlation of different signals when studying the variance captured by that group of signals. The more the signals are correlated, the more variance is captured in the first few components. The captured variance is also affected by the sampling frequency since this affects what kind of behavior that can be studied. It is important to note that, even though the PC 1 captures the most dominant relation in the dataset, this is not necessarily the most interesting one. The most interesting characteristic to record and investigate depends heavily on the intended use. As will be seen in Section 4.2, the most interesting characteristic can be the relation between the first and second PCs, or a combination of any of the PCs.

4.1.1.2 Impact of sampling frequency

Several tests, with different sampling frequency, have been executed and with a periodicity ranging from 50 ms to 10 s. The tests indicate that the chosen sampling frequency needs to be of the same scale as the studied behavior.

An example with different sampling frequencies is presented where the log file highway driving with load is used. Figure 11shows the total log file from the driving scenario highway driving with load. Two tests are shown, both using three signals; vehicle speed, engine speed and gear. The signals are chosen because they are closely correlated and easily understood in the context of the driving scenario.

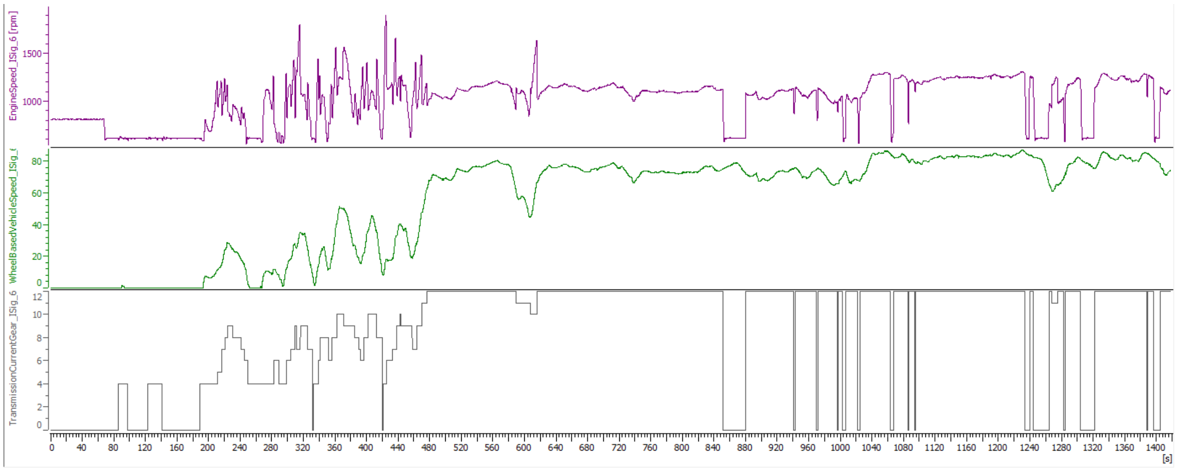

Figure 51. The unprocessed signal values from the log file highway driving with load. The signals are from the top; engine speed, vehicle speed, and gear. Note that the three signals closely follow each other's behavior, begin with a "shaky" phase, and after that are steadier. This is consistent with the driving scenario where the truck starts on a parking place and then is driven with low speed and some stops to a bigger road where a more constant speed is held.

In the first test, sampling is done every sixth second and in the second every ninth second. The characteristics of the dataset are well captured by the first PC when sampling is done every sixth or ninth second. This can be seen from the knee in the eigenvalues plot, see Figure 12, and from that the first PC captures 98% respectively 97% of the variance. Section 2.2.3.1 describes the knee method in more detail.

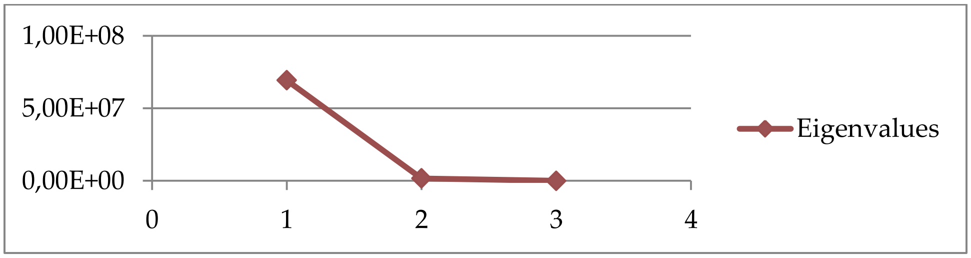

Figure 62. The eigenvalues (Y-axis) corresponding to the PCs (X-axis) for Test 1 and 2. The knee is situated at the second PC.

Figure 13 shows the scores for the first PC for Test 1 and Figure 14 shows the same for Test 2. The same behavior is apparent in both plots despite their difference in sampling frequency. However, viewing for example the varying part around 50 samples shown in Figure 13 and around 40 samples in Figure 14, difference can be seen. In Figure 14, parts of the behavior that is shown in Figure 13 have disappeared, since the sampling is too slow to catch this variation. The dip in the curves around 100 samples in Figure 13 and around 65 samples in Figure 14, is visible in both figures. However, in Figure 14 fewer samples are taken during this variation and with an even slower sampling frequency the behavior would no longer be seen.

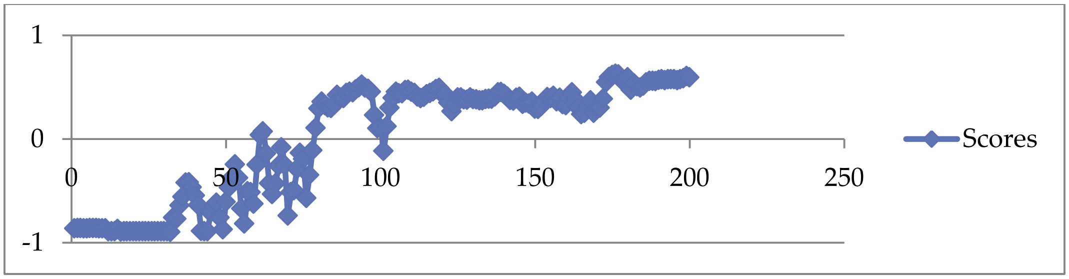

Figure 13. Scores for the first PC in Test 1 with sampling every sixth second. The likeness to the unprocessed signal values, in Figure 11, is clear.

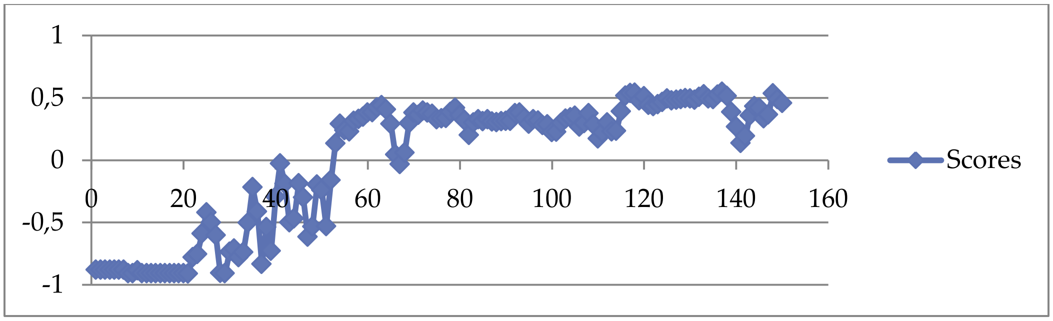

Figure 14. Scores for the first PC in Test 2 with sampling every ninth second. In this test only 150 samples are collected. The highly varying behavior seen around sample 40 is simplified in this figure in comparison to Figure 13.

The sampling frequency affects what behavior can be seen and must be chosen with care for the specific application scenario, to enable the behavior of interest to be accurately displayed.

4.1.1.3 Impact of scaling on observed behavior

The scaling is a very important calibration parameter since it greatly affects the observed behavior. An example with the drive profile with low gear will show how different behavior can be seen depending on the scaling. In these tests the data is scaled according to the mean and standard deviation in each test, to see if a difference can be detected despite that the scaling is relative to the observations during the given test.

Figure 15 shows the driving profile. The signals from the driving scenario are first varying because of acceleration/decelerations and after that they are smooth when the vehicle is on the highway. Seven signals are used in these tests; engine speed, vehicle speed, accelerator pedal position, longitudinal acceleration, gear, engine torque and engine retarder torque. The signals are chosen because they are related to the engine and the assumption is that a difference in load should be detectable by monitoring this set. Three different tests are run on parts of the same log file. In the first test a large part of the log file is used, while in the following two tests only smaller parts are used. This is because different sampling frequency is used and a higher sampling frequency means that a smaller time window will be the basis for the observations. When comparing the scores from the different test the impact that scaling has on the behavior can be seen.

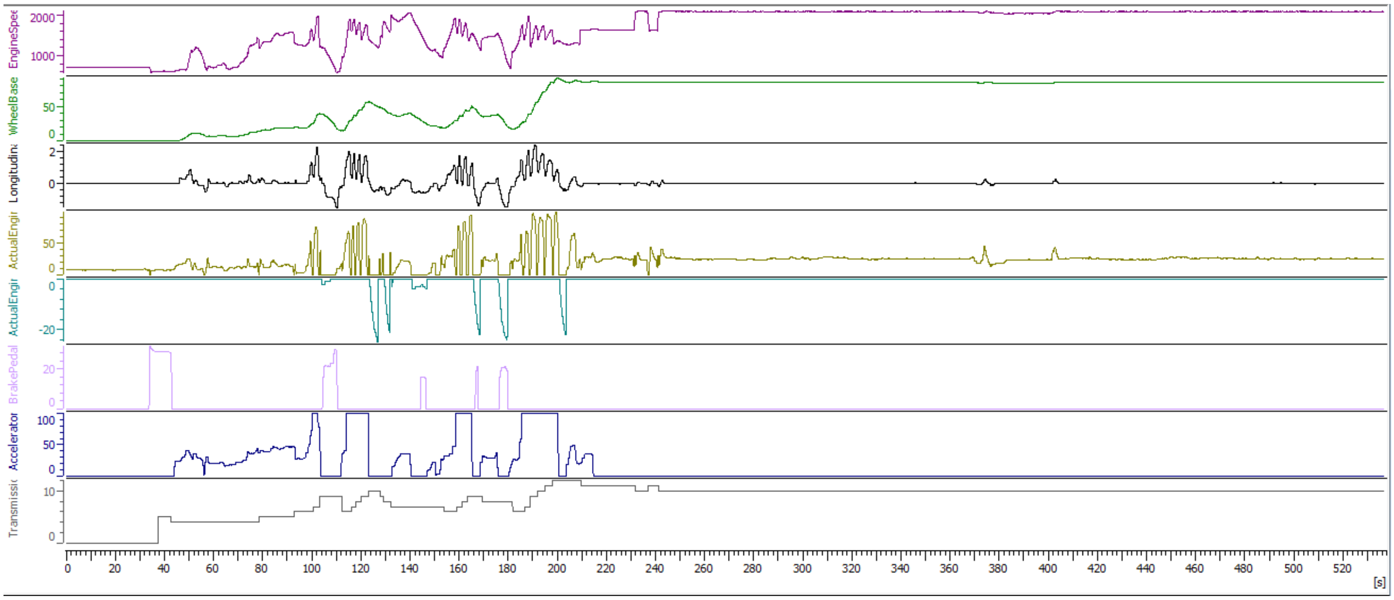

Figure 15. The log file highway driving with low gear. The signals are from the top; engine speed, vehicle speed, longitudinal acceleration, engine torque, engine retarder torque, accelerator pedal position and gear.

Before the score plots are compared, the variance captured in the different tests is presented along with a plot of the eigenvalues. From the plot of the eigenvalues the knee test can be done for deciding how many PCs are needed to capture the measured signals characteristic.

Test 1: In the first test, the log file from the driving scenario with low gear is executed and the samples are taken every third second. The first component captures 94% of the variance which can also be seen in the plot of the eigenvalues versus the PCs, see Figure 16. The knee occurs at the second component, as seen in Figure 16. This indicates that it is acceptable to only use the first PC since it is enough to catch 94% of the variance.

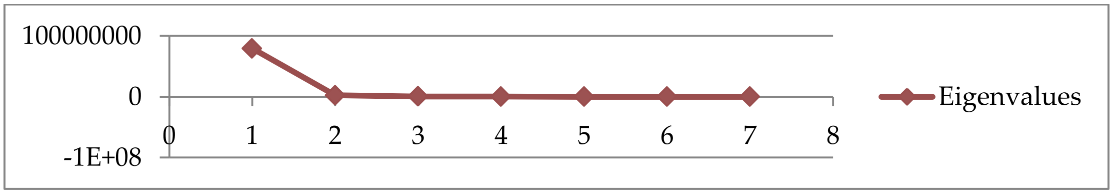

Figure 16. Eigenvalues for the PCs for Test 1. Knee is visible at second PC.

Test 2: In the second test, the recording of the driving scenario with low gear is started when the speed is already high, as can be seen in Figure 17. The highway driving is very smooth with the only exception of the slight decrease down to 83 km/h, which can be seen as a small dip in the speed curves. The period is from time 0 to 200s, the sampling frequency is 1 Hz and 200 samples are taken.

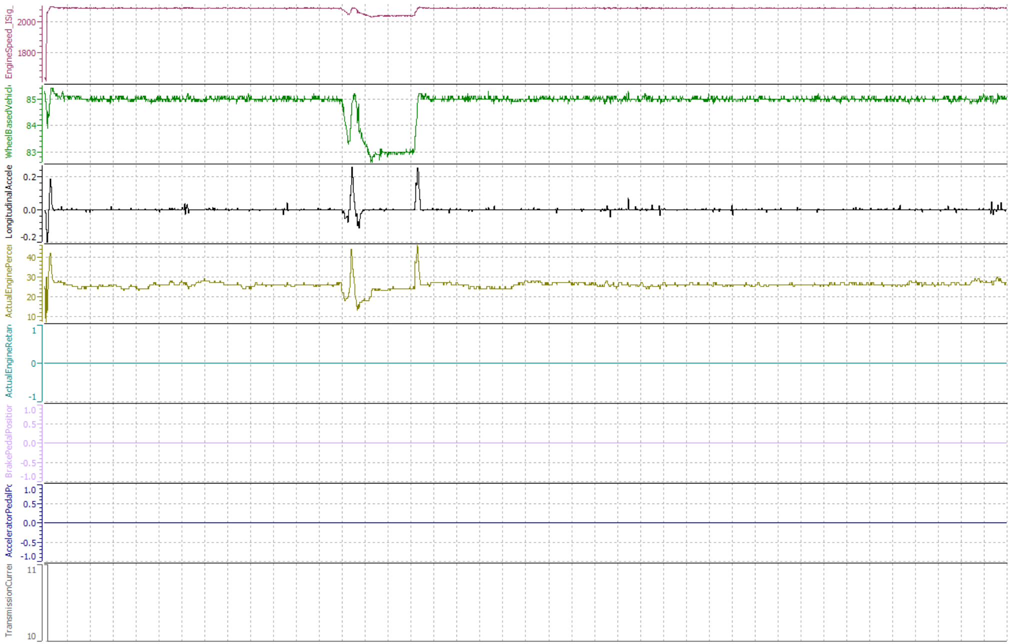

Figure 17. Part of log file from highway driving with low gear. The visible signals are in order from the top; engine speed, vehicle speed, longitudinal acceleration, engine torque, engine retarder torque, accelerator pedal position, and gear.

The first PC captures 78% of the variance, so less than when the sampling was done less frequently. More of the data's characteristic will be lost if only the first component is used. PC2 captures 17.52% of the variance and PC3 4.27%. The other PCs capture 0%. The knee in the eigenvalues plot is not as pronounced as before but is still situated at the second component, see Figure 18.

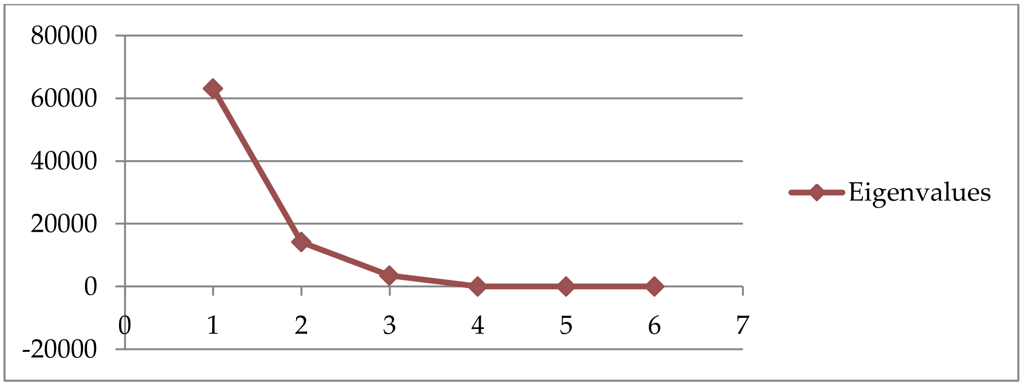

Figure 18. Eigenvalues for the PCs for Test 2. The knee is at the second PC.

Test 3: In test 3 the next part (200-400s) of the log file that can be seen in Figure 17 is sampled with a frequency of 1 Hz and 200 samples are taken. This part of the log file shows a non-changing situation. The first component captures even less of the variance 59% than in both the previous cases. The variance captured by PC2 is 28.51% and by PC3 12.42% while the rest of the PCs captures zero percent. Figure 19 shows the knee plot, where the knee is seen at PC4.

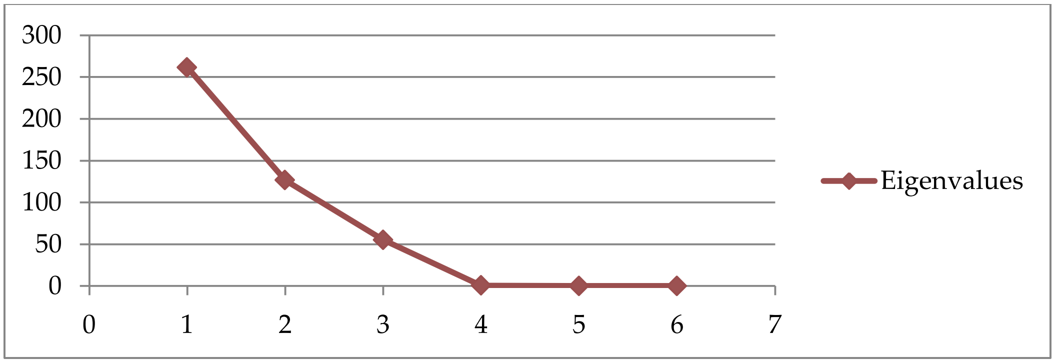

Figure 19. Eigenvalues for Test 3. The knee is between the second and third PC.

Why the situation is different can be explained by comparing the score of the first PC of the different tests, see Figure 20-22. Figure 22 shows a much more varying curve. This is because the data is scaled according to the observations within the same set and no large differences occur in this set, which makes it possible to see the smaller variations in the signals that are not visible in the other tests where larger variations occur. In the score plot for *Test 1*, see Figure 20, it is easy to see the resemblance to the raw signal values.

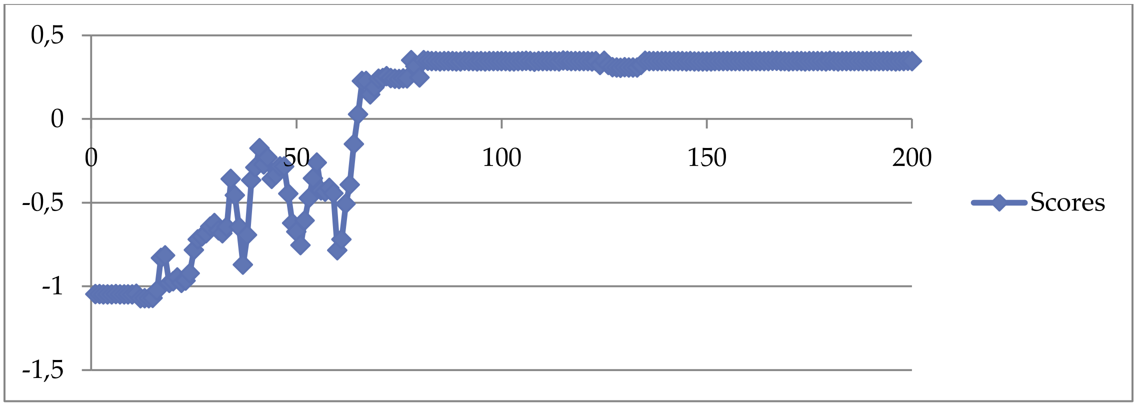

Figure 20. Scores for PC1 from test 1. Note the likeness to the unprocessed signal values.

In the score plot for *Test 2*, see Figure 21, it is easy to see the resemblance to the signal values in that a constant speed has been kept with no drastic changes. The small bump around 150 samples corresponds to the speed decrease at that Figure 17, with the raw data, shows.

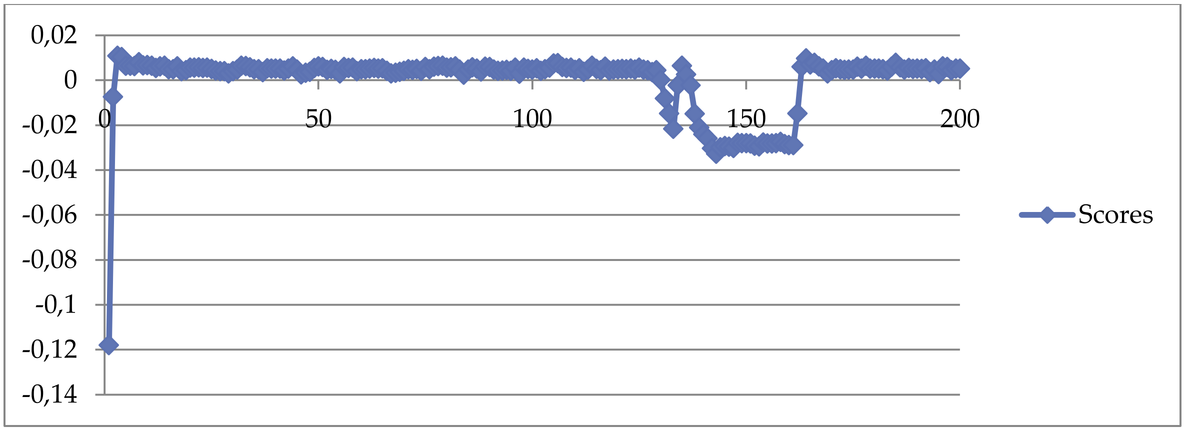

Figure 7. Scores for PC1 for test 2. Note the bump around 150 samples.

Figure 22 shows the score plot from *Test 3*, where a highly fluctuating behavior can be observed. The signal values are more fluctuating because the scaling of the data is done relative to the samples taken in the same test execution, as was mentioned above. Small changes in the captured behavior become enlarged. In this case the correlation for the signals is not as strong as for the more pronounced changes, which could be seen in the eigenvalues.

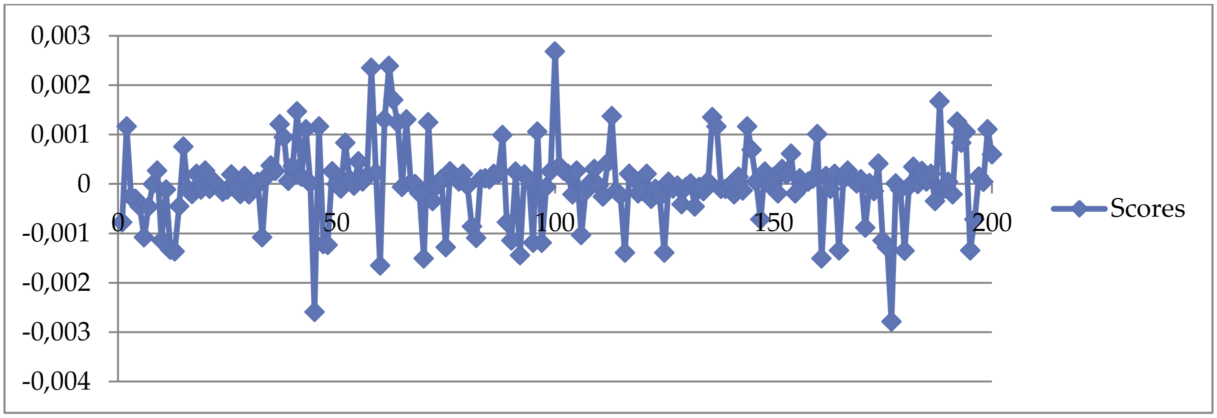

Figure 82. Scores for PC1 for test 3. The observed signal is varying even though the unprocessed signals are very flat, this is due to scaling.

The conclusion from the shown tests results is that the chosen scaling greatly affects which behavior can be studied. When using PCA this needs to be taken into consideration, to avoid incorrect conclusions to be drawn regarding the data and the behavior it describes.

4.1.2 Detection of general vehicle usage

The results from tests with different driving scenarios show that it is easy to detect and differentiate between the scenarios. Tests have been performed with all the four different driving scenarios; city-, highway- , low gear- and uphill-downhill driving. It is easy to differentiate between the scenarios from their first PC but also from the other PCs. Tests with varying sampling periodicity was performed, in the interval from 50 ms to 10 s, and with varying signal setup. The number of signals ranged from 2 to 10 and different configurations were used in the tests with both closely correlated and negatively correlated signals. The results from the other tests is similar to the examples displayed and will not be shown.

4.1.2.1 Detection of driving with different load

To show the capability of detecting different usage scenarios through the relation between signals, a test where the same route has been driven with and without load is presented. The significance of the test is that it is possible to see the difference of driving with and without load due to the impact this factor has on the other signals. Difference can be detected in the relation of the signals and not on the unprocessed signal values. The ability to detect the effect of changing conditions on the signals relation can be useful, as an example, for a brake performance monitoring system as is described in Section 4.3.

The goal is to see if any difference can be detected between driving up and down a hill with and without load, since these are two quite closely related scenarios. Sampling is done with a periodicity of 3 seconds and 200 samples are collected. The tests were executed with scaling done according to the mean and standard deviation in each sampling period, i.e. they were different for the two log files. It was also tested to use the same mean and standard deviation for scaling but it did not noticeably affect the result. The signals used are; engine speed, vehicle speed, accelerator pedal position, brake pedal position, longitudinal acceleration, gear, engine torque and engine retarder torque. The driving profiles can be seen below in Figure 23 and 24 respectively. The route is up and down the same hill, with an abrupt deceleration and turning between the runs. The driving profiles are displayed to exemplify how hard it is to see the difference between the two test cases by inspecting all the unprocessed signal values and to show how alike the two test runs were.

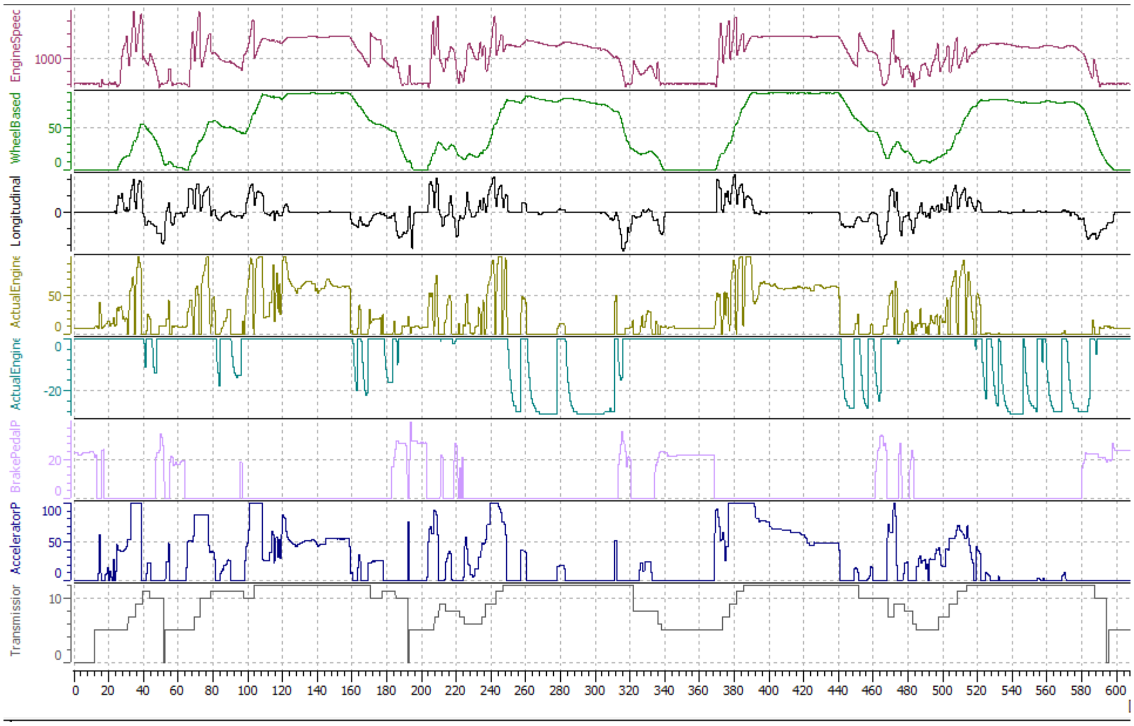

Figure 23. Unprocessed signal values for the log file downhill-uphill without load. The visible signals are in order from the top; engine speed, vehicle speed, longitudinal acceleration, engine torque, engine retarder torque, brake pedal position, accelerator pedal position, and gear.

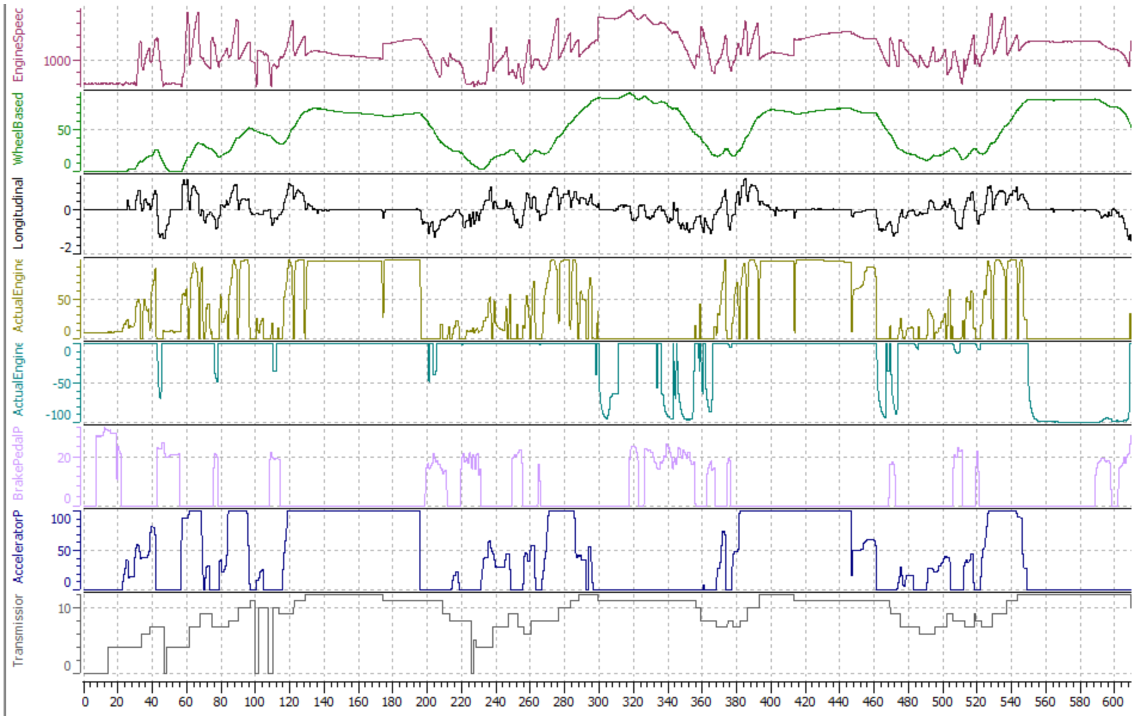

Figure 24. Unprocessed signal values for the log file uphill-downhill with load. The signals are in order from the top; engine speed, vehicle speed, longitudinal acceleration, engine torque, engine retarder torque, brake pedal position, accelerator pedal position, and gear.

In Figure 25 the score on PC1 with and without load can be seen. The observations related to driving in the slope are the four peaks above the X-axis, where the first and third are uphill, and the second and fourth are downhill. When driving up the slope is the plot for the truck without load above the one with load and vice versa in the downhill drive. Difference between the two driving scenarios can be detected by the exhibited behavior in the uphill-downhill setting. The score for PC1 is nearly identical regardless of if only vehicle and engine speed are used or if 8 signals are used as below. That is because the relation that these signals show is dominant in the dataset, if other relations are of interest the other PCs need to be investigated.

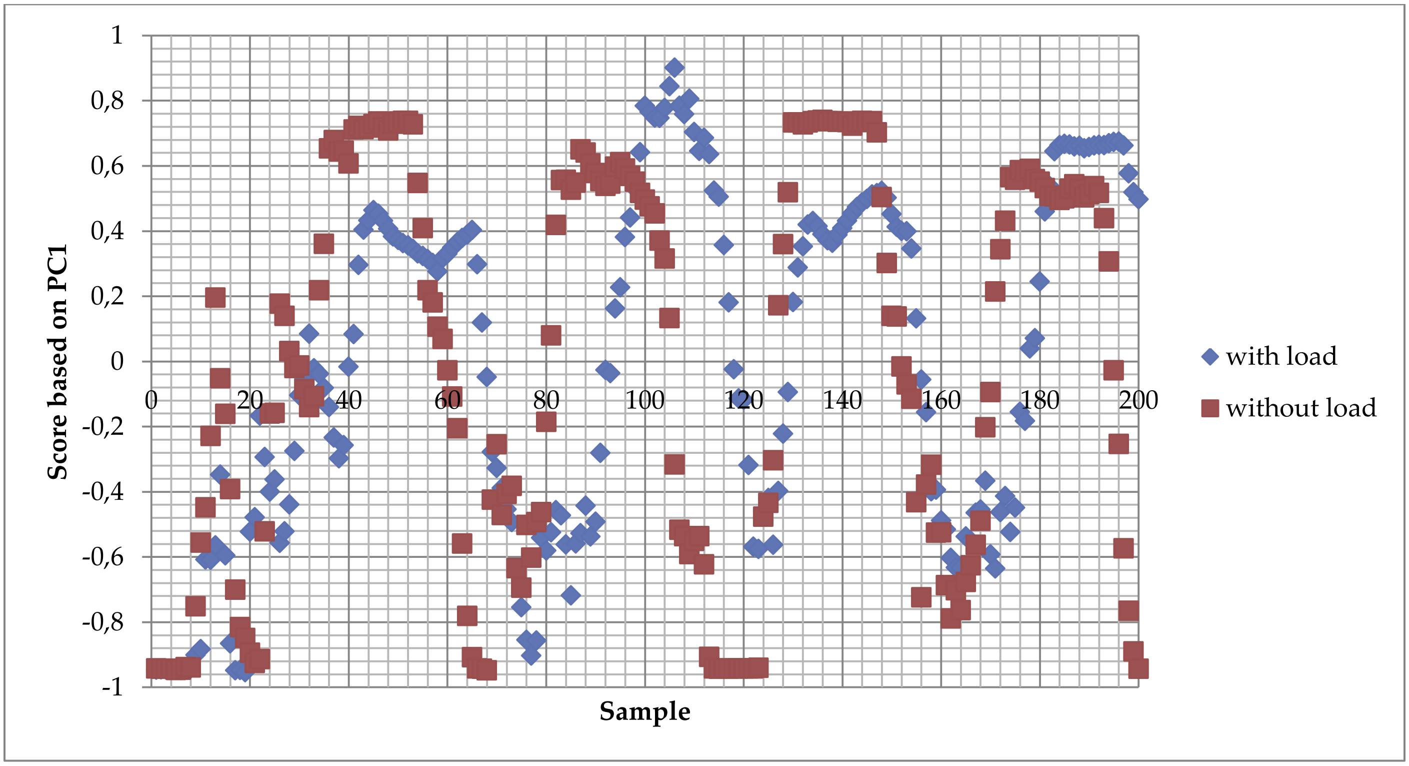

Figure 25. Score plots for uphill-downhill driving with and without load. The first uphill driving is around the 50th sample, then a turn is made and the first downhill drive is around the 100th sample. The second uphill drive is around the 135th sample, and downhill is around 185th sample. The observations below the X-axis are related to turning and driving to the slope, not in it. The importance in the figure is the four peaks above the X-axis, and the difference they show between the drive with and without load.

To study the detected difference more closely only the downhill route was compared between the different scenarios. In addition, here a significant difference is clearly detected. In Figure 26 the scores from the first component can be seen.

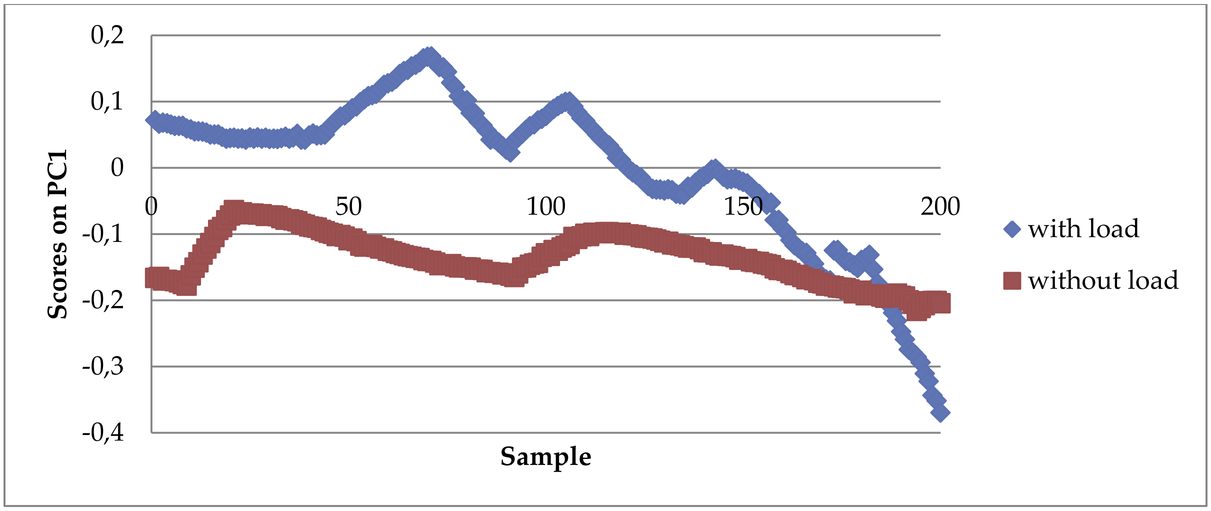

Figure 26. Scores for downhill-driving with and without load. Note the difference between the scenario with and without load.

The difference between the scenario with and without load is clear already on the first component but the other PCs have also been investigated. As was described in Section 2.2.3.1 can patterns be seen in all the different PCs, and the most atypical relation for the dataset will be captured by the least component, the PC with the highest number. The difference between the scenario with and without load is clear in all the PCs, see Figure 27 and 28 for examples. In Figure 27 the scores on the second component are plotted against their sample number and a clear difference between the scenarios can be seen.

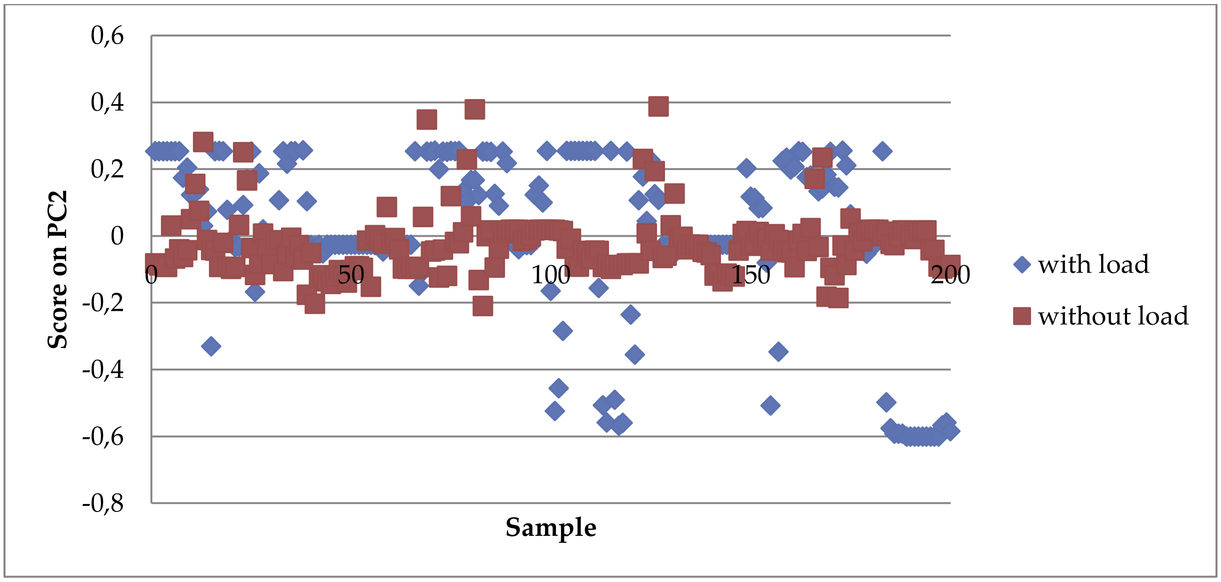

Figure 27. Score on PC2 on the Y-axis, sample number along the X-axis. The difference between the behaviors in the two test cases can clearly be seen.

In Figure 28 the scores on PC6 and 7 are plotted and also here a clear difference can be seen between the two scenarios.

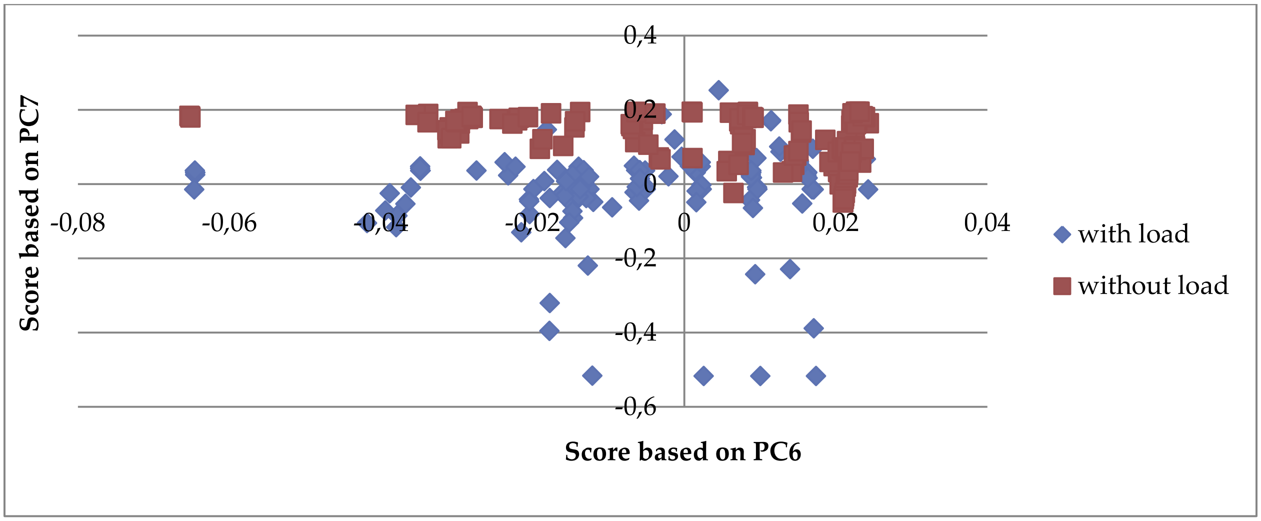

Figure 28. Score on PC6 and 7 on X and Y axis. Difference between the two test cases for this type of visualization can be seen.

What is clear from looking at the scores on a few of the PCs is that information that could be of value is contained there. There is a possibility that the difference between the scenario with and without load is best described with any combination of the PCs. The complication with analyzing the less significant PCs is that it is more difficult to be certain what pattern is described. The characteristic contained in the first PC is the easiest to guess, since that is the most dominant relation in the dataset. For the different relations to be identified, tests need to be run where all aspects of the truck behavior are strictly controlled so that only one variable is changed at a time.

4.1.3 Summary of vehicle usage tests (Scenario 1)

The signals in the studied vehicle application are highly correlated and linearly related. This characteristic enables PCA to give good results with regard to compression. Compression to 15-30% of the original dataset contains 70-90% of the variance. It has been shown in the tests that scaling of the datasets has large impact on the perspective and on what behavior that can be seen. The choice of the time window for observation, the combination of how many samples are taken and the sampling frequency, have also large impact on what is seen. The signals that have been used here are easy to describe with few PCs and still have a high degree of the variance.

The configurability of the developed LU is an important prerequisite for the tests. By using different configurations different behavior can be seen, as shown in Section 4.1.1.2, 4.1.1.3 and 4.1.2.1. Since which signals are used can be changed can we investigate different systems and relations that the set of signals are related to. By changing the sampling frequency can a choice be made regarding which behavior to be investigated, for example, faster sampling for volatile behavior while a slower rate can be used for long term monitoring of slow changes. The effort to calibrate the PCA for the specific scenarios is an opportunity with regard to new possible application but it also requires effort which could be time consuming.

The results of the performed tests showed that identification of different vehicle usage patterns and differentiation between the driving scenarios was obtained. Even scenarios that are closely related, as uphill-downhill driving with and without load, are easy to differentiate between. Furthermore, all the PCs contain information which can be useful for understanding of the signals and the system they describe. The difference between the closely related driving scenarios of driving up and down the same slope with varying load can be seen in the scores from PC1-7. The relation or pattern described by the different PCs is hard to map to the existing knowledge of vehicular systems without further tests. To perform these tests all signals need to be strictly controlled to ascertain which signal change corresponds to which change in relation. That was not possible to do within the scope of this project.

It is however promising that difference can be detected between the different vehicle usage scenarios and a similar analysis could also detect differences between driver styles and other factors. To further investigate the possibilities and limitation of the PCA method more tests are needed. Next, results from tests focusing on investigating how small changes can be detected, in the relation between signals, are described.

4.2 Scenario 2: Monitoring tire pressure

A series of tests were performed to investigate the sensitivity of the PCA method. The sensitivity is an important parameter for evaluation since low sensitivity limits the usefulness of the method. The idea was to see how small change that could be detected in the relation between two signals. The found sensitivity limits are also applicable to other signals relations.

The aim of the scenario is to monitor the tire pressure using two signals, vehicle speed and engine speed. A lower tire pressure is simulated by the use of a parameter which value is based on the tire radius. The basis for the test is that a decreased tire pressure means a decreased tire radius, which will affect the calculated vehicle speed. The reason for the existence of the parameter is that the calculation of vehicle speed needs to be fine-tuned with the rear axle ratio and the wheel dimensions to be accurate. The different values of the parameter correspond to different wheel dimensions because the parameter is calculated from the wheel circumference. The relation between engine speed and vehicle speed is affected by the parameter and it is this difference that is monitored. The parameter has its normal value as 5386 for the specific truck and tires used for the tests. In the different test runs the parameter is decreased in steps by 1, 5, 10, 25, 50, and 100 from 5386, see Table 2.

Parameter value	Parameter difference	Wheel radius difference (mm)
5386	0	0
5385	1	1
5381	5	4.5
5375	10	9.5
5361	25	23
5336	50	46.5
5286	100	94

Table 2, Parameter and wheel radius difference for the values of the parameter.

The idea of the tests is that if it is possible to detect changes of the parameter it is also possible to detect the corresponding change of the wheel radius. The possibility to detect changes of the wheel radius can be used to detect if the tire pressure has dropped due to air leakage, which is the task of a tire pressure monitoring system.

The log files are collected from a driving scenario at a low speed and few turns. The data was scaled with the same mean and standard deviation, to enable comparison between the different tests. 200 samples were collected with a periodicity of one per second.

In Figure 29 the score plot for the first two PCs can be seen. The largest change of the parameter (5286) is seen below the others. It is possible to see that already a parameter change of 10 (5376) is visually detectable, which corresponds to a radius change of 9.5 mm. Similar appearance was seen with both correlation and covariance based PCA.

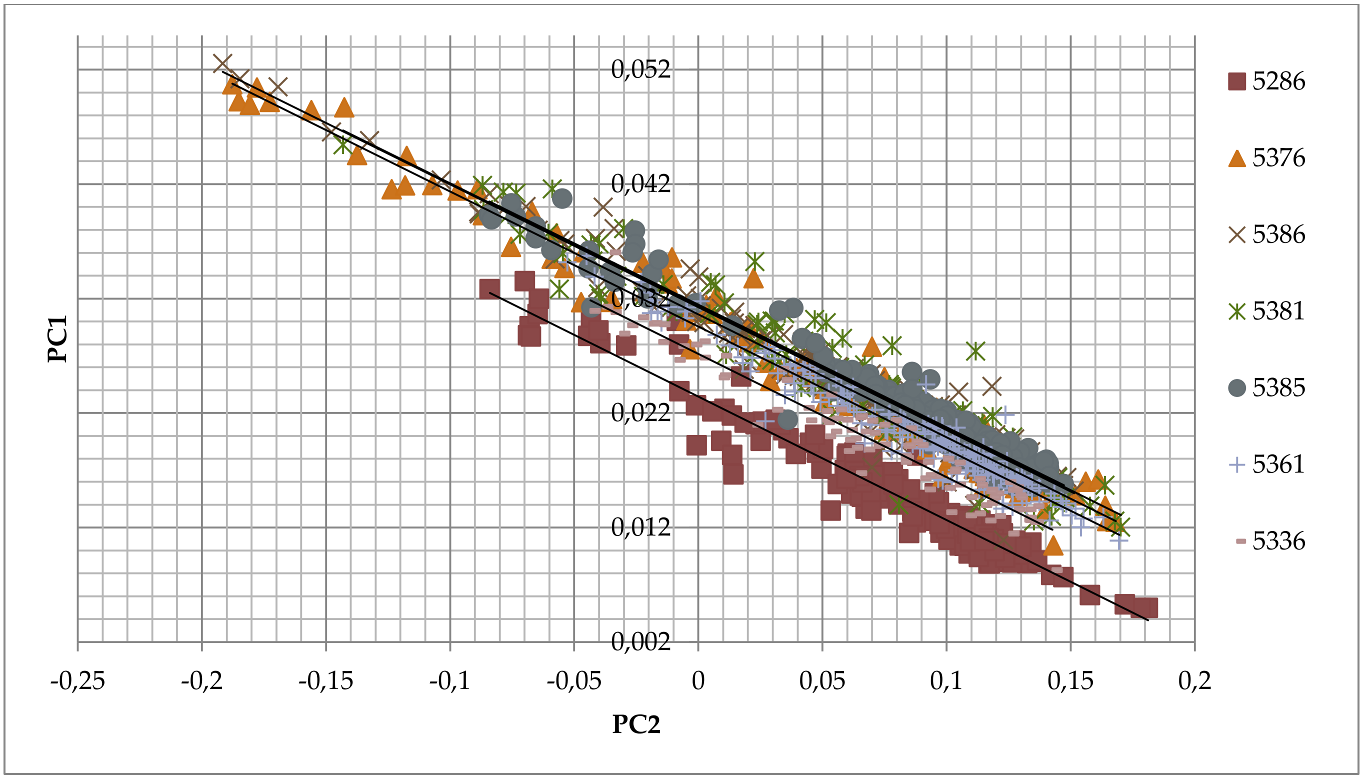

Figure 29. The scores for the different parameter values. Linear trend lines are also plotted. Note that the tests with different parameter value lies in order with the test with lowest parameter value at the bottom.

Compare Figure 29 to Figure 30, where the unprocessed values of engine and vehicle speed are plotted. Here, any differences between data with parameter values that differ with less than 100 are very hard to see. The black lines in Figure 29 and 30 are the linear approximations of the datasets.

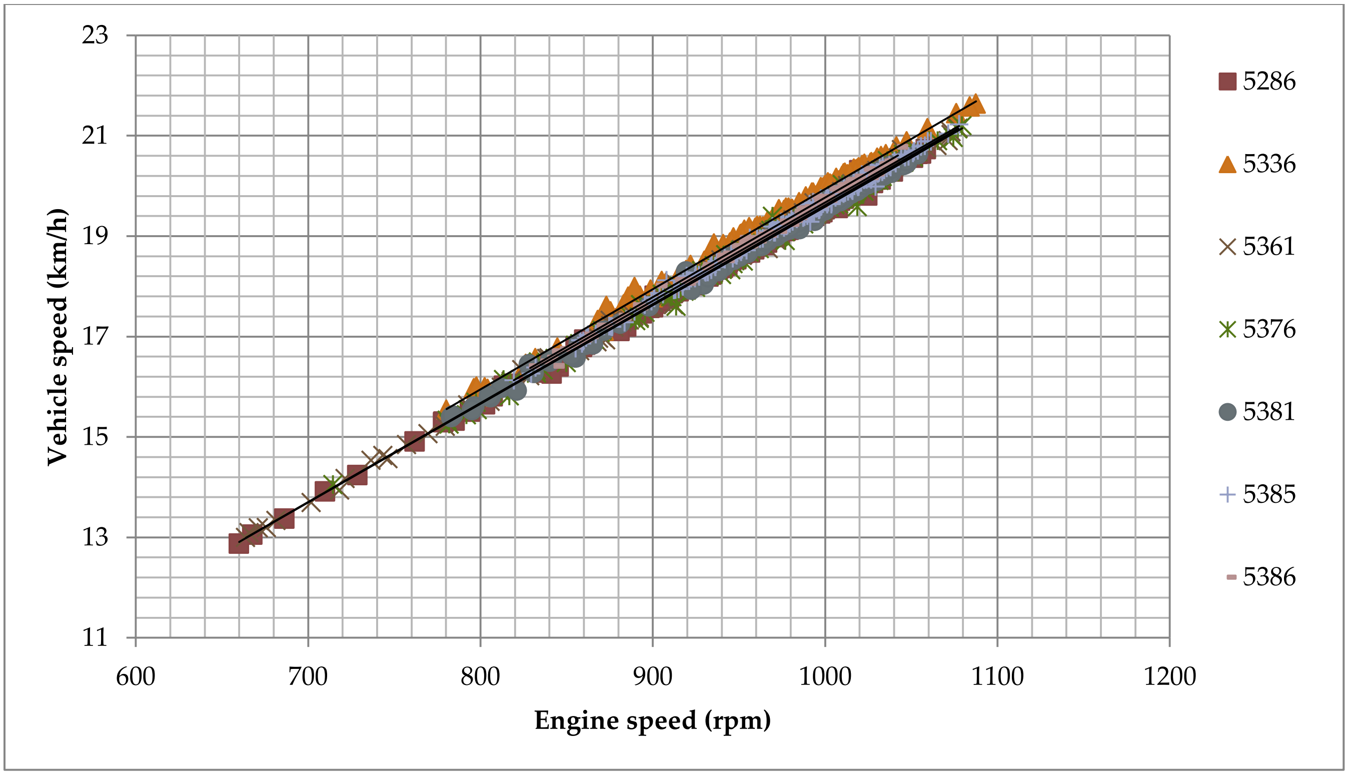

Figure 30. The raw values of engine and vehicle speed are plotted. Black trend lines are also plotted. Note that a the line at the top is for the test with parameter value of 5286 but that the other tests with different parameter values are hard to differentiate between.

To check the reliability of the models, and specifically of the detectable differences, the confidence limits on the linear regression models were calculated. The gradient of the linear regression models for the different values of the parameter are approximately the same for the PCA scores. The lines dissimilarity was calculated and the results can be seen in Figure 31. The confidence interval for a normal distribution with 95% confidence limit was calculated for the values seen in Figure 31 and 32. The confidence interval is between 0.000341 and 0.000664 and is too small to affect the result. This result ensures that it actually is detectable with a parameter change of 10, i.e. parameter 5376, which correspond to a radius change of 9.5 mm.

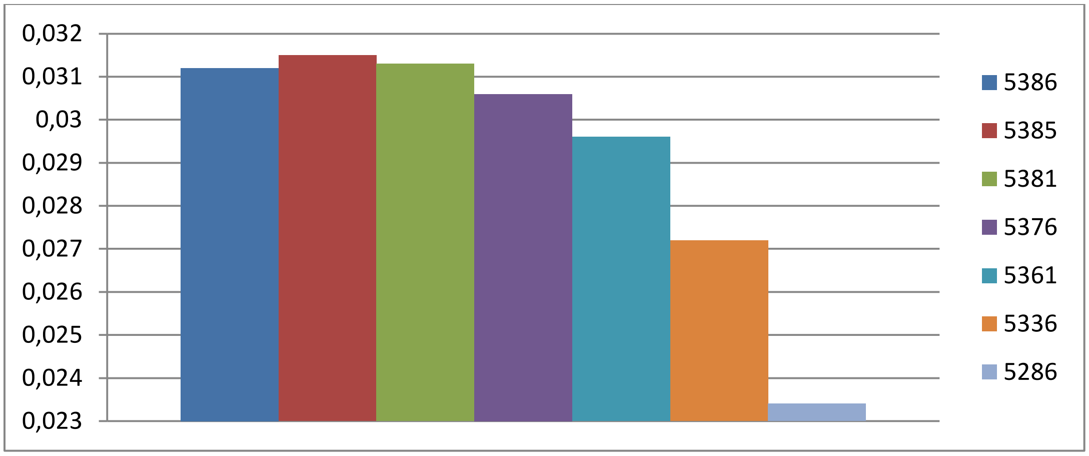

Figure 31. Crossing of the Y-axis for the linear models of tests with different parameter value. The first significant difference is for parameter value 5376.

For comparison was also the unprocessed signals linear regression models calculated. The result can be seen in Figure 32, where it is clear that the smallest detectable difference with unprocessed signals is parameter value of 5286. The parameter value of 5286 corresponds to a radius difference of 94 mm. This means that the PCA score give ten times better sensitivity than the unprocessed signals.

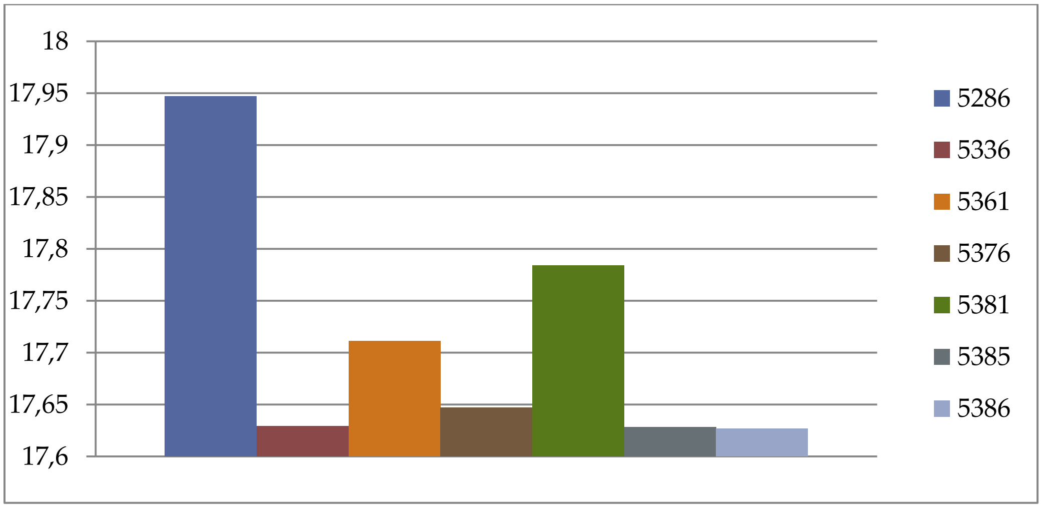

Figure 32. Crossing of Y-axis for the unprocessed signal values. Note that the parameter value 5286 is distinguishable from the other values but that no more values are.

To put the found sensitivity in contrast to a real tire pressure monitoring system, the air pressure drop corresponding to a tire indentation of 9.5 mm in radius was investigated. At nominal atmospheric pressure and axle load of 7.5 ton, 10 mm change in radius corresponds to a pressure alteration of 1-1.5 bar when tire type 315/70 is used [62]. The tire indentation is progressive which means that the radius change increases more with further pressure loss. Tire pressure in a truck is normally 9 bar. When the tire pressure originally is 9 bar a 1-1.5 bar change corresponds to a drop of pressure with 11.1-16.7%. In [63] tire pressure drop of 25% gives a warning to the driver from the indirect tire pressure monitoring system (TPMS). This gives an indication that the shown sensitivity of the PCA analysis makes it useful for the tire monitoring scenario.

4.2.1 Summary of tire pressure monitoring tests (Scenario 2)

The conclusion from the test is that using the scores on the PCs, is it possible to see small changes. A radius change of 9.5 mm was clearly detected, which correspond to a parameter change of 10. The change was validated by considering a linear regression and calculation of the 95% confidence interval. Comparison was also performed to the unprocessed signals. It is shown that the PCA method exhibit one magnitude greater sensitivity than the unprocessed signals. The difference between the parameter values can best be seen by plotting the scores for the first two components against each other. Plotting scores for different PCs is also discussed in 4.1.2. This is an adaptation of the analysis, which indicates that the PCA method needs calibration to exhibit its full potential. The adaptation of the method to the specific usage scenario takes time and creativity.

4.3 Scenario 3: Monitoring brake performance

Monitoring the relations regarding a specific functionality instead of the individual signals is one interesting application of the analysis method. The idea could be applied to different functionalities where the normal behavior is affected by several factors, and the factors relation could be monitored.

Since the brakes functionality is of great importance for the operation of the vehicle, are they of interest to monitor. Brake monitoring is complicated by that the normal case depends on many factors. These conditions are for example the load of the vehicle and road inclination. When monitoring the brakes very small changes are of interest, since that can indicate a fault.

Tests are executed to determine if the PCA method could be used for detecting deviations related to the brakes performance. Different brake scenarios were performed to determine how well the method captured the pattern and deviations. Signals used were brake pedal position, vehicle speed and longitudinal acceleration. The first test was to differentiate between braking on flat ground and in a slope with 5% inclination. Figure 33 and 34 shows the results. Figure 33 shows the characteristic of the braking maneuver. The two brake maneuvers in the slope differ from the ones done on flat ground. Figure 34 shows the score based on PC1 and PC3 and here the difference is slightly more pronounced.

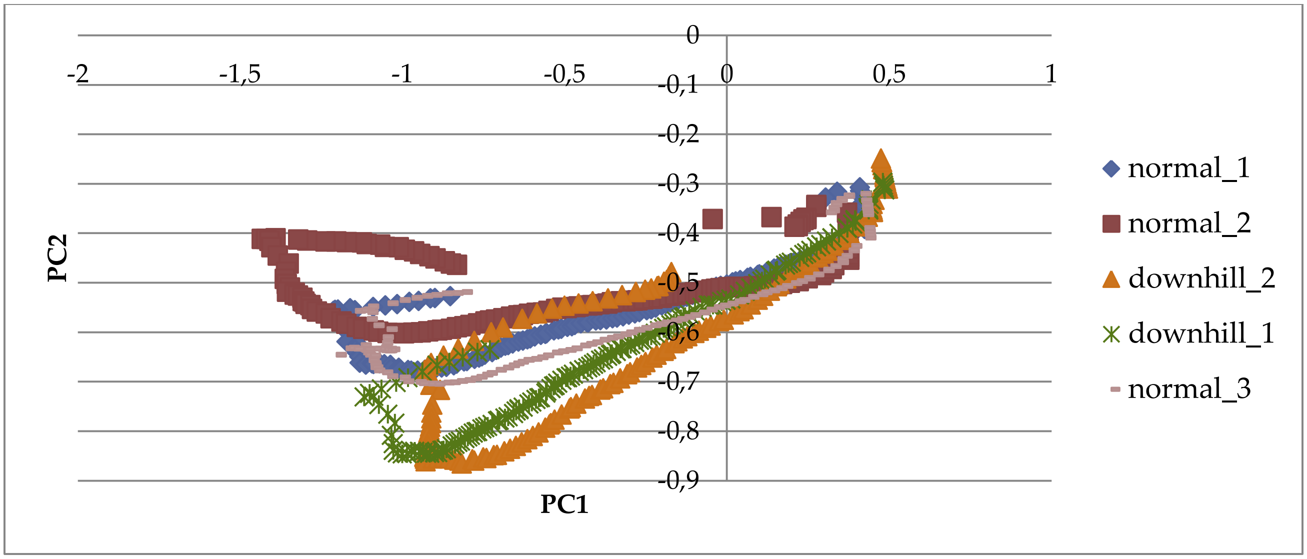

Figure 33. Score on PC1 and PC2 from braking on flat ground and in a slope with 5% inclination. The braking scenario has a distinct form that is similar in the different tests and the results from braking on sloping ground are positioned in the bottom of the plot.

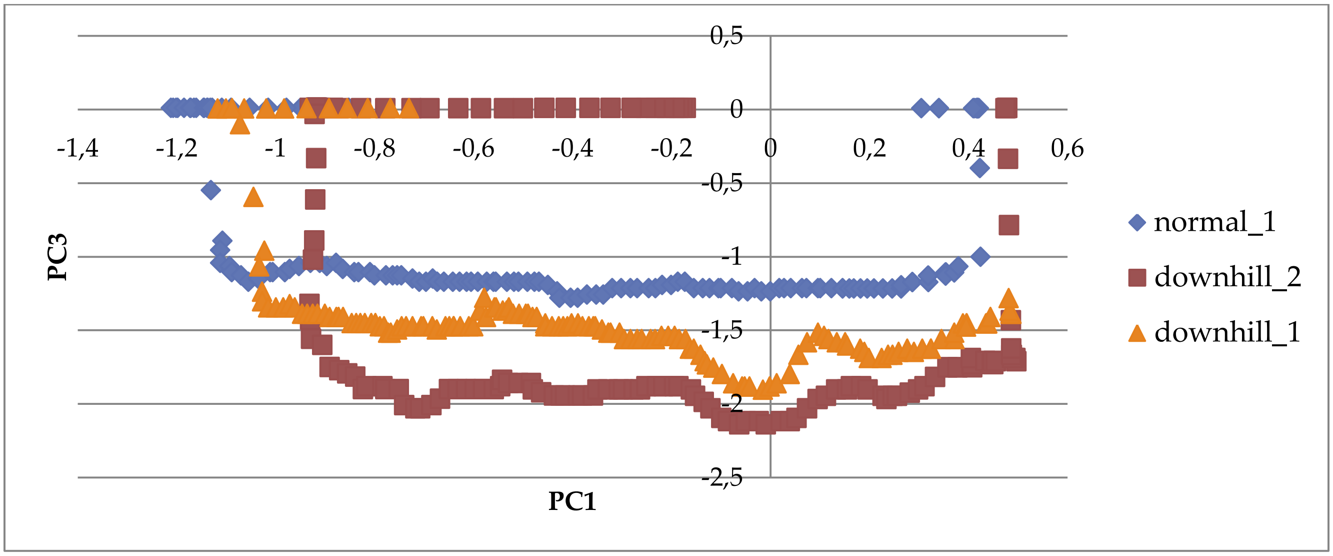

Figure 34. Score from PC1 and PC3 on flat and sloping ground. Only three signals are displayed to not crowd the figure. The results from braking in a slope are visible below the case with braking on flat ground.

To proceed with the method more tests need to be performed where braking on different slopes is tested with different loads. From these tests, sensitivity limits and calibration of the method can be determined. For example, it could be determined if the relation between scores on first and third PC, that is shown to be promising in Figure 34, is the best way of detecting brake differences. Further tests were not possible in this work due to time constraints.

4.3.1 Summary of brake performance tests (Scenario 3)

The LU can detect difference between the braking scenarios. The performed tests differentiate between braking on flat or sloping ground with an inclination of 5%. The next step is to perform further tests to see the full extent of the usefulness of the method. The intended use case of brake monitoring is promising but needs more tests to determine sensitivity of the method in this application.

Since promising results were obtained for monitoring the brakes, through the signals relations, also other functionality, as the traction control systems, could be monitored with a similar approach. Of course, further tests need to be applied to investigate and to calibrate the analysis in the applications.

5 Summary and discussion

The configurability of the developed PCA method is important for the usefulness of the method since it enables analysis of different behaviors. Due to the configurability is diverse analyses performed.

Three different scenarios is the basis of the evaluation. The first scenario investigates detection of different usage of the vehicles. The results from this scenario are that difference in usage is easy to detect using the PCA analysis. Tests also investigated the calibration of the method. Results have shown that the signals are highly correlated but the degree of correlation differs depending on the scenario and sampling frequency. The correlation is easy to understand from a basic knowledge of the vehicular system. Tests also shows that the scaling is importance for receiving accurate results, since it affects which behavior is easy to see.

The second scenario is concerning monitoring of the tire pressure. The results from this scenario show that the analysis is sensitive to small changes. The PCA showed detection of a change with sensitivity ten times higher than the unprocessed signals. In the tests, it was detectable with a change of radius of the tire with 10 mm, which corresponds to a pressure loss of 1-1.5 bar. The shown sensitivity is within the bounds of an indirect TPMS.

The third scenario is regarding monitoring of the brakes. The results from the tests show that it is possible to detect the difference on the relation of the signals related to braking. In the tests, it was possible to detect a difference between barking on flat or sloping ground with an inclination of 5%. These results can be the basis of further tests focusing on monitoring the relation between signals since a defective functionality could be easier detected through the relation than the unprocessed signals values.

The main finding in the tests is that the configurable PCA method can be adapted to suit different types of analyses, in line with the presented scenarios, depending on the intended application.

5.1 Thesis goal attainment

Main objectives of this work have been to investigate the usefulness of the proposed PCA approach for the vehicle application, its configurability, and resource demand. This thesis work shows results in two main areas; implementation of a configurable LU with PCA and test of what results the analysis can give in an automotive setting.

Basis for the evaluation of the usefulness of the LU is the results from the compression and PCA analysis. The prototype performed well with regard to compression in the tests that were carried out, 70-90% of the variance was captured in the score from the first component. A more general conclusion from these tests is hard to make since the correlation between the variables is the foundation of how good the compression becomes. The investigated signals are highly correlated but the correlation varied between different scenarios and is also dependent on the sampling frequency and scaling. It was possible to use the PCA method to get more information as was done in the case of detecting varying tire pressures (Section 4.2). It is in this area that the analysis could give new and valuable information. The feasibility of more sophisticated logging solutions to extract the relevant information, useful for monitoring, is strengthened. However, to take fully advantage of this new functionality more work needs to be performed to explore how to best use the possibilities in the vehicle. The results from the calibration section (Section 4.1) indicate the importance of calibration. This is closely related to one of the basic truths of data analysis; to get valuable results you need to understand your data, i.e. understand the application setting, and make relevant assumptions based on that knowledge.

Configurability is an important feature for the proposed LU. The completed implementation shows that the chosen level of configurability was possible to create and is vital for enabling the analysis to be modified to the different scenarios. The following parameters are configurable; a) which signals to investigate, b) how frequently they are sampled, c) how many samples is the basis of the analysis, and d) how many PCs that are of interest. The importance of the configuration for the analysis is found to be large, as previously discussed. The configurable nature of the LU also makes the logging functions less reliant on other functionality for testing and updates, which removes an unnecessary dependability.

An important aspect of the problem to solve was the limited resources available for on-board vehicle logging. In this area the results have shown that the PCA method gives an accurate compressed version of the data. The PCA decreases the dimensionality of the data as can be seen in the evaluation, Section 4. A low dimensional summary of the data's characteristics is rendered, where as much as possible of the variation in the original dataset is still represented. Section 4.1.1.1 presents results where a compression to 15-30%, of the original dataset, contains 70-90% of the variance. Memory demand is lower since the analysis creates a low-dimensional representation of the data. Nevertheless, since the memory demand is highly dependent of what the component should be used to log information about, can both higher and lower compression rates than shown, be achieved. The analysis method uses a lot of floating point calculations, which will affect the selection of HW. Because of time and resource limitations, was the prototype unit not tested on HW, which makes the conclusions with regard to resource demand incomplete.

The limitations for the proposed unit are the same as the limitation for the analysis method. PCA cannot analyze non-linear relations, or data that are not related. How significant limitation this is for the method to be useful in vehicles is not clear, indications from the tests are that the studied signals were correlated and often closely so. Another limitation on the type of scenario that the method can analyze is that the behavior of interest must be distinguishable from other behavior. This is mostly a calibration problem, finding a setting that separates between the behavior of interest and other behavior. However, if no calibration can be found which makes the behavior of interest marked, can this behavior not be analyzed by the method. Furthermore, there can be a high workload to calibrate

• • •

the unit to find useful results. It is not hard to find any relation in the data, but domain specific knowledge is needed to know if the found relation could be advantageously used.

6 Conclusions and future work

This thesis work demonstrates that a well-functioning configurable on-board vehicle data LU that uses PCA is feasible. The developed prototype LU performs well with respect to compression and produce useful analysis results. Tests showed that the signals chosen for this study were highly correlated and linearly interrelated. Compressions that contained 15-30% of the original dataset could capture 70-90% of the variance. The method also showed capability of detecting small changes. Tests monitoring the change in tire pressure and braking performance, investigated the sensitivity of the method. In the case of changing tire pressure ten times higher detection sensitivity compared to the unprocessed signals was found. While in the case of monitoring breaking performance a difference between braking on flat and sloping ground, with 5% inclination, could be detected. Finally, tests were performed that show the importance of correct calibration of the PCA for obtaining valid results.

The next step would be to run the created software unit on actual HW. This would give the opportunity to determine the resource usage in detail. Another area of further research is to find more situations where PCA can give valuable results. In this project, tests were conducted where the analysis gave useful results for monitoring tire pressure, brake performance, and vehicle usage, but there are many more settings in the vehicle where the analysis is likely to be applicable. Suitable areas could be to monitor and detect between different driver styles or to monitor specific subsystems as the engine, transmission, or air suspension.

It is clear that the interest in the area of vehicle data acquisition and analysis is growing from many different directions. The possibility to make better decisions based on available information is beneficial in all aspects related to vehicles. The possibilities for lower fatality in traffic accidents and better resource usage of both fuel and vehicle parts makes the search for better solutions important both with regard to resource management and safety requirements.

7 References

[1] S. Fryk and P.-O. Edlund, "The Right Truck for the Job with Global Truck Application Descriptions," in *SAE Commercial Vehicle Engineering-Congress and Exhibition*, Chicago, Illinois, 2004.

[2] J. Hétu and S. Plante, "Vehicle On-Board Diagnostics Added Values," in *Commercial Vehicle Engineering Congress & Exhibition*, Chicago, 2008.

[3] R. Ludes and B. Steeples, "Road Load and Customer Data from the Vehicle Data Bus - A New Approach for Quality Improvement," in *International Congress and Exposition*, Detroit, Michigan, 1999.

[4] M. Johanson and L. Karlsson, "Improving Vehicle Diagnostics through Wireless Data Collection and Statistical Analysis," in *Vehicular Technology Conference*, Baltimore, MD, Sept. 30-Oct. 3 2007.

[5] T. Rognvaldsson, G. Panholzer, S. Byttner and M. Svensson, "A self-organized approach for unsupervised fault detection in multiple systems," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, Tampa, FL, 8-11 Dec. 2008.

[6] S. Byttner, M. Svensson and G. Vachkov, "Incremental classification of process data for anomaly detection based on similarity analysis," in *Evolving and Adaptive Intelligent Systems (EAIS), 2011 IEEE Workshop on*, Paris, 11-15 April 2011.

[7] C.-P. Young, B. R. Chang and T.-Y. Wei, "Highway vehicle accident reconstruction using Cooperative Collision Warning based Motor Vehicle Event Data Recorder," *Intelligent Vehicles Symposium, 2009 IEEE,* pp. 1131-1136, 3-5 June 2009.

[8] H. Schweppe, A. Zimmermann and D. Grill, "Flexible On-Board Stream Processing for Automotive Sensor Data," in *IEEE Transactions on Industrial Informatics*, 2010.

[9] W. Wiberg, Interviewee, *Logging approaches.* [Interview]. 10 9 2013.

[10] A. Hervé and L. J. Williams, "Principal Component Analysis," *John Wiley & Sons, WIREs Comp Stat,* vol. 2, no. 1, pp. 433-459, 2010.

[11] I. T. Jolliffe, Principal Component Analysis, New York: Springer, 1986.

[12] R. I. Davis, A. Burns, R. J. Bril and R. I. Davis, "Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised," *Real-Time Systems,* vol. 35, no. 3, pp. 239-272, 2007.

[13] N. Navet and F. Simonot-Lion, Automotive Embedded Systems Handbook, CRC Press, 2008.

[14] P. Baltusis, "On Board Vehicle Diagnostics," in *Convergence International Congress & Exposition On Transportation Electronics*, 2004.

[15] "DELPHI-About," Delphi Automotive LLP, [Online]. Available: http://delphi.com/about/news/media/photos/. [Accessed 6 11 2013].

[16] J. Olsson and H. H. Bosch, "Post-deployment Data Collection in Software-Intensive Embedded Products," in *In Proceedings of 4th International Conference of Software Business (ICSOB)*, Potsdam, Germany, 2013.

[17] J. Olsson and H. H. Bosch, "Towards Data-Driven Product Development: A Multiple Case Study on Post-Deployment Data Usage in Software-Intensive Embedded Systems," in *In Proceedings of Lean Enterprise Software and Systems Conference (LESS)*, Galway, Ireland, 2013.

[18] A. Engström, "Methods and tools for evaluation of the Volvo Pre Crash Recorder," Linköpings Institute of Technology, Linköpings, 2001.

[19] P. Niehoff, H. C. Gabler, C. Ragland, J. Hinch, C. Chidester and J. Brophy, "EVALUATION OF EVENT DATA RECORDERS IN FULL SYSTEMS CRASH TESTS," 2005.

[20] "BMW ConnectedDrive," Bayerische Motoren Werke AG, [Online]. Available: http://www.bmw.com/com/en/insights/technology/connecteddrive/2013/index.html. [Accessed 7 11 2013].

[21] "OnStar," OnStar, LLC, [Online]. Available: https://www.onstar.com/web/portal/home?g=1. [Accessed 7 11 2013].

[22] "Volvo Trucks Sweden," AB Volvo, [Online]. Available: http://www.volvotrucks.com/trucks/sweden-market/sv-se/services/dynafleet/Pages/Default.aspx. [Accessed 7 11 2013].

[23] "MV EDR, IEEE Standards Association," [Online]. Available: http://standards.ieee.org/findstds/standard/1616-2004.html. [Accessed 7 11 2013].

[24] "VERONICA (Vehicle Event Recording based On Intelligent Crash Assessment)," 2009. [Online].

Available: http://www.veronica-project.net/index.php?option=com_frontpage&Itemid=1. [Accessed 7 11 2013].

[25] "PRIVACY OF DATA FROM EVENT DATA RECORDERS: STATE STATUTES," National Conference of State Legisture, [Online]. Available: http://www.ncsl.org/research/telecommunications-and-information-technology/privacy-of-data-from-event-data-recorders.aspx. [Accessed 7 11 2013].

[26] "Federal Motor Vehicle Safety," Federal Register, [Online]. Available: http://www.gpo.gov/fdsys/pkg/FR-2012-12-13/pdf/2012-30082.pdf. [Accessed 7 11 2013].

[27] M. Jensen, J. Wagner and K. Alexander, "Analysis of in-vehicle driver behaviour data for improved safety," *Internationa Journal of Vehicle Safety,* vol. 5, no. 3, pp. 197-212, 2011.

[28] H. Kargupta, R. Bhargava, K. Liu, M. Powers, P. Blair, S. Bushra, J. Dull, K. Sarkar, M. Klein, M. Vasa and D. Handy, "VEDAS: A Mobile and Distributed Data Stream Mining System for Real-Time Vehicle Monitoring," in *Proceedings of the Fourth SIAM International Conference on Data Mining*, Lake Buena Vista, Florida, USA, 2004.

[29] S. You, M. Krage and L. Jalics, "Overview of Remote Diagnosis and Maintenance for Automotive Systems," in *SAE World Congress*, Detroit, Michigan, 2005.

[30] R. Isermann, Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance, Darmstadt: Springer, 2006.

[31] N. B. Gallagher, B. M. Wise, S. Watts Butler, D. D. J. White and G. G. Barna, "Development And Benchmarking Of Multivariate Statistical Process Control Tools For A Semiconductor Etch Process: Improving Robustness Through Model Updating," in *Process: Impact of Measurement Selection and Data Treatment on Sensitivity", Safeprocess '97*, Hull, 1997.

[32] M. L. P. Rosani and J. W. Hines, "Using Principal Component Analysis Modeling to Monitor Temperature Sensors in a Nuclear Research Reactor".

[33] R. Dunia and S. J. Qin, "Joint diagnosis of process and sensor faults using principal component analysis," *Control Engineering Practice,* vol. 6, no. 1, pp. 457-469, 1998.

[34] S. H. D'Silva, "Diagnostics based on the Statistical Correlation of Sensors," *SAE Int. J. Passeng. Cars - Electron. Electr. Syst,* vol. 1, no. 1, pp. 53-61, 2008.

[35] R. Prytz, S. Nowaczyk and S. Byttner, "Towards Relation Discovery for Diagnostics," in *KDD4Service '11*, San Diego, 2011.

[36] S. Byttner, T. Rögnvaldsson and M. Svensson, "Self-organized Modeling for Vehicle Fleet Based Fault Detection," in *2008 World Congress*, Detroit, Michigan, 2008.

[37] M. Svensson, S. Byttner and T. Rögnvaldsson, "Vehicle Diagnostics Method by Anomaly Detection and Fault Identification Software," *SAE Int. J. Passeng. Cars - Electron. Electr. Syst.*, vol. 2, no. 1, pp. 352-358, 2009.

[38] G. Govaert, Data Analysis, Hoboken: Wiley, 2010.

[39] A. K. Jain, M. N. Murty and P. J. Flynn, "Data Clustering: A Review," in *ACM Computing Surveys (CSUR)*, New York, 1999.

[40] J. B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations," in *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, Berkerly, 1967.

[41] B.-D. Shai and M. Ackerman, "Which Data Sets are 'Clusterable'? - A Theoretical Study of Clusterability".

[42] A. Banerjee and R. Dave, "Validating clusters using the Hopkins statistic," in *Fuzzy Systems, 2004. Proceedings. 2004 IEEE International Conference on*, 2004.

[43] Y. Dodge, The Concise Encyclopedia of Statistics, Springer, 2008.

[44] J. S. Milton and J. C. Arnold, Introduction to probability and Statistics: Principles and Applications for engineering and the computing sciences, New York: McGrawHill, 2003.

[45] L. Smith, "A tutorial on Principal Components Analysis," 2002.

[46] S. Chatterjee and A. S. Hadi, "SIMPLE LINEAR REGRESSION," in *Regression Analysis by Example*, John Wiley & Sons, Inc., 2006, pp. 21-45.

[47] "MineFleet," Agnik, [Online]. Available: http://www.agnik.com/minefleet.html. [Accessed 7 11 2013].

[48] H. Kargupta, K. Sarkar and M. Gilligan, "MineFleet: an overview of a widely adopted distributed vehicle performance data mining system," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, Washington, DC, USA, 2010.

[49] M. Svensson, S. Byttner and T. Rögnvaldsson, "Self-organizing Maps for Automatic Fault Detection in a Vehicle Cooling System," in *Proc. IEEE International Conference on Intelligent Systems,*, Varna, Bulgaria, 2008.

[50] S. Byttner, T. Rognvaldsson, M. Svensson, G. Bitar and W. Chominsky, "Networked vehicles for automated fault detection," in *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, Taipei, 2009.

[51] S. Byttner, M. Svensson and T. Rögnvaldsson, "Finding the Odd-One-Out in Fleets of Mechatronic Systems using Embedded Intellient Agents," in *AAAI Spring Symposium*, Stanford, CA, March 22-24 2010.

[52] A. Mosallam, S. Byttner, M. Svensson and T. Rognvaldsson, "Nonlinear Relation Mining for Maintenance Prediction," in *Aerospace Conference, 2011 IEEE*, Big Sky, MT, 5-12 March 2011.

[53] K. F. Martin and M. H. Marzi, "Diagnostics of a coolant system via neural networks," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering,* vol. 213, pp. 229-243, 1999.

[54] "AUTOSAR, AUTomotive Open System ARchitecture," AUTOSAR, [Online]. Available: http://www.autosar.org/. [Accessed 6 11 2013].

[55] A. Kelley and I. Pohl, A Book on C: programmin in C, Indianapolis: Addison-Wesley, 2006.

[56] "CUnit-A Unit Testing Framework for C," [Online]. Available: http://cunit.sourceforge.net/index.html. [Accessed 14 11 2013].

[57] J. Dooley, "Unit Testing," in *Software Development and Professional Practice*, New York, Apress, 2011, pp. 193-208.

[58] Bowdler, Martin, Reinsch and Wilkinson, Handbook for Auto. Comp. Vol.ii-Linear Algebra.

[59] W. H. Press and W. T. Vetterling, Numerical recipes in C: The art of scientific computing, Cambridge: Cambridge University Press, 1988.

[60] "CANoe," Vector Informatik GmbH , 2010-2013. [Online]. Available: http://vector.com/vi_canoe_en.html. [Accessed 6 11 2013].

[61] "Python Programming Language – Official Website," Python Software Foundation, 1990-2013. [Online]. Available: http://www.python.org/. [Accessed 6 11 2013].

[62] I. Geert, Interviewee, *private correspondence.* [Interview]. 18 11 2012.

[63] "FAQs: IsYour TPMS Light On?," Schrader, [Online]. Available: http://www.tpmsmadesimple.com/faq.php. [Accessed 19 11 2013].