



Norwegian University of  
Science and Technology

# Condition Monitoring of Mechanical Equipment Using Bilinear Modeling of Vibrations

**Jørgen Lerøy Neset**

Master of Science in Industrial Cybernetics

Submission date: June 2018

Supervisor: Harald Martens, ITK

Norwegian University of Science and Technology  
Department of Engineering Cybernetics



# Abstract

The purpose of this project is to investigate the application of multivariate analysis methods on vibration data obtained from a diesel engine using acoustic contact microphones. A fully instrumented Mercedes-Benz OM613 diesel engine for passenger cars is the subject of the research. Data acquisition was carried out at the Internal Combustion Engine Laboratory at the Norwegian University of Science and Technology. A large number of measurements related to engine operation, including vibrations, were obtained for a range of different operating conditions. Analysis was initially carried out in the time domain where it was shown that vibration amplitude increases with engine speed and load.

In addition, both traditional time-frequency domain analysis and multivariate analysis methods were studied. Resampling of the vibration data revealed some frequencies that were present for all operating conditions, however increasing in magnitude with increasing engine speed. An attempt was made to connect these frequencies to engine events. A suggestion was that the common rail fuel system, with pressure sensor and fuel regulator could be connected to a static frequency, but with magnitude increasing with engine speed. This theory was further substantiated through that the return of fuel to the tank was increased with engine speed.

Principal component analysis revealed a resonance structure which was not visible in the time-frequency domain analysis. This resonance is thought to be the resonance of the engine block. Several regression models were made for different response variables, with vibration data in the time-frequency domain as predictors. The best regression models were found for the variables weight of exhaust gases, intake temperature and fuel return to tank. Regression models for cylinder pressure and oil temperature were evaluated against regression models where engine speed had been used as the predictor. For the cylinder pressure model, vibration data were better predictors than engine speed, but had an estimated prediction error of 12% with 7 factors. For the oil temperature model an estimated prediction error of 6% was obtained with 5 factors, but engine speed gave a better model with fewer factors. The vibration models showed promising results, but further research is needed before vibration sensors can replace other sensors.





# Sammendrag

Formålet med dette prosjektet er å undersøke bruken av multivariate analysemetoder på vibrasjonsdata hentet fra en dieselmotor ved bruk av akustiske kontaktmikrofoner. En fullt instrumentert Mercedes-Benz OM613 dieselmotor for personbiler er gjenstand for forskningen. Datainnsamling ble gjennomført ved laboratoriet for forbrenningsmotorer ved Norges teknisk-naturvitenskapelige universitet. Mange ulike målinger relatert til motoroperasjon, inkludert vibrasjoner, ble utført for en rekke forskjellige driftsforhold. Analysen ble innledningsvis utført i tidsdomenet, hvor det ble vist at vibrasjonsnivået økte med motorhastighet og belastning.

Videre ble både tradisjonell tidsfrekvensdomene-analyse og multivariate analysemetoder studert. Resampling av vibrasjonsdataene viste noen frekvenser som var tilstede for alle driftsforhold, men økte i styrke med økende motorhastighet. Det ble forsøkt å koble disse frekvensene til motorhendelser. Et forslag var at common rail drivstoffsystemet, bestående av blant annet trykksensor og drivstoffregulator kunne kobles til en konstant frekvens, men med styrke som økte med motorhastighet. Denne teorien ble videre underbygget gjennom at retur av drivstoff til tanken økte med økende motorhastighet.

Prinsippal komponent analyse avslørte en resonansstruktur som ikke var synlig i tidsfrekvensdomene-analysen. Denne resonansen antas å være resonansen i motorblokken. Flere regresjonsmodeller ble laget for ulike responsvariabler, med vibrasjonsdata i tidsfrekvensdomenet som prediktorer. De beste regresjonsmodellene ble funnet for variablene vekt av eksosgass, innsugningstemperatur og mengde returnert drivstoff til tank. Regresjonsmodeller for sylindetrykk og oljetemperatur ble evaluert mot regresjonsmodeller der motorhastigheten var brukt som prediktor. For sylindetrykkmodellen var vibrasjonsdata en bedre prediktor enn motorhastighet, men modellen hadde en estimert prediksjonsfeil på 12% med 7 faktorer. For oljetemperaturmodellen ble en estimert prediksjonsfeil på 6% oppnådd med 5 faktorer, men motorhastigheten ga en bedre modell med færre faktorer. Regresjonsmodellene med vibrasjonsdata i frekvensdomenet som prediktorer viste lovende resultater, men det er nødvendig med videre forskning før vibrasjonssensorer kan erstatte andre sensorer.



# Preface

This is a master's thesis submitted in partial fulfilment of the requirements for a Master of Science for the study program Industrial Cybernetics at the Norwegian University of Science and Technology in Trondheim. The project was carried out during the spring semester of 2018 in collaboration with the company Idletechs, and builds on a feasibility study that was done in the autumn semester of 2017. The objective of the project was to investigate the applicability of multivariate analysis on vibration data recorded on a diesel engine for condition monitoring purposes. Data acquisition has been carried out at the Internal Combustion Engine Laboratory at the Department of Energy and Process Engineering. The problem formulation was prepared in cooperation with Jan Otto Reberg at Idletechs as a result of the students desire to have an interdisciplinary project with some practical work, and the company's desire to implement multivariate analysis on vibration data from mechanical equipment.

This project has been challenging, but also very interesting; both the world of signal conditioning and multivariate analysis is quite new for me, so there were many things to learn. A lot of time has been spent on the processing and interpretation of measurement data. I would like to thank my supervisor at Idletechs, Jan Otto Reberg for the help with preparing the project formulation, all guidance, brainstorming and enthusiasm throughout the project. He has been an important person for the progress of the project, and was also the one who supplied the sensors used to obtain the vibration measurements. I would also like to thank David Emberson at the Department of Energy and Process Engineering who has helped with the data acquisition at the laboratory and shown great attitude towards the project from the start.

Trondheim, 2018-06-11

Jørgen Lerøy Neset



# Contents

<b>Abstract</b>	<b>i</b>
<b>Sammendrag</b>	<b>iii</b>
<b>Preface</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background and Motivation . . . . .	1
1.2 Goal and Method . . . . .	5
1.3 Objectives . . . . .	5
1.4 Approach . . . . .	6
1.5 Contributions . . . . .	7
1.6 Limitations . . . . .	7
1.7 Outline . . . . .	8
<b>2 Theoretical Background</b>	<b>9</b>
2.1 Condition Monitoring . . . . .	9
2.1.1 Maintenance Strategies . . . . .	9
2.1.2 Machine Condition Monitoring . . . . .	10
2.1.3 Condition Monitoring Techniques . . . . .	13
2.1.4 Vibration Measurement . . . . .	13
2.1.5 Vibration Analysis . . . . .	16
2.2 Signal Processing . . . . .	19
2.2.1 Frequency Domain . . . . .	20
2.2.2 Spectra . . . . .	23
2.2.3 Time-Frequency Domain . . . . .	25
2.3 Multivariate Analysis . . . . .	31
2.3.1 Principal Component Analysis . . . . .	32
2.3.2 Mathematical Derivation of PCA . . . . .	33
2.3.3 Important Plots and Interpretation of Plots . . . . .	37

2.3.4	Regression Methods . . . . .	38
2.3.5	Multiple Linear Regression . . . . .	40
2.3.6	Principal Component Regression . . . . .	40
2.3.7	Partial Least Squares Regression . . . . .	41
2.3.8	Validation . . . . .	42
2.4	Internal Combustion Engines . . . . .	43
2.4.1	Principle of Operation . . . . .	43
2.4.2	Noise and Vibration Sources . . . . .	46
<b>3</b>	<b>Materials and Methods</b>	<b>49</b>
3.1	Experimental setup . . . . .	49
3.1.1	The Diesel Engine . . . . .	49
3.1.2	Measurements and Sensors . . . . .	52
3.1.3	Preparations for Vibration Measurements . . . . .	55
3.2	Measurement Procedure . . . . .	59
3.3	Data Processing and Analysis . . . . .	61
3.3.1	Data Extraction . . . . .	61
3.3.2	Defining an Engine Cycle . . . . .	61
3.3.3	Fourier transform . . . . .	65
3.3.4	Adapting Measurements to Engine Cycle . . . . .	65
3.3.5	Resampling . . . . .	66
3.3.6	The Unscrambler®X . . . . .	67
<b>4</b>	<b>Results and Discussion</b>	<b>69</b>
4.1	The Effects of Speed and Load on Vibrations . . . . .	69
4.1.1	Time Domain . . . . .	69
4.1.2	Time-Frequency Domain . . . . .	73
4.2	PCA Models . . . . .	81
4.2.1	PCA Model 1 . . . . .	81
4.2.2	PCA Model 2 . . . . .	87
4.2.3	PCA Model 3 . . . . .	93
4.3	Regression models . . . . .	96
4.3.1	Cylinder Pressure . . . . .	96
4.3.2	Oil Temperature . . . . .	100
4.3.3	Other variables . . . . .	105
<b>5</b>	<b>Conclusion</b>	<b>109</b>
5.1	Recommendations for Further Work . . . . .	111

5.2 Closing Remarks . . . . .	113
<b>References</b>	<b>119</b>
<b>Appendix</b>	<b>119</b>
<b>A Abbreviations</b>	<b>121</b>
<b>B Regression plots</b>	<b>123</b>
<b>C MATLAB Code</b>	<b>125</b>
C.1 Data Extraction . . . . .	125
C.2 Data Processing . . . . .	129





# List of Tables

2.1	Time domain descriptors for vibration signals [22], [27] . . . . .	18
3.1	Technical details for the engine [52] . . . . .	51
3.2	High Speed Measurements . . . . .	53
3.3	Low Speed Measurements . . . . .	54
3.4	Measurement plan with test conditions . . . . .	60
4.1	Comparison of the regression measures for the three models with cylinder pressure as response variable . . . . .	99
4.2	Comparison of the regression measures for the the models with oil temperature as response variable . . . . .	104
4.3	Regression measures for several different measurement variables related to engine operation . . . . .	105



# List of Figures

2.1	Flowchart summarising the process of condition-based maintenance . . .	11
2.2	A compression accelerometer and its components (Copyright©Brüel & Kjør) . . . . .	15
2.3	Structure of a piezoelectric pickup microphone . . . . .	16
2.4	Typical vibration waveform (time domain) of an internal combustion engine	19
2.5	Graphical representation of the transition from time domain to frequency domain . . . . .	22
2.6	Amplitude spectrum of a 2.4second long vibration recording of a six cylinder diesel engine running at 1000rpm and with a load of 20Nm . . .	24
2.7	A zoomed view of Figure 2.6, where it is possible to recognise several frequencies and relate them to physical events. . . . .	25
2.8	An example of two different window functions for 512 samples, the Hanning and Hamming windows. . . . .	27
2.9	The original and windowed signal of one frame of length 512 samples from the vibration waveform presented in Figure 2.4 . . . . .	28
2.10	The principle of dividing a signal into frames . . . . .	29
2.11	Spectrogram of the vibration recording of a six cylinder diesel engine . .	30
2.12	Illustration of Principal Components and the score value [42] . . . . .	33
2.13	Geometrical representation of scores . . . . .	34
2.14	A simplified illustration of a diesel engine cylinder . . . . .	44
2.15	Illustrations of the TDC and BDC positions of the piston . . . . .	46
2.16	The different noise sources in internal combustion engines . . . . .	47
3.1	The instrumented diesel engine in the lab . . . . .	50
3.2	The engine and test bench with dynamometer . . . . .	51
3.3	The pickup microphone used for this project . . . . .	55
3.4	The new BNC plug an coaxial cable on the microphone . . . . .	56
3.5	The placement of microphone #1 . . . . .	57
3.6	Placement of microphone #2, on the air supply to the turbocharger. . . .	58
3.7	Schematic overview of microphone and pressure transducer locations . .	59

3.8	Vibrations from microphone #1 and cylinder pressures for Test1 (1000rpm20Nm) and Test3 (1200rpm50Nm) . . . . .	62
3.9	The three first cycles (or, 12288 first samples) of the cylinder pressure data for all tests, down scaled by a factor of 10. . . . .	63
3.10	Illustrations of the effect of matching engine cycles . . . . .	64
3.11	The workflow of the data processing procedure . . . . .	68
4.1	Time domain vibration signal for three different rpms . . . . .	70
4.2	Time domain vibration signal for three different rpms . . . . .	71
4.3	Time domain vibration signal for three different rpms . . . . .	72
4.4	The RMS value of the vibration measurements from microphone #1 of all tests. . . . .	73
4.5	Spectrogram of vibration measurements (microphone #1) for all tests prior to resampling . . . . .	74
4.6	Spectrogram of vibration measurements (microphone #1) after resampling	74
4.7	A 3D plot of the time-frequency data of the vibration measurements from microphone #1 . . . . .	75
4.8	Spectrogram of resampled vibration data from microphone #2 . . . . .	76
4.9	A 3D plot of the time-frequency data of the vibration measurements from microphone #2 . . . . .	76
4.10	A zoomed view of Figure 4.6 which shows the effect of the different load conditions . . . . .	78
4.11	Zoomed views of the high intensity bands of the vibration measurements	80
4.12	Scores of PC1 and PC2 from the first PCA model plotted against each other. The different rpms have been colourised . . . . .	82
4.13	The loadings of PC1 for the first PCA model . . . . .	83
4.14	Zoomed views of the peaks for PC1 loadings in the first PCA model . . .	84
4.15	The scores of PC1 for the first PCA model . . . . .	85
4.16	The loadings of PC2 for the first PCA model . . . . .	86
4.17	The scores of PC2 for the first PCA model . . . . .	86
4.18	Scores of PC1 and PC2 from the second PCA model plotted against each other. The different rpms have been colourised . . . . .	88
4.19	The loadings of PC1 for the second PCA model . . . . .	89
4.20	Zoomed view of the frequency structure in the loadings of PC1 for the second PCA model . . . . .	90
4.21	The loadings of PC2 for the second PCA model . . . . .	91
4.22	The scores of PC2 for the second PCA model . . . . .	91
4.23	Calibration scores and Validation scores in same plot . . . . .	92

4.24	Scores of PC1 and PC2 from the third PCA model plotted against each other	94
4.25	Predicted versus reference cylinder pressure values. Vibration data in time-frequency domain has been used as predictors in a PCR model. . . .	97
4.26	Predicted versus reference cylinder pressure values. Vibration data in time-frequency domain has been used as predictors in a PLSR model . . .	98
4.27	Predicted versus reference cylinder pressure values. Engine speed data has been used as predictors in an MLR model. . . . .	98
4.28	Predicted versus reference oil temperature values. Vibration data in the time-frequency domain has been used as predictors in a PLSR model. . .	101
4.29	Predicted versus reference oil temperature values. Engine speed data has been used as predictor in a MLR model. . . . .	102
4.30	Predicted versus reference oil temperature values. Engine speed and torque data has been used as predictors in a PLSR model. . . . .	103
B.1	Predicted versus reference exhaust gas weight values. Vibration data in time-frequency domain has been used as predictors in a PLSR model . .	123
B.2	Predicted versus reference intake temperature after cooler values. Vibration data in time-frequency domain has been used as predictors in a PLSR model . . . . .	124
B.3	Predicted versus reference engine torque values. Vibration data in time-frequency domain has been used as predictors in a PLSR model . . . . .	124



# Chapter 1

## Introduction

### 1.1 Background and Motivation

Condition monitoring of machines is far from a new field; the idea of monitoring machines as a way to realise predictive maintenance strategies has been used for decades. Some of the earliest methods that can be related are investigations of fatigue on axles of stage coaches and railways as well as fatigue of chains in the machines of the industrial revolution. The “monitoring” was carried out using the simplest sensing devices there is; the hands, ears and eyes of experienced operators. As machines and equipment grew in complexity, such sensing devices no longer gave satisfactory information about the conditions, and new methods emerged proportionally with the availability of instrumentation, technology and computational power [1].

Although many parameters have been used and still are in use for condition monitoring of machines, vibrations are without doubt the dominating indicators of machine health [2]. As sound is generated through vibration of objects, it is also an indicator of machine health. The reason for the extensive use of vibration analysis for condition monitoring purposes is due to the fact that a change in the condition of a machine will affect its dynamic behaviour, which in turn affects its vibratory behaviour [1]. This makes vibration analysis a good indicator of the condition of a machine, because emerging faults can be detected in real-time, as opposed to for example oil and lubrication analysis, where the time between samples taken and analysis of samples can be several days.

Condition monitoring of rotary equipment such as roller bearings and turbines is usually easier due to the stationary nature of the vibration signals; periodic events in such machines have a frequency of occurrence, and thus makes separation and study of such sources easier. Therefore, traditional statistical and frequency domain analysis has seen a lot of success in the area of vibro-acoustic condition monitoring for such machinery [3], [4]. Internal combustion (IC) engines also has recurring events; for each

engine cycle there is a combustion event in each of the cylinders of the engine. However, even if these events happen repeatedly there are variations from one cycle to the next, thus the vibration signals of IC engines are classified as non-stationary. What separates stationary signals from non-stationary signals is that stationary signals have statistical properties which does not change in time [5].

Research has been done in the time and frequency domains with various results; Abed *et al* [6] analyse the vibration signatures of a four-stroke, four cylinder diesel engine and study in particular the effects of load and speed variations as well as induced faults in the exhaust valve clearance of cylinder #1. Time domain analysis proves useful for relating engine events to the angle of the crank shaft, however in a condition monitoring approach the usefulness is less prominent. When examining the recorded signals in the frequency domain using the fast Fourier transform, differences can be seen between the faulty and healthy cases. Albarbar *et al* [7] conclude that limited information is found in the time and frequency domain of acoustic signals from diesel engines, due to the dominating appearance of the firing frequencies of the engine and its harmonics. Although apparent, this is not stated as an issue in [6].

The non-stationary nature of the vibration and acoustic signals from IC engines seems to lead to the need of more advanced signal processing methods. Analysing such signals in the joint time-frequency domain using signal processing methods such as the short-time Fourier transform (STFT), continuous wavelet transform (CWT), Wigner-Ville distribution (WVD) has shown to be effective for extraction of relevant features. Albarbar *et al* [7] show that applying the Wigner-Ville distribution on acoustic signals of a diesel engine makes it possible to define the frequency of combustion induced noise (below 5kHz), however defining other acoustic sources proved more difficult without carefully applying filters on the respective frequency bands. Gu *et al* [8] apply both the continuous wavelet transform and the smoothed Wigner-Ville distribution on acoustic signals of a diesel engine for comparison, and find that the CWT provide better detail of the high frequency combustion, while SPWVD provide more detailed information in the lower frequency band.

Yadav *et al* [9] propose a method where airborne acoustic signals from internal combustion engine are separated into frames and an FFT is applied to each frame, thus taking advantage of the assumption that a signal is stationary within a small time frame (“quasi-stationary”). Antoni *et al* [10] introduce a condition monitoring approach for IC engines based on cyclostationarity of vibration signals; some of the events (mechanical) in the engine cycle will have periodic patterns whereas other events (thermodynamics, fluid mechanics) will have random variations from one cycle to the next. The idea is that the random variations have a hidden periodicity and thus contain useful information.



This coincides with the signal classification given in [5]. This shows that there are many ways to handle the non-stationary signals of the IC engines, however most of these studies have been carried out in a laboratory environment, and thus it remains to see how these methods can be implemented in on-line monitoring systems.

As the industry's demand for reliability, availability, maintainability and safety continues to increase, condition monitoring is an area which can provide many benefits. The technology development has led to easier, better and cheaper equipment and sensors for data acquisition. A growing concern in our connected society is the vast amounts of data that is collected; the data in itself is not useful unless meaningful information can be extracted by efficient methods, preferably in real-time. Multivariate data analysis, tightly connected to chemometrics, has seen an increase in areas of application over the last two decades. Successful implementations include pattern recognition, process monitoring, multivariate calibration, classification and discriminant analysis. The idea of multivariate data analysis is to use mathematical and statistical tools to uncover (hidden) structures in data, and thus extract useful information. Nature is multivariate, meaning that a phenomenon usually is dependant upon more than one factor; this is assumed to hold for most data generating systems, technological and otherwise [11].

Over the past decade, different applications of multivariate analysis for condition monitoring purposes has been studied in academia, however few solutions have been implemented in real-world environments. Li *et al* [12] study the noise sources of a diesel engine using the independent component analysis (ICA). The argument for the proposed method is that traditional methods like time domain, frequency domain and time-frequency domain analysis are not successful in identifying low-level noise sources such as fuel injection and valve movements due to the small energy level of the signals of these events. In addition, these events often happen to be buried under events such as the combustion, which contain a high level of energy. Applying ICA to the acoustic measurements of a diesel engine and then transforming them to the time-frequency domain using the CWT made it possible to separate events such as fuel injection, inlet valve opening, exhaust valve opening and piston slap. A similar method is utilised in [7], with attention directed towards monitoring of the diesel engine fuel injection system.

Independent component analysis has also seen other applications; Li *et al* [13] suggest a method where ICA is used for sensor fusion; data from four vibration sensors are combined using ICA, and results of the study show that the recognition of faults is improved through information fusion. Another method with roots from the realm of multivariate analysis is the principal component analysis (PCA); it is a method which replaces the original variables in a data set with latent variables and at the same time reduces the dimension of the data [11]. Jafarian *et al* [14] apply principal component

analysis to vibration measurements done on a four-cylinder IC engine.

The motivation behind the study is the quality check in the last step of the manufacturing process of an automobile engine at the Iranian manufacturing company IKCO. Sound, vibration and resonance, among other things, are examined and compared to standards; any deviations lead to the respective engine being removed from the production line for further investigation. Although faults are detected, their location is not; thus the process of fixing an engine is time consuming, and the risk of incorrect fault detection and diagnosis is present which could lead to an engine being opened several times. The proposed method applies PCA on vibration signals acquired in healthy and faulty operating conditions of the IC engine, and clear differences are seen in the outputs of the PCA. Extracting standard deviation and mean as features, three fault categories and the healthy state are separated clearly.

In the study done by Li *et al* [13] which was mentioned above, after using ICA for sensor fusion, STFT for transforming the measurements to time-frequency domain and extracting relevant features, PCA is successfully used as a “feature fusion” technique, i.e reducing the dimension of the extracted features. PCA has also been proposed for other condition monitoring methods, mainly as a way to classify and separate different conditions of mechanical equipment from each other. Baydar *et al* [15] apply PCA to the vibration signals of a two-stage helical gearbox; a reference model and confidence region with normal operating conditions is first established, and deviations from these indicate a fault. It is found that the Q-statistics (squared prediction error), the  $T^2$ -statistic and confidence regions all are useful towards fault identification. Similar applications of PCA for condition monitoring purposes is also seen in [16], [17] and [18].

Another consequence of the technological development and increase in computational power is the appearance of intelligent systems for condition monitoring; systems that learn, and based on what they learn can make decisions. In engineering, such systems are usually connected to artificial intelligence and artificial neural networks (ANN). Such systems have seen some research in academia; Li *et al* [13] implement a fuzzy neural network for fault pattern identification in a marine diesel engine and Abraham *et al* [19] use an artificial neural network for classification of acoustic signals of a motorcycle engine. Applications of artificial neural networks is also seen in [20] and [21].

As can be seen condition monitoring is a major research area, and many types of (successful) systems and methods are proposed. However, few have found their way into real-world condition monitoring applications. This might indicate that more research is required, or that the proposed methods are too advanced or unpractical to implement in the industry. The intent of this master’s thesis is to build on the existing literature and research, with a main focus on internal combustion engines.

## 1.2 Goal and Method

The goal of this master's thesis is to use one or more vibration sensors (or contact microphones) and multivariate analysis to monitor an internal combustion (diesel) engine in order to characterise operation under varying conditions. The result can be used for one or more of the following:

- Replacing other sensors with vibration sensors (being able to characterise variations in for example temperature and pressure from vibration signals)
- Detecting and characterising long term trends
- Automatic control of performance (limit a cars performance when the engine is "cold")

This will be accomplished by making one or more bilinear models for establishing relationships between vibrations and other parameters. An attempt will be made to obtain sensor values (temperature, pressure) as a function of vibrations.

## 1.3 Objectives

The main objectives of this Master's project are

### 1. Literature study

- (a) Condition monitoring of mechanical equipment using vibrations and acoustic measurements, with main focus on internal combustion engines
- (b) Condition monitoring using multivariate methods, with focus on principal component analysis

### 2. Study of internal combustion engines

- (a) General principle
- (b) Noise (vibration) sources

### 3. Data acquisition

- (a) Determine method and measurement plan
- (b) Implement data acquisition according to plan

### 4. Data processing

- (a) Structure data appropriately

- (b) Consider and apply necessary pre-processing
  - (c) Study measurements using appropriate signal processing methods (time domain, frequency domain, time-frequency domain)
5. **Experimental study of vibration measurements for condition monitoring of diesel engine**
- (a) Establish a set of normal operating conditions, and a reference model based on normal operating condition vibrations
  - (b) Look into the effects of speed and load on vibrations
  - (c) Look for relationships between the vibration measurements and operating parameters (temperature, pressure) using multivariate methods

## 1.4 Approach

The literature study is done by a literature search in the available scientific databases provided by the Norwegian University of Science and Technology (NTNU). Relevant literature is read and summarised, and used as a source of information and inspiration for procedures and experiments. Data acquisition will be carried out at the Internal Combustion Engine Laboratory at the Department of Energy and Process Engineering at NTNU. In particular, the experiments will be performed on a fully instrumented Mercedes Benz OM 613 turbocharged diesel engine. New for this project is the integration of acoustic pickups on the engine and to the already existing data acquisition system, for measurement of vibrations. The obtained measurements from the engine system will be processed and analysed using the following software:

- MATLAB R2018a
- The Unscrambler®X v10.5

The measurement system and methodology is more thoroughly described in chapter 3.

## 1.5 Contributions

This master's project builds on existing literature and research on condition based maintenance for internal combustion engines. The first contribution is the study of how cheap, simple acoustic piezo pickups can be used for measuring vibrations on a six-cylinder diesel engine for condition monitoring purposes. The second, and most important contribution is the application of multivariate analysis on the measured data. Furthermore, the investigation of relationships between operating parameters such as temperature, pressure and vibrations can bring about new knowledge and thus reduce the need for many sensors (of which some are intrusive) in for example a laboratory rig.

## 1.6 Limitations

This project is limited to the spring semester of 2018, and funding and equipment other than what is available through the Norwegian University of Science and Technology is not looked into. Albeit, the available equipment and software should be sufficient to investigate the problem at hand. As the engine used for data acquisition in this project also is used by many other projects, inducing faults will not be possible. However, the engine system provides other options for applying changes to the operating conditions. Moreover, this project is not focused towards the development of a complete condition monitoring system, but rather studies the underlying opportunities for the creation of such a system.

## 1.7 Outline

The following gives an overview of how the thesis is organised.

- **Chapter 1 - Introduction:** Sets the background and motivation as well as the goal and the objectives for the project. Furthermore, the approach, limitations and contribution are presented.
- **Chapter 2 - Theoretical Background:** Gives a presentation of theory needed for the methods used and discussions made throughout the project. This includes condition monitoring, the general principle of an internal combustion engine and the most important noise sources, signal processing and multivariate analysis.
- **Chapter 3 - Materials and Methods:** Introduces the system and equipment that have been used in this study, and explains the methods and the pre-processing that have been applied, so that the experiments can be recreated.
- **Chapter 4 - Results and Discussion:** States the findings of the study, interpretation of results, and discussion of sources that might have influenced the results.
- **Chapter 6 - Conclusion:** Insights and ideas for further work.
- **References:** The reference list of the literature review.
- **Appendix - A:** Abbreviations
- **Appendix - B:** Regression plots
- **Appendix - C:** MATLAB code

# Chapter 2

## Theoretical Background

This chapter provides theory and background information related to condition monitoring, internal combustion engines, signal processing and multivariate analysis. The theory presented should give the necessary understanding for the methods and procedures to come, but is by no means an exhaustive representation of the subjects.

### 2.1 Condition Monitoring

#### 2.1.1 Maintenance Strategies

Most machines are degrading systems, meaning that wear eventually will lead to a need for maintenance or replacement. To reduce the downtime, and thus increase the availability of machines, several maintenance programs have been developed. Which maintenance program that is implemented for a given machine usually depends on the importance of that machine. Rao [1] classifies the maintenance programs into the following categories:

- Run to failure
- Preventive maintenance
- Predictive maintenance

The same categories are also used by Randall [2] and Mohanty [22], however the run to failure-category is here also known as *reactive maintenance*. The following paragraphs shortly explain each of these strategies.

Reactive maintenance is as mentioned above a strategy where the machines run until they break or fail to do their job. At this point, maintenance is required either in the form of changing of parts or complete replacement. This maintenance program is usually only

implemented for non-critical machines and equipment and is commonly associated with large repair costs and long downtimes.

*Preventive maintenance* is a strategy where maintenance is planned and performed according to guidelines from manufacturers or based on prior knowledge of failures. Overhauls are carried out periodically, and parts whose life expectancy have surpassed are replaced by new ones. The positive aspect of such a maintenance strategy is that the downtime is reduced, because the maintenance is planned, and machines are less prone to critical failures. However, a downside is that some of the parts that are being replaced probably have some remaining useful life (RUL).

*Predictive maintenance* is the condition-based strategy; indicators of machine health is monitored and deviations from “normal” values might indicate a problem. From the information of the current condition of the machine the future condition is predicted, and maintenance can be planned and carried out according to this. Thus downtime is minimised and availability is maximised.

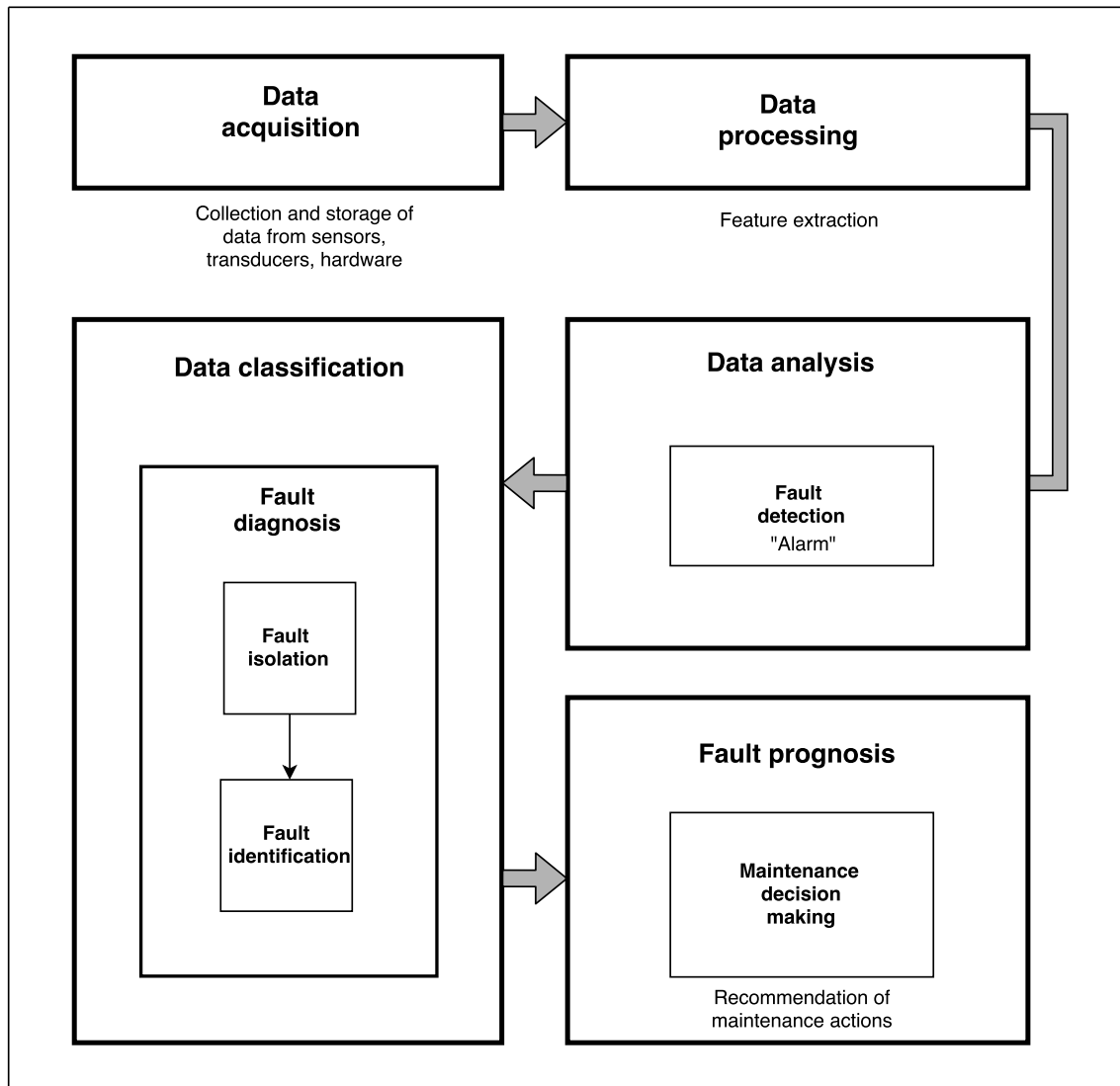
### 2.1.2 Machine Condition Monitoring

Within condition monitoring, it is normal to divide the monitoring into two categories; *permanent monitoring* and *intermittent monitoring*. Permanent monitoring relates to condition monitoring systems where the transducers are permanently attached to the machine being monitored. Such systems are usually implemented for machines that are critical; monitoring is done continuously, and thus abrupt changes can be discovered and the machine can be shut down if necessary. Intermittent monitoring is monitoring where there is a preset interval for the data acquisition; thus it is more directed towards the long-term trends and not the sudden changes that might occur. In general, intermittent monitoring can be used on most machines by setting the data acquisition interval to an appropriate value. Furthermore, intermittent monitoring can be carried out both by permanently mounted transducers and transducers which are moved between measurement points [2].

Although several definitions are used, it is common to divide machine condition monitoring into some main phases. Randall [2] divide machine condition monitoring into three phases: **fault detection**, **fault diagnosis** and **fault prognosis**. Prognosis is the phase where the future condition of the machine is predicted; i.e the remaining useful life of the machine (components). Jardine *et al* [23] divide the condition-based maintenance program into three steps: **(1) data acquisition**, **(2) data processing** and **(3) decision making**, however diagnostics and prognostics are deemed as two important parts of condition based maintenance. These three steps are also mentioned as key elements in [9]. Building on this, the process of condition-based maintenance can be summarised



with the flowchart presented in Figure 2.1.



**Figure 2.1:** Flowchart summarising the process of condition-based maintenance

The data acquisition step is rather self-explaining; it is the step where data is collected and stored. In general there are two different types of data, the first being data related to events such as installation, breakdown and overhauls and the second being condition monitoring data which are the measurements from various sensors. The latter type is the one being used in the process of condition-based maintenance, however the event-data can provide insight into the performance of the machine. That is, the performance of an engine or machine can be said to be decreasing as the wear rate increases through the running-in phase, the normal operations phase and the wear-out phase [24].

The data processing step can be divided into two: (1) pre-processing and (2) signal processing and feature extraction. The pre-processing vary considerably from application to application, however data cleaning i.e the searching and removal of erroneous data

is a common and important part of the pre-processing. Furthermore, methods like normalisation and mean centring are also typical in the pre-processing of data. As mentioned in the previous paragraph, there are two data types in the data acquisition step. The condition monitoring data can be further divided into three categories: (1) waveform data, (2) value data and (3) multidimensional data [23]. The waveform data are data that are recorded for a variable as a time series, like vibration and acoustic measurements. The value type data are data that are recorded as a single value like temperature and pressure. Finally, the multidimensional data are data with several dimensions such as images.

The data processing for waveform and multidimensional data is called signal processing, and there are many different approaches of which some are discussed in section 2.2. Raw data by itself seldom provide a lot of useful information, and therefore feature extraction is often necessary. Depending on the application and the data category, the process of feature extraction vary considerably. A brief overview of feature extraction techniques is given in section 2.1.3.

Although presented in separate boxes in the flowchart, the steps of data processing, data analysis and data classification are tightly interconnected. The idea of the data analysis step is that the extracted features from the processing step are investigated closer, and that “alarms” will give indication of something being out of order, i.e a fault detection. Alarms might be related to a preset threshold, or as a result of comparing the extracted features to features extracted from the same machine at a time of similar and healthy operation.

With a fault present, as indicated by alarms the next step is to classify the fault to discover its location, nature and severity. As seen in Figure 2.1, the main part of the data classification is the fault diagnosis; the isolation and identification of a fault. The fault isolation is the action of locating the component which is responsible for the fault whereas fault identification is the action of determining the nature of the fault, i.e why is there a fault and what is the severity of the fault. Having a database with known fault cases and data related to these fault cases can be used as a way to compare and classify the newly occurring fault. If the new fault has features that coincide with a fault in the database it is very likely that the fault is the same, however if the features of the new fault show no similarities with faults in the database further investigation is required. An example of such a way of classifying a fault can be that vibrations in a certain frequency range with a noticeable amplitude measured in the fault at hand matches the vibrations in the frequency range of a known fault in the database.

The final step of the condition-based maintenance process is the fault prognosis and the maintenance decision making. The prognosis and diagnosis in this scheme are

somewhat intertwined; in general the fault diagnosis consist of the fault detection, fault isolation and fault identification. So for there to be a fault diagnosis a fault will necessarily already have occurred, and thus diagnostics is a posterior event analysis [23]. With the fault diagnosed, recommendation of maintenance actions can be given directly. The fault prognosis on the other hand is a prior event analysis, and attempts to predict a fault before it occurs. The prediction includes an estimate of how likely and how soon the fault is to occur i.e the RUL of the machine before maintenance is necessary. So if an alarm has revealed that a fault is emerging, it can still be on a very early stage and the prognosis thus predicts when the fault is too severe for the machine to continue operation and that maintenance is necessary. Therefore it is often common to include several alarm levels which represents the severity of the faults.

### 2.1.3 Condition Monitoring Techniques

In the case of condition monitoring of machines there are several measurements that could be done to investigate the health of the machine. In all practical applications (outside laboratory environments) such measurements should be limited to those that can be done externally. With this in mind, the most common techniques are vibration analysis and lubricant oil analysis [2]. Airborne acoustic measurements and acoustic emission techniques has also seen an increase in condition monitoring research, however these are closely related to the vibration analysis. Put very shortly, the idea of oil analysis is to examine the chemical contaminants and the size and amount of wear particles in the lubricant oil. As the main focus of this project is towards vibration analysis, oil analysis will not be discussed further.

Vibration analysis for condition monitoring purposes has been used for decades and shows many advantages over other condition monitoring techniques. All machines emit vibrations during operation, whether they are in good condition or not. However the vibration of a machine that is in good condition and the vibration of the same machine with a fault will be different, which makes vibration analysis very well suited for condition monitoring purposes. In fact, as a fault emerges the vibration signature will change along with it, meaning that a fault can be uncovered at an early stage.

### 2.1.4 Vibration Measurement

Randall [2] describes three ways to interpret vibrations when used for condition monitoring purposes: (1) absolute vibration (f.ex of a machine housing), (2) relative vibration (f.ex between a shaft and the housing) and (3) torsional vibration. The three types give different yet relevant information for diagnostics. Most relevant for this project is the

absolute vibration of a machine housing, as contact microphones will be attached on the engine block. Furthermore, vibration can be expressed through the three parameters displacement, velocity and acceleration. The corresponding transducers most commonly used for measurements in condition monitoring applications are proximity probes, velocity transducers and accelerometers. Proximity probes measure relative displacement, while absolute motion is measured by most acceleration and velocity transducers. The following paragraphs shortly explain the principles of these transducers, but with the greatest emphasis on accelerometers since these are most closely related to the equipment used in this project.

### **Proximity Probes**

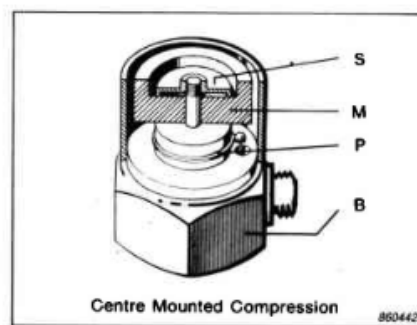
A proximity probe measures the distance between a surface and the probe tip. There are several principles on how this distance is measured; by creating a circuit which includes the distance under inspection, changes in capacitive, magnetic and inductive properties can be used to represent changes in the distance and thus give a measure of the displacement caused by vibration. Proximity probes are commonly used for monitoring of shaft vibration, however for other applications different transducers are preferred due to the limited dynamic range of 40dB. The dynamic range describes the ratio of the largest to the smallest measurable signal [2]. In reality the dynamic range limitation impacts the frequency range and reduces it significantly.

### **Velocity Transducers**

The velocity transducers give a signal proportional to absolute velocity. They usually consist of a coil and a permanent magnet, where the movement of the coil around the magnet causes an electromotive force (emf) to be generated. The emf is proportional to the movement of the coil in the magnetic field, and thus proportional to the velocity. There are also types where the role of magnet and coil are switched, i.e the coil is rigidly mounted to the transducer housing whereas the magnet is seismically suspended, meaning that it is attached to the transducer through a spring. In terms of frequency and dynamic ranges, such transducers are limited to a lower frequency limit of 10Hz and the upper frequency limit of 1-2kHz and with a dynamic range of around 60dB. Velocity transducers have a tendency of being larger and heavier than accelerometers, thus integration of such transducers will be more difficult without affecting the components which are being monitored.

### Accelerometers

In the same way as velocity transducers and velocity, the accelerometers produce a signal which is proportional to the acceleration of a body. The principle of piezoelectricity is the one most commonly encountered in accelerometers for condition monitoring purposes. Some materials have a piezoelectric property, meaning that they generate an electric charge when subject to mechanical strain. This property makes it possible to obtain an electric charge which is proportional to the acceleration of the surface which the transducer is attached to. A common design for accelerometers is the compression type, which utilise the principle of piezoelectricity. In the compression type the piezoelectric element acts as a very stiff spring between a base and a mass. The base, piezoelectric element and mass is clamped together with a spring. With the accelerometer attached to the surface of a vibrating object, the mass in the assembly will lead to deformation of the piezoelectric element, which in turn creates an electric charge. In the world of condition monitoring, accelerometers are the vibration transducers which are used the most; this is mainly due to the advantages of the wide amplitude (dynamic) and frequency ranges which typically are 120dB and 10Hz-10kHz [2]. Furthermore, the accelerometers are compact and thus have little influence on the equipment being monitored. Figure 2.2 shows an example of the compression accelerometer and its components.

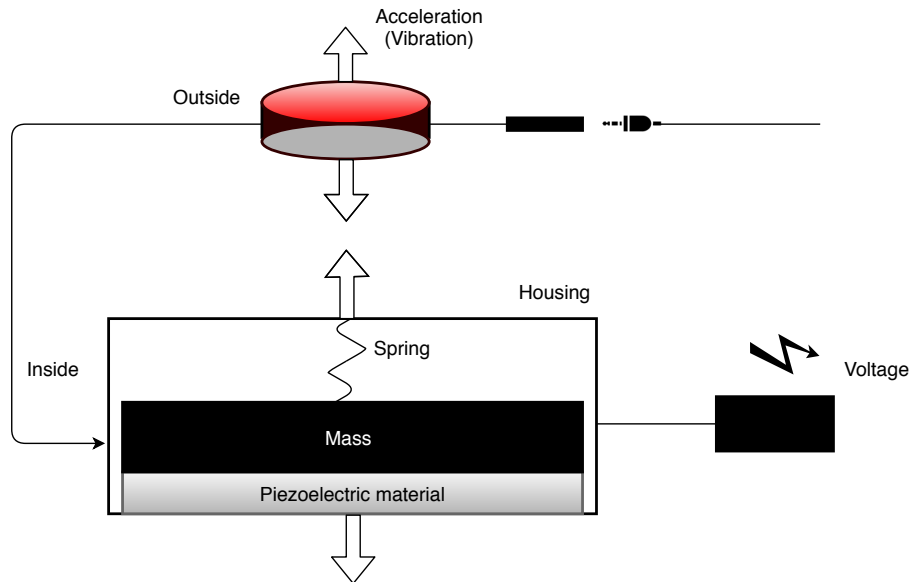


*Traditional Compression Design. M=Seismic Mass, P=Piezoelectric Element, B=Base, and S=Spring*

**Figure 2.2:** A compression accelerometer and its components (Copyright©Brüel & Kjær)

In this project the use of contact microphones in the form of piezoelectric pickups will be investigated for condition monitoring purposes. These are made for and used to collect vibrations on the surface of musical instruments and turn them into electrical signals which can be transmitted through an amplifier. These transducers, as the accelerometers, utilise the principle of piezoelectricity and thus work in much the same way. Accelerometers tend to be a lot more robust which is natural due to the environment they are exposed to. Both the piezoelectric accelerometer and the piezoelectric pickup have

a high output impedance, which generally means that an amplifier is required before anything else can be done with the electrical signals they generate. More on the specific equipment used in this project is discussed in chapter 3, but Figure 2.3 shows a drawing with the general structure of a piezoelectric pickup.



**Figure 2.3:** Structure of a piezoelectric pickup microphone

Requirements to instrumentation for measurement of vibrations on rotating and reciprocating machinery are given in International Organization for Standardization [ISO] 2954 [25]. As the chosen measurement equipment in this project is an acoustic guitar pickup microphone it is clear that there will be some deviations to the requirements. But the main purpose of the requirements in the standard is to reduce the inaccuracies of measurements to within a specific value. Another standard of interest with regards to vibration condition monitoring is ISO 13373-1 [26] which give general guidelines for the measurement procedures. As this project tends to be quite different from a traditional vibration analysis procedure, not all of the elements of these standards are equally relevant, however some of them are discussed further in chapter 3.

### 2.1.5 Vibration Analysis

As mentioned in section 2.1.3, there are many reasons as to why vibrations are good indicators of machine health. All machines vibrate, and emit certain vibration characteristics depending on the operating condition and health condition. This is especially true for machines with rotating components, or components with repetitive or cyclic work patterns. This makes frequency analysis one of the main diagnostic tools for interpretation of vibration signals from machines. As an example, take the rotation of a shaft; if it

rotates at a constant speed the frequency of rotation can be recognised in the frequency analysis, and other components that are dependent on the shaft speed or multiples of the shaft speed can be identified and separated.

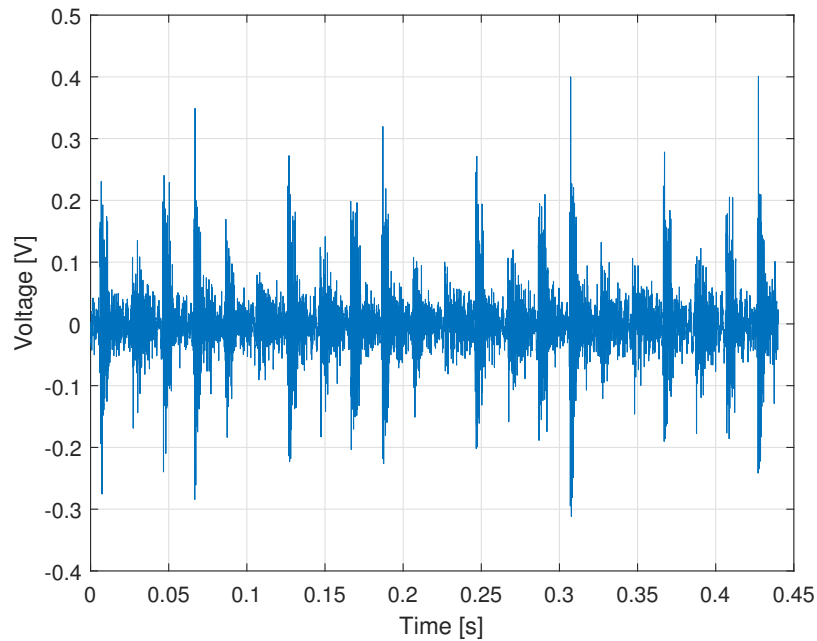
Vibration analysis in its entirety is a very extensive field, and the following paragraphs are only intended as a minor introduction to the area. Furthermore, the goal of this project is to take on a “new” approach to vibration analysis, parting from the traditional time domain and frequency domain approaches. Firstly, a short introduction to vibration in the time domain is given. Vibration is motion around an equilibrium position, and in the time domain the vibration signals appear as a curve which shifts around zero. Figure 2.4 shows an example of a vibration waveform from an internal combustion engine. One of the most useful aspects of the time domain analysis of vibration signals from IC engines is to plot the vibration signals with respect to rotation of the crank shaft. This makes it possible to relate the vibration signatures to the different events of the engine cycle. Information of the vibration signal in the time domain representation can be extracted through mathematical and statistical properties. However, these are most relevant for signals that are stationary in time, meaning that their statistical properties does not change in time [5]. As stated in the introduction (section 1.1), the vibration signals from internal combustion engines are non-stationary, thus the time domain descriptors are less useful for such signals. Nevertheless, the most common statistical descriptors are given in Table 2.1, as some of them might still have relevance for later parts of the project. In Table 2.1  $x_i$  is the signal at any time instant and  $N$  is the total number of time instants.

As mentioned these descriptors provide little information with regards to non-stationary signals and even for stationary signals, more insight can be obtained by translating the analysis into the frequency domain. ISO 13373-2 [28] gives some guidelines on how to process, present and analyse vibration data both in time and frequency domains, and ISO 20816 Part 1 [29] and ISO 10816 Part 6 [30], deals with the evaluation of the vibration measurements. Since the analysis in this project mostly takes place in the frequency domain, vibration analysis in the time domain will be discussed no further. Before presenting general procedures for vibration analysis in the frequency domain, some basic signal processing theory is ascertained, with main focus on the transition from time to frequency domain. The following section takes on much of the theory from the pre-project mentioned in the preface, as this theory also was necessary for implementation and understanding of the methods there.

**Table 2.1:** Time domain descriptors for vibration signals [22], [27]

Name	Formula	Description
Mean	$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$	The sum of the signal over all time instants divided by the number of time instants provides the mean value of the signal  The mean value can be used as a preliminary indicator, where a higher mean should lead to further investigation
Max	$x_{max} = \max x_i $	The max value, also known as the peak value gives useful information in regards to events of short duration like bursts or shocks (peak-to-peak can also be interesting for relation to vibration displacement)
Min	$x_{min} = \min x_i $	The minimum value of the vibration signal
Range	$x_{range} = x_{max} - x_{min}$	The range of the vibration signal
Sum	$x_{sum} = \sum_{i=1}^N x_i$	The sum of the vibration signal over all time instants
RMS	$x_{rms} = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}$	Root mean square, which basically is the squared average. It is useful for signals which have both positive and negative values and is related to vibration energy
Standard Deviation	$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$	Standard deviation provides the average distance of the values from the mean and thus acts as a measurement of the distribution of the data.
Variance	$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2$	The variance is the square of standard deviation, which means that it is the square of the average distance of the values from the mean
Kurtosis	$x_{kur} = \frac{\sum_{i=1}^N (x_i - \bar{x})^4}{(N - 1)\sigma^4}$	Kurtosis is a measure of the “heaviness” of the tails in a distribution. The tails in a distribution are the parts of the distribution that are far away from the mean.  A kurtosis larger than 0 implies that the data is distributed in such a way that samples away from the mean have a large weight compared to the rest of the distribution, and vice-versa
Skewness	$x_{skew} = \frac{\sum_{i=1}^N (x_i - \bar{x})^3}{(N - 1)\sigma^3}$	Skewness says something about the symmetry of a distribution of data, and the well known normal distribution has a skewness of 0
Crest Factor	$C_f = \frac{x_{max}}{x_{rms}}$	The crest factor is the ratio of the maximum value to the RMS value. An increase in the crest factor might be related to impulses or random vibrations, thus by using the crest factor as a monitoring variable faults might be discovered before breakdown occurs
Form Factor	$S_f = \frac{x_{rms}}{\bar{x}}$	The form factor is the ratio of the RMS value to the mean, and alongside the crest factor it gives some information of the waveshape of the signal





**Figure 2.4:** Typical vibration waveform (time domain) of an internal combustion engine. The peaks are related to the combustion events in the engine cylinders

## 2.2 Signal Processing

When the vibration measurements are collected, they are expressed in the time domain. The units may vary, but in this project the amplitude of the vibrations is expressed as a voltage, and the measurements are a time series ranging from the start of recording to the end of recording. From the vibratory movement of the engine and until a waveform of the vibrations can be expressed on a computer, the signal goes through several steps of processing. Most important is the analog-to-digital conversion; since the voltage produced by the piezoelectric material in the pickup microphone is a physical quantity, it has to be converted to a size which a computer can understand. Shortly put, this is done through the *sampling* and *quantisation* procedures, which deals with the discretisation of time and measurement values of the continuous signal respectively [31].

An important aspect of this step of the processing is to have sufficient *sample rate*. *Proper sampling* is used of a sampling process where the analog signal can be recovered from the digital signal. Then the opposite, *improper sampling* is when the analog signal cannot be recovered exactly from the digital signal. In the latter, it is usually the problem of *aliasing* that leads to errors in the transition; aliasing is when sine waves change their frequency in the analog-to-digital conversion. This happens because frequency components in the signal are too "fast" for the sampling rate. To overcome this problem, one must apply *The Sampling Theorem* which states that to obtain a proper sampling

of a continuous signal the frequency components of the signal must be no larger than one-half of the sampling rate. This frequency (one half of the sampling rate) is known as the Nyquist frequency. To clarify, if we have a sampling rate of 4000 samples per second, the frequency components of the signal cannot be larger than 2000 cycles per second [31].

### 2.2.1 Frequency Domain

In the frequency domain the representation of the vibration signal is dependent on frequency rather than time as in the time domain. Transforming the signal from the time domain to the frequency domain includes some mathematical formulas which will be discussed in this section, and the *Fourier Transform* is the core of it all. The basis of the Fourier transform is that every continuous periodic signal can be represented as a sum of sinusoidal waves. The *Discrete Fourier Transform* (DFT) is a mathematical technique applied to digitised signals [31], which compares the content of a signal to sinusoids of different frequencies [32]. Sinusoid is a term used for both cosine and sine waves. The sine and cosine waves are closely related, and each can be expressed by the other by including a phase shift. The importance of sine waves in signal processing lies within the fact that the sine wave retains its shape when added to another sine wave with the same frequency; it produces a new sinusoid with the same frequency. This property of sine waves is sometimes called sinusoidal fidelity, and only apply to linear systems which are time-invariant [31]. Most of this is common knowledge, at least for people working with digital signal processing, but it gives an introduction to how the DFT is carried out. Adopting the terminology and notation from [32] the frequency magnitude- and phase-coefficients can be expressed as the following, where  $\mathbb{R}$  is the range of real numbers.

$$\omega \in \mathbb{R} \tag{2.1}$$

$$d_\omega \in \mathbb{R} \geq 0 \tag{2.2}$$

$$\varphi_\omega \in \mathbb{R} \tag{2.3}$$

As mentioned, the Fourier transform compares the signal to sinusoids of different frequencies. If a sinusoid of a frequency  $\omega$  is resembling an oscillation in the signal, this will give rise to a large magnitude coefficient  $d_\omega$ . Similarly, if a frequency  $\omega$  does not resemble any of the oscillations in the signal, this will lead to a small magnitude coefficient  $d_\omega$ . Thus, for every frequency  $\omega \in \mathbb{R}$ , a magnitude coefficient  $d_\omega$  and a phase coefficient  $\varphi_\omega$  is obtained. The phase coefficient reflects how the sinusoid is shifted to best fit the original signal.

It is not the goal to give a comprehensive explanation of the Fourier transform in this project, thus in-depth explanations of the mathematics will not be given. Necessary formulas will simply be stated, and the reader is referred to other sources for more thoroughly explanations. In general, the (real) discrete Fourier transform takes in a real valued  $N$  point signal, and outputs two  $N/2 + 1$  arrays. The arrays can be thought of as the amplitudes of the sine and cosine waves for which the signal is represented by. For the input signal  $x[.]$  the arrays are given as  $ReX[.]$  and  $ImX[.]$ , where the real array represents the amplitude of the cosine waves, and the imaginary array represent the amplitudes of the sine waves. The arrays are calculated as shown in equation (2.4) and (2.5), where  $k$  is an integer in the range  $[0, N/2]$  and  $i$  runs from  $[0, N - 1]$ . Here the notation is adopted from [31].

$$ReX[k] = \sum_{i=0}^{N-1} x[i] \cos(2\pi ki/N) \quad (2.4)$$

$$ImX[k] = - \sum_{i=0}^{N-1} x[i] \sin(2\pi ki/N) \quad (2.5)$$

This is known as the *analysis equation*, and is an example of the concept of correlation and how the DFT can be calculated mathematically. The time domain signal is multiplied with sine and cosine for different frequencies and the resulting points are added to form the frequency domain signal. However, it is easier to interpret the signal in the frequency domain in terms of magnitude and phase, and the transformation is given by

$$MagX[k] = \sqrt{(ReX[k])^2 + (ImX[k])^2} \quad (2.6)$$

$$PhaseX[k] = \arctan\left(\frac{ImX[k]}{ReX[k]}\right) \quad (2.7)$$

Using Euler's formula, a more formal definition can be given for the discrete Fourier transform [32]

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{(-2\pi i kn/N)} \quad (2.8)$$

for integers

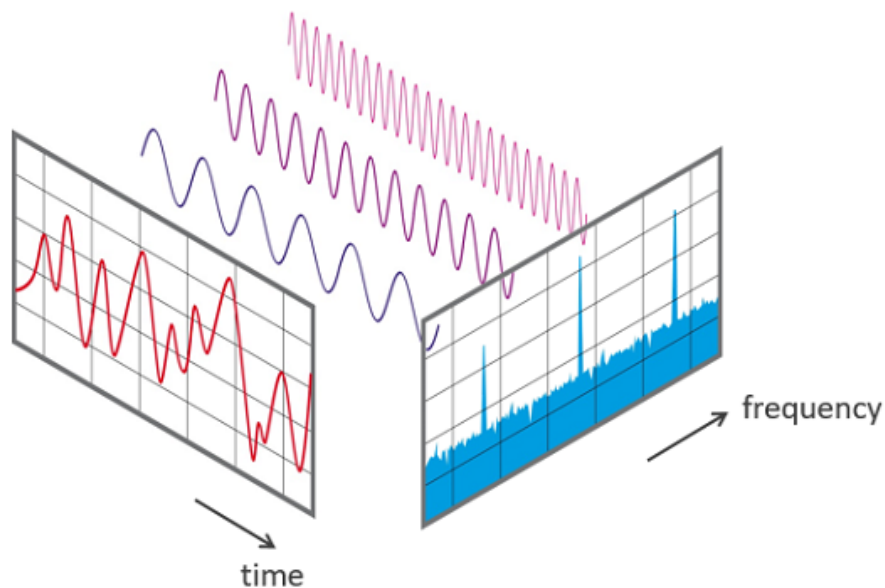
$$k \in [0 : N - 1]$$

where

$$N = \text{Number of samples}$$

$$e^{(-2\pi i k n / N)} = \cos(2\pi k n / N) - i \sin(2\pi k n / N)$$

However, calculating the DFT like this is very time consuming, and therefore many applications today use algorithms to calculate the DFT. One of the most widespread algorithms is the *Fast Fourier Transform* (FFT). What makes the FFT so much faster than the method presented above is that it utilises the redundancy found over sinusoids with varying frequencies, and then computes the Fourier coefficients by recursion [32]. Using the above calculation method would give  $n^2$  operations, while using the FFT will give a number of operations equal to  $n \log_2 n$ . MATLAB has a built-in function for the FFT which will be used throughout the project. No further details will be given of the FFT, but the reader is encouraged to read chapter 12 of [31] and chapter 2.4.3 of [32] for further explanations. Figure 2.5 nicely summarises the transform from the time to the frequency domain.



**Figure 2.5:** Graphical representation of the transition from time domain to frequency domain (Figure by "Phonical", distributed under the Creative Commons Attribution-ShareAlike 4.0 International license [33], [34])

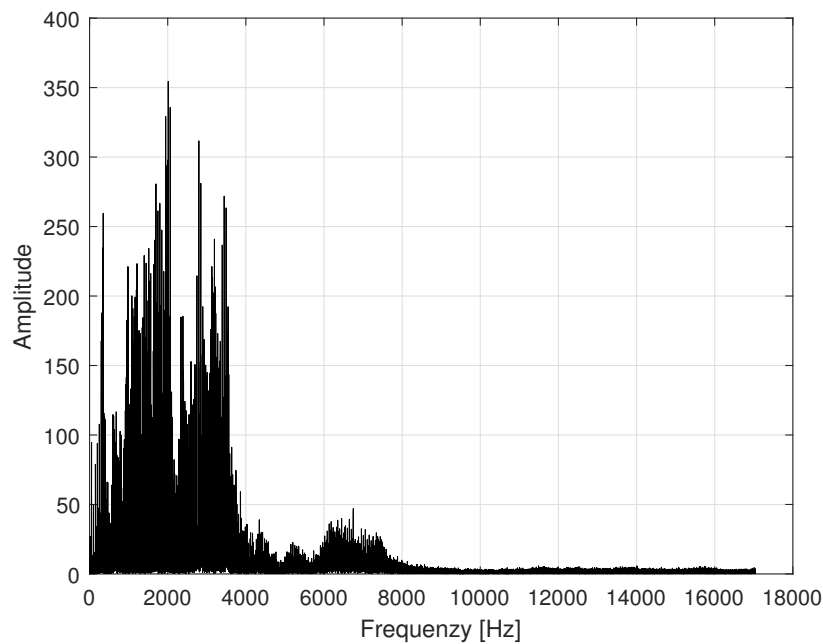
### 2.2.2 Spectra

In this section a short introduction to frequency spectra and their relevance towards the project is presented. A spectrum is a representation of the frequencies contained within a signal, and it is one of the main results from the Fourier transform of a time domain signal. There are several different types of spectra, but the most common are the *amplitude spectrum* and the *phase spectrum*. These are simply the plots of the magnitude and phase angle versus frequency [35]. The amplitude spectrum is also therefore commonly called the magnitude spectrum. The phase spectrum can provide a lot of useful information, however for this project it is the amplitude spectrum that is of most interest.

Another common spectrum representation is the *power spectrum (periodogram)*, or *power spectral density* which it also is called. The name "power" comes from the fact that sound and vibration signals very often are interpreted or expressed as a voltage, and the relation between power and voltage is given by  $Power = (Voltage)^2 / Resistance$ . The power spectrum thus shows how much power is contained within every frequency component of the signal [35]. The representations can differ depending on the value on the axes; the x-axis is usually the normalised frequencies (i.e with respect to the sample rate), and the y-axis is the magnitude given either a linear or logarithmic scale. Furthermore, the spectra can be given as single- or double-sided, where for the single-sided spectra the negative frequencies are omitted. Figure 2.6 shows the (single-sided) amplitude spectrum of a 2,4seconds long recording of a six cylinder diesel engine running at 1000rpm with a load of 20Nm. A zoomed view is given in Figure 2.7, where it is possible to locate several frequencies related to the operating condition. In both these figures, the amplitude has not been normalised according to the signal length, hence the large values. The horizontal axis contains the frequencies up until the Nyquist frequency.

In Figure 2.7 several frequency peaks are pointed out with red arrows. The first one, which is marked with "rpm" is found at approximately 16,6Hz, and is thus related to the rpm of the engine because 1000 revolutions per minute gives 16.67 revolutions per second. The subsequent peaks are related to the *firing frequency* of the engine; a six cylinder IC engine requires two complete revolutions of the crankshaft to complete an engine cycle, which means that there are three combustion events per revolution of the crankshaft. Taking the number of revolutions per second (16,67 at 1000rpm) and multiplying it with three gives 50 combustion events per second, and gives the peak marked with "F" in the figure. The following peaks that are marked are the *harmonics* of the firing frequency; these are integer multiples of the firing frequency. Since the firing frequency is related to the rotational speed, these can also be said to be integer multiples of the rotational speed, which is the basis for the field of *order analysis*.

In order analysis the frequency of rotation is the first order, and the second order is the

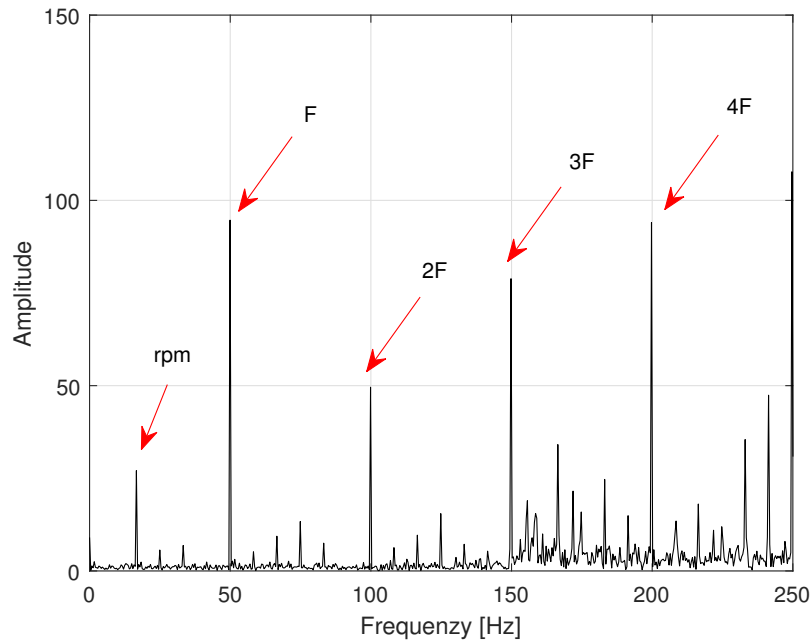


**Figure 2.6:** Amplitude spectrum of a 2.4-second long vibration recording of a six cylinder diesel engine running at 1000rpm and with a load of 20Nm

rotational frequency multiplied by two, whereas the third order is the rotational frequency multiplied by three and so on. In the case of a six cylinder engine, the third order and the firing frequency are the same. Order analysis is a technique used to find information for specific orders [36]. As can be seen in Figures 2.6 and 2.7 a lot of information can be extracted, and this is at the core of traditional vibration analysis. For example, Abed *et al* [6] managed to detect faults in exhaust valve clearances by comparison of amplitude spectra from the healthy and the faulty conditions. However, the spectrum is largely influenced by the firing frequency and its harmonics, which might make it difficult to separate out other frequencies that could be of interest [8]. Furthermore, this method requires knowledge of the engine being monitored and experience with frequency analysis if faults are to be detected, which is why new methods are constantly being developed.

Four more expressions should be presented as their meaning and effect has some relevance towards the chosen methods in this project; *smearing*, *leakage*, *truncation* and *windowing*. If an interval of a (deterministic) signal in the time domain is to be transformed to the frequency domain the following can be said: the interval is a rectangular window which has value 1 for the length of the interval but zero elsewhere<sup>1</sup>. The *truncated signal* is the product of the original signal and the rectangular window. The Fourier transform of the product of two time signals is given by equation (2.9)

<sup>1</sup>Having a rectangular window is thus the same as having no window at all, and it is just used here for a proof of concept



**Figure 2.7:** A zoomed view of Figure 2.6, where it is possible to recognise several frequencies and relate them to physical events.

$$\mathcal{F}\{x(t)\omega(t)\} = \int_{-\infty}^{\infty} X(g)W(f-g)dg = X(f) * W(f) \quad (2.9)$$

which basically states that a product in the time domain is convolution in the frequency domain. The convolution introduces *truncation errors* in the Fourier transform known as smearing and leakage. Shortly put, and without going into too much detail smearing is what happens when frequency components that are closely spaced are difficult to separate, or resolve and leakage is spreading of frequency components over the bandwidth, which can lead to addition of new frequencies not originally present in the signal [35]. There are many different window functions that are better suited to reduce these errors than the rectangular window, and two are shortly mentioned in the next section. The point of mentioning this is that the errors of smearing and leakage can be even more prominent when the signal being analysed is non-stationary, which means that interpreting amplitude spectra obtained through the Fourier transform of vibration signals from IC engines can be troublesome.

### 2.2.3 Time-Frequency Domain

In the introduction of the project, it is stated that the vibration signal of an IC engine is non-stationary, and some relevant research on different signal processing methods used is presented. In this section further discussion will be given as to why it can be

problematic to transform the non-stationary vibration signals from IC engines directly to the frequency domain using the Fourier transform, and what can be done instead.

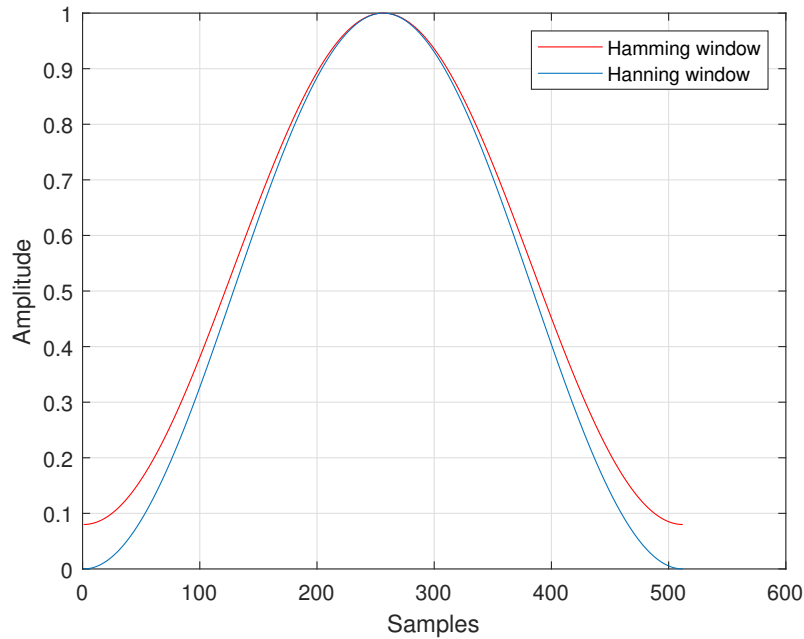
The cycles of an IC engine consist of the combustion happening in the cylinders and the other strokes that are before and after the combustion stroke. More detail on the combustion cycle is given in section 2.4. The engine cycles and the processes within each engine cycle happen repeatedly, however there will be variations from each cycle to the next due to the nature of the combustion events. Thus, the vibration signal from IC engines vary both in time and frequency, and by transforming the signal to the frequency domain by means of the regular Fourier transform the information of the time-variation is lost. Therefore it is useful to analyse vibration signals from IC engines in the joint *time-frequency domain*.

There are several methods for transforming a signal to the time-frequency domain, and as can be seen from the research presented in the introduction the most common are the Short-Time Fourier Transform (STFT), the Continuous Wavelet Transform (CWT) and the Wigner-Ville Distribution (WVD). For this project the STFT will be used mainly due to its easy integration with Principal Component Analysis (PCA) which will be discussed in section 2.3. Therefore no further information will be given on the CWT and the WVD.

What makes the STFT different from the DFT is that it only considers a small section of the signal at a time, to prevent information from being hidden, as may happen when considering the signal in one entity. These sections of the signal, commonly denoted *frames* or *segments* are of equal length, and their length is usually given in a number of samples. A Fourier transform is calculated for each of these frames using the FFT, which results in frequency information at different times of the signal. The result of all the Fourier transforms are gathered and then the frequency spectra can be plotted against time. [32].

Building on the discussion of window functions from the previous section, a window function can be said to be a function that is nonzero for a only a short period of time. This period defines the interval of the section being considered. The *windowed signal* is obtained by multiplying the window function by the frames of the original signal. Window functions are in general used to afford more influence to the data at the centre of the set than to the data at the edges, however this can lead to a loss of information. To overcome this obstacle, it is normal to use overlapping frames, and an overlap of 50% is quite common. This will reduce the loss of information. Because of the window functions, the STFT is also known as "the windowed Fourier transform". Two typical window functions are shown in Figure 2.8, and the effect of a Hamming window on a frame of a vibration signal is shown in Figure 2.9. Furthermore the concept of dividing a signal into frames is shown in Figure 2.10.



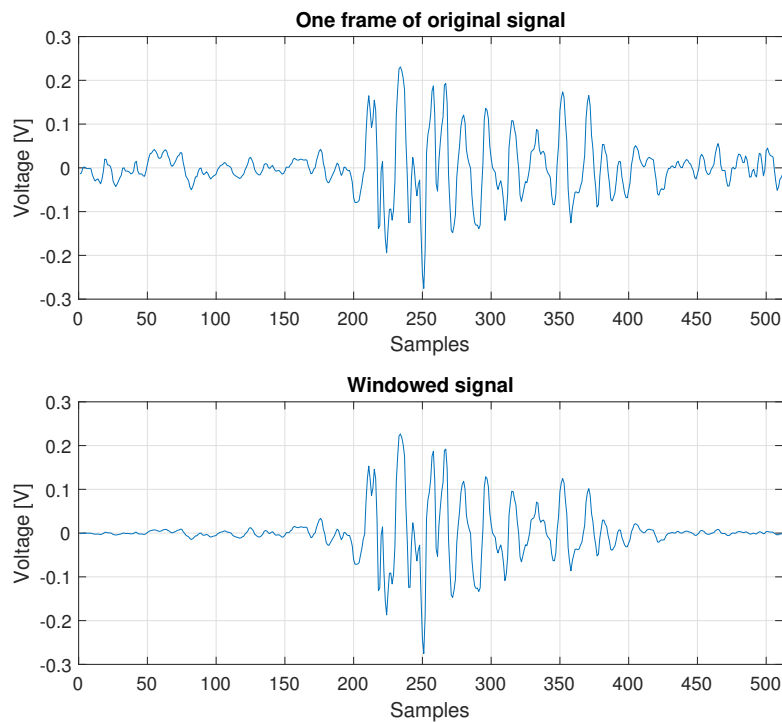


**Figure 2.8:** An example of two different window functions for 512 samples, the Hanning and Hamming windows.

When deciding for the frame length for a given application there are several things one should bear in mind. First of all the sample rate, i.e the number of samples collected per second should be considered. The sample rate together with the frame length determine the time interval of each frame. Next, one should contemplate on the physical origin of the signal to find a frame length that best suits the signal at hand. The following provides an example of this; returning to Figure 2.9, which contains one frame with a length of 512 samples. With the background knowledge that this is from an IC engine, it is possible to relate the impulses centred around 250 on the horizontal axis to a combustion event. If a frame length of 256 samples was to be used, the combustion event would be split in half, or at least spread over two subsequent frames. This is not necessarily negative, however it is something that should be thought of when selecting the frame length.

An other thing related to the selection of the frame length is the trade-off between *time resolution* and *frequency resolution*. Increasing the frame length gives an increased frequency resolution, i.e that closely valued frequencies are easier to separate. Shorter frame lengths gives better time resolution which means that it is easier to detect when frequencies change [32]. Thus one must make a choice of which resolution is most important for the given application. This trade-off, or dilemma is by some considered a weakness of the STFT [37], [35] which has led to the development of methods like the CWT.

Lastly, the way to visualise the result of the STFT is by using a spectrogram. A

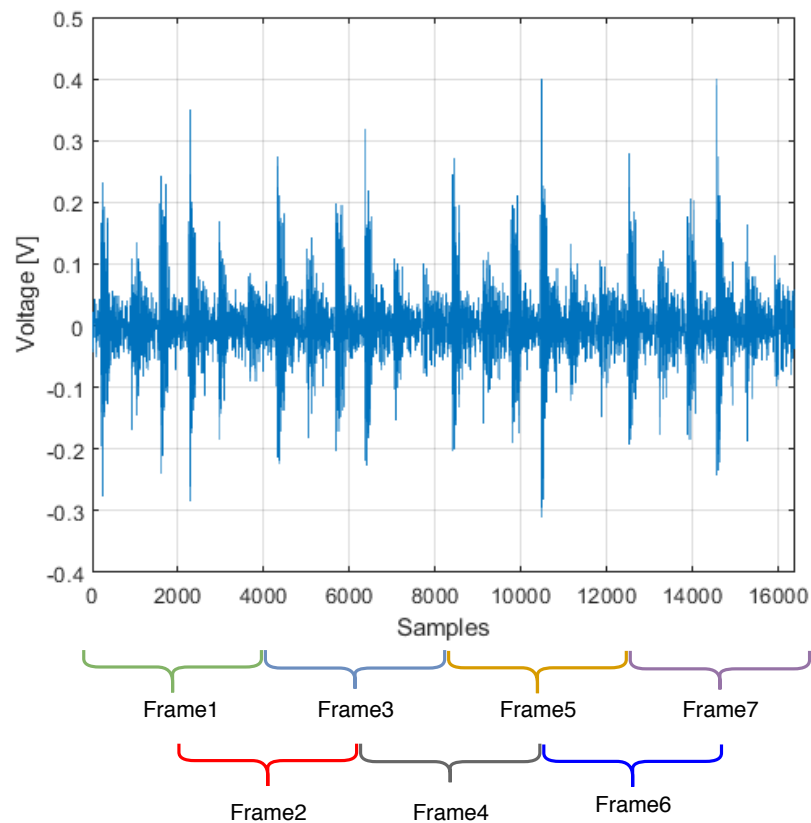


**Figure 2.9:** The original and windowed signal of one frame of length 512 samples from the vibration waveform presented in Figure 2.4

spectrogram is a representation of how the magnitude of the frequencies change in time [31]. It is thus a two-dimensional representation of the magnitude, because it shows the dependency in both time and frequency. The spectrogram is composed of the frequency spectra of all the frames which the signal is divided into; the spectrum of each frame represent one vertical line in the spectrogram image. The spectrogram is usually presented with frequency along the y-axis, time along the x-axis, and then colour or light intensity which represents the magnitude of the frequencies. Low magnitude is represented by light colours or white fields, whereas high magnitude is given by dark colours or black and grey fields. For spectrograms with colours, it is good custom to include a colorbar showing which colours represent which magnitudes.

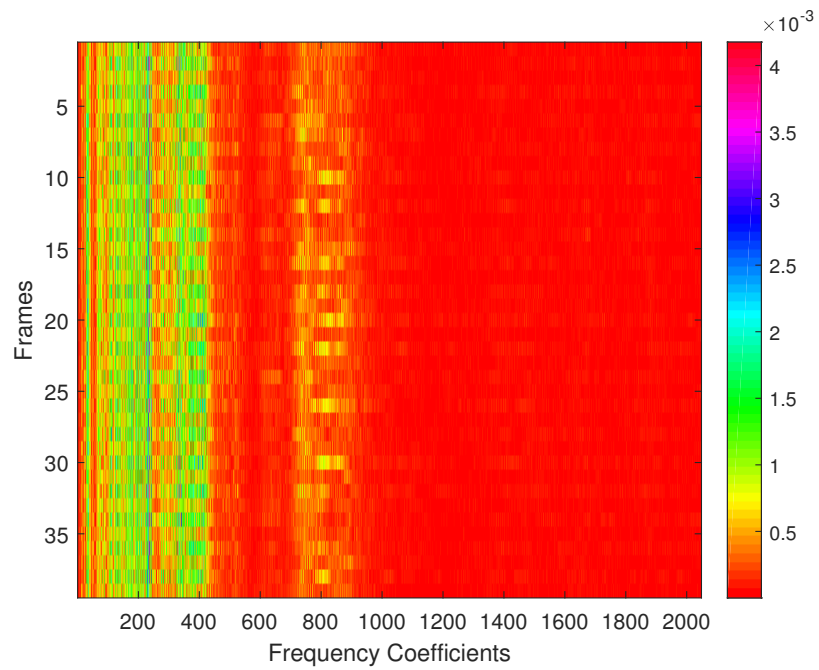
Figure 2.11 shows an example of a spectrogram<sup>2</sup> based on the same recording as was used for the amplitude spectrum in Figure 2.6. A frame length of 4096samples and 50% overlap has been used for the transformation to the time-frequency domain. The values along the horizontal axis are here denoted "frequency coefficients" i.e they are not frequencies directly, but they represent the frequencies from 0 to the Nyquist frequency. Thus, to get the frequency at an occurrence in the figure, the value along the horizontal

<sup>2</sup>In literature, a spectrogram is given as the squared magnitude of the STFT [32], however here the word spectrogram is used when visualising magnitudes that are not squared as well



**Figure 2.10:** The principle of dividing a signal into frames. The figure shows a vibration signal from an IC engine with a total of 16384 samples, frame length of 4096 samples and with 50% overlap between successive frames

axis has to be multiplied by the the Nyquist frequency divided by the number of frequency coefficients (which is determined by the frame length). In this recording the sampling rate is 34,1kHz, which makes the Nyquist frequency 17,05kHz. So, to give an example there is a line with high intensity at approximately frequency coefficient 242; using the relations provided above, this gives a frequency of roughly 2015Hz. This can be validated towards the amplitude spectrum in Figure 2.6, where the largest peak appears between 2014Hz and 2015Hz.



**Figure 2.11:** Spectrogram of the vibration recording of a six cylinder diesel engine. The frames on the vertical axis represent the time-dependency, whereas the frequency coefficients along the horizontal axis represent the frequencies, although not directly. The recording is the same as in Figure 2.6

## 2.3 Multivariate Analysis

This section moves on to some other methods used in this project, which all fall under the common name of multivariate analysis. The main focus will be on Principal Component Analysis and regression methods such as Principal Component Regression (PCR) and Partial Least Squares Regression (PLSR). As PCA was used for the pre-project done during the autumn semester of 2017, some of the theory presented there will be reused in the following paragraphs.

Some terminology must be introduced before delving into the Principal Component Analysis. A *multivariate* method is a method in which several variables are measured simultaneously as opposed to *univariate* methods, where only one variable is measured [38]. Multivariate analysis (MVA) is useful when relationships between variables becomes more complex, and when investigating one variable at a time is no longer practical. *Correlation* is a dimensionless property that describes how variables or measurements vary together, thus two variables are correlated if variations in values of one of the variables leads to variations in the values of the other variable. If this is not the case, the variables are said to be uncorrelated. *Covariance* describes the influence one variable has on others and its definition is very similar to that of correlation. Given two variables  $X_1$  and  $X_2$ , their covariance and correlation is mathematically expressed as [39], [40]

$$\text{Cov}(X_1, X_2) = E[(X_1 - E[X_1])(X_2 - E[X_2])] \quad (2.10)$$

$$\text{Corr}(X_1, X_2) = \frac{E[(X_1 - E[X_1])(X_2 - E[X_2])]}{(\sigma_{X_1} \sigma_{X_2})} \quad (2.11)$$

where

$E(X)$  = expected value (mean)

$\sigma_{X_1}$  = Standard deviation in  $X_1$

$\sigma_{X_2}$  = Standard deviation in  $X_2$

Formulas for mean and standard deviation are given in Table 2.1. The discussion of terminology will be concluded by introducing *causality*; causality describes the relation between cause and effect. An example of causality can be given by thinking of two random processes; if the first process is the reason for the second process to happen, and the second process is to some degree reliant upon the first, there is causality between the two processes.

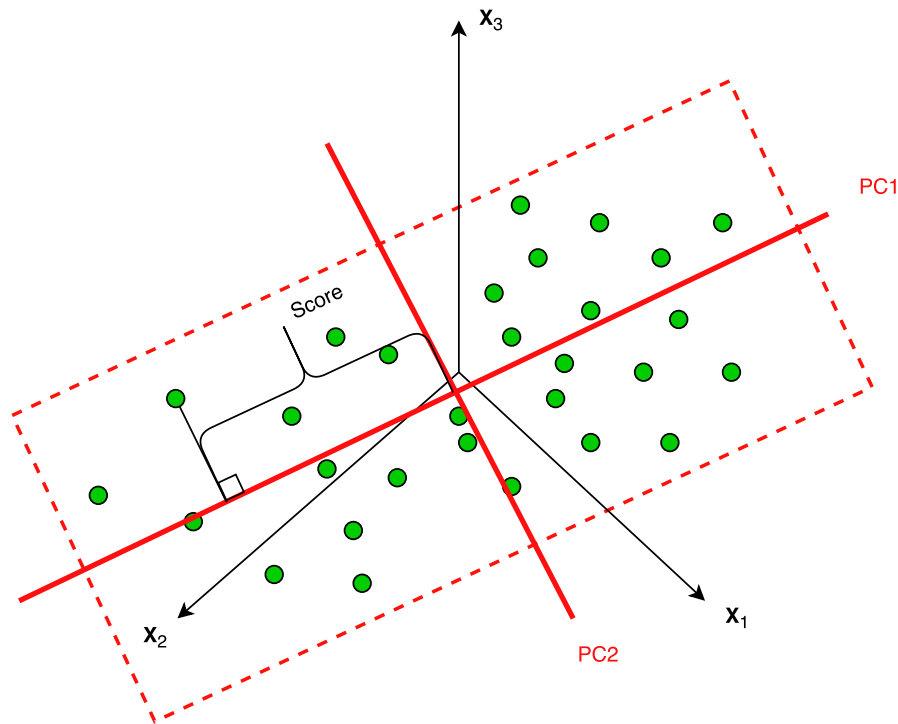
### 2.3.1 Principal Component Analysis

Principal component analysis is a statistical multivariate method used to find and understand relationships in large data sets [38]. It does this by reducing the dimensionality of the data by replacing the original variables with *latent variables*. The latent variables are "hidden" variables, meaning that they do not necessarily represent physical factors; they are linear combinations of the original variables. These linear combinations, or latent variables are known as *principal components* (PCs). PCA is useful if the original variables are correlated. One of the main goals of the PCA is to reduce the dimension of a set of data without losing too much of the information contained within it; the principal components are therefore uncorrelated and chosen on basis of the largest variation in the the original variables. The first principal component represent the largest variation in the data set, and the second principal component represent the second largest variation in the data set and so on [41].

Assuming the data is represented in a matrix, PCA utilises the principle of projection; the columns of the data matrix represent the original variables and each of the variables defines an axis in a coordinate system, which in total defines the variable space. The rows of the data matrix are the *objects*, *samples* or *observations* which are shown as points in the variable space [42]. These observations could for example represent measurements at a particular point in time. To best explain the variation of the observations in the variable space, a "trend-line" or "best-fit-line" can be drawn through the swarm of points; this line summarises and estimates what the points are describing. This first best-fit line is in fact the first principle component, and the distance from the *90degree* projection of an observation on the line and to the origin (of the latent variable space) is the *score* of that exact observation. Every observation is awarded its own score value, which can be either positive or negative. The direction of the line is therefore sometimes describes as "the direction of maximum variance of the projections onto the line" [43, p. 294]

The second PC is found in the same way, however it has to be perpendicular to the first PC. The second PC will point in the direction of the second largest variance, and the scores for this component are the distances from the origin to the projections of the observations along this line. The PCs are composed of two elements; The total number of score values (one for each observation) collected in a column vector denoted  $t$  and the direction of the component given by a direction vector denoted  $p$ . The direction vectors are unit column vectors known as *loadings*, and they act as a link between the original variable space and the latent variable space [43]. After having found two PCs, a plane can be defined where the two PCs act as a new coordinate system often called the latent variable coordinate system. This plane is the best description of the data we can obtain using two PCs. The theory presented in this and the previous paragraph is illustrated in

Figure 2.12. Here a 3D-space is shown with observations given as green dots; principle components are given as red lines, and the picture shows the projection of a score value onto the first PC.

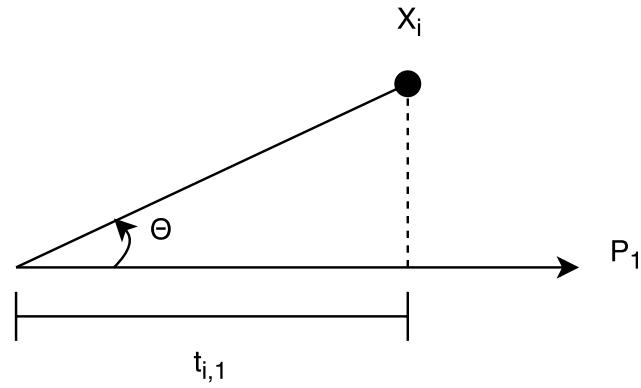


**Figure 2.12:** Illustration of Principal Components and the score value [42]

### 2.3.2 Mathematical Derivation of PCA

The following gives a deeper insight into the maths behind the PCA. It does not have any direct relevance to the project, other than that it provides a greater understanding of the method. Thus, skipping this section and going straight to section 2.3.3 where larger emphasis is put on interpreting the results of the PCA can be done if desired. The explanation that is presented here is obtained from [43, p. 296]. PCA is based on linear algebra and statistics, but can also be explained from a geometrical point of view. Suppose we have a data matrix  $X$  of size  $(N \times K)$ . Denote a row from the data matrix as  $X'_i$ , so that  $X'_i$  becomes a  $1 \times K$  vector.

In Figure 2.13 the following is given: the vector  $X_i$ , the direction vector  $P_1$  which is a unit vector, and the distance  $t_{i,1}$  which is the score value for the  $i_{th}$  observation with regards to the first principal component. Using the following relations of the cosine



**Figure 2.13:** Geometrical representation of scores

$$\cos(\theta) = \frac{\text{adjacent}}{\text{hypotenuse}} = \frac{t_{i,1}}{\|X_i\|}$$

$$a \cdot b = \|a\| \cdot \|b\| \cos(\theta) \quad (2.12)$$

$$\cos(\theta) = \frac{X_i' P_1}{\|X_i\| \cdot \|P_1\|}$$

$$\frac{t_{i,1}}{\|X_i\|} = \frac{X_i' P_1}{\|X_i\| \cdot \|P_1\|}$$

$$t_{i,1} = X_i' P_1 \quad (2.13)$$

It can be seen that the score value is a linear combination of the  $i$ th row of data  $X_i$  and the vector  $P_1$ . To calculate all score values for the same observation, one can do the computation

$$t_i' = X_i' P$$

and for a complete data matrix, the computation takes the form

$$T = XP \quad (2.14)$$



where

$T$  = The scores for all the observations given as a  $(N \times A)$

$X$  = The original data matrix  $(N \times K)$

$P$  = The loadings  $(K \times A)$

$A$  = The number of principal components

The geometrical point of view gives an easy and understandable introduction into the world of PCA, however it is natural to talk a bit about the *covariance matrix*, which forms the basis for the PCA. The following is obtained from [44]. Let  $X$  be a data matrix of size  $(m \times n)$ , where  $n = 30$  represent the samples and  $m = 3$  represent the variables. The  $i_{th}$  measurement can be represented as a  $(3 \times 1)$  vector  $\vec{x}_i$ . The expected value, or mean was presented in equations (2.10) and (2.11) for covariance and correlation. Formulas for variance and mean were also given in Table 2.1, however to follow the notation of the literature that is used as a basis here, they are given as

$$E[X] = \vec{\mu}_x = \frac{1}{n}(\vec{x}_1 + \vec{x}_2 + \vec{x}_3 + \dots + \vec{x}_n) \quad (2.15)$$

$$Var(A) = \frac{1}{n-1}((a_1 - \mu_A)^2 + \dots + (a_n - \mu_A)^2) \quad (2.16)$$

As a pre-processing it is often common perform data-centring the origin. This is done by subtracting the mean from each measurement vector. Using this, a new matrix is defined as

$$B = \begin{bmatrix} \vec{x}_1 - \vec{\mu}_x & \vec{x}_2 - \vec{\mu}_x & \dots & \vec{x}_n - \vec{\mu}_x \end{bmatrix} \quad (2.17)$$

Then, defining the covariance matrix as

$$S = \frac{1}{n-1}BB^T \quad (2.18)$$

It can be seen that this makes sense in relation to the definition of the covariance in equation (2.10). The  $S$ -matrix will be a symmetric  $(m \times m)$  matrix, with the entries on the diagonal being the variances of the variables, and the other entries being the covariance between variables. This can be illustrated by an example which is adopted from [44].

Given the following

$$\vec{x}_1 = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}, \vec{x}_2 = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}, \vec{x}_3 = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix}, \vec{\mu}_x = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{bmatrix}$$

This gives

$$B = \begin{bmatrix} a_1 - \mu_1 & b_1 - \mu_1 & c_1 - \mu_1 \\ a_2 - \mu_2 & b_2 - \mu_2 & c_2 - \mu_2 \\ a_3 - \mu_3 & b_3 - \mu_3 & c_3 - \mu_3 \\ a_4 - \mu_4 & b_4 - \mu_4 & c_4 - \mu_4 \end{bmatrix} \quad (2.19)$$

Then the first entry of S would be given by

$$S_{1,1} = \frac{1}{3-1}((a_1 - \mu_1)^2 + (b_1 - \mu_1)^2 + (c_1 - \mu_1)^2)$$

whereas the first entry of the second row would be given by

$$S_{2,1} = \frac{1}{3-1}((a_1 - \mu_1)(a_2 - \mu_2) + (b_1 - \mu_1)(b_2 - \mu_2) + (c_1 - \mu_1)(c_2 - \mu_2))$$

It is easy to see that the entry  $S_{1,1}$  is the variance of the first variable, whereas the entry  $S_{2,1}$  is the variance between the first and the second variable. To wrap this up, the eigenvalues of S (in decreasing order) can be found from the relation

$$A\vec{v}_i = \lambda_i\vec{v}_i \quad (2.20)$$

$$(A - \lambda I)\vec{v} = 0 \quad (2.21)$$

The corresponding eigenvectors to the eigenvalues of S are the principal components. Hence, the relation between statistics and linear algebra has been established. The most common algorithms for calculating PCA are the Singular-Value Decomposition (SVD) and the nonlinear iterative partial least squares (NIPALS) algorithms, however these will not be discussed further here.

### 2.3.3 Important Plots and Interpretation of Plots

PCA provides many powerful graphical representations of the data, however the most important results of the PCA analysis is the scores, the loadings and the explained or residual variance. Scores and loadings have already been introduced, but the explained variance remains to be presented. The explained variance shows how much information is carried by each principal component, and the residual is the complementary; that is how much variance remains to be explained after each PC [42]. Thus, the importance of a PC is determined by the size of its residual variance.

#### Score plot

The score plot often appears as a map of the scores of one PC plotted against the scores of another PC. This provides a scatter plot of the samples, and gives an illustration of the distribution of the data. Although you could plot the scores of any of the PCs against each other, the most interesting is the plot of the scores of PC1 and PC2, because these are the components with the largest explained variance. The score plot can be used to reveal patterns, groupings and outliers. Outliers are samples that are very different from the rest, meaning that they appear far away from all the other samples in the score plot. Outliers could be a result of errors in the data collection, and should be considered removed from the data set. As mentioned earlier, each sample has its own score value for each PC, and this value represents the location of that sample along the PC. It can be a good idea to look at the score plot of two PCs at the same time as a loadings plot of the same PCs, because this can provide insight into which variables that lead to differences between samples.

#### Loadings plot

The loadings of a PC describe how much each variable contribute to that PC, and is a value between -1 and +1. The value of the loadings is connected to its geometrical representation, which is the same as the relation given in equation (2.12), i.e a loading is the cosine of the angle between the corresponding PC and the variable [42]. A small angle gives a large value for the loading and indicates that the connection between the variable and PC is high. When interpreting loadings plot, it is wise to look for variables with loading values close to 1 or -1 for each PC, because this means that the PC describes the variable in a good way. Depending on the variables of the application, it can sometimes be easier to look at the loadings plot as a line plot, i.e look at the loadings for one PC at a time.

To end the discussion of the PCA a short summary is given. PCA can be applied on a

matrix of data ( $N \times K$ ), where the goals can be one or more of the following: simplification, data reduction, modelling, outlier detection, variable selection, classification, prediction [45]. PCA can be used to find an underlying data structure in a set of data and separate this from the noise in the data. PCA thus provides a *bilinear* matrix decomposition of the data matrix, where the scores and the loadings of the principal components are the main elements.

When interpreting a PCA model there are several things to pay attention to: firstly the quality of the obtained model must be assessed by looking at the number of principal components and the variance explained by each of them. Further on, outliers should be detected and investigated closer. Before using the model for prediction, it should be validated. Validation is discussed further in section 2.3.8. In the interpretation of scores and loadings, if the loading of a variable and the score of a sample on the same PC have positive values, the value of the sample is higher than average for that variable, and the other way around; if the signs are negative the value of the score is lower than the average for that variable. However, one should be careful to interpret loadings with small values, because this means that the variable is badly described by the PC.

### 2.3.4 Regression Methods

This section shortly describes the principles of three different regression methods that will be employed in the project, namely Multiple Linear Regression (MLR), Principal Component Regression and Partial Least Squares Regression. It is not the intention here to go into very specific details about these three methods, thus the mathematical representations will be kept at a minimum. Instead the focus is directed to what separates these three methods, how their results can be interpreted and what makes a good regression model.

Regression is a common name given to methods where the intention is to establish a relationship between variables. The regression methods used in this project are linear, thus the desired relation is that one variable can be approximated as a linear function of the other. These variables go by many names, but frequently encountered names are *predictors* (X-variables) and *responses* (Y-variables). They also sometimes go by the name "independent" and "dependent" variables, respectively. The established relationship can be used either to gain more insight into the variables or to predict new values. Prediction can be advantageous in cases where measurements are hard to obtain, or for economical reasons [42].

It is possible to do both *univariate regression* and *multivariate regression*; the difference is whether one or several prediction variables are used. In this project more than one predictor variable will be used most of the time. A good regression model is characterised

by (1) it extracts all relevant information, and (2) it does not model noise (does not overfit the model). Most regression methods seek to minimise the sum of the squared residuals, where the residuals are the differences between measured and predicted Y-variables [46]. Using the notation in [46], the general form of the linear (multivariate) regression model can be given as

$$f(X) = \beta_0 + \sum_{j=1}^p X_j \beta_j \quad (2.22)$$

where

$$f(X) = Y = \text{Response variable(s)}$$

$$\beta_0 = \text{intercept term}$$

$$\beta_j = \text{Regression coefficients}$$

$$X_j = \text{Predictor variables}$$

Before using a regression model to predict new values it is of great importance to validate and assess the quality of the model. Validation will be discussed in section 2.3.8, however there are some other characteristics or features common for all the regression methods used in this project that could and should be addressed to evaluate the quality of the model. Initially it can be a good idea to evaluate if the model is overfitted (has modelled noise), or if the model has a lack of fit, i.e there are nonlinearities in the data. For example, the residuals should have a random distribution, and show no trend [42]. Furthermore, one should look for outliers and consider them removed if they have large influence on the model.

Moreover, when assessing the quality of the model there should be a straight line relationship between predicted and measured values of the Y-variable, i.e an increase in the value of the measured Y-variable should give a corresponding increase in the value of the predicted Y-variable (for example when looking at a plot of measured versus predicted values). This line relationship should have a slope close to 1 to qualify as a "good fit". Some other statistical measures that indicate the quality of the model are R-squared, adjusted R-squared, R2(Pearson) and root-mean-square error (RMSE). The R-squared, which also goes by the name of *coefficient of determination*, gives a measure of the correlation between predicted and measured values, and ranges from 0-1, where a value close to 1 is indicative of a good model. The R-squared value comes from the sum  $1 - \text{Residual Y-variance}$  [42]. The adjusted R-squared, which is the R-squared for

validation, is a measure of how good the fit of future predictions are expected to be, and also range between 0-1, where 1 is a good result.  $R^2$ (Pearson) is simply the square of the correlation value. The RMSE can be given both for calibration and validation, and it says something about the spread of the samples in relation to the regression line [42]. More specifically there are three main types of RMS: root mean square error of calibration (RMSEC), root mean square error of cross validation (RMSECV) and root mean square error of prediction (RMSEP). The RMSEC is the spread of the samples around the regression line for the calibration samples, i.e the average deviation between reference and predicted values. RMSECV is the spread of samples around the regression line for cross validation, and is thus an estimate of the average prediction error. RMSEP can be thought of as the average prediction error, and it is the best measure on how a model will perform on future samples. The RMSEP is obtained with a separate test set, where the samples have not been used in calibration of the model. The following sections shortly outline the three regression methods that will be used in this project, where PCR and PLSR are bilinear methods.

### 2.3.5 Multiple Linear Regression

Multiple Linear Regression is a "standard" simple regression method where several predictors (X-variables) are used to predict one response (Y-variable). MLR should only be used if there are few predictors, and a rule of thumb gives 20 variables or less as a limit. As discussed in the previous section, most regression methods act on a "least squares" objective, i.e they strive to minimise the residual sum of squares. MLR works towards this directly by finding regression coefficients that minimise the error. Three more things should be noted about the MLR method: (1) if the predictor variables are not linearly independent of each other, the resulting regression can end up being unstable, (2) the number of sample must be larger than the number of variables, and (3) when using MLR on noisy data, one should pay extra attention to overfitting [42].

### 2.3.6 Principal Component Regression

Principal Component Regression is a method which consist of two main steps (1) Principal Component Analysis, and (2) Multiple Linear Regression. PCA was introduced in section 2.3.1, and there are many similarities here as PCA is one of the main steps. In this method the scores of the principal components from the PCA are used as predictors in the Multiple Linear Regression. One of the reasons that this works good is that the PCs are orthogonal; as mentioned the MLR can be unstable with variables that are linearly dependant, but by using the PCA first this is no longer a problem. In fact, PCR works better when the

original X-variables are correlated and express the same type of information. The scores and loadings of the PCR are the same as for the PCA, thus the reader is referred to the section of PCA for further explanation of these.

### 2.3.7 Partial Least Squares Regression

Partial Least Squares Regression is the third and final regression method presented here. It has many similarities with the two previous methods, however it has some characteristics that may require some extra explanation. Unlike PCR, PLSR uses information from both predictors and responses simultaneously when establishing the relationship between them [46]. As PCA, PLSR finds latent variables in X, however different from PCA it also finds latent variables in Y at the same time and uses this information to build a model with the best predictive ability [42]. The components created in PLSR are called *factors*, but they have many similarities with the principal components of PCA. In PLSR it is possible to model several responses simultaneously, which is in contrast to the two other methods where only one response can be modelled at a time.

A short explanation of the PLSR factors and associated scores and loadings is necessary to gain a brief understanding of the method, as there are some deviations between the scores and loadings of PLSR and the previously discussed scores and loadings of PCA. The largest difference is that PLSR has two sets of scores and loadings, since latent variables are found from both X-variables and Y-variables. These are namely the T- and U-scores and the W- and Q-loadings. T-scores and P-loadings are related to the X-variables and are very much like the scores and loadings of PCA. Thus, they can be interpreted the same way as for PCA. U-scores and Q-loadings are a bit different; they are related to the Y-variable(s). If we say that the T-scores summarise the parts of X used for prediction of Y, then U-scores summarise the parts of Y explained by X, and the Q-loadings give the relationship between the T-scores and the Y-variable(s) [42].

In general this leads to more diagnostic tools and a greater room for interpretation of the regression model [11]. PLSR is good towards modelling of data that is influenced by noise, and often reaches the same result as PCR with fewer components (factors). Truly, there is a lot more to be said of PLSR, however the interested reader is referred to other sources for a deeper explanation of the method, for example [11] which contains some practical examples, or [46] which has a deeper focus on the statistical and mathematical aspects of the method.

### 2.3.8 Validation

To complete the theory of multivariate analysis, a short explanation of the importance of validation and how it can be done is presented here. The following is mostly based on the information on validation given in [42]. In general there are two steps that are crucial to the creation of a model: (1) calibration and (2) validation. Therefore, it is normal to talk about a *training set* which consist of the samples used for calibration and a *test set* which contains the validation samples. It is important that the calibration samples and validation samples are separate, i.e that the test set does not contain samples used for calibration. As stated in the respective sections, validation is necessary both for the PCA models and the regression models. Without validating the models, one cannot be sure if future predictions or classification is to be trusted. Thus, validation is the same as testing whether a model can be applied for new data sets. Two well-known validation methods are *test set validation* and *cross validation*.

Test set validation is exactly as described above; the data is divided into a set used for calibration and a set used for validation, namely the test set. The test set should contain samples that are representative of the variations in the complete data set, and the number of samples should be 20-40% of the total number of samples. After a model has been created, the samples in the test set are fed into the model, and their predicted values are compared to the measured values. The residuals from prediction can then be used to calculate the root mean-square error of prediction, which gives the uncertainty of future predictions. Test set validation is a good method for data sets with many samples.

The cross validation method differs from the test set validation method in that all samples are used both for calibration and validation, however not in the same computation. The way it works is that a model is computed several times, each time with different samples kept out from the calibration set. For each computation, the predicted values are calculated for the samples that were kept out from the calibration, and compared to the corresponding measured values. This procedure is carried out until all samples have been kept out of the calibration set once. The residuals are kept for each computation, and after the last computation they are put together to form the RMSECV. Cross validation thus makes better use of the samples in data sets where the number of samples are low. The Unscrambler®X provides several types of cross validation, for instance full cross validation, where each sample is kept out separately, and segmented cross validation, where groups of samples are kept out simultaneously. A third validation method is known as leverage correction. Without going into more details on this method, it can be mentioned that when used for MLR models, it is comparable to full cross validation. Furthermore, when leverage correction is being used, the model error is given as the root mean square error of estimation (RMSEE).



## 2.4 Internal Combustion Engines

The following section gives a short introduction to the internal combustion engine and its main components, the four-stroke cycle, and the largest sources of noise and vibration in engines. This is deemed relevant for the project, as experimental data are recorded on a diesel engine, and several references are made to components and processes in the engine.

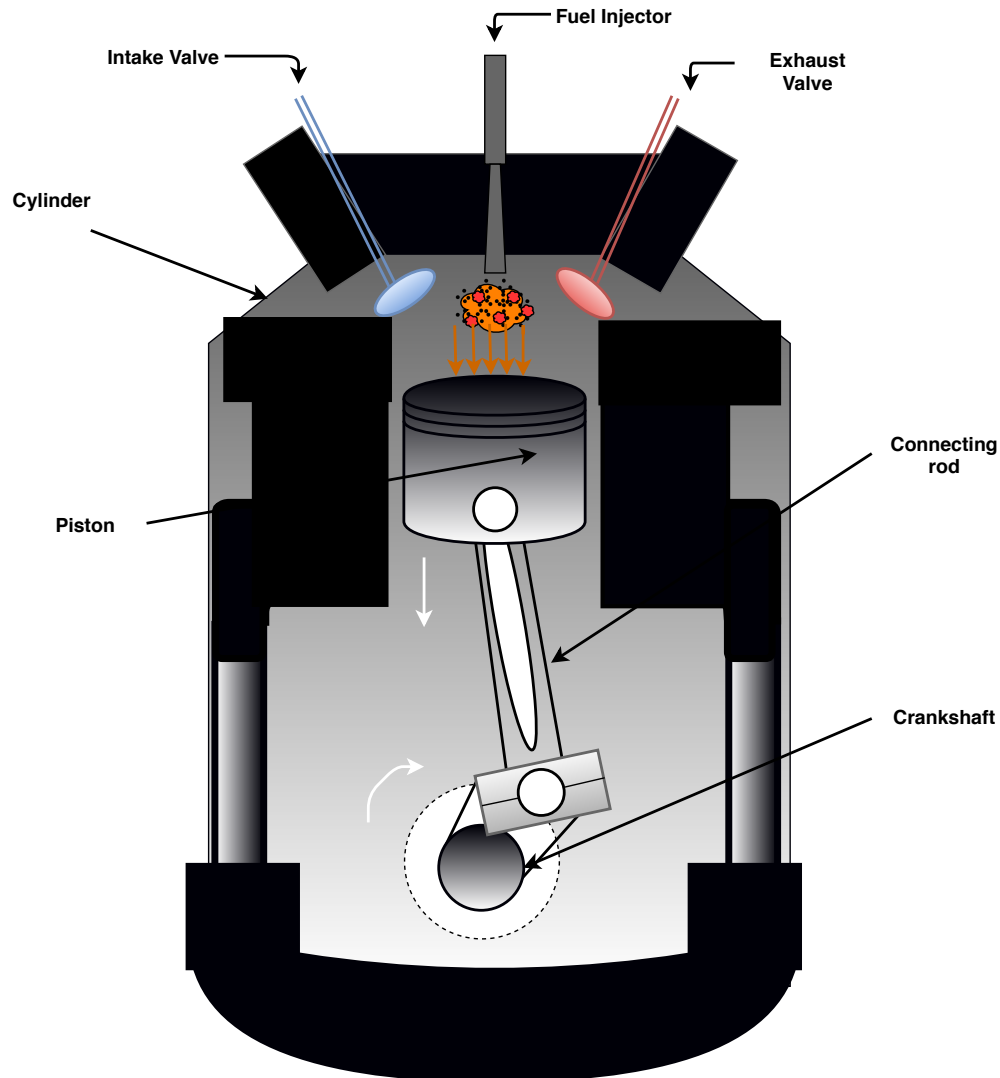
### 2.4.1 Principle of Operation

Internal combustion engines is a common name used for all engines where the combustion process happens internally and the result of the combustion process creates a force which acts on a mechanical component inside the engine. The combustion process is (usually) the ignition of a mixture of (fossil) fuel and air, which results in a gas expansion [47]. There are several engines which falls within the category of internal combustion engines; namely the piston engine, several types of rotary engines and gas turbines. For this project it is the piston engine (reciprocating engine) which is relevant. In the piston engine the expansion of gas "forces" the pistons down, and creates a rotary motion due to the assembly of the engine, which is further explained in the coming sections.

There are two main types of the reciprocating engine; the spark ignition (SI) engine, and the compression ignition (CI) engine. The spark ignition engines use gasoline as a fuel, and the mixture of gasoline and air is ignited by means of a spark plug, which exerts a spark at a certain point of the engine cycle. Compression ignition engines generally use diesel as a fuel, and the mixture of diesel and air is ignited through compression, i.e at a certain point in the engine cycle the temperature of the mixture reaches a value in which the mixture self-ignites. In newer CI engines, diesel is injected at a high pressure after the air has been compressed to a temperature that is higher than the ignition temperature.

The main components of an automotive piston engine are the following: **cylinders, pistons, connecting rods, crankshaft, camshaft, valves**. some of these components are illustrated in Figure 2.14. The **cylinders** are the containers of the combustion process; depending on the application the number of cylinders vary. For a motorcycle, two cylinders is quite common, whereas four cylinders and more is typical for automobiles. Together the cylinders form the cylinder block. Below the cylinder block another large structure of the engine can be found, namely the crank case; it contains the **crankshaft** of the engine. The crank case and cylinder block together forms the engine block, and in most cars they are one structure. The crankshaft is the part of the engine which turns the energy of the combustion process into rotary motion. Along the crankshaft there are several connecting rods (one for each cylinder) that join the pistons in the cylinders to

the crankshaft. As the combustion "forces" the pistons down due to the expansion of the gas, the movement is transferred to the crankshaft via the connecting rods and turned into rotation.



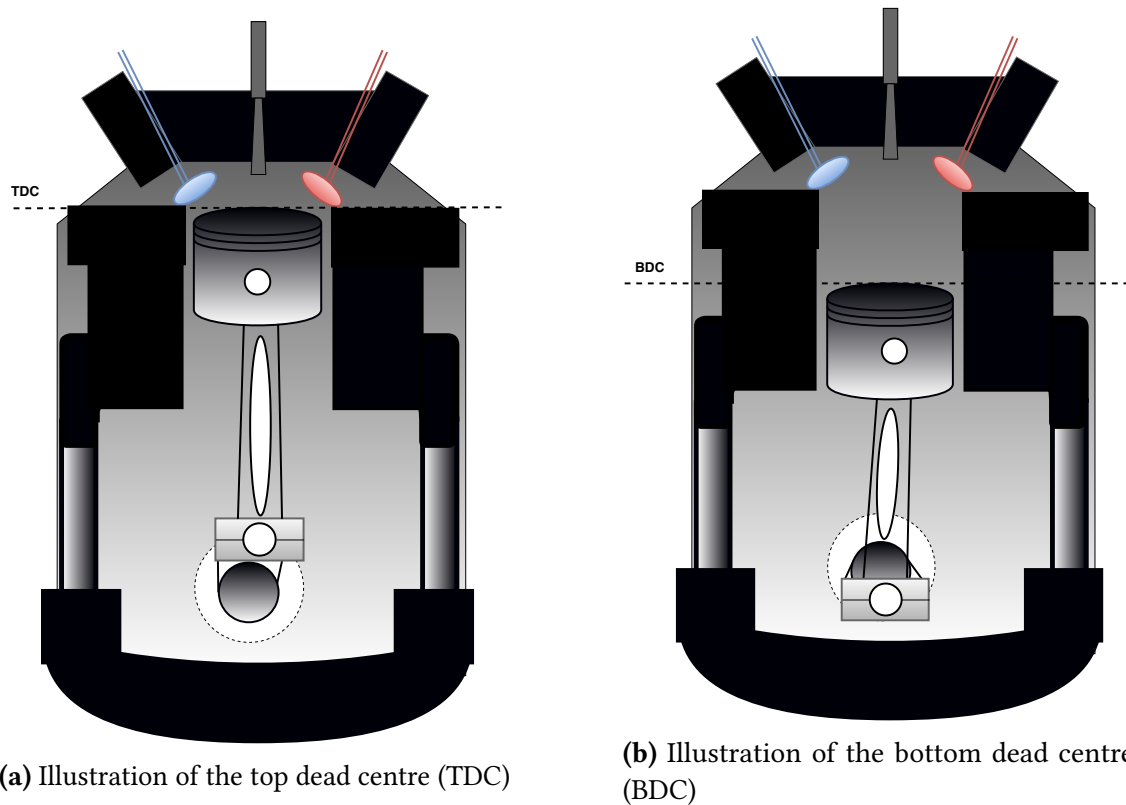
**Figure 2.14:** A simplified illustration of a diesel engine cylinder

The piston engine can work either on a two-stroke cycle or a four-stroke cycle, where the four-stroke cycle is the one most used today. The four-stroke cycle refers to the fact that a piston has to travel up and down the cylinder four times in order to complete one engine cycle. The four strokes are: (1) Intake stroke, (2) Compression stroke, (3) Expansion stroke and (4) Exhaust stroke [47]. Two strokes are done per rotation of the crankshaft, which means that a complete engine cycle requires two rotations (720 crank angle degrees (CAD)) of the crankshaft. Two other terms that are important towards the position of the pistons in the cylinders are *top dead centre* (TDC) and *bottom dead centre* (BDC). These are marked in Figures 2.15a and 2.15b .

The intake stroke starts with the piston in the TDC, and ends with the piston in BDC, creating a suction of air into the cylinder. Then the compression stroke follows, starting in BDC and ending with ignition shortly before TDC is reached again (depending on engine type, the start time of combustion varies). Next, the expansion stroke ensues, forcing the piston from TDC and back down to BDC. The cycle ends with the exhaust stroke, where the piston travels from BDC to TDC, "pushing" the exhaust gases through the now open exhaust valve [47].

The **camshaft** and the **valves** are essential for the engine cycle; each cylinder has valves solely for the purpose of intake of fresh air and release of exhaust gases. There is at least one intake valve and one exhaust valve per cylinder, and today it is common to have two of each. A typical layout for valve motion is through mechanical impact of a rocker arm, which is activated by the camshaft, although other solutions have been developed in recent years. The engine used in this project is fitted with a turbocharger. A turbocharger is a turbine-device which uses the exhaust gases to increase the amount of air fed to the combustion chambers in the cylinders, thus increasing the efficiency (and power) of the engine.

A typical measurement encountered in the world of engine related thermodynamics is the Air-to-fuel ratio. Not all too relevant for this project, a very short description is given as one of the measurements from the lab is based on this. The air-to-fuel ratio (AFR) describes the mass of air to the mass of fuel in a combustion process. The *stoichiometric* air-to-fuel ratio is the optimal air-to-fuel ratio; for this ratio all fuel is completely burned. As the stoichiometric AFR is the ideal situation, it is common to use this as a reference, and compare the actual AFR to this optimal situation. This gives the *air-to-fuel equivalence ratio*  $\lambda$ , which is the ratio of the actual AFR to the stoichiometric AFR. A value of  $\lambda = 1$  thus indicates that the actual AFR is equal to the stoichiometric AFR, whereas a  $\lambda > 1$  indicates that the mixture is *lean*, i.e. that there is excess air, and a  $\lambda < 1$  indicates that the mixture is *rich*, i.e. that there is excess fuel. In general, diesel engines always have excess air, meaning that  $\lambda > 1$  [48].



(a) Illustration of the top dead centre (TDC)

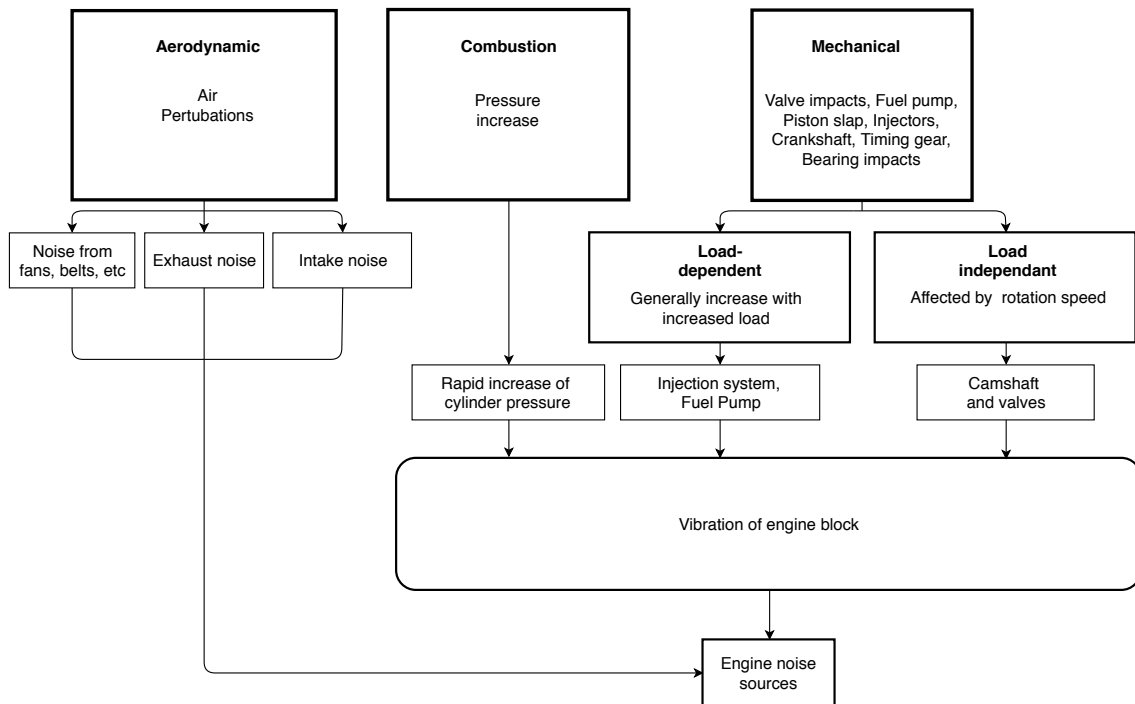
(b) Illustration of the bottom dead centre (BDC)

**Figure 2.15:** Illustrations of the TDC and BDC positions of the piston

### 2.4.2 Noise and Vibration Sources

The internal combustion engine is a complex composition of reciprocating and rotating parts, thus separating noise sources is a challenging task. However, the combustion process has been pointed out as the largest contributor to engine noise and vibration [8], [37], [49], this is especially true for diesel engines due to the higher cylinder pressure. The other noise sources are commonly summarised as "mechanical noise sources" [49], however Drouet *et al* [50] further divides the mechanical sources into "load-dependent" and "load-independent" groups, whereas others include a third group called "aerodynamic noise sources" for sources that directly produce air-borne noise [37], [51]. Figure 2.16 summarises the different noise sources, and it is based on the classifications given in [37], [49] and [50].

With this in mind, it is even more clear that using vibrations as an indicator of the condition of an engine makes perfectly good sense, even though it sometimes can be challenging to separate the different sources from each other. The combustion process is of course responsible for some of the mechanical impact noises, like the piston slap. Piston slap is the name given to the event in which the pistons hit the walls of the cylinder while moving up and down [8], and it is a large contributor to engine noise. Delvecchio *et al* [37] also give a great summary of typical faults and the transmission path of their



**Figure 2.16:** The different noise sources in internal combustion engines

vibrations, however as it is not possible to induce faults in the engine (other than maybe misfiring of cylinders) used for this project, it is less relevant but it can be used as a reference if such possibilities should appear.



# Chapter 3

## Materials and Methods

This chapter presents the experimental setup, information about the measurements done and the measurement plan and procedure. The experimental setup includes the diesel engine, the placement of the pickup microphones, the arrangement of the equipment in the lab and the data acquisition hardware. Furthermore, the implementation of the methods presented in chapter 2 are presented and linked to relevant code in Appendix C. Challenges encountered along the way and how they were solved is also given in this chapter.

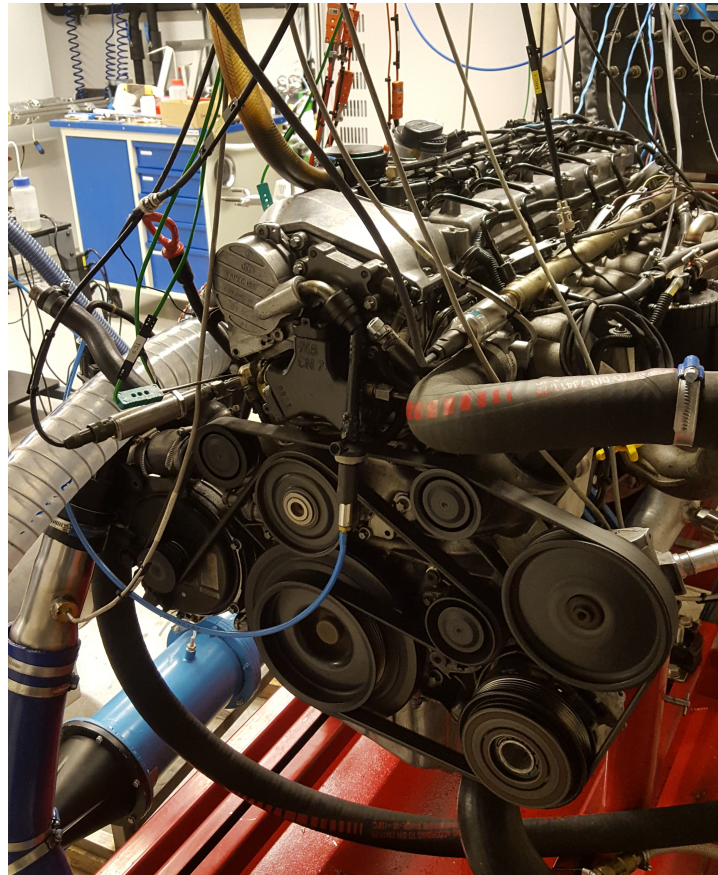
### 3.1 Experimental setup

As mentioned in the introductory chapter (ch.1), the Internal Combustion Engine Laboratory at the Department of Energy and Process Engineering at the Norwegian University of Science and Technology is used for data acquisition. Thus, the changes made to the test rig for this project were minimal. As the purpose of this laboratory is mainly directed towards energy efficiency and emissions research, the instrumentation is largely based on temperature sensors, pressure sensors and emission analysis equipment. The following gives an overview of the laboratory setup and the main parts of the instrumentation already found in the lab prior to this project. Then the changes made for this project are presented.

#### 3.1.1 The Diesel Engine

There are several test rigs at the Internal Combustion Engine Laboratory, however the one most relevant to this project is the diesel engine. The engine is a Mercedes-Benz six-cylinder inline diesel engine for passenger cars. The six cylinders are placed after each other in a line, and it is common to number the cylinders from the front of the

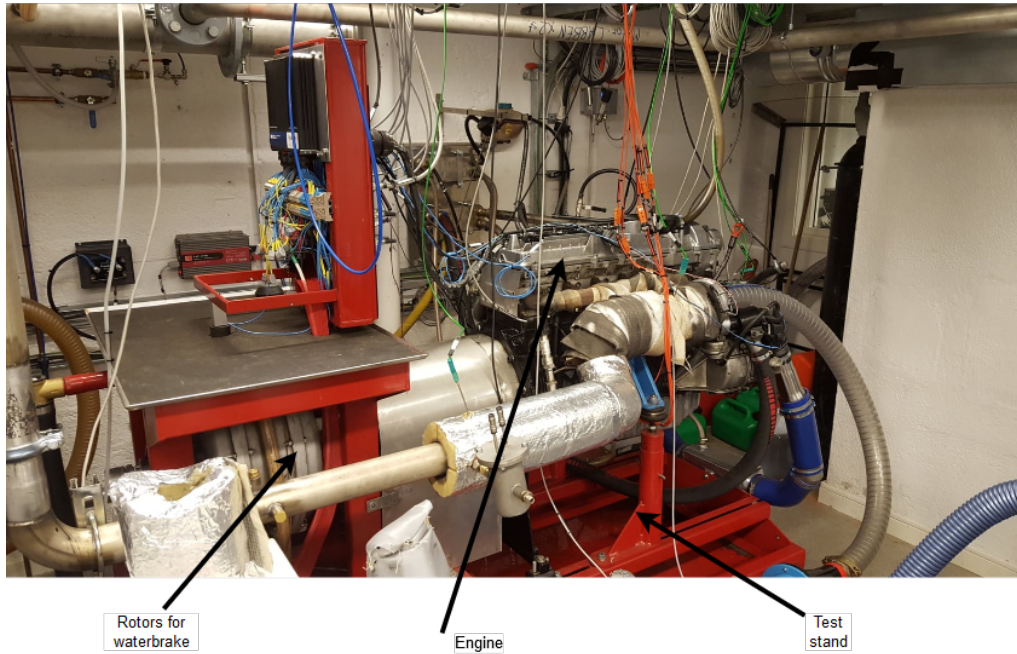
engine and towards the back, i.e the cylinder closest to the front of the car (when placed in a car) is cylinder #1 This is the numbering that will be used in this project. Figure 3.1 shows the engine, and from this cylinder-counting scheme, the cylinder that is closest in the picture is cylinder #1



**Figure 3.1:** The instrumented diesel engine in the lab

The engine resides on a Stuska test bench with integral hydrodynamometer; the dynamometer is connected to the output shaft of the engine, and acts as a break on the torque produced by the engine. Without going into details, the breaking effect is obtained by means of two rotors mounted on the shaft; a system of pumps and tanks supply water to the rotors, making rotation harder. Thus, increasing water flow to the rotors increases the load on the engine. Table 3.1 summarises the technical details of the engine, and Figure 3.2 shows the engine and test bench with dynamometer. Two more things with regards to the engine can be of interest; it has a variable geometry turbocharger and a common-rail injection system, i.e the vanes in the turbocharger are adjustable and the engine has a common high-pressure reservoir of fuel (instead of having one fuel pump for each cylinder.)





**Figure 3.2:** The engine and test bench with dynamometer

**Table 3.1:** Technical details for the engine [52]

<b>Engine type</b>	Turbocharged diesel engine for passenger cars
<b>Engine manufacturer</b>	Mercedes-Benz
<b>Engine code</b>	OM 613
<b>Number of cylinders</b>	6
<b>Orientation of cylinders</b>	Straight (inline)
<b>Total cylinder volume</b>	3,2L (3222 cc)
<b>Bore × Stroke</b>	88 × 88,3 mm
<b>Valve gear</b>	Double overhead camshaft (DOHC) 4 valves per cylinder
<b>Maximum power output</b>	194 break horsepower (bhp)(145kW)
<b>Maximum torque</b>	470Nm at 1800-2600rpm
<b>Compression ratio</b>	18:1
<b>Break mean effective pressure (BMEP)</b>	1883,1kPa
<b>Number of crankshaft bearings</b>	7

### 3.1.2 Measurements and Sensors

The measurements obtained at the lab are divided into "high speed measurements" and "low speed measurements", where the sampling frequency is the major difference. For some measurements it is more interesting to investigate trends, in this case for example over an engine cycle, whereas for other measurements it is more interesting to look at how values change from a very short time step to the next, for example during the combustion stroke in a cylinder. Thus, the low speed measurements are sampled at 10Hz, independent of the speed of the engine, and the high speed measurements are sampled at several kHz, and the sampling rate increases with the engine speed.

This is done in the following way; an encoder is attached to the output shaft of the engine. The encoder provides 2048 trigger-signals and one reference signal for each rotation of the shaft. Hence, an increase in the engine speed will lead to increased sampling rate. Each trigger signal leads to sampling of data through the associated sensors and data acquisition hardware. Thus, an engine speed of 1000rpm leads to a sampling rate of approximately 34,1kHz. By increasing the sampling rate while the engine speed increases, satisfying measurements can be obtained of the cylinder pressure during the combustion process, for all rotational speeds of the engine.

In-cylinder pressure is measured in cylinder #5. Measurements are obtained with a Kistler 6058 High Temperature Pressure Sensor, fitted inside the glow plug hole. The signal is fed to a Kistler Charge Amplifier Type 5073, before it is sent to the data acquisition unit, which is a National Instruments CompactRIO. A large number of temperature measurements are done; intake temperatures, exhaust gas temperatures, oil temperature, fuel temperature and coolant temperature to mention some. Most of the temperature measurements are done with type K thermocouples. Torque measurements are obtained by a force cell which is connected to the dynamometer with a torque arm.

Control of the engine is done with a Bosch Motorsport engine control unit (ECU) and Modus software. This makes it possible for the user to adjust engine parameters, like injection pressure, injection timing, turbo and engine speed. Thus, data about the injection timing, duration, volume, and fuel pressure and consumption is available through the ECU. When the load is adjusted, it is the ECU that ensures that the speed is kept constant, for example by increasing or decreasing the amount of fuel injected. Control of the load applied by the dynamometer and data logging is done through a program in LabView.

To measure the rate of air flow, an orifice plate with a nozzle is installed in the intake pipe (which is of a large diameter). Emission measurements are done by a Horiba Automotive Tests Systems Gas analyser, and all devices related to emissions originally mounted on the engine (from manufacturer, i.e catalysts etc. ) are removed. All meas-

measurements with their respective units and group affiliation (i.e high speed / low speed) are given in Table 3.2 and 3.3. Some of the measurements listed here might not be that relevant for this project, but the tables still provide a good overview of the content of the data sets, and thus the available data for closer investigation. Of the measurements that are given as boolean values, the reference signal from the encoder is the most important as it gives a signal (1) each time the crank shaft starts a new rotation, but is zero elsewhere.

**Table 3.2:** High Speed Measurements

<b>High Speed Measurements</b>	
<b>Measurement</b>	<b>Unit</b>
Time	Date
Cylinder Pressure	[MPaG]
Exhaust Pressure	[hPaG]
Fuel Pressure	[MPaG]
Engine Torque	[Nm]
Microphone 1	[Volt]
Microphone 2	[Volt]
Crank - encoder	Bool
CAM - ECU	Bool
Crank - ECU	Bool
Reference - Encoder	Bool
TDC - ECU	Bool

**Table 3.3:** Low Speed Measurements

Low Speed Measurements			
Measurements	Unit	Measurements	Unit
Time	Date	Fuel consumption ECU	[l/h]
EGT cylinder 1	[°C]	Air mass flow ECU	[kg/h]
EGT cylinder 2	[°C]	Air mass flow stroke ECU	[g/str]
EGT cylinder 3	[°C]	Duration of main injection ECU	[ $\mu$ s]
EGT cylinder 4	[°C]	Engine control mode ECU	[.]
EGT cylinder 5	[°C]	Lambda estimated ECU	[.]
EGT cylinder 6	[°C]	Fuel consumption inlet	[l/h]
Spill water	[°C]	Fuel consumption inlet total	[l]
Fuel temperature	[°C]	Fuel consumption return	[l/h]
Oil temperature	[°C]	Fuel consumption return total	[l]
EGT after turbine 1	[°C]	Engine torque - Nm	[Nm]
EGT after turbine 2	[°C]	Engine power - kW	[kW]
Intake temperature before charger	[°C]	Engine power - HP	[HP]
Intake temperature after cooler	[°C]	Mass flow rate	[kg/h]
Intake temperature before cooler	[°C]	Orifice plate pressure	[Pa]
Coolant temperature	[°C]	Engine speed	[rpm]
Water pressure	[hPaG]	Turbo speed	[rpm]
Oil pressure	[hPaG]	Cylinder peak pressure position	[deg]
Boost pressure	[hPaG]	Cylinder peak pressure	[MPaG]
Load pump	[%]	Room pressure	[hPaA]
Exhaust pressure	[hPaG]	Room temperature	[°C]
Battery voltage ECU	[Volt]	Room humidity	[[%]]
Fuel Pressure ECU	[MPaG]	NOx	[ppm]
Boost Pressure ECU	[hPaA]	CO	[ppm]
Engine speed ECU	[rpm]	CO2	[vol%]
Fuel quantity ECU	[mg/str]	O2	[vol%]
Fuel pressure valve ECU	[%]	Weight	[g]
Boost valve ECU	[\$%]	NO2	[ppm]
Begin of main injection ECU	[deg]	THC	[ppm]
Injection volume ECU	[ $mm^3/inj$ ]	NO	[ppm]

### 3.1.3 Preparations for Vibration Measurements

As the last section outlined the equipment already in the lab prior to this project, this section describes the changes that were made in order to obtain vibration measurements. In section 2.1.4 the piezoelectric pickup microphone was introduced. For this project, two cheap acoustic guitar microphones bought online was used for measuring vibrations. ISO 2954:2012 [25] gives some general requirements for the instrumentation used to measure the severity of vibrations. These requirements are given for the purpose of reducing inaccuracies in vibration measurements of a machine housing (casing).

From the requirements given in the standard it follows that the measurement instrument(s) should at least be able to measure RMS value of vibration velocity in the frequency range 10Hz-1000Hz; as no specifications are given of the pickup microphones from the supplier, a bold assertion is to assume that the frequency range of the pickup microphones are similar to that of the accelerometer, i.e 10Hz-10kHz [2]. The standard further provides requirements related to sensitivity, requirements to indicating unit, cables and power supply. As this project pursue a simplistic approach to vibration measurements, the requirements in this standard will be discussed no further. Thus, assumptions are made that the specifications of the piezoelectric pickup microphones are adequate. Figure 3.3 shows the microphones that are used for this project.



(a) Acoustic guitar pickup microphone in package as obtained from supplier



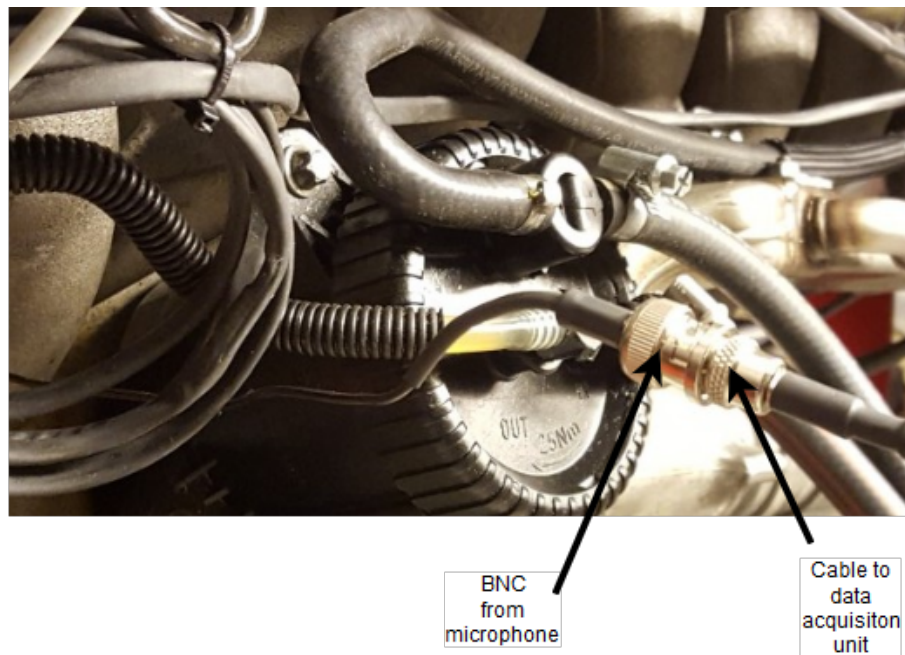
(b) The microphone is fitted with a 6,35mm female jack plug

**Figure 3.3:** The pickup microphone used for this project

Initial tests were performed outside the laboratory to determine the amplitude level of the output signal. As can be seen in Figure 3.3, the microphone output is a 6,35mm female stereo jack; appropriate connectors were used to connect this to the headphone jack port of a mobile phone. Recordings of short duration bursts or continuous knocking in the

vicinity of the microphones were carried out to investigate the amplitude of the output signal. After reviewing the recordings, it seemed as an amplifier was needed to get decent results. Transducers utilising the piezoelectric property tend to have a high impedance at the output [2], thus if connected directly to data acquisition unit some of the voltage signal might be lost due improper impedance matching. This pointed further towards the need for an amplifier, because an amplifier could turn the high output impedance of the pickup microphone into a low output impedance signal for the data acquisition unit (i.e an amplifier has better impedance matching with the piezoelectric pickup microphone). With this in mind it was decided that a review of the measurements at the lab was to be done before investing in an amplifier.

In general, the jack plugs and corresponding connectors are not well suited for laboratory measurements; data acquisition hardware seldom have such input channels, and they are also susceptible to noise (electromagnetic radiation, etc.) [2]. Thus, the cables on the pickup microphones were changed to coaxial cables with BNC plugs. These cables have a proven effect on minimising noise due to a braided wire shield around the central wire carrying the signal [2]. Furthermore, two 0-5V channels were added to the data acquisition hardware and the LabView program. These tasks were carried out by a laboratory technician working at the Department of Energy and Process Engineering. Figure 3.4 shows the connections and new cables on a microphone. With the signal level of measurements obtained at the lab being in the range of approximately  $\pm 1V$ , it was decided that the project would carry on without the procurement of an amplifier.

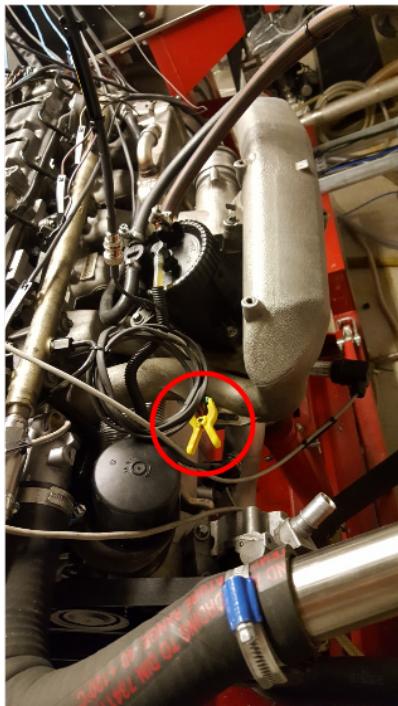


**Figure 3.4:** The new BNC plug and coaxial cable on the microphone



Finally, the placement of the microphones posed some challenges. ISO 13373-1:2002 [26] gives some general guidelines as well as typical transducer locations when measuring vibrations. As a general rule, a position near bearings can provide useful information with regards to vibration. However, based on the application one should consider to place the transducers at locations where vibration values are expected to be maximal. In an internal combustion engine, it is likely that the maximum vibration values will occur in the engine block near the cylinders. Thus, this was the desired transducer location initially.

Testing showed that the temperature on the engine block was too high for the chosen piezoelectric transducers, as they showed signs of melting (the microphone that showed signs of melting was taken out of service.) To overcome this obstacle, the microphones were moved to new locations where the temperature was sufficiently low. Microphone #1 was placed at the air-intake manifold, whereas microphone #2 was placed at the air supply pipe of the turbocharger. The placements are shown in Figure 3.5 and 3.6, respectively. Before the final tests were made, the signals obtained from these positions were investigated, to make sure that they captured an adequate amount of the engine vibration. Microphone #1 was attached with a clamp, whereas microphone #2 was attached with a glue pad which came along in the microphone package.

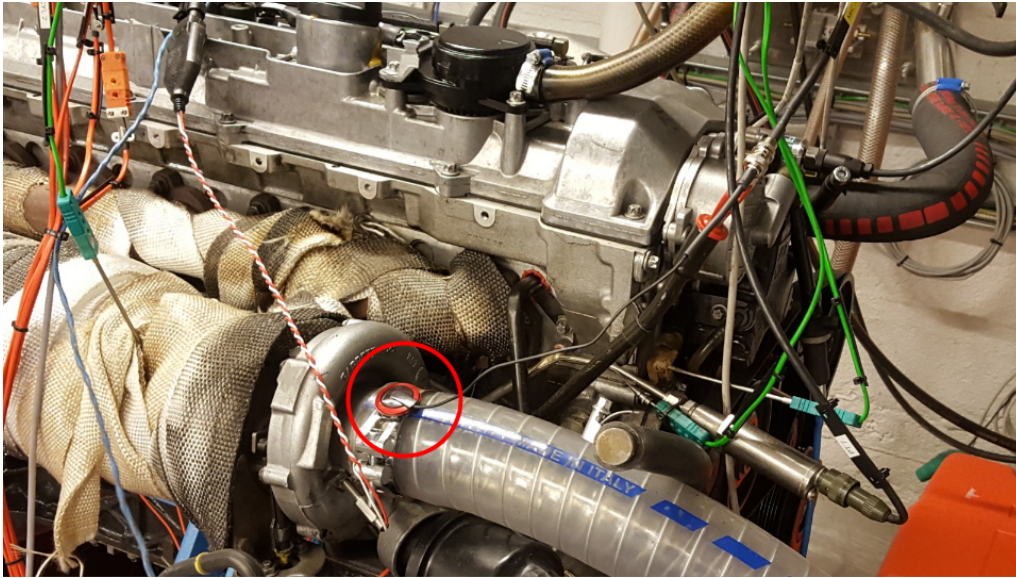


(a) Placement of microphone #1, on the air-inlet manifold.



(b) Placement of microphone #1. A clamp keeps the microphone in place.

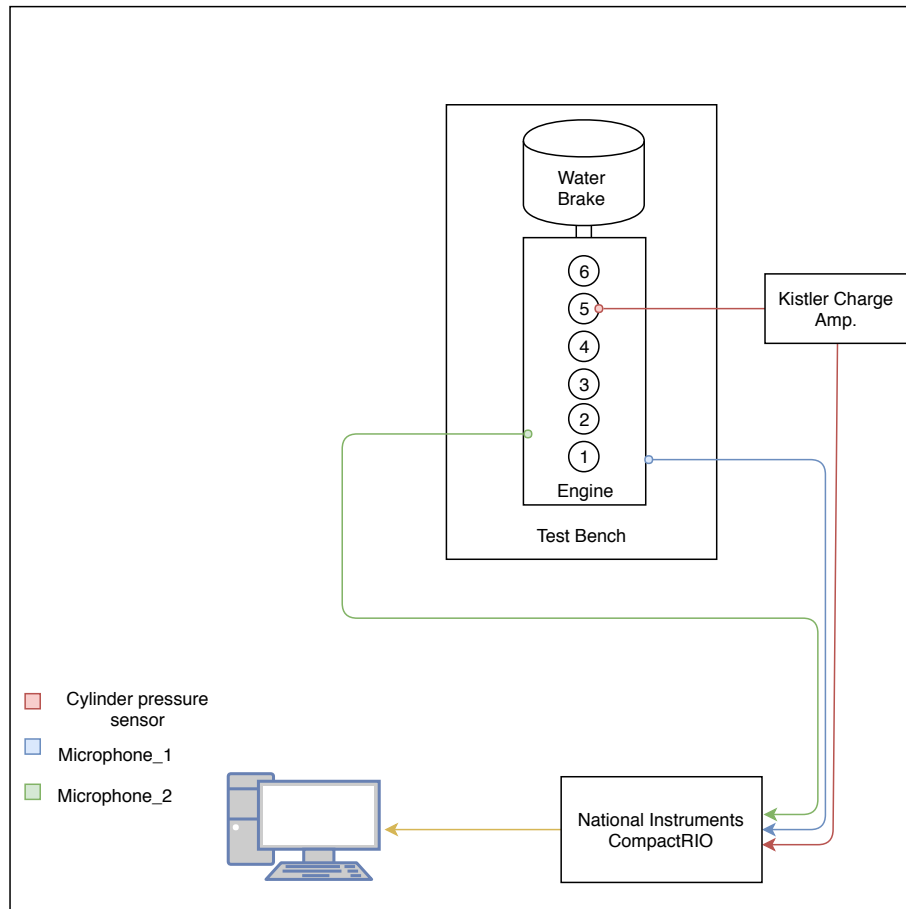
**Figure 3.5:** The placement of microphone #1



**Figure 3.6:** Placement of microphone #2, on the air supply to the turbocharger.

When looking at the data collected by microphone #2, it is important to keep in mind its placement, i.e it is glued to a hose clamp, which can vibrate differently than the rest of the engine. The reason for this particular placement was that it had to be placed at a location where the temperature was sufficiently low, and a location near the turbocharger was desired. To end this description of the equipment and setup at the lab, a schematic overview of the laboratory area is given in Figure 3.7, with the most important equipment outlined.





**Figure 3.7:** Schematic overview of microphone and pressure transducer locations

## 3.2 Measurement Procedure

After the preliminary tests, a measurement plan was made. The intention of the measurement plan was to gain a sufficient amount of data for different operating conditions. Three load conditions were chosen, and these loads were applied to the engine over a large range of rpms. Based on personal experience, the range of rpm was chosen from 1000rpm to 2600rpm, as this is a common area of operation for diesel engines in passenger cars. It also partly matches the rpm range where the engine can provide maximum torque, namely 1800-2600rpm (see Table 3.1). Thus, the range of rpm is a representative range for engine operation.

The chosen load conditions were 20Nm, 50Nm and 90Nm, and these loads were applied to nearly all rpms; however at the lower rpms the highest load conditions were skipped, due both to limitations of the dynamometer and partly the engine. As can be seen in Table 3.1, the maximum torque the engine can produce is 470Nm; thus the chosen load conditions are not as representative of the operating range of the engine, but still

gives the possibility of comparing equal load conditions between different rpms. Ideally, data logging should be done for a longer period of time, i.e the engine should be run for hours with data logging the entire time. But as the focus of the lab used for data acquisition in this project is quite different than that of the project, both data storage and equipment were not prepared for such endeavours. Thus, only approximately 30s of recording was done for each of the operating conditions. However, due to the high sampling rate this still provides more than 800 000 samples for the test with the lowest rpm. The measurement plan with the test conditions are summarised in Table 3.4.

**Table 3.4:** Measurement plan with test conditions

<b>Test</b>	<b>RPM</b>	<b>Load</b>	<b>Duration</b>	<b>Test</b>	<b>RPM</b>	<b>Load</b>	<b>Duration</b>
<b>Test1</b>	1000	20Nm	30sec	<b>Test13</b>	2000	20Nm	30sec
<b>Test2</b>	1200	20Nm	30sec	<b>Test14</b>	2000	50Nm	30sec
<b>Test3</b>	1200	50Nm	30sec	<b>Test15</b>	2000	90Nm	30sec
<b>Test4</b>	1400	20Nm	30sec	<b>Test16</b>	2200	20Nm	30sec
<b>Test5</b>	1400	50Nm	30sec	<b>Test17</b>	2200	50Nm	30sec
<b>Test6</b>	1400	90Nm	30sec	<b>Test18</b>	2200	90Nm	30sec
<b>Test7</b>	1600	20Nm	30sec	<b>Test19</b>	2400	20Nm	30sec
<b>Test8</b>	1600	50Nm	30sec	<b>Test20</b>	2400	50Nm	30sec
<b>Test9</b>	1600	90Nm	30sec	<b>Test21</b>	2400	90Nm	30sec
<b>Test10</b>	1800	20Nm	30sec	<b>Test22</b>	2600	20Nm	30sec
<b>Test11</b>	1800	50Nm	30sec	<b>Test23</b>	2600	50Nm	30sec
<b>Test12</b>	1800	90Nm	30sec	<b>Test24</b>	2600	90Nm	30sec

For each of the speed and load conditions stated in Table 3.4, all the measurements of Tables 3.2 and 3.3 were logged. Before the start of data acquisition, the engine was run for 15minutes to make sure that the temperature of components and fluids were warm and stable. For each of the operating conditions stated as "Tests" in Table 3.4, the engine was run for approximately 1minute before start of data logging, to make sure that the engine had stabilised at the new condition. This could be visually confirmed through the graphs and values provided by the user interface in the LabView program. The data acquisition was carried out according to plan, and without any large deviations. However, there was one test condition that seemed "unstable"; during Test18 with 2200rpm and 90Nm the engine stalled. It started again immediately, nonetheless this test was marked and kept in mind for the work to come.

### 3.3 Data Processing and Analysis

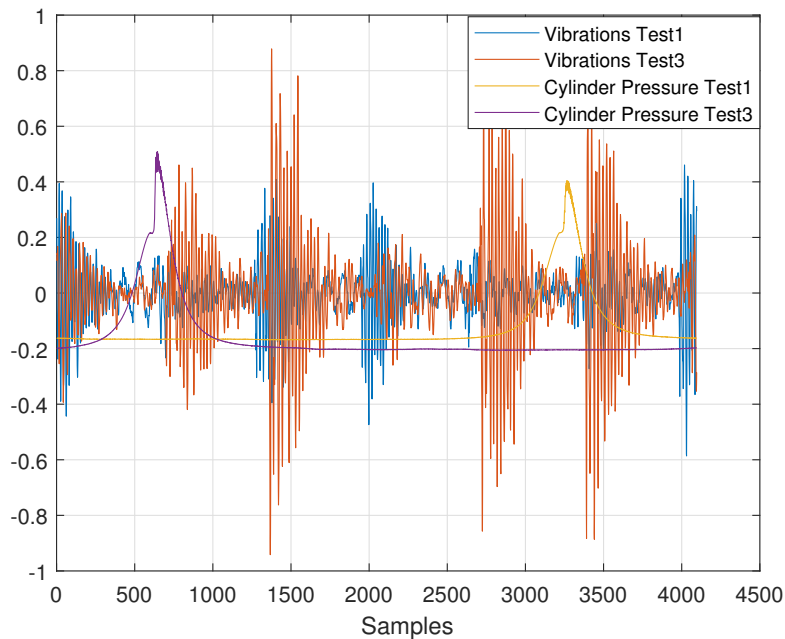
The data from the laboratory measurements was stored in the TDMS file format, which is a common format when using software and hardware from National Instruments. It was desired to do the data processing in MATLAB; thus, a TDMS reader program was downloaded from MathWorks File Exchanger [53]. By using the functions provided by the TDMS reader, the data sets could be read into MATLAB and stored as structs. It is also possible to open these files in excel, and read the data into MATLAB using the `xlsread`-function, however Excel has a maximum limit for the number of rows, and some of the data sets in this project are larger than this limit. In MATLAB, several processing steps were made before the data set was loaded into The Unscrambler®X for PCA and regression. The following sections shortly outline the processing that was done in MATLAB. Two scripts have been written for the purpose of the following procedures; the first script deals with data extraction, and the second deals with the processing of the data. Both of these scripts can be found in Appendix C.

#### 3.3.1 Data Extraction

After having loaded the data into MATLAB as structs, all the data of all the tests were gathered in two multidimensional arrays; one with all low speed measurements and one with all high speed measurements. The multidimensional arrays can be thought of as three-dimensional matrices, where the first and second dimension are the rows and columns, and the third dimension is like the pages in a book [54]; in this case each page correspond to the measurements of one test. This gave an easy and tidy collection of the data sets, where values could be indexed out from a certain measurement and/or a certain test. The process of extracting all the data and gathering it in the multidimensional arrays can be seen in the MATLAB code in Appendix C.1.

#### 3.3.2 Defining an Engine Cycle

From the theory presented about the four-stroke cycle in section 2.4 and the shaft encoder presented in section 3.1.2, i.e that a complete engine cycle requires two rotations of the crankshaft, and that the shaft encoder provides 2048 trigger-signals per rotation of the crankshaft, it is clear that the length of an engine cycle always will be 4096samples. Thus, engine speed and sampling rate will not affect the number of samples per cycle. Therefore it should be an easy task to extract an equal number of engine cycles from the tests. However, as there is no way to easily determine the position of the crank shaft when commencing data logging, it is not necessarily so that each recording starts at the same place in an engine cycle. This is illustrated in Figure 3.8, where the first 4096samples



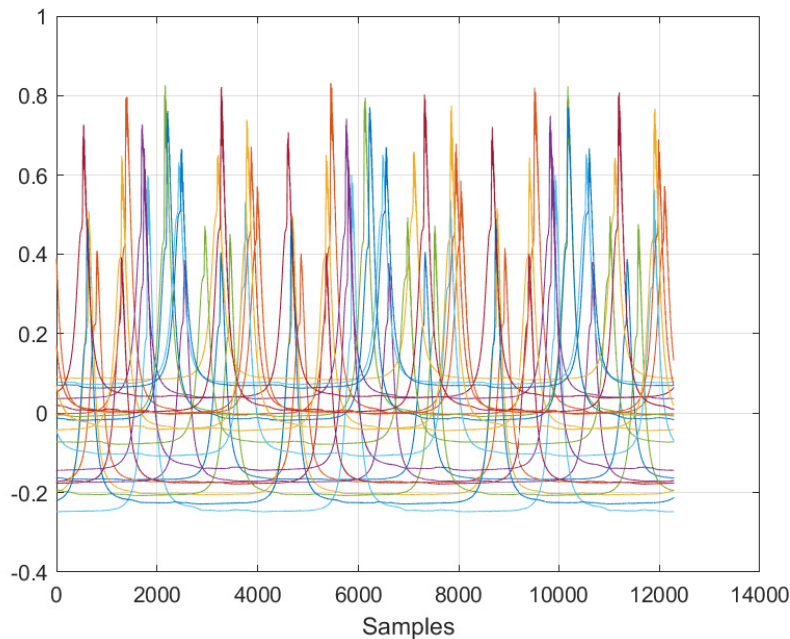
**Figure 3.8:** Vibrations from microphone #1 and cylinder pressures for Test1 (1000rpm20Nm) and Test3 (1200rpm50Nm). Values for cylinder pressures have been down scaled by a factor of 10 for easier illustration, and units along the y-axis have been omitted on purpose.

of the vibration measurements for two different test conditions are plotted along with their respective cylinder pressures. It is reasonable to expect that the cylinder pressure is shifted a few samples from one operating condition to the next, however as illustrated here the difference is more than 2500samples. To obtain comparable data for all of the test conditions, it was necessary to define the start of an engine cycle, and implement this definition for all tests.

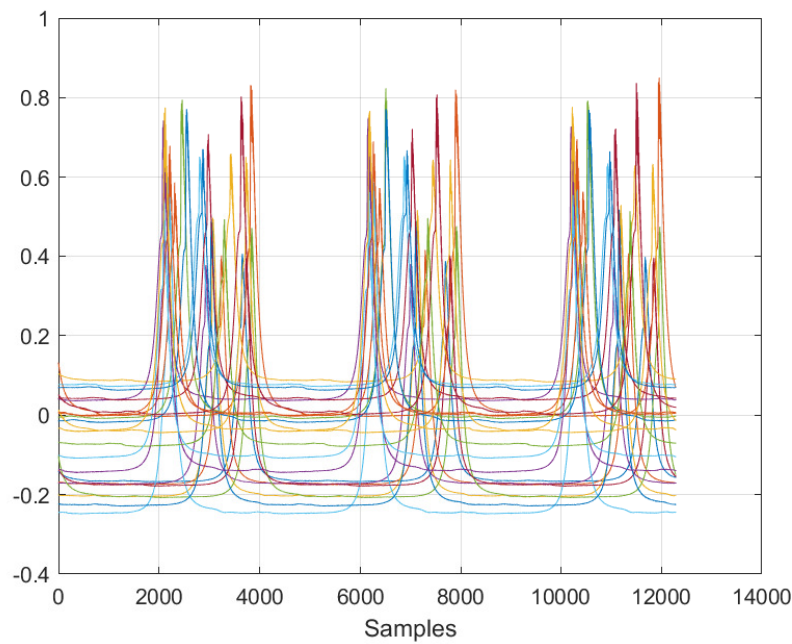
Two methods were proposed to fix this problem; the first one is based on a combination of the reference signal from the encoder and the cylinder pressure measurements, and the second one is only based on the cylinder pressure peaks. The first method builds on the following: the shaft encoder provides one reference signal per rotation (i.e per 2048 samples). Hence, an engine cycle can be defined as the interval between subsequent reference signals; starting the cycle at one reference cycle, passing a reference signal at 2048samples (after one rotation) and ending the cycle at the next reference signal (after two rotations). So, the first thing that is done is to find the index of the first reference signal; this will of course vary between the different tests. Then, the index of the first cylinder pressure peak is found. Both of these indices are found using the `max`-function in MATLAB. Then, it follows that if the index of the reference signal is lower than that of the pressure peak, the new index of the reference signals is updated to be the first

reference signal that comes after the index of the pressure peak. The idea is that the cycles can be matched by using the reference signals from the encoder; starting all cycles at the first reference signal that occurs after a pressure peak.

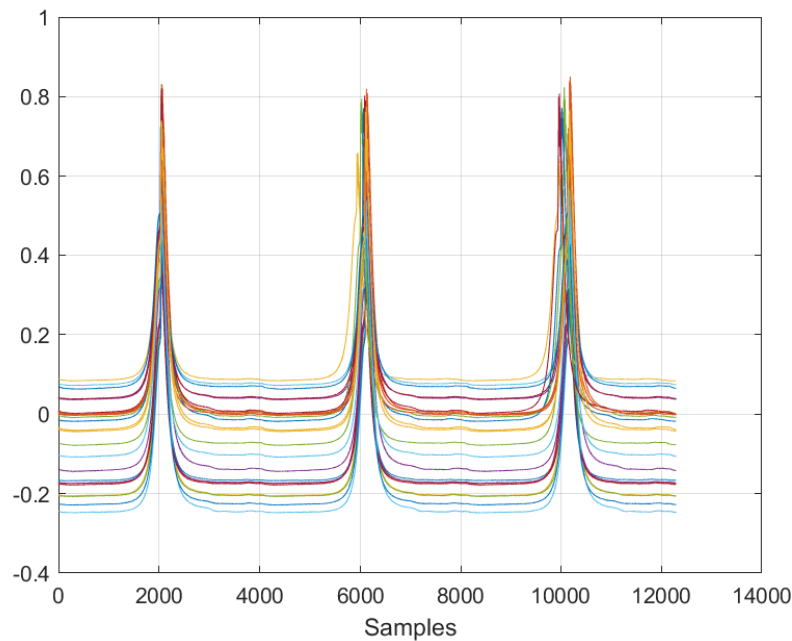
The second method is based on a similar approach; the index of the first pressure peak that is at least 2048samples into the recording but not further in than 2048samples plus one cycle (i.e 4096samples) is found using the `max`-function in MATLAB. This index is then used as the start of the cycles, so that each cycle starts 2048samples before a pressure peak, and ends 2048samples after a pressure peak. The code for this method can be seen in Appendix C.2, lines (30-65). Figure 3.9 shows the cylinder pressure peaks of the three first cycles of all tests before implementing the methods described above, and Figure 3.10 shows the cycles after having implemented each of them. Units on Y-axes have been omitted on purpose.



**Figure 3.9:** The three first cycles (or, 12288 first samples) of the cylinder pressure data for all tests, down scaled by a factor of 10.



(a) The cylinder pressure peaks of the three first cycles after implementing the first method for defining an engine cycle



(b) The cylinder pressure peaks of the three first cycles after implementing the second method for defining an engine cycle

**Figure 3.10:** Illustrations of the effect of matching engine cycles

From the figure, it seems as though the second method is the most successful one with regards to matching the cycles; however some fluctuations are seen in later engine cycles. After having decided to move on with the second method, the multidimensional

arrays were reset to start at the new indices. This was important to make sure that all the other measurements also follow the same cycles. Due to limitations in the length of the recording with the lowest rpm, a number of 200 engine cycles were extracted for comparison ( $200 \cdot 4096 = 819200$  samples) and further use in the project. Although not ideal, it should give sufficient information for this study.

### 3.3.3 Fourier transform

After having extracted the data sets and matched their starting indices, the next step was to transform the vibration measurements into the frequency domain. This was done using a windowed Fourier transform, i.e. the vibration signals were divided into frames, and an FFT was applied to each of the frames. Since the engine cycles have a length of 4096 samples, this was chosen as the frame lengths over which the FFT was applied. In addition, an overlap of 2048 samples (50%) were used together with a Hamming window in MATLAB. The results from the Fourier transform of all tests were gathered in one large matrix, with the frames acting as rows, the frequency coefficients acting as columns, and the values within the matrix as intensity, or magnitude of the frequencies.

MATLAB has a built-in function of the Short-Time Fourier Transform, however in this project a `for`-loop with MATLAB's `fft`-function was implemented instead. The MATLAB code for this procedure can be seen in Appendix C.2, lines (128-215). This procedure was carried out on the vibration signals from both microphone #1 and #2. A large frequency spectrum was created, which could be put in the same table as all the other measurements. Along the way, the effect of using overlapping frames in this specific application was reconsidered.

### 3.3.4 Adapting Measurements to Engine Cycle

As the frequency data now was based on frames matching the sample-length of the engine cycles, something had to be done to the other measurement data for them to correspond to the number of frames. For the high speed measurements, an equal number of samples existed as for the vibration data, i.e. 4096 samples per engine cycle. The solution that was implemented was to take the mean of the samples across each frame (=cycle). One should be careful with the use of mean values, i.e. information can be lost when taking the mean over a number of measurements. However, as the purpose was to compare between engine cycles, and not the values within engine cycles, the mean was chosen as a simple solution to the problem at hand. To make the high speed measurements fit the number of frames for the frequency vibration data, overlap was also implemented. That is, the mean of the measurements was taken with 50% overlap with the previous engine

cycle. The MATLAB code for this procedure can be seen in Appendix C.2, lines (218-270).

For the low speed measurements, a different approach had to be followed. As the sampling rate of the low speed measurements is constant at 10Hz, independent of engine speed, something had to be done to make sure that the measurement values corresponded to their respective frames (or cycles). First, an array where the rows correspond to the total number of frames and the columns correspond to the number of tests, was created. Then, the end-times of each frame of each test was calculated using the associated (high speed) sampling rates. The next step was then to match the frame times with the time of a low speed measurement; this was done using the `min`-function in MATLAB. The index of the value of the low speed measurement times that gave the minimum result when subtracting the frame-times was used as the measurement for that frame. The MATLAB code for this procedure can be found in Appendix C.2, lines (272-340). Finally, the high speed and low speed measurements for the frames of all tests were gathered in a common matrix.

### 3.3.5 Resampling

The last major processing step that was performed was resampling of the results of the Fourier transform. As the sampling rate was different for the vibration data for all tests, it was necessary to resample the data to a reference. The resampling was carried out using MATLABs `resample`-function. When resampling the vibration data, a reference rpm of 3000 was chosen; for each test the measurements were resampled at a number given by the original rpm divided by the reference rpm, times the original sample rate. Before resampling it was difficult to interpret the results with regards to frequencies, as all tests had different sampling rates. After resampling, it was possible to see how the intensity of a certain frequency changed with load and rpm. The MATLAB code for this procedure can be seen in Appendix C.2, lines (479-512). When looking at the results of the windowed Fourier transform, the word "frequency coefficients" has been used. An explanation of how to interpret this is given in section 2.2. After the resampling procedure, the same method can be used to transform the frequency coefficients into frequencies, but the sampling rate is now  $3000 * 2048 / 60 = 102400$  Hz. Thus, to transform a frequency coefficient to a frequency in a spectrogram obtained after resampling, the frequency coefficient has to be multiplied by the Nyquist frequency divided by the number of frequency coefficients. In this project a window length of 4096samples has been used, and this leads to 2048 frequency coefficients. Therefore, after resampling a frequency coefficient has to be multiplied by 25 to get the frequency in Hz.

After resampling, a common table was created for the data for all the measurements, including the resampled frequency data. The corresponding rownames (test conditions

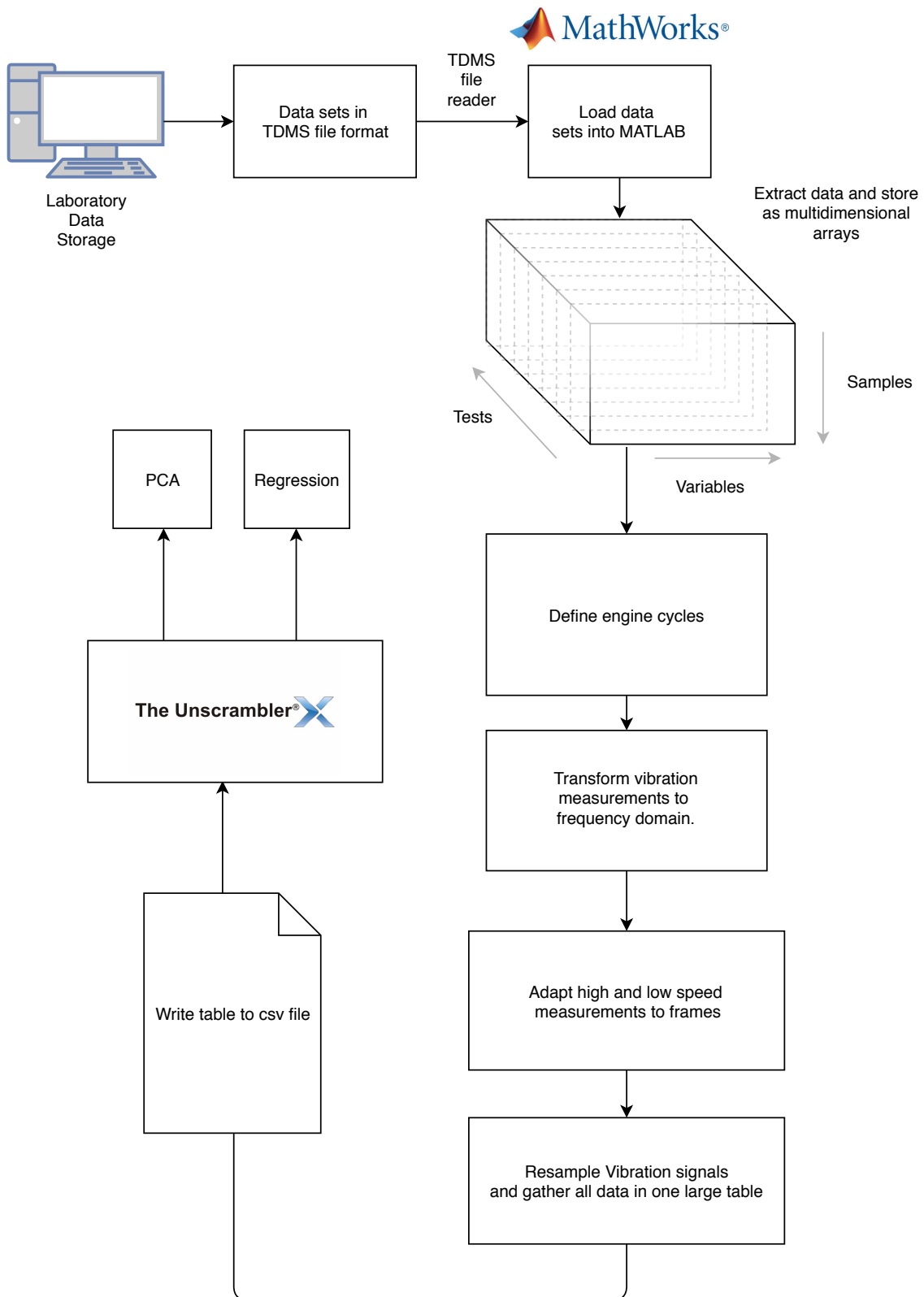


and frame nr.) and column names (variable names) were added, and the table was written to a csv file. Before the csv-file could be imported into The Unscrambler®X, the commas that were used as separators were replaced with semicolons, whereas the dots used for decimals were replaced with commas, in order for the program to interpret all values as numeric.

### 3.3.6 The Unscrambler®X

The final processing of the data set was done in The Unscrambler®X. Before each PCA and regression analysis, the data was mean centred. In addition, analyses were done with and without weighting. The weighting that was used was the inverse of the standard deviation. It works as follows: the standard deviation of each column (variable) of the X matrix is calculated. Then, each element in X is multiplied with the inverse of the standard deviation of the variable it belongs to. This procedure ensures that the variance of each variable is the same [11], and is thus particularly useful for data sets where there are variables with different units, like in this project. As an example, take the turbo speed, which may vary from 20000 to 150000rpm and the NO<sub>x</sub> measurements which are in the order of 1,11-1,15ppm. The weighting function is a built-in feature of The Unscrambler®X.

For the PCA analyses, the maximum number of components was limited to 7. Furthermore, the NIPALS algorithm was used, with the number of iterations being limited to 100. For the validation part, random segmented cross validation was employed. The number of cross validation segments were set to be 20 for the PCA analyses. For the regression analyses, the maximum number of factors was yet again limited to 7, as this was the standard input of the program. Moreover, random cross validation with 8 segments was used for the validation procedure. Due to the size of the data set (9576x4166) the number of cross validation segments had to be reduced when making several models. NIPALS algorithm with the number of iterations limited to 100 was used for the regression models as well. The functions in The Unscrambler®X are quite straightforward, and a comprehensive user manual is available online. The overall data processing procedure is summarised in Figure 3.11.



**Figure 3.11:** The workflow of the data processing procedure

# Chapter 4

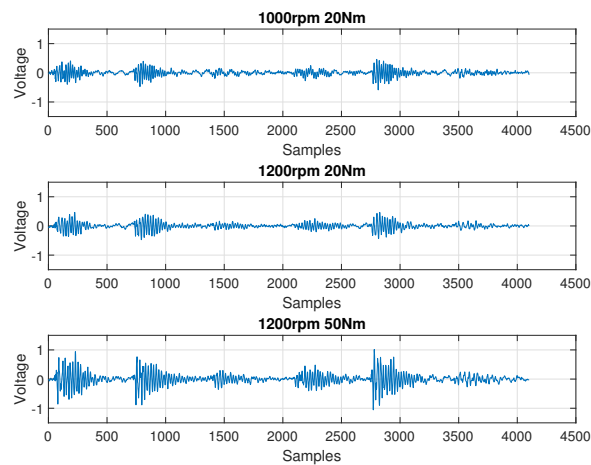
## Results and Discussion

This chapter presents the results obtained in this study and discusses the findings, their limitations and relevance. The chapter is divided into three sections, each section reflecting the results obtained in relation to one of the objectives stated under "Experimental study" in chapter 1.3. The sections are further divided into subsections to make it easy to follow the methods that have been used to obtain the given results. Discussion of the results are given for each section throughout the chapter. Results are mainly presented as graphs with explanatory text, however tables will be given in addition for some of the results; this mainly concerns the results for the regression section.

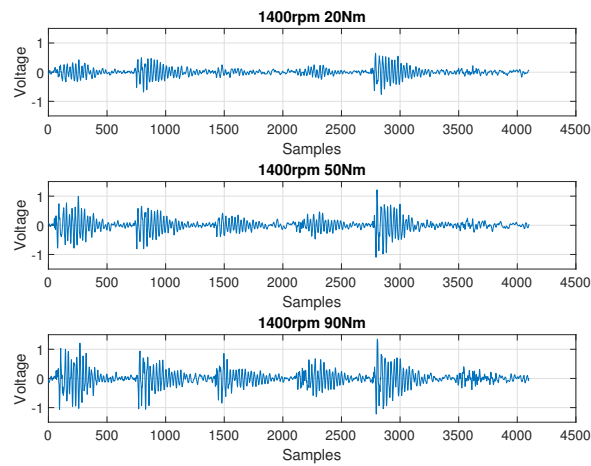
### 4.1 The Effects of Speed and Load on Vibrations

#### 4.1.1 Time Domain

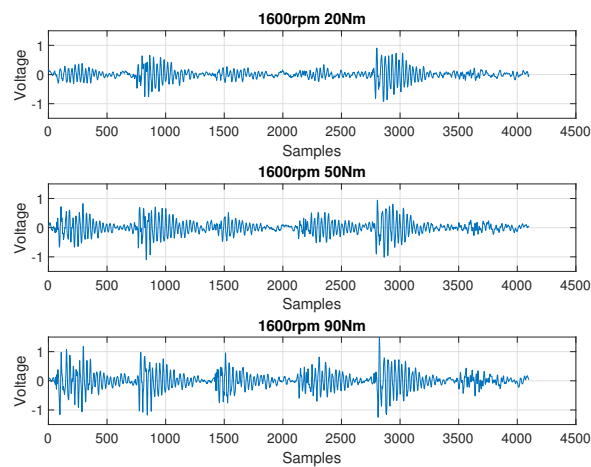
Figures 4.1-4.3 shows the first extracted engine cycle of all rpms, for all load conditions. The peaks correspond to the combustion events, of which there should be six for each engine cycle due to the six cylinders of the engine. It can be seen that these combustion peaks are more noticeable for the tests with lower rpm; the overall vibration amplitude increases with increased rpm. Furthermore, the combustion events seems to be stretched over a larger number of samples for the tests with the higher rpms, than for the tests with the lower rpms. However, load condition seem to be a more important factor for the vibration level than rpm. This can be seen from within each plot, where the vibration level increases with increasing load. Moreover, the vibration amplitude for the highest load condition for a given rpm is larger than for the lowest load condition of the next rpm. This can be seen from comparison of the 90Nm load condition in Figure 4.1c and the 20Nm load condition in Figure 4.2a.



(a) The vibration of the first extracted engine cycle for the different load conditions at 1000rpm and 1200rpm

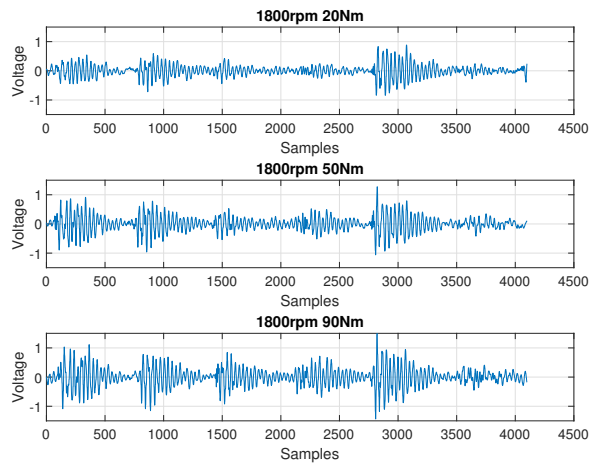


(b) The vibration of the first extracted engine cycle for the different load conditions at 1400rpm

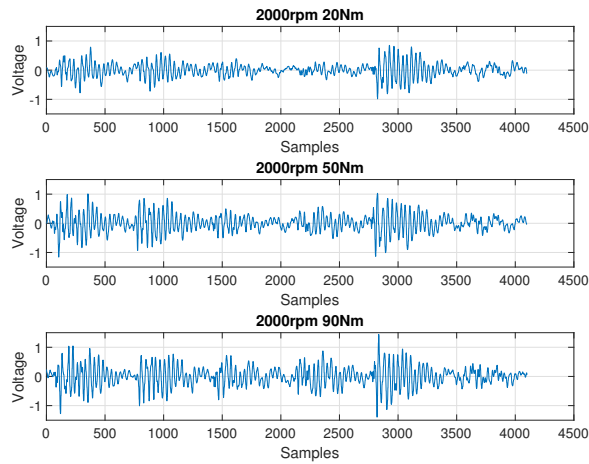


(c) The vibration of the first extracted engine cycle for the different load conditions at 1600rpm

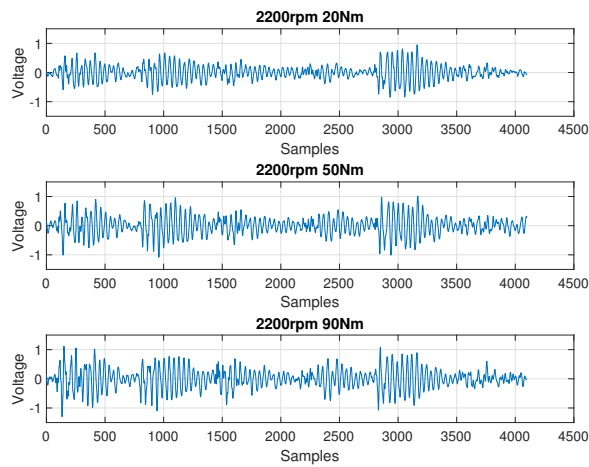
**Figure 4.1:** Time domain vibration signal for three different rpms



(a) The vibration of the first extracted engine cycle for the different load conditions at 1800rpm

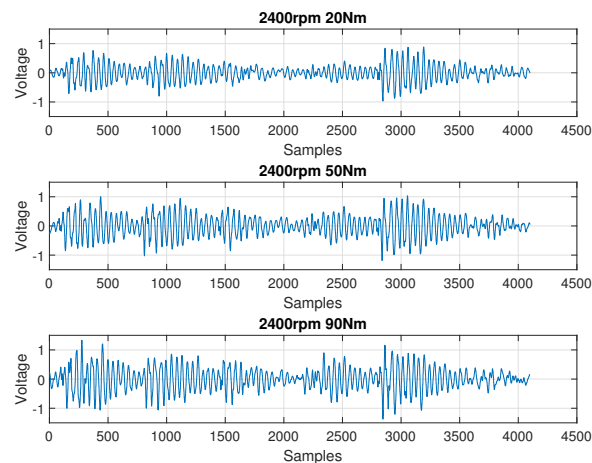


(b) The vibration of the first extracted engine cycle for the different load conditions at 2000rpm

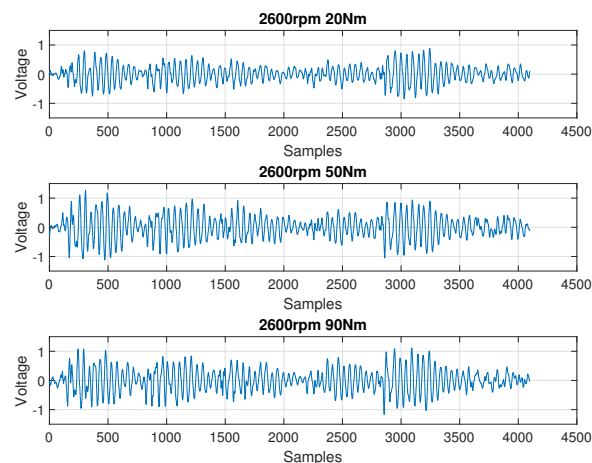


(c) The vibration of the first extracted engine cycle for the different load conditions at 2200rpm

**Figure 4.2:** Time domain vibration signal for three different rpms



(a) The vibration of the first extracted engine cycle for the different load conditions at 2400rpm

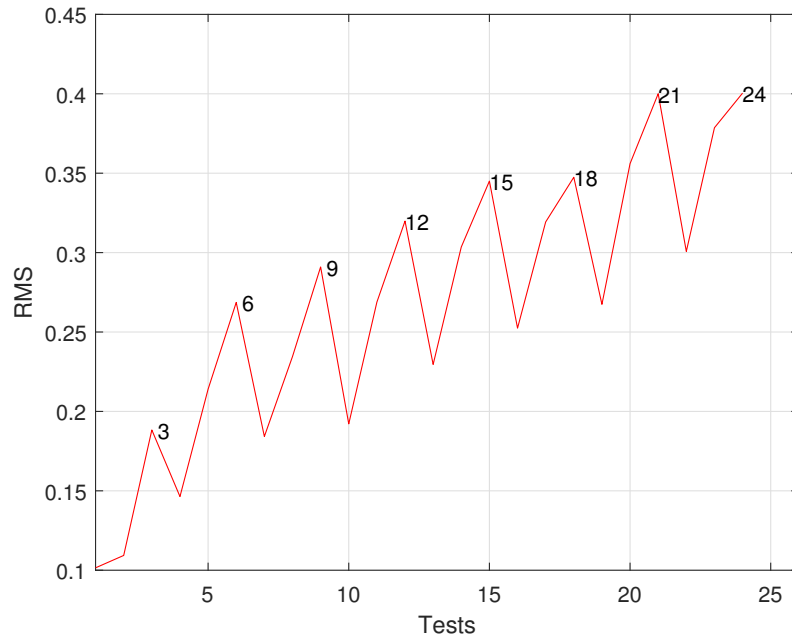


(b) The vibration of the first extracted engine cycle for the different load conditions at 2600rpm

**Figure 4.3:** Time domain vibration signal for three different rpms

The plots have been obtained from the vibrations measured by microphone #1, however the same trend is apparent in the vibrations from microphone #2, although the overall vibration levels in general are lower for the vibrations obtained from this microphone. These results thus point towards that vibration is affected by both speed and load. An interesting plot which further backs the given statement is the plot of the root mean square of the vibrations. The RMS was introduced in section 2.1.5, and the formula is given in Table 2.1. Figure 4.4 shows the root mean square value of the vibration measurements obtained from microphone #1 for all tests. The peaks are marked with numbers, and the numbers correspond to which test the RMS value applies to. It can be verified from Table 3.4 that the peaks in RMS value correspond to the tests with the highest load condition

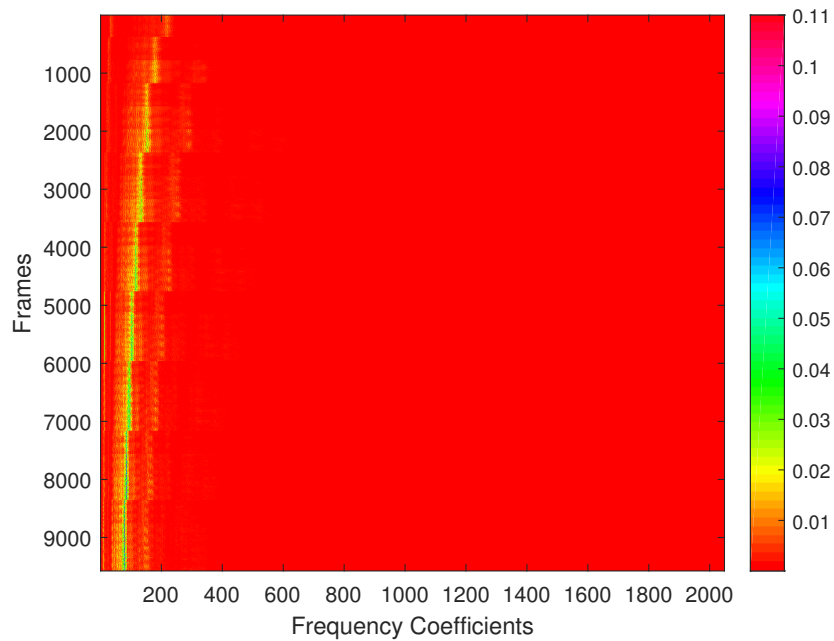
for all rpms, except from Test1 which only had a 20Nm load condition. As RMS value is related to vibration energy, this further points towards a relation between vibrations and load and vibrations and speed.



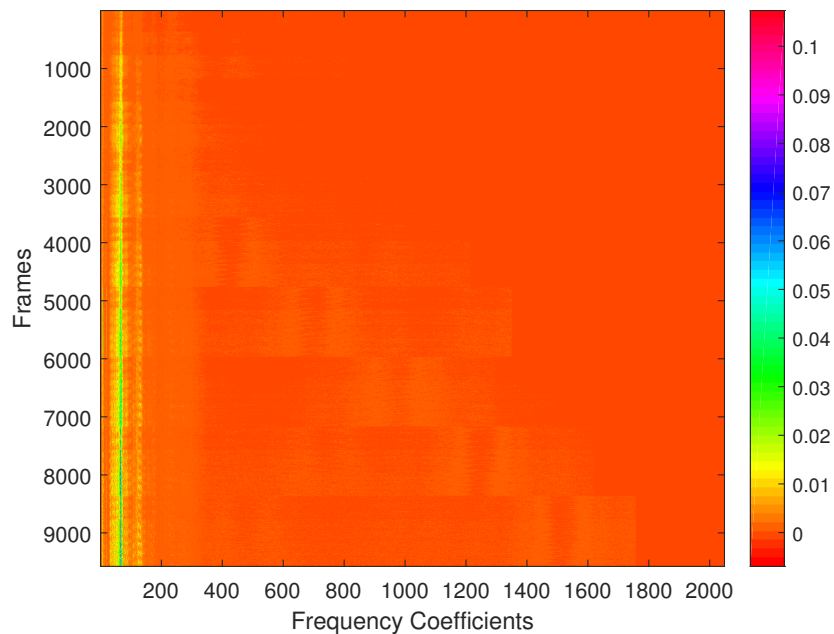
**Figure 4.4:** The RMS value of the vibration measurements from microphone #1 of all tests.

#### 4.1.2 Time-Frequency Domain

To further investigate this relationship, the vibration measurements were analysed in the time-frequency domain. Figure 4.5 shows a spectrogram of the vibration measurements from microphone #1 prior to resampling. In the figure, the frames for all the vibration measurements have been placed beneath each other; due to the fact that sampling rate is dependent on the engine speed (trigger signals from shaft encoder), different sample rates have been used for the different tests, and therefore it might be hard to interpret some of the contents of the figure. There is one interval with a higher intensity than the rest, which shifts to the left for increasing rpm (which is the same as increasing sample rate). Figure 4.6 shows the same data after all measurements have been resampled to a reference rpm.



**Figure 4.5:** Spectrogram of vibration measurements (microphone #1) for all tests prior to resampling

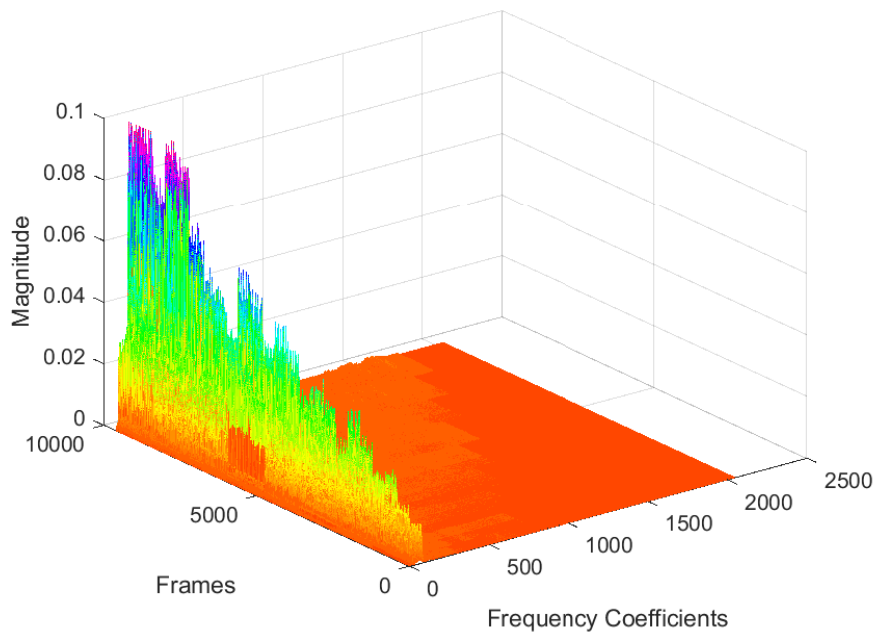


**Figure 4.6:** Spectrogram of vibration measurements (microphone #1) after resampling

From Figure 4.6 it can be seen that the interval of high intensity that was shifted towards the left now maintains a straight line; furthermore, it seems to increase in intensity for increasing frame number. The frame numbers correspond to the different

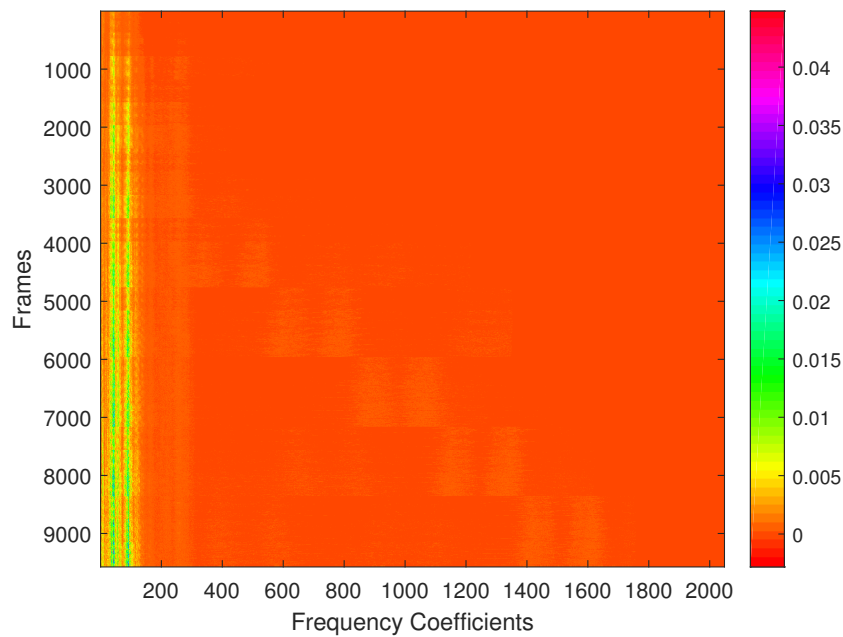


tests (i.e frame 1-399 is Test1, frame 400-799 is Test2 and so on). An increase in the intensity for increasing rpm means that the magnitudes of the frequencies are larger at the higher rpms; in other words, the vibrations are stronger. A 3D-plot showing the same data can aid in the interpretation. Figure 4.7 shows the same data as Figure 4.6, now as a three-dimensional figure.

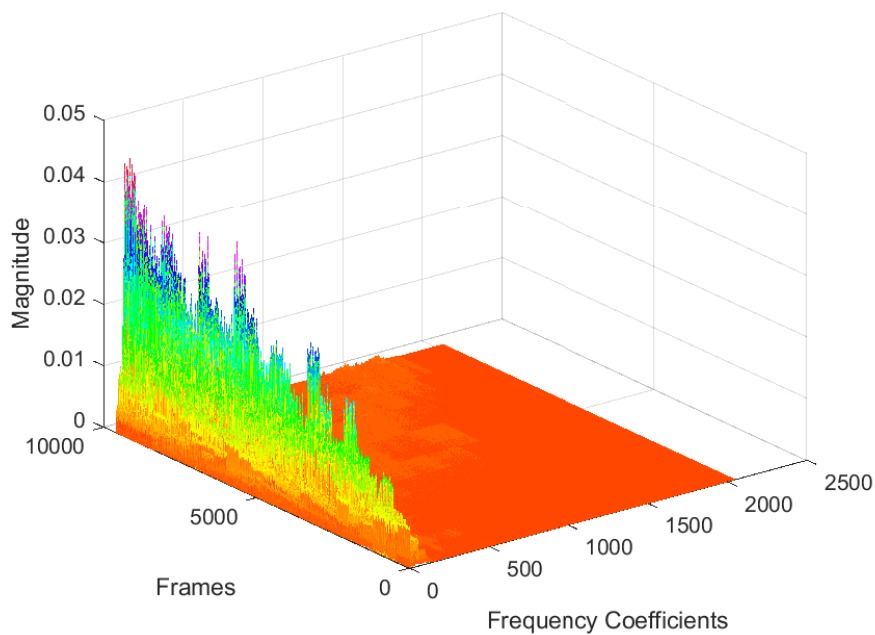


**Figure 4.7:** A 3D plot of the time-frequency data of the vibration measurements from microphone #1

It is easy to see from Figure 4.7 that the magnitude of the frequencies increase for increasing rpm; furthermore, the stair-like structure corresponds well with the RMS values in Figure 4.4. The load conditions appear as steps in a staircase; and as discussed in section 4.1.1, the vibration level is higher for the largest load for the preceding rpm than it is for the smallest load for the next higher rpm. Figure 4.8 and 4.9 shows the the same plots for the vibration measurements from microphone #2 after resampling. The same trends that appear in the measurements from microphone #1 are also apparent here; increasing vibration magnitude as load and rpm increase, and the stair-like structure.



**Figure 4.8:** Spectrogram of resampled vibration data from microphone #2



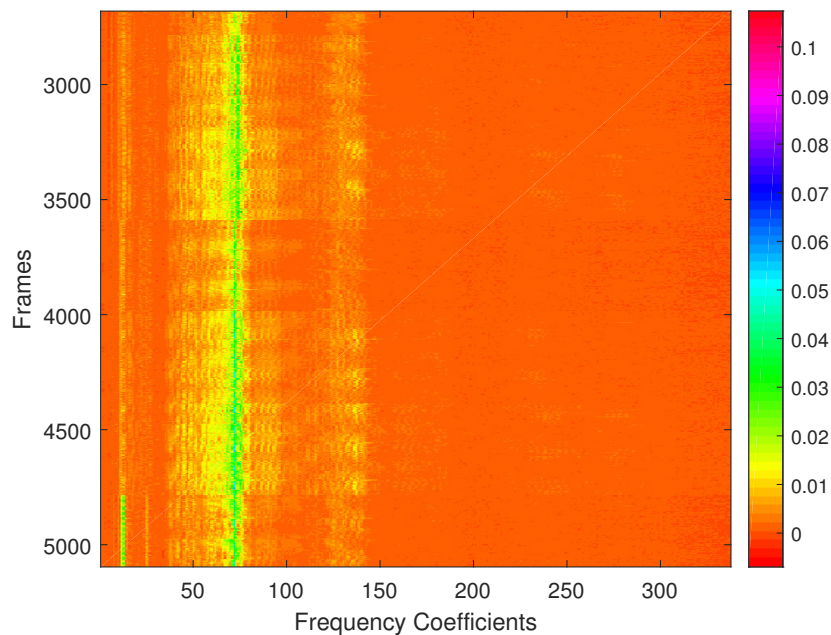
**Figure 4.9:** A 3D plot of the time-frequency data of the vibration measurements from microphone #2

In Figure 4.6 there are some intervals of frequency coefficients that show higher intensity than the others. From frequency coefficient 40 to 100 there is a lot of activity; firstly, a line of high intensity (green and blue) is visible between frequency coefficients

70-80. This line is continuous for all tests. The highest magnitude is seen for the tests of 2400rpm and 2600rpm (frames 7182-9576). The frequency coefficients on each side of this high-intensity interval (i.e 40-70 and 80-100) has a structure in which an increase in the intensity comes and goes with a fixed number of frames. In addition there is an interval of higher intensity at frequency coefficients 130-145, which seems to be quite consistent, but slightly affected by the same pattern as in the previous line. Lastly there is a band at frequency coefficients 13 and 14 which has a higher intensity at frames 4800-7200, and frames 8400 to 9576, with the highest intensity being from frame 4800 to frame 6000 approximately.

For the measurements obtained from the second microphone which are shown in Figure 4.8 there are two intervals of higher intensity. These are found at frequency coefficient 50 and 97, approximately. Some similarities can be seen when compared to Figure 4.6: the intervals around coefficient 50 and 97 seem to be continuous and increasing in intensity for the larger rpms. The highest intensity is found for the line at coefficient 50. The same pattern of varying intensity can be seen for the frequency coefficients on both sides of the high-intensity lines (which also are slightly affected). Moreover, there is a structure around frame 4000 that shifts to the right (higher frequency) for each new rpm; this is visible in both Figure 4.6 and 4.8, but is easiest to spot in the latter.

Using the approach described in the theory chapter, section 2.2.3 and in the method chapter, section 3.3.5, the above frequency coefficients may be transformed into frequencies. The interval of frequency coefficients 40-100 seen in Figure 4.6 correspond to frequencies 1000-2500Hz; the magnitude of these frequencies seems to be largely affected by the load conditions of each test. In Figure 4.6, the pattern of increase and decrease in intensity correspond well with the different load conditions. This is best illustrated with a zoomed view of Figure 4.6, which is given in Figure 4.10.



**Figure 4.10:** A zoomed view of Figure 4.6 which shows the effect of the different load conditions

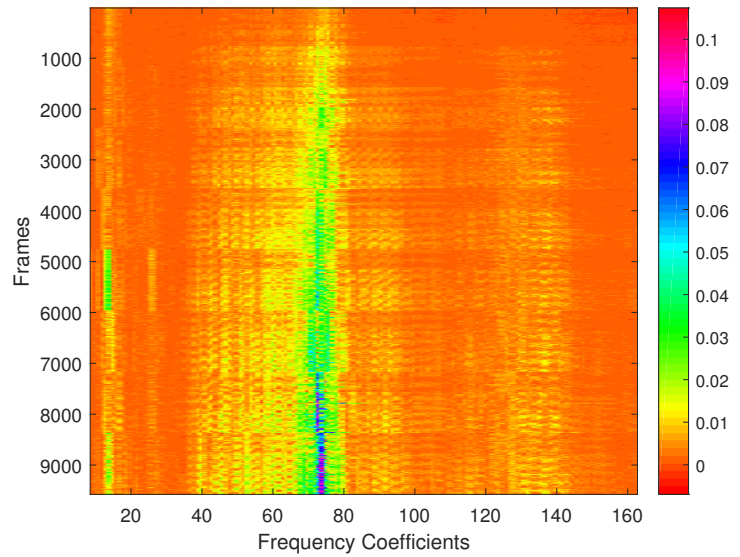
At approximately frame number 3600, a transition from higher intensity to lower intensity is seen in the frequency band given above. The lower intensity lasts for the next 400 frames, i.e. to frame number 4000, where the intensity once again increases. The transition from high intensity to lower intensity at around frame number 3600 corresponds to the transition from Test9 to Test10, which is 1600rpm and 90Nm to 1800rpm and 20Nm, whereas the transition at around frame number 4000 corresponds to the transition from Test11 to Test12, which is 1800rpm and 20Nm to 1800rpm and 50Nm. Thus, the repeating pattern reflects the load conditions for each test, and yet again shows that vibrations increase with increasing load. It is also possible to point out the transition from 50Nm to 90Nm for each test, but this is a bit less visible than the transitions to or from 20Nm. This same pattern is also visible in the measurements from microphone #2. These frequencies which come and go thus seem to be mostly affected by load (at least for the measurements from microphone #1). In section 2.4.2, different noise sources for combustion engines were introduced, and mechanical noise sources were divided into load dependent and load independent groups [50]. The frequencies described in this paragraph seem to relate mostly with the load dependent sources, which includes the fuel injection system and fuel pumps.

For the lines with the highest intensity, namely frequency coefficients 70-80 in Figure 4.6 and frequency coefficients in the vicinity of 50 and 97 for Figure 4.8 the intensity increases with increased rpm. These frequency coefficients correspond to frequencies

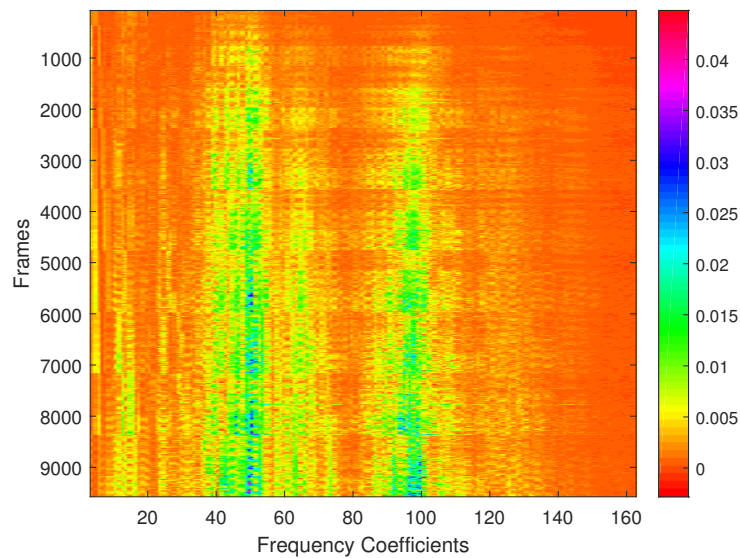
of 1750-2000Hz, 1250Hz, and 2425Hz, respectively. A zoomed view of these frequency coefficients are given in Figure 4.11. The frequency coefficient with the highest intensity in Figure 4.6 is 73,5, which correspond to a frequency of 1837Hz. Although some effect is seen from the load conditions, these frequencies seem to be more dependent on engine speed. As for the grouping of mechanical noise sources, these would relate to the load independent sources, like the camshaft and the valves. Whats interesting is that the frequencies are in the same band, but increase in magnitude with increasing rpm. When it comes to the slightly visible structures that shifts to the right in both Figure 4.6 and 4.8, these events increase in frequency with increasing engine speed. Since microphone #2 is placed near the turbocharger it is likely that the frequencies in Figure 4.11b are linked to the the operation of the turbocharger, whose rotational speed varies from approximately 20 000rpm to 125 000rpm throughout the tests, and increase and decrease with the load conditions.

It is worth mentioning that the plots in Figures 4.1a to 4.4 are made of the vibration data prior to resampling; i.e these measurements have been obtained with different sample rates. However, since the shaft encoder gives an equal number of samples per rotation, independent of engine speed, meaning that each engine cycle has an equal number of samples, it is still very relevant to compare these measurements. Nonetheless, this might be the reason to the "stretching" of the combustion peaks seen for the higher rpms; the combustion events happen faster in terms of time, but the number of samples remain the same. Thus, the effect of increased sample rate for the higher rpms is diminished. The increase of the vibration amplitude peaks as well as the increase of the overall vibration level might also contribute to the stretched appearance. The vibration measurements could have been resampled before comparison, however as the number of samples per engine cycle remained the same this was thought to a better method as it led to an equal number of samples for the same number of engine cycles, and thus easier data processing.

Results from both time-domain and time-frequency domain indicates that vibration is dependent on load and speed when measured in a diesel engine. The RMS values and the three-dimensional time-frequency plots have a stair-like structure, where the largest peaks correspond to the largest load conditions for all rpms. Furthermore, the smallest load condition for one rpm has less vibration than the largest load condition for the preceding rpm, thus indicating that load has more influence on the vibration than a (small) increase in rpm. The 90Nm peaks is even more apparent in the vibrations from microphone #2 (Figure 4.9). These findings correspond well with the findings of Ball *et al* [55], where it is shown that the RMS value of the (airborne) sound signals of an engine increases with engine load and speed. Therefore this is not a new revelation, but it shows how the obtained data correspond with research done elsewhere, and thus increases the



(a) Zoomed view of the high intensity area of the measurements from microphone #1



(b) Zoomed view of the high intensity area of the measurements from microphone #2

**Figure 4.11:** Zoomed views of the high intensity bands of the vibration measurements. Some of the frequencies show a tendency to increase in magnitude with the engine speed.

integrity of the data set.

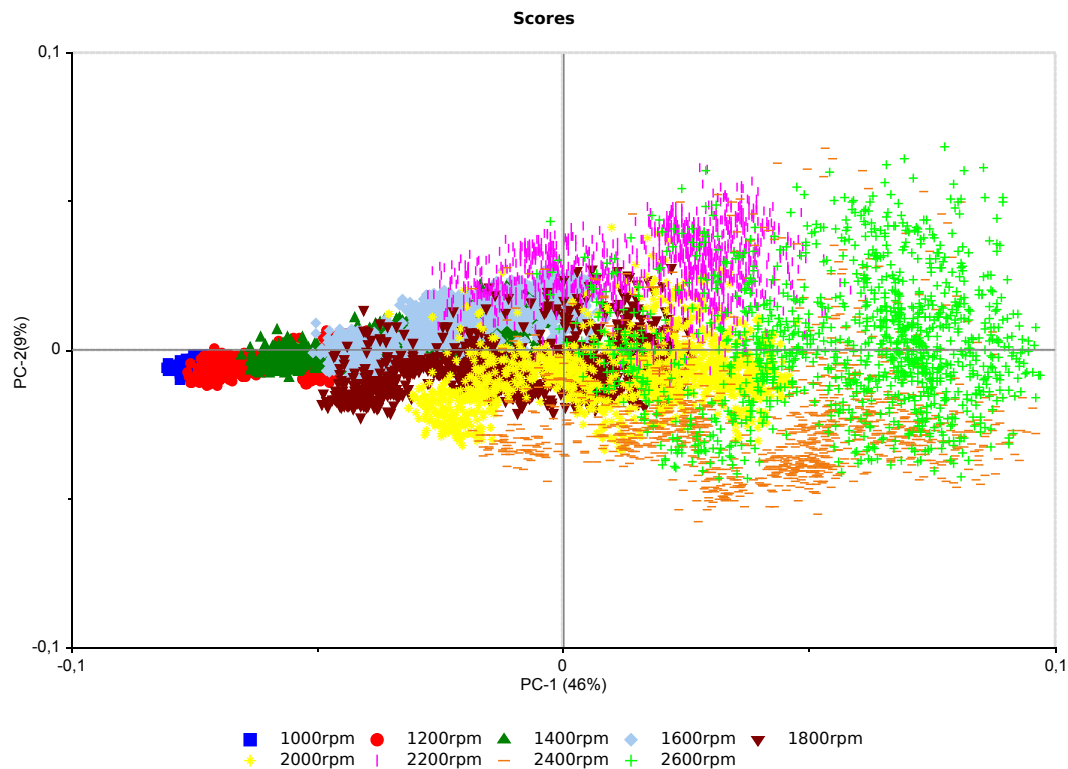
In an engine, it is reasonable to believe that most sources of noise and vibration are related to the engine speed in such a way that the vibration or noise would come with a higher frequency at higher rpms. However, the frequencies with highest intensity found here, seem to only increase in magnitude when the speed increases, and thus is not affected by the speed directly. A process that could be tied to these frequencies are the functions of the pressure regulator in the common rail injection system. When rpm increases, the fuel pump works faster to deliver the amount of fuel required and to maintain the pressure in the common rail. The pressure in the rail quickly reaches its maximum level. To prevent the pressure from exceeding its maximum level due to further increase in rpm, a control loop with a pressure sensor and a pressure control valve (PCV) regulates the pressure by returning fuel to the tank. Thus, the pressure remains the same but the amount of fuel returned is likely to increase at higher rpms. The increased amount of fuel returned could provide larger vibrations. That the frequency is "static" could be related to the pressure regulator (the PCV), which opens and closes at regular intervals. However, this is only a theory and more knowledge of the fuel system of diesel engines is necessary for further evaluation.

## 4.2 PCA Models

As described in chapter 3, the resulting matrix of the Fourier transform of the vibration measurements from microphone #1 and microphone #2 was loaded into The Unscrambler®X where a PCA was carried out. A PCA was also carried out on the large table where all measurements are included. In total three different PCA models were created; one where the frequency matrix was analysed directly, one where the frequency matrix was analysed with a weighting of 1/standard deviation and one where all measurements (frequency, high speed, low speed) were included with the weighting 1/standard deviation. Each of these models are presented in the following subsections.

### 4.2.1 PCA Model 1

This section shows the results after performing a PCA on the resampled vibration data in the time-frequency domain without the use of any weighting. Row ranges for the different rpms have been made which makes it possible to use colours to separate sample groups in the score plots. Figure 4.12 shows the score plot of the two first principal components.

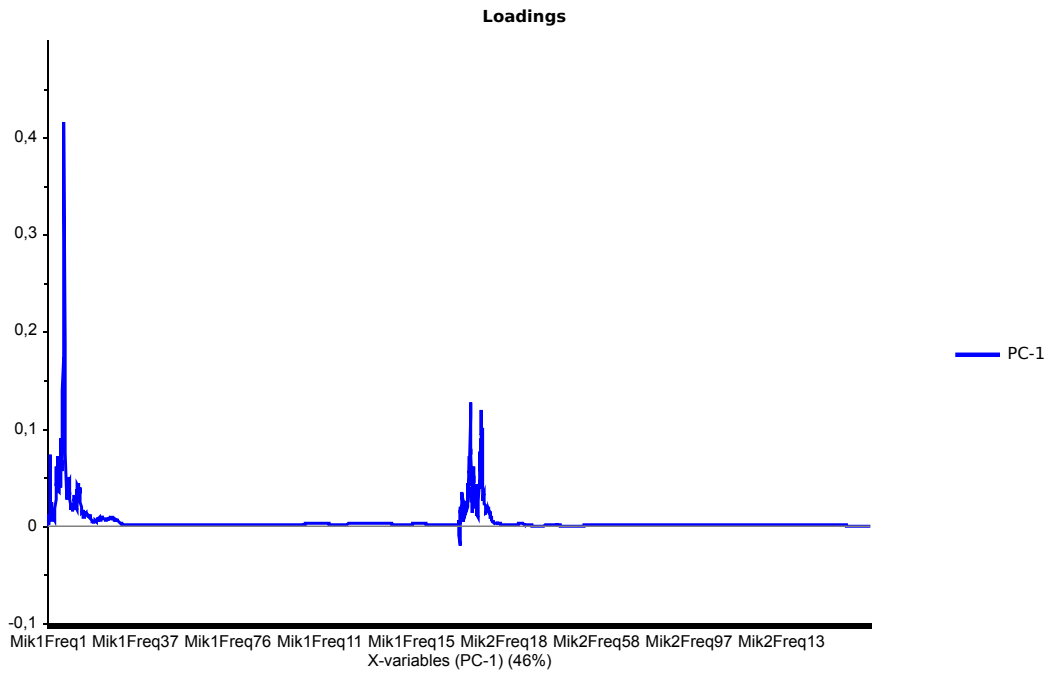


**Figure 4.12:** Scores of PC1 and PC2 from the first PCA model plotted against each other. The different rpms have been coloured

As can be seen from the figure, the different rpms have been coloured; i.e all load conditions for each rpm is expressed with the same colour. The first thing that is obvious from the figure is that the rpm increases along the horizontal axis, which is the first principal component. Moreover, the load conditions for each rpm increase along the first PC (this is hard to tell without labels on the samples, however this makes the plot very messy, thus they have been kept out in this presentation). The samples for each rpm have been clustered together, however as the rpm increases the samples are more spread. Starting at 1400rpm, the samples seem more spread due to the influence of the vertical axis (PC2). The largest spread of samples is seen for the highest rpm. An interesting observation is that the majority of samples for 2200rpm have positive values along PC2, whereas the majority of samples for 2400rpm have negative values for PC2.

Figure 4.13 shows the loadings of the first principal component. A large peak with a value of 0,4 is seen for frequency coefficient 73 and 74 from microphone #1, and two peaks of value 0,13 and 0,12 is seen at frequency coefficients  $50 \pm 1$  and  $97 \pm 2$  from microphone #2. This corresponds very well with the lines of high intensity seen in the spectrograms in section 4.1.2 ( Figures 4.6 and 4.8).



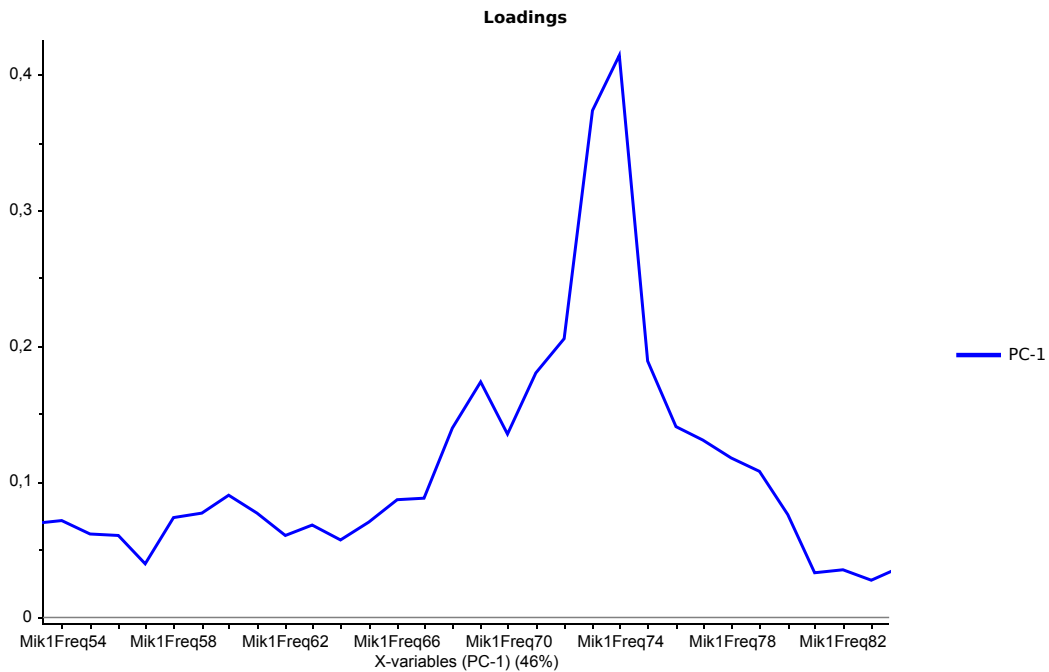


**Figure 4.13:** The loadings of PC1 for the first PCA model

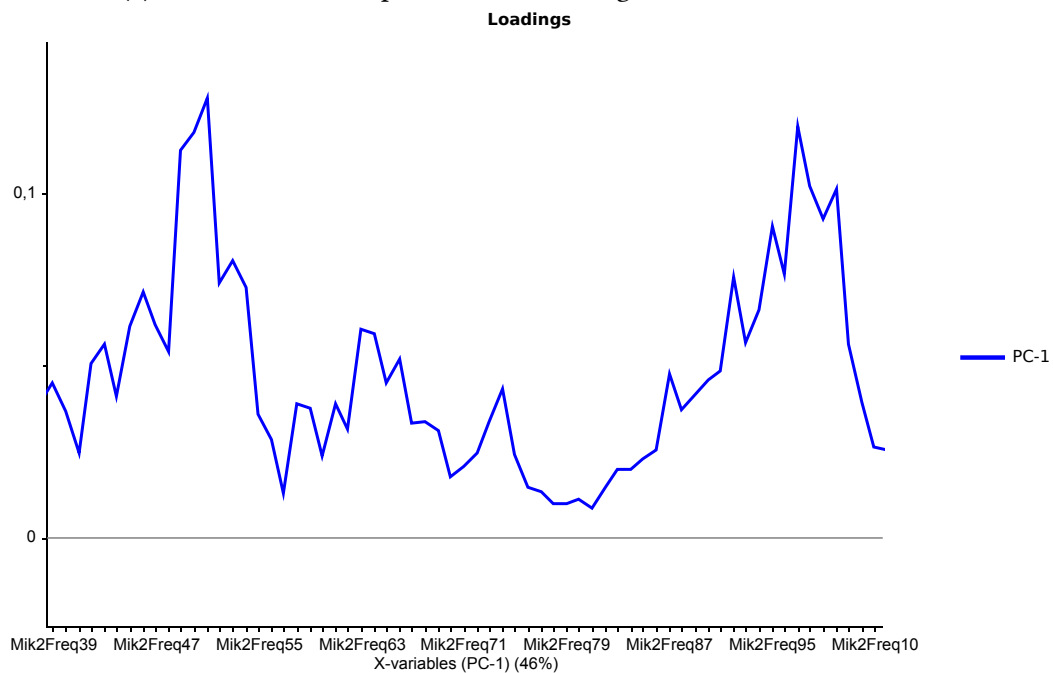
As there are very many frequency variables, the labels on the horizontal axis have been cut a bit short, therefore zoomed views of the peaks are given in Figure 4.14. In section 4.1.2 it was discussed that the intensity of these frequencies seemed to increase with increasing engine speed; that the same frequencies now are the variables that have contributed the most to the first principal component further supports the assertion that the engine speed is described by PC1. The first PC also has the largest amount of explained variance; this can be seen in the parentheses on the axis labels of all the plots. PC1 has 46% explained variance and PC2 has 9% explained variance. This means that the variation seen along the vertical axis is only responsible for 9% of the variation in the data set.

In the interpretation of PCA loadings it is so that if a large positive peak (large relative to the scale -1 to 1) is found for a variable in the loadings of a principal component, then the samples which have a high score value along the same PC have values for that variable that are larger than average. In this model a peak occurs at frequency coefficients 73-74 for PC1 loadings, thus the samples which have (large) positive score values for this PC will have larger values (magnitudes) for these frequencies. From Figure 4.12 it can be seen that the majority of the samples for 2400rpm and close to all samples for 2600rpm have positive score values along PC1, with the samples for 2600rpm having the highest values. This also correspond well with the information provided by the spectrogram in Figure 4.6, where the intensity increased for this frequency for higher rpms. The same

can be said for the two other peaks in the plot of PC1 loadings.



(a) Zoomed view of a peak for PC1 loadings for the first PCA model

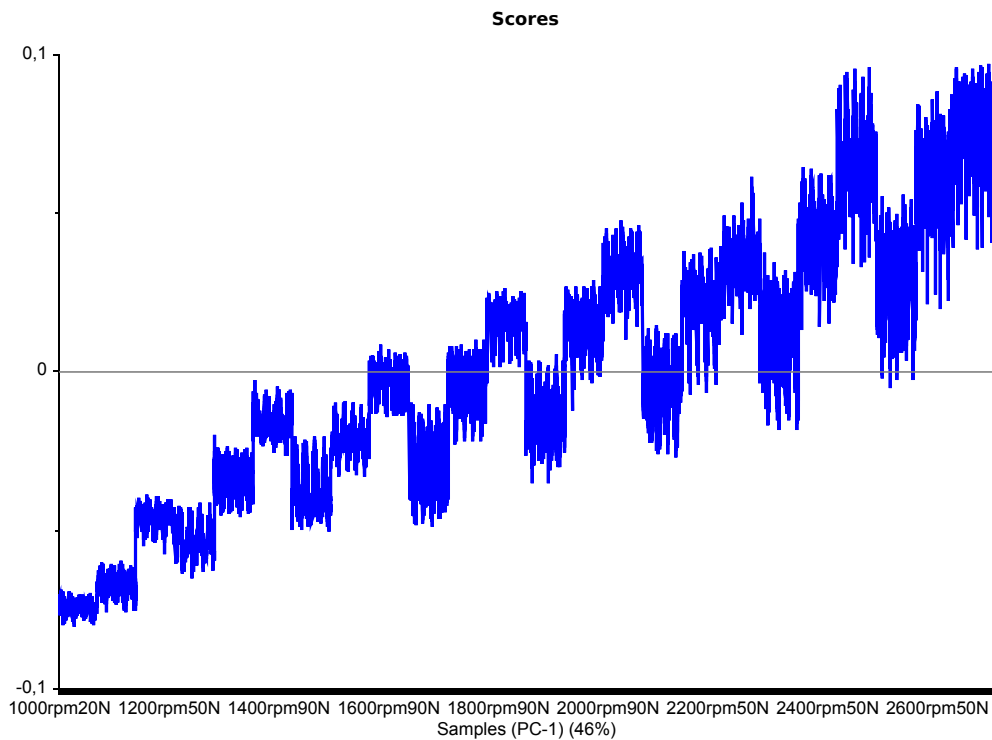


(b) Zoomed view of peaks for PC1 loadings for the first PCA model

**Figure 4.14:** Zoomed views of the peaks for PC1 loadings in the first PCA model

The scores of PC1 are shown in Figure 4.15. It has a familiar structure; once more the staircase structure is apparent, and it is easy to separate the different load conditions

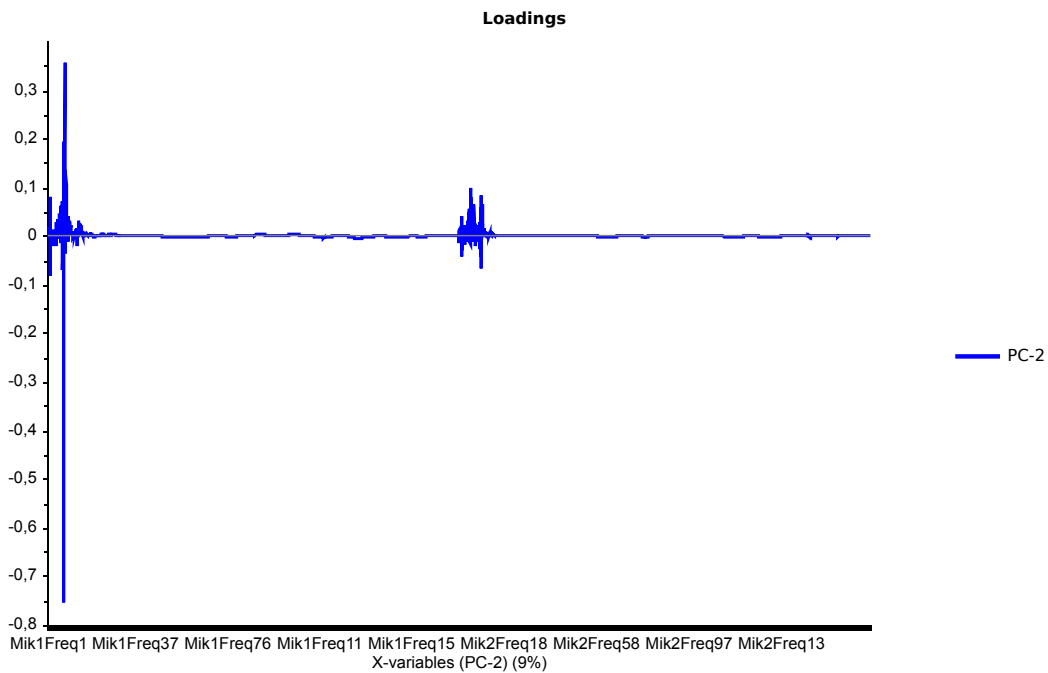
for each test. For 1000rpm there is only one load condition, whereas there are two load conditions for 1200rpm and three for all other rpms. It can be seen that the score value increases when the rpm increases; furthermore it also increases with increased load, and as before the largest load condition of one rpm has a higher score value than the lowest load condition of the succeeding rpm. This yet again matches very well with the results in the time and time-frequency domains. As the plot of PC1 loadings had three prominent peaks, the value of the scores in PC1 describes the value each sample has for the variables (here: frequency coefficients) that these peaks correspond to. It can be validated in the spectrograms in section 4.1.2 that the magnitude is larger for these frequencies at the higher rpms. The score values for the highest load conditions for 2400 rpm and 2600rpm is quite similar.



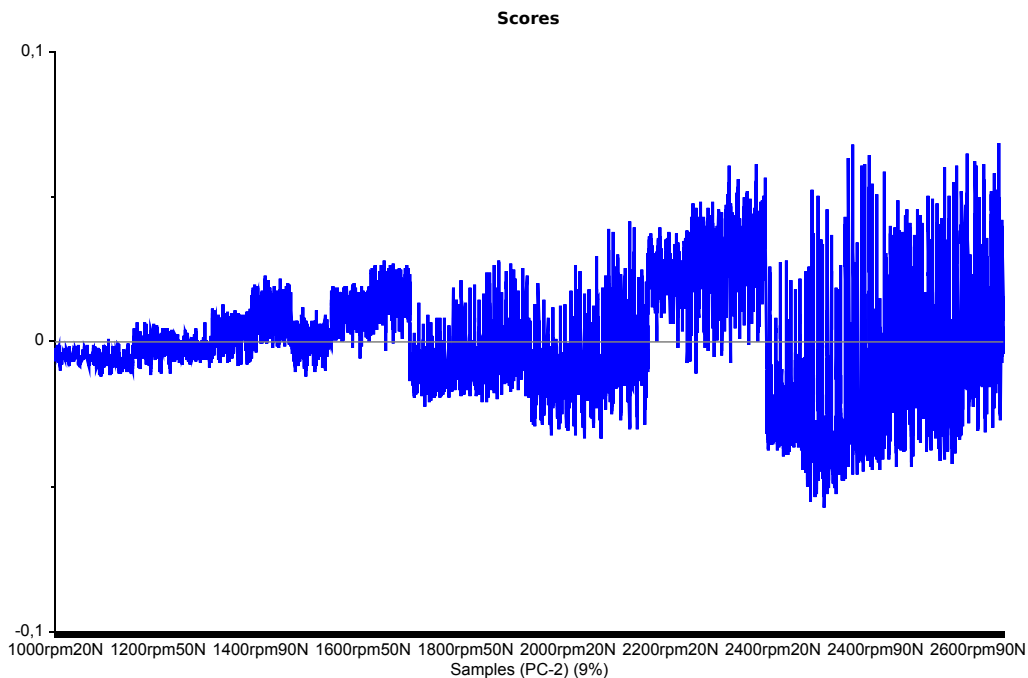
**Figure 4.15:** The scores of PC1 for the first PCA model

Figure 4.16 and 4.17 shows the loadings and score values for the second principal component. Some of the positive peaks seen for PC1 loadings appear here as negative peaks. This is expected, as there should be no correlation between the principal components. The largest negative peak is seen for frequency coefficient 73 from microphone #1, whereas the largest positive peak is seen for frequency coefficients 75 and 76 for microphone #1. In addition, there are both negative and positive peaks for some of the frequency coefficients from microphone #2; positive peaks are seen at frequency coefficient 50 and

98, and a negative peak can be found at frequency coefficient 97. However the values of these peaks are quite low, thus one should be careful when interpreting their relevance.



**Figure 4.16:** The loadings of PC2 for the first PCA model



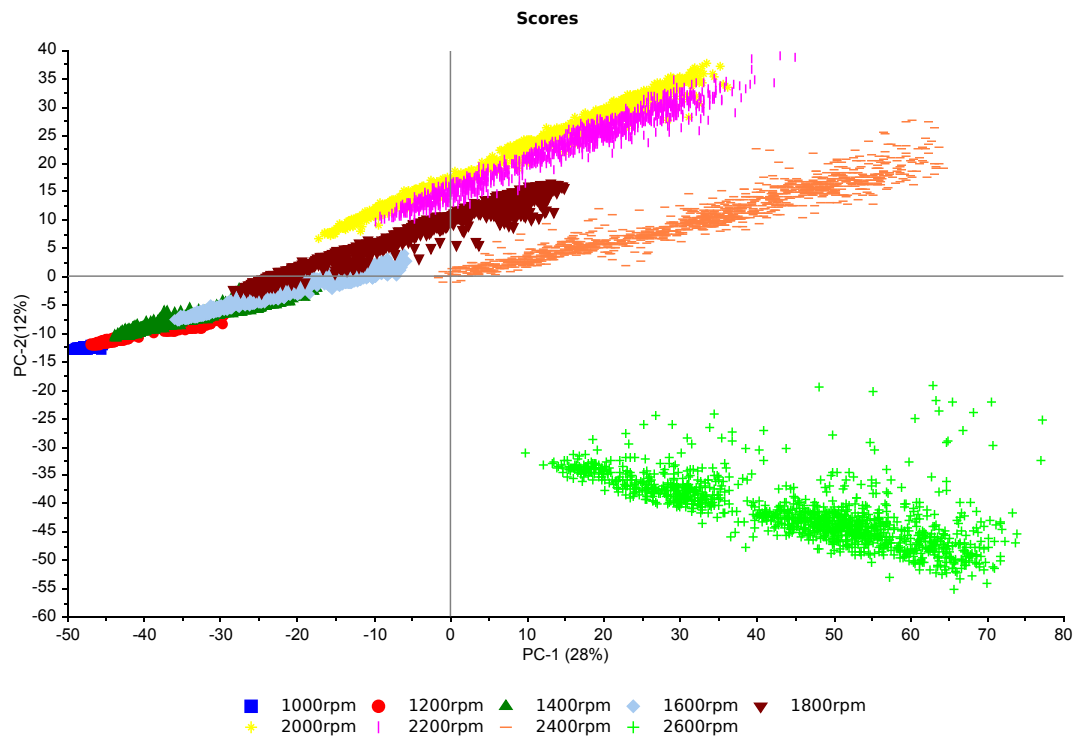
**Figure 4.17:** The scores of PC2 for the first PCA model

It can be seen that the scores and loadings for the second PC does not provide so

much new information; it generally shows how the spread in the samples increase with increasing rpm, i.e that the variation in intensity for frequency coefficient 75 and 76 is larger for higher rpms. The PCA analysis has reduced the original data set consisting of 4096 variables (frequency coefficients, 2048 for each microphone) to a data set with only two variables, PC1 and PC2. Only two PCs were returned; the model did not converge for PC3, meaning that a third score vector fulfilling the convergence criterion was not obtained by the limit of 100 iterations. As PC1 has 46% explained variance, whereas PC2 has 9% explained variance only a total of 55% of the original variation in the data set is explained by these two principal components. However, the most important frequencies were kept, and the stair-like structure was still visible. In section 4.1.2, an attempt was made to tie these frequencies to engine events. The regulation of fuel pressure in the common rail injection system was suggested as a possible source, but no conclusions were drawn. If it is so that these frequencies can be connected to specific events in the engine, then the PCA has managed to extract these events and used them to describe the different operating conditions.

#### 4.2.2 PCA Model 2

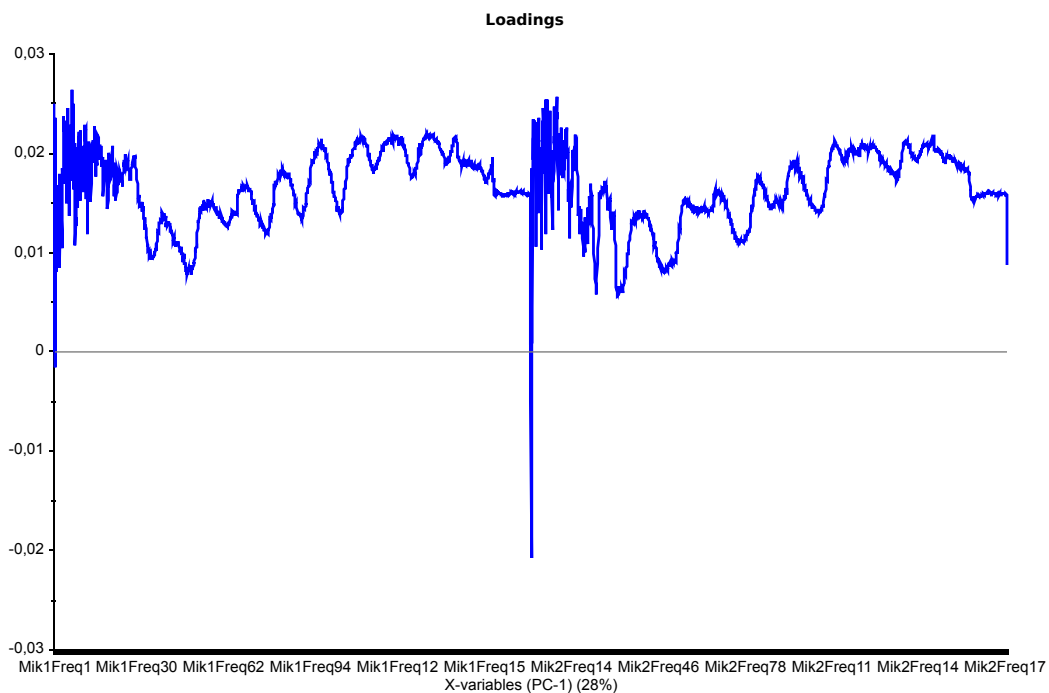
This section shows the results after performing a PCA on the resampled vibration data in the time-frequency domain with the weighting option  $A/SDev$ , where  $A = 1$  and  $B = 0$ . As described in chapter 3, this function finds the standard deviation of each column and multiplies each sample within each column with the ratio  $1/Standard\ deviation$  (of that column). This gave a new structure in the score plot of the two first principal components, which is seen in Figure 4.18.



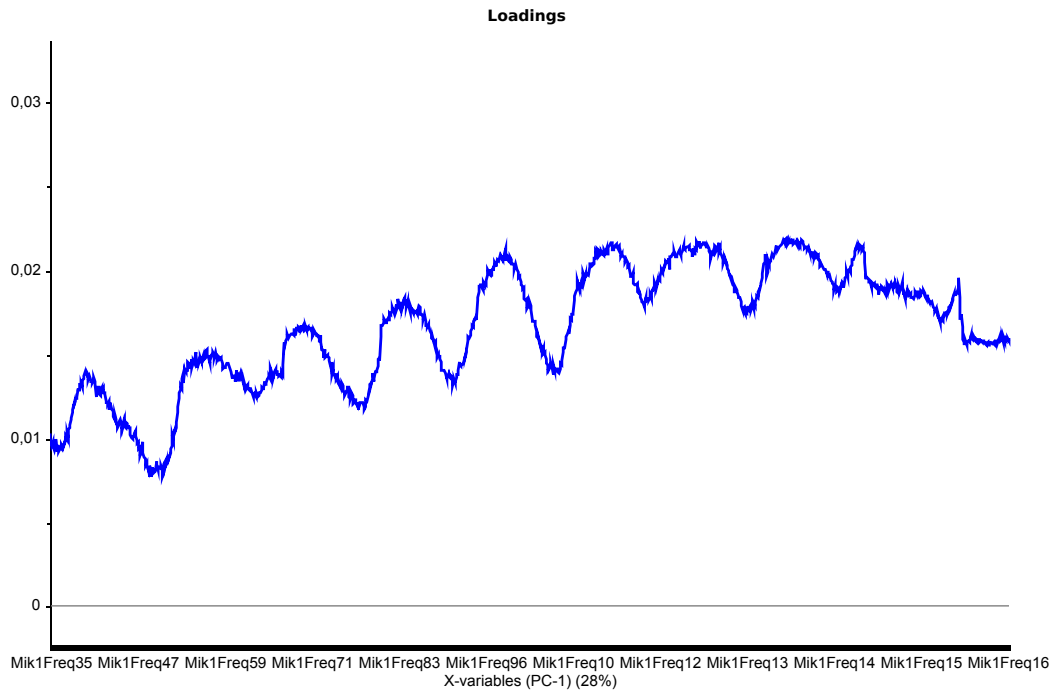
**Figure 4.18:** Scores of PC1 and PC2 from the second PCA model plotted against each other. The different rpms have been coloured

All samples have been coloured after their respective rpms. In this model the clusters for the different rpms are more clear, although some overlap can still be found. The groups for each rpm is shaped almost like lines, where the higher load conditions come after the lower load conditions, i.e. increased load leads to higher values for both horizontal (PC1) and vertical (PC2) axes. Each new rpm is placed slightly higher on the vertical axis and further to the right on the horizontal axis. Some load conditions typically align along PC1; for instance are the samples belonging to the condition 1800rpm and 20Nm at the same horizontal position as the samples of 1600rpm 50Nm and 1400rpm 90Nm. This is also seen for some of the other rpms and load conditions. What separates these samples is their values along PC2. The trend seems to continue up to and including the samples for 2000rpm. The samples for 2200rpm follow approximately the same pattern, however the values along PC2 are now mostly lower than those of the samples belonging to 2000rpm. For the samples of the 2400rpm operating condition, nearly all samples have positive values for PC1; apart from this the samples follow roughly the same trend as the lower rpms, where values increase for both PC1 and PC2 when the load increases. The largest deviation is seen for the samples belonging to 2600rpm, where all samples have positive values along PC1, but negative values along PC2. Furthermore, the values along PC2 decreases as the load increases.

Figure 4.19 shows the loadings of the first principal component. The first thing one can see is a pattern which is repeated for consecutive frequency coefficients, both for microphone #1 and microphone #2. This structure illustrates something that comes and goes periodically, but increases in magnitude. The structure is a bit more inconsistent in the frequency coefficients from microphone #2, but still seems to represent the same. The structure have an average period of 137 frequency coefficients for microphone #1, and 138 frequency coefficients for microphone #2. This corresponds to a frequency of approximately 3425Hz. A zoomed view of the structure for the frequency coefficients of microphone #1 is given in Figure 4.20. This could be related to a resonance effect. Other things worth noting from the loadings of the first PC is that there are positive peaks for frequency coefficient 69 (microphone #1) and 97 (microphone #2). This is almost the same as the peaks seen in PC1 loadings in the first PCA model, and indicates that the samples with positive score value along PC1 has a higher magnitude for these frequencies. The samples belonging to the higher load conditions for 2000rpm, 2200rpm and all samples for 2400rpm and 2600rpm have positive score values along PC1. One prominent negative peak is seen at frequency coefficient 4 of microphone #2, and means that the lower rpms have a higher magnitude for this frequency, because these have negative values along the first principal component. The score plot of PC1 is in fact almost identical to the one given for the first PCA model in Figure 4.15, thus it is not so interesting to present this again.



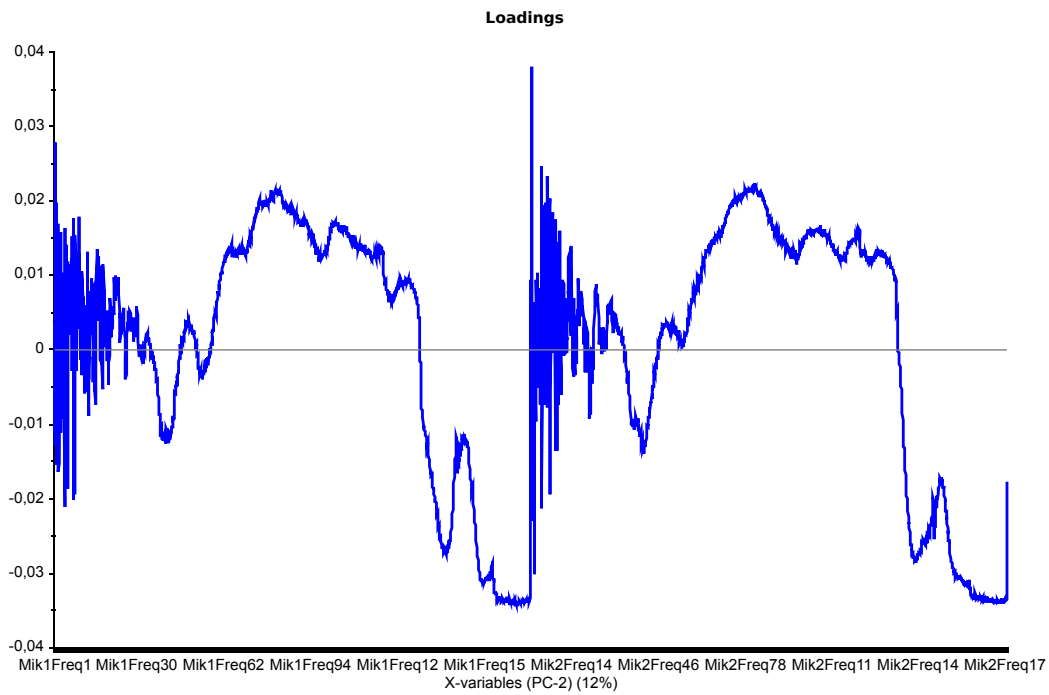
**Figure 4.19:** The loadings of PC1 for the second PCA model



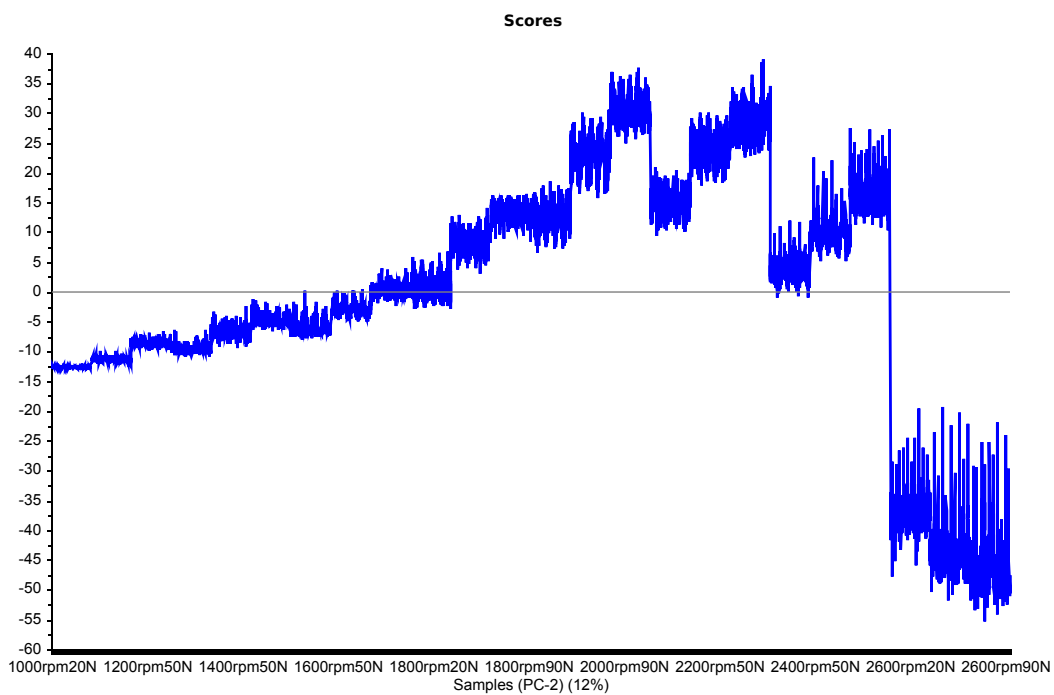
**Figure 4.20:** Zoomed view of the frequency structure in the loadings of PC1 for the second PCA model

Figure 4.21 and 4.22 shows the loadings and the scores for the second principal component. In the loadings plot there is a structure that resembles the one seen in Figure 4.19, however now it is much less periodic. The structure is caught by both microphones. In addition there are positive peaks at frequency coefficient 5 for both microphones, and the largest negative peaks are found at frequency coefficients 40, 74 and 79 for microphone #1 and 17, 7 and 43 for microphone #2. Figure 4.22 shows that the score values increases for rpm and load until the samples for 2200rpm, where they start to drop slightly and then a large drop is seen for the samples at 2600rpm. This of course corresponds with the separation seen in Figure 4.18.





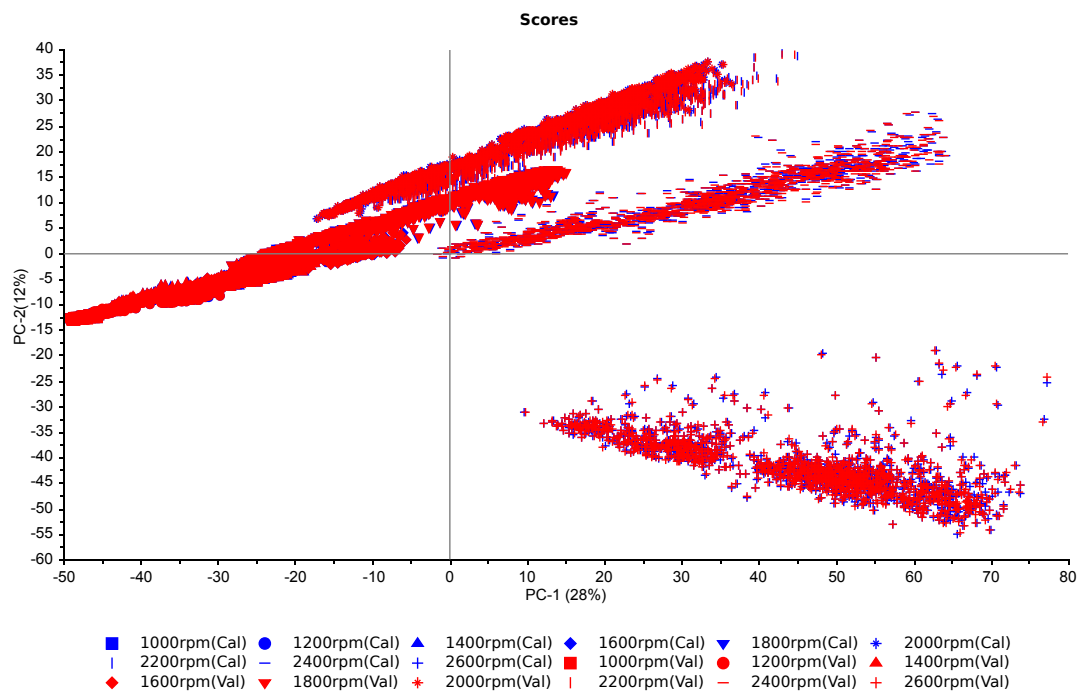
**Figure 4.21:** The loadings of PC2 for the second PCA model



**Figure 4.22:** The scores of PC2 for the second PCA model

The third principal component only has 6% explained variance and provides little new information. The loadings has many similarities with the loadings of the two first

components, with positive and negative peaks at various frequency coefficients, and a structure resembling the one discussed for the first PC, however yet again not periodic. As discussed in the theory chapter (ch. 2), validation is an important part of multivariate analysis when making models. In the case of PCA, validation is most relevant if the model is to be used to classify new data samples, and for this reason it is not so important here where the PCA mostly has been used as a descriptive tool. Nevertheless, because validation is automatically carried out by The Unscrambler®X, some validation has been done for the PCA model. As described in chapter 3, cross validation with 20 segments was carried out. Figure 4.23 shows the scores for PC1 and PC2 for both calibration and validation; as far as the eye can see, the model has obtained good results for validation, however this is not very exact. Although a bit hard to see from this plot, in general, the validation samples have been placed correctly in their rpm groups as divided by the calibration. For now, it is deemed good enough and will be discussed no further. Outliers have not been considered, but what can be said is that there are no extreme outliers, but that there is more spread among the samples for the higher rpms. This model could now act as a reference model for the operating range of the diesel engine. The model has a good range of different rpms, but only three load conditions have been considered. In Table 3.1 it can be seen that the engine can produce as much as 470Nm in the range 1800rpm-2600rpm. Thus, as the largest load condition considered here is 90Nm, the load range is not so representative of the engines (possible) operating range.



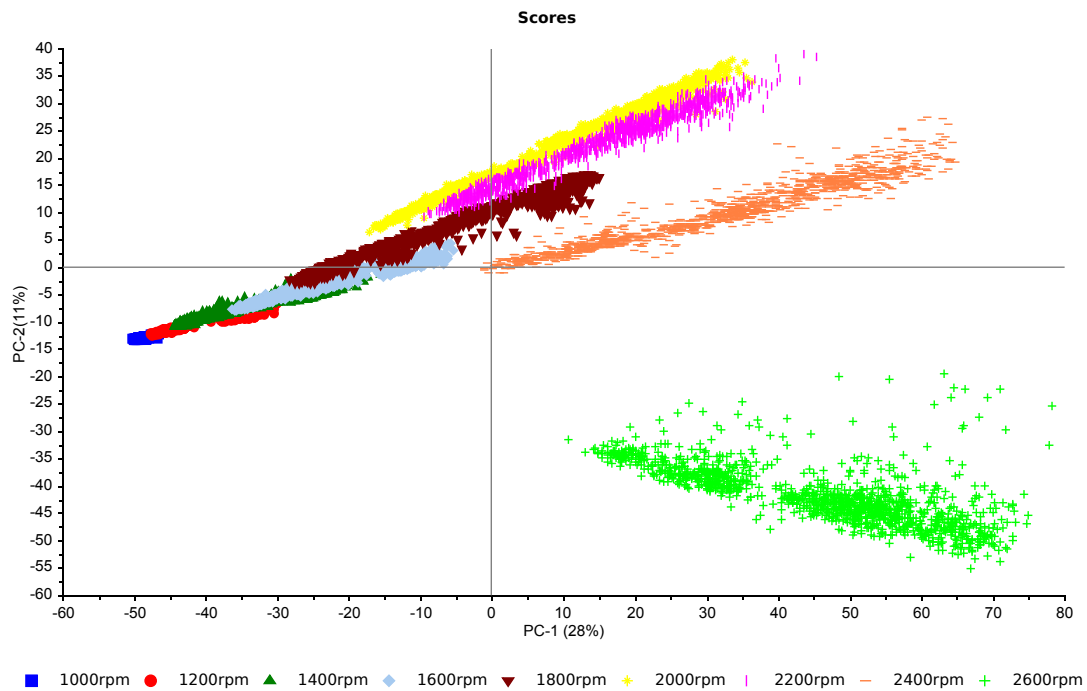
**Figure 4.23:** Calibration scores and Validation scores in same plot

An interesting feature of this model when compared to the first PCA model is the revelation of the periodic frequency structure. This structure was not visible in the loadings of the first PCA model, but by implementing the weighting of  $1/\text{standard deviation}$ , the structure appeared on the exact same data set. This frequency structure was very prominent in the loadings, and it was found that it had a period of approximately 3425Hz. A hint was given that this resembled a resonance effect. The engine block is made of cast iron, of which the speed of sound is 4600m/s. The product of the period of the frequency structure and the speed of sound in cast iron results in a distance of 1.34m, approximately. This is not far from the length of the engine block, which can be estimated through the following: the bore of the cylinders is found in Table 3.1 to be 88mm. For 6 cylinder this corresponds to 528mm, or approximately 0,53m. In addition, there is space between each cylinder and on either side of the cylinders. From this, the length of the engine block can be estimated to be around 1m. Thus, it is likely that the PCA has found the resonance in the engine block. This is an interesting result, as it probably is transmissible to other engines of similar design.

### 4.2.3 PCA Model 3

This section shows the results after performing a PCA on all the measurements combined, i.e all resampled vibration data in time-frequency domain along with the high speed measurements and the low speed measurements. This was done in order to investigate the relationship between the frequencies and the measurement variables. The results of this PCA gives an indication of if and which variables that are correlated with the vibration data, and thus gives a hint of which variables that will perform well during regression.

Figure 4.24 shows the scores of PC1 and PC2; this is almost identical with the representation given in Figure 4.18, which means that the frequencies are the among the most important variables to describe the variation between the different operating conditions. Looking at the scores and loadings for PC1, PC2 and PC3 they take the same shape as for the second PCA model; the only difference is the introduction of a few new variables which affect the model slightly. In PC1 loadings some new peaks appear for the measurement variables. Firstly, a large negative peak is seen for the variable "Weight". The second largest negative peak is the variable "Room humidity". In addition, a negative peak appear for "Lambda estimated ECU". The most prominent positive peaks are seen at variables "Room temperature", "Fuel consumption return total", "Fuel consumption inlet total", "Airmass flow ECU", "Fuel Consumption ECU", "Intake temperature before charger", "Intake temperature after cooler" and "Spill water temperature". There are also peaks for "Engine speed ECU", "Mass flow rate", "Oil temperature", "orifice plate pressure",



**Figure 4.24:** Scores of PC1 and PC2 for the third PCA model plotted against each other. The different rpms have been coloured. The plot is almost identical to that in Figure 4.18

"EGT cylinder 1", "Oil pressure" and "Begin of main injection", however they are less prominent.

In PC2 loadings there are negative peaks for "Exhaust pressure", "Fuel temperature", "Turbo speed", and "Boost valve" in addition to some of the peaks that were positive for PC1 loadings. The positive peaks for PC2 loadings are found at the variables "Fuel Quantity ECU", "Load pump", "Injection volume ECU", "Engine torque (high speed)", "Engine torque (low speed)" and "Duration of main injection ECU". There are also peaks at several other variables, for instance "Cylinder pressure", but these are less outstanding. PC3 only has 6% explained variance, and the most outstanding peaks of measurement variables are the same as for the two first components, with the largest positive peak being "Lambda estimated ECU" and "Injection volume ECU".

Although the frequencies and the frequency structure seem to dominate the analysis, some variables were found to have an influence on the model; these variables should have some correlation with the vibration data, and thus gives an indication of which variables that potentially could do well in the regression analysis. Due to the dominance of the frequency variables, one should be careful to interpret the model in relation to the other measurement variables. However it should still be possible to extract some information based on the most prominent peaks for the measurement variables. The

largest negative peak for PC1 loadings was the variable "Weight". This is the weight of the exhaust gases measured in [g]. In relation to the score plot this should mean that the weight of the exhaust gases decrease with increasing load and rpm, as the lower rpms have negative score values along PC1. This could be related to air-fuel ratio in the cylinders at different rpms. In fact, the third largest negative peak is as mentioned above "Lambda estimated ECU". As described in the theory chapter, this is a dimensionless number describing the ratio of actual air-to-fuel ratio to an optimal air-to-fuel ratio. That lambda estimated decreases alongside the weight makes perfect sense; at the lower rpms both weight and lambda has higher values, and a high value of lambda indicate that the combustion mixture is lean, i.e it has a ratio higher than the stoichiometric ratio. This will lead to production of more NO<sub>x</sub>-particles, which of course increases the weight. For the higher rpms, both weight and lambda have lower values. This means that the combustion mixture is "richer". As diesel engines in general have excess air ( $\lambda > 1$ ), this only means that the amount of excess air is less at higher rpms.

Looking at the positive peaks of PC1 loadings the majority was related to temperatures and fuel consumption. That fuel consumption increases for increased load and for higher rpms matches well with the above statement that the mixtures get richer (lambda decrease) at increased load and rpm. At higher rpms the engine burns more fuel, and produces more heat, which makes it logical that the room temperature and spill water temperature increase. However one should be careful when considering temperature measurements, because even though the engine was run at a 15-minute period prior to data logging, temperatures in the engine might still increase as a result of time in operation, and is therefore not necessarily only connected to the increased rpms and loads. That is, the tests with low rpm were acquired first, and then the rpm was increased gradually so that the test with the highest rpm was taken in the end. Thus, for the final test the engine had been run for a longer period than for the first test. The loadings for PC2 and PC3 are even more dominated by the frequencies and frequency structures, thus interpreting the relevance of the measurement variables will not be done for these components. The variable "Fuel consumption return total" were among the most positive peaks for PC1 loadings. This indicates that the return of fuel to the tank is larger for the higher rpms, because these have higher positive score values along PC1. This further substantiates the theory of the frequencies and the fuel regulator discussed in section 4.1.2. The theory was based on that the operation of the fuel regulator was approximately constant, but that the increase of return fuel flow could lead to larger vibrations.

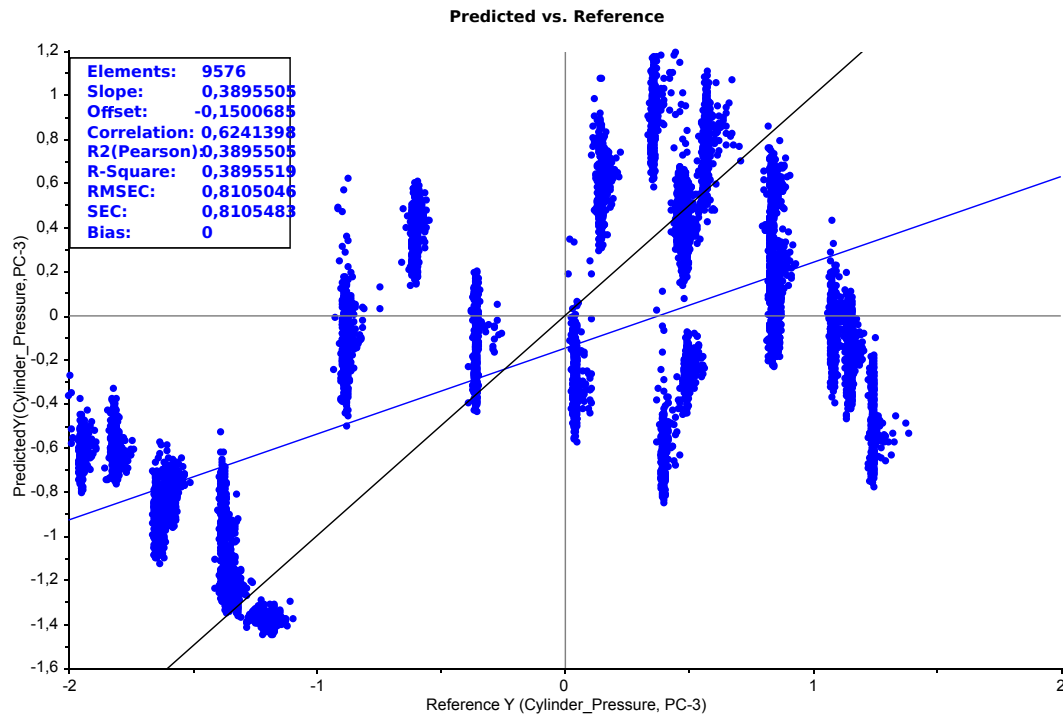
## 4.3 Regression models

This section shows the results of using the vibration data in the time-frequency domain as predictors for some of the measurement variables using different regression methods. Comparisons were made with regression models where engine speed is used as the predictor variable. As the intention of the project was to investigate relationships with variables such as oil temperature and cylinder pressure, most emphasis is put on the regression models of these variables. However, as the third PCA model provided information on the correlation of some other variables, regression models have been made for some of them as well. In addition, results are compared between methods, mainly PCR and PLSR.

### 4.3.1 Cylinder Pressure

The first regression model considers the cylinder pressure. Since the combustion process is the largest contributor to noise and vibration in diesel engines as discussed in section 2.4.2, it is reasonable to assume that there will be a correlation between the cylinder pressure and the vibrations. The reason to investigate the cylinder pressure is because the cylinder pressure sensors often are mounted inside the cylinders, and thus are intrusive in nature. In the test rig for this project, the pressure sensor is fitted in the glow plug hole of cylinder number 5. Initially a PCR was performed, with the vibration data in time-frequency domain from both microphones as predictors and the cylinder pressure as the response variable. Figure 4.25 shows the connection between the predicted cylinder pressure values and the reference cylinder pressure values. As described in section 2.3.4, ideally one would want a slope and R-squared close to 1. The regression line and target lines are shown in blue and black, respectively.

In the figure, the model is shown with three components. It is easy to see that the model is not good when looking at the regression measures. The first thing to notice is the large spread of the data and the deviation between the target line and the regression line. In the data table given in the figure, several regression measures are given. It can be seen that both the slope and R-squared have values lower than 0,4. Furthermore the correlation between the predicted and reference values are 0,62 and the average modelling error (RMSEC) is 0,81. Figure 4.26 shows the same predictors and response variables, but this time a PLSR has been done. The result of the PLSR is clearly better than that of the PCR; less dispersion is seen of the values and higher values are seen for slope, R-squared and correlation, whereas the RMSEC is reduced significantly. In the predicted versus reference plot of the PLSR there is initially a group which is placed a bit higher than what seems to be the trend along the regression line; these samples belong to the

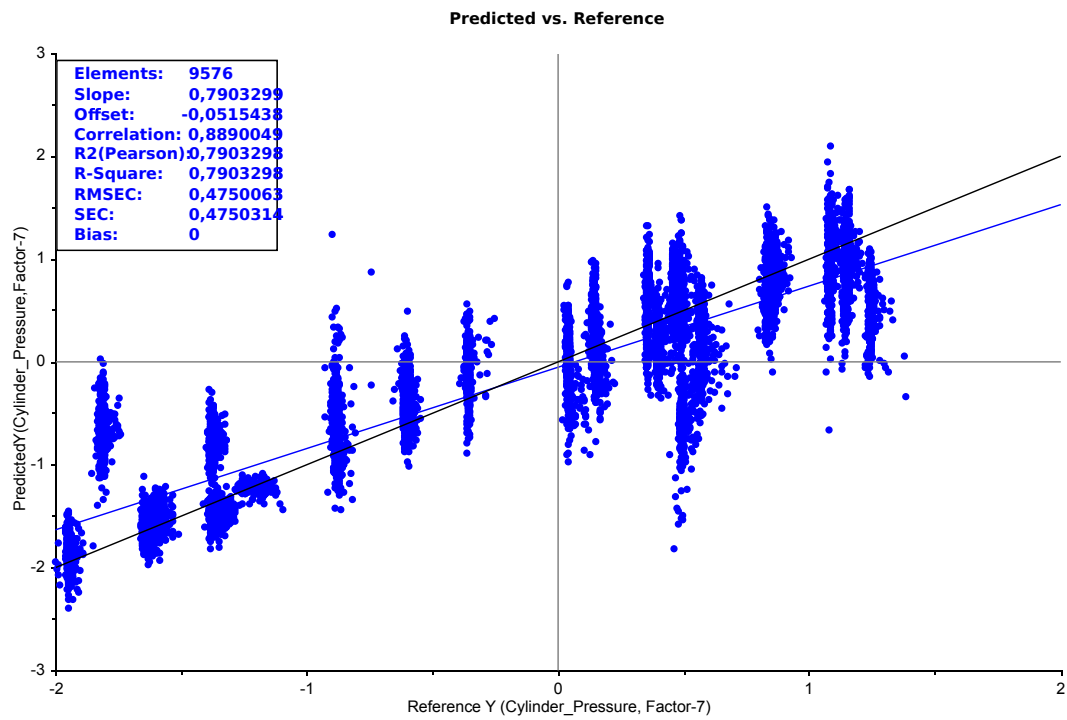


**Figure 4.25:** Predicted versus reference cylinder pressure values. Vibration data in time-frequency domain has been used as predictors in a PCR model.

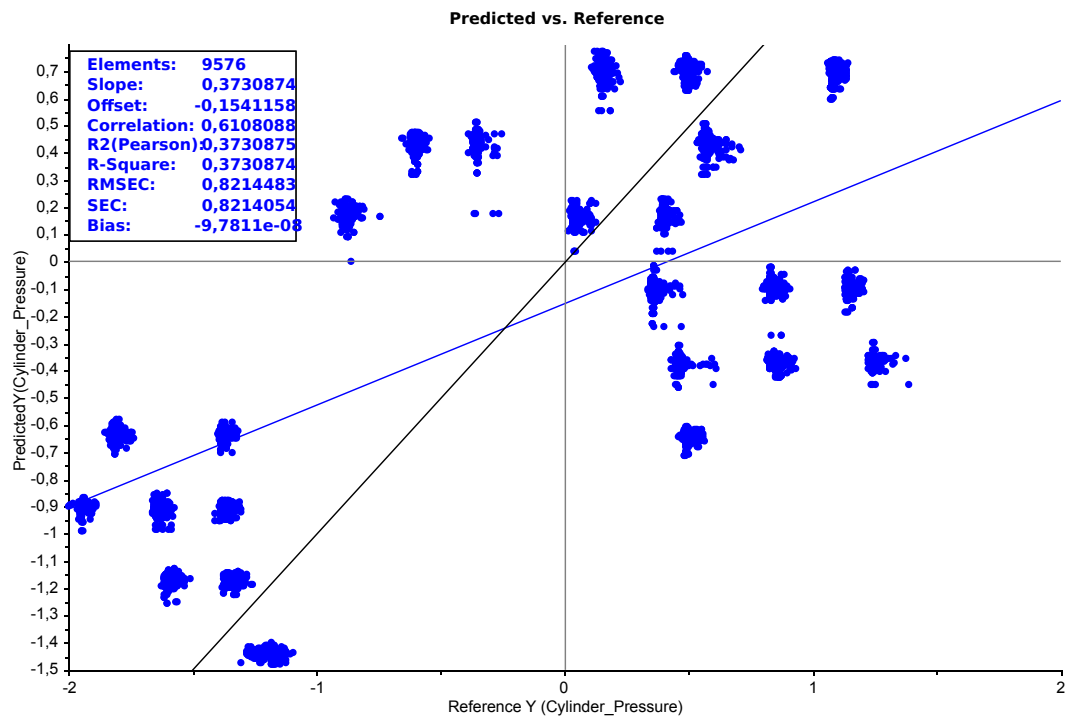
operating condition 1600rpm and 50Nm. No obvious trend is seen for the rpms but in general the cylinder pressure seems to increase for the larger rpms, with some exceptions. Furthermore, the load conditions for the respective rpms seem to go from high to low along the regression line; i.e the larger load conditions have lower cylinder pressures.

To evaluate whether it makes sense to use vibrations as predictors, the regression results are compared towards a regression model where engine speed has been used as the predictor. Since only one variable is used as a predictor, MLR is used. Figure 4.27 shows the predicted versus reference cylinder pressure values. What can be seen for the figure is that there are 24 clusters which are separated well from each other. Closer investigation shows that each of the clusters correspond to one of the operating conditions of the tests presented in Table 3.4. For the clusters, the general trend is an increase in rpm vertically, and a decrease in load from right to left in the plot; the higher load conditions are placed further to the left than the lower load conditions for the same rpm. As for the regression measures, they show similar values to those of the PCR model seen in Figure 4.25.

Table 4.1 shows some regression measures for the three models presented above. It can be seen that the PLSR model is better than both the PCR model and the MLR model where engine speed has been used as the predictor variable. Firstly, the slope and R-square values for the PLSR model is twice as large as for the other two models, and



**Figure 4.26:** Predicted versus reference cylinder pressure values. Vibration data in time-frequency domain has been used as predictors in a PLSR model



**Figure 4.27:** Predicted versus reference cylinder pressure values. Engine speed data has been used as predictors in an MLR model.



it has a higher correlation between the predicted and reference values. Furthermore, the root mean square error of calibration is smaller. From the regression measures of validation the model fit can be evaluated. It is important to keep in mind the number of components in the model, as this has relevance towards the fit of the model and the prediction measures. It can be seen that the root mean square error of cross validation is slightly higher than that of calibration. The deviation is only approximately 0,01, however this is for 7 factors. As the original unit of the Cylinder pressure is given in [MPaG] (megapascal gauge), so is the RMSECV. The RMSECV is approximately 0,48MPaG, which indicates that future predictions are expected to have a prediction error of approximately 0,48MPaG. In the range -2 to 2 this gives an error of 12%. Furthermore, the R-square is approximately 0,78 in validation for 7 factors. Thus, for 7 factors it is not a very good model. However, it still has better measures when compared to the PCR model using only 3 factors, and similar measures to the MLR model with 1 factor.

**Table 4.1:** Comparison of the regression measures for the three models with cylinder pressure as response variable

<b>Response: Cylinder Pressure</b>	<b>Model 1</b>	<b>Model 2</b>	<b>Model 3</b>
<b>Regression measures</b>	<b>PCR</b>	<b>PLSR</b>	<b>MLR</b>
<b>Calibration</b>	<b>Predictors: Frequencies</b>	<b>Predictors: Frequencies</b>	<b>Predictor: Engine Speed</b>
Slope	0,3895505	0,7903299	0,3730874
Correlation	0,6241398	0,8890049	0,6108088
R-square	0,3895519	0,7903298	0,3730874
R2 (Pearson)	0,3895505	0,7903298	0,3730875
RMSEC	0,8105046	0,4750063	0,8214483
<b>Validation</b>	<b>Cross validation, 8 segments</b>	<b>Cross validation, 8 segments</b>	<b>Leverage correction</b>
Slope	0,3890022	0,7840710	0,3729926
Correlation	0,6237821	0,8839393	0,6106493
Adjusted R-square	0,3892015	0,7813563	0,3728926
R2(Pearson)	0,3891041	0,7813486	0,3728926
RMSECV	0,8108446	0,4850848	0,8215321 (RMSEE)

As described in the theory chapter, PLSR is better at modelling noisy data, which is why it might have gotten a better result than PCR. The vibration measurements are

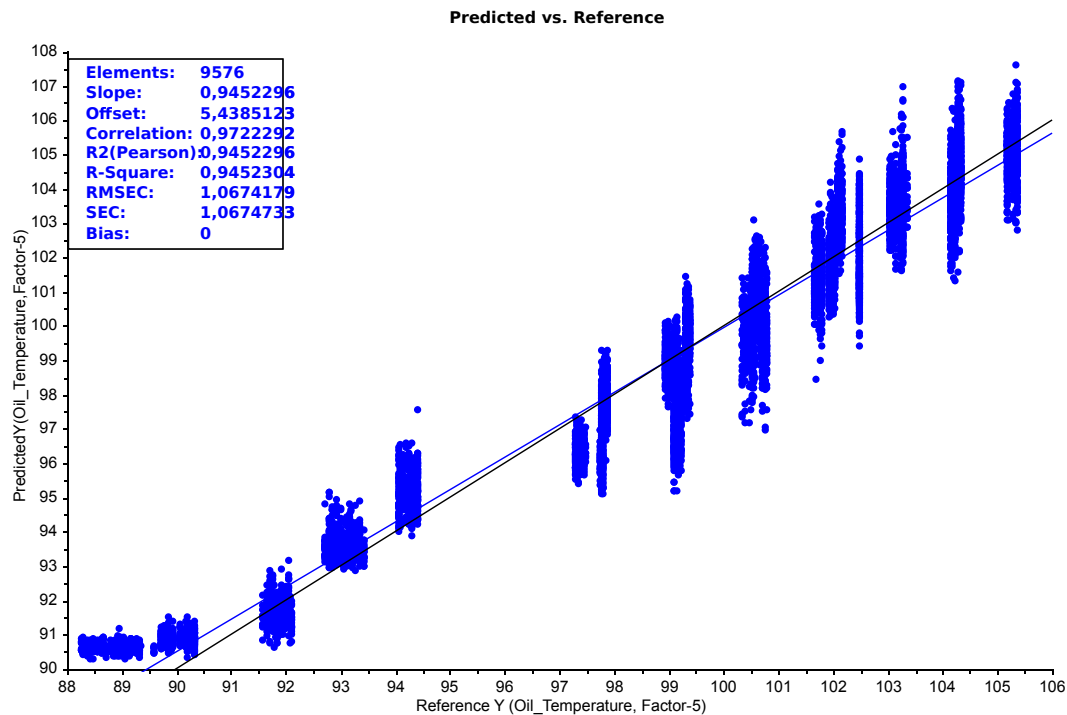
likely to be affected by noise as they were obtained without the use of an amplifier, and because the vibration sensors really were made for an entirely different application (guitar pickups). A PLSR model was also made where the use of overlapping frames in the frequency domain transformation was removed. Improvements in the order of 0,01 was seen for most of the regression measures, however at the same number of factors. The idea was that overlap between successive frames (which range over entire engine cycles) was not ideal for the prediction of the cylinder pressure, however as seen the improvements were so minimal that the effect was negligible. Furthermore, the effect of using the weighting  $1/\text{standard deviation}$  was investigated, and the best regression models were obtained with this weighting applied, the difference in most of the regression measures being about 0,04 for an equal number of factors. Finally, the effect of using two microphones instead of one was investigated. Using only the data from one microphone led to a worse regression model, however the differences were not as large as one would have imagined, as the number of predictor variables are doubled when using the data from two microphones.

All in all the regression model(s) for the cylinder pressure are not good enough to be used as replacements, for instance for the pressure sensor in the lab. As the work and research performed in the lab requires accurate measurements, the prediction error will be too large. However, it is likely that the PLSR model, which was the best of the three models, could be used to discover faults in compression in the cylinders of a car and cylinder misfire. Moreover, it is important to remember that the pre-processing of the cylinder pressure data, placement of microphones, and drift in pressure sensor can have affected the measurements negatively. As pre-processing, the mean of the cylinder pressures have been taken over each engine cycle; this might not be beneficial and can have influenced the data in a bad way. Furthermore, the pressure sensor was placed in cylinder #5, whereas the microphones were placed at a distance from this cylinder. Ideally, the microphones should have been placed at the cylinder head, or at least directly on the engine block. Finally, thermal drift in the pressure sensor might have influenced the measurements in a way that has made them non-representative of the different operating conditions.

### 4.3.2 Oil Temperature

In this section regression models for the oil temperature in the engine will be presented and analysed. The principle is the same as in the previous section; vibration data in time-frequency domain have been used as predictors and oil temperature is the response variable. Furthermore, the regression models with frequencies as predictors will be compared to a model where engine speed has been used as predictor. Oil temperature

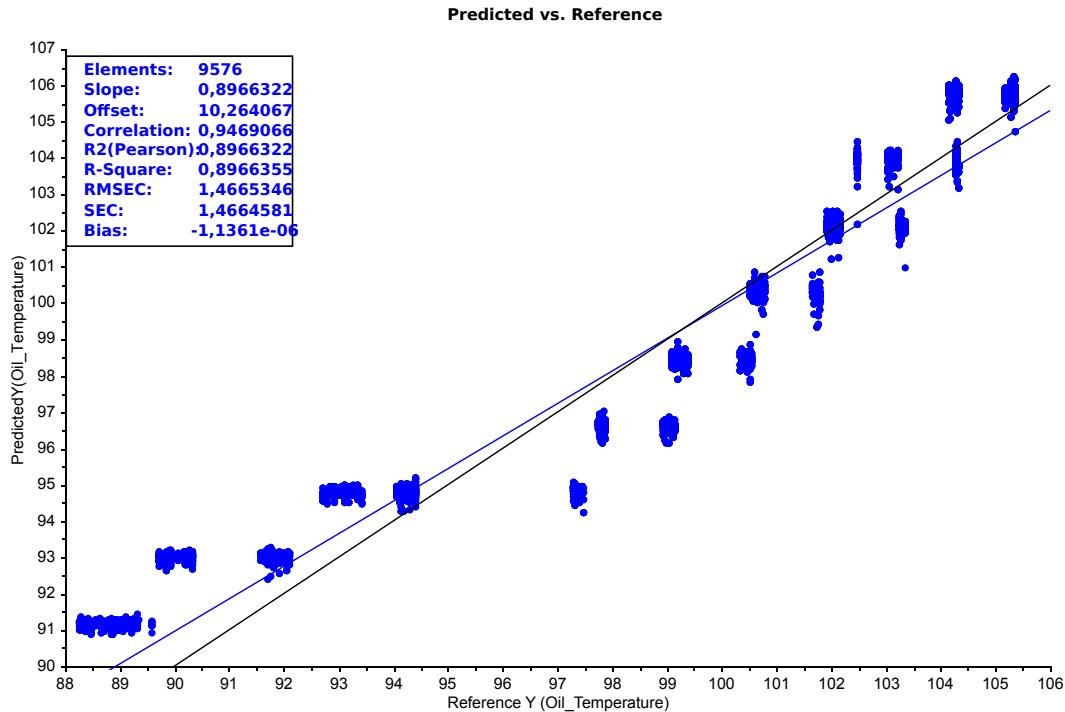
is a common variable which is being measured in most cars. However, if values of the oil temperature can be predicted by the use of vibrations it shows the versatility of the vibration measurements. Figure 4.28 shows the predicted versus reference values of a PLSR model for oil temperature where frequencies have been used as the predictors.



**Figure 4.28:** Predicted versus reference oil temperature values. Vibration data in the time-frequency domain has been used as predictors in a PLSR model.

The figure shows that there is a connection between the predictors and the responses; the samples are not too spread, and the regression line and the target line only have small deviations. Ideally, all the samples should lie clustered on the regression line, but as was the case for the cylinder pressure as well, the samples have some dispersion vertically around the regression line. When investigating the samples, the trend seems to be that the oil temperature increase for increasing rpm and load, and that the dispersion around the regression line increases at the higher temperatures. The temperature seems to increase gradually up until 94,5°C at 1400rpm and 50Nm, whereafter a sudden "jump" to 97,5°C appears for 1400rpm and 90Nm. As for the regression measures, these can be said to be good; the slope and R-squared are approximately 0,94, and the RMSEC is about 1,07. In the range of 88 – 106°C this gives an error of about 6%. It is important to remember that this is for 5 factors, but when compared to the PLSR model of the cylinder pressure which had 7 factors, this can be said to be a better model. Furthermore, the validation measures, which are given in Table 4.2, shows that the deviation between calibration and validation

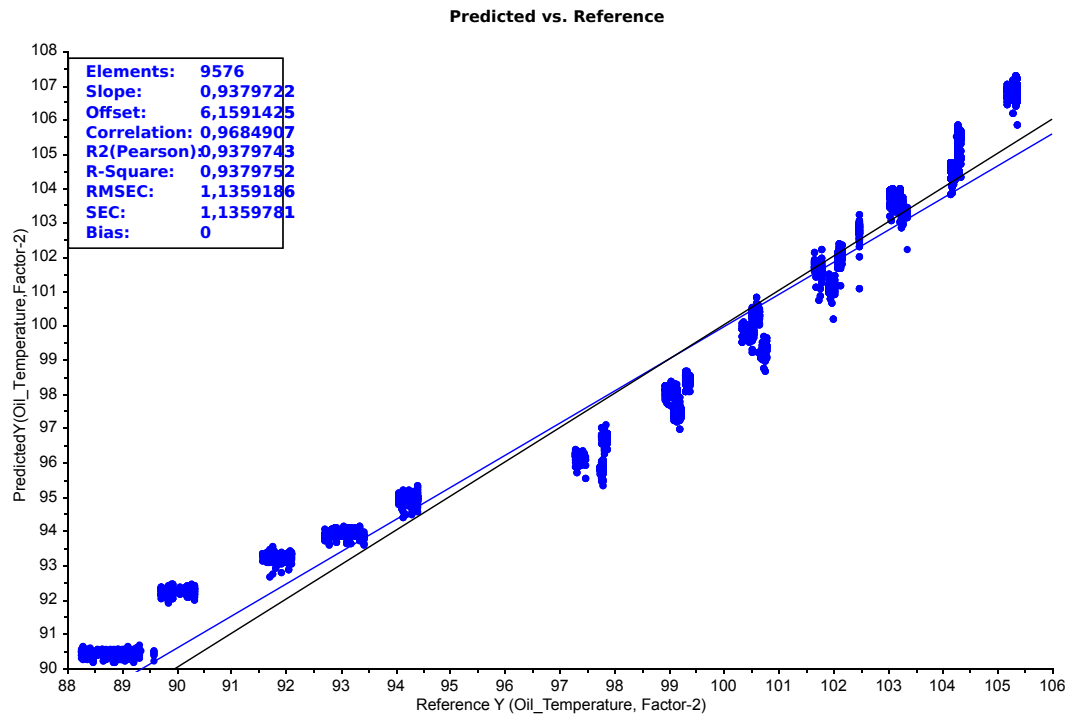
is minimal. Thus, the model does not suffer from overfitting. Figure 4.29 shows another regression model. In this figure engine speed has been used as the predictor variable in a MLR model of oil temperature.



**Figure 4.29:** Predicted versus reference oil temperature values. Engine speed data has been used as predictor in a MLR model.

Once again the samples have been placed in clusters; in the MLR model of the cylinder pressure (Figure 4.27 each of the clusters corresponded to one of the operating conditions of the 24 tests. This is not the case here as overlap is seen for a few of the operating conditions, mostly the 20Nm and 50Nm conditions of rpms from 1600-2600. As for the first oil temperature regression model, the temperature increases for increased load and rpm. This is of course decided by the data and not the regression model, but shows the connection between the operating conditions and the temperature. Earlier in this chapter it was found that vibrations increase with increasing load and rpm; now it seems that the same is true for oil temperature. Thus, the large correlation between the predictors and responses in these regression models is probably related to this. The regression measures for the MLR model are shown in the figure, but they are also given in Table 4.2 for easier comparison with validation measures and the two other regression models. When compared to the PLSR model where the frequencies were used as predictors, the MLR model has lower values for both slope, correlation, R-squared and higher values for RMSEEE. This indicates that the PLSR model with vibrations is a better model than

the MLR model with engine speed; however, the PLSR model uses 5 factors to get the regression measures shown here. At 3 factors, the regression measures of the PLSR model are surpassed by those of the MLR model.



**Figure 4.30:** Predicted versus reference oil temperature values. Engine speed and torque data has been used as predictors in a PLSR model.

Figure 4.30 shows a third regression model for the oil temperature; here both engine speed and engine torque has been used as predictors. This model shows similar measures to those of the first PLSR model, although slope, correlation and R-square have slightly lower values and RMSEC is a bit higher. The model gets these regression measures with only 2 factors, however here two different variables have been used as predictors. The engine speed is a very common measurement when working with engines; engine torque however, is maybe less so, at least for the typical car. In most laboratories and research facilities, engine torque is likely to be a common measurement. The point to be made is that vibration data in time-frequency domain gives an equally good regression model as when engine speed and engine torque is being used (albeit with more factors). This in itself shows that the vibration measurements have a lot of potential.

The regression models have shown that there is great correlation between vibration data and oil temperature, and engine speed (and load) and oil temperature. As opposed to the cylinder pressure PLSR model, the vibration data provided a better model here with a lower number of factors. In a car, the measurements for oil temperature does not

**Table 4.2:** Comparison of the regression measures for the the models with oil temperature as response variable

<b>Response: Oil Temperature</b>	<b>Model 1</b>	<b>Model 2</b>	<b>Model 3</b>
<b>Regression measures</b>	<b>PLSR</b>	<b>MLR</b>	<b>PLSR</b>
<b>Calibration</b>	<b>Predictors: Frequencies</b>	<b>Predictors: Engine Speed</b>	<b>Predictors: Engine Speed Engine Load</b>
Slope	0,9452296	0,8966322	0,9379722
Correlation	0,9722292	0,9469066	0,9684907
R2(Pearson)	0,9452296	0,8966322	0,9379743
R-square	0,9452304	0,8966355	0,9379752
RMSEC	1,067418	1,466535	1,135919
<b>Validation</b>	<b>Cross validation, 8 segments</b>	<b>Leverage correction</b>	<b>Cross validation, 8 segments</b>
Slope	0,9459367	0,8966156	0,9379626
Correlation	0,9715765	0,9468855	0,9684788
R2(Pearson)	0,9439609	0,8965921	0,9379512
Adjusted R-square	0,9439676	0,8965845	0,9379745
RMSECV	1,079756	1,466816 RMSEE	1,136130

necessarily have to be very exact, and the PLSR model with vibration data provided a root mean square error for cross validation of approximately 1,08, which means that new samples (on the same scale) are likely to have a prediction error of a bit more than 1°C. This is not much in the case of measurements in a car, but of course might not be exact enough for a laboratory environment. One should keep in mind that in an engine, temperatures are likely to increase as the period of operation increases. For the tests and data acquisition performed here, the tests with the lower rpms were carried out first, and then the rpm was gradually increased to log data for each operating condition. Thus, at the tests with the higher rpms, the engine has been run for a longer period of time than at the tests with lower rpm. This might have influenced the oil temperature and made the measurements less representative of their operating condition. However, the engine was run for 15minutes prior to the start of data logging to prevent or reduce the effect this could have on the measurements.

### 4.3.3 Other variables

This section investigates regression models of some other variables obtained from the engine measurement system. Analysis of the models will not be as thorough as for cylinder pressure and oil temperature; instead, regression measures for different responses are gathered in a large table for easy comparison. The predictors are vibration data in the time-frequency domain. Some models will be outlined and the predicted versus reference plot for these models will be given in Appendix B. The responses are essentially the variables discussed in section 4.2.3, i.e those that appeared in the loadings for the third PCA model.

**Table 4.3:** Regression measures for several different measurement variables related to engine operation. Calibration values are given in blue and validation values are given in red for each model. Validation has been done by random cross-validation with 8 segments.

Variables		Regression Measures				Number of factors
Responses	Prediktors	Slope	Correlation	R-squared	RMSE	
Weight [g]	Frequencies	0,9568371	0,9781804	0,9568366	170,0071	2
		0,8565927	0,9781305	0,9567481	170,1993	
Fuel consumption return total [l]	Frequencies	0,9746484	0,9872428	0,9746484	2,815842	4
		0,9749666	0,9869714	0,9741167	2,845924	
Spill water temperature [°C]	Frequencies	0,9683817	0,9840639	0,9683812	0,4416076	5
		0,9677584	0,9828694	0,9660458	0,4577434	
Lambda estimated ECU [-]	Frequencies	0,8939310	0,9454793	0,8939296	0,2050000	7
		0,8864679	0,9428360	0,8889420	0,2097853	
Engine Speed [rpm]	Frequencies	0,9599580	0,9797744	0,9599576	93,89131	5
		0,9600499	0,9791237	0,9586924	95,37666	
Engine torque [Nm]	Frequencies	0,9210295	0,9597028	0,9210292	7,890084	7
		0,9111322	0,9577693	0,9172873	8,075238	
Fuel consumption ECU [l/h]	Frequencies	0,9664701	0,9830921	0,9664699	0,2680566	5
		0,9637158	0,9822365	0,9647993	0,2747009	
Intake temperature after cooler [°C]	Frequencies	0,9537267	0,9765893	0,9537267	0,4837511	2
		0,9534999	0,9765552	0,9536645	0,4840999	
Begin of main injection [deg]	Frequencies	0,9521710	0,9757925	0,9521722	0,3900680	6
		0,9524339	0,9749004	0,9504398	0,4001120	
Fuel pressure ECU [MPaG]	Frequencies	0,7901476	0,8889025	0,7901360	3,218397	7
		0,7825182	0,8799335	0,7742751	3,338657	
NOx [ppm]	Frequencies	0,3498936	0,5915180	0,3498913	0,007793400	7
		0,3350571	0,5673864	0,321730	0,007962400	

From Table 4.3 it can be seen that many models have good regression measures, but that some require more factors than others to obtain these measures. The two models which stands out the most are the models with "Weight" and "Intake temperature after

cooler" as response variables. These models obtain equally good measures as many of the other models with only two factors. These were also among the most prominent peaks for the loadings of the first principal component of the third PCA model in section 4.2.3. The "Weight" model has a RMSECV of 170,1993; in the range of 12000-15000g (which spans the measurements for the samples) this corresponds to an estimated prediction error of 5,7%, approximately. The predicted vs reference values of the Weight-model is shown in Figure B.1 in Appendix B. It can further be seen from table that a specific exhaust gas is hard to predict: the regression model for NO<sub>x</sub>-measurements is not good, even with 7 factors. But as can be seen from the weight model, the total amount of exhaust gases has a good correlation with vibrations. It is possible that an increased amount of exhaust gases (larger weight) will lead to more vibrations in the exhaust gas pipes, thus making vibrations a good indicator of the amount of exhaust gas. However, when looking back to the discussion made on weight and lambda estimated in section 4.2.3, where it was found that lambda estimated and weight were both higher at the lower rpms, it seems to be the other way around. As the weight was found to be higher at lower rpms whereas vibrations were found to increase with rpm and load, it should mean that vibrations and exhaust gas weights are negatively correlated. It could be that the increased weight of the exhaust gases leads to increased damping in the affected systems, thus less vibration.

The model with the response variable "Intake temperature after cooler" also provided some really good regression measures. Since one of the microphones is placed directly to the intake manifold, it might be that it has managed to detect small changes in the air temperature. Figure B.2 in Appendix B shows the predicted versus reference values for the intake temperature after cooler model. As for all temperature measurements, one should be careful to draw conclusions too fast. It can be seen that the intake temperature after cooler increase with increasing rpm and load. In section 4.2.3 it was seen that "Room temperature" appeared in the loadings of the first principal component. The room temperature can be seen to increase over the time span of the different tests; this is mainly due to the engine operation which produces heat through the combustion process. If the air intake to the engine is in the room, and not through a pipe to the outside, it is clear that the temperature of the intake air will increase as the room temperature increases. This should be kept in mind when looking at the result. Another interesting model is the one where "Fuel consumption return total" is the response variable; as discussed earlier in the report this variable, or the function of the fuel system which leads to return of fuel might be the reason for the frequencies discussed in section 4.1.2 and 2.3.1. As can be seen here it has a very high correlation with the vibration data with 4 factors. Finally, the engine-torque model, although with many factors and not the best regression measures, shows that it could be possible to predict the load of an engine based on vibrations.



This could be useful in many contexts, where torque measurements are unavailable or expensive. In this project only three different load conditions have been used, and to further develop the torque model it would be necessary to have a much wider range of load conditions. The predicted versus reference values of the engine-torque model can be seen in Figure B.3 in Appendix B.

To end the results and discussion chapter of this report, some final remarks will be made. Regression has been performed with vibration data in time-frequency domain obtained from two contact microphones at two different locations. The regression models have shown a lot of potential, at least for some of the variables that have been predicted. The cylinder pressure-model and oil temperature-model was compared to models where engine speed was used as the predictor variable. The point was to see if an equally good or better model could be obtained from engine speed only, thus making the use of vibrations unnecessary as rpm measurements are very common in all engines. Table 4.3 shows that there is a lot of potential in the vibration measurements, but some important points must be mentioned. For all the regression models here, only cross validation has been performed, giving the root mean square error of cross validation. This gives a better measure on how the models will perform on future data than the root mean square error of calibration, but the best estimation of how a model will perform on future samples is given from the root mean square error of prediction, which requires a separate test set. Test-set validation have not been employed in this project, and would give a further indication on the potential of vibration data. Also, the vibration sensors (contact microphones) and all other measurement equipment are prone to errors, which might have affected the results obtained here.



# Chapter 5

## Conclusion

This chapter presents a conclusion for the work that has been carried out in this project. The main results will be summarised with respect to the objectives stated in section 1.3 in the introductory chapter.

The purpose of this project has been to investigate the application of multivariate analysis methods on vibration data obtained from a diesel engine using cheap acoustic contact microphones. Initially, a literature study on condition monitoring research for internal combustion engines was carried out. Both traditional frequency-domain analysis and multivariate analysis methods were studied. The literature study gave an overview of the research that has been done in the field, and provided inspiration for this project. Condition monitoring research has seen an increase in the later years, and to get an overview of all the research in the field is a task that exceeds the objectives of this project. However, a thorough literature review was carried out, with the results being put in context throughout the project. The main parts of the literature studied is woven into the introduction chapter to set the background for this project. Furthermore, a short review on the working principles of the internal combustion engine was done in order to better understand the connections between the measurements.

Data acquisition was carried out according to plan in the Internal Combustion Engine Laboratory at the Norwegian University of Science and Technology. More specifically, data was obtained from a fully instrumented Mercedes-Benz OM613 diesel engine. The instrumentation and test rig was set up prior to this project, and only small changes were made to the existing setup. Of the changes worth mentioning was the inclusion of the contact microphones to the existing instrumentation system, and the changes made in the LabView software for data logging. In addition, the connectors on the microphones were changed from 6,35mm female stereo jacks to BNC plugs to reduce the influence of noise, and to make them fit the existing data acquisition hardware. This work was carried out by laboratory technicians the Department of Energy and Process Engineering. When

preparing the plan for the measurements, emphasis was put on getting a representative range of engine speeds. Measurements were obtained in the range 1000-2600rpm, which can be said to be describing of the typical operating range of a diesel engine for passenger cars. Three different load conditions were applied at each rpm (except at 1000rpm and 1200rpm), namely 20Nm, 50Nm and 90Nm. In hindsight, this is less representative of the typical operation of a diesel engine, as it can produce as much as 470Nm at max.

Two problems were encountered in the data acquisition process; microphone placement and the need for an amplifier. As the microphones are intended for guitars, they were not fit for the high temperatures that take place on the engine block. Thus, new placements where the temperature was adequately low had to be found. Furthermore, the initial impression was that an amplifier had to be used to get usable vibration measurements from the contact microphones. However, preliminary tests showed that the measurements were representative, and it was decided to move on with the project without an amplifier. Moreover, for the purpose of condition monitoring, it would be beneficial to obtain data over a longer time period. But due to the limitations in storage, the data acquisition system in its whole and the fact that the lab was used for several projects simultaneously, only short time intervals of operation was logged. However, the amount of data acquired was deemed sufficient for the objectives of this project.

For the experimental study, a superficial analysis was initially carried out in the time domain. It was shown that vibration amplitude increases with increased rpm and load, and that this matches results from other research papers. Furthermore, analysis was carried out in the time-frequency domain. Resampling of the vibration data revealed some frequencies that appeared as "static", i.e the frequencies were apparent for all rpms. The interesting part of these frequencies was that they increased in magnitude for increasing rpm. The frequencies were found at approximately 1837Hz, 1250Hz and 2425Hz. An attempt was made to connect these frequencies to engine events. A suggestion was that the common rail fuel system, with pressure sensor and fuel regulator could be connected to a static frequency, but with magnitude increasing with engine speed. This theory was further substantiated through that the variable "Fuel consumption return total" appeared in the loadings of the first principal component of the PCA model where all measurements were included.

Implementation of the weighting function  $1/\text{standard deviation}$  revealed a resonance-like structure in the loadings of the frequencies which was not visible prior to the weighting. The period of the structure was found to be 3425Hz on average. Multiplication of the period with the speed of sound in cast iron resulted in a distance of 1,34m, which is possible that represents the length of the engine block in the laboratory. In addition to the PCA models, several regression models were made for different response variables,

with vibration data in the time-frequency domain as predictors. The models showed good regression measures for many different response variables, and the best results were found for the variables "Weight", "Intake temperature after cooler" and "Fuel consumption return total". It was further found that exhaust gas weight and vibrations were negatively correlated. Models for cylinder pressure and oil temperature were compared to models where engine speed was used as the predictor variable. An estimated prediction error of 12% with 7 factors was found for the cylinder pressure model, whereas the estimated prediction error for oil temperature was approximately 6% with 5 factors. For the cylinder pressure model, vibration data gave a better model than engine speed, however the prediction error is too large for vibration sensors to replace pressure sensors in for instance the laboratory. For the oil temperature model, vibration data gave a better model when used as predictors than when engine speed and torque was used as the predictors, however with more factors. All in all the vibration models showed promising results, but further research must be done before vibration sensors can replace other sensors.

## 5.1 Recommendations for Further Work

There are many things that can be done to further investigate the vibration data of diesel engines. The recommendations are divided into three subsections, where the first is related to measurement equipment, the second is related to the measurements and operating conditions of the engine, and the third is related to the processing of the data.

### Data acquisition - Equipment

For further work, it would be interesting to use better microphones, or vibration sensors which are more suited to the application, like accelerometers. This could make it possible to place the vibration sensors directly on the engine block, which could produce stronger vibrations, or even different vibrations than those obtained in this project. Furthermore, an amplifier could be implemented as part of the data acquisition hardware to further improve the quality of the measurement and reduce the influence of noise and potentially the loss of information. In addition, using several vibration sensors at different locations could lead to the discovery of new vibrations which can be related to specific events in the engine. It was seen in this project that using two microphones instead of one gave better results in the regression models.

## Data acquisition - Measurements

The range of rpm measurements in this project was quite representative of the operation of a diesel engine for passenger cars, however the range of loads were not. Therefore, new measurements done at several different load conditions could reveal more information, and make the reference model more representative of normal operation. For instance, load conditions from 0Nm to 150Nm or more could be applied to the engine at speeds where it is feasible. The step between load conditions could for example be 15Nm. Furthermore, as the software and ECU at the lab makes it possible to adjust the injection timing in the cylinders and the activation level of the turbocharger, it could be interesting to see the effects this would have on vibrations. The ECU and Modus software makes it possible to completely close off the injection of fuel to one cylinder; thus simulating misfiring. Recording vibrations from this operating condition, and then projecting these samples onto the reference model could be very interesting in a condition monitoring perspective. If the samples of misfiring are clearly separated from the other samples in the reference model, this is promising towards a CM system.

## Data Processing

In this project, the vibration measurements have been transformed into the time-frequency domain using a windowed Fourier transform. Moreover, frame lengths of 4096samples which is the same as the number of samples in an engine cycle have been used in the transformation. It might be that other frame lengths are better suited for the vibrations, thus trying various frame lengths could lead to different results. In addition, the processing of the high speed measurements, where the mean was taken over the engine cycles, might not be the best way to process the data. It was necessary in order to have an equal number of measurements as frames from the time-frequency transformation, but it is likely that there are other solutions that will provide a better result. Finally, only cross validation has been used for the regression models, and to further evaluate these models data should be collected which could be used as separate test sets. This would give a better indication of the prediction error the models will have for new samples. A different approach could be to make new models of the existing data, but with some of the samples being kept out as a separate test set. This would be the next step in the process of investigating if vibrations could give exact enough predictions for vibration sensors to act as replacements for other sensors.

## 5.2 Closing Remarks

This project has been truly experimental, and what would be the outcome of it was not known at the start. PCA has been used in an exploratory manner, and exactly what would be found was left for surprise. Thus, it has also been challenging to know what to look for, and to tie the findings to actual engine events or components. However it has been shown that there is information to be found in the vibration data, and that vibrations have a lot of potential towards prediction of other variables. When it comes to the world of multivariate analysis, this project has probably only touched the tip of the iceberg of different possibilities, and it is likely a lot more information to be found in the vibration data of diesel engines.





# Bibliography

- [1] J. S. Rao, *Vibratory Condition Monitoring of Machines*. CRC Press/Narosa Pub. House, 2000.
- [2] R. B. Randall, 'Introduction and Background', in *Vibration-based Condition Monitoring: Industrial, Aerospace and Automotive Applications*, John Wiley & Sons, Ltd, 2011, ch. 1, pp. 1–23.
- [3] M. Entezami, E. Stewart, J. Tutcher, W. Driscoll, R. Ellis, G. Yeo, Z. Zhang, C. Roberts, T. Kono and S. Bayram, 'Acoustic analysis techniques for condition monitoring of roller bearings', *6th IET Conference on Railway Condition Monitoring (RCM 2014)*, pp. 1–8, 2014.
- [4] R. B. Randall, *Frequency Analysis*. Brüel & Kjær Primers and Handbooks, 1987.
- [5] —, 'Vibration Signals from Rotating and Reciprocating Machines', in *Vibration-based Condition Monitoring: Industrial, Automotive and Aerospace Applications*, John Wiley & Sons, Ltd, 2011, ch. 2, pp. 25–61.
- [6] Y. Alhouli B, A. Abed, A. Alkhaledi, A. Alzayedi and M. Alardhi, 'Study of Diesel Engine Vibration Condition Monitoring', *Global Journal of Researches in Engineering*, vol. 15, no. 6, 2015.
- [7] A. Albarbar, F. Gu and A. D. Ball, 'Diesel engine fuel injection monitoring using acoustic measurements and independent component analysis', *Measurement: Journal of the International Measurement Confederation*, vol. 43, no. 10, pp. 1376–1386, 2010.
- [8] F. Gu, W. Li, A. D. Ball and A. Y. T. Leung, 'The Condition Monitoring of Diesel Engines Using Acoustic Measurements Part 1 : Acoustic Characteristics of the Engine and Representation of the Acoustic Signals', *SAE TECHNICAL*, 2000.
- [9] S. K. Yadav, K. Tyagi, B. Shah and P. K. Kalra, 'Audio signature-based condition monitoring of internal combustion engine using FFT and correlation approach',

- IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 4, pp. 1217–1226, 2011.
- [10] J. Antoni, J. Daniere and F. Guillet, ‘Effective Vibration Analysis Of IC Engines Using Cyclostationarity. Part 1: A Methodology For Condition Monitoring’, *Journal of Sound and Vibration*, vol. 257, no. 5, pp. 815–837, 2002.
- [11] K. H. Esbensen, D. Guyot, F. Westad and L. P. Houmoller, *Multivariate Data Analysis - in Practice: An Introduction to Multivariate Data Analysis and Experimental Design*, 5th ed. CAMO, 2010.
- [12] W. Li, F. Gu, A. D. Ball, A. Y. T. Leung and C. E. Phipps, ‘A Study of the Noise From Diesel Engines Using the Independent Component Analysis’, *Mechanical Systems and Signal Processing*, vol. 15, no. March 2016, pp. 1165–1184, 2001.
- [13] Z. Li, X. Yan, Z. Guo, Y. Zhang, C. Yuan and Z. Peng, ‘Condition Monitoring and Fault Diagnosis for Marine Diesel Engines using Information Fusion Techniques’, *Electronics and Electrical Engineering*, vol. 123, no. 7, pp. 1392–1215, 2012.
- [14] K. Jafarian, ‘Vibration Analysis for Fault Detection of Automobile Engine Using PCA Technique’, in *4th International Conference on Control , Instrumentation and Automation (ICCIA)*, 2016, pp. 372–376.
- [15] N. Baydar, Q. Chen, A. Ball and U. Kruger, ‘Detection of incipient tooth defect in helical gears using multivariate statistics’, *Mechanical Systems and Signal Processing*, vol. 15, no. 2, pp. 303–321, 2001.
- [16] R. Zimroz and A. Bartkowiak, ‘Investigation on spectral structure of gearbox vibration signals by principal component analysis for condition monitoring purposes’, *Journal of Physics: Conference Series*, vol. 305, no. 1, 2011.
- [17] T. Plante, L. Stanley, A. Nejadpak and C. X. Yang, ‘Rotating machine fault detection using principal component analysis of vibration signal’, in *2016 IEEE AUTOTEST-CON*, IEEE, Sep. 2016.
- [18] A. D. Nembhard, J. K. Sinha, A. J. Pinkerton and K. Elbhah, ‘Combined vibration and thermal analysis for the condition monitoring of rotating machinery’, *Structural Health Monitoring*, vol. 13, no. 3, pp. 281–295, May 2014.
- [19] J. Abraham and A. R. Bhende, ‘IC Engine Fault Diagnosis Using Vibration & Acoustic Signals – A Review’, *International Journal on Recent and innovation Trends in Computing and Communication*, vol. 3, no. 2, pp. 129–132, 2015.

- [20] S. Yadav and P. Kalra, 'Automatic fault diagnosis of internal combustion engine based on spectrogram and artificial neural network', in *Proceedings of the 10th WSEAS Int. Conference on ROBOTICS, CONTROL and MANUFACTURING TECHNOLOGY*, 2010.
- [21] G. O. Chandroth, A. J. C. Sharkey and N. E. Sharkey, 'Cylinder pressures and vibration in internal combustion engine condition monitoring', in *Proceedings Comadem 99*, 1999, pp. 141–151.
- [22] A. R. MOHANTY, *MACHINERY CONDITION MONITORING : principles and practices*. CRC PRESS, 2017.
- [23] A. K. Jardine, D. Lin and D. Banjevic, 'A review on machinery diagnostics and prognostics implementing condition-based maintenance', *Mechanical Systems and Signal Processing*, vol. 20, no. 7, pp. 1483–1510, 2005.
- [24] R. Jiang and X. Yan, 'Condition Monitoring of Diesel Engines', in *Complex System Maintenance Handbook*, Springer London, 2008, pp. 533–557.
- [25] International Organization for Standardization 2954, *Mechanical vibration of rotating and reciprocating machinery - Requirements for instruments for measuring vibration severity*. 2012.
- [26] International Organization for Standardization 13373-1, *Condition monitoring and diagnostics of machines - Vibration condition monitoring - Part 1: General procedures*, 2002.
- [27] J. T. Broch, *Mechanical Vibration and Shock Measurements*. Brüel & Kjær, 1984, pp. 1–436.
- [28] International Organization for Standardization 13373-2, *Condition monitoring and diagnostics of machines — Vibration condition monitoring - Part 2: Processing, Analysis and presentation of vibration data*, 2016.
- [29] International Organization for Standardization 20816-1, *Mechanical vibration - Measurements and evaluation of machine vibration - Part 1: General guidelines*, 2016.
- [30] International Organization for Standardization 10816-6, *Mechanical vibration - Evaluation of machine vibration by measurements of non-rotating parts - Part 6: Reciprocating machines with power ratings above 100kW*, 1995.
- [31] S. W. Smith, *Digital signal processing*. California Technical Publishing, 1999.

- [32] M. Müller, *Fundamentals of Music Processing*. Springer International Publishing, 2015.
- [33] Phonical, *-FFT-Time-Frequency-View.png*, 2017. [Online]. Available: <https://commons.wikimedia.org/wiki/File:FFT-Time-Frequency-View.png>.
- [34] Creative Commons, *Attribution-ShareAlike 4.0 International*. [Online]. Available: <https://creativecommons.org/licenses/by-sa/4.0/legalcode>.
- [35] K. Shin and J. K. Hammond, *Fundamentals of Signal Processing for Sound and Vibration Engineers*. John Wiley & Sons, Ltd, 2008.
- [36] Head Acoustics, *Order analysis in ArtemiS SUITE 1 HEAD acoustics Application Note*, 2016.
- [37] S. Delvecchio, P. Bonfiglio and F. Pompoli, 'Vibro-acoustic condition monitoring of Internal Combustion Engines: A critical review of existing techniques', *Mechanical Systems and Signal Processing*, vol. 99, pp. 661–683, Jan. 2018.
- [38] B. Swarbrick, *Multivariate Data Analysis for Dummies*. 2012, p. 43. [Online]. Available: [www.camo.com](http://www.camo.com).
- [39] Encyclopedia of Mathematics, *Covariance*. [Online]. Available: <https://www.encyclopediaofmath.org/index.php/Covariance>.
- [40] E. W. Weisstein, *Statistical Correlation*. [Online]. Available: <http://mathworld.wolfram.com/StatisticalCorrelation.html>.
- [41] I. Jolliffe, *Principal Component Analysis*. Springer Verlag New York, Inc, 1986.
- [42] CAMO, 'The Unscrambler X v10.3 User Manual', 2014.
- [43] K. Dunn, *Process Improvement Using Data*. 2016.
- [44] J. Jauregui, 'Principal component analysis with linear algebra', pp. 1–10, 2012.
- [45] S. Wold, K. Esbensen and P. Geladi, 'Principal Component Analysis', *Chemometrics and Intelligent Laboratory Systems*, vol. 2, pp. 37–52, 1987.
- [46] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. Springer, 2009.

- [47] F. Badin, 'Internal combustion engines', in *Hybrid Vehicles - From Components to System*, P. Leduc, Y. Touzani and F. Bouvart, Eds., Editions Technip, 2013, ch. 2, pp. 60–113.
- [48] L. Eriksson and L. Nielsen, *MODELING AND CONTROL OF ENGINES*. John Wiley & Sons, Ltd, 2014.
- [49] A. Albarbar, F. Gu, A. D. Ball and A. Starr, 'Acoustic monitoring of engine fuel injection based on adaptive filtering techniques', *Applied Acoustics*, vol. 71, no. 12, pp. 1132–1141, 2010.
- [50] J. Drouet, Q. Leclère and E. Parizet, 'Experimental modeling of Wiener filters estimated on an operating diesel engine', *Mechanical Systems and Signal Processing*, vol. 50-51, pp. 646–658, 2015.
- [51] A. Rousounelos, 'Reduction of sound radiation from automotive-type panels', PhD thesis, Loughborough University, 2010.
- [52] *1999 Mercedes-Benz E320 CDI W210 DE 32 LA*, 2013. [Online]. Available: <https://www.carfolio.com/specifications/models/car/?car=249887>.
- [53] J. Hokanson, *MathWorks File Exchange - TDMS Reader*, 2012. [Online]. Available: <https://se.mathworks.com/matlabcentral/fileexchange/30023-tdms-reader>.
- [54] MathWorks, *Multidimensional Arrays*. [Online]. Available: <https://se.mathworks.com/help/matlab/math/multidimensional-arrays.html>.
- [55] A. D. Ball, F. Gu and W. Li, 'The Condition Monitoring of Diesel Engines Using Acoustic Measurements Part 2 : Fault Detection and Diagnosis', *SAE TECHNICAL*, 2000.



# Appendix A

## Abbreviations

**CM** Condition monitoring

**CBM** Condition based maintenance

**RAMS** Reliability, availability, maintainability, and safety

**RUL** Remaining useful life

**AE** Acoustic Emission

**ANN** Artificial neural network

**MEMS** Microelectromechanical systems

**emf** Electromotive force

**GUI** Graphical user interface

**MVA** Multivariate analysis

**ICA** Independent component analysis

**PCA** Principal component analysis

**PLSR** Partial least squares regression

**MLR** Multiple linear regression

**PCR** Principle component regression

**RMSE** Root mean square error

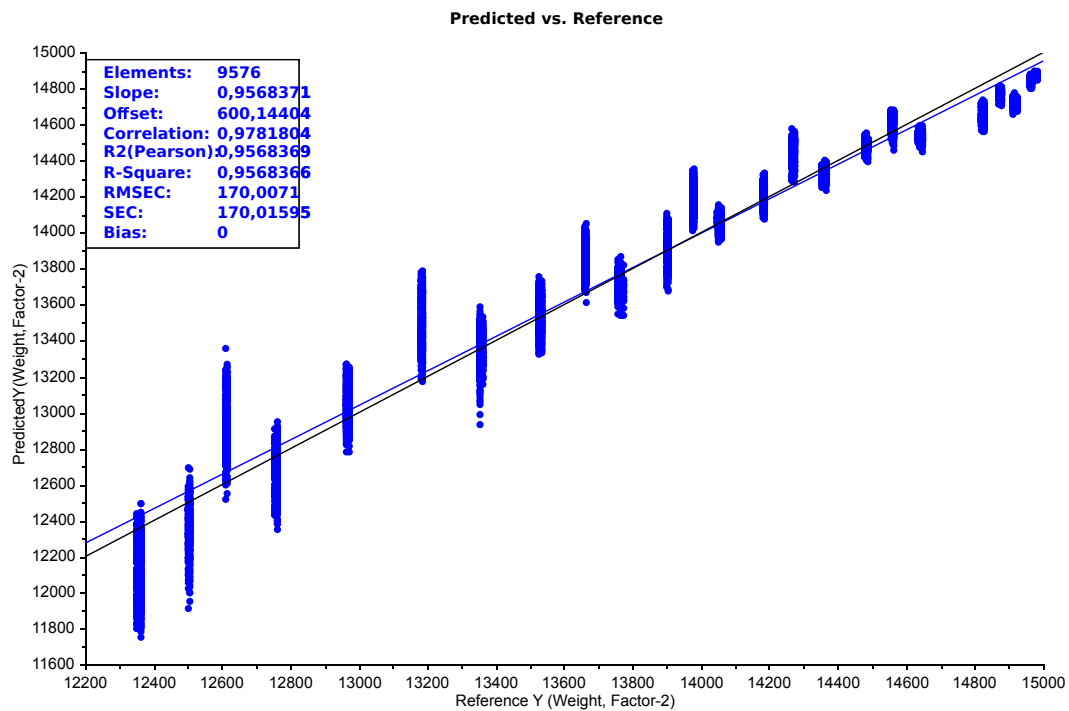
**RMSEC** Root mean square error of calibration

<b>RMSEP</b>	Root mean square error of prediction
<b>RMSECV</b>	Root mean square error of cross validation
<b>RMSEE</b>	Root mean square error of estimation
<b>IC</b>	Internal combustion
<b>ICE</b>	Internal combustion engine
<b>VICE</b>	Vehicle internal combustion engine
<b>TDC</b>	Top dead centre
<b>BDC</b>	Bottom dead centre
<b>OHV</b>	Overhead valve
<b>CAD</b>	Crank angle degrees
<b>ECU</b>	Engine control unit
<b>rpm</b>	Revolutions per minute
<b>AFR</b>	Air-to-fuel ratio
<b>DFT</b>	Discrete Fourier transform
<b>FFT</b>	Fast Fourier transform
<b>STFT</b>	Short-time Fourier transform
<b>CWT</b>	Continuous wavelet transform
<b>HT</b>	Hilbert transform
<b>HHT</b>	Hilbert-Huang transform
<b>SPWVD</b>	Smoothed pseudo Wigner-Ville distribution
<b>BSS</b>	Blind source separation
<b>RMS</b>	Root mean square

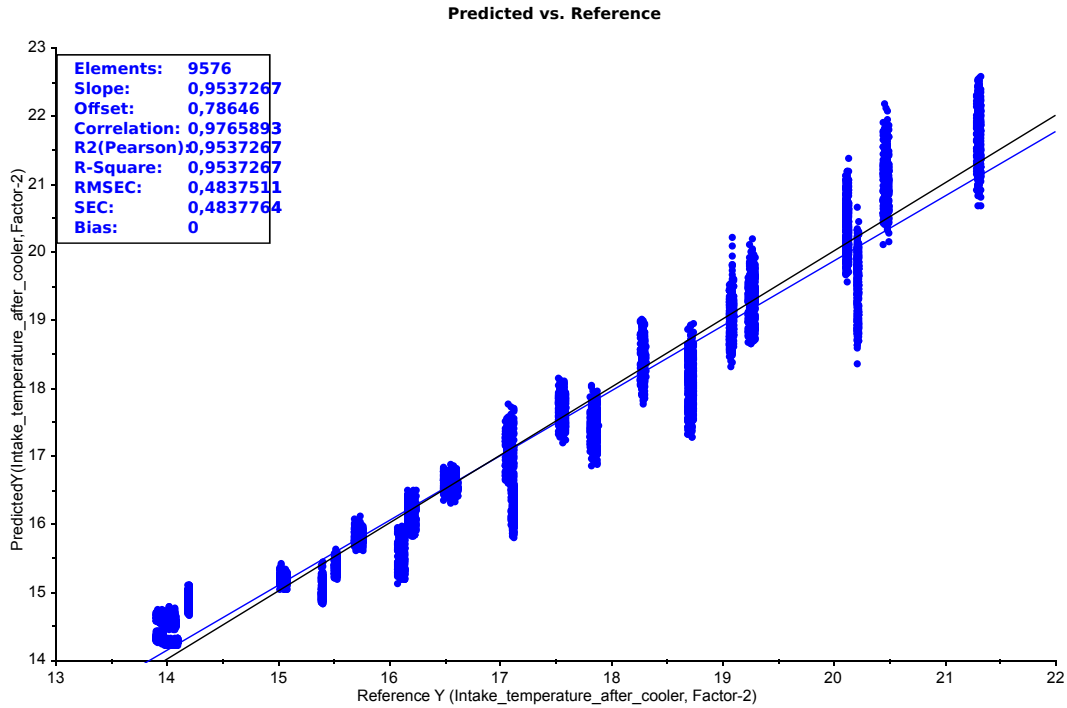


# Appendix B

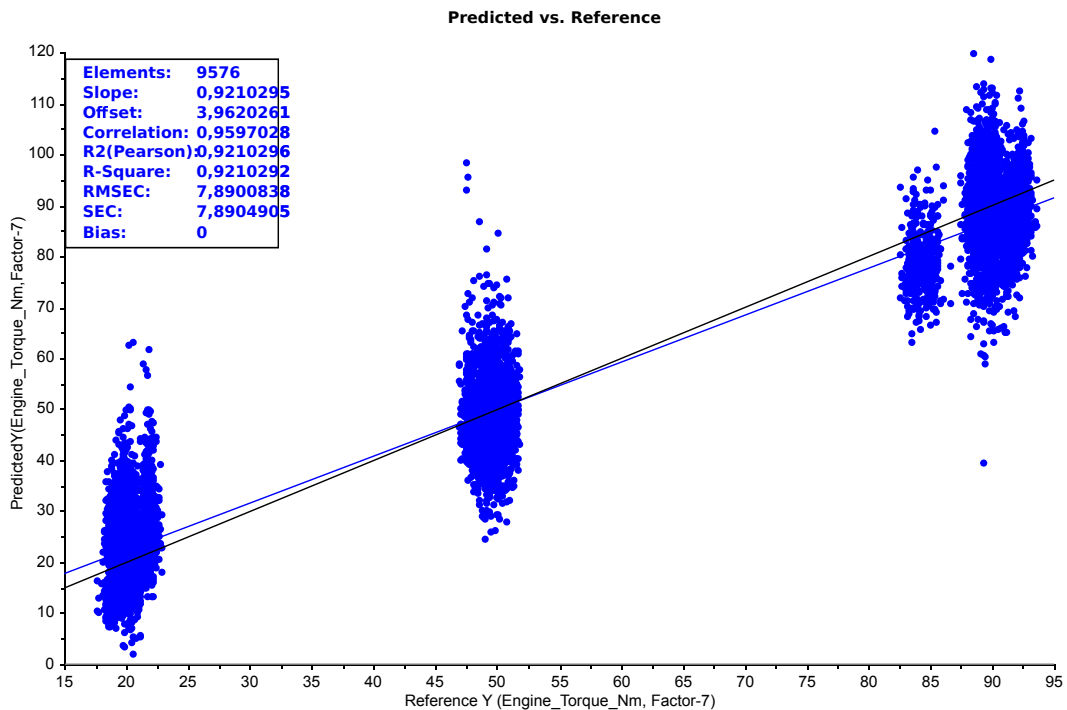
## Regression plots



**Figure B.1:** Predicted versus reference exhaust gas weight values. Frequency vibration data has been used as predictors in a PLSR model. The weight values increase for the lower rpms, i.e the lowest values are found for 2600rpm90Nm and the highest values are found for 1000rpm20Nm



**Figure B.2:** Predicted versus reference intake temperature after cooler values. Frequency vibration data has been used as predictors in a PLSR model. The temperature increase for increasing rpm and load



**Figure B.3:** Predicted versus reference engine torque values. Frequency vibration data has been used as predictors in a PLSR model. The three load conditions are very clear, with some spread.

# Appendix C

## MATLAB Code

### C.1 Data Extraction

This script deals with extraction of data from tdms files and creation of multidimensional arrays. The script has to be placed in the same folder as the MATLAB files provided by the TDMS-Reader package.

```
1
2 %Initializing
3 close all;
4 clear;
5 clc;
6
7 %Number of tests
8 NumTests = 24;
9 %Number of variables in High Speed measurements
10 NumVarHigh = 12;
11 %Number of variables in Low Speed measurements
12 NumVarLow = 60;
13 %Number of samples in one engine cycle
14 CycleLength = 4096;
15 %Low speed sampling rate
16 FsLow = 10;
17
18
19
20
21
22
23
24
```

```

25
26 %Storing filenames in array
27 %Preallocate
28 Filenames=num2cell(zeros(NumTests,1));
29 for i=1:NumTests
30
31     CurFilename = ((strjoin([{'Test'},num2str(i),{'.tdms'}])));
32     CurFilename = CurFilename(CurFilename ~= ' ');
33     Filenames(i) = cellstr(CurFilename);
34
35 end %end for i
36
37 %Reading datafiles using tdms-reader
38 for i=1:NumTests
39
40     Tests(i) = TDMS_getStruct(char(Filenames(i)));
41
42 end
43
44 %Multidimensional array for High Speed measurements
45 %Preallocate size of each test
46 TestSizeHigh= zeros(length(Tests(24).High_speed.Time.data),...
47     NumVarHigh);
48
49 Tests_High = cat(3,TestSizeHigh,TestSizeHigh,TestSizeHigh,...
50     TestSizeHigh,TestSizeHigh,TestSizeHigh,TestSizeHigh,...
51     TestSizeHigh,TestSizeHigh,TestSizeHigh,TestSizeHigh,...
52     TestSizeHigh,TestSizeHigh,TestSizeHigh,TestSizeHigh,...
53     TestSizeHigh,TestSizeHigh,TestSizeHigh,TestSizeHigh,...
54     TestSizeHigh,TestSizeHigh,TestSizeHigh,TestSizeHigh,...
55     TestSizeHigh);
56
57 %Multidimensional array for Low speed measurements
58 %Preallocate size of each test
59 TestSizeLow = zeros(1000,NumVarLow);
60
61 Tests_Low = cat(3, TestSizeLow,TestSizeLow,TestSizeLow,...
62     TestSizeLow,TestSizeLow,TestSizeLow,TestSizeLow,...
63     TestSizeLow,TestSizeLow,TestSizeLow,TestSizeLow,...
64     TestSizeLow,TestSizeLow,TestSizeLow,TestSizeLow,...
65     TestSizeLow,TestSizeLow,TestSizeLow,TestSizeLow,...
66     TestSizeLow,TestSizeLow,TestSizeLow,TestSizeLow,TestSizeLow);
67
68
69 %Create arrays of the variable names

```

```
70 HighSpeedVar = struct2table(Tests(1).High_speed, 'AsArray',true);
71 HighSpeedVar = HighSpeedVar.Properties.VariableNames;
72
73 LowSpeedVar = struct2table(Tests(1).Low_speed, 'AsArray',true);
74 LowSpeedVar = LowSpeedVar.Properties.VariableNames;
75
76 %%Data extraction
77 for i=1:NumTests
78
79     %Get Current Test
80     CurTest = Tests(i);
81
82     %Extract High speed measurements
83     for j=1:NumVarHigh
84
85         %Index out each variable, starting from 'Time'.
86         CurVar_H = char(HighSpeedVar(j+2));
87
88         %Extract data for current variable, and transpose
89         %to get data in rows
90         CurVarData_H = (CurTest.High_speed.(char(CurVar_H)).data)';
91
92         %Store the extracted data in the multidimensional array
93         Tests_High(1:length(CurVarData_H),j,i) = CurVarData_H;
94
95     end %end for j:NumVarHigh
96
97     %Extract Low speed measurements
98     for k=1:NumVarLow
99
100        %Index out each variable starting from 'Time';
101        CurVar_L = char(LowSpeedVar(k+2));
102
103        %Extract data for current variable, and transpose to
104        %get data in %rows.
105        CurVarData_L = (CurTest.Low_speed.(char(CurVar_L)).data)';
106
107        %Store the extracted data in multidimensional array
108        Tests_Low(1:length(CurVarData_L),k,i) = CurVarData_L;
109
110    end %end for k:NumVarLow
111
112 end %end for i:NumTests
113
114 %%Determine number of cycles to extract
```

```
115 %Find the number of samples in the shortest test, i.e the test with
116 %the lowest rpm
117 Duration_High = length(Tests(1).High_speed.Microphone_1.data);
118 Duration_Low = length(Tests(1).Low_speed.Time.data);
119
120 %The largest possible number of cycles that can be extracted:
121 %NumCycles = Duration/CycleLength;
122
123 %Chosen number of cycles
124 NumCycles = 200;
125
126 %Extract High speed measurements starting from sample number 4097
127 %due to an oddity in the time measurements. To match the low speed
128 %measurements,extract these starting from sample number 2.
129
130 %Extracting number of samples equal to the samples in
131 %NumCycles*CycleLength from High_Speed measurements,
132 %with a buffer of 2cycles.
133 Tests_High = Tests_High(4097:(NumCycles+2)*CycleLength+4096, :, :);
134
135 %Extracting Low speed measurements (fitted to numCycles,
136 %with a buffer)
137 Tests_Low = Tests_Low(2:257, :, :);
```

## C.2 Data Processing

This script deals with the data processing of the measurements, including Fourier transform, adaption of high and low speed measurements, and resampling. The output is a large table, where all measurements are gathered. Tables before and after resampling of the vibration measurements are given.

```
1 %Initialize
2 clearvars -except HighSpeedVar LowSpeedVar Tests Tests_High...
3     Tests_Low NumVarHigh NumVarLow NumTests NumCycles...
4     CycleLength FsLow;
5
6 close all;
7 clc;
8
9 %% Data extraction
10
11 %Preallocate matrices
12 %Microphone1
13 Vibarray_1=zeros (NumCycles*CycleLength,NumTests);
14 %Microphone2
15 Vibarray_2=zeros (NumCycles*CycleLength,NumTests);
16 %cylinder pressure
17 Pressurearray=zeros (NumCycles*CycleLength,NumTests);
18 %RPM
19 RPM =zeros (length (Tests_Low (:,1,1)),NumTests);
20 %reference enc.
21 Reference_encoder = zeros (NumCycles*CycleLength,NumTests);
22 Index = zeros (NumTests,1); %index array
23
24 %In order for the extracted cycles to have the "same content",
25 %a reference must be used when extracting the measurements.
26 %Thus, by finding the index of a pressure peak, and subtracting 2048
27 %samples from this index, a cycle will be
28 %-2048 samples : pressure peak : +2048 samples.
29
30 %For 1 to number of tests
31 for i=1:NumTests
32
33     %Extract cylinder pressure measurements and gather in array
34     Pressurearray_Temp = Tests_High (:,2,i);
35
36     %Find the first pressure peak that is at least 2048 samples in
37     % but not further in than 2048+1cycle
```

```

38     [M,I] = max(Pressurearray_Temp(2048:6144));
39
40     %Store index for later use
41     Index(i,1) = I;
42
43     %Find the reference signals from new index
44     Reference_encoder(:,i) = ...
45         Tests_High(I:NumCycles*CycleLength+I-1,11,i);
46
47     %Use the index to extract the pressure measurements so as to fit
48     %the cycles of the coming vibration measurements
49     Pressurearray(:,i) = Tests_High(I:NumCycles*CycleLength+I-1,2,i);
50
51     %Extract vibration measurements and gather in array
52     %Use the index found in previous line to define start point for
53     %first cycle
54
55     %Microphone1
56     Vibarray_1(:,i) = Tests_High(I:NumCycles*CycleLength+I-1,6,i);
57
58     %Microphone2
59     Vibarray_2(:,i) = Tests_High(I:NumCycles*CycleLength+I-1,7,i);
60
61     %Extract RPM measurements and gather in array
62     RPM(:,i) = Tests_Low(:,46,i);
63
64 end
65 clear Iref Ip Referene_encoder_temp Pressurearray_temp I;
66
67 %By starting the cycles at a later index, the start time of
68 %high speed and low speed measurements are no longer identical,
69 %however the difference is so small that it is negligble.
70 %Also, the low speed measurements are sampled at a lower sample
71 %rate,and seldom change from one measurement to the next.
72
73 %%Calculating means of rpm-measurements
74 %Preallocate
75 RPM_mean = zeros(NumTests,1);
76
77 %For 1 to number of tests
78 for i=1:NumTests
79
80     %Calculate mean of RPM vectors
81     RPM_mean(i,1) = mean(RPM(:,i));
82

```



```
83 end
84
85 %%Calculating sample rates
86 %Preallocate
87 Fs = zeros(NumTests,1);
88
89 %For 1 to number of tests
90 for i=1:NumTests
91
92     %Calculate sample rate which is dependant upon RPM
93     Fs(i,1) = round(2048*RPM_mean(i,1))/60;
94
95 end
96
97 %%Calculating time vectors
98 %Preallocate
99 Time = zeros(NumCycles*CycleLength,NumTests);
100
101 %For 1 to number of tests
102 for i=1:NumTests
103
104     %Calculate time vectors based on sample rate
105     Time(:,i) = 0:1/Fs(i,1):NumCycles*CycleLength/...
106         (Fs(i,1))-1/Fs(i,1);
107
108 end
109
110 %%Calculate Cycle times
111 %Preallocate
112 CycleTimes = zeros(NumCycles,NumTests);
113
114 %For 1 to number of tests
115 for i=1:NumTests
116
117     %For 1 to the number of cycles
118     for j=1:NumCycles
119
120         % Append the end time of each cycle to a matrix
121         CycleTimes(j,i) = j*(1/(RPM_mean(i,1)/(60*2)));
122
123     end %end for j:NumCycles
124
125 end %end for i:NumTests
126
127
```

```

128 %% Calculating Fourier transform of vibration signals
129
130 %Length
131 L = NumCycles*CycleLength;
132
133 %Frame lengths
134 windowLength = 4096;    %Frame length equal to one engine cycle
135 step = 2048;           % 50% overlap
136
137 %Shorter window lengths give better time resolution
138 %Longer window lengths give better frequency resolution
139
140 %Compute the total number of frames;
141 numFrames = floor((L-windowLength)/step) + 1;
142 %floor rounds down
143 %Take length of signal, subtract 1 frame
144 %divide by step to obtain number of frames
145
146 %Init;
147 CurPos = 1;
148 Position = 1;
149
150 %Preallocate matrices
151 %Microphone1
152 X_1 = zeros(ceil(windowLength/2), numFrames);
153 XBig_1 = zeros(numFrames*NumTests, round(windowLength/2));
154
155 %Microphone2
156 X_2 = zeros(ceil(windowLength/2), numFrames);
157 XBig_2 = zeros(numFrames*NumTests, round(windowLength/2));
158
159 % Calculate the FFT of each frame of each vibration measurement
160 % and append all FFT results in a large table
161
162 for i=1:NumTests %For 1 to the number of tests
163
164     % Get vibration measurements
165     Vib_1 = Vibarray_1(:,i);
166     Vib_2 = Vibarray_2(:,i);
167
168     % For each frame
169     for j=1:numFrames
170
171         % Get current frame
172         frame_1 = Vib_1(CurPos:CurPos+windowLength-1);

```

```

173     frame_2 = Vib_2(CurPos:CurPos+windowLength-1);
174
175     % Multiply the frame with the hamming window
176     frameW_1 = frame_1.*window(@hamming,length(frame_1));
177     frameW_2 = frame_2.*window(@hamming,length(frame_2));
178
179     % Find FFT of each frame
180     %compute magnitude of each frame spectrum
181     %normalize by the number of samples
182     FFT_1 = abs(fft(frameW_1))/windowLength;
183     FFT_2 = abs(fft(frameW_2))/windowLength;
184
185     % Due to symmetric property of the magnitude of DFT coeff:
186     %return first half of spectrum
187     FFT_1 = FFT_1(1:ceil(windowLength/2)); %ceiling rounds up
188     FFT_2 = FFT_2(1:ceil(windowLength/2));
189
190     % Gather all frames in one matrix
191     ...where the DFT coefficients of each frame are placed into
192     ...separate columns
193
194     %Microphone1
195     X_1(:,j) = FFT_1;
196     %Microphone2
197     X_2(:,j) = FFT_2;
198
199     %Update index
200     CurPos = CurPos + step;
201 end
202
203 %Gather cycles in a matrix
204 XBig_1(Position:Position+length(X_1(1,:))-1,:)= X_1';
205 XBig_2(Position:Position+length(X_2(1,:))-1,:)= X_2';
206
207 %Update index for matrix appending
208 Position = Position + length(X_1(1,:));
209
210 %Restore index for inner loop
211 CurPos = 1;
212
213 end
214 clear Position CurPos frame_1 frame_2 frameW_1 frameW_2 FFT_1...
215     FFT_2 Vib_1 Vib_2
216
217

```

```

218 %% Making a matrix of High speed measurements by taking the mean of
219 ...the samples across each frame
220
221 %Preallocating matrices:
222 % Temporary matrix used in loop
223 MeanMeas_H = zeros(numFrames,length(Tests_High(1,2:end,1)));
224
225 % Matrix that will contain the low speed measurements for each cycle
226 ...of each test
227 Measurements_H = zeros(NumTests*numFrames, ...
228     length(Tests_High(1,2:end,1)));
229
230 %Initialize indices
231 Pos = 1;
232 Loc = 1;
233
234 %For 1 to the number of tests
235 for i=1:NumTests
236
237     CurIdx = Index(i);
238
239     %Get measurements
240     Measurements = ...
241         Tests_High(CurIdx:NumCycles*CycleLength+CurIdx-1,2:12,i);
242
243     %For each cycle
244     for j=1:numFrames
245
246         %For each column of High Speed measurements
247         for k=1:length(Measurements(1,:))
248
249             %Take mean over number of samples equal to frame length
250             MeanMeas_H(j,k) = ...
251                 mean(Measurements(Pos:Pos+windowLength-1,k));
252
253         end %end for k:columns
254
255         %Update index position
256         Pos=Pos+step;
257
258     end %end for j:NumCycles
259
260     %Restore index
261     Pos = 1;
262

```

```

263     %Append matrices from each test to a common matrix
264     Measurements_H(Loc:Loc+numFrames-1,:) = MeanMeas_H;
265
266     %Update index
267     Loc = Loc+numFrames;
268
269 end %end for i:NumTests
270 clear Loc Pos Measurements MeanMeas_H CurIdx;
271
272 %% Creating a matrix of low speed measurements by connecting
273 ...measurements to frame times
274
275
276 %Low speed measurements are sampled at 10times per second
277 Time_Low = (0.1:1/FsLow:length(Tests_Low(:,1,1))/FsLow-1/FsLow)';
278
279 %%Calculate frame times
280 %Preallocate
281 FrameTimes = zeros(numFrames,NumTests);
282 %For 1 to number of tests
283 for i=1:NumTests
284
285     %For 1 to the number of cycles
286     for j=1:numFrames
287
288         % Append the end time of each frame to a matrix
289         FrameTimes(j,i) = j*((windowLength/2)/Fs(i));
290
291     end %end for j:numFrames
292
293 end %end for i:NumTests
294
295 %Preallocating matrices:
296 % Temporary matrix used in loop
297 Meas_L = zeros(numFrames,length(Tests_Low(1,2:end,1)));
298
299 % Matrix that will contain the low speed measurements for each
300 ...cycle of each test
301 Measurements_L = zeros(NumTests*numFrames, ...
302     length(Tests_Low(1,2:end,1)));
303
304 %Initialize index;
305 Position = 1;
306
307 % For 1 to the number of tests

```

```

308 for i=1:NumTests
309
310     %Get measurements
311     CurMeasurements = Tests_Low(:,2:end,i);
312
313     %Obtain frame times of current test
314     CurFrames = FrameTimes(:,i);
315
316     %For each frame
317     for j=1:numFrames
318
319         %Return the index of the time for samples that best matches
320         %the frame time.
321         [~,idx] = min(abs(Time_Low-CurFrames(j)));
322
323         %For each column of Low Speed measurements
324         for k=1:length(CurMeasurements(1,:))
325
326             %Append the measurements of the index found above to the
327             %subsequent frames
328             Meas_L(j,k) = (CurMeasurements(idx,k));
329         end %end for k:Columns
330
331     end %end for j:numframes
332
333     %Append matrices from each test to a common matrix
334     Measurements_L(Position:Position+numFrames-1,:) = Meas_L;
335
336     %Update index
337     Position = Position+numFrames;
338
339 end %end for i:NumTests
340 clear Position CurFrames CurMeasurements Meas_L idx;
341
342 %% Putting Low Speed and High Speed measurements in same matrix
343
344 %Preallocate
345 Measurements_H_L = zeros(numFrames*NumTests, ...
346     length(Measurements_H(1,:))+length(Measurements_L(1,:)));
347
348 %Append High Speed measurements
349 Measurements_H_L(:,1:length(Measurements_H(1,:))) = Measurements_H;
350
351 %Append Low speed measurements
352 Measurements_H_L(:,length(Measurements_H(1,:))+1:...
```

```

353     length(Measurements_L(1,:))+length(Measurements_H(1,:)) = ...
354     Measurements_L;
355
356     %%Include the FFT coefficients in the same table as the measurements
357     Measurements_ALL = zeros(numFrames*NumTests, length(XBig_1(1,:))+...
358         length(Measurements_H_L(1,:))+ length(XBig_2(1,:)));
359     Measurements_ALL(:,1:length(XBig_1(1,:)))= XBig_1;
360
361     Measurements_ALL(:,length(XBig_1(1,:))+1:length(XBig_1(1,:))+...
362         length(XBig_2(1,:))) = XBig_2;
363
364     Measurements_ALL(:,length(XBig_1(1,:))+length(XBig_2(1,:))+1:...
365         length(XBig_1(1,:))+length(XBig_2(1,:))+...
366         length(Measurements_H_L(1,:)))= Measurements_H_L;
367
368     %% Preparing column and row names for measurement table
369
370     colNames=num2cell(zeros(1,round(2*windowLength/2)+...
371         length(Measurements_H_L(1,:))));
372
373     FreqVarName_1 = {'Mik1Coeff'};
374     FreqVarName_2 = {'Mik2Coeff'};
375
376     for i=1:round(windowLength/2)
377         colNames(i) = cellstr(strjoin([FreqVarName_1,num2str(i)], '_'));
378     end
379
380     for i=round(windowLength/2)+1:windowLength
381         colNames(i) = cellstr(strjoin([FreqVarName_2,num2str(i-2048)],...
382             '_'));
383     end
384
385     colNames(2*round(windowLength/2)+1:2*round(windowLength/2)+...
386         length(Measurements_H_L(1,:))) = {...
387         'Cylinder_Pressure','Exhaust_Pressure_H','Fuel_Pressure',...
388         'Engine_Torque','Microphone_1','Microphone_2',...
389         'Crank_Encoder','CAM_ECU','Crank_ECU','Reference_Encoder',...
390         'TDC_ECU','EGT_Cylinder_1','EGT_Cylinder_2','EGT_Cylinder_3',...
391         'EGT_Cylinder_4','EGT_Cylinder_5','EGT_Cylinder_6',...
392         'Spill_water','Fuel_Temperature','Oil_Temperature',...
393         'EGT_after_turbine_1','EGT_after_turbine_2',...
394         'Intake_temperature_before_charger',...
395         'Intake_temperature_after_cooler',...
396         'Intake_temperature_before_cooler','Coolant_Temperature',...
397         'Water_Pressure','Oil_Pressure','Boost_Pressure',...

```

```

398 'Load_Pump', 'Exhaust_Pressure_L', 'Battery_voltage_ECU', ...
399 'Fuel_Pressure_ECU', 'Boost_Pressure_ECU', 'Engine_Speed_ECU', ...
400 'Fuel_Quantity_ECU', 'Fuel_Pressure_valve_ECU', ...
401 'Boost_valve_ECU', 'Begin_of_main_injection_ECU', ...
402 'Injection_Volume_ECU', 'Fuel_Consumption_ECU', ...
403 'Airmass_flow_ECU', 'Airmass_flow_stroke_ECU', ...
404 'Duration_of_main_injection_ECU', 'Engine_control_mode_ECU', ...
405 'Lambda_estimated_ECU', 'Fuel_Consumption_inlet', ...
406 'Fuel_consumption_inlet_total', 'Fuel_consumption_return', ...
407 'Fuel_consumption_return_total', 'Engine_Torque_Nm', ...
408 'Engine_Power_kW', 'Engine_Power_HP', 'Mass_flow_rate', ...
409 'Orifice_plate_pressure', 'Engine_speed', 'Turbo_speed', ...
410 'Cylinder_peak_pressure_position', 'Cylinder_peak_pressure', ...
411 'Room_pressure', 'Room_temperature', 'Room_humidity', 'NOx', ...
412 'CO', 'CO2', 'O2', 'Weight', 'NO2', 'THC', 'NO'};
413
414 %Preallocate
415 rowNames = cell(numFrames*NumTests, 1);
416 Test_names = {'1000rpm20Nm', '1200rpm20Nm', '1200rpm50Nm', ...
417 '1400rpm20Nm', '1400rpm50Nm', '1400rpm90Nm', '1600rpm20Nm', ...
418 '1600rpm50Nm', '1600rpm90Nm', '1800rpm20Nm', ...
419 '1800rpm50Nm', '1800rpm90Nm', '2000rpm20Nm', '2000rpm50Nm', ...
420 '2000rpm90Nm', '2200rpm20Nm', '2200rpm50Nm', '2200rpm90Nm', ...
421 '2400rpm20Nm', '2400rpm50Nm', '2400rpm90Nm', '2600rpm20Nm', ...
422 '2600rpm50Nm', '2600rpm90Nm'};
423
424 Basename = 'F_'; %Frame nr. (because rownames must be unique)
425 %Index
426 indicator = 1;
427
428 %For each test
429 for i=1:NumTests
430
431     %Get name of current test
432     Rowname = (Test_names(i));
433
434     %For each cycle
435     for j=1:numFrames
436
437         %Append test name and cycle name to rows
438         rowNames(indicator+j-1,1) = cellstr(strjoin([Rowname, ...
439             Basename,num2str(j)]));
440
441     end %for j:NumCycles
442

```



```

443     %Update index
444     indicator = indicator + numFrames;
445
446 end %end for i
447 clear Rowname indicator;
448
449 %% Create table and write to file
450
451 %Create table with all measurements, included FFT results
452 Measurements_ALL = array2table(Measurements_ALL, 'RowNames', ...
453     rowNames, 'VariableNames', colNames);
454
455 %Write to CSV
456 % writetable(Measurements_ALL, 'Measurements_ALL.csv', ...
457 %     'WriteRowNames', true);
458
459 %% Resample the FFT data and put into matrix together with all other
460 ...data
461
462 %%Creating an RPM vector with equal number of rows as XBig
463 %Preallocate %large RPM vector
464 RPM_Big = zeros(length(XBig_1(:,1)),1);
465 steg=1;
466
467 %For 1 to the number of different RPMs
468 for i=1:length(RPM_mean(:,1))
469
470     %Append the current rpm to the corresponding test rows
471     RPM_Big(steg:steg+numFrames-1) = round(RPM_mean(i));
472
473     %Update index
474     steg = steg+numFrames;
475
476 end %end for i:RPM_mean
477 clear steg;
478
479 %%Resampling
480 Reference_rpm = 3000; %chosen reference rpm for resampling
481
482 %Preallocate the resampled matrix
483 %After resampling, the number of columns will vary,
484 ...so pad with some extra zeros.
485 %Approximating the number of columns required and adding some extra
486 Pad = round((RPM_mean(end)/Reference_rpm)*length(XBig_1(1,:)));
487 XBig_Resampled_1 = zeros(length(XBig_1(:,1)),Pad+288);

```

```

488 XBig_Resampled_2 = zeros(length(XBig_2(:,1)),Pad+288);
489
490 %For all rows of XBig
491 for i=1:length(XBig_1(:,1))
492
493     %Get current row
494     CurRow_1 = XBig_1(i,:);
495     CurRow_2 = XBig_2(i,:);
496
497     %Get current rpm
498     CurRPM=RPM_Big(i);
499
500     %Resample
501     CurRow_resampled_1 = resample(CurRow_1,CurRPM,Reference_rpm);
502     CurRow_resampled_2 = resample(CurRow_2,CurRPM,Reference_rpm);
503
504     %Append to large matrix
505     XBig_Resampled_1(i,1:length(CurRow_resampled_1)) = ...
506         CurRow_resampled_1;
507     XBig_Resampled_2(i,1:length(CurRow_resampled_2)) = ...
508         CurRow_resampled_2;
509
510 end %end for i
511 clear CurRow_1 CurRow_2 CurRPM CurRow_resampled_1 ...
512     CurRow_resampled_2;
513
514 %Create a common matrix for the resampled FFT matrices
515 XBigBig = XBig_Resampled_1;
516 XBigBig = [XBigBig XBig_Resampled_2];
517
518 %%Add the resampled FFT matrix to the other measurements in a
519 ...large table
520 %Preallocate
521 Measurements_ALL_Resampled = zeros(length(XBig_1(:,1)),...
522     length(XBig_Resampled_1(1,:))+ length(Measurements_H_L(1,:))...
523     +length(XBig_Resampled_2(1,:)));
524
525 %Adding resampled FFT data
526 Measurements_ALL_Resampled(:,1:length(XBig_Resampled_1(1,:)))...
527     = XBig_Resampled_1;
528
529 Measurements_ALL_Resampled(:,length(XBig_Resampled_1(1,:))+1:...
530     (length(XBig_Resampled_1(1,:))+length(XBig_Resampled_2(1,:))))...
531     = XBig_Resampled_2;
532

```

```

533 %Adding high and low speed measurements
534 Measurements_ALL_Resampled(:,2*length(XBig_Resampled_1(1,:))+1:...
535     2*length(XBig_Resampled_1(1,:))+...
536     length(Measurements_H_L(1,:)))= Measurements_H_L;
537
538 %Finally, creating a table with corresponding row-and columnnames
539 rowNames; % The same as before resampling
540 %Preallocate
541 colNames_Resampled = num2cell(zeros(1,...
542     length(Measurements_ALL_Resampled(1,:))));
543
544 %Add Frequencies
545 for i=1:length(XBig_Resampled_1(1,:))
546     CurColName = strjoin({'Mik1Freq'},num2str(i));
547     CurColName = CurColName(CurColName ~= ' ');
548     colNames_Resampled(i) = cellstr(CurColName);
549 end
550 clear CurColName;
551
552 %Add Frequencies
553 for i=length(XBig_Resampled_1(1,:))+1:...
554     length(XBig_Resampled_1(1,:))+length(XBig_Resampled_2(1,:))
555     CurColName = strjoin({'Mik2Freq'},...
556         num2str(i-length(XBig_Resampled_1(1,:))));
557     CurColName = CurColName(CurColName ~= ' ');
558     colNames_Resampled(i) = cellstr(CurColName);
559 end
560 clear CurColName;
561
562 %Adding measurement names
563 colNames_Resampled(length(XBig_Resampled_1(1,:))+...
564     length(XBig_Resampled_2(1,:))+1:end) = ...
565     colNames(length(XBig_1(1,:))+length(XBig_2(1,:))+1:...
566         length(XBig_1(1,:))+length(XBig_2(1,:))+...
567         length(Measurements_H_L(1,:)));
568
569 %Creating table
570 Measurements_ALL_Resampled = array2table(...
571     Measurements_ALL_Resampled,'RowNames',rowNames,...
572     'VariableNames',colNames_Resampled);
573
574 clear i j k FreqVarName Basename;
575 %%Writing to csv
576 % writetable(Measurements_ALL_Resampled,...
577 % 'Measurements_ALL_Resampled_v4.csv','WriteRowNames',true);

```

