Vibration-based Monitoring of the Zwartewaterbrug

A Machine Learning Approach

Janno de Bruijn

September 2019

Vibration-based Monitoring of the Zwartewaterbrug: a Machine Learning Approach

by

Janno de Bruijn

to obtain the degree of Master of Science
at the Delft University of Technology.
to be defended publicly on Tuesday September 24, 2019 at 13:00.

Thesis committee:	Prof. dr. A.V. Metrikine,	TU Delft
	Prof. dr. E. Lourens,	TU Delft
	Ir. D. Fallais,	TU Delft
	Ir. F. Besseling,	Witteveen+Bos

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft Delft University of Technology

Witteveen + Bos

Acknowledgment

With this thesis my time as a student comes to an end. I am grateful to finalize this project as it was one of the hardest challenges I faced as student. On the other hand it gave me the opportunity to explore something news, the field of machine learning algorithms.

I want to thank a few persons who supported me in the past year. Floris, you made it possible for me to graduate at Witteveen+Bos. I appreciate our discussions, they were very valuable as you triggered me to think out of the box. Whenever I had a question or something to discuss, I could account on you. Eliz-Mari, you inspired me to go for a topic related to structural health monitoring and machine learning. Your knowledge on these topics helped me a lot and the discussions with you were always interesting. Both of you helped me to organize my thesis project during all phases.

I would also like thank my family and friends for their endless support during my study period. Finally, a special thank to my wife Femke. I cannot thank you enough for your love, support, patience and many bike rides which I needed to clear my mind.

Janno de Bruijn
Delft, September 2019

Abstract

In the Netherlands there are many bridges that crosses waterways, which are key nodes in the transportation system. The safety and integrity of these bridges is nowadays assessed by (visual) inspections at regular intervals. This approach cannot provide information about damage development in between inspections, leading to potential failure or high intervention costs. A maintenance strategy for bridges based on continuous vibration-based monitoring provides a possible cost-effective solution that may replace the periodic (visual) inspections. Vibration-based structural health monitoring of civil engineering structures is receiving increasing attention in recent years. This is due to new developments in related areas such as sensing technology, system identification, data mining and condition assessment.

A particular damage scenario that frequently occurs in steel orthotropic deck structures is the development of fatigue cracks in the welds between the longitudinal U-shaped stiffeners and the transverse beams or the steel deck plate. This damage scenario was also observed in the steel deck structure of the Zwartewaterbrug, located in Hasselt in the Netherlands. Vibration tests were performed on the bridge to develop a damage detection algorithm for early detection of the fatigue cracks in the bridge deck. In order to simulate the effect of damage, small masses were attached to the bottom of the bridge deck. The measurements were performed over a relatively short time period to exclude — in as far as possible — the effects of environmental variability. However, the bridge was kept open to traffic during the testing and structural response was primarily triggered by traffic load. The vibration data (accelerations) obtained during these tests was used for validation of the proposed methodology in this thesis.

The aim of the thesis is to detect the added masses from the vibration data. To achieve this, a data-based approach for damage detection was proposed, which, besides the data acquisition, consists of four main stages: data preprocessing, feature extraction, pattern recognition and decision making. The analysis is focused on the high-frequency acceleration response, because the associated higher-frequency local modes are more sensitive to the small changes induces by the added mass. The *similarity filtering* preprocessing procedure was applied to filter out the variability related to the operational loading, and the Frequency Domain Decomposition system identification technique was used to extract damage indicators. Subsequently, a support vector machine algorithm was applied to "learn" patterns in the extracted damage indicators, which enables the algorithm to assign a damage state to a given measurement vector. Alternatively, a novelty detection technique was used to distinguish data from the "healthy" structure from data corresponding to a "damage" state.

It was not possible to reliable detect the added masses from the high-frequency acceleration data due to the large variations in the damage indicators of a mass class. These variations are attributed to the changing environmental conditions over the measurement period. The principal component analysis was used in an attempt to remove these variations from the damage indicators, but the results were still not satisfactory.

In conclusion, it was not possible to properly test the proposed methodology for damage detection using the data of the Zwartewarterbrug, mainly because the data was not suitable for the machine learning algorithms employed. Recommendations are given related to the methodology for the similarity filtering, determination of the high-frequency range, and machine learning algorithm to be used. Additionally, recommendations are given for future measurement campaigns to obtain data that is more suitable for successful application of the methodologies employed in the present project.

Contents

List of Acronyms ix

1 Introduction 1
- 1.1 Problem Definition . . . 2
- 1.2 Objectives . . . 2
- 1.3 Thesis Outline . . . 2

2 The Zwartewaterbrug Project 5
- 2.1 Bridge Description . . . 5
- 2.2 Measurement Campaign . . . 6
  - 2.2.1 Sensor Network . . . 6
  - 2.2.2 Added Mass . . . 6
  - 2.2.3 Data Acquisition . . . 7
- 2.3 Environmental Variability . . . 9
- 2.4 Methodology . . . 12

3 Theory: Similarity Filtering and Frequency Domain Decomposition 15
- 3.1 Similarity Filtering . . . 15
  - 3.1.1 Single-Input/Single-Output System . . . 15
  - 3.1.2 Multiple-Input/Single-Output System . . . 19
  - 3.1.3 PSD of Processed Signal . . . 24
  - 3.1.4 Practical Matters: Segments . . . 26
  - 3.1.5 Summary . . . 29
- 3.2 Frequency Domain Decomposition . . . 31
  - 3.2.1 Modal Decomposition . . . 31
  - 3.2.2 Uncorrelated Modal Coordinates . . . 33
  - 3.2.3 Spectral Matrix Decomposition . . . 34
  - 3.2.4 Identification Algorithm . . . 35
  - 3.2.5 Modal Assurance Criterion . . . 38

4 Application: Similarity Filtering and Frequency Domain Decomposition 39
- 4.1 Bandpass Filtering . . . 40
- 4.2 Similarity Filtering . . . 44
  - 4.2.1 Algorithm Composition . . . 44
- 4.3 Frequency Domain Decomposition . . . 49
  - 4.3.1 Modal Analysis . . . 55
  - 4.3.2 Conclusions . . . 56

5 Feature Engineering 65
- 5.1 Visual Inspection of the Spectra . . . 66
- 5.2 Feature Selection . . . 70

6 Supervised Learning - Classification 75
- 6.1 Introduction . . . 75
- 6.2 Bayesian Decision Theory . . . 76
- 6.3 Support Vector Machines . . . 81
  - 6.3.1 Nonseparable Classes . . . 84
  - 6.3.2 The Multiclass Case . . . 86
  - 6.3.3 The Nonlinear Case . . . 86
  - 6.3.4 Summary . . . 88

6.4 Application . . . 89
6.4.1 Classification per segment . . . 89
6.4.2 Classification per sensor . . . 95
6.5 Conclusion . . . 100

7 Unsupervised Learning - Novelty Detection 101
7.1 Discordance Measures . . . 101
7.1.1 A Gaussian Distributed Normal Condition . . . 102
7.1.2 Calculation of Threshold Values . . . 103
7.1.3 Process of Novelty Detection . . . 105
7.2 Application . . . 105
7.2.1 Feature Selection . . . 105
7.2.2 Threshold Value . . . 110
7.2.3 Comparison results per segment . . . 112
7.3 Conclusions . . . 116

8 Data Normalization 117
8.1 Overview of Studies on Environmental and Operational Variability . . . 117
8.2 Temperature Dependency of Natural Frequencies . . . 121
8.2.1 Mass effect or temperature effect? . . . 129
8.3 Data Normalization . . . 132
8.3.1 Regression and Interpolation models . . . 132
8.3.2 Decomposition methods . . . 135
8.3.3 Novelty Detection Under Changing Environmental Conditions . . . 138
8.3.4 Application . . . 140
8.4 Chapter Summary . . . 145

9 Conclusions and Recommendations 147
9.1 Conclusions . . . 147
9.2 Recommendations . . . 148
9.2.1 Recommendations related to the methodology . . . 148
9.2.2 Recommendations for future measurement campaign . . . 149

A Convolution & Fourier Transform 153
A.1 Fourier Transform . . . 153
A.2 Convolution . . . 154
A.3 Basic Dynamic Characteristics . . . 154

B Linear Algebra 157
B.1 Singular Value Decomposition . . . 157
B.2 Kronecker Notation . . . 158

C Principal Component Analysis 161

D PSD decomposition 165

E Performance of Similarity Filtering 167

F Spectra of Processed Signals 173
F.1 PSD spectrograms . . . 173
F.2 Singular value spectra segment 1 & 3 . . . 180

G Modal Analysis 183
G.1 Identification Algorithm . . . 183
G.2 Results Segment 1 . . . 187
G.3 Results Segment 3 . . . 190

H Classification results per sensor 193

Bibliography 199

List of Acronyms

CDF	Cumulative Density Function
CSD	Cross Spectral Density
DAQ	Data Acquisition
DFT	Discrete Fourier Transform
DOF	Degree-of-Freedom
EFDD	Enhanced Frequency Domain Decomposition
EOV	Environmental and Operational Variability
EVD	Eigenvalue Decomposition
FA	Factor Analysis
FDD	Frequency Domain Decomposition
FEM	Finite Element Model
FFT	Fast Fourier Transform
FRF	Frequency Response Function
IRF	Impulse Response Function
LTI	Linear Time-Invariant
MAC	Modal Assurance Criterion
MDOF	Multi-Degree-of-Freedom
MISO	Multiple-Input/Single-Output
MSD	Mahalanobis Squared Distance
NDE	Non-Destructive Evaluation
PCA	Principal Component Analysis
PSD	Power Spectral Density
RBF	Radial Basis Function
SDOF	Single-Degree-of-Freedom
SHM	Structural Health Monitoring
SISO	Single-Input/Single-Output
SVD	Singular Value Decomposition
SVM	Support Vector Machine

1 Introduction

The state of civil structures such as bridges deteriorates over time. For example, fatigue crack development in steel bridges may suddenly lead to brittle failure. Maintenance programs are supposed to mitigate safety risks associated with deterioration. Nowadays, the safety and integrity of bridges is assessed by visual inspections performed at regular time intervals, and in some cases supplemented by Non-Destructive Evaluation (NDE) such as acoustic or ultrasonic measurement techniques, magnetic field, and radiography. All these methods require that the location of the damage is known a priori and is easily accessible for inspection. Subject to these limitations, the NDE methods can detect damage on or near the surface of a structural element. However, detection of damage too far beneath the surface is not guaranteed and the methods cannot provide information about the health of internal members out of reach. A major drawback of periodic inspections is that damage or failure could happen in between inspections. Also, costs can be high. Continuous vibration-based monitoring of bridges provides a possible alternative solution to these challenges and has been studied extensively in the last decades aiming to develop cost-effective Structural Health Monitoring (SHM) systems [17, 18].

The topic of this thesis is vibration-based monitoring, which is a subset of SHM. The main idea behind vibration-based damage detection methods is that damage will alter the physical properties of a structure, which will cause detectable changes in the dynamic response properties of the structure [17]. Originally, changes in modal parameters (natural frequencies, mode shapes and modal damping) were used as damage indicators. However, as damage is typically a local phenomenon it will not significantly influence the low-frequency global response of the structure that is normally measured during vibration tests with ambient excitation. The local response that is captured by high-frequency modes instead shows higher sensitivity to local damage. This response is typically more difficult to excite in vibration tests as more energy is required to produce measurable responses at these higher frequencies, particularly when a structure is excited by ambient loading. In addition, the modal properties appear to be sensitive to Environmental and Operational Variability (EOV) as well. For bridge decks specifically, the most important sources of variability in modal responses were found to be temperature and traffic loading. These problems have led to the need of other damage indicators, or methods that are capable of amplifying the effect of damage and removing the effects of EOV. Although many attempts were made to obtain better damage indicators, this is still an active research topic being partially unresolved.

Over the years, the damage detection algorithms that have been developed for data-driven approaches are mainly drawn from the discipline of pattern recognition, or more broadly, machine learning [52]. The data-driven approach avoids the need of a high-fidelity physical model of the structure and offers the possibility of automating the SHM process. The idea of machine learning in the context of SHM is to "learn" the relationship between some *features* derived from the measured data and the damaged state of the structure. The learning problem then is to estimate the function that describes this relationship using data acquired from the structure—the training data. The training data consist of feature vectors extracted from the data, possibly supplemented with the corresponding class label/output. The feature vectors must be sensitive to the damage and insensitive to EOV. This is one of the main challenges when a machine learning approach is adopted and forms a major issue dealt with in this thesis.

1.1 Problem Definition

The current research continues the research performed in the thesis of Milosevic [32], and contributes to the development of a vibration-based structural health monitoring system for early detection of small structural faults in steel bridge decks. In particular, to further automate the damage detection procedure proposed in [32], a machine learning approach is adopted here.

The methodology proposed in this thesis is tested using measurement data from the Zwartewaterbrug, an arch bridge in the Province of Overijssel in the Netherlands. As many other orthotropic bridge decks, the deck structure of the Zwartewaterbrug is prone to fatigue cracks in the welds between the U-shaped longitudinal stiffeners and the transverse beams, and in the welds between the stiffeners and the steel deck plate. In March 2017 a SHM system was installed on the Zwartewaterbrug to measure the acceleration response of the orthotropic bridge deck. One of the purposes of this short measurement campaign was to demonstrate the possibility to detect the frequently occurring fatigue cracks in the deck. In order to simulate the effects of these fatigue cracks, small masses were attached to the bottom of the bridge deck. The use of such *pseudo faults* allows for the generation of data from a modified or "damaged" structure without actually damaging the structure.

The detection of the small added masses from the vibration data of the Zwartewaterbrug through a machine learning approach is the main goal of this thesis, leading to the following main question:

"Is it possible to detect small added masses from the vibration data of the Zwartewaterbrug using machine learning algorithms."

The following challenges to answer the research question will be faced:

- The small dataset due to the short measurement campaign and the possibly varying environmental conditions.
- The low sensitivity of the dynamic response to the small structural changes. The effect of damage is simulated by very small masses attached to the bottom of the bridge deck; e.g. the smallest mass of 25 kg amounts to only 0.0625% of the weight of an instrumented bridge segment.
- The non-stationary acceleration time-histories caused by the unknown traffic load with vehicle weight ranging from 200 kg to 50000 kg.

1.2 Objectives

The objectives of this thesis are as follows:

- To remove non-stationarities in the acceleration time histories caused by varying traffic load.
- To extract features from the vibration data that are sensitive to small structural changes and insensitive to environmental and operational variabilities.
- To detect the small added masses that were attached to the bridge deck from the vibration measurements of the Zwartewaterbrug using machine learning algorithms.
- To assess whether small local defects in bridge decks can be detected based on high-frequency acceleration measurements only.
- To investigate possibilities of machine learning applications using data from a short measurement campaign.

1.3 Thesis Outline

The outline of the thesis is schematically presented in Figure 1.1.

In Chapter 2 the Zwartewaterbrug project is briefly described. The vibration data from this project forms the basis of the analyses performed in the next chapters. The followed methodology for damage detection is also described in this chapter.

Chapter 3 presents the theoretical background of two important techniques that are applied in the first part of the thesis: similarity filtering and Frequency Domain Decomposition (FDD). The method of similarity filtering is a preprocessing technique, which aims to filter out the variability related to the operational loading (e.g. traffic loading) from the data. The FDD technique is an output-only system identification technique that is used to extract the damage-sensitive features from the vibration data.

In Chapter 4 the preprocessing (i.e. the process from raw data to initial features) of the Zwartewaterbrug data is described. The preprocessing consist of three steps: band-pass filtering, similarity filtering and FDD. In the first step a reliable high frequency range is determined being sensitive to small changes in the bridge deck, and the acceleration time histories are band-pass filtered based on this frequency range. Secondly, the similarity filtering preprocessing procedure is applied to filter out the variability related to the operational loading. In the third step, the FDD is applied to the processed signals in order to obtain the *initial features*. These features are further processed in the next chapters before they are used as inputs for the machine learning algorithms. At last, The natural frequencies and mode shapes of the bridge resulting from the FDD are analysed to confirm their local character.

In Chapter 5 the concept of feature engineering is briefly described. Feature engineering is the general process of generating proper inputs for the machine learning algorithms. In SHM context, these inputs are referred to as *damage-sensitive features*. The initial features that are extracted in previous chapter are visually inspected, and preliminary conclusions regarding the mass class separability are drawn. Finally, the proposed feature selection procedure is presented.

Chapters 6 and 7 are concerned with the detection of the small added masses that were attached to the bridge deck using the damage-sensitive features extracted from the vibration data. Two types of machine learning algorithms are employed that "learn" the relationship between the features and the damaged state of the structure.

In Chapter 6 the Support Vector Machine (SVM) classification is applied to learn this relation. The SVM is a supervised learning algorithm meaning that the diagnostic is trained by showing the true labels of the data samples. The chapter starts with the introduction of some important concepts for classification through the Bayesian decision theory. Next, the theoretical background of SVM classification is briefly described. The last part of the chapter concerns the application of the SVM algorithm to the Zwartewaterbrug data, with the goal to train a SVM model that correctly classify the data samples according to the corresponding weight of the added mass.

In contrast to the previous chapter, in Chapter 7 the diagnostic is trained using training data from the normal operating condition (i.e. healthy condition) of the structure only. The applied algorithm belongs to the class of outlier or novelty detection methods. In these methods a model of normal condition is created, and the test data are compared with this model to detect abnormalities. A threshold level is used to determine if a certain observation significantly deviates from the normal condition and should be flagged as abnormal. The theory of novelty detection and different approaches for setting the threshold level are discussed first, followed by an application on the data of the Zwartewaterbrug.

In Chapter 8 the importance of data normalization for the design of a reliable damage detection algorithm is discussed. It starts with an overview of studies on Environmental and Operational Variability (EOV), the main focus being the effect of temperature variations on the natural frequency of a structure. Thereafter, the correlation between the ambient temperature and natural frequencies of the Zwartewaterbrug is analysed. In the second part of the chapter different methods for data normalization are discussed. The discussion is restricted to two classes of methods: regression methods and decomposition methods. In the end, a data normalization approach based on the Principal Component Analysis (PCA) is applied to the data of the Zwartewaterbrug and it is illustrated that the algorithm applied in Chapter 7 is able to filter out the environmental variability in much the same way as the method based on linear PCA.

Finally, Chapter 9 provides the overall conclusions of the work based on the obtained results, and recommendations are given for future research and successful future measurement campaigns.

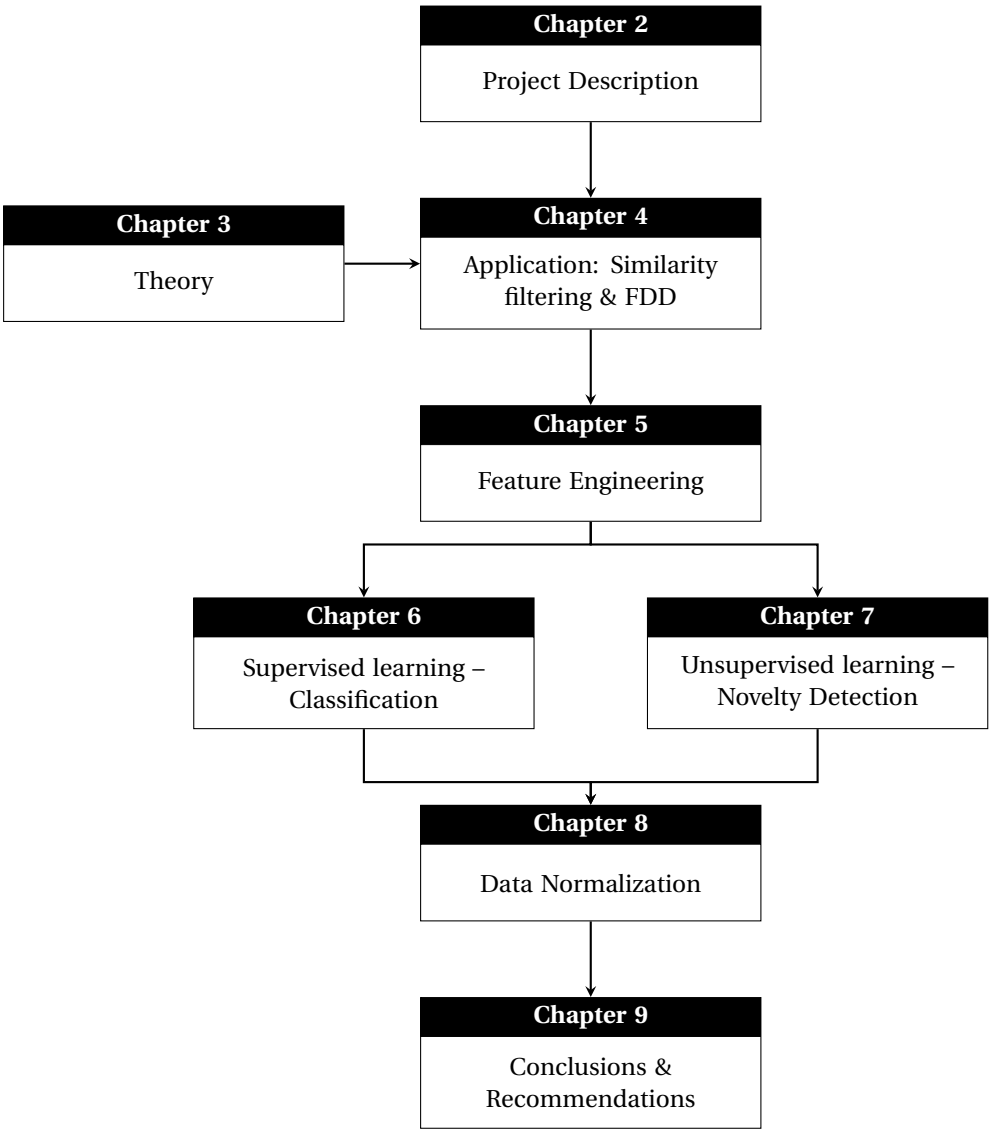


Figure 1.1: Thesis outline.

2

The Zwartewaterbrug Project

2.1 Bridge Description

The Zwartewaterbrug, depicted in Figure 2.1, crosses the Zwartewater river near the city of Hasselt in the Dutch province Overijssel. It is a steel arch bridge with a span of 104 meters, built in 1969.

Figure 2.1: The Zwartewaterbrug.

The orthotropic deck consists of an asphalt-covered steel deck plate, I-shaped longitudinal main girders, I-shaped transverse beams, and a total of 66 smaller U-shaped longitudinal stiffeners. Figure 2.2 shows a part of one of the spans between two transverse beams. Also seen on this figure are the U-shaped longitudinal stiffeners. The detection of frequently occurring fatigue cracks in the welds between these U-shaped stiffeners and the transverse beams was targeted with the proposed Structural Health Monitoring (SHM) installation.

Figure 2.2: A part of the orthotropic deck as seen from the bottom.

2.2 Measurement Campaign

The main idea behind the Zwartewaterburg project was to acquire acceleration data from the structure in normal condition, and the structure with the presence of small damage. It is assumed that the damage reflects itself in terms of reduced stiffness. The reduction of stiffness has an influence on the dynamic properties of the bridge. In particular, natural frequencies are expected to decrease, and it is assumed that this reduction could equally well be due to an increase in mass. This latter is motivated by the expression for the natural frequency of a simple mass-spring system ($\omega_n = \sqrt{k/m}$). With this idea in mind, the measurement campaign of the Zwartewaterbrug project was designed. It must be noted that the campaign was initially not intended for machine learning applications, but will be the main focus of this thesis.

2.2.1 Sensor Network

The orthotropic deck of the bridge is instrumented with 32 accelerometers. The sensors are divided over three adjacent spans between transverse beams, and are placed over half the width of the bridge. The spatial positioning of the sensors is illustrated in Figure 2.3. In this thesis, the span with sensor 1 to 10 is referred to as segment 1, the span with sensor 11 to 22 is referred to as segment 2, and the span with sensor 23 to 32 is referred to as segment 3. The location of the ten sensors in segment 1 and 3 deviates from the twelve sensors in segment 2. The added mass is located in the middle of segment 2, between sensor 16 and 17 (blue square in Figure 2.3). The instrumented section is located at north side of the bridge near the concrete pillar.

2.2.2 Added Mass

Small masses were used to simulate the presence of damage. The added mass system, depicted in Figure 2.4, was attached to the bridge deck using a magnet, as shown in Figure 2.5. Sand was used to regulate the weight of the added mass. Measurements with added mass were performed in steps of 25 kg, with a maximum total mass of 100 kg including the magnet's own weight of 25 kg. It must be mentioned that the masses are extremely small in relation to the weight of the deck itself: the mass of 25 kg amounts to only ±0.0625% of the weight of the instrumented section. In terms of damage sensitivity, it is important to note that these masses will have absolutely no influence on the lower modes of the bridge, hence the focus on higher resonances for early damage detection.

In Figure 2.4 it can be observed that the added mass is hanging below the bridge deck with flexible hangers. This could cause non-linear effects, for example, induced by wind of different direction and intensity.

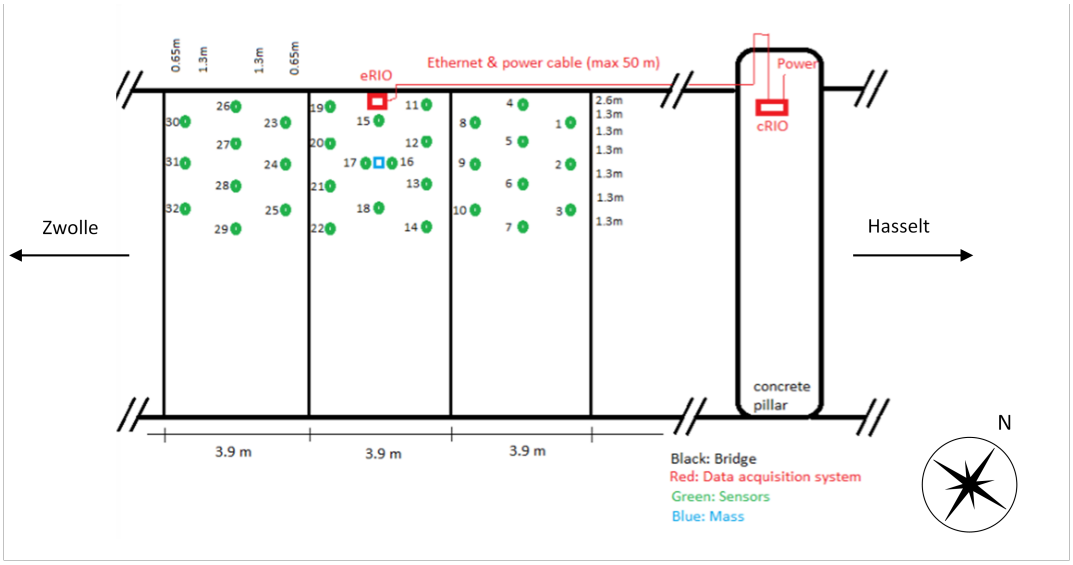


Figure 2.3: Spatial position of sensors.

Figure 2.4: The mass system attached to the bottom of the deck to act as pseudo-faults.

2.2.3 Data Acquisition

The vertical acceleration response of the bridge deck due to ambient loading is simultaneously measured by 32 accelerometers with a sample rate of 1000 Hz. The measurements are performed on three different days (not continuously measured, and not during the whole day), and five different added masses are considered. In total 764 different recordings of varying length are received, each one consisting of 32 signals. The received data is summarized in Table 2.1. In general, seven data sets can be distinguished, each one corresponding to a different time moment and/or added mass. The temperature listed in the last column of the table represents the ambient temperature that is measured at the data acquisition system. More detailed temperature data is not available.

(a) (b)

Figure 2.5: Attachment of added mass to bridge deck using a magnet.

Figure 2.6a illustrates the vertical acceleration time histories of the bridge deck at the 32 sensor locations. It can be observed that the measured dynamic responses present the non-stationary characteristic due to the ambient loading (e.g. varying traffic load). The large variations in traffic load is considered as an important factor that complicates the detection of small changes in the structure. This issue will be addressed in the following chapters on the preprocessing of the data. Figure 2.6b shows the power spectral density functions corresponding to previous time histories. From this it can be observed that most energy is contained in the frequency components below 100 Hz; in particular, frequencies around 20 Hz and 60 Hz.

Table 2.1: Summary received data from Zwartewaterbrug.

Ref. no.	Date	From	To	Added mass (kg)	Available data (minutes)	Number of records	Temperature (°C)
1	23-03-2017	08:29:07	09:47:50	0	18.9	52	7.64 ± 0.51
2		10:40:04	12:23:35	100	56.12	151	11.31 ± 0.47
3		12:40:42	13:40:40	50	34.43	85	12.42 ± 0.20
4		14:12:23	15:36:38	0	51.13	112	13.01 ± 0.12
5	28-03-2017	09:32:27	10:51:50	25	38.89	111	13.07 ± 0.96
6		11:34:02	12:53:36	50	48.46	126	16.59 ± 0.61
7	31-03-2017	07:47:30	09:12:22	75	53.34	127	15.22 ± 0.34

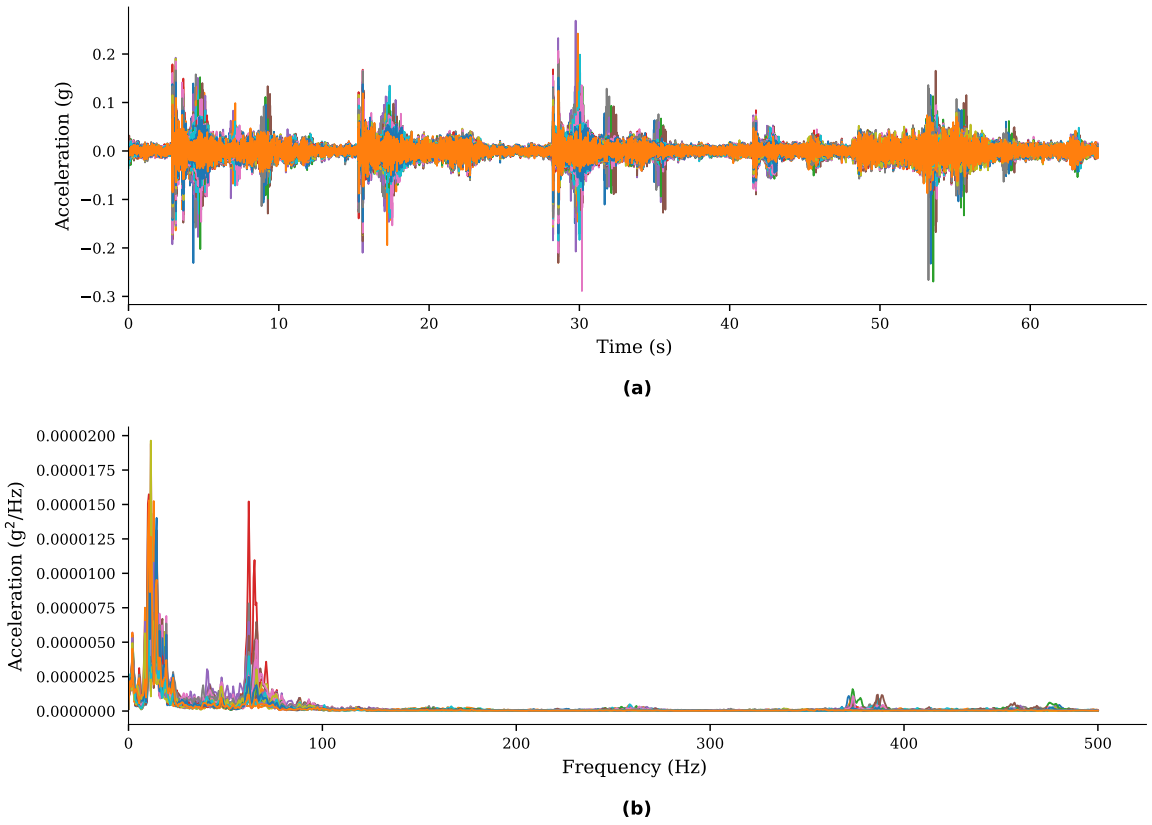


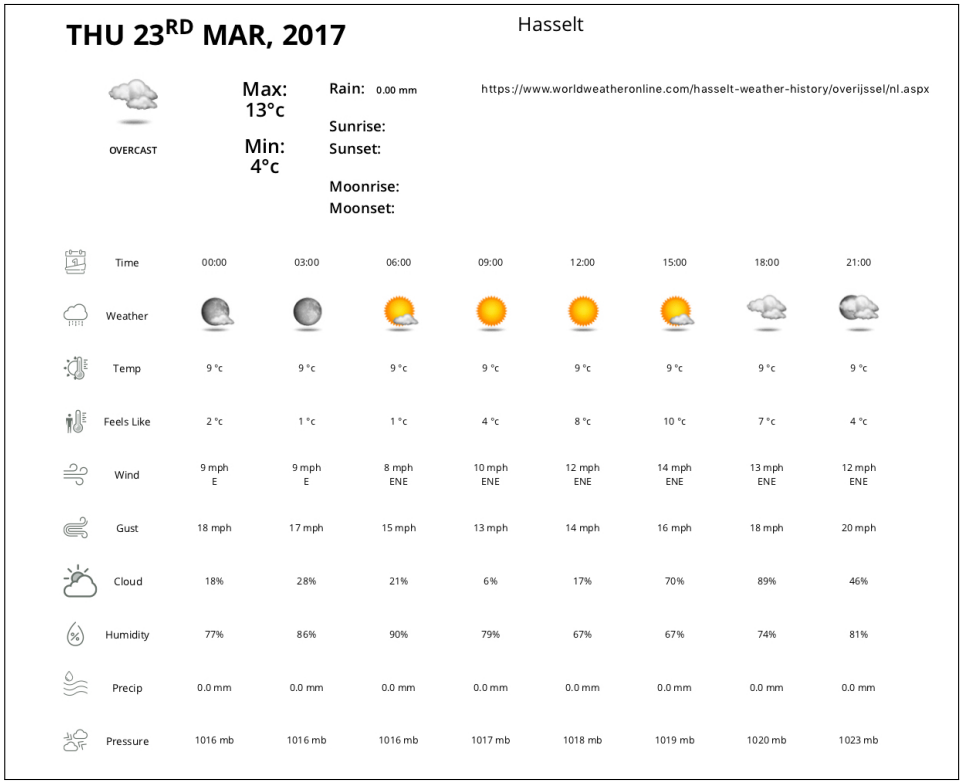
Figure 2.6: An example of (a) the time histories of 32 sensors, and (b) the corresponding power spectral density functions.

2.3 Environmental Variability

Information about the environmental conditions during the measurement campaign is obtained from Netherlands Historical Weather Almanac, and the temperature data recorded during the experiments. Figures 2.7–2.9 show the 3-hours average weather forecast for the 23rd, 28th and 31st of March 2017, respectively. The ambient temperature recorded at the start of each measurement is shown in Figures 2.10–2.12. It must be noted that the ambient temperature is recorded at a single location only, and might not be representative for the temperature distribution across the bridge deck.

The temperature is slightly different on the three measurement days. On 23rd March the temperature during the measurements ranges from 7°C in the morning to 13°C in the afternoon, on 28th of March the temperature ranges from 12°C in the morning to 17°C in the early afternoon, and on 31st of March the temperature ranges from 15°C to 16°C. The latter were measured early in the morning (from 7:47 to 9:12) and were completed before the rain started later that day.

The main goal of this thesis is to develop a damage detection algorithm sensitive to small changes, the environmental conditions (e.g. temperature) are considered as possible factor for deviations in the results. For the Zwartewaterbrug only limited temperature data is available complemented with some weather charts, which is not enough to include the impact of environmental conditions in the development of the damage detection algorithm.



THU 23RD MAR, 2017 — Hasselt

OVERCAST

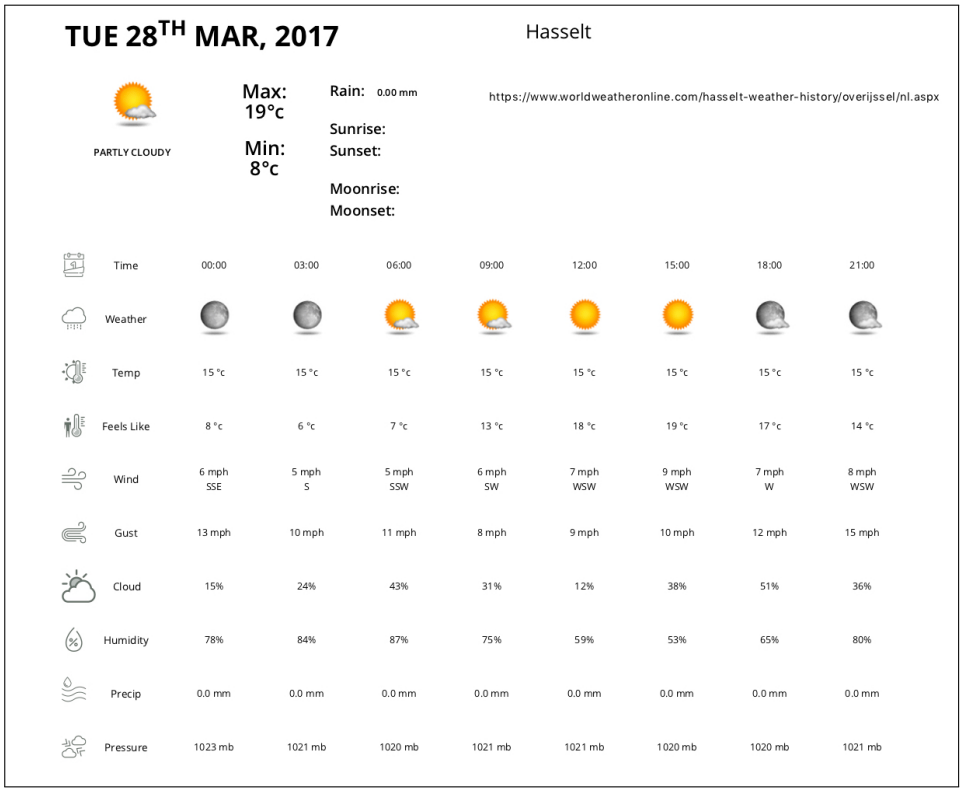
Max: 13°c
Min: 4°c

Rain: 0.00 mm
Sunrise:
Sunset:
Moonrise:
Moonset:

https://www.worldweatheronline.com/hasselt-weather-history/overijssel/nl.aspx

	00:00	03:00	06:00	09:00	12:00	15:00	18:00	21:00
Time								
Weather								
Temp	9 °c	9 °c	9 °c	9 °c	9 °c	9 °c	9 °c	9 °c
Feels Like	2 °c	1 °c	1 °c	4 °c	8 °c	10 °c	7 °c	4 °c
Wind	9 mph E	9 mph E	8 mph ENE	10 mph ENE	12 mph ENE	14 mph ENE	13 mph ENE	12 mph ENE
Gust	18 mph	17 mph	15 mph	13 mph	14 mph	16 mph	18 mph	20 mph
Cloud	18%	28%	21%	6%	17%	70%	89%	46%
Humidity	77%	86%	90%	79%	67%	67%	74%	81%
Precip	0.0 mm	0.0 mm	0.0 mm	0.0 mm	0.0 mm	0.0 mm	0.0 mm	0.0 mm
Pressure	1016 mb	1016 mb	1016 mb	1017 mb	1018 mb	1019 mb	1020 mb	1023 mb

Figure 2.7: Hasselt, Overijssel, Netherlands Historical Weather Almanac, March 23, 2017.



TUE 28TH MAR, 2017 — Hasselt

PARTLY CLOUDY

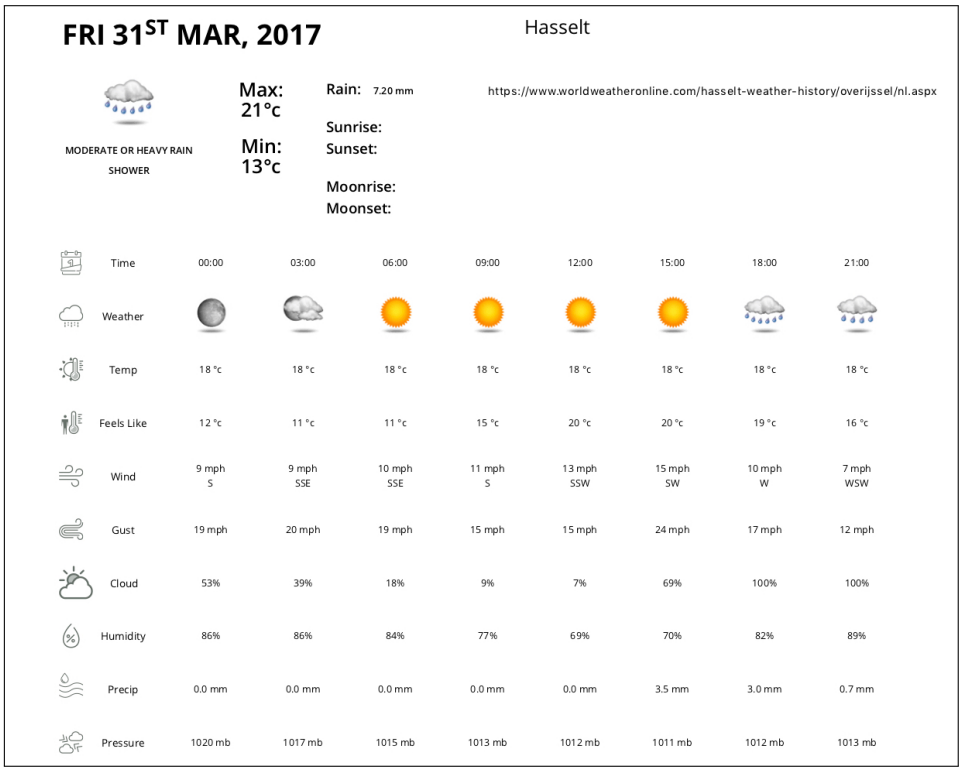
Max: 19°c
Min: 8°c

Rain: 0.00 mm
Sunrise:
Sunset:
Moonrise:
Moonset:

https://www.worldweatheronline.com/hasselt-weather-history/overijssel/nl.aspx

	00:00	03:00	06:00	09:00	12:00	15:00	18:00	21:00
Time								
Weather								
Temp	15 °c	15 °c	15 °c	15 °c	15 °c	15 °c	15 °c	15 °c
Feels Like	8 °c	6 °c	7 °c	13 °c	18 °c	19 °c	17 °c	14 °c
Wind	6 mph SSE	5 mph S	5 mph SSW	6 mph SW	7 mph WSW	9 mph WSW	7 mph W	8 mph WSW
Gust	13 mph	10 mph	11 mph	8 mph	9 mph	10 mph	12 mph	15 mph
Cloud	15%	24%	43%	31%	12%	38%	51%	36%
Humidity	78%	84%	87%	75%	59%	53%	65%	80%
Precip	0.0 mm	0.0 mm	0.0 mm	0.0 mm	0.0 mm	0.0 mm	0.0 mm	0.0 mm
Pressure	1023 mb	1021 mb	1020 mb	1021 mb	1021 mb	1020 mb	1020 mb	1021 mb

Figure 2.8: Hasselt, Overijssel, Netherlands Historical Weather Almanac, March 28, 2017.



FRI 31ST MAR, 2017

Hasselt

MODERATE OR HEAVY RAIN SHOWER

Max: 21°c

Min: 13°c

Rain: 7.20 mm

Sunrise:

Sunset:

Moonrise:

Moonset:

https://www.worldweatheronline.com/hasselt-weather-history/overijssel/nl.aspx

Time	00:00	03:00	06:00	09:00	12:00	15:00	18:00	21:00
Temp	18 °c	18 °c	18 °c	18 °c	18 °c	18 °c	18 °c	18 °c
Feels Like	12 °c	11 °c	11 °c	15 °c	20 °c	20 °c	19 °c	16 °c
Wind	9 mph S	9 mph SSE	10 mph SSE	11 mph S	13 mph SSW	15 mph SW	10 mph W	7 mph WSW
Gust	19 mph	20 mph	19 mph	15 mph	15 mph	24 mph	17 mph	12 mph
Cloud	53%	39%	18%	9%	7%	69%	100%	100%
Humidity	86%	86%	84%	77%	69%	70%	82%	89%
Precip	0.0 mm	0.0 mm	0.0 mm	0.0 mm	0.0 mm	3.5 mm	3.0 mm	0.7 mm
Pressure	1020 mb	1017 mb	1015 mb	1013 mb	1012 mb	1011 mb	1012 mb	1013 mb

Figure 2.9: Hasselt, Overijssel, Netherlands Historical Weather Almanac, March 31, 2017.

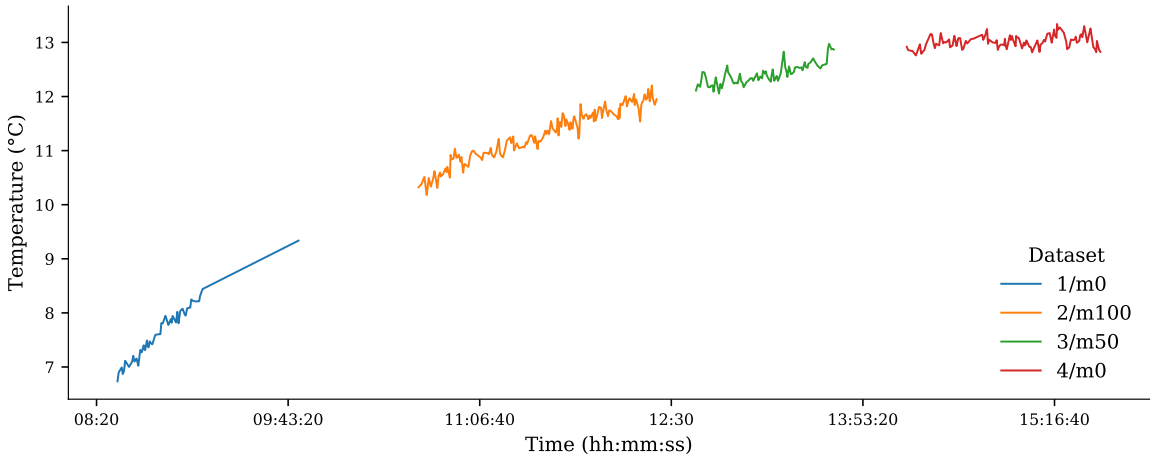


Figure 2.10: Ambient temperature recorded at the start of each measurement on March 23, 2017.

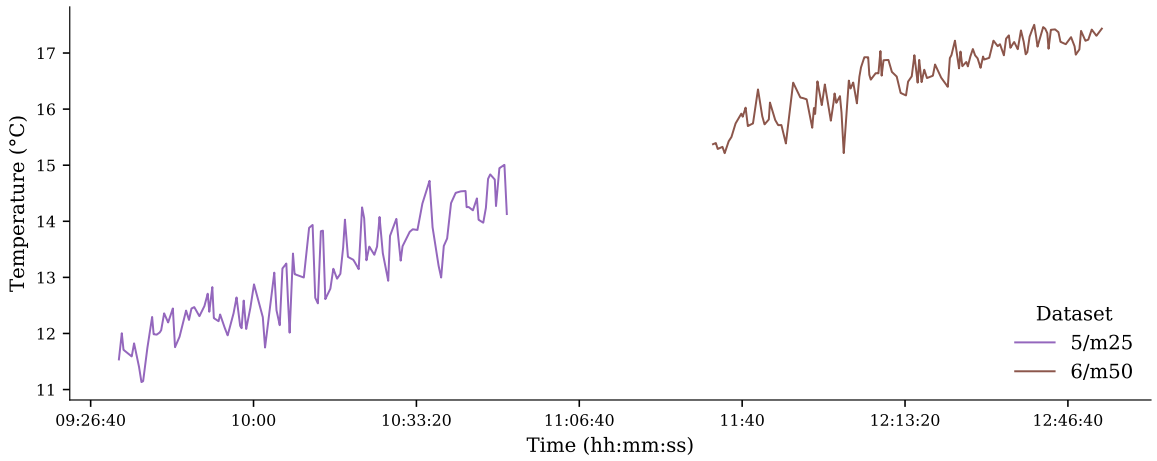


Figure 2.11: Ambient temperature recorded at the start of each measurement on March 28, 2017.

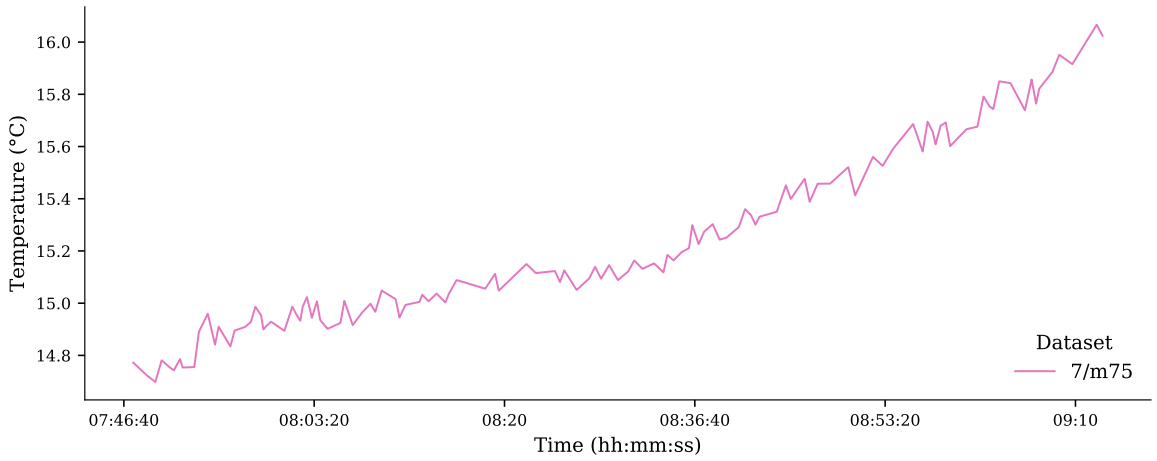


Figure 2.12: Ambient temperature recorded at the start of each measurement on March 31, 2017.

2.4 Methodology

The followed methodology is schematically illustrated in Figure 2.13. The main steps are: *data acquisition*, *preprocessing*, *feature extraction*, *pattern recognition*, and *decision making*. At the end of the analysis it should be possible to make a decision as to the state of the structure.

The first stage is data acquisition which is concerned with the generation of vibration data. The data acquisition is not part of this thesis, but the raw data of the Zwartewaterbrug as described above is used to test the methodology.

The second stage is preprocessing and consist of two steps: band-pass filtering and similarity filtering. The first step aims to increase the sensitivity of the algorithm to small structural changes while the second step aims to remove the operational variability from the measurements.

The third stage is feature extraction. The purpose of this stage is to generate proper inputs for the machine learning algorithms, called *features*. A damage-sensitive feature is some quantity extracted from the preprocessed data that indicates the presence (or not) of damage in a structure. This stage involves data compression and data fusion (i.e. combining information from multiple sensors) in an effort to obtain low dimensional feature vectors that are sensitive to small structural faults (e.g. added mass).

The next stage is the application of a machine learning algorithm that can decide the state of the structure based on the given feature vectors. Two types of algorithms are distinguished: a Support Vector Machine

(SVM) classifier and a novelty detection method. The algorithm differ in the type of data that is used for training the diagnostic. The SVM classifier uses labelled training data from every class (supervised learning), while the novelty detection uses only the data from the undamaged structure for training (unsupervised learning). The diagnostic of the latter algorithm is restricted to the lowest levels of damage identification—that of damage detection and possibly damage localization—while the former algorithm may progress to higher levels such as damage quantification.

The final stage is the decision, which is a matter of considering the outputs of the algorithms and deciding whether an alarm should be issued. In this case, the detection of the small added masses is the main goal.

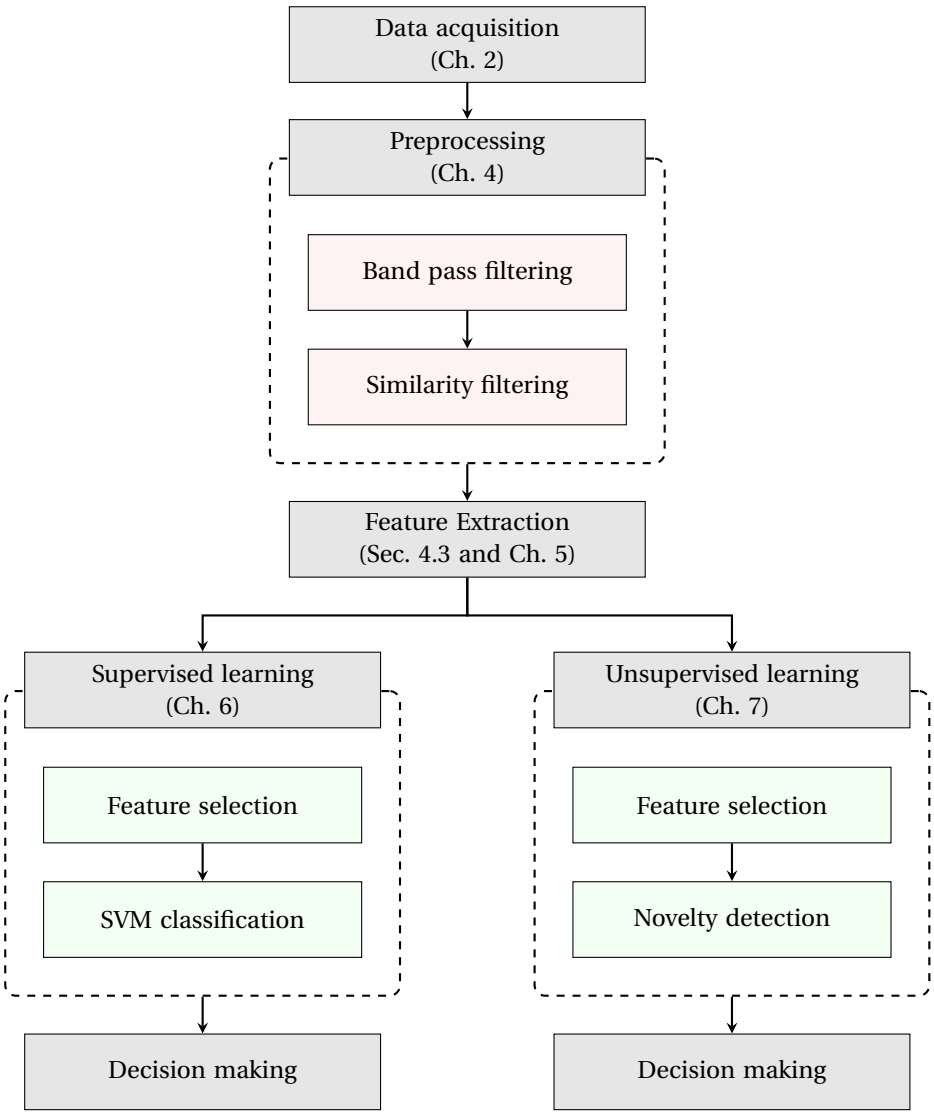


Figure 2.13: Overview methodology

3

Theory: Similarity Filtering and Frequency Domain Decomposition

3.1 Similarity Filtering

The aim of the thesis is to classify the different added masses based on acceleration data only. The features used for classification are extracted from the response spectra. In order to get clear spectra from the acceleration data, the data is preprocessed using a method called *similarity filtering*. The idea of similarity filtering is, roughly speaking, to amplify the similarities in the measurements and to damp the differences between the measurements. It is assumed that the dynamic properties of the structure do not change from one measurement to another (i.e. the system is assumed time-invariant) so that the characteristics of the system are amplified. On the other hand, the structure is excited by varying (unknown) traffic loads. The source of such a load might be a motorcycle or a heavy truck, each of them producing a considerably different signal. These differences are damped by the application of similarity filtering. The resulting signal is ideally a signal that clearly shows the characteristic response of the system.

The method of similarity filtering was developed by Milosevic in his thesis [32] (there it was called probabilistic filtering). The theory about similarity filtering presented in the next sections is largely based on the work of Milosevic. Some additional aspects are considered, and new notations are introduced that are more convenient to describe the method in its general sense.

3.1.1 Single-Input/Single-Output System

The methodology of similarity filtering is first explained for a Single-Input/Single-Output (SISO) system, also known as as a Single-Degree-of-Freedom (SDOF) system. First, the considered system is introduced, followed by an explanation of the method in both the time and frequency domain.

The considered system in this section is a linear time-invariant SDOF system with Impulse Response Function (IRF) $h(t)$ and Frequency Response Function (FRF) $H(\omega)$. The meaning of these functions is discussed in Appendix A.3, but it is repeated here that both functions contain the same information about the system, but in different domains. In fact, $h(t)$ and $H(\omega)$ form a Fourier pair. It is also emphasized that the parameters describing the system are constant (independent of time) as we assume a time-invariant system. For a mechanical system (e.g. a bridge) this means that the masses, springs and dampers are not changing with time. This is often a reasonable assumption during the time over which we measure, but over a long enough time span, very few systems are time invariant. For example, the characteristics of a bridge can change due to the changing temperature between day and night, or its characteristics can change (on a more long-term span) due to aging or fatigue of the structure [3].

The system is subjected to a single (unknown) force $f(t)$ and the response is denoted as $x(t)$. The response of the linear system is obtained as the convolution between the time history of the force and the impulse response function

$$x(t) = h(t) * f(t) = \int_{-\infty}^{\infty} h(\tau) f(t-\tau)\,\mathrm{d}\tau \tag{3.1}$$

in which the lower limit is essentially zero because the system only responds to past inputs (causal system).

Equivalently, the response of the system can be given in the frequency domain, which is obtain by the Fourier transform of equation (3.1) and by virtue of the convolution theorem

$$X(\omega) = H(\omega)F(\omega) \tag{3.2}$$

where $F(\omega)$ denotes the Fourier transform of the input $f(t)$ and $X(\omega)$ the Fourier transform of the response $x(t)$ (assuming these transforms exist). For similarity filtering we need the response of the system due to a wide range of variable input forces. Therefore, equation (3.1) and (3.2) are slightly adapted to indicate the different responses. Let $\{f(t)\} = \{f_1(t), f_2(t), \ldots\}$ and $\{x(t)\} = \{x_1(t), x_2(t), \ldots\}$ denote the set of forces and corresponding system responses, respectively, where $f_k(t)$ and $x_k(t)$ denote the k-th time history in the set. Then the time domain response, equation (3.1), becomes

$$x_k(t) = h(t) * f_k(t) \tag{3.3}$$

in which the IRF $h(t)$ is constant as we assume a time-invariant system. Analogue to this, the response in the frequency domain, equation (3.2), is reformulated. Note that the subscript k should not be confused with the subscript used in case of multiple input/output, where it refers to the input or output location. In general, the meaning of the subscript will be clear from the context.

Similarity filtering produces a new signal $y(t)$, also referred to as *processed signal*, which is obtained by successive convolution of multiple response time histories from the set $\{x(t)\}$. Mathematically this can be expressed as

$$\begin{aligned} y(t) &= x_1(t) * x_2(t) * \cdots * x_n(t) \\ &= \big(h(t) * f_1(t)\big) * \big(h(t) * f_2(t)\big) * \cdots * \big(h(t) * f_n(t)\big) \end{aligned} \tag{3.4}$$

where n denotes the number of signals involved, this number will be referred to as the *order of similarity filter*. Note that the number of convolutions performed is one less than the order of similarity filtering. Introducing the convolution operator $\circledast_{k=1}^{n}$ for n-fold convolution, equation (3.4) can be reformulated as

$$y(t) = \circledast_{k=1}^{n} x_k(t) = \circledast_{k=1}^{n} \big(h(t) * f_k(t)\big) = h^{*n}(t) * \left(\circledast_{k=1}^{n} f_k(t)\right) \tag{3.5}$$

where $h^{*n}(t)$ denotes the n-fold convolution of $h(t)$ with itself. The last equality in equation (3.5) follows from the fact that the convolution operation is commutative and associative.

The resulting convolution in equation (3.5) does not have a very clear interpretation. Therefore the same relation will be presented in the frequency domain. From the convolution theorem it follows that convolution in the time domain is equivalent to multiplication in the frequency domain. Hence, the convolution operation in equation (3.5) becomes a multiplication operation in the frequency domain, which can be expressed as

$$Y(\omega) = \prod_{k=1}^{n} X_k(\omega) = \prod_{k=1}^{n} H(\omega)F_k(\omega) = H(\omega)^n \cdot \prod_{k=1}^{n} F_k(\omega) \tag{3.6}$$

Note that the quantities in equation (3.6) are generally complex-valued, which can be written in terms of a real-valued amplitude and associated phase angle as

$$Y(\omega) = |Y(\omega)|\mathrm{e}^{i\varphi_Y(\omega)} \tag{3.7}$$

where

$$|Y(\omega)| = |H(\omega)|^n \cdot \prod_{k=1}^{n} |F_k(\omega)| \tag{3.8}$$

$$\varphi_Y(\omega) = n \cdot \varphi_H(\omega) + \sum_{k=1}^{n} \varphi_{F_k} \tag{3.9}$$

Equation (3.8) and (3.9) are referred to as the *amplitude spectrum* and then *phase spectrum*, respectively. The first term in equation (3.8) holds all information regarding the dynamical properties of the system whereas the last term describes the “average” frequency content of the excitation. The term “average" is related to the observation that if we take the n-th root of equation (3.6), we would obtain the geometric mean of the

frequency response functions (which is just the FRF $H(\omega)$) times the geometric mean of the forces. If the frequency content of each force in the set $\{f(t)\}$ is different, then the geometric mean of the frequency content will become more constant (i.e. differences in the spectrum are smoothen by averaging). This leads to the idea of similarity filtering in which successive convolution of different excitation time histories results in a white noise signal with constant intensity in the frequency domain. This idea is further elaborated in the following.

White noise

Up to now the excitation $f(t)$ was undefined, but we will now assume that the product $\prod_{k=1}^{n} F_k(\omega)$ tends to a constant intensity. In other words, we assume that multiple convolutions of variable (unknown) input time signals tends to approximate white noise. White noise is a random signal whose correlation function is of the form

$$R_w(\tau) = \mathbf{E}[w(t)\,w(t+\tau)] = C\delta(\tau) \tag{3.10}$$

in which $\mathbf{E}[\bullet]$ is the expectation operator and C is the intensity of the white noise process. The correlation function implies that two samples of a white noise process are not correlated when they are taken at different time instances; while the variance is infinite when two samples are taken at the same time instance. The Power Spectral Density (PSD) of a white noise process equals

$$S_w(\omega) = \int_{-\infty}^{\infty} R_w(\tau)\mathrm{e}^{-i\omega\tau}\,\mathrm{d}\tau = C\int_{-\infty}^{\infty} \delta(\tau)\mathrm{e}^{-i\omega\tau}\,\mathrm{d}\tau = C \tag{3.11}$$

This means that the PSD of a white noise process is real and constant, and has infinite bandwidth. This latter property is not feasible for describing most physical processes. Therefore, it is usually assumed that the white noise process has constant PSD over a finite frequency bandwidth. In order to apply the white noise concept in the methodology of the similarity filtering we will establish the PSD of the processed signal.

The power spectral density function of a random signal $x(t)$ is defined as [2, p. 127]

$$S_{xx}(\omega) = \lim_{T\to\infty} \frac{1}{T}\mathbf{E}\left[X(\omega,T)X(\omega,T)^*\right] = \lim_{T\to\infty} \frac{1}{T}\mathbf{E}\left[|X(\omega,T)|^2\right] \tag{3.12}$$

where $(\bullet)^*$ denotes the complex conjugate and $X(\omega,T)$ represent the finite Fourier transform of $x(t)$ with $t \in [0,T]$, which is defined as

$$X(\omega,T) = \int_0^T x(t)\mathrm{e}^{-i\omega t}\,\mathrm{d}t \tag{3.13}$$

This finite Fourier transform will exist for the (stationary) random data, whereas their infinite Fourier transform would not exist, because the random signal theoretically persist forever. In reality, the acquired signals are always finite hence the Fourier transform will exist. In fact, we can always compute the Discrete Fourier Transform (DFT) of a discrete signal (random or not) but should take care of bias introduced by sampling.

Using equation (3.12), the PSD of the processed signal $y(t)$ after similarity filtering is given by (omitting the limit for brevity)

$$S_{yy}(\omega) = \frac{1}{T}\mathbf{E}\left[|Y(\omega)|^2\right] = \frac{1}{T}\mathbf{E}\left[\prod_{k=1}^{n}|X_k(\omega)|^2\right] \tag{3.14}$$

where the last equality follows from the substitution of the relation for similarity filtering in the frequency domain, equation (3.6). It is assumed that the response signals in the set $\{x(t)\}$ are uncorrelated; that is, $\mathbf{E}[x_k(t)x_l(t)] = \mathbf{E}[x_k(t)]\mathbf{E}[x_l(t)]$ for $k \neq l$. Consequently, the amplitude spectra of these signals are also uncorrelated. Hence, equation (3.14) can be written as

$$S_{yy}(\omega) = \frac{1}{T}\prod_{k=1}^{n}\mathbf{E}\left[|X_k(\omega)|^2\right] \tag{3.15}$$

Let $S_{xx,k}(\omega) = \mathbf{E}\left[|X_k(\omega)|^2\right]/T_k$ denote the PSD of the k-th response signal with finite length T_k, then (3.15) becomes

$$S_{yy}(\omega) = c\prod_{k=1}^{n} S_{xx,k}(\omega) \tag{3.16}$$

where c is a scaling factor that depends on the individual length of the signals $x_k(t)$ and the total length of the processed signal. Let T_k denote the length of $x_k(t)$ and T denote the length of the processed signal $y(t)$, then

$$c = \frac{1}{T}\prod_{k=1}^{n} T_k \tag{3.17}$$

The PSD of a processed signal is thus simply proportional to the product of the power spectra of the response signals used for similarity filtering. The latter spectra can be written in terms of the frequency response function and the random inputs as

$$S_{xx,k}(\omega) = |H(\omega)|^2 S_{ff,k}(\omega) \tag{3.18}$$

where $S_{ff,k}(\omega)$ denotes the PSD of the random input signal $f_k(t)$, which is defined according to equation (3.12) as (again omitting the limit for brevity)

$$S_{ff,k}(\omega) = \frac{1}{T_k}\mathbf{E}\left[|F_k(\omega)|^2\right] \tag{3.19}$$

Substituting equation (3.18) into equation (3.16) yields the PSD of a processed signal in terms of the frequency response function and the PSDs of the random inputs

$$S_{yy}(\omega) = c\prod_{k=1}^{n}|H(\omega)|^2 S_{ff,k}(\omega) = c|H(\omega)|^{2n}\cdot\prod_{k=1}^{n} S_{ff,k}(\omega) \tag{3.20}$$

where the last equality follows from the assumption that the system is time invariant (i.e. the frequency response function is constant).

As mentioned before, it is assumed that by successive convolution of the random input signals in the set $\{f(t)\}$, the resulting signal tends to approximate a white noise signal $w(t)$, that is

$$\overset{n}{\underset{k=1}{*}} f_k(t) = f_1(t) * f_2(t) * \cdots * f_n(t) \approx w(t)$$

which implies that the product of the input spectra in equation (3.20) becomes approximately constant. Consequently, the PSD of the processed signal becomes proportional to the amplitude spectrum of the frequency response function raised to the order $2n$

$$S_{yy}(\omega) = c|H(\omega)|^{2n}\cdot\underbrace{\prod_{k=1}^{n} S_{ff,k}(\omega)}_{\text{constant}} \propto |H(\omega)|^{2n} \tag{3.21}$$

which illustrates the fundamental idea of similarity filtering; that is, by multiple convolutions of response measurements the system characteristics are amplified and the variations from the random excitation are removed. It must be noted that this is only true when the following two assumptions are satisfied: 1) the system is time invariant, and 2) the n-fold convolution of the input signals in the set $\{f(t)\}$ tends to approximate a white noise signal (at least in some sense). The relative short time window in the application of the similarity filtering ensures that the first assumption is satisfied. The second assumption is often more difficult to validate, because the excitation is not always directly measured. In general, the excitation can be considered as random signals, which are characterized by multivariate density functions. The random nature of the excitation makes a general proof of the assumption difficult. The white noise assumption was validated with some simple numerical examples by Milosevic in his thesis [32].

Ensemble averaging

The similarity filtering can be related to a method that is commonly applied to estimate the PSD of a random signal. To illustrate this, we take the n-th root of equation (3.16) and omit the constant c for the moment, which yields

$$\left(S_{yy}(\omega)\right)^{\frac{1}{n}} = \left(\prod_{k=1}^{n} S_{xx,k}(\omega)\right)^{\frac{1}{n}} \tag{3.22}$$

which is essentially a *geometric mean* of the power spectra of the response signals in the set $\{x(t)\}$. The resulting spectrum can be seen as an estimate of the power spectrum of the random signal $x(t)$. So, similarity

filtering yields some kind of average spectrum, which is very similar to the averaging methods that are often used in practice for estimating the power spectrum of a random signal.

In practice, the method of averaging provides a way to reduce the variance of an estimate in exchange for a reduction of frequency resolution. One of the simplest among these methods is Bartlett's method (also known as the method of averaged periodograms). In this method the random signal $x(t)$ is split into n segments, each of length T_k (thereby reducing the frequency resolution), and for each segment an estimate of the power spectrum is computed as

$$S_{xx,k}(\omega) = \frac{1}{T_k}|X_k(\omega)|^2 \tag{3.23}$$

which is also referred to a *periodogram*. A final estimate of the power spectrum of the random signal $x(t)$ is defined as the *arithmetic mean* of the periodograms

$$\hat{S}_{xx}(\omega) = \frac{1}{n}\sum_{k=1}^{n} S_{xx,k}(\omega) \tag{3.24}$$

The averaging reduces the variance, compared to a single estimate. Note that the periodogram in equation (3.23) can be obtained from the definition of the PSD (3.12) by dropping the limit and the expectation operator.

The estimates in equations (3.22) and (3.24) are very similar. The former uses the geometric mean to obtain an estimate of the power spectrum of the random signal $x(t)$, while the latter uses the arithmetic mean. Based on this observation we can state that similarity filtering is similar to Bartlett's method in which the arithmetic ensemble mean is replaced by the geometric ensemble mean. Note however that the two are not exactly equivalent because we have omitted the constant in equation (3.22).

3.1.2 Multiple-Input/Single-Output System

This section is concerned with the generalization of the fundamental idea of similarity filtering, as discussed in the previous section, to Multi-Degree-of-Freedom (MDOF) systems. MDOF systems are defined such that each response is caused by a linear combination of all the inputs, and there are no causal relations between any of the responses (i.e. one response does not have to be known to determine the other responses). The MDOF system can therefore be seen as a number of parallel Multiple-Input/Single-Output (MISO) systems. Since there are no causal relations between the responses, similarity filtering is applied to the response from each Degree-of-Freedom (DOF) separately. This means that response signals from different DOFs are not combined in the similarity filtering method. We can thus concentrate on looking at the MISO system, at least from a conceptual point of view. It is noted that computationally it may be more efficient to compute the entire MDOF system at once, but the understanding of the method is best illustrated by a MISO system. We will start with the introduction of matrix formulas for MDOF systems, from which the formulas for MISO systems are readily obtained.

Let $\mathbf{x}(t) \in \mathbb{R}^{p\times 1}$ be a column vector containing p response signals, and $\mathbf{f}(t) \in \mathbb{R}^{q\times 1}$ be a column vector containing q input signals

$$\mathbf{x}(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_p(t) \end{pmatrix}, \qquad \mathbf{f}(t) = \begin{pmatrix} f_1(t) \\ f_2(t) \\ \vdots \\ f_q(t) \end{pmatrix} \tag{3.25}$$

Then, the response vector $\mathbf{x}(t)$ is the result of convolution between the input vector $\mathbf{f}(t)$ and an *impulse response matrix* $\mathbf{h}_{xf}(t) \in \mathbb{R}^{p\times q}$, which in matrix form is formulated as

$$\mathbf{x}(t) = \mathbf{h}_{xf}(t) * \mathbf{f}(t) = \int_{-\infty}^{\infty} \mathbf{h}_{xf}(\tau)\mathbf{f}(t-\tau)\,\mathrm{d}\tau \tag{3.26}$$

in which the convolution operates analogue to a matrix-vector multiplication. The impulse response matrix $\mathbf{h}_{xf}(t)$ has the following form

$$\mathbf{h}_{xf}(t) = \begin{bmatrix} h_{11}(t) & h_{12}(t) & \dots & h_{1q}(t) \\ h_{21}(t) & h_{22}(t) & \dots & h_{2q}(t) \\ \vdots & \vdots & \ddots & \vdots \\ h_{p1}(t) & h_{p2}(t) & \dots & h_{pq}(t) \end{bmatrix} \tag{3.27}$$

in which the element $h_{ij}(t)$ is referred to as an impulse response function, which is the response at DOF i due to an impulse loading at DOF j and zero loading at all other DOFs. We will often drop the time variable t for brevity. As already mentioned we will concentrate on looking at a single response of the system, which will be denoted by $x_j(t)$ where the index j refers to a specific DOF; for example, this might be a sensor measuring the vertical acceleration at a specific location of a structure. From equation (3.26) it follows that the response of DOF j can be expressed as

$$x_j(t) = \mathbf{h}_{j*}(t) * \mathbf{f}(t) = \sum_{l=1}^{q} h_{jl}(t) * f_l(t) \qquad \text{for } j = 1,\dots,p \tag{3.28}$$

where the row vector $\mathbf{h}_{j*}(t) \in \mathbb{R}^{1\times q}$ denotes the j-th row of the impulse response matrix.

Let $\mathbf{X}(\omega) \in \mathbb{C}^{p\times 1}$ be a column vector representing the Fourier transform of the response signals $X_j(\omega) = \mathscr{F}\{x_j(t)\}$, $j = 1,\dots,p$, and $\mathbf{F}(\omega) \in \mathbb{C}^{q\times 1}$ be a column vector representing the Fourier transform of the input signals $F_j(\omega) = \mathscr{F}\{f_j(t)\}$, $j = 1,\dots,q$; that is

$$\mathbf{X}(\omega) = \begin{pmatrix} X_1(\omega) \\ X_2(\omega) \\ \vdots \\ X_p(\omega) \end{pmatrix}, \qquad \mathbf{F}(\omega) = \begin{pmatrix} F_1(\omega) \\ F_2(\omega) \\ \vdots \\ F_q(\omega) \end{pmatrix} \tag{3.29}$$

By virtue of the convolution theorem, equation (3.26) is transformed to the frequency domain

$$\mathbf{X}(\omega) = \mathbf{H}_{xf}(\omega)\mathbf{F}(\omega) \tag{3.30}$$

in which $\mathbf{H}_{xf}(\omega) \in \mathbb{C}^{p\times q}$ is the frequency response matrix defined as

$$\mathbf{H}_{xf}(\omega) = \begin{bmatrix} H_{11}(\omega) & H_{12}(\omega) & \dots & H_{1q}(\omega) \\ H_{21}(\omega) & H_{22}(\omega) & \dots & H_{2q}(\omega) \\ \vdots & \vdots & \ddots & \vdots \\ H_{p1}(\omega) & H_{p2}(\omega) & \dots & H_{pq}(\omega) \end{bmatrix} \tag{3.31}$$

The element $H_{ij}(\omega)$ is referred to as a frequency response function which is the Fourier transform of the impulse response function $h_{ij}(t)$. The frequency dependence will often be omitted for brevity. The frequency response of DOF j will be denoted as

$$X_j(\omega) = \mathbf{H}_{j*}(\omega)\mathbf{F}(\omega) = \sum_{l=1}^{q} H_{jl}(\omega)F_l(\omega) \qquad \text{for } j = 1,\dots,p \tag{3.32}$$

where the row vector $\mathbf{H}_{j*}(\omega) \in \mathbb{C}^{1\times q}$ denotes the j-th row of the frequency response matrix. Notice that the single response is caused by a linear combination of *all* inputs, weighted by the respective elements in the frequency response matrix, and that this response does not cause another response (i.e. no causal relations between any of the responses).

As for the SDOF system, we need the response of the system due to variable input forces, the latter being distributed over multiple DOFs. The set of forces is denoted by $\{\mathbf{f}(t)\} = \{\mathbf{f}_1(t), \mathbf{f}_2(t), \dots\}$ and the set of corresponding responses at DOF j is denoted by $\{x_j(t)\} = \{x_{j,1}(t), x_{j,2}(t), \dots\}$. The different signals are indicated with an additional index k, which is separated by a comma from the index that refers to the input or output DOF. For example, equation (3.28) for the k-th signal now becomes

$$x_{j,k}(t) = \mathbf{h}_{j*}(t) * \mathbf{f}_k(t) = \sum_{l=1}^{q} h_{jl}(t) * f_{l,k}(t) \qquad \text{for } j = 1,\dots,p \tag{3.33}$$

Analogue to this, the response in the frequency domain can be reformulated. It must be noted that the elements of the impulse response matrix and frequency response matrix do not change from one response to the other, as we assume a time-invariant system, and thus the index k is omitted.

Let $y_j(t)$ now denote a processed signal obtained by applying n-th order similarity filtering to the set of responses $\{x_{j,k}(t)\colon\ k=1,\dots,n\}$ from DOF j. This signal can be expressed as follows, where the time dependence is omitted for brevity

$$\begin{aligned} y_j(t) &= x_{j,1} * x_{j,2} * \cdots * x_{j,n} \\ &= \left(\sum_{l_1=1}^{q} h_{jl_1} * f_{l_1,1}\right) * \left(\sum_{l_2=1}^{q} h_{jl_2} * f_{l_2,2}\right) * \cdots * \left(\sum_{l_n=1}^{q} h_{jl_n} * f_{l_n,n}\right) \\ &= \sum_{l_1=1}^{q}\sum_{l_2=1}^{q}\cdots\sum_{l_n=1}^{q} \left(h_{jl_1} * f_{l_1,1}\right) * \left(h_{jl_2} * f_{l_2,2}\right) * \cdots * \left(h_{jl_n} * f_{l_n,n}\right) \\ &= \sum_{l_1=1}^{q}\sum_{l_2=1}^{q}\cdots\sum_{l_n=1}^{q} \left(h_{jl_1} * h_{jl_2} * \cdots * h_{jl_n}\right) * \left(f_{l_1,1} * f_{l_2,2} * \cdots * f_{l_n,n}\right) \end{aligned} \tag{3.34}$$

By virtue of the convolution theorem, equation (3.34) can be transformed to the frequency domain; that is, convolution in the time domain is equivalent to multiplication in the frequency domain. Letting $Y_j(\omega)$ denote the Fourier transform of $y_j(t)$ and omitting the frequency dependence for brevity, the transformation of equation (3.34) yields

$$\begin{aligned} Y_j(\omega) &= X_{j,1} \cdot X_{j,2} \cdots X_{j,n} \\ &= \left(\sum_{l_1=1}^{q} H_{jl_1} F_{l_1,1}\right) \cdot \left(\sum_{l_2=1}^{q} H_{jl_2} F_{l_2,2}\right) \cdots \left(\sum_{l_n=1}^{q} H_{jl_n} F_{l_n,n}\right) \\ &= \sum_{l_1=1}^{q}\sum_{l_2=1}^{q}\cdots\sum_{l_n=1}^{q} \left(H_{jl_1} F_{l_1,1}\right) \cdot \left(H_{jl_2} F_{l_2,2}\right) \cdots \left(H_{jl_n} F_{l_n,n}\right) \\ &= \sum_{l_1=1}^{q}\sum_{l_2=1}^{q}\cdots\sum_{l_n=1}^{q} \left(H_{jl_1} H_{jl_2} \cdots H_{jl_n}\right) \cdot \left(F_{l_1,1} F_{l_2,2} \cdots F_{l_n,n}\right) \end{aligned} \tag{3.35}$$

Equation (3.34) and (3.35) show that it is possible to isolate the excitation term in the same way as for the SDOF system, although the notation is more complex than before. In fact, from the last expression in equation (3.35) we observe that it is possible to isolate two n-dimensional arrays with q^n components. The components of the first array are the products $H_{jl_1} H_{jl_2} \cdots H_{jl_n}$ and the components of the second array are the products $F_{l_1,1} F_{l_2,2} \cdots F_{l_n,n}$ where the indices $l_1, l_2, \dots, l_n$ range from 1 to q. The frequency response $Y_j(\omega)$ is then the sum of the products of the corresponding components in the two arrays. To illustrate this, equation (3.35) is expanded for $q=2$ and $n=2$ (i.e. the system is excited at two DOFs and second order similarity filtering is considered)

$$\begin{aligned} Y_{2,j}(\omega) &= \sum_{l_1=1}^{2}\sum_{l_2=1}^{2} \left(H_{jl_1} H_{jl_2}\right) \cdot \left(F_{l_1,1} F_{l_2,2}\right) \\ &= \left(H_{j1} H_{j1}\right) \cdot \left(F_{1,1} F_{1,2}\right) + \left(H_{j1} H_{j2}\right) \cdot \left(F_{1,1} F_{2,2}\right) + \left(H_{j2} H_{j1}\right) \cdot \left(F_{2,1} F_{1,2}\right) \\ &\quad + \left(H_{j2} H_{j2}\right) \cdot \left(F_{2,1} F_{2,2}\right) \end{aligned} \tag{3.36}$$

where the subscript 2 in the left hand side indicates the corresponding order of similarity filtering; in general this subscript will be omitted. The equation is reformulated as the following matrix equation

$$Y_{2,j}(\omega) = \boldsymbol{H}_{2,j}(\omega) : \boldsymbol{F}_2(\omega) \tag{3.37}$$

where the two-dimensional arrays (matrices) are defined as

$$\boldsymbol{H}_{2,j}(\omega) = \begin{bmatrix} H_{j1} H_{j1} & H_{j1} H_{j2} \\ H_{j2} H_{j1} & H_{j2} H_{j2} \end{bmatrix}_{2\times 2} \quad , \qquad \boldsymbol{F}_2(\omega) = \begin{bmatrix} F_{1,1} F_{1,2} & F_{1,1} F_{2,2} \\ F_{2,1} F_{1,2} & F_{2,1} F_{2,2} \end{bmatrix}_{2\times 2} \tag{3.38}$$

The "double dot" product (denoted by **:**) in equation (3.37) implies the summation of the products of the corresponding components of two matrices having the same size (analogue to the dot product for vectors).

Increasing the order of similarity filtering leads to higher-order tensors that takes the form of multi-dimensional arrays. Third order similarity filtering yields a third order tensor equation that implies the following summation (expanding equation (3.35) with $n = 3$)

$$\begin{aligned} Y_{3,j}(\omega) &= \sum_{l_1=1}^{2}\sum_{l_2=1}^{2}\sum_{l_3=1}^{2}\left(H_{jl_1}H_{jl_2}H_{jl_3}\right)\cdot\left(F_{l_1,1}F_{l_2,2}F_{l_3,3}\right) \\ &= \left(H_{j1}H_{j1}H_{j1}\right)\cdot\left(F_{1,1}F_{1,2}F_{1,3}\right) + \left(H_{j2}H_{j1}H_{j1}\right)\cdot\left(F_{2,1}F_{1,2}F_{1,3}\right) \\ &\quad + \left(H_{j1}H_{j2}H_{j1}\right)\cdot\left(F_{1,1}F_{2,2}F_{1,3}\right) + \left(H_{j2}H_{j2}H_{j1}\right)\cdot\left(F_{2,1}F_{2,2}F_{1,3}\right) \\ &\quad + \left(H_{j1}H_{j1}H_{j2}\right)\cdot\left(F_{1,1}F_{1,2}F_{2,3}\right) + \left(H_{j2}H_{j1}H_{j2}\right)\cdot\left(F_{2,1}F_{1,2}F_{2,3}\right) \\ &\quad + \left(H_{j1}H_{j2}H_{j2}\right)\cdot\left(F_{1,1}F_{2,2}F_{2,3}\right) + \left(H_{j2}H_{j2}H_{j2}\right)\cdot\left(F_{2,1}F_{2,2}F_{2,3}\right) \end{aligned} \tag{3.39}$$

which can be reformulated as the following tensor equation, similar to equation (3.37)

$$Y_{3,j}(\omega) = \boldsymbol{H}_{3,j}(\omega) \bullet \boldsymbol{F}_3(\omega) \tag{3.40}$$

where the third-order tensors (three-dimensional arrays) $\boldsymbol{H}_{3,j}(\omega)$ and $\boldsymbol{F}_3(\omega)$ are defined as

$$\boldsymbol{H}_{3,j}(\omega) = \left[\begin{pmatrix} H_{j1}H_{j1}H_{j1} & H_{j1}H_{j2}H_{j1} \\ H_{j2}H_{j1}H_{j1} & H_{j2}H_{j2}H_{j1} \end{pmatrix}, \begin{pmatrix} H_{j1}H_{j1}H_{j2} & H_{j1}H_{j2}H_{j2} \\ H_{j2}H_{j1}H_{j2} & H_{j2}H_{j2}H_{j2} \end{pmatrix}\right]_{2\times2\times2} \tag{3.41}$$

$$\boldsymbol{F}_3(\omega) = \left[\begin{pmatrix} F_{1,1}F_{1,2}F_{1,3} & F_{1,1}F_{2,2}F_{1,3} \\ F_{2,1}F_{1,2}F_{1,3} & F_{2,1}F_{2,2}F_{1,3} \end{pmatrix}, \begin{pmatrix} F_{1,1}F_{1,2}F_{2,3} & F_{1,1}F_{2,2}F_{2,3} \\ F_{2,1}F_{1,2}F_{2,3} & F_{2,1}F_{2,2}F_{2,3} \end{pmatrix}\right]_{2\times2\times2} \tag{3.42}$$

The "big dot" product (denoted by •) generalizes the dot product for vectors to higher-order tensors. For three-dimensional arrays this product must be interpreted as a "triple dot" product. Observe that the dimension of the arrays is equal to the order of similarity filtering while the shape is determined by the number of inputs.

The notation with multidimensional arrays can be extended to the general case of q input DOFs and order n similarity filtering (i.e. the convolution of n response measurements). However, this leads to a rather complex equation that is not easy to handle in further computations. Therefore, the Kronecker notation will be introduced in which the components of a potentially multi-dimensional array are ordered in a block matrix, which enables one to use the more convenient matrix notation.

Generalization using Kronecker notation

One difficulty in describing the similarity filtering for MDOF systems is the choice of an appropriate notation. We found that for single-input systems, the expressions describing the method of similarity filtering simply contain scalar quantities. For multiple-input systems, we found a matrix equation when the order of similarity filtering is $n = 2$; this was illustrated for the case that the system is excited at two DOFs resulting in 2×2 matrices. However it can be shown that this holds for any number of inputs, resulting in $q \times q$ matrices. Increasing the order of the similarity filtering leads to multi-dimensional arrays. These arrays take the form of higher-order tensors and we could refer to its components using appropriate indices. However, the algebra for higher-order tensors is rather abstract and will be avoided. Kronecker algebra is used instead for which the basic concepts are presented in Appendix B.2. The Kronecker product is denoted by ⊗ and operates on two matrices of arbitrary size resulting in a *block matrix*.

We will now rewrite the result in equation (3.39) using the Kronecker notation to illustrate its application. Using the matrix formulation from equation (3.32), the third-order similarity filtering in the frequency domain is expressed as

$$Y_{3,j}(\omega) = X_{j,1} \cdot X_{j,2} \cdot X_{j,3} = \left(\mathbf{H}_{j*}\mathbf{F}_1\right)\cdot\left(\mathbf{H}_{j*}\mathbf{F}_2\right)\cdot\left(\mathbf{H}_{j*}\mathbf{F}_3\right) \tag{3.43}$$

where $\mathbf{H}_{j*}$ is a row vector and $\mathbf{F}_k$, $k = 1,2,3$ is a column vector, both having a length of $q = 2$ (corresponding to two different input DOFs). Notice that the last term in equation (3.43) is a product of three vector products which, according to Kronecker's mixed-product property for vectors (B.14), can be rewritten as

$$Y_{3,j}(\omega) = \left(\mathbf{H}_{j*} \otimes \mathbf{H}_{j*} \otimes \mathbf{H}_{j*}\right)\cdot\left(\mathbf{F}_1 \otimes \mathbf{F}_2 \otimes \mathbf{F}_3\right) \tag{3.44}$$

The Kronecker products of the vectors between the brackets are again vectors with a length of 2^3, which are referred to as *block vectors*. Lets introduce new variables for the two block vectors that results from the Kronecker products between the brackets

$$\hat{\mathbf{H}}_{3,j}(\omega) = \mathbf{H}_{j*} \otimes \mathbf{H}_{j*} \otimes \mathbf{H}_{j*} \tag{3.45a}$$

$$\hat{\mathbf{F}}_3(\omega) = \mathbf{F}_1 \otimes \mathbf{F}_2 \otimes \mathbf{F}_3 \tag{3.45b}$$

where the block vectors $\hat{\mathbf{H}}_{3,j}(\omega)$ and $\hat{\mathbf{F}}_3(\omega)$ have dimensions 1×8 and 8×1, respectively. The subscript 3 in the notation of a block vector refers to the order of similarity filtering and must not be confused with the indices used to refer to the components of a matrix. It is noted that $\hat{\mathbf{H}}_{3,j}(\omega)$ contains every component of the multi-dimensional arrays that was defined in (3.41), and $\hat{\mathbf{F}}_3(\omega)$ contains every component of the multi-dimensional arrays that was defined in (3.42). The only difference is that the components are now arranged in vectors of length 2^3 instead of in arrays of dimension $2 \times 2 \times 2$. Using the previously defined variables for the Kronecker products, equation (3.44) can be reformulated as

$$Y_{3,j}(\omega) = \hat{\mathbf{H}}_{3,j}(\omega)\hat{\mathbf{F}}_3(\omega) \tag{3.46}$$

which shows that the frequency response of a processed signal can be expressed in two terms; one related to the system characteristics and one related to the excitation of the system. Hence, it is possible to isolate the excitation term in the same way as we did for the SDOF system.

The previous can be extended to the general case of n-th order similarity filtering for a system with q inputs at different DOFs, leading to the following definition for the frequency response of the processed signal

$$Y_j(\omega) = \hat{\mathbf{H}}_{n,j}(\omega)\hat{\mathbf{F}}_n(\omega) \tag{3.47}$$

where the block vectors $\hat{\mathbf{H}}_{n,j}(\omega) \in \mathbb{C}^{1\times q^n}$ and $\hat{\mathbf{F}}_n(\omega) \in \mathbb{C}^{q^n\times 1}$ are formed from all possible combinations of the components in the vectors from different observations (indicated with the subscript k); that is, the components of the block vectors have the form

$$H_{jl_1}(\omega) \cdot H_{jl_2}(\omega) \cdots H_{jl_n}(\omega)$$

and

$$F_{l_1,1}(\omega) \cdot F_{l_2,2}(\omega) \cdots F_{l_n,n}(\omega)$$

respectively, where all possible combinations of the indices $l_1, l_2, \ldots, l_n$ ranging from 1 to q are included. The length of the block vectors increases thus with the order of similarity filtering n and the number of input DOFs q; that is the vectors consist of q^n components. The block vectors are in general defined as

$$\hat{\mathbf{H}}_{n,j}(\omega) = \mathbf{H}_{j*} \otimes \mathbf{H}_{j*} \otimes \cdots \otimes \mathbf{H}_{j*} = \mathbf{H}_{j*}^{[n]} \tag{3.48}$$

$$\hat{\mathbf{F}}_n(\omega) = \mathbf{F}_1 \otimes \mathbf{F}_2 \otimes \cdots \otimes \mathbf{F}_n = \bigotimes_{k=1}^{n} \mathbf{F}_k \tag{3.49}$$

where $\mathbf{H}_{j*}^{[n]}$ denotes the n-fold Kronecker product of the row vector $\mathbf{H}_{j*}$ with itself, this product is also known as the n-th Kronecker power. The operator $\bigotimes_{k=1}^{n}$ is used to indicate the n-fold Kronecker product of matrices or vectors that are not constant. Hence, the definition in equation (3.47) can alternatively be expressed as

$$Y_j(\omega) = \mathbf{H}_{j*}^{[n]}(\omega) \bigotimes_{k=1}^{n} \mathbf{F}_k(\omega) \tag{3.50}$$

which is very similar to equation (3.6); the result of similarity filtering in the frequency domain in case of a SDOF system. In fact, if $q = 1$ (i.e. only one input DOF) the Kronecker products of vectors reduce to multiplications of scalar quantities.

The remarkable result in equation (3.50) for a MDOF system is very similar to the expression for $Y(\omega)$ in equation (3.6) for a SDOF system, showing the correspondence of similarity filtering between the two systems. Moreover, the result for $Y_j(\omega)$ in equation (3.47) is very similar to equation (3.32) for $X_j(\omega)$. The only difference is that the former involves block vectors resulting from the Kronecker product of multiple observations. Note that for $q = 1$ the result in equation (3.50) reduces to the result for a SDOF system; that is, the vectors become scalars and the Kronecker product becomes a normal multiplication.

Finally, by considering a different row of the frequency response matrix (i.e. by changing the index j) processed signals for all p output locations are obtained. Notice that the excitation block vector (3.49) remains the same for all output locations.

3.1.3 PSD of Processed Signal

Similarity filtering will be used as a preprocessing step before the actual damage detection, to enhance the changes in structural properties caused by damage. In the following chapters, we will use the principal values of the spectral density matrix to detect these changes and thus potential faults in the structure. Therefore, it is interesting at this point to see what the spectral density matrix of a processed signal will look like. Moreover, the main assumption for similarity filtering relates to the input spectra, as we will assume that these become approximately constant for a sufficient high order.

The PSD function of a random signal $x(t)$ is already defined in equation (3.12). Similarly, the Cross Spectral Density (CSD) function of two random signals $x(t)$ and $f(t)$ is defined as

$$S_{xf}(\omega) = \lim_{T\to\infty} \frac{1}{T}\mathbf{E}\left[X(\omega)F(\omega)^*\right] \tag{3.51}$$

where $\mathbf{E}[\bullet]$ is the expectation operator, and $X(\omega)$ and $F(\omega)$ are the *finite* Fourier transforms as defined in equation (3.13). For multiple-input/multiple-output systems, the input and output processes could be collected in vectors $\mathbf{f}(t)$ and $\mathbf{x}(t)$, respectively. The power spectral density matrix and the cross spectral density matrix are then defined as

$$\mathbf{S}_{xx}(\omega) = \lim_{T\to\infty} \frac{1}{T}\mathbf{E}\left[\mathbf{X}(\omega)\mathbf{X}(\omega)^{\mathrm{H}}\right] \tag{3.52}$$

$$\mathbf{S}_{ff}(\omega) = \lim_{T\to\infty} \frac{1}{T}\mathbf{E}\left[\mathbf{F}(\omega)\mathbf{F}(\omega)^{\mathrm{H}}\right] \tag{3.53}$$

$$\mathbf{S}_{xf}(\omega) = \lim_{T\to\infty} \frac{1}{T}\mathbf{E}\left[\mathbf{X}(\omega)\mathbf{F}(\omega)^{\mathrm{H}}\right] \tag{3.54}$$

where $(\bullet)^{\mathrm{H}}$ denotes the complex conjugate transpose. Then from the relation between input and output $\mathbf{X}(\omega) = \mathbf{H}(\omega)\mathbf{F}(\omega)$ we can derive the following relations between the spectral density matrices and the frequency response matrix

$$\mathbf{S}_{xx}(\omega) = \mathbf{H}(\omega)\mathbf{S}_{ff}(\omega)\mathbf{H}(\omega)^{\mathrm{H}} \tag{3.55}$$

$$\mathbf{S}_{xf}(\omega) = \mathbf{H}(\omega)\mathbf{S}_{ff}(\omega) \tag{3.56}$$

The cross spectral density function between the response of DOF j and the response of DOF l is defined as

$$S_{jl}(\omega) = \mathbf{H}_{j*}(\omega)\mathbf{S}_{ff}(\omega)\mathbf{H}_{l*}(\omega)^{\mathrm{H}} \tag{3.57}$$

where the row vectors $\mathbf{H}_{j*}(\omega)$ and $\mathbf{H}_{l*}(\omega)$ correspond to the j-th and the l-th row of the frequency response matrix, respectively.

Now, the spectral density matrix $\mathbf{S}_{yy}(\omega)$ for the processed signals in the vector $\mathbf{y}(t)$ will be derived. For this, it is sufficient to consider the CSD for two processed signals $y_j(t)$ and $y_l(t)$ for two arbitrary DOFs j and l. The PSD of the processed signal $y_j(t)$ is then obtained for the particular case that $l = j$ (i.e. diagonal elements of the matrix). The two processed signals are obtained by similarity filtering using the response signals from the corresponding DOF; that is, the set $\{x_{j,k}(t)\colon k = 1,\ldots,n\}$ is used to compute $y_j(t)$ and the set $\{x_{l,k}(t)\colon k = 1,\ldots,n\}$ is used to compute $y_l(t)$. The CSD for two processed signals $y_j(t)$ and $y_l(t)$ is defined as

$$S_{y_j y_l}(\omega) = \frac{1}{T}\mathbf{E}\left[Y_j(\omega)\,Y_l(\omega)^*\right] \tag{3.58}$$

where $Y_j(\omega)$ and $Y_l(\omega)$ are the the Fourier transforms of the processed signals, which can be expressed as the product of the Fourier transforms of the response signals, as shown in the first expression of equation (3.35). Hence, the CSD in equation (3.58) can be expanded as

$$\begin{aligned} S_{y_j y_l}(\omega) &= \frac{1}{T}\mathbf{E}\left[\left(X_{j,1}(\omega)X_{j,2}(\omega)\cdots X_{j,n}(\omega)\right)\cdot\left(X_{l,1}(\omega)^* X_{l,2}(\omega)^*\cdots X_{l,n}(\omega)^*\right)\right] \\ &= \frac{1}{T}\mathbf{E}\left[\left(X_{j,1}(\omega)X_{l,1}(\omega)^*\right)\cdot\left(X_{j,2}(\omega)X_{l,2}(\omega)^*\right)\cdots\left(X_{j,n}(\omega)X_{l,n}(\omega)^*\right)\right] \\ &= \frac{1}{T}\mathbf{E}\left[X_{j,1}(\omega)X_{l,1}(\omega)^*\right]\cdot\mathbf{E}\left[X_{j,2}(\omega)X_{l,2}(\omega)^*\right]\cdots\mathbf{E}\left[X_{j,n}(\omega)X_{l,n}(\omega)^*\right] \end{aligned} \tag{3.59}$$

The last equality implies that the response signals from different time moments are uncorrelated so that the expectation of the products is the product of the expectations. The cross spectrum of two response signals $x_{j,k}(t)$ and $x_{l,k}(t)$ both with length T_k is defined as

$$S_{x_j x_l,k}(\omega) = \frac{1}{T_k}\mathbf{E}\left[X_{j,k}(\omega)X_{l,k}(\omega)^*\right] \tag{3.60}$$

Hence, the last expression in equation (3.59) can be written as

$$S_{y_j y_l}(\omega) = c \cdot S_{x_j x_l,1}(\omega) \cdot S_{x_j x_l,2}(\omega) \cdots S_{x_j x_l,n}(\omega) = c \prod_{k=1}^{n} S_{x_j x_l,k}(\omega) \tag{3.61}$$

where c is a scaling factor that depends on the length T_k of the individual response signals and the length T of the processed signals. This factor was earlier defined in equation (3.17). The cross spectrum of two processed signals is thus proportional to the product of the cross spectra of the response signals that are used for similarity filtering. The same holds for the power spectrum of a processed signal which is obtained by setting $l = j$. Notice that this was also found for the SDOF system considered previously.

The relation between the spectra of the processed signals and the spectra of the forces that excite the structure can be found upon substitution of equation (3.47) into equation (3.58), which yields

$$S_{y_j y_l}(\omega) = \frac{1}{T}\mathbf{E}\left[Y_j(\omega)Y_l(\omega)^{\mathrm{H}}\right] = \frac{1}{T}\mathbf{E}\left[\hat{\mathbf{H}}_{n,j}(\omega)\hat{\mathbf{F}}_n(\omega)\hat{\mathbf{F}}_n(\omega)^{\mathrm{H}}\hat{\mathbf{H}}_{n,l}(\omega)^{\mathrm{H}}\right] \tag{3.62}$$

Substituting the definition of the block vector $\hat{\mathbf{F}}_n(\omega)$ (3.49) into the last equation and omitting the frequency dependence for brevity yields

$$\begin{aligned} S_{y_j y_l}(\omega) &= \hat{\mathbf{H}}_{n,j}\frac{1}{T}\mathbf{E}\left[(\mathbf{F}_1 \otimes \mathbf{F}_2 \otimes \cdots \otimes \mathbf{F}_n)\cdot(\mathbf{F}_1 \otimes \mathbf{F}_2 \otimes \cdots \otimes \mathbf{F}_n)^{\mathrm{H}}\right]\hat{\mathbf{H}}_{n,l}^{\mathrm{H}} \\ &= \hat{\mathbf{H}}_{n,j}\frac{1}{T}\mathbf{E}\left[\mathbf{F}_1{\mathbf{F}_1}^{\mathrm{H}} \otimes \mathbf{F}_2{\mathbf{F}_2}^{\mathrm{H}} \otimes \cdots \otimes \mathbf{F}_n\mathbf{F}_n^{\mathrm{H}}\right]\hat{\mathbf{H}}_{n,l}^{\mathrm{H}} \\ &= \hat{\mathbf{H}}_{n,j}\frac{1}{T}\mathbf{E}\left[\mathbf{F}_1{\mathbf{F}_1}^{\mathrm{H}}\right] \otimes \mathbf{E}\left[\mathbf{F}_2{\mathbf{F}_2}^{\mathrm{H}}\right] \otimes \cdots \otimes \mathbf{E}\left[\mathbf{F}_n\mathbf{F}_n^{\mathrm{H}}\right]\hat{\mathbf{H}}_{n,l}^{\mathrm{H}} \end{aligned} \tag{3.63}$$

where the second equality follows from the mixed-product property of the Kronecker product (B.14), which is used to change the order of the vector products. The last equality implies that the input signals from different time moments are uncorrelated so that the expectation of the Kronecker product is the Kronecker product of the expectations. Using the definition of the spectral density matrix for the input vectors $\mathbf{f}_k(t)$, $k = 1, \ldots, n$ (3.53), the last equation can be rewritten as

$$S_{y_j y_l}(\omega) = c \cdot \hat{\mathbf{H}}_{n,j}\left(\mathbf{S}_{ff,1} \otimes \mathbf{S}_{ff,2} \otimes \cdots \otimes \mathbf{S}_{ff,n}\right)\hat{\mathbf{H}}_{n,l}^{\mathrm{H}} \tag{3.64}$$

where the scaling factor c is given by equation (3.17). The result in equation (3.64) is similar to the relation in equation (3.57) and shows that the CSD of two processed signals can be written in terms of the spectral density matrices associated with the force vectors of the different observations and two block vectors constructed from the rows of the frequency response matrix. Again, it is assumed that by successive convolution of multiple response signals, the part related to the random excitation tends to approximate white noise. Consequently, the Kronecker product in between the brackets in equation (3.64) tend to approximate a constant matrix $\mathbf{C} \in \mathbb{C}^{q^n \times q^n}$; that is

$$\bigotimes_{k=1}^{n}\mathbf{S}_{ff,k}(\omega) = \left(\mathbf{S}_{ff,1} \otimes \mathbf{S}_{ff,2} \otimes \cdots \otimes \mathbf{S}_{ff,n}\right) \approx \mathbf{C} \tag{3.65}$$

Consequently, the cross-spectrum in equation (3.64) becomes

$$S_{y_j y_l}(\omega) = c \cdot \hat{\mathbf{H}}_{n,j}\underbrace{\left(\mathbf{S}_{ff,1} \otimes \mathbf{S}_{ff,2} \otimes \cdots \otimes \mathbf{S}_{ff,n}\right)}_{\text{constant}}\hat{\mathbf{H}}_{n,l}^{\mathrm{H}} \approx \hat{\mathbf{H}}_{n,j}(\omega)\mathbf{C}\hat{\mathbf{H}}_{n,l}(\omega)^{\mathrm{H}} \tag{3.66}$$

The result obtained here is not so straightforward as the result that was found in equation (3.21) for the SDOF system, but the concept is the same; the spectrum of the processed signal becomes "proportional" to the FRFs. Although for the SDOF system there is only one FRF characterizing the system, in the present case

there are multiple FRFs, one for each input-output pair (e.g. if two outputs are considered at DOF j and l and q inputs, there are $2q$ different FRFs given by $H_{jk}(\omega)$, $H_{lk}(\omega)$, $k = 1,\ldots,q$).

To illustrate how the resulting spectrum will look like, equation (3.64) is evaluated for the case that the system has two inputs and the similarity filtering is performed with two different observations; that is $q = 2$ and $n = 2$

$$S_{y_j y_l}(\omega) = \hat{\mathbf{H}}_{2,j}\left(\mathbf{S}_{ff,1} \otimes \mathbf{S}_{ff,2}\right)\hat{\mathbf{H}}_{2,l}^{\mathrm{H}} \tag{3.67}$$

which applies to any combination of indices j and l, being the DOFs where the responses are measured. The vectors $\hat{\mathbf{H}}_{2,j}$ and $\hat{\mathbf{H}}_{2,l}$ are given by equation (3.48), and for DOF j reads as

$$\hat{\mathbf{H}}_{2,j}(\omega) = \mathbf{H}_{j*} \otimes \mathbf{H}_{j*} = \begin{bmatrix} H_{j1}H_{j1} & H_{j1}H_{j2} & H_{j2}H_{j1} & H_{j2}H_{j2} \end{bmatrix} \tag{3.68}$$

which consists of the elements of the j-th row in the FRF matrix (omitting the frequency dependency). The spectral density matrix $\mathbf{S}_{ff,k}$ can be estimated as

$$\hat{\mathbf{S}}_{ff,k}(\omega) = \frac{1}{T}\mathbf{F}_k\mathbf{F}_k^{\mathrm{H}} = \frac{1}{T}\begin{bmatrix} F_{1,k}F_{1,k}^* & F_{1,k}F_{2,k}^* \\ F_{2,k}F_{1,k}^* & F_{2,k}F_{2,k}^* \end{bmatrix} \tag{3.69}$$

which is simply obtained by omitting the limit and expectation operator in the definition of the spectral density matrix (3.53). Subsequently, expanding the Kronecker product in equation (3.67) yields the following 4×4 matrix

$$\hat{\mathbf{S}}_{ff,1} \otimes \hat{\mathbf{S}}_{ff,2} = \frac{1}{T}\begin{bmatrix} F_{1,1}F_{1,1}^*\hat{\mathbf{S}}_{ff,2} & F_{1,1}F_{2,1}^*\hat{\mathbf{S}}_{ff,2} \\ F_{2,1}F_{1,1}^*\hat{\mathbf{S}}_{ff,2} & F_{2,1}F_{2,1}^*\hat{\mathbf{S}}_{ff,2} \end{bmatrix} \tag{3.70}$$

The elements in this matrix are composed of different combinations of the Fourier transforms (and their complex conjugate) of the forces from different DOFs and from distinct observations, indicated by the two indices. The number of combinations (and thus the size of the matrix) increases for increasing order of similarity filtering. In the general case where the number of inputs is q and the order of similarity filtering is n, the matrix has dimensions $q^n \times q^n$. The fundamental idea of similarity filtering is that the elements in the latter matrix become constant.

3.1.4 Practical Matters: Segments

So far we have assumed time-continuous signals of *infinite* length, hence the limits in the convolution integrals are infinite. However, in practice we have finite measurements and thus the convolution integral can only be evaluated over a finite time interval. Suppose we have measured the response of a structure over a certain period of time, and repeated this measurement n_m times. The response measurements are collected in a set denoted by $Z = \{u_k(t,T_k)\colon k = 1,2,\ldots,n_m\}$, where T_k denotes the measurement window of the k-th response signal. The response signals in the set are subsequently processed with similarity filtering to filter out the operational variability. The method of similarity filtering is based on successive convolution of the response signals, and the basic formation of n-th order similarity filtering can be written as

$$y(t,T) = u_1(t,T_1) * u_2(t,T_2) * \cdots * u_n(t,T_n) \tag{3.71}$$

where $y(t,T)$ denotes the processed signal of length T. Notice that the measurement window T_k might vary between different measurement and thus the input signals in equation (3.71) might have different length. Although this is not a problem for the convolution operation, the operation will be computationally very expensive when the measurement window is very long. Therefore, it is chosen to split the measurement into consecutive segments of T_s seconds (with $T_s < T_k$) and perform the similarity filtering using these smaller time segments. This means that the signals in the set of response measurements, Z, are split into a number of segments, each with length T_s seconds. Assuming that each measurement can be split into n_s segments of T_s seconds, a new set of signals is obtained denoted by $S = \{x_k(t,T_s)\colon k = 1,2,\ldots,n_s \cdot n_m\}$ and equation (3.71) becomes

$$y(t,T) = x_1(t,T_s) * x_2(t,T_s) * \cdots * x_n(t,T_s) \tag{3.72}$$

The question now is, what should be the length of the segments? To answer this question three aspects are considered:

1. the computational cost of similarity filtering;

2. the time scale on which similarity filtering operates;

3. the expected natural frequencies of the structure.

The first aspect is merely related to the amount of data that is necessary to produce one processed signal $y(t,T)$. From equation (3.72) it follows that for n-th order similarity filtering, n segments of T_s seconds are used meaning that for one processed signal $T = n \cdot T_s$ seconds of data is required. So, longer segments means that there is more data needed for processing. In fact, the amount of data also depends on the number of segments (i.e. the order of similarity filtering). Hence, from this point of view it makes more sense to choose the length of the segments in combination with the order of similarity filtering. Although, the discussion here is limited to the choice of a suitable length of the segments, it is emphasized that the final selection of the parameters n and T_s will be based on an analysis of the combination of both parameters. The analysis that is performed to select suitable parameters for the data considered in this thesis will be discussed in Chapter 4.

Recall that the purpose of similarity filtering is, loosely speaking, to amplify the similarities in the data and at the same time damp the differences in the data to improve the performance of damage detection. For the bridge under consideration, it is assumed that the differences are primarily related to variations in traffic load (operational variability), and that the similarities are related to the underlying system, which is assumed to be linear and time-invariant (see theory). The latter assumption is often reasonable during a short time window, but over a longer time span, very few real systems are time-invariant. The characteristics of the bridge, for example, can change under varying environmental conditions such as temperature. Hence, we must ensure that the data of a processed signal spans over a short enough time window so that the system can be considered time-invariant, and at the same time, the operational conditions will change from one time segment to another. In general, the time scale on which the characteristics of the system change due to changing environmental conditions are much larger than the time scale of the operational variability. For example, the passage of a vehicle over the bridge segment with sensors takes less than a second, while the change of environmental conditions (and thus the system properties) can take minutes to an hour; temperature changes in less than a second are rarely the case. Therefore, it is concluded that the length of a time segment T_s can best be chosen in the order of magnitude of seconds.

The smaller the segments, the less data is required and the less expensive is the computational cost to process one signal according to equation (3.72). However, the length of the signals must not be chosen too short, which brings us to the third aspect: the expected natural frequencies of the structure. The frequency components of interest must be well represented in a time segment; that is, the time segment must properly captures the dynamic behavior of the system, and thus a sufficient number of cycles of the lowest frequency component of interest must be included. Moreover, one should take care of the following, the length of the segments governs the frequency resolution of the processed signal and the ability to distinguish between closely spaced modal frequencies. In fact, the successive convolutions performed in similarity filtering "smoothen" the Fast Fourier Transform (FFT) of a signal. When very short segments are used, then a large amount of smoothing is applied and closely-spaced frequency components are possibly merged into one single peak and can no longer be distinguished from each other.

This is illustrated using the synthetic signal shown in Figure 3.1, which has two dominant frequency components at 5.0 Hz and 5.1 Hz and a smaller frequency component at 0.5 Hz. The signal has a length of 100 seconds and is sampled with a sample rate of 100 Hz resulting in $N = 10000$ samples. The amplitude spectrum shown in Figure 3.1b is computed with a N-point FFT using all samples, resulting in a frequency resolution of 0.01 Hz. To separate the two closely-spaced components, the frequency resolution should be at least 0.1 Hz, which means that we should use a time window of at least 10 seconds (i.e. 1000 samples) in the FFT. The signal will be split into a number of segments, which are then convolved with each other (as in similarity filtering). Two cases are considered:

1. The signal is split into 10 segments of 10 s length (i.e. 1000 samples per segment), yielding a frequency resolution of 0.1 Hz.

2. The signal is split into 20 segments of 5 s length (i.e. 500 samples per segment), yielding a frequency resolution of 0.2 Hz.

The intermediate results after 1, 2, 3, 8 and 9 convolutions are shown in Figures 3.2a and 3.2b for the first and second case, respectively. From the evolution of the amplitude spectrum during successive convolution it is clear from that the second case should be avoided as it is no longer possible to separate the closely spaced frequency components from each other.

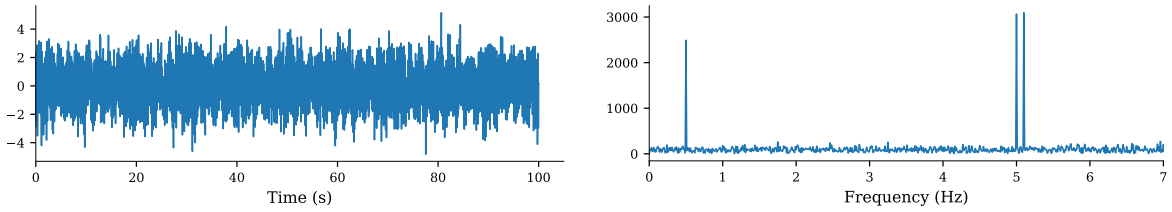


Figure 3.1: Three superimposed harmonics embedded in random noise, presented in (a) the time domain, and (b) the frequency domain. The two closely spaced frequency components are located 0.1 Hz apart at 5.0 and 5.1 Hz, respectively.

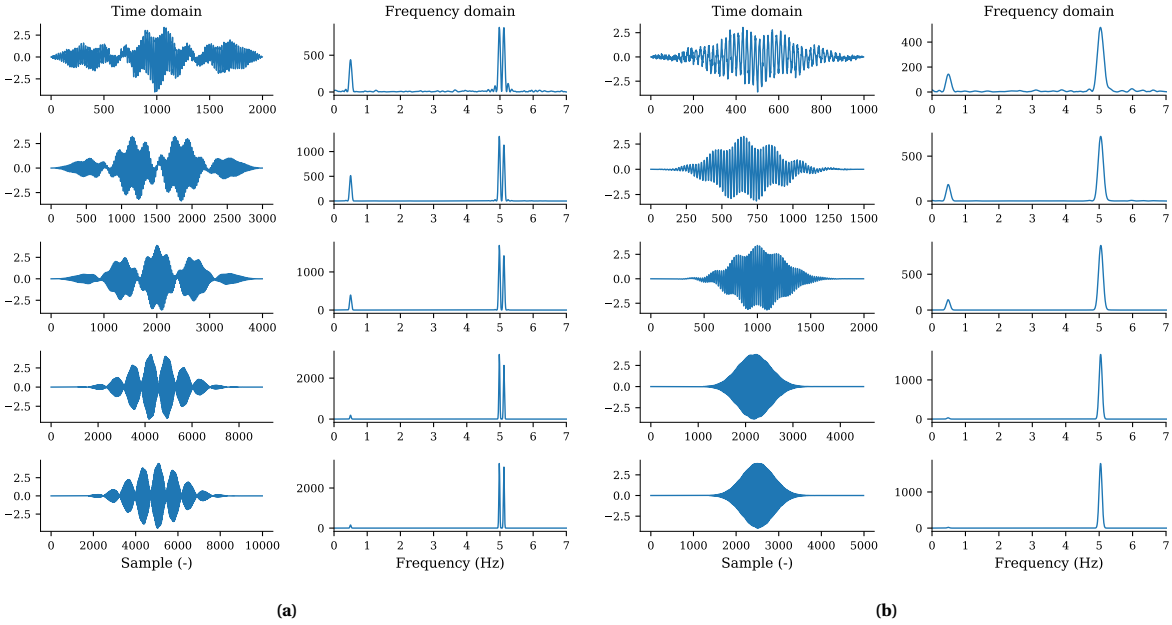


Figure 3.2: Evolution of the time signal and corresponding amplitude spectrum for successive convolutions of (a) 10 second segments and (b) 5 second segments. The figures from the top to the bottom show intermediate results for 1, 2, 3, 8 and 9 convolutions.

To summarize, short time segments are used for similarity filtering rather than the raw signal length. The length of the time segments must be chosen in the order of magnitude of the time scale of the operational variability, which is generally much smaller than the time scale of the environmental variability.

3.1.5 Summary

In this chapter we have presented the theory of similarity filtering for linear time-invariant systems. The aim of similarity filtering is to remove the variations in the response measurements that are induced by changing operational conditions (e.g. varying traffic load). To achieve this, multiple response measurements (from a single sensor) are convolved with each other. The idea behind this operation is that by multiple convolutions, the similarities (system characteristics) are amplified and the differences (operational conditions) are damped. Ideally, the resulting signal represent a response signal of the system that one would obtain if the system is excited by white noise. The input signals being white noise is a common assumption in many system identification algorithms (e.g. frequency domain decomposition).

The main assumptions of the similarity filtering are:

- The system is Linear Time-Invariant (LTI);
- By multiple convolutions of response measurements, the part related to the random excitation tends to approximate white noise.

General

In general it is assumed that we have a set of response measurements, denoted by $\{x_{j,k}(t)\colon k = 1,\dots,n\}$ where index j indicates the DOF which is measured and index k indicates the measurement number. The variable n denotes the number of signals used for processing and defines the order of similarity filtering. The basic formation of n-th order similarity filtering for the response at DOF j is

$$y_j(t) = x_{j,1}(t) * x_{j,2}(t) * \cdots * x_{j,n}(t) = \overset{n}{\underset{k=1}{\ast}} x_{j,k}(t) \tag{3.73}$$

which by virtue of the convolution theorem can be transformed to the frequency domain

$$Y_j(\omega) = X_{j,1}(\omega) \cdot X_{j,2}(\omega) \cdots X_{j,n}(\omega) = \prod_{k=1}^{n} X_{j,k}(\omega) \tag{3.74}$$

The multiplication in the frequency domain is more appealing than the convolution in the time domain, hence most of the relations are presented in the frequency domain.

SDOF system

To show the idea of similarity filtering a linear time-invariant system was introduced and the formulation of similarity filtering was written in terms of the system characteristics and excitation. For a SDOF system with frequency response function $H(\omega)$, where the k-th response can be written as $X_k(\omega) = H(\omega)F_k(\omega)$, the following relation was obtained for the processed signal in the frequency domain

$$Y(\omega) = H(\omega)^n \cdot \prod_{k=1}^{n} F_k(\omega) \tag{3.6 revisited}$$

which shows that the (unknown) excitation can be isolated from the system characteristics. It is assumed that by multiple convolutions of the response measurements, the part related to the excitation tends to approximate a white noise signal $w(t)$. A white noise signal is characterized by a constant PSD. Therefore the PSD of the processed signal was derived and it was shown that if the assumption holds, the PSD becomes proportional to the amplitude spectrum of the FRF raised to the order $2n$

$$S_{yy}(\omega) = c|H(\omega)|^{2n} \cdot \underbrace{\prod_{k=1}^{n} S_{ff,k}(\omega)}_{\text{constant}} \propto |H(\omega)|^{2n} \tag{3.21 revisited}$$

which illustrates the fundamental idea of similarity filtering; that is, by multiple convolutions of response measurements the system characteristics are amplified and the variations from the random excitation are removed.

MDOF system

The idea of similarity filtering was then extended to a MDOF system with q inputs and p outputs. Since there are no causal relations between any of the outputs, similarity filtering is applied to the response of each DOF separately. A response of the system at DOF j due to input forces at q different DOFs can be expressed as

$$X_{j,k}(\omega) = \mathbf{H}_{j*}(\omega)\mathbf{F}_k(\omega) \tag{3.75}$$

where $\mathbf{H}_{j*} \in \mathbb{C}^{1\times q}$ is the j-th row of the FRF matrix and the index k indicates the different observations that are used for similarity filtering. The *Kronecker product*, denoted by $\otimes$, was introduced to obtain convenient notations of the results for a MDOF system. Using this product, a processed signal can be expressed in the frequency domain as

$$Y_j(\omega) = \mathbf{H}_{j*}^{[n]}(\omega) \bigotimes_{k=1}^{n} \mathbf{F}_k(\omega) \tag{3.50 revisited}$$

with

$$\mathbf{H}_{j*}^{[n]}(\omega) = \mathbf{H}_{j*} \otimes \mathbf{H}_{j*} \otimes \cdots \otimes \mathbf{H}_{j*} = \hat{\mathbf{H}}_{n,j}(\omega)$$
$$\bigotimes_{k=1}^{n} \mathbf{F}_k(\omega) = \mathbf{F}_1 \otimes \mathbf{F}_2 \otimes \cdots \otimes \mathbf{F}_n = \hat{\mathbf{F}}_n(\omega)$$

where $\mathbf{H}_{j*}^{[n]}$ denotes the n-fold Kronecker product of the row vector $\mathbf{H}_{j*}$ with itself. The operator $\bigotimes_{k=1}^{n}$ is used to indicate the n-fold Kronecker product of matrices or vectors that are not constant. Two new variables were introduced to denote the *block vectors* that result from the Kronecker product; $\hat{\mathbf{H}}_{n,j}(\omega) \in \mathbb{C}^{1\times q^n}$ and $\hat{\mathbf{F}}_n(\omega) \in \mathbb{C}^{1\times q^n}$, respectively. The components of these vectors have the form

$$H_{jl_1}(\omega) \cdot H_{jl_2}(\omega) \cdots H_{jl_n}(\omega)$$

and

$$F_{l_1,1}(\omega) \cdot F_{l_2,2}(\omega) \cdots F_{l_n,n}(\omega)$$

respectively, where all possible combinations of the indices $l_1, l_2, \ldots, l_n$ ranging from 1 to q are included. Hence, the length of the block vectors increases with the order of similarity filtering n and the number of input DOFs q; that is the vectors consist of q^n components.

Subsequently the cross-spectral density function between two processed signals $y_j(t)$ and $y_l(t)$ was derived

$$S_{y_j y_l}(\omega) = c \cdot \hat{\mathbf{H}}_{n,j} \underbrace{\left(\mathbf{S}_{ff,1} \otimes \mathbf{S}_{ff,2} \otimes \cdots \otimes \mathbf{S}_{ff,n}\right)}_{\text{constant}} \hat{\mathbf{H}}_{n,l}^{\mathrm{H}} \approx \hat{\mathbf{H}}_{n,j}(\omega)\mathbf{C}\hat{\mathbf{H}}_{n,l}(\omega)^{\mathrm{H}} \tag{3.66 revisited}$$

where it was again assumed that by multiple convolutions of the response measurements the part related to the random excitation tends to approximate white noise, implying that the spectrum is constant. The correspondence with the SDOF system is very clear, in fact when $q = 1$ (i.e. only one input DOF) the result in equation (3.50 revisited) reduces to (3.6 revisited), and the result in equation (3.66 revisited) reduces to (3.21 revisited).

3.2 Frequency Domain Decomposition

The Frequency Domain Decomposition (FDD) is an output-only system identification technique developed by Brincker et al. [7]. The FDD method is based on the fact that if a lightly damped system is excited by white noise inputs, its modes can be identified using the spectral densities of the output signals. It is a non-parametric method that estimates the modal parameters of a system directly using some basic signal processing.

The idea behind FDD is to perform a decomposition of the system responses into a set of independent SDOF systems, one for each mode. The decomposition can directly be obtained with the Singular Value Decomposition (SVD) of the spectral density matrix. Plotting the singular values as function of frequency yields a so-called *singular value spectrum*. This spectrum can be seen as an overlaid plot of the auto-spectral density functions of the SDOF systems, whose resonance frequencies can be observed from the peaks in the spectrum. So, the modes of the system are identifiable by simply looking at the singular value spectrum. By picking the peaks in the spectrum, estimates of the natural frequencies can be obtained together with the corresponding mode shape. The FDD technique is able to identify closely spaced modes, and effectively deals with noise in the signals. However, since the information from a single frequency line is used, it is not possible to estimate the modal damping ratios directly with this method. For this reason the Enhanced Frequency Domain Decomposition (EFDD) method has been developed [26].

The FDD technique is based on the following assumptions regarding the unknown input and the system:

- The system is Linear Time-Invariant (LTI);
- The system is lightly damped;
- The input signals are white noise and uncorrelated in space.

Violation of one of these assumption does not mean that the method cannot be applied, but one must be careful with the interpretation of the results. For example, if the input signals are not white noise or contain some harmonic components, spurious peaks might occur in the singular value spectrum that are not related to the system characteristics.

3.2.1 Modal Decomposition

The explanation of the FDD technique starts with the modal decomposition of the output spectral density matrix. Let $\mathbf{x}(t) \in \mathbb{R}^{q\times 1}$ denote the vector with q unknown inputs and $\mathbf{y}(t) \in \mathbb{R}^{p\times 1}$ denote the vector with the p measured displacement responses of the system. The relation between the unknown inputs and the measured displacements can be expressed as [2, p. 241]

$$\mathbf{S}_{yy}(\omega) = \mathbf{H}(\omega)\mathbf{S}_{xx}(\omega)\mathbf{H}(\omega)^{\mathrm{H}} \tag{3.76}$$

where $\mathbf{H}(\omega) \in \mathbb{C}^{p\times q}$ is the force-displacement Frequency Response Function (FRF) matrix, $\mathbf{S}_{xx}(\omega) \in \mathbb{C}^{q\times q}$ and $\mathbf{S}_{yy}(\omega) \in \mathbb{C}^{p\times p}$ are the input and output Power Spectral Density (PSD) matrix, respectively, and $(\bullet)^{\mathrm{H}}$ denotes the complex conjugate transpose. The FRF matrix can be expressed in partial fraction form via poles λ_r and residue matrices $\mathbf{R}_r$ [3, p. 136]

$$\mathbf{H}(\omega) = \sum_{r=1}^{M} \frac{\mathbf{R}_r}{i\omega - \lambda_r} + \frac{\mathbf{R}_r^*}{i\omega - \lambda_r^*} \tag{3.77}$$

with

$$\lambda_r = -\sigma_r + i\omega_{dr}$$

where M is the number of modes of interest, λ_r is the pole of the r-th mode, σ_r is the modal damping (decay rate) and ω_{dr} is the damped natural frequency of the r-th mode. $\mathbf{R}_r \in \mathbb{C}^{p\times q}$ is the residue matrix corresponding to the r-th pole, which can be expressed as

$$\mathbf{R}_r = Q_r \boldsymbol{\theta}_r \boldsymbol{\gamma}_r^{\mathrm{T}} \tag{3.78}$$

where $\boldsymbol{\theta}_r \in \mathbb{C}^{p\times 1}$, $\boldsymbol{\gamma}_r \in \mathbb{C}^{q\times 1}$ and Q_r are the r-th mode shape, modal participation vector and modal scaling factor, respectively. The latter depends on the normalization of the mode shapes [3, p. 139]. When all

output measurements points are taken as reference (i.e. $q = p$), then $\mathbf{H}(\omega)$ is a square matrix and $\boldsymbol{\gamma}_r = \boldsymbol{\theta}_r$. Substituting equation (3.77) into equation (3.76) yields

$$\mathbf{S}_{yy}(\omega) = \left(\sum_{r=1}^{M} \frac{\mathbf{R}_r}{i\omega - \lambda_r} + \frac{\mathbf{R}_r^*}{i\omega - \lambda_r^*} \right) \mathbf{S}_{xx}(\omega) \left(\sum_{s=1}^{M} \frac{\mathbf{R}_s^{\mathrm{H}}}{-i\omega - \lambda_s^*} + \frac{\mathbf{R}_s^{\mathrm{T}}}{-i\omega - \lambda_s} \right) \tag{3.79}$$

Multiply the two partial fraction factors and assume that the input is white noise (i.e. constant input PSD $\mathbf{S}_{xx}(\omega) = \mathbf{S}_{xx}$). Subsequently, using Heaviside partial fraction theorem (cover-up method), the output PSD matrix can be reduced to the following pole-residue form (see Appendix D for derivation)

$$\mathbf{S}_{yy}(\omega) = \sum_{r=1}^{M} \frac{\mathbf{A}_r}{i\omega - \lambda_r} + \frac{\mathbf{A}_r^{\mathrm{H}}}{-i\omega - \lambda_r^*} + \frac{\mathbf{A}_r^*}{i\omega - \lambda_r^*} + \frac{\mathbf{A}_r^{\mathrm{T}}}{-i\omega - \lambda_r} \tag{3.80}$$

where $\mathbf{A}_r \in \mathbb{C}^{p \times p}$ is the residue matrix of the PSD matrix corresponding to the r-th pole λ_r, which is defined as

$$\mathbf{A}_r = \sum_{s=1}^{M} \frac{\mathbf{R}_r \mathbf{S}_{xx} \mathbf{R}_s^{\mathrm{H}}}{-\lambda_r - \lambda_s^*} + \frac{\mathbf{R}_r \mathbf{S}_{xx} \mathbf{R}_s^{\mathrm{T}}}{-\lambda_r - \lambda_s} \tag{3.81}$$

Using the definition for $\mathbf{R}_r$, equation (3.78), the residue matrix of the PSD matrix can be expressed in terms of modal quantities as

$$\mathbf{A}_r = \boldsymbol{\theta}_r \underbrace{\sum_{s=1}^{M} \frac{Q_r \boldsymbol{\gamma}_r^{\mathrm{T}} \mathbf{S}_{xx} Q_s^* \boldsymbol{\gamma}_s^* \boldsymbol{\theta}_s^{\mathrm{H}}}{-\lambda_r - \lambda_s^*} + \frac{Q_r \boldsymbol{\gamma}_r^{\mathrm{T}} \mathbf{S}_{xx} Q_s \boldsymbol{\gamma}_s \boldsymbol{\theta}_s^{\mathrm{T}}}{-\lambda_r - \lambda_s}}_{\triangleq \boldsymbol{\chi}_r^{\mathrm{T}}} = \boldsymbol{\theta}_r \boldsymbol{\chi}_r^{\mathrm{T}} \tag{3.82}$$

where $\boldsymbol{\chi}_r \in \mathbb{C}^{p \times q}$ is the *stochastic modal participation vector* of mode r. This vector depends on the modal parameters of *all* modes and the PSD matrix of the unknown excitation (assumed to be white noise). Hence, the output PSD matrix can be rewritten as

$$\mathbf{S}_{yy}(\omega) = \sum_{r=1}^{M} \frac{\boldsymbol{\theta}_r \boldsymbol{\chi}_r^{\mathrm{T}}}{i\omega - \lambda_r} + \frac{\boldsymbol{\chi}_r^* \boldsymbol{\theta}_r^{\mathrm{H}}}{-i\omega - \lambda_r^*} + \frac{\boldsymbol{\theta}_r^* \boldsymbol{\chi}_r^{\mathrm{H}}}{i\omega - \lambda_r^*} + \frac{\boldsymbol{\chi}_r \boldsymbol{\theta}_r^{\mathrm{T}}}{-i\omega - \lambda_r} \tag{3.83}$$

which is the modal decomposition of an output PSD matrix in case of displacement measurements. A similar expression can be derived for accelerations, see for instance [36, Eq. A.8].

Narrow-banded spectrum

As seen from equation (3.83) the output PSD matrix can be decomposed into four terms expressed by the modal parameters $\{\lambda_r, \boldsymbol{\theta}_r, \boldsymbol{\chi}_r\}$. Approximating this PSD matrix for spectral lines in the vicinity of a modal frequency (i.e. $\omega \to \omega_{dr}$), then only the first two terms in equation (3.83) are dominating [5]

$$\mathbf{S}_{yy}(\omega) = \sum_{r=1}^{S} \frac{\boldsymbol{\theta}_r \boldsymbol{\chi}_r^{\mathrm{T}}}{i\omega - \lambda_r} + \frac{\boldsymbol{\chi}_r^* \boldsymbol{\theta}_r^{\mathrm{H}}}{-i\omega - \lambda_r^*} \tag{3.84}$$

where S denotes the number of significantly contributing modes, typically one or two modes. The contribution from mode r to the modal participation vector $\boldsymbol{\chi}_r$ can be derived from equation (3.82) as

$$\boldsymbol{\chi}_r^T = \frac{Q_r \boldsymbol{\gamma}_r^{\mathrm{T}} \mathbf{S}_{xx} Q_r^* \boldsymbol{\gamma}_r^* \boldsymbol{\theta}_r^{\mathrm{H}}}{2\sigma_r} + \frac{Q_r \boldsymbol{\gamma}_r^{\mathrm{T}} \mathbf{S}_{xx} Q_r \boldsymbol{\gamma}_r \boldsymbol{\theta}_r^{\mathrm{T}}}{-2\lambda_r} \tag{3.85}$$

where σ_r is minus the real part of the pole $\lambda_r = -\sigma_r + i\omega_{dr}$. For lightly damped structures (i.e. $\sigma_r \ll \omega_{dr}$) the first term in equation (3.85) is dominating and thus the modal participation vector becomes proportional to the mode shape vector

$$\boldsymbol{\chi}_r^{\mathrm{T}} \approx \underbrace{\frac{|Q_r| \boldsymbol{\gamma}_r^{\mathrm{T}} \mathbf{S}_{xx} \boldsymbol{\gamma}_r^*}{2\sigma_r}}_{\triangleq c_r} \boldsymbol{\theta}_r^{\mathrm{H}} = c_r \boldsymbol{\theta}_r^{\mathrm{H}} \tag{3.86}$$

where c_r is a real-valued scaling factor. Substituting this result into equation (3.84) yields

$$\mathbf{S}_{yy}(\omega) \approx \sum_{r=1}^{S} \frac{c_r \boldsymbol{\theta}_r \boldsymbol{\theta}_r^{\mathrm{H}}}{i\omega - \lambda_r} + \frac{c_r \boldsymbol{\theta}_r \boldsymbol{\theta}_r^{\mathrm{H}}}{-i\omega - \lambda_r^*} = \sum_{r=1}^{S} \Re\left\{ \frac{2c_r}{i\omega - \lambda_r} \right\} \boldsymbol{\theta}_r \boldsymbol{\theta}_r^{\mathrm{H}} \tag{3.87}$$

where $\Re\{\bullet\}$ denotes the real-part of its complex argument. Equation (3.87) can be rewritten in matrix form as

$$\mathbf{S}_{yy}(\omega) \approx \boldsymbol{\Theta}\,\mathrm{diag}\left(\Re\left\{\frac{2c_r}{i\omega-\lambda_r}\right\}\right)\boldsymbol{\Theta}^{\mathrm{H}} \tag{3.88}$$

where $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_S) \in \mathbb{C}^{p\times S}$ is the mode shape matrix that contains only the mode shapes of dominating modes as columns. The diagonal matrix in the middle is real-valued and contains the poles of dominating modes. This diagonal matrix can be thought of as an power spectral density matrix of uncorrelated modal coordinates, analogue to the observation in Section 3.2.2.

The contribution to the spectral density matrix from a single mode can be expressed as

$$\Delta\mathbf{S}_{yy,r}(\omega) \approx \Re\left\{\frac{2c_r}{i\omega-\lambda_r}\right\}\boldsymbol{\theta}_r\boldsymbol{\theta}_r^{\mathrm{H}} = \alpha_r(\omega)\boldsymbol{\theta}_r\boldsymbol{\theta}_r^{\mathrm{H}} \tag{3.89}$$

where $\alpha_r(\omega)$ is a real-valued function which can be thought of as a PSD function of the modal coordinate of mode r. Substituting the pole $\lambda_r = -\sigma_r + i\omega_{dr}$ yields

$$\alpha_r(\omega) = \Re\left\{\frac{2c_r}{i\omega-\lambda_r}\right\} = \frac{2c_r\sigma_r}{\sigma_r^2 + (\omega-\omega_{dr})^2} \tag{3.90}$$

which shows that the spectrum $\Delta\mathbf{S}_{yy,r}(\omega)$ reaches a local maximum at the *damped* natural frequencies $\omega = \omega_{dr}$.

It should be noted that if the decomposition is only conducted in a narrow frequency band, the requirement of white noise excitation spectrum, can be reduced to broadband excitation; that is, approximately flat spectrum in the frequency band around the considered mode.

3.2.2 Uncorrelated Modal Coordinates

The decomposition of the spectral density matrix can also be seen as a result of uncorrelated modal coordinates. In fact, any dynamic response of a linear system can be written in modal coordinates as

$$\mathbf{y}(t) = \boldsymbol{\Phi}\mathbf{q}(t) = \sum_{r=1}^{M}\boldsymbol{\phi}_r q_r(t) \tag{3.91}$$

where $\boldsymbol{\Phi} = \left(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_M\right) \in \mathbb{R}^{p\times M}$ is the *real-valued* mode shape matrix and $\mathbf{q}(t) \in \mathbb{R}^{M\times 1}$ is a column vector with the modal coordinates. Note that in contrast to previous section, the mode shapes must be real-valued here to ensure that the response in the left hand side of equation (3.91) is real valued (hence the different symbol for the mode shapes).

For zero-mean stationary responses $\mathbf{y}(t)$ the covariance matrix (or correlation matrix) is given by the following relation

$$\mathbf{C}_{yy}(\tau) = \mathbf{E}\left[\mathbf{y}(t+\tau)\mathbf{y}(t)^{\mathrm{T}}\right] \tag{3.92}$$

where $\mathbf{E}[\bullet]$ is the expectation operator. Substituting the modal coordinates form of equation (3.91) into the covariance matrix leads to

$$\mathbf{C}_{yy}(\tau) = \mathbf{E}\left[\boldsymbol{\Phi}\mathbf{q}(t+\tau)\mathbf{q}(t)^{\mathrm{T}}\boldsymbol{\Phi}^{\mathrm{T}}\right] = \boldsymbol{\Phi}\mathbf{E}\left[\mathbf{q}(t+\tau)\mathbf{q}(t)^{\mathrm{T}}\right]\boldsymbol{\Phi}^{\mathrm{T}} = \boldsymbol{\Phi}\mathbf{C}_{qq}(\tau)\boldsymbol{\Phi}^{\mathrm{T}} \tag{3.93}$$

where $\mathbf{C}_{qq}(\tau) \in \mathbb{R}^{M\times M}$ is the covariance matrix of the modal coordinates. If we assume that the modal coordinates are uncorrelated, then all off-diagonal elements in $\mathbf{C}_{qq}(\tau)$ are zero, and thus $\mathbf{C}_{qq}(\tau)$ is diagonal holding the auto-covariance functions of the modal coordinates.

Now, transforming equation (3.93) into the frequency domain by applying the Fourier transform, we obtain the corresponding output PSD matrix (Wiener-Khinchin theorem), which can be expressed as

$$\mathbf{S}_{yy}(\omega) = \boldsymbol{\Phi}\mathbf{S}_{qq}(\omega)\boldsymbol{\Phi}^{\mathrm{T}} = \sum_{r=1}^{M} S_{qq,r}(\omega)\boldsymbol{\phi}_r\boldsymbol{\phi}_r^{\mathrm{T}} \tag{3.94}$$

where $\mathbf{S}_{qq}(\omega) \in \mathbb{R}^{M\times M}$ is the PSD matrix of the modal coordinates. Since the covariance matrix of the modal coordinates is assumed to be diagonal, the PSD matrix $\mathbf{S}_{qq}(\omega)$ will also be diagonal with positive real components $S_{qq,r}(\omega)$.

For the general case with complex mode shapes a similar result can be derived, see for instance [5].

3.2.3 Spectral Matrix Decomposition

In Section 3.2.1 and Section 3.2.2 it was shown that the output PSD matrix can be decomposed into the mode shapes of the system and the PSD functions of the modal coordinates associated with each mode. In both sections we found a decomposition of the following form

$$\mathbf{S}_{yy}(\omega) = \mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^{\mathrm{H}} \tag{3.95}$$

The factorization $\mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^{\mathrm{H}}$ at a discrete frequency $\omega = \omega_k$ can be described as a SVD where $\boldsymbol{\Sigma}$ is a diagonal matrix holding the singular values and $\mathbf{U}$ is a unitary matrix holding the singular vectors corresponding to the singular values.

In general, the SVD of an arbitrary (real or complex) matrix $\mathbf{A} \in \mathbb{C}^{m\times n}$ is given by

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{H}} = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\mathrm{H}} \tag{3.96}$$

where $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m) \in \mathbb{C}^{m\times m}$ and $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n) \in \mathbb{C}^{n\times n}$ are unitary matrices holding the left and right singular vectors, respectively, and $\boldsymbol{\Sigma} \in \mathbb{C}^{m\times n}$ is a "diagonal" matrix holding the nonzero singular values in descending order; that is $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$. The rank r of matrix $\mathbf{A}$ is the number of nonzero diagonal elements of $\boldsymbol{\Sigma}$, or in practice due to the presence of noise; the number of elements large enough to be considered nonzero. For a positive definite Hermitian matrix (i.e. all eigenvalues are positive and $\mathbf{A} = \mathbf{A}^{\mathrm{H}}$) the left and right singular vectors are equal, hence $\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{H}} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^{\mathrm{H}}$.

Notice that instead of the SVD we could also use the Eigenvalue Decomposition (EVD) as the basis for the FDD. This can be obtained by post-multiplying equation (3.95) by $\mathbf{U}$ and using the fact that $\mathbf{U}$ is unitary (i.e. $\mathbf{U}^{-1} = \mathbf{U}^{\mathrm{H}}$)

$$\mathbf{S}_{yy}\mathbf{U} = \mathbf{U}\boldsymbol{\Sigma} \tag{3.97}$$

which we could recognize as an eigenvalue problem. However, in practice the result is not exactly the same, as SVD normally sorts the singular values in descending order and the EVD usually does not, furthermore the singular vectors and eigenvectors tolerate some arbitrary phases that may not be the same [5, p. 620].

Modal identification

As explained previously, the key is to conduct a factorization like equation (3.95) of the spectral density matrix at discrete frequencies $\omega_k \in [0, 2\pi f_s/2]$. When the frequency approaches a modal frequency $\omega_k \to \omega_{dr}$ and in case of well-separated modes, the spectral density matrix approximates to a rank one matrix as

$$\underset{\omega_k\to\omega_{dr}}{\mathbf{S}_{yy}(\omega_k)} \approx \sigma_1(\omega_k)\mathbf{u}_1(\omega_k)\mathbf{u}_1(\omega_k)^{\mathrm{H}} \tag{3.98}$$

where $\sigma_1(\omega_k)$ and $\mathbf{u}_1(\omega_k)$ are the first singular value and first singular vector, respectively, computed at discrete frequency ω_k. Comparing equation (3.98) with the contribution of a single mode to the PSD matrix, equation (3.89), it is observed that the singular values reaches a local maximum at the damped natural frequency. Therefore, modes can be identified as peaks in the singular value spectrum; which is obtained by plotting the singular values as a function of frequency. The frequency at a local maximum is an estimate of the natural frequency, denoted by ω_r, and the associated singular vector is an estimate of the mode shape with unitary normalization

$$\widehat{\boldsymbol{\phi}}_r = \mathbf{u}_1(\omega_r) \tag{3.99}$$

If two or more modes have about the same natural frequency, the rank of the PSD matrix will be two or more at that frequency. Consequently, equation (3.98) contains two or more terms and the closely-spaced (or repeated) modes can be identified from two or more distinct singular values. This means that we have to plot an appropriate number of singular values as function of frequency and identify possible modes as peaks in these curves.

It must be noted that not all peaks in singular value spectrum indicate modes [39]. Errors such as noise, leakage, non-linearity and a cross singular value effect can also generate apparent peaks. The cross singular value effect is an computational characteristic that develops due to the way the singular values are plotted, and only occurs in the lower singular value curves. In the singular value spectrum, the singular values are plotted as a function of frequency, where the largest singular value is plotted first at each frequency followed

by subsequently smaller singular values. Since the contributions from different modes vary along the frequency axis, the contribution of two modes can be approximately equal at a specific frequency. At this frequency the functions of the corresponding SDOF systems cross each other. Consequently, the lower singular value curve appears to peak, while the higher singular value curve appears to dip. Hence, the peak in this case is not due to a system pole but is caused by an equal contribution from two modes. This characteristic is identifiable since the peak occurs in the lower singular value curve at the same frequency as a dip in the higher singular value curve.

3.2.4 Identification Algorithm

In this section the different steps of the FDD procedure are briefly described. The first step is to estimate the output spectral density matrix from the measured responses. Suppose that the responses are measured at p locations and are collected in a vector denoted by $\mathbf{y}(t) \in \mathbb{R}^p$. The spectral density matrix has the following form

$$\mathbf{S}_{yy}(\omega) = \begin{bmatrix} S_{11}(\omega) & S_{12}(\omega) & \dots & S_{1p}(\omega) \\ S_{21}(\omega) & S_{22}(\omega) & \dots & S_{2p}(\omega) \\ \vdots & \vdots & \ddots & \vdots \\ S_{p1}(\omega) & S_{p2}(\omega) & \dots & S_{pp}(\omega) \end{bmatrix} \tag{3.100}$$

where the diagonal elements represent the PSD functions of the responses and the off-diagonal elements represent the CSD functions of two response signals. The elements of this matrix are defined as

$$S_{ij} = \lim_{T\to\infty} \frac{1}{T} \mathbf{E}\left[Y_i(\omega) Y_j(\omega)^* \right] \qquad \text{for } i,j = 1,2,\dots,p \tag{3.101}$$

where $\mathbf{E}[\bullet]$ denotes the expectation operator, $(\bullet)^*$ denotes the complex conjugate and $Y_j(\omega)$ is the Fourier transform of signal $y_j(t)$. In the particular case that $i = j$, the PSD functions are obtained corresponding to the diagonal in the spectral density matrix. The definition in equation (3.101) can be written in matrix notation as

$$\mathbf{S}_{yy}(\omega) = \lim_{T\to\infty} \frac{1}{T} \mathbf{E}\left[\mathbf{Y}(\omega) \mathbf{Y}(\omega)^{\mathrm{H}} \right] \tag{3.102}$$

where $(\bullet)^{\mathrm{H}}$ denotes the complex conjugate transpose and $\mathbf{Y}(\omega) \in \mathbb{C}^{p\times 1}$ is the Fourier transform of the response vector $\mathbf{y}(t)$. The definition of the spectral density functions assumes that the signals have infinite length. In reality this is never the case and the spectral density functions are thus estimated. A "raw" estimate of the spectral density matrix is obtained by simply dropping the limit and expectation operator in (3.102); that is

$$\hat{\mathbf{S}}_{yy}(\omega) = \frac{1}{T} \mathbf{Y}(\omega) \mathbf{Y}(\omega)^{\mathrm{H}} \tag{3.103}$$

However, this estimation is often not very accurate and suffer from high variance. In order to reduce this variance, the time series are usually divided into (possible overlapping) data segments $\mathbf{y}_k(t)$, $k = 1,\dots,N_s$ where N_s is the number of data segments. Then for each segment a raw estimate of the spectral density matrix is computed according to equation (3.103), where the Fourier transforms are computed for the (shorter) data segments. A final estimate of the spectral density matrix is obtained by averaging the raw estimates of the spectral density matrices computed for the data segments

$$\hat{\mathbf{S}}_{yy}(\omega) = \frac{1}{N_s} \sum_{k=1}^{N_s} \mathbf{Y}_k(\omega) \mathbf{Y}_k(\omega)^{\mathrm{H}} \tag{3.104}$$

where $\mathbf{Y}_k(\omega) \in \mathbb{C}^{p\times 1}$ is the Fourier transform of data segment $\mathbf{y}_k(t)$. The method of averaging is also known as Welch's method (or Bartlett's method in case of non-overlapping data segments) and provides a way to reduce the variance of the "raw" spectral density estimate in exchange for a reduction of frequency resolution (due to the shorter time segments). Moreover, the rank of the PSD matrix is increased, which is necessary to resolve the mode multiplicity with the SVD.

Once the spectral density matrix is constructed the matrix can be factorized using SVD. Repeating the SVD for discrete frequencies $\omega_k \in [0, 2\pi f_s/2]$ results into a collection of singular values and corresponding singular vectors as function of frequency. By plotting the singular values as function of the relative frequency a singular value spectrum is obtain. Modes can be identified as peaks in this spectrum. The frequencies

corresponding to these peaks are estimates of the natural frequencies of the system, and the left singular vectors corresponding to these peaks are estimates of the mode shape vectors. In the case of well-separated natural frequencies only the first singular values and corresponding singular vectors are important. On the other hand, in case of closely-spaced or repeated modes, the number of relevant singular values is equal to the mode multiplicity. Then the closely-spaced modes can be identified as peaks in distinct singular values. The most dominating mode will show up as peak in the first singular value, the second dominating mode in the second singular value and so forth. Typically, not more than two modes have almost the same natural frequency, thus considering the first two singular values as a function of frequency is often sufficient. Figure 3.3 shows an example of the singular value spectrum, in which five different modes can be observed, of which two have the same frequency (around 56 Hz). The repeated mode shows up in the second singular value curve with approximately the same magnitude. The peak in the second singular value around 20 Hz is an computational characteristics that develops due to the way the singular values are plotted as mentioned at the end of previous section (cross singular value effect).

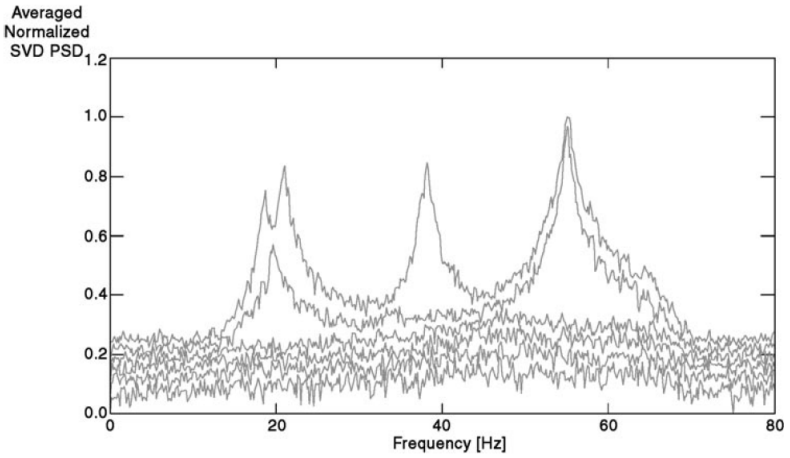


Figure 3.3: Example of singular values of the response spectral density matrix [7].

So far, the natural frequencies and mode shapes are estimated from the information of the frequency lines corresponding to the peaks in the singular value spectrum. Damping ratios could not be estimated from these frequency lines. For this reason the EFDD technique has been developed. The EFDD method is an extension of the FDD technique in order to estimate the modal damping of the system under observation. The EFDD method is based on the identification of the SDOF auto-spectral density functions from the singular value spectrum.

Once a mode is identified in the singular value spectrum, the auto-spectral density function of the corresponding SDOF system (modal coordinate) can be estimated by comparing the mode shape vector $\widehat{\boldsymbol{\phi}}_r$ (3.99) with the singular vectors of the frequency lines around the peak. The consistency between the two vectors is measured using the Modal Assurance Criterion (MAC) [1] (see Section 3.2.5 for a description of this measure). As long as singular vectors are found that has high MAC value with $\widehat{\boldsymbol{\phi}}_r$, then the corresponding singular value belongs to the modal domain of the SDOF system. If at a certain line none of the singular values has a singular vector with a MAC value larger than a certain threshold, the search for matching parts of the modal domain is terminated. In the end, we have identified a frequency range $[\omega_r - \Delta\omega_1, \omega_r + \Delta\omega_2]$ around a peak at frequency ω_r of modal dominance called the modal domain, see Figure 3.4. The associated singular values estimate the auto-spectral density function of the corresponding SDOF system, which can be transformed back to the time domain by inverse Fourier transform, yielding the auto correlation function of the SDOF system from which the natural frequency and damping ratio can be found by estimating crossing times and logarithmic decrements [6].

In this work the modal domain is used to distinguish between different modes (or modes and noise) rather than estimating modal damping ratios. Therefore we introduce the following discriminator function

$$d_r(\omega_k) = \mathrm{MAC}\left(\widehat{\boldsymbol{\phi}}_r, \mathbf{u}_1(\omega_k)\right) \tag{3.105}$$

where $\widehat{\boldsymbol{\phi}}_r$ is the estimated mode shape vector and $\mathbf{u}_1(\omega_k)$ is the left singular vector at neighboring frequency

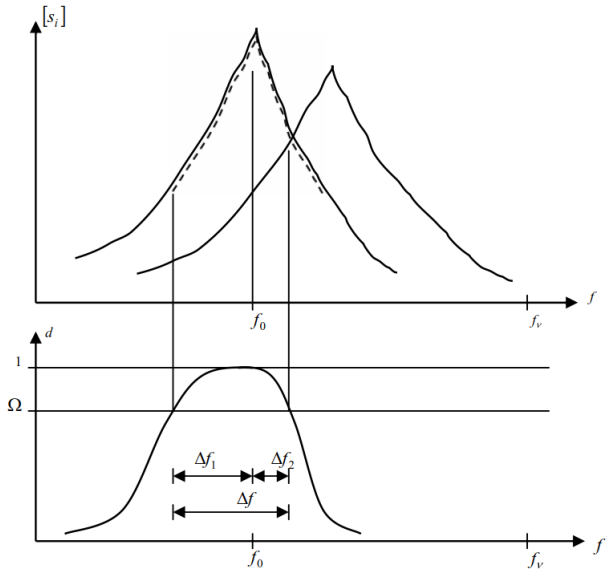


Figure 3.4: Definition of the modal domain for an identified mode. Top figure shows the modal decomposition using the SVD of the spectral density matrix. Bottom figure shows how the modal domain is defined by the part where the discriminator function is larger than the threshold [8].

ω_k. The modal domain of mode r is defined by the following criterion

$$d_r(\omega_k) \geq \Omega \tag{3.106}$$

The lower the value of Ω the larger the bandwidth of the modal domain $\Delta\omega = \omega_1 + \omega_2$.

Summary process

The steps of the FDD technique can be summarized as follows:

Step 1. Estimate the spectral density matrix $\mathbf{S}_{yy}(\omega)$ according to equation (3.104).

Step 2. Factorize $\mathbf{S}_{yy}(\omega)$ as in equation (3.95) at discrete frequencies $\omega_k \in [0, 2\pi f_s/2]$ using the singular value decomposition, to obtain a set with singular values and corresponding singular vectors. Usually it is sufficient to consider the set with first singular values and vectors denoted by $\{\sigma_1(\omega_k), \mathbf{u}_1(\omega_k)\}$ (except in the case of closely-spaced modes).

Step 3. Identify peaks in the singular value spectrum and let ω_r, $r = 1, \ldots, M$ denote the corresponding frequencies. Estimates of the natural frequencies are obtained by the frequencies corresponding to the peaks, and estimates of the mode shapes are obtained by the left singular vectors corresponding to the peaks; that is

$$\widehat{\boldsymbol{\phi}}_r = \mathbf{u}_1(\omega_r) \tag{3.107}$$

The natural frequencies and mode shapes are estimated from the information of a single frequency line, namely the one associated with a peak in the singular value spectrum. The estimation of the damping ratios requires the following additional steps [6]:

Step 4. For each mode, estimate the SDOF auto-spectral density function from the singular value spectrum based on the MAC value between the mode shape estimate $\widehat{\boldsymbol{\phi}}_r$ and the singular vectors of neighboring frequency lines.

Step 5. Transform a fully or partially identified auto-spectral density function back to the time domain and determine the natural frequency and damping ratio based on zero crossings and logarithmic decrements. The natural frequencies estimated in this step are usually more accurate than the ones obtained in **Step 3**.

Rank of PSD matrix

The ability to separate closely-spaced modes with the FDD method depends on the rank of the spectral density matrix. The rank of this matrix is limited by the following factors [5, p. 622]:

- The number of *distinct* measurement points. If multiple sensors are close to each other, it is possible that these sensors measure the same information, and this does not contribute to the matrix rank.
- The number of independent force inputs.
- The number of data segments used to estimate the PSD matrix according to equation (3.104). If the information of only one data segment is used to estimate a PSD matrix, then the rank of this matrix will always be one.

3.2.5 Modal Assurance Criterion

The Modal Assurance Criterion (MAC) is used as a measure of consistency (degree of linearity) between estimates of mode shapes. This provides a confidence factor in the evaluation of mode shapes from different reference locations or different modal parameter estimation algorithms. The MAC is defined as a scalar constant measuring the degree of linearity between one mode shape and another reference mode shape as follows

$$\mathrm{MAC}(\boldsymbol{\phi}_r, \boldsymbol{\phi}_s) = \frac{|\boldsymbol{\phi}_r^{\mathrm{H}} \boldsymbol{\phi}_s|^2}{\boldsymbol{\phi}_r^{\mathrm{H}} \boldsymbol{\phi}_r \boldsymbol{\phi}_s^{\mathrm{H}} \boldsymbol{\phi}_s} \tag{3.108}$$

The MAC takes on values in the range from zero to one, where zero implies no consistent correspondence between the mode shapes and one implies a consistent correspondence. In this manner, if the mode shape under consideration truly exhibit a consistent, linear relationship, then the MAC should approach unity. However, the MAC can only indicate consistency, not validity or orthogonality. If the same errors exist on all mode shape estimates, then the MAC is close to unity, even though the mode shape estimates are not valid. So, careful interpretation of MAC is import to avoid potential errors. Possible reasons of the MAC being close to zero or unity are listed below [1].

Possible reasons for the MAC being zero (modal vectors are not consistent):

- The system is non-stationary.
- There is noise on the reference modal vector.
- The modal parameter estimation is invalid.
- The modal vectors are from linearly unrelated mode shape vectors, which implies that the modal vectors are orthogonal.

Possible reasons for the MAC being unity (modal vectors are consistent):

- The modal vectors have been incompletely measured.
- The modal vectors are the result of a forced excitation other than the desired input.
- The modal vectors are primarily coherent noise.
- The modal vectors represent the same modal vector with different arbitrary scaling.

4

Application: Similarity Filtering and Frequency Domain Decomposition

In this chapter the processing of the Zwartewaterbrug data from raw data to initial features is described. An overview of the processing steps is given in Figure 4.1. First, in Section 4.1 a reliable high frequency range is determined that is sensitive to small changes in the bridge deck, and a band-pass filter is designed to operate in this range. In Section 4.2 the similarity filtering is applied, which is a necessary step to reduce the non-stationarities in the acceleration time histories due to large variations in traffic load. Finally, in Section 4.3 the Frequency Domain Decomposition (FDD) is applied to the processed signals (i.e. band-pass and similarity filtered signals), in order to obtain 1) the singular value spectra and 2) the natural frequencies and mode shapes of the bridge. The singular value spectra and the natural frequencies will both serve as damage-sensitive features in the subsequent chapters about machine learning.

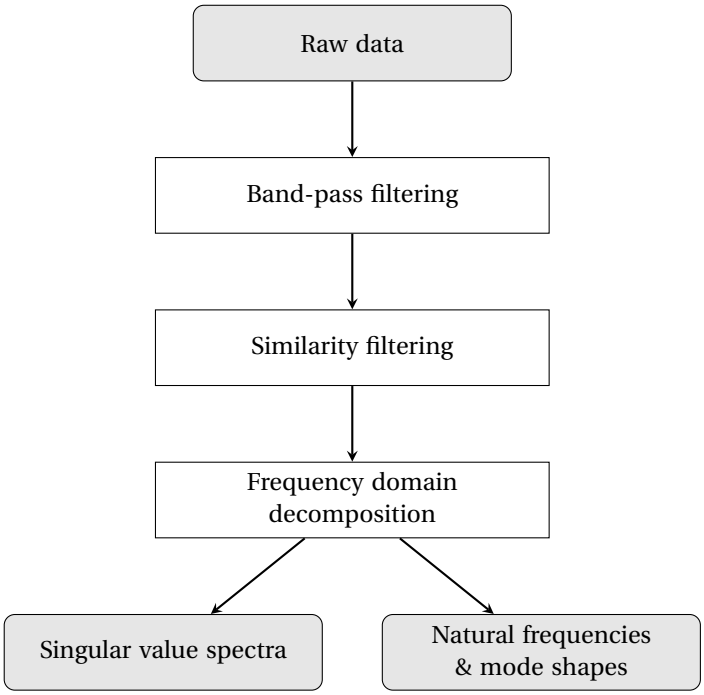


Figure 4.1: Overview of processing steps.

4.1 Bandpass Filtering

The aim of this section is to determine a reliable high frequency range in the vibration data for damage detection purposes. The reason for this is twofold: 1) the higher the frequency region, the more the response spectra are sensitive to small changes, and 2) to reduce the number of frequency lines that will be used as damage-sensitive features in the machine learning algorithms. The first reason is directly related with the main goal of the Zwartewaterbrug project, which is to detect small changes in the deck structure of the bridge (e.g. local increase of the structural mass which simulates the presence of damage). The second reason is primarily related to the performance of the machine learning algorithms as it is known that some algorithm perform poorly with high dimensional feature vectors. This dimensionality problem is often referred to as the *curse of dimensionality* and will be explained in more detail in Chapter 5. Before proceeding to the methodology, first some words are spent on the meaning of a "reliable high frequency range". The term "reliable" implies that the frequency region is active in most of the measurements and the term "high frequency range" implies that the selected frequency region covers primarily local modes (i.e. local with respect to the spatial size of the sensor network), so the fundamental modes are not considered.

The modes are identified from the vibration data using the Frequency Domain Decomposition (FDD) technique. The key of the FDD technique is to perform a Singular Value Decomposition (SVD) of the spectral density matrix to obtain the singular values and singular vectors at discrete frequencies $f_k \in [0, f_s/2]$. Plotting the singular values as function of frequency will provide an overlaid plot of the auto spectral densities of the modal coordinates and thus modes can be identified as peaks in the singular value spectrum. Hence, we can identify the peaks in the singular value spectra of all the measurements and draw conclusions about the frequency content based on the number of identified peaks in certain frequency regions. However, not every peak in the singular value spectrum indicates a mode. Errors such as noise, leakage, and non-linearities could generate spurious peaks, also referred to as noise peaks. So, it is important to distinguish between modal peaks and spurious peaks. In this section a discriminator function is introduced that will be used to distinguish between the peaks of modes and those related to noise. In fact, the discriminator function introduced here is very similar to the function that defines the modal domain, as explained in Section 3.2.4.

Suppose a peak is identified in the singular value spectrum at frequency f_r. Calculating the Modal Assurance Criterion (MAC), equation (3.108), between the singular vector associated with the peak and the singular vectors of neighboring frequency lines defines the following discriminator function

$$d_r(f_r) = \mathrm{MAC}\left(\mathbf{u}_1(f_r), \mathbf{u}_1(f_k)\right) \tag{4.1}$$

where $\mathbf{u}_1(f_r)$ is the singular vector at the peak and $\mathbf{u}_1(f_k)$ is the singular vector at a neighboring frequency f_k. If the components of each of the singular vectors are random, which is generally the case for a noise peak, then this function is close to zero. On the other hand, if the singular vectors are consistent (linear correlated), the MAC is close to unity meaning that the associated singular values correspond to the same modal coordinate. Hence, modal and noise peaks can be discriminated using the requirement that for neighboring frequency lines the following criterion is met

$$d_r(f_k) \geq \Omega \tag{4.2}$$

where Ω is a threshold value that defines the minimum MAC value so that the two vectors can be considered consistent. When several frequency lines are considered at each side of the peak, then a peak is accepted as modal peak if the criterion (4.2) is met for all frequency lines. Hence, the singular vectors $\mathbf{u}_1(f_k)$, with $f_k = f_r \pm \Delta f$, should be consistent with the singular vector associated with the peak at a frequency f_r. For discrete frequencies, the bandwidth Δf can be expressed as $\Delta f = k \cdot f_0$, where k is the considered number of frequency lines around the peak and f_0 is the frequency resolution which is defined as $f_0 = f_s/N$, in which f_s is the sample rate and N are the number of samples used in the Fast Fourier Transform (FFT). Notice that this bandwidth is similar to the modal domain introduced in Section 3.2.4 and shown in Figure 3.4. So, it is equivalent to state that the modal peaks are discriminated from noise peaks based on the width of the associated modal domain.

Application

For the selection of the reliable high frequency range in the vibration data of the Zwartewaterbrug it is chosen to consider a bandwidth of 0.5 Hz around a peak in combination with a threshold level $\Omega = 0.90$. The bandwidth is fixed to 0.5 Hz based on visual inspection of some singular value spectra. However, it is probably better to have a frequency dependent bandwidth as the modal peaks of the lower modes are different than

for the higher modes due to different amount of damping. Higher modes have generally larger damping values resulting in a wider modal peak while the lower modes with lower damping value have more sharp peaks. Hence, for the former the bandwidth may be larger than for the latter.

The process to identify the consistent peaks (and thus the reliable high frequency range) in the vibration data is summarized below.

Step 1. Preparation of data. Since the measurements do not have the same length, the spectral density estimation will not have the same quality. Therefore, measurement samples are concatenated to a common length of 60 s (60000 samples).

Step 2. Estimate the spectral density matrix $\mathbf{S}_{yy}(f_k)$ using the data from all 32 sensors resulting in a complex matrix of size 32×32. Welch's method of averaged periodograms is applied for estimating the power spectral densities (diagonal elements of the matrix) and the cross spectral densities (off-diagonal elements of the matrix). The averaged periodograms are computed using 11 segments of 10 s (10000 samples) length with 50% overlap and a hamming window is applied to the data of each segment to reduce spectral leakage.

Step 3. For all discrete frequencies $f_k \in [0, f_s/2]$ perform a singular value decomposition of the spectral density matrix obtained in **Step 2** to obtain a set of singular values and singular vectors.

Step 4. Find a peak in the first singular values $\sigma_1(f_k)$, the frequency corresponding to this peak is denoted f_r.

Step 5. Evaluate the discriminator function (4.1) for frequency lines within a bandwidth of $\Delta f = 0.5$ Hz around the peak. The number of frequency lines depends on the sample rate f_s and the number of samples used in the FFT. With $f_s = 1000$ Hz and $N = 10000$ the frequency resolution $f_0 = 0.1$ Hz and thus the number of frequency lines in a bandwidth of 0.5 Hz is 5.

Step 6. Accept the peak as modal peak if the discriminator function (MAC value) for all five neighboring frequency lines is larger than the threshold value $\Omega = 0.90$. Otherwise, the peak is labeled as spurious.

Step 7. Repeat **Step 4** to **Step 6** until no more new peaks can be found or until a user defined number of peaks are found.

Step 8. Repeat **Step 1** to **Step 7** until no more data is available.

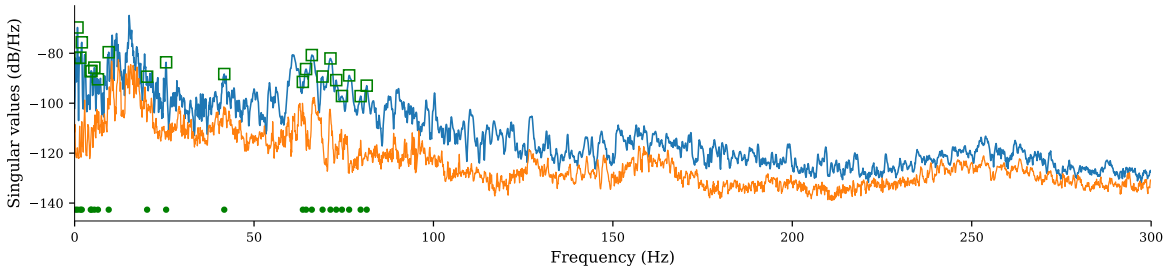
In the end, we have a set of consistent peaks for each measurement that was prepared in **Step 1**. Subsequently, histograms are formed to show the distribution of the frequencies corresponding to the consistent peaks, from which the reliable high frequency content is determined.

Results

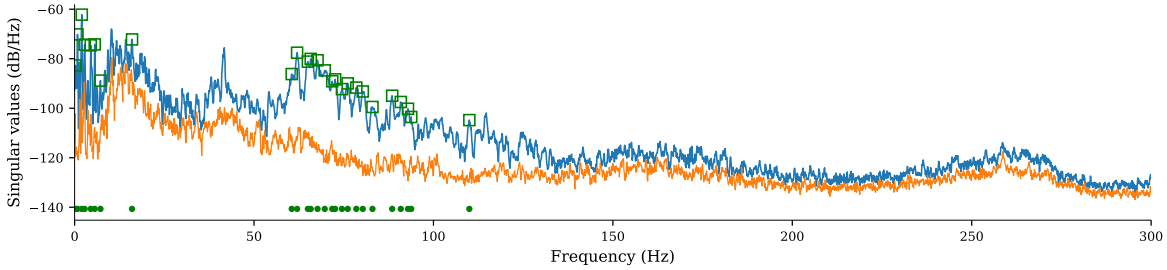
In total there are 764 measurements divided over five different mass classes (see Table 2.1). In order to obtain equal quality of the FDD, multiple measurements are combined to form data blocks of 60 seconds. Per mass class, the following number of data blocks are extracted: 69 (0 kg), 38 (25 kg), 82 (50 kg), 53 (75 kg) and 56 (100 kg). In Figure 4.2, five examples of the singular value spectrum with indication of consistent peaks (possible modes) are provided. It must be noted that 32 singular values per frequency line are obtained and that only the largest two singular values per frequency line are presented in the figures, the other thirty values are omitted because these are less important and most of them are close to zero. It is further noted that the consistent peaks are obtained from the largest/first singular values only. From the set of consistent peaks, histograms of frequency count are formed separated by added mass class, as shown in Figure 4.3. The frequency axis in the figures is limited to 300 Hz, since no peaks were found above 300 Hz.

Conclusion

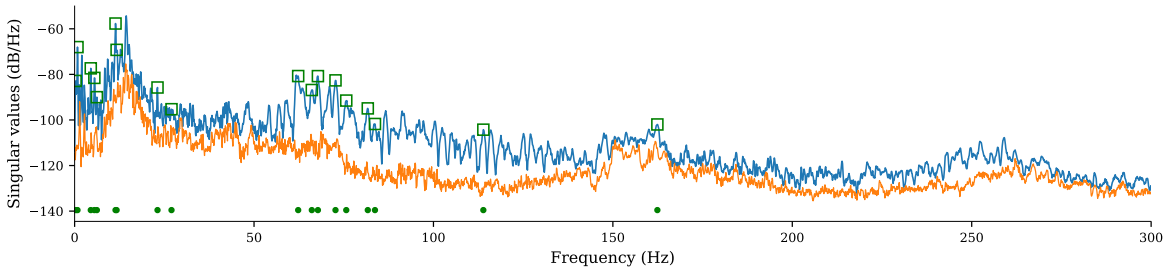
From the histograms, it can be observed that the frequency region from 50 to 100 Hz has a large number of consistent modes. It is assumed that the aforementioned frequency region corresponds to the local modes related to the spatial size of the sensor network and thus is the region of interest. Notice that the local modes are related to the complete sensor network (i.e. over all three instrumented segments) since the spectral density matrix in **Step 2** of the process above is estimated using the data from all 32 sensors. Similar results



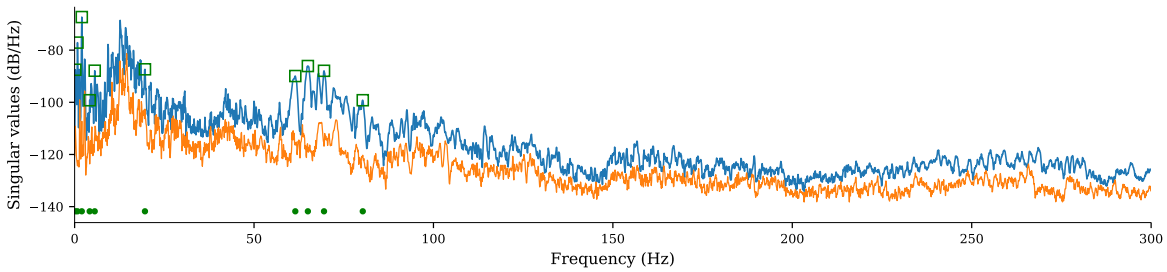
(a) Example 1



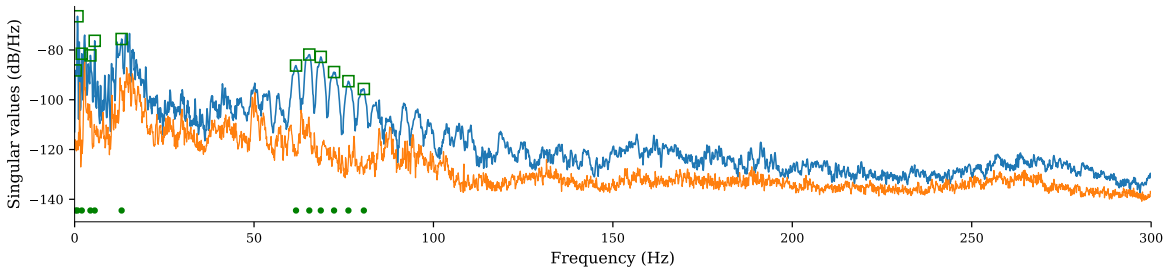
(b) Example 2



(c) Example 3

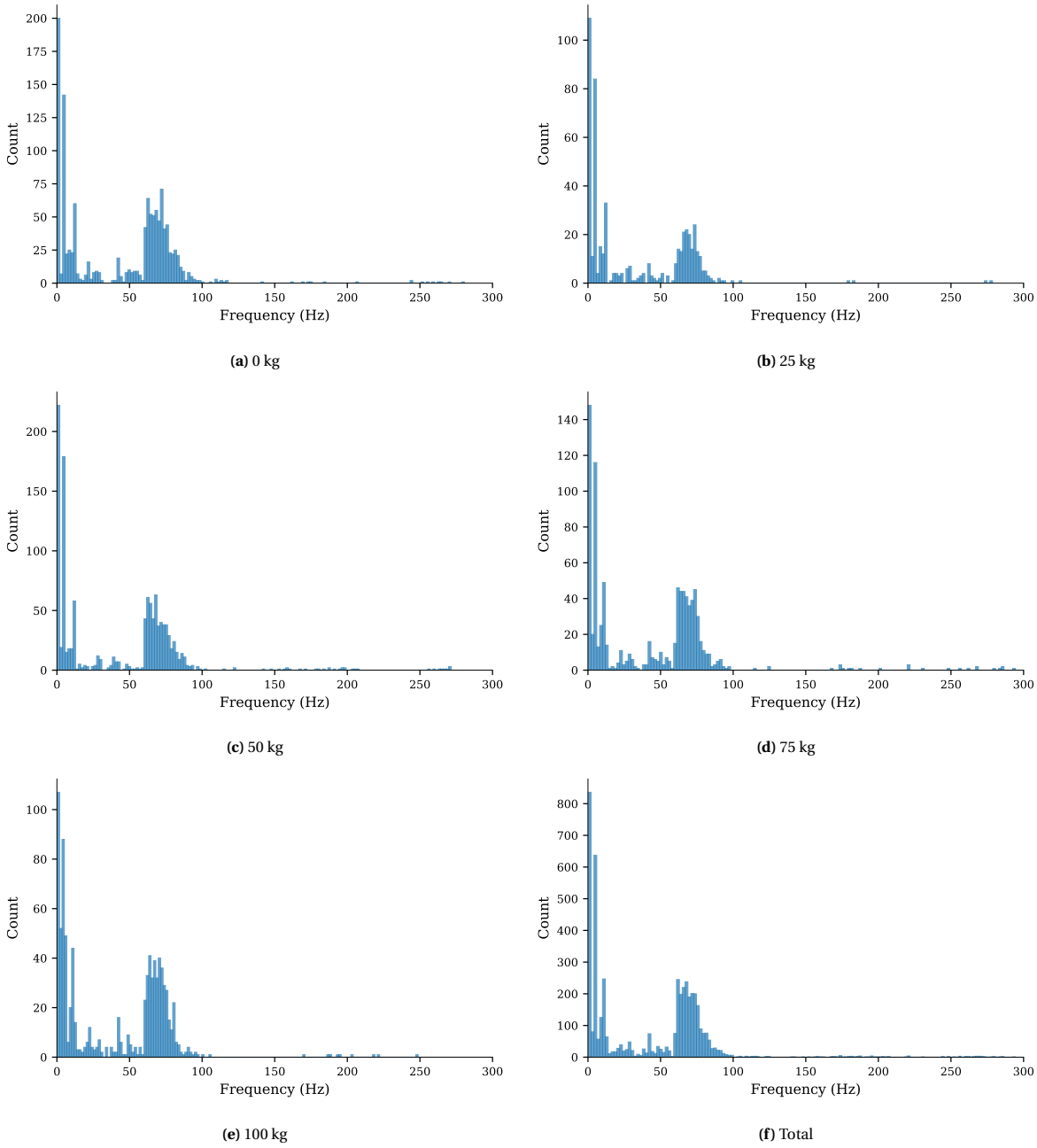


(d) Example 4



(e) Example 5

Figure 4.2: Singular value spectrum with consistent modes.



(a) 0 kg

(b) 25 kg

(c) 50 kg

(d) 75 kg

(e) 100 kg

(f) Total

Figure 4.3: Histograms of consistent mode count.

are obtained if the analysis is performed for each bridge segment separately, showing that the selected high frequency content applies to all segments.

In correspondence with the analysis goals we want to find a reliable high frequency range that potentially increases the sensitivity of damage for the damage detection algorithm. Based on the histograms, the frequency range from 50 to 100 Hz is chosen and subsequently a band-pass filter is designed to operate in the given frequency range. In the next section, the data will be band-pass filtered to the given frequency region before the similarity filtering.

The selected frequency region is broader than the one from 60 to 75 Hz used in the thesis of Milosevic on the Zwartewaterbrug [32]. In that thesis it was found that the added mass could be detected based on one dominant mode in the data of the second segment of the bridge. This mode had a local mode shape with a nonzero displacement at the added mass, which makes it reasonable that this mode is sensitive to the added mass. In fact, it is expected that the modes with a local mode shape having a nonzero displacement at the

added mass have the largest mass sensitivity. Now the question raised, *what would happen if the mass was located at another position in the instrumented bridge section, for example between sensor 12 and 13?* Then it seems to me that the frequency region from 60 to 75 Hz might not be suitable because the modes in this range are not sensitive to the added mass at the different location. So to be less restricted to the location of the added mass, a broader frequency region from 50 to 100 Hz is considered in this thesis.

Another reason for the choice of a broader frequency region is related to the similarity filtering. As stated in the thesis of Milosevic, the data must be band-pass filtered to the frequency range of interest before application of the similarity filtering, otherwise you would not get the expected results. The similarity filtering appeared to work well when the data was band-pass filtered to the range 60-75 Hz, but no other frequency ranges were considered in his report. So, it is interesting to investigate whether the similarity filtering also works when the data is band-pass filtered to a broader frequency range.

4.2 Similarity Filtering

Once the reliable high frequency region has been established, the similarity filtering can be performed. It is emphasized that the aim of this processing step is to reduce the operational variability. For the bridge under consideration this variability is primarily caused by the large differences in traffic load, as the vehicle weight ranges from 200 kg motor cycles to 50000 kg heavy trucks. In this section, a description of the processing algorithm is given first, followed by some results.

4.2.1 Algorithm Composition

In Section 3.1 the theory of similarity is discussed. In this section it is shown how the similarity filtering is applied on real data. Recall that similarity filtering is based on successive convolution of different measurements (from the same sensor), which can be expressed as

$$y_j(t) = \overset{n}{\underset{k=1}{\circledast}} x_{j,k}(t) = x_{j,1}(t) * x_{j,2}(t) * \cdots * x_{j,n}(t) \tag{4.3}$$

where n is the order of similarity filtering, and the index $j = 1,2,\ldots,32$ refers to the sensor location. Recall that the order of similarity filtering is defined as the number of signals/segments being convolved and that the number of convolutions is one less than the order of similarity filtering. The signals $x_{j,k}(t)$ might have different lengths, but for now it will be assumed that the signals have equal length, which will be denoted by T_s. In order to reduce the operational variability to an acceptable level, a sufficient number of measurements should be used in the convolution. In other words, the order n of similarity filtering should be high enough. In Section 3.1.4 it was concluded that it is better to use smaller time segments in the order of magnitude of seconds rather than the raw signal length. This conclusion was based on the consideration of three aspects, namely 1) the computational cost of similarity filtering, 2) the time scale on which similarity filtering operates, and 3) the expected natural frequencies of the structure. It was also mentioned that the final selection of the order n and the segment length T_s should be based on an analysis of the combination of both parameters. In Appendix E the evolution of the singular value spectrum (resulting from FDD) for an increasing number of convolutions and for different segment lengths are shown. Suitable parameters are chosen based on visual inspection of the figures and consideration of previous aspects. The chosen parameters are listed in Table 4.1, in which T denotes the length of the data blocks in seconds, necessary for one processed signal and α denotes the percentage overlap between two successive data blocks. It is noted that the processed signals will contain much more samples than the original signals due to the convolution operations, and thus require more memory for storage (i.e. the processed signals consist of 249901 samples while the original signals have on average about 25000 samples). To reduce the number of samples, the processed signals are truncated at both ends at the expense of the quality of the (singular value) spectrum, as will be explained later.

Table 4.1: Selected parameters for similarity filtering, in seconds and number of samples (sample rate is 1000 Hz).

n	Segment length		Total length		Overlap		
	T_s (s)	N_s	T (s)	N	T_o (s)	N_o	α (%)
100	2.5	2500	250	250000	225	225000	90

Let $Z = \{x_{j,k}(t, T_k) \colon k = 1, 2, \ldots, n_m\}$ denote the set of acceleration time histories (raw data) from sensor j, where T_k is the length of the k-th signal and n_m is the number of signals available. It is assumed that the signals in the set are associated with the same added mass class. Once the order of similarity filtering n and the length of the segments T_s are chosen, the application of similarity filtering to one data set involves the following steps:

Step 1. Form a data block with at least a length of T seconds by concatenating measurement samples from the set of measurements $Z = \{x_{j,k}(t, T_k) \colon k = 1, 2, \ldots, n_m\}$. To ensure that variations, other than operational variations, within the selected data will be minimal, consecutive samples are used rather than randomly selecting the samples.

Step 2. Apply a band-pass filter of 50-100 Hz to the data block from the previous step. The given frequency region was determined as a reliable high-frequency region in Section 4.1.

Step 3. Split the data block into n segments, each with a length of T_s seconds.

Step 4. Perform similarity filtering of the segmented data; that is, successive convolution of the data segments according to equation (4.3).

Step 5. Repeat **Step 1** to **Step 3** until all measurement samples in the set $\{x_{j,k}(t, T_k) \colon k = 1, 2, \ldots, n_m\}$ are used. In **Step 1** some amount of overlap between the data blocks from successive runs is applied to obtain a reasonable number of processed signals.

Step 6. Repeat **Step 1** to **Step 4** for all sensors.

There are two remarks on the described approach: 1) the applied overlap between the data blocks, and 2) the selection of the measurement samples, which are explained in more detail below.

First, it is noted that the set with processed signals can be considerably smaller than the original set with raw signals, because multiple measurements are used to produce one processed signal. For example, when similarity filtering is performed with 100 segments of 2.5 s, then 250 s of data is required to generated one processed signal. The acceleration time histories from the Zwartewaterbrug have on average a length of 25 seconds, meaning that approximately ten measurements are concatenated to form the data block in the first step of the process above. The set with processed signals after similarity filtering will be approximately ten times smaller than the original set.

Recall that the aim of this thesis is to detect the small added masses using machine learning algorithms. The performance of machine learning algorithms depends on the size of the data set and is generally better for large data sets. Hence, the reduction of the data set due to the application of similarity filtering might be problematic for the machine learning algorithms, especially when the original data set is already small (as is the case for the Zwartewaterbrug data). To overcome this problem, some amount of overlap is applied in **Step 1** between the data blocks from successive runs. This is illustrated in Figure 4.4, in which the overlapping data blocks of five successive runs are shown. There will be a large correlation between the signals from overlapping data blocks, which might negatively affect the performance of the machine learning algorithms. When a very large data set is acquired then overlapping data blocks do not have to be applied, because the size of the resulting set with processed signals will be satisfactory.

In this thesis it is chosen to use 90% overlap (see Table 4.1) between the data blocks from successive runs, in order to obtain a reasonable number of processed signals.

Secondly, it is noted that the measurement samples in **Step 1** should not be randomly selected, because then the assumption of a time-invariant system might be violated. In particular, the environmental conditions (e.g. temperature) vary during the measurements, which change the characteristics of the bridge. If the changes in structural properties are large within a data block, the similarity filtering cannot be applied as it was developed to remove operational variations under the assumption that the system is time-invariant. To illustrate this, the mean temperature and standard deviation for a number of data blocks are shown in Figure 4.5a. The data blocks are formed using a random and normal (i.e. in chronological order) sample selection to a total length of 250 seconds, followed by computing the mean and standard deviation of the temperature corresponding to the selected samples. Obviously, the variations in temperature are larger when the measurement samples are randomly selected and the characteristics of the system might not be constant. Hence, the time-invariant assumption will no longer hold and the similarity filtering should not be applied.

Notice that with normal sample selection, the variations within a data block are smaller, but the variations between the data blocks are larger. However, this is not relevant for the application of the similarity filtering,

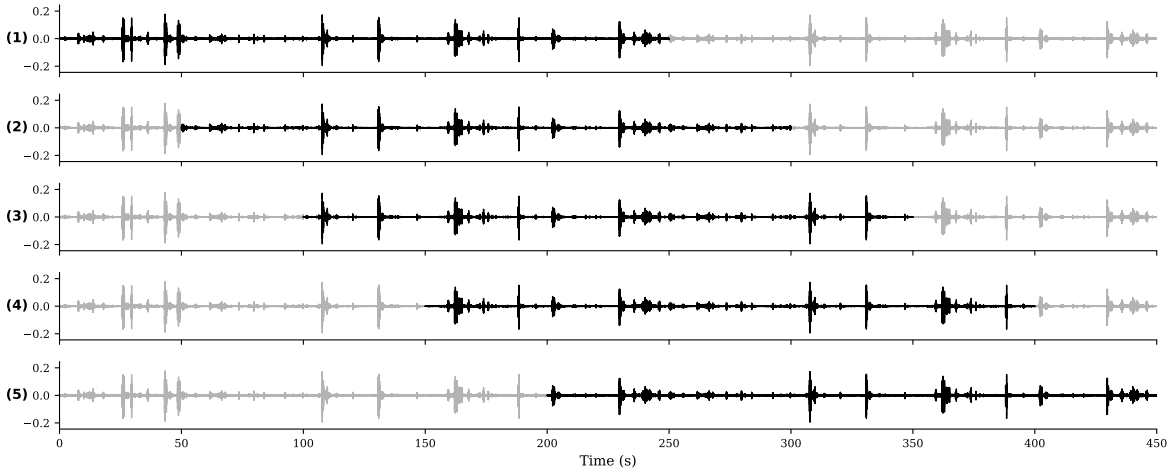


Figure 4.4: Illustration of overlapping data blocks that are formed in **Step 1** during five successive runs. The length of the data blocks is $T = 250$ seconds, and the amount of overlap is $\alpha = 80\%$.

where one data block at the time is considered. The variations between the data blocks will be present in the processed signals and might also be reflected on the damage-sensitive features. If this is the case, then the temperature effects must be removed from the features (i.e. data normalization, which will be discussed in Chapter 8), otherwise the temperature effects might mask the damage effects and complicates the damage detection. This also applies to other environmental variables, although temperature is usually seen as most important.

The time-invariant assumption might also be violated when the length of a data block is very long, because a longer data block covers a larger time range with potentially larger variations in the environmental conditions. This is illustrated in Figure 4.5b in which the mean temperature and standard deviation for a number of data blocks are shown. The data blocks are formed with a fixed number of segments $n_s = 50$ and the length of the segments is 2.5 or 10 seconds; for the 2.5 s segments this means that one data block spans 125 s and for the 10 s segments one data block spans 500 s of data. The data blocks are formed with the same set of data, so there are less data blocks formed with the longer segments than with the shorter segments. Here the variation of temperature within a data block are comparable for the two different lengths, but the variation between the data blocks are much larger for the longer data blocks. However, only the variation within a data block is important for the time-invariant assumption in similarity filtering.

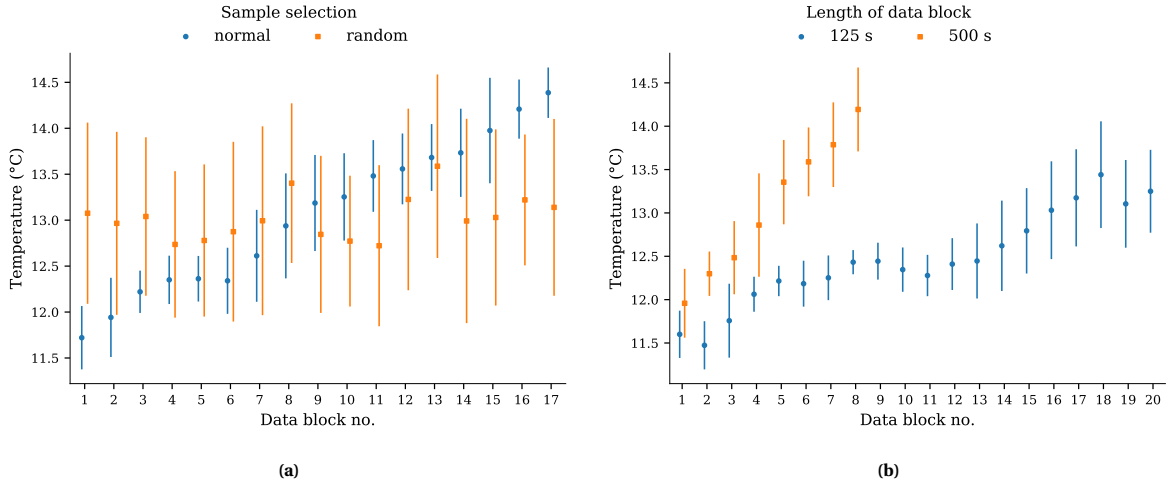


Figure 4.5: Mean temperature and variation of the selected measurement samples that form a data block for similarity filtering. The data blocks are formed with (a) random sample selection and (b) different segment length.

In the end, we have a new data set of processed signals consisting of 32 channels as in the original data set, but less observations because multiple measurement samples are used to produce one processed signal. The reduction of the data set is however limited because we have applied overlapping data blocks. The sizes of the original data sets and the data sets with processed signals are listed in Table 4.2. In the table the number of records per dataset are presented, each record consists of 32 (processed) signals.

An example of a processed signal per sensor is shown in Figure 4.6. The signals are generated using 100 segments of 2.5 seconds, resulting in processed signals of 250 s length. In general, the signals contain many leading and trailing zeros, which are the results of how the discrete convolution was performed. These zeros are of minor importance for further processing, what is important is the *more stationary* central part of the signals, which is marked gray in the figures. However, in the next section, it will become apparent that important information might be lost when only the central part of the signal is considered.

Table 4.2: Overview of the number of received data records with raw signals, and the number of data records with processed signals after similarity filtering. Each record consists of 32 signals, one signal per sensor.

	Dataset/ added mass							Total
	1/m0	2/m100	3/m50	4/m0	5/m25	6/m50	7m75	
Raw data	52	150	85	109	111	125	120	752
After similarity filtering	43	141	74	101	100	114	114	687

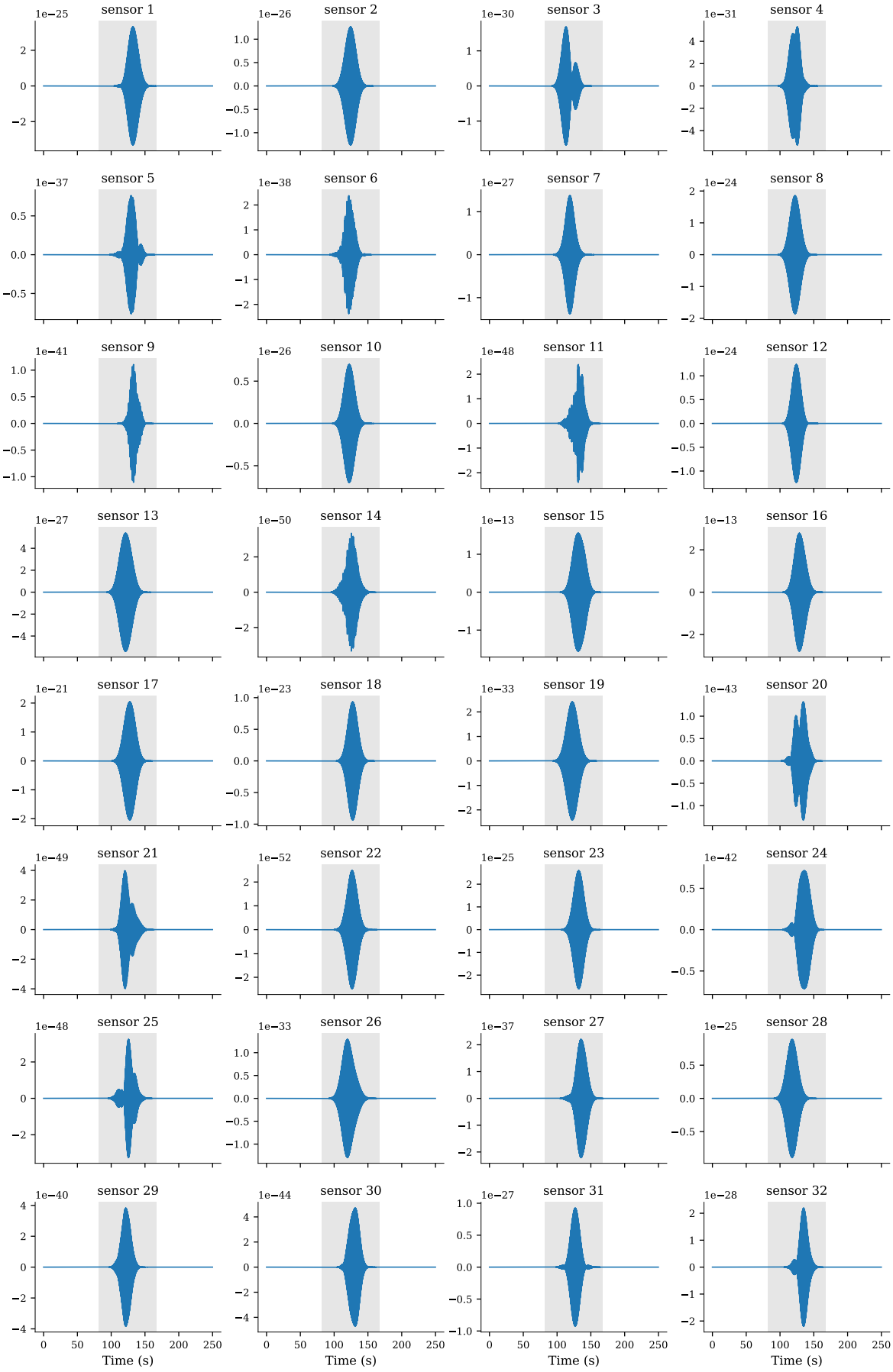


Figure 4.6: Example of processed signals after the application of similarity filtering with $n = 100$ and $T_s = 2.5$ seconds. The marked area indicates the centered signal with a length of 80 seconds (i.e. 80000 samples). The units on the vertical axis are omitted because these are meaningless and not relevant for our application.

4.3 Frequency Domain Decomposition

This section is concerned with the application of the FDD method to the processed signals obtained in previous section. The theory about FDD is described in Section 3.2. The first step of the FDD method is to estimate the spectral density matrix as described in Section 3.2.4. The spectral density matrix of the processed signals (i.e. band-pass and similarity filtered signals) is simply computed (omitting any scaling factor) as

$$\mathbf{S}_{yy}(\omega) = \mathbf{Y}(\omega)\mathbf{Y}(\omega)^{\mathrm{H}} \tag{4.4}$$

where $\mathbf{Y}(\omega)$ is the Fourier transform of the processed signals $\mathbf{y}(t)$. The FDD is performed for each bridge segment separately because of the following two reasons: 1) to enhance the mass sensitivity of a particular segment, and 2) to allow for some global damage localization; for example, if the mass can be detected with the data from segment 2, but not with the data from segment 1 and 3, the mass will most likely be located in segment 2. This means that the vector $\mathbf{y}(t)$ contains the processed signals from one bridge segment only. Furthermore, the same output measurement points are taken as reference and thus $\mathbf{S}_{yy}(\omega)$ is a square *Hermitian* matrix.

A diagonal element of the spectral density matrix represents the Power Spectral Density (PSD) function, also called auto-spectral density function. This function describes the rate of change of the variance with frequency in the corresponding signal. In Figure 4.7 the PSDs of the processed signals (Figure 4.6) are shown. To illustrate the effect of similarity filtering, Figure 4.8 shows the PSDs of the unprocessed signals. It must be noted that these signals have already been band-pass filtered, so the term "unprocessed" merely refers to the fact that similarity filtering has not been applied to these signals. In this example, the spectra before and after similarity filtering are computed using the same data; that is, a data block of 250 seconds. The spectra of the unprocessed signals are estimated using Welch's method of averaged periodograms.

The effect of similarity filtering can readily be seen from the figures. Figure 4.7 shows that the power spectrum of the majority of the sensors contain one dominant frequency component, except for 14 sensors. A closer look to the figure reveals that the magnitude of the spectral values in the spectra of these 14 sensors is much lower. The lower magnitude can be explained by the purpose of similarity filtering, which is among others to amplify the system characteristics in a measured signal. In other words, similarity filtering was unable to amplify the system characteristics in the signals of the 14 sensors. The reason for this might be related to the local modes that were excited at these senors; either one of the following reasons might be the case: 1) no local modes are excited at these sensors, or 2) only modes with very low energy are excited at these sensors, or 3) the local modes are not consistently excited at these sensors and thus not present in all the signals/segments that are used in the similarity filtering for the generation of a processed signal.

The sensors with low quality power spectra (i.e. large variability and low magnitude) are not useful for the purpose of damage detection due to the large variability in spectral density, which are obviously not related to damage. These sensors will have a low contribution in the singular value decomposition of the spectral density matrix in the next step.

Figures F.1–F.4 in the appendix show the PSDs per sensor of all processed signals in the form of a spectrogram. These figures show that the observations done for the example above apply to all processed signals. Moreover, they show that the dominant frequency component changes over time, presumably due to variations in the environmental conditions.

The next step in the FDD method is to factorize the spectral density matrix using SVD at discrete frequencies $f_k \in [50, 100]$ Hz (i.e. the frequency range of interest as determined in Section 4.1). This factorization results into a collection of singular values and corresponding singular vectors. The number of singular values and singular vectors is equal to the number of sensors used to form the square spectral density matrix. For segment 1 a 10×10 matrix is constructed using the signals from sensors 1 to 10; for segment 2 a 12×12 matrix is constructed using the signals from sensors 11 to 22; and for segment 3 a 10×10 matrix is constructed using the signals from sensors 23 to 32.

For each of the three segments, an example of the singular value spectrum before and after similarity filtering is shown in Figures 4.10–4.12. Only the first two singular values (i.e. the largest two singular values) are shown. This shows that the singular value spectra after similarity filtering are much smoother and multiple dominant frequency components are clearly visible. These frequency components cannot be observed in the spectra of the unprocessed signals before similarity filtering. The dominant frequency components are assumed to be related to the dynamic characteristics of the system, which are amplified by the application of similarity filtering.

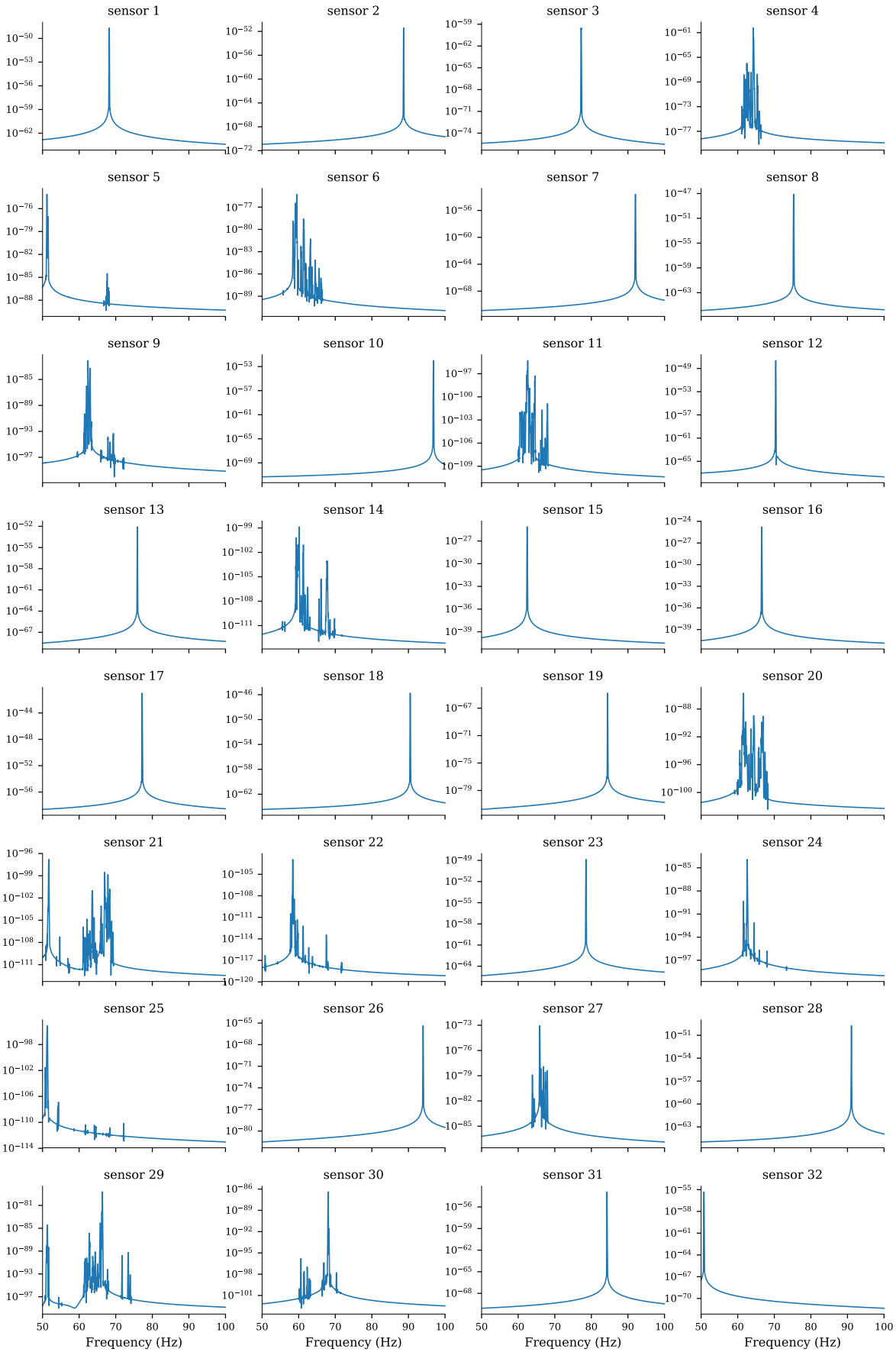


Figure 4.7: Power spectral density functions of the processed signals shown in Figure 4.6 (i.e. band-pass and similarity filtered signals). The processed signals were obtained by successive convolution of 100 segments of 2.5 seconds (i.e. using a data block with a total length of 250 s).

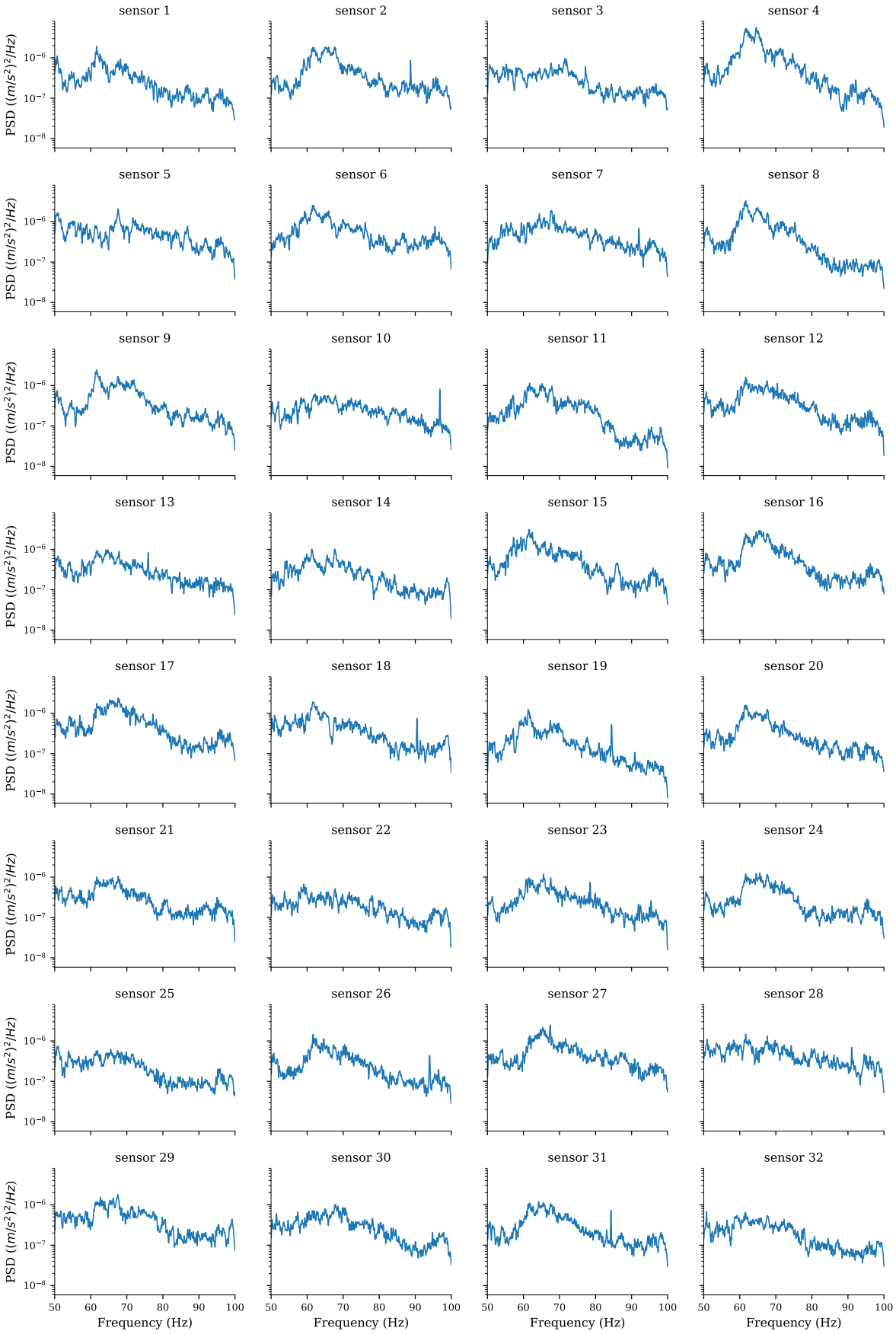


Figure 4.8: Power spectral densities of band-pass filtered signals (50–100 Hz). The PSD is estimated using Welch's method, in which the data block of 250 seconds is divided into 49 segments of 10 seconds with 50% overlap and the segments are weighted with a Hann window.

The spectrum of the largest singular values is the most important for FDD, and this spectrum is often intended when referring to *the singular value spectrum*. This spectrum can be seen as an overlaid plot of the PSDs of the sensors used to construct the spectral density matrix. (i.e. the diagonal elements in the spectral density matrix). So, the information of all PSDs is combined in a single spectrum, where a PSD with a relative low magnitude has a low contribution to the singular value spectrum. Previously, it was observed that the PSD of 14 sensors shows a lot of noise with considerably lower magnitude than the other sensors. The noise in these PSDs is not reflected in the singular value spectrum because of the minor importance of these sensors.

One remark must be made regarding the computation of the singular value spectrum of the processed signals (after similarity filtering). The singular value spectra shown in Figures 4.10–4.12 are obtained using the central parts of the of the processed signals (i.e. marked area in Figure 4.6). This means that the leading and trailing zeros in the processed signal are removed, resulting in shorter "centered" signals. However, some information is lost when computing the singular value spectrum using the central parts of the signals rather than the full length of the signal. This is demonstrated in Figure 4.9 (first singular value only), where the "regular" spectrum is computed using all 250000 samples in the FFT, while for the "centered" spectrum only 80000 samples are used. It can be observed that some smaller frequency components are overpowered by dominant frequency components. This is the consequence of spectral leakage, which is an inherent effect of the Discrete Fourier Transform (DFT). The DFT assumes that the signals are periodic. In our case, the edges of a centered signal will not be completely zero, meaning that the periodic extension of this signal contains discontinuities, which leads to spectral leakage when computing the DFT/FFT of the centered signal.

Figure 4.13 shows the singular value spectra (first singular value only) of all processed signals of segment 2. The results are presented per dataset, where each sample represents a singular value spectrum. The number of samples in each dataset corresponds to the number of data files after similarity filtering as listed in Table 4.2. The singular value spectra of all samples in a dataset are shown in the form of a spectrogram (top axis). The average spectrum over all samples are shown in the bottom axis. From the figures it can be observed that the dominant frequency components changes over time, presumably due to variations in the environmental conditions. Similar observations can be done from the singular value spectra for segment 1 and 3, which are shown in the appendix, Figure F.7 and Figure F.8, respectively.

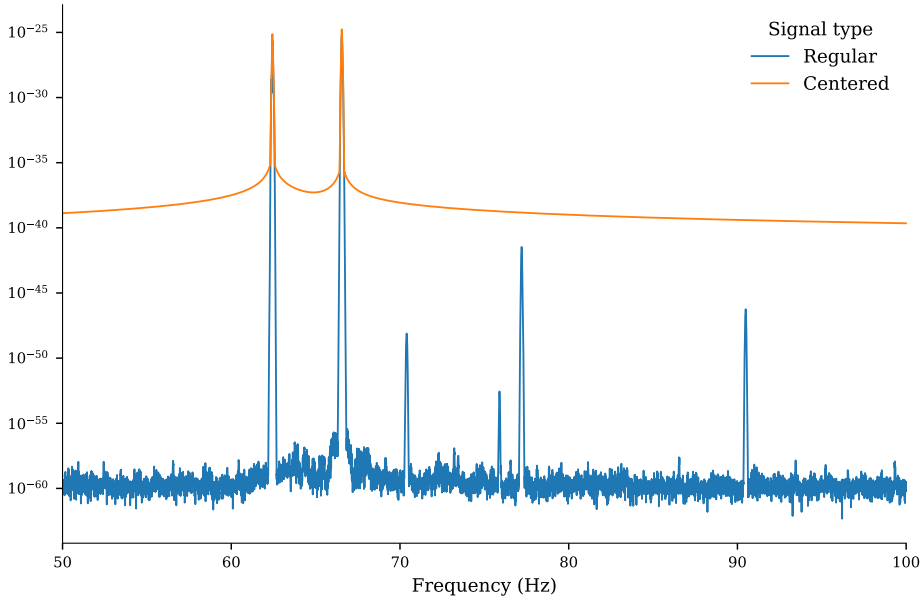
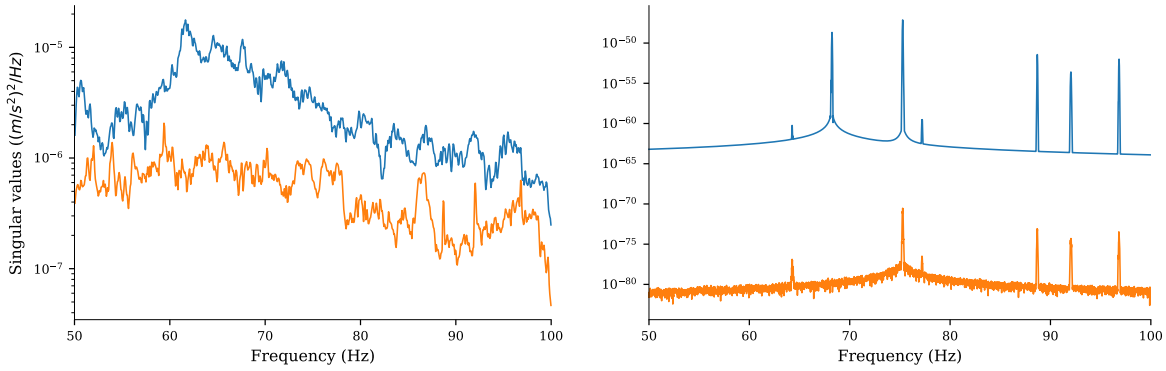
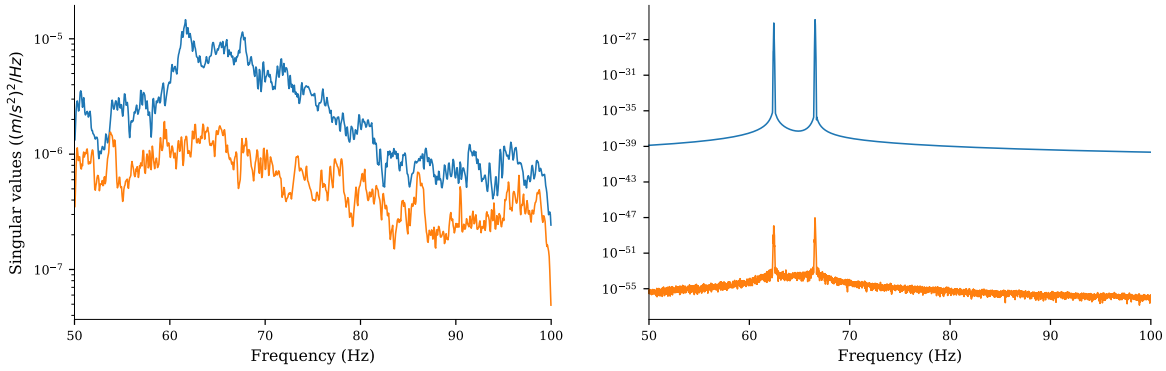


Figure 4.9: Comparison of the singular value spectrum for processed signals in case the full measurement is considered (regular) and when only the central part is used. The central part of the processed signals is illustrated in Figure 4.6.



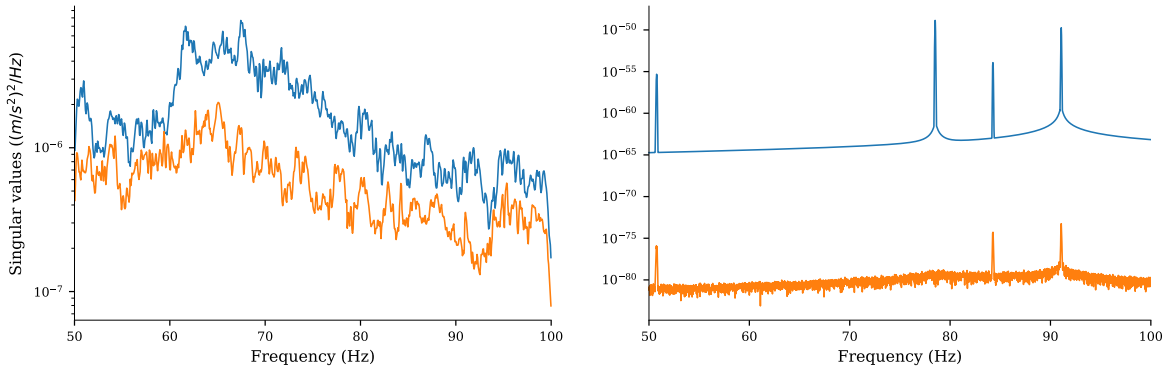
(a) (b)

Figure 4.10: Segment 1 – Singular value spectrum obtained from vibration data (a) before similarity filtering and (b) after similarity filtering. Only the first two singular values of the spectral density matrix are shown.



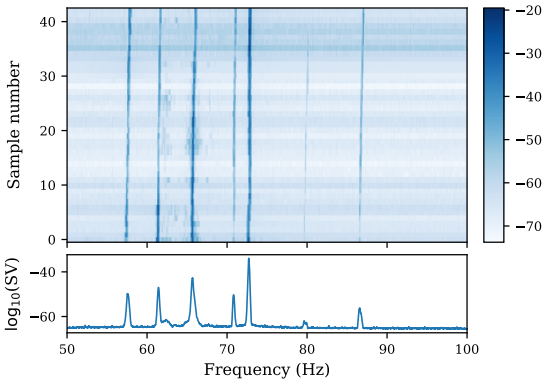
(a) (b)

Figure 4.11: Segment 2 – Singular value spectrum obtained from vibration data (a) before similarity filtering and (b) after similarity filtering. Only the first two singular values of the spectral density matrix are shown.

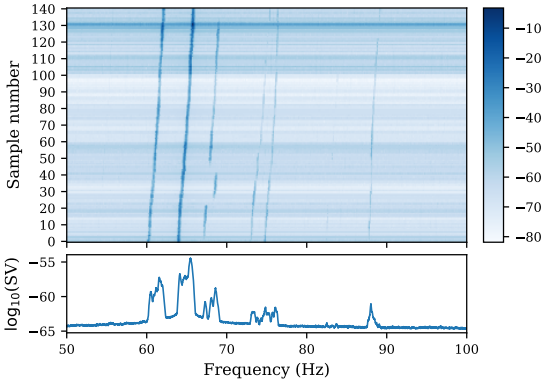


(a) (b)

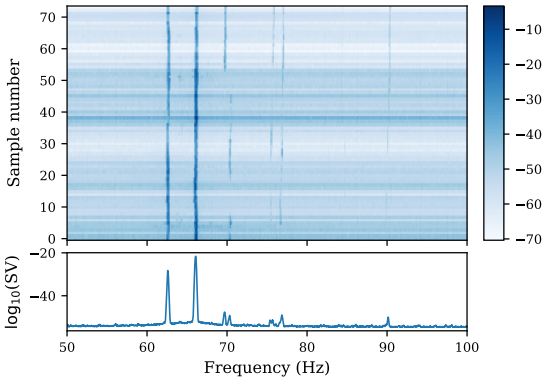
Figure 4.12: Segment 3 – Singular value spectrum obtained from vibration data (a) before similarity filtering and (b) after similarity filtering. Only the first two singular values of the spectral density matrix are shown.



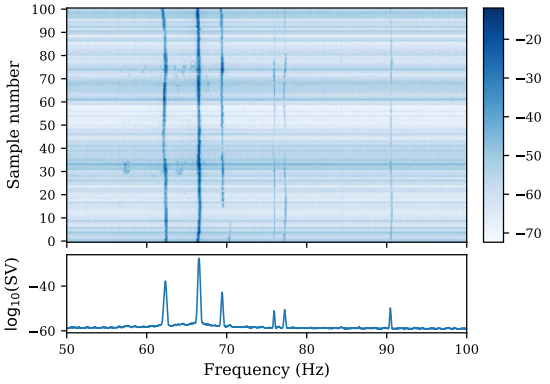
(a) Dataset 1 (0 kg)



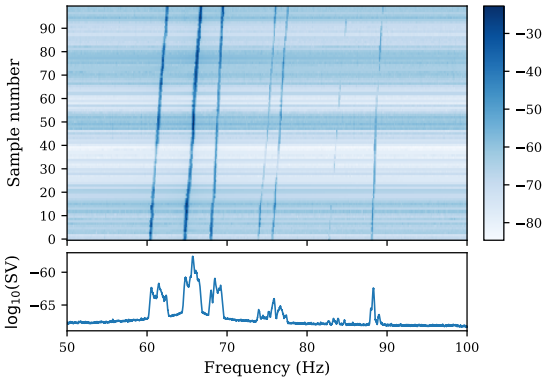
(b) Dataset 2 (100 kg)



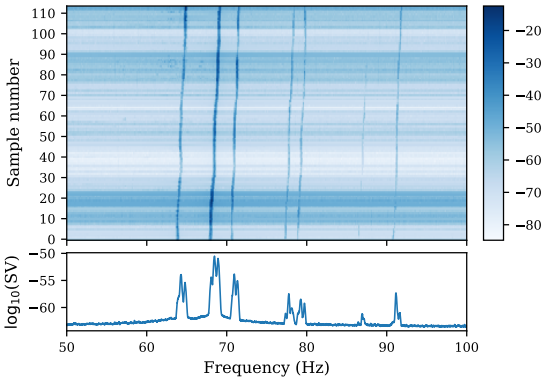
(c) Dataset 3 (50 kg)



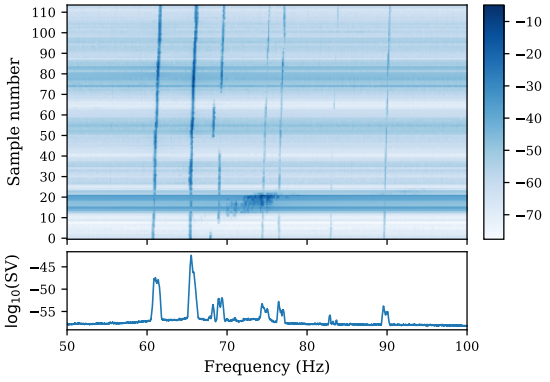
(d) Dataset 4 (0 kg)



(e) Dataset 5 (25 kg)



(f) Dataset 6 (50 kg)



(g) Dataset 7 (75 kg)

Figure 4.13: Singular value spectra of the processed signals from segment 2, separated by dataset (added mass). The top figure shows a spectrogram with observations in chronological order on the vertical axis; the bottom figure shows the average spectrum over all observations of the particular dataset.

4.3.1 Modal Analysis

In this section the modal parameters, natural frequencies and mode shapes, of the bridge are identified using the FDD technique. The identification process is explained in Section 3.2.4, where the modal parameters are identified from a set of singular values and corresponding singular vectors, denoted by $\{\sigma_1(f), \mathbf{u}_1(f)\}$. These sets were obtained in previous section, resulting from the singular value decomposition. An overview of the number of sets that are considered for the modal analysis is provided in Table 4.3, where the number of samples in each dataset corresponds to the number of data files after similarity filtering. In the following, a single set will also be referred to as an observation or a sample.

Table 4.3: Number of measurements in dataset 1 to 7 that are considered for the modal analysis.

	1	2	3	4	5	6	7
Added mass (kg)	0	100	50	0	25	50	75
Number of samples	43	141	74	101	100	114	114

To determine the modes from all observations efficiently, an automatic identification algorithm is implemented in python. This algorithm involves the following steps:

Step 1. Find peaks in the singular value spectrum $\sigma_1(f)$ of a single observation; let f_r, $r = 1, \ldots, N_p$ denote the frequencies of the N_p peaks found. Each peak represents a possible mode, whose natural frequency is estimated by the peak frequency f_r and the corresponding mode shape is estimated by the left singular vector; that is, $\boldsymbol{\phi}_r = \mathbf{u}_1(f_r)$. At the end of this step, we have a set $\{(f_r, \boldsymbol{\phi}_r) : r = 1, \ldots, N_p\}$ consisting of N_p possible modes, where a mode is characterized by its natural frequency and mode shape.

Step 2. Remove non-uniquely identified modes. For this we compute the pairwise distance (4.5) between all possible modes that were identified from a single observation. The distance between two identified modes, $\{f_j, \boldsymbol{\phi}_j\}$ and $\{f_l, \boldsymbol{\phi}_l\}$, is defined as

$$d_{jl} = \frac{|f_j - f_l|}{\max(f_j, f_l)} + 1 - \mathrm{MAC}(\boldsymbol{\phi}_j, \boldsymbol{\phi}_l) \tag{4.5}$$

which considers both the difference in natural frequency and mode shape; the latter is given by the MAC, which is introduced in Section 3.2.5. The distance in equation (4.5) is close to zero for two modes that belong to the same mode. If multiple modes are found with pairwise distances close to zero (lower than a certain threshold Ω_1) then these modes are removed from the set of possible modes except the mode with the largest peak value in the singular value spectrum. In this way a set of certainly different modes is obtained.

Step 3. Repeat **Step 1** and **Step 2** for all the observations, to obtain a set of uniquely determined possible modes for each observation.

Step 4. Cluster the modes from different observations using hierarchical clustering. Again, the pairwise distance between two modes is computed as in (4.5). Average linkage is used to form successive clusters. Modes with similar characteristics will have a distance close to zero.

Step 5. Obtain clusters with similar modes. The distances between observations in a cluster should be small enough for the observations to belong to the same mode, while the distance between two clusters must be large enough for the clusters to represent two distinct modes.

Step 6. Count the number of elements in each cluster and disregard clusters with less elements than a certain threshold; For example, 50% of the total number of observations, which means that a mode must be present in 50% of the observations. In this way spurious modes are removed.

In the end, we have a set of modal parameters $\{(f_n, \boldsymbol{\phi}_n) : n = 1, \ldots, N_m\}$ for each observation. A set consist of N_m pairs formed by a natural frequency f_n and a mode shape $\boldsymbol{\phi}_n$. A step-by-step description of the identification algorithm including visualization is given in Appendix G.1.

Examples of singular value spectra with identified modes are shown in Figure 4.14. The mode shapes corresponding to the identified modes in Figure 4.14c are shown in Figure 4.15. It is noted that the description of a mode shape is limited to the spatial position of the sensors in the considered segment (i.e. the mode shape values are only known at the sensor positions). From the figure it can be observed that the mode shapes have a nonzero amplitude at a single sensor and are zero at the other sensors, confirming the local character of the modes. The same mode shapes are found for all five mass classes, which means that the mode shapes are not sensitive to the added mass. The frequency count of the identified modes are shown in Figure 4.16 in the form of histograms, separated by added mass. Similar figures are presented in Appendix G.2 and Appendix G.3 for the identification results of segment 1 and 3, respectively.

For each segment, the natural frequencies of all 687 observations are summarized in Tables 4.4–4.6 and visualized in Figures 4.17–4.19. Here, the modes are notated as X/SY, where X indicates the mode number and Y refers to the sensor number where the amplitude of the local mode shape is unity (amplitude is zero at the other sensors, visualized in Figure 4.15). This notation is chosen so that a certain mode can directly be related to a position on the bridge deck. The "success rate" in the tables expresses the percentage of successful identification of a certain mode, with respect to the size of the dataset. For example, mode 1/S4 has a success rate of 16% for dataset 1 with size 43, so this mode is successfully determined in 7 of the observations.

Seven modes are identified per bridge segment. However, it appears that not all modes are correctly determined. The natural frequencies of mode 1/S4 (segment 1) and mode 2/S27 (segment 3) show large scatter, which can be attributed to the erroneous identification by the automatic algorithm. Furthermore, mode 1/S32 and mode 7/S25 (both segment 3) could not be determined for all datasets. This is due to the chosen frequency range of 50–100 Hz for band-pass filtering in combination with the varying environmental conditions. More specifically, these two modes have a natural frequency around 50 Hz and 100 Hz, respectively. Due to varying environmental conditions, the natural frequencies change and might fall outside the considered frequency range, and consequently cannot be identified. For correct identification of these modes, the frequency range should have been chosen slightly larger (e.g. from 45 to 105 Hz).

In general, the largest variations in the natural frequencies are found for datasets 2 (100 kg) and 5 (25 kg), and to a lesser extend for datasets 6 (50 kg) and 7 (75 kg). These variations are presumably caused by changes in the environmental conditions (e.g. temperature). The correlation between the natural frequency and temperature is analyzed in more depth in Chapter 8.

4.3.2 Conclusions

In this chapter the Frequency Domain Decomposition is applied on the processed data. First, the PSD and the singular value spectra were analyzed, followed by the identification of the modal parameters (natural frequencies and mode shapes). The results in this section are complemented with the results in Appendix G.

Based on the outcomes, the following observation can be made:

- The power spectrum and singular value spectrum of the processed signals (after band-pass and similarity filtering) are smoother than of the unprocessed signals and dominant frequency components are clearly visible. The dominant frequency components are related to the dynamic characteristics of the system, which are amplified by the application of similarity filtering. Moreover, all the singular value spectra of a segment (for all processed signals) look very similar, the same applies to all the power spectra of a sensor. These observations indicate that in the processed signals the operational variability is reduced (i.e. smoother spectra, and less variations between the signals), and the dynamic characteristic of the system are amplified (i.e. dominant frequency components are clearly visible). So, the similarity filtering seems to work for its intended goal; that is, to reduce the operational variability in a measurement and at the same time enhance the similarities (i.e. the dynamic characteristics of the system).

  It must be noted that although the resonances in the spectra are clearly visible, the "shape" of the spectra are highly modified. This means that important information for damage detection might be lost due to the similarity filtering, given that changes in the structural properties could also affect the shape of a spectrum.

- Figures F.1–F.4 in the appendix show that most of the sensors have a power spectrum with one dominant frequency component only, while the other sensors have low quality spectra (i.e. large variability and very low magnitude). In particular, the following sensors have considerably lower quality power spectra

for all processed signals: 4, 5, 6, 9, 11, 14, 20, 21, 22, 24, 27, 29. Furthermore, the power spectra of sensors 25, 30 and 32 have a dominant frequency component close to one of the ends of the frequency range 50–100 Hz, which is not consistently observed in all processed signals. Those 15 sensors are not useful for damage detection due to the large variability in spectral density that are obviously not related to damage.

So, the similarity filtering did not succeed in amplifying a dominant frequency component in the signals of 15 sensors, which might be related to the local modes that were excited (or were not excited) at these sensors. However, no obvious reason is found why these sensors have low quality power spectra and the other sensors not.

From the figures it can also be observed that the dominant frequency components in the power spectra changes over time, presumably due to the variations in environmental conditions.

- Seven modes were identified per bridge segment. However, the natural frequencies of modes 1/S4 (segment 1) and 2/S27 (segment 3) showed large scatter, and modes 1/S32 and 7/S25 (both segment 3) were not consistently determined in all datasets. The large scatter is attributed to the erroneous identification of these modes by the automatic algorithm. The inability to identify the modes 1/S32 and 7/S25 in all datasets is due to the chosen frequency range of 50–100 Hz for band-pass filtering in combination with the varying environmental conditions. For correct identification of these modes, the frequency range should have been chosen slightly larger (e.g. from 45 to 105 Hz).

  Those four modes will be disregarded in the analyses performed in the following chapters.

- The identified natural frequencies (Figures 4.17–4.19) show large variations across the different observations, especially for dataset 2 (100 kg) and 5 (25 kg). The standard deviations of the natural frequencies of these two datasets are considerably larger compared to the other datasets. The change of natural frequency over time might be the result of variations in the environmental conditions (e.g. temperature). The correlation between a natural frequency and the temperature is analyzed in more depth in Chapter 8. It must be noted that the variation within a dataset (with constant added mass) will not be the result of the added mass.

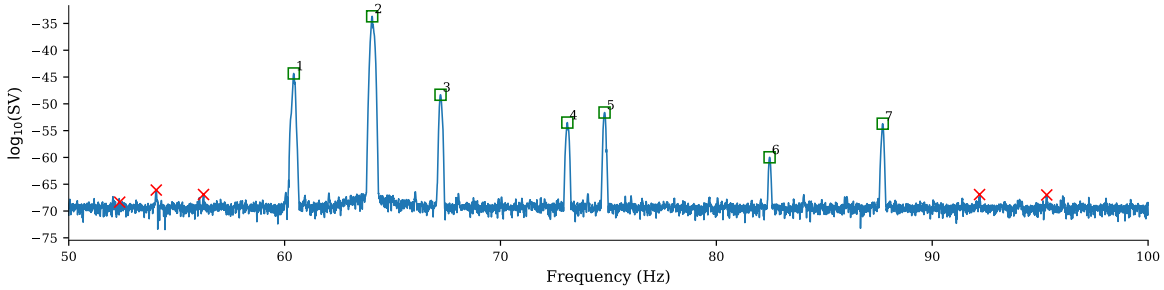
- The measurements of dataset 1 and 4 both are related to the 0 kg mass class. The average value of the natural frequencies of these two datasets show absolute differences of more than 4 Hz (maximum 5.1 Hz), this is clearly visible in Figure 4.16a. The datasets are both obtained on March 23, 2017, dataset 1 in the morning from 8:29 to 9:48 and dataset 4 in the afternoon from 14:12 to 15:36, see Table 2.1.

  The measurements of dataset 3 and 6 are both related to the 50 kg mass class. The average value of the natural frequencies of these datasets show difference of more than 2 Hz (maximum 2.6 Hz), this is clearly visible in Figure 4.16c. Dataset 3 is measured on March 23, 2017 from 12:40 to 13:40 and dataset 6 is measured on March 28, 2017 from 11:34 to 12:53. So, both sets are measured around the same time of the day but on different days.

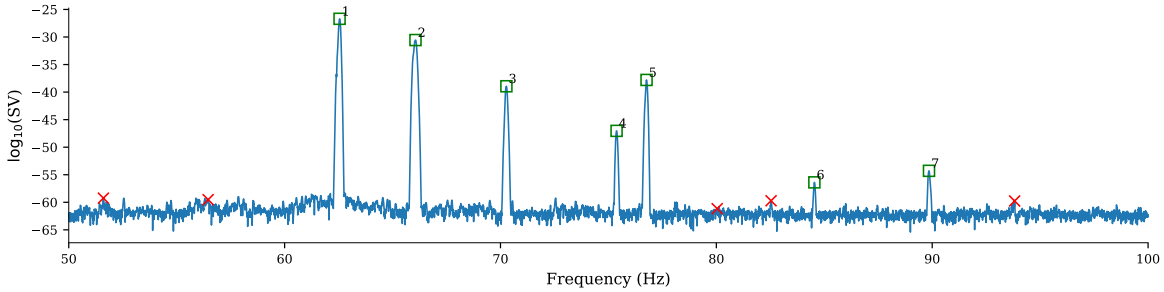
  Obviously, the frequency differences in these two cases cannot be caused by the presence of an added mass. Hence, these must be the result of other factors, such as varying environmental conditions (e.g. temperature). The correlations between the natural frequencies and the temperature is analyzed in more depth in Chapter 8.

- Figure 4.15, and Figures G.6 and G.9 in the appendix, show that the mode shapes have a nonzero amplitude at a single sensor and are zero at the other sensors. This confirms the local character of the modes (with respect to the spatial size of the sensor network). The mode shapes are not sensitive to the added mass, because the same mode shapes are found for all five mass classes. By analyzing the figures of the mode shapes it can be concluded that no modes were found with a local mode shape having a nonzero amplitude at one of the sensors with low quality PSD (i.e. the sensors mentioned in the second bullet).

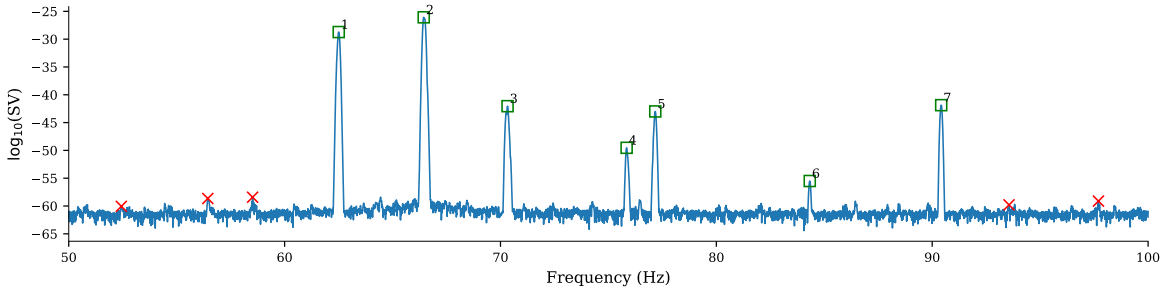
  It is noted that the description of the mode shapes is limited to the spatial position of the sensors in a segment, but the same mode shapes would have been found if the signals from all sensors were used in the FDD (not shown here).



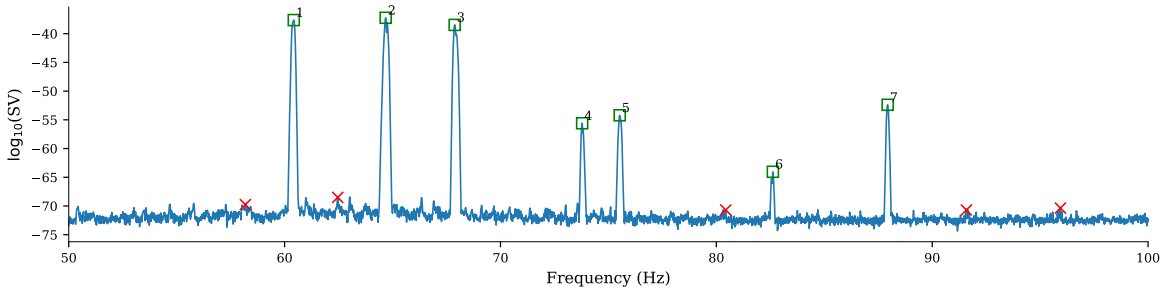
(a) Example 5: dataset 2 (100 kg)



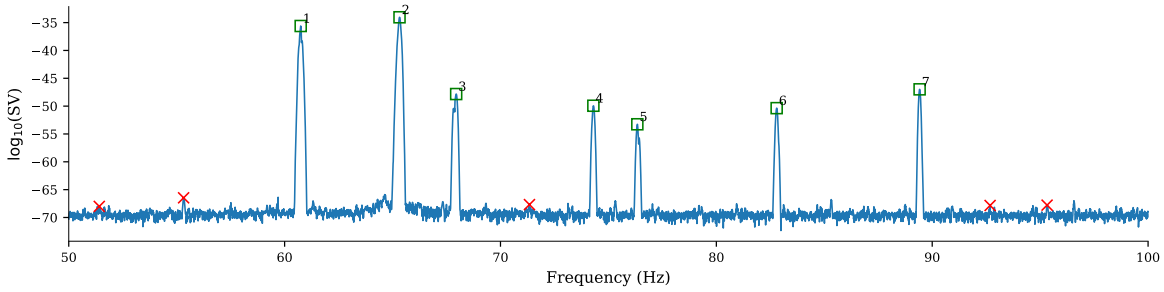
(b) Example 3: dataset 3 (50 kg)



(c) Example 1: dataset 4 (0 kg)



(d) Example 2: dataset 5 (25 kg)



(e) Example 4: dataset 7 (75 kg)

Figure 4.14: Examples of the singular value spectrum of the processed signals from segment 2, including the identified modes (□). The red crosses indicate peaks that are removed during the modal clustering.

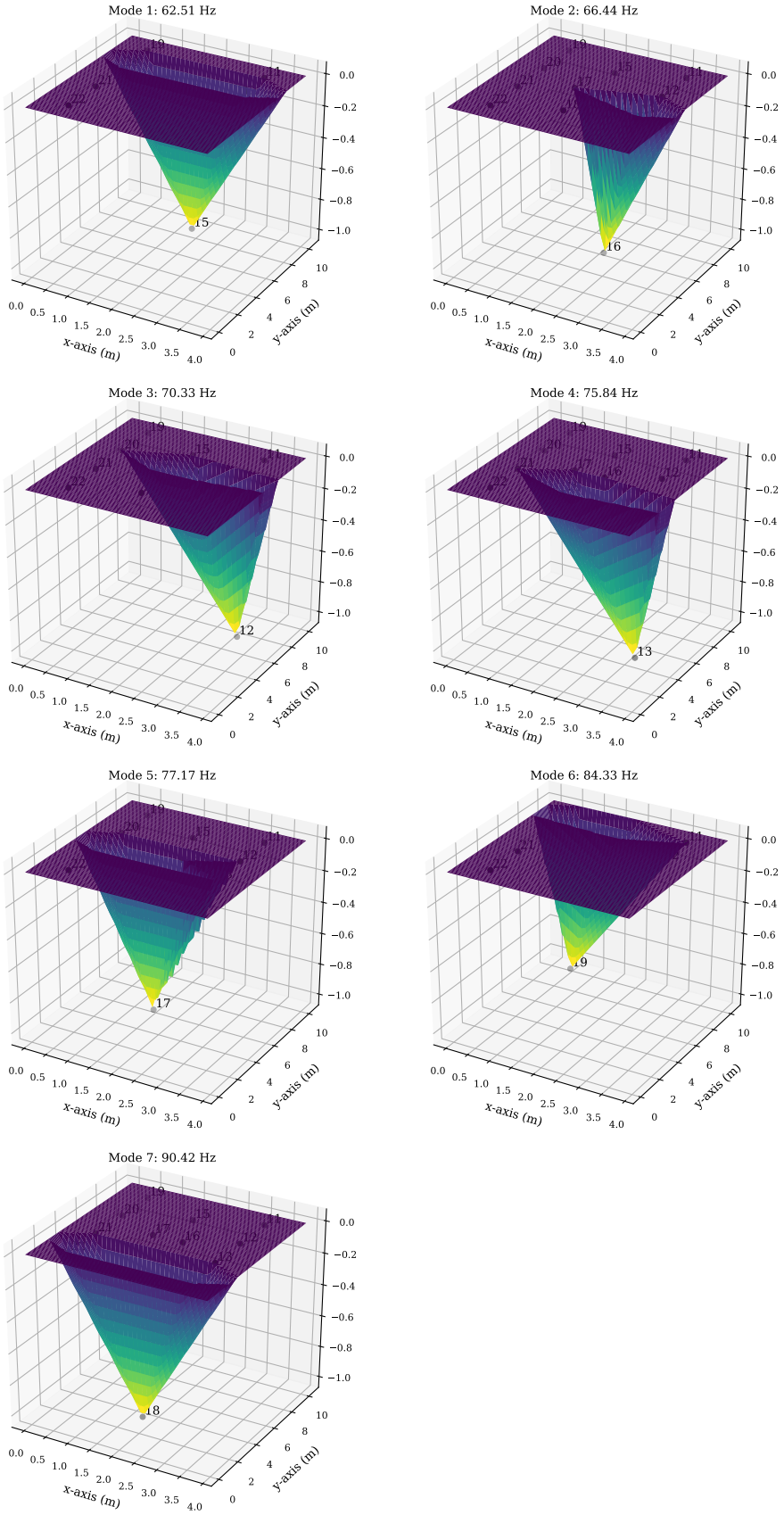
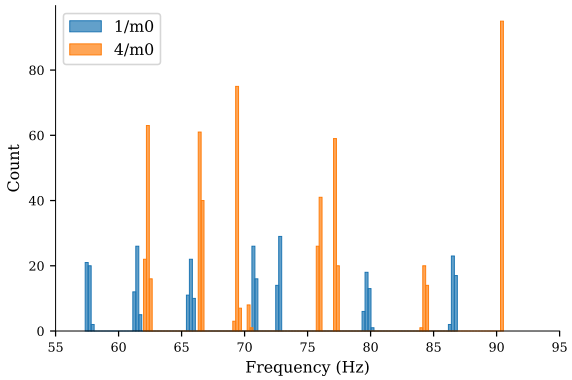
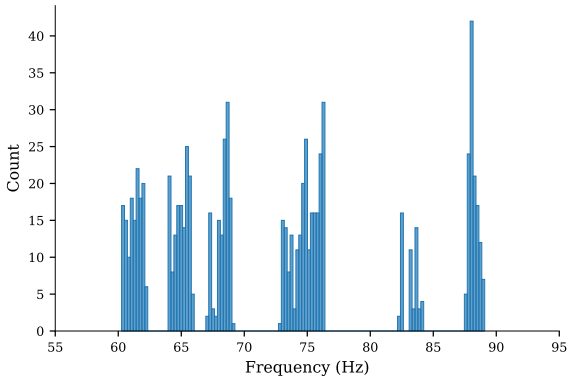


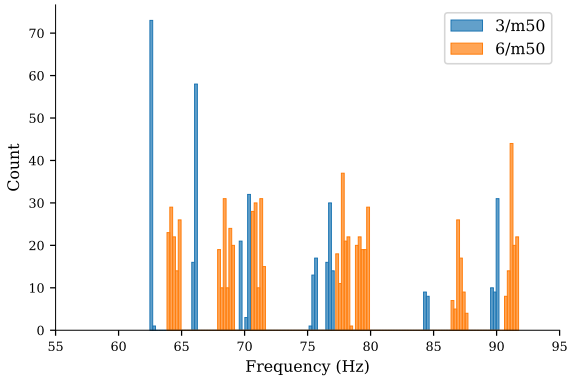
Figure 4.15: Local mode shapes of segment 2, corresponding to the identified modes in Figure 4.14c.



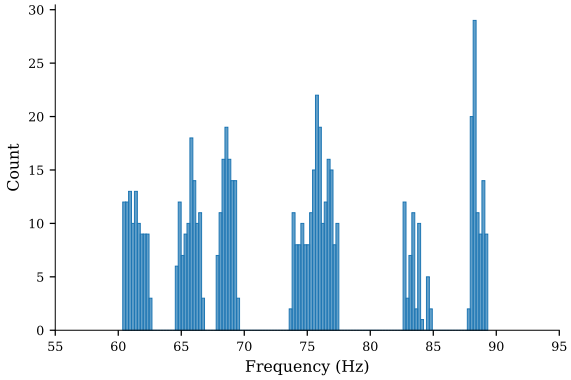
(a) Datasets 1 and 4 (0 kg)



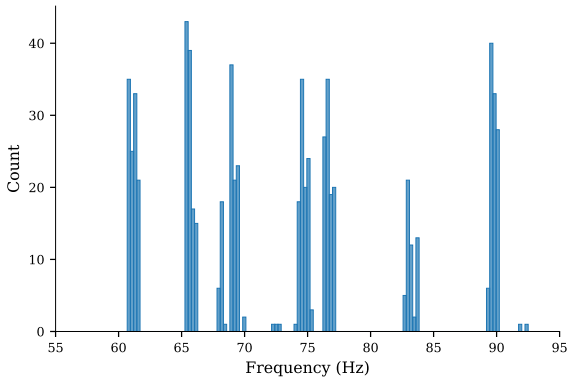
(b) Dataset 2 (100 kg)



(c) Datasets 3 and 6 (50 kg)



(d) Dataset 5 (25 kg)



(e) Dataset 7 (75 kg)

Figure 4.16: Frequency count of identified modes from the singular value spectra of the processed signals from segment 2, separated by added mass class.

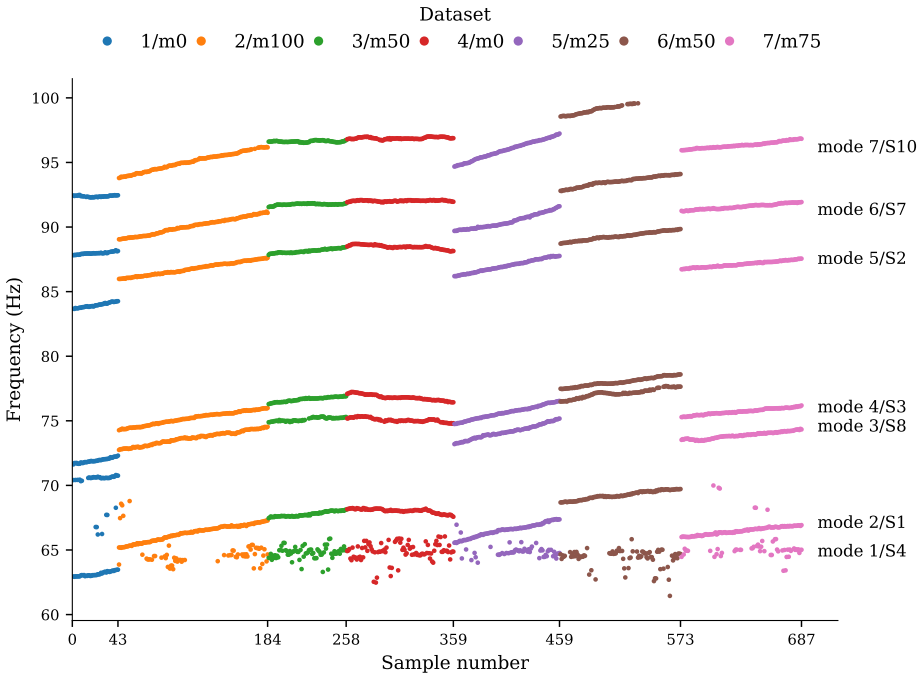


Figure 4.17: Segment 1 – Identified natural frequencies from processed data sets. The horizontal axis represents the samples/observations in chronological order.

Table 4.4: Segment 1 – Automatic modal analysis results for different datasets. The mean value and standard deviation (Std) of the natural frequencies are specified, together with the success rate.

Mode		Dataset/added mass						
		1/m0	2/m100	3/m50	4/m0	5/m25	6/m50	7/m75
1/S4*	Succes rate (%)	16	48	100	97	56	72	50
	Mean (Hz)	67.10	64.76	64.79	64.87	64.89	64.39	65.37
	Std (Hz)	0.80	1.09	0.46	0.69	0.46	0.71	1.36
2/S1	Succes rate (%)	100	100	100	100	100	100	100
	Mean (Hz)	63.15	66.30	67.78	68.03	66.52	69.24	66.47
	Std (Hz)	0.17	0.61	0.19	0.16	0.51	0.33	0.29
3/S8	Succes rate (%)	81	100	95	97	99	94	100
	Mean (Hz)	70.59	73.66	75.12	75.07	74.15	77.12	73.87
	Std (Hz)	0.11	0.51	0.14	0.15	0.54	0.33	0.26
4/S3	Succes rate (%)	100	100	100	100	100	100	100
	Mean (Hz)	71.93	75.20	76.66	76.82	75.65	78.02	75.72
	Std (Hz)	0.18	0.50	0.17	0.21	0.51	0.34	0.24
5/S2	Succes rate (%)	100	100	100	100	100	100	100
	Mean (Hz)	83.95	86.78	88.15	88.47	87.00	89.30	87.12
	Std (Hz)	0.18	0.49	0.15	0.15	0.48	0.32	0.23
6/S7	Succes rate (%)	98	100	100	100	100	100	100
	Mean (Hz)	87.98	90.08	91.76	92.03	90.45	93.54	91.58
	Std (Hz)	0.09	0.63	0.06	0.04	0.56	0.38	0.20
7/S10	Succes rate (%)	98	96	100	99	100	59	100
	Mean (Hz)	92.38	95.04	96.63	96.88	95.97	99.05	96.35
	Std (Hz)	0.06	0.70	0.06	0.08	0.73	0.33	0.25

* Not well determined mode, natural frequencies show large scatter.

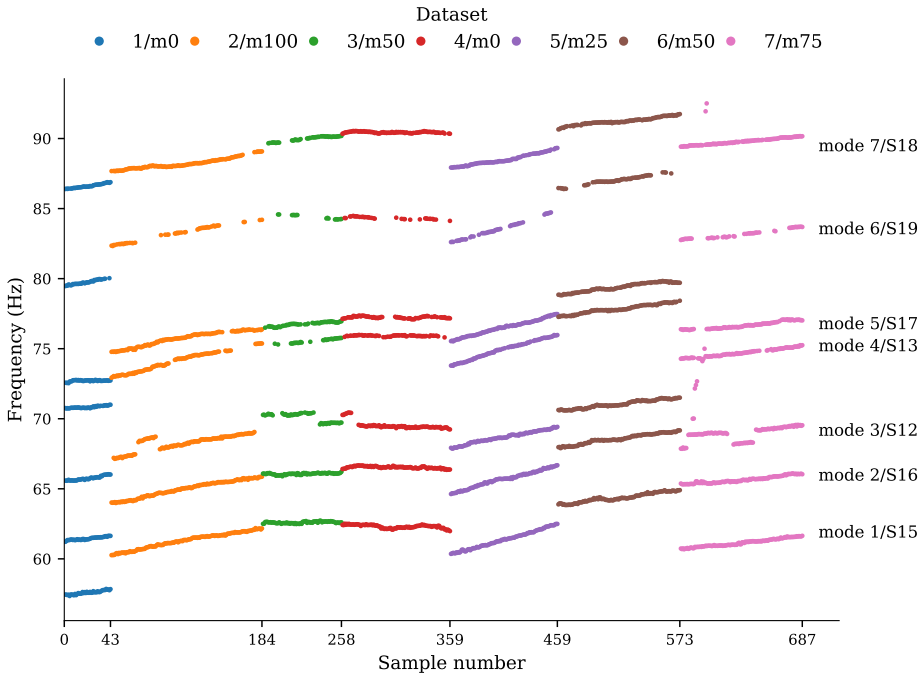


Figure 4.18: Segment 2 – Identified natural frequencies from processed data sets. The horizontal axis represents the samples/observations in chronological order.

Table 4.5: Segment 2 – Automatic modal analysis results for different datasets. The mean value and standard deviation (Std) of the natural frequencies are specified, together with the success rate.

Mode		Dataset/added mass						
		1/m0	2/m100	3/m50	4/m0	5/m25	6/m50	7/m75
1/S15	Succes rate (%)	100	100	100	100	100	100	100
	Mean (Hz)	57.60	61.29	62.60	62.31	61.37	64.37	61.15
	Std (Hz)	0.13	0.55	0.06	0.12	0.63	0.33	0.28
2/S16**	Succes rate (%)	100	100	100	100	100	100	100
	Mean (Hz)	61.45	64.99	66.07	66.53	65.68	68.57	65.65
	Std (Hz)	0.11	0.58	0.06	0.08	0.58	0.36	0.24
3/S12	Succes rate (%)	100	90	76	93	100	100	93
	Mean (Hz)	65.76	68.29	70.08	69.52	68.68	71.04	68.93
	Std (Hz)	0.15	0.53	0.32	0.29	0.43	0.29	0.48
4/S13	Succes rate (%)	98	77	42	66	93	96	82
	Mean (Hz)	70.83	74.04	75.56	75.92	74.92	77.84	74.72
	Std (Hz)	0.08	0.70	0.15	0.04	0.67	0.31	0.28
5/S17**	Succes rate (%)	100	92	81	78	100	96	89
	Mean (Hz)	72.71	75.69	76.76	77.24	76.50	79.38	76.68
	Std (Hz)	0.06	0.52	0.14	0.07	0.55	0.32	0.24
6/S19*	Succes rate (%)	88	38	23	35	53	60	59
	Mean (Hz)	79.74	83.20	84.40	84.33	83.48	87.01	81.66
	Std (Hz)	0.16	0.58	0.15	0.09	0.62	0.29	4.36
7/S18	Succes rate (%)	98	91	68	94	94	95	94
	Mean (Hz)	86.61	88.20	90.00	90.44	88.48	91.24	89.78
	Std (Hz)	0.15	0.36	0.18	0.05	0.40	0.26	0.23

* Inconsistently determined mode, low success rate.
** Mode with a mode shape having a nonzero displacement close to the added mass; at the position of sensor 16 and 17.

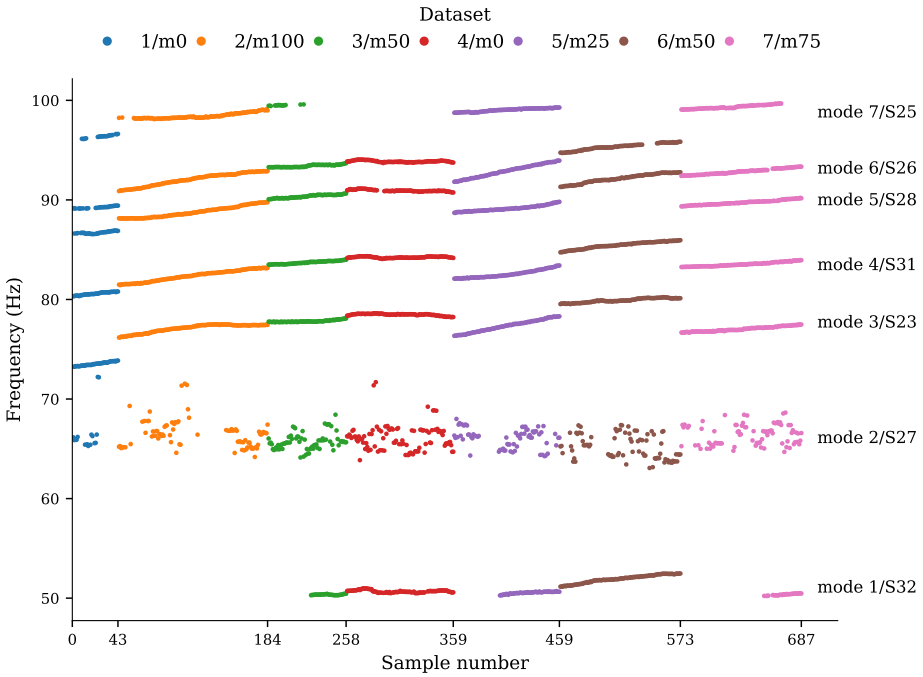


Figure 4.19: Segment 3 – Identified natural frequencies from processed data sets. The horizontal axis represents the samples/observations in chronological order.

Table 4.6: Segment 3 – Automatic modal analysis results for different datasets. The mean value and standard deviation (Std) of the natural frequencies are specified, together with the success rate.

Mode		Dataset/added mass						
		1/m0	2/m100	3/m50	4/m0	5/m25	6/m50	7/m75
1/S32**	Succes rate (%)	0	0	46	95	57	100	28
	Mean (Hz)	-	-	50.35	50.69	50.54	51.93	50.36
	Std (Hz)	-	-	0.04	0.13	0.11	0.40	0.08
2/S27*	Succes rate (%)	37	60	97	98	70	75	75
	Mean (Hz)	66.64	66.51	65.73	66.15	66.09	65.15	66.42
	Std (Hz)	2.20	1.42	0.85	1.31	0.96	1.22	1.02
3/S23	Succes rate (%)	100	100	100	100	100	100	100
	Mean (Hz)	73.53	77.09	77.82	78.47	77.34	79.91	77.03
	Std (Hz)	0.19	0.41	0.10	0.09	0.60	0.21	0.24
4/S31	Succes rate (%)	100	100	100	100	100	100	100
	Mean (Hz)	80.59	82.37	83.71	84.24	82.56	85.45	83.53
	Std (Hz)	0.15	0.54	0.15	0.07	0.39	0.34	0.21
5/S28	Succes rate (%)	86	100	100	92	100	100	96
	Mean (Hz)	86.72	88.79	90.36	90.94	89.14	92.15	89.73
	Std (Hz)	0.11	0.54	0.16	0.09	0.28	0.46	0.23
6/S26	Succes rate (%)	72	100	100	98	100	89	96
	Mean (Hz)	89.26	92.04	93.41	93.87	92.90	95.32	92.84
	Std (Hz)	0.10	0.63	0.12	0.09	0.64	0.33	0.28
7/S25**	Succes rate (%)	53	91	22	0	100	0	80
	Mean (Hz)	96.38	98.44	99.50	-	99.03	-	99.33
	Std (Hz)	0.15	0.27	0.04	-	0.17	-	0.18

* Not well determined mode, natural frequencies show large scatter.
** Natural frequencies could not be determined for some datasets, as they fall outside the range 50–100 Hz.

5

Feature Engineering

In this chapter some important aspects of feature engineering are briefly discussed. Feature engineering is the general process of generating proper inputs for the machine learning algorithms. The inputs of the algorithms are then referred to as *features*; in the context of structural health monitoring they are often referred to as *damage-sensitive features*. A damage-sensitive feature is some quantity extracted from the measured system response that indicates the presence of damage in a structure. These features vary considerably in their complexity. Ideally, we wish to have low-dimensional feature vectors that are highly sensitive to the condition of the structure and insensitive to all forms of operational and environmental variability. The dimension of a feature vector refers to the number of scalar quantities in the vector. Many techniques used for machine learning have difficulties in dealing with feature vectors of high dimensionality; this problem is usually referred to as the *curse of dimensionality*. The curse is simply that, in order to obtain acceptable performance of these techniques, the number of training data theoretically grows exponentially with the dimensionality p of the feature space. To get an idea of this curse, suppose a one-dimensional interval needs, say, N equidistant points to be properly described, the corresponding two-dimensional square will need N^2, the three-dimensional cube N^3, and so on. This would mean that for a p-dimensional space we need N^p samples. Another problem associated with high-dimensional spaces is that, in practice, due to the lack of enough training data points some regions in the feature space may be sparsely represented in the data set [46].

This curse is problematic for most probability density estimation techniques, as well as for classifiers that make their decision based on a certain distance metric. In a high-dimensional space two samples are, on average, further located from each other compared to the same number of points in a low-dimensional space. When the distance between samples grows, classification becomes more difficult because prediction for new samples are less likely to be based on learning from similar training samples.

From a practical point of view, there are two solutions to this problem. The first solution is to obtain adequate training data sets. This would mean that we have to acquire more data, which is not always possible due to the size and expense of the measurement campaign. The second solution is to reduce the dimensions of the features to a point where the available data is sufficient. The latter approach is often more feasible and is an important aspect in feature engineering. A large range of techniques exists for dimensionality reduction, which differ in their criteria used for deciding which information should be preserved. Obviously, the dimensions of the feature space should be reduced without the loss of important, damage-related information.

Feature extraction and *feature selection* are the two main classes for reducing the dimensions of the feature space. Feature extraction refers to the process of transforming the measured data into some alternative lower-dimensional form where the correlation with the damage is more readily observed. This process often also includes the fusion of data from multiple sensors. Feature selection is the process of determining which feature to use in the damage detection process. Ideally one should select a feature that is sensitive to the presence of damage in the structure and insensitive to all forms of operational and environmental variability. However, in practice, features that are sensitive to damage are also sensitive to changes in the structural response that are not related to damage. This issue often requires an additional processing step after the feature engineering, which will be discussed in Chapter 8 (data normalization).

Although feature extraction and selection both aim at reducing the dimensions of the feature space, they use a different approach to achieve this. Feature extraction maps (in a linear or nonlinear manner) all p ob-

served variables to a smaller number of variables, while feature selection selects a subset of variables. The main advantage of feature extraction is that it is not necessarily axis-aligned, implying that the process may involve nonlinear mappings. Principal Component Analysis (PCA) is one of the most-popular method for feature extraction and dimensionality reduction in general. The theory of PCA is briefly described in Appendix C. PCA is also often used for visualization of the data, where the dimensions of the features are reduced to two or three. Depending on which information content is preserved by the transformation, the important structure of the data may be accessible by visual inspection.

In our case, the feature extraction process is actually already performed in Chapter 4. In that chapter, the processed signals (i.e. time series after band-pass and similarity filtering) are transformed to alternative forms that are more useful for damage detection. For the signals per sensor, the Power Spectral Density (PSD) functions were computed for the frequency range of 50 to 100 Hz. The information of multiple sensors (sensors per bridge segment) was combined into a singular value spectrum; that is, the first/largest singular values of the spectral density matrix computed from the processed signals of multiple sensors. The computed spectra for the processed signals consist of 4000 spectral lines between 50 and 100 Hz (frequency resolution 0.0125 Hz). Using all the spectral lines as damage-sensitive features results in a 4000-dimensional feature vector, which is not desirable. The very high dimensionality of the feature space in combination with the the small number of observations might result in bad performance of the machine learning algorithms. Therefore, the dimensions of the feature space are further reduced using a simple but pragmatic feature selection process. The idea of the feature selection process for a spectrum is rather straightforward and consists of selecting the spectral lines centered around the peaks in the spectrum, as these are known to be sensitive to the damage of a structure. In this way, the dimension of the feature vector is reduced without losing the damage-related information. However, the peaks in a response spectrum are also sensitive to changes in the structural response that are not related to damage. Recall that in Chapter 4, estimates of the bridge's natural frequencies were identified from the peaks in a singular value spectrum. So, in the particular case that we select the spectral lines at the peaks only, the features correspond to estimates of the resonance frequencies of the considered structure. Using the natural frequencies of the structure as damage sensitive features results in low dimensional feature vectors, which are desirable for most machine learning applications.

In this thesis, features selected from a singular value spectrum are used for the classification task in Chapter 6. The resulting feature space is high-dimensional which is problematic for many machine learning algorithms. However, not for the Support Vector Machine (SVM) algorithm that will be applied for the classification problem. The computational complexity of the SVM is independent of the input feature space (or the kernel space where the input feature space is mapped), hence bypassing the curse of dimensionality.

On the other hand, the natural frequencies are used for the novelty detection in Chapter 7 because of their low dimensionality. The employed method for novelty detection is based on the Mahalanobis Squared Distance (MSD), which involves the inverse of a covariance matrix. An accurate estimate of the covariance matrix requires a lot of data, especially for high dimensional feature vectors. Therefore, the natural frequencies of the structure are used rather than the singular values spectrum. The former yields a feature vector with only five or six elements while the for latter the feature vector easily contains more than 200 elements.

5.1 Visual Inspection of the Spectra

In previous chapter, the spectra associated with the different mass classes were considered, but the results of the different mass classes were not compared yet. Here, the singular value spectra and PSD of four mass classes will be compared to get an idea of the possible class separability.

The singular value spectra of the processed signals were previously shown in the form of spectrograms, displaying the frequency content across a set; see Figure 4.13 for the data of segment 2 and Figures F.7 and F.8 in the appendix for the data of segment 1 and 3, respectively. Based on these figures the frequency range of 50 to 100 Hz will be further reduced as part of the feature selection process. Recall that the frequency range of 50 to 100 Hz was determined using the Frequency Domain Decomposition (FDD) in which the spectral density matrix was constructed with all the sensors. However, for the processed signals after similarity filtering it was chosen to perform the FDD for each bridge segment separately (using only a subset of the sensors for the construction of the spectral density matrix). It then appears that the resonance frequencies of the three segments do not coincide and that these frequency components appear in a smaller frequency range. By visual inspection of the spectrograms it is chosen to consider the following frequency ranges: 65 to 100 Hz for segment 1, 55 to 72 Hz for segment 2, and 75 to 95 Hz for segment 3. The frequency ranges of the three segments are not the same, so are the number of spectral lines.

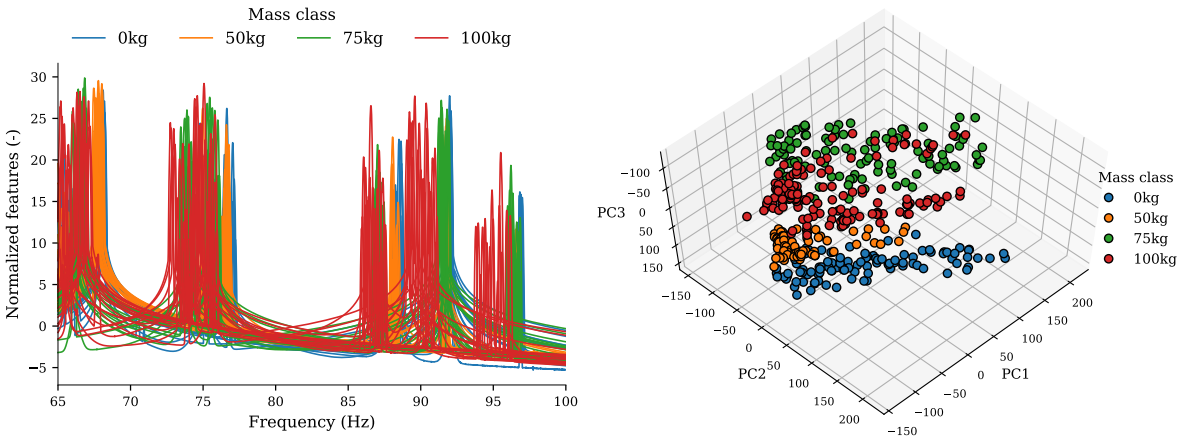
In Figures 5.1a–5.3a overlaid plots of the normalized singular value spectra for four different mass classes are shown. In these figures we would expect a frequency shift of the peaks towards the left for increasing weight of the mass; that is, an increase in mass leads to a reduction of the natural frequency, which is based on the formula for the natural frequency of a SDOF system $\omega_n = \sqrt{k/m}$. Moreover, it is expected that this effect will be most apparent for the spectra of segment 2 in which the mass is located. The expected behavior can indeed be observed from the spectra of segment 2 for the peaks around 66 Hz, but is less obvious for the other peaks. It can also be seen that there is a large spread in the location of the peaks for the 100 kg mass class. Similar frequency shifts can be observed in the spectra of segment 1 and segment 3, although this is less clear in the former. The observations for segment 1 and 3 seem a bit strange because the mass is located in segment 2 and it is unlikely that the mass will cause these changes, given that the mass is very small compared to the weight of a segment and the segments are separated by stiff transverse beams. It is more likely that the observed frequency shifts are the result of changes in the environmental conditions (e.g. temperature). The correlation between natural frequency and the temperature will be investigated in Chapter 8.

Next to the usual representation of the spectra, the data is also visualized in an alternative form in an attempt to show the separability of the mass classes. The PCA is employed to obtain the first three principal components of the data matrix (i.e. matrix holding the singular value spectra). In Figures 5.1b–5.3b the three principal components are shown. From these figures it can be seen that the observations from the different mass classes cluster quite well. In particular, the observations without mass (0 kg class) is clearly separated from the observations with mass. It is noted that the PCA is solely applied as visualization tool in this thesis. The PCA could also be applied as feature extraction tool to obtain low-dimensional feature vectors; for example, the three principal components form a three-dimensional feature vector. However, the PCA involves the estimation of a covariance matrix (see Appendix C), which requires a lot of data for an accurate estimate. Besides, the damage-related information may be lost by the projection of the data on the principal components. Therefore, the PCA will not be used for dimensionality reduction of the high-dimensional feature vectors. Instead, another method will be used for feature selection as will be explained later.

The PSDs of the processed signals for each sensor were also shown in the form of spectrograms, see Figures F.1–F.4 in the appendix. From these figures it was observed that for 17 sensors the PSDs have one dominant frequency component (i.e. resonance frequency of the structure), while for the other 15 sensors the spectra is of low quality. The latter spectra are not useful for damage detection and will be disregarded. Figure 5.4 shows overlaid plots of the normalized PSD for four different mass classes. Similar plots with the data grouped by measurement date are shown in Figures F.5 and F.6. From these figures the shift in resonance frequency is clearly visible for all sensors. For most of the sensors, the shift is consistent with the increasing weight of the added mass. The shift in resonance frequency is not more pronounced for the sensors closest to the added mass (i.e. sensors 16 and 17). For example, for sensor 26 and 28 (located in the middle of segment 3) the shift in the resonance frequency is also very consistent with the increase of the mass. Besides in Figure F.6 in the appendix the opposite behavior is observed; that is, the resonance frequencies for the 50 kg mass are higher than for the 25 kg mass, while it is expected that the heavier mass leads to reduced resonance frequencies. The location of the peaks in the spectra is very distinct between the two mass classes; the difference between them is for all sensors about 2 Hz. Based on these two observations it is questionable whether the added mass has an observable effect on the spectra. It fact, it seems that the the shift in frequency reflects the time moment at which the data is measured. This statement is supported by the following observations:

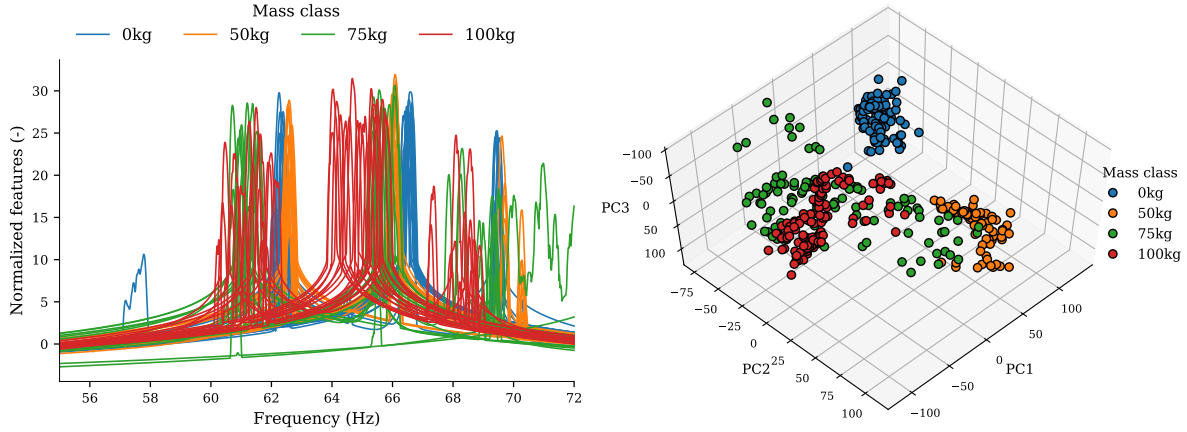
- The resonance frequencies also change considerably in the spectra of the same added mass class, which will not be the result of the added mass. Considering the measurement time corresponding to the spectra it appears that the resonance frequencies generally increase while time elapsed.
- The resonance frequencies in the spectra of a certain dataset/mass class are generally the lowest for the dataset that is measured first (i.e. in the morning). For example, the measurements on March 23, 2017 are performed with a mass of 0 kg, 100 kg, 50 kg, and again 0 kg between 08:29 and 15:37 (see Table 2.1 for the time periods per dataset). From Figure F.5 it can be observed that the resonance frequencies in the spectra corresponding to these measurements also occur in this order, with the lowest frequency for the 0 kg class measured first. The same can be observed in Figure F.6 for the measurements on March 28, 2017, where the 25 kg class is measured first followed by the 50 kg class.

Of course, the time moment itself does not cause the observed frequency shifts; it will be the variables that are related to the time moments, with the environmental conditions presumably as the main source. The correlation between natural frequency and the temperature will be investigated in Chapter 8.



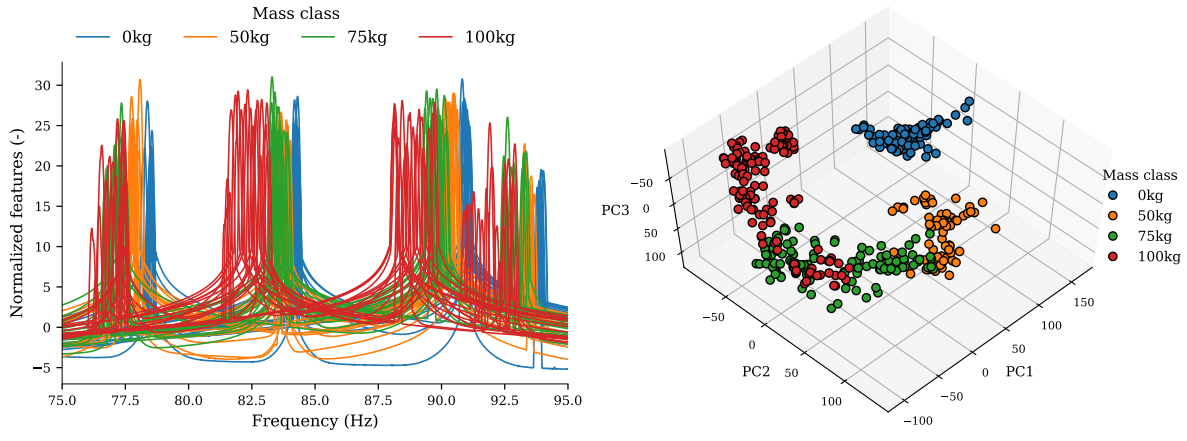
(a) (b)

Figure 5.1: Normalized features for bridge segment 1. In (a) the features represent 2800 frequency lines from singular value spectra between 65 and 100 Hz, and in (b) the first three principal components are shown which explain 44% of the variance in the original 2800 features.



(a) (b)

Figure 5.2: Normalized features for bridge segment 2. In (a) the features represent 1360 frequency lines from singular value spectra between 55 and 72 Hz, and in (b) the first three principal components are shown which explain 44% of the variance in the original 1360 features.



(a) (b)

Figure 5.3: Normalized features for bridge segment 3. In (a) the features represent 1600 frequency lines from singular value spectra between 75 and 95 Hz, and in (b) the first three principal components are shown which explain 44% of the variance in the original 1600 features.

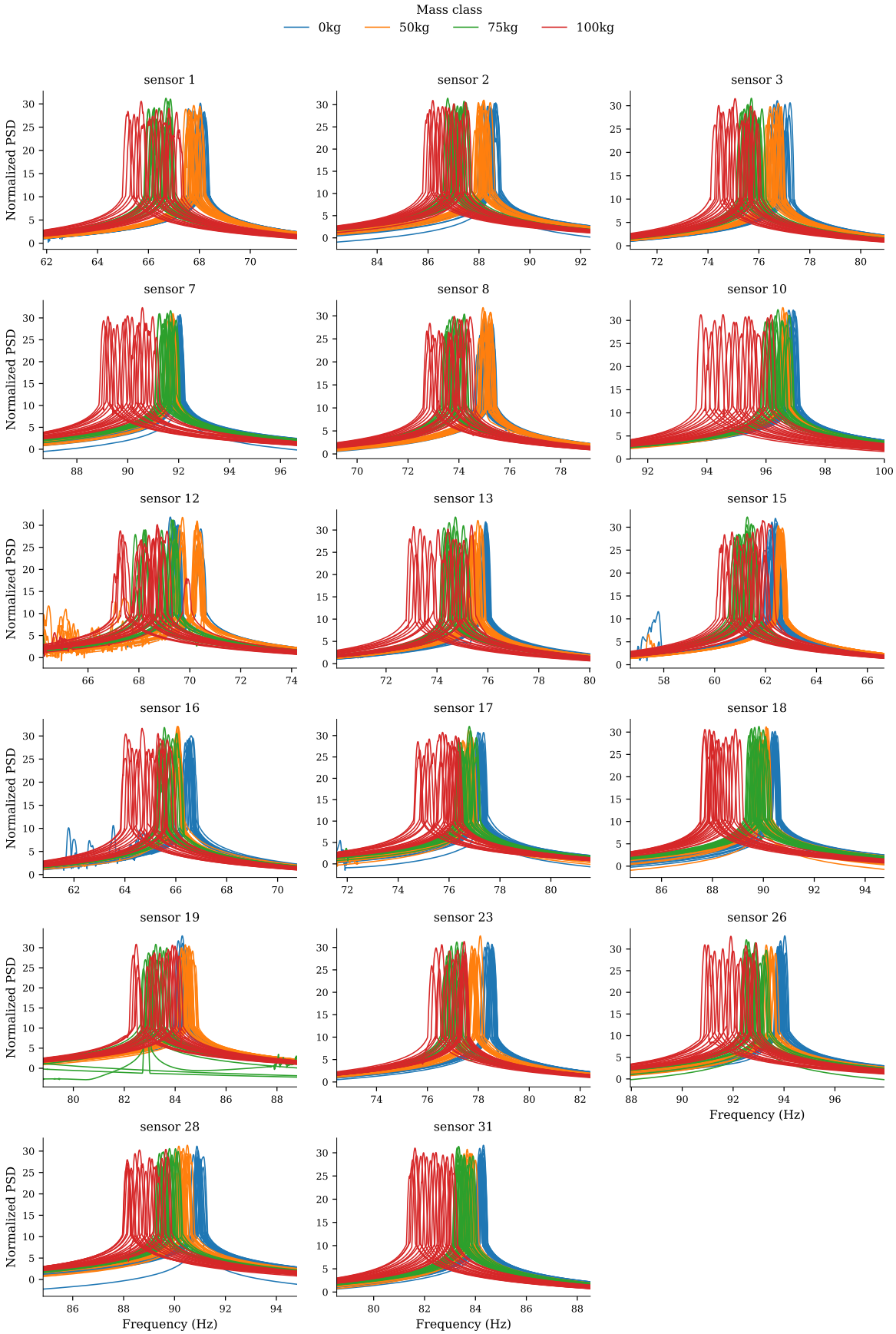


Figure 5.4: Normalized PSD for four different mass classes; not shown are the sensors for which the PSDs are of low quality. Sensors in the range from 1 to 10 are located in segment 1, from 11 to 22 in segment 2, and from 23 to 32 in segment 3. The added mass is attached to the bridge deck between sensor 16 and 17.

5.2 Feature Selection

In this section the feature selection procedures that are applied for the selection of damage-sensitive features in Chapter 6 and Chapter 7 are explained.

Spectrum

In Chapter 6 the SVM algorithm is applied for the classification problem. The features for the SVM algorithm are selected from the spectrum (particularly, the singular value spectrum or the power spectrum). Not all the spectral lines of a spectrum will be used as features, only a subset of the spectral lines will be used, which is defined in the following.

The spectra are scaled to equal order of magnitude before the feature selection, so that the features from different observations will be comparable. The normalization is performed for each spectrum individually as follows: first the spectrum is log-transformed, followed by the extraction of the mean value of the spectrum. In this way all the observations will have zero mean, while retaining the relative magnitude of the peaks. It is noted that in the following the features are not standardized because the features already have the same units (i.e. all the features represents a spectral quantity).

The spectra obtained from the processed signals after similarity filtering consist of 4000 spectral lines between 50 and 100 Hz (frequency resolution 0.0125 Hz). These spectral lines are the potential features for the machine learning algorithms, but as already mentioned it is not desirable to have so many of them. To reduce the number of features in a feature vector, the following feature selection procedure is applied:

Step 1. The spectra are down sampled to 1666 spectral lines in the frequency range of 50 to 100 Hz (compared to 4000 lines in the original spectrum). The resulting spectra have a frequency resolution of 0.03 Hz. At the end of this step the number of potential features is already reduced by almost 60%.

Step 2. The actual features are selected based on the variance of each spectral line. This variance is computed from the set of observations that includes all the mass classes (in general this will be a set with training vectors). Let $\{\mathbf{x}_j : j = 1, \ldots, N\}$ denote this set with N number of observations (i.e. the spectra resulting from the previous step), then the variance of the k-th spectral line is computed as

$$\mathrm{Var}\,(x_k) = \frac{1}{N-1}\sum_{j=1}^{N}\left(x_{k,j} - \bar{x}_k\right)^2, \qquad \bar{x}_k = \frac{1}{N}\sum_{j=1}^{N} x_{k,j} \tag{5.1}$$

where $\bar{x}_k$ is the mean value of the k-th spectral line. Subsequently, the spectral lines with a variance larger than some threshold are selected as features for damage detection. For example, we can select the spectral lines that have a variance larger than 5% of the maximum variance. The idea behind this step is that by choosing features with high variance the observations of different classes are better separable. There is a caveat associated with this selection. The features with low variance may in practice be most relevant for damage detection an may be erroneously disregarded. Besides, a large variance is not necessarily related to the class separability but can also be the result of large variation between the observations from a single class; for example when a certain state of the structure is measured under varying environmental conditions. For the spectrum, the largest variance is expected around the peak, which is the region of interest and thus it is acceptable to apply this selection procedure.

The feature selection process will be further clarified with an application to the singular value spectrum. Recall that the singular value spectra are computed for each bridge segment separately. So, for each segment the damage sensitive features must be selected using the process described above. In previous section, a new frequency range was defined for the singular value spectra of each segment; 65 to 100 Hz for segment 1, 55 to 72 Hz for segment 2, and 75 to 95 Hz for segment 3. The frequency ranges of the three segments are not the same, so are the number of spectral lines. These spectra are down sampled to a frequency resolution of 0.03 Hz in **Step 1**, so at the end of this step a singular value spectrum corresponding to segment 1, 2 and 3 consists of 1166, 566 and 666 spectral lines, respectively. It is noted that the shape of the spectra after this step does not differ from the spectra shown in Figures 5.1a–5.3a.

In **Step 2** the final features are selected based on a variance threshold. Figure 5.5b shows an example of the set with singular value spectra for segment 3 that results from the first step. The horizontal axis represents now the spectral lines of the spectra rather than the frequency, each line is referred to as a feature. The set

consists of four mass classes, but the class labels are not important here because these are not considered for the feature selection. For this particular set with singular value spectra, the variance of each spectral line is computed according to equation (5.1) and shown in Figure 5.5a. The dashed line indicates the 5% variance threshold, which is equal to 5% of the maximum variance. By comparing Figure 5.5a and Figure 5.5b it is clear that the regions with largest variance correspond to the regions with peaks in the spectra, which are the regions of interest for damage detection. Subsequently, the spectral lines with a variance larger than the threshold level are selected as features, which in this case are roughly the lines in the ranges 40–125, 200–325, and 420–640. In total 430 of the 666 spectral lines are selected as features. The selected features are highlighted in the figures.

Examples of the selected features from the singular value spectra are shown in Figures 5.6–5.8 for the three segments. The figures in the left column show the spectra after the first step, the highlighted regions indicate the spectral lines with a variance larger than 5% of the maximum variance. The spectral lines in these regions are selected as features and shown in the figures in the right column.

It is noted that the number of features selected depends on the data that is used to compute the variances of the spectral lines. Therefore, this feature selection step should be part of the classification algorithm, where the variances of the spectral lines are usually computed using a set of training vectors.

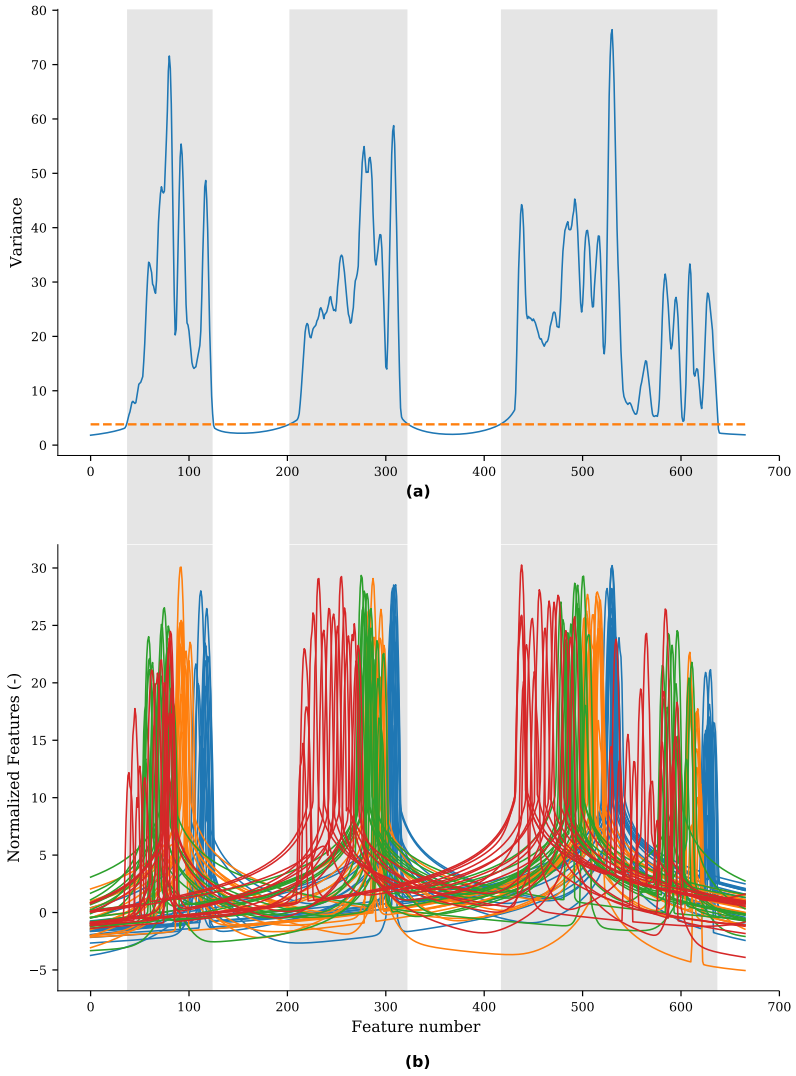


Figure 5.5: Example variance-based feature selection. In (a) the variance of each spectral line is shown together with the 5% threshold level (dashed line). The variances are computed for the spectra shown in (b). The spectral lines with a variance larger than the threshold are selected as features and highlighted in the figure.

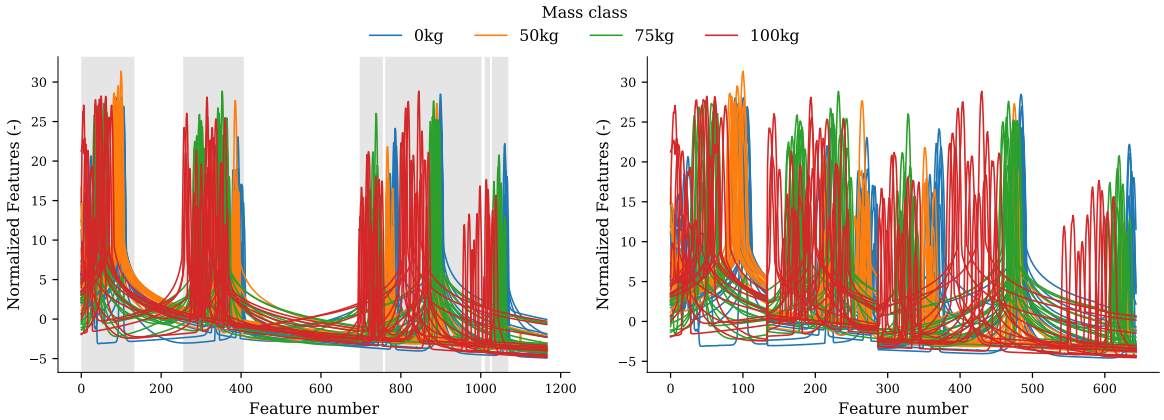


Figure 5.6: Segment 1 – Selected features after **Step 1** (left) and after **Step 2** (right). The selected spectral lines shown in the figure on the right are highlighted in the figure on the left. The spectra in the left figure consist of 1166 spectral lines (frequency resolution 0.03 Hz), and 644 spectral lines are selected with a variance of at least 5% of the maximum variance in the data set.

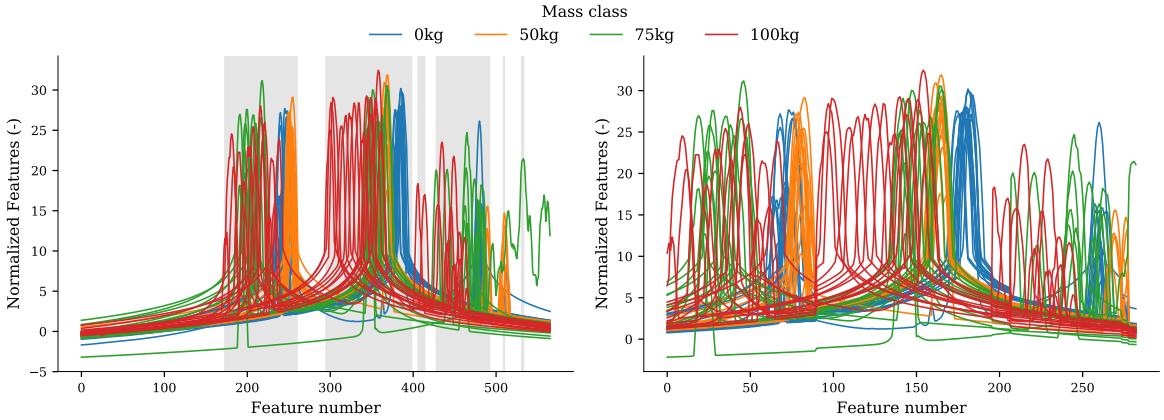


Figure 5.7: Segment 2 – Selected features after **Step 1** (left) and after **Step 2** (right).The selected spectral lines shown in the figure on the right are highlighted in the figure on the left. The spectra in the left figure consist of 566 spectral lines (frequency resolution 0.03 Hz), and 283 spectral lines are selected with a variance of at least 5% of the maximum variance in the data set.

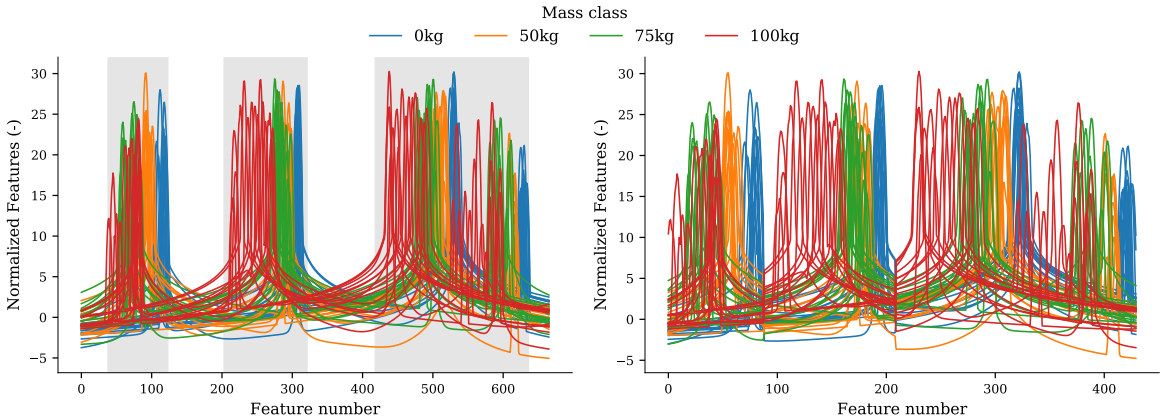


Figure 5.8: Segment 3 – Selected features after **Step 1** (left) and after **Step 2** (right). The selected spectral lines shown in the figure on the right are highlighted in the figure on the left. The spectra in the left figure consist of 666 spectral lines (frequency resolution 0.03 Hz), and 430 spectral lines are selected with a variance of at least 5% of the maximum variance in the data set.

The previous application of the feature selection procedure was tailored to the singular value spectra. The same procedure is applied for the selection of features from the PSD (to be used for the classification per sensor). In that case, there is only one peak in the PSD and the spectral lines centered around the peak frequency are selected as features. For the spectra shown in Figure 5.4, the number of spectral lines that are selected as features vary between 100 and 350.

Natural frequencies

In Chapter 7 the novelty detection method is applied for the detection of the added mass. For this method the natural frequencies will be used as damage-sensitive features, because of their low dimensionality. The natural frequencies were estimated in Section 4.3.1. The feature selection in this case is simple; the natural frequencies with a success rate larger than 80% are selected as features. This results in a 6-dimensional feature vector for segment 1, a 5-dimensional feature vector for segment 2, and a 4-dimensional feature vector for segment 3.

6

Supervised Learning - Classification

6.1 Introduction

The preceding chapters focused on the data acquisition, preprocessing of the data and feature engineering. At this point we are now faced with the challenge of making an accurate assessment of the damage condition of the structure based on the extracted features. For this we will use an approach that is based on the *pattern recognition* aspect of *machine learning*. The idea is that we can learn relationships from the measured data. In the context of SHM, this means that we can *learn* to assign a damage condition or *class* to a given feature vector of the structure. The feature vectors are formed from the measurement data and must be sensitive to damage. In this thesis, we have chosen to use the singular values of the spectral density matrix or the natural frequencies as features. The former are used as features in this chapter.

Most pattern recognition algorithms work by training a diagnostic. The type of learning algorithms in which the diagnostic is trained by showing it the true label for each data set is called *supervised learning*. This type of learning suggests that data from every conceivable damage condition should be available, which is not always possible. The two possible sources to acquire such data are physically-based modeling and experiments. The former can be very complex in case the structure and damage are difficult to model. In case of experiments it would be necessary to make copies of the structure of interest and damage it in all the ways that might naturally occur. For operational structures like bridges, this is simply not possible. However, we can still obtain data from experiments by simulating the damage rather than actually damaging the structure. The damage can be simulated with added masses, which is believed to have an equivalent effect on the selected features as the stiffness reduction caused by damage.

Fortunately there is an alternative to supervised learning, that is *unsupervised learning*. In this case, only training data from the normal operating condition of the structure is used to establish the diagnostic. In this way a model of the normal condition is trained. Then, during monitoring, newly acquired data are compared with this model. If there are any significant deviations, then the data is flagged as abnormal, which means that the structure has departed from its normal condition, so potential damage has occurred. The techniques used here are often referred to as *novelty detection* or *anomaly detection* methods. Novelty detection methods can only be used for detection and possibly locating damage (quantification of damage and prediction of remaining life time—level 3 and 4 diagnostics—are not possible). It is an important qualifier that the novelty detector should flag only if significant deviations from its normal operating condition occur. In reality, the structure is subjected to measurement noise and usually operate in changing environmental conditions, so the algorithm must be able to distinguish between a statistical fluctuation in the data and a real deviation from normality.

The damage identification problem in this sense is thus considered as a *pattern recognition problem*. Three types of algorithms can be distinguished depending on the desired diagnostic [19]:

1. *Novelty detection*; the algorithm simply indicate if the data come from a normal operating condition or not. This is a one-class problem, which has the advantage that unsupervised learning can be used. Novelty detection will be discussed in Chapter 7.

2. *Classification*; the algorithm returns a discrete label for given feature vector. In most basic form, the algorithm might assign a “damaged” or “not damaged” label to the features. In general, more classes

are used in order to quantify the damage states; for example, for the type of damage, different labels are used for different damage mechanisms that may occur and, for damage localization, the structure may be divided into labeled substructures. Classification is usually expressed as a pattern recognition problem and will be discussed in this chapter.

3. *Regression*; the algorithm returns values for one or more continuous variables. Regression is a supervised learning problem and could be useful, for e.g. severity assessment in which the output of the algorithm is the length of a fatigue crack. This algorithm is not discussed in this report, the interested reader is referred to [19] or [46].

Although the novelty detection is referred to as an unsupervised learning method, this is not completely correct. The term "unsupervised learning" is usually reserved for self-learning algorithms that search for previously unknown patterns in a dataset without pre-existing labels (e.g. cluster algorithms). However, for novelty detection it is assumed that the labels of one class are known, namely the normal condition, making it not completely unsupervised. Therefore, the novelty detection is sometimes referred to as *semi-supervised anomaly detection*.

In the next section, the pattern recognition approach is described in more detail. The discussion is limited to the classification problem. Although the presented classifier will not be applied in this work, it is very intuitive and allows for some important concepts to be introduced.

6.2 Bayesian Decision Theory

The problem of pattern recognition is to associate classes ω_i, $i = 1,2,\ldots,M$ with measured data usually expressed in terms of feature vectors $\mathbf{x}$. In the simplest case there will be just two classes, but in a sense this is the most important problem in SHM as one wish to distinguish between the classes "healthy" (H) and "damaged" (D). The considered problem is one of supervised learning, hence we will assumed that data from the damaged condition is available. In a probabilistic context, we are interested in the probability that the system is damaged or undamaged given that one has observed the feature vector $\mathbf{x}$, and this can be expressed in terms of the conditional probabilities $P(H\,|\,\mathbf{x})$ and $P(D\,|\,\mathbf{x})$, respectively. If these probabilities can be computed, it is natural to assign an unknown pattern to the class that has the highest probability of the two. Hence, the basic decision rule now is

$$\text{Choose } H \text{ if } P(H\,|\,\mathbf{x}) > P(D\,|\,\mathbf{x}), \text{ otherwise choose } D \tag{6.1}$$

Unfortunately, the two probabilities in question are usually not directly available. However, it is assumed that training data from the structure in healthy and damaged condition is available; that is, the data sets $\mathscr{D}_H = \{(\mathbf{x}_j, H)\colon\ j = 1,\ldots,N\}$ and $\mathscr{D}_D = \{(\mathbf{x}_j, D)\colon\ j = 1,\ldots,N\}$ for the healthy and damaged condition, respectively, are known. Both sets consists of N observations of the p-dimensional feature vector, $\mathbf{x}_j \in \mathbb{R}^p$, extracted from the measurements. Then it is possible to determine the class-conditional probability density functions $P(\mathbf{x}\,|\,H)$ and $P(\mathbf{x}\,|\,D)$, also referred to as the *likelihood functions*, which specify the probabilities that a measurement vector $\mathbf{x}$ arises from class H and D, respectively. Now the required probabilities $P(H\,|\,\mathbf{x})$ and $P(D\,|\,\mathbf{x})$ can be computed based on *Bayes theorem*, which, for a general class ω_i, is defined as

$$P(\omega_i\,|\,\mathbf{x}) = \frac{p(\mathbf{x}\,|\,\omega_i)P(\omega_i)}{p(\mathbf{x})} \tag{6.2}$$

where $P(\omega_i)$ is the *prior probability* of finding an example from class ω_i without considering any measurement information. $p(\mathbf{x})$ is the probability density function of $\mathbf{x}$, which can be computed based on the *law of total probability*

$$p(\mathbf{x}) = \sum_{i=1}^{M} p(\mathbf{x}\,|\,\omega_i)P(\omega_i) \tag{6.3}$$

Using equation (6.2), the decision rule (6.1) can equivalently be based on the inequality

$$p(\mathbf{x}\,|\,H)P(H) > p(\mathbf{x}\,|\,D)P(D) \tag{6.4}$$

in which $p(\mathbf{x})$ is not taken into account, because it is the same for all classes and it does not affect the decision. In the context of SHM, one will usually not know the prior class probabilities $P(H)$ and $P(D)$, although one would hope that $P(H) \gg P(D)$. The prior probabilities can be determined based on engineering knowledge,

or in the absence of any information the classes may be assumed equally likely; that is $P(H) = P(D) = 1/2$. If this latter applies, then the decision rule is based solely on the likelihoods calculated from the training set.

Figure 6.1 shows an example of two equiprobable classes and shows the likelihoods as a function of x for the simple case of a single feature ($p = 1$). The dotted line at x_0 is a threshold partitioning the feature space into two regions, R_1 and R_2. According to the decision rule, the classifier assigns all values of x in R_1 to H and all values in R_2 to D. However, the probability density functions overlap so classification errors are unavoidable. From the figure it is readily seen that the total probability of committing a classification error is given by

$$\begin{aligned} P_e &= P(x \in R_2 \cap H) + P(x \in R_1 \cap D) \\ &= P(x \in R_2 \mid H)P(H) + P(x \in R_1 \mid D)P(D) \\ &= P(H)\int_{x_0}^{\infty} P(x \mid H)\,\mathrm{d}x + P(D)\int_{-\infty}^{x_0} P(x \mid D)\,\mathrm{d}x \end{aligned} \tag{6.5}$$

which is equal to the total shaded area under the curves in Figure 6.1. It can be proven that, in general, the Bayesian classifier is optimal with respect to minimizing the classification error probability [46]. This means that if R_1 is the region of space in which $P(\omega_1 \mid \mathbf{x}) > P(\omega_2 \mid \mathbf{x})$, and R_2 the region where the reverse is true, then the probability of error is minimized.

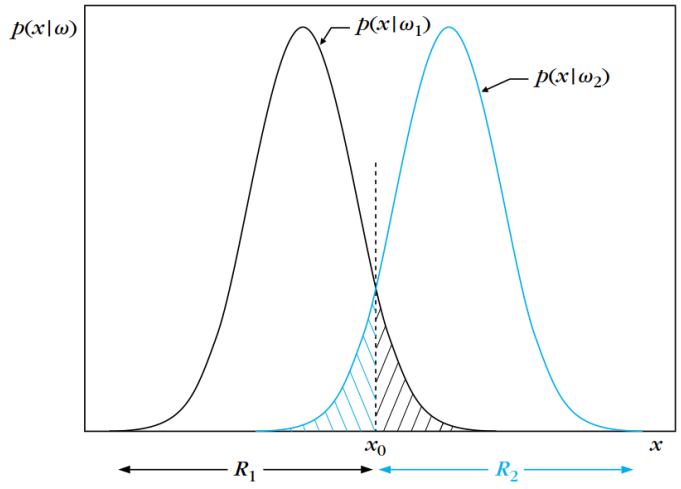


Figure 6.1: Example of two regions R_1 and R_2 formed by the Bayesian classifier for the case of two classes [46].

It is noted that the classification error probability is not always the best criterion to be adopted for minimization. This is because it assigns the same importance to all errors. However, there are cases in which some wrong decisions may have more serious consequences than others. In fact, in the two class problem considered above two types of error can occur:

1. Type I error, also called *false-positive* or *false alarm*. This type of error occurs when the structure is indicated as damaged while it is actually healthy. This could result in a structure being withdrawn from service or closed and has potentially severe financial consequences.

2. Type II error, also called *false-negative* or *missed detection*. This type of error occurs when the structure is indicated as healthy even though it is damaged. This means that the damaged structure is allowed to continue to operate in a misdiagnosed damaged state, which potentially may lead to complete loss of the structure and consequent loss of life. Hence the consequences of this type of error are more severe than a type I error as there will be a threat to safety.

So, it may be more appropriate to assign a penalty term to weight each error. In our case, the probability to commit a type I error is $P_I = P(x \in R_2 \mid H)P(H)$ and for type II this is $P_{II} = P(x \in R_1 \mid D)P(D)$. Instead of selecting R_1 and R_2 so that P_e is minimized, we can also minimize the following weighted version

$$P_e = \lambda_I P(H)\int_{x_0}^{\infty} P(x \mid H)\,\mathrm{d}x + \lambda_{II} P(D)\int_{-\infty}^{x_0} P(x \mid D)\,\mathrm{d}x \tag{6.6}$$

in which each of the two terms that contribute to the overall probability is weighted according to its significance. In our case, the reasonable choice would be to have $\lambda_{II} > \lambda_I$, because type II error has more severe consequences. Alternative cost functions can be established depending on the considered problem.

So far, we have dealt with the simple case of two classes. Generalization to the multiclass case is straightforward. In a classification problem with M classes, $\omega_1, \omega_2, \ldots, \omega_M$, an unknown pattern, represented by the feature vector $\mathbf{x}$, is assigned to class ω_i if

$$p(\mathbf{x}\,|\,\omega_i)P(\omega_i) > p(\mathbf{x}\,|\,\omega_j)P(\omega_j) \quad \forall\, j \neq i \tag{6.7}$$

It can be shown that such a choice also minimizes the classification probability [46].

The Bayesian classifier is a supervised learning algorithm, which means that measurement data from all conceivable classes are available. The general Bayesian classifier can be summarized as follows:

Step 1. Establish a training set $\mathscr{D}_i = \{(\mathbf{x}_j, y_j = \omega_i) :\ j = 1,2,\ldots,N_i\}$ consisting of N_i number of observations of the p-dimensional feature vector, $\mathbf{x}_j \in \mathbb{R}^p$, for each class ω_i, $i = 1,2,\ldots,M$.

Step 2. Estimate the likelihood functions $p(\mathbf{x}\,|\,\omega_i)$ and the a prior probabilities $P(\omega_i)$ for each class. This is the training stage of the classifier.

Step 3. Given a new unclassified measurement $\mathbf{y} \in \mathbb{R}^p$, compute the probability that the vector $\mathbf{y}$ can arise from each class ω_i; that is, evaluate $p(\mathbf{y}\,|\,\omega_i)$.

Step 4. Assign $\mathbf{y}$ to class ω_i so that $p(\mathbf{y}\,|\,\omega_i)P(\omega_i) > p(\mathbf{y}\,|\,\omega_j)P(\omega_j)$ for all $j \neq i$. This is the testing/monitoring stage of the classifier.

Some remarks on the Bayesian classifier;

- The new unclassified observations are denoted $\mathbf{y}$ to distinguish them from the training vector $\mathbf{x}$, but it must be remembered that both vectors represents the same type of features.
- The prior probabilities in **Step 2** are usually estimated from the data as $P(\omega_i) = N_i/N$, where N is the total number of samples from all classes. However, in the context of SHM, the data associated with the damage classes is usually obtained by inducing different types of damage (real or pseudo), meaning that the number of observations of a certain type of damage will not be equal to the real number of events, and the prior probabilities that are estimated from the data might differ significantly from the true probabilities. Alternatively, the prior probabilities can be determined based on engineering judgment or in the absence of any information regarding the occurrences of the different damage classes the classes may be assumed equally likely. The latter might be better than the estimation of the prior probabilities from the data, in that the resulting classifier does not favor the classes that are measured most. Obviously, we wish that the probabilities of the damaged classes are much smaller than the probabilities of the undamaged classes.
- The estimation of the probability density functions $p(\mathbf{x}\,|\,\omega_i)$ in **Step 2** is one of the most challenging tasks of all machine learning problems. There are two main possibilities here: a parametric form can be assumed for the probability density with the training data set used to determine the values of the parameters or a nonparametric form can be obtained. In the latter case, the probability density function can be constructed from a frequency histogram, Kernel Density Estimate (KDE) or Gaussian Mixture Model (GMM) over the training and stored as an array of values. In the former approach, in the vast majority of cases, the probability density is assumed to be Gaussian.
- Once new observations are classified in **Step 4**, it is possible to diagnose the condition of the structure. If some observations are classified to one of the damage classes, then an alarm should be issued.

Discriminant Functions and Decision Surfaces

It should be clear that, in case of M classes, the feature space is partitioned into M regions. If regions R_i and R_j happen to be contiguous, then they are separated by a *decision surface* in the multidimensional feature space. In case of minimum error probability, the decision surface is described by

$$g_{ij}(\mathbf{x}) = P(\omega_i\,|\,\mathbf{x}) - P(\omega_j\,|\,\mathbf{x}) = 0 \tag{6.8}$$

Instead of working directly with probabilities it may be more convinient, from a mathematical point of view, to work with an equivalent function of them, for example $g_i = f(P(\omega_i \mid \mathbf{x}))$, where $f(\bullet)$ is a monotonically increasing function. $g_i(\mathbf{x})$ is known as a *discriminant function*. The decision decision rule can now be stated as

$$\text{assign } \mathbf{x} \text{ to } \omega_i \text{ if } g_i(\mathbf{x}) > g_j(\mathbf{x}) \text{ for all } j \neq i \tag{6.9}$$

The decision surfaces, separating contiguous regions are then described by

$$g_{ij}(\mathbf{x}) = g_i(\mathbf{x}) - g_j(\mathbf{x}) = 0, \quad i, j = 1, 2, \ldots, M, \quad i \neq j \tag{6.10}$$

The concepts of discriminant functions and decision boundaries are also used in the theory about support vector machines (Section 6.3), where $g(\mathbf{x})$ will be a linear function of $\mathbf{x}$ and the corresponding decision surface is a hyperplane.

In the previous, we have approached the classification problem via Bayesian probabilistic arguments and the goal was to minimize the classification error probability or the risk. However, not all problems are well suited to such approaches. For example, in many cases the involved probability density functions are complicated and their estimation is not an easy task (especially in case of high dimensional features). In such cases, it may be preferable to compute decision surfaces directly by means of alternative costs; for example, maximizing the margin in support vector machine classifier or minimizing a certain distance metric in k-nearest neighbor classifier. Such approaches give rise to discriminant functions and decision surfaces, which are entities with no direct relation to Bayesian classification, and they are, in general, suboptimal with respect to Bayesian classifiers. The support vector machine classifier is discussed in more detail later, but first we will elaborate the Bayesian classification for the special case of Gaussian density function, which allows for the introduction of some important concepts that are used in the novelty detection method in Chapter 7.

The Bayesian Classifier for Gaussian Distributed Classes

As noted earlier, the estimation of the involved probability density functions, $p(\mathbf{x} \mid \omega_i)$ (likelihood functions of ω_i with respect to $\mathbf{x}$), is far from trivial. Here we will assume a parametric form for the probability density. More specifically, we will assume that the likelihood functions describing the data distribution in each one of the classes are multivariate Gaussian distributions, that is

$$p(\mathbf{x} \mid \omega_i) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}_i|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right) \tag{6.11}$$

where $\boldsymbol{\mu}_i \in \mathbb{R}^p$ and $\boldsymbol{\Sigma}_i \in \mathbb{R}^{p\times p}$ are respectively the mean vector and the covariance matrix associated with class ω_i, and $|\boldsymbol{\Sigma}_i|$ denotes the determinant of $\boldsymbol{\Sigma}_i$. In this case, the problem of parameter estimation is reduced to obtaining $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$. One popular method for achieving this is the *maximum likelihood* (ML) approach. It can be shown that the maximum likelihood estimate of the mean vector is the *sample mean vector*, denoted as $\bar{\mathbf{x}}$, taken over all training vectors

$$\bar{\mathbf{x}} = \sum_{j=1}^{N} \mathbf{x}_j \tag{6.12}$$

and that the maximum likelihood estimate of the covariance matrix is equal to the *sample covariance matrix*, denoted as $\mathbf{C}$

$$\mathbf{C} = \frac{1}{N}\sum_{j=1}^{N}\left(\mathbf{x}_j - \boldsymbol{\mu}\right)\left(\mathbf{x}_j - \boldsymbol{\mu}\right)^{\mathrm{T}} \tag{6.13}$$

in which $\boldsymbol{\mu} = \mathbf{E}[\mathbf{x}]$ is the population mean. If this mean is unknown it can be replaced with the sample mean (6.12), then the factor in front should be $1/(N-1)$ rather than $1/N$ in order to yield unbiased estimates of the covariance matrix.

Returning to the problem, the Bayesian classifier depends on maximizing the discriminator function

$$g_i(\mathbf{x}) = p(\mathbf{x} \mid \omega_i)P(\omega_i) \tag{6.14}$$

over all classes. In case of Gaussian distributed classes, it is more convenient to work with the following discriminant function, which is obtained by taking the natural logarithm of equation (6.14)

$$g_i(\mathbf{x}) = \ln\left(p(\mathbf{x} \mid \omega_i)P(\omega_i)\right) = \ln p(\mathbf{x} \mid \omega_i) + \ln P(\omega_i) \tag{6.15}$$

and using equation (6.11) we obtain

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) + \ln P(\omega_i) + c_i \tag{6.16}$$

in which the constant $c_i = -(p/2)\ln 2\pi - (1/2)\ln|\boldsymbol{\Sigma}_i|$. In general, this discriminator function has a quadratic form.

We will now show that maximizing $g_i(\mathbf{x})$ is equivalent to minimizing a certain distance. Assuming equiprobable classes and ignoring the constant, equation (6.16) simplifies to

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) \tag{6.17}$$

Maximizing this discriminant function is equivalent to minimizing

$$\|\mathbf{x}-\boldsymbol{\mu}_i\|^2_{\boldsymbol{\Sigma}^{-1}} \equiv (\mathbf{x}-\boldsymbol{\mu}_i)^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) \tag{6.18}$$

which has the form of a distance squared, based on a norm weighted by the inverse of the covariance matrix, $\boldsymbol{\Sigma}_i^{-1}$, also known as the *precision matrix*. The distance $\|\mathbf{x}-\boldsymbol{\mu}\|_{\boldsymbol{\Sigma}^{-1}}$ is known as the *Mahalanobis distance*. Hence, from a geometrical point of view, under restricted circumstances, classification is performed by minimizing the Mahalanobis distance between a feature vector $\mathbf{x}$ and the class mean vector $\boldsymbol{\mu}_i$.

In the particular case that the covariance matrix is diagonal with equal elements; that is, the features are uncorrelated and of the same variance $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}_p$, the norm in equation (6.18) becomes

$$\frac{\|\mathbf{x}-\boldsymbol{\mu}_i\|^2}{\sigma^2} \equiv \frac{1}{\sigma^2}(\mathbf{x}-\boldsymbol{\mu}_i)^{\mathrm{T}}(\mathbf{x}-\boldsymbol{\mu}_i) \tag{6.19}$$

in which $\|\bullet\|$ is the Euclidean norm. So, in this case the feature vectors are assigned to classes based on their *Euclidean distance* from the respective mean points.

In Figure 6.2 both distance are illustrated in a two-dimensional feature space ($p = 2$). Figure 6.2a shows the case of diagonal covariance matrix in which the curves of equal Euclidean distance from the mean points are circles (in a high-dimensional feature space these will be hyperspheres). Figure 6.2b shows the general case of nondiagonal covariance matrix, in which the curves of constant Mahalanobis distance are ellipses (or hyperellipses in a high-dimensional feature space). So, all points having the same Mahalanobis distance from a specific point are located on an ellipse (rather than a circle), and it can be shown that the distance from the center of mass of an ellipse to a point with Mahalanobis distance c, in the direction of a principal axis, is $2\sqrt{\lambda_j}c$, where λ_j, $j = 1,\ldots,p$ are the eigenvalues of the covariance matrix and the principal axis are aligned with the corresponding eigenvectors.

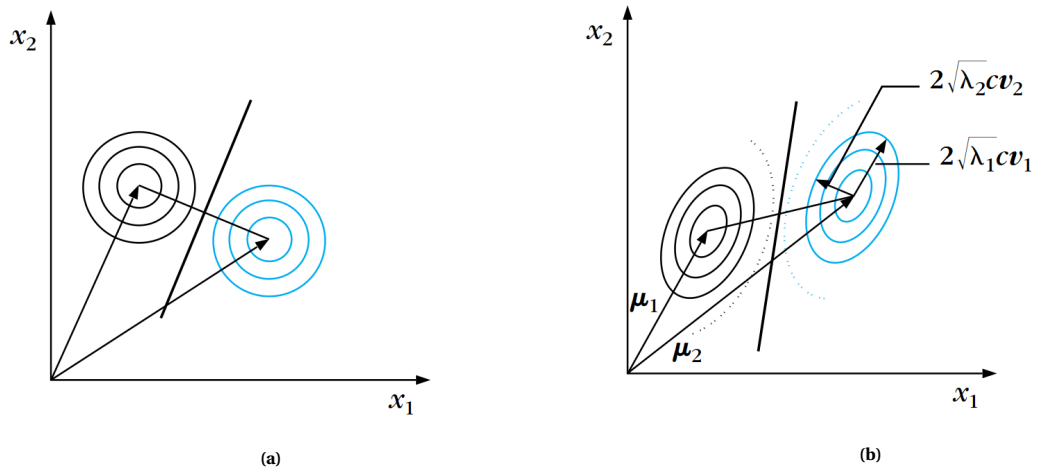


Figure 6.2: An example of curves of (a) equal Euclidean distance and (b) equal Mahalanobis distance from the mean points of each class [46]. In the two-dimensional space, they are circles in case of Euclidean distance and ellipses in the case of Mahalanobis distance.

Previous consideration serves as an introduction of the Mahalanobis distance that will be used for novelty detection in Chapter 7. The geometrical interpretation, illustrated in Figure 6.2, might be useful for understanding the basics of this method.

Remarks

- The most challenging task while designing a Bayesian classifier is the estimation of the probability density functions $p(\mathbf{x} \mid \omega_i)$, especially in the case of a high-dimensional feature space. Usually a large number of samples are necessary for acceptable performance of the density estimation techniques. This number grows exponentially with the dimension of the feature space, which is usually referred to as the *curse of dimensionality*.
- In practice it is common to assume that the data in each class are adequately described by a Gaussian distribution. This means that the data of each class can be characterized by its first two statistical moments (i.e. the mean vector and covariance matrix). The problem now reduces to one of estimating the mean vector and covariance matrix from the data of each class. Maximum likelihood is often used for the estimation of the unknown parameters that define the mean vectors and the covariance matrices. The number of unknown parameters to be estimated depends on the adopted assumption concerning the covariance matrices. That is, if they are all equal or different for the classes.

6.3 Support Vector Machines

We will start with a linearly separable two-class problem, and then extend the method to more general cases where the data are not separable and/or multiple classes are involved.

Let $\mathscr{D}_N = \{(\mathbf{x}_j, y_j) : j = 1, \ldots, N\}$ denote a data set consisting of N number of p-dimensional feature vectors, $\mathbf{x}_j \in \mathbb{R}^p$, extracted from measurements and the corresponding labels, $y_j \in \{-1, 1\}$. The labels belong to either of the two classes ω_1 or ω_2, which are assumed to be linearly separable. The task now is to find a linear separating boundary between these two classes, which classifies correctly all the training vectors. The boundary will be a hyperplane (or a straight line in case of two-dimensional feature vectors), which can be expressed as

$$g(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b = 0 \tag{6.20}$$

in which $\mathbf{w} \in \mathbb{R}^p$ determines the direction of the hyperplane and b determines the exact position in space. Figure 6.3 illustrates the classification task with three possible hyperplane solutions. Obvious, all three hyperplanes perfectly separate the two classes. However, some hyperplanes are closer to the training data than other, which means that there is less room for the data to move without causing an error.

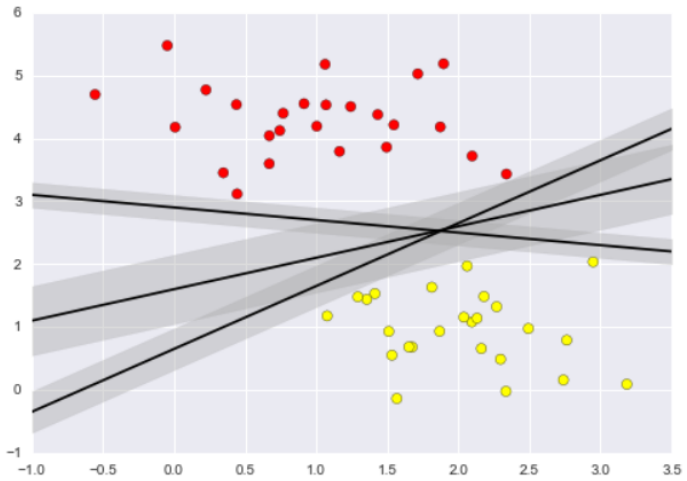


Figure 6.3: An example of a linearly separable two-class problem with three possible separating lines [47]. The width around each line up to the nearest point in each class (margin) is marked.

In general there will be (infinite) many separating hyperplanes. The problem with most of them is that they will not generalize well; that is, the classifier, designed using the training data set, will not operate well with data outside this set. The idea of a Support Vector Machine (SVM) is that it will select the hyperplane that generalizes the best, and in this case this means that the one that is furthest from the data in some sense. The total distance that a hyperplane leaves from both classes is referred to as the *margin*. Our goal is to find

the hyperplane that gives the maximum possible margin. However, the parametrisation of the hyperplane (6.20) is currently arbitrary. This will be fixed by appropriate scaling of all the candidate hyperplanes.

The training vectors $\mathbf{x}_j$ can be seen as points in a p-dimensional space, and the distance of each point from the separating hyperplane is

$$z_j = \frac{|g(\mathbf{x}_j)|}{\|\mathbf{w}\|} \tag{6.21}$$

We can now scale $\mathbf{w}$, b so that the value of $g(\mathbf{x})$, at the nearest points in ω_1, ω_2, is equal to 1 for ω_1 and equal to -1 for ω_2 (we want to give no preference to any of the classes, so the hyperplane is placed at the same distance from the respective nearest points in ω_1 and ω_2). The margin will then be $2/\|\mathbf{w}\|$, and the separation conditions are given by

$$g(\mathbf{x}_j) = \mathbf{w}^{\mathrm{T}}\mathbf{x}_j + b \geq 1, \qquad \forall \mathbf{x}_j \in \omega_1 \tag{6.22}$$

$$g(\mathbf{x}_j) = \mathbf{w}^{\mathrm{T}}\mathbf{x}_j + b \leq -1, \qquad \forall \mathbf{x}_j \in \omega_2 \tag{6.23}$$

or more concisely

$$g(\mathbf{x}_j) = y_j\left(\mathbf{w}^{\mathrm{T}}\mathbf{x} + b\right) \geq 1 \tag{6.24}$$

where ω_1 and ω_2 are the two classes, and y_j in this case is a class label $y_j = 1$ if $\mathbf{x}_j \in \omega_1$, and $y_j = -1$ if $\mathbf{x}_j \in \omega_2$.

It should be clear now that maximizing the margin will place the hyperplane at the furthest point from the data and this can be accomplished by minimizing the norm of the weight vector, $\|\mathbf{w}\|$, subject to the constraint of equation (6.24). This can be formulated as the following *primal optimization problem*

$$\begin{aligned} &\underset{\mathbf{w},b}{\text{minimize}} \quad J(\mathbf{w},b) = \frac{1}{2}\|\mathbf{w}\|^2 \\ &\text{subject to} \quad y_j\left(\mathbf{w}^{\mathrm{T}}\mathbf{x} + b\right) \geq 1, \quad j = 1,2,\ldots,N \end{aligned} \tag{6.25}$$

The optimization variables $\mathbf{w}$, b are referred to as *primal variables* and the optimal values of the problem are denoted as $\mathbf{w}^*$, b^*, respectively. The optimization problem is a *quadratic programme* one because the cost function is quadratic and the constraints are all linear.

To solve this constrained optimization problem we use the method of Lagrange multipliers. This method involves a Lagrangian function, which for optimization problem in (6.25) is defined as

$$\mathscr{L}(\mathbf{w},b,\boldsymbol{\lambda}) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} - \sum_{j=1}^{N}\lambda_j\left(y_j\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_j + b\right) - 1\right) \tag{6.26}$$

where $\lambda_j \geq 0$, $j = 1,2,\ldots,N$ are the Lagrangian multipliers. Furthermore, we make use of the special nature of the optimization problem. That is, the cost function is convex and the constraints are linear and define a convex set of feasible solutions. Such problems can be solved by considering the *Lagrangian duality*. Lagrangian treatment of convex optimization problems lead to an alternative *dual* description, which often turns out to be easier to solve than the primal problem since handling inequality constraints directly is difficult. Moreover, the dual formulation has the attractive property that the data only appear inside an inner product, which allows for efficient generalizations in the nonlinear case.

We can transform the primal optimization problem into the dual by simply setting to zero the derivatives of the Lagrangian function with respect to the primal variables, and substituting the so obtained relations back into the Lagrangian. This transformation removes the dependence on the primal variables. The resulting function contains only dual variables and must be maximized under simpler constraints.

The corresponding dual representation of (6.25) is found by differentiating the Lagrangian with respect to $\mathbf{w}$, b, imposing stationarity

$$\frac{\partial}{\partial \mathbf{w}}\mathscr{L}(\mathbf{w},b,\boldsymbol{\lambda}) = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{j=1}^{N}\lambda_j y_j \mathbf{x}_j \tag{6.27}$$

$$\frac{\partial}{\partial b}\mathscr{L}(\mathbf{w},b,\boldsymbol{\lambda}) = 0 \quad \Rightarrow \quad \sum_{j=1}^{N}\lambda_j y_j = 0 \tag{6.28}$$

and substituting the so obtained relations back into the Lagrangian (6.26); We obtain an equivalent optimization task

$$\begin{aligned} \underset{\lambda}{\text{maximize}} \quad & \sum_{j}^{N} \lambda_j - \frac{1}{2} \sum_{j}^{N} \sum_{k}^{N} \lambda_j \lambda_k y_j y_k \mathbf{x}_j^{\mathrm{T}} \mathbf{x}_k \\ \text{subject to} \quad & \sum_{j=1}^{N} \lambda_j y_j = 0 \\ & \lambda_j \geq 0, \quad j = 1, 2, \ldots, N \end{aligned} \tag{6.29}$$

Once the optimal Lagrange multipliers have been obtained by maximizing equation (6.29), the optimal weight vector $\mathbf{w}^*$ that realizes the maximum margin hyperplane is obtained via equation (6.27) as

$$\mathbf{w}^* = \sum_{j=1}^{N} \lambda_j^* y_j \mathbf{x}_j \tag{6.30}$$

The value of b does not appear in the dual problem and so b^* must be found making use of some other conditions. To this end we introduce the *Karush-Kuhn-Tucker (KKT) complementary conditions*. These conditions state that the optimal solution λ_j^*, $\mathbf{w}^*$, b^* must satisfy

$$\lambda_j^* \left(y_j \left(\mathbf{w}^{*\mathrm{T}} \mathbf{x}_j + b^* \right) - 1 \right) = 0, \quad j = 1, 2, \ldots, N \tag{6.31}$$

Either one of the terms in these conditions must be zero. If λ_j^* is nonzero, then the second term in equation (6.31) must be zero, which implies that the corresponding feature vector x_j lies on the boundary of the margin, these vectors are termed *support vectors*. The set of indices of the support vectors will be denoted with Ω_{sv}. Now b^* can be obtained from any of the complementary conditions (6.31) satisfying $\lambda_j^* \neq 0$, that is

$$b^* = y_j - \mathbf{w}^{*\mathrm{T}} \mathbf{x}_j = y_j - \sum_{k \in sv} \lambda_k^* y_k \mathbf{x}_k^{\mathrm{T}} \mathbf{x}_j \quad \forall j \in sv \tag{6.32}$$

Notice that this parameter is solely determined by the position of the support vectors. In practice, b^* is computed as an average value obtained using all conditions of this type.

The optimal hyperplane can be expressed in terms of the dual variables

$$g(\mathbf{x}) = \mathbf{w}^{*\mathrm{T}} \mathbf{x} + b^* = \sum_{j \in \Omega_{\mathrm{sv}}} \lambda_j^* y_j \mathbf{x}_j^{\mathrm{T}} \mathbf{x} + b^* \tag{6.33}$$

The Lagrange multipliers (or dual variables) associated with each point quantifies how important a given point is in forming the final solution. Points that are not support vectors ($\lambda_j = 0$) have no influence, so slight perturbations of such points will not affect the solution.

Remarks

- The Lagrange multipliers can be either zero or positive. Thus, the vector parameter $\mathbf{w}^*$ of the optimal solution is a linear combination of $N_s \leq N$ feature vectors that are associated with $\lambda_j^* \neq 0$; that is

$$\mathbf{w}^* = \sum_{j \in \Omega_{\mathrm{sv}}} \lambda_j^* y_j \mathbf{x}_j \tag{6.34}$$

  Where Ω_{sv} denotes a subset of indices. The considered feature vectors $\mathbf{x}_j$, $j \in \Omega_{\mathrm{sv}}$ are known as *support vectors* and the optimum hyperplane classifier as a *support vector machine*. As already mentioned, the set of constraints in (6.31) suggests for $\lambda_j^* \neq 0$ that the support vectors lie on the boundary of the margin, that is

$$\mathbf{w}^{*\mathrm{T}} \mathbf{x} + b = \pm 1 \tag{6.35}$$

  In other words, they are the training vectors that are closest to the linear classifier, and they constitute the critical elements of the training set. The resulting separating hyperplane only depends on the number and position of the support vectors.

- The cost function in equation (6.25) is strict convex. Furthermore, the inequality constraints consist of linear functions. These two conditions guarantee that any local minimum is also global and unique. Hence, the optimal hyperplane classifier of a support vector machine is unique.

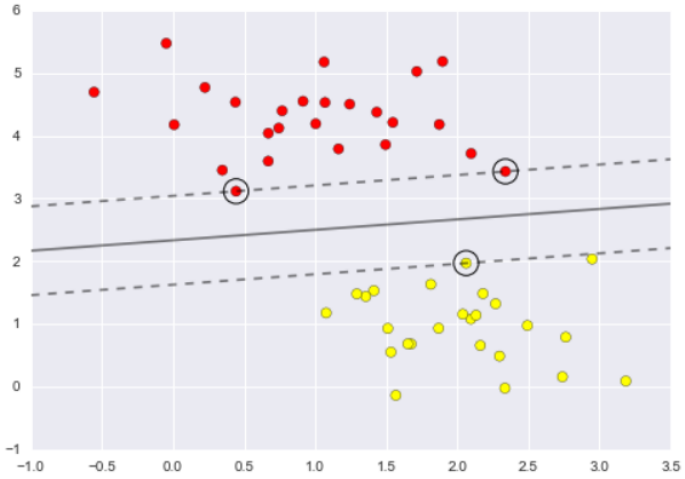


Figure 6.4: An example of a linearly separable two-class problem with the dividing line (solid) that maximizes the margin between the two sets of points [47]. The support vectors are enclosed by circles.

- The training vectors enter the dual problem (6.29) via equality constraints and not inequality ones, as was the case for (6.25), which are easier to handle. Furthermore, the vectors enter the dual problem in pairs, in the form of inner products. Hence, the cost function does not depend explicitly on the dimensionality of the input space, which allows for efficient generalizations in the nonlinear case.

6.3.1 Nonseparable Classes

When the classes are not separable, previous classifier cannot be used since it is designed to always produce a perfect separation, that is all training points are correctly classified. Figure 6.5 illustrates the case in which two classes are not separable. Any attempt to draw a hyperplane will never result in a perfect separation with no data points inside the margin, as was previously the case.

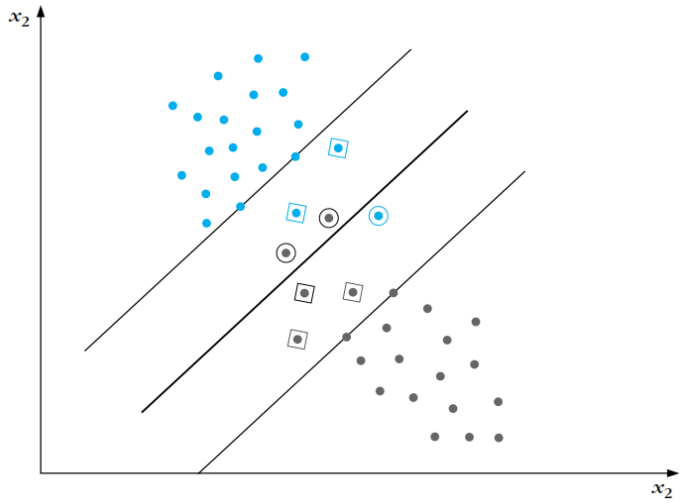


Figure 6.5: An example of linearly nonseparable classes [46].

In order to allow the margin constraints to be violated, a new set of variables is introduced, namely

$$y_j\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_j + b\right) \geq 1 - \xi_j \tag{6.36}$$

The variables ξ_j are known as *slack variables*. The training vectors can now belong to one of the following categories, depending on the value these variables:

1. $\xi_j = 0$, vectors that fall outside the margin and are correctly classified. These vectors comply with the constraints in (6.25).

2. $0 < \xi_j \leq 1$, vectors falling inside the margin and are correctly classified. They are placed in squares in Figure 6.5.

3. $\xi_j > 1$, vectors that are misclassified. They are enclosed by circles in Figure 6.5.

The goal now is to make the margin as large as possible but at the same time to keep the number of points with $\xi_j > 0$ (i.e. vectors inside the margin and misclassified vectors) as small as possible. The mathematical formulation of this problem involves a discontinuous function, which complicates the optimization tasks. Therefore, a closely related cost function is optimized that contains a penalty term of the form $\sum_{j=1}^{N} \xi_j^p$. If $p = 1$ or 2, the optimization problem remains one of quadratic programming. Furthermore, if $p = 1$ then the slack variables do not appear in the dual problem (which simplifies the computations). With $p = 1$, the primal optimization problem becomes

$$\begin{aligned} \underset{\mathbf{w},b}{\text{minimize}} \quad & J(\mathbf{w}, b, \boldsymbol{\xi}) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{j=1}^{N} \xi_j \\ \text{subject to} \quad & y_j\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_j + b\right) \geq 1 - \xi_j, \quad \xi_j \geq 0, \quad j = 1, 2, \ldots, N \end{aligned} \tag{6.37}$$

in which the parameter C is a positive constant that controls the cost of nonseparation. The problem is again a convex quadratic programming one, and the corresponding Lagrangian is given by

$$\begin{aligned} \mathscr{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = & \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{j=1}^{N} \xi_j - \sum_{j=1}^{N} \mu_j \xi_j \\ & - \sum_{j=1}^{N} \lambda_j \left(y_j\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_j + b\right) - 1 + \xi_j\right) \end{aligned} \tag{6.38}$$

with $\lambda_j \geq 0$ and $\mu_j \geq 0$; the latter are the Lagrangian multipliers that force the ξ_j to be positive. The corresponding dual formulation is found by differentiating the Lagrangian with respect to the primal variable, $\mathbf{w}$, $\boldsymbol{\xi}$ and b, imposing stationarity

$$\frac{\partial}{\partial \mathbf{w}} \mathscr{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{j=1}^{N} \lambda_j y_j \mathbf{x}_j \tag{6.39}$$

$$\frac{\partial}{\partial b} \mathscr{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = 0 \quad \Rightarrow \quad \sum_{j=1}^{N} \lambda_j y_j = 0 \tag{6.40}$$

$$\frac{\partial}{\partial \xi_j} \mathscr{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = 0 \quad \Rightarrow \quad C - \lambda_j - \mu_j = 0 \tag{6.41}$$

and substituting the so obtained relations into the Lagrangian (6.38); we obtain the following dual representation of the optimization problem

$$\begin{aligned} \underset{\lambda}{\text{maximize}} \quad & \sum_{j=1}^{N} \lambda_j - \frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{N} \lambda_j \lambda_k y_j y_k \mathbf{x}_j^{\mathrm{T}}\mathbf{x}_k \\ \text{subject to} \quad & \sum_{j=1}^{N} \lambda_j y_j = 0 \\ & 0 \leq \lambda_j \leq C, \quad j = 1, 2, \ldots, N \end{aligned} \tag{6.42}$$

Note that this problem is identical to that for the separable case. The only difference is that the constraint $C - \lambda_j - \mu_j = 0$, together with $\mu_j \geq 0$, enforces $\lambda_j \leq C$. In particular, the linearly separable case corresponds to $C \to \infty$. Further note that the slack variables ξ_j, and their associated Lagrangian multipliers μ_j, do not enter into the problem explicitly. Their presence is reflected through the parameter C.

Solving (6.53) for the optimal parameters $\boldsymbol{\lambda}^*$, the corresponding expression for the hyperplane is exactly as before, given by equation (6.33). Such a classifier is said to have a *soft margin*.

Remarks

- For larger values of C, the second term in (6.36) has more influence in the cost. The optimization process tries to satisfy this demand by reducing the margin and consequently the number of points with $\xi_j > 0$. In other words, the width of the margin does not depend entirely on the data distribution, as was the case with the completely separable classes, but is heavily affected by the choice of C. This is the reason SVM classifiers of this type are also known as soft margin classifiers.
- One problem with the soft margin approach is the choice of the parameter C. Typically a range of values must be tried before the best choice for a particular training set can be selected. Furthermore, the scale of the parameter is affected by the size of the feature space.

6.3.2 The Multiclass Case

The discussion so far was restricted to a two-class classification problem. In an M-class problem, a straightforward extension is to consider it as a set of M two class problems (*one-versus-rest*). For each one of the classes, we seek to design an optimal discriminant function $g_m(\mathbf{x})$, $m = 1, 2, \ldots, M$ so that $g_m(\mathbf{x}) > g_n(\mathbf{x})$ for all $m \neq n$, if $\mathbf{x} \in \omega_m$. Then classification is achieved according to the following rule

$$\text{assign } \mathbf{x} \text{ to } \omega_m \text{ if } m = \arg\max_n \{g_n(\mathbf{x})\} \tag{6.43}$$

This technique, however, may lead to indeterminate regions, where more than one discriminant function is positive. Furthermore, each binary classification is rather asymmetric in the sense that training is carried out with many more negative than positive examples. This becomes more serious when the number of classes is relative large.

An alternative technique is the *one-versus-one*. In this case $M(M-1)/2$ binary classifiers are trained and each classifier separates a pair of classes. Classification is then achieved based on majority vote. The obvious disadvantage of this technique is that a relatively large number of binary classifiers has to be trained. Nevertheless, the one-versus-one technique is used in this thesis because it has less drawbacks. Although this method is computationally more expensive than the one-versus-rest technique, this will not be excessive because the considered data sets are relative small.

6.3.3 The Nonlinear Case

The analysis so far has assumed that the classifier is linear, but this can easily be extended to nonlinear problems. These problems are basically solved by first transforming the data into a high-dimensional space in a nonlinear manner and then linearly separating the classes in this space. The underlying justification is found in *Cover's theorem* [12] on the separability of patterns, which, in quality terms may be stated as follows

> *"A complex pattern-classification problem, cast in a high-dimensional space nonlinearly, is more likely to be linearly separable than in a low-dimensional space, provided that the space is not densely populated."*

The input data are therefore mapped into a higher-dimensional *feature space* using a nonlinear mapping $\phi\colon \mathbb{R}^p \to \mathbb{R}^Q$, that is

$$\mathbf{x} \mapsto \phi(\mathbf{x}) \in \mathbb{R}^Q \tag{6.44}$$

Recall that once the optimal hyperplane has been computed, classification is performed according to the following rule

$$\text{assign } \mathbf{x} \text{ to } \begin{cases} \omega_1 \text{ if } g(\mathbf{x}) \geq 0 \\ \omega_2 \text{ if } g(\mathbf{x}) < 0 \end{cases} \tag{6.45}$$

where the decision function $g(\mathbf{x})$ is given in equation (6.33), and repeated here

$$g(\mathbf{x}) = \mathbf{w}^{*\mathrm{T}}\mathbf{x} + b^* = \sum_{j \in \Omega_{\mathrm{sv}}} \lambda_j^* y_j \mathbf{x}_j^{\mathrm{T}}\mathbf{x} + b^* \tag{6.46}$$

where Ω_{sv} denotes the subset of indices corresponding to the support vectors; that is, the vectors with nonzero weights λ_j^*. The feature vectors participate in pairs, via the inner product operation. Thus, if the classification takes place in a new Q-dimensional space, then the only difference is that the involved vectors will be the Q-dimensional mappings of the original input vectors. At first sight it seems that the complexity for making the

classes linear separable is much higher now, since Q is usually much higher than the dimension of the input space, in order to make the classes linear separable. However, according to *Mercer's theorem* [46] it is possible to express the inner product of the vectors in the high-dimensional feature space as a function of the inner product of the corresponding vectors evaluated in the original input space; that is

$$K(\mathbf{x},\mathbf{z}) = \phi(\mathbf{x})^{\mathrm{T}}\phi(\mathbf{z}) \tag{6.47}$$

where $K(\mathbf{x},\mathbf{z})$ denotes a kernel function evaluated in the input space.

We will illustrate the concept of a kernel function with a simple example. Assume that $\phi\colon \mathbb{R}^2 \to \mathbb{R}^3$

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \mapsto \phi(\mathbf{x}) = \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \end{pmatrix} \tag{6.48}$$

Then the inner product between $\phi(\mathbf{x})$ and $\phi(\mathbf{z})$ is

$$\begin{aligned} \phi(\mathbf{x})^{\mathrm{T}}\phi(\mathbf{z}) &= \begin{pmatrix} x_1^2 & x_2^2 & \sqrt{2}x_1x_2 \end{pmatrix} \begin{pmatrix} z_1^2 \\ z_2^2 \\ \sqrt{2}z_1z_2 \end{pmatrix} \\ &= x_1^2z_1^2 + x_2^2z_2^2 + 2x_1x_2z_1z_2 \\ &= (x_1z_1 + x_2z_2)^2 \\ &= \left(\mathbf{x}^{\mathrm{T}}\mathbf{z}\right)^2 \end{aligned} \tag{6.49}$$

In words, the inner product of vectors in the new (higher-dimensional) space has been expressed as function of the inner product of the corresponding vectors in the original space. So, we can compute the inner product without explicitly computing $\phi(\mathbf{x})$ and $\phi(\mathbf{z})$. All that is required, in this specific example, is the kernel $K(\mathbf{x},\mathbf{z}) = \left(\mathbf{x}^{\mathrm{T}}\mathbf{z}\right)^2$.

Some common kernels that are often used for SVM are

Polynomials	$K(\mathbf{x},\mathbf{z}) = \left(\mathbf{x}^{\mathrm{T}}\mathbf{z}+1\right)^q, \quad q>0$	(6.50)
Radial Basis Functions (RBF)	$K(\mathbf{x},\mathbf{z}) = \exp\left(-\gamma\|\mathbf{x}-\mathbf{z}\|^2\right)$	(6.51)
Hyperbolic Tangent	$K(\mathbf{x},\mathbf{z}) = \tanh\left(\alpha\mathbf{x}^{\mathrm{T}}\mathbf{z}+\beta\right)$	(6.52)

for appropriate values of α and β so that Mercer's conditions are satisfied. Working values are $\alpha = 2$, $\beta = 1$. The actual mapping ϕ is in general not known, just like the dimension of the space, which can even be infinite.

Once an appropriate kernel has been adopted that implicitly defines a mapping into a higher dimensional space, the dual optimization task, equations (6.42), becomes

$$\begin{aligned} \underset{\lambda}{\text{maximize}} \quad & \sum_{j=1}^{N}\lambda_j - \frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{N}\lambda_j\lambda_k y_j y_k K(\mathbf{x}_j,\mathbf{x}_k) \\ \text{subject to} \quad & \sum_{j=1}^{N}\lambda_j y_j = 0 \\ & 0 \le \lambda_j \le C, \quad j = 1,2,\ldots,N \end{aligned} \tag{6.53}$$

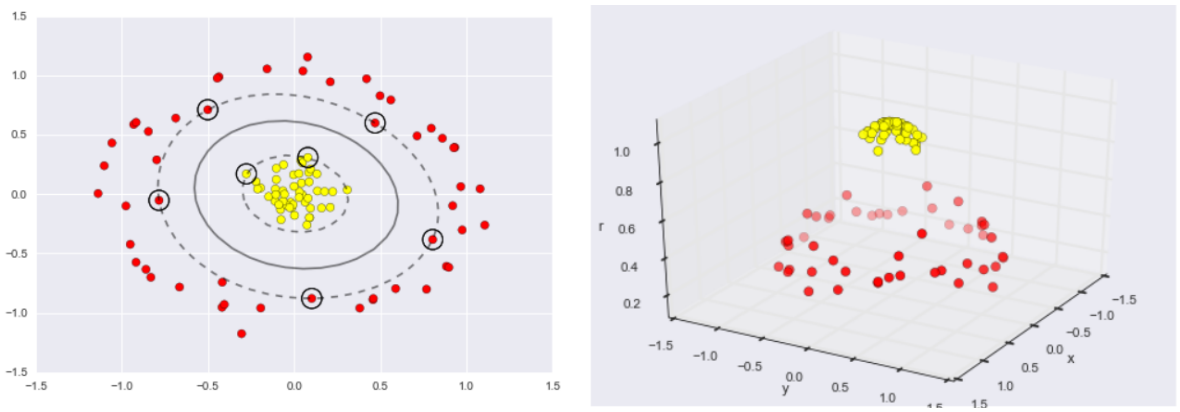
Solving this optimization problem yields the nonzero weights λ_j^*, $j = 1,2,\ldots,N_s$ for the support vectors, then $\mathbf{w}^*$ can be obtained from equation (6.34). The resulting linear (in the high-dimensional space) classifier is the same as in equation (6.45), but now the decision function is given, in $\mathbb{R}^Q$, by

$$g(\mathbf{x}) = \mathbf{w}^{*\mathrm{T}}\phi(\mathbf{x}) + b^* = \sum_{j\in\Omega_{\mathrm{sv}}}\lambda_j^* y_j K(\mathbf{x}_j,\mathbf{x}) + b^* \tag{6.54}$$

where the offset b^* can be obtained from the support vectors as

$$b^* = y_j - \sum_{k\in\Omega_{\mathrm{sv}}}\lambda_k^* y_k K(\mathbf{x}_j,\mathbf{x}_k) \tag{6.55}$$

for any j provided that $\lambda_j \neq 0$. The resulting classifier is nonlinear in the original space, because of the nonlinearity of the kernel function, see Figure 6.6.



(a) (b)

Figure 6.6: Example of a nonlinear SVM classifier for the case of two nonlinearly separable classes [47]. (a) The input space with nonlinear decision boundary, dotted lines mark the margin while points enclosed by circles are the support vectors. (b) The projected data, using a radial basis function centered on the middle clump, in three-dimensions, in which the data is linearly separable.

Remarks

- A notable characteristic of the support vector machines is that the computational complexity is independent of the dimensionality of the kernel space, where the input feature space is mapped. Thus the curse of dimensionality is bypassed.
- The generalization capabilities of the classifier depends on the selected kernel parameters and smoothing parameter C in the cost function. The most common procedure to obtain the best set of parameters, also known as *hyperparameters*, is to solve the SVM task for different sets of hyperparameters and finally select the SVM classifier corresponding to the set optimizing the adopted bound. For example, the bound may represent the number of correctly classified points (score) when the classifier is confronted with data outside the training set.

6.3.4 Summary

The Support Vector Machine (SVM) classifier is a linear model for classification problems. It can solve linear and nonlinear problems and work well for many practical problems. The SVM aims to find a hyperplane which separates the data into two/multiple classes. The SVM maximizes the margin around the separating hyperplane. The decision function is fully specified by a usually very small subset of training samples, known as the support vectors. This becomes a quadratic programming problem that can be easily be solved by standard methods.

The support vector machine is a powerful classification method for a number of reasons [47]:

- The method depends on relatively small number of support vectors, which means that the models are very compact and require very little memory.
- Once a model is trained, the prediction phase is very fast due to the simple formulation of the decision function/rule.
- The method is only affected by points near the margin, and therefore work well with high-dimensional data, even data with more dimensions than samples, which is challenging for most other algorithms.
- The formulation with kernels makes the method very flexible and suitable for many types of data.

However, SVM has several disadvantages as well:

- The scaling with the number of samples N is $\mathcal{O}[N^3]$ at worst, or $\mathcal{O}[N^2]$ for efficient implementations. For large number of training samples, this computational cost can be prohibitive.
- The results are strongly dependent on a suitable choice for the smoothing parameter C and the kernel parameters. This must be carefully chosen, for example via cross-validation, which can be computationally expensive for large data sets.

- The results do not have a direct probabilistic interpretation, as is the case for the Bayes classifier.

6.4 Application

In this section the SVM classifier is applied to the Zwartewaterbrug data. The goal is to correctly classify the data samples according to the corresponding weight of the added mass. In a SHM context, this is equivalent to detecting damage of different degrees of severity in the structure. The data for classification consists of N number of p-dimensional feature vectors $\mathbf{x} \in \mathbb{R}^p$ extracted from the measurements, with corresponding class label $y \in \{0\text{kg}, 25\text{kg}, 50\text{kg}, 75\text{kg}, 100\text{kg}\}$. The known class labels for the feature vectors makes the classification problem a supervised learning problem. The features used here are selected spectral lines of a spectrum; the type of spectrum depends on the chosen approach for classification as will be explained below.

Two approaches for classification are considered:

1. *Classification per segment* – a SVM model will be trained for each of the three bridge segments using the data of the particular segment. The spectral lines around the peaks in the singular value spectrum of the considered segment are used as damage-sensitive features. The singular value spectra associated with each segment were extracted from the processed signals (i.e. band-pass and similarity filtered signals) in Chapter 4.

2. *Classification per sensor* – a SVM model will be trained for each sensor using the data of the particular sensor. The spectral lines around the peak in the Power Spectral Density (PSD) functions of the considered sensor are used as damage-sensitive features. In Chapter 4 the PSDs associated with each sensor were computed from the processed signals.

For both approaches the number of samples/observations that are available for a SVM model is the same. The number of samples are divided over seven different datasets (corresponding to the seven datasets with raw signals). An overview of the number of samples per dataset together with the corresponding mass class is presented in Table 6.1. The number of samples apply for each segment/sensor, where a single sample represents either a singular value spectrum or a power spectrum depending on the problem being analyzed. Dataset 1 and 6 are not used in the training phase of the models, but are used for model validation in the first approach.

Table 6.1: Available number of samples per dataset for the classification problem, together with the corresponding mass class.

	1	2	3	4	5	6	7
Added mass (kg)	0	100	50	0	25	50	75
Number of samples	43	141	74	101	100	114	114

6.4.1 Classification per segment

Here, per segment a SVM model will be trained using the data of the particular segment. Hence, three different SVM models will be considered. The idea behind this distinction is that one would expect that the model of segment 2 shows better performance for classification than the other two models, because the added mass is located in segment 2. In this way more informative results are obtained.

Feature selection

Recall that feature extraction involves the generation of damage sensitive features from the measurements. Here, the singular value spectra extracted from the processed signals are used. A singular value spectrum represents the first singular value of the spectral density matrix per frequency. The resulting spectra are down sampled to a frequency resolution of 0.03 Hz. Next, the spectra are scaled to an equal order of magnitude as follows; per spectrum the singular values are first log-transformed, followed by the extraction of the mean value. The resulting spectral lines are referred to as *initial features*. Standardization of the features is not applied because the features already have the same units. Since the feature space is still high dimensional, further reduction is achieved by a simple feature selection approach consisting of two parts. First, a frequency region with many modal peaks is manually selected, followed by the selection of spectral lines in this region

with a variance larger than 5% of the maximum variance. The last feature selection step, actually already forms part of the classifier and the variance of each feature should be computed using the training data set only. This feature selection method is motivated by the idea that low variance features are less important for classification. In our case the initial features have the largest variance around the peaks, which is the region that is expected to be most important for fault detection. Hence, dimensionality reduction is achieved without significant loss of fault related information. The initial features and the selected features for classification are shown in the left and right figure of Figures 6.7–6.9, respectively. In the left figure, the features that are selected from the initial features are highlighted, and these features are shown in the figure on the right.

Classification

Once the damage-sensitive features are defined, classification can be performed. In summary the analysis consists of the following steps:

Step 1. Choose a classification model.
Step 2. Split the data set into two subsets for training and testing.
Step 3. Determine appropriate values for the hyper-parameters of the model (if there are any); e.g. using cross-validation.
Step 4. Train the model using the training data set.
Step 5. Evaluate the performance of the model using the testing data set.

In **Step 1** any classification model can be chosen. However, the analysis here is restricted to the SVM classifier because this model is very suitable for small data sets with high dimensional features. A Radial Basis Function (RBF) kernel (6.56) is used to implicitly map the features into a higher-dimensional space.

$$K(\mathbf{x},\mathbf{z}) = \exp\left(-\gamma\|\mathbf{x}-\mathbf{z}\|^2\right) \tag{6.56}$$

where the parameter γ controls the size of the RBF kernel, and should be specified before training. Furthermore, the one-versus-one approach is employed to separate the multiple classes, as discussed in Section 6.3.2. This means that for each pair of classes a classifier is trained (for 5 classes, 10 classifiers are trained). Classification is then achieved based on majority vote.

The available data per segment is split into two subsets for training and testing of the models. The three models will be trained using 80% of the data, and the other 20% is held back for testing. The training data set is also used to tune the hyper-parameters (C and γ) by means of cross-validation. The number of samples in the training and testing data set per mass class are listed in Table 6.2. This shows that the different classes are more or less equally represented in both the training and testing data set, implying that the data sets are balanced. The applied training/testing splits are the same for the data of each segment, in other words the same number of data samples are used for training and testing of the three models.

As already mentioned, the performance of the models will also be validated using two completely new data sets (i.e. set 1 and 6 in Table 6.1). While the previously defined testing sets contain samples from the same sets that are used for training the model, the two new data sets contain observations that are obtained under different environmental conditions than those in the training/testing sets.

Table 6.2: Number of data samples in training and testing data set per mass class.

Mass class	Number of samples (% of total)	
	Training	Testing
0 kg	81 (15.28)	20 (3.77)
25 kg	83 (15.66)	17 (3.21)
50 kg	57 (10.75)	17 (3.21)
75 kg	91 (17.17)	23 (4.34)
100 kg	112 (21.13)	29 (5.47)
Total	424 (80.00)	106 (20.00)

The SVM model with RBF kernel contains two *hyper-parameters*, C and γ, which must be chosen in **Step 3** before training. The parameter C controls the hardness of the margin; it trades off misclassification of

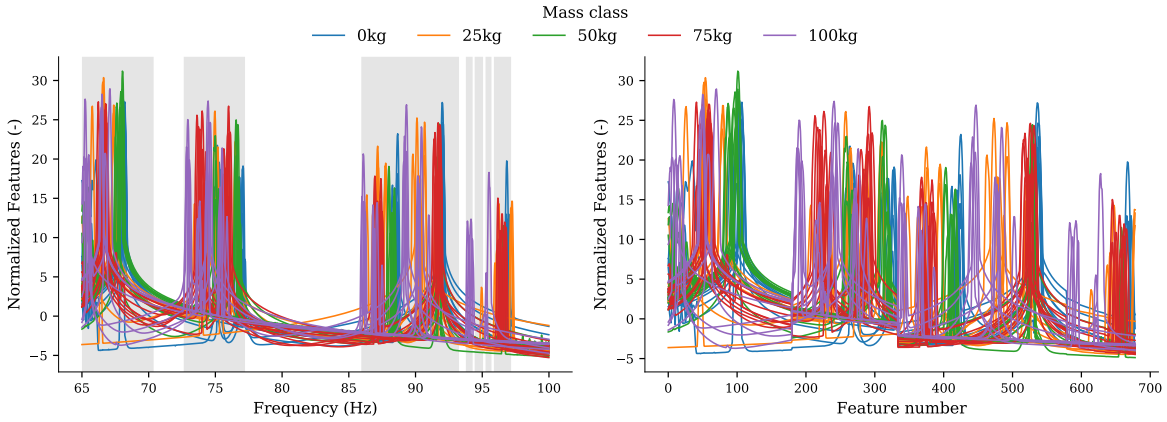


Figure 6.7: Segment 1 – Features for classification before (left) and after (right) variance-based feature selection. Initially, the spectra consist of 1166 spectral lines between 65 and 100 Hz (frequency resolution 0.03 Hz), and 679 spectral lines are selected with a variance of at least 5% of the maximum variance in the training data set.

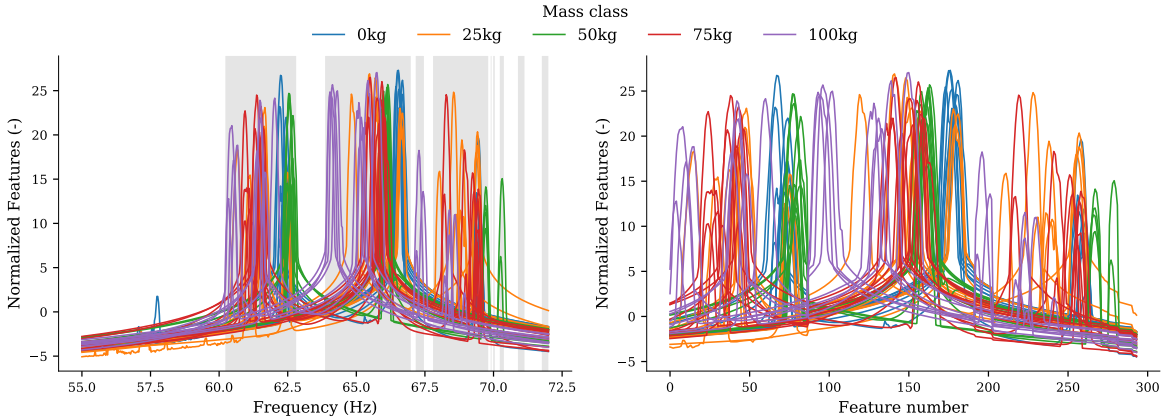


Figure 6.8: Segment 2 – Features for classification before (left) and after (right) feature selection. Initially, the spectra consist of 566 spectral lines between 55 and 72 Hz (frequency resolution 0.03 Hz), and 294 spectral lines are selected with a variance of at least 5% of the maximum variance in the training data set.

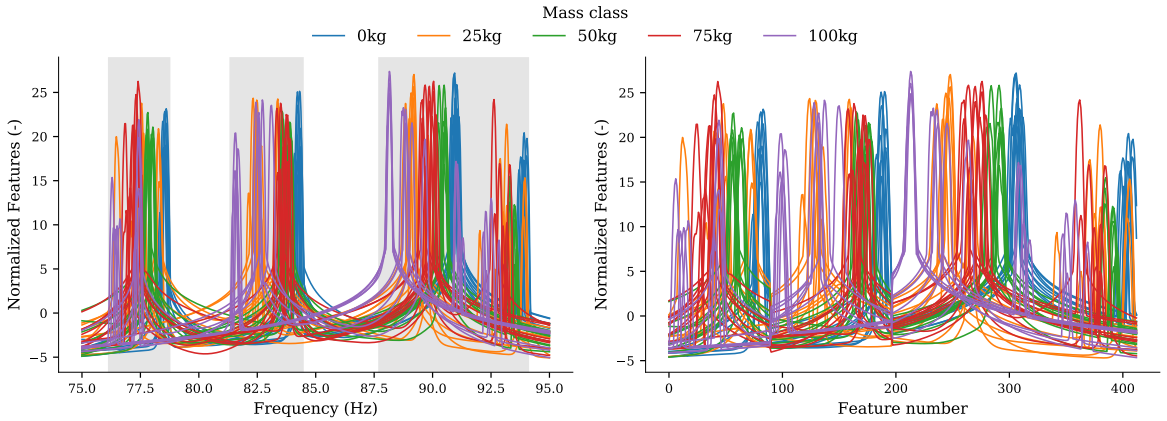


Figure 6.9: Segment 3 – Features for classification before (left) and after (right) feature selection. Initially, the spectra consist of 666 spectral lines between 75 and 95 Hz (frequency resolution 0.03 Hz), and 413 spectral lines are selected with a variance of at least 5% of the maximum variance in the training data set.

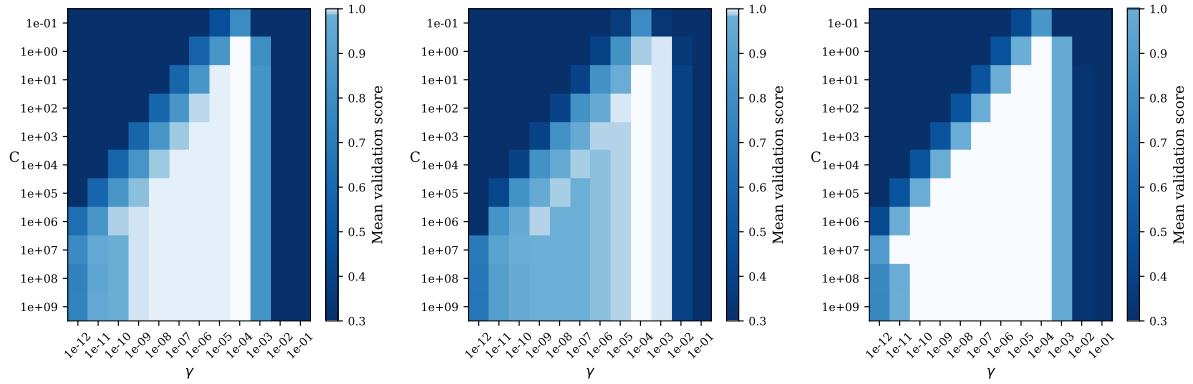
training examples against the simplicity of the decision surface. A low C makes the decision surface "soft", while a high C aims at classifying all training examples correctly. The γ parameter controls the size of the RBF kernel. Intuitively, it defines how much influence a single training example has. The larger γ is, the closer other examples must be to be affected. This parameter can also be seen as the inverse of the radius of the area of influence of the support vectors of the model. The behavior of the model is very sensitive to the choice of the parameters, hence it is important to determine appropriate values. The optimal parameters are determined from a parameter grid via an optimization approach based on cross-validation. For the C values a logarithmic grid from 1e−1 to 1e+9 is considered and for the the γ values a logarithmic grid from 1e−12 to 1e−1 is considered (in practice, a logarithmic grid from 1e−3 to 1e+3 is usually sufficient but this is extended here for illustrative purposes). For each combination of parameters a number of models are trained and validated, each time using a different fraction of the training data set for training and validation (i.e. cross-validation). The training and validation score of each run and for each combination of parameter are stored. Finally, the optimal parameters are chosen as the one that result in the highest mean validation score. It is noted that using an average score rather than using a single score is better for determining the hyper-parameters. The optimal hyper-parameters are not the same for the three models, so the optimization should be performed for each model separately.

A note on cross-validation; one could split the data set into three smaller data sets for training, validation and testing, respectively. The training data set is used to fit the model (e.g. to determine coefficient of separating hyperplane in the case of SVM). The validation set is used to evaluate a given model while tuning hyper-parameters, hence this is for frequent evaluation. The testing set is used to evaluate a model once it is completely trained (using the training and validation sets). This latter set can be seen as unknown data, because the model has not "seen" it before during training. One disadvantage of using one particular subset of the data for validation is that a portion of the data have been lost for training the model. One way to address this is to use *cross-validation*; that is, to do a sequence of fits where each subset of the data is used both as a training set and as a validation set. The validation scores could be combined, for example by taking the mean, to obtain a single validation score. This score is in general a better measure of model performance than the score of a single validation set. The most popular method is K-fold cross validation, in which the data set is split into K (randomized) folds, each fold is then used once as a validation set while the $K-1$ remaining folds form the training set. Several variations of this method exist; for example, the stratified K-fold cross validation method in which the percentage of data samples for each class is preserved.

For the optimization of C and γ, the training data set is randomly split in a stratified fashion, into 90%/10% sets for training and validation. Fifteen splits are performed for each combination of parameters to smooth out spurious variation in the scores that are the result of random splits. Figures 6.10a–6.10c show the mean validation scores for each combination of parameters for the three models. This shows that if γ is smaller than 1e−12, the models are too constrained and cannot capture the complexity or shape of the data; that is, the models are *underfitting* the data. The region of influence of any selected support vector will include the whole training set. The resulting model will behave similarly to a linear model with a set of hyperplanes that separate the centers of high density of any pair of classes. On the other hand, if γ is larger than 1e−2, then the radius of the area of influence of the support vectors only includes the support vectors itself and no amount of regularization with C will be able to prevent *overfitting*. Both, underfitting and overfitting leads to poor performance of the models when evaluated on the validation set (i.e. poor generalization).

For intermediate values, the figures show that good models can be found on a "diagonal" of C and γ values. Smooth models (lower γ values) can be made more complex by increasing the importance of classifying each point correctly (larger C values), without loss of generality. It can also be observed that for some intermediate values of γ, equally well performing models are obtained when C becomes very large. In particular, for γ = 1e−4 for the models of segment 1 and segment 2, and for values of γ between 1e−10 and 1e−4 for the model of segment 3. This implies that it is not necessary to regularize by enforcing a larger margin. It appears that the radius of the RBF kernel alone acts as a good structural regularizer. In practice though it might still be interesting to simplify the decision function with a lower value of C, to favor models that use less memory and that are faster to predict.

In Table 6.3 the optimal parameters are presented that were found for the three models. The optimal values for γ are the same for all three models, while the optimal values for C slightly differ. However, from Figure 6.10 it was observed that for the optimal values of γ the amount of regularization by C is not so important.



(a) Model 1 **(b)** Model 2 **(c)** Model 3

Figure 6.10: Grid search optimal parameter selection. The three models correspond to the three bridge segments.

Table 6.3: Optimal hyper-parameters for SVM model per model.

Model	C	γ
1	1	1e−4
2	10	1e−4
3	1	1e−4

One important aspect of the hyper-parameters (i.e. model complexity) is that the optimal values will generally depend on the size of the training set. Therefore, it is useful to explore the behavior of the model with given complexity as a function of the number of training points. A graph showing the training and validation score with respect to the size of the training set is known as a *learning curve*. This kind of graph is often used to find out how much the model benefits from adding more training data, and whether the estimator suffers more from a variance error (overfitting) or a bias error (underfitting). Figures 6.11a–6.11c show the *learning curves* of the three models corresponding to the three different segments. The notable feature of the learning curve is the convergence to a particular score as the number of training samples grows. In particular, if enough training examples are used so that a particular model has converged, then adding more training data will not improve the model any further. If the model performance in this case is satisfactory, then the only way to improve it is to use another (often more complex) model. The curves show that the model overfits small data sets (relatively high training score, but relatively low validation score), while better fits are obtained when increasing the size of the data set. The models of segment 1 and 3 have converged to a high score when all the training examples are considered, while model 2 is not converged yet (although the difference with the other two models is small). The performance of model 2 will most likely increase when adding more training samples. The model for segment 1 has the least steep learning curve. The number of training samples that are necessary for the three models to obtain a (mean) validation score larger than 0.9 is 87, 73 and 61, respectively. Hence, the model of segment 3 requires least amount of training samples, i.e. has steepest learning curve.

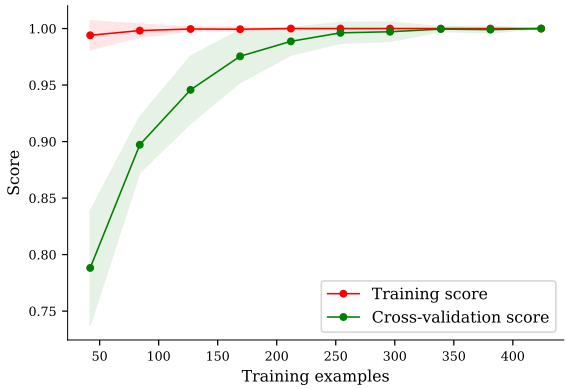
Once the optimal hyper-parameters are selected, the final models can be trained using all samples of the training set. The training phase of a model consists of the following steps:

Step 1. Selection of the spectral lines that will be used as features. The selection is based on a minimum amount of variance of the spectral values in the training data set as explained before.

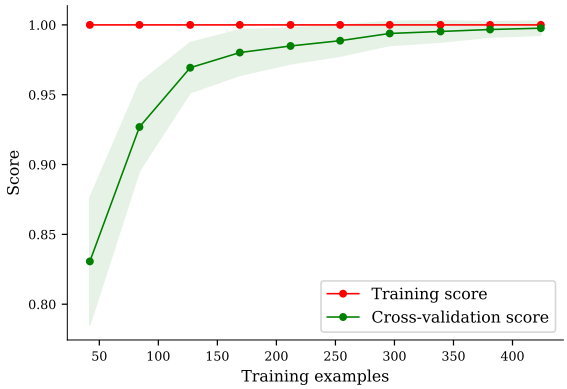
Step 2. Training of the SVM model.

Both steps must be part of the same training process to avoid data leakage; in other words, to avoid that some testing samples are also considered in the training phase. The features that are selected in **Step 1**, were presented in Figures 6.7–6.9 for each model.

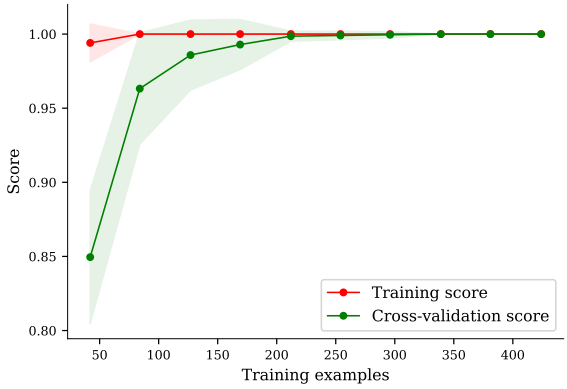
In **Step 2**, training a SVM model implies solving the optimization problem (6.53) for the dual coefficients λ_j, $j = 1,2,\ldots,N$, which quantify the importance of each training sample in forming the decision boundary. As many coefficients as observations in the considered training set are solved, many of them will be zero



(a) Model 1



(b) Model 2



(c) Model 3

Figure 6.11: Learning curve. The three models correspond to the three bridge segments.

while the nonzero coefficients define the support vectors. In Table 6.4 the number of support vectors per model are listed. The value in between brackets represents the percentage of the training samples that are support vectors. The total number of support vectors is subdivided into the number of vectors per mass class. For model 1 almost 60% of the training samples are used as support vector, for model 2 and 3 this is about 46% and 42%, respectively. The highest percentage of support vectors are found for the 25 kg followed by the 100 kg mass class. The large number of support vectors implies that almost all training vectors of the particular class are considered as the critical elements of the training set. Consequently, the resulting decision surfaces has high complexity and the models might perform poorly for unseen data (except when the unseen data is very close to the data used for training). The large number of support vectors can be related to the large variations in the training data. Recall that the 25 kg and 100 kg mass classes correspond to dataset 5 and 2, respectively. For these two datasets it was observed that the natural frequencies of the bridge changes considerably over time, see for example Figures 4.17–4.19. It is noted that the natural frequencies are related to the peaks in the singular value spectra, in the sense that a change of natural frequency is equivalent to a shift of the corresponding peak in a singular value spectrum. So, large variations exist among the feature vectors of these two mass classes and to account for these variations many support vectors are required. As mentioned before, the observed variations are presumably related to the varying environmental conditions.

Subsequently, the trained models are used to predict labels for the data samples in the testing data sets (note that the testing sets, just like the training sets, are not the same for the three models). The results of the three models are presented in Figures 6.12–6.14. The figures on the left show the initial testing vectors in which the highlighted regions indicate the selected features. It is noted that the selected features are based on the training data set and not on the testing set (i.e. the highlighted regions in these figures are the identical to the ones in Figures 6.7–6.9). The figures on the right visualize the performance of the models in the form of a confusion matrix, displaying the predicted class against the true class. Both the number of observations predicted and the corresponding percentage of test observations in the true class (i.e. the class supports) are

Table 6.4: Number of support vectors used by the SVM model per segment.

Mass class	Number of support vectors (% of training samples)		
	Model 1	Model 2	Model 3
0 kg	36 (44.44)	22 (27.16)	20 (24.69)
25 kg	67 (80.72)	57 (68.67)	56 (67.47)
50 kg	27 (47.37)	14 (24.56)	17 (29.82)
75 kg	43 (47.25)	43 (47.25)	32 (35.16)
100 kg	75 (66.96)	57 (50.89)	53 (47.32)
Total	248 (58.49)	193 (45.52)	178 (41.98)

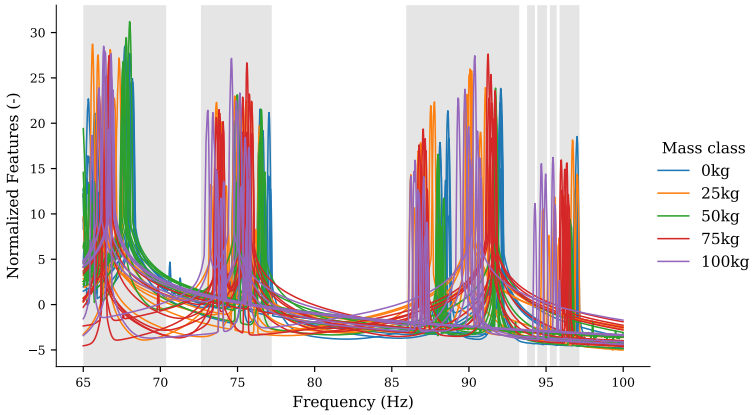
shown in each cell of the matrix. This shows that the models are able to correctly classify almost all samples from the testing data set. Most misclassification are observed for the 25 kg and 100 kg class.

A brief explanation for the interpretation of the confusion matrix. The diagonal elements represent the number of test samples for which the predicted label is equal to the true label, while the off-diagonal elements are those that are mislabeled by the classifier. In particular, the off-diagonal elements in the top row (true class 0 kg and predicted class a nonzero mass) represent the false-negatives or false alarm; that is, the model predicts "damage" while this is actually not the case. On the other hand, the off-diagonal elements in the column on the far left (predicted class 0 kg and true class a nonzero mass) represent the false-positives or missed detection; that is, the model predicts "no damage", while there might actually be damage. As mentioned earlier, the latter might have more severe consequences.

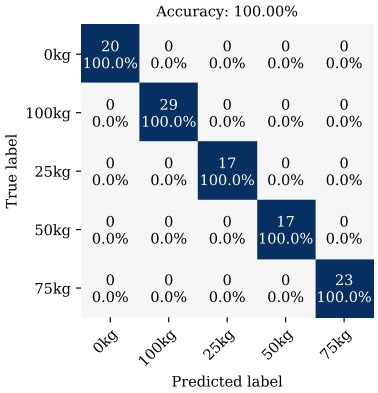
The performance of all three models is remarkable high. However, as already mentioned, the observations in the training and testing sets are from the same data sets. This means that the testing set most likely consist of similar patterns as those in the training set used for training the models. So, even though the models have not seen the testing data during training, it is not surprising that the models are able to classify all test observations correctly. It is more interesting to see whether the models generalize well on completely new data. For this we will use dataset 1 (0kg) and 6 (50 kg) as defined in Table 6.1. The samples (i.e. the singular value spectra) of these sets and the corresponding classification results are shown in Figures 6.15–6.17. This shows that the models are unable to classify the new data correctly. Most of the samples are assigned to either the 25 kg or 100 kg class. It appears that the current models do not have enough flexibility to suitably account for all the features in the new data. The best way to address this problem is to train new models using samples from all the available data sets. Hence, the process described above is repeated, but now using all the available data samples. The performance of the models is again validated using 20% of the data that was held back during training. The testing samples are shown in Figures 6.18a–6.20a, including the selected spectral lines that are used as features for the new models. From this it is immediately clear that the new models use considerably more spectral lines as features than previous models, which is necessary to account for all the features in the data. The performance of the new models are visualized in Figures 6.18b–6.20b. This shows that the new models are able to correctly classify most of the samples, including those for which the previous models failed.

6.4.2 Classification per sensor

Previously, a SVM model was trained for each bridge segment separately, and the damage-sensitive features were extracted from the measurements of the particular segment. Alternatively, we could extract features from the measurements of a single sensor or from a subset of sensors. Suppose that for each sensor a model is trained, then the performance of a model might be an indication of the sensitivity of the corresponding sensor/location to the added mass, which on its turn might allow for the localization of the mass. For this it is presumed that a model associated with a sensor closest to the added mass performs the best. This is reasonable because the features extracted from a sensor close to the added mass are most likely the most sensitive to the added mass. However, the features are also sensitive to other factors such as the environmental conditions. So, the performance of a model will also be affected by these factors, and a good performing model does not necessarily implies that the model is good for detecting the added mass.

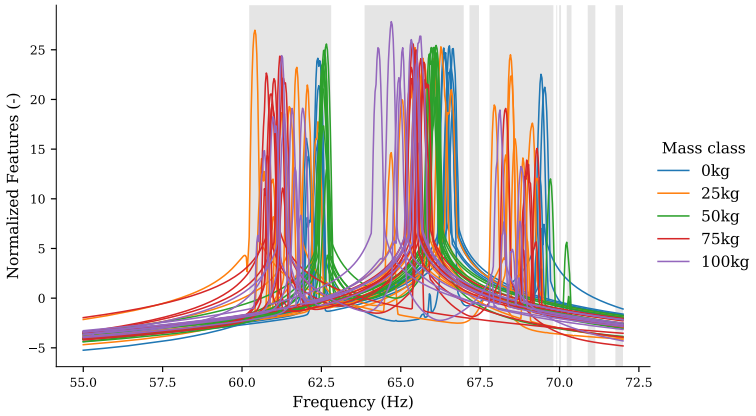


(a) Testing samples.

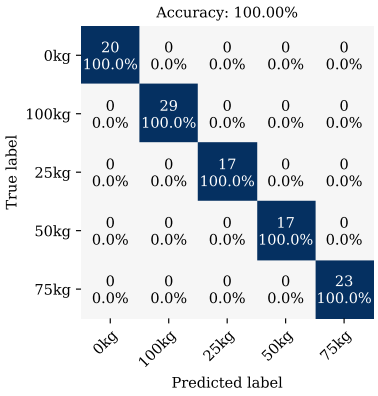


(b) Confusion matrix.

Figure 6.12: Segment 1 – Classification of the test data using model 1 trained with the samples shown in Figure 6.7. In (a), the selected spectral lines that are used as features are highlighted.

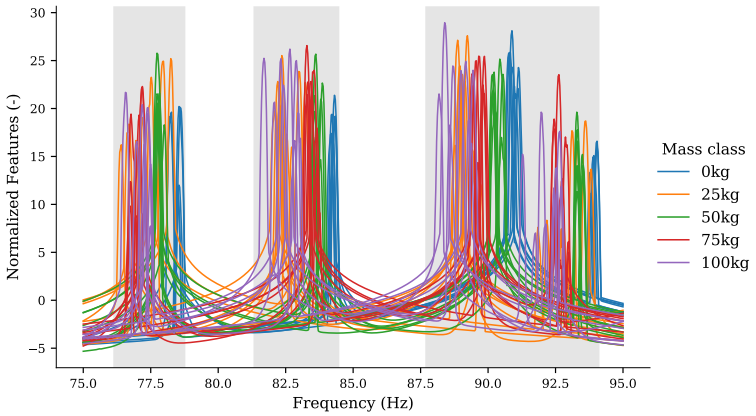


(a) Testing samples.

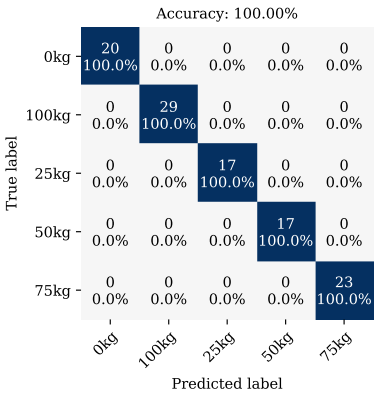


(b) Confusion matrix.

Figure 6.13: Segment 2 – Classification of the test data using model 2 trained with the samples shown in Figure 6.8. In (a), the selected spectral lines that are used as features are highlighted.

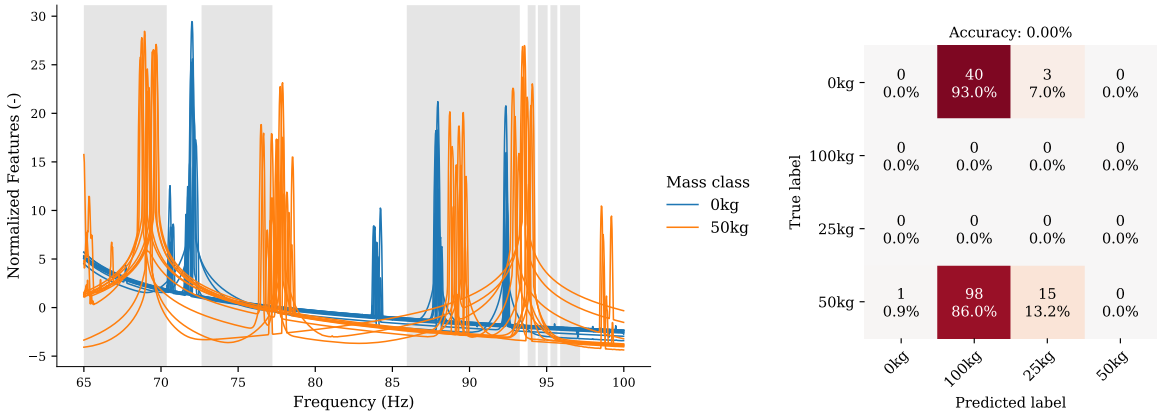


(a) Testing samples.



(b) Confusion matrix.

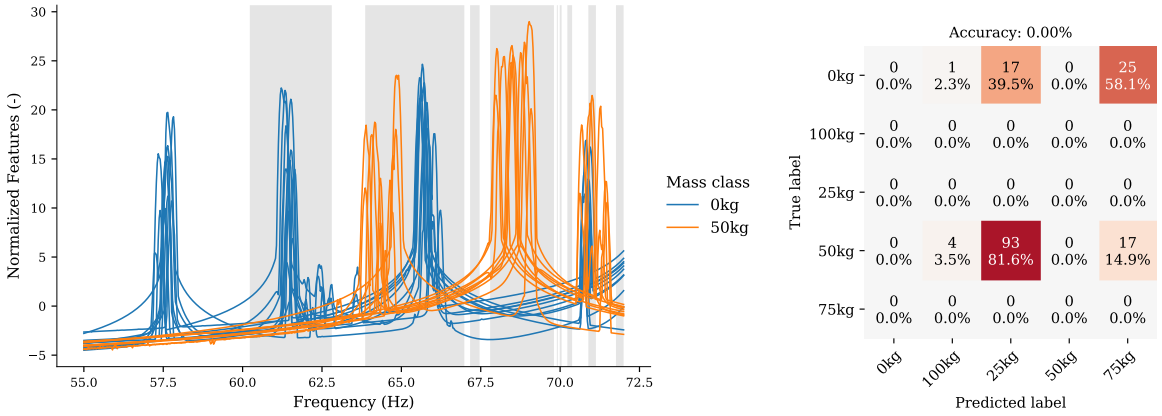
Figure 6.14: Segment 3 – Classification of the test data using model 3 trained with the samples shown in Figure 6.9. In (a), the selected spectral lines that are used as features are highlighted.



(a) Samples from dataset 1 (0 kg) and 6 (50 kg).

(b) Confusion matrix.

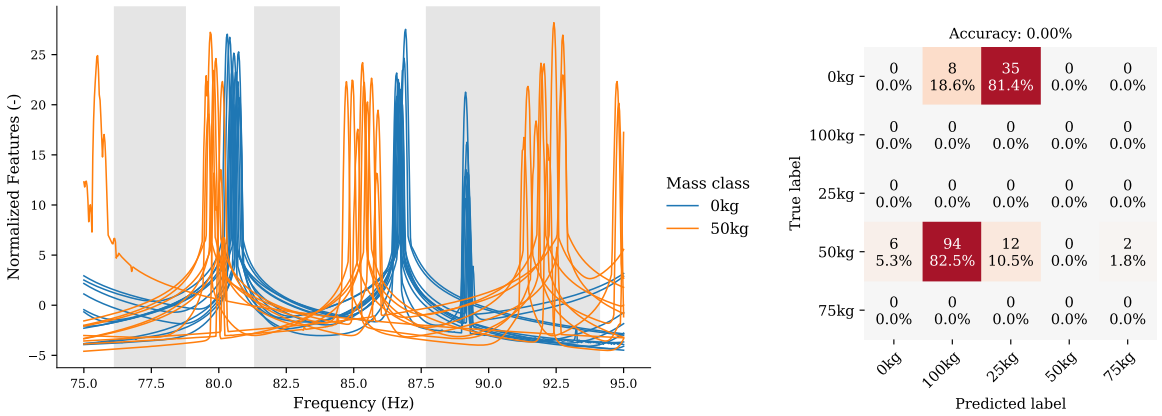
Figure 6.15: Segment 1 – Classification of the new data using model 1 trained with the samples shown in Figure 6.7. In (a), the selected spectral lines that are used as features are highlighted.



(a) Samples from dataset 1 (0 kg) and 6 (50 kg).

(b) Confusion matrix.

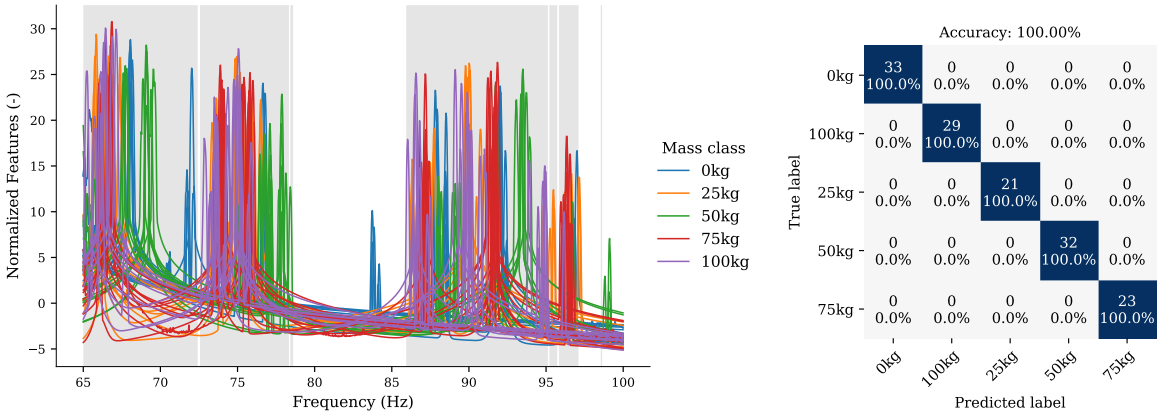
Figure 6.16: Segment 2 – Classification of the new data using model 2 trained with the samples shown in Figure 6.8. In (a), the selected spectral lines that are used as features are highlighted.



(a) Samples from dataset 1 (0 kg) and 6 (50 kg).

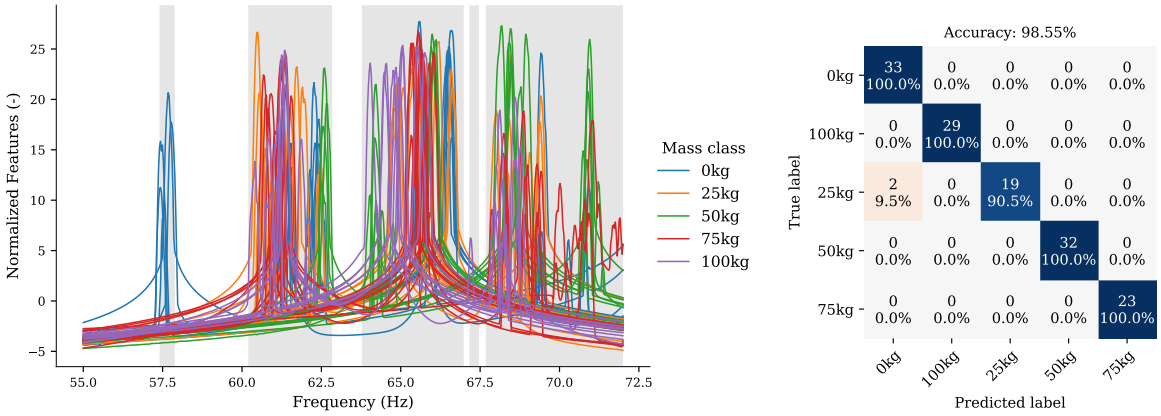
(b) Confusion matrix.

Figure 6.17: Segment 3 – Classification of the new data using model 3 trained with the samples shown in Figure 6.9. In (a), the selected spectral lines that are used as features are highlighted.



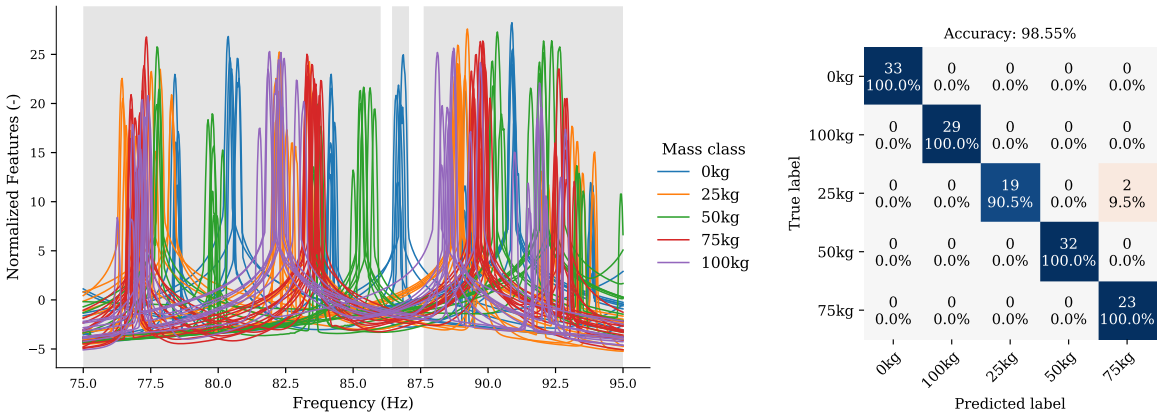
(a) Testing samples. **(b)** Confusion matrix.

Figure 6.18: Segment 1 – Classification of testing samples using a new model for segment 1 trained with feature vectors extracted from all the datasets (training vectors are not shown). In (a), the selected spectral lines that are used as features are highlighted.



(a) Testing samples. **(b)** Confusion matrix.

Figure 6.19: Segment 2 – Classification of testing samples using a new model for segment 2 trained with feature vectors extracted from all the datasets (training vectors are not shown). In (a), the selected spectral lines that are used as features are highlighted.



(a) Testing samples. **(b)** Confusion matrix

Figure 6.20: Segment 3 – Classification of testing samples using a new model for segment 3 trained with feature vectors extracted from all the datasets (training vectors are not shown). In (a), the selected spectral lines that are used as features are highlighted.

Next, the idea discussed above is further elaborated with an application on the data. A SVM model is trained using feature vectors extracted from the measurements of a single sensor, in order to investigate whether certain sensors are more sensitive to the added mass. In this analysis, data set 1 and 6 are disregarded, and the number of samples and the corresponding mass class of the remaining datasets can be found in Table 6.1. The features are now selected from the PSD of a single sensor. The feature selection procedure is conducted in the same way as before; that is, the power spectra are down sampled to a frequency resolution of 0.03 Hz and scaled to equal order of magnitude. The spectral lines with a variance larger than 5% of the maximum variance in the training set will be selected during training of the classifier model. The variance of each spectral line is computed across the training data set. The selected spectral lines are used as features for classification. A SVM model will be trained for every *relevant* sensor. In Chapter 4 it was found that for 15 sensors the spectra have relatively low magnitude and show large variations, as can be observed from Figures F.1–F.4 in the appendix. So, it is impossible to extract useful features from the spectra of these sensors and thus no model will be trained for these sensors. For each of the remaining sensors a SVM model is trained in the same way as we did before. Once a model is trained, labels are predicted for the testing samples and the performance of the model can be validated. The results per sensor are shown in Figures H.2–H.6 in the appendix. There the testing samples are presented, and the performance of the model is visualized in the form of a confusion matrix. An overview of the accuracy scores of the different models, each one associated with a different sensor, is shown Figure 6.21. The accuracy score is computed as the number of correct predictions divided by the total number of predictions. In this figure the scores are related to the spatial position of the sensors, and the location of the added mass is indicated with a red dot between sensors 16 and 17. The 15 sensors for which no model could be trained are labeled with NA (Not Available).

In general, the scores are relatively high for all considered sensors. This means that the features extracted from a sensor close to the added mass are not necessarily more suitable for classification than features that are extracted from sensors further away from the added mass. So, from this it cannot be concluded that a specific sensor/location is more sensitive to the added mass. By analyzing the power spectra (i.e. feature vectors) that are used here, it can be observed that the shift in peak frequency is very similar for all sensors. This implies that there are some variables that affect the dynamic properties of the bridge globally, while the added mass will only have a local effect. Environmental variables (e.g. temperature) are the most obvious variables that will cause these global changes.

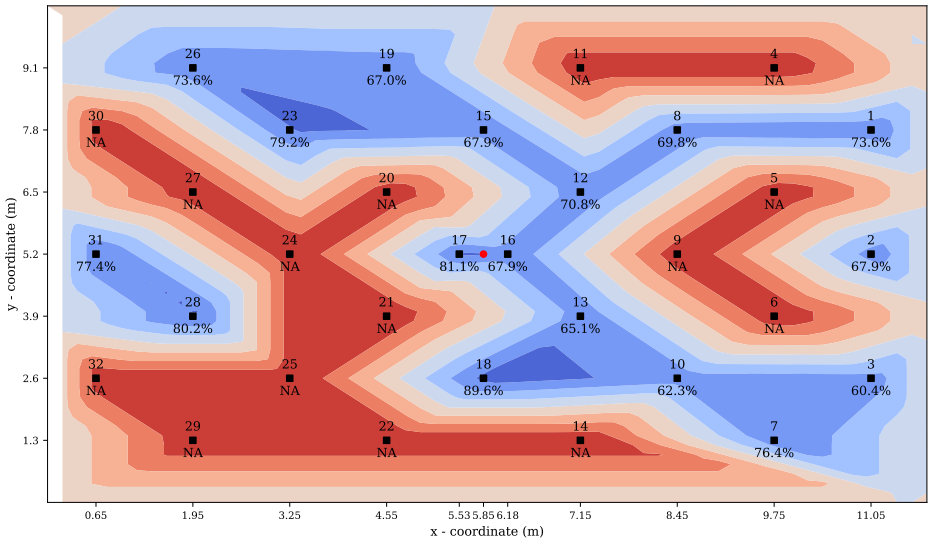


Figure 6.21: Accuracy scores of SVM models that are trained using feature vectors extracted from measurements of a single sensor. The selected spectral lines of the power spectra are used as features. The location of the added mass is indicated with a red dot between sensors 16 and 17.

6.5 Conclusion

In this section the capability of the support vector machine classification for damage detection was analyzed. Two different approaches were considered: 1) classification per segment, and 2) classification per sensor. In both approaches multiple models were trained; in the first approach a SVM model was trained for each bridge segment using feature vectors extracted from the particular segment, and in the second approach a SVM was trained for each relevant sensor using feature vectors extracted from the particular sensor. The main difference between the two approaches is the type of features used. In both approaches the features are spectral lines that are selected from of a certain spectrum; in the first approach the singular value spectrum of a segment is considered, while in the second approach the power-spectral density of a sensor is considered. The steps that are performed to build each SVM model are identical. The performance of a model was quantified with an accuracy score, and visualized in the form of a confusion matrix, displaying the predicted labels versus the true labels. From the results obtained in previous two sections the following observations were made:

- The feature vectors in a dataset of a mass class show large variations (especially for the 25 kg and 100 kg mass class). Moreover, the variations are very similar for the three segments, and also for the 17 considered sensors. Since the added mass is very small and only present at a single location, the effects of the added mass are very local and will not have caused the "overall" variations in the feature vectors. This implies that the feature vectors are affected by some global factors, presumably related to the environmental conditions. Recall that the features are selected from the singular value spectrum (classification per segment) and power spectrum (classification per sensor). Both types of spectra are related to the dynamic characteristics of the bridge and will be sensitive to environmental variability. Hence previous statement is reasonable. This was already found in previous chapters, where it was observed that the peaks in the spectrum changes considerably over time.
- In general, all the SVM models that were trained use a large number of support vectors to separate the different mass classes. Especially for the 25 kg and 100 kg class, which can be attributed to the large variations of the feature vectors as mentioned in the previous point.
- The classification results of the three SVM models associated with the three bridge segments are comparable. Even though, it is reasonable to expect that the model associated with segment 2 (with the added mass) would perform better than the the models for segment 1 and 3. Similarly, the classification results of the models associated with the 17 sensors were also comparable, while it was expected that the models associated with the sensors close to the added mass perform better than the other sensors. Based on these results it is very unlikely that the separability of the mass classes is truly caused by the added mass itself. Instead, it is concluded that the large variations of the features vectors, as mentioned in the first point above, mask the effects of the added mass on the feature vectors.
- The SVM models perform very poorly when observations are tested that were not part of the train/test datasets. In other words, the models do not generalize well for new data. Again, this can be attributed to the large variations of the feature vectors. If a portion of the new observations is considered in the training phase of the models, the newly trained models are able to classify all test observations correctly, including those for which the previous models failed.

The observations have in common that they can all be related to the large variations in the feature vectors that are presumable caused by changing environmental conditions. However, it must be noted that the proposed methodology is not intended to account for the environmental variability, but only for the operational variability by means of the similarity filtering. So, these observations are not very surprising. Nevertheless, they emphasize the importance of *data normalization*, which aims to remove the variations of the feature vectors that are caused by changing environmental conditions.

In conclusion, the current dataset of the Zwartewaterbrug is not suitable for the application of a SVM classification (as part of the followed methodology). In general it may not be suitable for any type of machine learning algorithm because of the limited amount of data in combination with the distinct environmental conditions under which the data for the different mass classes are obtained. The latter issue could be solved by proper data normalization as discussed in Chapter 8. However this generally requires a dataset that is obtained over a sufficient long time period (to account for a full range of environmental conditions).

7

Unsupervised Learning - Novelty Detection

In previous chapter we have introduced the basic idea of the pattern recognition aspect of machine learning. It was assumed that the training data for the expected damage states is available, and the employed methods were supervised learning methods. As already mentioned, it is not always possible to obtain data of the damaged structure. In these cases, unsupervised learning algorithm can be employed. Two of the main methods used in unsupervised learning are principal component analysis and cluster analysis. These methods use unlabeled training data, hence no prior information regarding the nature of the data is used. However, it is often assumed that the training data describes the normal condition of the structure; meaning that the structure is assumed in normal operating state during acquisition of the training data. Then, during monitoring, one needs to determine whether a new observation belongs to the same distribution as the normal condition, or whether it is significantly different. The involved techniques are called *novelty detection* or *outlier detection*, which both can be used for anomaly detection. The names of the methods are often used interchangeable, however the methods make different assumptions regarding the training data. In outlier detection, the training data contains outliers which are defined as observations that are far from the others. Outliers detection estimators thus try to fit the regions where the training data is the most concentrated, ignoring the deviant observations. In novelty detection, the training data is not polluted by outliers and the aim is to detect whether a new observation is an outlier. In this context an outlier is also called a novelty. Both methods are used for anomaly detection, where one is interested in detecting abnormal observations. Outlier detection is then also known as unsupervised anomaly detection and novelty detection as semi-supervised anomaly detection.

In general, the anomaly detection is based on certain statistics/measures that are derived from the training data. These statistics are either *inclusive* or *exclusive* depending on whether or not the potential outliers are included in the training data [19]. Hence, outlier detection essentially uses inclusive measures, while exclusive measures are used in novelty detection.

In the context of SHM, it is often assumed that the training data represents the normal/healthy condition of the structure and does not contain outliers. The outliers correspond to observations from an abnormal/damaged condition of the structure. In this sense, novelty detection is of most interest and will be applied in this thesis.

7.1 Discordance Measures

The basic idea of novelty detection is to compute discordance measures (also called novelty scores) for the data and then compare the measures with a threshold; if a measure exceeds the threshold then the data is flagged as novelty, which implies that the structure has departed from its normal condition. The decision function can be expressed as

$$D(\mathbf{x}) < \xi \tag{7.1}$$

where $D(\mathbf{x})$ is some appropriate discordance measure and ξ is a threshold value. The discordance measure indicates how much a certain observation $\mathbf{x}$ departs from the distribution of the normal condition data. A suitable choice for this measure is [19]

$$D(\mathbf{x}) = -\ln\big(p(\mathbf{x})\big) \tag{7.2}$$

where $p(\mathbf{x})$ is the probability density function of the normal condition data. Note that the probability density yield values between zero and one, so the discordance measure ranges from zero to infinity, where low scores correspond to high probabilities and high scores correspond to low probabilities.

It is assumed that the probability density function $p(\mathbf{x})$ of the damage-sensitive features can be estimated from the measured data, and that this function represents the normal/healthy condition of the structure. Note that this function is essentially the same as the conditional probability $p(\mathbf{x}|H)$ that was used in previous chapter, because it is assumed that the data comes from a healthy structure. The probability distribution associated with the damaged condition of the structure does not need to be known. This is a big advantage over the damage detection methods discussed in previous chapter. Once this probability density function is known, new observations can be labeled as normal or abnormal on the basis of the novelty score for this new feature vector; that is, feature vectors that have a novelty score higher than a predefined threshold value are diagnosed as abnormal (implying significant departures from the normal condition of the structure). This can be interpreted as follows: features with large novelty score have low probability values and are thus unlikely to have come from the undamaged distribution. The determination of a suitable threshold value will be discussed later.

It must be noted that the estimation of the probability density function in its full generality, is essentially the hardest problem of all machine learning problems [19]. There are numerous nonparametric methods of estimating densities for multivariate data, such as kernel density estimation (KDE) or Gaussian mixture models (GMM). Alternatively, the problem can be made more tractable at the expense of generality if a parametric form for the probability density is assumed. The Gaussian distribution is adopted in most cases, which is restricted to data that can be characterized by its first two statistical moments; the mean and variance (or covariance for multidimensional feature vectors). The Gaussian assumption leads to a much simpler discriminator function that depends on only a small number of parameters (i.e. much smaller than the number of parameters involved in the nonparametric density estimation methods).

7.1.1 A Gaussian Distributed Normal Condition

As discussed above, one can make progress in novelty detection by assuming a particular form for the probability distribution of the features that define the normal conditions. The method discussed in this section is tailored to the Gaussian distribution and therefore depends on the implicit assumption that the data can be characterized by its first two statistical moments: the mean and covariance. This assumption reduces the problem to one where the probability density function of the normal condition data is fixed by the estimation of a small number of parameters; the main drawback is that the assumption of Gaussianity is by no means always merited. However, the method may still perform well if the departures from a Gaussian distribution are small; that is, the normal condition data looks "elliptical" in the feature space and its probability density function is unimodal (only has one peak). If the data departs radically from these assumptions, for example, if the distribution is not convex or, in the worst case, actually separated into distinct unconnected regions, the novelty detection may fail badly. It is thus important to assess the assumption of Gaussianity for the normal condition data.

Recall that the probability density function of a multivariate Gaussian distribution is given by

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{p/2}|\mathbf{C}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}})^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{x}-\bar{\mathbf{x}})\right) \tag{7.3}$$

where $\bar{\mathbf{x}} \in \mathbb{R}^p$ and $\mathbf{C} \in \mathbb{R}^{p\times p}$ are the sample mean vector and the sample covariance matrix, respectively, and $|\mathbf{C}|$ denotes the determinant of $\mathbf{C}$. Let $\mathscr{D}_N = \{\mathbf{x}_j \in \mathbb{R}^p : j = 1,\ldots,N\}$ denote the training data set consisting of N number of p-dimensional feature vectors obtained from the structure in normal/healthy condition. The sample mean vector and sample covariance matrix are computed using the training vectors as

$$\bar{\mathbf{x}} = \sum_{j=1}^{N} \mathbf{x}_j \tag{7.4}$$

$$\mathbf{C} = \frac{1}{N-1}\sum_{j=1}^{N}\left(\mathbf{x}_j-\bar{\mathbf{x}}\right)\left(\mathbf{x}_j-\bar{\mathbf{x}}\right)^{\mathrm{T}} \tag{7.5}$$

These two measures are also referred to as *sample statistics* and can be inclusive or exclusive depending on whether or not the potential outliers are included in the training data. In the context of SHM, it is assumed that the potential outlier is not included in the training data and so the statistics are exclusive.

Evaluating the discordance measure (7.1) for the multivariate Gaussian distribution yields

$$D(\mathbf{x}) = \frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}})^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{x}-\bar{\mathbf{x}}) + c \tag{7.6}$$

where the constant $c = (p/2)\ln 2\pi + (1/2)\ln|\mathbf{C}|$, which is the same for each observation and can thus be ignored. Hence, the discordance measure for novelty detection in case of Gaussian distributed data simply becomes

$$D(\mathbf{x}) = (\mathbf{x}-\bar{\mathbf{x}})^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{x}-\bar{\mathbf{x}}) \tag{7.7}$$

which can be recognized as the Mahalanobis Squared Distance (MSD) that was introduced in previous chapter in the context of a minimum distance classifier for Gaussian distributed classes. The squared distance is based on a norm that is weighted by the inverse of the covariance matrix $\mathbf{C}^{-1}$, also known as the precision matrix. The MSD is illustrated in Figure 6.2b for a two dimensional feature space. This shows that all points having the same Mahalanobis distance from a specific point are located on an ellipse (or hyperellipses in a high-dimensional feature space).

In novelty detection, observations that have a MSD larger than a certain threshold are labeled as abnormal. The MSD is much simpler than the general discordance measure for an arbitrary probability distribution of the data, because it only requires the estimation of the mean vector and covariance matrix rather than the full multivariate probability density function. This latter is one of the most challenging tasks in machine learning, especially in the case of a high-dimensional feature space, due to the curse of dimensionality.

7.1.2 Calculation of Threshold Values

In order to detect damage using a novelty detector, it is necessary to set a threshold above which the condition of the system will be considered as abnormal. This value depends on both the number of training vectors and of the number of dimensions of the problem being analyzed. The value also depends on whether an inclusive or exclusive threshold is required. As explained in the introduction of this chapter, in the context of SHM it is assumed that the potential outlier is not included in the normal condition data and so an exclusive threshold should be established.

The threshold value is determined based on statistical analysis, by choosing a type I error level γ. This level is related to the percentage of false-positive errors (also called type I errors or false alarm) in the normal condition data. For example, a value of $\gamma = 0.01$ will correspond to a threshold level in which the top 1% of the data is considered a false-positive. In other words, it is statistically normal to have 1 out of 100 samples above the threshold. It is noted that for the discordance measure only the upper limit of the distribution is of interest.

In case that the data is Gaussian distributed, the threshold can be computed in terms of a chi-squared statistic. In fact, the MSD can be written as a sum of squares of p independent standard normal random variables, which forms a chi-squared distribution with p degrees of freedom, denoted as $\chi^2(p)$. The threshold can then be computed for an error level γ by inverting the expression $\gamma = 1-F(\xi)$, where $F(\bullet)$ is the Cumulative Density Function (CDF) of the chi-squared distribution and ξ is the threshold value. The resulting threshold corresponds to the value below which $100(1-\gamma)\%$ of the observations may be found.

Alternatively, the threshold value can be determined using a Monte Carlo simulation in which it is assumed that the data is Gaussian distributed [19, 51]. This numerical method consist of the following steps:

Step 1. Construct a data set $\mathscr{D}_N = \{\mathbf{x}_j \in \mathbb{R}^p : j = 1,\ldots,N\}$ consisting of N number of p-dimensional feature vectors, randomly generated from a standard normal distribution; Gaussian distribution with zero mean and unit standard deviation.

Step 2. Compute the sample mean vector $\bar{\mathbf{x}} \in \mathbb{R}^p$ and the sample covariance matrix $\mathbf{C} \in \mathbb{R}^{p\times p}$ from the data set $\mathscr{D}_N$, using equation (7.4) and (7.5). The potential outliers are included in the computation, yielding *inclusive* statistics.

Step 3. Compute the MSD for all the observations in $\mathscr{D}_N$, using equation (7.7), and store the largest distance value.

Step 4. Repeat **Step 1** to **Step 3** for a large number of trials, to obtain a vector that contains the largest MSD of each trial. Let N_t denote the number of trials (e.g. $N_t = 1000$) and $\mathbf{d}_{max} \in \mathbb{R}^{N_t\times 1}$ the resulting vector.

Step 5. Determine the threshold value for the $100\gamma\%$ test of discordance. This value can be obtained as follows; sort the values of the vector $\mathbf{d}_{max}$ in ascending order of magnitude, and select the distance below which $100(1-\gamma)\%$ of the trials occur (i.e. the $100(1-\gamma)$th percentile).

Note that above process yields an inclusive threshold, because the potential outliers are included in the data that was used to compute the mean vector and the covariance matrix. To compute the exclusive threshold, the exclusive mean vector and covariance matrix (exclusive statistics) must be computed for every subset of the data that is formed by excluding one observation. Then the MSD is computed for each observation using the corresponding exclusive statistics, implying that the covariance matrix must be inverted each time again. Computing the inverse of a matrix is computational expensive, which makes the determination of the exclusive threshold directly from this process not very appealing. Fortunate, the exclusive threshold is related to the inclusive threshold by means of a simple formula [19, Eq. 6.74]

$$\xi_{exc} = \frac{(N-1)(N-1)^2\xi_{inc}}{N\left(N^2-(N+1)\xi_{inc}\right)} \tag{7.8}$$

in which ξ_{inc}, ξ_{exc} are the inclusive and exclusive threshold, respectively. So, for the speed of computation, the inclusive threshold is always computed first. If an exclusive threshold is required, then this can be obtained from the simple relation above.

Extreme Value Statistics

In general, one will infer damage if the new observation is far from the mean of the distribution of the normal condition data or in a region of very low probability density. In both these cases, the implication is that the new measurement point will be somewhere in the tails of the distribution. This means that that the threshold will be sensitive to how well the distributions of the tails are understood. The methods of determining the threshold that has been discussed so far are based on assuming that the normal condition data is Gaussian distributed, which involves a severe assumption on the nature of the tails that may not be justified [19]. Therefore, more refined methods for modeling the tails of arbitrary distributions are considered. The relevant field is referred to as *extreme value statistics*, which is generally applicable to any type of distribution.

Suppose that one is given a set of samples $\{X_1, X_2, \ldots, X_n\}$ (e.g. values of the MSD) from an arbitrary parent distribution. Then the right tail of the parent distribution is described by the maximum operator, $\max\left(\{X_1, X_2, \ldots, X_n\}\right)$, which selects the maximum value from the sample vector. The theorem of extreme value statistics states that in the limit as the number samples (from an arbitrary parent distribution) tends to infinity, the induced distribution on the maximums of these samples can only take one of the following three forms: Gumbel, Weibull or Frechet distribution. Note that the distribution of the maximums is relevant for the right tail of a univariate distribution only. For the left tail, the minimum of the samples should be considered, and for this a similar theorem exists. The minimum distribution is not relevant in this thesis, because, the discordance measure used for the novelty detection has the form of a distance, where large values might indicate departures from the normal condition. Hence, the distribution for the maximums should be used for setting the upper threshold level.

The three types of limit distribution for maximums are given by [53]

$$\text{FRECHET:} \qquad F_1(x) = \begin{cases} \exp\left(-\left(\dfrac{x-\lambda}{\delta}\right)^{-\beta}\right), & \text{if } x \geq \lambda \\ 0, & \text{otherwise} \end{cases} \tag{7.9}$$

$$\text{WEIBULL:} \qquad F_2(x) = \begin{cases} 1, & \text{if } x \geq \lambda \\ \exp\left(-\left(\dfrac{\lambda-x}{\delta}\right)^{\beta}\right), & \text{otherwise} \end{cases} \tag{7.10}$$

$$\text{GUMBEL:} \qquad F_3(x) = \exp\left(-\exp\left(-\left(\frac{x-\lambda}{\delta}\right)\right)\right), \quad -\infty < x < \infty \text{ and } \delta > 0 \tag{7.11}$$

where λ and δ are the location and scale parameter, respectively, and β is a shape parameter. These parameters should be estimated from the data. Now given a vector of samples of maximum data, it is possible to select an appropriate limit distribution and fit a parametric model to the data. Once the parametric model is

obtained, it can be used to compute an effective threshold for novelty based on the true statistics of the data as opposed to statistics based the assumption of a Gaussian distribution. For example, the $100(1-\gamma)\%$ upper threshold level can be determined by inverting the relation $\gamma = 1 - F(\xi)$, where ξ is the threshold value.

7.1.3 Process of Novelty Detection

The general process of novelty detection can be summarized as follows:

Step 1. Establish a training set $\mathscr{D}_N = \{\mathbf{x}_j \in \mathbb{R}^p : j = 1,\ldots,N\}$ consisting of N number of p-dimensional feature vectors extracted from measurements. The training data should represent the normal/healthy condition of the structure.

Step 2. Estimate the probability density function $p(\mathbf{x})$ of the features using the training data.

Step 3. Determine the threshold value for the discordance measure above which the condition of the structure will be considered abnormal. This value can be computed using extreme value statistics, which is generally applicable to any type of distribution.

Step 4. For a new observation $\mathbf{y} \in \mathbb{R}^p$, evaluate the discordance measure $D(\mathbf{y}) = -\ln\big(p(\mathbf{y})\big)$ using the previously estimated probability density function. Note that a new observation is denoted $\mathbf{y}$ to distinguish it from the training vectors $\mathbf{x}$, but it must be remembered that both vectors represents the same type of features.

Step 5. Diagnose the condition of the structure based on new observations. If the discordance measure exceeds the threshold level, then the structure has departed significantly from its normal condition, which might indicate that the structure is damaged and an alarm should be issued.

As mentioned before, the estimation of the probability density function in **Step 2** is one of the most challenging tasks of all machine learning problems. To avoid this problem, it is often assumed that the training data is Gaussian distributed, which implies that the data can be characterized by its mean vector and covariance matrix. In that case, the following changes can be made to the above process:

- The estimation of the probability density function in **Step 2** reduces to the estimation of the mean vector and covariance matrix. Unbiased estimators for these statistics are given by equation (7.4) and (7.5).
- The threshold level in **Step 3** can also be computed using other methods that rely on the assumption of Gaussian distributed data, such as the chi-squared statistic and the numerical method as described in Section 7.1.2.
- The discordance measure in **Step 4** reduces to the MSD, which is given by equation (7.7).

It is clear that the Gaussian assumption leads to a simple method for damage detection. However, if this assumption cannot be justified (e.g. non-elliptical distributions), then the damage detection may be unreliable.

7.2 Application

In this section, the application of novelty detection is illustrated for the Zwartewaterbrug data. Similar to the previous chapters, the damage detection is performed separately for each bridge segment. First, the analysis of segment 2 is discussed in detail and the results of the all three segments are compared at the end of this section.

7.2.1 Feature Selection

First, suitable features are selected for the novelty detection. In contrast to previous chapter it is chosen to use the natural frequencies of a particular bridge segment as features rather than the spectral lines of the singular value spectrum. This results in a much smaller dimension of the feature space, which requires less data samples for an accurate estimation of the probability density or other statistics such as the mean vector and covariance matrix. Seven different modes were identified per segment in the range of 50–100 Hz (Tables 4.4–4.6), but these modes were not consistently identified in all measurements (indicated by the success rate in the tables). This means that the original data contains a lot of missing values. The dataset that is used for the analysis related to each one of the three segments is formed as follows:

1. Modes of the particular segment that were identified in less than 80% of the measurements are disregarded, the remaining number of modes determines the number of features.

2. The observations for which the natural frequencies of all remaining modes were identified forms the dataset. Observations with missing values are disregarded.

The resulting sets might have different number of features and observations.

It is assumed that the data from dataset four (0 kg) and five (25 kg) represent the normal condition of the bridge. There are two reasons why the 25 kg class is also considered for the normal condition data: 1) the number of training vectors is increased resulting in better estimations for the mean vector and covariance matrix, and 2) the 25 kg class is obtained under varying environmental conditions which is required for data normalization as will become clear in Chapter 8. Note that the 25 kg mass accounts for only 0.0625% of the total weight of one bridge segment, and that this mass is only located in segment 2. Hence, it is reasonable to assume that the dynamic properties of the bridge are not affected by the added mass, which justifies the choice to consider this for the normal condition of the structure.

The selected features for segment 2 are the natural frequencies of five modes as shown in Figure 7.1. The normal/healthy condition data is labeled as "H" and comprises samples of the 0 kg and 25 kg class as explained above. This data will be used for training the novelty detector, while the remaining observations are used for testing. It is noted that normally if novelty detection is performed there will be no data from the damage structure available no data is available from the damaged structure. It is assumed that this data is Gaussian distributed so that the MSD can be used as discordance measure.

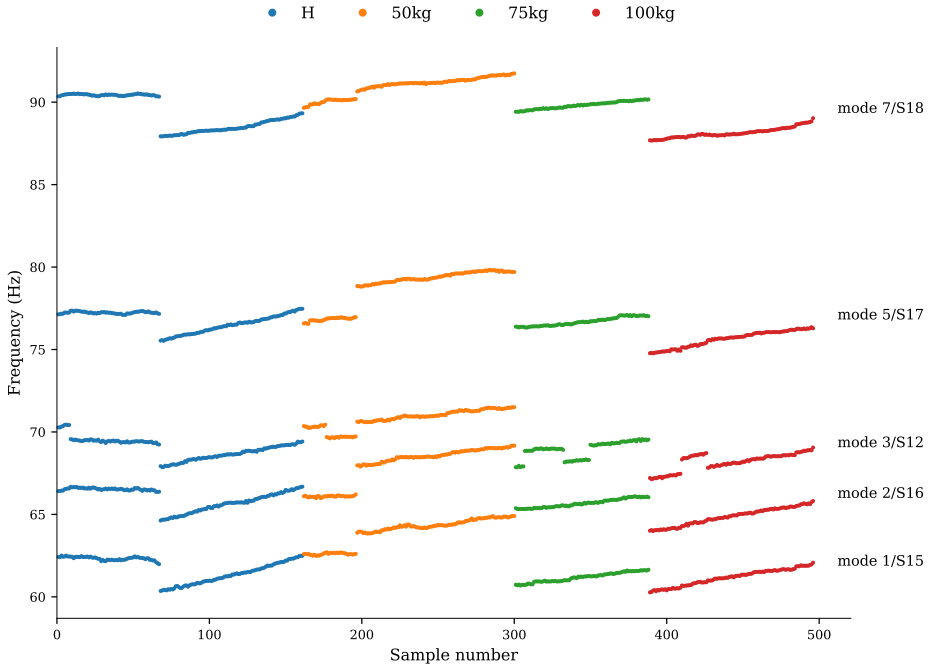


Figure 7.1: Features for damage detection, being the natural frequencies of five modes of bridge segment 2 (segment in which the added mass is located) in the range 50–100 Hz. Horizontal axis represents the different observations/samples, and the horizontal axis represents the frequency axis.

Normality test

It is assumed that the normal condition data is Gaussian distributed, so that the MSD can be used as discordance measure for the novelty detection. However, for valid interference of the results, this assumption must be validated. The assessment of normality for data with more than two variables is not so straightforward as for data with one or two variables. However, for multivariate Gaussian distributed data, the marginal distributions and any linear combination of variables should be Gaussian distributed as well. These conditions are necessary but insufficient conditions for multivariate normality [9]. So, satisfying these conditions does not guarantee that the data is multivariate Gaussian distributed, but these conditions provide a starting point for

assessing the multivariate normality. The approach that will be followed for assessing multivariate normality of the data, proceeds from univariate to bivariate to multivariate examination of the data. The advantage of this procedure over direct assessment of multivariate normality is that it results in better understanding of the underlying distribution. It is noted that if any of the variables in the data set is not univariate Gaussian, and/or any pair of variables is not bivariate Gaussian, the dataset cannot be multivariate Gaussian.

There are several techniques that can be used to check whether the data deviates from a Gaussian distribution, called *normality tests* [9, 35]. Two classes of normality tests can be distinguished: 1) graphical methods, which are methods for plotting the data and qualitatively evaluating whether the data looks Gaussian, and 2) statistical tests, which are methods that calculate statistics (e.g. kurtosis and skewness) on the data and quantify how likely it is that the data was drawn from a Gaussian distribution. One of the most popular graphical methods for testing univariate normality (and the one that will be used here) is the *probability plot* or Q–Q plot (quantile versus quantile plot) in which the observations are arranged in increasing order of magnitude and then plotted against the theoretical normal distribution values. A perfect match between the observations and the theoretical distribution is found when the samples lie on a straight line. Deviations by the samples from this line indicate a deviation from the expected distribution. The probability plot is not restricted to the normal distribution but can be used for any type of distribution. This test allows a quick and simple means of evaluating the shape of the univariate distribution for each dependent variable.

A scatter plot of all possible pairs of variables can be used in checking the bivariate normality. If the sample points do not look ellipsoidal in the two dimensional space, then the normality of the data is questionable. On the other hand, if the plot is ellipsoidal, there is no guarantee of bivariate normality since many other distributions have this shape. Another method for evaluating the normality of a dataset of two or more variables is by means of the MSD. It can be shown that if the data is drawn from a p dimensional multivariate Gaussian distribution, then the MSDs of the data has a chi-square distribution with p degrees of freedom. The Q–Q plot can be used to check the correspondence between the MSDs of the data and the chi-squared distribution. If the data is multivariate Gaussian, then the data points must lie on the line $y = x$ in the Q–Q plot.

Statistical tests may be used complementary to the graphical methods. A large range of statistical tests exist having their own strength and weaknesses [35]. In the following, D'Agostino-Pearson omnibus test is used for assessing univariate normality. This test uses the skewness (measure of the asymmetry in the distribution) and kurtosis (measure of the shape) of the data to quantify how far from Gaussian the data is in terms of asymmetry and shape.

The assessment of multivariate normality starts with examining the univariate normality and bivariate normality. Recall that the dataset consist of five features, namely the natural frequencies of the five selected modes (considering the data of segment 2). The feature vectors are standardized to normal scores as

$$\mathbf{z}_j = \mathbf{S}^{-1}\left(\mathbf{x}_j - \bar{\mathbf{x}}\right), \qquad j = 1, \ldots, N \tag{7.12}$$

where $\mathbf{x}_j \in \mathbb{R}^p$ is an observation of the original feature vector, $\bar{\mathbf{x}} \in \mathbb{R}^p$ is the sample mean vector of the features, and $\mathbf{S} \in \mathbb{R}^{p\times p}$ is a diagonal matrix whose diagonal elements are the sample standard deviations of the features. Consequently, all the individual features have zero mean and unit standard deviation, so that the expected univariate distribution is a standard normal distribution. Figure 7.2 shows the probability plots of the individual features, in which the horizontal axis represents the theoretical quantiles of a standard normal distribution (zero mean and unit standard deviation), and the vertical axis represents the ordered standardized feature values. As a reference, a straight line is fitted to the data points. The large deviations of the data points from the straight line as well as the very low p-values suggest that the individual features are not Gaussian distributed. Figure 7.3 shows the marginal distributions of the individual features and the pairs of features; the univariate distributions are shown in the form of a histogram on the "diagonal" and the bivariate distributions are shown in the form of a scatter plot in the "lower-triangle". It is noted that only the lower triangle is shown because of symmetry. As a reference, the estimated (joint) probability density function is also shown in the figures. If the data is Gaussian distributed then the histograms will have a bell shape, and the data points in the scatter plot will look ellipsoidal in the two dimensional feature space.

The third level of assessment is to examine the MSD of the data, which will have a chi-squared distribution in case the data is multivariate Gaussian. The sample mean vector $\bar{\mathbf{x}}$ and sample covariance matrix $\mathbf{C}$ are computed using all feature vectors in the normal condition set. Subsequently, the MSDs are computed using these statistics. It is noted that standardization of the features vectors does not affect the MSD, so the distances can be computed using either the original or standardized feature vectors. The MSD for all vectors in the normal condition set is shown in Figure 7.4. By comparing the histogram with the theoretical distribu-

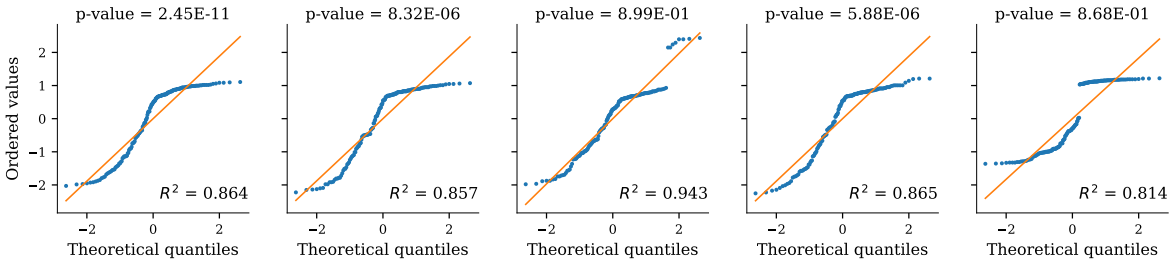


Figure 7.2: Normal probability plot of the individual features. The horizontal axis represents the theoretical quantiles of a standard normal distribution, and the vertical axis represents the ordered standardized feature values. Deviations by the data points from the straight line suggest departures from normality. The R^2 value is the coefficient of determination and indicates how well the data points can be represented by the best-fitted straight line. The p-value is computed using D'Agostino-Pearson omnibus test which is based on the skewness and kurtosis value of the data, the null hypothesis that a sample originates from a normal distribution can be rejected if the p-value is less than a chosen confidence level α, typically $\alpha = 0.05$.

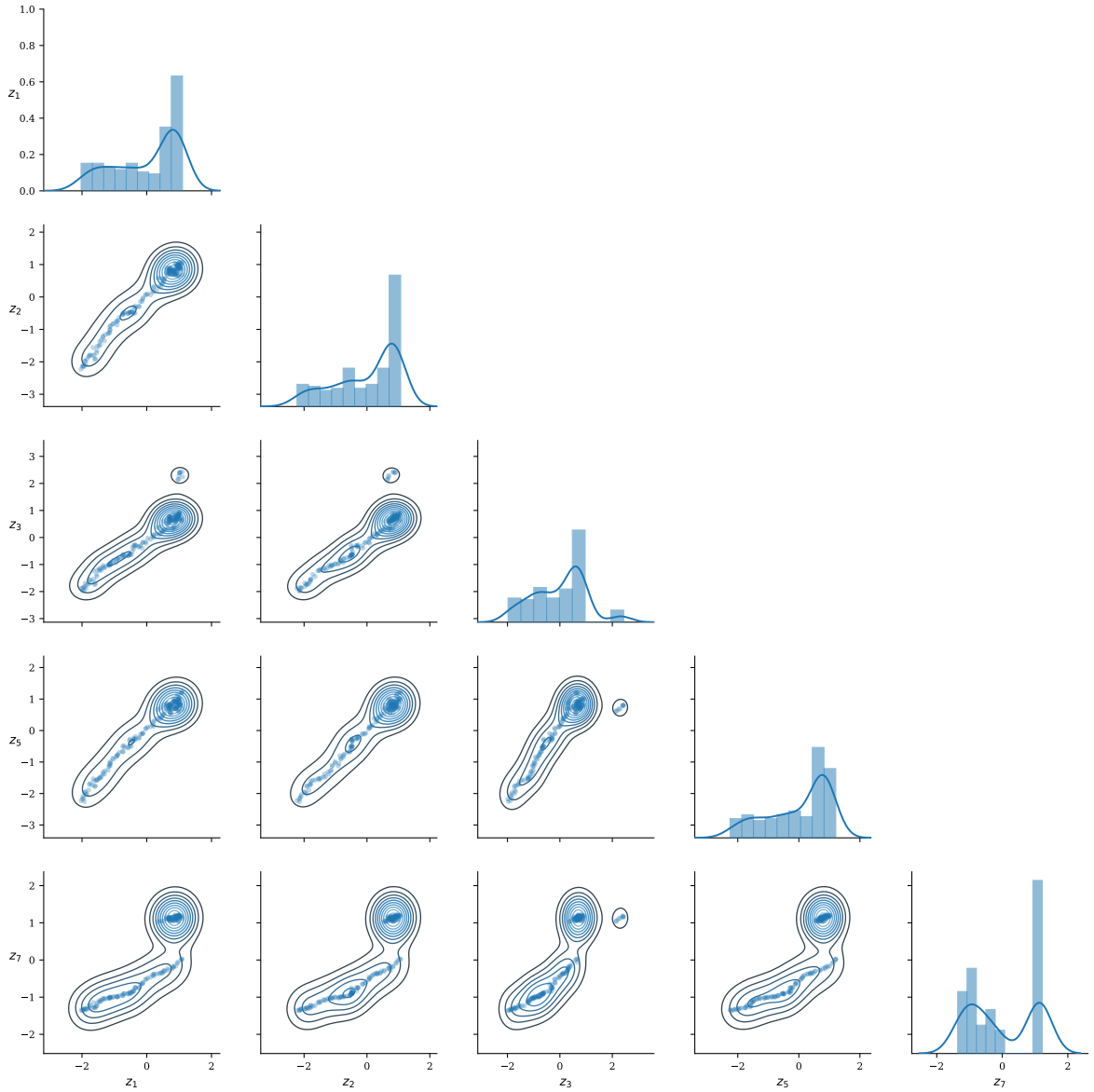


Figure 7.3: Sample distributions of the individual features and all possible pairs of features (the features are standardized to zero mean and unit standard deviation). The "diagonal" shows the univariate distributions of the features, the solid line represents an estimation the probability density function. The "lower-triangle" shows the bivariate distributions of the pairs of features, the contour lines represents an estimation of the joint probability density function. The density functions are estimated with kernel density estimation method.

tion it can be observed that the distribution of the MSD is indeed close to a chi-squared distribution, however quantification of the correspondence is not possible. The correspondence of the distributions is assessed in Figure 7.5 by means of a probability plot in which the vertical axis represents the ordered MSD values and the horizontal axis represents the theoretical quantiles of a chi-squared distribution with 5 degrees of freedom. As a reference the line $y = x$ is plotted, which represents perfect correspondence. A good correspondence between the MSD and the theoretical distribution is found for MSD smaller than ten, while large deviations of the data points from the reference line are found above this value, which suggest that these data points are not multivariate Gaussian.

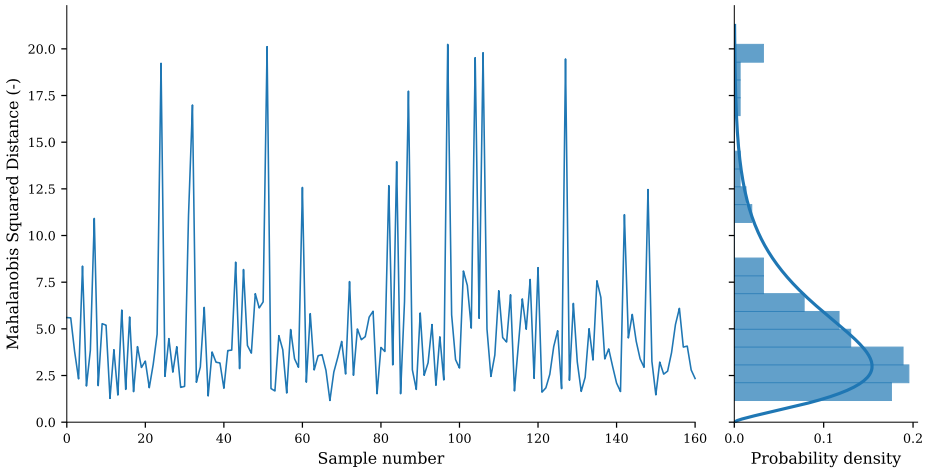


Figure 7.4: Mahalanobis squared distance of the normal condition data set. The involved covariance matrix and mean vector are computed using all samples in the considered data set. The distribution of the MSD is shown in the form of a histogram in the figure on the right, the solid line represents the theoretical chi-squared distribution with 5 degrees of freedom. The MSD of multivariate Gaussian distributed data has a chi-squared distribution, hence deviations suggest departures from normality.

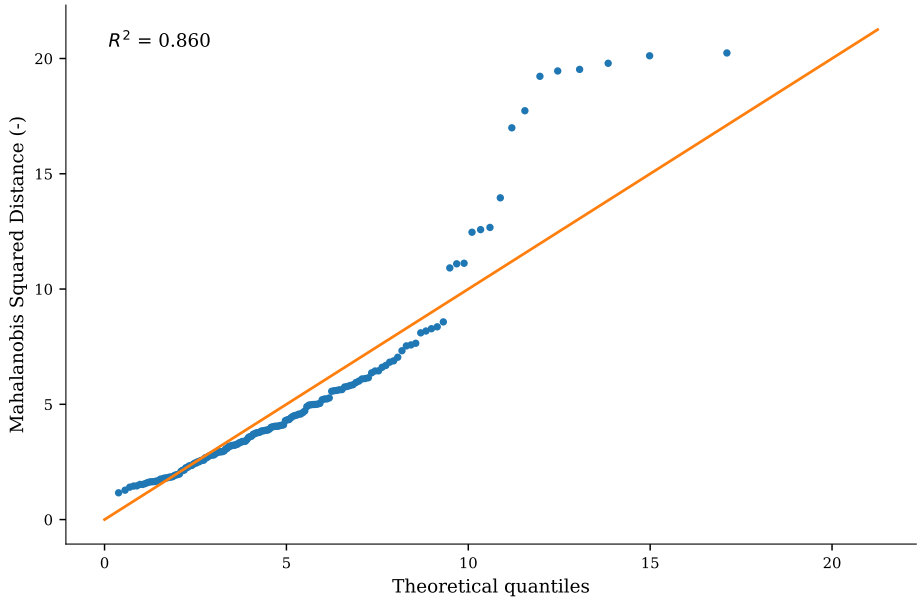


Figure 7.5: Chi-squared probability plot of the Mahalanobis squared distance of the normal condition data. Vertical axis represents the ordered MSD values and the horizontal axis represents the theoretical quantiles of a chi-squared distribution with 5 degrees of freedom. Deviations from the line $y = x$ suggest departures from the chi-squared distribution. The R^2 value (coefficient of determination) quantifies the correspondence of the data with the chi-squared distribution. An R^2 value of 1 indicates perfect correspondence.

From Figure 7.2 and Figure 7.3 it is clear that the individual features and the pairs of features are not Gaussian. Consequently, the multivariate distribution of the data will also not be Gaussian because univariate and bivariate normality of the data are necessary conditions for multivariate normality. This could have been concluded directly on the basis of Figure 7.5, since the MSD of the data does not correspond to a chi-squared distribution. The Gaussian assumption for the normal condition data is thus *not* satisfied. Nevertheless, the MSD will be used as discordance measure for the novelty detection, justified by the following two arguments:

1. Figure 7.3 shows that the the data looks elliptical in the feature space (at least in the two dimensional space), and that the probability density function has only one peak, except for the fifth feature. This suggest that the departures from a Gaussian distribution are small and that the method based on MSD may still perform well.

2. The estimation of a probability density in its full generality is a challenging task, especially given the limited number of observations (*curse of dimensionality*). Hence, the general approach will not necessarily provide more accurate results then when a parametric form of the distribution is assumed (e.g. Gaussian). The reliability of the general approach is mainly determined by the accuracy of the density estimation, which is known to be poor for small data sets.

7.2.2 Threshold Value

The threshold value is determined using the three methods introduced in Section 7.1.2; the methods are 1) based on a Monte Carlo simulation, 2) based on the chi-squared statistic, and 3) based on extreme value statistics. The first two methods assume that the data is Gaussian distributed, while the third method is applicable to any type of distribution. The type I error level is fixed to $\gamma = 0.01$ for all the methods, resulting in a threshold value for the 1% test of discordance.

For the first approach a (161×5) matrix (number of observation × number of features; same size as the normal condition set) is generated with each element being a randomly generated number from a zero mean and unit standard deviation Gaussian distribution; for all elements the MSD is then computed and the largest value stored. This is repeated a large number of times (5000 – 10000 times suggested in literature), each time storing the largest MSD in an array. The inclusive threshold ξ_{incl} can then be obtained from the array as the value below which 99% of the trials occur. The desired exclusive threshold is computed from equation (7.8). The 1% (exclusive) threshold value for the 5-dimensional problem with 161 observations was found to be 30.34 after 10000 trials.

The second method based on the chi-squared statistic is the simplest among the three methods. The MSD of the 5-dimensional feature vectors has a chi-squared distribution with five degrees of freedom (assuming that the data is Gaussian distributed). The threshold value ξ is computed by inverting the relation $F(\xi) = 1-\gamma$, where $F(\bullet)$ is the cumulative density function of the chi-squared distribution. The 1% threshold value for the 5-dimensional problem was found to be 15.09. Note that this threshold depends only on the number of features.

In the third method, the threshold value is computed based on extreme value statistics. The methodology is briefly described in Section 7.1.2. The method consist of fitting a parametric model to the right tail of the empirical cumulative density function of the MSDs computed from the training data. The only three possible parametric models for the distribution of maximums are the Frechet, Weibull and Gumbel distribution whose cumulative density functions are given in equation (7.9), (7.10) and (7.11), respectively. The distribution that describes the maximum data best is chosen for the computation of the threshold. The approach to find the best model consists of fitting the parameters of the three distributions to the data using maximum likelihood estimation and selecting the model which gives the highest likelihood. In this case the Gumbel distribution with parameters $\delta = 3.346$ and $\lambda = 6.292$ has the largest likelihood and is chosen for the computation of the threshold. The threshold value ξ is computed by inverting the expression $F_3(\xi) = 1-\gamma$, where $F_3(\bullet)$ is the CDF of the Gumbel distribution given by equation (7.11). This leads to the following threshold

$$\xi = \lambda - \delta \ln\big(-\ln\big(1-\gamma\big)\big) \tag{7.13}$$

The 1% threshold value obtained using the fitted Gumbel distribution is 21.70. This threshold is presented in Figure 7.6 together with the threshold of the second method. In this figure only the MSDs of the normal condition data is presented, which should ideally be below the thresholds. However, considering the threshold based on the chi-squared statistic there are several false-positives observed (i.e. values above the threshold),

but not for the other threshold. This implies that the threshold based on extreme value statistics is much more adequate than the one computed based on the normality assumption.

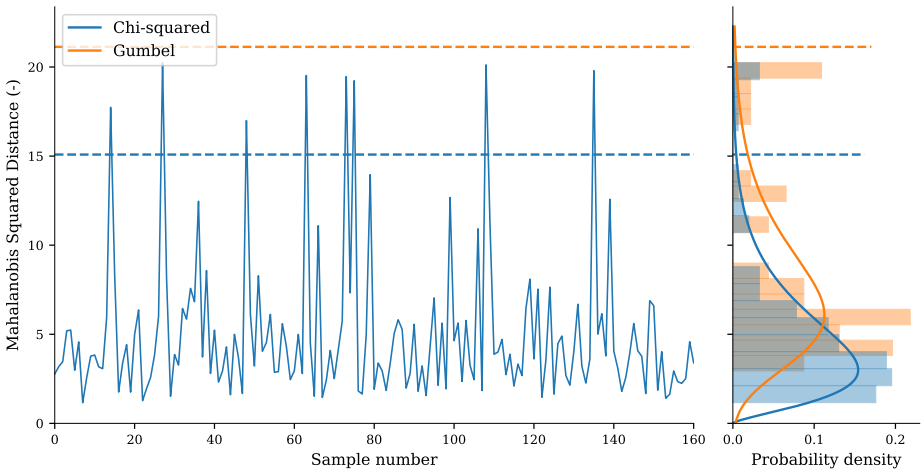


Figure 7.6: Threshold levels ($\gamma = 0.01$) based on the chi-squared distribution with five degrees of freedom and the Gumbel distribution fitted on the maximums of the MSD. The dashed lines represent the threshold levels. The histograms on the right represent the empirical probability density function of the MSD and the maximums. The density function of both the fitted Gumbel distribution and the chi-squared distribution are plotted with solid lines in the corresponding colors.

The three threshold values for the 1%, 2% and 5% test of discordance computed using the three different methods are listed in Table 7.1. This shows that the threshold values that are computed with Monte Carlo simulation (first method) are in general the largest. The values computed with the second method are in general the lowest and might lead to several false-positives.

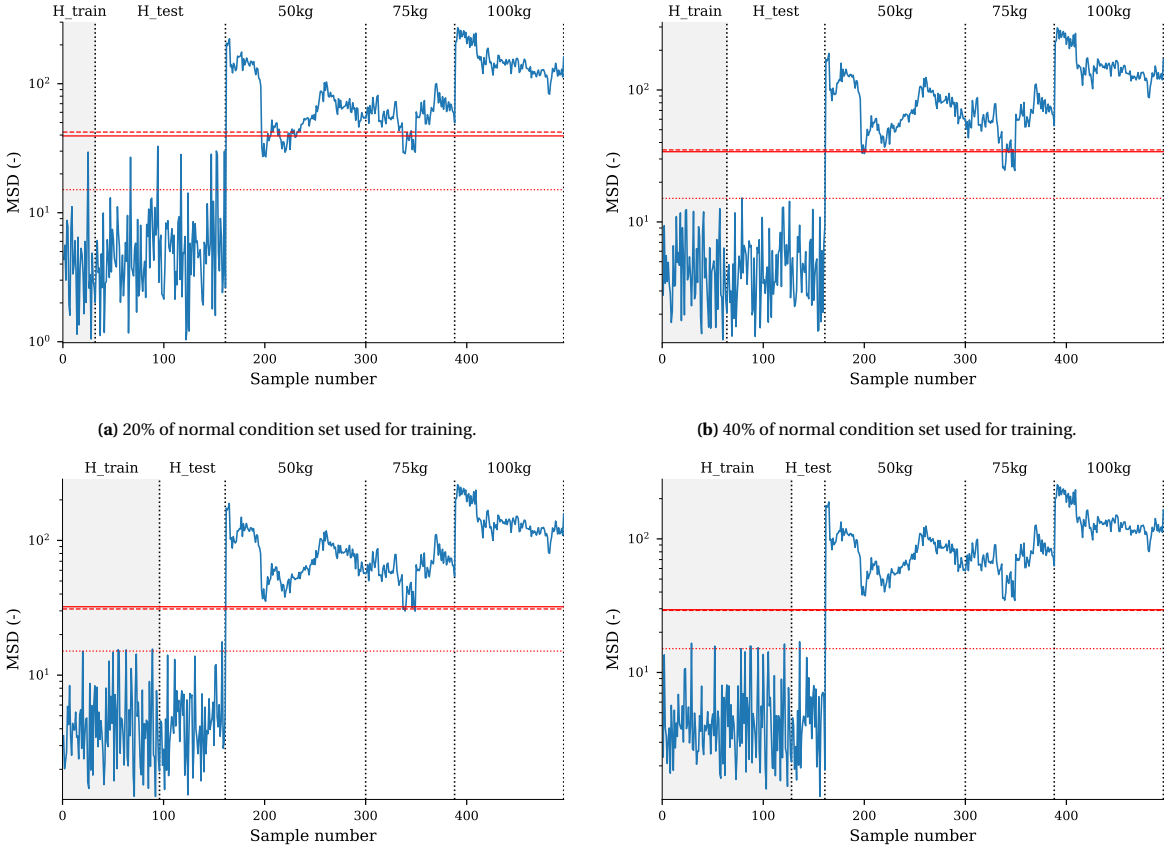
Table 7.1: Threshold values for the 1%, 2% and 5% test of discordance. The values are computed using all observations from the healthy state of the structure.

Method	Threshold level		
	1%	2%	5%
1. Monte Carlo simulation	30.34	28.36	25.72
2. Chi-squared distribution	15.09	13.39	11.07
3. Gumbel distribution	21.70	19.33	16.16

So far we only considered the data from the healthy state of the bridge (segment 2) for assessing the normality and computing threshold levels to give insight in the methodology. Now will will apply the method for damage detection as described in Section 7.1.3, where the other observations are considered as anomalies (samples 162 to 496 in Figure 7.1). To show the influence of the training set size; 20%, 40%, 60%, and 80% of the healthy data set is randomly taken to form the training set for computing the sample mean vector and sample covariance matrix. The remaining observations are used for testing the novelty detector models. The results of these four models are shown in Figure 7.7. The three threshold levels as discussed above, are plotted in the figures too. It must be noted that both the thresholds based on Monte Carlo simulation and extreme values statistics depend on the training size while the threshold based on the chi-squared is independent of the training size. From the figures is can be observed that the training size is only of minor importance, as the results are very similar for all four cases. However, including more observations from the healthy state of the structure in the training set will in general improve the performance of the model. The threshold levels indicate that it is possible to separate the abnormal observations (with added mass) from the normal ones (labeled as healthy). The threshold based on the chi-squared distribution is in general the lowest, while the other two are comparable at a higher level.

A larger threshold value is more conservative with respect to the false-positive errors (observations above the threshold in the healthy state), but at the same time there are potentially more false-negative errors (observations below the threshold for the damaged state). In the latter, no alarm is issued while the structure is

damaged (missed detection) and will usually have more severe consequences than false-positive errors (false alarm). Hence, we are looking for a threshold that minimizes the number of false-negative errors while being the number of false-positive errors at an acceptable level. To determine the most optimal threshold level, knowledge about the expected number of false-negative errors is needed for which data of the faulted structure is required. However, this is usually not possible in an unsupervised learning mode, when this type of data is not available.



(a) 20% of normal condition set used for training.

(b) 40% of normal condition set used for training.

(c) 60% of normal condition set used for training.

(d) 80% of normal condition set used for training.

Figure 7.7: Mahalanobis squared distance and threshold levels for novelty detection. The three horizontal lines represents a threshold level based on the Monte Carlo method (solid line), extreme value statistics (dashed line), and chi-squared distribution (dotted line). The size of the training data set in (a), (b), (c) and (d) is 20%, 40%, 60% and 80% of the normal condition set, respectively.

7.2.3 Comparison results per segment

In previous sections, the method of novelty detection have been applied to the data of segment 2 only. The same analysis is performed for segment 1 and 3, and in this section the results of all three segments are discussed.

Figures 7.8 and 7.9 show the selected features associated with segment 1 and 3, respectively. The approach for feature selection is the same as done for segment 2, which is described in Section 7.2.1. For segment 1, the natural frequency of six modes are selected as features and the total set consist of 582 samples of which the first 196 samples represent the healthy condition. For segment 3, the natural frequency of four modes are selected as features and the total set consist of 614 samples of which the first 192 samples represent the healthy condition.

For each segment, a novelty detection model is designed using the process described in Section 7.1.3 for Gaussian distributed data. Here, 80% of the healthy condition is used to train the model and the remaining 20% are used for testing. Training the model implies computing the mean vector and the covariance matrix of the features, which characterize the probability distribution of the features associated with the healthy state of the structure. These statistics are subsequently used to evaluate the mahalahobis squared distance. The

threshold level is computed based on the extreme value statistics as described before.

The results are shown in Figures 7.10–7.12 for segment 1, 2, and 3, respectively. The confusion matrix visualizes the performance of the model. The diagonal elements of this matrix represent the true negatives (upper left) and true positives (lower right). The off-diagonal elements represent the false-positives (lower left) and false-negatives (upper right). The overall accuracy is computed as the number of correctly labeled observations divided by the total number of observation.

The model for segment 1 and segment 3 seems to perform poorly showing a lot of false-negatives. However, it should be realized that the dynamic properties of these two segments might not be affected by the added mass because the added mass is located in segment 2 (given that the added mass is very small compared to the weight of the segments and the segments are separated by stiff transverse beams). In fact, the values below the threshold implies that the structure has not be departed from the undamaged state of the corresponding segment. The false-negatives of segment 1 and 3 might thus suggest that these segments are less sensitive to the added mass, as expected. The large values of the MSD in these segments are probably caused by other factors (e.g. variations in environmental conditions) that were not included in the training data set. It is emphasized that the current training set covers only a small range of environmental conditions and thus outliers cannot be attributed to the added mass alone.

The model of segment 2 performs well since it labels almost all observations correctly. The dynamic properties (e.g. natural frequencies) of this segment are most affected by the added mass and one might expect that the MSD increases for increasing weight, which cannot be observed for the 50 kg and 75 kg mass class. This might be because the effect of damage is compensated by the effect of other factors that were not included in the training set.

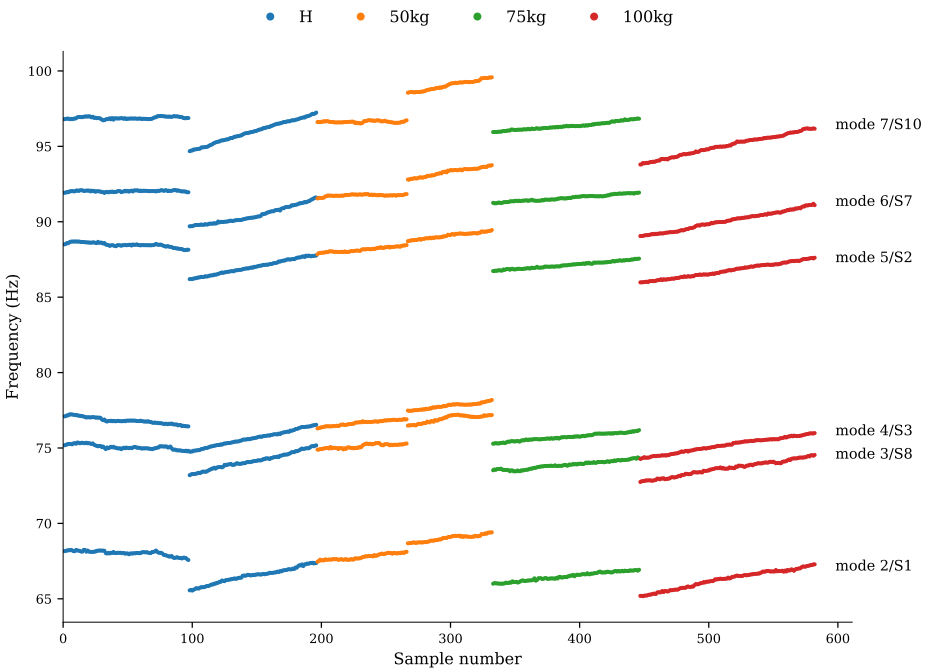


Figure 7.8: Features for damage detection, being the natural frequencies of six modes of bridge segment 1 (segment without added mass) in the range 50–100 Hz. Horizontal axis represents the different observations/samples, and the horizontal axis represents the frequency axis.

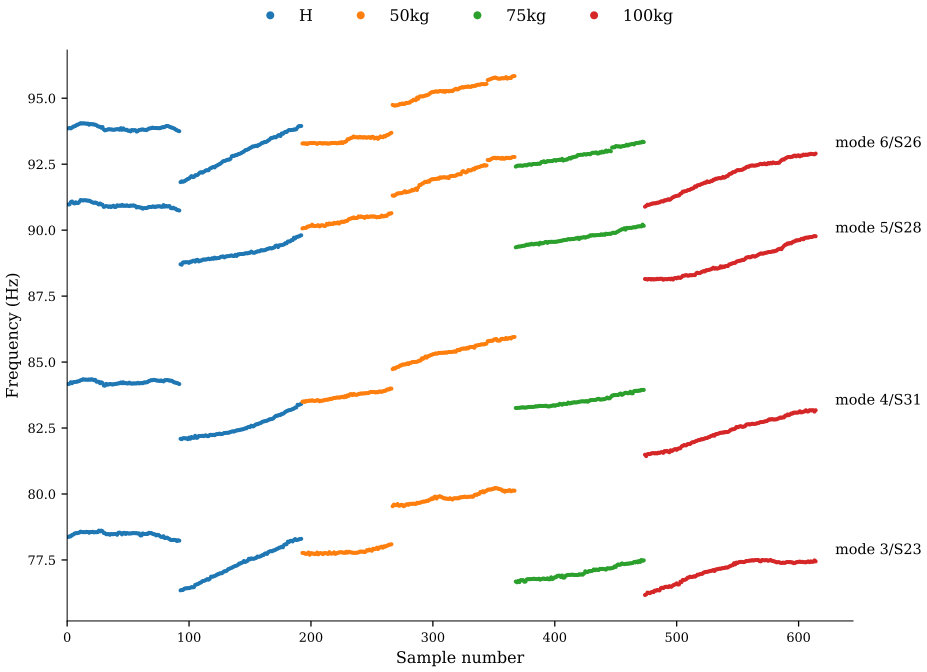
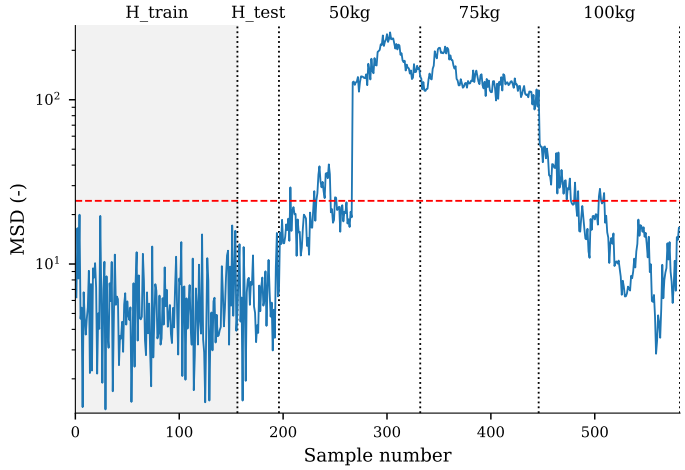
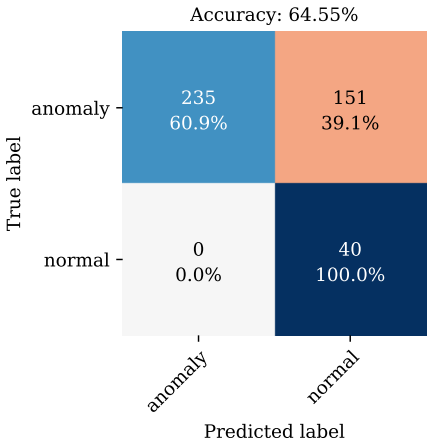


Figure 7.9: Features for damage detection, being the natural frequencies of four modes of bridge segment 3 (segment without added mass) in the range 50–100 Hz. Horizontal axis represents the different observations/samples, and the horizontal axis represents the frequency axis.

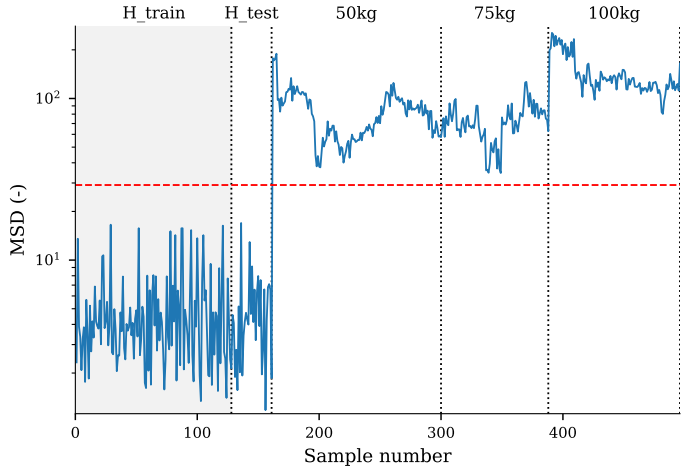


(a) Mahalanobis squared distance

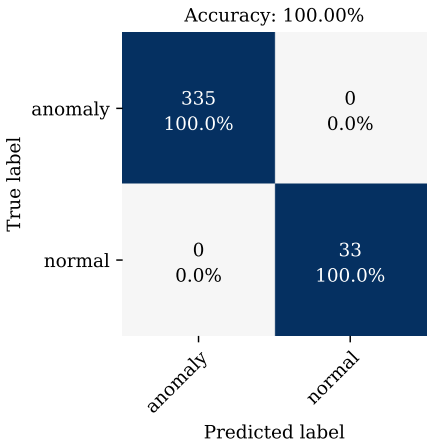


(b) Confusion matrix

Figure 7.10: Detection of anomalies in the observations of segment 1. Observations that are labeled with a weight correspond to observations obtained when the added mass was present in segment 2.

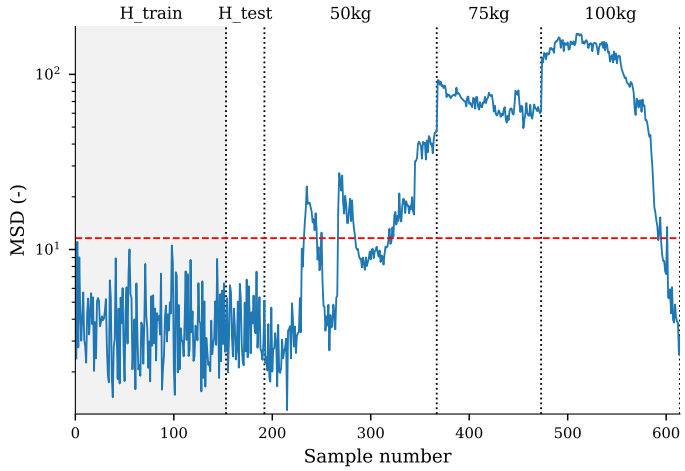


(a) Mahalanobis squared distance

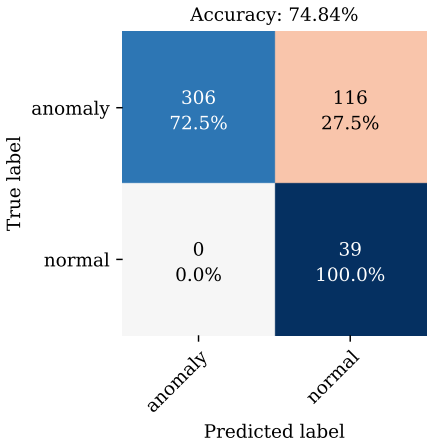


(b) Confusion matrix

Figure 7.11: Detection of anomalies in the observations of segment 2, in which the added mass is located.



(a) Mahalanobis squared distance



(b) Confusion matrix

Figure 7.12: Detection of anomalies in the observations of segment 3. Observations that are labeled with a weight correspond to observations obtained when the added mass was present in segment 2.

7.3 Conclusions

In this chapter the method of novelty detection is discussed. The aim of this method is to detect anomalies in the data, which are related to observations that are far from the probability distribution of the reference data. In the context of SHM, the probability distribution of the reference state is established using data obtained from the healthy condition of the structure. For new observations it is then checked whether it is likely that these observations come from the distribution of the reference state. A discordance measure is used to quantify the correspondence of a new observation with the reference data. For Gaussian distributed data this measure is simply the MSD. The normality of the reference data is checked. Although it appears that the data is not normally distributed, the Gaussian assumption is applied in this thesis, justified by two arguments: 1) the departures from a Gaussian distribution are small, and 2) due to the limited number of observation, the estimation of a probability density in its full generality is a challenging task. The threshold is based on three methods: the chi-squared distribution, the Monte Carlo simulation and extreme value statistics. The first two methods assume that the data is Gaussian distributed, while the third method is applicable to any type of distribution. For each one of the three segments, a novelty detection model was constructed using the natural frequencies of the particular segment as feature.

Due to the limited data available no hard conclusions can be drawn, but the following observation were made from the results:

- The threshold based on the extreme value statistics is more adequate as the one based on the chi-squared distribution. Moreover, the extreme value statistics can be applied to any type of distributions and is not restricted to Gaussian distributed data.
- All the observation are correctly labeled with a detector model that uses the natural frequencies of segment 2 as features. On the other hand, models that uses the natural frequencies of either segment 1 or 3 yield a large number of false-negatives. This suggests that the natural frequencies/ dynamic properties of segment 1 and 3 are less sensitive to the added mass located in segment 2.
- Other factor (e.g. variations in the environmental conditions) that were not included in the training set might also yield large values of the MSD, which might mask or compensate the effect of damage. This can explain the suspicious large values that were obtained from the models of segment 1 and 3.

8

Data Normalization

The basic concept underlying the use of vibration-based damage detection is that global vibration properties, notably natural frequencies, mode shapes and modal damping, are functions of the physical parameters of the structural mass, damping and stiffness. Changes in the physical parameters will cause detectable changes in the vibration properties. This sounds simple, but for in-service structures, the application of vibration-based damage detection is complicated because the vibration properties of a real structure are strongly affected by factors other than structural damages. Variations in environmental conditions (e.g. temperature, humidity, wind) and operational conditions (e.g. traffic loads) also cause changes in dynamic properties which may mask the changes caused by structural damage.

The employed damage detection methods in previous chapters were based on changes in the dynamic response of the system. In fact, it is assumed that damage will alter the stiffness of a system (could also be mass, or energy dissipation properties), which in turn will alter the measured dynamic response of the system. However, variations in the environmental and operational conditions of the structure will often mask subtler structural changes caused by damage. The damage-sensitive features used in the damage detection techniques often appear to be sensitive to these changes in environmental and operational conditions of the structure. This was already observed from the results of the modal analyses, in some datasets the identified natural frequencies show large variations, which cannot be allocated to the added mass, but must be caused by other sources like environmental and operational conditions. In order to design a reliable damage detection model it is necessary to account for the changes in environmental and operational conditions, this is often referred to as *data normalization*. Data normalization is a procedure to normalize data sets, so that signal changes caused by operational and environmental variations of the system can be distinguished from structural changes of interest, such as structural damage.

8.1 Overview of Studies on Environmental and Operational Variability

In this section, a brief summary about the Environmental and Operational Variability (EOV) is provided. The focus will mainly be on the influence of environmental variability. It must be noted that the similarity filtering method was intended to remove the effects due to variations in operational conditions, but not in the environmental conditions. The operational conditions include ambient loading conditions, operational speed and mass loading.

Mass loading effects

The influence of traffic loading on modal parameters of bridge structures has been investigated by Kim et al. and Zhang et al. [28, 58]. In both studies it was found that the mass loading effect of moving vehicles varies depending on the mass of the vehicles relative to the mass of the considered section of the bridge. Kim et al. [28] reported that the natural frequencies of a 46 meter long simply supported bridge decreased by 5.4% as a result of heavy traffic (mass ratio of heavy vehicles to the super structure is about 3.8%). On the other hand, for middle-and long span bridges, changes in the measured natural frequencies due to different types of vehicle masses (mass ratio for heavy vehicles is about 0.38%) were hardly detectable. Zhang et al. [58] found for a cable stayed bridge that the natural frequencies the global modes, up to 2 Hz can exhibit as much as

1% change within a day. In these studies it was assumed that changes due to environmental variations were limited but in general the structures are exposed to the combined action of environmental conditions and operational conditions.

Temperature effects

Most studies about environmental conditions in the context of vibration-based damage detection investigate the effects of temperature variability on the measured dynamic response of a structure.

Wood [50] reported that changes of bridge responses were closely related to the structural temperature based on the vibration test of five bridges in the UK. Analyses based on the data compiled suggested that the variability of the asphalt elastic modulus due to temperature effects was a major contributor to the changes in structural stiffness.

Based on vibration tests conducted on four different footbridges, Weber [49] found a large variation of dynamic properties with the asphalt temperature. The bridges were also modeled numerically using a visco-elastic material for asphalt. The experiments and models show that both the natural frequencies and the damping ratios can change significantly with temperature. In one particular case (timber bridge), an increase in natural frequency of 25% has been when asphalt temperature decreased. In general, the natural frequencies (of the first three modes) decrease with increasing temperature. For high temperatures, the asphalt is weak and gives practically no contribution to the stiffness. For colder temperatures, the asphalt stiffness increases and the fundamental frequencies increase. However, not all the bridges were affected the same way. This is mostly explained by considering the cross-sections of the bridges. The further away the pavement is from the neutral axis, the higher is the influence. The author also concluded that other effect may play a role as well: 1) if the pavement is not fully bonded to the beam the interface stiffness may reduce the effect, 2) if the dynamic behavior is not entirely governed by bending, the effect of asphalt is reduced, and 3) short bridges might vibrate more like a plate than like a beam, in which case the cross section might not fully describe the dynamic behavior.

Researchers from Los Alamos National Laboratory performed several tests on the Alamos Canyon bridge located in southern New Mexico [11, 21, 22]. The results revealed that the first three natural frequencies of the bridge varied about 4.7%, 6.6%, and 5.0% during a 24 h period as the temperature of the bridge deck changed by approximately 22°C. The temperature gradient largely influences the variation of the fundamental frequency. They have also studied the I-40 bridge over the Rio Grande which was artificially damaged by making various torch cuts in the web and flange of one of the two plate girders, which support the concrete deck [20]. It was found that no change in dynamic properties can be observed until the final level of damage is introduced (10 mm wide cut through half the height of the plate girder, including the bottom flange). At the final level of damage the natural frequencies for the first two modes have dropped to values 7.6 and 4.4% less, respectively, than those measured during the undamaged tests.

Peeters et al. [37, 38] conducted a study devoted to monitoring the effect of changing environmental conditions on structural vibration properties on the Z24 Bridge in Switzerland. It was continuously monitored for nearly a year. It was reported that the relative variation of the first four natural frequencies varied between 14% and 18% during the 10 months. It was also found that the natural frequencies of all the modes analyzed, except the second mode, decreased when the temperature increase. The natural frequency of the second mode increased with increasing temperature (for positive temperatures). In particular, a bilinear relationship was observed between the measured frequencies and temperature. There was a clear linear relationship above the freezing temperature (0°C) and a different linear correlation below the freezing temperature. It was demonstrated that this bilinear behavior could be attributed to the asphalt layer on the deck. Although the asphalt layer did not contribute to the overall stiffness at warm temperatures, it added significant stiffness to the bridge at cold temperatures. This was also mentioned by Weber in his work [49].

Ni et al. [34] addresses the modeling of temperature effects on the modal frequencies for the cable-stayed Ting Kau Bridge (Hong Kong), which has been instrumented with a long-term monitoring system. The bridge comprises two main spans of 448 m and 475 m respectively, and two side spans of 127 m each. Based on one-year measurement data obtained from 45 accelerometers and 83 temperature sensors, natural frequencies and mode shapes of the first eight modes, in the range from 0.1 to 0.4 Hz, and temperatures at different locations of the bridge are obtained at one-hour intervals. The one-hour average temperature from 20 temperature sensors ranges from 2.83 to 53.46 degree Celsius. It was observed that environmental and operational variations account for relative variations in the modal frequencies between 1.505% and 6.689% for the first eight modes [60], which may mask the modal changes caused by structural damage. For all the measured modes, an overall decrease in modal frequency is observed with increase in temperature of the bridge. The

variation in mode shapes does not depend on the environmental effects.

Ding and Li [16] investigated the daily and seasonal correlations modal frequency-temperature using 215 days of health monitoring data obtained on the Runyang Suspension Bridge. The bridge is a single-span steel suspension bridge with main span of 1490 m. The aerodynamically shaped closed box steel girder is 36.3 m wide and 3.0 m high. A total of 27 uni-axial accelerometers were installed on nine sections of the bridge deck to measure the dynamic response, and a total of 40 temperature sensors were installed on five sections which effectively measure the global temperature of the steel girder. The natural frequencies of the first six modes, in the range of 0.1 to 0.5 Hz, were identified at 10 minutes intervals. In order to eliminate the variations rising from the identification algorithm, the daily averaged modal frequencies were computed. It was shown that the daily averaged modal frequencies have remarkable seasonal correlation with the daily averaged temperature. For all six modes, an overall decrease in modal frequency was observed with the increase of average temperature of the bridge, and the measured modal frequencies of higher modes are more sensitive to ambient temperatures. Average temperature ranges from -2 to 38 degrees Celsius and relative variation of daily averaged modal frequencies ranges from 0.324% to 2.132%.

As the temperature of an entire structure is usually nonuniformly distributed, Xia et al. [54] states that using the air temperature and/or surface temperatures alone may not sufficiently capture the relation between the vibration properties and temperatures. Based on FEM analyses and experiments with a simply supported reinforced concrete slab, he concluded that the consideration of the temperature distribution of the whole structure (especially over the height of the RC slab) will lead to more accurate results of the temperature effect on the vibration properties of the structure. It was found that the frequency of the structure decreases when temperature goes up.

Geng et al. [23] performed experiments to investigate the vibration and acoustic response characteristic of a clamped rectangular aluminum plate (0.4 x 0.3 x 0.003 m^3) in thermal environment. Results show that the natural frequencies of the plate decrease when temperature is elevated. Thermal stresses were regarded as the major reason for the reduction of the natural frequencies of the heated clamped plate. This thermally induced softening effect has unequal influence on the structural stiffness along the two in-plane directions. Besides, it was observed that the initial deflection of the plate has a large influence on the natural vibration of the heated plate, because it reduces the thermally induced softening effect.

Boundary condition effects

Temperature variations not only affect the material properties (in most studies the modulus of elasticity was found to be most affected), but could also change the boundary conditions. Freezing of the bridge supports was supposed to be one of the main driving forces behind the observed piecewise linear relationship between ambient temperature and the fundamental frequencies of the Z24 bridge that was found by Peeters and Roeck [37] based on one-year monitoring data. As already mentioned, the bilinear behavior could also be attributed to the stiffness of the asphalt layer on the deck. It is likely that both effects happen when the temperature fall below the freezing point. Other researchers also show that freezing of the bridge supports significantly affects the natural frequencies. In general, the natural frequencies increase due to the increased overall stiffness of the structure.

Theoretical derivation

In the above literature, most studies show that the modal frequencies decreases when the temperature rises. The magnitude of frequency change varies between different studies and depends on the type of structure, the materials and the temperature range. Next it will be shown that the observations can partially be explained by Euler-Bernoulli beam theory and plate theory. Consider a simply supported single-span prismatic beam made of an isotropic material. It is assumed that variation in the temperature will not affect the mass and boundary conditions, but only the geometry of the structure and the mechanical properties of the material. For this beam, the undamped natural frequency of mode n are given by [10]

$$\omega_n = \frac{n^2\pi^2}{L^2}\sqrt{\frac{EI}{\mu}} \tag{8.1}$$

in which L is the length of the beam, $\mu = \rho A$ is the mass per unit length, E is the modulus of elasticity, and I is the moment of inertia of the cross-sectional area. The dimensional rate of natural frequency change can be

expressed in terms of an increment of the involved parameters [55]

$$\frac{\delta\omega_n}{\omega_n} = -2\frac{\delta L}{L} + \frac{1}{2}\frac{\delta E}{E} + \frac{1}{2}\frac{\delta I}{I} - \frac{1}{2}\frac{\delta\mu}{\mu} \tag{8.2}$$

Lets denote the thermal coefficient of linear expansion of the material as α_T, and the thermal coefficient of modulus as α_E. Assuming that the variations in modulus of elasticity with temperature are linear (for small changes in temperature), the variations in moment of area are four times the variation in linear expansion, and the mass per unit length is inversely proportional to the length as the total mass is assumed constant, one obtains

$$\frac{\delta L}{L} = \alpha_T \delta T, \qquad \frac{\delta E}{E} = \alpha_E \delta T, \qquad \frac{\delta I}{I} = 4\alpha_T \delta T, \qquad \frac{\delta\mu}{\mu} = -\alpha_T \delta T \tag{8.3}$$

Consequently, equation (8.2) can be simplified to

$$\frac{\delta\omega_n}{\omega_n} = \frac{1}{2}(\alpha_T + \alpha_E) \tag{8.4}$$

which estimates the dimensionless rate of the frequency change with the temperature change. For most engineering materials α_E is much larger than α_T (in absolute sense), which implies that the variation in natural frequency subjected to a temperature change are governed by α_E. For example, the linear coefficients (α_T) of steel, aluminum and concrete are 1.1e−5, 2.3e−5 and 1.0e−5/°C, respectively. The thermal coefficients of modulus (α_E) of the three materials are −3.6e−4, −5.6e−4 and −3.0e−3/°C, respectively [55]. Hence, the theoretical variation of the natural frequencies of steel beams is 0.018% per degree Celsius (positive for temperature decrease and negative for temperature increase).

In addition, temperature changes may induce thermal axial forces in the structure, which alters the natural frequencies by changing the longitudinal bending stiffness [55]. These forces arise, for example, when the thermal expansion/contraction of the structure is constrained or when changes in the support bearings occur. The natural frequencies of a simply supported single-span prismatic beam under constant axial force are [10]

$$\bar{\omega}_n = \omega_n\sqrt{1 + \frac{N}{n^2 N_{cr}}} \tag{8.5}$$

in which ω_n is the natural frequency of the beam without axial force given by equation (8.1), N represents the constant axial force (positive is tension, and negative is compression), $N_{cr} = \pi^2 EI/L^2$ is the critical buckling load for a pin ended column with length L and flexural rigidity EI. For instance, the axial force that arises due to a uniform temperature increase ΔT is equal to $N = -EA\alpha_T\Delta T$, where α_T is the thermal coefficient of linear expansion and EA is the axial stiffness of the beam. This shows that the natural frequency increases when the beam is tensioned, and decreases when the beam is compressed. The latter is due to the softening of the beam in compression. It also shows that the higher order modes are less sensitive to the (thermally induced) axial forces. Previously it was assumed that the temperature change is constant and uniform across the beam. In reality this is rarely the case, for instance, when some parts of a structure are more exposed to heating than others (e.g. the top surface of a bridge deck will often experience higher temperatures than the bottom surface, due to the solar radiation on the top surface). Moreover, temperature variations inside a structure lag behind those on the surface because of temperature inertia, especially for concrete structures.

Similar derivation can be done for plate structures. Consider an isotropic rectangular thin plate simply supported on each edge. The plate is lying in the plane $z = 0$ with dimensions a in the x-direction an b in the y-direction and t denotes the thickness of the plate. It is assumed that the plate is stress-free at the reference temperature T_0. Subjected to the variation of temperature (uniform across the thickness of the plate), it is assumed that static membrane forces will be generated in the plate

$$N_x = N_y = -\frac{Et\alpha_T\Delta T}{1-\nu}, \qquad N_{xy} = 0 \tag{8.6}$$

In which α_T is the thermal expansion coefficient, ΔT is the temperature change with positive values denoting a temperature increase (i.e. increase of temperature results in compression membrane forces). The natural frequencies of a simply supported plate in thermal environment are given by [23]

$$\omega_{mn} = \sqrt{\frac{\pi^4 D}{\rho t}\left(\frac{m^2}{a^2} + \frac{n^2}{b^2}\right)^2 - \frac{\pi^2 E\alpha_T\Delta T}{\rho(1-\nu)}\left(\frac{m^2}{a^2} + \frac{n^2}{b^2}\right)} \tag{8.7}$$

in which $D = Et^3/12(1-\nu^2)$ is the flexural rigidity of the plate, E is the modulus of elasticity, ν is the Poisson's ratio, and ρ is the mass density. The corresponding mode shape of mode (m,n) is

$$\phi_{mn}(x,y) = \sin\frac{m\pi x}{a}\sin\frac{n\pi y}{b} \tag{8.8}$$

From equation (8.7) it is clear that there are two factors affecting the natural frequencies of the plate, namely, variations in material properties and thermally induced forces. When a constrained plate is heated, the thermal expansion of the plate will be limited by the boundaries. Then, compression forces are generated in the plate, which reduces the natural frequencies. At the same time, for most structural materials, the modulus of elasticity will decrease as the temperature rises. Both these two variations will reduce the natural frequency of the plate. When the plate is cooled, the natural frequencies will increase. It is noted again that a uniform temperature variation has been assumed. When the temperature at the top face of the plate is higher than at the bottom side, then the top face expands more than the bottom, and moments are generated in addition to membrane forces.

8.2 Temperature Dependency of Natural Frequencies

In this section we will analyze the temperature dependency of the natural frequencies of the Zwartewaterbrug, that were identified in section Chapter 4. In particular, the natural frequencies of the four most dominant modes for each of the three bridge segments are considered (12 in total). These frequencies are summarized in Tables 8.1–8.3. Recall that the notation of the modes is X/SY, where X indicates the mode number and Y refers to the sensor number where the amplitude of the local mode shape is unity. The relative frequency differences presented in the tables is the frequency difference normalized by the mean value and computed as

$$\Delta f(\%) = \frac{f_{max} - f_{min}}{\bar{f}} \cdot 100\% \tag{8.9}$$

where f_{max}, f_{min}, and $\bar{f}$ are the maximum, minimum, and mean frequency, respectively.

The relative frequency difference considering all data, ranges between 6.76% and 12.21%. The smallest relative frequency difference can be found in segment three and the largest can be found in segment two (in particular found by modes in segment two with a local mode shape close to the added mass). When the mass classes are taken into account, the 0 kg mass class showed for all modes the highest relative frequencies differences, ranging from 4.72% to 8.47% (note that the data of the 0 kg mass class is measured once in the morning and once in the afternoon with 7 hours between them). The range of the 0 kg mass class have to be explained by normal environmental changes.

Based on the studies from previous literature, it is suspected that the variations in the natural frequencies are the result of changing environmental conditions, like the temperature. The ambient temperature measured during the tests on the Zwartewaterbrug were previously shown in Chapter 2, Figures 2.10–2.12 and are summarized in Tables 8.1–8.3. Note that the temperature is measured at the data acquisition system (location of eRIO in Figure 2.3), and applies to all segments.

Figures 8.1, 8.3 and 8.5 show the variation of temperature and natural frequency of the four most dominant modes of each of the three bridge segment. The sample numbers on the horizontal axis are sorted in chronological order and includes all measurements (for the samples per dataset, see Table 4.3). In most cases, the datasets can easily be distinguished based on the “jumps" in the plots. Although the mass classes are not distinguished in the figures, some might argue that the added mass can affect the natural frequency. However, the natural frequencies of the outer segments without added mass (segment 1 and 3) are hardly affected by the small mass in the center segment (segment 2), because the weight of the added mass is relatively small compared to the weight of the segments, and the segments are separated by stiff transverse I-beams. So, any deviations in the natural frequencies of segment one or segment three are almost certainly caused by other factors like temperature. The added mass will influence the modes with a local mode shape that is close to the added mass the most, but also for segment two other factors (like temperature) will have more effect. From the figures it is readily seen that the natural frequency follows the temperature pattern. The small deviations imply that the relation between the temperature and frequency is not perfectly linear.

The correlation between the natural frequencies and the ambient temperature is analyzed in more detail, considering 1) the data of the five mass classes separately to exclude the potential influence of the mass on the variation of the natural frequency, and 2) all the available data (i.e. data of all mass classes combined) to

analyze the overall relation. For the latter, the temperature is plotted versus the natural frequencies in Figures 8.2, 8.4 and 8.6. These figures show that all the frequencies increase with increasing temperature, and that there is a linear relation between the temperature and frequencies. The temperature-frequency relation is discussed in more depth after the introduction of the correlation coefficient and the linear regression model.

The strength of a linear relationship between two variables x and y is measured by the (Pearson product-moment) correlation coefficient, ρ_{xy}, which is defined as

$$\rho_{xy} = \frac{C_{xy}}{\sigma_x \sigma_y} \tag{8.10}$$

where C_{xy} is the sample covariance

$$C_{xy} = \frac{1}{N-1} \sum_{k=1}^{N} (x_k - \bar{x})\left(y_k - \bar{y}\right), \qquad \bar{x} = \frac{1}{N} \sum_{k=1}^{N} x_k \tag{8.11}$$

and $\sigma_x = \sqrt{C_{xx}}$, $\sigma_y = \sqrt{C_{yy}}$ are the sample standard deviations, $\bar{x}$ is the sample mean value, and N is the number of observations. The correlation coefficient has a value between -1 and +1, where -1 indicates a perfect negative correlation, 0 means no linear correlation, and +1 indicates a perfect positive linear correlation. This metric is used as a measure for the goodness of fit of the linear model introduced next.

A linear regression model is fitted to the temperature–frequency data in order to examine the suggested linear relationship between the temperature and a natural frequency. The relationship between the temperature input T_k at time instant k, and the natural frequency can be expressed as

$$f_k = w_1 T_k + w_0 + \varepsilon_k \tag{8.12}$$

where w_0 (intercept) and w_1 (slope) are the regression coefficients, and ε_k is the error term. The coefficient w_1 indicates the frequency change for a unit temperature increase, and w_1 / w_0 gives the percentage of the frequency change for a unit temperature increase with respect to the intercept w_0. Suppose that N observations are available, then equation (8.12) can be written in matrix notation as

$$\mathbf{f} = \mathbf{X}\mathbf{w} + \boldsymbol{\varepsilon} \tag{8.13}$$

where $\mathbf{w} = (w_0, w_1)^{\mathrm{T}}$, and with N observations

$$\mathbf{f} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{pmatrix}, \qquad \mathbf{X} = \begin{bmatrix} 1 & T_1 \\ 1 & T_2 \\ \vdots & \vdots \\ 1 & T_N \end{bmatrix}, \qquad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_N \end{pmatrix}$$

The method of least squares (LS) is used to estimate the coefficients of the linear model. In this method the coefficients are determined such that the sum of the squares of the errors is minimized

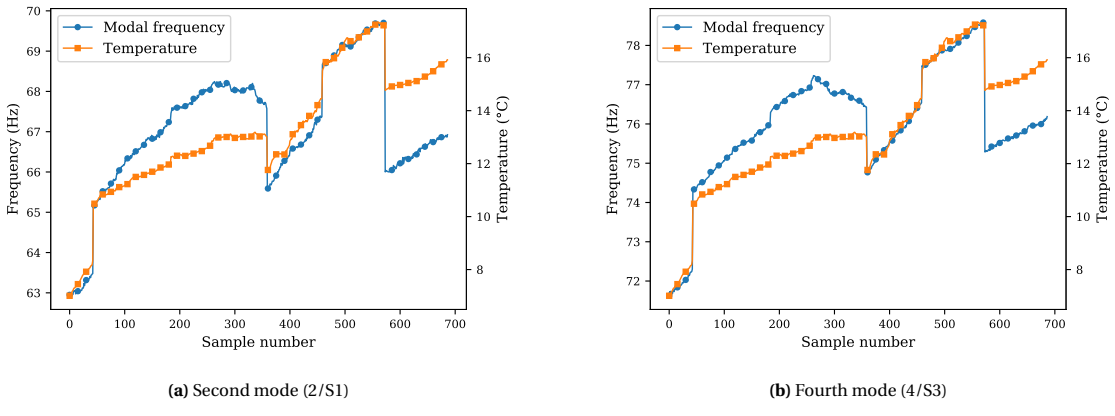
$$\min_{\mathbf{w}} \sum_{k=1}^{N} \varepsilon_k^2 \equiv \min_{\mathbf{w}} \|\mathbf{f} - \mathbf{X}\mathbf{w}\|^2 \tag{8.14}$$

Solving this quadratic minimization problem with respect to $\mathbf{w}$ yields

$$\hat{\mathbf{w}} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1} \mathbf{X}^{\mathrm{T}}\mathbf{f} \tag{8.15}$$

where $\hat{\mathbf{w}}$ denote the *estimated* coefficients. Equation (8.15) is used to compute the intercept and slope of the linear line for a given set of input-output pairs.

Multiple linear regression models are fitted to the temperature-frequency data, each corresponding to one of the twelve modes. Moreover, linear regression models are fitted to the data of the five mass classes separately, and one to all the available data (i.e. data of all mass classes combined), resulting in a total of 72 models. The regression coefficients of these models and corresponding correlation coefficients are listed in Tables 8.1–8.3.



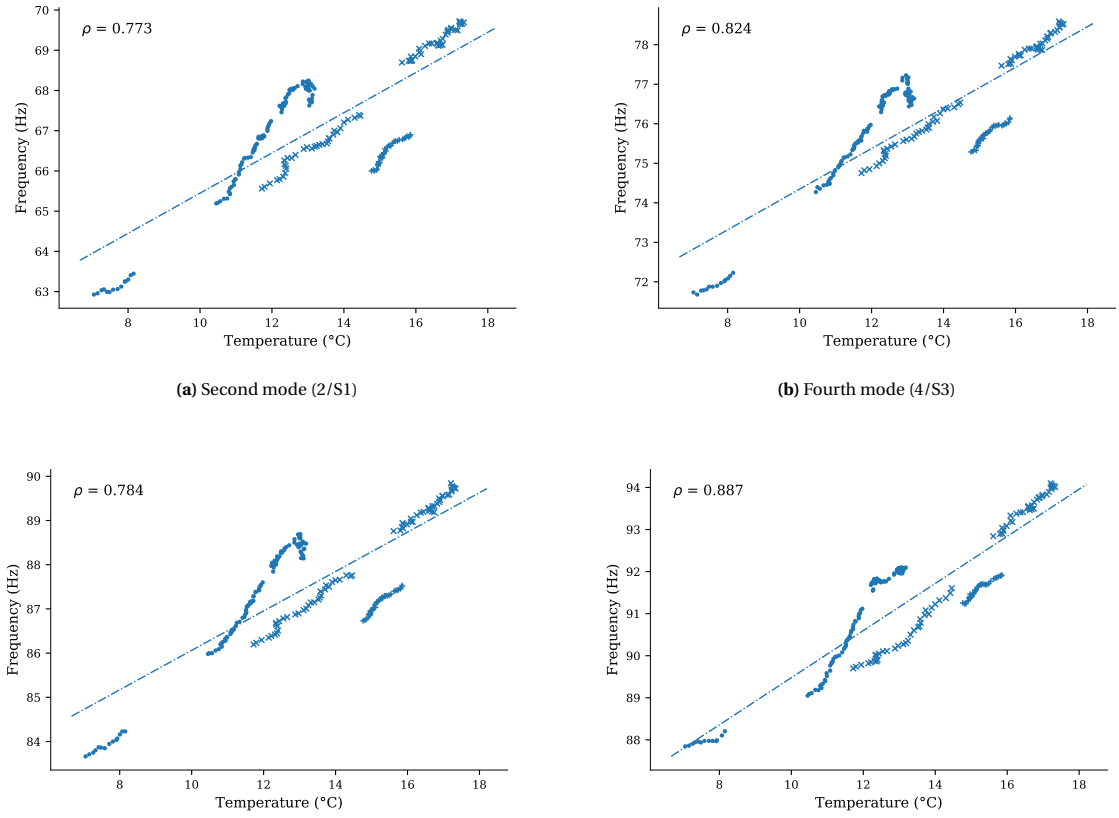
(a) Second mode (2/S1)

(b) Fourth mode (4/S3)

(c) Fifth mode (5/S2)

(d) Sixth mode (6/S7)

Figure 8.1: Segment 1 – Variation of temperature and natural frequency of the four most dominant modes in segment 1. The samples on the horizontal axis are sorted in chronological order.



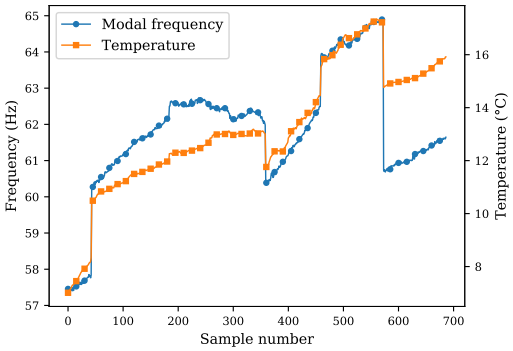
(a) Second mode (2/S1)

(b) Fourth mode (4/S3)

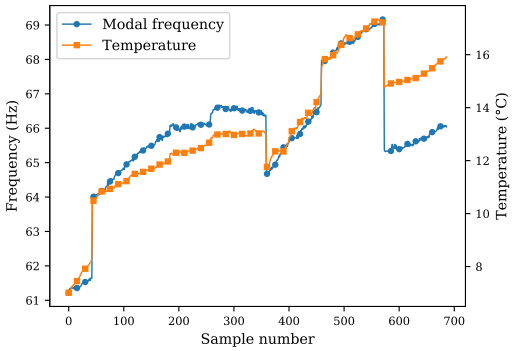
(c) Fifth mode (5/S2)

(d) Sixth mode (6/S7)

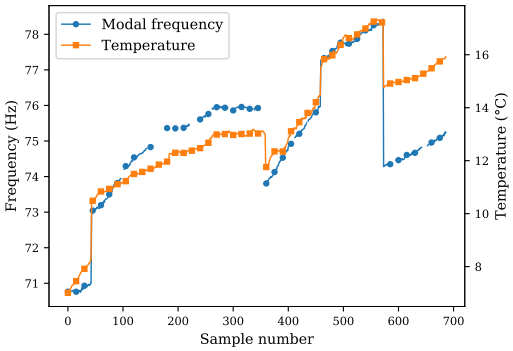
Figure 8.2: Segment 1 – Correlation between natural frequency and temperature for the four most dominant modes in segment 1. The suggested linear relationship among the variables is emphasized with a linear regression fit (dashed line). Markers: (.) 23 March 2017, (x) 28 March 2017, (+) 31 March 2017.



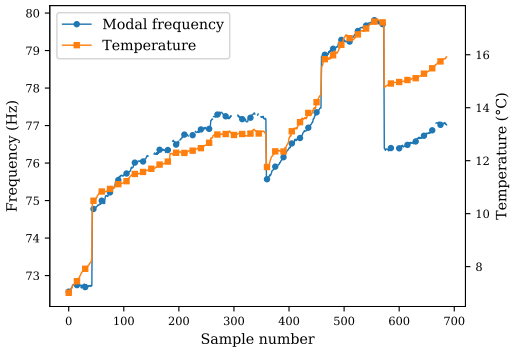
(a) First mode (1/S15)



(b) Second mode (2/S16)

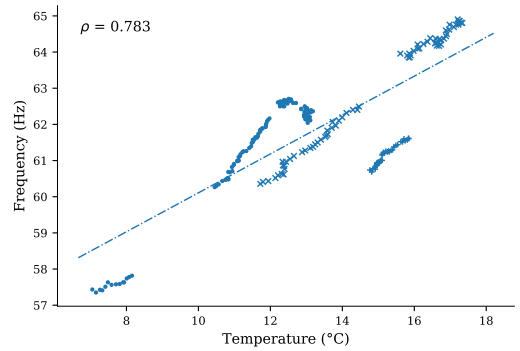


(c) Fourth mode (4/S13)

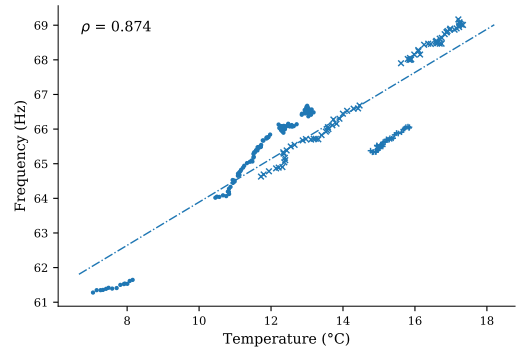


(d) Fifth mode (5/S17)

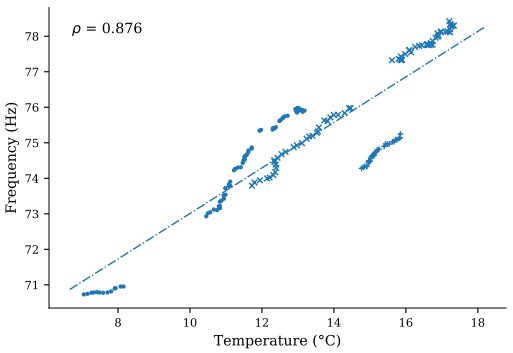
Figure 8.3: Segment 2 – Variation of temperature and natural frequency of the four most dominant modes in segment 2. The samples on the horizontal axis are sorted in chronological order.



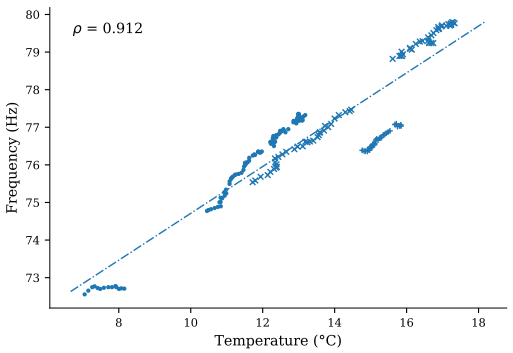
(a) First mode (1/S15)



(b) Second mode (2/S16)

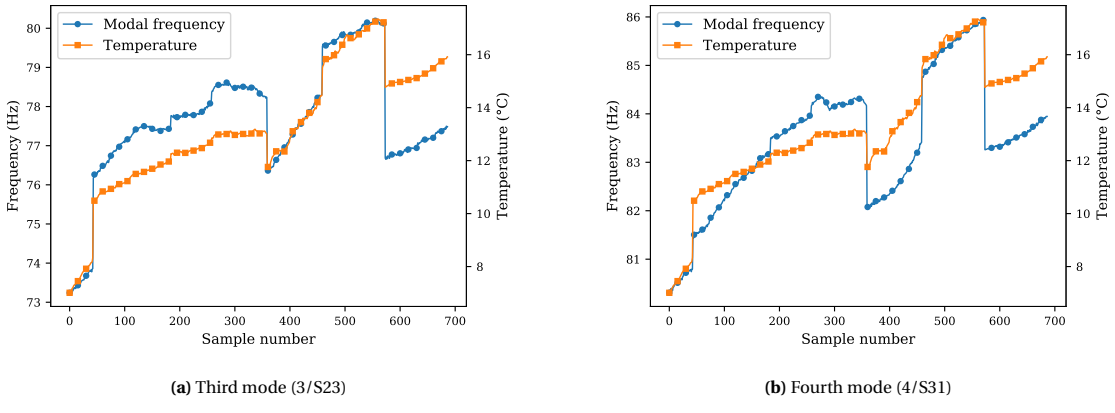


(c) Fourth mode (4/S13)



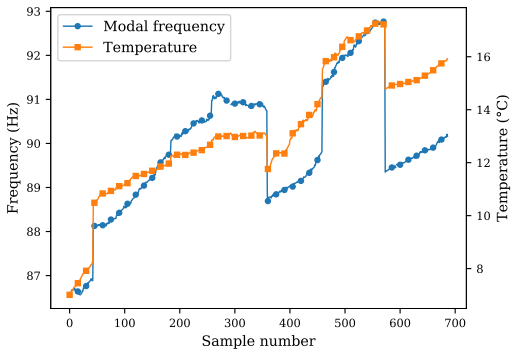
(d) Fifth mode (5/S17)

Figure 8.4: Segment 2 – Correlation between natural frequency and temperature for the four most dominant modes in segment 2. The suggested linear relationship among the variables is emphasized with a linear regression fit (dashed line). Markers: (.) 23 March 2017, (x) 28 March 2017, (+) 31 March 2017.

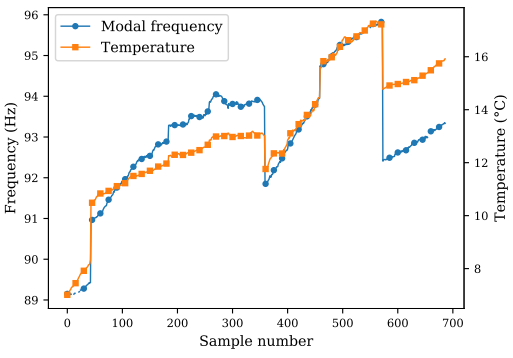


(a) Third mode (3/S23)

(b) Fourth mode (4/S31)

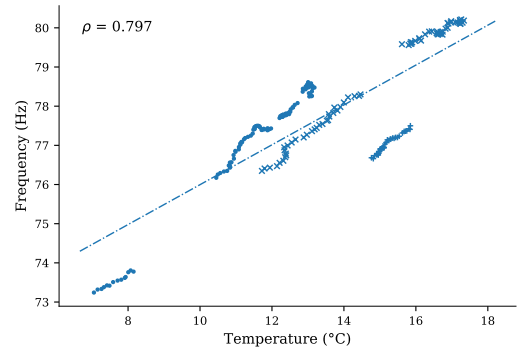


(c) Fifth mode (5/S28)

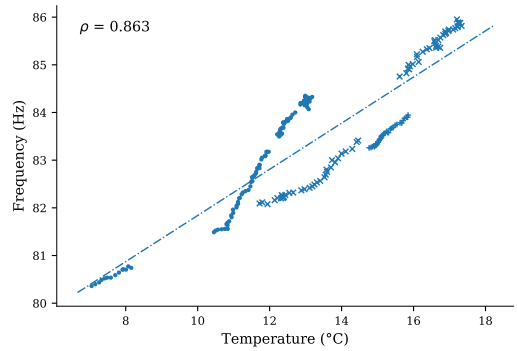


(d) Sixth mode (6/S26)

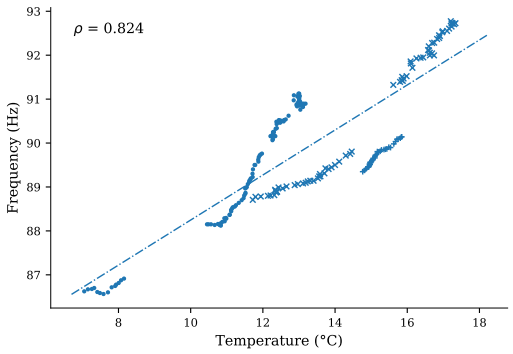
Figure 8.5: Segment 3 – Variation of temperature and natural frequency of the four most dominant modes in segment 3. The samples on the horizontal axis are sorted in chronological order.



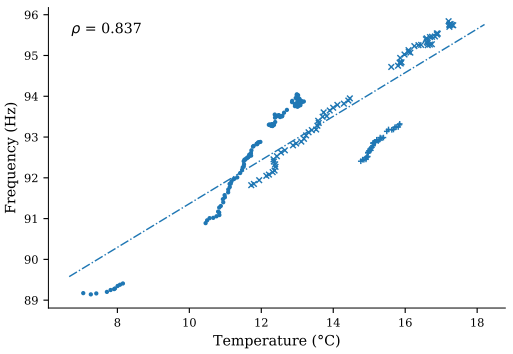
(a) Third mode (3/S23)



(b) Fourth mode (4/S31)



(c) Fifth mode (5/S28)



(d) Sixth mode (6/S26)

Figure 8.6: Segment 3 – Correlation between natural frequency and temperature for the four most dominant modes in segment 3. The suggested linear relationship among the variables is emphasized with a linear regression fit (dashed line). Markers: (.) 23 March 2017, (x) 28 March 2017, (+) 31 March 2017.

Table 8.1: Segment 1 – Summary of the natural frequency data and temperature data, together with the results of the corresponding linear regression for different datasets (separated by added mass). The natural frequency of the four most dominant modes in segment 1 are presented. The temperature data is the same for each mode because no spatial temperature data is available. The slope (w_1) and intercept (w_0) of the fitted line are specified, together with the percentage of the frequency change with respect to the intercept for unit temperature increase and the correlation coefficient ρ.

Mode	Dataset	Added mass (kg)*	Natural frequency				Temperature			Linear regression			
			Mean (Hz)	Min (Hz)	Max (Hz)	Rel. diff. (%)	Mean (°C)	Min (°C)	Max (°C)	w_0 (Hz)	w_1 (Hz/°C)	$w_1/w_0 \cdot 100\%$	ρ
2/S1	1 & 4	0	66.57	62.93	68.24	7.99	11.42	7.01	13.18	56.24	0.91	1.61	0.996
	2	100	66.30	65.17	67.28	3.19	11.32	10.45	11.99	50.05	1.44	2.87	0.993
	3 & 6	50	68.67	67.46	69.72	3.29	14.95	12.20	17.34	63.30	0.36	0.57	0.981
	5	25	66.52	65.54	67.39	2.78	13.07	11.72	14.46	58.03	0.65	1.12	0.977
	7	75	66.47	66.00	66.93	1.39	15.23	14.77	15.92	53.36	0.86	1.61	0.939
	all		**67.07**	**62.93**	**69.72**	**10.12**	**13.24**	**7.01**	**17.34**	**60.45**	**0.50**	**0.83**	**0.773**
4/S3	1 & 4	0	75.36	71.62	77.23	7.44	11.42	7.01	13.18	65.01	0.91	1.39	0.995
	2	100	75.20	74.27	75.99	2.28	11.32	10.45	11.99	61.95	1.17	1.89	0.994
	3 & 6	50	77.49	76.29	78.59	2.96	14.95	12.20	17.34	72.48	0.33	0.46	0.974
	5	25	75.65	74.75	76.54	2.36	13.07	11.72	14.46	67.15	0.65	0.97	0.987
	7	75	75.72	75.29	76.19	1.19	15.23	14.77	15.92	64.75	0.72	1.11	0.954
	all		**76.01**	**71.62**	**78.59**	**9.16**	**13.24**	**7.01**	**17.34**	**69.20**	**0.51**	**0.74**	**0.824**
5/S2	1 & 4	0	87.12	83.66	88.69	5.77	11.42	7.01	13.18	77.55	0.84	1.08	0.997
	2	100	86.78	85.98	87.62	1.88	11.32	10.45	11.99	73.76	1.15	1.56	0.991
	3 & 6	50	88.85	87.84	89.84	2.25	14.95	12.20	17.34	84.56	0.29	0.34	0.967
	5	25	87.00	86.20	87.76	1.80	13.07	11.72	14.46	78.93	0.62	0.78	0.986
	7	75	87.12	86.73	87.55	0.95	15.23	14.77	15.92	76.36	0.71	0.93	0.970
	all		**87.51**	**83.66**	**89.84**	**7.06**	**13.24**	**7.01**	**17.34**	**81.60**	**0.45**	**0.55**	**0.784**
6/S7	1 & 4	0	90.84	87.81	92.10	4.72	11.42	7.01	13.18	82.26	0.75	0.91	0.998
	2	100	90.08	89.05	91.19	2.38	11.32	10.45	11.99	73.42	1.47	2.00	0.990
	3 & 6	50	92.84	91.55	94.10	2.75	14.95	12.20	17.34	86.35	0.43	0.50	0.989
	5	25	90.45	89.70	91.61	2.11	13.07	11.72	14.46	81.25	0.70	0.87	0.976
	7	75	91.58	91.22	91.93	0.77	15.23	14.77	15.92	82.05	0.63	0.76	0.966
	all		**91.30**	**87.81**	**94.10**	**6.89**	**13.24**	**7.01**	**17.34**	**83.88**	**0.56**	**0.67**	**0.887**

* The added mass is not located in this segment, but this weight refers to the added mass that was attached to the bridge deck of segment 2.

Table 8.2: Segment 2 – Summary of the natural frequency data and temperature data, together with the results of the corresponding linear regression for different datasets (separated by added mass). The natural frequency of the four most dominant modes in segment 2 are presented. The temperature data is the same for each mode because no spatial temperature data is available. The slope (w_1) and intercept (w_0) of the fitted line are specified, together with the percentage of the frequency change with respect to the intercept for unit temperature increase and the correlation coefficient ρ.

Mode	Dataset	Added mass (kg)*	Natural frequency				Temperature			Linear regression			
			Mean (Hz)	Min (Hz)	Max (Hz)	Rel. diff. (%)	Mean (°C)	Min (°C)	Max (°C)	w_0 (Hz)	w_1 (Hz/°C)	$w_1/w_0 \cdot 100\%$	ρ
1/S15	1 & 4	0	60.90	57.35	62.51	8.47	11.42	7.01	13.18	50.93	0.87	1.72	0.997
	2	100	61.29	60.26	62.19	3.15	11.32	10.45	11.99	46.58	1.30	2.79	0.993
	3 & 6	50	63.67	62.48	64.90	3.80	14.95	12.20	17.34	57.29	0.43	0.74	0.992
	5	25	61.37	60.36	62.49	3.48	13.07	11.72	14.46	50.82	0.81	1.59	0.989
	7	75	61.15	60.68	61.65	1.57	15.23	14.77	15.92	48.14	0.85	1.77	0.973
	all		**61.85**	**57.35**	**64.90**	**12.21**	**13.24**	**7.01**	**17.34**	**54.73**	**0.54**	**0.98**	**0.783**
2/S16**	1 & 4	0	65.01	61.22	66.67	8.39	11.42	7.01	13.18	54.27	0.94	1.73	0.998
	2	100	64.99	63.99	65.86	2.89	11.32	10.45	11.99	49.63	1.36	2.73	0.992
	3 & 6	50	67.59	65.90	69.17	4.84	14.95	12.20	17.34	58.67	0.60	1.02	0.998
	5	25	65.68	64.63	66.68	3.12	13.07	11.72	14.46	56.05	0.74	1.31	0.980
	7	75	65.65	65.32	66.10	1.19	15.23	14.77	15.92	54.26	0.75	1.38	0.981
	all		**65.91**	**61.22**	**69.17**	**12.06**	**13.24**	**7.01**	**17.34**	**57.65**	**0.62**	**1.08**	**0.874**
4/S13	1 & 4	0	73.96	70.73	75.98	7.10	11.42	7.01	13.18	63.67	0.94	1.48	0.997
	2	100	74.04	72.93	75.38	3.31	11.32	10.45	11.99	54.76	1.72	3.14	0.990
	3 & 6	50	77.34	75.30	78.42	4.03	14.95	12.20	17.34	68.66	0.55	0.80	0.997
	5	25	74.92	73.78	75.97	2.92	13.07	11.72	14.46	64.01	0.84	1.31	0.984
	7	75	74.72	74.28	75.25	1.30	15.23	14.77	15.92	62.33	0.81	1.30	0.978
	all		**75.14**	**70.73**	**78.42**	**10.23**	**13.24**	**7.01**	**17.34**	**66.60**	**0.64**	**0.96**	**0.876**
5/S17**	1 & 4	0	75.64	72.54	77.35	6.36	11.42	7.01	13.18	66.33	0.84	1.26	0.997
	2	100	75.69	74.77	76.38	2.13	11.32	10.45	11.99	62.13	1.20	1.93	0.983
	3 & 6	50	78.45	76.49	79.83	4.26	14.95	12.20	17.34	69.04	0.62	0.90	0.998
	5	25	76.50	75.52	77.47	2.54	13.07	11.72	14.46	67.34	0.70	1.04	0.988
	7	75	76.68	76.32	77.10	1.00	15.23	14.77	15.92	65.18	0.75	1.16	0.981
	all		**76.72**	**72.54**	**79.83**	**9.51**	**13.24**	**7.01**	**17.34**	**68.48**	**0.62**	**0.91**	**0.912**

* The added mass is located in this segment.
** Modes with a local mode shape close to the added mass.

Table 8.3: Segment 3 – Summary of the natural frequency data and temperature data, together with the results of the corresponding linear regression for different datasets (separated by added mass). The natural frequency of the four most dominant modes in segment 3 are presented. The temperature data is the same for each mode because no spatial temperature data is available. The slope (w_1) and intercept (w_0) of the fitted line are specified, together with the percentage of the frequency change with respect to the intercept for unit temperature increase and the correlation coefficient ρ.

Mode	Dataset	Added mass (kg)*	Natural frequency				Temperature			Linear regression			
			Mean (Hz)	Min (Hz)	Max (Hz)	Rel. diff. (%)	Mean (°C)	Min (°C)	Max (°C)	w_0 (Hz)	w_1 (Hz/°C)	$w_1/w_0 \cdot 100\%$	ρ
3/S23	1 & 4	0	77.00	73.24	78.61	6.98	11.42	7.01	13.18	66.53	0.92	1.38	0.998
	2	100	77.09	76.17	77.50	1.72	11.32	10.45	11.99	66.73	0.92	1.37	0.940
	3 & 6	50	79.09	77.70	80.23	3.20	14.95	12.20	17.34	71.74	0.49	0.68	0.998
	5	25	77.34	76.35	78.30	2.53	13.07	11.72	14.46	67.36	0.76	1.13	0.988
	7	75	77.03	76.65	77.50	1.10	15.23	14.77	15.92	65.74	0.74	1.13	0.977
	all		**77.64**	**73.24**	**80.23**	**9.00**	**13.24**	**7.01**	**17.34**	**70.91**	**0.51**	**0.72**	**0.797**
4/S31	1 & 4	0	83.15	80.31	84.35	4.86	11.42	7.01	13.18	75.41	0.68	0.90	0.999
	2	100	82.37	81.44	83.18	2.11	11.32	10.45	11.99	67.95	1.27	1.87	0.992
	3 & 6	50	84.76	83.49	85.95	2.91	14.95	12.20	17.34	78.47	0.42	0.54	0.990
	5	25	82.56	82.08	83.41	1.62	13.07	11.72	14.46	76.22	0.49	0.64	0.964
	7	75	83.53	83.26	83.95	0.82	15.23	14.77	15.92	73.74	0.64	0.87	0.988
	all		**83.41**	**80.31**	**85.95**	**6.77**	**13.24**	**7.01**	**17.34**	**77.00**	**0.48**	**0.63**	**0.863**
5/S28	1 & 4	0	89.74	86.56	91.14	5.10	11.42	7.01	13.18	80.65	0.79	0.98	0.997
	2	100	88.79	88.12	89.78	1.86	11.32	10.45	11.99	74.90	1.23	1.64	0.965
	3 & 6	50	91.44	90.07	92.77	2.96	14.95	12.20	17.34	84.83	0.44	0.52	0.978
	5	25	89.14	88.69	89.81	1.25	13.07	11.72	14.46	84.50	0.35	0.42	0.967
	7	75	89.73	89.35	90.21	0.96	15.23	14.77	15.92	78.94	0.71	0.90	0.979
	all		**89.93**	**86.56**	**92.77**	**6.91**	**13.24**	**7.01**	**17.34**	**83.12**	**0.51**	**0.62**	**0.824**
6/S26	1 & 4	0	92.77	89.13	94.06	5.31	11.42	7.01	13.18	82.57	0.87	1.05	0.997
	2	100	92.04	90.89	92.90	2.19	11.32	10.45	11.99	75.44	1.47	1.94	0.992
	3 & 6	50	94.51	93.26	95.84	2.73	14.95	12.20	17.34	87.65	0.46	0.53	0.995
	5	25	92.90	91.82	93.95	2.29	13.07	11.72	14.46	82.19	0.82	1.00	0.989
	7	75	92.84	92.41	93.35	1.02	15.23	14.77	15.92	79.84	0.85	1.07	0.971
	all		**93.11**	**89.13**	**95.84**	**7.21**	**13.24**	**7.01**	**17.34**	**86.01**	**0.54**	**0.62**	**0.837**

* The added mass is not located in this segment, but this weight refers to the added mass that was attached to the bridge deck of segment 2.

The following observation based on the Tables 8.1–8.3 and Figures 8.2, 8.4 and 8.6 can be made:

- In general for all modes, the natural frequency increases with increasing temperature. This is not in line with the studies in literature, where in the majority of the cases the natural frequency decreases with increasing temperature. It was tried to find a reasonable explanation for this contradiction, but this was not possible with the available data. More research is needed to get a better understanding of the influence of temperature on the dynamic properties (e.g. natural frequencies) of the bridge. It must be noted that the studies in the literature consider the temperature–frequency relation of the fundamental modes, while in this thesis higher order modes were considered. This is important as higher order modes might have different behavior. No studies were found that analyze the influence of temperature on such modes.
- The correlation coefficients of the temperature-frequency data of the five mass classes range between 0.939 and 0.999, implying that there is a strong positive linear correlation between the two variables.
- The correlation coefficient between the temperature and frequency using all the available data range between 0.597 and 0.912, showing that there is a good linear correlation between the variables, but the correlation is less strong than for the data when the mass classes are analyzed separately.
- When short time periods (i.e. data obtained at one day) are analyzed, the temperature-frequency relation tends to be linear (not shown in the table, but can be observed in Figures 8.2, 8.4 and 8.6 by considering the different markers).
- The frequency change per unit temperature increase, indicated by the coefficient w_1 (i.e. slope of the line), is not constant over all datasets, which can probably be explained by the time moments. The measurement periods of the datasets are listed in Table 2.1. The largest value of the coefficient w_1 is found for dataset two, which is measured in the morning between 10:40 to 12:23.
- Considering all data, the relative frequency differences range between 6.76% and 12.21%. The smallest relative frequency difference can be found in segment three and the largest can be found in segment two (in particular found by modes in segment two with a local mode shape close to the added mass). For higher modes the frequency difference tends to decrease, except for the modes of segment three, suggesting that the higher modes are less sensitive to variations in the dependent variables such as the environmental conditions. The corresponding temperature increase was in all cases 10.33°C.

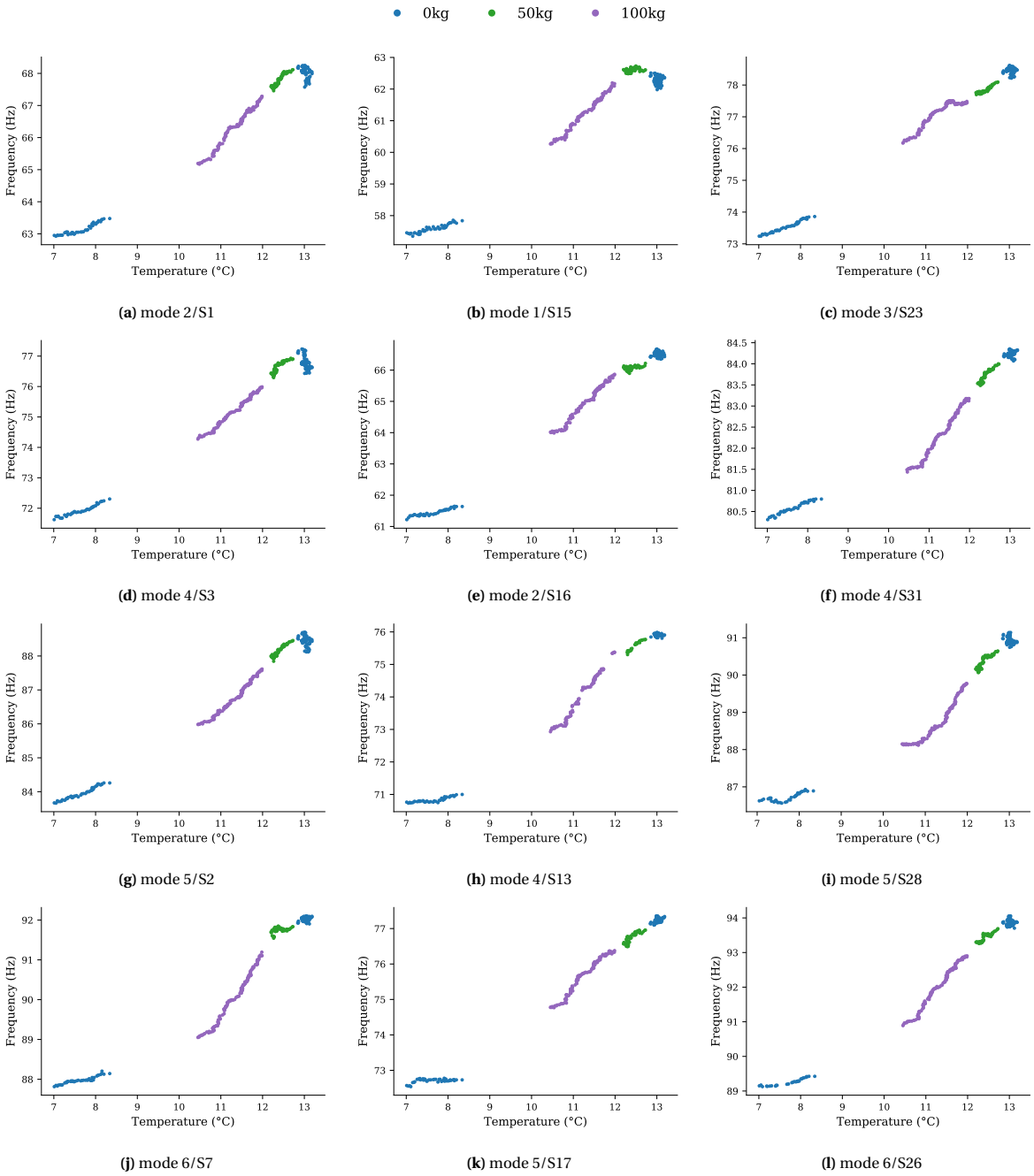
8.2.1 Mass effect or temperature effect?

The added mass classes are disregarded in previous analysis. However, it is worth noting the following observations regarding the class separability. In Chapter 7 and Chapter 6 it was assumed that the added mass changes the natural frequencies of the bridge, resulting into a shift of the peaks in the PSD (or singular value spectrum resulting from FDD). In particular, based on the formula for the natural frequency of a SDOF system it is expected that the natural frequencies decrease when a heavier mass is attached to the bridge. Based on Figure 8.7 someone might argue that this is indeed the case (except for 0 kg at low temperatures), but Figure 8.8 contradicts this observation. The changes in the natural frequencies caused by the temperature variations seem to completely mask the changes caused by the the added mass. This is supported by the following observations that can be made from figures:

- It is expected that the two modes 2/S16 and 5/S17 have the highest sensitivity to the added mass, because these modes have a mode shape close to the added mass. However, the frequency variation is very similar for all modes and no obvious deviations can be found for the two modes with highest mass sensitivity.
- The variations of a natural frequency that can be observed within the dataset of a particular mass class will not be the result of the added mass. Instead, an instant change of structural mass will result in an abrupt change in natural frequency, particularly for the modes closest to the added mass. This abrupt change is not observed in the figures.

In addition, in Chapter 6 we found that the support vector machine exhibit good capabilities for predicting the mass classes of data. However, considering the temperature data and the results of previous analysis, the following comment can be made: since each mass class is measured by a different temperature, the mass

classes can be distinguished from each other by temperature. The temperature is not directly included as feature in the classification algorithm. However, from previous analysis it can be concluded that the temperature has a linear effect on the natural frequencies of the bridge. Thus, the differences in temperature is reflected in the differences of the natural frequencies. So when using the natural frequencies as features for the classification algorithm (without data normalization), the classification is implicitly based on the temperature. This is supported by the figures where it can be observed that the temperature has more effect on the natural frequency than the added mass; that is the natural frequency of all modes show similar relation with temperature, with no obvious deviations for modes close to the added mass.



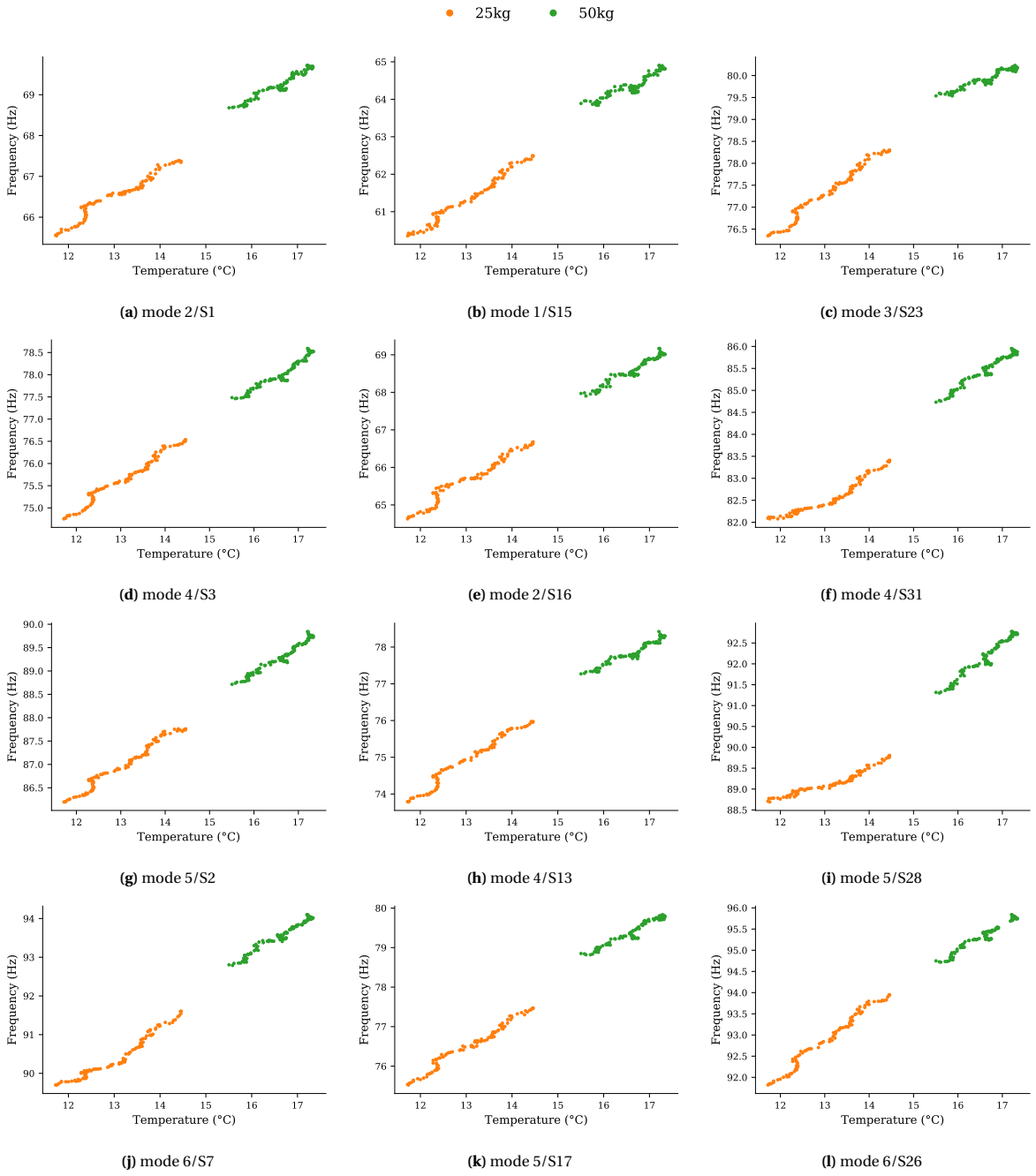
(a) mode 2/S1 **(b)** mode 1/S15 **(c)** mode 3/S23

(d) mode 4/S3 **(e)** mode 2/S16 **(f)** mode 4/S31

(g) mode 5/S2 **(h)** mode 4/S13 **(i)** mode 5/S28

(j) mode 6/S7 **(k)** mode 5/S17 **(l)** mode 6/S26

Figure 8.7: Correlation between natural frequency and temperature for the data obtained on **March 23, 2017**. Each one of the three columns presents the modes from a different bridge segment; the column on the left for the first segment, the middle one for the second segment, and the column on the right for the third segment. The added mass is located in second (middle) segment.



(a) mode 2/S1 **(b)** mode 1/S15 **(c)** mode 3/S23

(d) mode 4/S3 **(e)** mode 2/S16 **(f)** mode 4/S31

(g) mode 5/S2 **(h)** mode 4/S13 **(i)** mode 5/S28

(j) mode 6/S7 **(k)** mode 5/S17 **(l)** mode 6/S26

Figure 8.8: Correlation between natural frequency and temperature for the data obtained on **March 28, 2017**. Each one of the three columns presents the modes from a different bridge segment; the column on the left for the first segment, the middle one for the second segment, and the column on the right for the third segment. The added mass is located in second (middle) segment.

8.3 Data Normalization

When the environmental and/or operational variabilities are significant, the effects must be removed from the features before any damage detection algorithm is employed. Two complementary approaches can be used for this purpose.

The first one consists in extracting features that are mainly sensitive to damage but insensitive to the variability of the system and its environment. For example, Deraemaeker and Preumont [15] propose to use the appearance of spurious peaks in the outputs of modal filters as features for damage detection, because this feature is very sensitive to a local damage scenario, but not very sensitive to global changes due to the environment. Cross et al. [13] uses the univariate novelty index and the principal component analysis to identify a subset of the feature variables which are insensitive to temperature variations yet sensitive to damage. This approach requires data from different structural conditions (healthy and damaged) measured under varying environmental and operational conditions, which might be difficult to obtain. In fact, this is a supervised learning approach for feature selection.

The second approach uses a model of the impact of the environment on the features of interest in order to remove it from the extracted features. Three methods can be used for this purpose. First, one can attempt to directly model the impact of the environment on the dynamical characteristic of the structure (based on physical laws). This is a difficult task, because there may be many factors which need to be taken into account, and the types of models (e.g. constitutive equations and its parameters) are generally unknown.

One alternative is to model the relation between the dynamic characteristics of the structure and the environmental and/or operational variables based on measurements on real structures. Different types of models including linear models, nonlinear models and learning models can be used for this. However, these models do not have a real physical meaning, as they only model an input-output relationship, and they are not derived from physical laws. This restricts their use for the structure for which they have been developed. In practical applications, most researchers have limited their studies to modeling the relationship between the first few natural frequencies and one environmental variable (e.g. temperature). Examples of such are: Peeters and Roeck use an ARX model for the Z24 bridge, Sohn et al. use a linear filter for the Alamos Canyon bridge and Ni et al. use the Support Vector Machine (SVM) and neural networks for the Ting Kau bridge, Ding and Li use a polynomial regression model for the Runyang Suspension Bridge.

Previous approach has several practical drawbacks. As pointed out by Kullaa [30], the optimal locations of the temperature sensors may be difficult to determine or to reach. Another difficulty is that other environmental and/or operational variables might also affect the features, some of them might be difficult to measure. Moreover, once the correlation models has been established, the sensors that measure the environmental and/or operational variables must remain at same place on the structure, any change in the monitoring system (e.g. failure of a sensor) might cause problems for the damage detection. In order to overcome these drawbacks, other methods have been proposed that could remove the variability due to environment without measuring the environmental variables. Examples of such are: linear or nonlinear principal component analysis [56, 57], factor analysis [30], singular value decomposition [41, 48] and Auto-associative Neural Network [45].

8.3.1 Regression and Interpolation models

Several researchers show that it is possible to train a model that accurately describe the relation between modal parameters (often the natural frequencies) and some environmental parameters. These models can subsequently be used to eliminate the environmental effects in vibration-based damage detection.

Sohn et al. and Peeters and Roeck both proposed a (multivariate) linear model to accommodate the changes in temperature to the damage detection system. Here it was assumed that the dynamic properties change with temperature linearly. The proposed linear models are very easy to implement and are consistent with the simplified theoretical formulations [59].

Sohn et al. [43, 44] proposed a linear adaptive filter for predicting changes in modal parameters of a full-scale bridge due to environmental temperature. Data from the Alamosa Canyon Bridge in New Mexico were used to demonstrate the applicability of the adaptive filter. The first dataset from 1996 was used to train the adaptive filter, while the second dataset from 1997 was used to test the prediction performance. The modal parameters were extracted using the Eigensystem Realization Algorithm (ERA). Approximately nine meaningful modes were identified from the ERA within the range of 0–30 Hz. Only the first two modes are considered in the study. The temperature is measured at nine different locations across the centre of the span.

In previous studies by Farrar et al. [22] it was shown that the measured first modal frequency varied approximately 5% during the 24 hour test period, and the change in the measured fundamental frequency was found to correlate with the temperature difference across the deck. Similar variations and correlations with deck temperature difference were observed for the other modes of the bridge.

A simple multivariate linear regression model was used to describe the relationship between the selected bridge temperatures and its measured fundamental frequency, which can be expressed as

$$y = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_r x_r + \varepsilon = w_0 + \mathbf{x}^{\mathrm{T}}\mathbf{w} + \varepsilon \tag{8.16}$$

where y is the fundamental frequency, $\mathbf{x} = (x_1, x_2, \ldots, x_r)^{\mathrm{T}}$ are the selected temperature inputs, $\mathbf{w}$ is a vector with regression coefficients that weight each input, and ε is the error term. The inputs of the model represents the spatial and temporal temperature distributions to account for the geographical (north-south) orientation of the structure, and the thermal inertia of the structure. Based on statistical analyses, a satisfactory model was selected that reproduced the variation of the natural frequencies reasonably well. The selected model includes two temporally separated and two spatially separated temperature measurements, which reveals that 1) the response change of the Alamosa Canyon Bridge lags the temperature of the bridge, and 2) the temperature gradient between west and east largely influences the variation of the fundamental frequency (similar as was observed by Farrar et al. [22]).

Once the regression model is trained, it can be used to establish confidence intervals of the frequencies for a new temperature profile. A newly measured frequency can be compared against the confidence interval. If the fundamental natural frequency falls outside the confidence interval, then one may suspect with the given confidence that some changes in the underlying structural characteristic are caused by damage or other effects. The comparison of the prediction intervals obtained from the first data set and the measured frequencies from the second test data reveals that the bridge experienced a statistically significant decrease in the first and second mode frequencies. This implies that the stiffness of the structure is deteriorated and/or the mass of the structure is increased. The authors supposed that this consistent decrease of fundamental natural frequency was mainly caused by the increase of the bridge mass as the Alamosa Canyon Bridge absorbed significant amount of moisture and the bridge retained some of the rainfall on its surface.

Peeters and Roeck [37] performed a regression analysis of the natural frequencies of the Z24 Bridge on temperature to filter out the temperature effects from the measured frequencies. The Z24-bridge is located in Switzerland and has been monitored for almost one year before it was artificially damaged. The bridge was instrumented with 49 sensors to capture environmental parameters such as ambient temperature, soil temperature at the boundaries, concrete temperatures, wind, humidity, and bridge expansion, but it appears that only the temperatures were correlated with the natural frequencies of the bridge, and the study considers only this relation. Four modes of the undamaged bridge, in the range 0–12 Hz, were identified using automated modal analysis. Eigenfrequency differences between 14% and 18% are observed, which have to be explained by normal environmental changes. For almost all combination of natural frequency versus temperature, a bilinear relationship between temperature and frequency was found, with the knee situated around 0 degrees Celsius. It was concluded that this bilinear behaviour can be attributed to the asphalt layer. During warm periods the asphalt did not contribute to the overall stiffness, but during cold periods, it added significant stiffness of the structure. In general, the frequency decreases when the temperature is elevated, except for the second mode for which the frequency increases with increasing temperatures (for positive temperatures).

To take into account the thermal inertia of the asphalt and the concrete, a dynamic linear regression model called ARX was used to describe the relation between eigenfrequency and temperature. The ARX model consist of an auto-regressive output and an eXogeneous input part

$$y_k + a_1 y_{k-1} + \cdots + a_{n_a} y_{k-n_a} = b_1 u_{k-n_k} + b_2 u_{k-n_k-1} + \cdots + b_{n_b} u_{k-n_k-n_b+1} + \varepsilon_k \tag{8.17}$$

where y_k is the output (in this case an eigenfrequency) at time instant k, u_k is the input (in this case a temperature), and ε_k is the error term. Six temperature readings were selected as representative variables for the environmental variability. The monitored wind characteristics, rainfall and humidity were also considered as possible inputs. However, no relation was found between these quantities and the eigenfrequencies.

The strategy followed in this study is as follows: Single-Input/Single-Output (SISO) ARX models are constructed for all four identified natural frequencies and six selected input candidates. A single representative input is selected that yields on the average the best models for all four frequencies. The best models have lowest loss function and final prediction error. The best ARX models are used for simulating the eigenfrequencies of the Z24-bridge and estimating confidence intervals. If a new measured eigenfrequency lies outside the estimated confidence interval, it is likely that the bridge is damaged. In this study, the damage introduced was

the incremental settlement of one of the bridge piers, which were successfully detected using the proposed approach. Different types of ARX models were considered; it was found that an ARX model that include the thermal dynamics of the bridge is superior to a static regression model. Also it turned out that a temperature measurement at one location (single input) was sufficient to find an accurate dynamic ARX model. On the other hand, the static regression models could be improved by adding more input variables.

The regression approaches performed by Peeters and Roeck [37] and Sohn et al. [43, 44] were compared by Sohn [42]. It was mentioned that Peeters' work emphasizes the thermal dynamics of the bridge by using a single temperature measurement with multiple time lags (the temporal variation of temperature), while the work by Sohn used temperature readings from multiple thermocouples to take into account the temperature gradient across the bridge (the spatial variation of temperature) as well as the temporal variation. The comparison of these two approaches clearly demonstrates that data normalization is problem-specific; one kind for each individual structure. The orientation and location of the Alamos Canyon Bridge makes it reasonable that the spatial variation of the bridge temperature might have been the main driving factor for the frequency variation. On the other hand, the magnitude of the Z24 Bride (30 m long main span and two 20 m side spans with 8.6 m width) is larger than the 7.3 m wide and 15.2 m long span of the Alamos Canyon Bridge tested, which makes it reasonable that it takes a longer time before the temperature affected the dynamic properties of the bridge and thus the time-lag information of the temperature was more important for the Z24 bridge than for the Alamos Canyon Bridge.

Ding and Li [16] proposed a polynomial regression model to describe the frequency-temperature seasonal correlations of the Runyang Suspension Bridge, which can be mathematically described as

$$f(T,n) = p_n T^n + p_{n-1} T^{n-1} + \cdots + p_2 T^2 + p_1 T + p_0 \tag{8.18}$$

where T is the daily averaged value of temperature, f denotes the daily averaged values of the frequency, n denotes the order of polynomial regression model, and p_i denotes the regression coefficients. It was found that a sixth order polynomial regression model exhibits good capabilities for mapping between the temperature and measured modal frequency (a different model was used for each modal frequency). The established models were subsequently used to normalize all the measured natural frequencies to a fixed reference temperature, resulting in features that are independent of the temperature effects.

More advanced methods can also be used to model the correlation between dynamic properties and environmental parameters. For instance, Ni et al. [34] applied the SVM regression algorithm to establish regression models which quantify the effect of temperature on modal frequencies for the cable-stayed Ting Kau Bridge (Hong Kong) based on long-term monitoring data. The results obtained by the SVM models were compared with those produced by a multivariate linear regression model and showed that the SVM models exhibit good capabilities for mapping between the temperature and modal frequencies.

The proposed SVM model uses only the input at the current time instant (average temperature) to describe the current output (modal frequency), which is often referred to as a "static" model. In order to examine the influence of thermal inertia, a dynamic SVM model is formulated using continuous measured data [25]. By comparing the obtained results from the static and dynamic models it was concluded that the dynamic model is superior to the static model, which shows that the change of modal frequency lags behind the temperature variation, and dynamic regression models are preferable to represent the temperature-frequency correlation when continuous measurement data is available. Subsequently, the Principal Component Analysis (PCA) was added before conducting SVM algorithm. The PCA extracts uncorrelated principal components from the measured temperatures for dimensionality reduction and removes the redundancy in the input data. The removal of the redundancy might improve the generalization performance of the regression models. The proposed method using the PCA-compressed features was compared with the method directly using measurement data to train the SVM models. It was found that the SVM model trained using PCA-compressed feature vectors outperforms the SVM model trained using the original data in both model accuracy and computational costs.

Ni et al. and Zhou et al. [33, 60] use neural network models to establish the correlation models between the modal frequencies and the temperatures, for the monitoring data of the Ting Kau bridge. Different strategies for the construction of an appropriate input to the neural networks, including mean temperatures, effective temperatures and principal components of temperatures, were addressed by Zhou et al. [61].

Ko and Ni [29] used the monitoring data from the Ting Kau bridge to compare different learning methods including linear regression, nonlinear regression, neural network, and SVM for modeling the effect of temperature on modal frequencies. It was observed that linear and nonlinear regression models are unable to accurately predict the frequency variations. The neural network and SVM models exhibited good capabilities

in both reproduction and prediction. This indicate that more advanced methods are superior to the simplistic methods.

In Section 8.2 an attempt was made to model the relation between the natural frequencies and ambient temperature for the Zwartewaterbrug data. A linear model was used to describe the relation. However, the suggested model was not suitable for modeling the relation between temperature and natural frequency over multiple days. The limited temperature data from a single location and the relatively small dataset of vibration measurements are the main difficulties for the analysis. Moreover, the vibration measurements (including different added mass) cover a wide range of temperatures, but there are only a few observations per temperature which is insufficient for a reliable statistical analysis.

8.3.2 Decomposition methods

In previous studies a correlation model was established using the vibration measurements and the measurements of some relevant environmental parameters. Sohn et al. [44] use the temperature information from nine different locations across the center of the span. Peeters and Roeck [37] use the temperature below the asphalt layer from a single location in the middle of the main span of the Z24 bridge. Ni et al. [34] uses the average temperature of 20 temperature sensors across the bridge. Li et al. [31] developed a model including both the temperature and wind velocity effects.

There are situations in which direct measurements of these operational and environmental parameters are impractical or difficult to achieve, or it is unknown which parameters to measure or where to measure. For these situations, it may be possible to find an underlying model or some embedded variables that can explain the correlation among the features. The principal component analysis and factor analysis are the two most commonly used techniques for data normalization when measurements of environmental and operational parameters are not available. Both methods are based on a decomposition of the covariance matrix of the features monitored over a long period of time with changing environmental conditions. For data normalization they rely on the assumption that damage produces changes in the extracted features that are "orthogonal" to the changes caused by the environmental and operation variations of the structure. If this assumption is not satisfied, some damage effect might also be removed from the data with the consequent inability of damage detection. In such cases, nonlinear methods which consist in identifying a nonlinear manifold instead of orthogonal subspaces can be used.

Principal Component Analysis and Factor Analysis

Kullaa [30] proposed a data normalization technique based on Factor Analysis (FA). The objective of factor analysis is to identify the underlying latent factors that can explain the correlation among the observed dependent variables. In particular, the factor analysis assumes that there are a smaller number of underlying factors that describe the variation of the measured variables and the measured variable are subjected to random errors, which can be written as

$$\mathbf{x} = \mathbf{\Lambda}\boldsymbol{\xi} + \boldsymbol{\varepsilon} \tag{8.19}$$

where $\mathbf{x} \in \mathbb{R}^p$ is the vector of measured variables, $\mathbf{\Lambda} \in \mathbb{R}^{p\times d}$ is the matrix of *factor loadings* ($d < p$), $\boldsymbol{\xi} \in \mathbb{R}^d$ is the vector of unobservable *factors* (in this case the environmental and operational variables such as temperature that influence the observed feature vectors), and $\boldsymbol{\varepsilon} \in \mathbb{R}^p$ is the vector of *unique factors*. The unique factors are assumed to be normally distributed with zero mean and a diagonal covariance matrix; that is, $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Psi})$.

It is assumed that the factors represent the environmental and operational parameters, so that the unique factors $\boldsymbol{\varepsilon}$ are variables independent of the common factors and can be used for damage detection. That is, if the structural condition deteriorates so that the measured variables change, the previously trained factor model cannot explain the multivariate correlation of the newly measured data, causing an increase in unique factors. The training phase of the factor model involves the estimation of the factor loadings $\mathbf{\Lambda}$ and the covariance matrix of the unique factors $\mathbf{\Psi} \in \mathbb{R}^{p\times p}$. The training set is formed from feature vectors obtained from data measure on the undamaged structure under a full range of environmental and/or operational conditions. The factor model is subsequently used with new data (from potentially damaged structure) resulting in the corresponding unique factors/ residuals $\hat{\boldsymbol{\varepsilon}}$. The state of the structure can then be assessed using a damage index, which can be defined as the Euclidean distance or the Mahalanobis distance of the residuals. The former is simply defined as

$$DI_k = \|\hat{\boldsymbol{\varepsilon}}_k\| \tag{8.20}$$

If the damage index significantly deviates from zero, this indicate that the new feature vector $\mathbf{x}_k$ are extracted

from data corresponding to the structure in an abnormal condition that cannot be attributed to the environmental or operational variability represented in the training data. Statistical analysis is usually used to determine a threshold level for the damage index, when this limit is exceeded an alarm should be issued.

It is emphasized that factor analysis is limited to linear relationships among the observed features. For the cases in which the environmental or operational variations have a nonlinear effect on the observed features, the nonlinear relationships can be modeled by mixing different linear factor models over different data spaces [30].

Yan et al. [56] proposed a damage detection method for structural health monitoring under varying environmental and operational conditions based on the PCA. Similar to the previous method, the data normalization is performed without measuring the environmental variables. The underlying physical quantities do not need to be known, but the environmental effects are treated as embedded variables instead. In particular, the PCA assumes that the total variance in the observed dependent variables (feature vectors of dimension p) can be explained by a smaller number of uncorrelated variables of dimension d called the the *principal component scores* ($d < p$). The linear PCA model can be written as

$$\mathbf{y} = \mathbf{A}^{\mathrm{T}}\mathbf{x} \tag{8.21}$$

In which $\mathbf{x} \in \mathbb{R}^p$ is the vector of measured features, $\mathbf{A} \in \mathbb{R}^{p\times d}$ is the *loading matrix* that projects the measured features into an environmental-factor characterized space, and $\mathbf{y} \in \mathbb{R}^d$ is the vector with principal component scores. The dimension d may be thought as the physical order of the system which corresponds here to the number of combined environmental factors that affect the features. This dimension reduction process forces the model to learn the inherent variables driving changes of the features and to capture the embedded relationship between the environmental factors and the features. It is assumed that the principal component scores computed for the data obtained from the undamaged structure represent the variations due to environmental variations. By remapping the projected data back to the original space using the following relation

$$\hat{\mathbf{x}} = \mathbf{A}\mathbf{y} = \mathbf{A}\mathbf{A}^{\mathrm{T}}\mathbf{x} \tag{8.22}$$

and removing it from the observed data, a residual vector $\mathbf{e}$ that is independent of the environmental factors is obtained, which can be used for damage detection. That is, the residual error remains small if the PCA model is applied on data of a healthy structure (i.e. the same structural condition that was used to form the training dataset), and it increases significantly when structural damage occurs. The residuals of a newly obtained feature vector $\mathbf{x}_k$ from a potentially damaged structure can be expressed as

$$\mathbf{e}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k = \left(\mathbf{I} - \mathbf{A}\mathbf{A}^{\mathrm{T}}\right)\mathbf{x}_k \tag{8.23}$$

Again, the damage index can be defined as the Euclidean distance of the residuals

$$DI_k = \|\mathbf{e}_k\| \tag{8.24}$$

and a statistical analysis technique can be used to indicate damage (e.g. control charts or hypothesis testing).

The training phase of the PCA model consists of constructing the loading matrix $\mathbf{A}$ using the training data obtained from the undamaged structure under a range of environmental conditions. It can be shown that the desired projection with the PCA model is obtained when the columns of $\mathbf{A}$ are the eigenvectors of the covariance matrix of the observed features corresponding to the d largest eigenvalues. The covariance matrix is estimated from the training data obtained from the undamaged structure. The feature vectors are often normalized by removing the mean vector of the training data. It is emphasized that the damaged-state data should not be normalized by removing its own mean value, because then both healthy and damaged-state data are centered at the origin and the separability between the two classes is lost, hence damage could no longer be detected, see Figure 8.9.

The PCA is restricted to mapping only linear correlations among the observed variables, but a nonlinear extension of the PCA exist, which can be used for modeling nonlinear relationships present in the data [45]. Alternatively, Yan et al. [57] propose to use local PCA models for local regions in the data space.

Methodology

The damage detection method based on either factor analysis or principal component analysis can be summarized by the following steps

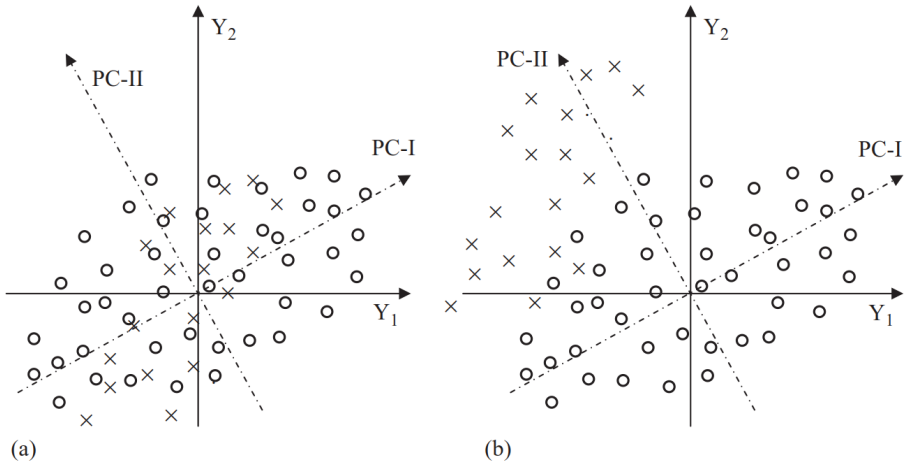


Figure 8.9: Geometric interpretation of PCA with data normalization: (a) removing mean value of each set; (b) removing mean value of reference data for both sets [56].

Step 1. Form a training dataset with measured variables/ features obtained from the undamaged structure under a full range of environmental or operational conditions.

Step 2. Train the latent variable model; This step involves the estimation of the covariance matrix for the observed features from training data, which is then decomposed into some particular matrices that will be denoted **T**.

Step 3. Compute the residual vector $\mathbf{e}_k = \left(\mathbf{I} - \mathbf{T}\mathbf{T}^{\mathrm{T}}\right)\mathbf{x}_k$, where $\mathbf{x}_k$ is a feature vector obtained from a potentially damaged structure under arbitrary environmental conditions.

Step 4. Perform novelty detection analysis or some other statistical analysis to assess whether the residual vector deviates significantly from the reference level, such that damage could be expected.

The main difference between the two methods is **Step 2**, where either the factor model or the PCA can be used. The construction of matrix **T** for the factor model is more complex than in the PCA, and is usually obtained through an iterative process. It can be shown that the results of FA after one iteration is identical to the result of PCA [14].

Although the FA and PCA techniques can both be used for damage detection, the techniques have different underlying ideas. For instance, the FA assumes the existence of an underlying linear model driving the relation between the environmental factors and the features, but there is no such explicit model in the PCA method. The reader may refer to [27] for the similarities and differences between both techniques.

In the study of Deraemaeker and Worden [14], the efficiency of both methods for filtering out environmental variabilities is demonstrated using experimental data, and the results are compared. The experiment concerns a wooden bridge model that have been monitored in a laboratory. Five artificial damage scenarios have been created by adding five different lumped masses to the structure. The additional masses are attached on the top flange at 600 mm left from the structural mid-span. The masses are small and ranges from 0.065% to 0.54% of the total mass of the structure (36 kg). The bridge model is randomly excited using an electrodynamic shaker, and acceleration measurements were collected at 15 different locations. The feature vector is made of the natural frequencies and normalized modal coordinates of nine modes, resulting in a feature vector of dimension $p = 261$. A fan, which was running during the day and turned off during the night, was mainly responsible for change in temperature and humidity. It was shown that the environmental effects could be effectively removed without loss of damage sensitivity, using $d = 81$ principal components/ unobservable factors for the damage index. It was observed that the results of PCA and FA after convergence of the iterative process are very similar. The different levels of damages were clearly detected using both methods. It was noted that the training set should contain the full range of environmental conditions and the feature vectors must be large enough in order to ensure some separability between the damage effects and the environmental effects.

In the same paper it is also shown that the Mahalanobis Squared Distance (MSD) can be used to filter out confounding effects in a very similar way to the two linear techniques, PCA and FA. The MSD was already

introduced in Chapter 7 in the context of damage detection. It is therefore interesting to see how this method can be used for filtering of the environmental effects.

8.3.3 Novelty Detection Under Changing Environmental Conditions

Recall that the MSD of a potential outlier $\mathbf{x} \in \mathbb{R}^p$ is defined as

$$D^2 = (\mathbf{x} - \bar{\mathbf{x}})^{\mathrm{T}} \mathbf{C}_x^{-1} (\mathbf{x} - \bar{\mathbf{x}}) \tag{8.25}$$

where $\bar{\mathbf{x}} \in \mathbb{R}^p$, $\mathbf{C}_x \in \mathbb{R}^{p \times p}$ are the mean vector and covariance matrix, respectively. Given the set of N feature vectors $\{\mathbf{x}_j \in \mathbb{R}^p \colon j = 1, \ldots, N\}$, representing N samples of the "healthy state" of the structure, of which the mean vector and the covariance matrix can be estimated as follows

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{j=1}^{N} \mathbf{x}_j \tag{8.26}$$

and the sample covariance matrix as

$$\mathbf{C}_x = \frac{1}{N-1} \sum_{j=1}^{N} (\mathbf{x}_j - \bar{\mathbf{x}}) (\mathbf{x}_j - \bar{\mathbf{x}})^{\mathrm{T}} \tag{8.27}$$

The components in a feature vector are usually not statistically independent, so that the covariance matrix is not diagonal. However, it is possible to perform a transformation of the feature vector in order to diagonalize the covariance matrix. The real, symmetric and non-singular covariance matrix and thus can always be diagonalized by the following unitary transform

$$\mathbf{C}_x = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^{\mathrm{T}} \tag{8.28}$$

where $\mathbf{\Lambda} \in \mathbb{R}^{p \times p}$ is a diagonal matrix whose elements are the eigenvalues of $\mathbf{C}_x$, assume that the eigenvalues are sorted in descending order of magnitude so that $\lambda_1 > \lambda_2 > \cdots > \lambda_p$. The orthogonal matrix $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_p)$ contains the corresponding (orthonormal) eigenvectors, with the following orthogonality properties $\mathbf{U}\mathbf{U}^{\mathrm{T}} = \mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{I}$. Combining equation (8.25) and (8.28) yields

$$D^2 = (\mathbf{x} - \bar{\mathbf{x}})^{\mathrm{T}} \mathbf{U} \mathbf{\Lambda}^{-1} \mathbf{U}^{\mathrm{T}} (\mathbf{x} - \bar{\mathbf{x}}) \tag{8.29}$$

Define the projection of $\mathbf{x}$ onto the eigenvectors as $\mathbf{y} = \mathbf{U}^{\mathrm{T}}\mathbf{x}$ (i.e. same transformation as the one used in the principal component analysis). The coordinates of $\mathbf{y}$ are equal to $\mathbf{u}_j^{\mathrm{T}}\mathbf{x}$, $j = 1, \ldots, p$. In other words, they are the coordinates of $\mathbf{x}$ in the new space whose axes are determined by the eigenvectors in $\mathbf{U}$. Note that the mean vector of the transformed features is simply $\bar{\mathbf{y}} = \mathbf{U}^{\mathrm{T}}\bar{\mathbf{x}}$. Equation (8.29) can now be written as

$$D^2 = (\mathbf{y} - \bar{\mathbf{y}})^{\mathrm{T}} \mathbf{\Lambda}^{-1} (\mathbf{y} - \bar{\mathbf{y}}) \tag{8.30}$$

Which can be expanded as follows

$$D^2 = \frac{(y_1 - \bar{y}_1)^2}{\lambda_1} + \cdots + \frac{(y_p - \bar{y}_p)^2}{\lambda_p} = \sum_{j=1}^{p} \frac{(y_j - \bar{y}_j)^2}{\lambda_j} \tag{8.31}$$

This shows that the MSD can be decomposed into a sum of independent contributions from each component of the transformed variables $y_j = \mathbf{u}_j^{\mathrm{T}}\mathbf{x}$, $j = 1, \ldots, p$. The contributions are weighted by the inverse of the associated eigenvalues λ_j, which can be interpreted as the variance of the new, transformed variables. In fact, the covariance matrix of the transformed variables is $\mathbf{C}_y = \mathbf{\Lambda}$, and thus the variance of the j-th component is $\sigma_j^2 = \lambda_j$. If the variance is large, then the contribution to the distance is small.

Filtering of the environmental effects

The total variability in the feature vectors extracted from the undamaged structure can usually be explained by a smaller number of transformed features. Strictly speaking, this occurs when some eigenvalues of $\mathbf{C}_x$ are equal to zero, which implies that some of the columns/rows are linearly dependent. In practice, the eigenvalues are never strictly zero due to the noise and numerical precision that is involved in the analysis, but a

significant drop can be observed in the eigenvalues which can be used to define the number of principal components which account for most of the variability. A practical way to determine the number d of dimensions in the principal subspace is to define the following indicator [14]

$$I = \frac{\sum_{j=1}^{d} \sigma_j^2}{\sum_{j=1}^{p} \sigma_j^2} \tag{8.32}$$

and determine the lowest integer d such that $I > e(\%)$, where e is a threshold value (e.g. 99.9%). The thresholds should be interpreted as follows: d principal components are needed in order to explain e% of the variance in the observed data. Now, equation (8.31) can be split into two parts

$$D^2 = \sum_{j=1}^{d} \frac{\left(y_j - \bar{y}_j\right)^2}{\sigma_j^2} + \sum_{j=d+1}^{p} \frac{\left(y_j - \bar{y}_j\right)^2}{\sigma_j^2} = D_1^2 + D_2^2 \tag{8.33}$$

where D_1^2 is the MSD of $\mathbf{x}$ projected on the *major* principal components, and D_2^2 is the MSD of $\mathbf{x}$ projected on the *minor* principal components, sometimes referred to as the null-space of the principal components.

Now assume that the first d principal components, with the largest variance, represent the environmental factors which have strong influence on the extracted features from the undamaged structure. Because the MSD scales each independent component with respect to the inverse of its variance, the distance will have a very low sensitivity to the environmental changes. So, when the covariance matrix $\mathbf{C}_x$ is estimated from the set of features that are extracted from the undamaged structure under a full range of environmental conditions, then the MSD is made insensitive to the environmental changes.

In Chapter 7 the MSD for damage detection was applied on the data of the Zwartewaterbrug. The natural frequencies of five identified modes were used as features. The features exhibit large variations, presumably due to environmental changes, which were naturally filtered by the application of the MSD. It must be mentioned that this filtering will work well only when the features that are used to estimate the covariance matrix cover a full range of environmental conditions (or at least all the environmental conditions under which the other data sets were obtained). Because the features of the healthy condition (0 kg added mass) were measured under more or less constant environmental conditions, it was chosen to use also the features of the 25 kg mass to build the model of the reference state. These features cover a broader range of environmental conditions which makes the MSD less sensitive to environmental changes.

The link between the MSD and the two most commonly used linear techniques to filter confounding effects: principal component analysis and factor analysis, is demonstrated in [14]. Both methods consist of computing a residual vector of the following form

$$\mathbf{e} = \left(\mathbf{I} - \mathbf{T}\mathbf{T}^{\mathrm{T}}\right)\mathbf{x} \tag{8.34}$$

where $\mathbf{x} \in \mathbb{R}^p$ is the original feature vector, and $\mathbf{e} \in \mathbb{R}^p$ is a residual vector that is made insensitive of the environmental variability but still sensitive to damage. $\mathbf{T} \in \mathbb{R}^{p \times d}$ is a transformation matrix that describes the linear mapping between the original variables and some embedded/unobservable variables. For principal component analysis, this matrix contains as its columns the eigenvectors of the covariance matrix $\mathbf{C}_x$ corresponding to the d largest eigenvalues in descending order. Factor analysis is slightly more complex as the factor model must be determined iteratively, but it can be shown that the identified subspace of the first iteration corresponds to the subspace of the principal components in PCA (subspace spanned by the eigenvectors of $\mathbf{C}_x$ corresponding to the d largest eigenvalues) [14]. Although the two methods have different interpretation, the subspaces identified with PCA and FA are often very close.

If the MSD of the residual vector is subsequently used to perform novelty detection, it is essentially the same as computing

$$D^2 = \sum_{j=d+1}^{p} \frac{\left(y_j - \bar{y}_j\right)^2}{\sigma_j^2} = D_2^2 \tag{8.35}$$

which is the MSD of $\mathbf{x}_j$ projected on to the subspace of the minor principal components. The latter equation is equivalent to considering that $\sigma_j^2 = \infty$ for $j \leq d$ in equation (8.33). Hence, if there is a clear drop in the eigenvalues of $\mathbf{C}_x$, so that $\sigma_1, \ldots, \sigma_d \gg \sigma_{d+1}, \ldots, \sigma_p$, then computing the full MSD is equivalent to computing the MSD of the feature vector projected onto the subspace of the minor components.

Next, it will be shown that the residual vector (8.34) is identical to the projection of the feature vector $\mathbf{x}$ onto the subspace of the minor principal components, so that (8.35) is indeed true. The subspace of the

minor principal components is spanned by the last $p-d$ eigenvectors of the covariance matrix $\mathbf{C}_x$, which corresponds to the $p-d$ smallest eigenvalues (recall that it was assumed that the eigenvalues are sorted in descending order of magnitude, and the corresponding eigenvectors are arranged in the same order). Hence, the eigenvectors and eigenvalues in equation (8.28) can be partitioned as

$$\mathbf{U} = \begin{pmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{pmatrix}, \qquad \mathbf{\Lambda} = \begin{pmatrix} \mathbf{\Lambda}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_2 \end{pmatrix} \tag{8.36}$$

where $\mathbf{\Lambda}_1 \in \mathbb{R}^{d\times d}$ and $\mathbf{\Lambda}_2 \in \mathbb{R}^{(p-d)\times(p-d)}$ are diagonal matrices containing the largest and smallest eigenvalues of $\mathbf{C}_x$, respectively, in a decreasing order. $\mathbf{U}_1 \in \mathbb{R}^{p\times d}$ and $\mathbf{U}_2 \in \mathbb{R}^{p\times(p-d)}$ are the corresponding eigenvectors. Then the subspace of the minor principal components is spanned by the column vectors in $\mathbf{U}_2$. By introducing the identity $\mathbf{I} \equiv \mathbf{U}\mathbf{U}^{\mathrm{T}} = \mathbf{U}_1\mathbf{U}_1^{\mathrm{T}} + \mathbf{U}_2\mathbf{U}_2^{\mathrm{T}}$ and noting that $\mathbf{T}$ is equal to $\mathbf{U}_1$, the residual vector in equation (8.34) reduces to

$$\mathbf{e} = \mathbf{U}_2\mathbf{U}_2^{\mathrm{T}}\mathbf{x} \tag{8.37}$$

This shows that residual vector is essentially the projection of the original feature vector $\mathbf{x}$ onto the subspace of the minor principal components. In this way it is possible to derive features that are insensitive to environmental factors yet sensitive to structural damage [13]. It must be noted that the dimension of the feature vector should be large enough in order to ensure some separability between the damage effects and the environmental effects [14].

8.3.4 Application

The PCA-based filtering method will be applied to the data of the Zwartewaterbrug and compared to the results obtained in Chapter 7, in order to illustrate that the MSD is indeed able to filter out the environmental variability in the data, in much the same way as the method based on linear PCA.

Consider again the feature vector that was obtained in Chapter 7 from the data of bridge segment two (segment in which the added mass is located). The feature vector is made of five natural frequencies ($p = 5$) as shown in Figure 7.1. The available data is listed in Table 8.4. The first $N = 161$ samples (comprising data from the 0 kg and 25 kg class) are considered as training data for the computation of the covariance matrix $\mathbf{C}_x$ and the mean vector $\bar{\mathbf{x}}$, and the total set consist of 496 samples. As mentioned before, the 25 kg case is also used for training in order to broaden the range of environmental conditions in the training data. It is expected that the 25 kg does not cause noticeable variation in the natural frequencies, which is reasonable because this mass accounts for only 0.0625% of one instrumented segment. Therefore, it is reasonable to assume that the samples from the 25 kg case represent the healthy state of the structure. The pairwise correlation between the five natural frequencies is plotted in Figure 8.12. This shows that the natural frequencies varied considerably and that they are more or less linearly correlated. The latter implies that changes in environmental or operational variables have linear effect on the natural frequencies. It is suspected that the large variations are mainly due to changes in environmental variables, because the operational variability have been reduced through similarity filtering. Moreover, a strong correlation between temperature and natural frequency was found previously. The MSD for each of the 496 feature vectors is computed according to equation (8.25), and results were shown in Chapter 7.

Table 8.4: Added mass cases and corresponding data samples.

	H*	50 kg	75 kg	100 kg
Size	161	139	88	108
First sample no.	1	162	301	389
Last sample no.	161	300	388	496

* Comprises 67 samples of 0 kg class and 94 samples of 25 kg.

In order to illustrate the correspondence between the MSD and PCA-based data normalization, a new set of feature vectors is constructed by computing the residual vectors according to equation (8.34). Alternatively, the new set of features could be obtained by projecting the feature vectors onto the subspace of minor principal components according to (8.37). It is noted that all the feature vectors are normalized by removing the mean vector of the training data, before application of the PCA.

The PCA involves the decomposition of the covariance matrix according to (8.28), resulting in five eigenvalues and corresponding orthonormal eigenvectors. Each one of the eigenvalues indicate the amount of

variance that is explained by the corresponding principal component. Figure 8.10 shows the normalized variance associated with each principal component, which is computed as follows

$$\lambda_k / \sum_{j=1}^{5} \lambda_j \cdot 100\%, \qquad k = 1,2,\ldots,5$$

Obviously, most of the variance is explained by the first principal component (93%), the last two components explains only 0.2% of the variance in the original data, and it is assumed that these eigenvalues are only non-zero due to the noise in the training data. On the other hand, it is assumed that the variability explained by the first three components (99.8%) can be attributed to the unobserved environmental factors (and possible other factors, but not related with the deterioration of the structure). Consequently, the matrix $\mathbf{T}$ in equation (8.34) has dimensions 5×3 and its columns are the three eigenvectors corresponding to the three largest eigenvalues. Figure 8.13 shows the pairwise correlation of the five components in the residual vectors. By comparing this figure with Figure 8.12, it can be observed that the residuals are less sensitive to the variations in the training data yet still sensitive to the added masses. An exception to this observation is the data of the 50 kg class, for which the samples do not cluster at one location. This is probably because the training data used does not cover the full range of environmental conditions or there may be other variables that also affect the natural frequencies, but their effect was not reflected by the training data. In both cases, the PCA-based normalization will be unable to filter out the variations that are the result of (environmental) conditions that are not included in the training data. Another reason might be that the changes in the latent variables have nonlinear effect on the natural frequencies rather than the assumed linear effect. In particular, it was already observed that the relationship between the temperature and natural frequency is lightly nonlinear, especially for data from different days (Figures 8.2, 8.4 and 8.6). It is also emphasized that the observed separability between the mass classes can still be the result of changes in the environmental conditions, because of the same reasoning just mentioned. Figure 8.14 shows the normalized natural frequencies/features in a more convenient form, which can be compared directly with Figure 7.1. The normalized vectors are obtained by adding the mean vector of the training data to the residual vectors. In Figure 8.11, the MSDs of the original feature vectors and the normalized vectors are compared. It appears that PCA do not result in a major improvement of the damage detection. This could be expected due to the significant drop in the eigenvalues which essentially reduces the full MSD to the MSD of the feature vectors $\mathbf{x}$ projected on the minor principal components, which is equivalent to the MSD of the residual vectors.

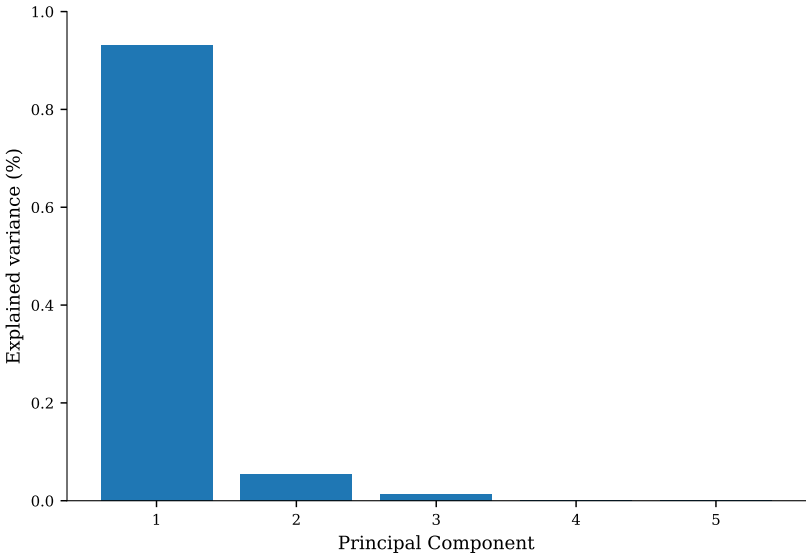


Figure 8.10: The explained variance by each one of the principal components resulting from the PCA. The variances are normalized by the total variance in the features.

In conclusion, when using a proper training set, the MSD is able to filter out environmental effects while keeping a high sensitivity to structural changes (e.g. damage or added mass). It is important that the training

set covers a full range of environmental conditions and that the feature vector is large enough in order to ensure some separability between the damage effects and the environmental effects. For the considered data is was found that the environmental changes have a more or less linear effect on the features, there are however situations in which the environmental or operational variations have a strong nonlinear effect on the features. For example, as the temperature falls below zero, its influence on the features can change abruptly [37]. This usually results in a nonlinear correlation structure between the features. For these situations one can use a nonlinear model or a mixture of linear models to compensate the nonlinear effects [30, 45, 57].

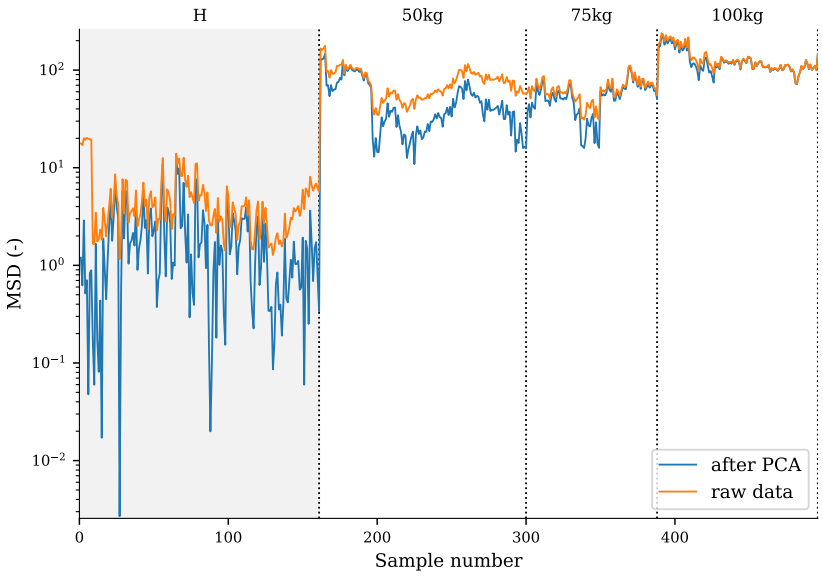


Figure 8.11: Comparison of the MSD obtained from the raw features (without normalization) and the PCA-based normalized features. The first 161 samples are used for training the model.

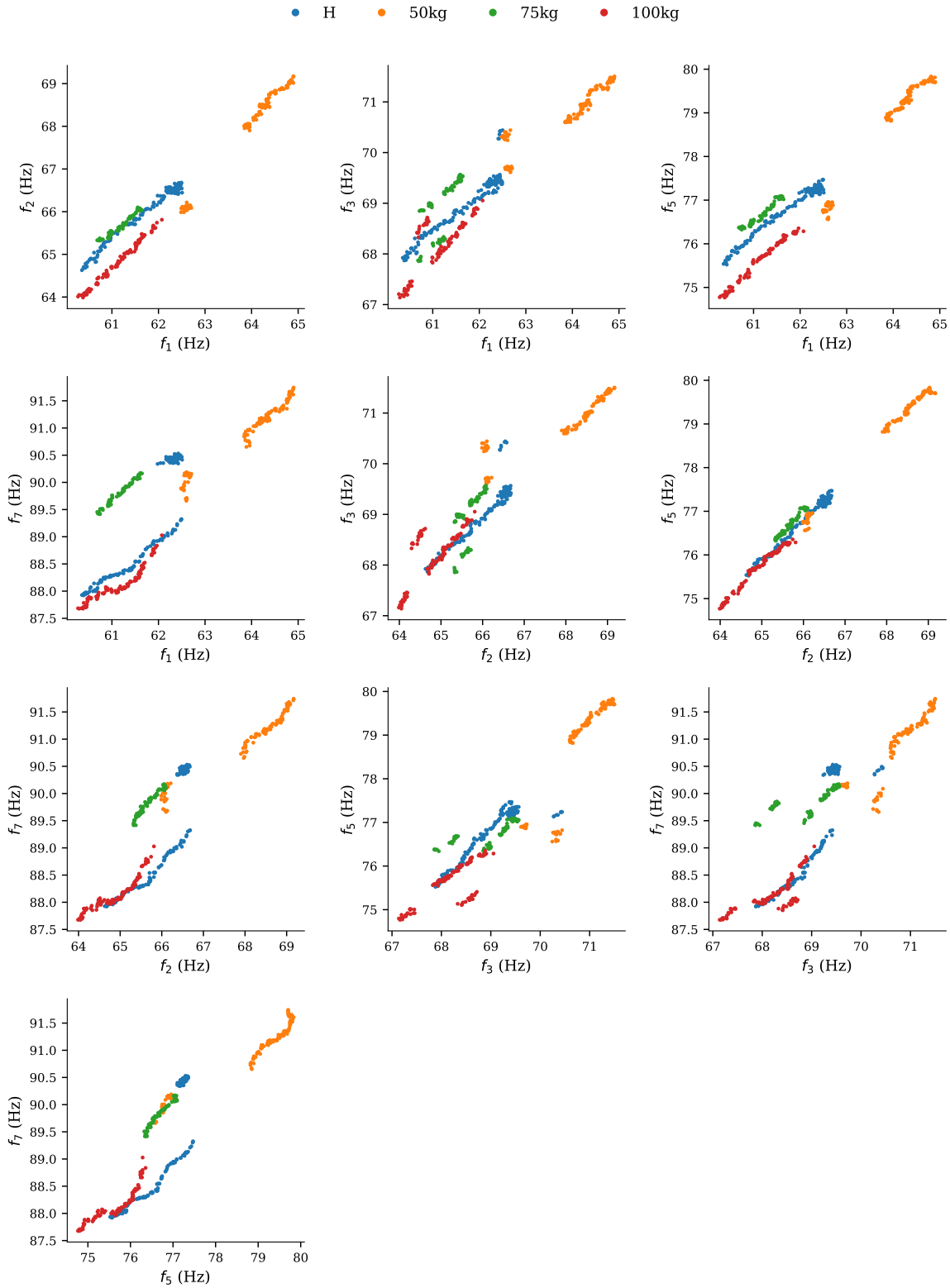


Figure 8.12: Pairwise correlation between five natural frequencies of the second bridge segment.

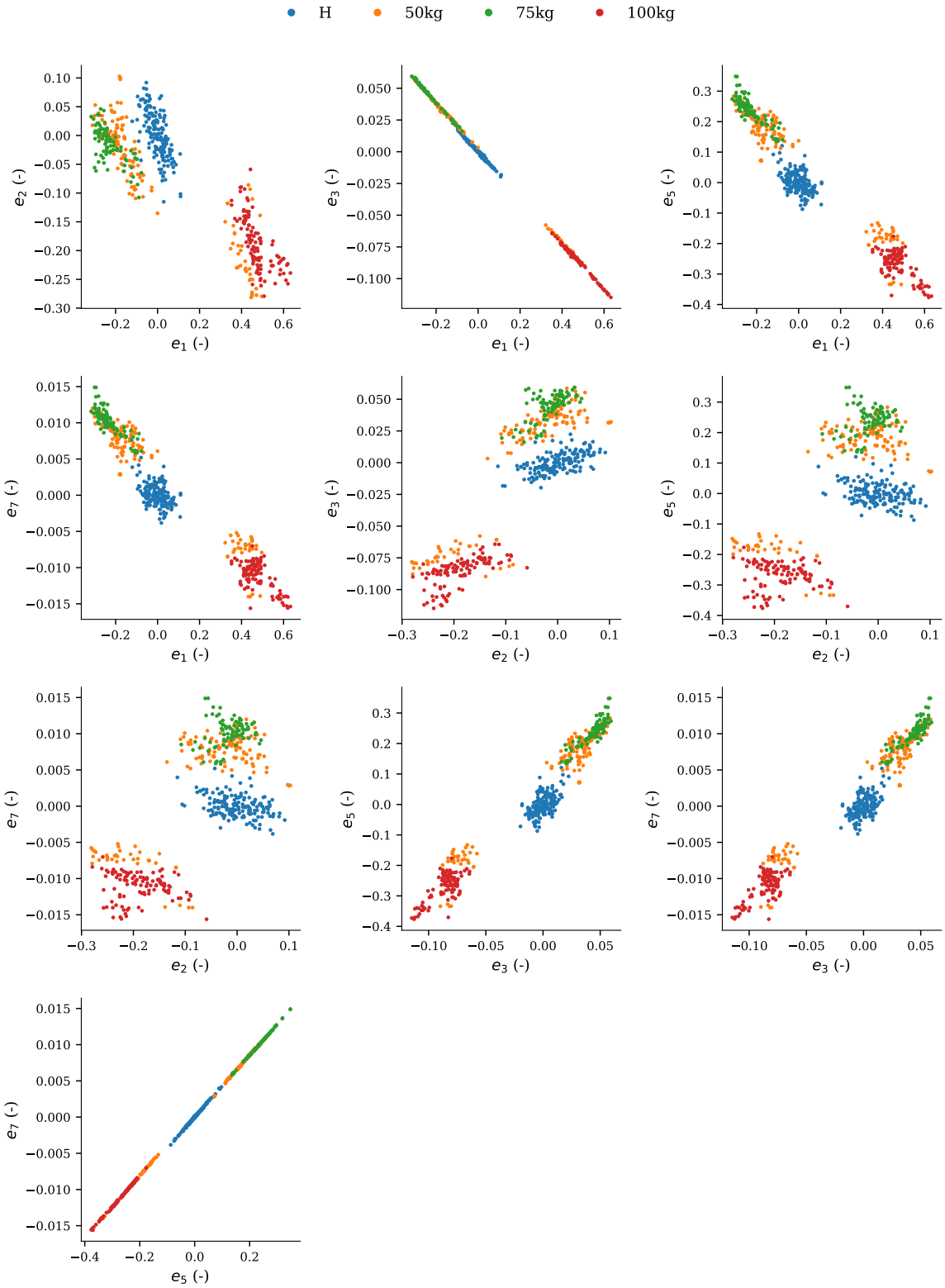


Figure 8.13: Pairwise correlation between the residuals after PCA-based data normalization.

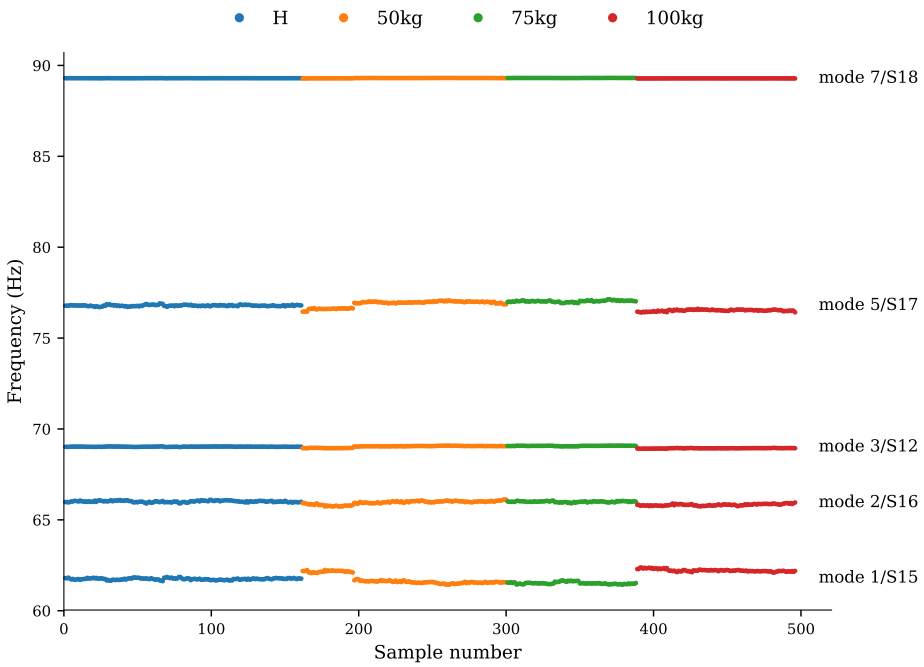


Figure 8.14: Natural frequencies/ features after PCA-based data normalization. These are obtained by adding the mean vector of the training data to the residual vectors.

8.4 Chapter Summary

In this chapter the importance of data normalization and its application was illustrated with some literature studies and the data from the Zwartewaterbrug.

First, some studies were summarized that reported the influence of the environmental variability on the dynamic properties of the structure. Most studies consider the effect of temperature variations on the natural frequency, mode shape and damping ratio of a structure. Some important points that were found in the studies are:

- The natural frequencies of a structure decrease for increasing structural temperature.
- When the temperature drops below freezing point, the natural frequencies are significantly affected due to freezing of the supports and the increased stiffness of the asphalt layer.
- Thermally induced stresses were regarded as the main reason for the reduction of the natural frequencies of heated clamped plates.
- Theoretically it can be shown that the natural frequency decreases when temperature is increased due to two factors: 1) variations in material properties (i.e. for most engineering materials the modulus of elasticity decreases for increasing temperature), and 2) thermally induced compression forces resulting from constrained expansion of the structure.
- The changes in the dynamic properties caused by variations in environmental and operational conditions may mask the changes caused by structural damage, which makes damage detection impossible without proper data normalization.
- When the correlation between temperature and the natural frequency is modeled with a regression or interpolation model it is important to consider the thermal inertia of the structure (i.e. the time lag information) as well as the temperature distribution across the structure (i.e. the spatial temperature data). Which one of these variables is more important depends on the structure being analyzed.

Next, the temperature dependency of the natural frequencies of the Zwartewaterbrug was analyzed. In particular, the correlation between the ambient temperature and the natural frequencies of the four most dominant modes of each segment was analyzed. In short the following was observed:

- The natural frequencies generally increase with increasing temperature, which is not in line with the studies in literature. No reasonable explanation could be found for this deviating behavior based on the available data.
- A strong positive correlation is found between the temperature and natural frequency for a single mass class, correlation is less strong when considering all available data. In the latter case the relative frequency differences between 6.76% and 12.21%.
- A linear model is not suitable for modeling the relation between temperature and natural frequency over multiple days. The limited temperature data from a single location and the relatively small dataset of vibration measurements are the main difficulties for the analysis. Moreover, the vibration measurements (including different added mass) cover a relatively wide range of temperatures, but there are only a few observations per temperature which is insufficient for a reliable statistical analysis.

Furthermore, it was mentioned that the changes in the natural frequencies due to the temperature variations are more prominent than the changes caused by the added mass. This was supported by the following two observations:

- The variations in natural frequency are very similar for all modes and no obvious deviations can be found for the modes “close” to the added mass (i.e. modes with highest mass sensitivity).
- The variations in natural frequency that can be observed within the dataset of a particular mass class will not be the result of the added mass.

At several points in this thesis it is observed that the effects of the environmental variabilities on the features is significant. These effects should be removed by proper data normalization before any damage detection algorithm is employed.

Two classes of methods for data normalization were discussed. The first class consist of regression and interpolation methods which require monitoring of the environmental variables of interest, which might be impractical or difficult to achieve. The second class consist of decomposition methods which overcome the difficulties of direct measurements of the environmental variables. The second class of methods is more appealing because it can remove the effects caused by multiple variables without actually measuring these variables. The Principal Component Analysis (PCA) and Factor Analysis (FA) are two most commonly used decomposition methods for data normalization.

The PCA for data normalization was applied to the data of the Zwartewaterbrug and it was illustrated that the MSD used in Chapter 7 is able to filter out the environmental variability in much the same way as the method based on linear PCA. For this it is important that the training set is obtained under a full range of environmental conditions and that the feature vector is large enough in order to ensure some separability between the damage effects and the environmental effect.

The available data of the Zwartewaterbrug is not suitable for the application of data normalization because the normal condition data (0 kg mass class) does not cover a full range of environmental conditions. In fact, this data is measured at almost constant temperature, while the data of the mass classes are obtained at different and varying temperatures. To partially overcome this issue the data from 0 kg and 25 kg mass class are considered as normal condition data. Subsequently, a PCA-based data normalization was employed. The residuals resulting from the PCA are less sensitive to the environmental variability, though still sensitive to the added masses. However, some deviations were observed as well resulting from the fact that the training dataset do not cover the full range of environmental conditions. It is noted that the PCA-based normalization will be unable to filter out the variations due to conditions that are not included in the training data.

In conclusion, the natural frequencies of the bridge are highly correlated with the ambient temperature, and the large variations in the features can be attributed to the changing temperature. To remove the variations in the features caused by changing environmental conditions it is necessary to have datasets for the healthy and damaged state of the structure that cover the full range of environmental conditions. For the current research this would mean that the data of the healthy structure (0 kg mass) should be obtained under the same environmental conditions as the data with added masses. This data is not available for the Zwartewaterbrug, hence data normalization is not possible.

9

Conclusions and Recommendations

9.1 Conclusions

The main objective of this thesis was to detect the small added masses that were attached to the bridge deck from the vibration measurements of the Zwartewaterbrug using machine learning algorithms. This research contributes to the development of a vibration-based structural health monitoring system for early detection of small structural faults. A data-based methodology was proposed that uses simple signal processing techniques, output-only system identification technique, and machine learning algorithms. Vibration data of the Zwartewaterbrug was used to test the methodology.

The main challenges of the research were related to the sensitivity of the algorithm to small structural changes (e.g. added mass of 25 kg amounts for only 0.0625% of the weight of an instrumented bridge segment), the nonstationary characteristics of the dynamic responses due to large variations in traffic load, and the limits associated with the small dataset obtained under various environmental conditions.

To increase the sensitivity of the algorithm to the small masses, the analysis was focused on the higher frequency modes rather than the lower modes of the bridge deck. The frequency range of interest was determined to be 50–100 Hz and the measurements were band-pass filtered accordingly. The large variations in traffic load may mask the presence of the small added mass. To overcome this issue the measurements were processed with a method called "similarity filtering". The aim of this method is to damp the differences between measurements and to amplify similarities, which then are considered to be related to the dynamic characteristics of the bridge deck. The results in this thesis show that the similarity filtering seems to work for its intended goal as local resonances could clearly be identified from the spectra of the processed signals.

Next, the damage sensitive features were extracted from the processed signals using the frequency domain decomposition technique. Two different types of features were used throughout the thesis: selected frequency lines from the singular value spectrum and the natural frequencies. The former were chosen for its simplicity because they can be obtained directly from the (processed) data using simple signal processing tools. The resulting feature vectors however are high-dimensional which is challenging for most machine learning algorithms, especially when the number of observations is small. Therefore, the natural frequencies were used as alternative features (for novelty detection) because of their low-dimensionality. Different sets of features were determined for the three bridge segments, allowing to investigate the mass sensitivity of the different bridge segments.

As a final step, these features were used to detect the presence of the added mass on the structure. To achieve this, two machine learning techniques were applied: a supervised learning technique (i.e. support vector machine classification) and an unsupervised learning technique (i.e. novelty detection). Although the results of both techniques suggest that the added mass could be detected, the results were not reliable due to the large variations that exist in the feature vectors of a mass class.

These variations are attributed to the changing environmental conditions over the measurement period. A strong positive correlation was found between the ambient temperature and the natural frequencies of the bridge. The positive correlation is not in line with the literature, where mostly a negative correlation was reported. No explanation could be found for this deviating behavior with the available data of the Zwartewaterbrug. Furthermore, it is concluded that the changes in the natural frequencies caused by the temperature

variations are more prominent than the changes caused by the added mass. Therefore, for the case considered, it is impossible to detect the added masses with the proposed strategy. This conclusion is further supported by the following observations:

- The change of natural frequency over time is very similar for all identified local modes. In other words, there is no indication that local modes "close" to the added mass are more sensitive the added mass.
- The SVM classification is performed for the three bridge segments separately, and the classification results are the comparable for all segments. The same applies for the SVM classification per sensor, where similar accuracy scores were found for each sensor.
- The large variations in natural frequency that can be observed within the dataset of a particular mass class will not be the result of the added mass.

So, when the feature vectors are highly sensitive to the environmental conditions the proposed strategy will not work. The changes of the features caused by environmental variabilities should be removed by proper data normalization before any damage detection algorithm could potentially be successfully employed.

Based on the conclusions above, the proposed methodology must be supplemented with data normalization to be able to detect small structural anomalies. From the large range of data normalization techniques that could be found in literature, two classes of methods for data normalization are discussed. The first class consist of regression and interpolation method which requires monitoring of the environmental variables of interest, which might be impractical or difficult to achieve. The second class consist of decomposition methods (e.g. PCA) which overcome the difficulties of direct measurements of the environmental variables. The second class of methods is more appealing because it can remove the effects caused by multiple variables without actually measuring these variables. Moreover, the second class is data-based which fits better in the framework of the proposed strategy. In general, it is necessary to have a dataset that covers the full range of environmental conditions.

An attempt was made to remove the changes in the features caused by environmental variabilities using a PCA-based normalization technique. The normalized features seem less sensitive to the environmental variability. However, some deviations were observed resulting from the fact that the dataset for the healthy structure (i.e. reference dataset) do not cover the environmental conditions that are present in the dataset of the added masses. Therefore, it is concluded that the dataset of the Zwartewaterbrug is not suitable for data normalization.

In conclusion, it was not possible to properly test the proposed methodology for damage detection using the data of the Zwartewarterbrug, mainly because the datasets were not suitable for the machine learning algorithms employed. In the following some recommendations are given for future measurement campaigns to obtain data that is more suitable for application of the methodologies employed in the present project.

9.2 Recommendations

Resulting from the research that is carried out, some recommendation are given below related to the methodology and for future measurement campaigns.

9.2.1 Recommendations related to the methodology

Similarity filtering

The method of similarity filtering is an important part of the damage detection algorithm, and the following recommendation are made:

- The theory of similarity filtering is extensively presented, however in this thesis no attention was given to the numerical aspects of the method; e.g. discrete convolution of sampled time series. *It is recommended to address more research to the optimization of the method of similarity filtering.*
- The parameters of similarity filtering were determined by means of visual inspection of the singular value spectrum; this selection procedure is far from optimal. *More appropriate approaches to select the parameters of similarity filtering should be developed.*
- The similarity filtering is based on multiple convolutions of short time signals (e.g. in this thesis 100 convolutions of 2.5 seconds were applied). This process is computationally expensive and the compu-

tation time will become very long for large datasets. *To speed up the similarity filtering process, a more efficient implementation of the method is needed.*

- So far, the method of similarity filtering is only applied to the data of the Zwartewaterbrug. The method seems to work well for this particular case, but this does not guarantee that it will work for other similar problems or cases. *It is recommended to test the method of similarity filtering with vibration data from other bridges and/or other structural parts; e.g. main girders rather than an orthotropic deck structure.*

High frequency range

It is recommended to develop an algorithm for robust selection of the high frequency region that is sensitive to small faults in the structure. To increase the sensitivity of the algorithm to the small added masses, the analysis was focused on the higher resonances rather than the lower modes of the bridge. The frequency range of interest was determined based on the number of consistent modes that could be identified from the data. The process relies on the assumption that the modal peaks in the singular value spectrum can be distinguished from the noise peaks based on the Modal Assurance Criterion (MAC). However, the MAC value also implies consistency when, for example, the inputs are primarily coherent noise, or the inputs are the result of forced excitation rather than the structural modes. Furthermore, the process involves a bandwidth over which the consistency of a possible mode is measured, it was noted that this should be frequency dependent.

Added mass

It is recommended to determine the actual effect of the added mass on the dynamic properties. In this thesis the actual effect of the added mass on the resonance frequencies is not known. Therefore, it was impossible to quantify the changes caused by the added mass and those caused by the environmental conditions.

Machine learning algorithm

In this thesis a supervised learning and an unsupervised learning algorithm were applied, both having their strengths and weaknesses. Although both methods can be used to detect the structural faults, the unsupervised learning algorithm is more appealing as it requires only data from the healthy structure (i.e. normal operating conditions) to establish the normal condition model. During monitoring newly acquired data is compared with the model and any significant deviations indicate that the structure has departed from the healthy state. On the other hand, supervised learning algorithms generally require that data from every conceivable damage state of the structure is available. This mode of learning could potentially be used for damage localization and quantification. Since the main goal is initially to detect structural faults (i.e. abnormalities), *it is suggested to focus on the unsupervised learning algorithms (i.e. novelty detection) for damage detection.*

9.2.2 Recommendations for future measurement campaign

It was concluded that the current dataset of the Zwartewaterbrug is not suitable for damage detection using the proposed methodology. One of the main problems with the dataset is that the data with and without masses are measured in different unknown environmental conditions and thus not suitable for data normalization. Therefore, it was impossible to remove the environmental variability from the feature vectors, which is necessary before any damage detection algorithm could be successfully employed. Based on the findings in this thesis some general recommendations are given for future (long-term) measurement campaigns, aiming to generate data that is more suitable for testing the methodology.

- Design the measurement setup with the intended algorithm for machine learning in mind. Supervised learning algorithms generally require that data from every conceivable damage state of the structure is available for training, and the dataset of each damage state should contain approximately the same number of measurements to avoid imbalanced datasets. On the other hand, for unsupervised learning algorithms only data from the healthy structure is required to train the algorithm. Nevertheless, it is useful to have some data from the "damaged" structure for testing the algorithm. This dataset may be smaller than for the healthy structure, because the data of the damaged structure is not used for training and unbalanced datasets is not a problem for unsupervised learning algorithms.

- Measure the response of the healthy structure (i.e. normal operating conditions) continuously (e.g. 10-minutes interval) over a sufficient long time period. These measurements are used to establish the

reference dataset that is used for data normalization (i.e. all measurements are normalized with respect to this dataset). The measurements should be obtained over the total range of possible environmental conditions, or at least the range of environmental conditions under which the damage states of the structure are measured (e.g. structure with added mass).

- Depending on the type of machine learning algorithm that will be used, the vibration data of the damaged structure (e.g. with added mass) can be obtained over shorter time periods. However, the environmental conditions in which these are measured should be approximately equal to the conditions in which the data for the healthy structure (i.e. reference dataset) is obtained.
- Measure the structural temperature at several positions of the structure to capture the nonuniform temperature distribution; e.g. at every accelerometer. It is noted that these measurements are not required when a decomposition method is used for data normalization. However, the temperature data is potentially very useful to validate the performed normalization in that case.

  The temperature data can be used to establish correlation models describing the relationship between the temperature and modal parameters (e.g. natural frequency). The thermally induced stresses were seen as one of the main factors causing changes in the natural frequencies of the structure. Therefore, even more reliable relationships can be obtained when strains are measured at different parts of the structure, in addition to temperatures.
- Use a dense sensor network in order to increase the possibility of detecting the small added mass or equivalent small damage. The expected damage mechanism is usually known beforehand so it possible to design the sensor network based on this knowledge. The expected mode shapes (e.g. from FEM analysis) of the structure can be used to determine suitable locations for the sensors.

Based on the current measurement setup of the Zwartewaterbrug (3 measurement days, several mass classes, 32 sensors and 1 DAQ system), and taking into account the above recommendations, the following changes are proposed:

- Consider two spans between the transverse beams (instead of three); one span with added mass and one without added mass (reference). Use 16 sensors per span and place them closer to each other (exact position to be determined).
- Measure the response of the structure without added mass every 6 minutes for 1-2 minutes during two consecutive days (about 240 measurements).
- Measure the response of the structure with added masses during one day (same environmental conditions as the measurements without added mass). For each added mass, it is sufficient to measure continuously for ~2 hours. Three different mass classes (e.g. 50 kg, 75 kg and 100 kg) will be adequate for testing the methodology.
- Measure the structural temperature at different location across the bridge deck, particularly across the section where the accelerometers are located. In addition, monitor the (thermal) expansion/contraction of the main structural parts in this section (notably the bridge deck, the transverse beams and the main girders) using strain gauges.

Appendices

A

Convolution & Fourier Transform

A.1 Fourier Transform

The Fourier Transform provides a method for decomposing a time history into its frequency components. Suppose that a function $x(t)$ is defined on the interval $(-\infty, +\infty)$ and satisfy the so-called Dirichlet conditions. These conditions ensure that $x(t)$ is reasonably "well-behaved"; that is, $x(t)$ is absolutely integrable, has finite number of discontinuities and has a finite number of maxima and minima. Then the following *forward* Fourier transform exist

$$X(\omega) = \int_{-\infty}^{\infty} x(t) \mathrm{e}^{-i\omega t} \, \mathrm{d}t \tag{A.1a}$$

The *inverse* Fourier transform is then

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) \mathrm{e}^{i\omega t} \, \mathrm{d}\omega \tag{A.1b}$$

Note that the factor in front of the integrals is not universal and different versions could be found in literature. However, the product of the factors should always be $1/2\pi$. Substituting the relation $\omega = 2\pi f$ in the previous defined Fourier pair one obtains the following Fourier pair in terms of cyclic frequency f

$$X(f) = \int_{-\infty}^{\infty} x(t) \mathrm{e}^{-i2\pi f t} \, \mathrm{d}t \tag{A.2a}$$

$$x(t) = \int_{-\infty}^{\infty} X(f) \mathrm{e}^{i2\pi f t} \, \mathrm{d}f \tag{A.2b}$$

Algebraic properties

Some important algebraic properties of the Fourier transform are presented, without their proofs.

Linearity For any real or complex numbers a and b, if $z(t) = ax(t) + by(t)$, then

$$Z(\omega) = aX(\omega) + bY(\omega) \tag{A.3}$$

Translation For any real number τ, if $y(t) = x(t - \tau)$, then

$$Y(\omega) = X(\omega) \mathrm{e}^{-i\omega\tau} \tag{A.4}$$

Modulation For any real number ω_0, if $y(t) = x(t) \mathrm{e}^{i\omega_0 t}$, then

$$Y(\omega) = X(\omega - \omega_0) \tag{A.5}$$

Scaling For any non-zero real number a, if $y(t) = x(at)$, then

$$Y(\omega) = \frac{1}{|a|} X\left(\frac{\omega}{a}\right) \tag{A.6}$$

The case $a = -1$ leads to the *time reversal* property, which states that if $y(t) = x(-t)$, then $Y(\omega) = X(-\omega)$.

Differentiation For any derivative of $x(t)$, if $\frac{\mathrm{d}^n x(t)}{\mathrm{d}t^n}$ exists, then

$$\mathscr{F}\left\{\frac{\mathrm{d}^n x(t)}{\mathrm{d}t^n}\right\} = (i\omega)^n X(\omega) \tag{A.7}$$

This property is very convenient for solving differential equations using Fourier transform.

A.2 Convolution

The convolution between two time signals $f(t)$ and $g(t)$ is defined as

$$f(t) * g(t) \triangleq \int_{-\infty}^{\infty} f(\tau) g(t-\tau)\,\mathrm{d}\tau \tag{A.8}$$

where the integration time variable is substituted by the variable τ. In words the convolution of two signals is defined as the integral of the product of two functions after one is reversed and shifted by amount t. The computed integral is a weighted average of the function $f(\tau)$ at the moment t where the weighting is given by $g(-\tau)$ shifted by amount t. By changing t, the weighting function emphasizes different parts of the input function.

Algebraic properties

Some important algebraic properties of the convolution are presented, without their proofs.

$$f(t) * g(t) = g(t) * f(t) \qquad \text{commutativity} \tag{A.9a}$$
$$h(t) * \big(f(t) * g(t)\big) = \big(h(t) * f(t)\big) * g(t) \qquad \text{associativity} \tag{A.9b}$$
$$h(t) * \big(f(t) + g(t)\big) = h(t) * f(t) + h(t) * g(t) \qquad \text{distributivity} \tag{A.9c}$$

Convolution Theorem

The *convolution theorem* states that under suitable conditions the Fourier transform of a convolution of two signals is the multiplication of their Fourier transforms. In other words convolution in one domain (e.g. time domain) is equivalent to multiplication in the other domain (e.g. frequency domain). Let $f(t)$ and $g(t)$ be two functions with convolution $f(t) * g(t)$, and let $\mathscr{F}\{f(t)\}$ and $\mathscr{F}\{g(t)\}$ denote their Fourier transforms. Then

$$\mathscr{F}\{f(t) * g(t)\} = k \cdot \mathscr{F}\{f(t)\} \cdot \mathscr{F}\{g(t)\} \tag{A.10}$$

where k is a constant that depends on the specific normalization of the Fourier transform.

A.3 Basic Dynamic Characteristics

In this section some important concepts and fundamental definitions related to the dynamic behavior of linear systems are introduced. More specifically we will restrict ourselves to the class of *linear time-invariant* or *LTI* systems. A system is time-invariant if all system properties are invariant with respect to time. For example, a simple mass-spring system would be a time-invariant system if the mass and stiffness do not change from one time to another. A system is linear if the *principle of superposition* holds. Superposition can be defined by the following properties

$$f(x_1 + x_2) = f(x_1) + f(x_2) \qquad \text{additivity} \tag{A.11}$$
$$f(cx) = cf(x) \qquad \text{homogeneity} \tag{A.12}$$

where $f(x)$ denotes the response due to an input x and c is an arbitrary constant.

In words *Additivity* means that the output to a sequence of inputs is equal to the sum of outputs produced by each input individually. *Homogeneity* means that the output produced by a constant times the input is equal to the constant times the output produced by the input alone.

The Impulse Response Characteristic

The dynamic characteristic of a linear time-invariant system can be described by an *impulse response function*, denoted as $h(t)$, which is defined as the output of a system to a unit impulse input applied at time $t = 0$. The usefulness of the impulse response function as a description of the system is the following; for any arbitrary input $x(t)$, the system output $y(t)$ can be computed as a weighted sum over the entire history of the input $x(t)$. The impulse response act as *weighting function* and the infinite sum is given by the convolution integral

$$y(t) = h(t) * x(t) = \int_{-\infty}^{\infty} h(\tau)x(t-\tau)\,\mathrm{d}\tau \tag{A.13}$$

A linear time-invariant system is *physically realizable* or *causal* if it only responds to past inputs. This implies that

$$h(t) = 0 \qquad \text{for } t < 0 \tag{A.14}$$

Hence, for physical systems, the effective lower limit of integration in equation (A.13) is zero rather than $-\infty$.

A system is said to be *stable* if every possible bounded input function produces a bounded output function. It can be proven that if the impulse function $h(t)$ is absolutely integrable; that is

$$\int_{-\infty}^{\infty} |h(t)|\,\mathrm{d}t < \infty \tag{A.15}$$

then the output will be bounded and the system is stable. A proof of this is presented in [2].

The Frequency Response Characteristic

If a linear time-invariant system is physically realizable and stable, then the dynamic characteristic of the system can be described by a *frequency response function*, denoted as $H(\omega)$, which is defined as the Fourier transform of the impulse response function $h(t)$; that is

$$H(\omega) = \int_{-\infty}^{\infty} h(t)\mathrm{e}^{-i\omega t}\,\mathrm{d}t \tag{A.16}$$

Note that the lower limit of integration is effectively zero because of the causality property (A.14). An important relationship for the frequency response function is obtained by the convolution theorem, which states that convolution in the time domain is equivalent to multiplication in the frequency domain and vice verse, see Appendix A.2. Let $X(\omega)$ denote the Fourier transform of an input $x(t)$ and $Y(\omega)$ the Fourier transform of the resulting output $y(t)$, assuming these transforms exist, then from equation (A.13) and by virtue of the convolution theorem it follows

$$Y(\omega) = H(\omega)X(\omega) \tag{A.17}$$

So, the response of a system in the frequency domain is given by a simple algebraic expression rather than the convolution operation in the time domain.

The frequency response function is generally a complex-valued quantity that can be written in complex polar notation as

$$H(\omega) = |H(\omega)|\mathrm{e}^{i\varphi(\omega)} \tag{A.18}$$

where $|H(\omega)|$ is a real-valued amplitude and $\varphi(\omega)$ is the associated phase angle. The interpretation is as follows; if the system is subjected to a sinusoidal input with a frequency ω_0, then the output will also be sinusoidal with the same frequency. The ratio of the output amplitude to the input amplitude is equal to $|H(\omega_0)|$, and the phase shift between the output and the input is equal to the $\varphi(\omega_0)$ (both evaluated at the frequency of the input).

B

Linear Algebra

B.1 Singular Value Decomposition

This appendix provides a minimum explanation of the Singular Value Decomposition (SVD), which has been extensively applied in this work. More information about SVD can be found in almost any book about Linear Algebra, for instance [40].

In general, every *symmetric* matrix $\mathbf{A}$ can be decomposed as $\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}^{\mathrm{T}}$, where $\mathbf{P}$ is an orthogonal matrix and $\mathbf{D}$ is a diagonal matrix displaying the eigenvalues of $\mathbf{A}$. If $\mathbf{A}$ is not symmetric, such a decomposition is not possible, but we may still be able to decompose a square matrix $\mathbf{A}$ as $\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}^{-1}$, where $\mathbf{D}$ is as before, but $\mathbf{P}$ is now an invertible matrix. However, not every matrix has an invertible matrix $\mathbf{P}$ such that $\mathbf{D} = \mathbf{P}^{-1}\mathbf{D}\mathbf{P}$ is diagonal.

The SVD generalizes above concepts from square matrices to any kind of matrix (symmetric or not, square or not). In particular, any (real or complex) matrix $\mathbf{A} \in \mathbb{C}^{m\times n}$ can be decomposed as follows

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}} \tag{B.1}$$

where $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m) \in \mathbb{C}^{m\times m}$ and $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n) \in \mathbb{C}^{n\times n}$ are unitary matrices holding the left and right singular vectors, respectively, and $\mathbf{\Sigma} \in \mathbb{C}^{m\times n}$ is a "diagonal" matrix holding the singular values. The superscript $(\bullet)^{\mathrm{H}}$ denotes the Hermitean conjugate (complex conjugate transpose); if $\mathbf{A}$ is real, this simply becomes the superscript $(\bullet)^{\mathrm{T}}$.

It is conventional to arrange the singular values of $\mathbf{A}$ in descending order as

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0 \quad , \quad p = \min(m, n)$$

The rank r of matrix $\mathbf{A}$ is equal to the number of nonzero singular values (obviously $r \leq \min(m, n)$). Suppose $r < p$, then the last $p - r$ singular values are zero; that is $\sigma_{r+1} = \sigma_{r+2} = \cdots = \sigma_p = 0$, and the SVD can be partitioned as

$$\mathbf{A} = \begin{pmatrix}\mathbf{U}_r & \mathbf{U}_0\end{pmatrix} \begin{bmatrix}\mathbf{\Sigma}_r & \mathbf{O} \\ \mathbf{O} & \mathbf{O}\end{bmatrix} \begin{pmatrix}\mathbf{V}_r^{\mathrm{H}} \\ \mathbf{V}_0^{\mathrm{H}}\end{pmatrix} = \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^{\mathrm{H}} \tag{B.2}$$

in which each matrix $\mathbf{O}$ is a zero matrix of the appropriate size (if $r = m$ or $r = n$, some of these will not appear) and $\mathbf{\Sigma}_r$ is defined as

$$\mathbf{\Sigma}_r = \begin{bmatrix}\sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_r\end{bmatrix} \tag{B.3}$$

In many applications it is useful to rewrite equation (B.1) in the *outer product form*, that is

$$\mathbf{A} = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^{\mathrm{H}} + \cdots + \sigma_r \mathbf{u}_r \mathbf{v}_r^{\mathrm{H}} = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\mathrm{H}} \tag{B.4}$$

Here, only the singular vectors associated with nonzero singular values are included. If the singular values die off quickly, then it may be possible to accurately reconstruct the original matrix $\mathbf{A}$ using only the singular

vectors associated with the first few singular values (i.e. the largest singular values). This leads to the idea of *truncated* SVD, where the upper limit in equation (B.4) is changed to $k < r$. The integer k is chosen such that $\sigma_{k+1}, \ldots, \sigma_r \approx 0$.

The connection between eigenvectors, eigenvalues from an Eigenvalue Decomposition (EVD) and SVD is the following. For an arbitrary (real or complex) matrix $\mathbf{A}$, if $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{H}}$, then

$$\begin{aligned} \mathbf{A}^{\mathrm{H}}\mathbf{A} &= \mathbf{V}\boldsymbol{\Sigma}^{\mathrm{H}}\mathbf{U}^{\mathrm{H}}\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{H}} \\ &= \mathbf{V}\boldsymbol{\Sigma}^{\mathrm{H}}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{H}} \\ &= \mathbf{V}\mathbf{D}\mathbf{V}^{\mathrm{H}} \end{aligned} \tag{B.5}$$

where $\mathbf{D} = \boldsymbol{\Sigma}^{\mathrm{H}}\boldsymbol{\Sigma} = \boldsymbol{\Sigma}^2$ is a diagonal matrix containing the squared singular values. From equation (B.5) it follows that

$$\mathbf{A}^{\mathrm{H}}\mathbf{A}\mathbf{V} = \mathbf{V}\mathbf{D} \tag{B.6}$$

which we can recognize as an EVD of $\mathbf{A}^{\mathrm{H}}\mathbf{A}$ and thus the eigenvectors of $\mathbf{A}^{\mathrm{H}}\mathbf{A}$ are equal to $\mathbf{V}$, the right singular vectors of $\mathbf{A}$, and the eigenvalues of $\mathbf{A}^{\mathrm{H}}\mathbf{A}$ are equal to $\mathbf{D}$, the squared singular values. Similarly, the left singular vectors, $\mathbf{U}$, are obtained from the eigendecomposition of $\mathbf{A}\mathbf{A}^{\mathrm{H}}$

$$\mathbf{A}\mathbf{A}^{\mathrm{H}}\mathbf{U} = \mathbf{U}\mathbf{D} \tag{B.7}$$

So, the symmetric matrices $\mathbf{A}\mathbf{A}^{\mathrm{H}}$ and $\mathbf{A}^{\mathrm{H}}\mathbf{A}$ share the same real *nonnegative* eigenvalues, equal to the squared singular values of $\mathbf{A}$.

B.2 Kronecker Notation

The Kronecker product and its most important properties are briefly described below. The properties of the Kronecker product are given without proof. Complete proofs of each of the properties can be found in [24] and [4].

The Kronecker product is denoted by $\otimes$ and defined as follows: if $\mathbf{A}$ is an $(n \times m)$ rectangular matrix and $\mathbf{B}$ is an $(p \times q)$ rectangular matrix, then $\mathbf{C} = \mathbf{A} \otimes \mathbf{B}$ is a rectangular matrix of dimension $(np \times mq)$ with components $C_{p(j_1-1)+j_2,q(l_1-1)+l_2} = A_{j_1 l_1} B_{j_2 l_2}$. More explicitly, this relationship can be written as

$$\mathbf{C} = \mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} A_{11}\mathbf{B} & A_{12}\mathbf{B} & \ldots & A_{1m}\mathbf{B} \\ A_{21}\mathbf{B} & A_{22}\mathbf{B} & \ldots & A_{2m}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1}\mathbf{B} & A_{n2}\mathbf{B} & \ldots & A_{nm}\mathbf{B} \end{bmatrix} \tag{B.8}$$

Note that in the particular case in which $\mathbf{A}$ and $\mathbf{B}$ are both vectors, we find that $\mathbf{A} \otimes \mathbf{B}$ is also a vector. If $\mathbf{a}$ is an $(n \times 1)$ column vector and $\mathbf{b}$ is an $(p \times 1)$ column vector, then $\mathbf{c} = \mathbf{a} \otimes \mathbf{b}$ is a column vector with dimensions $(np \times 1)$;

$$\mathbf{c} = \mathbf{a} \otimes \mathbf{b} = \begin{pmatrix} a_{11}\mathbf{b} \\ a_{21}\mathbf{b} \\ \vdots \\ a_{n1}\mathbf{b} \end{pmatrix} \tag{B.9}$$

This situation will be of particular interest in our case, because most of the equations used involve vectors. Note that the Kronecker product is much more general than matrix multiplication and requires no restrictions on the dimensions of the arrays being multiplied. The Kronecker product fulfills the following linearity properties

$$\mathbf{A} \otimes (\mathbf{B} + \mathbf{C}) = \mathbf{A} \otimes \mathbf{B} + \mathbf{A} \otimes \mathbf{C} \tag{B.10a}$$

$$(\mathbf{A} + \mathbf{B}) \otimes \mathbf{C} = \mathbf{A} \otimes \mathbf{C} + \mathbf{B} \otimes \mathbf{C} \tag{B.10b}$$

$$\mathbf{A} \otimes (\mathbf{B} \otimes \mathbf{C}) = (\mathbf{A} \otimes \mathbf{B}) \otimes \mathbf{C} \tag{B.10c}$$

$$\lambda\,(\mathbf{A} \otimes \mathbf{B}) = (\lambda\mathbf{A}) \otimes \mathbf{B} = \mathbf{A} \otimes (\lambda\mathbf{B}) \tag{B.10d}$$

for any matrix **A**, **B** and **C**, and constant λ, showing that the operation is distributive and associative. On the other hand $\mathbf{A}\otimes\mathbf{B} \neq \mathbf{B}\otimes\mathbf{A}$, so the operation is not commutative. A simplified notation is used for the Kronecker product of a matrix with itself. In particular, the n-th *Kronecker power* of matrix **A** is defined as

$$\mathbf{A}^{[n]} = \underbrace{\mathbf{A}\otimes\mathbf{A}\otimes\cdots\otimes\mathbf{A}}_{n} \tag{B.11}$$

Some other useful properties of the Kronecker product are

$$(\mathbf{A}\otimes\mathbf{B})^{\mathrm{H}} = \mathbf{A}^{\mathrm{H}}\otimes\mathbf{B}^{\mathrm{H}} \tag{B.12a}$$

$$(\mathbf{A}\otimes\mathbf{B})^{-1} = \mathbf{A}^{-1}\otimes\mathbf{B}^{-1} \quad \text{if } \mathbf{A} \text{ and } \mathbf{B} \text{ are invertible} \tag{B.12b}$$

$$\mathbf{A}^{[n+1]} = \mathbf{A}^{[n]}\otimes\mathbf{A} \tag{B.12c}$$

$$(\mathbf{AB})^{[n]} = \mathbf{A}^{[n]}\otimes\mathbf{B}^{[n]} \tag{B.12d}$$

Because the relations for multiple-input systems contain matrix products, we also need a general relationship involving the combination of matrix products and Kronecker products. This *mixed-product* property is

$$(\mathbf{AB})\otimes(\mathbf{CD}) = (\mathbf{A}\otimes\mathbf{C})\cdot(\mathbf{B}\otimes\mathbf{D}) \tag{B.13}$$

provided that the dimensions of the matrices are such that the products **AB** and **CD** exist. In the particular case that one of these products is a scalar (which is only possible if the product is essentially a vector product), we find that $(\mathbf{AB})\otimes(\mathbf{CD}) \equiv (\mathbf{AB})\cdot(\mathbf{CD})$. Hence, we can define the following *mixed-product property for vectors*

$$\left(\mathbf{a}^{\mathrm{T}}\mathbf{b}\right)\cdot\left(\mathbf{c}^{\mathrm{T}}\mathbf{d}\right) = (\mathbf{a}\otimes\mathbf{c})^{\mathrm{T}}\cdot(\mathbf{b}\otimes\mathbf{d}) \tag{B.14}$$

in which **a**, **b**, **c** and **d** are column vectors which dimensions are such that the products $\mathbf{a}^{\mathrm{T}}\mathbf{b}$ and $\mathbf{c}^{\mathrm{T}}\mathbf{d}$ exist. This latter relation is most important for the notations used in the theory about similarity filtering.

C

Principal Component Analysis

This appendix provides a minimum explanation of the Principal Component Analysis (PCA), which has been used in this work for data normalization and visualization. More information about the PCA can be found in many textbooks about pattern recognition or machine learning, for example [46] or [19]. Some good examples on real data sets can be found in [47].

The feature vectors used in machine learning are often high-dimensional, which are hard to visualize and could cause problems in case of small data sets. Therefore, one often seeks to reduce the number of features without loosing important information. The PCA is one of the most popular methods for dimensionality reduction in pattern recognition. Furthermore, PCA is very useful in producing compact representations of the original high-dimensional feature space, which makes it also very suitable for visualization purposes.

The PCA uses a linear transformation to transform a set of possibly correlated variables $\{\mathbf{x}_i\} \in \mathbb{R}^p$ into a new set of uncorrelated features $\{\mathbf{y}_i\} \in \mathbb{R}^d$ called the principal components. This transformation is defined in such a way that the first principal component has the largest possible variance, and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors are a linear combination of the original variables and form an orthogonal basis set.

Mutually uncorrelated features

Let $\{\mathbf{x}_i\} \in \mathbb{R}^p$, $i = 1, \ldots, n$ be the set of training vectors. In order to simplify the notations, we will assume that the variables have zero mean, that is $\mathbf{E}[\mathbf{x}] = \mathbf{0}$. If this is not the case, we can always subtract the mean value. In practice, this is achieved by subtracting the *sample mean* $\bar{\mathbf{x}}$, which is computed from the set of training vectors as

$$\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \tag{C.1}$$

The PCA exploits the following linear transformation

$$\mathbf{y}_i = \mathbf{A}^{\mathrm{T}} \mathbf{x}_i \tag{C.2}$$

where $\mathbf{A} \in \mathbb{R}^{p \times d}$ is the transformation matrix (to be determined) and $\mathbf{y}_i \in \mathbb{R}^d$ is a vector of uncorrelated features; that is, its covariance matrix $\mathbf{C}_y$ is diagonal. Since we have assumed that $\mathbf{E}[\mathbf{x}] = \mathbf{0}$, it is readily seen that $\mathbf{E}[\mathbf{y}] = \mathbf{0}$. For zero mean variables the covariance matrix $\mathbf{C}_y$ coincides with the correlation matrix $\mathbf{R}_y$, and thus, from the definition of the correlation matrix, we have

$$\mathbf{R}_y = \mathbf{E}[\mathbf{y}\mathbf{y}^{\mathrm{T}}] = \mathbf{E}[\mathbf{A}^{\mathrm{T}}\mathbf{x}\mathbf{x}^{\mathrm{T}}\mathbf{A}] = \mathbf{A}^{\mathrm{T}}\mathbf{R}_x\mathbf{A} \tag{C.3}$$

In practice, the correlation matrix $\mathbf{R}_x$ is estimated as an average over the given set of training vectors $\{\mathbf{x}_i\}$, that is

$$\mathbf{R}_x \approx \frac{1}{n-1} \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^{\mathrm{T}} \tag{C.4}$$

which is known as the *sample correlation matrix*, which is an unbiased estimator of the correlation matrix, the reason the sample correlation matrix has $n-1$ in the denominator rather that n is essentially that the mean $\mathbf{E}[x]$ is not known and is replaced by the sample mean $\bar{\mathbf{x}}$, equation (C.1).

The correlation matrix $\mathbf{R}_x$ is symmetric matrix, and hence its eigenvectors are orthogonal ($\mathbf{a}_i^T\mathbf{a}_j = 0$ for $i \neq j$). Thus if the transformation matrix $\mathbf{A}$ is chosen so that its columns are the orthonormal eigenvectors $\mathbf{a}_i$, $i = 1,2,\ldots,p$ of $\mathbf{R}_x$, then $\mathbf{R}_y$ is diagonal

$$\mathbf{R}_y = \mathbf{A}^T\mathbf{R}_x\mathbf{A} = \mathbf{\Lambda} \tag{C.5}$$

where $\mathbf{\Lambda} \in \mathbb{R}^{p\times p}$ is a diagonal matrix that contain the respective eigenvalues λ_i, $i = 1,2,\ldots,p$ of $\mathbf{R}_x$ on its diagonal. It can be shown that the correlation matrix is positive semidefinite and thus the eigenvalues are nonnegative.

Now, we have found a diagonal correlation matrix $\mathbf{R}_y$ which implies that the transformed features are uncorrelated. From the respective definitions we have that the variance of the transformed features is equal to

$$\mathrm{Var}\left(y_j\right) = \mathbf{R}_y(j,j) = \lambda_j \qquad \text{for } j = 1,2,\ldots,p \tag{C.6}$$

where y_j refers to the j-th feature. Equation (C.6) shows that the eigenvalues of the correlation matrix $\mathbf{R}_x$ are equal to the variances of the transformed features.

Singular Value Decomposition

Previously, the PCA is explained by means of an eigenvalue decomposition of the correlation matrix $\mathbf{R}_x$ (or covariance matrix $\mathbf{C}_x$). Alternatively, PCA can be done by a Singular Value Decomposition (SVD) of the data matrix $\mathbf{X}$ as shown in the following. The SVD is usually applied in practice because efficient algorithms exist to calculate the SVD of $\mathbf{X}$ without having to form the matrix $\mathbf{X}^T\mathbf{X}$. The data matrix is defined as follows

$$\mathbf{X} = \begin{bmatrix}\mathbf{x}_1 & \mathbf{x}_2 & \ldots & \mathbf{x}_n\end{bmatrix}^T \tag{C.7}$$

That is, $\mathbf{X} \in \mathbb{R}^{n\times p}$ is a matrix whose rows are the available feature vectors in the set $\{\mathbf{x}_i\} \in \mathbb{R}^p$, $i = 1,\ldots,n$. The SVD of $\mathbf{X}$, with rank $r \leq \min(n,p)$ reads as (Appendix B.1)

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{C.8}$$

where $\mathbf{\Sigma} \in \mathbb{R}^{n\times p}$ contains the nonnegative real-valued singular values of $\mathbf{X}$ on its diagonal and $\mathbf{U} \in \mathbb{R}^{n\times n}$, $\mathbf{V} \in \mathbb{R}^{p\times p}$ are orthogonal matrices. The sample correlation matrix, equation (C.4), can be written as

$$\begin{aligned}\mathbf{R}_x &= \frac{1}{n-1}\mathbf{X}^T\mathbf{X} \\ &= \frac{1}{n-1}\mathbf{V}\mathbf{\Sigma}^T\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \\ &= \frac{1}{n-1}\mathbf{V}\hat{\mathbf{\Sigma}}^2\mathbf{V}^T\end{aligned} \tag{C.9}$$

where $\hat{\mathbf{\Sigma}}^2 = \mathbf{\Sigma}^T\mathbf{\Sigma} \in \mathbb{R}^{p\times p}$ is a diagonal matrix containing the squared singular values of $\mathbf{X}$.

Using the orthogonality property of the right-singular vectors ($\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}$), the correlation matrix $\mathbf{R}_x$ can be diagonalized as

$$\frac{1}{n-1}\hat{\mathbf{\Sigma}}^2 = \mathbf{V}^T\mathbf{R}_x\mathbf{V} \tag{C.10}$$

Comparing equation (C.10) with equation (C.5), it is readily seen that $\mathbf{\Lambda} = \mathbf{\Sigma}^2/(n-1)$ and $\mathbf{A} = \mathbf{V}$. Hence, using the SVD the transformation is expressed as

$$\mathbf{y}_i = \mathbf{V}^T\mathbf{x}_i \tag{C.11}$$

which can be reformulated in matrix form using the definition of the data matrix, equation (C.7), as

$$\mathbf{Y} = \mathbf{X}\mathbf{V} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{V} = \mathbf{U}\mathbf{\Sigma} \tag{C.12}$$

That is, each column of $\mathbf{Y}$ (the principal components) is given by one of the left singular vectors of $\mathbf{X}$ multiplied by the corresponding singular value. The variances of the transformed features y_j are equal to

$$\mathrm{Var}\left(y_j\right) = \frac{\sigma_j^2}{n-1} \qquad \text{for } j = 1,2,\ldots,p \tag{C.13}$$

where σ_j is the j-th singular value of the data matrix $\mathbf{X}$, and n is the number of observations (i.e. rows in the data matrix).

Dimensionality Reduction

In many cases, when the number of features is large enough, the total variability in the feature vector can be explained by a smaller number of transformed features and thereby reducing the dimensions of the feature space. Such dimensionality reduction can be very useful for visualizing and processing high dimensional data sets while still remaining as much of the variance in the data set as possible. The reduction is achieved by disregarding the transformed features with low variance, which occur when the original p variables are actually a linear combination of $d < p$ variables.

If some eigenvalues of the correlation matrix $\mathbf{R}_x$ are equal to zero, then some of the variables in the original feature vector are redundant. In practice, due to the noise and measurement uncertainty, the eigenvalues are not strictly equal to zero, but a significant drop in the eigenvalues can be observed and can be used to define the number of principal components which account for most of the variability. A practical way to determine the number d of principal components for retention is to define the following indicator [14]

$$I = \frac{\sum_{i=1}^{d} \lambda_i}{\sum_{i=1}^{p} \lambda_i} \tag{C.14}$$

and determine d as the lowest integer such that $I > e(\%)$, where e is a threshold value. The meaning of this threshold is as follows: d principal components are needed in order to explain $e(\%)$ of the total variance associated with the original observed variables. Thus, selecting those features $y_j = \mathbf{a}_j^{\mathrm{T}}\mathbf{x}$, corresponding to the d largest eigenvalues makes their sum variance $\sum_j \lambda_j$ maximum. In other words, the selected d features retain most of the total variance associated with the original features $\mathbf{x}$.

The transformed d-dimensional feature vectors are now computed as

$$\mathbf{y}_i = \mathbf{A}^{\mathrm{T}}\mathbf{x}_i \qquad \text{for } i = 1, 2, \ldots, n \tag{C.15}$$

where $\mathbf{A} \in \mathbb{R}^{p \times d}$ is a transformation matrix whose columns are the eigenvectors of the correlation matrix corresponding to the d largest eigenvalues, where d is determined according to (C.14) (e.g. such that $e = 99\%$ of the total variance in the original features is explained by the transformed features). When the vectors $\mathbf{x}_i$ are collected as the rows of matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, then the transformation can be written in matrix notation as

$$\mathbf{Y} = \mathbf{X}\mathbf{A} \tag{C.16}$$

where $\mathbf{Y} \in \mathbb{R}^{n \times d}$ is the matrix with transformed feature vectors.

Alternatively, the PCA can be performed by a SVD of the data matrix $\mathbf{X}$ as explained above. Using the SVD, the transformation is given by equation (C.12). For dimensionality reduction this equation is slightly modified as

$$\mathbf{Y} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}} \tag{C.17}$$

where $\hat{\mathbf{\Sigma}} \in \mathbb{R}^{d \times d}$ is a diagonal matrix holding the largest d singular values on its diagonal, and $\hat{\mathbf{U}} \in \mathbb{R}^{n \times d}$ contains the corresponding left singular vectors as its columns; that is

$$\hat{\mathbf{U}} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \ldots & \mathbf{u}_d \end{bmatrix}, \qquad \hat{\mathbf{\Sigma}} = \begin{bmatrix} \sigma_1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \sigma_d \end{bmatrix} \tag{C.18}$$

D

PSD decomposition

This proof is provided for completeness. Starting with the relation between the unknown inputs $\mathbf{x}(t)$ and the measured displacements $\mathbf{y}(t)$

$$\mathbf{S}_{yy}(\omega) = \mathbf{H}(\omega)\mathbf{S}_{xx}(\omega)\mathbf{H}(\omega)^{\mathrm{H}} \tag{D.1}$$

where $\mathbf{H}(\omega)$ is the force-displacement Frequency Response Function (FRF) matrix, which can be expressed as partial fraction form via poles λ_r and residue matrices $\mathbf{R}_r$

$$\mathbf{H}(\omega) = \sum_{r=1}^{M} \frac{\mathbf{R}_r}{i\omega - \lambda_r} + \frac{\mathbf{R}_r^*}{i\omega - \lambda_r^*} \tag{D.2}$$

Substitution of equation (D.2) into (D.1) yields

$$\begin{aligned}
\mathbf{S}_{yy}(\omega) &= \left(\sum_{r=1}^{M} \frac{\mathbf{R}_r}{i\omega - \lambda_r} + \frac{\mathbf{R}_r^*}{i\omega - \lambda_r^*} \right) \mathbf{S}_{xx}(\omega) \left(\sum_{s=1}^{M} \frac{\mathbf{R}_s^{\mathrm{H}}}{-i\omega - \lambda_s^*} + \frac{\mathbf{R}_s^{\mathrm{T}}}{-i\omega - \lambda_s} \right) \\
&= \sum_{r=1}^{M} \sum_{s=1}^{M} \frac{\mathbf{R}_r \mathbf{S}_{xx}(\omega) \mathbf{R}_s^{\mathrm{H}}}{(i\omega - \lambda_r)\left(-i\omega - \lambda_s^*\right)} + \frac{\mathbf{R}_r^* \mathbf{S}_{xx}(\omega) \mathbf{R}_s^{\mathrm{H}}}{\left(i\omega - \lambda_r^*\right)\left(-i\omega - \lambda_s^*\right)} \\
&\quad + \frac{\mathbf{R}_r \mathbf{S}_{xx}(\omega) \mathbf{R}_s^{\mathrm{T}}}{(i\omega - \lambda_r)(-i\omega - \lambda_s)} + \frac{\mathbf{R}_r^* \mathbf{S}_{xx}(\omega) \mathbf{R}_s^{\mathrm{T}}}{\left(i\omega - \lambda_r^*\right)(-i\omega - \lambda_s)}
\end{aligned} \tag{D.3}$$

Now, assume that the input is white noise, one has that $\mathbf{S}_{xx}(\omega) = \mathbf{S}_{xx}$. To simplify notations the following definitions are introduced

$$\mathbf{A}_{1rs} = \mathbf{R}_r \mathbf{S}_{xx} \mathbf{R}_s^{\mathrm{H}} \tag{D.4}$$

$$\mathbf{A}_{2rs} = \mathbf{R}_r \mathbf{S}_{xx} \mathbf{R}_s^{\mathrm{T}} \tag{D.5}$$

Hence, the expression for $\mathbf{S}_{yy}(\omega)$ reduces to

$$\begin{aligned}
\mathbf{S}_{yy}(\omega) = \sum_{r=1}^{M} \sum_{s=1}^{M} &\frac{\mathbf{A}_{1rs}}{(i\omega - \lambda_r)\left(-i\omega - \lambda_s^*\right)} + \frac{\mathbf{A}_{2rs}^*}{\left(i\omega - \lambda_r^*\right)\left(-i\omega - \lambda_s^*\right)} \\
&+ \frac{\mathbf{A}_{2rs}}{(i\omega - \lambda_r)(-i\omega - \lambda_s)} + \frac{\mathbf{A}_{1rs}^*}{\left(i\omega - \lambda_r^*\right)(-i\omega - \lambda_s)}
\end{aligned} \tag{D.6}$$

Since the denominator of each term has two zeros, a partial fraction expansion is possible. Heaviside cover-up method is a fast way to do the decomposition. For the first term this yields

$$\frac{\mathbf{A}_{1rs}}{(i\omega - \lambda_r)\left(-i\omega - \lambda_s^*\right)} = \frac{\dfrac{\mathbf{A}_{1rs}}{-\lambda_r - \lambda_s^*}}{i\omega - \lambda_r} + \frac{\dfrac{\mathbf{A}_{1rs}}{-\lambda_r - \lambda_s^*}}{-i\omega - \lambda_s^*} \tag{D.7}$$

Similar partial fraction expansions of the other terms, yield the following expression for $\mathbf{S}_{yy}(\omega)$

$$\begin{aligned}\mathbf{S}_{yy}(\omega) = \sum_{r=1}^{M}\sum_{s=1}^{M} &\frac{\dfrac{\mathbf{A}_{1rs}}{-\lambda_r-\lambda_s^*}}{i\omega-\lambda_r} + \frac{\dfrac{\mathbf{A}_{1rs}}{-\lambda_r-\lambda_s^*}}{-i\omega-\lambda_s^*} + \frac{\dfrac{\mathbf{A}_{2rs}^*}{-\lambda_r^*-\lambda_s^*}}{i\omega-\lambda_r^*} + \frac{\dfrac{\mathbf{A}_{2rs}^*}{-\lambda_r^*-\lambda_s^*}}{-i\omega-\lambda_s^*} \\ &+ \frac{\dfrac{\mathbf{A}_{2rs}}{-\lambda_r-\lambda_s}}{i\omega-\lambda_r} + \frac{\dfrac{\mathbf{A}_{2rs}}{-\lambda_r-\lambda_s}}{-i\omega-\lambda_s} + \frac{\dfrac{\mathbf{A}_{1rs}^*}{-\lambda_r^*-\lambda_s}}{i\omega-\lambda_r^*} + \frac{\dfrac{\mathbf{A}_{1rs}^*}{-\lambda_r^*-\lambda_s}}{-i\omega-\lambda_s}\end{aligned} \tag{D.8}$$

Partial fraction yields eight terms with four different denominators, combining the terms with the same denominator leads to

$$\begin{aligned}\mathbf{S}_{yy}(\omega) = \sum_{r=1}^{M}\sum_{s=1}^{M} &\frac{\dfrac{\mathbf{A}_{1rs}}{-\lambda_r-\lambda_s^*} + \dfrac{\mathbf{A}_{2rs}}{-\lambda_r-\lambda_s}}{i\omega-\lambda_r} + \frac{\dfrac{\mathbf{A}_{1rs}}{-\lambda_r-\lambda_s^*} + \dfrac{\mathbf{A}_{2rs}^*}{-\lambda_r^*-\lambda_s^*}}{-i\omega-\lambda_s^*} \\ &+ \frac{\dfrac{\mathbf{A}_{1rs}^*}{-\lambda_r^*-\lambda_s} + \dfrac{\mathbf{A}_{2rs}^*}{-\lambda_r^*-\lambda_s^*}}{i\omega-\lambda_r^*} + \frac{\dfrac{\mathbf{A}_{1rs}^*}{-\lambda_r^*-\lambda_s} + \dfrac{\mathbf{A}_{2rs}}{-\lambda_r-\lambda_s}}{-i\omega-\lambda_s}\end{aligned} \tag{D.9}$$

By changing the order of summation in the second and fourth term (i.e. exchange the indices) and the fact that, from their definitions, it follows that $\mathbf{A}_{1sr} = \mathbf{A}_{1rs}^{\mathrm{H}}$ and $\mathbf{A}_{2sr} = \mathbf{A}_{2rs}^{\mathrm{T}}$ (under the assumption that $\mathbf{S}_{xx}$ is real and constant), equation (D.9) reduces to the following convenient form for the modal decomposition of an output PSD

$$\mathbf{S}_{yy}(\omega) = \sum_{r=1}^{M} \frac{\mathbf{A}_r}{i\omega-\lambda_r} + \frac{\mathbf{A}_r^{\mathrm{H}}}{-i\omega-\lambda_r^*} + \frac{\mathbf{A}_r^*}{i\omega-\lambda_r^*} + \frac{\mathbf{A}_r^{\mathrm{T}}}{-i\omega-\lambda_r} \tag{D.10}$$

where $\mathbf{A}_r$ is defined as

$$\mathbf{A}_r = \sum_{s=1}^{M} \frac{\mathbf{A}_{1rs}}{-\lambda_r-\lambda_s^*} + \frac{\mathbf{A}_{2rs}}{-\lambda_r-\lambda_s} \tag{D.11}$$

E

Performance of Similarity Filtering

In this appendix the performance of similarity filtering is analyzed by visual inspection of the singular value spectrum of the processed signals. The main purpose of the analysis is to select suitable filter parameters. Therefore, the spectra corresponding to different combinations of filter parameters are considered. The processed signals are generated with data from the Zwartewaterbrug and so the results are not generally applicable. First the analysis procedure is briefly described, followed by the discussion of the results.

Similarity filtering is based on successive convolution of different time series, which can be expressed as

$$y(t,T) = \overset{n}{\underset{k=1}{\ast}} x_k(t) = x_1(t) * x_2(t) * \cdots * x_n(t) \tag{E.1}$$

where n is the order of filtering (i.e. the number of response signals used), and T denotes the length of the signal after similarity filtering. Although the time series $x_k(t)$, $k = 1,\ldots,n$ might have different lengths, it will be assumed that the signals have an predefined length of T_s seconds. It is noted that the time series might also be some segments of a long measurement, therefore the time series are also referred to as *segments*. The formulation of similarity filtering involves two free parameters: the length T_s of the segments, and the order of similarity filtering n (i.e. the number of segments). The choice of the parameters is related to the goal of similarity filtering, which is filter out the operational variability from the measurements. The following three aspects should be taken into account when selecting the parameters (see also Section 3.1.4):

1. the computational cost of similarity filtering;
2. the time scale on which similarity filtering operates;
3. the expected natural frequencies of the structure.

In the following, suitable parameters are determined by visual inspection of the results after similarity filtering. In particular the singular value spectrum is considered because this clearly indicates which frequency components are amplified. Moreover, the peaks in this spectrum are related to the modes of the considered structure.

To analyze the performance of similarity filtering, different combinations of parameters n (filter order) and T_s (segment length) are considered. The considered segment lengths are: 2.5, 4.0, 5.0, 6.0, 7.5, and 10 seconds, and the order of similarity filtering ranges from 5 to 100 in steps of 5. It is noted that the number of convolutions performed is one less than the order of similarity filtering. For each combination of parameters, the following steps are performed:

Step 1. Apply a band-pass filter 50–100 Hz to the raw time series.

Step 2. Generate one vector of processed signals using the similarity filtering with the particular combination of parameters. The vector consists of 12 signals, generated from the measurements of sensor 11 to 22 (located in bridge segment 2).

Step 3. (optional) remove the zeros at both ends of the processed signals; that is, extract the nonzero part of the signals also referred to as *centering*.

Step 4. Perform a singular value decomposition of the spectral density matrix $\mathbf{S}_{yy}(f)$ at each discrete frequency to obtain a set of singular values (only largest singular values are stored). The spectral density matrix is computed from the processed signals as $\mathbf{S}_{yy}(\omega) = \mathbf{Y}(\omega)\mathbf{Y}(\omega)^{\mathrm{H}}$ where $\mathbf{Y}(\omega) \in \mathbb{C}^{12\times 1}$ is the Fourier transform of the processed signals $\mathbf{y}(t)$. The set of singular values forms the *singular value spectrum*.

Step 5. Repeat **Step 2** to **Step 4** for all combinations of parameters.

Following these steps, 180 singular value spectra are obtained, one for each combination of parameters (i.e. 6 different time segments and 20 different number of convolutions).

It is noted that **Step 3** is an optional step after the similarity filtering, which is not related with the similarity filtering itself. In this step the dominant middle part of the processed signal is extracted, and thus disregarding the edges of the signal that are close to zero. This might have influence on the further processing of the signals; for example when computing the singular value spectrum considered here. Therefore, two options are considered: 1) continue with the length of the processed signals (regular signals), and 2) remove the parts that are approximately zero at both ends of the processed signal; the signal is assumed to be zero when it has an amplitude smaller than 1e−5 times the maximum amplitude of the signal. So, two sets of 180 singular value spectra are generated, one for the centered signals and one for the regular signals (i.e. without centering).

The evolution of the singular value spectrum as a function of the number of convolutions is visualized in two ways: 1) 3D waterfall plot of the singular value spectra, and 2) 2D heatmap of the singular value spectra, in which the darkness of the colors indicates the intensity of the frequency components. Figures E.1 and E.3 show the results for the regular signals and Figures E.2 and E.4 show the result for the centered signals. For the interpretation of the figures it must be remembered that one of the goals of similarity filtering is to amplify the similarities in the signals, which are related to the system characteristics. The peaks (i.e. dominant frequency components) in the singular value spectrum are related to the modes of the considered structure. So, an important criterion is the existence of peaks in the spectrum, and the evolution of these peaks for increasing number of convolutions.

The following can be observed from the figures:

- In general, the singular value spectrum becomes smoother when the number of convolutions are increased. However, at some point increasing the number of convolutions might not improve the spectrum any further. In fact, useful information might be lost when too many convolutions are performed. This can clearly be observed from the figures where some peaks in the spectra “disappear” when the number of convolution increases. Recall that the peaks in the singular value spectrum are related to the modes of the considered structure. So, by increasing the number of convolutions it may happen that a mode is damped rather than amplified. This will be the case if the mode is not excited during all measurements and thus is not present in every segment (i.e. the signals $x_k(t)$ in equation (E.1)) used for similarity filtering. Consequently, this mode becomes a variation rather than a similarity between the different segments and will be damped by the similarity filtering. So, it is important that the modes of interest are excited during all measurements.
- In most cases, longer segments require more convolution than short segments to obtain the same “quality” of the singular value spectrum. Together with the fact that long segments require more data than short segments it is better to use short segments (i.e. $T_s \leq 5$ s).
- By comparing the results of the centered and regular signals it can be observed that some frequency components might be lost due to the *centering* of the processed signals. In general, the amplitude of the centered signals is larger, and the spectrum looks smoother. This effect is presumably related to the computation of the spectrum using the Fast Fourier Transform (FFT). Furthermore, the spectra of the regular signals seems to be bounded at low values which is presumably a numerical artifact resulting from the large number of convolutions. The numerical aspects of the similarity filtering are not considered in this thesis, but more research into these aspects is needed.

For the application of similarity filtering in this thesis it is chosen to use 100 segments of 2.5 seconds length. Other combinations of parameters might also give acceptable results, but these are not considered.

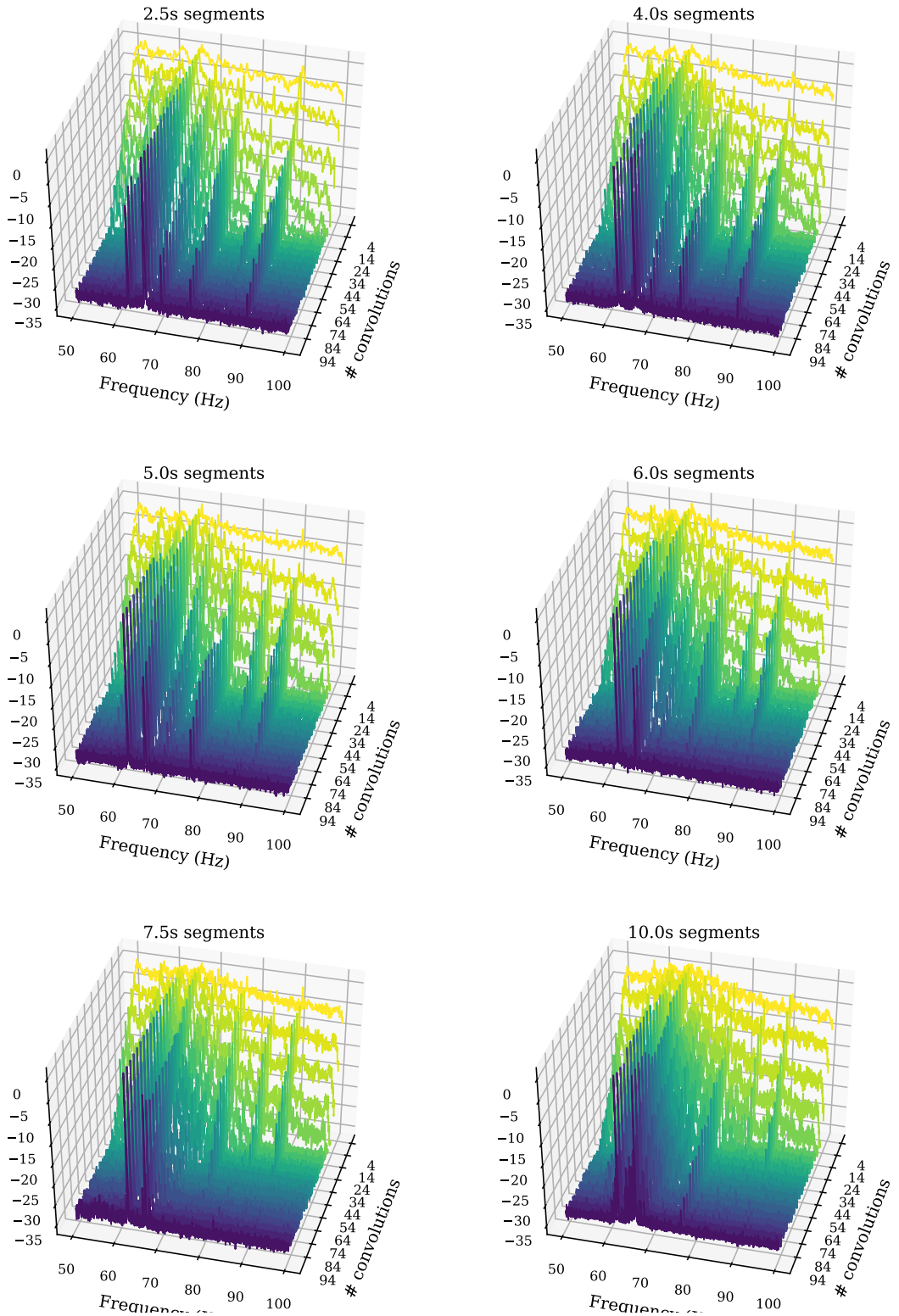


Figure E.1: Regular signals – 3D waterfall plot of the evolution of the singular value spectrum when increasing the number of convolutions in similarity filtering.

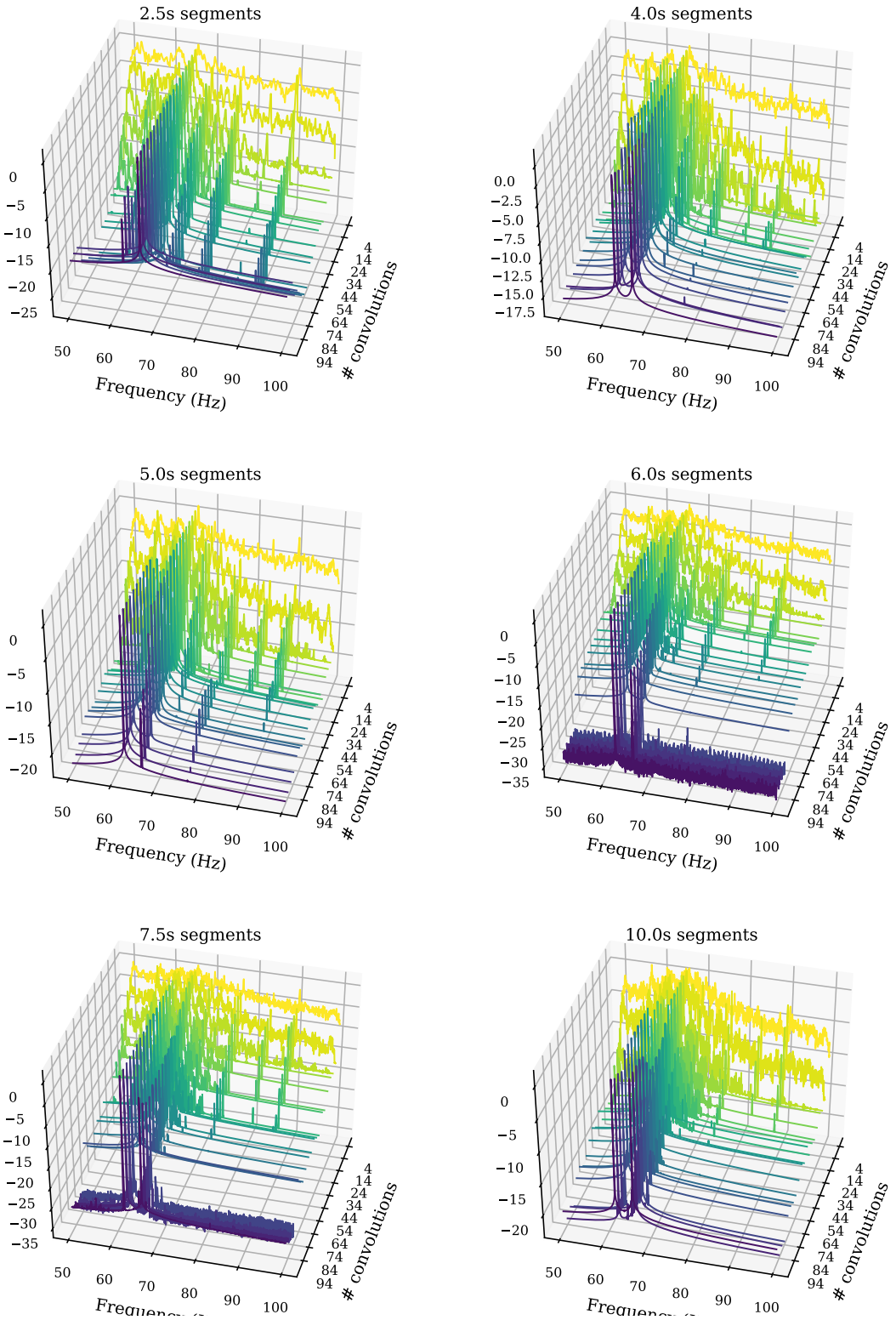


Figure E.2: Centered signals – 3D waterfall plot of the evolution of the singular value spectrum when increasing the number of convolutions in similarity filtering.

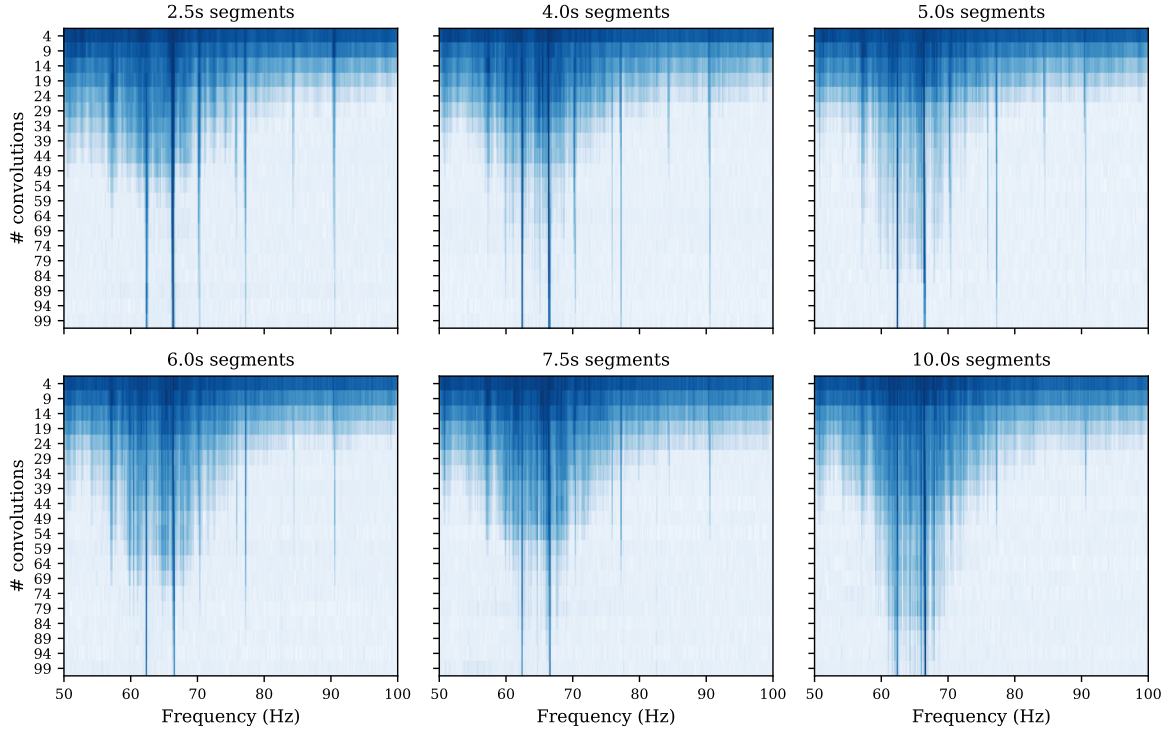


Figure E.3: Regular signals – 2D image of the evolution the of singular value spectrum when increasing the number of convolutions in similarity filtering.

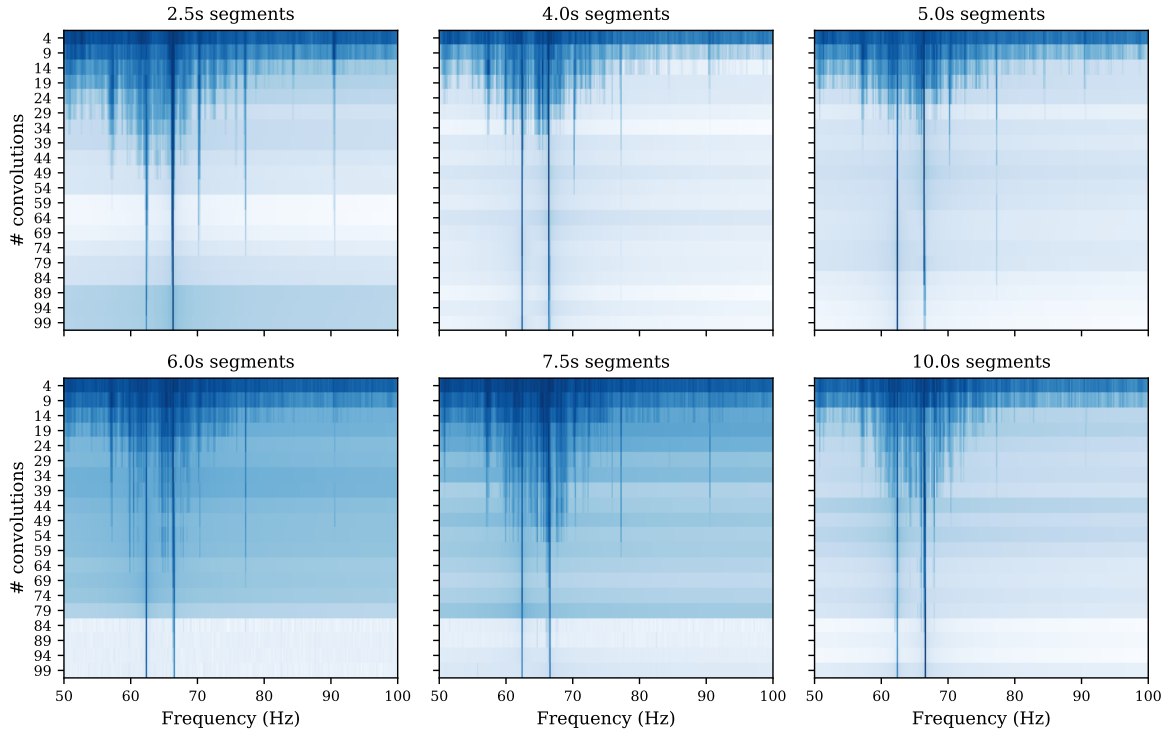


Figure E.4: Centered signals – 2D image of the evolution of the singular value spectrum when increasing the number of convolutions in similarity filtering.

F

Spectra of Processed Signals

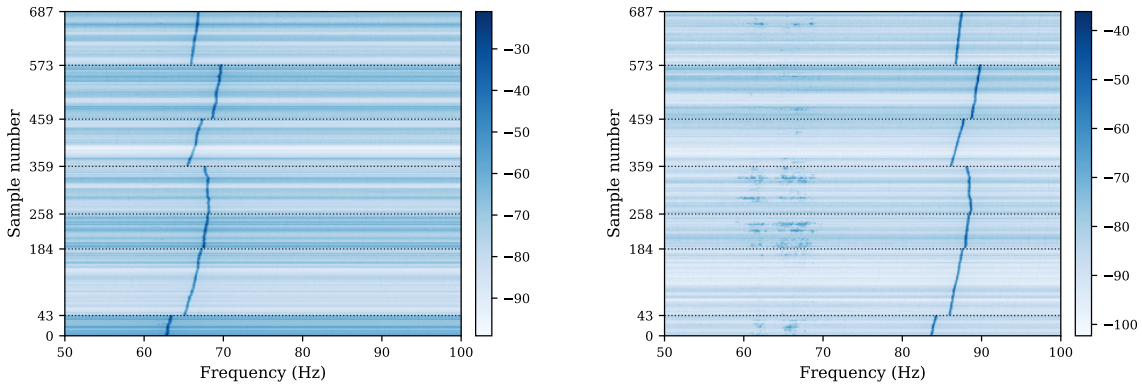
F.1 PSD spectrograms

Figures F.1–F.4 show the Power Spectral Density (PSD) of all processed signals (i.e. band-pass and similarity filtered signals) per sensor in the form of spectrograms. In the following, the power spectrum of a single signal is also referred to as a sample or observation. In the figures, the samples are sorted in chronological order, with the first observation at the bottom. The different datasets are separated by the dotted horizontal lines and the cumulative size of the datasets is indicated on the vertical axis. The order of the datasets corresponds to Table 2.1. The position of the sensors are shown in Figure 2.3, and they are divided over three segments; sensor 1 to 10 in segment 1, sensor 11 to 22 in segment 2, and sensor 23 to 32 in segment 3.

From the figures it can be observed that most sensors have a PSD with one dominant frequency component only, while the other sensors have low quality spectra (i.e. large variability and very low magnitude). In particular, the following sensors have considerably lower quality power spectra for all processed signals: 4, 5, 6, 9, 11, 14, 20, 21, 22, 24, 27, 29. Furthermore, the power spectra of sensors 25, 30 and 32 have a dominant frequency component close to the boundary of the frequency range 50–100 Hz, which is not consistently observed in all processed signals. Those 15 sensors are not useful for damage detection due to the large variability in spectral density that are obviously not related to damage. A closer look to the figures reveals that the spectra of these sensors has much lower magnitude than the other sensors. Hence, these sensors will also have a low contribution in the singular value decomposition of the spectral density matrix in the frequency domain decomposition.

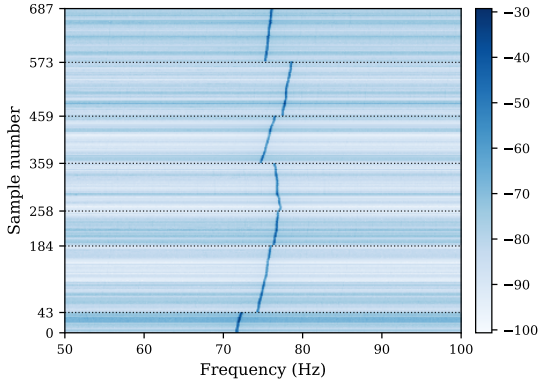
The “jumps” in the spectral line with highest intensity (i.e. dominant frequency component) occurs between two distinct datasets, resulting from the instant change of measurement conditions due to the time gap between the datasets. Moreover, the dominant frequency component changes over time, presumably due to variations in the environmental conditions.

Figures F.5 and F.6 show the PSDs of different mass classes that were measured on 23 March 2017 and 28 March 2017, respectively.

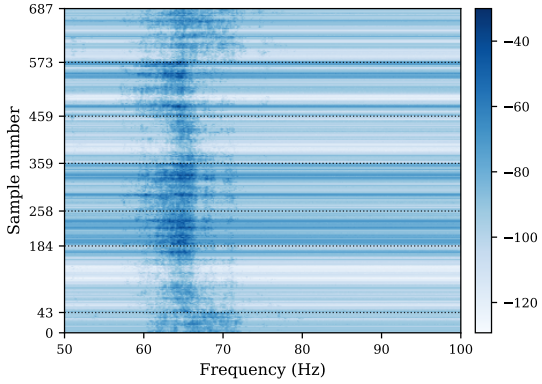


(a) Sensor 1

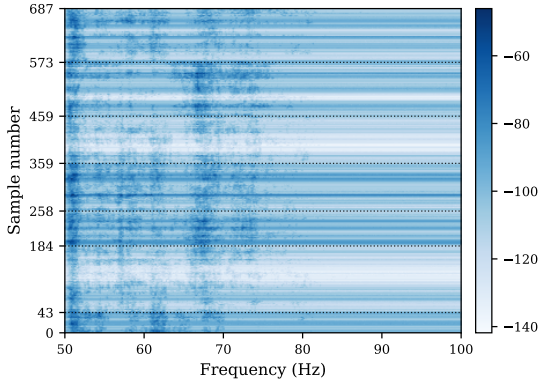
(b) Sensor 2



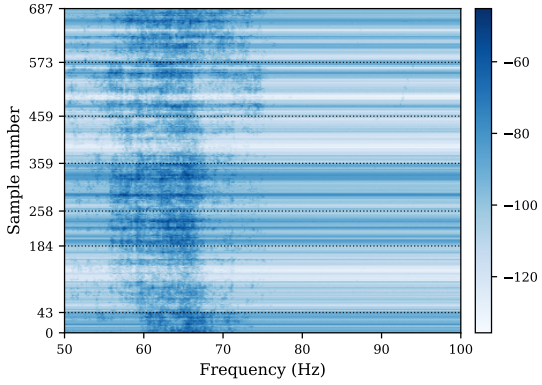
(c) Sensor 3



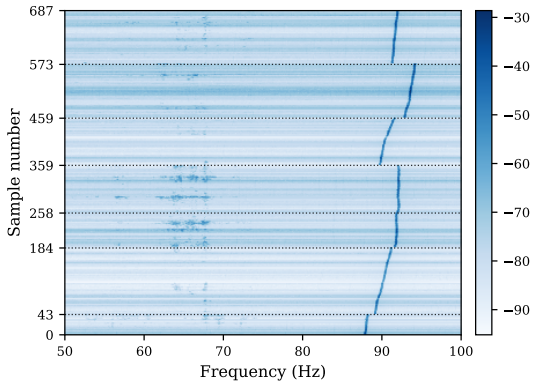
(d) Sensor 4



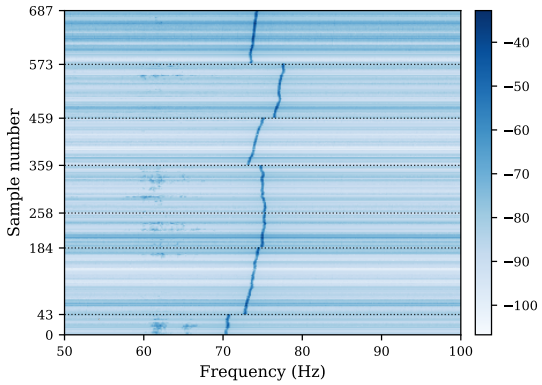
(e) Sensor 5



(f) Sensor 6

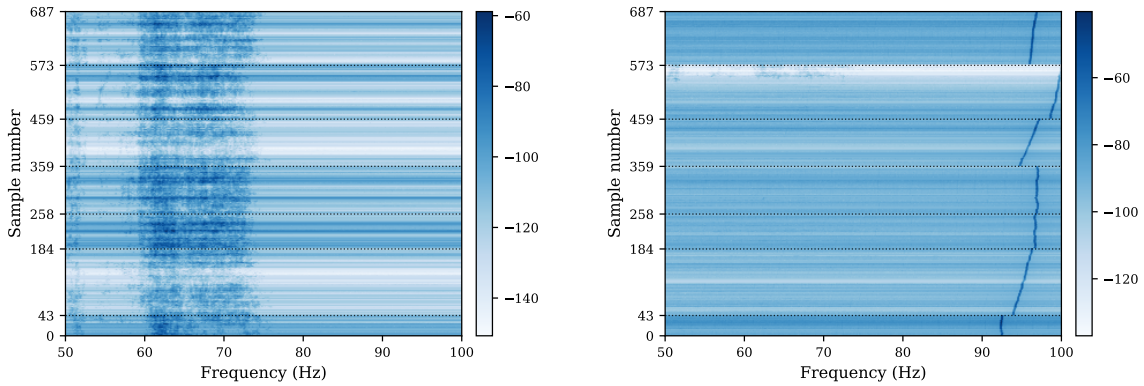


(g) Sensor 7



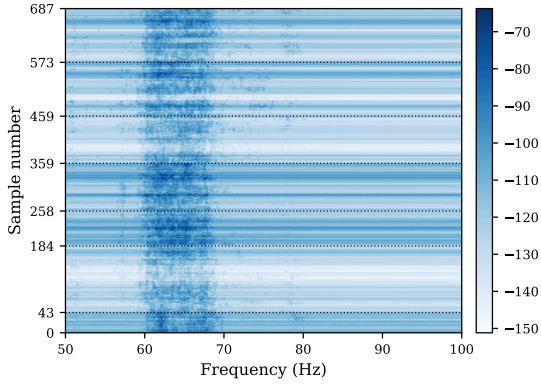
(h) Sensor 8

Figure F.1: PSD spectrogram for sensor 1 to 8 of all data sets. The data sets for different mass classes are separated with a dotted line and the cumulative size of these data sets is indicated on the vertical axis. The observations are sorted in chronological order, with the first observation at the bottom.

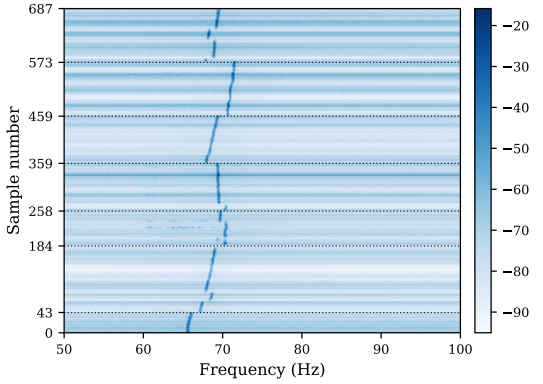


(a) Sensor 9

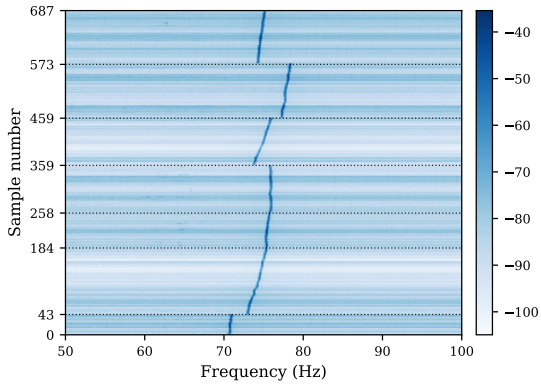
(b) Sensor 10



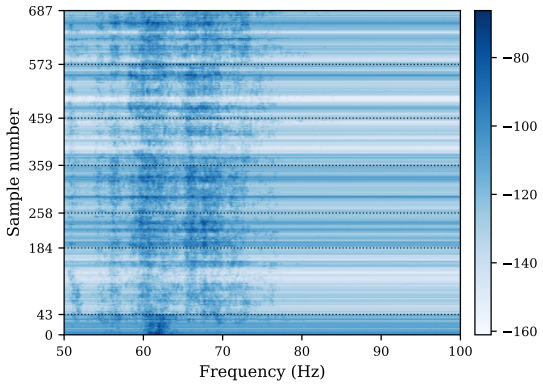
(c) Sensor 11



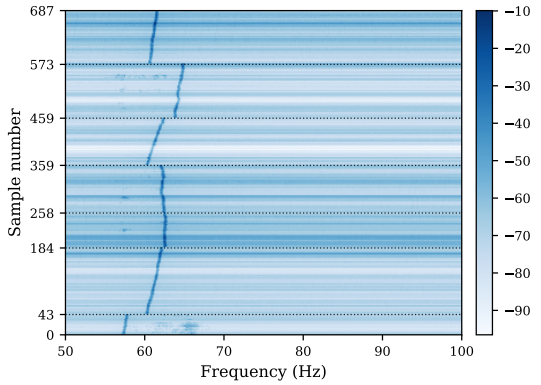
(d) Sensor 12



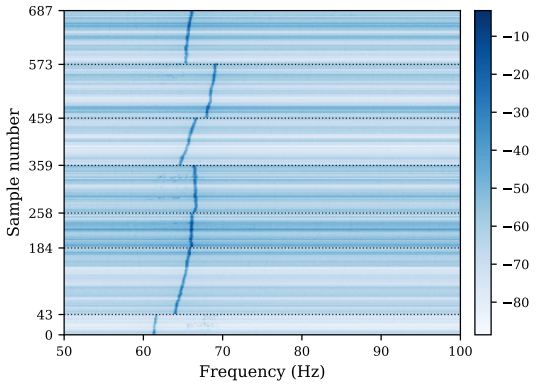
(e) Sensor 13



(f) Sensor 14

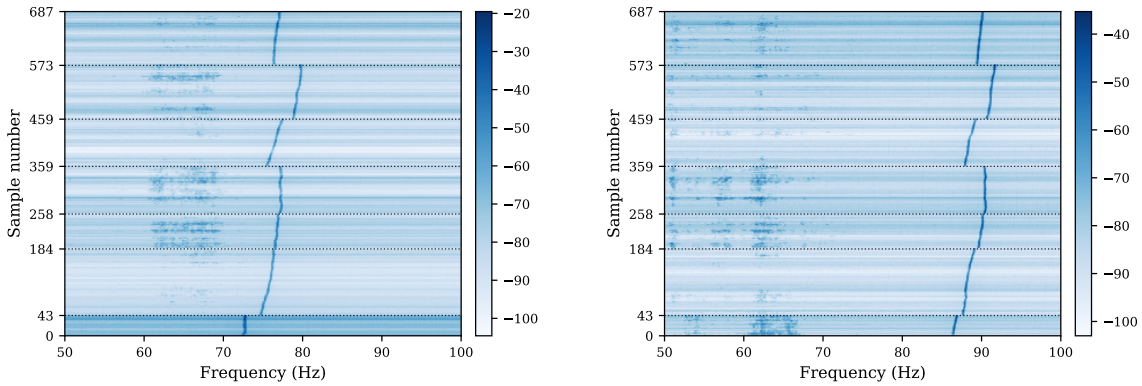


(g) Sensor 15



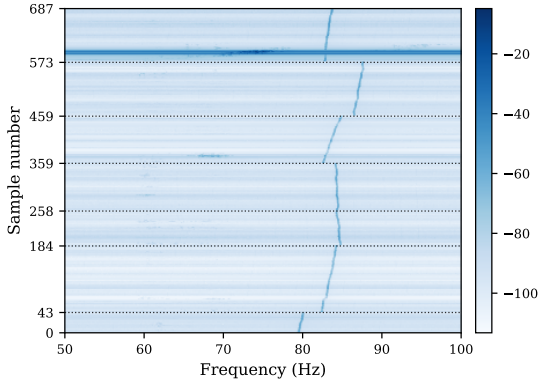
(h) Sensor 16

Figure F.2: PSD spectrogram for sensor 9 to 16 over all datasets. The datasets are separated by a dotted line and the cumulative size of the datasets is indicated on the vertical axis. The observations are sorted in chronological order, with the first observation at the bottom.

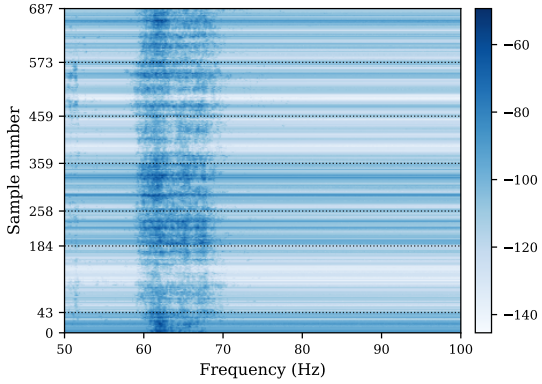


(a) Sensor 17

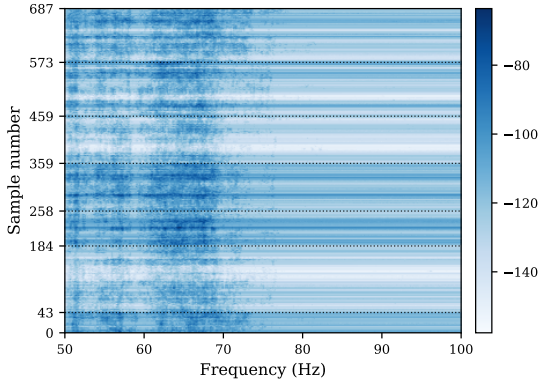
(b) Sensor 18



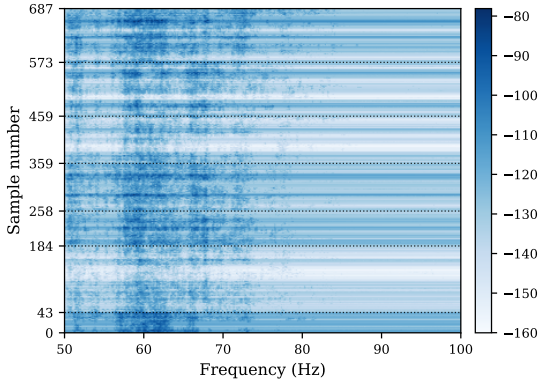
(c) Sensor 19



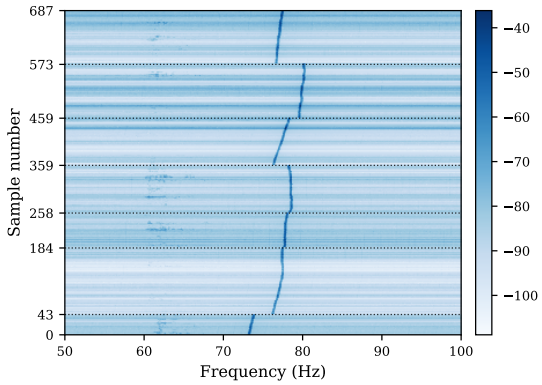
(d) Sensor 20



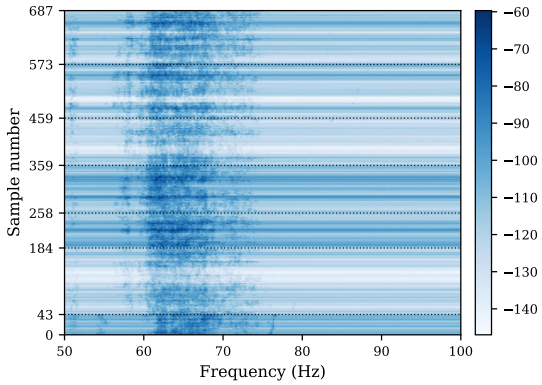
(e) Sensor 21



(f) Sensor 22

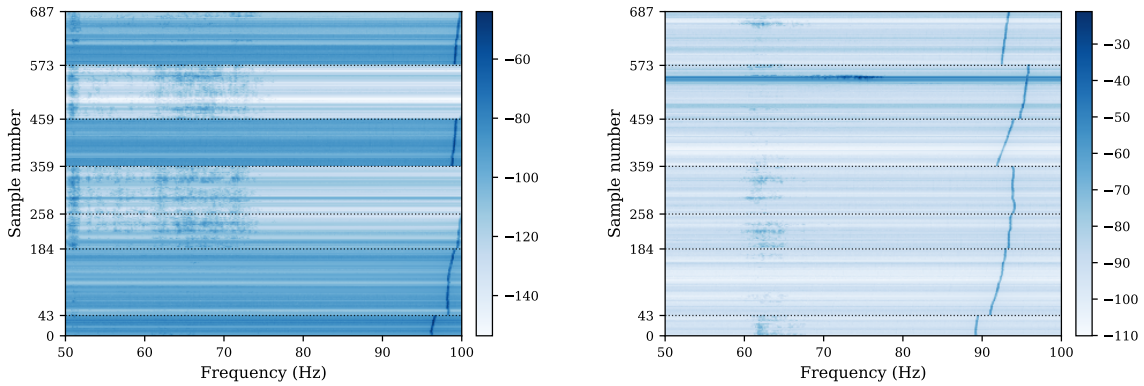


(g) Sensor 23



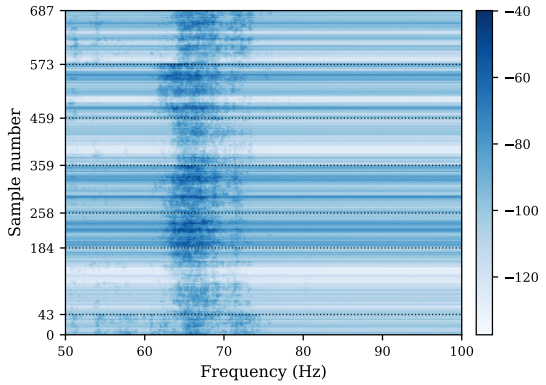
(h) Sensor 24

Figure F.3: PSD spectrogram for sensor 17 to 24 over all datasets. The datasets are separated by a dotted line and the cumulative size of the datasets is indicated on the vertical axis. The observations are sorted in chronological order, with the first observation at the bottom.

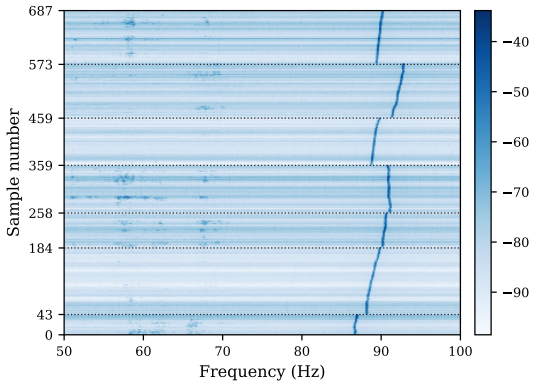


(a) Sensor 25

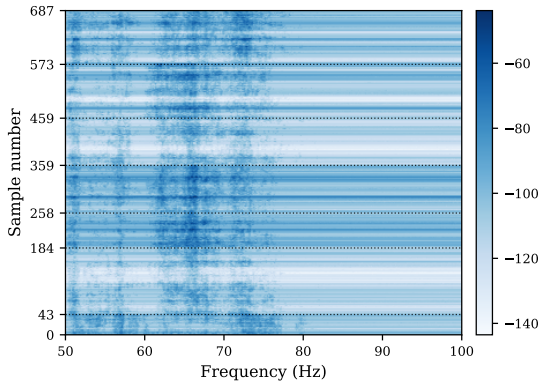
(b) Sensor 26



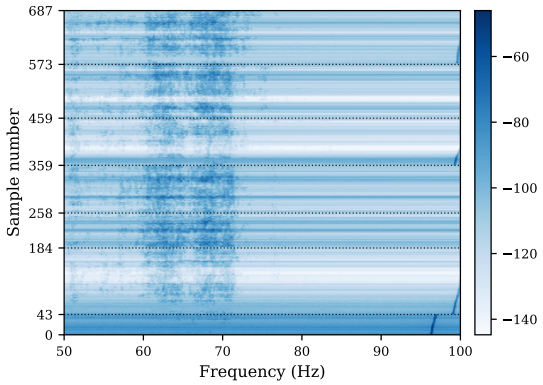
(c) Sensor 27



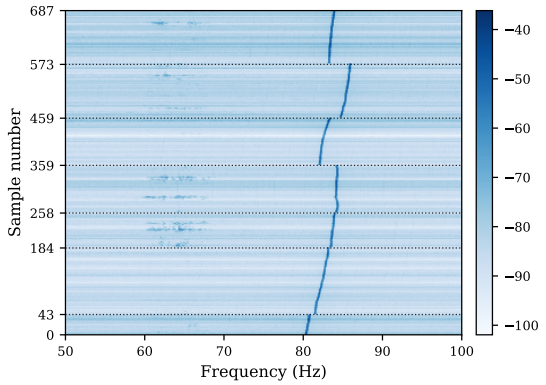
(d) Sensor 28



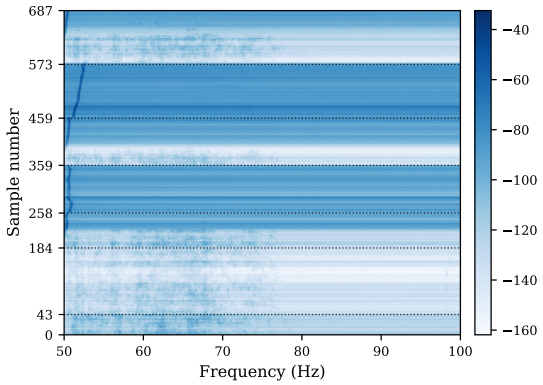
(e) Sensor 29



(f) Sensor 30



(g) Sensor 31



(h) Sensor 32

Figure F.4: PSD spectrogram for sensor 25 to 32 over all datasets. The datasets are separated by a dotted line and the cumulative size of the datasets is indicated on the vertical axis. The observations are sorted in chronological order, with the first observation at the bottom.

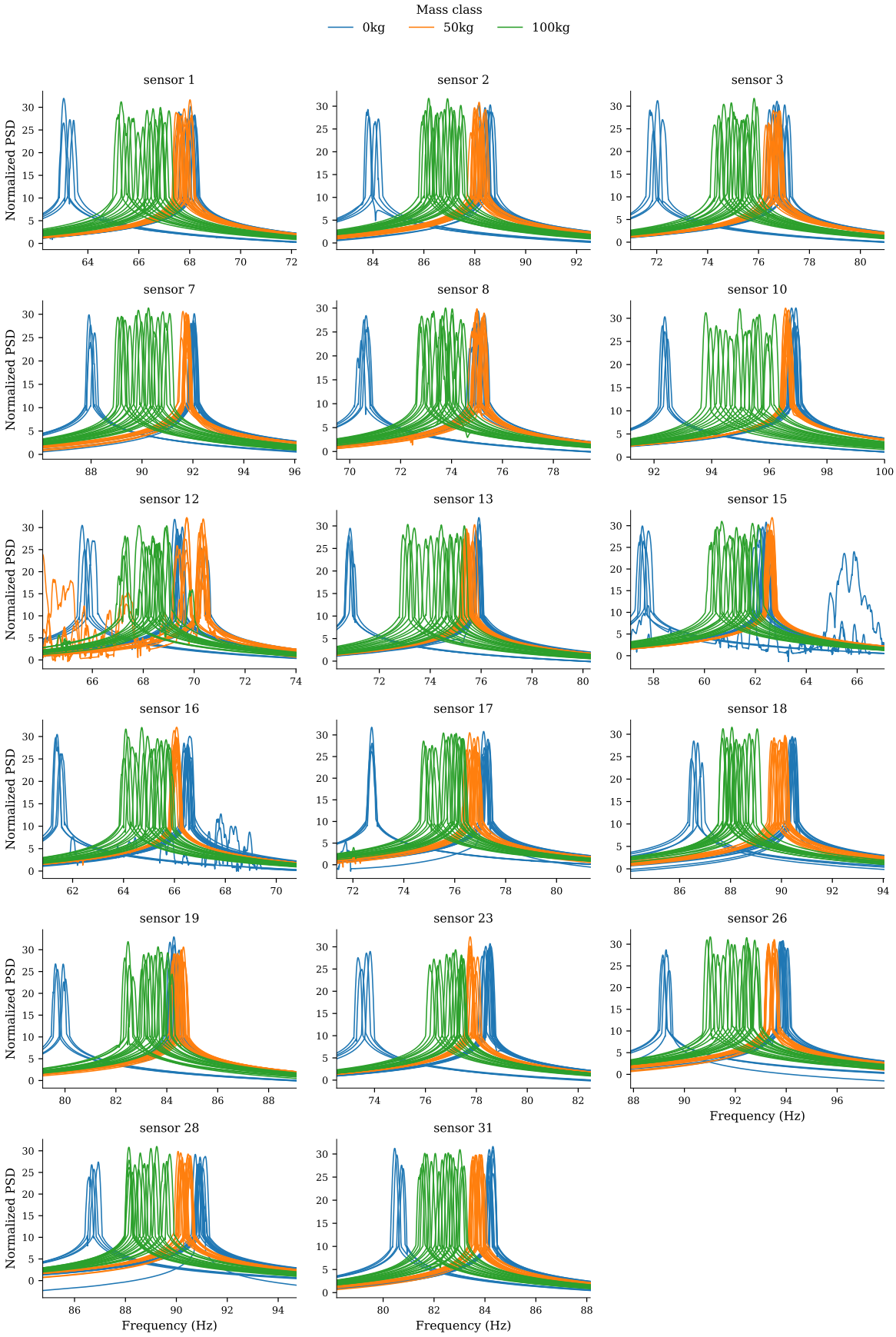


Figure F.5: March 23, 2017 – Normalized PSDs for the data measured on March 23, 2017 (i.e. dataset 1, 2, 3 and 4); not shown are the sensors for which the PSDs are of low quality. Sensors in the range from 1 to 10 are located in segment 1, from 11 to 22 in segment 2, and from 23 to 32 in segment 3. The added mass is attached to the bridge deck between sensor 16 and 17.

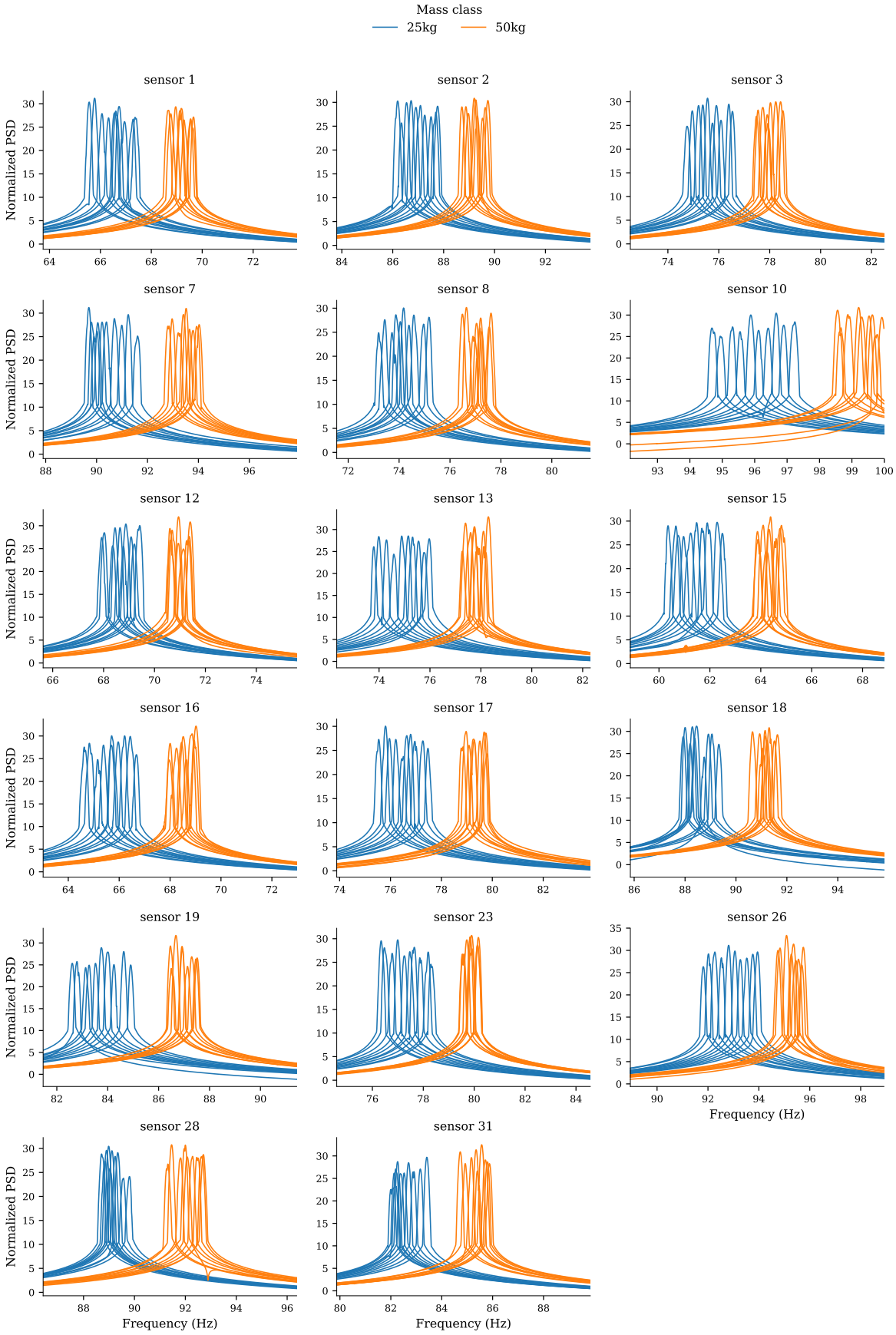
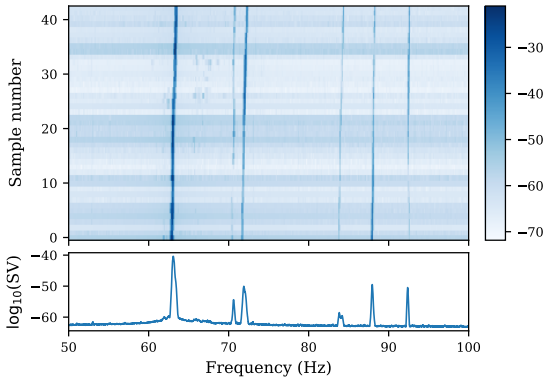
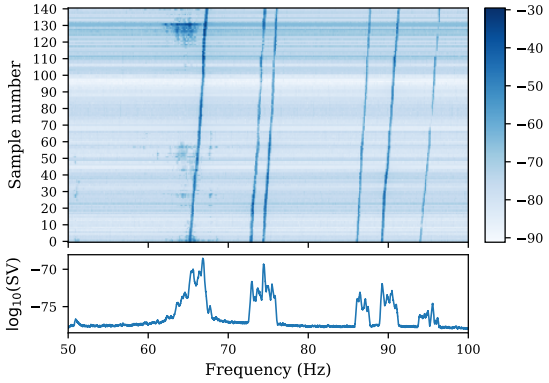


Figure F.6: March 28, 2017 – Normalized PSDs for the data measured on March 28, 2017 (i.e. dataset 5 and 6); not shown are the sensors for which the PSDs are of low quality. Sensors in the range from 1 to 10 are located in segment 1, from 11 to 22 in segment 2, and from 23 to 32 in segment 3. The added mass is attached to the bridge deck between sensor 16 and 17.

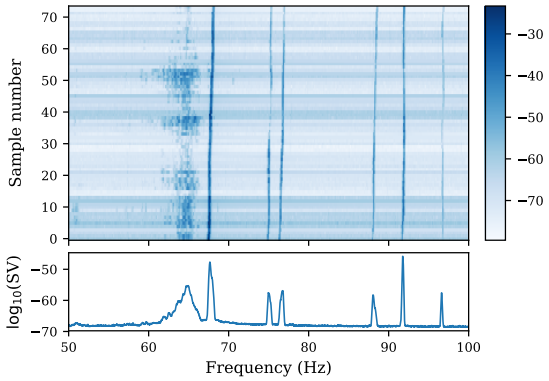
F.2 Singular value spectra segment 1 & 3



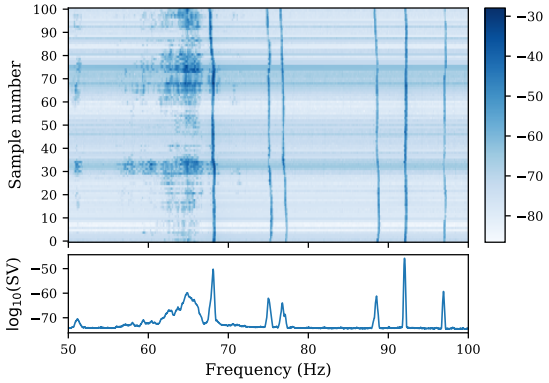
(a) Dataset 1 (0 kg)



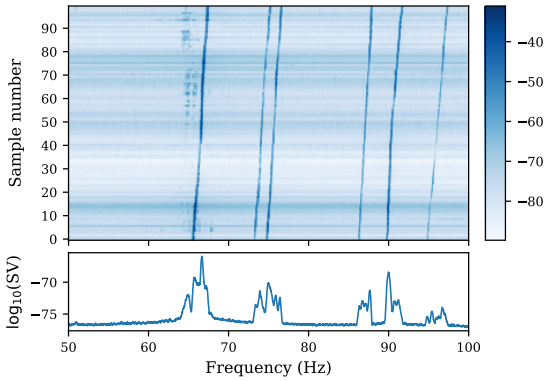
(b) Dataset 2 (100 kg)



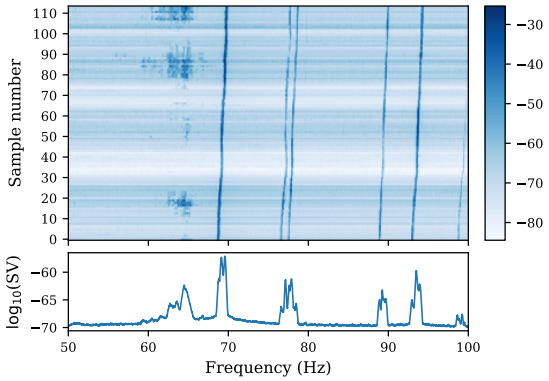
(c) Dataset 3 (50 kg)



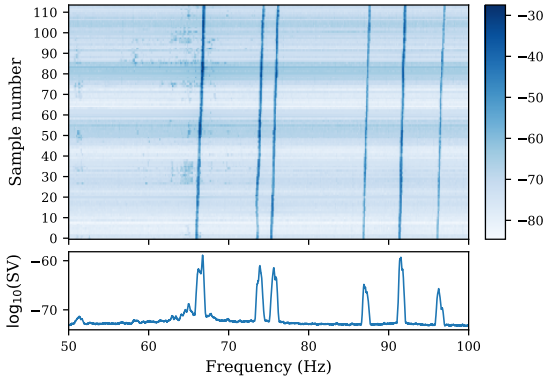
(d) Dataset 4 (0 kg)



(e) Dataset 5 (25 kg)

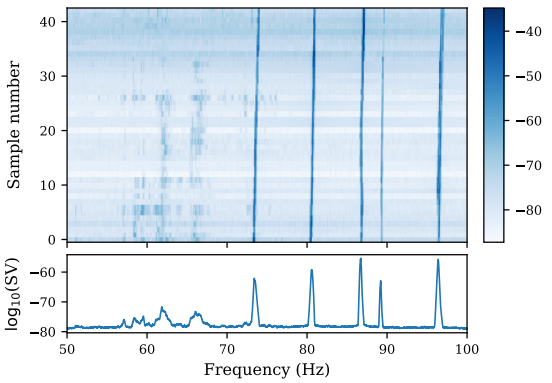


(f) Dataset 6 (50 kg)

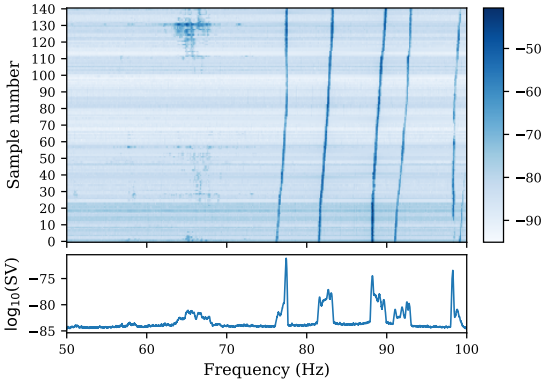


(g) Dataset 7 (75 kg)

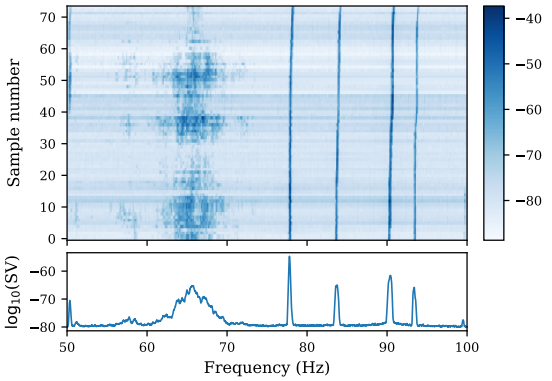
Figure F.7: Singular value spectra of the processed signals from segment 1, separated by dataset (added mass). The top figure shows a spectrogram with the observations in chronological order on the vertical axis; the bottom figure shows the average spectrum over all observations of the particular dataset.



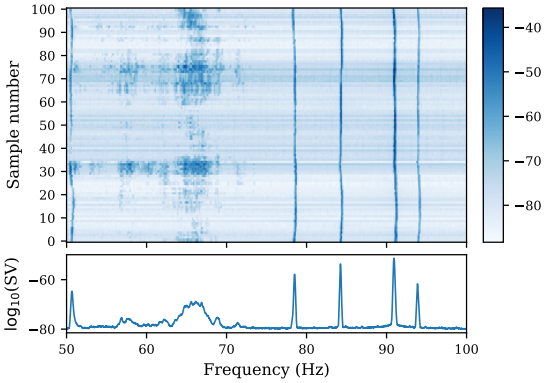
(a) Dataset 1 (0 kg)



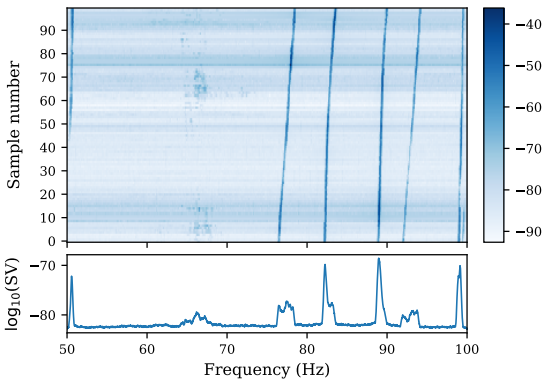
(b) Dataset 2 (100 kg)



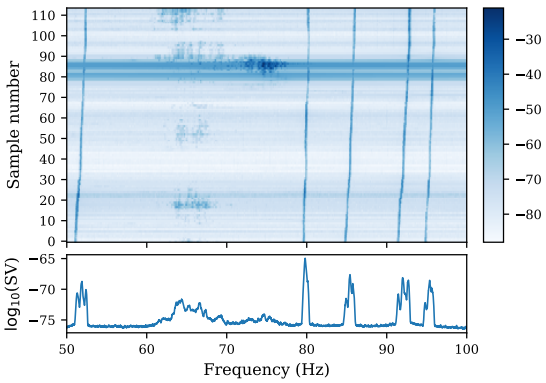
(c) Dataset 3 (50 kg)



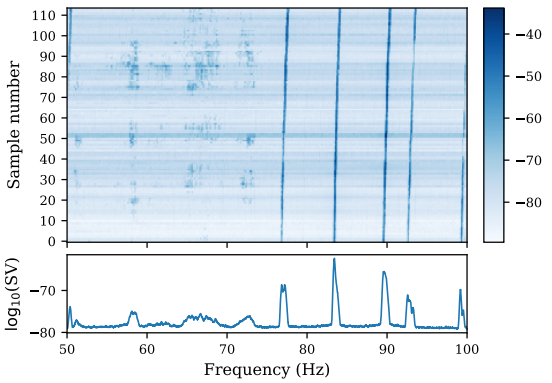
(d) Dataset 4 (0 kg)



(e) Dataset 5 (25 kg)



(f) Dataset 6 (50 kg)



(g) Dataset 7 (75 kg)

Figure F.8: Singular value spectra of the processed signals from segment 3, separated by dataset (added mass). The top figure shows a spectrogram with the observations in chronological order on the vertical axis; the bottom figure shows the average spectrum over all observations of the particular dataset.

G

Modal Analysis

G.1 Identification Algorithm

Description of the identification algorithm that is used for estimating the natural frequencies and mode shapes in Chapter 4. The modal parameters are estimated using the Frequency Domain Decomposition (FDD) method. With this method the modes of a structure can be identified from the peaks in a singular value spectrum. This process of peak picking can easily be automated, such that the modal parameter estimation for a large number of measurements can be done automatically. Below a simple algorithm for automatic mode identification is described, which will be used to estimate natural frequencies and mode shapes from the data in this thesis. It must be noted that the algorithm has its limitations and might not work well on any dataset. Nevertheless, it works well for the dataset used in this thesis.

Suppose we have N vectors of response measurements $\mathbf{y}(t)$, then for each vector the spectral density matrix can be estimated by $\hat{\mathbf{S}}_{yy}(f) = \mathbf{Y}(f)\mathbf{Y}(f)^{\mathrm{H}}$ where $\mathbf{Y}(f)$ is the Fourier transform of $\mathbf{y}(t)$. By decomposition of the spectral density matrix using the Singular Value Decomposition (SVD) the *first* singular values $\sigma_1(f)$ and corresponding left singular vectors $\mathbf{u}_1(f)$ are obtained. This can be done for all N vectors of response measurements resulting in the set $\{(\sigma_{1,k}(f), \mathbf{u}_{1,k}(f)) : k = 1, \ldots, N\}$. This set serves as the starting point of the identification algorithm explained below, where an element of the set will be referred to as an observation.

To determine the modes from all observations efficiently, an automatic identification algorithm is implemented in python. This algorithm involves the following steps:

Step 1. Find peaks in the singular value spectrum $\sigma_1(f)$ of a single observation; let f_r, $r = 1, \ldots, N_p$ denote the frequencies of the N_p peaks found. Each peak represents a possible mode, whose natural frequency is estimated by the peak frequency f_r and the corresponding mode shape is estimated by the left singular vector; that is, $\boldsymbol{\phi}_r = \mathbf{u}_1(f_r)$. At the end of this step, we have a set $\{(f_r, \boldsymbol{\phi}_r) : r = 1, \ldots, N_p\}$ consisting of N_p possible modes, where a mode is characterized by its natural frequency and mode shape. In Figures G.1b, G.1d, G.1f and G.1h four examples of the singular value spectrum are shown including the peaks that were found with an automatic peak picking algorithm. Here, twelve peaks ($N_p = 12$) are selected in each spectrum indicated by the numbers. From the figures it can be observed that the peaks are correctly found, but also some spurious peaks are selected which are not related to a structural mode. These modes are removed in the following steps.

Step 2. Remove non-uniquely identified modes. For this we compute the pairwise distance (G.1) between all possible modes that were identified from a single observation. The distance between two identified modes, $\{f_j, \boldsymbol{\phi}_j\}$ and $\{f_l, \boldsymbol{\phi}_l\}$, is defined as

$$d_{jl} = \frac{|f_j - f_l|}{\max(f_j, f_l)} + 1 - \mathrm{MAC}(\boldsymbol{\phi}_j, \boldsymbol{\phi}_l) \tag{G.1}$$

which considers both the difference in natural frequency and mode shape; the latter is given by the Modal Assurance Criterion (MAC), which is introduced in Section 3.2.5. The distance in equation

(4.5) is close to zero for two modes that belong to the same mode. If multiple modes are found with pairwise distances close to zero (lower than a certain threshold Ω_1) then these modes are removed from the set of possible modes except the mode with the largest peak value in the singular value spectrum. In this way a set of certainly different modes is obtained. In Figures G.1a, G.1c, G.1e and G.1g four examples of the pairwise distances are shown in the form of a heatmap. The pairwise distances correspond to the possible modes that were found in the four examples of previous step. The diagonal elements in the matrix represents the pairwise distance of a mode with itself, which is of course zero. The off-diagonal elements represents the pairwise distance between two different identified modes. If this value is close to zero (i.e. lower than a certain threshold) the two modes belong to the same mode and are not unique. If multiple values on the off-diagonal for a certain peak are close to zero, then these peaks represent all the same mode. Only the peak with the largest peak value in the singular value spectrum, while the others are removed from the set. For example, in Figure G.1a the modes corresponding to peak 7 to 12 are similar to the mode corresponding to the peak 1. The latter has the largest peak value and is retained, while the modes corresponding to peak 7 to 12 are removed form the set of possible modes.

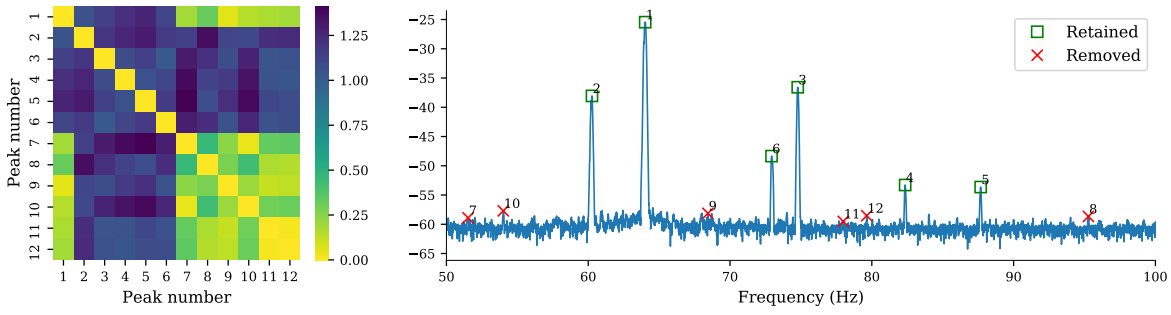
Step 3. Repeat **Step 1** and **Step 2** for all the observations, to obtain a set of uniquely determined possible modes for each observation. The results of four runs were illustrated in the previous steps.

Step 4. Cluster the modes from different observations using hierarchical clustering. Again, the pairwise distance between two modes is computed as in (4.5). Average linkage is used to form successive clusters. Modes with similar characteristics will have a distance close to zero.

Step 5. Obtain clusters with similar modes. The distances between observations in a cluster should be small enough for the observations to belong to the same mode, while the distance between two clusters must be large enough for the clusters to represent two distinct modes. To distinguish between the different clusters a threshold Ω_2 is introduced. The clusters are then formed by observations whose cluster distance is smaller than this threshold. The result of hierarchical clustering is best visualized in the form of a dendogram, which shows the hierarchy of the clusters of observations that are formed. Each observation starts in its own cluster at the bottom of the diagram, and pairs of clusters are merges as one moves up the hierarchy. An example of a dendrogram is shown in Figure G.2. The *tree* is cut at a distance of 0.1 and clusters of similar modes are indicated with different colors. In total nine clusters are formed, seven of them are clearly visible in the dendrogram while the other two clusters are less clear because the clusters contain only a few observations.

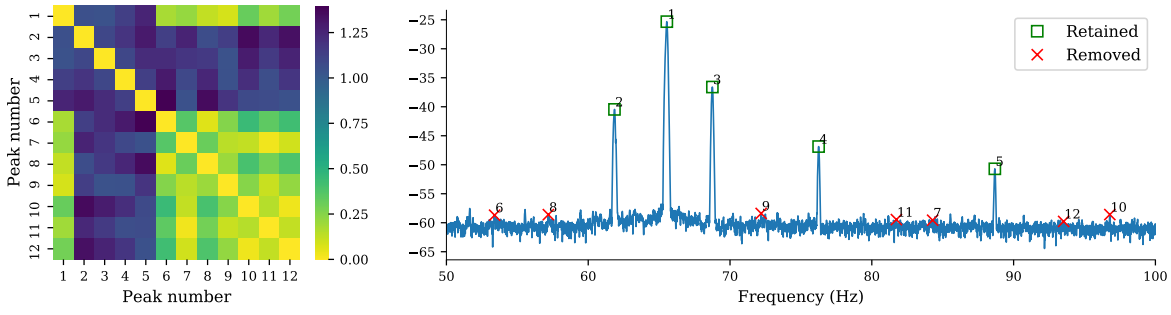
Step 6. Count the number of elements in each cluster and disregard clusters with less elements than a certain threshold. In this way spurious modes are removed. Figure Figure G.3 shows the number of modes in the nine clusters that were formed in previous step. It is clear that two clusters contain only a few modes, and these clusters can be removed by introducing a threshold for the minimum number of elements in a cluster. Here, a 20% threshold level is used meaning that the modes must be present in 20% of the observations (i.e. $0.2 \cdot N$).

In the end, we have a set of modal parameters $\{(f_n, \boldsymbol{\phi}_n) : n = 1, \ldots, N_m\}$ for each observation. A set consist of N_m pairs formed by a natural frequency f_n and a mode shape $\boldsymbol{\phi}_n$.



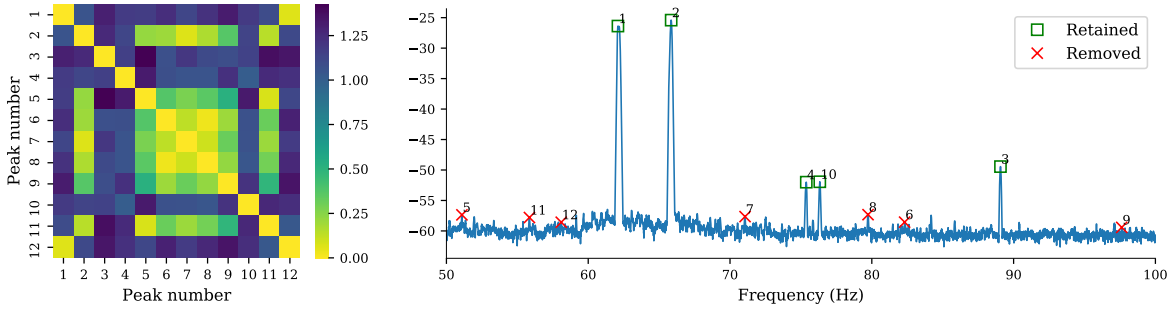
(a) Example 1: Distances

(b) Example 1: singular value spectrum



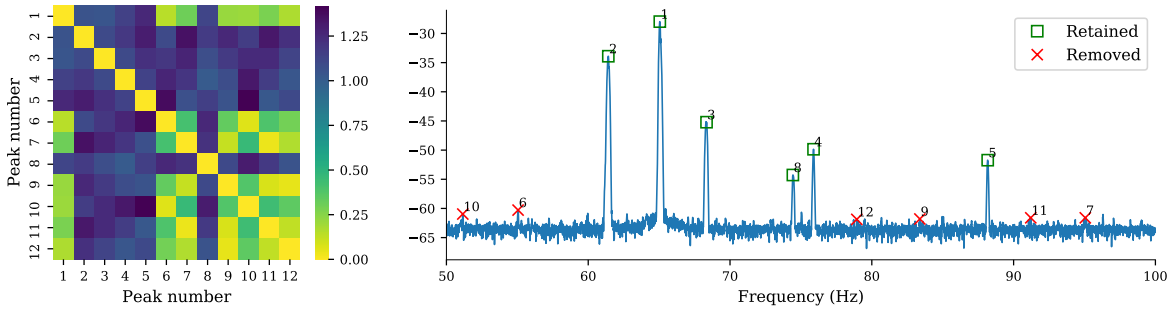
(c) Example 2: Distances

(d) Example 2: singular value spectrum



(e) Example 3: Distances

(f) Example 3: singular value spectrum



(g) Example 4: Distances

(h) Example 4: singular value spectrum

Figure G.1: Examples of mode identification in single observations. The figures on the right show examples of singular value spectra with identified peaks. Red crosses indicate peaks that are removed from the set based on the pairwise distances illustrated in the left figures. The pairwise distance is computed based on the natural frequency and mode shape corresponding to the peaks. Non-uniquely identified modes with a distance close to zero are removed, except the one with largest peak value.

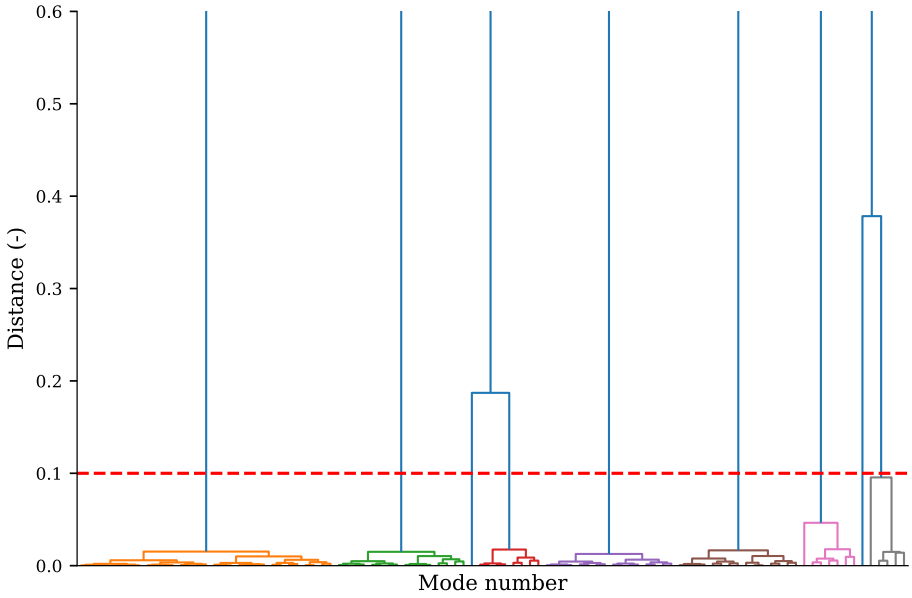


Figure G.2: Hierarchical clustering dendrogram.

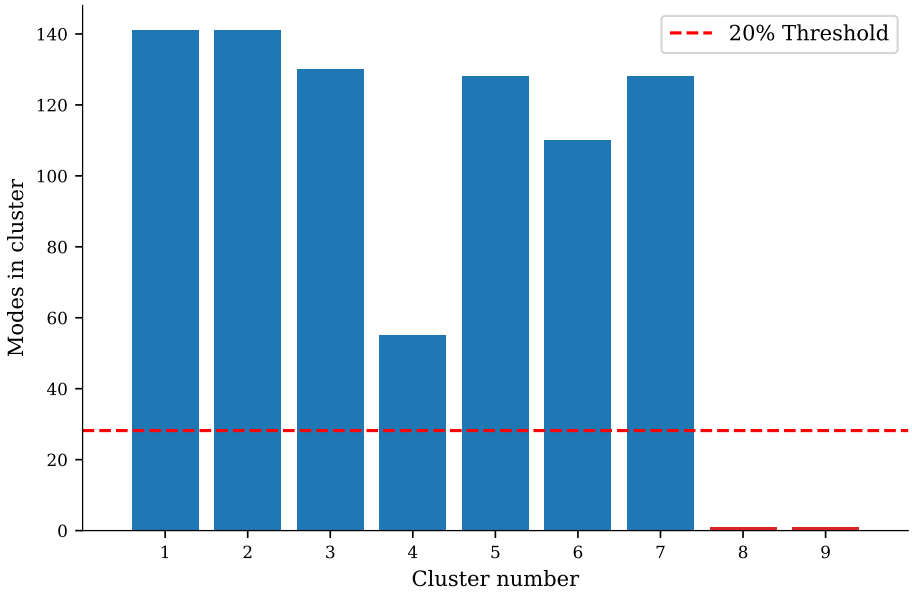
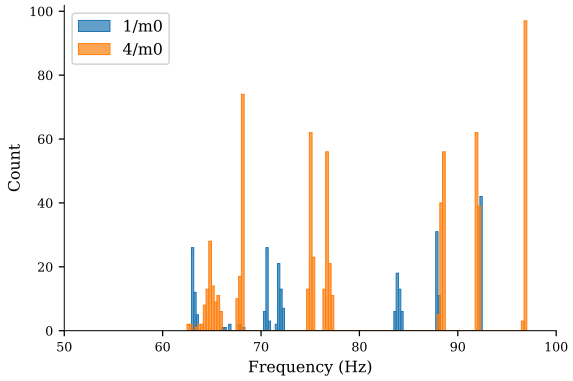
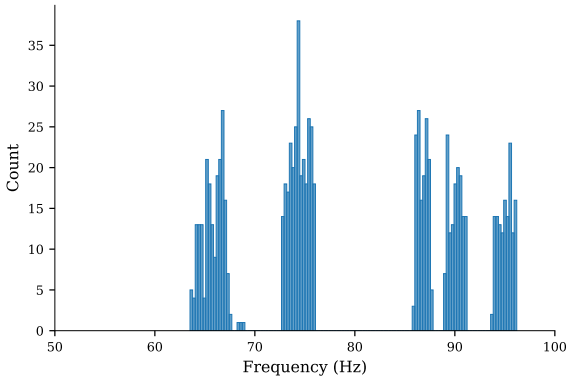


Figure G.3: Number of modes in clusters.

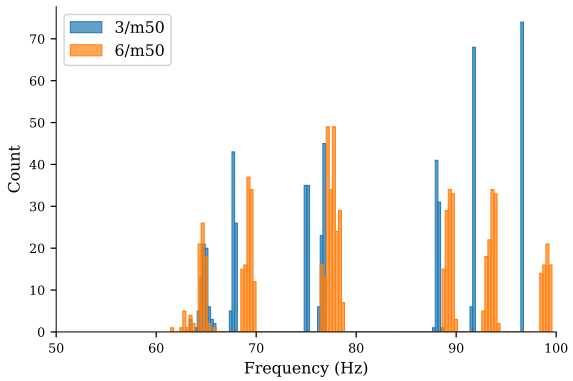
G.2 Results Segment 1



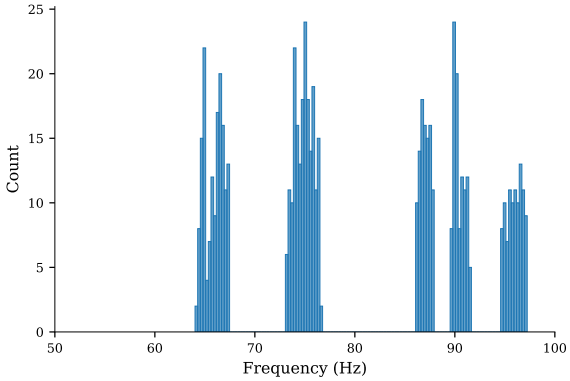
(a) Datasets 1 and 4 (0 kg)



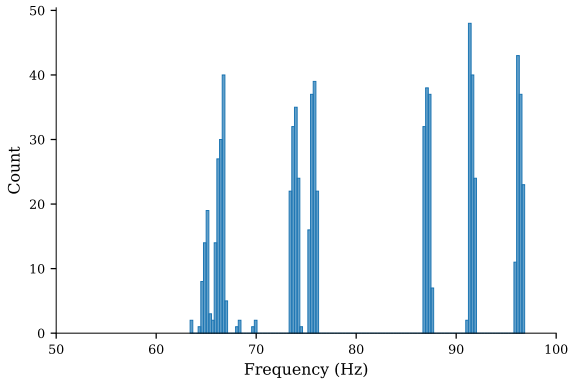
(b) Dataset 2 (100 kg)



(c) Datasets 3 and 6 (50 kg)

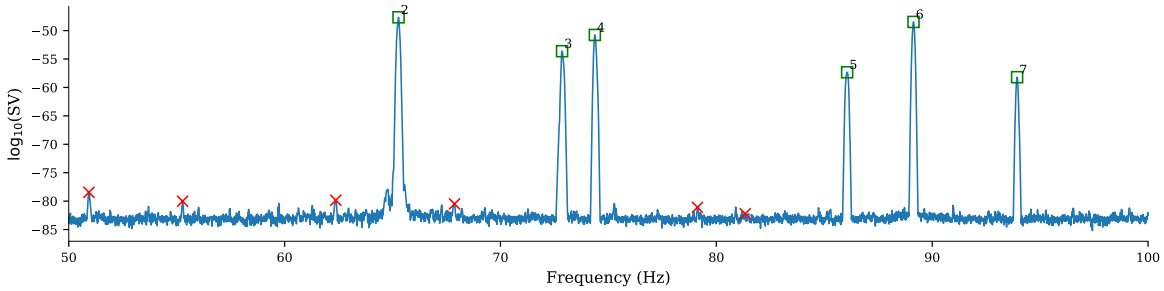


(d) Dataset 5 (25 kg)

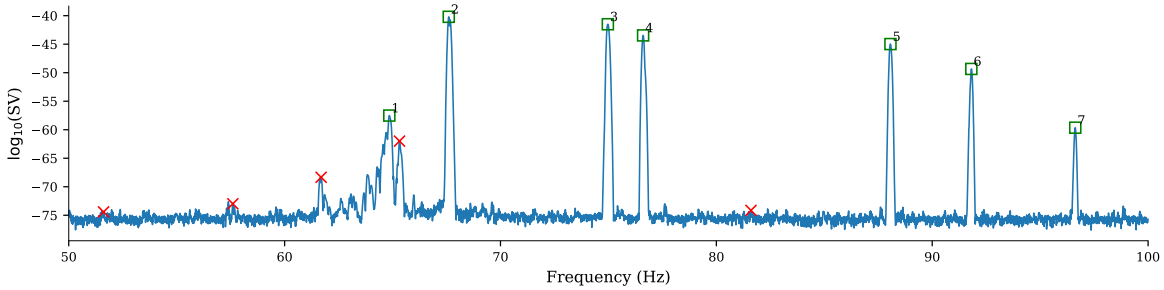


(e) Dataset 7 (75 kg)

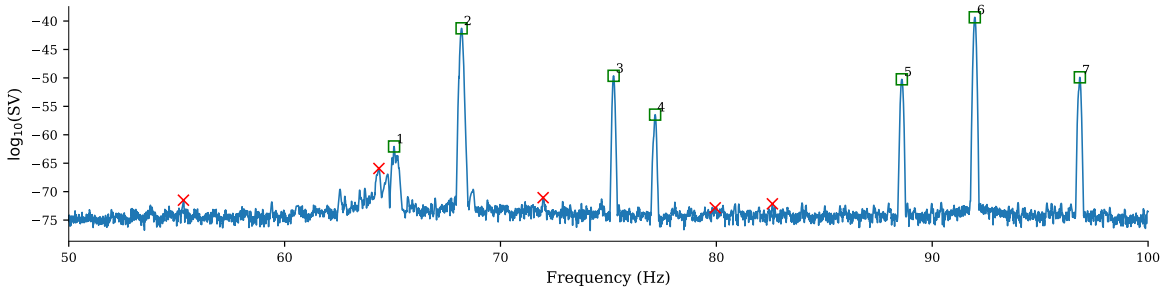
Figure G.4: Frequency count of the identified modes from the singular value spectra of the processed signals from segment 1, separated by added mass class.



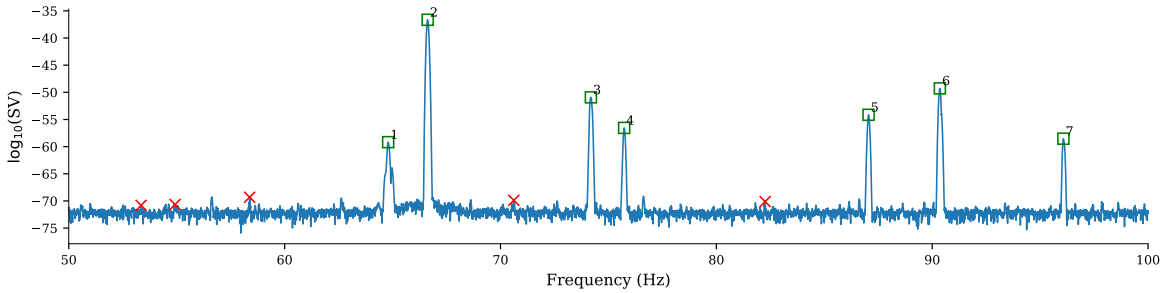
(a) Example 5: dataset 2 (100 kg)



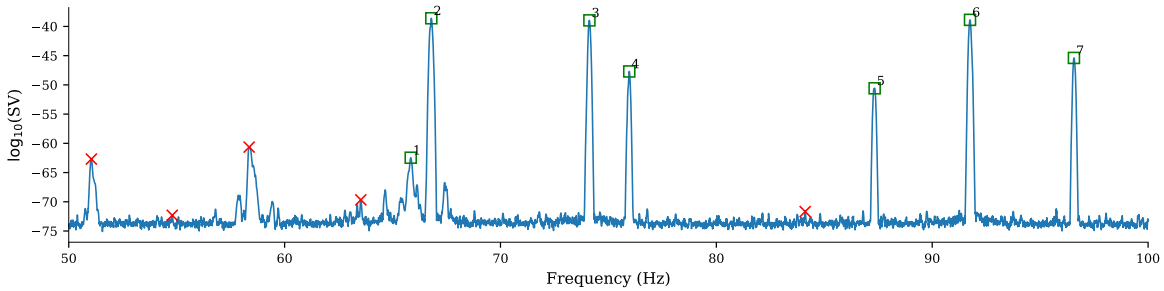
(b) Example 3: dataset 3 (50 kg)



(c) Example 1: dataset 4 (0 kg)



(d) Example 2: dataset 5 (25 kg)



(e) Example 4: dataset 7 (75 kg)

Figure G.5: Examples of the singular value spectrum of the processed signals from segment 1, including the identified modes (□). The red crosses indicate the peaks that are removed during the modal clustering.

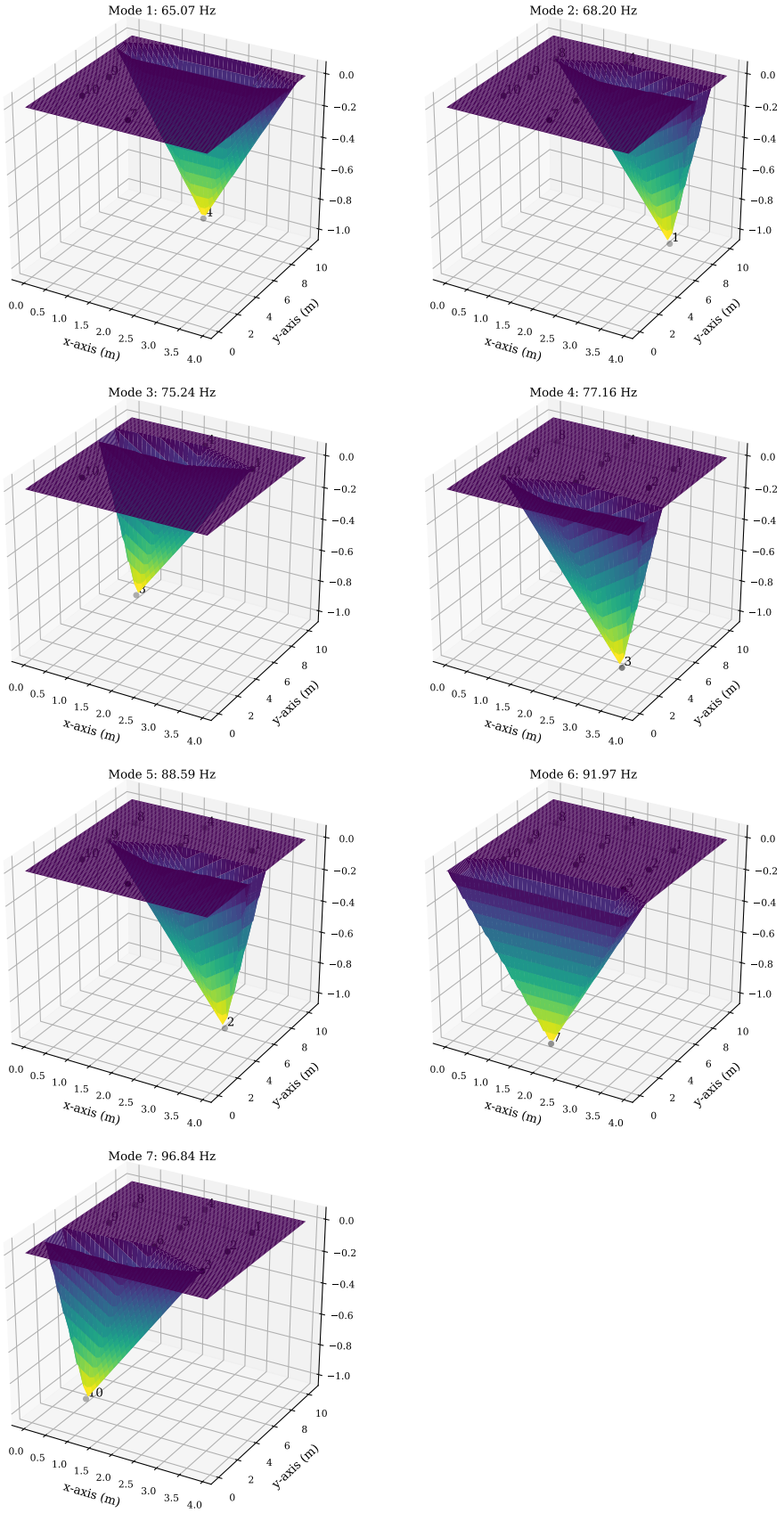
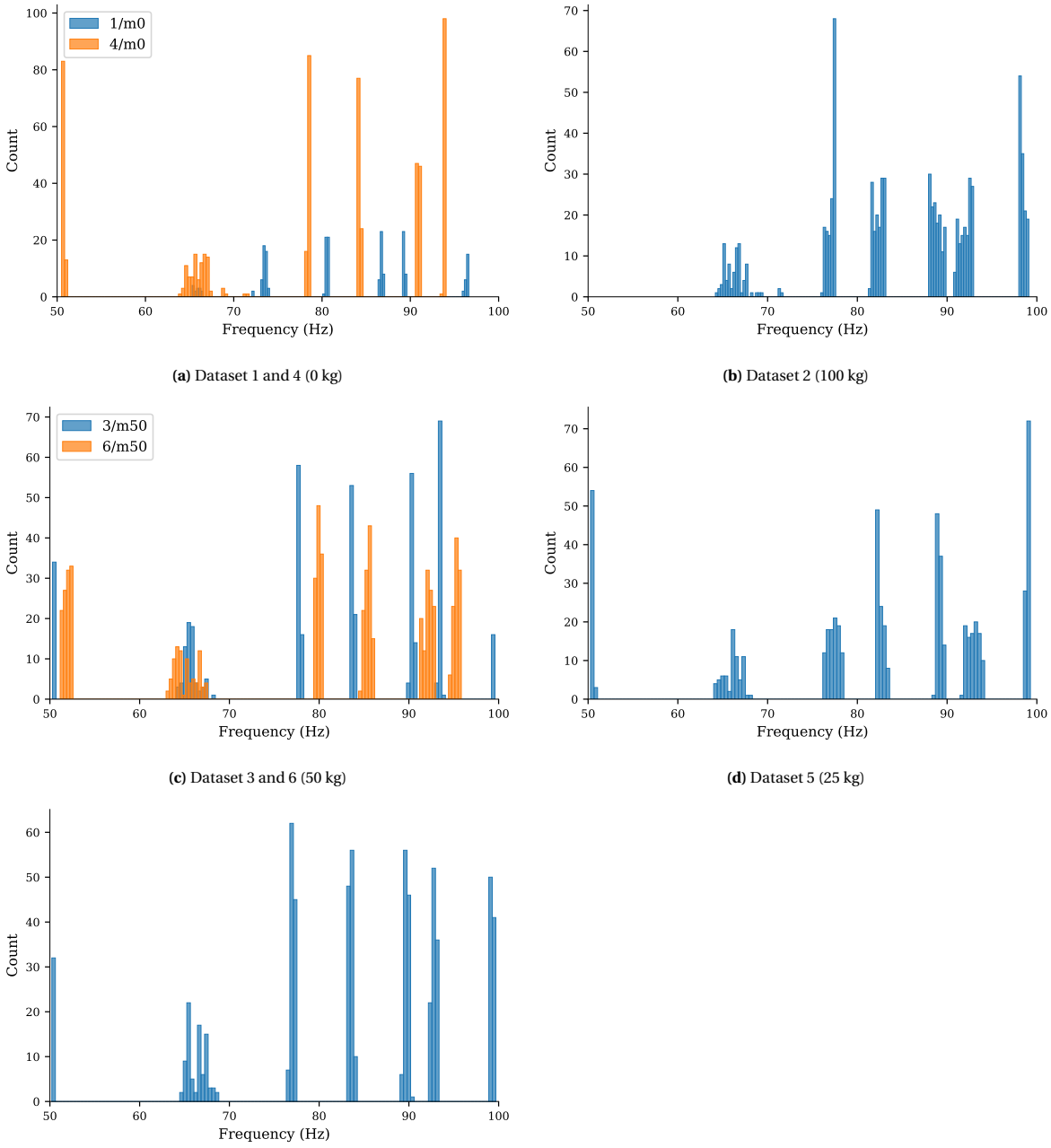


Figure G.6: Local mode shapes of segment 1, corresponding to the identified modes in Figure 4.14c.

G.3 Results Segment 3



(a) Dataset 1 and 4 (0 kg)

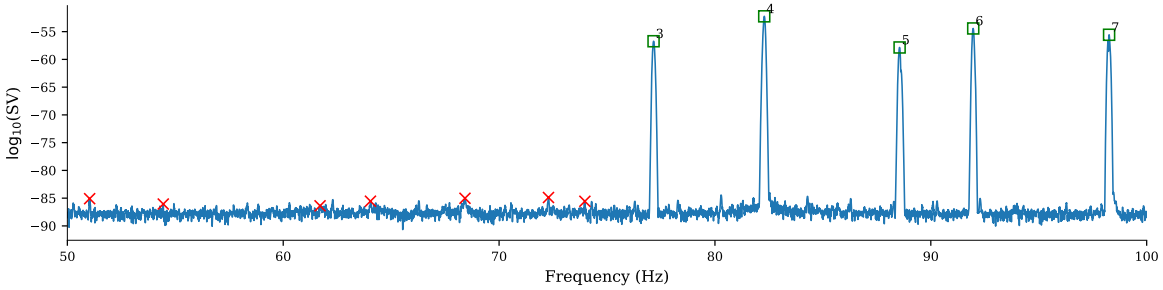
(b) Dataset 2 (100 kg)

(c) Dataset 3 and 6 (50 kg)

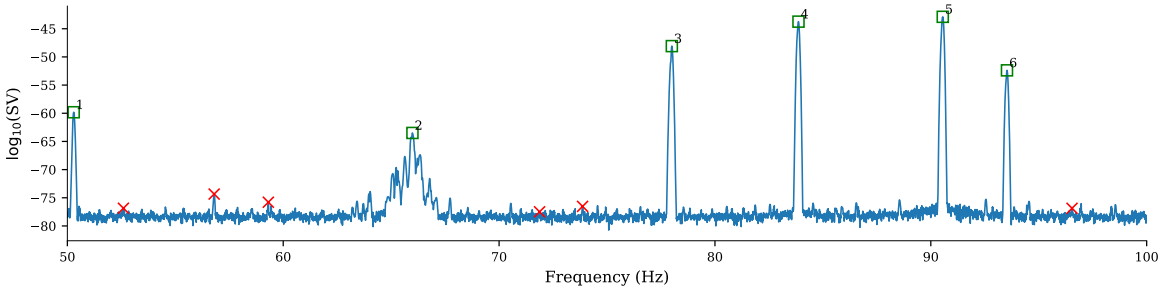
(d) Dataset 5 (25 kg)

(e) Dataset 7 (75 kg)

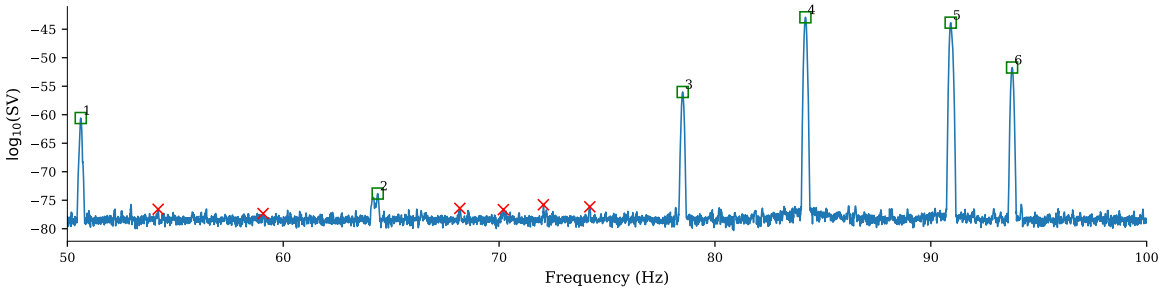
Figure G.7: Frequency count of the identified modes from the singular value spectra of the processed signals from segment 3, separated by added mass class.



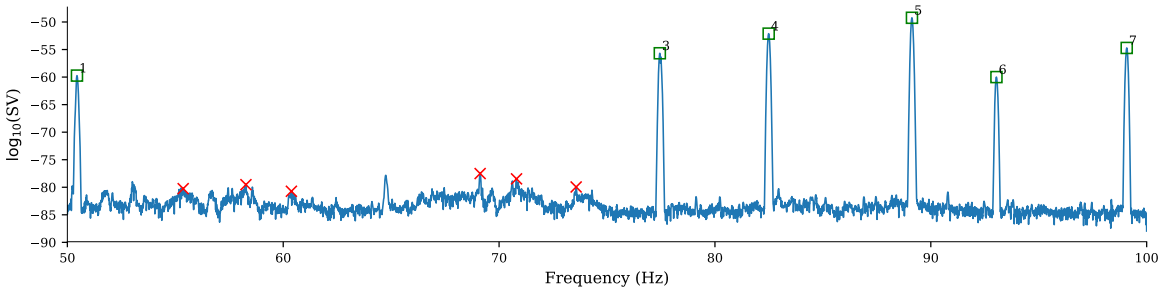
(a) Example 5: dataset 2 (100 kg)



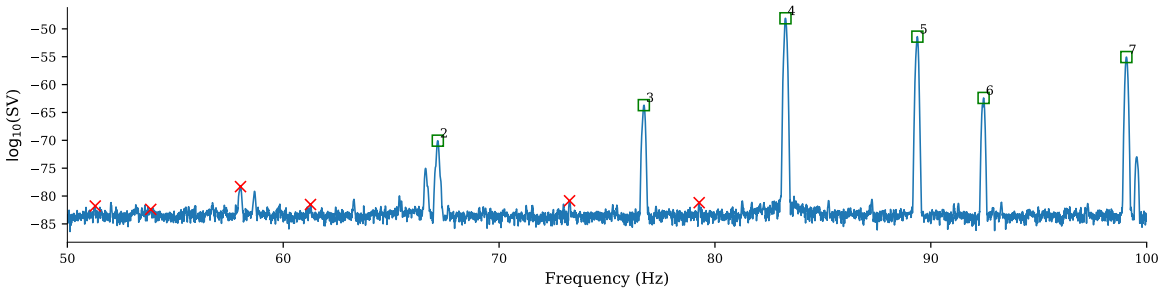
(b) Example 3: dataset 3 (50 kg)



(c) Example 1: dataset 4 (0 kg)



(d) Example 2: dataset 5 (25 kg)



(e) Example 4: dataset 7 (75 kg)

Figure G.8: Examples of the singular value spectrum of the processed signals from segment 3, including the identified modes (□). The red crosses indicate the peaks that are removed during the modal clustering.

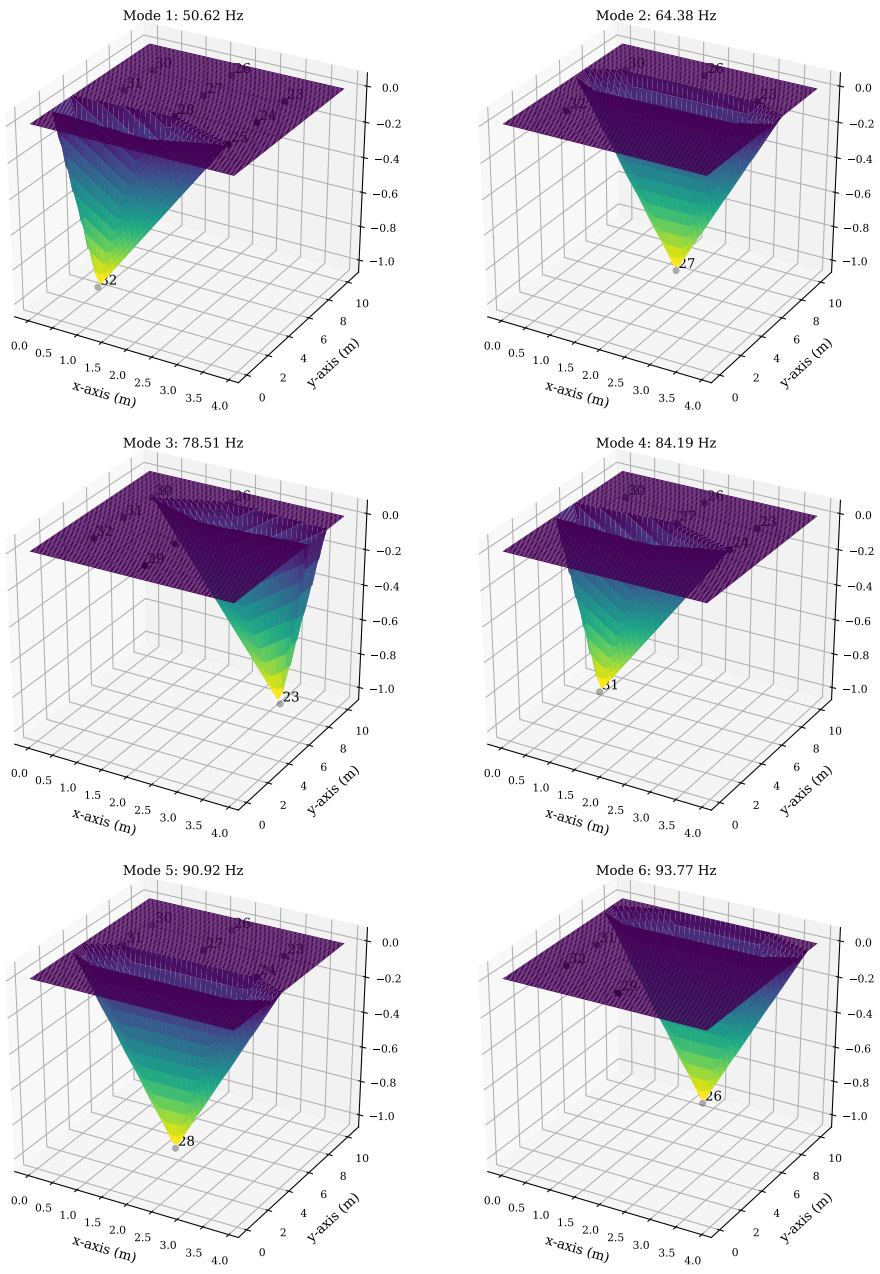


Figure G.9: Local mode shapes of segment 3, corresponding to the identified modes in Figure 4.14c.

H

Classification results per sensor

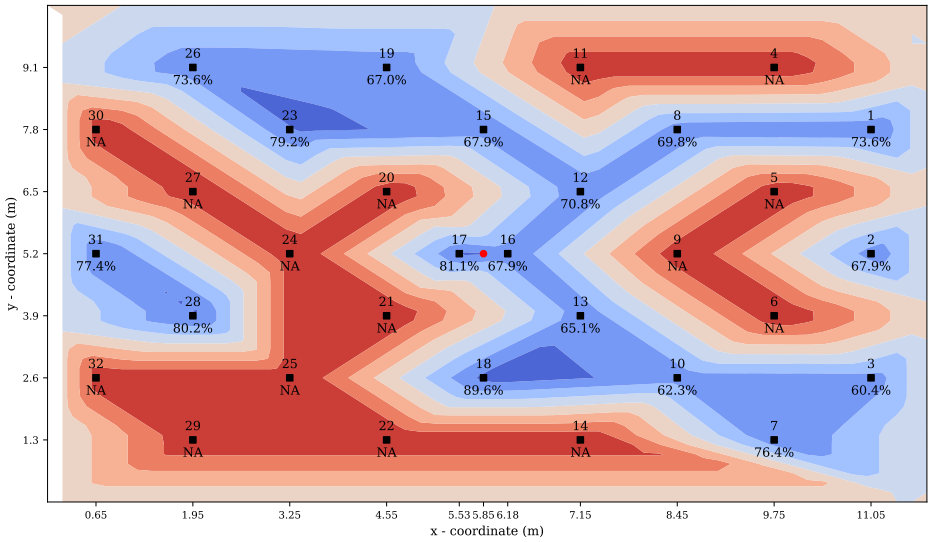
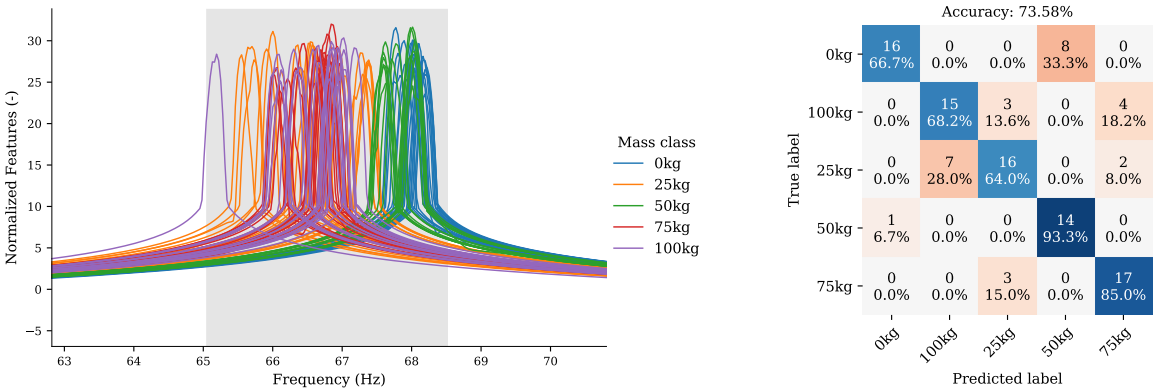
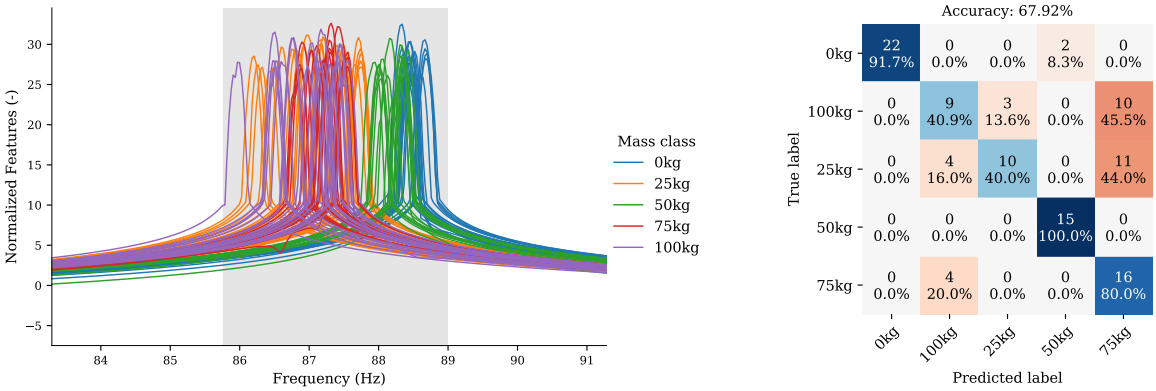


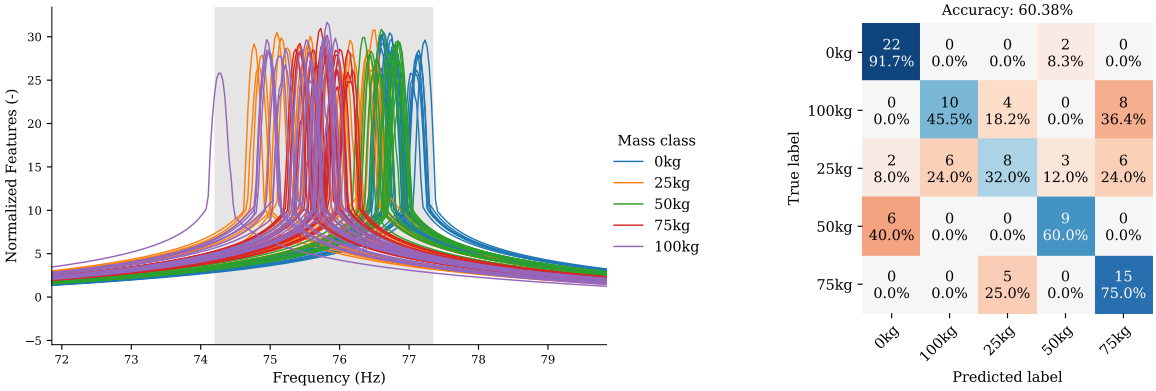
Figure H.1: Overview accuracy scores.



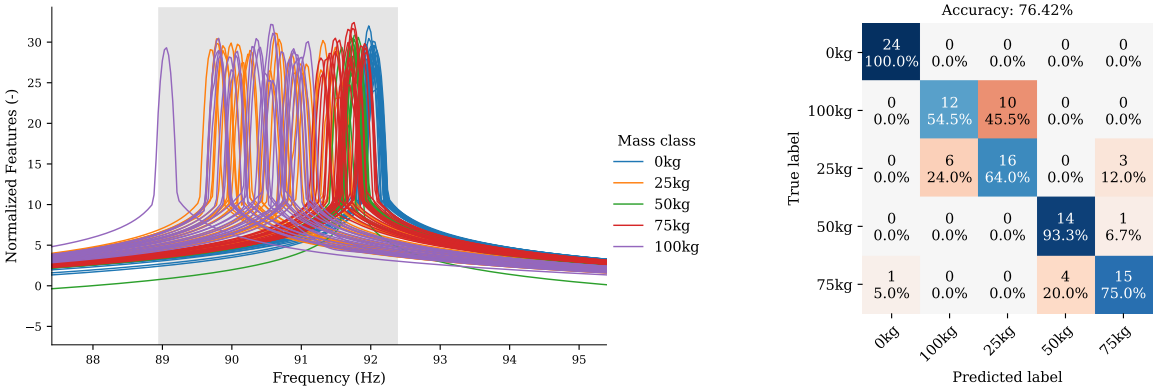
(a) Sensor 1



(b) Sensor 2

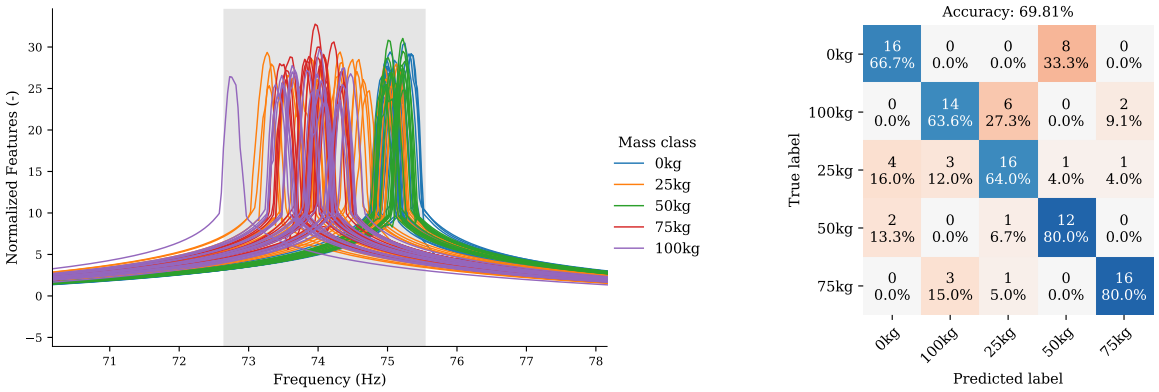


(c) Sensor 3

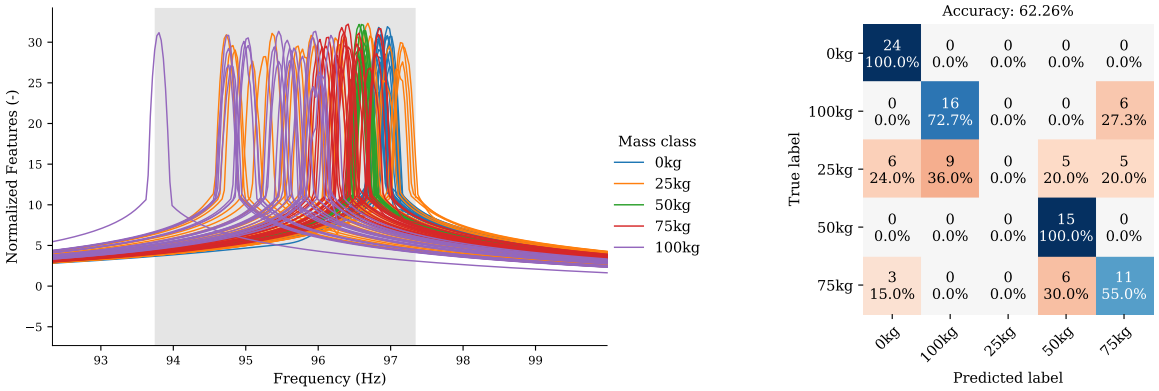


(d) Sensor 7

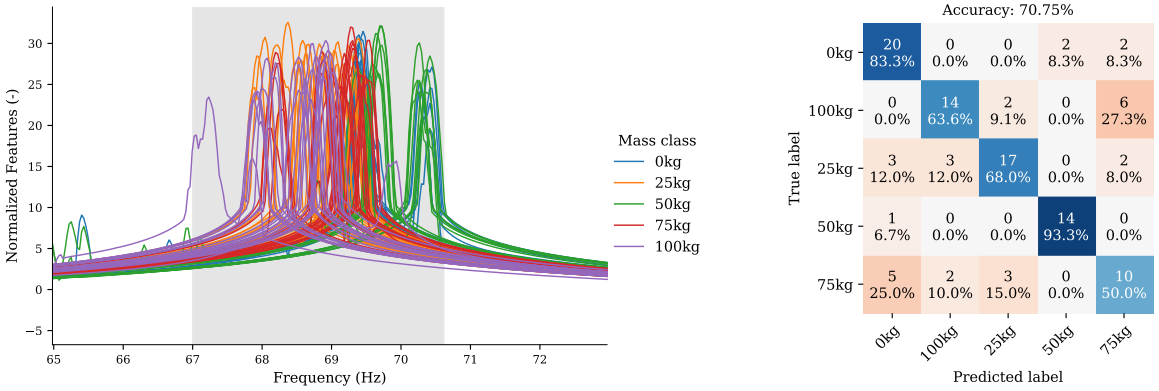
Figure H.2: Support vector machine classification using the spectral lines of a power spectra as features. The selected spectral lines that are used as features are highlighted and correspond to frequencies around the peak in the spectrum. The features are selected using a variance threshold of 5% of the maximum variance in the training dataset. The shown samples form the testing dataset for which the confusion matrix is shown in the figure on the right.



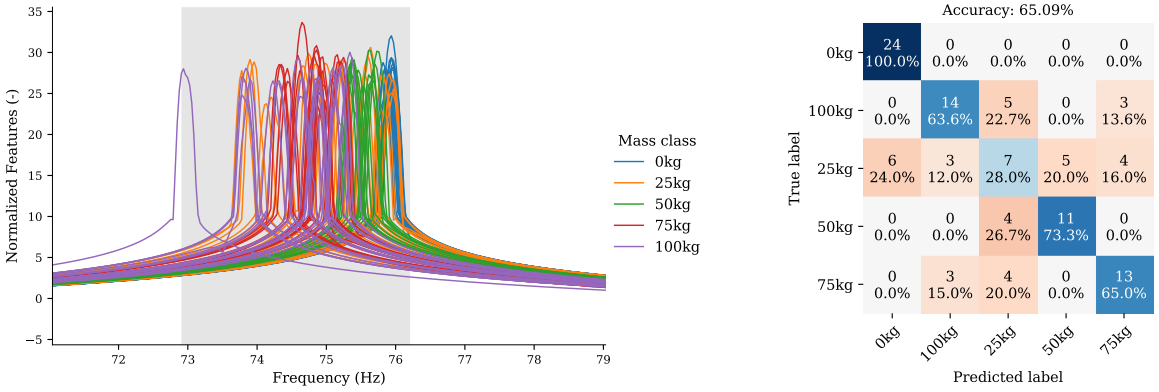
(a) Sensor 8



(b) Sensor 10

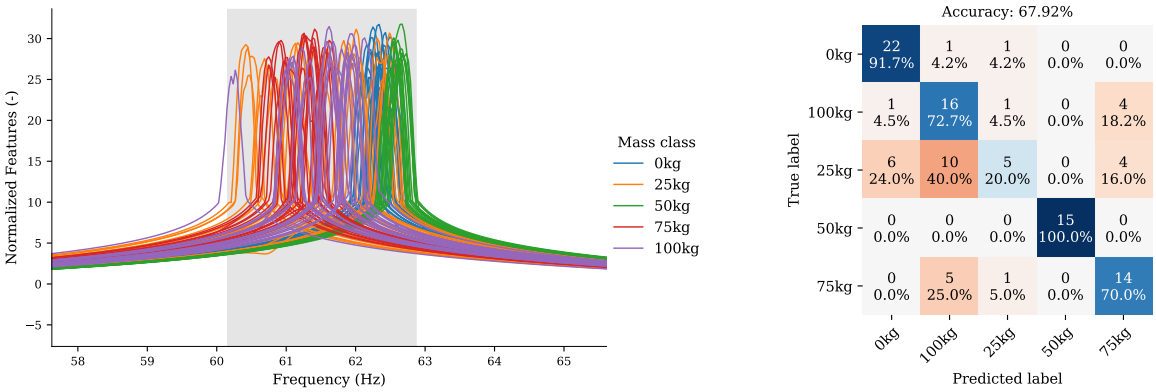


(c) Sensor 12

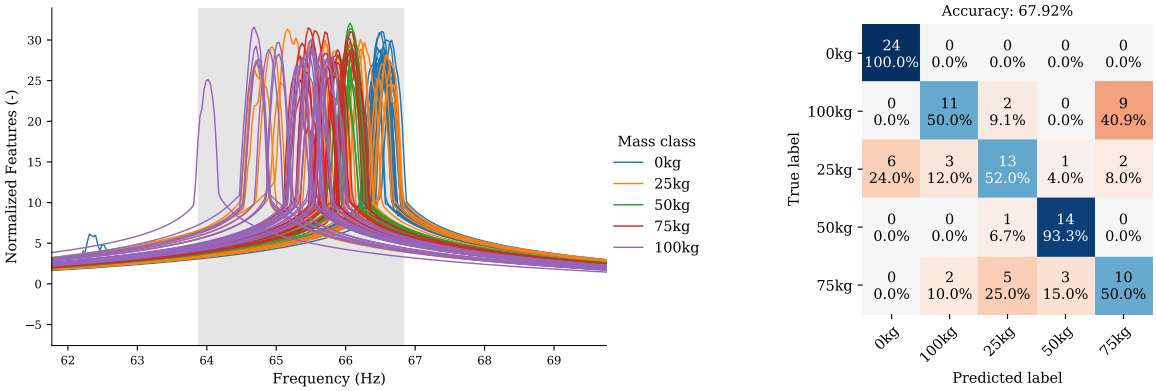


(d) Sensor 13

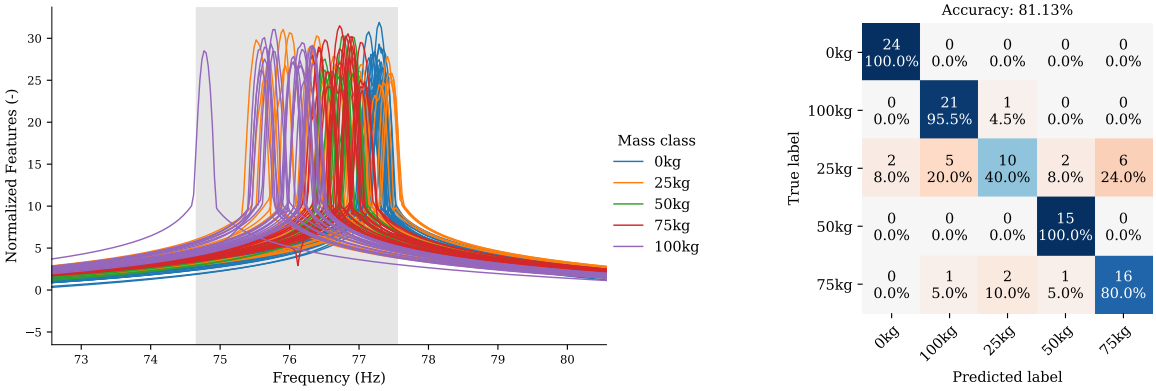
Figure H.3: Support vector machine classification using the spectral lines of a power spectra as features. The selected spectral lines that are used as features are highlighted and correspond to frequencies around the peak in the spectrum. The features are selected using a variance threshold of 5% of the maximum variance in the training dataset. The shown samples form the testing dataset for which the confusion matrix is shown in the figure on the right.



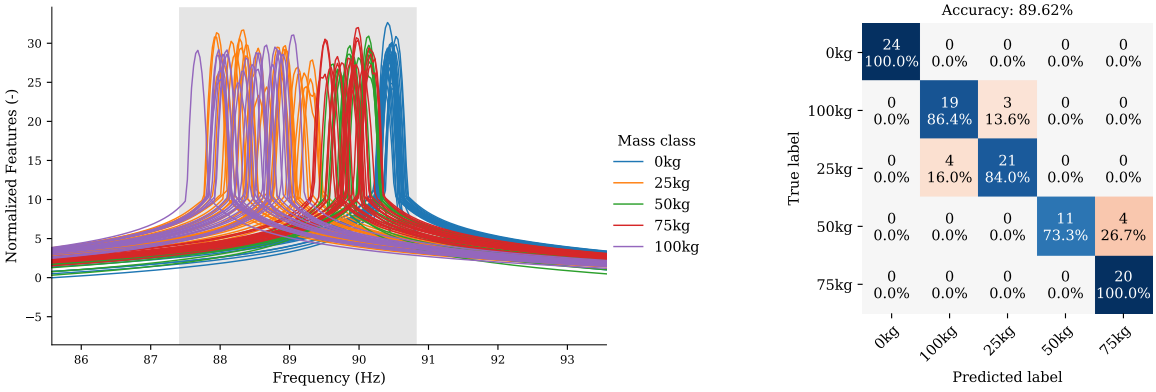
(a) Sensor 15



(b) Sensor 16

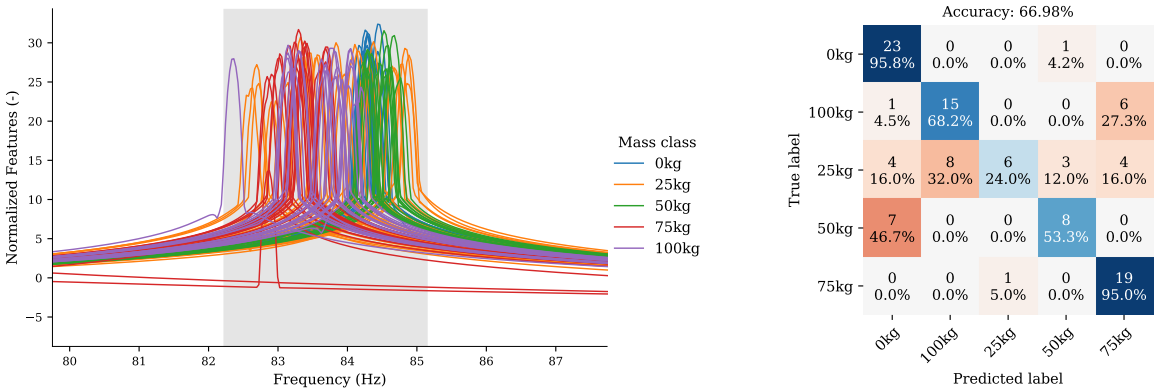


(c) Sensor 17

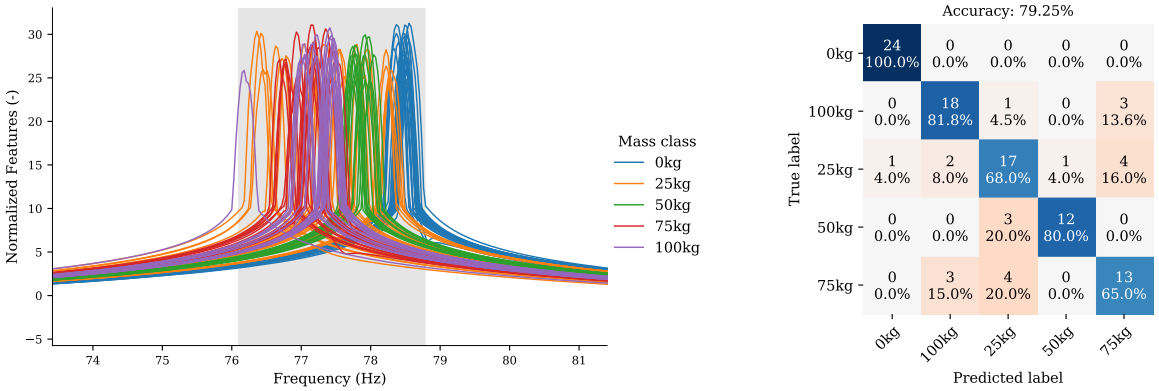


(d) Sensor 18

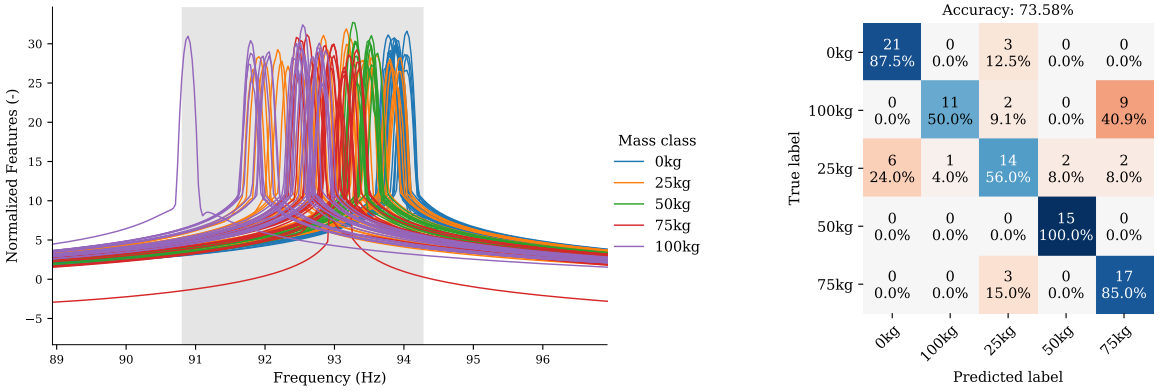
Figure H.4: Support vector machine classification using the spectral lines of a power spectra as features. The selected spectral lines that are used as features are highlighted and correspond to frequencies around the peak in the spectrum. The features are selected using a variance threshold of 5% of the maximum variance in the training dataset. The shown samples form the testing dataset for which the confusion matrix is shown in the figure on the right.



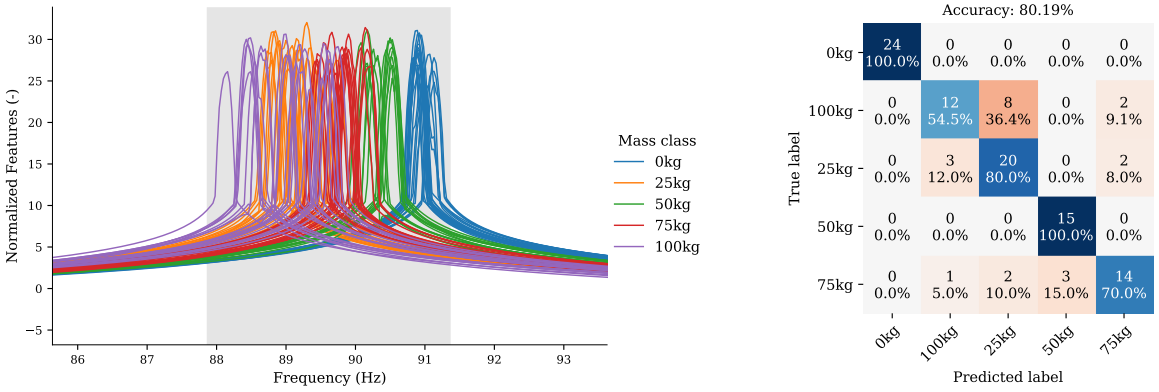
(a) Sensor 19



(b) Sensor 23

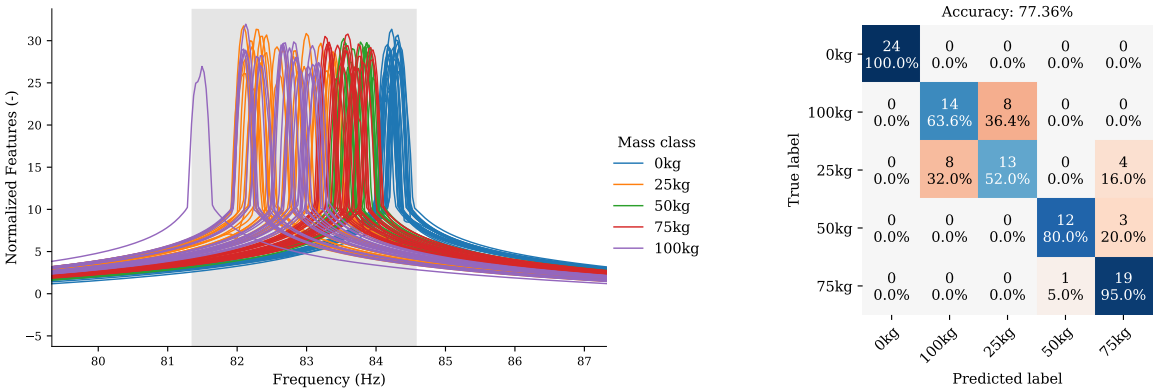


(c) Sensor 26



(d) Sensor 28

Figure H.5: Support vector machine classification using the spectral lines of a power spectra as features. The selected spectral lines that are used as features are highlighted and correspond to frequencies around the peak in the spectrum. The features are selected using a variance threshold of 5% of the maximum variance in the training dataset. The shown samples form the testing dataset for which the confusion matrix is shown in the figure on the right.



(a) Sensor 31

Figure H.6: Support vector machine classification using the spectral lines of a power spectra as features. The selected spectral lines that are used as features are highlighted and correspond to frequencies around the peak in the spectrum. The features are selected using a variance threshold of 5% of the maximum variance in the training dataset. The shown samples form the testing dataset for which the confusion matrix is shown in the figure on the right.

Bibliography

[1] Randall J. Allemang. The modal assurance criterion (mac): Twenty years of use and abuse. *Sound and Vibration*, 37(8):14–23, 2003.

[2] Julius S. Bendat and Allan G. Piersol. *Random Data: Analysis and Measurement Procedures*. John Wiley & Sons, 2010. ISBN 0470248777.

[3] Anders Brandt. *Noise and Vibration Analysis*. John Wiley and Sons Ltd, 2011. ISBN 0470746440.

[4] John Brewer. Kronecker products and matrix calculus in system theory. *IEEE Transactions on circuits and systems*, 25(9):772–781, 1978.

[5] Rune Brincker and Lingmi Zhang. Frequency domain decomposition revisited. In *Proceedings of the 3rd International Operational Modal Analysis conference - IOMAC*, pages 615–626, 2009.

[6] Rune Brincker, C. E. Ventura, and Palle Andersen. Damping estimation by frequency domain decomposition. In *19th International Modal Analysis Conference*, 2001.

[7] Rune Brincker, Lingmi Zhang, and Palle Andersen. Modal identification of output-only systems using frequency domain decomposition. *Smart Materials and Structures*, 10:441–445, 2001. ISSN 0964-1726. doi: 10.1088/0964-1726/10/3/303.

[8] Rune Brincker, Palle Andersen, and Niels-Jørgen Jacobsen. Automated frequency domain decomposition for operational modal analysis. In *Proceedings of The 25th International Modal Analysis Conference (IMAC)*. Society for Experimental Mechanics, 2007.

[9] Thomas Burdenski. Evaluating univariate, bivariate, and multivariate normality using graphical procedures. *Multiple Linear Regression Viewpoints*, pages 15–28, 2000.

[10] Ray W. Clough and Joseph Penzien. *Dynamics of Structures*. McGraw-Hill College, 3rd edition, 2003. ISBN 978-0070113923.

[11] P. Cornwell, C.R. Farrar, S.W. Doebling, and H. Sohn. Environmental Variability of Modal Properties. *Experimental Techniques*, 23(6):45–48, nov 1999. doi: 10.1111/j.1747-1567.1999.tb01320.x.

[12] Thomas M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, EC-14(3):326–334, 1965. doi: 10.1109/pgec.1965.264137.

[13] E. J. Cross, G. Manson, K. Worden, and S. G. Pierce. Features for damage detection with insensitivity to environmental and operational variations. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 468(2148):4098–4122, oct 2012. doi: 10.1098/rspa.2012.0031.

[14] A. Deraemaeker and K. Worden. A comparison of linear approaches to filter out environmental effects in structural health monitoring. *Mechanical Systems and Signal Processing*, 105:1–15, may 2018. doi: 10.1016/j.ymssp.2017.11.045.

[15] Arnaud Deraemaeker and André Preumont. Vibration based damage detection using large array sensors and spatial filters. *Mechanical Systems and Signal Processing*, 20(7):1615–1630, oct 2006. doi: 10.1016/j.ymssp.2005.02.010.

[16] YouLiang Ding and AiQun Li. Temperature-induced variations of measured modal frequencies of steel box girder for a long-span suspension bridge. *International Journal of Steel Structures*, 11(2):145–155, jun 2011. doi: 10.1007/s13296-011-2004-4.

[17] S. W. Doebling, C. R. Farrar, and M. B. Prime. A summary review of vibration-based damage identification methods. *The Shock and Vibration Digest*, 30(2):91–105, mar 1998. doi: 10.1177/058310249803000201.

[18] Wei Fan and Pizhong Qiao. Vibration-based damage identification methods: A review and comparative study. *Structural Health Monitoring: An International Journal*, 10(1):83–111, apr 2010. doi: 10.1177/1475921710365419.

[19] C. R. Farrar and K. Worden. *Structural Health Monitoring: A Machine Learning Perspective*. John Wiley & Sons, 2012. ISBN 1119994330.

[20] Charles R Farrar and David A Jauregui. Comparative study of damage identification algorithms applied to a bridge: I. experiment. *Smart Materials and Structures*, 7(5):704–719, oct 1998. doi: 10.1088/0964-1726/7/5/013.

[21] Charles R Farrar, Scott W Doebling, Phillip J Cornwell, and Erik G Straser. Variability of modal parameters measured on the alamosa canyon bridge. In *Proceedings of SPIE - The International Society for Optical Engineering*, 1996.

[22] Charles R. Farrar, Phillip J. Cornwell, Scott W. Doebling, and Michael B. Prime. Structural health monitoring studies of the alamosa canyon and i-40 bridges. Technical report, Los Alamos National Laboratory, jul 2000.

[23] Qian Geng, Huan Li, and Yueming Li. Dynamic and acoustic response of a clamped rectangular plate in thermal environments: Experiment and numerical simulation. *The Journal of the Acoustical Society of America*, 135(5):2674–2682, may 2014. doi: 10.1121/1.4870483.

[24] Alexander Graham. *Kronecker Products and Matrix Calculus: With Applications (Mathematics and its Applications)*. Ellis Horwood Limited, 1981. ISBN 0-85312-427-2.

[25] X. G. Hua, Y. Q. Ni, J. M. Ko, and K. Y. Wong. Modeling of temperature–frequency correlation using combined principal component analysis and support vector regression technique. *Journal of Computing in Civil Engineering*, 21(2):122–135, mar 2007. doi: 10.1061/(asce)0887-3801(2007)21:2(122).

[26] Niels-Jørgen Jacobsen, Palle Andersen, and Rune Brincker. Using enhanced frequency domain decomposition as a robust technique to harmonic excitation in operational modal analysis. In *Proceedings of ISMA2006: international conference on noise & vibration engineering*, pages 18–20. Belgium Leuven, 2006.

[27] I. T. Jolliffe. *Principal Component Analysis*, chapter 7, pages 150–165. Springer, 2nd edition, 2002. ISBN 0387954422.

[28] Chul-Young Kim, Dae-Sung Jung, Nam-Sik Kim, Soon-Duck Kwon, and Maria Q. Feng. Effect of vehicle weight on natural frequencies of bridges measured from traffic-induced vibration. *Earthquake Engineering and Engineering Vibration*, 2(1):109–115, jun 2003. doi: 10.1007/bf02857543.

[29] J.M. Ko and Y.Q. Ni. Technology developments in structural health monitoring of large-scale bridges. *Engineering Structures*, 27(12):1715–1725, oct 2005. doi: 10.1016/j.engstruct.2005.02.021.

[30] Jyrki Kullaa. Vibration-based structural health monitoring under variable environmental or operational conditions. In *New Trends in Vibration Based Structural Health Monitoring*, pages 107–181. Springer Vienna, 2010. doi: 10.1007/978-3-7091-0399-9_4.

[31] Hui Li, Shunlong Li, Jinping Ou, and Hongwei Li. Modal identification of bridges under varying environmental conditions: Temperature and wind effects. *Structural Control and Health Monitoring*, pages n/a–n/a, 2009. doi: 10.1002/stc.319.

[32] Marko Milosevic. Structural health monitoring of the zwartewaterbrug bridge. Master's thesis, Delft University of Technology, 2018.

[33] Y. Q. Ni, H. F. Zhou, and J. M. Ko. Generalization capability of neural network models for temperature-frequency correlation using monitoring data. *Journal of Structural Engineering*, 135(10):1290–1300, oct 2009. doi: 10.1061/(asce)st.1943-541x.0000050.

[34] Y.Q. Ni, X.G. Hua, K.Q. Fan, and J.M. Ko. Correlating modal properties with temperature using long-term monitoring data and support vector machine technique. *Engineering Structures*, 27(12):1762–1773, oct 2005. doi: 10.1016/j.engstruct.2005.02.020.

[35] Felix Boakye Oppong and Senyo Yao Agbedra. Assessing univariate and multivariate normality, a guide for non-statisticians. *Math. Theory Modeling*, 6(2):26–33, 2016.

[36] Bart Peeters. *System Identification and Damage Detection in Civil Engineering*. PhD thesis, Katholieke Universiteit te Leuven, 2000.

[37] Bart Peeters and Guido De Roeck. One-year monitoring of the z24-bridge: environmental effects versus damage events. *Earthquake Engineering & Structural Dynamics*, 30(2):149–171, 2001. doi: 10.1002/1096-9845(200102)30:2<149::aid-eqe1>3.0.co;2-z.

[38] Bart Peeters, Johan Maeck, and Guido De Roeck. Vibration-based damage detection in civil engineering: excitation sources and temperature effects. *Smart Materials and Structures*, 10(3):518–527, jun 2001. doi: 10.1088/0964-1726/10/3/314.

[39] Allyn W. Phillips, Randall J. Allemang, and William A. Fladung. Complex mode indicator function (cmif) as a parameter estimation method. In *Proceedings of the International Modal Analysis Conference - IMAC*, volume 1, pages 705–710, 1998.

[40] D. Poole. *Linear Algebra A Modern Introduction*. Cengage, 3rd edition, 2011. ISBN 9780538735445.

[41] R. Ruotolo and C. Surace. Using SVD to detect damage in structures with different operational conditions. *Journal of Sound and Vibration*, 226(3):425–439, sep 1999. doi: 10.1006/jsvi.1999.2305.

[42] Hoon Sohn. Effects of environmental and operational variability on structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851):539–560, feb 2007. doi: 10.1098/rsta.2006.1935.

[43] Hoon Sohn, Mark Dzwonczyk, Erik G. Straser, Kincho H. Law, Teresa H. Meng, and Anne S. Kiremidjian. Adaptive modeling of environmental effects in modal parameters for damage detection in civil structures. In S.-C. Liu, editor, *Smart Structures and Materials 1998: Smart Systems for Bridges, Structures, and Highways*. SPIE, jun 1998.

[44] Hoon Sohn, Mark Dzwonczyk, Erik G. Straser, Anne S. Kiremidjian, Kincho H. Law, and Teresa Meng. An experimental study of temperature effect on modal parameters of the alamosa canyon bridge. *Earthquake Engineering & Structural Dynamics*, 28(8):879–897, aug 1999. doi: 10.1002/(sici)1096-9845(199908)28:8<879::aid-eqe845>3.0.co;2-v.

[45] Hoon Sohn, Keith Worden, and Charles R. Farrar. Statistical damage classification under changing environmental and operational conditions. *Journal of Intelligent Material Systems and Structures*, 13(9):561–574, sep 2002. doi: 10.1106/104538902030904.

[46] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern Recognition*. Elsevier LTD, Oxford, 4th edition, 2008. ISBN 1597492728.

[47] Jake Van der Plas. *Python Data Science Handbook*. O'Reilly UK Ltd., 2016. ISBN 1491912057.

[48] S. Vanlanduit, E. Parloo, B. Cauberghe, P. Guillaume, and P. Verboven. A robust singular value decomposition for damage detection under changing operating conditions and structural uncertainties. *Journal of Sound and Vibration*, 284(3-5):1033–1050, jun 2005. doi: 10.1016/j.jsv.2004.07.016.

[49] Benedikt Weber. Dynamic properties of footbridges: Influence of asphalt pavement and support conditions. *MATEC Web of Conferences*, 24:01004, 2015. doi: 10.1051/matecconf/20152401004.

[50] Michael G Wood. *Damage analysis of bridge structures using vibrational techniques*. PhD thesis, Aston University, 1992.

[51] K. Worden, G. Manson, and N. R. J. Fieller. Damage detection using outlier analaysis. *Journal of Sound and Vibration*, 229(3):647–667, jan 2000. doi: 10.1006/jsvi.1999.2514.

[52] Keith Worden and Graeme Manson. The application of machine learning to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851):515–537, dec 2006. doi: 10.1098/rsta.2006.1938.

[53] Keith Worden, David W. Allen, Hoon Sohn, and Charles R. Farrar. Damage detection in mechanical structures using extreme value statistics. In Vittal S. Rao, editor, *Smart Structures and Materials 2002: Modeling, Signal Processing, and Control*, jul 2002. doi: 10.1117/12.475226.

[54] Yong Xia, You-Lin Xu, Ze-Long Wei, Hong-Ping Zhu, and Xiao-Qing Zhou. Variation of structural vibration characteristics versus non-uniform temperature distribution. *Engineering Structures*, 33(1):146–153, jan 2011. doi: 10.1016/j.engstruct.2010.09.027.

[55] Yong Xia, Bo Chen, Shun Weng, Yi-Qing Ni, and You-Lin Xu. Temperature effect on vibration properties of civil structures: a literature review and case studies. *Journal of Civil Structural Health Monitoring*, 2 (1):29–46, jan 2012. doi: 10.1007/s13349-011-0015-7.

[56] A.-M. Yan, G. Kerschen, P. De Boe, and J.-C. Golinval. Structural damage diagnosis under varying environmental conditions—part i: A linear analysis. *Mechanical Systems and Signal Processing*, 19(4):847–864, jul 2005. doi: 10.1016/j.ymssp.2004.12.002.

[57] A.-M. Yan, G. Kerschen, P. De Boe, and J.-C. Golinval. Structural damage diagnosis under varying environmental conditions—part II: local PCA for non-linear cases. *Mechanical Systems and Signal Processing*, 19(4):865–880, jul 2005. doi: 10.1016/j.ymssp.2004.12.003.

[58] Q. W. Zhang, L. C. Fan, and W. C. Yuan. Traffic-induced variability in dynamic properties of cable-stayed bridge. *Earthquake Engineering & Structural Dynamics*, 31(11):2015–2021, 2002. doi: 10.1002/eqe.204.

[59] Guang-Dong Zhou and Ting-Hua Yi. A summary review of correlations between temperatures and vibration properties of long-span bridges. *Mathematical Problems in Engineering*, 2014:1–19, 2014. doi: 10.1155/2014/638209.

[60] H. F. Zhou, Y. Q. Ni, and J. M. Ko. Eliminating temperature effect in vibration-based structural damage detection. *Journal of Engineering Mechanics*, 137(12):785–796, dec 2011. doi: 10.1061/(asce)em.1943-7889.0000273.

[61] H.F. Zhou, Y.Q. Ni, and J.M. Ko. Constructing input to neural networks for modeling temperature-caused modal variability: Mean temperatures, effective temperatures, and principal components of temperatures. *Engineering Structures*, 32(6):1747–1759, jun 2010. doi: 10.1016/j.engstruct.2010.02.026.